Quantum computation using NMR

Kavita Dorai‡, T. S. Mahesh‡, Arvind‡ and Anil Kumar‡,*,#

‡Department of Physics, Indian Institute of Science, Bangalore 560 012, India
‡Department of Physics, Guru Nanak Dev University, Amritsar 143 005, India
*Sophisticated Instruments Facility, Indian Institute of Science, Bangalore 560 012, India

This article reviews recent work done by us in some initial steps towards the implementation of quantum computation using liquid state NMR. We describe how special kinds of states required for such computation (called pseudo-pure states) can be created from a thermal ensemble of spins. We demonstrate the implementation of several quantum logic gates through one- and two-dimensional NMR methods, using transition- and spin-selective pulses. Finally, we discuss the implementation of the Deutsch–Jozsa algorithm using NMR.

All present-day computations use two-state binary logic and have led to a large revolution in data processing and manipulation. However, several scientists have wondered if quantum mechanical systems could provide a new paradigm for computation. Feynmann in particular, hypothesized that it might be possible to simulate quantum evolution efficiently, provided the simulator was itself quantum-mechanical in nature.

The question that arises is: can real quantum systems be used to build computers that operate in the quantum-mechanical regime, and how much more computing power would such devices be able to achieve? These are some interesting issues being addressed by researchers today, and this fusion of ideas from quantum physics and information theory has led to exciting new developments in quantum cryptography, teleportation, error correction, and quantum computation.

It was proved early on, that traditional Boolean logic gates can be implemented using a set of modified ‘reversible’ gates, and that one can find a minimal set of such gates that are sufficient for computation. Furthermore, researchers in the early 1980s made a fundamental connection between quantum mechanics and reversible computation by proposing that reversible Boolean computation can be simulated by the time evolution of a quantum system, which is a unitary reversible dynamic. Logical operations in quantum computation are implemented on quantum bits (qubits), the basic units of quantum information. A qubit can be visualized as the states of a two-level quantum system, like the two spin states of a spin-1/2 particle or the two different polarization states of a single photon. The realization that, the two eigenstates labelled by |0⟩ and |1⟩ can be mapped onto logical 0 and 1, leads to the possibility of the quantum-mechanical implementation of logic gates and circuits. Unlike the classical bit which can exist only in two states, the permitted states for a qubit (cosθ |0⟩ + sinθ e⁻iφ |1⟩), span a 2D complex vector space. A state for n qubits can be represented by a 2ⁿ-dimensional complex vector. While all classical computation can be performed using the mapping between eigenstates and logical states, the fact that a qubit can exist in a general coherent superposition of the eigenstates, leads to new possibilities for computation. What is intrinsically different about a quantum computation? The answer lies in the fact that it exploits inherently quantum features like quantum superposition and entanglement to solve problems hitherto deemed intractable, on any classical computer. If two qubits are in a state such as |00⟩ + |11⟩, which is not resolvable into the tensor product of the states of the individual qubits, the qubits are said to be entangled. Neither qubit by itself has a definite state, in contrast to a classical system which can be completely resolved into the states of each part of the system. The existence of such entangled states is fundamental to the quantum world and leads to counter-intuitive phenomena like the violation of Bell’s inequalities. It is interesting that such intriguing states have now found an application in computation and information processing, to reduce the level of complexity of computational tasks.

A measure of computational complexity is how the number of steps required for the computation (denoted s) evolves mathematically as a function of the size of the problem (denoted L). If s is a polynomial function of L, the problem is tractable; if s rises exponentially with L, the problem is thought to be intractable. An example of a computationally hard problem is that of the

![Figure 1](image_url)

Figure 1. Units of information: bits and qubits. A classical bit can only take the values 0 and 1, whereas a qubit can exist in any coherent superposition of the two.

For correspondence. (e-mail: anilmr@physics.iisc.ernet.in)
factorization of a number, i.e., to find all the prime factors of a given large integer. Classically, this is known to be computationally intractable since the time taken by the best-known classical algorithm grows exponentially with the size of the input number. This is the security on which most cryptographic systems rely on today. A quantum computer can factor large integers in a time which is polynomial in the logarithm of the best classical time. This brings the non-tractable factoring problem to the domain of tractability and illustrates the true power of quantum computing. In quantum computation, the computation itself can be thought of as a set of unitary transformations $U_i$ acting on a set of input qubits. The main task is to identify suitable quantum systems, prepare them in pure initial states and physically achieve Hamiltonians that generate these unitary transformations, where $U_i = e^{iH_0}$. A massive parallelism is achieved in quantum computation, since one is able to perform the computation on all the inputs in one go, by preparing the state of the qubits in a quantum superposition of all the possible classical inputs. However, not all tasks can be speeded up by a uniform amount on a quantum computer—problems like factoring and the simulation of quantum dynamics gain an exponential speed-up, while the database search problem is speeded up polynomially. On the other hand, many problems like inserting an element in an ordered list do not get speeded up at all. The identification of classes of problems that can be solved more efficiently by a quantum computer opens up newer avenues for research. Three path-breaking quantum algorithms in this direction are, the Deutsch–Jozsa (DJ) algorithm to distinguish between two classes of mathematical functions, Shor’s quantum factoring algorithm, and Grover’s algorithm to rapidly search a database.

The choice of physical systems to build elements of quantum hardware explored thus far, ranges from optical photons, cavity QED, quantum dots, trapped ions, to nuclear spins. As a candidate for quantum computing, NMR is attractive because of the long coherence times exhibited by the spins, and also due to the complexity of logical operations that can be executed on modern spectrometers. A molecule with $N$ spin $1/2$ nuclei can be visualized as an $N$-bit quantum computer, provided the spins are able to interact, one can manipulate their states in a desired fashion and there is a well-defined method of reading out the result of the computation. In the case of liquid-state NMR, one has an ensemble of such $N$-bit computers with $N$ determined by the number of manipulable, interacting spins. Quantum computing is usually performed on pure input states, whereas nuclear spins at thermal equilibrium are in a statistical mixture of pure states, with the signal being an average over all the individual states of the ensemble. It has been shown recently that it is possible to perform quantum computations using special kinds of mixed state ensembles as well (ensembles in pseudo-pure states), and the challenge now is to explore the possibility of carrying out bulk quantum computing using NMR-based information processors.

Previous workers in the field have employed various NMR methods like non-selective pulses, rf gradients, coherence transfer via $J$-coupling and simultaneous multi-site excitation to create pseudo-pure states, construct universal quantum logic gates, and implement quantum algorithms for two- and three-qubit systems. In this research account, we demonstrate the utility of transition- and spin-selective pulses in achieving quantum computing using NMR. We exploit the non-commuting nature of operations on connected transitions to prepare the spin system in a pseudo-pure state and execute different logical operations simultaneously. Different quantum logic gates have been demonstrated using one- and two-dimensional (1D and 2D respectively) NMR techniques. As an illustration of these ideas, we have implemented the DJ quantum algorithm using 1D and 2D schemes. It turns out that a modified version of the DJ algorithm, which uses lesser qubits, brings out its subtle aspects more clearly. The implementation of the algorithm for up to one and two bits does not involve the manipulation of quantum entangled states, and thus finds a classical explanation. It is only for the three-bit case that entangling transformations become essential to solve the problem.

Experiments have been performed at 300 K on Bruker AMX-400 and DRX-500 spectrometers operating at $^1$H resonance frequencies of 400 and 500 MHz, respectively. Two, three and four-spin systems have been chosen as qubits for the computation. The $T_1$ relaxation times in the molecules used, is of the order of a few seconds (3.4–4.6 s), whereas $T_2$ relaxation occurs within an interval of about 1 second. Selective excitation has been achieved using low power, long duration rectangular-shape and Gaussian-shaped pulses. The lengths of these pulses are tailored to achieve sufficient selectivity in the frequency domain without perturbing the nearest line and hence depends on the magnitude of the smallest $J$ coupling present. The duration of the pulses applied varies from 100 to 263 ms (for $J$ couplings of 9.6 to 3.8 Hz). For small computations, such as the ones performed here, drastic decoherence or dephasing does not occur during the duration of these selective pulses. However, the deleterious effects of such selective pulses must be considered and compensated for, whenever larger computations are attempted.

Creation of pseudo-pure states

Quantum algorithms are usually formulated in terms of pure quantum-mechanical states. The purity of a state...
can be best visualized in the ensemble interpretation. A pure state corresponds to an ensemble with each member being described by the same state vector. A mixed ensemble on the other hand, cannot be represented by a single vector in Hilbert space. The density matrix approach has been devised to handle this physical situation, whereby a density operator can be assigned to pure as well as mixed ensembles. Given a pure ensemble with the state vector $|\psi\rangle$, the corresponding density operator $\rho = |\psi\rangle\langle\psi|$ contains all the relevant information about the state. For a mixed ensemble, with say a fraction $f$ members in a state $|\psi\rangle$ and $1-f$ members in state $|\phi\rangle$, the density operator is a statistical mixture of operators corresponding to both the state vectors and is given by $\rho = f|\psi\rangle\langle\psi| + (1-f)|\phi\rangle\langle\phi|$. Therefore, we can always associate a density operator with a general quantum-mechanical system drawn from a general (pure or mixed) ensemble. When the system is in a pure state (drawn from a pure ensemble), it is also possible to assign a state vector to it. A necessary and sufficient condition for the density matrix corresponding to a pure state is $\rho^2 = \rho$. A quantum computer comprising these spins is therefore an ensemble of independent quantum computers, doing the same computation. The initial state of each quantum computer (spin) is randomly determined and the only accessible measurement is an average of all the computers’ registers (states of the spins). The result of a computation attempted on this mixed ensemble might get erased by the averaging process over the ensemble. The problem of working with such mixed initial states has been circumvented by creating within the overall density matrix of the system, a subsystem that behaves exactly like a pure state. Such states are termed ‘pseudo-pure’ states and different schemes to create pseudo-pure states in NMR have been proposed and experimentally implemented.39,40,43–45,63

The preparation of pseudo-pure states from a thermal ensemble involves isolating a sub-manifold of spin states whose transformation properties are similar to those of a pure state. In effect, the excess (or deficit) population of a state among a uniformly populated background of levels, causes it to behave like a pure state. Consider a system of $n$ spins with a $2^n$-dimensional Hilbert space. If we now have a state for which all levels except one are equally populated and we are interested in measurements of those operators for which the uniform background (density matrix is a multiple of identity) does not contribute, then for such measurements this state behaves like a pure state, corresponding to the level with excess population. Such states will henceforth be referred to as pseudo-pure states.

At high temperature, the density matrix of an $N$-spin system is given by

$$\hat{\sigma} = \frac{1}{2^N}(\hat{I} + \beta \hat{\sigma}_\Lambda),$$

which consists of a background part proportional to the identity $\hat{I}$ and a traceless, deviation density matrix $\hat{\sigma}_\Lambda$. The factor $\beta = \hbar \omega / kT$ is very small ($=10^{-6}$) for nuclear spins at room temperature. The identity part of the density matrix does not evolve under rf pulses and does not contribute to the measured NMR signal. After manipulation by rf pulses and/or gradients, the deviation part of the density matrix can be made to behave like a pure quantum state. To illustrate this point, consider a system of two spin-1/2 particles, with the thermal equilibrium density matrix being described by

$$\sigma_{eq} = \frac{1}{4} \left[ \begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{array} \right].$$

A small fraction of the spins are in the state describable by $(\sigma_x)_{eq} = I_{z1} + I_{z2}$; the subscripts 1, 2 label the spins and $I_{z\alpha}$, etc. are the usual product operator representations of angular momentum vectors.65 The rest of the spins are in the uniformly populated background and do not contribute to the signal.

Consider now the matrix representation of a traceless deviation density matrix corresponding to the state $I_{z1} + I_{z2} + 2I_{z1}I_{z2}$ is

$$\langle \sigma_\Lambda \rangle_{\text{pure}} = \frac{3}{2} \begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & -1/2 & 0 & 0 \\ 0 & 0 & -1/2 & 0 \\ 0 & 0 & 0 & -1/2 \end{array}.$$  

This matrix can be visualized as

$$\langle \sigma_\Lambda \rangle_{\text{pure}} = \begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{2} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}$$

where the identity term does not evolve under rf pulses and is ignored in NMR experiments. This combination of product operators, $I_{z1} + I_{z2} + 2I_{z1}I_{z2}$, hence represents a pseudo-pure state. It is to be noted that the equilibrium density matrix $\sigma_{eq}$ (eq. (2)) cannot be resolved into two such matrices (one of them being identity) and hence the thermal equilibrium state is not a pseudo-pure state.60

Techniques to experimentally prepare pseudo-pure states can be broadly classified as (i) spatial averaging methods which rely on rf gradients for their implemen-
RESEARCH ACCOUNT

tation\(^{40}\), (ii) temporal averaging, where the result of different experiments performed sequentially is time-averaged\(^{38}\) and (iii) logical labelling methods which use manipulations by rf pulses to re-label the states of the spins\(^{39}\).

**Spatially averaged pseudo-pure states**

A spatially averaged pseudo-pure state can be prepared by a judicious combination of rf gradients and pulses with different flip angles as shown by Cory et al.\(^{30}\). It is termed spatial since unitary transformations are applied as a function of a spatial degree of freedom. For example, the scheme \([\frac{\pi}{2}]_{12} -[\Delta]_{2} -[\frac{\pi}{2}]_{1} -[\frac{\pi}{2}]_{2} -[\Delta]_{2}\), where 1, 2 are the spin labels, \(J\) the coupling constant, and \(\Delta\), a gradient in the \(z\) direction, leads to a final pseudo-pure density matrix represented by \(\frac{1}{2}[I_{1z} + I_{2z} + 2I_{1z}I_{2z}]\).

**Temporally averaged pseudo-pure states**

The temporal averaging technique uses several experiments with different preparation steps, the results of which are averaged to give a pseudo-pure state as demonstrated by Knill et al.\(^{41}\). The unitary transformations are performed sequentially in time. As an example, the average of three different experiments \(P_n\) yields a temporally-averaged pseudo-pure state whose density matrix is given by

\[
\sigma_{\text{avg}} = \frac{1}{3} \sum_{n=1}^{3} P_n \sigma P_n^T.
\]  

(5)

The unitary transformations corresponding to the three experiments are given by

\[
P_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad P_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}
\]

and

\[
P_3 = P_2^T.
\]  

(6)

**Logically labelled pseudo-pure states**

The logical labelling technique to create pseudo-pure states is categorized by the fact that unitary transformations are used to redistribute the populations of states, such that an effective pure state is obtained in the sub-manifold of qubits (spins) to be used for computation, and ancillary qubits are used as ‘labels’. While the concept underlying the logical labelling method of pseudo-pure state creation has been delineated by Gershenfeld and Chuang\(^{39}\) and Chuang et al.\(^{32}\), there have been very few experimental implementations of such an elegant technique\(^{45}\). We have designed novel pulse schemes using transition-selective pulses to create such logically labelled pseudo-pure states\(^{33}\). Consider a three spin-1/2 system (AMX), with the energy levels labelled as in Figure 2. The selective inversion of two unconnected single-quantum transitions of the A spin (\(|001\rangle \rightarrow |101\rangle\) and \(|010\rangle \rightarrow |110\rangle\)) would lead to the creation of a logically labelled pseudo-pure state, with A being the ‘label qubit’ and M, X being the ‘work qubits’ available for computation. The first four eigenstates (labelled by the first spin being in the |0) state) now form a manifold that corresponds to a two-qubit pseudo-pure state, while the last four (labelled by the first spin being in the |1) state) form a different manifold that corresponds to another two-qubit pseudo-pure state. The creation of a pseudo-pure state by this method leads to relative population differences of

\[
\begin{bmatrix}
000 & 001 & 010 & 011 & 100 & 101 & 110 & 111 \\
3/2 & -1/2 & -1/2 & -1/2 & 1/2 & 1/2 & 1/2 & -3/2
\end{bmatrix}
\]  

(7)

\[\]
and has been implemented in NMR by a selective $\pi$ pulse on a single transition\textsuperscript{40}. The action of the quantum XOR is an evolution under a Hamiltonian that achieves a mapping of two-qubit basis states according to the XOR truth table, i.e., $\{\langle 00 | \rightarrow |00\rangle, |01 | \rightarrow |01\rangle, |10 | \rightarrow |11\rangle, |11 \rangle \rightarrow |10\rangle \}$. It has been proved that the reversible quantum XOR gate, supplemented by a set of general one-qubit quantum gates, is sufficient to perform any arbitrary quantum computation\textsuperscript{40}. Various other gates which combine many logical operations might be useful in reducing the number of pulses when larger computations are attempted. We detail the design and experimental implementation of such gates here, using 1D and 2D NMR methods; borrowing from Lewis Carroll, we call such 'many-in-one' gates Portmanteau gates\textsuperscript{66}.

In the 1D method, various transition- and spin-selective pulses have been applied on a two-spin system in equilibrium, to implement different logic gates\textsuperscript{63}. In the context of quantum computing, it was recently recognized by Madi \textit{et al.}\textsuperscript{31}, that the four stages of a quantum computation (namely initial state preparation, state labelling, computation and readout of result), can be directly related to various steps in two-dimensional NMR (namely preparation, evolution, mixing and detection).

In the two-dimensional method, the states of the ‘computation’ qubits are labelled by the individual transitions of an extra ‘observer’ qubit (instead of being identified by their energy level labels, as is usually the case in one-dimensional experiments). The 2D experiment begins with a $\pi/2$ pulse on the observer spin, which is then allowed to evolve under the free Hamiltonian for a time period $t_1$. After the $t_1$ period, a selective $\pi/2$ pulse on the observer spin restores its magnetization to $z$-direction, followed by a gradient pulse which destroys unwanted transverse magnetization, if any. The quantum logic gate is then implemented on the computation qubits and the detection of the final state is performed by a spin-selective $\pi/2$ pulse on the observer qubit (Figure 4). The positions of the different multiplet components of the observer spin 2D spectrum, represent the results of the computation.

Consider a two-spin system (AX) with each spin being a qubit, the spin A being the first qubit and the spin X, the second qubit. The eigenstates of this system can be represented by $|e_1\rangle, |e_2\rangle$, where $e_1$ and $e_2$ are 0 or 1. Various logic operations and portmanteau gates that can be implemented on this two-spin system are categorized in Table 1, along with their truth tables and corresponding unitary transformations. The NOP gate corresponds to a no operation (with no pulses being applied on the qubits).

**The SWAP operation**

The logical SWAP operation exchanges the states of a pair of qubits. This gate might be useful dur-
Table 1. Characterization of various two-bit logic operations and portmanenteau gates. The truth tables and corresponding unitary transformations are given in the different columns.

<table>
<thead>
<tr>
<th>Gate</th>
<th>In</th>
<th>Out</th>
<th>$U_f$</th>
<th>Circuit diagram</th>
</tr>
</thead>
</table>
| NOP       | 00  | 00  | \[
\begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0
\end{pmatrix}
\] | 1 |
| | 01  | 01  | \[
\begin{pmatrix}
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\] | 2 |
| | 10  | 10  | \[
\begin{pmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{pmatrix}
\] |  |
| | 11  | 11  | \[
\begin{pmatrix}
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\] |  |
| SWAP      | 00  | 00  | \[
\begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0
\end{pmatrix}
\] | 1 |
| | 01  | 10  | \[
\begin{pmatrix}
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\] | 2 |
| | 10  | 01  | \[
\begin{pmatrix}
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\] | 2 |
| | 11  | 11  | \[
\begin{pmatrix}
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\] |  |
| XOR1 + SWAP + NOT1 | 00 | 10 | \[
\begin{pmatrix}
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\] | 1 |
| | 01  | 01  | \[
\begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0
\end{pmatrix}
\] | 2 |
| | 10  | 11  | \[
\begin{pmatrix}
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\] | 2 |
| | 11  | 00  | \[
\begin{pmatrix}
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\] |  |
| XNOR2 + SWAP + NOT2 | 00 | 11 | \[
\begin{pmatrix}
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\] | 1 |
| | 01  | 01  | \[
\begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0
\end{pmatrix}
\] | 2 |
| | 10  | 00  | \[
\begin{pmatrix}
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\] | 2 |
| | 11  | 01  | \[
\begin{pmatrix}
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\] |  |
| NOT + SWAP | 00 | 11 | \[
\begin{pmatrix}
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\] | 1 |
| | 01  | 01  | \[
\begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0
\end{pmatrix}
\] | 2 |
| | 10  | 10  | \[
\begin{pmatrix}
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\] | 2 |
| | 11  | 00  | \[
\begin{pmatrix}
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\] |  |

The symbol $\oplus$ stands for 'addition modulo 2', while the bar denotes a complement. The symbol $\odot$ represents the operation NOT. The NOP stands for 'No Operation', in which no pulse is applied and the unity operator performs no operation.

Figure 4. Pulse scheme for quantum computing using two-dimensional NMR. $I_0$ is the observer qubit and $I_1, I_2, \ldots$ are the computation qubits. The gradient pulse $G_z$ eliminates unwanted transverse magnetization before the computation. A two-dimensional data set $s(t_1, t_2)$ is collected, which on double Fourier transform yields a two-dimensional spectrum $s(\omega_0, \omega_2)$. The two-dimensional spectrum correlates the input and output states of the work qubits depending on the computation performed during the mixing/computation period.

The symbol $\oplus$ stands for 'addition modulo 2', while the bar denotes a complement. The symbol $\odot$ represents the operation NOT. The NOP stands for 'No Operation', in which no pulse is applied and the unity operator performs no operation.

In the course of a computation when qubits need to be permuted$^{59}$. In spin systems where some scalar $J$ couplings are not well resolved, the logical SWAP could be used to compensate for the missing couplings$^{51}$. Madi et al.$^{54}$ have implemented the SWAP operation using an INEPT-type sequence, with non-selective rf pulses and $J$-evolution. It is interesting to note that the logical SWAP operation can be achieved by selectively interchanging the populations of the zero quantum levels. Since these levels are not connected by single-quantum transitions, the population exchange will have to be achieved indirectly. A pulse sequence to implement this uses transition selective pulses on regessively connected single-quantum transitions, after the creation of a non-equilibrium population distribution$^{67}$. The 1D implementation of the logical SWAP operation achieved by a selective manipulation of the zero-quantum is shown in Figure 5 on the two-spin system of coumarin. The 2D implementation of the SWAP gate is especially interesting and is shown in Figure 6. As discussed earlier, the 1D spectrum of homo-nuclear spins after the execution of a SWAP operation is indistinguishable from that of an equilibrium spectrum and a one-dimensional demonstration of the SWAP gate begins.
with the creation of a non-equilibrium state. The 2D method begins with both qubits in the equilibrium state and yields a spectrum characteristic of the SWAP gate.

The 1D implementation of various portmanteau gates on a thermal initial state is shown in Figure 7 for the two-spin system of coumarin. The same pulse schemes will implement the desired logic operations on other initial states (for instance, a pseudo-pure or a coherent superposition of states) as well, without requiring prior knowledge of the state of the system. The implementation of an XNOR gate (with the output on the first qubit), followed by a SWAP operation and then a NOT gate on the first qubit can be experimentally achieved by transition-selective π pulses applied consecutively on two progressively connected transitions. Reversing the order of the pulses corresponds to an XNOR + SWAP + NOT gate, with the output on the second qubit. These operations do not commute, so the order in which the pulses are applied is important and its reversal leads to different logical operations. The implementation of a NOT gate followed by a logical SWAP operation (or vice versa, since these operations commute) is achieved by a cascade of transition-selective π pulses on two progressively connected transitions. This tantamounts to a selective inversion of populations of the double-quantum levels.

The 2D implementation of these quantum logic gates on two computation qubits (using an extra observer qubit) is experimentally demonstrated on 2,3-dibromo propionic acid (Figure 8). A complete set (24) of one-to-one reversible 2-qubit gates using two-dimensional NMR on a three-spin system has been carried out.

Figure 9 shows several 3-qubit gates implemented on a 4-spin system, using selective pulses. Once again, No OPeration during computation period yields NOP gate and inverting spin 1 yields the NOT (11) gate. The more interesting ones are Toffoli gate (or AND/NAND gate) and OR/NOR gates. The operations of Toffoli and OR/NOR gates (13, 15) are respectively,

\[ |s, t, u \rangle \rightarrow |s \oplus (t \land u), t, u \rangle, \]

and

\[ |s, t, u \rangle \rightarrow |s \oplus (t \lor u), t, u \rangle, \]

where \( \oplus \) = addition modulo 2, \( \land \) = AND, \( \lor = OR \) and \( s, t, u \) are the states of the control spin 1 and the input spins \( I_2 \) and \( I_3 \). The Toffoli gate is a universal gate for reversible computation.

The 1D pulse schemes for the implementation of gates are useful and easy to implement. The advantage of the two-dimensional method is that it correlates the input and output states and thus the readout of the com-
RESEARCH ACCOUNT

Computation is graphic. However, the 2D method requires an extra qubit to encode the computation.

The Deutsch–Jozsa quantum algorithm

The DJ algorithm determines whether an unknown function $f(x)$ is constant or balanced\(^\text{28}\). The algorithm affords the simplest demonstration of the power of a quantum computer over its classical counterpart. In general, consider an $n$-bit binary string $x$; a Boolean function $f$ can be defined on this $n$-bit domain space to a 1-bit range space, with the restriction that either the output is the same for all inputs (the function is constant) or the output is 0 for half the inputs and 1 for the other half (the function is balanced). All the $2^n$ possible input strings are valid inputs for the function ($f(x) = \{0, 1\}$). In quantum computation, these $n$-bit logical strings are in one-to-one correspondence with the eigenstates of $n$-qubits, and one can hence label the logical string $x$ by the eigenstate $|x\rangle$. Classically, for an $n$-bit domain space, one needs to compute the function at least $2^{n/2} + 1$ times in order to determine whether it is constant or balanced. The DJ algorithm achieves this on a quantum computer using only a single function call.

The usual implementation of the DJ algorithm (the Cleve version\(^\text{29}\)) for $n$ bits requires $n + 1$ qubits, the function $f$ being encoded through an $f$-dependent unitary transformation,

$$|x\rangle_{n\text{-bit}} |y\rangle_{1\text{-bit}} U_f \rightarrow |x\rangle_{n\text{-bit}} |y \oplus f(x)\rangle_{1\text{-bit}},$$

where $\oplus$ denotes addition modulo 2. As shown in Figure 10, the implementation of the unitary transformation $U_f$ along with the Hadamard transformation, then suffices to distinguish the function as constant or balanced\(^\text{28,29}\). A Hadamard transformation on one qubit mixes the eigenstates maximally,

$$|0\rangle \rightarrow \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle),$$
$$|1\rangle \rightarrow \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle).$$

The Hadamard transformation for $n$-qubits is the tensor product of the one-qubit transformation ($H^n = H \otimes H \otimes H \ldots \otimes H$), its action on the $n$-qubit eigen states being

$$H^n |x\rangle = \sum_{y=0}^{2^n-1} (-1)^{x \cdot y} |y\rangle,$$

where $x_j$ and $y_j$ are the $j$th entries of the $n$-bit strings $x$ and $y$.

The four possible functions for the one-bit DJ algorithm (encoded using one extra qubit) are categorized as shown in Table 2.

The unitary transformations corresponding to the four possible propagators $U_f$ are easily constructed:

$$U_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$
$$U_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

$$U_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$
$$U_4 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$
Figure 8. Implementation of a complete set of 24 one-to-one reversible 2D quantum gates on a three-
spin system using one observer and two work qubits. a, 2,3-dibromopropionic acid; b, 1D 1H NMR
spectrum of (a) dissolved in CDCl3, recorded on a Bruker DRX-500 spectrometer at 300 K; c, 2D
spectrum of 1 observer spin (l0), corresponding to various gates implemented using spin- and transi-
tion-selective pulses. The computation pulses for each of these gates are given in ref. 68. Low-power
rectangular/Gaussian pulses were utilized for various gates. Spin-selective pulses were 10 ms long and
the transition-selective pulses were 100–300 ms long. The phase of the computation pulses were cy-
cled through (x, –x) to suppress the distortions due to pulse imperfections. All experiments were car-
ried out in the time domain with 256 τp values and 256 complex data points along τp and with 2 scans
for each τp point. Zero filling to 512 × 512 complex data points was done prior to the 2D Fourier trans-
formation. All plots are shown in magnitude mode.

qubit (the ‘control’ qubit) remains in the superposition
state, while the desired result (f(0) ⊗ f(1)) is encoded as
the appearance or disappearance of the lines of the tar-
get qubit. The \( U_1 \) transformation corresponds to the
unity operation or ‘do nothing’, while the \( U_2 \) transform
is achieved by a spin-selective \( \pi \) pulse on the control
qubit. The \( U_3 \) and \( U_4 \) transformations are implemented by
selective \( \pi \) pulses on the \( |10\rangle \rightarrow |11\rangle \) and the
\( |00\rangle \rightarrow |01\rangle \) transitions, respectively.

In a single measurement, one can distinguish between
constant and balanced functions on the basis of the dis-
appearance of the lines of the target qubit in the spec-
trum. These predictions are borne out by the
experimental spectra in Figure 11. The phase of the
transition-selective pulse (used to implement the \( U_3 \) and
\( U_4 \) transformations) has been stepped through \( (x, –x, \ y, –y) \) to suppress phase distortions and leads to the total
suppression of the target qubit lines and the retention of
only one line of the control qubits. The implementation
of the algorithm does not require pure initial states,
since similar results can be read out from the spectrum
if one starts with thermal initial states instead. The 2D
implementation of the DJ algorithm on the same system
is shown in Figure 12 (ref. 68).
A modified scheme can be designed to solve the $n$ bit Deutsch problem, using $n$ qubits alone\(^6\). In this scheme, for every function $f$ a unitary transformation is constructed, such that its action on the eigenstates of $n$ qubits is
\[
\left| x \right\rangle_{n\text{-bit}} U_f (-1)^{f(x)} \left| x \right\rangle_{n\text{-bit}}.
\]

Consider $n$ qubits, all in the state $|0\rangle$; a Hadamard transformation $H^n$ converts this state to a linear superposition of all $2^n$ eigenstates with equal amplitudes and no phase differences. The unitary transformation $U_f$ (defined in eq. (14)) acting on this state, introduces an $f$-dependent phase factor in each eigenstate in the superposition. At this juncture, all information about $f$ is encoded in the quantum state of the $n$ qubits. A Hadamard transformation $H^n$ is once again applied in order to extract the function's constant or balanced nature.
The categorization of the function as constant or balanced through a single function call using \( n \) qubits, is shown pictorially in Figure 13. The number of functions for the \( n \)-bit Deutsch problem is \( 2^C_{N/2} + 2 \) (where \( N = 2^n \)).

The NMR implementation of the modified DJ algorithm for one, two and three qubits, would require the implementation of unitary transformations corresponding to 4, 8 and 72 functions, respectively. This implementation of the DJ algorithm does not require the initial preparation of the spins in a pseudo-pure state, since the thermal equilibrium state serves equally well as a good initial state. The observable spectral result is the same in both cases, though beginning with a pseudo-pure state creates some (undetectable) multiple-quantum coherence. We have implemented the unitary transformations using composite-\( z \) pulses, spin-selective pulses and evolution under the scalar \( J \)-coupling. We discovered that it is only at the level of three or more input qubits that entangling transformations are essential\(^{14}\). The three-qubit DJ algorithm thus affords the simplest example where quantum entanglement plays a definitive role in the computation.

Concluding remarks

This article seeks to give a short account of the state-of-the-art in NMR quantum computing, by demonstrating experimentally various selective pulse and multi-dimensional implementations of pseudo-pure states, novel quantum logic gates and quantum algorithms. In conclusion, we would like to reiterate that NMR is a promising technique to illustrate and explore ideas in quantum computation. The ease with which quantum circuits can be implemented in multi-pulse FTNMR experiments and the facility with which spin dynamics can be manipulated using a variety of techniques is a major advantage. However, one of the challenges for NMR spectroscopists at the moment is to increase the number of qubits available for computation. Efforts are on in this direction, using liquid crystal solvents, labelled nuclei in hetero-nuclear systems, quadrupolar nuclei and even solid-state NMR techniques. We hope NMR will be able to better address current problems in quantum computing such as implementing quantum algorithms with a greater number of qubits, constructing error correcting circuits for fault-tolerant computing, and performing large-scale quantum simulations.

RESEARCH ACCOUNT


ACKNOWLEDGMENTS. The use of 400 and 500 MHz FTNMR spectrometers of the Sophisticated Instruments Facility (SIF), Indian Institute of Science, funded by Department of Science and Technology, New Delhi is gratefully acknowledged. We also thank Prof. K. V. Ramanathan and Dr G. A. Nagannagowda of SIF for useful discussions.

Received 29 August 2000; revised accepted 27 October 2000