Quantum phase estimation with optimal confidence interval using three control qubits

Kaur Kristjuhan[1,2,3] and Dominic W. Berry[2]

[1]Molecular Quantum Solutions ApS, Blegdamsvej 17, 2100 Copenhagen, Denmark
[2]School of Mathematical and Physical Sciences, Macquarie University, Sydney, NSW 2109, Australia
[3]Sydney Quantum Academy, Sydney, NSW, Australia

arXiv:2601.16474v3 [quant-ph] 7 Aug 2026

Quantum phase estimation is an important routine in many quantum algorithms, particularly for estimating the ground state energy in quantum chemistry simulations. This estimation involves applying powers of a unitary to the ground state, controlled by an auxiliary state prepared on a control register. In many applications the goal is to provide a confidence interval for the phase estimate, and optimal performance is provided by a discrete prolate spheroidal sequence. We show how to prepare the corresponding state in a far more efficient way than prior work. We find that a matrix product state representation with a bond dimension of 4 is sufficient to give a highly accurate approximation for all dimensions tested, up to 2^{24}. This matrix product state can be efficiently prepared using a sequence of simple three-qubit operations. When the dimension is a power of 2, the phase estimation can be performed with only three qubits for the control register, making it suitable for early-generation fault-tolerant quantum computers with a limited number of logical qubits.

1 Introduction

Quantum phase estimation (QPE) is a widely used subroutine in quantum computing and is indispensable for many important applications such as factorisation [1] and ground state energy estimation in quantum chemistry [2, 3, 4, 5, 6, 7]. The purpose of QPE is to estimate the phase ϕ of an eigenvalue $e^{i\phi}$ of a unitary operator U. The algorithm for QPE [8] invokes the application of controlled powers of U on a quantum state many times.

The estimate can be improved to arbitrary precision by using larger powers of U. In the case of factoring, these powers may be performed with similar complexity to U, so it is convenient to use the simple textbook form of QPE with large powers of U. In contrast, for ground state energy estimation, U is an operator that encodes the Hamiltonian describing the physical system of interest. In that context, higher powers of U correspond to Hamiltonian simulations with longer evolution time, or more applications of the qubitised operator [9, 10]. This means that the complexity increases linearly in the power of U, and these powers constitute the majority of the cost of the algorithm.

This motivates us to find ways to modify the QPE algorithm to maximise the accuracy of the phase estimate with minimal applications of U. The first step of the QPE algorithm,

Kaur Kristjuhan: kaur@mqs.dk

as described in most textbooks [8], is to prepare an equal superposition state on a control register. Replacement of this step with preparation of a different state is one way to improve the accuracy of QPE.

A closely related modification is used in classical signal processing. Assigning equal weight to every sample in a dataset is called a rectangular window, and it is well known to cause problems in spectral analysis [11]. Uniform weighting causes the discrete Fourier transform to spread energy across neighbouring frequency bins, rather than concentrating it at the true frequency. The solution is to replace the rectangular window with a smoother window function that suppresses this spectral leakage. Precision metrology faces the same issue, and gravitational wave observatories such as LIGO taper the edges of their time series data before Fourier analysis for this reason [12, 13]. Quantum phase estimation relies on an analogous Fourier transform between the control register and the estimated phase, and so encounters the same leakage issue. We can adapt these classical tapering techniques to the quantum setting by applying a smooth amplitude profile across the control register. This acts as a quantum window function, which concentrates the measured phase distribution around the true eigenphase.

If the goal is to perform measurements with minimum mean-square error, then the control register can be in a state with amplitudes according to a sine function, which is easily prepared [14, 15, 16, 17]. This provides the optimal average case accuracy for QPE, but does not provide any guarantees of the estimate being within a desired error of the true answer. One example where this is especially relevant is ground state energy estimation, in the case where we are unable to precisely prepare the ground state. In such a situation, the phase measurement may erroneously probe the energy of an excited state instead. To isolate the ground state from excited states, we can take multiple samples and select the minimum value. However, this strategy requires a careful balance: under-sampling risks missing the ground state entirely, while over-sampling increases the likelihood of underestimating the true energy.

The alternative is to optimise for the confidence interval of the phase estimate instead. This minimises the probability of obtaining a result with large error [18]. The use of a discrete prolate spheroidal sequence (DPSS) state in the control register has been shown to minimise the width of the confidence interval [19] and also enables QPE to be used as a viable technique for filtering [20].

The DPSS state shares a direct physical analogue with optimal apodisation in classical optics [21]. Just as high-precision optical systems use prolate spheroidal transmission masks to smoothly taper a beam's intensity profile and maximise the concentration of focused light by suppressing diffraction rings [22, 23], the DPSS state maximally concentrates the quantum measurement probability around the true phase.

Due to the difficulty of calculating the DPSS [24], some works have opted to use the Kaiser window state instead [25], which approximates the DPSS state with good accuracy and has asymptotic behaviour that can be analytically described [26, 18]. Both the DPSS state and Kaiser window state are considerably more complicated than the sine window from Ref. [16, 17], so a new state preparation scheme is needed.

For preparing the DPSS state, variations of Grover-Rudolph state preparation [27, 28] or state preparation based on the quantum singular value transformation are commonly proposed methods [29, 30, 31]. Reference [32] also describes a specialised state preparation procedure for the DPSS state, which involves discarding low amplitude components of the DPSS window in frequency space to enable an approximate, but less costly preparation. For all of these approaches, the number of non-Clifford gates scales polynomially in the dimension (so exponentially in the number of qubits). Moreover, all qubits for the state

are prepared at once, and many more additional qubits may be used in schemes based on quantum read-only memory (QROM) [28, 18]. These features are particularly undesirable for early generations of fault-tolerant quantum computers, where qubit counts are limited and gate operations are slow.

We propose a more resource-efficient state preparation procedure to prepare a DPSS state of any dimension based on a matrix product state (MPS) representation. The implementation cost of the state preparation grows linearly with the number of qubits in the control register and enables the user to perform the minimal number of controlled applications of U to achieve a target confidence interval width. Furthermore, if the dimension of the DPSS state is a power of two, we are able to exploit the structure of the MPS preparation circuit to sequentially measure and recycle the control register qubits used during phase estimation so that only three ancilla qubits are required for implementation.

The remainder of this paper is structured as follows. Section 2 explains the necessary background theory of QPE, DPSS states, and MPS approximations. In Section 3 we provide numerical evidence that using an MPS approximation with bond dimension 4 is suitable for preparing a DPSS state and provides the optimal confidence interval at any desired precision for confidence levels used in practical applications. Finally, in Section 4 we propose an efficient gate implementation for preparing an MPS with real-valued amplitudes, provide explicit quantum circuits comprised of single qubit rotations and Clifford gates, and demonstrate reduced implementation cost compared to prior work.

2 Quantum phase estimation with discrete prolate spheroidal sequence states

2.1 Overview of quantum phase estimation

In this section, we will provide an overview of the QPE algorithm. This mainly reviews prior work, but we also provide some more explicit explanations of how a DPSS state can be used to enhance the performance of the algorithm and discuss alternative implementations where the dimension of the state is not necessarily a power of two.

The quantum phase estimation algorithm begins with the premise that we have a system register where we have already prepared an eigenstate $|\phi\rangle$ of an operator U. The task is to estimate the phase of the eigenvalue ϕ corresponding to that state, meaning that $U|\phi\rangle = e^{i\phi}|\phi\rangle$. The first step of the algorithm is to prepare a D-dimensional state $|\psi\rangle = \sum_{k=0}^{D-1} \gamma_k |k\rangle$ on a control register comprising n qubits. In the textbook version of QPE [8], the amplitudes are equal and D is a power of two.

The next step is to apply the operator U^k to the system register, where k is controlled on each value of k in the control state. The resulting state is

$$|\psi\rangle|\phi\rangle \mapsto \sum_{k=0}^{D-1} \gamma_k |k\rangle U^k |\phi\rangle = \sum_{k=0}^{D-1} \gamma_k e^{i\phi k} |k\rangle |\phi\rangle . \tag{1}$$

Instead of explicitly performing U^k controlled on each value of k, the most common implementation is to apply powers of U that are powers of two, each controlled on just one of the qubits in the control register. This saves considerable resources if U^k can be implemented with lower cost than k applications of U. This is true in the context of factoring [1], but not for ground state energy estimation problems, where the cost of U^k is linear in k at best, as k corresponds to the simulation time in Hamiltonian simulation algorithms [33].

One way to reduce the cost of this step is to opt for a smaller value of D, at the expense of the accuracy of the result. To provide the most flexibility in this choice, it

would ideally be possible to select any value of D which need not be a power of two. This entails modifying $|\psi\rangle$ accordingly and then performing a unary iteration [16] procedure on the control register [34]. During each step of the unary iteration, the operator U would be applied to the system register, controlled on an ancilla qubit, which is initialised in the $|1\rangle$ state, as shown in Figure 1. The unary iteration then controls the flip of the ancilla, meaning that if the control register is in the state $|k\rangle$, then the ancilla is flipped and U will have been applied to the system register k times. This procedure achieves the desired effect of applying U^k to the system register, using a total of $D-1$ applications of controlled U, where D can be any integer. It is also possible to increase the precision by controlling between U and $U^\dagger$ [16]. For U a qubitised operator [9, 10], the unary iteration can instead be used to control the reflection, and the extra ancilla qubit is not needed [34].

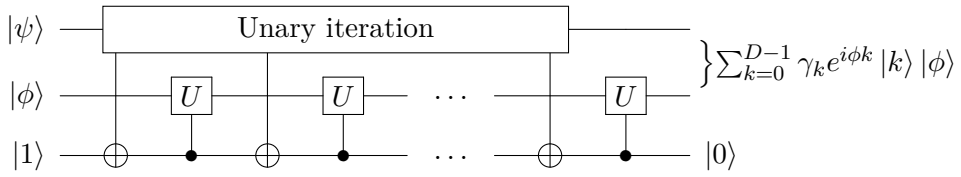

Figure 1: Variant of the first steps of the QPE algorithm with $D-1$ applications of controlled unitary operations, where D may be any positive integer.

The final step of QPE is to apply an inverse quantum Fourier transform (QFT) to the control register and perform a measurement in the computational basis. The state before measurement is

$$\sum_{k=0}^{D-1}\gamma_k e^{i\phi k}|k\rangle|\phi\rangle \overset{\mathrm{QFT}^{-1}}{\mapsto} \frac{1}{\sqrt{N}}\sum_{k=0}^{D-1}\sum_{l=0}^{N-1}\gamma_k e^{ik(\phi-2\pi l/N)}|l\rangle|\phi\rangle = \sum_{l=0}^{N-1}\Gamma\left(\phi-\frac{2\pi l}{N}\right)|l\rangle|\phi\rangle\,, \quad (2)$$

where $N=2^n \geq D$ and we have defined

$$\Gamma(x) \equiv \frac{1}{\sqrt{N}}\sum_{k=0}^{D-1}\gamma_k e^{ikx}. \quad (3)$$

The value of N is chosen as a power of 2, because that enables the inverse QFT to be performed efficiently. Based on the measurement, we obtain an estimate of ϕ, which we denote $\hat{\phi}$. A commonly used procedure is to set the estimate as $\hat{\phi}=2\pi l/N$ for a measurement outcome l, which occurs with probability $|\Gamma(\phi-2\pi l/N)|^2=|\Gamma(\phi-\hat{\phi})|^2$.

A disadvantage of doing so is that it restricts our phase estimates to a discrete set of outcomes. This induces an error in our estimate that depends on how close the true value of ϕ is to the values in this set. This feature is undesirable because the true value of ϕ is unknown and in the worst case scenario where ϕ is precisely between two consecutive multiples of $2\pi/N$, the phase estimation would always yield an error of at least π/N, resulting in larger average error than for other values of ϕ. This method of performing the phase estimation complicates the error analysis, because the performance of QPE averaged over all possible values of ϕ can be different from the performance of QPE for a particular unknown ϕ.

A known solution to this is to add a randomly chosen, known phase $\tilde{\phi}$ to ϕ within the QPE circuit, such that $-\pi/N<\tilde{\phi}\leq\pi/N$ [35]. We explain this approach below to clarify how it can be applied in practice. The probability density function of the continuous variable $\tilde{\phi}$ is a uniform distribution. We will use $p(\cdot)$ to denote probability densities of continuous variables and $\Pr(\cdot)$ to denote probabilities of discrete outcomes, with the

argument specifying the variable under consideration in each instance. The probability density of $\tilde{\phi}$ is

$$p(\tilde{\phi}) = \begin{cases} \frac{N}{2\pi} \text{ for } -\pi/N < \tilde{\phi} \leq \pi/N \\ 0 \text{ elsewhere} \end{cases}. \tag{4}$$

Implementing this phase shift can be achieved by inserting a phase shift gate on each qubit in the control register after the state preparation step, where the applied phase is $2^j\tilde{\phi}$ with j an integer incremented by 1 for each qubit in the control register, starting from $j = 1$ for the least significant qubit and ending with $j = n$ for the most significant qubit. The range we have defined for $\tilde{\phi}$ ensures that $-\pi < 2^j\tilde{\phi} \leq \pi$ for all values of j. After doing this, the state on the control register is $\sum_{k=0}^{D-1} \gamma_k e^{i(\phi+\tilde{\phi})k} |k\rangle |\phi\rangle$, which becomes $\sum_{l=0}^{N-1} \Gamma\left(\phi + \tilde{\phi} - \frac{2\pi l}{N}\right) |l\rangle |\phi\rangle$ after applying the inverse quantum Fourier transform.

Proceeding with the measurement, we obtain a measurement outcome l with the conditional probability

$$\Pr(l|\phi, \tilde{\phi}) = \left|\Gamma\left(\phi + \tilde{\phi} - \frac{2\pi l}{N}\right)\right|^2. \tag{5}$$

We can define a continuous variable y to express the associated conditional probability density function of the discrete variable l:

$$p(y|\phi, \tilde{\phi}) \equiv \sum_{l=0}^{N-1} \Pr(l|\phi, \tilde{\phi})\delta(y - l). \tag{6}$$

We can now define the estimate of ϕ as a continuous variable $\hat{\phi} = 2\pi y/N - \tilde{\phi}$, where the subtraction of $\tilde{\phi}$ is accounting for the adjustment made in the circuit. Inserting $y = (\hat{\phi} + \tilde{\phi})N/2\pi$ into the previous expression yields the conditional probability density of $\hat{\phi}$:

$$p(\hat{\phi}|\phi, \tilde{\phi}) = \frac{N}{2\pi} \sum_{l=0}^{N-1} \Pr(l|\phi, \tilde{\phi})\, \delta\left(\frac{(\hat{\phi} + \tilde{\phi})N}{2\pi} - l\right), \tag{7}$$

which has been multiplied by a factor of $N/2\pi$ to ensure normalisation of the probability density upon integrating over the new variable, as $dy = (N/2\pi)d\hat{\phi}$. The marginal probability density of $\hat{\phi}$, obtained by averaging all possible values of $\tilde{\phi}$, is

$$\begin{aligned} p(\hat{\phi}|\phi) &= \int_{-\infty}^{\infty} p(\hat{\phi}|\phi, \tilde{\phi})p(\tilde{\phi})\, d\tilde{\phi} = \frac{N}{2\pi} \sum_{l=0}^{N-1} \int_{-\infty}^{\infty} \Pr(l|\phi, \tilde{\phi})\, \delta\left(\frac{(\hat{\phi} + \tilde{\phi})N}{2\pi} - l\right) p(\tilde{\phi})\, d\tilde{\phi} \\ &= \sum_{l=0}^{N-1} \Pr(l|\phi, \tilde{\phi})p(\tilde{\phi})\Big|_{\tilde{\phi}=2\pi l/N-\hat{\phi}} = |\Gamma(\phi - \hat{\phi})|^2 \sum_{l=0}^{N-1} p(\tilde{\phi})\Big|_{\tilde{\phi}=2\pi l/N-\hat{\phi}}, \end{aligned} \tag{8}$$

where we have used the property of the Dirac delta function that if $g(x) = 0$ only for $x = x_0$, then

$$\int_{-\infty}^{\infty} f(x)\delta(g(x))dx = \frac{f(x_0)}{g'(x_0)}. \tag{9}$$

Regardless of the value of $\hat{\phi}$, Eq. (4) ensures that the expression $p(\tilde{\phi})|_{\tilde{\phi}=2\pi l/N-\hat{\phi}}$ has the value $N/2\pi$ for exactly one value of l and is zero elsewhere, so

$$\sum_{l=0}^{N-1} p(\tilde{\phi})\Big|_{\tilde{\phi}=2\pi l/N-\hat{\phi}} = \frac{N}{2\pi}, \tag{10}$$

and Eq. (8) reduces to

$$p(\hat{\phi}|\phi) = \frac{N}{2\pi}|\Gamma(\phi - \hat{\phi})|^2. \tag{11}$$

The probability density of the error $\Delta\phi \equiv \phi - \hat{\phi}$ is therefore

$$p(\Delta\phi) = \frac{N}{2\pi}|\Gamma(\Delta\phi)|^2, \tag{12}$$

which is independent of the value of ϕ, as desired. The confidence level $1-\delta$ with which the true value ϕ lies within a confidence interval of width $2d$ centred around the estimate $\hat{\phi}$ can be calculated as

$$1 - \delta = \Pr[-d \leq \Delta\phi \leq d] = \int_{-d}^{d} p(\Delta\phi) d\Delta\phi = \frac{N}{2\pi}\int_{-d}^{d} |\Gamma(\Delta\phi)|^2 d\Delta\phi. \tag{13}$$

For this analysis we require $0 \leq d \leq \pi$, because the confidence interval is defined on the circle. Determining the coefficients γ_k that maximise the value of this expression has been studied under the name of the spectral concentration problem, where the objective is to find a bandlimited sequence with maximal concentration [36]. For the discrete case, the sequence $\{\gamma_k\}$ which solves this problem is the first DPSS [24]. Each DPSS can be defined as a solution to the eigenvalue problem $S(d)\,|\psi\rangle = \lambda(d)\,|\psi\rangle$, where

$$S(d) \equiv \frac{d}{\pi}\sum_{j=0}^{D-1}\sum_{k=0}^{D-1} \frac{\sin(d(j-k))}{d(j-k)}\,|j\rangle\,\langle k|\,. \tag{14}$$

The first DPSS corresponds to the solution with the largest eigenvalue, which is equal to the confidence level $1-\delta$, and throughout the paper we refer to the DPSS state as the corresponding eigenstate. The DPSS state therefore provides the largest possible confidence level for a given confidence interval width, or conversely, achieves a desired confidence level at the tightest possible confidence interval. The next sections focus on how to efficiently prepare the DPSS state on the control register using a matrix product state approximation.

2.2 Matrix product states

A matrix product state $|\psi_{\text{MPS}}\rangle$ for n subsystems describes a state where a set of matrices $W^{[k]}$ are associated with each subsystem k and the wavefunction of the total system is expressed using a product of these matrices:

$$|\psi_{\text{MPS}}\rangle = \sum_{s_1,\ldots,s_n} \text{Tr}\left(W_{s_1}^{[1]} W_{s_2}^{[2]} \ldots W_{s_n}^{[n]}\right)|s_1 \ldots s_n\rangle\,, \tag{15}$$

where each $W_{s_k}^{[k]}$ is a matrix prescribed to subsystem k in the state $|s_k\rangle$. The number of columns in one matrix must correspond to the number of rows in the next matrix so that matrix multiplication is properly defined. This number shared between two neighbouring matrices is called the bond dimension χ. The indices that span the rows and columns of these matrices are called virtual indices α_k, to distinguish them from physical indices s_k, which describe the subsystems. An MPS with bond dimension χ is an MPS where χ is the maximum bond dimension among all neighbouring pairs of matrices. If the subsystems are qubits, then any quantum state can be expressed exactly as an MPS with bond dimension $2^{\lfloor n/2 \rfloor}$ [37, 38].

If we restrict χ to a fixed value less than $2^{\lfloor n/2 \rfloor}$, then only a subset of n-qubit states can be represented. However, an arbitrary state can be approximated by choosing the closest state from this subset. The quality of the approximation depends on both the target state and the bond dimension. The state that provides the best approximation from this subset can be found with an algorithm based on singular value decomposition (SVD) [38]. This can be applied in different ways, producing different representations of the same state [38, 37]. We are most interested in representations where the matrices $W_{s_k}^{[k]}$ are unitaries or isometries that can be implemented directly in a quantum circuit.

To implement each $W_{s_k}^{[k]}$ as an operator, these matrices are first calculated classically. Then, a procedure for unitary synthesis [39, 40, 41, 42, 43, 44] or isometry synthesis [18, 28, 45, 46] is used to generate a quantum circuit that implements these matrices. Increasing the bond dimension increases the amount of classical data that needs to be incorporated into the circuit, thereby incurring a T-gate cost scaling as $\mathcal{O}(\chi^{3/2})$ [47]. The number of qubits that each operator acts on also scales logarithmically with bond dimension, further increasing the size and complexity of the circuit as bond dimension is increased. For these reasons, it is of practical interest to use MPS approximations of states that have minimal bond dimension while providing sufficient accuracy.

2.3 Preparation of DPSS states

To prepare a quantum state $|\psi\rangle$ that spans n qubits as an MPS, we can follow the procedure described in Ref. [38], which involves classically computing successive applications of reshaping and singular value decomposition (SVD) of the state. In this section we explain this procedure in detail, showing how the state preparation quantum circuit is constructed from the target state. A small-scale example is provided in Appendix C.

We start with the amplitude vector $\psi_{s_1 \ldots s_n}$ of the target state

$$|\psi\rangle = \sum_{s_1 \ldots s_n} \psi_{s_1 \ldots s_n} |s_1 \ldots s_n\rangle \,, \tag{16}$$

and reshape it into a rectangular matrix of dimension $2^{n-1} \times 2$, which we denote by $\psi_{s_1 \ldots s_{n-1}, s_n}$ (the comma separates the row index from the column index). For this state, s_1 refers to the most significant qubit and s_n refers to the least significant qubit. Performing an SVD on this matrix yields $\psi_{s_1 \ldots s_{n-1}, s_n} = \sum_{\alpha_{n-1}} \psi_{s_1 \ldots s_{n-1}, \alpha_{n-1}} V^{\dagger[n]}_{\alpha_{n-1}, s_n}$, where $V^{\dagger[n]}$ has orthonormal rows containing the right singular vectors and $\psi_{s_1 \ldots s_{n-1}, \alpha_{n-1}}$ is the remaining part of the SVD. We then repeat this process, each time reshaping the remaining part so that one physical index becomes a column index

$$\psi_{s_1 \ldots s_k, \alpha_k} \to \psi_{s_1 \ldots s_{k-1}, s_k \alpha_k} = \sum_{\alpha_{k-1}} \psi_{s_1 \ldots s_{k-1}, \alpha_{k-1}} V^{\dagger[k]}_{\alpha_{k-1}, s_k \alpha_k}. \tag{17}$$

After performing this procedure for all sites, we obtain an MPS in the right canonical form

$$\psi_{s_1 \ldots s_n} = \sum_{\alpha_1 \ldots \alpha_{n-1}} V^{\dagger[1]}_{s_1 \alpha_1} V^{\dagger[2]}_{\alpha_1, s_2 \alpha_2} \cdots V^{\dagger[n-1]}_{\alpha_{n-2}, s_{n-1} \alpha_{n-1}} V^{\dagger[n]}_{\alpha_{n-1}, s_n}, \tag{18}$$

where $V^{\dagger}_{s_1 \alpha_1}$ is a row vector instead of a matrix. Clearly, this is a matrix product state with $W^{[k]}_{s_k} = V^{\dagger[k]}_{\alpha_{k-1}, s_k \alpha_k}$. For real-valued amplitudes, we can make use of the trace property $\mathrm{Tr}(X) = \mathrm{Tr}(X^\dagger)^*$ to rewrite the amplitudes as

$$\begin{aligned} \psi_{s_1 \ldots s_n} = \psi^*_{s_1 \ldots s_n} &= \mathrm{Tr}\left(W^{[1]}_{s_1} W^{[2]}_{s_2} \ldots W^{[n]}_{s_n} \right)^* = \mathrm{Tr}\left(W^{\dagger[n]}_{s_n} \ldots W^{\dagger[2]}_{s_2} \ldots W^{\dagger[1]}_{s_1} \right) \\ &= \sum_{\alpha_1 \ldots \alpha_{n-1}} V^{[n]}_{s_n, \alpha_{n-1}} V^{[n-1]}_{\alpha_{n-1} s_{n-1}, \alpha_{n-2}} \cdots V^{[2]}_{\alpha_2 s_2, \alpha_1} V^{[1]}_{\alpha_1 s_1}. \end{aligned} \tag{19}$$

Each V now has orthonormal columns and is square or has more rows than columns. The only exception is $V^{[1]}$, which is a column vector, but we can redefine it as an isometry that prepares the two-qubit state described by that column $V^{[1]} \mapsto V^{[1]}(|0\rangle \otimes |0\rangle)$. Based on each $V^{[k]}$, it is also convenient to construct an operator $M^{[k]}$, obtained by swapping indices internally within rows and columns as shown in Figure 2. Finally, we can prepare the matrix product state by sequentially synthesising each isometry $M^{[k]}$ and applying it to the $|0\rangle^{\otimes n}$ state, as shown in Figure 3. Each individual operator is applied to at most $1 + \log_2 \chi$ qubits.

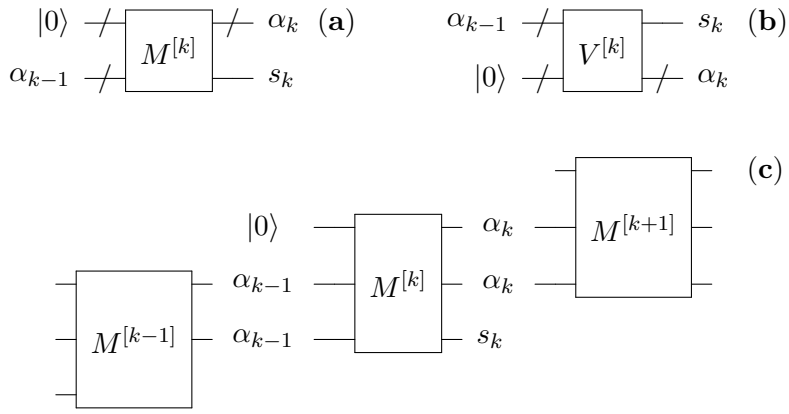

Figure 2: Diagram (a) illustrates an operator $M^{[k]}$, which can be constructed by reordering the rows and columns of $V^{[k]}$, shown in diagram (b). The bottommost qubit is the most significant qubit. The slashes on the lines indicate that the number of qubits needed varies according to the dimension needed to span the corresponding index. Diagram (c) shows how an operator $M^{[k]}$ connects to neighbouring operators in an MPS. For $\chi = 4$, most operators in the MPS will have two qubits indexing α_{k-1} and α_k, and one qubit in the state $|0\rangle$ needed to match the dimension on either side.

The fact that we prepare the state starting from the most significant qubit is a consequence of choosing to generate an MPS in the right-canonical form. Using the left-canonical form produces a circuit which prepares the state starting from the least significant qubit [48]. Using the isometries $M^{[k]}$ as opposed to the isometries $V^{[k]}$ has enabled us to match the most significant qubit in the register to the index s_1, which indexes the most significant qubit of the target state. This ensures that all qubits in the circuit diagram are represented with horizontal lines, with no need for ancilla qubits or vertical rerouting to match indices between operators. Using this specific construction of the MPS to approximate a DPSS for QPE is also beneficial because it provides compatibility with the semi-classical quantum Fourier transform.

2.4 The semi-classical Fourier transform

The semi-classical Fourier transform is a variant of the quantum Fourier transform that uses measurements and classically controlled phase shifts [49]. This can be beneficial because once a qubit is measured, it can be recycled for other purposes. Using this in QPE necessitates that the most significant qubit of the control state is measured first, yielding the least significant bit of the phase estimate. In Figure 4 we draw the first part of the circuit for performing QPE, incorporating the MPS preparation circuit of Figure 3 and the semi-classical Fourier transform implemented with sequential measurements and classically controlled phase shift gates. The phase shift gates $P(2^k\tilde{\phi})$ that are applied immediately

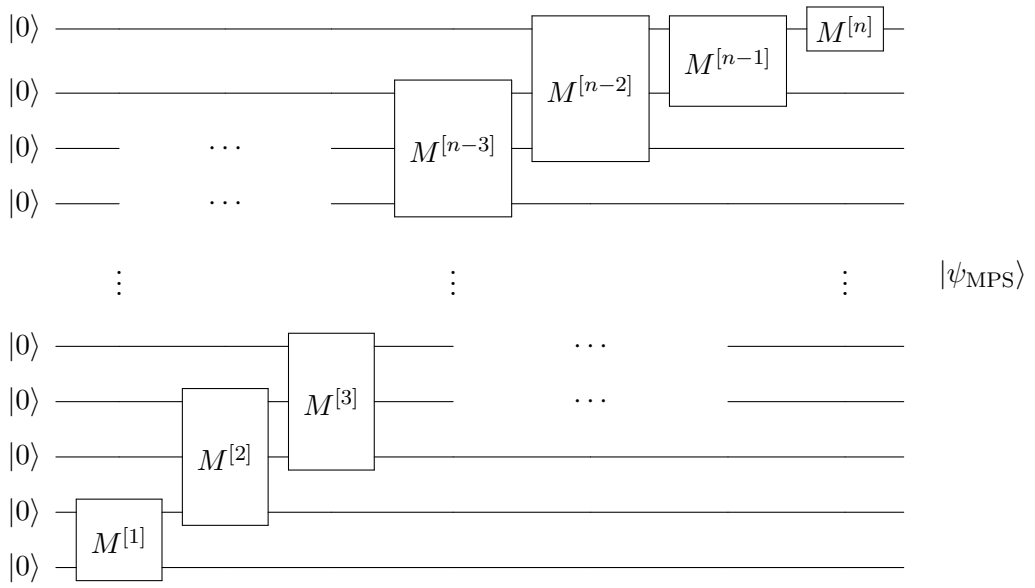

Figure 3: Quantum circuit for preparing an MPS with $\chi = 4$. All operators between $M^{[2]}$ and $M^{[n-2]}$ act on three consecutive qubits. The bottommost qubit is the most significant qubit.

after each of the controlled U operators implement the addition of the random phase discussed in Section 2.1.

The circuit in Figure 4 uses an MPS with $\chi = 4$, meaning each $M^{[k]}$ acts on at most three qubits. The circuit has structure analogous to the circuits of Ref. [17] which use an MPS with $\chi = 2$. The dimension of the state that the MPS is approximating is $D = 2^n$, but instead of allocating n qubits to a control register and preparing the entire state on it at once, we can delay the application of each $M^{[k]}$ until a previously used qubit is made available after a measurement. The pattern of measuring out a qubit before recycling it to apply the next operator in the MPS preparation continues until all bits of the phase estimate have been obtained through measurements. The procedure for determining the classically controlled phase is described in detail in Ref. [49], which we will not repeat here.

2.5 Preparation of MPS with arbitrary dimension

In Section 2.1 we explained how it can be beneficial to prepare a state which has a dimension that is not a power of two. Here we will explain how to prepare an MPS approximation of such a state. First, we follow the procedure described in Section 2.3 but use a different target state which is a padded version of the original one. Namely, if we have a D-dimensional DPSS state $|\psi_{\mathrm{DPSS}}\rangle = \sum_{k=0}^{D-1} \gamma_k |k\rangle$ that we are preparing on an $n = \log_2 N = \lceil \log_2 D \rceil$ qubit register, then we construct

$$|\psi_{\mathrm{DPSS}}\rangle' = \sqrt{\frac{1}{\lambda}} \left(\sum_{j=0}^{P-1} \mu_j |j\rangle + \sum_{k=P}^{P+D-1} \gamma_k |k\rangle + \sum_{l=P+D}^{N-1} \nu_l |l\rangle \right), \tag{20}$$

where P is an integer which determines how the padding is distributed on either side of the state, μ_j and ν_l are padding amplitudes that we can choose, and λ is a positive normalisation constant. Once an MPS approximation of this state has been prepared using the circuit in Figure 3, we apply an inequality test to retrieve the DPSS state as shown in Figure 5.

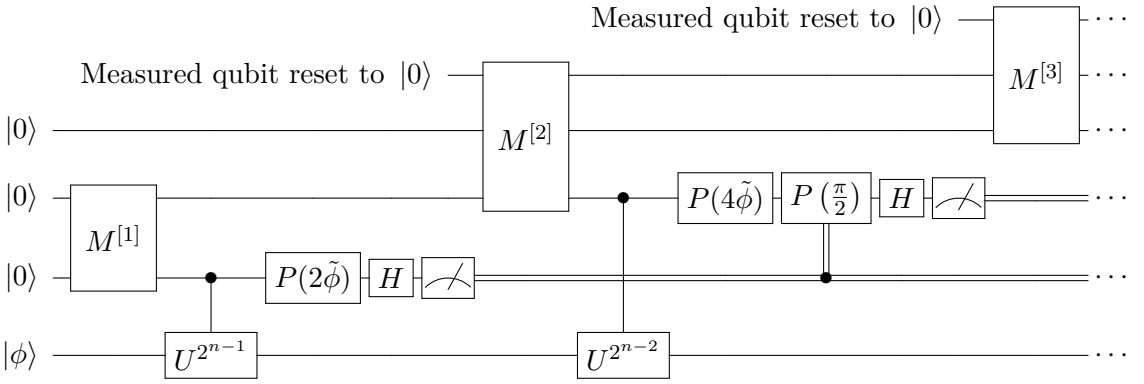

Figure 4: Circuit for performing QPE for a unitary operator U and its eigenstate $|\phi\rangle$ using the semi-classical Fourier transform. An MPS with $\chi = 4$ is being used to prepare a control state with dimension $D = 2^n$. Since qubits can be measured and recycled, no more than 3 qubits are needed for the control register at any given time. The circuit depicts the measurement of the first two least significant bits of the phase estimate, and it continues until all n bits have been obtained.

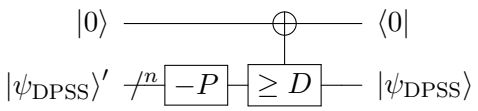

Figure 5: Circuit for preparing a D-dimensional DPSS state on n qubits. The $-P$ operator subtracts P from the register and the $\geq D$ operator flips the ancilla qubit if the value in the bottom register is greater than or equal to D. The preparation succeeds if the state $|0\rangle$ is measured on the ancilla qubit.

Assuming that the original coefficients were normalised $\sum_k \gamma_k^2 = 1$, the success probability of the circuit is $|1/\lambda|$. Although producing a state with $\mu_j = \nu_l = 0$ would provide a 100% success rate, using such a padding creates a sharp transition in the amplitudes that cannot be approximated well with an MPS. A better objective is to choose the padding amplitudes such that the MPS approximates the DPSS state amplitudes γ_k as well as possible while maintaining a reasonably high success rate. In the event of a state preparation failure, flagged by measuring the $|1\rangle$ state on the ancilla qubit, the preparation can be repeated until success.

A suitable choice of padding for even D can be obtained by setting $P = (N - D)/2$ and extending the DPSS in both directions, based on the extended definition of the prolate spheroidal wave function [50], which is the continuous limit of the DPSS as $D \to \infty$. In the Digital Library of Mathematical Functions [51], this is listed with the notation $\text{Ps}_n^m(x, \gamma^2)$, and the function values that we seek correspond to $\text{Ps}_0^0(x/N, dN/2)$, at values of x outside the region $-1 \leq x \leq 1$. Figure 6 shows the values of the padding amplitudes for $D = 34$ and $n = 6$ as an example.

These values can be numerically obtained by using the `SpheroidalPS` function available in Mathematica [52]. At higher values of D, determining the padding values with Mathematica becomes the computational bottleneck. This can be circumvented by performing a coarser sampling of padding values and interpolating between the results. We use this technique in the results shown in Section 3 for states on 17 qubits and beyond. Such an interpolation is an acceptable approximation because ultimately the padding amplitudes

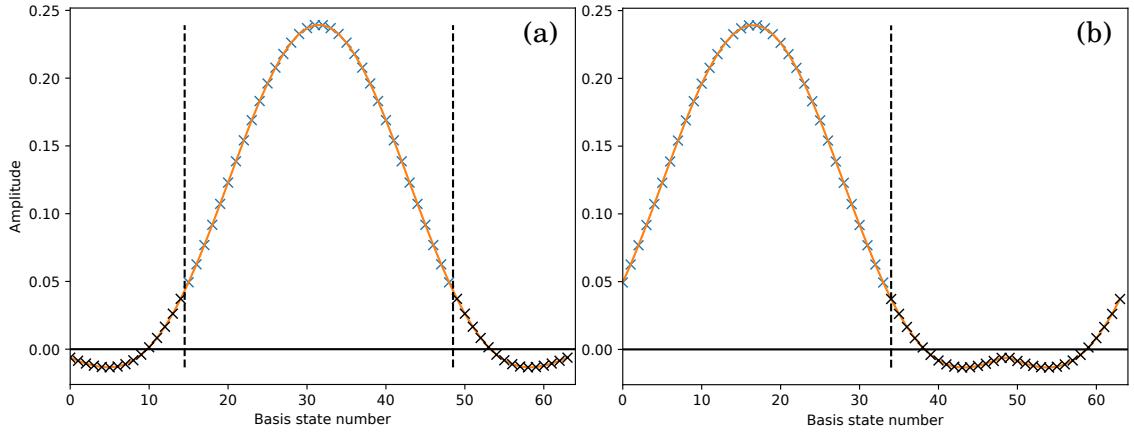

Figure 6: Plot (a) shows the state $|\psi_{\text{DPSS}}\rangle'$, which is approximated with an MPS. Blue crosses between the black vertical dashed lines indicate the values of γ_k for DPSS state with dimension $D = 34$, configured for confidence level $1 - \delta = 99\%$. The orange solid line is the prolate spheroidal wave function, from which the padding amplitudes μ_j and ν_l can be sampled, indicated with black crosses. Plot (b) shows the state after the circuit in Figure 5 has been applied. In the event of a success, the amplitudes beyond the black dashed line are set to zero by the inequality test, retrieving the desired state $|\psi_{\text{DPSS}}\rangle$. In this example, the probability for success is $1/\lambda \approx 99.3\%$. The values of the amplitudes in both plots are normalised with $\sum_k \gamma_k^2 = 1$.

are set to zero anyway by the inequality test, and their precise value is unimportant.

Once the state $|\psi_{\text{DPSS}}\rangle$ has been prepared on the control register, we can proceed to apply the unary iteration shown in Figure 1 and finally perform an inverse quantum Fourier transform and measurement. Although we can also use the semi-classical Fourier transform here, it is important to note that we cannot recycle qubits as shown in Figure 4 in this case, because both the inequality test and the unary iteration require access to the entire state.

Using a DPSS state with a dimension that is a power of two enables phase estimation with few qubits, whereas selecting a dimension that is not a power of two can reduce circuit complexity by up to half. Therefore, using a dimension that is a power of two is particularly useful for enabling the use of early-generation fault-tolerant quantum computers where the number of available qubits is severely limited. More advanced quantum computers, where qubit availability is not a limitation, will benefit from faster runtime achieved when using a lower dimension, which does not necessarily have to be a power of two.

3 QPE with an MPS approximation of a DPSS

We now turn to the practical implementation of DPSS states using MPS approximations. Our analysis pursues three primary objectives: First, we determine the minimum bond dimension required to accurately represent DPSS states across a range of system sizes and parameter values. Second, we examine how the approximation quality varies with state dimension and evaluate how the approximation error introduced by the MPS representation propagates into the phase estimation protocol itself. These investigations establish the conditions under which MPS approximations of DPSS states can serve as effective surrogates for their exact counterparts in QPE. Third, we explain in detail how to generate the MPS that achieves the optimal confidence interval for a given confidence level and control register size, and showcase the result with a representative example.

3.1 Selecting bond dimension

The bond dimension of an MPS determines both its representational capacity and its implementation cost, making the selection of an appropriate bond dimension crucial for practical applications. We begin by evaluating the fidelity between MPS approximations and target DPSS states across different bond dimensions. Fidelity provides a direct mathematical measure of approximation quality. The definition of fidelity $\mathcal{F}$ between these two states is

$$\mathcal{F} = |\langle \psi_{\text{DPSS}} | \psi_{\text{MPS}} \rangle|^2. \tag{21}$$

For calculating the amplitudes of the DPSS, we use the publicly available implementation in SciPy [53], which is based on the tridiagonal eigenvector formulation introduced in Ref. [24]. The MPS is constructed based on the target state as described in Section 2.3. This requires fewer classical computational resources than calculating the DPSS state. All of these tasks can be completed in less than a minute on a desktop computer for up to 24 qubits. More details on classical compute time are provided in Appendix B.

To determine the appropriate bond dimension for approximating DPSS states, we calculated the fidelity using Eq. (21) for MPS approximations with bond dimension $2, 4, 8$ and 16. The results for bond dimensions 2 and 4 are tabulated in Table 1. The fidelity increases sharply with bond dimension; for bond dimension 8 and above, the fidelity is equal to 1 up to numerical precision for all parameter values evaluated in the table. For 5 qubits or less, the MPS with bond dimension 4 also has perfect fidelity, which is due to the fact that an exact representation of any 5 qubit state is possible with bond dimension 4. Beyond that, when restricted to powers of two, the dimension of the state has almost no impact on the fidelity of the approximation.

n	Bond dimension 2			Bond dimension 4		
	$\delta_{\text{DPSS}} = 10^{-2}$	$\delta_{\text{DPSS}} = 10^{-3}$	$\delta_{\text{DPSS}} = 10^{-4}$	$\delta_{\text{DPSS}} = 10^{-2}$	$\delta_{\text{DPSS}} = 10^{-3}$	$\delta_{\text{DPSS}} = 10^{-4}$
5	1.27×10^{-4}	5.61×10^{-4}	1.16×10^{-3}	0	0	0
6	1.35×10^{-4}	5.95×10^{-4}	1.24×10^{-3}	3.38×10^{-14}	2.53×10^{-12}	4.22×10^{-11}
7	1.37×10^{-4}	6.04×10^{-4}	1.25×10^{-3}	4.93×10^{-14}	3.78×10^{-12}	6.28×10^{-11}
8	1.37×10^{-4}	6.06×10^{-4}	1.26×10^{-3}	5.44×10^{-14}	4.13×10^{-12}	6.88×10^{-11}
9	1.37×10^{-4}	6.07×10^{-4}	1.26×10^{-3}	5.51×10^{-14}	4.23×10^{-12}	7.03×10^{-11}
10	1.37×10^{-4}	6.07×10^{-4}	1.26×10^{-3}	5.51×10^{-14}	4.25×10^{-12}	7.07×10^{-11}
11	1.37×10^{-4}	6.07×10^{-4}	1.26×10^{-3}	5.77×10^{-14}	4.25×10^{-12}	7.08×10^{-11}
12	1.37×10^{-4}	6.07×10^{-4}	1.26×10^{-3}	5.60×10^{-14}	4.26×10^{-12}	7.08×10^{-11}
13	1.37×10^{-4}	6.07×10^{-4}	1.26×10^{-3}	5.57×10^{-14}	4.26×10^{-12}	7.08×10^{-11}
14	1.37×10^{-4}	6.07×10^{-4}	1.26×10^{-3}	5.66×10^{-14}	4.26×10^{-12}	7.08×10^{-11}
15	1.37×10^{-4}	6.07×10^{-4}	1.26×10^{-3}	5.44×10^{-14}	4.26×10^{-12}	7.08×10^{-11}
16	1.37×10^{-4}	6.07×10^{-4}	1.26×10^{-3}	5.82×10^{-14}	4.26×10^{-12}	7.08×10^{-11}
17	1.37×10^{-4}	6.07×10^{-4}	1.26×10^{-3}	5.51×10^{-14}	4.25×10^{-12}	7.08×10^{-11}
18	1.37×10^{-4}	6.07×10^{-4}	1.26×10^{-3}	5.53×10^{-14}	4.25×10^{-12}	7.08×10^{-11}
19	1.37×10^{-4}	6.07×10^{-4}	1.26×10^{-3}	5.55×10^{-14}	4.26×10^{-12}	7.08×10^{-11}
20	1.37×10^{-4}	6.07×10^{-4}	1.26×10^{-3}	5.37×10^{-14}	4.26×10^{-12}	7.09×10^{-11}
21	1.37×10^{-4}	6.07×10^{-4}	1.26×10^{-3}	5.71×10^{-14}	4.26×10^{-12}	7.08×10^{-11}
22	1.37×10^{-4}	6.07×10^{-4}	1.26×10^{-3}	5.55×10^{-14}	4.22×10^{-12}	7.04×10^{-11}
23	1.37×10^{-4}	6.07×10^{-4}	1.26×10^{-3}	3.97×10^{-14}	4.15×10^{-12}	7.21×10^{-11}
24	1.37×10^{-4}	6.07×10^{-4}	1.26×10^{-3}	3.48×10^{-13}	4.30×10^{-12}	7.29×10^{-11}

Table 1: Values of infidelity $1-\mathcal{F}$ for MPS approximations of DPSS states with dimension 2^n configured for confidence level δ_{DPSS}.

These results suggest that increasing the bond dimension beyond 8 has no benefit in

terms of approximation accuracy, and that the bond dimension of the MPS can be chosen independently of the dimension of the state. Since lower bond dimensions reduce implementation cost, we primarily focus on bond dimension 4 for the remainder of this work, as it represents the best balance between implementation efficiency and approximation quality, with bond dimension 2 exhibiting substantially lower fidelity as shown in Table 1.

3.2 Evaluating impact on phase estimation

While high fidelity indicates accurate state preparation, it does not directly reveal how MPS approximation errors affect the achievable confidence levels in phase estimation. To assess the practical impact of these approximations, we examine the degradation of confidence levels and confidence intervals of the phase estimate when using MPS-approximated DPSS states.

In phase estimation, if the normalised state on the control register is $|\psi\rangle = \sum_{k=0}^{D-1} f(k)\,|k\rangle$ (with integer values of k enumerating the basis states in the control register), then the probability of the phase ϕ being within a confidence interval of size $[-d, d]$ can be calculated as [18]:

$$1 - \delta = \frac{d}{\pi} \sum_{j=0}^{D-1} \sum_{k=0}^{D-1} f(j) f(k) \mathrm{sinc}(d(j-k)) = \langle\psi|S|\psi\rangle = \mathrm{Tr}\left[\rho S(d)\right], \tag{22}$$

where $\mathrm{sinc}(x) \equiv \sin(x)/x$, $\rho = |\psi\rangle\langle\psi|$, and $S(d)$ is defined in Eq. (14). The value of the sinc function at zero (which occurs when $j = k$) is 1, as given by the continuous limit. The DPSS state minimises d for any given δ, or conversely minimises δ for any given d, because it is by definition the eigenstate of $S(d)$ corresponding to the largest eigenvalue. Any other control state will perform worse, either by producing an increase in d, δ, or both. In Section 3.4 we analyse in more detail how these variables and the choice of the control state are related when using MPS approximations of DPSS states to achieve a target confidence level.

In this section, we instead focus on a direct comparison between DPSS states and their MPS approximations at fixed values of d. Since each DPSS state is an eigenstate of $S(d)$ for a specific value of d, we can evaluate the performance of an MPS approximation by fixing this value and calculating the increase in δ when the DPSS state is replaced with its MPS approximation.

We calculate the relative increase in δ induced by the MPS approximation as $(\delta_{\mathrm{MPS}} - \delta_{\mathrm{DPSS}})/\delta_{\mathrm{DPSS}}$, where δ_{MPS} and δ_{DPSS} correspond to the confidence levels achieved by the MPS and DPSS state respectively. We evaluate these quantities from Eq. (22). The classical computational resources required to directly evaluate the double sum in Eq. (22) scale as $\mathcal{O}(D^2)$. This can be done more efficiently by recognising that $S(d)$ is a Toeplitz matrix and that multiplication with a Toeplitz matrix is equal to discrete linear convolution with its first column $s(d)$ [54]:

$$\langle\psi|S|\psi\rangle = \langle\psi|s(d) * \psi\rangle\,. \tag{23}$$

Fast convolution algorithms based on the discrete Fourier transform and the circular convolution theorem can be used to evaluate this expression with computational complexity $\mathcal{O}(D \log D)$ [55].

We plot the relative increase in error probability $(\delta_{\mathrm{MPS}} - \delta_{\mathrm{DPSS}})/\delta_{\mathrm{DPSS}}$ induced by the MPS approximation as a function of the dimension D in Figure 7. For values of D that are not powers of 2, these results use padding values obtained by extending the prolate

spheroidal window function beyond the regular domain as explained in Section 2.5. We also tested zero-amplitude padding, but this produces relative errors up to two orders of magnitude larger while exhibiting qualitatively similar behaviour; these results are not shown here.

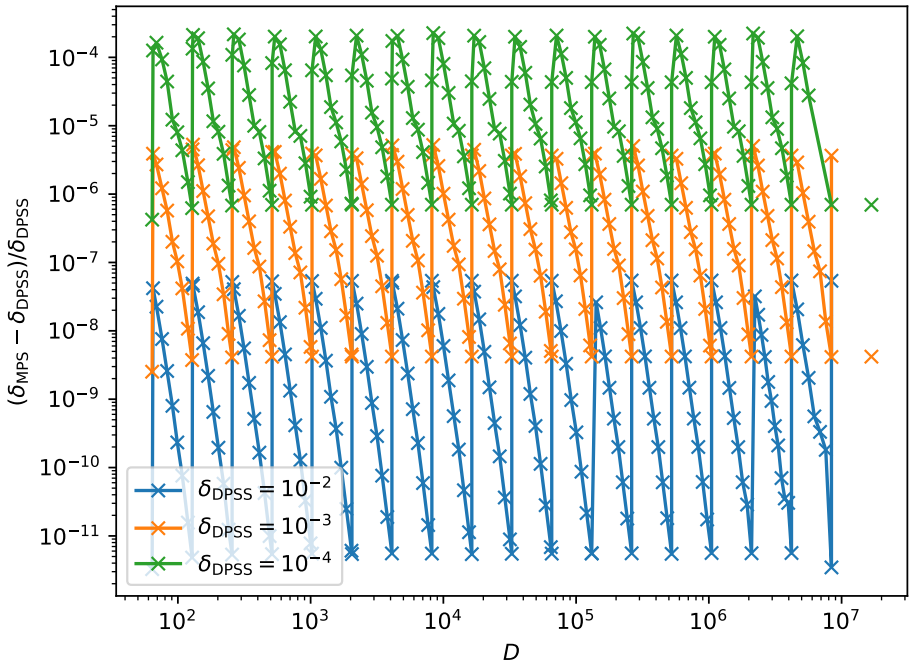

Figure 7: Relative increase in error probability δ arising from approximating a DPSS state of dimension D with an MPS with bond dimension 4. The minimal values occur at dimensions that are powers of two. Crosses indicate where the dimensions where the error was evaluated, and solid lines interpolate for visual guidance.

Overall, the relative increase in error induced by the MPS approximation is well below 0.1% for confidence levels up to $1 - 10^{-4}$, which is negligible for practical purposes. We attribute this to the fact that the MPS approximates the DPSS state with high fidelity.

We also observe a repeating pattern in the performance of the MPS approximation, where the maximum value of relative increase in error probability occurs at all values of D that are one larger than a power of two. This gradually improves until the next power of two, after which the error resets to approximately the same level as the previous maximum value. This indicates that the relative amount of padding matters, with MPS approximation performance improving when padding amplitudes constitute a smaller fraction of the total state. This leaves room for improvement in future work, where it may be possible to find even better padding values to reduce this effect.

3.3 Convergence with dimension

The periodic behaviour observed in Figure 7 indicates that MPS approximation performance remains approximately constant as the target state dimension doubles, even with fixed bond dimension. We hypothesise that this pattern reflects an inherent property of DPSS states that continues to hold beyond the dimensions we have numerically tested. To

support this hypothesis, we provide a theoretical explanation for the periodic behaviour and test whether the underlying mechanism persists as dimension increases.

As D is increased, each DPSS converges to a sequence obtained by sampling the prolate spheroidal wave function at D evenly spaced points. From this perspective, doubling the dimension corresponds to sampling the same underlying function at twice the resolution, inserting new sampling points between each existing pair. This suggests that doubling D requires only refining the least significant qubits while preserving the coarse structure encoded in the more significant qubits. If this property holds consistently, then the periodic pattern should persist for dimensions beyond those in Figure 7.

The structure of the MPS preparation circuit in Figure 3 accommodates this refinement naturally: doubling the state dimension introduces new operators exclusively on the least significant qubits. This does not necessarily mean that it is optimal to leave all other operators unchanged when new ones are added, but no changes in the previous operators are strictly required to produce refinements only in the least significant qubits. Numerically, we find that as qubits are added, previous operators may change noticeably with the new sequence of SVD-s, but this is only indicative of local unitary operators being transferred between neighbouring MPS operators. To account for this freedom, we will proceed by comparing states directly, rather than comparing the individual operators that are used to prepare them.

To test whether the difference between dimension D and $2D$ states occurs primarily in the least significant qubits, we compare an MPS approximation $|\psi_{(2D)}\rangle = M_{(2D)}^{[n+1]} \dots M_{(2D)}^{[1]} |0\rangle$ of a DPSS with dimension $2D$ to an MPS approximation $|\psi_{(D)}\rangle = M_{(D)}^{[n]} \dots M_{(D)}^{[1]} |0\rangle$ of a DPSS with dimension D. We calculate the fidelity $F(D)$ between $|\psi_{(D)}\rangle$ and the reduced state on the n most significant qubits obtained by tracing out the least significant qubit from $|\psi_{(2D)}\rangle$

$$F(D) = |(\langle\psi_{(D)}| \otimes \langle 0|) |\psi_{(2D)}\rangle|^2 + |(\langle\psi_{(D)}| \otimes \langle 1|) |\psi_{(2D)}\rangle|^2. \tag{24}$$

This expression is obtained by taking the reduced density matrix as

$$\rho = \langle 0|\psi_{(2D)}\rangle \langle\psi_{(2D)}|0\rangle + \langle 1|\psi_{(2D)}\rangle \langle\psi_{(2D)}|1\rangle \ , \tag{25}$$

and then using the expression for the fidelity between a pure state and a mixed state as $\langle\psi_{(D)}| \rho |\psi_{(D)}\rangle$.

We plot $1 - F(D)$ in Figure 8, and observe that the fidelity is high for any D, and increases with D. This convergence indicates that the state on the most significant qubits becomes increasingly stable as dimension grows, which is precisely what is required for the periodic pattern to persist at higher dimensions. The observed convergence supports our hypothesis that the pattern will continue beyond the tested range, because the underlying mechanism of refining only the least significant qubits becomes more pronounced rather than degraded as D increases. This provides evidence that the periodic pattern in Figure 7 will persist for MPS approximations of DPSS states beyond the dimensions that we have numerically evaluated.

3.4 Performance at fixed confidence level

From Figure 7, we have seen that approximating a DPSS state with an MPS produces a small relative decrease in confidence level, but in practical applications, it is desirable to use a procedure which guarantees a fixed confidence level. As explained in the beginning of Section 3.2, a desired confidence level $1 - \delta$ can be achieved either by changing the control state, increasing d, or a combination of the two.

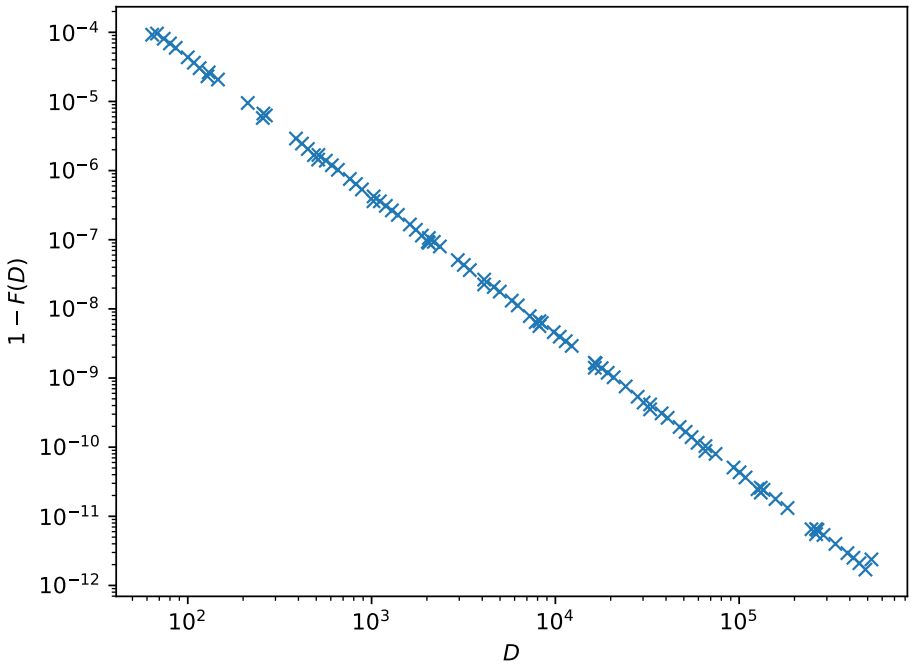

Figure 8: Infidelity between $|\psi_{(D)}\rangle$ and the mixed state left on the n most significant qubits when tracing out the least significant qubit from $|\psi_{(2D)}\rangle$.

We will now investigate how the required value of d changes when using different control states to achieve the target confidence level. The control states that we consider are the uniform superposition state, the DPSS state, and MPS approximations of DPSS states. The relation between d and δ is plotted in Figure 9 for these types of control states, when the size of the control register is 8 qubits. This serves as a representative example with similar behaviour observed for other control register sizes.

For the uniform superposition, there are no parameters to change for the control state. This means that δ can only be decreased at the expense of increasing d, as shown with the blue curve in Figure 9.

For DPSS states, there is a clear optimal choice of how to modify the state and d simultaneously. For any given value of d, the optimal choice among DPSS states is the eigenstate of the operator $S(d)$ that corresponds to the largest eigenvalue. The orange curve in Figure 9 shows the lowest possible value of d for any given δ, which is achieved by using the appropriate DPSS state. We have shaded the region below this curve to indicate that no other control state of the same dimension can provide the same confidence level at a lower value of d. The only way to access this region is by increasing the size of the control register.

For MPSs of bond dimensions 2 and 4, we plot two curves, depending on how the MPS is chosen. For the solid curves, the MPS used on each point in the curve is the approximation of the DPSS state that would be optimal for that value of d. At smaller values of d, these curves are indistinguishable in the plot, but at higher values of d, the MPSs provide confidence levels which are significantly worse than those provided by the corresponding DPSS states. The dashed curves show the best performance achievable by using MPS approximations, when optimizing over different possible target DPSS states,

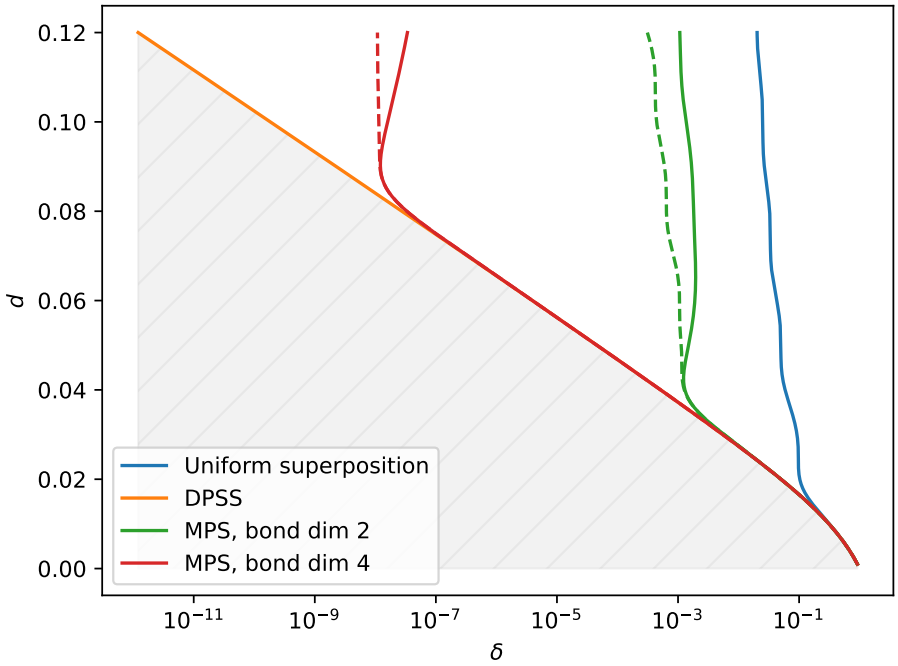

Figure 9: Confidence interval half-width d as a function of error probability δ for phase estimation using 8 control qubits. The shaded region below the DPSS line is unachievable by any 8 qubit control state and can only be accessed by increasing the dimension of the control state. Solid MPS curves correspond to MPS approximations of DPSS states chosen to be optimal for that value of d. Dashed MPS curves indicate the best performing MPS approximations, when optimising the target state over DPSS states configured for alternative values of d.

configured for alternative values of d.

The results demonstrate that MPS approximations achieve the near-optimal confidence interval width of the ideal DPSS up to a sharp confidence level cutoff that depends on bond dimension. Beyond this cutoff, the MPS cannot maintain optimal performance. Modifying the MPSs by optimising over different target DPSS states slightly improves performance (dashed curves), but still produces a result far from optimal. This is especially pronounced for the $\chi = 4$ case, where the dashed curve rises almost vertically beyond $\delta = 10^{-8}$, indicating that confidence levels beyond this threshold can only be achieved by substantially increasing d. When d approaches π, the confidence level is guaranteed to approach 100% for any control state.

To understand why the MPS approximations exhibit a sharp performance cutoff at certain confidence levels, we examine how the cutoff relates to approximation fidelity. In Appendix A, we derive an analytic upper bound on the relative increase in δ, as a function of fidelity:

$$\frac{\delta_{\mathrm{MPS}} - \delta_{\mathrm{DPSS}}}{\delta_{\mathrm{DPSS}}} \leq \left(\sqrt{\mathcal{F}} + \sqrt{\frac{1-\mathcal{F}}{\delta_{\mathrm{DPSS}}} + \mathcal{F} - 1} \right)^2 - 1. \tag{26}$$

As the desired confidence level increases (δ_{DPSS} decreases), the bound diverges when δ_{DPSS} becomes comparable to $1 - \mathcal{F}$ due to the term $(1 - \mathcal{F})/\delta_{\mathrm{DPSS}}$. This divergence predicts the confidence level regime where the MPS approximation can no longer be guaranteed to

perform well, explaining why the cutoff occurs when the desired confidence level approaches the same order of magnitude as the MPS infidelity (see Table 1). Conversely, in the regime $1-\mathcal{F} \ll \delta_{\mathrm{DPSS}}$, the bound remains small, indicating that high-fidelity MPS approximations produce negligible increases in error probability that require only negligible increases in d for compensation.

Across all bond dimensions tested, MPS-approximated DPSS states substantially outperform the uniform distribution commonly used as the default initial state in quantum phase estimation.

4 Implementation costs

The performance of the MPS preparation circuit is contingent on our ability to synthesise the operators $M^{[k]}$ accurately. One straightforward way of doing this is to extend the isometries to unitary operators and then use one of the many existing methods to perform unitary synthesis [39, 40, 41, 42, 43, 44]. A different and possibly more cost-efficient approach would be to use a technique for isometry synthesis [18, 28, 45, 46]. Reference [18] describes a custom isometry synthesis scheme designed specifically for MPS preparation, where certain parts between neighbouring operators can be combined. Here, we provide an adapted version of this, reinterpreted as a series of simplifications that arise after performing the cosine-sine decomposition (CSD) [56, 57, 58] on each of the operators.

When the bond dimension is 4, all operators $M^{[k]}$ besides $M^{[1]}, M^{[2]}, M^{[n-1]}$ and $M^{[n]}$ are isometries which have 8 rows and 4 columns. Splitting the rows in half to define 4×4 sub-matrices $M_0^{[k]}$ and $M_1^{[k]}$, we can use the 2-by-1 CSD, which is a decomposition of the form

$$M^{[k]} = \begin{pmatrix} M_0^{[k]} \\ M_1^{[k]} \end{pmatrix} = \begin{pmatrix} M_{b,0}^{[k]} & \\ & M_{b,1}^{[k]} \end{pmatrix} \begin{pmatrix} C \\ -S \end{pmatrix} M_a^{[k]}, \tag{27}$$

where $M_{b,0}^{[k]}$, $M_{b,1}^{[k]}$, and $M_a^{[k]}$ are two-qubit unitaries, and C and S are diagonal matrices with $C^2 + S^2 = \mathbb{1}$, which can be implemented in a quantum circuit with a controlled R_y gate. Figure 10(a) shows a part of the MPS circuit from Figure 3, and Figure 10(b) shows the corresponding circuit part when the CSD is applied to $M^{[k]}$. Note that Eq. (27) is presented with the common convention that the R_y gate is applied to the most significant qubit, allowing convenient use of the block-matrix notation. However, this is not a limitation of the 2-by-1 CSD, meaning that the decomposition can be applied to any qubit ordering by switching the appropriate rows and columns before and after applying the decomposition in the form of Eq. (27). In our MPS preparation circuit, we apply the decomposition with the rotation on the least significant qubit to remain compatible with the ordering established in Section 2.3.

We can reduce the cost of the circuit by combining the left two gates to obtain $\widetilde{M}^{[k-1]} = (\mathbb{1}_2 \otimes M_a^{[k]})M^{[k-1]}$ as shown in Figure 10(c). This circuit is equivalent to the construction in Ref. [18] with the adaptation that it is configured on an MPS circuit that uses no ancilla qubits. To implement the full MPS circuit, we can start by combining $M^{[n]}$ and $M^{[n-1]}$ with $M^{[n-2]}$ to retrieve $\widetilde{M}^{[n-2]} = (\mathbb{1}_4 \otimes M^{[n]})(\mathbb{1}_2 \otimes M^{[n-1]})M^{[n-2]}$. Then, we apply the CSD and merge steps iteratively to obtain $\widetilde{M}^{[n-3]} \dots \widetilde{M}^{[2]}$. We can also apply the CSD to $\widetilde{M}^{[2]}$, but $M_a^{[2]}$ cannot be merged with $M^{[1]}$ because it is a two-qubit operator that is not applied to the same qubits.

Since we are preparing a state with real-valued amplitudes, all of the gates in the circuit implement orthogonal operators. Controlled orthogonal operators can be decomposed with

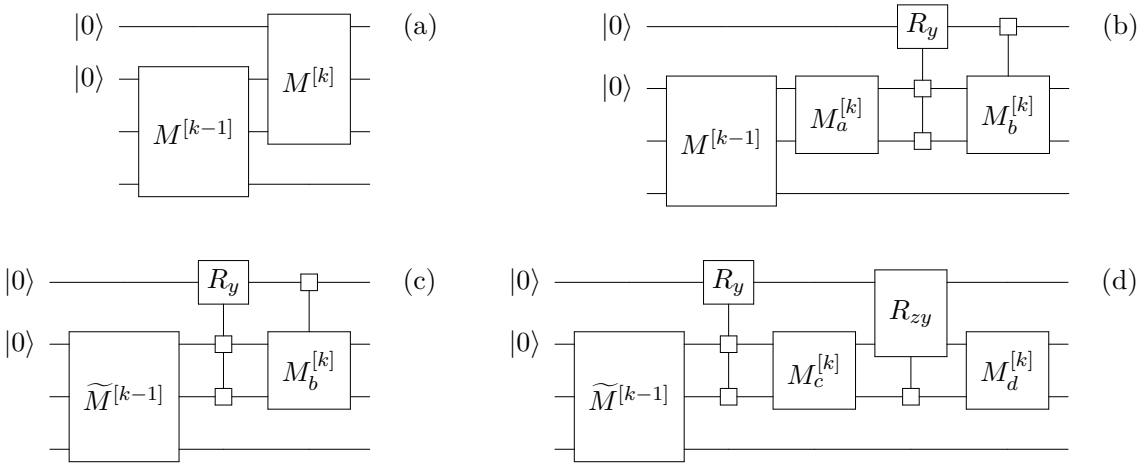

Figure 10: Circuits obtained by applying a series of useful decompositions to facilitate synthesis of an operator $M^{[k]}$ within an MPS preparation circuit. Circuit (a) shows a part of the MPS circuit from Figure 3. Circuit (b) is obtained by applying the CSD to (a). Circuit (c) is obtained by merging $M^{[k-1]}$ with $M_a^{[k]}$ in circuit (b). Circuit (d) is obtained by performing a type D Cartan decomposition to $M_b^{[k]}$ in circuit (c). This sequence of decompositions (a)-(d) can be sequentially applied for each value of k to obtain a decomposed circuit for the entire MPS.

a type D Cartan decomposition [59]. We will apply this to the $M_b^{[k]}$ gates to simplify the circuit further. The type D Cartan decomposition can be written as

$$\begin{pmatrix} M_{b,0}^{[k]} & \\ & M_{b,1}^{[k]} \end{pmatrix} = M_d^{[k]} \begin{pmatrix} C & S \\ -S & C \end{pmatrix} M_c^{[k]}, \tag{28}$$

where $M_c^{[k]}$ and $M_d^{[k]}$ are two-qubit unitaries and $C^2 + S^2 = \mathbb{1}_2$, which can be implemented in a quantum circuit with a uniformly controlled R_{zy} gate. Figure 10(d) shows the result after applying this decomposition to the circuit in Figure 10(c).

Each of the gates in Figure 10(d) has a known efficient implementation. The gates $M_c^{[k]}$ and $M_d^{[k]}$ can be constructed with the optimal decomposition of an orthogonal two-qubit gate [60]. The controlled rotations can be decomposed with the method of Ref. [61]. The result from applying these techniques to each of the gates is shown in Figure 11(a). The rightmost CZ gate of the decomposition of the controlled R_y gate has been omitted, since it can be absorbed into the controlled $M_b^{[k]}$ gate before performing the type D Cartan decomposition.

An alternative circuit with one fewer rotation gate is provided in Figure 11(b), which is obtained by processing the circuit in Figure 11(a) with a ZX-calculus-based circuit optimisation algorithm [62, 63]. The algorithm is implemented with publicly available code [64], provided by the authors of Ref. [62]. The first S gates in the circuit for $M^{[k]}$ could be removed by combining them with the $S^\dagger$ gates produced in the circuit for $\widetilde{M}^{[k-1]}$, but they are displayed in Figure 11(b) for clarity to convey that the operator $\widetilde{M}^{[k-1]}$ was not modified during the circuit optimisation procedure.

A numerical solver can be used to determine the angles for each rotation. We tested this approach using the `minimize` function in Scipy [53] with default parameters to minimise the Frobenius distance between the desired operator and the operator implemented by the circuits in Figure 11. We find that for both the MPS matrices for a DPSS state, as well as for randomly generated orthogonal matrices, the optimisation procedure is successful, achieving a Frobenius distance below 10^{-10} regardless of how the initial values of the angles

are chosen. Alternatively, the angles could be directly calculated by explicitly performing the decompositions and subsequent ZX-calculus based optimisation procedure on $\widetilde{M}^{[k]}$. With fixed bond dimension, the classical computational resources needed for calculating the angles scale linearly with n (logarithmically with state dimension), because there are n operators for which the angles need to be determined. This is a negligible cost compared to the SVD and DPSS amplitude calculation routines, which scale exponentially with n.

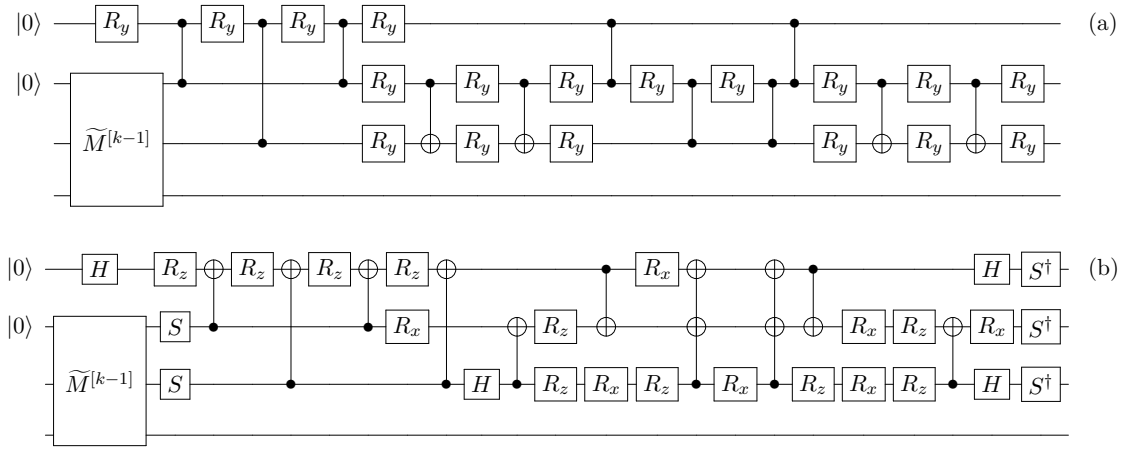

Figure 11: Two possible circuits for synthesising one operator $\widetilde{M}^{[k]}$. Both circuits contain only single qubit rotations and Clifford gates. Circuit (a) is obtained from using known efficient gate implementations of the gates in Figure 10(d). Circuit (b) is obtained by optimising Circuit (a) and contains one fewer rotation gate.

As a result, the implementation of each $\widetilde{M}^{[k]}$ involves synthesising 17 single-qubit rotations if we use the circuit in Figure 11(b). As special cases, $\widetilde{M}^{[2]}$ can be implemented with 16 rotations (the first two rotations can be combined) and $(\mathbb{1}_2 \otimes M_a^{[2]})(M^{[1]} \otimes \mathbb{1}_2)$ can be implemented with eight rotations because it performs a three-qubit state preparation with real amplitudes [65, 66]. In total, the state preparation circuit uses $24 + 17(n-4)$ single-qubit rotations. Assuming access to the Clifford+T gate set, single-qubit rotations can be synthesised probabilistically, with the aid of one ancilla qubit, with an average cost of $0.53\log_2(1/\epsilon) + 4.86$ T gates [67], where ϵ is the maximum allowed error in the value of the rotation angle. The total number of T gates to prepare an MPS on n qubits is therefore

$$\text{T-cost} \approx 9.01n\log_2(1/\epsilon) + 82.62n - 23.32\log_2(1/\epsilon) - 213.84\,. \tag{29}$$

If the goal is to prepare a DPSS state with dimension D, then $n = \lceil \log_2 D \rceil$ and if D is not a power of two, then an additional cost of at most $2n$ Toffoli gates is incurred for the subtraction [68] and comparator [69] operations described in Section 2.5, regardless of the value of ϵ. This extra cost would be present independently of how the MPS is prepared or how the rotations are synthesized.

In Table 2, we compare this cost to two different methods of preparing a DPSS state. The first is Grover-Rudolph state preparation [27], which can be used to prepare a state with arbitrary amplitudes. This prepares the exact DPSS state, but is not taking advantage of any properties of the state to reduce cost. We calculate the cost based on the implementation provided in Ref. [18], which optimises the usage of advanced QROM [28] (also known as QROAM [70]).

The second method that we compare to is the method of preparing the MPS approximation of the DPSS state (with $\chi = 4$), using the MPS preparation procedure of Ref. [18],

which also uses QROM. For both of these competing approaches, the dominant cost is expressed as the number of Toffoli gates used.

The number of T or Toffoli gates is used as a proxy for the overall complexity of the quantum circuits because they are the most difficult gates to implement in the surface code [71]. The best known technique for implementing these gates in the surface code involves magic state injection, where a high code distance magic state is designed and prepared specifically for the type of gate to be implemented. For this reason, it is most efficient to avoid implementing Toffolis with T gates, which requires 4 T gates per Toffoli [68], and instead directly use $|CCZ\rangle$ magic states [72, 73, 74] for circuits containing Toffoli gates.

Currently, the best known method for preparing magic states [75, 76] entails the following procedure. First, low code distance $|T\rangle$ magic states are prepared by cultivation [77, 78, 79]. These are then used to produce one high code distance $|CCZ\rangle$ magic state using $8\,|T\rangle$ to $|CCZ\rangle$ distillation [80]. This magic state can then be consumed to implement a Toffoli gate. To implement a T gate, one must further distil two high code distance $|T\rangle$ magic states from the $|CCZ\rangle$ magic state using a catalysed $|CCZ\rangle$ to $2\,|T\rangle$ conversion [73, 76]. Although this means that two T gates can be produced from the same resource state as one Toffoli gate, there is the added cost of conversion and the doubled cost of consumption. Due to this overhead, one Toffoli gate may be cheaper to implement than two T gates, but the exact ratio will depend on the compilation strategy and the application [76].

Table 2 compares the T gate count of our MPS preparation circuit to the Toffoli gate count of the MPS preparation method of Ref. [18]. Both MPS implementations provide considerable resource savings when compared to Grover-Rudolph DPSS state preparation at higher dimensions. Furthermore, both of the methods that we are comparing against use many more ancilla qubits, which are required for storing rotation angle values written by QROM. It is also important to note that ϵ describes the error allowed for the values of the angles in individual rotations. The overall error of the entire circuit is difficult to account for, but generally the total error increases with the number of gates in the circuit. This means that for larger circuits, a more stringent requirement on ϵ is also required to achieve the same overall error for the entire circuit. From this perspective, a reduced gate count as a function of n has an additional benefit of enabling a more lenient choice of ϵ, which further decreases the number of gates needed in total.

n	Method	Gate type	$\log_2(1/\epsilon)$					
			5	10	15	20	25	30
10	GR, Refs. [18, 27, 28]	Toffoli	490	636	761	858	948	1038
	MPS, Ref. [18]	Toffoli	697	952	1207	1462	1717	1972
	MPS, Eq. (29)	T	947	1281	1615	1948	2282	2616
20	GR, Refs. [18, 27, 28]	Toffoli	16362	20972	24817	27642	30212	32782
	MPS, Ref. [18]	Toffoli	1497	2052	2607	3162	3717	4272
	MPS, Eq. (29)	T	2223	3008	3792	4577	5361	6145
30	GR, Refs. [18, 27, 28]	Toffoli	524266	671724	794609	884730	966660	1048590
	MPS, Ref. [18]	Toffoli	2297	3152	4007	4862	5717	6572
	MPS, Eq. (29)	T	3500	4735	5970	7205	8440	9675

Table 2: Cost of state preparation for a range of values of n and ϵ. In each table entry, the top number is the number of Toffoli gates needed to prepare the exact DPSS state using a variant of arbitrary state preparation based on the Grover-Rudolph method [27, 28] that optimises the usage of QROM [18]. The middle number is the number of Toffoli gates needed to prepare an MPS with $\chi = 4$ using the method of Ref. [18]. The bottom number is the number of T-gates needed for preparing an MPS with $\chi = 4$ using the implementation given in Figure 11(b), with the cost calculated from Eq. (29).

It may also be of practical interest to estimate the number of T gates required to prepare a DPSS state that achieves a desired target confidence interval half-width d at a given confidence level $1-\delta$. To do this, we can use the asymptotic expression that relates the required DPSS state dimension D to δ and d [18]:

$$D \simeq \frac{\ln(1/\delta)}{2d} + \frac{\ln(8\pi \ln(4\sqrt{\pi}/\delta))}{4d} + \mathcal{O}\left(\frac{\ln\ln(1/\delta)}{d\ln(1/\delta)}\right). \tag{30}$$

A more precise value can be obtained by numerically searching for the value of D that produces the desired value of δ at a given value of d in Eq. (22). This dimension may need to be further increased slightly, to compensate for the finite value of ϵ. Once the dimension has been determined, the cost can be calculated with Eq. (29).

Table 3 shows the value of D for a variety of values of δ and d, calculated from Eq. (22). We provide a comparison to the textbook control state, as well as the sine window control state. At higher confidence levels, the difference in performance between the control states becomes increasingly pronounced.

Confidence level	Control state	Confidence interval half-width (radians)					
		0.2	0.1	0.05	0.01	0.005	0.001
95%	DPSS	26	52	103	513	1026	5127
	Sine	29	58	115	573	1146	5727
	Uniform	61	129	260	1303	2605	13025
99%	DPSS	36	71	141	702	1403	7013
	Sine	38	75	149	743	1486	7426
	Uniform	323	646	1293	6463	12926	64628
99.9%	DPSS	48	96	191	954	1907	9535
	Sine	86	172	344	1719	3438	17187
	Uniform	3169	6369	12742	63717	127434	637167
99.99%	DPSS	60	120	240	1198	2396	11980
	Sine	177	356	712	3561	7122	35608
	Uniform	31721	63614	127304	636700	1273404	6367028

Table 3: Control state dimension D required to achieve a given confidence level and interval width.

Besides the control state preparation costs described in this section, there are additional costs for performing phase estimation. If unary iteration is used as described in Section 2.1, then D applications to controlled U are used, as well as $D-2$ Toffoli gates for the iteration procedure itself [16]. If unary iteration is not used, then $\lceil 2^{\log_2 D} \rceil$ applications of controlled U are required. One phase shift gate per qubit is required for adding the randomly selected phase discussed in Section 2.1. The semi-classical Fourier transform requires at most one phase shift gate per qubit (depending on the measurement outcome), but this incurs no additional cost, because it can be combined with the phase shift gate used for the random phase that directly precedes it in the circuit, as can be seen in Figure 4.

The cost of implementing the controlled U operations will depend entirely on the application, but it is expected to incur the majority of the total cost of phase estimation. In many cases the cost of performing the controlled unitary is similar to the cost of performing the unitary without the control. For example, when using qubitization, the effect of controlling U can be achieved by controlling a reflection instead, incurring an overhead of just one Toffoli gate [16].

The application of the decompositions shown in Figure 10 generalises to the case of higher bond dimension and can be used to systematically take advantage of the sequential

structure of an MPS. This procedure can be used for preparing the MPS approximation of any state with real-valued amplitudes, making it also applicable to state preparation problems that are not necessarily connected to QPE or DPSS states.

5 Conclusion

In this work, we have demonstrated that a DPSS state can be prepared with high fidelity using an MPS approximation with bond dimension 4. The DPSS provides phase estimates with optimal performance in terms of the confidence interval. In practical applications, the confidence level is typically no more than 99%. In that case, the relative increase in the error is no more than one part in 10^7. Even if the required confidence level is 99.99%, the relative increase in the error is still no more than 3×10^{-4}. Since the relative increase in error probability is negligibly small, we have shown that our MPS approximation provides performance which is indistinguishable from the optimal DPSS result in a practical setting.

Our methods can be applied to DPSS states of any dimension, but they have additional benefits if the dimension is a power of 2. First, the accuracy of the approximation is very high. If the confidence level is 99%, then the relative increase in error is no more than one part in 10^{11}. Second, the state preparation circuit is compatible with the semi-classical Fourier transform. This enables the state to be generated as it is used for the phase estimation, so that no more than three qubits are needed for the control at once. This is important in situations where the quantum computer has a very limited number of logical qubits.

Our analysis of the performance of the MPS approximation is numerical, and so limited to a finite set of dimensions, but the features of the results strongly indicate that the MPS approximation is accurate for any dimension. We observe periodic behaviour with the dimension, where the performance of the MPS approximation is very similar as the dimension is increased by a factor of 2. The performance is best for dimensions that are a power of 2, and worst if the dimension is one more than a power of 2. As the dimension is increased, the performance improves until it is best again at the next power of 2.

We expect that this behaviour is because doubling the dimension corresponds to approximating the same function, but with one extra less-significant bit, so only a small adjustment to the state is needed. To support this explanation, we showed that the state prepared by the MPS circuit converges as the dimension of the DPSS is increased. This further strengthens the claim that our state preparation method is suitable for DPSS states with any dimension, because the high performance observed at low dimensions is predictive of high performance at higher dimensions.

We have constructed a circuit to efficiently prepare the MPS with a cost that scales logarithmically with both the dimension and desired precision of the prepared state. We have calculated the total number of T gates needed to implement the circuit. Our circuit has reduced non-Clifford gate count compared to prior work and avoids relying on quantum read-only memory, which has a high ancilla qubit overhead. Our circuit is applicable for preparing an MPS approximation of any state with real-valued amplitudes and is useful for applications that do not necessarily involve DPSS states or QPE.

In this work, we have concentrated on providing numerical evidence. There is still the open question of how to obtain a formal proof that these MPS approximations are accurate for any dimension. Our interpretation is that the DPSS sequences are well approximated by matrix product states because they converge to the prolate spheroidal window function in the continuum limit, which is a smooth function. Recent theoretical work has shown that target states with amplitudes that sample a smooth function can be well approximated

with low bond dimension [81].

Future work could also consider how to reduce the number of qubits used when the dimension is not a power of 2. That appears to be very difficult, because the unary iteration approach requires accessing all qubits of the control state prior to measurement, as do the shift and inequality test operations that suppress unwanted amplitudes. An alternative direction for improvement is developing better MPS approximations through optimised padding schemes for non-power-of-2 dimensions.

6 Code availability

Python code for reproducing the results in all Figures and Tables is available on GitHub [82]. The Python scripts used for generating Figures 6–8 internally call Mathematica's `SpheroidalPS` function [52] to implement the padding technique described in Section 2.5. A Mathematica installation is therefore required for these Figures, though no direct interaction with Mathematica is necessary. The results of prior work, presented in Table 2, were produced with Matlab code provided by the authors of Ref. [18].

7 Author contributions

KK performed the numerical work and wrote the manuscript, under the supervision of DWB. Both authors contributed to the analysis and revised the text. LLM tools were used to generate some of the code. All of the code was manually reviewed and verified for correctness by the authors. None of the manuscript text is LLM generated, but LLM suggestions were used to improve structure, grammar, and clarity of presentation.

8 Acknowledgments

KK worked on this project with funding from Molecular Quantum Solutions ApS and Sydney Quantum Academy, and would like to thank Mark Nicholas Jones and Kristín Björg Arnardóttir for their feedback on this manuscript. DWB worked on this project under a sponsored research agreement with Google Quantum AI. DWB is also supported by Australian Research Council Discovery Projects DP210101367 and DP220101602. The authors thank Guang Hao Low for helpful comments regarding the cost of preparing magic states.

References

[1] Peter W. Shor. "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer". SIAM Journal on Computing **26**, 1484–1509 (1997).

[2] Alán Aspuru-Guzik, Anthony D Dutoi, Peter J Love, and Martin Head-Gordon. "Simulated quantum computation of molecular energies". Science **309**, 1704–1707 (2005).

[3] Sam McArdle, Suguru Endo, Alán Aspuru-Guzik, Simon C. Benjamin, and Xiao Yuan. "Quantum computational chemistry". Rev. Mod. Phys. **92**, 015003 (2020).

[4] Yudong Cao, Jonathan Romero, Jonathan P Olson, Matthias Degroote, Peter D Johnson, Mária Kieferová, Ian D Kivlichan, Tim Menke, Borja Peropadre, Nicolas PD Sawaya, et al. "Quantum chemistry in the age of quantum computing". Chemical Reviews **119**, 10856–10915 (2019).

[5] Bela Bauer, Sergey Bravyi, Mario Motta, and Garnet Kin-Lic Chan. "Quantum algorithms for quantum chemistry and quantum materials science". Chemical Reviews **120**, 12685–12717 (2020).

[6] Mario Motta and Julia E Rice. "Emerging quantum computing algorithms for quantum chemistry". Wiley Interdisciplinary Reviews: Computational Molecular Science **12**, e1580 (2022).

[7] Alexander M. Dalzell, Sam McArdle, Mario Berta, Przemyslaw Bienias, Chi-Fang Chen, András Gilyén, Connor T. Hann, Michael J. Kastoryano, Emil T. Khabiboulline, Aleksander Kubica, Grant Salton, Samson Wang, and Fernando G. S. L. Brandão. "Quantum algorithms: A survey of applications and end-to-end complexities". Cambridge University Press. (2025). arXiv:2310.03011.

[8] Michael A Nielsen and Isaac L Chuang. "Quantum computation and quantum information". Cambridge university press. (2010).

[9] Dominic W. Berry, Mária Kieferová, Artur Scherer, Yuval R. Sanders, Guang Hao Low, Nathan Wiebe, Craig Gidney, and Ryan Babbush. "Improved techniques for preparing eigenstates of fermionic Hamiltonians". npj Quantum Information **4**, 22 (2018).

[10] David Poulin, Alexei Kitaev, Damian S. Steiger, Matthew B. Hastings, and Matthias Troyer. "Quantum algorithm for spectral measurement with a lower gate count". Phys. Rev. Lett. **121**, 010501 (2018).

[11] F.J. Harris. "On the use of windows for harmonic analysis with the discrete Fourier transform". Proceedings of the IEEE **66**, 51–83 (1978).

[12] Benjamin P Abbott, Rich Abbott, Thomas D Abbott, Fausto Acernese, Kendall Ackley, Carl Adams, Thomas Adams, Paolo Addesso, Rana X Adhikari, Vaishali B Adya, et al. "GW170817: Observation of gravitational waves from a binary neutron star inspiral". Phys. Rev. Lett. **119**, 161101 (2017).

[13] Samantha A Usman, Alexander H Nitz, Ian W Harry, Christopher M Biwer, Duncan A Brown, Miriam Cabero, Collin D Capano, Tito Dal Canton, Thomas Dent, Stephen Fairhurst, Marcel S Kehl, Drew Keppel, Badri Krishnan, Amber Lenon, Andrew Lundgren, Alex B Nielsen, Larne P Pekowsky, Harald P Pfeiffer, Peter R Saulson, Matthew West, and Joshua L Willis. "The PyCBC search for gravitational waves from compact binary coalescence". Classical and Quantum Gravity **33**, 215004 (2016).

[14] A. Luis and J. Peřina. "Optimum phase-shift estimation and the quantum description of the phase difference". Phys. Rev. A **54**, 4564–4570 (1996).

[15] Andrew M. Childs. "On the relationship between continuous- and discrete-time quantum walk". Commun. Math. Phys **294**, 581–603 (2009).

[16] Ryan Babbush, Craig Gidney, Dominic W Berry, Nathan Wiebe, Jarrod McClean, Alexandru Paler, Austin Fowler, and Hartmut Neven. "Encoding electronic spectra in quantum circuits with linear T complexity". Phys. Rev. X **8**, 041015 (2018).

[17] Peyman Najafi, Pedro C. S. Costa, and Dominic W. Berry. "Optimum phase estimation with two control qubits". AVS Quantum Science **5**, 023802 (2023).

[18] Dominic W Berry, Yu Tong, Tanuj Khattar, Alec White, Tae In Kim, Guang Hao Low, Sergio Boixo, Zhiyan Ding, Lin Lin, Seunghoon Lee, et al. "Rapid initial-state preparation for the quantum simulation of strongly correlated molecules". PRX Quantum **6**, 020327 (2025).

[19] Hiroshi Imai and Masahito Hayashi. "Fourier analytic approach to phase estimation in quantum systems". New Journal of Physics **11**, 043034 (2009).

[20] Rei Sakuma, Kaito Wada, Shu Kanno, Kimberlee Keithley, Kenji Sugisaki, Takashi Abe, Hajime Nakamura, and Naoki Yamamoto. "Quantum-phase-estimation-based filtering: Performance analysis and application to low-energy spectral calculations". Phys. Rev. A **113**, 012602 (2026).

[21] David Slepian. "Analytic solution of two apodization problems". Journal of the Optical Society of America **55**, 1110–1115 (1965).

[22] Sandrine J Thomas, Rémi Soummer, Daren Dillon, Bruce Macintosh, Julia W Evans, Donald Gavel, Anand Sivaramakrishnan, Christian Marois, and Ben R Oppenheimer. "Testing the APLC on the LAO ExAO testbed". In Adaptive Optics Systems. Volume 7015, pages 1845–1855. SPIE (2008).

[23] Changqing Xie, Xiaoli Zhu, Lina Shi, and Ming Liu. "Spiral photon sieves apodized by digital prolate spheroidal window for the generation of hard-x-ray vortex". Optics Letters **35**, 1765–1767 (2010).

[24] David Slepian. "Prolate spheroidal wave functions, Fourier analysis, and uncertainty—V: The discrete case". Bell System Technical Journal **57**, 1371–1430 (1978).

[25] J Kaiser and R Schafer. "On the use of the I_0-sinh window for spectrum analysis". IEEE Transactions on Acoustics, Speech, and Signal Processing **28**, 105–107 (1980).

[26] Dominic W Berry, Yuan Su, Casper Gyurik, Robbie King, Joao Basso, Alexander Del Toro Barba, Abhishek Rajput, Nathan Wiebe, Vedran Dunjko, and Ryan Babbush. "Analyzing prospects for quantum advantage in topological data analysis". PRX Quantum **5**, 010319 (2024).

[27] Lov Grover and Terry Rudolph. "Creating superpositions that correspond to efficiently integrable probability distributions" (2002). arXiv:quant-ph/0208112.

[28] Guang Hao Low, Vadym Kliuchnikov, and Luke Schaeffer. "Trading T gates for dirty qubits in state preparation and unitary synthesis". Quantum **8**, 1375 (2024).

[29] Sean Greenaway, William Pol, and Sukin Sim. "A case study against QSVT: assessment of quantum phase estimation improved by signal processing techniques" (2024). url: `https://doi.org/10.48550/arXiv.2404.01396`.

[30] Oliver O'Brien and Christoph Sünderhauf. "Quantum state preparation via piecewise QSVT". Quantum **9**, 1786 (2025).

[31] Sam McArdle, András Gilyén, and Mario Berta. "Quantum state preparation without coherent arithmetic" (2025). arXiv:2210.14892.

[32] Dhrumil Patel, Shi Jie Samuel Tan, Yi ğit Subaş ı, and Andrew T. Sornborger. "Optimal coherent quantum phase estimation via tapering". PRX Quantum **7**, 020302 (2026).

[33] Guang Hao Low and Isaac L Chuang. "Optimal Hamiltonian simulation by quantum signal processing". Phys. Rev. Lett. **118**, 010501 (2017).

[34] Joonho Lee, Dominic W. Berry, Craig Gidney, William J. Huggins, Jarrod R. McClean, Nathan Wiebe, and Ryan Babbush. "Even more efficient quantum computations of chemistry through tensor hypercontraction". PRX Quantum **2**, 030305 (2021).

[35] Dominic W. Berry, Michael J. W. Hall, Marcin Zwierz, and Howard M. Wiseman. "Optimal Heisenberg-style bounds for the average performance of arbitrary phase estimates". Phys. Rev. A **86**, 053813 (2012).

[36] Frederik J. Simons. "Slepian functions and their use in signal estimation and spectral analysis". Page 891–923 in Handbook of Geomathematics. Springer Berlin Heidelberg. (2010).

[37] Guifré Vidal. "Efficient classical simulation of slightly entangled quantum computations". Phys. Rev. Lett. **91**, 147902 (2003).

[38] Ulrich Schollwöck. "The density-matrix renormalization group in the age of matrix product states". Annals of Physics **326**, 96–192 (2011).

[39] Christopher M. Dawson and Michael A. Nielsen. "The Solovay-Kitaev algorithm" (2005). arXiv:quant-ph/0505030.

[40] Tianyi Hao, Amanda Xu, and Swamit Tannu. "Reducing T gates with unitary synthesis". In Proceedings of the 31st ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2. Page 1589–1604. ASPLOS '26New York, NY, USA (2026). Association for Computing Machinery.

[41] Matthew Amy, Dmitri Maslov, Michele Mosca, and Martin Roetteler. "A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits". IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **32**, 818–830 (2013).

[42] Vlad Gheorghiu, Michele Mosca, and Priyanka Mukhopadhyay. "T-count and T-depth of any multi-qubit unitary". npj Quantum Information **8**, 141 (2022).

[43] Anouk Paradis, Jasper Dekoninck, Benjamin Bichsel, and Martin Vechev. "Synthetiq: Fast and versatile quantum circuit synthesis". Proceedings of the ACM on Programming Languages **8**, 55–82 (2024).

[44] Mathias Weiden, Justin Kalloor, John Kubiatowicz, Ed Younis, and Costin Iancu. "High-precision multi-qubit Clifford+T synthesis by unitary diagonalization". Electronic Proceedings in Theoretical Computer Science **426**, 215–230 (2025).

[45] Raban Iten, Roger Colbeck, Ivan Kukuljan, Jonathan Home, and Matthias Christandl. "Quantum circuits for isometries". Phys. Rev. A **93**, 032318 (2016).

[46] Aaron Szasz, Ed Younis, and Wibe De Jong. "Numerical circuit synthesis and compilation for multi-state preparation". In 2023 IEEE International Conference on Quantum Computing and Engineering (QCE). Volume 01, pages 768–778. (2023).

[47] Stepan Fomichev, Kasra Hejazi, Modjtaba Shokrian Zini, Matthew Kiser, Joana Fraxanet, Pablo Antonio Moreno Casares, Alain Delgado, Joonsuk Huh, Arne-Christian Voigt, Jonathan E Mueller, et al. "Initial state preparation for quantum chemistry on quantum computers". PRX Quantum **5**, 040339 (2024).

[48] William J Huggins, Oskar Leimkuhler, Torin F Stetina, and K Birgitta Whaley. "Efficient state preparation for the quantum simulation of molecules in first quantization". PRX Quantum **6**, 020319 (2025).

[49] Robert B Griffiths and Chi-Sheng Niu. "Semiclassical Fourier transform for quantum computation". Phys. Rev. Lett. **76**, 3228 (1996).

[50] Wolfram Research (2007), SpheroidalPS, Wolfram Language function, `https://reference.wolfram.com/language/ref/SpheroidalPS.html`.

[51] "*NIST Digital Library of Mathematical Functions*". `https://dlmf.nist.gov/30.4`, Release 1.2.3 of 2024-12-15. F. W. J. Olver, A. B. Olde Daalhuis, D. W. Lozier, B. I. Schneider, R. F. Boisvert, C. W. Clark, B. R. Miller, B. V. Saunders, H. S. Cohl, and M. A. McClain, eds.

[52] Wolfram Research, Inc. "Mathematica, Version 14.3". Champaign, IL, 2025.

[53] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". Nature Methods **17**, 261–272 (2020).

[54] Robert M. Gray. "Toeplitz and circulant matrices: A review". Foundations and Trends in Communications and Information Theory **2**, 155–239 (2006).

[55] A.V. Oppenheim, R.W. Schafer, and J.R. Buck. "Discrete-time signal processing". Prentice Hall International Editions Series. Prentice Hall. (1999).

[56] G.H. Golub and C.F. Van Loan. "Matrix computations". Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press. (2013).

[57] C.C. Paige and M. Wei. "History and generality of the CS decomposition". Linear Algebra and its Applications **208-209**, 303–326 (1994).

[58] Brian D Sutton. "Computing the complete CS decomposition". Numerical Algorithms **50**, 33–65 (2009).

[59] David Wierichs, Maxwell West, Roy T. Forestano, M. Cerezo, and Nathan Killoran. "Recursive Cartan decompositions for unitary synthesis" (2025). arXiv:2503.19014.

[60] Hai-Rui Wei and Yao-Min Di. "Decomposition of orthogonal matrix and synthesis of two-qubit and three-qubit orthogonal gates". Quantum Info. Comput. **12**, 262–270 (2012).

[61] Vivek V. Shende, Stephen S. Bullock, and Igor L. Markov. "Synthesis of Quantum Logic Circuits". IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **25**, 1000–1010 (2006). arXiv:quant-ph/0406176.

[62] Aleks Kissinger and John van de Wetering. "Pyzx: Large scale automated diagrammatic reasoning". Electronic Proceedings in Theoretical Computer Science **318**, 229–241 (2020).

[63] Ross Duncan, Aleks Kissinger, Simon Perdrix, and John van de Wetering. "Graph-theoretic Simplification of Quantum Circuits with the ZX-calculus". Quantum **4**, 279 (2020).

[64] `https://github.com/zxcalc/pyzx`.

[65] Oscar Perdomo, Nelson Castaneda, and Roger Vogeler. "Preparation of 3-qubit states" (2022). arXiv:2201.03724.

[66] Yonghae Lee and Taewan Kim. "Three-qubit-state preparation: Classification and explicit circuits". Phys. Rev. A **113**, 062444 (2026).

[67] Vadym Kliuchnikov, Kristin Lauter, Romy Minko, Adam Paetznick, and Christophe Petit. "Shorter quantum circuits via single-qubit gate approximation". Quantum **7**, 1208 (2023).

[68] Craig Gidney. "Halving the cost of quantum addition". Quantum **2**, 74 (2018).

[69] Vivien Vandaele. "Asymptotically optimal quantum circuits for comparators and incrementers" (2026). arXiv:2603.12917.

[70] Dominic W. Berry, Craig Gidney, Mario Motta, Jarrod R. McClean, and Ryan Babbush. "Qubitization of Arbitrary Basis Quantum Chemistry Leveraging Sparsity and Low Rank Factorization". Quantum **3**, 208 (2019).

[71] Austin G. Fowler, Matteo Mariantoni, John M. Martinis, and Andrew N. Cleland. "Surface codes: Towards practical large-scale quantum computation". Phys. Rev. A **86**, 032324 (2012).

[72] Sungyeon Kook, Yujin Kang, Ilkwon Sohn, and Jun Heo. "Low spatial cost CCZ magic state factory" (2026). arXiv:2606.24170.

[73] Craig Gidney and Austin G. Fowler. "Efficient magic state factories with a catalyzed $|CCZ\rangle$ to $2|T\rangle$ transformation". Quantum **3**, 135 (2019).

[74] Jeongwan Haah and Matthew B. Hastings. "Codes and Protocols for Distilling T, controlled-S, and Toffoli Gates". Quantum **2**, 71 (2018).

[75] Craig Gidney. "How to factor 2048 bit RSA integers with less than a million noisy qubits" (2025). arXiv:2505.15917.

[76] Guang Hao Low, William J. Huggins, Dominic W. Berry, Tanuj Khattar, Alec F. White, Nicholas C. Rubin, and Ryan Babbush. "A denser planar surface code" (2026). arXiv:2605.30455.

[77] Craig Gidney, Noah Shutty, and Cody Jones. "Magic state cultivation: growing T states as cheap as CNOT gates" (2024). arXiv:2409.17595.

[78] Jahan Claes. "Cultivating T states on the surface code with only two-qubit gates" (2025). arXiv:2509.05232.

[79] Emma Rosenfeld, Craig Gidney, Gabrielle Roberts, Alexis Morvan, Nathan Lacroix, Dvir Kafri, Jeffrey Marshall, Ming Li, Volodymyr Sivak, et al. "Magic state cultivation on a superconducting quantum processor" (2025). arXiv:2512.13908.

[80] Cody Jones. "Low-overhead constructions for the fault-tolerant Toffoli gate". Phys. Rev. A **87**, 022328 (2013).

[81] Vladyslav Bohun, Illia Lukin, Mykola Luhanko, Georgios Korpas, Philippe JS De Brouwer, Mykola Maksymenko, and Maciej Koch-Janusz. "Entanglement scaling in matrix product state representation of smooth functions and their shallow quantum circuit approximations". Phys. Rev. Res. **8**, 023062 (2026).

[82] `https://github.com/MQSdk/DPSS-QPE`.

[83] John Watrous. "The theory of quantum information". Cambridge University Press. (2018).

A Derivation of bounds

Since $S(d) \equiv \frac{d}{\pi}\sum_{j,k}\mathrm{sinc}(d(j-k))\,|j\rangle\langle k|$ is a positive semi-definite operator, then $\{S(d), \mathbb{1}-S(d)\}$ is a positive operator valued measure and we can use a well-known inequality relating the fidelity of quantum states to the Bhattacharyya coefficient [83]:

$$\begin{aligned}\sqrt{\mathcal{F}} &\leq \sqrt{\mathrm{Tr}[\rho_{\mathrm{MPS}}S(d)]\mathrm{Tr}[\rho_{\mathrm{DPSS}}S(d)]} + \sqrt{\mathrm{Tr}[\rho_{\mathrm{MPS}}(\mathbb{1}-S(d))]\mathrm{Tr}[\rho_{\mathrm{DPSS}}(\mathbb{1}-S(d))]} \\ &= \sqrt{(1-\delta_{\mathrm{MPS}})(1-\delta_{\mathrm{DPSS}})} + \sqrt{\delta_{\mathrm{MPS}}\delta_{\mathrm{DPSS}}}\,. \end{aligned} \tag{31}$$

Rearranging this equation yields

$$\sqrt{\mathcal{F}} - \sqrt{\delta_{\mathrm{DPSS}}\delta_{\mathrm{MPS}}} \leq \sqrt{(1-\delta_{\mathrm{DPSS}})(1-\delta_{\mathrm{MPS}})}\,. \tag{32}$$

If $\mathcal{F} > \delta_{\mathrm{DPSS}}\delta_{\mathrm{MPS}}$, then both sides are positive. This inequality holds for the regime we are interested in, where we have $\mathcal{F}$ quite close to 1 and $\delta_{\mathrm{DPSS}}\delta_{\mathrm{MPS}}$ quite close to zero. Therefore, we can square both sides and then divide by δ_{DPSS} to obtain

$$\begin{aligned}
\mathcal{F} - 2\sqrt{\mathcal{F}\delta_{\mathrm{DPSS}}\delta_{\mathrm{MPS}}} + \delta_{\mathrm{DPSS}}\delta_{\mathrm{MPS}} &\leq 1 - \delta_{\mathrm{DPSS}} - \delta_{\mathrm{MPS}} + \delta_{\mathrm{DPSS}}\delta_{\mathrm{MPS}} \\
\mathcal{F} - 2\sqrt{\mathcal{F}\delta_{\mathrm{DPSS}}\delta_{\mathrm{MPS}}} &\leq 1 - \delta_{\mathrm{DPSS}} - \delta_{\mathrm{MPS}} \\
\frac{\delta_{\mathrm{MPS}}}{\delta_{\mathrm{DPSS}}} - 2\sqrt{\mathcal{F}\frac{\delta_{\mathrm{MPS}}}{\delta_{\mathrm{DPSS}}}} + 1 - \frac{1-\mathcal{F}}{\delta_{\mathrm{DPSS}}} &\leq 0. \qquad (33)
\end{aligned}$$

This is a quadratic inequality with respect to $\sqrt{\delta_{\mathrm{MPS}}/\delta_{\mathrm{DPSS}}}$ that has critical numbers $w_{\pm}$ at

$$w_{\pm} = \frac{2\sqrt{\mathcal{F}} \pm \sqrt{4\mathcal{F} - 4\left(1 - \frac{1-\mathcal{F}}{\delta_{\mathrm{DPSS}}}\right)}}{2} = \sqrt{\mathcal{F}} \pm \sqrt{\mathcal{F} - 1 + \frac{1-\mathcal{F}}{\delta_{\mathrm{DPSS}}}}. \qquad (34)$$

The critical numbers are real-valued when $\mathcal{F} - 1 + (1-\mathcal{F})/\delta_{\mathrm{DPSS}} \geq 0$. This inequality is equivalent to $(1-\mathcal{F})(1-\delta_{\mathrm{DPSS}})\delta_{\mathrm{DPSS}} \geq 0$, which is always true because $0 \leq \mathcal{F} \leq 1$ and $0 \leq \delta_{\mathrm{DPSS}} \leq 1$. Therefore, we have a real-valued upper bound of $\sqrt{\delta_{\mathrm{MPS}}/\delta_{\mathrm{DPSS}}}$ at

$$w_{+} = \sqrt{\mathcal{F}} + \sqrt{\mathcal{F} - 1 + \frac{1-\mathcal{F}}{\delta_{\mathrm{DPSS}}}}. \qquad (35)$$

Therefore

$$\frac{\delta_{\mathrm{MPS}} - \delta_{\mathrm{DPSS}}}{\delta_{\mathrm{DPSS}}} \leq (w_{+})^2 - 1 = \left(\sqrt{\mathcal{F}} + \sqrt{\frac{1-\mathcal{F}}{\delta_{\mathrm{DPSS}}} + \mathcal{F} - 1}\right)^2 - 1. \qquad (36)$$

There is also a lower bound for $\sqrt{\delta_{\mathrm{MPS}}/\delta_{\mathrm{DPSS}}}$ at w_{-}, however this only implies a positive lower bound for $(\delta_{\mathrm{MPS}} - \delta_{\mathrm{DPSS}})/\delta_{\mathrm{DPSS}}$ when

$$\begin{aligned}
w_{-} &\geq 1 \\
\sqrt{F} - \sqrt{\mathcal{F} - 1 + \frac{1-\mathcal{F}}{\delta_{\mathrm{DPSS}}}} &\geq 1 \\
1 - \sqrt{\mathcal{F}} &\geq \sqrt{\mathcal{F} - 1 + \frac{1-\mathcal{F}}{\delta_{\mathrm{DPSS}}}} \\
1 - 2\sqrt{\mathcal{F}} + \mathcal{F} &\geq \mathcal{F} - 1 + \frac{1-\mathcal{F}}{\delta_{\mathrm{DPSS}}} \\
\delta_{\mathrm{DPSS}} &\geq \frac{1+\sqrt{\mathcal{F}}}{2}, \qquad (37)
\end{aligned}$$

which does not occur in the regime we are interested in, where we have $\mathcal{F}$ quite close to 1 and δ_{DPSS} quite close to zero.

B Classical computation time

Figure 12 shows the average time used to perform the classical calculations, as a function of state dimension. For the DPSS amplitude calculation, we average the calculation time over three different DPSS states (corresponding to 99%, 99.9% and 99.99% confidence intervals)

with 100 trials for each setting. Subsequently, we calculated the isometries for the MPS preparation circuit as described in Section 2.3 for bond dimensions $\chi = 2$ and $\chi = 4$, for a total of 150 states each. We repeated this for power of two state dimensions from $D = 2^5$ to $D = 2^{24}$.

The computer used for generating this plot has 256 GB of RAM and a 64-core 2.9 GHz AMD Threadripper 3990X CPU. The DPSS amplitudes are calculated with the `signal.windows.dpss` function from the SciPy python package and the singular value decompositions are performed with the `linalg.svd` function from the NumPy Python package. No explicit parallelisation techniques were used for these calculations.

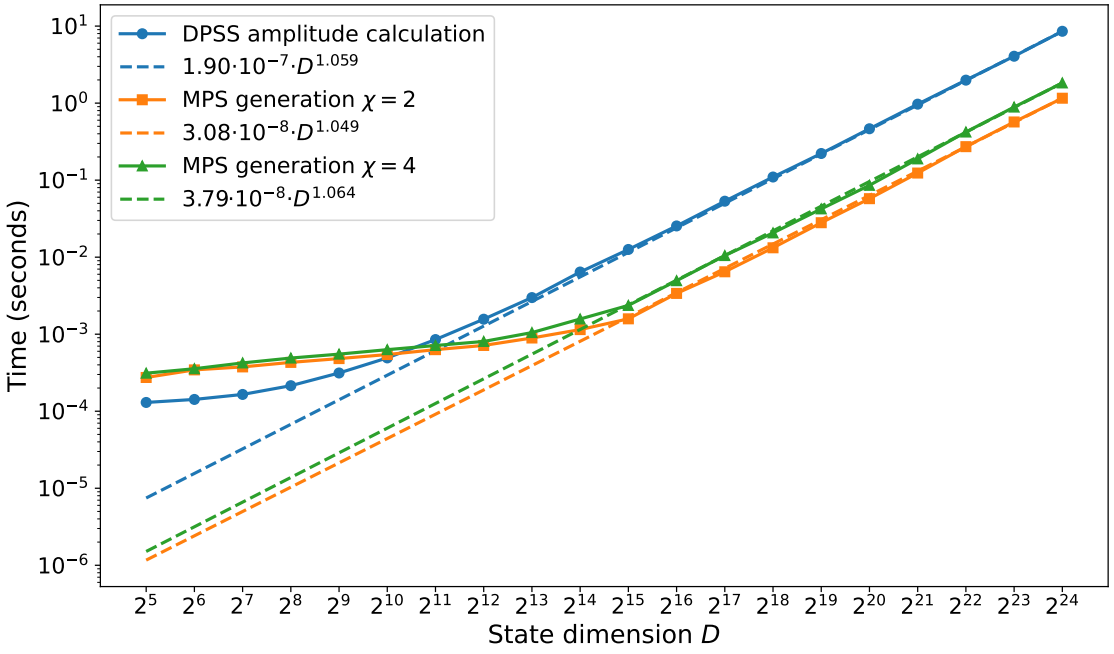

Figure 12: Classical processing time for generating the DPSS amplitudes and MPS operators. The fitted lines are approximately linear in D.

C Example calculation of MPS operators

Section 2.3 explains how a quantum circuit can be constructed to prepare a matrix product state that approximates a target state. To illustrate this in more detail, we work through an example using the DPSS state with dimension $D = 32$, configured to be optimal for a 99% confidence level. Here the bond dimension is $\chi = 4$, which is sufficient to prepare the state exactly with no truncation of singular values involved.

First, we perform the successive applications of reshaping and SVD, as shown in Eq. (18). We repeat this until we have retrieved all $V^{\dagger[k]}$, which correspond to the rightmost

matrices in Eqs. (38)-(41) below.

$$\begin{bmatrix} 0.05 & 0.07 \\ 0.08 & 0.10 \\ 0.12 & 0.13 \\ 0.15 & 0.17 \\ 0.18 & 0.20 \\ 0.21 & 0.22 \\ 0.23 & 0.24 \\ 0.24 & 0.25 \\ 0.25 & 0.24 \\ 0.24 & 0.23 \\ 0.22 & 0.21 \\ 0.20 & 0.18 \\ 0.17 & 0.15 \\ 0.13 & 0.12 \\ 0.10 & 0.08 \\ 0.07 & 0.05 \end{bmatrix} = \begin{bmatrix} -0.08 & -0.01 \\ -0.13 & -0.01 \\ -0.17 & -0.01 \\ -0.22 & -0.01 \\ -0.27 & -0.01 \\ -0.31 & -0.01 \\ -0.33 & -0.01 \\ -0.35 & 0.00 \\ -0.35 & 0.00 \\ -0.33 & 0.01 \\ -0.31 & 0.01 \\ -0.27 & 0.01 \\ -0.22 & 0.01 \\ -0.17 & 0.01 \\ -0.13 & 0.01 \\ -0.08 & 0.01 \end{bmatrix} \begin{bmatrix} -0.71 & -0.71 \\ 0.71 & -0.71 \end{bmatrix} \tag{38}$$

$$\begin{bmatrix} -0.08 & -0.01 & -0.13 & -0.01 \\ -0.17 & -0.01 & -0.22 & -0.01 \\ -0.27 & -0.01 & -0.31 & -0.01 \\ -0.33 & -0.01 & -0.35 & 0.00 \\ -0.35 & 0.00 & -0.33 & 0.01 \\ -0.31 & 0.01 & -0.27 & 0.01 \\ -0.22 & 0.01 & -0.17 & 0.01 \\ -0.13 & 0.01 & -0.08 & 0.01 \end{bmatrix} = \begin{bmatrix} -0.15 & 0.03 & 0.00 & 0.00 \\ -0.28 & 0.04 & 0.00 & 0.00 \\ -0.41 & 0.03 & 0.00 & 0.00 \\ -0.48 & 0.01 & 0.00 & 0.00 \\ -0.48 & -0.01 & 0.00 & 0.00 \\ -0.41 & -0.03 & 0.00 & 0.00 \\ -0.28 & -0.04 & 0.00 & 0.00 \\ -0.15 & -0.03 & 0.00 & 0.00 \end{bmatrix} \begin{bmatrix} 0.71 & 0.00 & 0.71 & 0.00 \\ 0.63 & -0.32 & -0.63 & -0.32 \\ 0.00 & -0.71 & 0.00 & 0.71 \\ -0.32 & -0.63 & 0.32 & -0.63 \end{bmatrix} \tag{39}$$

$$\begin{aligned} &\begin{bmatrix} -0.15 & 0.03 & 0.00 & 0.00 & -0.28 & 0.04 & 0.00 & 0.00 \\ -0.41 & 0.03 & 0.00 & 0.00 & -0.48 & 0.01 & 0.00 & 0.00 \\ -0.48 & -0.01 & 0.00 & 0.00 & -0.41 & -0.03 & 0.00 & 0.00 \\ -0.28 & -0.04 & 0.00 & 0.00 & -0.15 & -0.03 & 0.00 & 0.00 \end{bmatrix} \\ &= \begin{bmatrix} -0.30 & -0.11 & 0.01 & 0.00 \\ -0.63 & -0.06 & 0.00 & 0.00 \\ -0.63 & 0.06 & 0.00 & 0.00 \\ -0.30 & 0.11 & 0.01 & 0.00 \end{bmatrix} \begin{bmatrix} 0.71 & -0.01 & 0.00 & 0.00 & 0.71 & 0.01 & 0.00 & 0.00 \\ -0.62 & -0.34 & 0.00 & 0.00 & 0.62 & -0.34 & 0.00 & 0.00 \\ -0.01 & -0.69 & -0.15 & 0.00 & -0.01 & 0.69 & -0.15 & 0.00 \\ 0.24 & -0.45 & -0.49 & -0.05 & -0.24 & -0.45 & 0.49 & -0.05 \end{bmatrix} \end{aligned} \tag{40}$$

$$\begin{aligned} &\begin{bmatrix} -0.30 & -0.11 & 0.01 & 0.00 & -0.63 & -0.06 & 0.00 & 0.00 \\ -0.63 & 0.06 & 0.00 & 0.00 & -0.30 & 0.11 & 0.01 & 0.00 \end{bmatrix} \\ &= \begin{bmatrix} -0.66 & -0.26 \\ -0.66 & 0.26 \end{bmatrix} \begin{bmatrix} 0.71 & 0.04 & 0.00 & 0.00 & 0.71 & -0.04 & 0.00 & 0.00 \\ -0.63 & 0.33 & -0.02 & 0.00 & 0.63 & 0.33 & 0.02 & 0.00 \end{bmatrix} \end{aligned} \tag{41}$$

Finally, we can retrieve $V^{\dagger[1]}$ as the reshaped remainder matrix. The results are

$$V^{\dagger[5]} = \begin{bmatrix} -0.71 & -0.71 \\ 0.71 & -0.71 \end{bmatrix} \tag{42}$$

$$V^{\dagger[4]} = \begin{bmatrix} 0.71 & 0.00 & 0.71 & 0.00 \\ 0.63 & -0.32 & -0.63 & -0.32 \\ 0.00 & -0.71 & 0.00 & 0.71 \\ -0.32 & -0.63 & 0.32 & -0.63 \end{bmatrix} \tag{43}$$

$$V^{\dagger[3]} = \begin{bmatrix} 0.71 & -0.01 & 0.00 & 0.00 & 0.71 & 0.01 & 0.00 & 0.00 \\ -0.62 & -0.34 & 0.00 & 0.00 & 0.62 & -0.34 & 0.00 & 0.00 \\ -0.01 & -0.69 & -0.15 & 0.00 & -0.01 & 0.69 & -0.15 & 0.00 \\ 0.24 & -0.45 & -0.49 & -0.05 & -0.24 & -0.45 & 0.49 & -0.05 \end{bmatrix} \tag{44}$$

$$V^{\dagger[2]} = \begin{bmatrix} 0.71 & 0.04 & 0.00 & 0.00 & 0.71 & -0.04 & 0.00 & 0.00 \\ -0.63 & 0.33 & -0.02 & 0.00 & 0.63 & 0.33 & 0.02 & 0.00 \end{bmatrix} \tag{45}$$

$$V^{\dagger[1]} = \begin{bmatrix} -0.66 & -0.26 & -0.66 & 0.26 \end{bmatrix} \tag{46}$$

To retrieve the operators $M^{[k]}$ to be used in the quantum circuit, we follow the procedure described in Section 2.3:

1. Apply the conjugate transpose to retrieve $V^{[k]} = (V^{\dagger[k]})^{\dagger}$.
2. Pad the matrices $V^{[k]}$ so that they are square. Here we will pad them with values denoted as $*$, which indicate that the actual values corresponding to the operator implemented by the quantum circuit do not affect the final result. Instead of padding on one side, pad equally between the existing columns. This ensures the correct ordering of the columns for $M^{[k]}$ as shown in Figure 2.
3. Reorder the rows by making the last bit of each row the first bit. For example, row 6 becomes row 3 because $110 \rightarrow 011$. This achieves the swapping of s_k with α_k as shown in Figure 2.

The result of performing these steps is shown below:

$$M^{[5]} = \begin{bmatrix} -0.71 & 0.71 \\ -0.71 & -0.71 \end{bmatrix} \tag{47}$$

$$M^{[4]} = \begin{bmatrix} 0.71 & 0.63 & 0.00 & -0.32 \\ 0.00 & -0.32 & -0.71 & -0.63 \\ 0.71 & -0.63 & 0.00 & 0.32 \\ 0.00 & -0.32 & 0.71 & -0.63 \end{bmatrix} \tag{48}$$

$$M^{[3]} = \begin{bmatrix} 0.71 & * & -0.62 & * & -0.01 & * & 0.24 & * \\ -0.01 & * & -0.34 & * & -0.69 & * & -0.45 & * \\ 0.00 & * & 0.00 & * & -0.15 & * & -0.49 & * \\ 0.00 & * & 0.00 & * & 0.00 & * & -0.05 & * \\ 0.71 & * & 0.62 & * & -0.01 & * & -0.24 & * \\ 0.01 & * & -0.34 & * & 0.69 & * & -0.45 & * \\ 0.00 & * & 0.00 & * & -0.15 & * & 0.49 & * \\ 0.00 & * & 0.00 & * & 0.00 & * & -0.05 & * \end{bmatrix} \tag{49}$$

$$M^{[2]} = \begin{bmatrix} 0.71 & * & * & * & -0.63 & * & * & * \\ 0.04 & * & * & * & 0.33 & * & * & * \\ 0.00 & * & * & * & -0.02 & * & * & * \\ 0.00 & * & * & * & 0.00 & * & * & * \\ 0.71 & * & * & * & 0.63 & * & * & * \\ -0.04 & * & * & * & 0.33 & * & * & * \\ 0.00 & * & * & * & 0.02 & * & * & * \\ 0.00 & * & * & * & 0.00 & * & * & * \end{bmatrix} \tag{50}$$

$$M^{[1]} = \begin{bmatrix} -0.66 & * & * & * \\ -0.26 & * & * & * \\ -0.66 & * & * & * \\ 0.26 & * & * & * \end{bmatrix} \tag{51}$$

The original state can then be prepared with the circuit shown in Figure 3, which performs the following operation:

$$|\psi\rangle = (\mathbb{1}_{16} \otimes M^{[5]})(\mathbb{1}_8 \otimes M^{[4]})(\mathbb{1}_4 \otimes M^{[3]})(\mathbb{1}_2 \otimes M^{[2]} \otimes \mathbb{1}_2)(M^{[1]} \otimes \mathbb{1}_8)\,|0\rangle\,. \tag{52}$$

It is straightforward to numerically check that this reproduces the original state (the vector corresponding to the left hand side of Eq. (38)). Code for performing all of these calculations in available on Github [82].