Pramana – J. Phys. (2019) 92:26
https://doi.org/10.1007/s12043-018-1684-0

Hybrid scheme for factorisation: Factoring 551 using a 3-qubit NMR quantum adiabatic processor

SOHAM PAL[1,*], SARANYO MOITRA[2], V S ANJUSHA[1], ANIL KUMAR[3] and T S MAHESH[1]

[1]Department of Physics, Indian Institute of Science Education and Research, Pashan 411 008, India
[2]SISSA – International School for Advanced Studies, via Bonomea 265, 34136 Trieste, Italy
[3]Department of Physics and NMR Research Centre, Indian Institute of Science, Bengaluru 560 012, India
*Corresponding author. E-mail: soham.pal@students.iiserpune.ac.in

MS received 21 November 2017; revised 25 May 2018; accepted 22 June 2018; published online 2 January 2019

Abstract. Quantum processors are potentially superior to their classical counterparts for many computational tasks including factorisation. Circuit methods as well as adiabatic methods have already been proposed and implemented for finding the factors of a given composite number. The main challenge in scaling it to larger numbers is the unavailability of large number of qubits. Here, we propose a hybrid scheme that involves both classical and quantum computation, based on the previous work of Peng *et al*, *Phys. Rev. Lett.* **101(22)**, 220405 (2008), which reduces the number of qubits required for factorisation. The classical part involves setting up and partially simplifying a set of bit-wise factoring equations and the quantum part involves solving these coupled equations using a quantum adiabatic process. We demonstrate the hybrid scheme by factoring 551 using a 3-qubit NMR quantum register.

Keywords. Quantum computation, quantum cryptography; quantum communication; NMR implementation of quantum computation.

PACS Nos 03.67.Ac; 03.67.Dd; 03.67.Lx

1. Introduction

Multiplying two large numbers is an easy task, but the other way, i.e. to find the prime factors of a large number, is very difficult. In fact, there is no known classical algorithm to factor a number with polynomial resources. For many present cryptographic techniques, such as Rivest–Shamir–Adelman (RSA), this fact forms the basis for ensuring secure communication [1].

Peter Shor [2,3] in his milestone paper introduced a quantum algorithm to factorise numbers with polynomial complexity. Since then, several experimental architectures, including NMR [4], photonic systems [5] and trapped ions [6], have been used to demonstrate Shor's algorithm by factoring small numbers. Factoring larger numbers has been hindered by the unavailability of a quantum register with large number of qubits. As the size of the quantum register increases, one has to encounter the challenges of increased complexity of qubit-selective quantum controls, decreased coherence times and difficulty in quantum measurements. Moreover, it is also believed that the quantum processors may only be as efficient as their classical counterparts in certain computational tasks [7]. In this context, it is practical and may even be advantageous to look for a hybrid processor which can reduce the burden on the quantum processor without compromising the overall efficiency of computation.

In this work, we provide such an example by describing a hybrid procedure that uses both classical and quantum routines. We describe factorisation of large numbers using two stages: (i) construction and simplification of bit-wise factoring equations using a classical processor and (ii) solving the bit-wise factoring equations using an adiabatic quantum processor. The adiabatic quantum factorisation was previously used to factor 21 [8] and 143 [9] using 3 and 4 qubits, respectively. The hybrid method allows significant reduction in the number of qubits, and hence the complexity of quantum operations. Here, we describe the factorisation of 551 using only 3 qubits as an example. It can be noted that 551 is not the only number or the highest number that can be factorised using 3 qubits, as shown later. Moreover, we experimentally demonstrate the adiabatic

solution of bit-wise factoring equations using a 3-qubit NMR system.

In the next section, we describe the theoretical aspects of the hybrid procedure for factorisation. In §3 we describe the NMR experiments to factor 551, and finally we conclude in §4.

2. Theory

Let n be an l_n-bit biprime which is to be factored into its two prime factors p and q, i.e. $n = p \times q$. We can encode the factors on two quantum registers with l_p and l_q qubits. In binary form, the composite number and its factors are

$$n = \sum_{i=0}^{l_n-1} 2^i n_i, \quad p = \sum_{j=0}^{l_p-1} 2^j p_j \quad \text{and} \quad q = \sum_{k=0}^{l_q-1} 2^k q_k. \tag{1}$$

Except for the cases where one of the factors p or q is 2, all biprimes n are odd and hence the least significant bit of n, p and q are 1, i.e. $p_0 = q_0 = 1$. The most significant bits can also be set to 1 by construction, i.e. $p_{l_p-1} = q_{l_q-1} = 1$.

We set up the bit-wise multiplication table and each column of the table gives rise to a factoring equation. An example for the said multiplication table is shown in table 1 for the composite number $N = 551$ ($l_n = 10$) with factors $p = 29$ ($l_p = 5$) and $q = 19$ ($l_q = 5$) following the prescription in [9]. Here, the first row indicates the bit places and the subsequent two rows (having bit variables p_1 to p_3 and q_1 to q_3) represent the two factors. The remaining rows indicate bit-wise products as well as the carry bits (c_{ij}) from one column to another as indicated in the table. In the following, we discuss how factoring can be achieved using a hybrid computer with lesser number of qubits.

2.1 *Bit-wise factoring equations*

It can be seen that there are two possible cases regarding the bit lengths of the factors: Case A: $l_n = l_p + l_q$ and Case B: $l_n = l_p + l_q - 1$. Without loss of generality, assuming $q < p$, one can show that $l_q \leq \lceil l_n/2 \rceil \leq l_p$, where $\lceil \cdot \rceil$ is the ceiling function. Therefore, depending on the bit size l_n of the composite number, one may try various possibilities for the bit sizes of factors, and there can be at most $\lceil l_n/2 \rceil$ of them. Typically, in cryptosystems which rely on the difficulty of prime factorisation, l_p and l_q are chosen to be comparable, else the factoring could be rendered easier. In the following, we set up the factoring equations for general l_p and l_q and then eventually focus on the case where $l_p = l_q = \lceil l_n/2 \rceil$.

First, it is important to note that not all the bits of the two factors contribute to ith bit of n. As $n = pq$,

$$\sum_{i=0}^{l_n-1} 2^i n_i = \sum_{k=0}^{l_q-1} \sum_{j=0}^{l_p-1} 2^{j+k} p_j q_k. \tag{2}$$

Reshuffling the sum on the right-hand side to collect terms with the same power of 2, we have

$$\sum_{i=0}^{l_n-1} 2^i n_i = \sum_{m=0}^{l_p+l_q-2} 2^m \sum_{k=\alpha_m}^{\beta_m} p_{m-k} q_k, \tag{3}$$

where $\alpha_m = \max(0, m - l_p + 1)$ and $\beta_m = \min(m, l_q - 1)$.

At every order m the sum $\sum p_{m-k} q_k$ can be broken up into a binary residue along with a carry variable (not necessarily binary) which adds to the terms in the next order $m + 1$. By the same token, the mth order will have

Table 1. Bit-wise multiplication table for $n = 551 = pq$ with $l_n = 10$, $l_p = l_q = 5$. c_{ij} is the carry bit from column i to column j.

	B_9	B_8	B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0
$p =$						1	p_3	p_2	p_1	1
$q =$						1	q_3	q_2	q_1	1
R_0						1	p_3	p_2	p_1	1
R_1					q_1	p_3q_1	p_2q_1	p_1q_1	q_1	
R_2				q_2	p_3q_2	p_2q_2	p_1q_2	q_2		
R_3			q_3	p_3q_3	p_2q_3	p_1q_3	q_3			
R_4		1	p_3	p_2	p_1	1				
	$8 \to 9$	$7 \to 8$	$6 \to 7$	$5 \to 6$	$4 \to 5$	$3 \to 4$	$2 \to 3$	$1 \to 2$		
Carry	c_{89}	c_{78}	c_{67}	c_{56}	c_{45}	c_{34}	c_{23}	c_{12}		
	$7 \to 9$	$6 \to 8$	$5 \to 7$	$4 \to 6$	$3 \to 5$	$2 \to 4$				
	c_{79}	c_{68}	c_{57}	c_{46}	c_{35}	c_{24}				
551	1	0	0	0	1	0	0	1	1	1

Table 2. Bit-wise multiplication table for $n = 551 = pq$. C_i are the cumulative carries from column $i-1$ to column i.

	B_9	B_8	B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0
$p =$						1	p_3	p_2	p_1	1
$q =$						1	q_3	q_2	q_1	1
R_0						1	p_3	p_2	p_1	1
R_1					q_1	p_3q_1	p_2q_1	p_1q_1	q_1	
R_2				q_2	p_3q_2	p_2q_2	p_1q_2	q_2		
R_3			q_3	p_3q_3	p_2q_3	p_1q_3	q_3			
R_4		1	p_3	p_2	p_1	1				
	$8 \to 9$	$7 \to 8$	$6 \to 7$	$5 \to 6$	$4 \to 5$	$3 \to 4$	$2 \to 3$	$1 \to 2$	$0 \to 1$	
Carry	C_9	C_8	C_7	C_6	C_5	C_4	C_3	C_2	C_1	0
551	1	0	0	0	1	0	0	1	1	1

an 'incoming' carry variable C_m from the $(m-1)$th order. Thus, the factoring stands as

$$\sum_{k=\alpha_m}^{\beta_m} p_{m-k}q_k + C_m = n_m + 2C_{m+1} \tag{4}$$

for $0 \le m \le l_p + l_q - 2$. The advantage is that unlike in the prescription in [9] the factoring equations in (4) only couple adjacent orders, i.e. the mth equation gets connected only to the $(m-1)$th and $(m+1)$th equations. The trade-off is that these 'cumulative' carry variables C_m will, in general, take values in the set of non-negative integers.

The next step is, using the elements (R_j, B_i) of table 1, to form a new table, i.e., table 2. Here each cell has three elements: first element s_{ji} denotes least significant bit of the sum of the elements of cell $(j-1, i)$ the second element is the bit-wise product (R_j, B_i) of table 1 and the third element c_{ji} is the carry from the cell $(j, i-1)$. As each cell has only three bits, the carry is always a single bit and is always from the cell in the right. This is an advantage of table 2 over table 1 in keeping track of the carry. From these definitions it is clear that adding a cell leads to a sum and a carry, i.e.,

$$s_{ji} + p_{i-j}q_j + c_{ji} = s_{j+1,i} + 2c_{j,i+1}. \tag{5}$$

Adding the cells column-wise, we obtain

$$\sum_{j=\alpha_i}^{\beta_i}(s_{ji} - s_{j+1,i}) + \sum_{j=\alpha_i}^{\beta_i} p_{i-j}q_j + \sum_{j=\alpha_i}^{\beta_i} c_{ji} = \sum_{j=\alpha_i}^{\beta_i} 2c_{j,i+1}.$$

As the first term is nothing but $s_{\alpha_i,i} - s_{\beta_i+1,i}$, and as $s_{\beta_i+1,i} = n_i$,

$$s_{\alpha_i,i} + \sum_{j=\alpha_i}^{\beta_i} p_{i-j}q_j + \sum_{j=\alpha_i}^{\beta_i} c_{ji} = n_i + 2\sum_{j=\alpha_i}^{\beta_i} c_{j,i+1}. \tag{6}$$

It can be proved that

$$s_{\alpha_i,i} + \sum_{j=\alpha_i}^{\beta_i} c_{ji} = \sum_{j=\alpha_{i-1}}^{\beta_{i-1}} c_{ji} = C_i \ \text{(say)}. \tag{7}$$

Note that $C_0 \equiv 0$ because the first column cannot have an 'incoming' carry. Furthermore, substituting (4) into (3) we get

$$\sum_{i=0}^{l_n-1} 2^i n_i = \sum_{m=0}^{l_p+l_q-2} 2^m n_m + 2^{l_p+l_q-1}C_{l_p+l_q-1}$$

from which we can conclude that

$$C_{l_p+l_q-1} = \begin{cases} n_{l_n-1} = 1 & \text{for Case A: } l_n = l_p + l_q, \\ 0 & \text{for Case B: } l_n = l_p + l_q - 1. \end{cases}$$

From the structure of the factoring equations it is possible to readily assign values to some of the C_i, namely C_1 and $C_{l_p+l_q-2}$

$$\begin{aligned} m = 0 \qquad & : 1 = 1 + 2C_1 \Rightarrow C_1 = 0, \\ m = l_p + l_q - 2 & : C_{l_p+l_q-2} = n_{l_p+l_q-2} \\ & \quad + 2C_{l_p+l_q-1} - 1. \end{aligned}$$

The factoring equations can be put into a convenient matrix form as well. For concreteness, for $n = 551$, the matrix representation of eq. (4) is

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ q_1 & 1 & 0 & 0 & 0 \\ q_2 & q_1 & 1 & 0 & 0 \\ q_3 & q_2 & q_1 & 1 & 0 \\ 1 & q_3 & q_2 & q_1 & 1 \\ 0 & 1 & q_3 & q_2 & q_1 \\ 0 & 0 & 1 & q_3 & q_2 \\ 0 & 0 & 0 & 1 & q_3 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ p_1 \\ p_2 \\ p_3 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \\ C_7 \\ C_8 \\ C_9 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} + 2 \begin{bmatrix} 0 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \\ C_7 \\ C_8 \\ C_9 \\ 0 \end{bmatrix}. \tag{8}$$

Thus, a general factoring problem can be converted into solving equations of the above structure.

2.2 *Simplifying bit-wise factoring equations via classical processor*

Even though the C_i variables are not binary, it is possible to place bounds on them by noting that

$$\max[C_{i+1}] = \left\lfloor \frac{1}{2} \max \left(\sum_{k=\alpha_i}^{\beta_i} p_{i-k} q_k + C_i \right) - \frac{n_i}{2} \right\rfloor, \tag{9}$$

where $\lfloor \cdot \rfloor$ denotes the floor function. This is arrived at from rearranging the factoring equations. It is also possible to inductively determine an absolute upper bound for individual C_i irrespective of n_i, namely

$$\max[C_i] = \begin{cases} i-1 & \text{for } 1 \le i \le l_q - 1, \\ l_q - 1 & \text{for } l_q \le i \le l_p, \\ l_p + l_q - i & \text{for } l_p + 1 \le i \le l_p + l_q - 2. \end{cases} \tag{10}$$

The above values are used to initialise $\{Ci\}$ and then the bound on each element can be iteratively refined using eq. (9), where the maximum over the binary variables $\{p_i, q_i\}$ is evaluated in accordance with the constraints between them.

When $n = 551$, considering column B_1 from table 2 we find that $p_1 + q_1 = 1 + 2C_2$ while

$$\max[C_2] = \left\lfloor \frac{1}{2} \left(\max \sum_{j=0}^{1} p_{1-j} q_j + \max[C_1] - n_1 \right) \right\rfloor = 0,$$

as $\alpha_1 = \max(0, 1-10+5+1) = 0, \beta_1 = \min(1, 4) = 1$ and therefore $f_2 = 0$. In the same way, using bit-wise logic, the classical processor can determine values of all other f_i's. For $n = 551$, using a simple numerical procedure we found that

$$C_3 = 0, \quad C_4 = 1, \quad C_5 = 2, \quad C_6 = 1, \\ C_7 = 1, \quad C_8 = 1 \quad \text{and} \quad C_9 = 1. \tag{11}$$

The simplified matrix representation of the relevant factoring equations now becomes

$$\begin{bmatrix} q_1 & 1 & 0 & 0 \\ q_2 & 0 & 1 & 0 \\ q_3 & 0 & 0 & 1 \\ 0 & q_2 & q_1 & 0 \\ 0 & q_3 & 0 & q_1 \\ 0 & 0 & q_3 & q_2 \end{bmatrix} \begin{bmatrix} 1 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}. \tag{12}$$

Since they involve six unknowns, namely $\{p_1, p_2, p_3\}$ and $\{q_1, q_2, q_3\}$, it takes six variables to factor 551. However, a further reduction in the number of variables is possible by exploiting the first three equations, namely $p_1 + q_1 = 1$, $p_2 + q_2 = 1$ and $p_3 + q_3 = 1$, which together imply that $q_j = 1 - p_j$. Finally, only three unknowns define the factoring equations:

$$p_1(1-p_2) + (1-p_1)p_2 - 1 = 0, \\ p_1(1-p_3) + (1-p_1)p_3 - 1 = 0, \\ (1-p_2)p_3 + p_2(1-p_3) = 0, \tag{13}$$

which can be solved using 3 qubits. Here, we would like to stress upon the fact that these 3 qubits can be used to factor even larger bi-primes, given that the factors of the bi-prime differ from each other by only 3 bits. For example, the bi-prime 6767 with factors 67 $(1000011)_2$ and 101 $(1100101)_2$ can be factorised with only 3 qubits. In general, we state that any bi-prime with factors differing from each other by n bits can be factorised using n qubits [10]. Here 551 has only been used as an example.

In the following, we describe how these equations are solved using a 3-qubit adiabatic quantum processor.

2.3 *Solving the bit-wise factoring equations via quantum adiabatic processor*

2.3.1 *Quantum adiabatic algorithm.* Consider a closed quantum system existing in an eigenstate $|\psi_i\rangle$ of the initial Hamiltonian $\mathscr{H}_\mathrm{i}$ which is slowly changed to a new Hamiltonian $\mathscr{H}_\mathrm{f}$. Then, according to the quantum adiabatic theorem, the system mostly remains in an eigenstate of the instantaneous Hamiltonian and ultimately reaches the corresponding eigenstate of the final Hamiltonian, provided the system does not find two or more crossing eigenstates during the process [7,11].

Given a problem, adiabatic quantum computation typically involves encoding the solution to the problem in the ground state of the final Hamiltonian. A suitable initial Hamiltonian is chosen for which ground state can be prepared easily. Then the Hamiltonian of the system is slowly varied such that the system stays in the ground state of the instantaneous Hamiltonian. The intermediate Hamiltonian can be seen as an interpolation (linear or nonlinear) between the initial and final Hamiltonian

[12]. If T is the total time of evolution and $0 \leq s \leq 1$ is the interpolation parameter, then

$$\mathscr{H}(s) = (1-s)\mathscr{H}_{\mathrm{i}} + s\mathscr{H}_{\mathrm{f}}. \tag{14}$$

For linear interpolation we choose $s = t/T$, where t is the instantaneous time of evolution [13]. The adiabatic theorem requires that

$$T = \left| \frac{\max\{\mathrm{d}\mathscr{H}(s)/\mathrm{d}s\}}{\epsilon\Delta^2/\hbar} \right|, \tag{15}$$

where Δ is the minimum energy gap between the ground and the first excited states. Probability of reaching the ground state of the final Hamiltonian is given by $1-\epsilon^2$. From here onwards, we set $\hbar = 1$ and express the Hamiltonian in angular frequency units.

Now the entire time evolution of the system from $\mathscr{H}_{\mathrm{i}}$ to $\mathscr{H}_{\mathrm{f}}$ can be thought of as a unitary transformation U_{T} generated by a piece-wise constant Hamiltonian

$$\mathscr{H}_m = (1-m/M)\mathscr{H}_{\mathrm{i}} + (m/M)\mathscr{H}_{\mathrm{f}} \tag{16}$$

with M pieces, each of duration τ, and $0 \leq m \leq M$. Defining $U_m = \exp(-\mathrm{i}\mathscr{H}_m\tau)$, the total evolution operator $U_{\mathrm{T}} = \prod_{m=1}^{M} U_m$.

2.3.2 *Quantum adiabatic factoring.* In order to convert the factorisation problem into an optimisation problem, Peng *et al* [8] constructed a cost function $f(p,q) = (n - p\cdot q)^2$ which is minimum when p and q are the factors. They replace the scalar variables p and q with operators

$$P = \sum_{i=0}^{l_p-1} 2^i W_i \quad \text{and} \quad Q = \sum_{i=0}^{l_q-1} 2^i W_i. \tag{17}$$

Here the number operator $W_i = (I_2 - \sigma_{iz})/2$ is constructed in terms of the identity operator I_2 and the Pauli z-operator σ_z of the ith qubit. Note that eigenvectors $|0\rangle$ and $|1\rangle$ of W_i have the eigenvalues 0 and 1, respectively, the values a classical bit can take. Using this method, Peng *et al* [8] could factor the number 21 from the adiabatically prepared ground state of the final Hamiltonian

$$\mathscr{H}_{\mathrm{f}} = (NI_{2^n} - P\cdot Q)^2. \tag{18}$$

It can be noted that the ground state of the above Hamiltonian represents the factors. However, extending this method for factorising larger numbers is difficult because the Hamiltonian in eq. (18) can have many-body terms and required a large number of qubits.

Xu *et al* [9] improved upon this scheme using table 1. Each column of table 1 represents an equation which is subsequently encoded into a bitwise Hamiltonian, whose ground state contains the information about respective bits of the two factors. For example,

$$\begin{aligned} B_1:&\quad p_1 + q_1 - 1 - 2c_{12} = 0,\\ B_2:&\quad p_2 + p_1q_1 + q_2 + c_{12} - 1 - 2c_{23} - 4c_{24} = 0 \end{aligned}$$

and so on.

Now, the bit variables are replaced by the number operators: $p_j \to W_j$, $q_j \to W_{j+l_p-2}$. The carry bits $\{c_{j,j+i}\}$ are organised in a list according to increasing i for the same j and then in the order of increasing j. Each element k of the list is mapped onto $W_{k+l_q+l_p-4}$. The bit-wise Hamiltonians are then

$$\begin{aligned} B_1:&\quad \mathscr{H}_1 = (W_1 + W_4 - 1 - 2W_7)^2,\\ B_2:&\quad \mathscr{H}_2 = (W_2 + W_1W_4 + W_5 + W_7\\ &\qquad - 1 - 2W_8 - 4W_{15})^2 \end{aligned}$$

and so on. Thus, the final Hamiltonian of the factorisation problem is the sum

$$\mathscr{H}_{\mathrm{f}} = \sum_{i=1}^{l_n-1} \mathscr{H}_{\mathrm{i}}. \tag{19}$$

If the Hamiltonian is varied slowly enough, the adiabatic theorem ensures that the system ends up, with high probability, in the ground state of the target Hamiltonian. Therefore, on measuring the adiabatically prepared ground state of $\mathscr{H}_{\mathrm{f}}$, it is possible to retrieve the factors. Although the above encoding requires 20 qubits to factor the number 551, our hybrid scheme (§2.2) requires only 3 qubits.

2.3.3 *Quantum adiabatic factoring of* 551. In a hybrid computer, we first reduce the bit-wise factoring equations as described in §2.2 and then apply the quantum adiabatic algorithm to solve the residual equations. For the specific case of 551, the factoring equations are given by eq. (13). Replacing $p_j \to W_j = (I_2 - \sigma_{jz})/2$, we form the bit-wise Hamiltonian $\mathscr{H}_{\mathrm{i}}$. Final Hamiltonian (eq. (19)) becomes

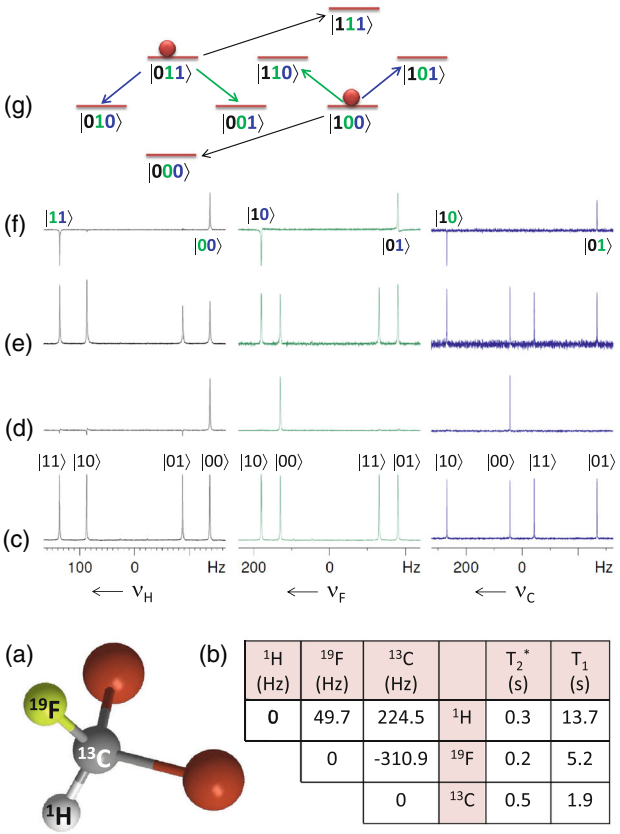
$$\mathscr{H}_{\mathrm{f}} = (3I_8 + \sigma_z^1\sigma_z^2 - \sigma_z^2\sigma_z^3 + \sigma_z^1\sigma_z^3)/2. \tag{20}$$

In the next section, we describe the experimental determination of the ground state of the above Hamiltonian which reveals the factors of 551.

3. Experiment

We implement the adiabatic factorisation of 551 on a 3-qubit NMR register involving ^{1}H, ^{19}F and ^{13}C of dibromofluoromethane (DBFM) dissolved in acetone-D6 [14]. All the experiments were carried out on a Bruker 500 MHz NMR spectrometer at an ambient temperature of 300 K.

The internal Hamiltonian for the 3-qubit system under week-coupling approximation [15,16] can be written as



^{1}H (Hz)	^{19}F (Hz)	^{13}C (Hz)		T_2^* (s)	T_1 (s)
0	49.7	224.5	^{1}H	0.3	13.7
	0	-310.9	^{19}F	0.2	5.2
		0	^{13}C	0.5	1.9

Figure 1. The molecular structure of DBFM is shown in (**a**). Resonance offsets (ν_i, diagonal elements), coupling constants (J_{ij}, off-diagonal elements) and relaxation parameters are tabulated in (**b**). The experimental NMR spectra correspond to thermal equilibrium (**c**), PPS (**d**), the ground state of initial Hamiltonian $\mathscr{H}_\mathrm{i}$ (**e**) and the solution, i.e. the ground state of the final Hamiltonian $\mathscr{H}_\mathrm{f}$ (**f**). The energy-level diagram (**g**) describes the deviation populations in the final state.

$$\mathscr{H}_\mathrm{int} = -2\pi \sum_{i=1}^{3} \nu_i I_z^i + 2\pi \sum_{i=1,j>i}^{i=2} J_{ij} I_z^i I_z^j, \tag{21}$$

where ν_i are the resonance offsets, J_{ij} are the coupling constants and I_z^i are the z-components of spin angular momentum operators. The molecular structure, Hamiltonian parameters and the thermal equilibrium spectra of DBFM are shown in figures 1a–1c, respectively.

The complete circuit for the experiment is shown in figure 2. The experiment mainly involves the following four stages:

(i) *Initialisation*: Preparation of $|000\rangle$ pseudopure state (PPS) from thermal equilibrium state was achieved by standard methods [17–19]. The PPS spectra shown in figure 1d corresponds to a fidelity of over 0.99.

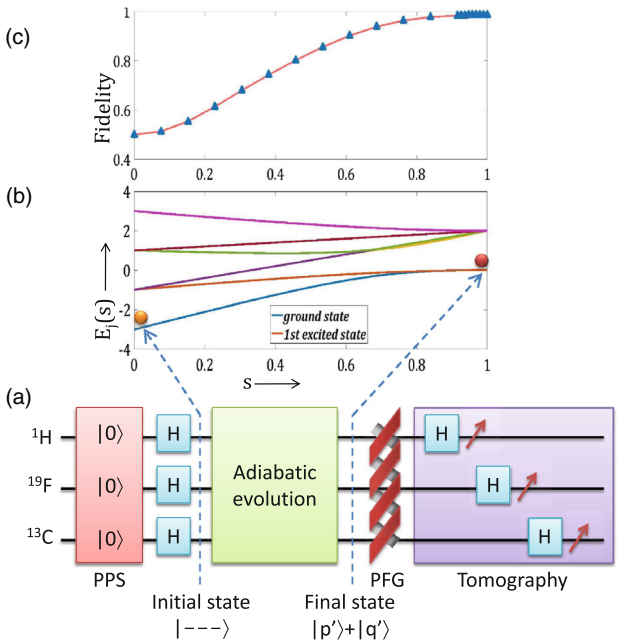


Figure 2. Three-qubit circuit for solving the bit-wise factoring equations (**a**), the transformation of energy spectrum during the adiabatic evolution (**b**) and the simulated fidelity of the solution state with the instantaneous ground state during the adiabatic evolution (**c**).

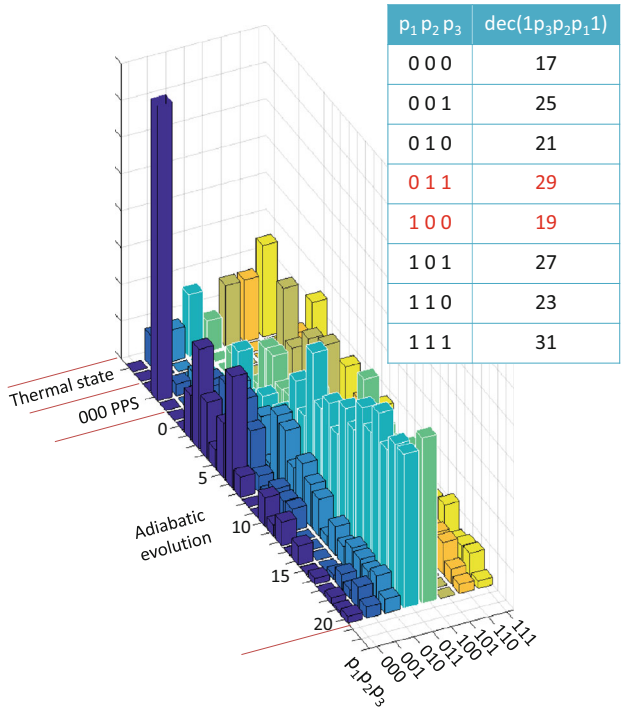
(ii) *Preparing the ground state*: We choose the initial Hamiltonian as

$$\mathscr{H}_\mathrm{i} = \sigma_x^1 + \sigma_x^2 + \sigma_x^3, \tag{22}$$

whose ground state is $|{-}{-}{-}\rangle$ (where $|\pm\rangle = (|0\rangle \pm |1\rangle)/\sqrt{2}$). Transforming the PPS into $|{-}{-}{-}\rangle$ was achieved by using three pseudo-Hadamard gates ($H = \exp[\mathrm{i}(\pi/2)\sigma_y/2]$) and the corresponding experimental spectra are shown in figure 1e.

(iii) *Adiabatic evolution*: The ground state of the initial Hamiltonian was driven adiabatically towards the ground state of the final Hamiltonian $\mathscr{H}_\mathrm{f}$ (as in eq. (19)) over a duration $T = 3.5$ s in 20 steps. The progression of energy eigenvalues $E_j(s)$ as a function of the interpolation parameter s is shown in figure 2b. Note that the ground state has no cross-over except at the end of the evolution where it becomes doubly degenerate. Each of these degenerate eigenstates encodes a factor. To quantify the overlap between the expected probabilities p_j^th and the simulated probabilities p_j^s after the sth step, we define a fidelity measure

$$F(s) = \frac{\sum_j p_j^\mathrm{th} p_j^s}{\sqrt{\sum_j (p_j^\mathrm{th})^2 \sum_j (p_j^s)^2}}. \tag{23}$$



$p_1\,p_2\,p_3$	dec($1p_3p_2p_11$)
0 0 0	17
0 0 1	25
0 1 0	21
0 1 1	29
1 0 0	19
1 0 1	27
1 1 0	23
1 1 1	31

Figure 3. Experimental probabilities of all the eight eigenstates at various stages of circuit in figure 2. Evolution of the probabilities during all the 20 adiabatic steps is shown. The table describes decoding the various eigenstates into respective decimal numbers. Factors highlighted in red achieve the highest probabilities during the adiabatic process.

The profile of $F(s)$ vs. the interpolation parameter s ultimately reaches a value of 0.99 at the end of evolution (see figure 2c).

The propagators corresponding to these adiabatic steps are realised using the recently developed Bang–Bang quantum control technique [20]. The obtained RF sequences were robust within an RF inhomogeneity of $\pm 10\%$ and had average fidelities above 0.99.

(iv) *Measurement of probabilities*: To demonstrate the evolution of the probabilities during the adiabatic process, we carried out 20 experiments each with varying length of the adiabatic sequence. In each experiment, after dephasing the coherences using a pulsed-field gradient [21], we measured the probabilities of various eigenstates in the computational basis (see figure 2a) [22,23]. The bar plots of the probabilities vs. the number of steps are shown in figure 3.

The experimental spectra of the final state and the corresponding population distributions are shown in figures 1f and 1g, respectively. The fidelity of the final state with the desired target state was over 0.99.

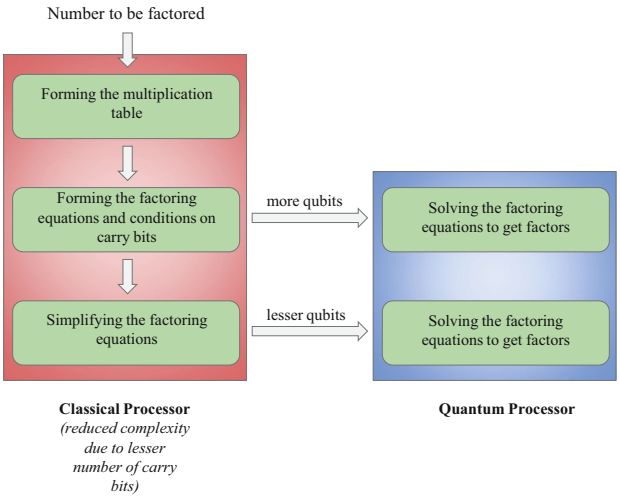


Figure 4. Flowchart of the overall process.

3.1 *Discussions*

It is clear from the table in figure 3 that the final state encodes the factors 19 and 29 with high probability. As with an NP problem, these factors can be verified easily.

An important issue is the complexity of the whole process, which is discussed qualitatively in the following. Formulating the bit-wise factoring equations (eq. (8)) involves mainly bit-wise multiplications, and hence polynomial in the bit size of the composite number (l_n). In principle, these factoring equations can directly be passed on to a quantum processor with a large number of qubits. Instead, we used some simple classical routines to reduce the size of the quantum register. This procedure involves computing upper bounds of cumulative carries C_i (see eq. (10)) and its complexity depends on the particular classical algorithm used. We presume that this optional procedure can be carried out efficiently without any exponential complexity. The quantum adiabatic process for solving the linear equations itself is believed to be polynomial [24,25]. Therefore, we believe that the overall factorisation procedure is efficient.

The crucial point in a hybrid scheme is to maximise the efficiency of the overall computation by optimising the switching point from classical to quantum processor. In this particular problem, simplifying the factoring equations to a higher extent will mean lesser number of required qubits during the quantum procedure. However, the complexity of classical simplification by itself should remain polynomial. The exact point of the crossover depends on the particular problem at hand and needs further investigation (figure 4).

In the case of factoring 551, it so happened that calculating the upper and lower bounds of carries C_i were enough to fix the values of the same. However, it is probable that for larger numbers, this procedure may

not be able to fix the values of all the carry variables, and the simplified factoring equations that are passed to the quantum routine may involve those unknown carries C_i. Nevertheless, these variables will be bounded from above and below, making the number of qubits required to encode them less than in the unbounded case.

4. Conclusions

Although, classical computers have seen an enormous progress over the past few decades, their difficulty in factorisation has become the corner stone of classical cryptography. Quantum computers are capable of factoring large numbers with polynomial complexity. Although prototype quantum computers capable of factoring small numbers have already been built, a large quantum computer outperforming a classical computer is just not around the corner. In this scenario, it is possibly more realistic to look for a hybrid computer having both classical and quantum processors.

In the present work, we analysed a possible scheme to factor a composite number by combining certain bitwise operations using a classical processor, and then solving a set of linear equations using an adiabatic quantum processor. We described the algorithm with respect to factoring the number 551 into 19 and 29 using only 3 qubits. Finally, we experimentally demonstrated the adiabatic quantum algorithm using a 3-qubit NMR quantum simulator, and obtained the factors with high probability. We believe this as a first step in exploiting the best of both the classical and quantum computational capabilities.

Acknowledgements

The authors acknowledge the useful discussions with Sudheer Kumar and Abhishek Shukla. SP and SM acknowledge the hospitality from Indian Institute of Science where this work was initiated. SM would like to thank Indian Academy of Sciences for the support during this period. This work was supported by the Department of Science and Technology, India (Grant Number DST/SJF/PSA-03/2012-13) and Council of Scientific and Industrial Research, India (Grant Number CSIR-03(1345)/16/EMR-II).

References

[1] N Koblitz, *A course in number theory and cryptography* (Springer, New York, 1994) Vol. 114
[2] P W Shor, *Proc. 35th Annual Symposium on Foundations of Computer Science* (IEEE Computer Society Press, Los Alatimos, CA, 1994) pp. 124–134
[3] P W Shor, *SIAM Rev.* **41(2)**, 303 (1999)
[4] L M K Vandersypen, M Steffen, G Breyta, C S Yannoni, M H Sherwood and I L Chuang, *Nature* **414(6866)**, 883 (2001)
[5] C-Y Lu, D E Browne, T Yang and J-W Pan, *Phys. Rev. Lett.* **99**, 250504 (2007)
[6] T Monz, D Nigg, E A Martinez, M F Brandl, P Schindler, R Rines, S X Wang, I L Chuang and R Blatt, *Science* **351(6277)**, 1068 (2016)
[7] M A Nielsen and I L Chuang, *Quantum computation and quantum information* (Cambridge University Press, Cambridge, 2010)
[8] X Peng, Z Liao, N Xu, G Qin, X Zhou, D Suter and J Du, *Phys. Rev. Lett.* **101(22)**, 220405 (2008)
[9] N Xu, J Zhu, D Lu, X Zhou, X Peng and J Du, *Phys. Rev. Lett.* **108(13)**, 130501 (2012)
[10] N S Dattani and N Bryans, *Quantum factorization of 56153 with only 4 qubits*, arXiv preprint arXiv:1411.6758 (2014)
[11] E Farhi, J Goldstone, S Gutmann, J Lapan, A Lundgren and D Preda, *Science* **292(5516)**, 472 (2001)
[12] H Hu *et al Phys. Rev. A* **93(1)**, 012345 (2016)
[13] A Messiah, *Quantum mechanics* (North-Holland, Amsterdam, 1962) Vol. 2
[14] A Mitra, T S Mahesh and A Kumar, *J. Chem. Phys.* **128(12)**, 124110 (2008)
[15] J Cavanagh, W J Fairbrother, A G Palmer III and N J Skelton, *Protein NMR spectroscopy: Principles and practice* (Academic Press, San Diego, 1995)
[16] M H Levitt, *Spin dynamics: Basics of nuclear magnetic resonance* (Wiley-VCH, Chichester, 2001)
[17] D G Cory, M D Price and T F Havel, *Phys. D: Nonlinear Phenom.* **120(1)**, 82 (1998)
[18] E Knill, I Chuang and R Laflamme, *Phys. Rev. A* **57**, 3348 (1998)
[19] J R Samal, A K Pati and A Kumar, *Phys. Rev. Lett.* **106**, 080401 (2011)
[20] G Bhole, V S Anjusha and T S Mahesh, *Phys. Rev. A* **93**, 042339 (2016)
[21] J A Jones, S D Karlen, J Fitzsimons, A Ardavan, S C Benjamin, G A D Briggs and J J L Morton, *Science* **324(5931)**, 1166 (2009)
[22] I L Chuang, N Gershenfeld, M G Kubinec and D W Leung, *Proc. R. Soc. London A: Math. Phys. Eng. Sci.* **454**, 447 (1998)
[23] R Das, T S Mahesh and A Kumar, *Phys. Rev. A* **67(6)**, 062304 (2003)
[24] A M Childs, E Farhi and J Preskill, *Phys. Rev. A* **65**, 012322 (2001)
[25] W Van Dam, M Mosca and U Vazirani, *Proceedings of the 42nd Symposium on Foundations of Computer Science* (IEEE, Los Alatimos, 2001) pp. 279–287