Quantum Factoring Algorithm using Grover Search

S. Whitlock[1*] and T. D. Kieu[2]

[1]European Center for Quantum Sciences (CESQ-ISIS, UMR 7006), University of Strasbourg and CNRS.
[2]Centre for Quantum Technology Theory and Optical Sciences Centre, Swinburne University of Technology, Victoria, Australia.

*Corresponding author(s). E-mail(s): whitlock@unistra.fr;

Abstract

We present a quantum algorithm for factoring products of prime numbers which exploits Grover search to factor any n-bit biprime using $2n - 5$ qubits or less. The algorithm doesn't depend on any properties of the number to be factored, has guaranteed convergence, and doesn't require complex classical pre or post-processing. Large scale simulations confirm a success probability asymptotically reaching 100% for >800 random biprimes with $5 \leq n \leq 35$ (corresponding to $5 - 65$ qubits) with the largest being $30398263859 = 7393 \times 4111763$. We also present a variant of the algorithm based on digital adiabatic quantum computing and show that Grover based factorization requires quadratically fewer iteration steps in most cases.

1 Introduction and context

Quantum computing represents a fundamental shift in how we process information which promises to solve hard computational problems out of the reach of even the most powerful classical computers. A celebrated example is Shor's quantum algorithm, discovered in 1994, which can factor a composite integer into its prime factors with a superpolynomial speedup over the best classical algorithms [1]. Soon after, in 1996 Grover proposed a quantum algorithm to find a given element in an unordered database of N elements in $\mathcal{O}(\sqrt{N})$ queries compared to $\mathcal{O}(N)$ required classically [2].

Shor's algorithm in particular has potentially enormous implications for information security and cryptography and has become a key algorithm for benchmarking noisy, intermediate-scale quantum computers [3–9]. However it is estimated that factoring integers relevant for cryptography with Shor's algorithm would require a quantum computer with tens of millions of qubits with error correction [10, 11], far beyond the capabilities of today's quantum computers. This has motivated more efficient implementations of Shor's algorithm [12–17] and new algorithms, based on e.g., quantum annealing, or quantum approximate optimization which potentially reduce the required number of qubits and/or gate counts [18–25]. However it is an open question if these approaches can solve large problem instances without relying on oversimplification [26–28].

In this paper we present a quantum algorithm to factor products of prime numbers exploiting a Grover search that uses only $2n - 5$ qubits, where n is the number of bits in the integer to be factored.

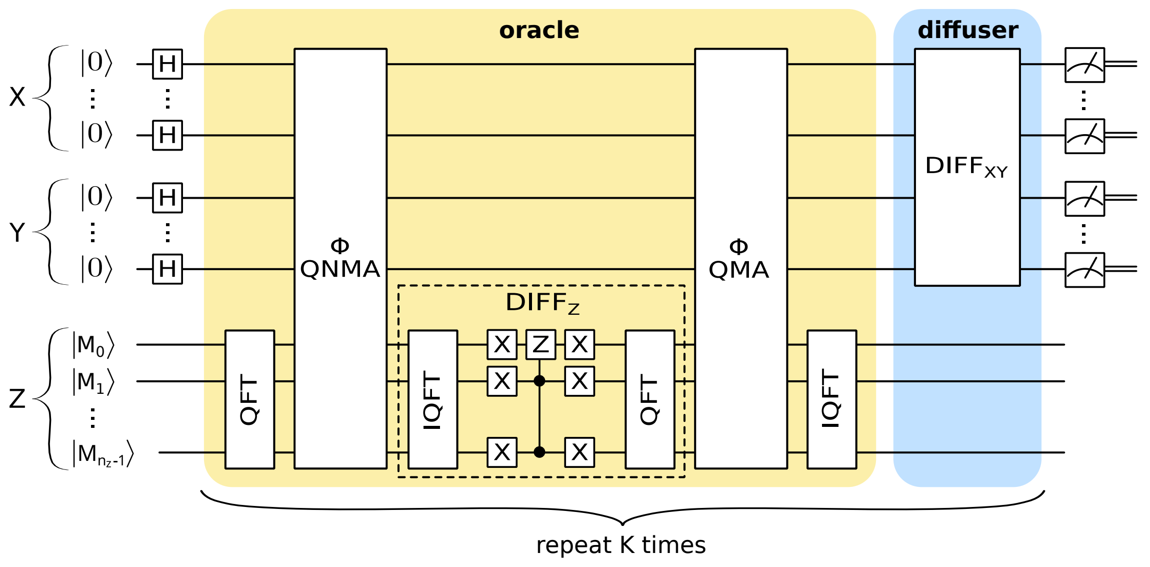

Fig. 1 Quantum algorithm for factoring a biprime N using Grover search. The input to the algorithm are the bits of $M = (N - S)/6 - 1$ encoded as the initial state of the Z register. The output is obtained by measuring the X, Y registers in the computational basis. The algorithm involves the repeated application of an Oracle and Diffuser composed of the following primitives: Quantum Fourier Transform (QFT), Inverse Quantum Fourier Transform (IQFT), Fourier Quantum Multiply Add (ΦQMA), Fourier Quantum Negative Multiply Add (ΦQNMA) and Diffusion (DIFF).

This improves on state of the art implementations of Shor's algorithm that require at least $2n + 1$ qubits [15] and facilitates implementation on quantum computers since the number of gates and thus also the sensitivity to errors typically scales strongly with the number of qubits. Furthermore, the algorithm is completely general and returns the prime factors with close to 100% probability without complex classical processing steps or midcircuit measurements and feedback. We also present a variant based on digital adiabatic quantum computing [29]. Through large scale numerical simulations we compare the performance of both algorithms and show that Grover based factoring requires quadratically fewer iteration steps than the corresponding digital adiabatic algorithm for the majority of problems.

2 Quantum factoring algorithm using Grover search

Grover's search algorithm efficiently locates a marked item in an unstructured database with 2^n entries using $\mathcal{O}(2^{n/2})$ queries, providing a quadratic speedup over classical searching. It can also be used to find the prime factors p, q of a n-bit integer $N = pq$. To illustrate, consider two quantum registers X, Y, with computational basis states $|x\rangle, |y\rangle$, both initialized in the uniform state. One then applies a loop, consisting in each step of an oracle operator which applies a π phase shift $|x, y\rangle \to -|x, y\rangle$ iff $xy = N$, where x, y is the integer representation of $|x\rangle, |y\rangle$. A concrete example of such an oracle operation built from elementary quantum circuits is shown in Fig. 1. This is followed by the Grover diffuser operation [30] which increases the amplitude in the marked state. The probability in the solution state increases in each step, approaching 100% after $\sim 2^{n/2}$ steps, such that the factors p, q can be directly read out by simply measuring X, Y in the computational basis. This approach can also be straightforwardly generalized to factor products of more than two primes.

Algorithm 1 and Fig. 1 present an implementation of a Grover based factoring algorithm which can be implemented on a quantum computer using $2n - 5$ qubits and elementary quantum arithmetic operations. Without loss of generality, we reduce the number of qubits required to represent N, p, and q taking advantage of the fact that primes and biprimes with $p, q > 3$ can be written as $N = 6(M+1)+S$,

$p = 6(a+1) + s$, and $q = 6(b+1) + sS$, where M, a, b are non-negative integers and $S, s = \pm 1$. M and S are uniquely determined from N with $S = \frac{3}{2} - \frac{1}{2}(N \bmod 6)$, noting that $N \bmod 6 \neq 3$ assuming 3 is not a factor of N. Thus prime factorization can be re-expressed as the problem of finding the integers a, b which satisfy $M = f(a,b)$ with $f(a,b) = 6(a+1)(b+1) + s(b+1) + sS(a+1) - 1$.

Implementing the algorithm using quantum circuits requires three quantum registers, X, Y, Z, with n_x, n_y and n_z qubits respectively, where n_x, n_y must be large enough to hold the integers a, b and $n_z = n_x + n_y + 3$ auxiliary qubits are used for computing $f(x,y)$. The X, Y registers are initially prepared in the uniform state, e.g., by applying Hadamard gates to each qubit. The Z register is initialized in the state $|z = M\rangle$ where $M = 2^0 M_0 + 2^1 M_1 \ldots 2^{n_z - 1} M_{n_z - 1}$.

The core of the factorization algorithm consists of a loop with K identical steps (Fig. 1). In each step we apply an oracle operator which adds a π phase shift to any solution states which satisfy $x = a, y = b$ (or $x = b, y = a$), with $M = f(a,b)$ for a specific choice of s, e.g., $s = +1$. This is followed by the diffusion operator from Grover's algorithm applied on the X and Y registers which increases the amplitude in the solution state(s). Finally the state of the X and Y registers are measured in the $0, 1$ basis giving the values a, b, from which the factors p, q can be straightforwardly calculated. Convergence of the algorithm, with probability in the solution state(s) $P \to 1$, is guaranteed within $K_{\text{opt}} = \lfloor (\pi/4) 2^{(n_x+n_y)/2} \rfloor$ steps. In cases where there are two valid solutions, i.e. $|x = a, y = b\rangle$ and $|x = b, y = a\rangle$ (possible if the factors are of a similar size, or if n_x and n_y are chosen larger than needed) then the algorithm will converge in $K_{\text{opt}} = \lfloor (\pi/4) 2^{(n_x+n_y)/2} / \sqrt{2} \rfloor$ steps. In case $N \neq pq$ then the algorithm should be run a second time with $s = -1$.

The oracle can be decomposed in terms of several circuit primitives as shown in Fig. 1. A key operation is the quantum multiply-and-add gate (QMA) and its inverse (QNMA), which are similar to

Algorithm 1 Factorize(N) using Grover search

```
Input:
    • The integer $N$ to be factored and the number of iteration steps $K$.
      For convenience we define $M = (N - S)/6 - 1$ and $S = \frac{3}{2} - \frac{1}{2}(N \bmod 6)$
    • Three registers $X, Y, Z$ containing, $n_x$, $n_y$ and $n_z = n_x + n_y + 3$ qubits respectively.
Output:
    • Two bitstrings $x = a, y = b$ satisfying $N = (6a + 6 + s)(6b + 6 + sS)$ with $s = \pm 1$
Procedure:
    Step 0. Check that $N$ is not a multiple of 2 or 3. Otherwise return the trivial factors.
    Step 1. Initialize registers
        • Prepare $X$ and $Y$ registers in the uniform state, i.e., $\frac{1}{\sqrt{2^{n_x+n_y}}} (|0\rangle + |1\rangle)^{\otimes(n_x+n_y)}$
        • Prepare $Z$ register in the state $|z = M\rangle$
    Step 2. Choose $s = 1$
    for $1 \leq k \leq K$ do
        Step 3. Apply oracle
            • $|x, y\rangle |z = M\rangle \to |x, y\rangle |z = 0\rangle$
            • Mark solution state $|z = 0\rangle \to -|z = 0\rangle$
            • $|x, y\rangle |z = 0\rangle \to |x, y\rangle |z = M\rangle$
        Step 4. Apply Grover's diffusion operator
    end for
    Step 5. Measure the $X$, $Y$ registers in the $0, 1$ basis. Calculate $p = 6x + s, q = 6y + sS$.
            If $N \neq pq$ restart with $s = -1$ in Step 2.
```

the fused multiply-add and fused negative multiply-add instructions in modern CPUs. These operations act on all three registers to realize the unitary transformation $|x, y\rangle|z\rangle \rightarrow |x, y\rangle|z \pm (axy + bx + cy + d) \mod 2^{n_z}\rangle$ where a, b, c, d are integer parameters. This can be realized with $\mathcal{O}(n^3)$ gates in the Fourier domain as a sequence of one-qubit phase gates and two- and three-qubit controlled phase gates acting on the Z register, preceded by a quantum Fourier transform QFT and followed by an inverse quantum Fourier transform IQFT (Supplementary Information). The QNMA, QMA operations envelop a (inverted) multicontrolled Z gate which applies a π phase shift to the $|z = 0\rangle$ basis state. This can be efficiently implemented using approximately $4n_z^2$ Toffoli gates or $12n_z^2$ elementary two qubit gates using the X, Y registers as auxiliary qubits [31].

There are a few simplifications which can further reduce the gate count and sensitivity to errors. The first QFT and the last IQFT in the oracle can be trivially pulled out of the loop and only need to be applied once. Alternatively one could re-initialize the Z register in each iteration which might help reduce the accumulation of gate errors. We then note that the middlemost gates, including the IQFT/QFT performs an operation that is effectively equivalent to the diffusion operator (DIFF_Z). Thus they can be replaced by $\mathrm{RY}(\pm\pi/2)$ rotation gates applied to each qubit surrounding a closed multicontrolled Z gate (Supplementary Information). In terms of elementary gates, the entire algorithm requires approximately $\mathcal{O}(n^3 2^{n/2})$ gates, where the factor $2^{n/2}$ is the number of Grover steps and n^3 gates are required for the multiply operations. The Supplementary Information includes a breakdown of the number of elementary gates required for the algorithm assuming an architecture with all-to-all connectivity.

The maximum number of qubits required for the algorithm can be constrained depending on the number of bits in N as

$$n_x = \left\lfloor \frac{n}{2} - 2 - d \right\rfloor, \qquad n_y = \left\lceil \frac{n}{2} - 2 + d \right\rceil$$

where d is the bit distance $d = (n_y - n_x)/2$. In many cases of practical relevance such as the RSA cryptosystem, the bit distance is close to zero. Otherwise the algorithm must be run for different values of d in the range $0 \leq d \leq \lfloor \frac{n}{2} - 2 \rfloor$ until a solution is found. Since $n_x + n_y \leq n - 4$ and $n_z = n_x + n_y + 3$ (to prevent overflow), the algorithm requires a total of $2n - 5$ qubits in the worst case.

3 Digital adiabatic factoring algorithm

The presented factoring algorithm also permits a variant in terms of adiabatic quantum computing, which provides a direct means to compare the Grover based approach. In adiabatic quantum computing the solution to a computational problem is encoded in the ground state of a Hamiltonian H_P. Starting from a readily prepared ground state of an initial Hamiltonian H_I one then leverages the adiabatic principle to slowly transform the system to the ground state of H_P. In this case we search for the state $|x, y\rangle$ satisfying $(f(x, y) - M)^2 = 0$.

We consider the time dependent Hamiltonian $H(k) = (1 - k/K)H_I + (k/K)H_P$ that can be realized on a digital quantum computer through the Trotter expansion in terms of elementary quantum gates. Up to order $\mathcal{O}(\epsilon^2)$ the time evolution operator for a given step k can be approximated using the Trotter formula

$$U_k \approx e^{-i\epsilon(1-k/K)H_I} \times e^{-i\epsilon(k/K)H_P}. \tag{1}$$

The initial state for the X, Y registers coincides with the ground state of $H_I = -\frac{1}{2}\sum_j \sigma_x^j$ where σ_x^j is the Pauli X operator acting on qubit j. H_P can be written as

$$H_P = QMA\left(\sum_{\zeta=0}^{n_z-1} P_1^\zeta\right) QNMA \tag{2}$$

where $P_1^\zeta = |1\rangle_\zeta \langle 1|$ is a single-bit projection operator acting on qubit ζ of register Z, $QNMA = \sum_{x,y,z} |x, y, z - f(x, y) \mod 2^{n_z}\rangle \langle x, y, z|$ and $QMA = QNMA^\dagger$. The ground state of H_P is $|x = a, y = b\rangle |z = M\rangle$.

Algorithm 2 presents a digital adiabatic factoring algorithm which is straightforward to construct from Algorithm 1. It replaces the diffusion operator DIFF_{XY} with single qubit rotation gates $\mathrm{RX} = \exp(-\epsilon(1 - k/K)\sigma_x/2)$ and replaces the multicontrol Z gate in the oracle operator with single-qubit phase shift gates $\mathrm{R} = \exp(-i\epsilon(k/K)P_1)$ acting on all the qubits in the Z register. A special aspect of our digital adiabatic algorithm is the explicit inclusion of multiplication and addition gates in Eq. (2). We find this approach converges much faster (using fewer Trotter steps) than alternative implementations which directly minimize the energy of $H_P = (N - \hat{p}\hat{q})^2$ where $\hat{p}, \hat{q}$ are single-qubit projection operators acting on the X and Y registers respectively [18, 24]. We attribute this to the smaller spectral norm of Eq. (2), which is of $\mathcal{O}(poly(n))$ compared to $\mathcal{O}(poly(2^n))$ [32, 33].

While the adiabatic factoring algorithm requires fewer multiqubit gates per iteration step than the Grover-based algorithm, its convergence is not guaranteed (in case of a vanishing spectral gap) and it is an open question which types of problems might allow for a computational quantum speedup in adiabatic quantum computing [26, 34, 35].

Algorithm 2 Factorize(N) using digital adiabatic evolution

```
Input:
    • The integer $N$ to be factored, the number of iteration steps $K$ and step size $\epsilon$.
      For convenience we define $M = (N - S/6) - 1$ and $S = \frac{3}{2} - \frac{1}{2}(N \bmod 6)$
    • Three registers $X, Y, Z$ containing, $n_x$, $n_y$ and $n_z = n_x + n_y + 3$ qubits respectively.
Output:
    • Two bitstrings $x = a, y = b$ satisfying $N = (6a + 6 + s)(6b + 6 + sS)$ with $s = \pm 1$
Procedure:
    Step 0. Check that $N$ is not a multiple of 2 or 3. Otherwise return the trivial factors.
    Step 1. Initialize registers
        • Prepare $X$ and $Y$ registers in the uniform state, i.e., $\frac{1}{\sqrt{2^{n_x+n_y}}}(|0\rangle + |1\rangle)^{\otimes(n_x+n_y)}$
        • Prepare $Z$ register in the state $|z = M\rangle$
    Step 2. Choose $s = 1$
    for $1 \le k \le K$ do
        Step 3. Apply $H_P$ terms
            • $|x, y\rangle |z = M\rangle \to |x, y\rangle |z = 0\rangle$
            • Apply phase shift $\bigotimes_\zeta^Z e^{-i\epsilon(k/K)P_1^\zeta}$
            • $|x, y\rangle |z = 0\rangle \to |x, y\rangle |z = M\rangle$
        Step 4. Apply transverse field $\bigotimes_j^{X,Y} e^{i\epsilon(1-k/K)\sigma_x^j/2}$
    end for
    Step 5. Measure the $X$, $Y$ registers in the 0, 1 basis. Calculate $p = 6x + s, q = 6y + sS$.
            If $N \neq pq$ restart with $s = -1$ in Step 2.
```

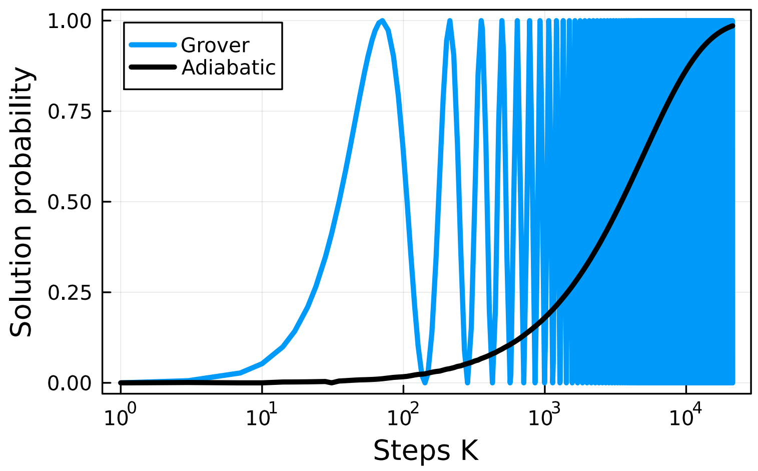

Fig. 2 Comparison of quantum factorization algorithms for $N = 101911 = 223 \times 457$ as a function of the iteration steps K. The blue line shows the success probability for the Grover based factorization which reaches ≈ 1 after $K = 71$ steps and then undergoes periodic oscillations. The digital adiabatic algorithm (black) requires considerably more iteration steps, only reaching $P > 0.9$ after $\gtrsim 10^4$ steps.

4 Numerical simulations

We evaluate both algorithms using a high performance quantum circuit simulator by QPerfect [36]. We use the exact state vector simulator which exploits highly optimized SIMD instructions to efficiently simulate deep quantum circuits. To study the largest possible range of biprimes we have also implemented a compiled version of the algorithm which applies the oracle directly to a state vector without the need for the auxiliary Z register [37]. This makes it possible to simulate the algorithm exactly for problem sizes up to $n \approx 35$ on a desktop computer, corresponding to ≈ 65 qubits. We verified that the compiled algorithm yields the same results as the decomposed version for problem sizes that can be simulated with the decomposed algorithm.

Figure 2 shows simulation results for the Grover based factorization algorithm and the adiabatic factorization algorithm for $N = 101911 = 223 \times 457$ ($n = 17$ corresponding to a simulation with $n_x + n_y = 13$) as a function of the number of iteration steps. For the adiabatic algorithm we use the step size $\epsilon = 0.45$ which maximizes the solution probability in the minimum number of steps. The Grover based algorithm reaches a probability $P \approx 1$ in the solution state in just $K_{\text{opt}} = 71$ steps and afterwards undergoes periodic oscillations. In comparison, the digital adiabatic algorithm requires many more steps, reaching $P > 0.9$ only after $> 10^4$ steps. Qualitatively similar behavior is seen for other biprimes.

We have tested the algorithm on > 800 factoring problems randomly selected with $n \leq 35$, confirming successful factorization in all cases, with $P \to 1$ for the Grover based algorithm. For the digital adiabatic algorithm we define success when the probability in the solution state reaches $P = 0.9$. Figure 3 summarizes the minimum required number of qubits and iteration steps for each problem. Both algorithms use the same number of qubits which ranges from $2n - 9$ to $2n - 5$ in the worst case depending on the specific problem. The number of gates per iteration step is dominated by the multiply and add operations scaling as n^3 in both algorithms. The scale of the simulations, here obtained using a single CPU with 64 GiB of total memory, are comparable to recent results for a highly optimized version of Shor's algorithm emulated on a 2048 GPU cluster [37]. The largest biprime we have factored

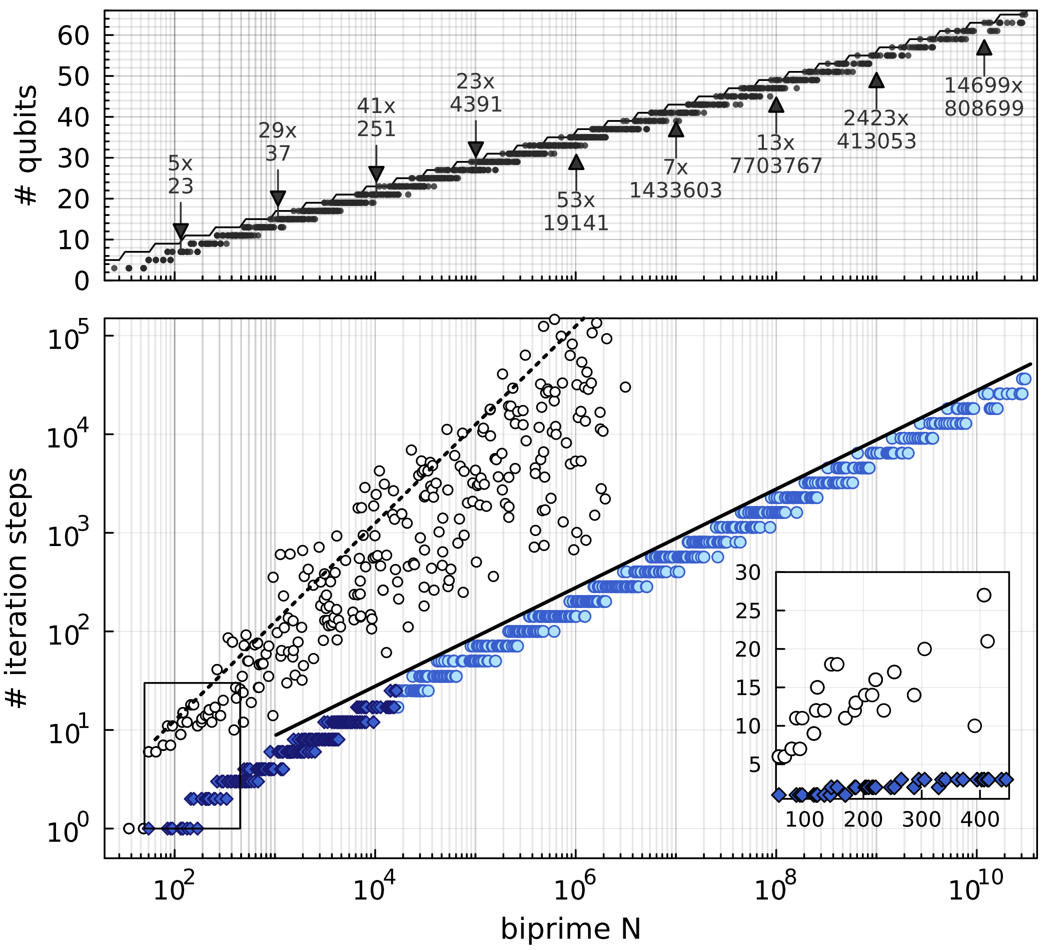

Fig. 3 Simulation results and required quantum resources for Grover based and digital adiabatic factoring algorithms for randomly selected biprimes N from 25 to $\approx 2^{35}$. (a) Total number of qubits versus problem size N. The black line depicts $2n-5$ scaling where n is the number of bits in N. (b) Total number of iteration steps versus problem size N for Grover (filled symbols) and digital adiabatic (open symbols) factoring algorithms. Dark diamonds indicate simulations performed of the full quantum algorithm using only one and two qubit gates. Light circles correspond to simulations using the compiled version of the algorithm. The solid black and dashed black lines show asymptotic scalings $\pi 2^{n/2-4}$ and 2^{n-4} respectively. The inset shows a zoomed in view of the section indicated in a box.

using the Grover based factoring algorithm on an emulator is $N = 30398263859 = 7393 \times 4111763$ (a $n = 35$ bit number).

The scale of these simulations allows us to investigate and compare the convergence and asymptotic complexity of both algorithms. Generally we find the Grover based algorithm converges with far fewer iteration steps than the digital adiabatic algorithm, with the difference increasing with N. In terms of time complexity the digital adiabatic algorithm requires between 2^{n-4} iteration steps (superlinear gate complexity in N) and $2^{\frac{n}{2}-1}$ (sublinear gate complexity in N) with large scatter depending on the problem instance. Convergence for the Grover algorithm however is much more deterministic, with the required number of iteration steps scaling asymptotically as $\pi 2^{n/2-4}$ (sublinear gate complexity). In terms of the required number of two-qubit gates, the digital adiabatic algorithm is competitive with the Grover algorithm for only a small fraction $\lesssim 0.1$ of problem instances studied.

5 Conclusion

We have presented a quantum algorithm for finding the prime factors of a composite integer that uses $2n - 5$ qubits or less, which compares favorably with other algorithms including state-of-the-art implementations of Shor's algorithm. The algorithm is general in that it can be used to factor any biprime just by changing several input bits and tends to 100% probability on the solution state after $2^{n/2}$ iteration steps and $\mathcal{O}(n^3 2^{n/2})$ elementary two-qubit gates. This is quadratically faster than a comparable algorithm based on digital adiabatic quantum evolution for the majority of problem instances. Grover's algorithm is optimal for unstructured problems [38, 39], however in terms of speed, the presented algorithm is likely not competitive against Shor's algorithm or the best classical factoring algorithms for large problem sizes, which highlights the need to exploit structure in the problem for achieving a practical quantum advantage over classical computers [40]. On the other hand we anticipate practical applications for benchmarking quantum computers on verifiable problems involving variable numbers of qubits. We estimate that with this new algorithm it should be possible to set new quantum factorization records for digital quantum computers. For example, with this new algorithm the number $N = 329$ could be factored with just 9 qubits and 278 elementary two-qubit gates. This could likely be further reduced using more efficient implementations of the quantum multiplier operation [41], or advanced circuit compilation techniques. We also remark that the presented algorithm can be straightforwardly generalized to solve multivariate polynomial equations with integer coefficients (Diophantine equations).

Acknowledgements We acknowledge valuable discussions with Johannes Schachenmayer and Guido Masella. This work has benefited from a state grant managed by the French National Research Agency under the Investments of the Future Program with the reference ANR-21-ESRE-0032 "aQCess - Atomic Quantum Computing as a Service", the Horizon Europe programme HORIZON-CL4-2021-DIGITAL-EMERGING-01-30 via the project "EuRyQa - European infrastructure for Rydberg Quantum Computing" grant agreement number 10107014 and support from the Institut Universitare de France (IUF).

Data availability The data reported in this manuscript and Julia scripts for constructing the circuit are available from https://github.com/aQCess/QuantumFactoring.

Competing interests SW is co-founder and shareholder of QPerfect.

References

[1] Shor, P.W.: Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings 35th Annual Symposium on Foundations of Computer Science, pp. 124–134 (1994). https://doi.org/10.1109/SFCS.1994.365700

[2] Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing. STOC '96, pp. 212–219. Association for Computing Machinery, New York, NY, USA (1996). https://doi.org/10.1145/237814.237866 . https://doi.org/10.1145/237814.237866

[3] Vandersypen, L.M., Steffen, M., Breyta, G., Yannoni, C.S., Sherwood, M.H., Chuang, I.L.: Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance. Nature **414**(6866), 883–887 (2001)

[4] Lu, C.-Y., Browne, D.E., Yang, T., Pan, J.-W.: Demonstration of a compiled version of Shor's quantum factoring algorithm using photonic qubits. Phys. Rev. Lett. **99**, 250504 (2007) https://doi.org/10.1103/PhysRevLett.99.250504

[5] Lanyon, B.P., Weinhold, T.J., Langford, N.K., Barbieri, M., James, D.F.V., Gilchrist, A., White, A.G.: Experimental demonstration of a compiled version of Shor's algorithm with quantum entanglement. Phys. Rev. Lett. **99**, 250505 (2007) https://doi.org/10.1103/PhysRevLett.99.250505

[6] Lucero, E., Barends, R., Chen, Y., Kelly, J., Mariantoni, M., Megrant, A., O'Malley, P., Sank, D., Vainsencher, A., Wenner, J., *et al.*: Computing prime factors with a Josephson phase qubit quantum processor. Nature Physics **8**(10), 719–723 (2012)

[7] Martin-Lopez, E., Laing, A., Lawson, T., Alvarez, R., Zhou, X.-Q., O'Brien, J.L.: Experimental realization of Shor's quantum factoring algorithm using qubit recycling. Nature photonics **6**(11), 773–776 (2012)

[8] Monz, T., Nigg, D., Martinez, E.A., Brandl, M.F., Schindler, P., Rines, R., Wang, S.X., Chuang, I.L., Blatt, R.: Realization of a scalable shor algorithm. Science **351**(6277), 1068–1070 (2016) https://doi.org/10.1126/science.aad9480 https://www.science.org/doi/pdf/10.1126/science.aad9480

[9] Amico, M., Saleem, Z.H., Kumph, M.: Experimental study of Shor's factoring algorithm using the IBM Q experience. Phys. Rev. A **100**, 012305 (2019) https://doi.org/10.1103/PhysRevA.100.012305

[10] Mosca, M.: Cybersecurity in an era with quantum computers: Will we be ready? IEEE Security & Privacy **16**(5), 38–41 (2018) https://doi.org/10.1109/MSP.2018.3761723

[11] Gidney, C., Ekerå, M.: How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits. Quantum **5**, 433 (2021) https://doi.org/10.22331/q-2021-04-15-433

[12] Cleve, R., Watrous, J.: Fast parallel circuits for the quantum Fourier transform. In: Proceedings 41st Annual Symposium on Foundations of Computer Science, pp. 526–536 (2000). https://doi.org/10.1109/SFCS.2000.892140

[13] Beauregard, S.: Circuit for Shor's algorithm using 2n+3 qubits. arXiv:quant-ph/0205095 (2002)

[14] Häner, T., Roetteler, M., Svore, K.M.: Factoring using 2n+2 qubits with toffoli based modular multiplication. Quantum Inf. Comput. **17**(7&8), 673–684 (2017) https://doi.org/10.26421/QIC17.7-8-7

[15] Gidney, C.: Factoring with n+ 2 clean qubits and n-1 dirty qubits. arXiv:1706.07884 (2017)

[16] Bernstein, D.J., Heninger, N., Lou, P., Valenta, L.: Post-quantum rsa. In: Post-Quantum Cryptography: 8th International Workshop, PQCrypto 2017, Utrecht, The Netherlands, June 26-28, 2017, Proceedings 8, pp. 311–329 (2017). Springer

[17] Regev, O.: An efficient quantum factoring algorithm. arXiv:2308.06572 (2023)

[18] Peng, X., Liao, Z., Xu, N., Qin, G., Zhou, X., Suter, D., Du, J.: Quantum adiabatic algorithm for factorization and its experimental implementation. Phys. Rev. Lett. **101**, 220405 (2008) https://doi.org/10.1103/PhysRevLett.101.220405

[19] Dridi, R., Alghassi, H.: Prime factorization using quantum annealing and computational algebraic geometry. Scientific reports **7**(1), 43048 (2017)

[20] Jiang, S., Britt, K.A., McCaskey, A.J., Humble, T.S., Kais, S.: Quantum annealing for prime factorization. Scientific reports **8**(1), 17667 (2018)

[21] Anschuetz, E.R., Olson, J.P., Aspuru-Guzik, A., Cao, Y.: Variational quantum factoring. arXiv:1808.08927 (2018)

[22] Kieu, T.D.: A factorisation algorithm in adiabatic quantum computation. Journal of Physics Communications **3**(2), 025014 (2019) https://doi.org/10.1088/2399-6528/ab060d

[23] Wang, B., Hu, F., Yao, H., Wang, C.: Prime factorization algorithm based on parameter optimization of ising model. Scientific reports **10**(1), 7106 (2020)

[24] Hegade, N.N., Paul, K., Albarrán-Arriagada, F., Chen, X., Solano, E.: Digitized adiabatic quantum factorization. Phys. Rev. A **104**, 050403 (2021) https://doi.org/10.1103/PhysRevA.104.L050403

[25] Yan, B., Tan, Z., Wei, S., Jiang, H., Wang, W., Wang, H., Luo, L., Duan, Q., Liu, Y., Shi, W., Fei, Y., Meng, X., Han, Y., Shan, Z., Chen, J., Zhu, X., Zhang, C., Jin, F., Li, H., Song, C., Wang, Z., Ma, Z., Wang, H., Long, G.-L.: Factoring integers with sublinear resources on a superconducting quantum processor. arXiv:2212.12372 (2022)

[26] Dam, W., Mosca, M., Vazirani, U.: How powerful is adiabatic quantum computation? In: Proceedings 42nd IEEE Symposium on Foundations of Computer Science, pp. 279–287 (2001). https://doi.org/10.1109/SFCS.2001.959902

[27] Smolin, J.A., Smith, G., Vargo, A.: Oversimplifying quantum factoring. Nature **499**(7457), 163–165 (2013)

[28] Khattar, T., Yosri, N.: A comment on "factoring integers with sublinear resources on a superconducting quantum processor". arXiv:2307.09651 (2023)

[29] Albash, T., Lidar, D.A.: Adiabatic quantum computation. Rev. Mod. Phys. **90**, 015002 (2018)

[30] Nielsen, M., Chuang, I.: Quantum Computation and Quantum Information. Cambridge University Press, Cambridge (2010). https://doi.org/10.1017/CBO9780511976667

[31] Barenco, A., Bennett, C.H., Cleve, R., DiVincenzo, D.P., Margolus, N., Shor, P., Sleator, T., Smolin, J.A., Weinfurter, H.: Elementary gates for quantum computation. Phys. Rev. A **52**, 3457–3467 (1995) https://doi.org/10.1103/PhysRevA.52.3457

[32] Aharonov, D., Dam, W., Kempe, J., Landau, Z., Lloyd, S., Regev, O.: Adiabatic quantum computation is equivalent to standard quantum computation. SIAM Review **50**, 755–787 (2008)

[33] Kieu, T.D.: A class of time-energy uncertainty relations for time-dependent hamiltonians. Proc. R. Soc. A. **475**, 20190148 (2019) https://doi.org/10.1098/rspa.2019.0148

[34] Rønnow, T.F., Wang, Z., Job, J., Boixo, S., Isakov, S.V., Wecker, D., Martinis, J.M., Lidar, D.A., Troyer, M.: Defining and detecting quantum speedup. Science **345**(6195), 420–424 (2014) https://doi.org/10.1126/science.1252319 https://www.science.org/doi/pdf/10.1126/science.1252319

[35] Hastings, M.B.: The Power of Adiabatic Quantum Computation with No Sign Problem. Quantum **5**, 597 (2021) https://doi.org/10.22331/q-2021-12-06-597

[36] MIMIQ Quantum Circuit Simulator. https://github.com/qperfect-io

[37] Willsch, D., Willsch, M., Jin, F., De Raedt, H., Michielsen, K.: Large-scale simulation of Shor's quantum factoring algorithm. Mathematics **11**(19) (2023) https://doi.org/10.3390/math11194222

[38] Bennett, C.H., Bernstein, E., Brassard, G., Vazirani, U.: Strengths and weaknesses of quantum computing. SIAM Journal on Computing **26**(5), 1510–1523 (1997) https://doi.org/10.1137/S0097539796300933

[39] Zalka, C.: Grover's quantum searching algorithm is optimal. Phys. Rev. A **60**, 2746–2751 (1999) https://doi.org/10.1103/PhysRevA.60.2746

[40] Aaronson, S.: How much structure is needed for huge quantum speedups? arXiv:2209.06930 (2022)

[41] Rines, R., Chuang, I.: High performance quantum modular multipliers. arXiv:1801.01081 (2018)

[42] Ruiz-Perez, L., Garcia-Escartin, J.C.: Quantum arithmetic with the quantum Fourier transform. Quantum Information Processing **16**, 1–14 (2017)

6 Supplementary Information

6.1 Decomposition of QMA in terms of elementary gates

To facilitate implementation on quantum computers we provide circuit based decompositions of the key elements of the quantum factoring algorithm in terms of elementary gates. The QMA operation performs the unitary transformation

$$QMA = \sum_{x,y,z} |x, y, z + axy + bx + cy + d \mod 2^{n_z}\rangle \langle x, y, z|$$

where a, b, c, d are integer coefficients and x, y, z refer to the basis states of the X, Y, Z registers in their integer representation. This operation can be efficiently implemented in the Fourier domain as a concatenation of weighted addition operations [42] shown as ΦADD and controlled ΦADD blocks in Fig. 4, preceded and followed by a QFT and IQFT respectively, which can be implemented in the standard way [30]. The circuit for ΦADD is depicted in Fig. 5. It multiplies a first register by a constant and adds it to a second register. The circuit is built using a series of controlled phase shift gates with exponentially decreasing phase angles.

$$R^j(c) = \begin{bmatrix} 1 & 0 \\ 0 & e^{ic\pi/2^j} \end{bmatrix}.$$

Controlled ΦADD blocks used for multiplication can be implemented by replacing the controlled phase gates with doubly controlled phase gates. Doubly controlled phase gates can be subsequently decomposed to three controlled phase gates and two CNOT gates each [31].

6.2 Decomposition of DIFF in terms of elementary gates

The diffusion operators DIFF$_{X,Y}$ and DIFF$_Z$ realize the operation $DIFF_Q = I - 2|s_Q\rangle\langle s_Q|$ where $|s_Q\rangle$ is the uniform state over the register Q,

$$|s_Q\rangle = \frac{1}{\sqrt{2^{n_q}}} \sum_{q=0}^{2^{n_q}-1} |q\rangle.$$

This can be implemented by the parallel application of RY($\pi/2$) rotation gates to each qubit (equivalent to a Hadamard gate followed by an X gate), then a multicontrolled Z gate and followed by another parallel application of RY($-\pi/2$) gates as depicted in Fig. 6. The multicontrolled Z gate can be further decomposed using a recursive application of Lemma 7.5 of [31] which expresses a general controlled 2×2 unitary in terms of $\mathcal{O}(n^2)$ elementary gates. This decomposition in turn uses multicontrolled X gates which can be efficiently decomposed into Toffoli gates (Lemma 7.2) and subsequently two-qubit gates using the other register(s) as auxiliary qubits.

Table. 1 summarizes the number of elementary gates needed for implementing each operation.

Primitive	Calls	Two-qubit gate count
ΦQMA	$2K$	$(5n_x n_y + n_x + n_y)(2n_z - n_x - n_y) + (12n_x n_y + n_x + n_y)$
DIFF$_{X,Y}$	K	$12(n_x + n_y)^2 - 82(n_x + n_y) + 149$
DIFF$_Z$	K	$12n_z^2 - 82n_z + 149$
QFT	2	$n_z(n_z - 1)$

Table 1 Two-qubit gate counts for the different elements of the quantum algorithms. The number of iterations required for convergence is $K = \lfloor(\pi/4)2^{(n_x+n_y)/2}\rfloor$

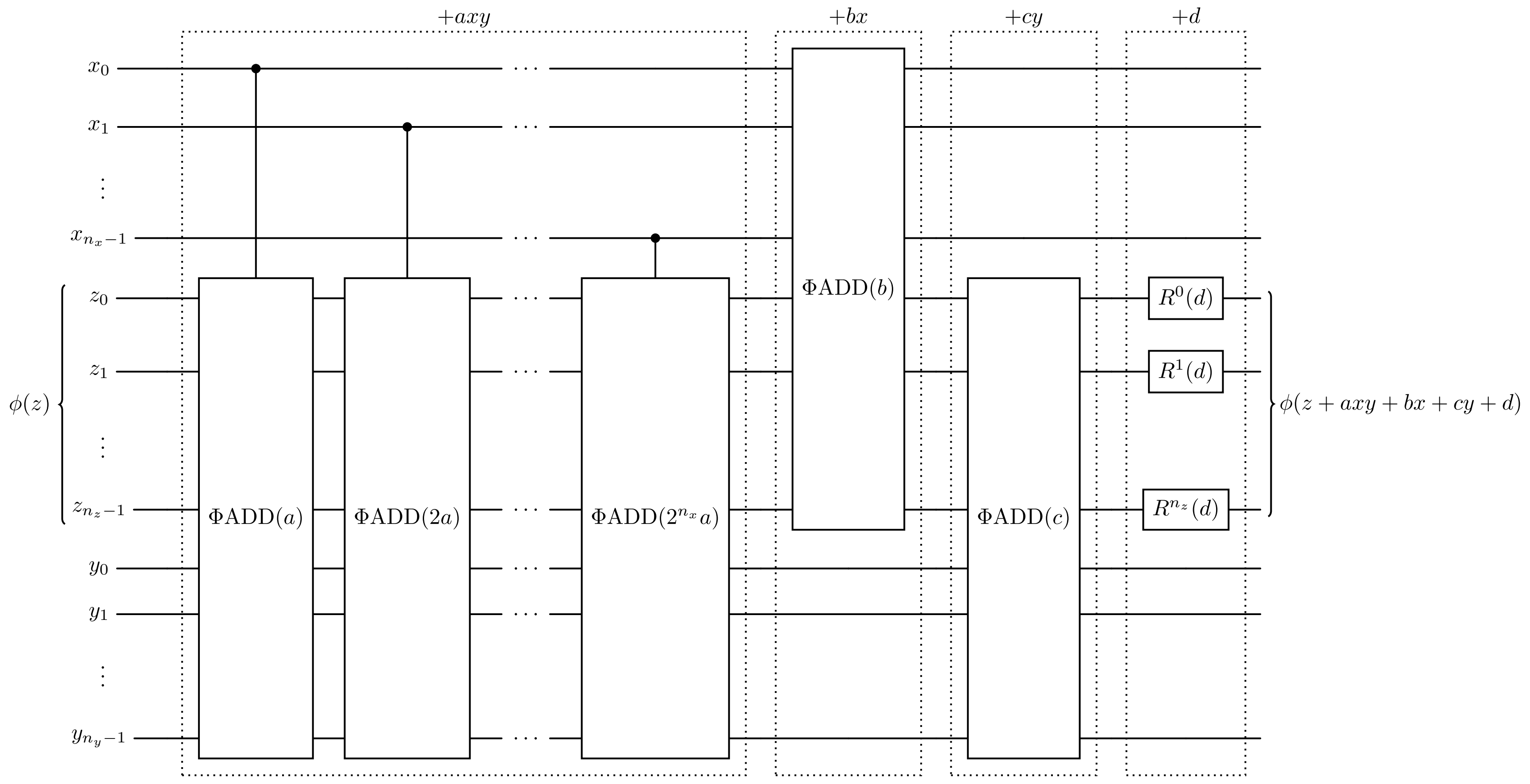

Fig. 4 ΦQMA(a). Quantum circuit for multiplying and adding two registers with integer scaling constants a, b, c, d

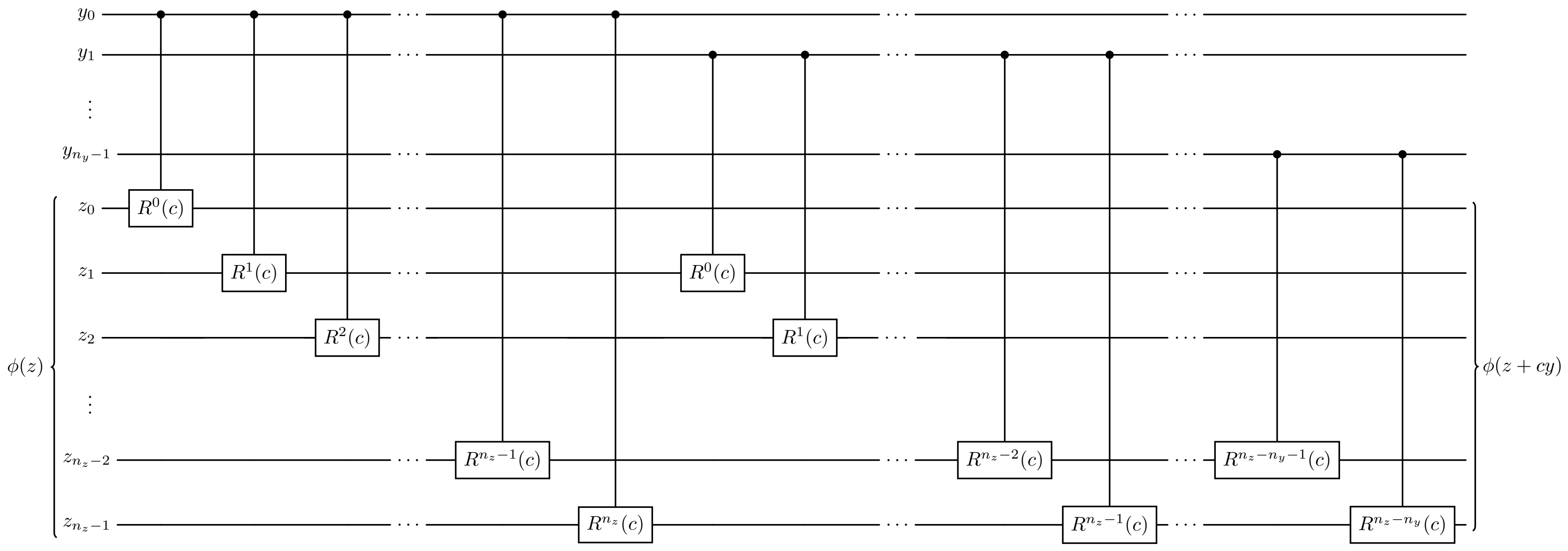

Fig. 5 ΦADD(c). Quantum circuit for adding two registers with integer scaling constant c

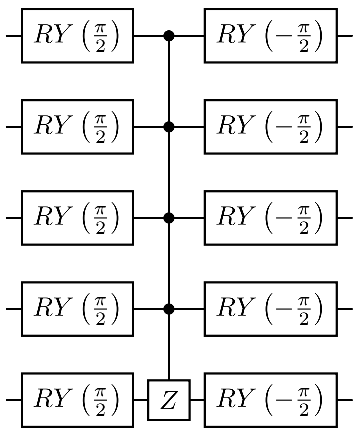
$RY\left(\frac{\pi}{2}\right)$
Z
$RY\left(-\frac{\pi}{2}\right)$

Fig. 6 DIFF. Quantum circuit for implementing the diffusion operation using a multicontrolled Z gate. The multicontrolled Z gate can be further decomposed using the procedure in Ref. [31]