Optimising matrix product state simulations of Shor's algorithm

Aidan Dang, Charles D. Hill, and Lloyd C. L. Hollenberg

Centre for Quantum Computation and Communication Technology,
School of Physics, The University of Melbourne, Parkville, Victoria 3010, Australia

2019-01-09

We detail techniques to optimise high-level classical simulations of Shor's quantum factoring algorithm. Chief among these is to examine the entangling properties of the circuit and to effectively map it across the one-dimensional structure of a matrix product state. Compared to previous approaches whose space requirements depend on r, the solution to the underlying order-finding problem of Shor's algorithm, our approach depends on its factors. We performed a matrix product state simulation of a 60-qubit instance of Shor's algorithm that would otherwise be infeasible to complete without an optimised entanglement mapping.

1 Introduction

With the potential for quantum computers to outperform the best classical computing resources available, achieving this quantum *supremacy* promises to be a major milestone in computing. However, the ability to demonstrate quantum supremacy [1–3] depends on several factors, including the computational task under consideration, as well as properties of the physical quantum computer (e.g. connectivity and error rate). On the side of classical computation, the difficulty in defining a quantum supremacy point for a given problem stems from bounding the computational power of classical machines. Therefore, we seek techniques to perform simulations of quantum algorithms and circuits more efficiently [3–5]. Generally, quantum algorithms possess some sort of structure that might be exploited to provide some advantage to classical simulation. In this paper, we examine how the entanglement structure of Shor's algorithm for integer factorisation lends itself to a particular matrix product state representation that quantifiably reduces the computational requirements for classical simulation. Additionally, we show how particular instances of Shor's algorithm become significantly easier to simulate when this structure is exploited.

Shor's algorithm [6] may be used to factor a semiprime integer N of l binary digits in polynomial time with respect to l on a quantum computer. Presently, there is no known algorithm to perform this factorisation efficiently (i.e. in polynomial time) on a classical computer. The presumed difficulty of factoring very large semiprime numbers is the foundation underpinning the security of the RSA public-key cryptosystem [7] used to secure online communications [8]. This provides motivation to build quantum computers capable of performing large enough instances of Shor's algorithm and to develop public-key cryptosystems resistant to quantum computers [9].

Existing physical implementations of Shor's algorithm [10–13] have been produced to factor small semiprimes no longer than five bits in length. This is significantly less than even the 15-bit instances first simulated in [14], requiring 45 qubits. Therefore, the simulated instances presented in this paper may be presented as medium-term goals for quantum hardware to benchmark against.

A typical state vector representation of an n-qubit system requires the storage of 2^n complex scalars, regardless of the state being stored. While simulations of quantum circuits may be performed by operating on this collection of scalars, the exponential space complexity ultimately limits the size of the systems that can reasonably be simulated. Instead, by using the matrix product state representation [15] of a quantum state, the space requirements scale according to the amount of entanglement present in the system. As tensor networks [16], matrix product states were originally used for simulating one-dimensional quantum many-body systems [17, 18], but have since been adapted for simulating quantum circuits [14, 15, 19]. As such, even states of many qubits may feasibly be stored using a matrix product state representation, provided its entanglement is sufficiently limited. Other examples of tensor networks that may be used to simulate quantum circuits include PEPS [20], MERA [21] and tree tensor networks [22].

By examining the entanglement introduced by a high-level circuit of Shor's algorithm, we were able to improve space requirements over previous simulations [14] by sensibly mapping this entanglement across the one-dimensional structure of a matrix product state. Furthermore, in treating the task of simulating Shor's algorithm as a sampling problem, we could take advantage of single-qubit measurement to collapse entanglement and hence reduce our space usage. The

Aidan Dang: aidan.dang@unimelb.edu.au
Charles D. Hill: cdhill@unimelb.edu.au
Lloyd C. L. Hollenberg: lloydch@unimelb.edu.au

combined result of our optimisations allowed us to simulate a nontrivial instance of Shor's algorithm involving 60 qubits on a supercomputer using an approximate total of 14 TB of RAM. In comparison to the $2^{60} \times 128\,\text{bit} \simeq 1.8 \times 10^7\,\text{TB}$ to store a state vector for 60 qubits in double precision, this represents a significant reduction in the required memory.

To outline this paper, we begin with a review of Shor's algorithm in Sec. 2 and a review of the matrix product state formalism in Sec. 3. We then examine the entanglement introduced at particular stages in Shor's algorithm in Sec. 4 and then detail in Sec. 5 how the subsystems of a matrix product state may be partitioned to take advantage of this entanglement evolution. Finally, we provide benchmarks of our implementation in Sec. 6, including our simulation of a 60-qubit instance.

2 Review of Shor's Algorithm

We consider the circuit shown in Fig. 1 for this review of Shor's algorithm and for the general structure of our following simulations. Given an odd, squarefree semiprime $N = pq$ represented by at least l binary digits, a high-level circuit for Shor's algorithm as detailed in [23] consists of an 'upper' register of $2l$ qubits initialised to $|0\rangle$ and a 'lower' register R of l qubits initialised to the state $|1\rangle$:

$$|\psi_{\text{init}}\rangle \equiv |0\rangle \otimes |1\rangle\,.$$

The Hadamard gate is then applied to each qubit of the upper register, creating the superposition

$$\begin{aligned}|\psi_{\text{superpos}}\rangle &\equiv (H^{\otimes 2l}) \otimes (I^{\otimes l})\,|\psi_{\text{init}}\rangle \\ &= \sum_{i=0}^{2^{2l}-1} |i\rangle \otimes |1\rangle\,,\end{aligned}$$

where we shall ignore normalisation constants for clarity. Randomly choosing integer $a \neq 1$ from $\mathbf{Z}_N^*$, the set of integers modulo and coprime to N, exponents of the unitary operator

$$U\,|x\rangle \equiv |ax \bmod N\rangle \tag{1}$$

are applied to the lower register, controlled by qubits in the top:

$$\begin{aligned}|\psi_{\text{modexp}}\rangle &\equiv \sum_{i=0}^{2^{2l}-1} |i\rangle \otimes U^i\,|1\rangle \\ &= \sum_{i=0}^{2^{2l}-1} |i\rangle \otimes \left|a^i \bmod N\right\rangle \\ &= \sum_{\substack{j=0 \\ kr+j<2^{2l}}}^{r-1} \sum_{\substack{k=0 \\ k<2^{2l}/r}} |kr+j\rangle \otimes \left|a^j \bmod N\right\rangle,\end{aligned} \tag{2}$$

where r is the period of U. To determine this period r, the lower register is measured, forcing a choice of the index j in Eq. (2). This collapses the entanglement between the two registers, and we can now separate them. The upper register is now in state

$$|\psi_{\text{upper}}\rangle \equiv \sum_{\substack{k=0 \\ k<2^{2l}/r}} |kr+j\rangle \tag{3}$$

for the $0 \leq j < r$ corresponding to the value ($a^j \bmod N$) measured from the lower register in Eq. (2).

The quantum Fourier transform (QFT) is then applied to this upper register:

$$\begin{aligned}\text{QFT}\,|\psi_{\text{upper}}\rangle &= \sum_{s=0}^{2^{2l}-1} \sum_{\substack{k=0 \\ k<2^{2l}/r}} e^{2\pi i(kr+j)s/2^{2l}}\,|s\rangle \\ &= \sum_{s=0}^{2^{2l}-1} e^{2\pi ijs/2^{2l}} \sum_{\substack{k=0 \\ k<2^{2l}/r}} e^{2\pi ikrs/2^{2l}}\,|s\rangle\,.\end{aligned}$$

The upper register is then measured to produce a value s with probability

$$\Pr(S=s) \propto \left| \sum_{\substack{k=0 \\ k<2^{2l}/r}} e^{2\pi ikrs/2^{2l}} \right|^2 \tag{4}$$

for random variable S. This implies that values of s such that $rs/2^{2l}$ is close to an integer are more likely to be measured, resulting in the peaks shown in the example probability distribution in Fig. 1(c). The quantum processing component of Shor's algorithm is now complete, and the result of S may be classically processed using the continued fractions and Euclidean algorithms to successfully factor N with high probability [23].

3 Review of Matrix Product States

To briefly review the matrix product state (MPS) [15] formalism for representing quantum states, we begin with some general composite state of n subsystems given by

$$|\psi\rangle = \sum_{i_1=0}^{d_1-1} \sum_{i_2=0}^{d_2-1} \cdots \sum_{i_n=0}^{d_n-1} \psi_{i_1 i_2 \ldots i_n}\,|i_1\rangle \otimes |i_2\rangle \otimes \cdots \otimes |i_n\rangle\,. \tag{5}$$

A subsystem m with dimensionality d_m is referred to as a d_m-level *qudit*. In the case where $d_m = 2$, m is a qubit. Conventionally, the *state-vector* approach for computationally storing this state involves recording each of the complex coefficients $\psi_{i_1 i_2 \ldots i_n}$ in a single, perhaps multidimensional, array. The space complexity of this particular representation is fixed at $O(d)$ regardless of the specific state $|\psi\rangle$, where $d = \prod_{m=1}^{n} d_m$.

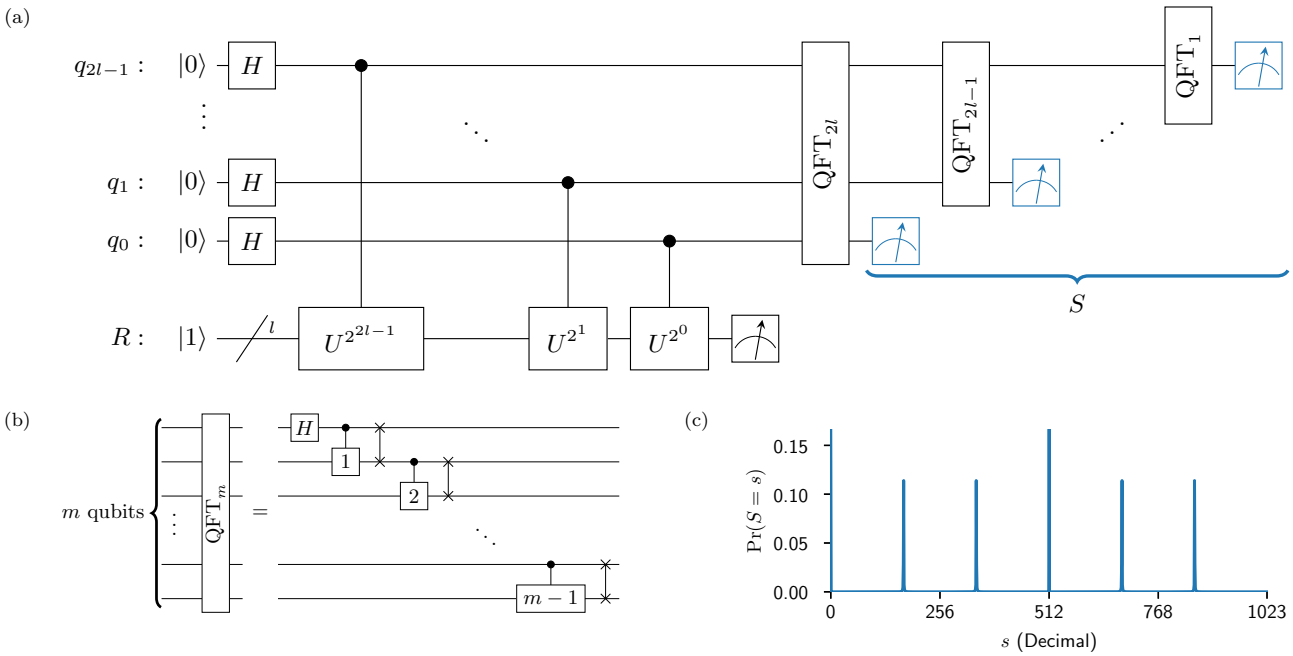

Figure 1: Schematic for the order-finding quantum subroutine in Shor's algorithm. (a) High-level circuit diagram for factoring a semiprime N of l binary digits. For randomly chosen a such that $1 < a < N$, U performs $U|x\rangle \to |ax \bmod N\rangle$. S is the result of final measurement of the upper register. (b) A single block in the linear nearest-neighbour quantum Fourier transform. A phase gate labelled by x performs $|0\rangle \to |0\rangle$ and $|1\rangle \to \exp(-i\pi/2^x)|1\rangle$. (c) Example probability distribution for S, where $l = 5$, $N = 21$, and $a = 2$.

At its core, the MPS representation of a state involves storing matrices for each qudit, whose product equals each of the coefficients in Eq. (5):

$$\psi_{i_1 i_2 \ldots i_n} = \Gamma^{[1]i_1}_{\alpha_1} \Gamma^{[2]i_2}_{\alpha_1 \alpha_2} \cdots \Gamma^{[n]i_n}_{\alpha_{n-1}}, \tag{6}$$

where we have used the usual summation notation. Notably, qudits whose matrices are adjacent in Eq. (6) may be contracted and redefined into a higher-dimensional qudit:

$$\Gamma^{[m,m+1]\{i_m,i_{m+1}\}}_{\alpha_{m-1}\alpha_{m+1}} \equiv \Gamma^{[m]i_m}_{\alpha_{m-1}\alpha_m} \Gamma^{[m+1]i_{m+1}}_{\alpha_m \alpha_{m+1}}. \tag{7}$$

By sequentially contracting adjacent qudits, the state vector representation may be obtained from an MPS representation of a state. Additionally, the reverse operation of Eq. (7) may be performed to decompose the matrices of two combined qudits. This involves rearranging the elements of $\Gamma^{[m,m+1]\{i_m,i_{m+1}\}}_{\alpha_{m-1}\alpha_{m+1}}$ in Eq. (7) and applying a matrix decomposition such as the trivial decomposition

$$M_{ab} = \begin{cases} M_{a\alpha_m} I_{\alpha_m b}, & \dim\{a\} \geq \dim\{b\} \\ I_{a\alpha_m} M_{\alpha_m b}, & \dim\{a\} < \dim\{b\} \end{cases}, \tag{8}$$

where I is the appropriately sized identity matrix, or the rank-revealing singular value decomposition (SVD)

$$M_{ab} = U_{a\alpha_m} S_{\alpha_m} V_{\alpha_m b}, \tag{9}$$

where U and V are unitary matrices and the singular values, the elements of S, are nonnegative reals. The intermediate index α_m is referred to as the bond index of its respective bipartition. By sequentially decomposing qudits into constituent subsystems, an MPS representation of a state represented in the state vector form may be obtained.

Though the trivial decomposition, Eq. (8), is simple to perform, an MPS produced solely from such decompositions on a state vector representation offers no computational advantage in terms of time or space usage. Instead, if the SVD, Eq. (9), is used as the decomposition, the MPS representation might look like

$$\psi_{i_1 i_2 \ldots i_n} = \Gamma^{[1]i_1}_{\alpha_1} \lambda^{[1]}_{\alpha_1} \Gamma^{[2]i_2}_{\alpha_1 \alpha_2} \lambda^{[2]}_{\alpha_2} \cdots \lambda^{[n-1]}_{\alpha_{n-1}} \Gamma^{[n]i_n}_{\alpha_{n-1}}, \tag{10}$$

where each $\lambda^{[m]}$ is the vector of singular values obtained from Eq. (9). If care is taken during the treatment of these singular values [15], a canonical MPS form can be defined such that these $\lambda^{[m]}_{\alpha_m}$ are equal to the Schmidt coefficients [23] across bipartition $[1, \ldots, m] : [m+1, \ldots, n]$. Furthermore, if $\chi^{[m]}$ is the Schmidt rank across this bipartition, $\dim\{\alpha_m\}$ can be set to $\chi^{[m]}$ by only storing values corresponding to nonzero Schmidt coefficients. Consequently, with the Schmidt ranks as a measure of the entanglement within a state, 'less entangled' states in a canonical MPS representation have lower space requirements to store.

To simulate measurements on qudit m of a composite system, the reduced density matrix of m may be obtained from the general MPS expression, Eq. (6),

by tracing over all other qudit subsystems:

$$\rho^{[m]}_{i_m i'_m} = \Gamma^{[1]i_1}_{\alpha_1} \cdots \Gamma^{[m]i_m}_{\alpha_{m-1}\alpha_m} \cdots \Gamma^{[n]i_n}_{\alpha_{n-1}} \times \overline{\Gamma^{[1]i_1}_{\beta_1}} \cdots \overline{\Gamma^{[m]i'_m}_{\beta_{m-1}\beta_m}} \cdots \overline{\Gamma^{[n]i_n}_{\beta_{n-1}}}, \quad (11)$$

where the bars denote complex conjugation. From this, the measurement probabilities of the d_m possible values of m may be read from the diagonal elements of $\rho^{[m]}$. For a canonical MPS in the form of Eq. (10), $\rho^{[m]}$ may instead be obtained locally:

$$\rho^{[m]}_{i_m i'_m} = \left|\lambda^{[m-1]}_{\alpha_{m-1}}\right|^2 \Gamma^{[m]i_m}_{\alpha_{m-1}\alpha_m} \overline{\Gamma^{[m]i'_m}_{\alpha_{m-1}\alpha_m}} \left|\lambda^{[m]}_{\alpha_m}\right|^2, \quad (12)$$

due to unitarity of the resulting matrices of an SVD. Even if an MPS is not in canonical form, the reduced density matrix $\rho^{[m]}$ of qudit m may still be obtained from Eq. (12) if pairwise contractions and SVDs are performed from the ends of the MPS in toward m. We refer to series of such pairwise contractions and SVDs as a sweep. Following a simulated measurement of m, which might partially collapse the entire state's entanglement, sweeps outward from m may be used to propagate this collapse and reduce our space usage.

Finally, applying a single-qudit gate U as a unitary transformation to qudit m in an MPS may be performed as a local operation:

$$\Gamma^{[m]i_m}_{\alpha_{m-1}\alpha_m} \leftarrow U_{i_m i'_m} \Gamma^{[m]i'_m}_{\alpha_{m-1}\alpha_m}.$$

This can be generalised to multiple-qudit gates acting on consecutive qudits in an MPS by contracting them, applying the gate, and then decomposing. SWAP gates may be used to rearrange the order of qudits in an MPS.

4 Entanglement in Shor's Algorithm

Since the space usage of an MPS scales with the amount of entanglement in the state, we should examine the entangling properties of Shor's algorithm in order to determine an efficient embedding of the circuit's $3l$ qubits into an MPS with its inherently linear connectivity. By rewriting Eq. (3) for the state of the upper register following measurement of the lower register R in the form

$$|\psi_{\text{upper}}\rangle \propto |j\rangle + |j+r\rangle + |j+2r\rangle + \cdots, \quad (13)$$

we observe that the α least-significant bits of each state on the right hand side of Eq. (13) are identical, where α is the number of trailing zeroes in the binary representation of r. Therefore, immediately prior to the measurement of R, these α least-significant qubits of the upper register only exhibit entanglement with R and not between themselves or other qubits of the upper register. Also at this point in the circuit, the remaining $2l-\alpha$ most-significant qubits of the upper register exhibit entanglement between themselves and with the lower register R, due to the odd factor $\beta \equiv r/2^\alpha$ of r which cannot be localised to specific qubits.

Following measurement of the lower register, all l qubits in R are now completely separable. Entanglement between the upper register with R is consequently collapsed, leaving the upper register's α least-significant qubits completely separable and its remaining $2l-\alpha$ most-significant qubits only entangled amongst themselves.

To examine some properties of α, we note that r divides $\lambda(N)$, the Carmichael function at N, which is defined as the smallest integer such that $x^{\lambda(N)} \equiv 1 \bmod N$ for all $x \in \mathbf{Z}^*_N$. By the Chinese remainder theorem and Carmichael's theorem,

$$\lambda(N) = \operatorname{lcm}(p-1, q-1) \quad (14)$$

for our odd, squarefree semiprime N. Therefore if d_p is the largest integer such that $2^{d_p}|(p-1)$ and similarly for d_q, then $\max(d_p, d_q)$ is the largest value α can take for a specific N over all choices of a.

Also by the Chinese remainder theorem, uniformly choosing $a \neq 1$ from $\mathbf{Z}^*_N$ is equivalent to independently and uniformly choosing $a_p \in \mathbf{Z}^*_p$ and $a_q \in \mathbf{Z}^*_q$ so that $a_p, a_q \neq 1$. Therefore, by [23, *Lemma A4.12*],

$$\Pr(\alpha = \max(d_p, d_q)) \geq \frac{1}{2} \quad (15)$$

asymptotically for randomly and uniformly chosen $a \neq 1$.

Furthermore, Dirichlet's theorem implies that there is asymptotically a 2^{1-m} probability that $2^m|(p-1)$ for randomly chosen prime p. Therefore, d_p is distributed geometrically such that $\Pr(d_p = m) = 2^{-m}$. If p and q are independently chosen, d_p and d_q are i.i.ds whose maximum has an expected value

$$\mathbf{E}(\max(d_p, d_q)) = \frac{8}{3} \quad (16)$$

by [24, Eq. (2.6)].

This analysis suggests a further partitioning of the upper register. From now, we will refer to A as the set of α least-significant qubits in the upper register and B as the remaining $2l-\alpha$ most-significant qubits of the upper register.

5 MPS Implementation

5.1 Previous approaches

High-level simulations of Shor's algorithm were performed in [14] with a static MPS qudit ordering that places the upper-register qubits on one side of the lower register R, per the example in Fig. 2(a). To briefly summarise this static method, the l qubits of R were kept contracted as a single qudit. With R effectively stored as a 1-dimensional qudit in the state $|1\rangle$, each of the $2l$ upper-register qubits q_i for $0 \leq i < 2l$ is entangled with R by applying the following procedure to each q_i in order of descending i:

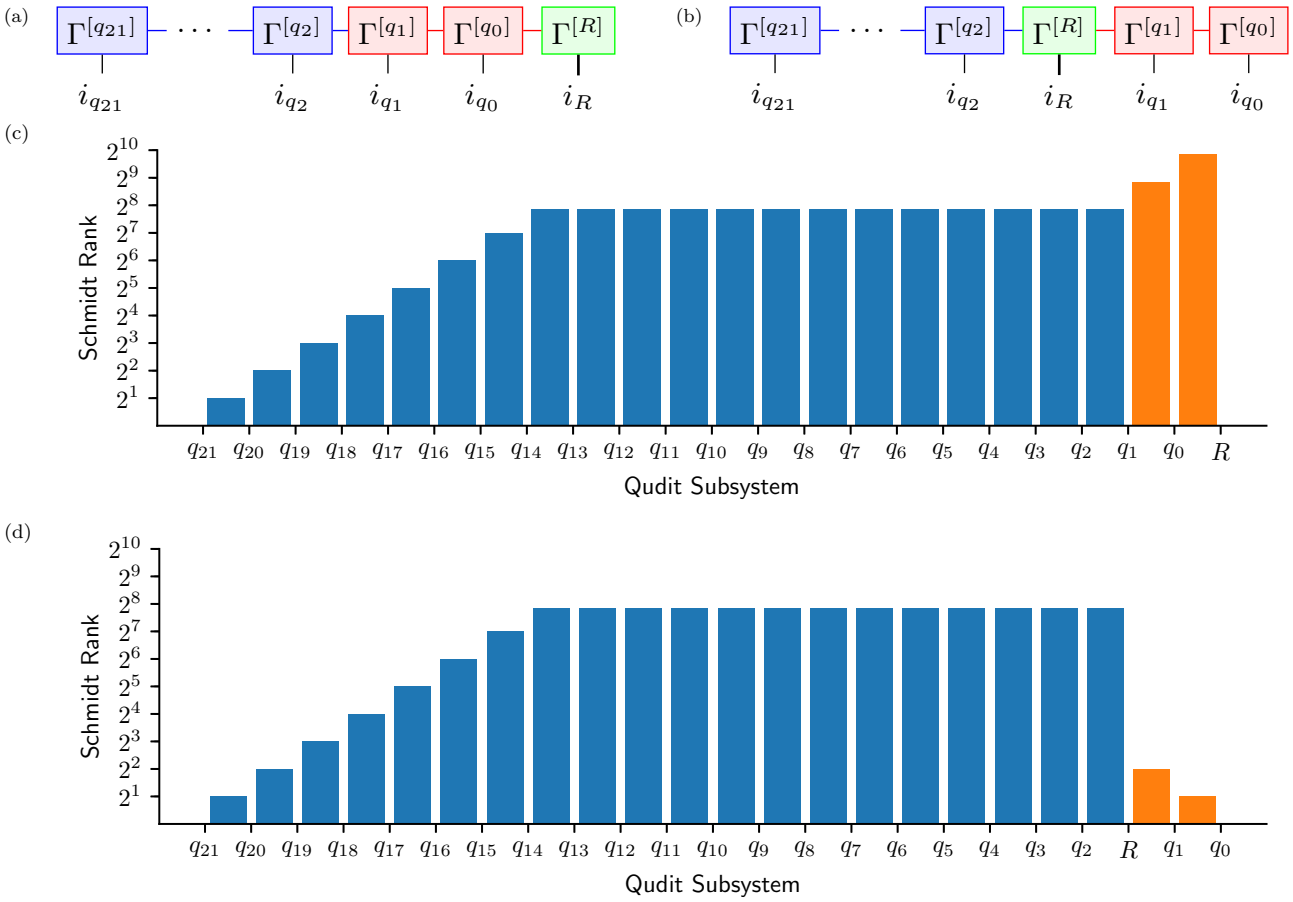

Figure 2: Comparison between MPS qudit layouts for example parameters $l = 11$, $N = 1943$, and $a = 2$, resulting in $r = 924 = 231 \times 2^2$. (a) and (b) show tensor network diagrams [16] for the static and dynamic qudit layouts respectively, where red (blue) tensors correspond to tensors in A (B) and the green tensor represents the lower-register qudit R. (c) and (d) show the Schmidt ranks across each bipartition in the static and dynamic layouts immediately following application of the controlled exponentiated U gates, with significant reductions across the highlighted bipartitions.

- An initially separable qubit q_i is inserted between the previous upper-register qubit and R by altering the bond sizes appropriately and is set to the state $(|0\rangle + |1\rangle)/\sqrt{2}$, representing the starting round of Hadamard gates.
- The gate U^{2^i}, which may be formed by repeated squaring of U, Eq. (1), is applied to R and controlled by q_i. This involves contracting q_i with R beforehand and may require R's current effective dimensionality to be increased.
- This contraction is then decomposed back into qubit q_i and the lower-register qudit R by using the trivial decomposition, Eq. (8). Rank minimisation through a rank-revealing decomposition is not required at this stage since the apparent rank from the trivial decomposition is equal to the Schmidt rank.

With sufficient index book-keeping, these steps may be combined into a single operation per qubit q_i. The effect of these controlled operations between each of the $2l$ upper-register qubits with the lower register R is to quickly initialise an MPS that encodes the state $|\psi_{\text{modexp}}\rangle$ in Eq. (2) with the MPS qudit layout $[B] : [A] : [R]$, defined in Sec. 4. As all upper-register qubits are on one side of the lower-register qudit, applying these controlled gates ensures that the Schmidt ranks at each bipartition are nondecreasing toward the lower-register qudit. This culminates in a rank of r between the least-significant qubit q_0 of the upper register with the lower register R [25], as demonstrated in Eq. (2) and Fig. 2(c). Additionally, R is effectively stored as an r-dimensional qudit at this point.

This MPS is not in canonical form, since the trivial decomposition, Eq. (8), was used instead of the SVD, Eq. (9). As such, nonlocal measurement of the R was simulated by directly calculating its reduced density matrix through Eq. (11), and upon accordingly setting a value for R, entanglement is collapsed to reduce space usage by performing a sweep outward from R. Since the lower register is now separable at this stage, it is removed from the MPS and the remaining upper register of $2l$ qubits encodes the state $|\psi_{\text{upper}}\rangle$ in Eq. (3).

The quantum Fourier transform (QFT) was then

performed in [14] by sequentially contracting qubits together, eventually resulting in a 2^{2l}-dimensional state vector of the upper register. Using this full contraction ultimately limits the size of an instance of Shor's algorithm that may be simulated. However, it does scale well timewise with parallelism since decompositions are not required. Though in that case, for these high-level simulations, the use of a distributed discrete Fourier transform library [26] as a single operation on the fully contracted state vector should result in greater performance.

A separate approach to simulating Shor's algorithm up to the lower-register measurement, making use of a tree tensor network, is detailed in [22]. Due to the strict linear layout of an MPS, this tree tensor network more evenly represents the multipartite entanglement induced by the controlled exponentiated U gates at the cost of a more complicated series of tensor operations when performing basic tasks such as measurement or applying nearest-neighbour gates. While [22] elucidates how such a tree tensor network might be converted to an MPS with a more even entanglement distribution than in [14], we detail how such an MPS might be generated solely through MPS operations.

5.2 Simulation optimisations

Instead of returning a state vector following a simulation of Shor's algorithm, which ultimately negates the space savings made by using the MPS formalism, we performed simulations of Shor's algorithm as a sampling problem where we only wish to sample measurements according to the underlying probability distribution resulting from the circuit. This method mimics the process of obtaining results from an actual quantum computer, and is therefore a task that may be performed by both classical and quantum machines. Furthermore, performing simulations in this way is ideal for MPS since it mitigates the need to contract to a state vector, and as the entanglement collapse following a measurement reduces memory usage.

From the description of Shor's algorithm in Sec. 2, the coefficients in all quantum states prior to the QFT are real. Therefore, we can approximately halve our time and space requirements for this section by storing our MPS elements as real scalars instead of complex scalars of equal precision, and then later converting to complex scalars prior to the QFT. These savings come at an opportune time due to the amount of entanglement before measurement of the lower register R.

The centrepiece of our additional optimisations, however, is to embed the lower-register qudit R within our MPS. Whilst the static approach in the previous section had qudits positioned in the order $[B] : [A] : [R]$ as in Fig. 2(a), this has a significant space disadvantage since A is not entangled to B, but the entanglement between B and R crosses through A and multiplies the Schmidt ranks within A by β. Instead, we have chosen to perform our simulations by positioning our partitions as $[B] : [R] : [A]$ in our MPS as in Fig. 2(b), which allows direct connectivity to R from A and B simultaneously. With this dynamic layout, the size of the MPS matrices for each of the α qubits in A is reduced by a factor of β^2 compared to the static approach as seen by comparing Figs. 2(c) and 2(d), resulting from a factor of β for the left and right bond indices of a matrix. Therefore, the dynamic qudit layout refines classification of the difficulty in simulating Shor's algorithm to include the factors of r, rather than just r per the static layout.

Since simulations should not depend on prior knowledge of the values r, α or β, our layout requires us to dynamically detect the transition from qubits in B to qubits in A when applying the controlled exponentiated U gates. In Sec. 4, we noted that B was entangled with R prior to the lower-register measurement, and that the qubits within B were entangled amongst one another. Therefore, as the controlled exponentiated U gates are sequentially operated on qubits in order of descending significance, the Schmidt ranks with R will temporarily stop increasing (at a value of β) when this amount of entanglement has been reached. Only when we start operating on qubits in A does the entanglement to R begin to increase again, as seen in Fig. 2(c). By detecting this plateau and subsequent rise in the Schmidt ranks, we can detect the boundary between A and B and begin to place qubits from A on the opposite side of R.

At this point, we also note from Eq. (15) that the choice of a has probability at least one half to result in the maximum α permitted by semiprime N. Therefore, classical simulations exceeding a given memory limit may be retried a constant number of times with different values of a before this maximum α is expected to be observed. Furthermore, from Eq. (16), when the choice of a results in the maximum α, A is expected to contain 8/3 qubits across all choices of N. In our dynamic qudit layout where the size of each MPS matrix in A is reduced by a factor of β^2, this may result in significant space savings compared to the static layout.

When sufficiently many controlled exponentiated U gates have been operated to initialise the systems $[B] : [R]$, we perform a right sweep as required for the local measurement of R. As each of the α qubits in A is interacted with R, their Schmidt ranks toward R must sequentially double to reach a lower-register occupancy of $r = \beta \times 2^{\alpha}$, given the contribution of β from B. This is demonstrated in Fig. 2(d). Use of the trivial decomposition Eq. (8) with careful normalisation removes the need for the left sweep across A. R may then be measured using Eq. (12). Sweeps are then performed outward from this site to collapse entanglement involving R and with R now separable, it is removed to leave an MPS consisting of systems

$[B] : [A]$. Since the only remaining entanglement exists between the qubits of B, the highest Schmidt rank within the MPS at this stage is β, and the qubits of A are completely separable at this point.

We then applied the linear nearest-neighbour (LNN) circuit for the QFT to our remaining qubits, as shown in Figs. 1(a) and 1(b). Since each controlled phase gate is immediately followed by a SWAP on the same qubits, we applied these as a combined operation. We note that the structure of the LNN QFT circuit allows us to measure any qubits that no longer require further operations. By performing these measurements as soon as possible and sweeping afterwards, we help mitigate any extra entanglement introduced during the progression of the QFT circuit. The final result is a completely separable MPS, rather than a full state vector, of the upper register corresponding to a value distributed according to Eq. (4).

6 Benchmarks

l	r	α	β	n_{proc}	t_U	t_{meas}	t_{QFT}	t_{total}	Advantage (×)
13	3870	1	1935	4	118	24	288	420	4.5
				16	48	10	105	163	3.2
14	8036	2	2009	4	146	38	585	769	4.6
				16	58	15	205	278	3.3
15	16104	3	2013	4	157	48	970	1175	8.0
				16	63	19	339	421	

Table 1: Benchmarks for simulating Shor's algorithm with our optimisations, with times for the exponentiated U gates, measurement, and QFT listed in seconds. With n_{proc} cores, we simulated the three cases $l = 13$, $N = 8189$, $a = 10$; $l = 14$, $N = 16351$, $a = 2$; and $l = 15$, $N = 32663$, $a = 6$, and compared the total times to the static approach with full state vector contraction presented in [14] to obtain the relative advantage.

l	r	α	β	n_{node}	t_U	t_{meas}	t_{QFT}	t_{total}
16	28140	2	7035	2	1538	353	4290	6181
17	57516	2	14379	24	1694	406	4544	6644
20	479568	4	29973	216	4271	1496	20236	26003

Table 2: Larger instances, this time across multiple nodes of a supercomputer. Times for the various stages are again listed in seconds. Each node has 24 cores and 64 GB of RAM. With n_{node} nodes, we simulated the three cases $l = 16$, $N = 56759$, $a = 2$; $l = 17$, $N = 124631$, $a = 2$; and $l = 20$, $N = 961307$, $a = 5$.

Our simulations were implemented using the Elemental [27] library for distributed-memory linear algebra, mainly due to its divide and conquer SVD [28]. All results were obtained in double precision, with 64 bit real scalars before the quantum Fourier transform and 128 bit complex scalars during. Square process grids were used and elements follow a two-dimensional element-wise cyclic distribution.

We began by simulating Shor's algorithm with the same parameters (l, N, a) benchmarked in [14]. Our results in Table 1 were obtained on a single Intel Xeon E5-2683 v4 with a base core clock of 2.10 GHz, using the Intel Parallel Studio XE suite for compilers and intraprocess BLAS and LAPACK [29], and Open MPI [30] for our MPI implementation. We also limited the space usage for these results to 8 GB of RAM to showcase our optimisations against the 16 GB required in [14]. Despite using a processor of lower clock speed, our implementation appears to result in a significant overall performance increase.

The parameters (l, N, a) chosen in [14] produce values of r equal to its upper bound of $\text{lcm}(p-1, q-1)$ from Eq. (14), since the static layout bases difficulty on the size of r. Our dynamic approach classifies difficulty according to the factors of r, especially the odd factor β of r which is similar across the three cases in Table 1. This is somewhat reflected in our results by our increasing time advantage over the results of [14].

Though our individual timings for the application of the exponentiated U gates and the QFT appear slower than the results of [14], this is due to our t_U being used to record the time to perform the sweep prior to lower-register measurement and us choosing to use the LNN QFT with interleaved measurements respectively. Our time savings made during the lower-register measurement more than makes up for this. Additionally, whilst [14] is able to claim $1/n_{\text{proc}}$ time scaling in their results due to the parallelism of MPS contraction, we relied more on MPS decomposition through the SVD to keep our memory requirements low. It would appear then that the SVD scales differently with n_{proc} since we do not observe $1/n_{\text{proc}}$ time scaling.

To benchmark a truly distributed implementation, we also performed some simulations across multiple nodes on Magnus [31], a Cray XC40 supercomputer with 24 cores at 2.60 GHz and 64 GB of RAM per node. Our results in Table 2 were run using the GNU Compiler Collection suite for compilers, OpenBLAS [32] for intranodal BLAS and LAPACK, and Cray's proprietary MPI implementation.

For our flagship 60-qubit $l = 20$ instance, we chose the parameters $N = 961307$ and $a = 5$. These parameters were specifically chosen to highlight the differences between the static and dynamic methods, by having the maximum period $r = 479568$ permitted by N, but with $\alpha = 4$ higher than expected. Since r here is relatively high, it would be infeasible to simulate this with the static method even if the final MPS contraction was not performed. In our simulation, the $\alpha = 4$ qubits in A significantly decrease the amount of resources required, just from an understanding of the entangling properties of Shor's algorithm. As such, we were able to perform a high-level simulation of sampling a measurement from Shor's algorithm on 60 qubits using a total of 216 nodes, 5184 cores, and 13.824 TB of memory within 8 h. To the best of our knowledge, this is the largest high-level simulation of

Shor's algorithm.

7 Conclusion

We performed classical simulations of Shor's algorithm as a sampling problem using a dynamic MPS qudit layout. Compared to previous approaches that relied on static qudit layouts, our approach better maps the entanglement induced by the circuit for Shor's algorithm onto a system with linear connectivity as represented by an MPS. The use of this dynamic layout also refines classification of the difficulty in simulating Shor's algorithm not only to include the size of the period r, but also its factors. Furthermore, by simulating Shor's algorithm as a sampling problem, we were able to take advantage of measurement to collapse entanglement within the MPS. This reduces space usage during the quantum Fourier transform with respect to contracting MPS matrices into a full state vector.

In particular, we found that asymptotically, on average, the power of 2 that divides r is 8/3. This number of qubits would become completely separable following measurement of the lower register, greatly reducing the simulation difficulty. We also note that instances with semiprime $N = pq$ such that $p - 1$ or $q - 1$ is divisible by a high power of 2 are especially easy to simulate if r does not possess a correspondingly large odd factor.

Our optimisations resulted in significant time and space savings for instances with up to 45 qubits when compared to previous benchmarks using the static qudit layout with full state-vector contraction. Through the use of supercomputing resources, we were able to simulate a 60-qubit instance of Shor's algorithm with high r, rendering it infeasible via the static approach. In terms of the number of qubits, this represents one of the largest simulations of a nontrivial quantum circuit ever performed.

Acknowledgements

This work was supported by the Australian Research Council (ARC) under the Centre of Excellence scheme (Project No. CE110001027). Computational resources were provided by the Pawsey Supercomputing Centre with funding from the Australian Government and the Government of Western Australia.

References

[1] S. Aaronson and A. Arkhipov, in *STOC '11 Proceedings of the Forty-third Annual ACM Symposium on Theory of Computing* (ACM, New York, 2011) pp. 333–342.

[2] M. J. Bremner, R. Jozsa, and D. J. Shepherd, Proc. Royal Soc. A **467**, 459 (2011).

[3] S. Boixo, S. V. Isakov, V. N. Smelyanskiy, R. Babbush, N. Ding, Z. Jiang, M. J. Bremner, J. M. Martinis, and H. Neven, Nat. Phys. **14**, 595 (2018).

[4] S. Bravyi and D. Gosset, Phys. Rev. Lett. **116**, 250501 (2016).

[5] E. Pednault, J. A. Gunnels, G. Nannicini, L. Horesh, T. Magerlein, E. Solomonik, and R. Wisnieff, (2017), arXiv:1710.05867 .

[6] P. W. Shor, SIAM J. Comput. **26**, 1484 (1997).

[7] R. L. Rivest, A. Shamir, and L. Adleman, Commun. ACM **21**, 120 (1978).

[8] T. Dierks and E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.2*, RFC 5246 (RFC Editor, 2008).

[9] D. J. Bernstein, "Introduction to post-quantum cryptography," in *Post-Quantum Cryptography*, edited by D. J. Bernstein, J. Buchmann, and E. Dahmen (Springer Berlin Heidelberg, Berlin, Heidelberg, 2009) pp. 1–14.

[10] L. M. K. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood, and I. L. Chuang, Nature **414**, 883 (2001).

[11] B. P. Lanyon, T. J. Weinhold, N. K. Langford, M. Barbieri, D. F. V. James, A. Gilchrist, and A. G. White, Phys. Rev. Lett. **99**, 250505 (2007).

[12] C.-Y. Lu, D. E. Browne, T. Yang, and J.-W. Pan, Phys. Rev. Lett. **99**, 250504 (2007).

[13] E. Lucero, R. Barends, Y. Chen, J. Kelly, M. Mariantoni, A. Megrant, P. O'Malley, D. Sank, A. Vainsencher, J. Wenner, T. White, Y. Yin, A. N. Cleland, and J. M. Martinis, Nat. Phys. **8**, 719 (2012).

[14] D. S. Wang, C. D. Hill, and L. C. L. Hollenberg, Quantum Inf. Process. **16**, 176 (2017).

[15] G. Vidal, Phys. Rev. Lett. **91**, 147902 (2003).

[16] R. Orús, Ann. Phys. (N. Y.) **349**, 117 (2014).

[17] U. Schollwöck, Ann. Phys. (N. Y.) **326**, 96 (2011).

[18] G. Vidal, Phys. Rev. Lett. **98**, 070201 (2007).

[19] K. J. Woolfe, C. D. Hill, and L. C. L. Hollenberg, Quantum Inf. Comput. **17**, 1 (2017).

[20] F. Verstraete and J. I. Cirac, (2004), arXiv:cond-mat/0407066 .

[21] G. Vidal, Phys. Rev. Lett. **99**, 220405 (2007).

[22] E. Dumitrescu, Phys. Rev. A **96**, 062322 (2017).

[23] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, 10th Anniversary ed. (Cambridge University Press, Cambridge, 2010).

[24] W. Szpankowski and V. Rego, Computing **43**, 401 (1990).

[25] R. Orús and J. I. Latorre, Phys. Rev. A **69**, 052308 (2004).

[26] M. Frigo, SIGPLAN Not. **39**, 642 (2004).

[27] J. Poulson, B. Marker, R. A. van de Geijn, J. R. Hammond, and N. A. Romero, ACM Trans. Math. Softw. **39**, 13:1 (2013).

[28] E. R. Jessup and D. C. Sorensen, SIAM J. Matrix Anal. Appl. **15**, 530 (1994).

[29] E. Anderson, Z. Bai, C. Bischof, L. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users' Guide*, Third ed. (Society for Industrial and Applied Mathematics, Philadelphia, 1999).

[30] R. L. Graham, T. S. Woodall, and J. M. Squyres, in *Parallel Processing and Applied Mathematics: 6th International Conference, PPAM 2005, Poznań, Poland, September 11-14, 2005, Revised Selected Papers*, edited by R. Wyrzykowski, J. Dongarra, N. Meyer, and J. Waśniewski (Springer Berlin Heidelberg, Berlin, Heidelberg, 2006) pp. 228–239.

[31] "The Pawsey Supercomputing Centre," `https://www.pawsey.org.au/` (2017).

[32] Q. Wang, X. Zhang, Y. Zhang, and Q. Yi, in *SC '13 Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis* (ACM, New York, 2013) pp. 25:1–25:12.