

Supplementary Materials: Parallel Quantum Simulation of Large Systems on Small NISQ Computers

Fergus Barratt,¹ James Dborin,² Matthias Bal,³ Vid Stojovic,³ Frank Pollmann,⁴ and A. G. Green²

¹*Department of Mathematics, King's College London, Strand, London WC2R 2LS, UK*

²*London Centre for Nanotechnology, University College London, Gordon St., London, WC1H 0AH, UK*

³*GTN Limited, Clifton House, 46 Clifton Terrace, Finsbury Park, London N4 3JP, UK*

⁴*Department of Physics, T42, Technische Universität München,*

James-Franck-Straße 1, D-85748 Garching, Germany

(Dated: April 14, 2021)

In this supplementary material we further explain the methods used in the quantum circuit implementation of iMPS algorithms, and the effects of imperfections in that realisation. We begin with a summary of the main ideas of MPS and how they translate to quantum circuits. After this, we show how these formal circuit equations can be realised in practice. We give details of appropriate ansätze and an optimization method that we have found effective. Next we detail the natural generalisation to higher bond order matrix product states, before giving additional data on the effects of shallow circuit realisations and finite gate fidelity.

I. SUPPLEMENTARY METHODS

A. Quantum Circuit Realisations of MPS

Matrix product state (MPS) techniques are now canonical in the condensed matter physics community. There are many fine reviews[1–3]. Here we give a condensed summary of the additional details relevant to this work and how they translate to quantum circuits.

1. Matrix Product States

An MPS state for a spin-1/2 chain is written

$$|\psi\rangle = \sum_{\{\sigma_\alpha\}} \sum_{\{i_\alpha\}} A_{i_1}^{\sigma_1} A_{i_1, i_2}^{\sigma_2} A_{i_2, i_3}^{\sigma_3} \dots A_{i_N}^{\sigma_N} |\sigma_1, \sigma_2, \sigma_3, \dots, \sigma_N\rangle \\ = \begin{array}{c} \text{Penrose diagram} \end{array} \quad (1)$$

where we use a penrose diagrammatic representation in the second line[1–3]. The tensor indices i_1, i_2, \dots, i_N run over values $1 \rightarrow D_i$. for a spin -1/2 system these states cover the Hilbert space when $D_i = 2^{\min(i, N-i)}$. In practice, D_i is usually truncated at some value.

Expectations of local operators involve contracting the state along the whole chain and are nonlocal;

$$\langle \psi | \hat{\theta} | \psi \rangle = \dots \begin{array}{c} \text{Penrose diagram} \end{array} \dots \quad (2)$$

However, identifying the result of contracting everything to the left of the site α as l_α and to the right as r_α we

may write this as a local expectation:

$$\langle \psi | \hat{\theta} | \psi \rangle = l_\alpha^{\alpha} \hat{\theta} r_{\alpha+1}^{\alpha+1}, \quad (3)$$

$$l_\alpha^{\alpha} = \begin{array}{c} \text{Penrose diagram} \end{array} \dots = l_{\alpha-1}^{\alpha-1}, \quad (4)$$

$$r_\alpha^{\alpha+1} = \begin{array}{c} \text{Penrose diagram} \end{array} \dots = r_{\alpha+1}^{\alpha+1}, \quad (5)$$

The tensors l_α and r_α are often called the environment for the site α and we will use that nomenclature here. Using a sequential singular-value decomposition from left to right of Eq.(1) allows the tensor to be put in a canonical form - or more precisely, left orthogonal form - such that $l_\alpha \equiv I_{D_{i-1} \otimes D_{i-1}} \forall \alpha$. This *isometric form* is the basis of our mapping to the quantum circuit shown in Fig.6.

2. Translation to Quantum Circuit

A quantum circuit realisation of Supplementary Eq.(1) is given by

$$|\psi\rangle = \begin{array}{c} \text{Quantum circuit} \end{array} . \quad (6)$$

Notice by comparing with Fig.6a in the main paper that the bond order increases and reduces at the ends of the chain. We have truncated the bond order at $D = 4$ in this equation. We note also that the greyed-out unitary at

the end of the chain is redundant, *i.e.* it may be replaced with the identity. It represents the MPS tensors on the last site in this bond order 4 case. There are $n - 1$ such redundant unitaries at bond order $D = 2^n$.

The circuits for the left environment l_α can be written

$$l_\alpha = \begin{array}{c} \text{Quantum Circuit Diagram} \\ \dots \\ \text{Quantum Circuit Diagram} \\ = \text{Simpler Quantum Circuit Diagram} \end{array} \quad (7)$$

As indicated, the unitaries cancel in conjugate pairs and the left environment is trivial reflecting the left-orthogonal form of the MPS state that the circuit represents. The right environment can similarly be expressed by the following circuits

$$r_\alpha = \begin{array}{c} \text{Quantum Circuit Diagram} \\ \dots \\ \text{Quantum Circuit Diagram} \\ = \begin{array}{c} |0\rangle \xrightarrow{V_\alpha} |0\rangle \\ |0\rangle \xrightarrow{V_\alpha} |0\rangle \\ |0\rangle \xrightarrow{V_\alpha} |0\rangle \\ |0\rangle \xrightarrow{V_\alpha} |0\rangle \end{array} = \begin{array}{c} |0\rangle \xrightarrow{R_\alpha} |0\rangle \\ |0\rangle \xrightarrow{R_\alpha} |0\rangle \\ |0\rangle \xrightarrow{R_\alpha} |0\rangle \\ |0\rangle \xrightarrow{R_\alpha} |0\rangle \end{array} . \end{array} \quad (8)$$

The existence of the compressed tensors V is the basis of our construction. It is guaranteed as follows: The right environment is Hermitian and has eigen-decomposition $r = U\Lambda U^\dagger$. Define $X = U\sqrt{\Lambda}$, and embed X across the legs of the unitary V by assigning X , reshaped into a vector, to the first column of V - since V is always applied to a state $|0\rangle^{\otimes N}$, these are the only entries of V that feature in any calculation. The other columns of V are chosen orthonormally and arbitrarily. We can choose to compress further by choosing only the k highest eigenvalue eigenvector pairs to construct U and Λ . The resulting environment $U'\Lambda'U'^\dagger$ is the rank- k matrix that best approximates r in Frobenius norm. This is the circuit realisation of the truncation scheme habitually applied in MPS calculations. In essence our work uses a variational scheme to determine V .

A very similar argument can be used to produce the tensor R : The right environment r is embedded in the appropriate diagonal block of R the remaining elements - upon which no computation depends - are completed by an appropriate singular valued decomposition.

From Supplementary Eq.(8) it is evident that we iteratively construct r_α or V_α from $r_{\alpha-1}$ or $V_{\alpha-1}$ according

to

$$\begin{array}{c} |0\rangle \xrightarrow{V_\alpha} |0\rangle \\ |0\rangle \xrightarrow{U_{\alpha-1}} |0\rangle \\ |0\rangle \xrightarrow{U_{\alpha-1}^\dagger} |0\rangle \\ |0\rangle \xrightarrow{V_\alpha^\dagger} |0\rangle \end{array} = \begin{array}{c} |0\rangle \xrightarrow{V_\alpha} |0\rangle \\ |0\rangle \xrightarrow{V_\alpha^\dagger} |0\rangle \end{array}. \quad (9)$$

or equivalently

$$\begin{array}{c} |0\rangle \xrightarrow{R_\alpha} |0\rangle \\ |0\rangle \xrightarrow{U_{\alpha-1}} |0\rangle \\ |0\rangle \xrightarrow{U_{\alpha-1}^\dagger} |0\rangle \\ |0\rangle \xrightarrow{R_{\alpha-1}} |0\rangle \end{array} = \begin{array}{c} |0\rangle \xrightarrow{R_\alpha} |0\rangle \\ |0\rangle \xrightarrow{R_{\alpha-1}} |0\rangle \end{array}. \quad (10)$$

These are the quantum circuit counterparts of the iterative expression for the MPS environment contained in the last part of Supplementary Eq.(5). In *translationally invariant systems*, V (and R) is independent of the site on the chain and Supplementary Eq.(10) reduces to the fixed point equation seen in Fig. 2 c.

3. Time-evolving Quantum Circuit MPS

The basis of our time-evolution algorithm is to find the state $|\psi(X(t+dt))\rangle$, parametrised by the set of parameters $X(t+dt)$, whose fidelity with the time-evolution of the state $|\psi(X(t))\rangle$ is optimised:

$$\max_{X(t+dt)} \langle \psi(X(t+dt)) | e^{iHdt} | \psi(X(t)) \rangle^2.$$

This overlap can be written for the $D = 4$ quantum circuit MPS as

$$\langle \psi(X(t+dt)) | e^{iHdt} | \psi(X(t)) \rangle$$

$$\begin{array}{c} \vdots \\ \text{Quantum Circuit Diagram} \\ \dots \\ \text{Quantum Circuit Diagram} \\ = \begin{array}{c} |0\rangle \xrightarrow{W_{\text{odd}}} |0\rangle \\ |0\rangle \xrightarrow{W_{\text{odd}}} |0\rangle \\ |0\rangle \xrightarrow{W_{\text{odd}}} |0\rangle \\ |0\rangle \xrightarrow{W_{\text{odd}}} |0\rangle \\ |0\rangle \xrightarrow{W_{\text{even}}} |0\rangle \\ |0\rangle \xrightarrow{W_{\text{even}}} |0\rangle \\ |0\rangle \xrightarrow{W_{\text{even}}} |0\rangle \\ |0\rangle \xrightarrow{W_{\text{even}}} |0\rangle \\ |0\rangle \xrightarrow{U'_\alpha} |0\rangle \\ |0\rangle \xrightarrow{U'_\alpha} |0\rangle \\ |0\rangle \xrightarrow{U'_\alpha} |0\rangle \\ |0\rangle \xrightarrow{U'_\alpha} |0\rangle \\ |0\rangle \xrightarrow{U_\alpha} |0\rangle \\ |0\rangle \xrightarrow{U_\alpha} |0\rangle \\ |0\rangle \xrightarrow{U_\alpha} |0\rangle \\ |0\rangle \xrightarrow{U_\alpha} |0\rangle \\ |0\rangle \xrightarrow{R_\alpha} |0\rangle \\ |0\rangle \xrightarrow{R_\alpha} |0\rangle \\ |0\rangle \xrightarrow{R_\alpha} |0\rangle \\ |0\rangle \xrightarrow{R_\alpha} |0\rangle \end{array} \quad (11) \end{array}$$

where we have expanded the time-evolution operator in the usual second order Trotterization as

$$\begin{aligned} \exp[iHdt] &= \exp[iH_{\text{odd}}dt/2] \exp[iH_{\text{even}}dt] \exp[iH_{\text{odd}}dt/2] \\ &= W_{\text{odd}} W_{\text{even}} W_{\text{odd}}. \end{aligned} \quad (12)$$

The unitaries U_α encode the state at time t . When the modulus of this overlap is optimised, the unitaries U'_α encode the state at time $t + dt$.

The circuit shown in Supplementary Eq.(11) can be implemented on a quantum computer that is much

smaller than the system under consideration by splitting along the red dashed lines. The resulting circuit takes the form

$$\langle \psi(X(t+dt)) | e^{iHdt} | \psi(X(t)) \rangle = \begin{array}{c} \text{[Circuit Diagram]} \\ \text{[Circuit Diagram]} \end{array}, \quad (13)$$

The circuit diagram shows a sequence of qubits (labeled |0>) connected by horizontal lines. A red dashed line splits the circuit into two parts: the top part above the dashed line contains a red block labeled R_α , and the bottom part below the dashed line contains a green block labeled $L_{\alpha-1}$. Between these blocks are several blue boxes labeled U_α and $U'_{\alpha-1}$, and green boxes labeled W_{even} and W_{odd} . There are also H gates and CNOT gates between the blocks and the environment.

where the mixed environment circuits (so called because U and U' are not the same) R_α and L_α are the portions of the circuit above and below the red dashed lines in Supplementary Eq.(11), compressed to tensors R_α and L_α as in Supplementary Eq.(8). Following the same procedure as in Supplementary Eq.(10), R_α and L_α can be determined iteratively from site to site from the equations

$$\begin{array}{c} \text{[Circuit Diagram]} \\ \text{[Circuit Diagram]} \end{array} = \begin{array}{c} \text{[Circuit Diagram]} \\ \text{[Circuit Diagram]} \end{array}, \quad (14)$$

The circuit diagram shows the iterative update of $R_{\alpha-1}$ and R_α . It consists of two main parts: the top part contains a red block labeled $R_{\alpha-1}$ followed by a sequence of blocks U_α , $U'_{\alpha-1}$, W_{odd} , W_{even} , U_α , and $U'_{\alpha-1}$; the bottom part contains a green block labeled $L_{\alpha-1}$ followed by the same sequence of blocks. The circuit is split into two parts by a red dashed line, with the top part being $R_{\alpha-1}$ and the bottom part being $L_{\alpha-1}$.

and

$$\begin{array}{c} \text{[Circuit Diagram]} \\ \text{[Circuit Diagram]} \end{array} = \begin{array}{c} \text{[Circuit Diagram]} \\ \text{[Circuit Diagram]} \end{array}. \quad (15)$$

The circuit diagram shows the iterative update of L_α . It consists of two main parts: the top part contains a green block labeled L_α followed by a sequence of blocks U_α , $U'_{\alpha-1}$, W_{odd} , W_{even} , U_α , and $U'_{\alpha-1}$; the bottom part contains a green block labeled $L_{\alpha-1}$ followed by the same sequence of blocks. The circuit is split into two parts by a green dashed line, with the top part being L_α and the bottom part being $L_{\alpha-1}$.

When Supplementary Eq.(13) is optimised alongside Supplementary Eqs.(14) and (15), the resulting evolution is equivalent to the TDVP equations over the variational parameters of U . Indeed, as noted in the Methods, setting the derivative of Supplementary Eq.(13) with respect to U' equal to zero recovers the usual TDVP equations. The present implementation benefits from the encoding of

the tangent space to the variational manifold in addition to the state in the unitaries U_α .

For translationally invariant systems, the environment unitaries R_α and L_α become independent of site and the iterative updates in Supplementary Eqs.(14) and (15) reduce to the fixed point equations giving the highest weight right and left eigenvectors of the transfer matrix

$$E_{U'}^U = \begin{array}{c} \text{[Circuit Diagram]} \\ \text{[Circuit Diagram]} \end{array}. \quad (16)$$

The circuit diagram shows the fixed-point equation $E_{U'}^U$. It consists of two main parts: the top part contains a red block labeled W_{even} followed by a sequence of blocks U_α , $U'_{\alpha-1}$, W_{odd} , W_{odd} , U_α , and $U'_{\alpha-1}$; the bottom part contains a green block labeled W_{odd} followed by the same sequence of blocks. The circuit is split into two parts by a red dashed line, with the top part being W_{even} and the bottom part being W_{odd} .

At this point we make a further simplification to allow for easier implementation on NISQ devices. We reduce to a first order Trotterization scheme take account of only $\exp[iH_{\text{odd}}dt]$. Taking advantage of the translational invariance of U and U' appears to reduce these errors somewhat. With this simplification the circuits in Supplementary Eqs.(13), (14) and (15) reduce to

$$\langle \psi(X(t+dt)) | e^{iHdt} | \psi(X(t)) \rangle = \begin{array}{c} \text{[Circuit Diagram]} \\ \text{[Circuit Diagram]} \end{array}, \quad (17)$$

The circuit diagram shows the simplified fixed-point equation. It consists of two main parts: the top part contains a red block labeled R_α followed by a sequence of blocks U_α , $U'_{\alpha-1}$, W_{odd} , W_{odd} , U_α , and $U'_{\alpha-1}$; the bottom part contains a green block labeled $L_{\alpha-1}$ followed by the same sequence of blocks. The circuit is split into two parts by a red dashed line, with the top part being R_α and the bottom part being $L_{\alpha-1}$.

for the overlap, and

$$\begin{array}{c} \text{[Circuit Diagram]} \\ \text{[Circuit Diagram]} \end{array} = \begin{array}{c} \text{[Circuit Diagram]} \\ \text{[Circuit Diagram]} \end{array}, \quad (18)$$

The circuit diagram shows the simplified fixed-point equation. It consists of two main parts: the top part contains a red block labeled $R_{\alpha-1}$ followed by a sequence of blocks U_α , $U'_{\alpha-1}$, W_{odd} , W_{odd} , U_α , and $U'_{\alpha-1}$; the bottom part contains a red block labeled R_α followed by the same sequence of blocks. The circuit is split into two parts by a red dashed line, with the top part being $R_{\alpha-1}$ and the bottom part being R_α .

and

$$\begin{array}{c} \text{[Circuit Diagram]} \\ \text{[Circuit Diagram]} \end{array} = \begin{array}{c} \text{[Circuit Diagram]} \\ \text{[Circuit Diagram]} \end{array}. \quad (19)$$

The circuit diagram shows the simplified fixed-point equation. It consists of two main parts: the top part contains a green block labeled L_α followed by a sequence of blocks U_α , $U'_{\alpha-1}$, W_{odd} , W_{odd} , U_α , and $U'_{\alpha-1}$; the bottom part contains a green block labeled $L_{\alpha-1}$ followed by the same sequence of blocks. The circuit is split into two parts by a green dashed line, with the top part being L_α and the bottom part being $L_{\alpha-1}$.

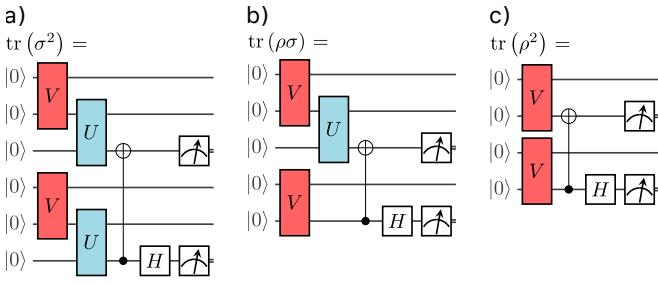
for the fixed-point equations.

The simulations presented in the main body of the paper are carried out for $D = 2$. This is mainly to obtain compact circuits that are feasible to implement on available (or imminently available) NISQ machines. Our code (in both Cirq and Qiskit) can run with arbitrary bond order. Restricting to bond order $D = 2$ recovers the circuits shown in Fig. 3 in the main text.

B. Implementing Quantum Circuit MPS

1. Finding the Environment

In order to determine the environment on the quantum chip we need to implement the equality of Fig. 2 from the main paper. Supplementary Fig. 1 details the quantum circuits required to do so. We use the SWAP test [4] to compare both sides of the equation. The minimal application of this is illustrated by the circuit shown in Supplementary Fig. 1b. However, the overlap $F(\rho, \sigma) = \text{tr}(\rho\sigma)$ is not necessarily maximised by $\rho = \sigma$. In fact, $F(\rho, \rho) = \text{tr}(\rho^2)$ is the purity, so that if the reduced density matrices of Supplementary Fig. 1b are mixed. The optimizer will incorrectly try to increase the purity. We circumvent this by using a quantity related to the trace distance $\text{tr}((\rho - \sigma)^\dagger(\rho - \sigma))$, which is minimized at $\rho = \sigma$. The additional circuits required to determine the trace distance are shown in Supplementary Figs. 1a and c.



Supplementary Figure 1. Quantum circuits to find the environment: In order to determine an approximation to the overlap, the fraction of 00 outputs is measured on the measurement symbols.

2. Finding the Overlap

In order to time evolve a matrix product state on a quantum computer, we minimize the overlap between the perfect evolution of the state from time t to $t + \delta t$ and the manifold of matrix product states of the available bond dimension. In the infinite translationally invariant case, the definition of the overlap requires care. It can be given as the largest eigenvalue of the *mixed transfer matrix*.

The overlap as largest eigenvalue: The overlap is the result of applying the matrix of Supplementary Eq.(16) an infinite number of times to boundary vectors. The result (with appropriate normalisation) is equal to the largest eigenvalue. In order to find the largest eigenvalue, we use a variational method to find the largest eigenvalue-eigenvector pair. Since the matrix in Supplementary Eq.(16) is not Hermitian, we cannot use the Rayleigh-Ritz variational principle. Instead, we minimize the distance between the result of applying the transfer matrix

to a vector, and the product of that vector with a candidate eigenvalue. Denoting the matrix of Supplementary Eq. 16 as $E_{U'}^U$, we seek to solve:

$$\max_{\eta} \min_r \|E_{U'}^U r - \eta r\|^2. \quad (20)$$

In the applications considered in this work, we need only the overlap between very similar states, and can take advantage of the fact that we know that η can deviate from 1 only by a correction of order δt . As a result, we have found it effective to apply a heuristic, wherein we minimise over both η and r , using an appropriate initial condition for $\eta \sim 1$, and restart the optimization if it fails to attain $\eta \sim 1$ at its conclusion:

$$\min_{\eta, r} \|E_{U'}^U r - \eta r\|^2 = \min_{\eta, r} v(\eta, r), \quad (21)$$

where we have defined our objective function $v(\eta, r)$. In practice, only rarely are repetitions required to find the largest eigenvalue and even then a small number of repetitions suffices (See Supplementary Fig. 2).

Solving Supplementary Eq. (21) on a quantum circuit: Expanding Supplementary Eq. (21) allows us to put it in a form that can be implemented as quantum circuits:

$$v(\eta, r) = r^\dagger E_{U'}^U E_{U'}^U r + |\eta|^2 r^\dagger r - r^\dagger E_{U'}^U r - r^\dagger E_{U'}^U r. \quad (22)$$

The first two terms can be implemented on quantum circuits as shown in Figs. 3d and 3f (a and b for the corresponding left eigenvectors). The objective function is found by measuring in the computational basis, determining the probability of the bit string of all 0s, and taking the square root of the corresponding probability (since we know the result should be real, there is no problem determining the phase). The same is not true of the third and fourth terms of Supplementary Eq. (22), for which there is an undetermined phase. At the minimum, the sum of the last two terms will be equal to the largest eigenvalue, which is real to $O(dt^2)$, since the largest eigenvalue of $E_{U'}^U$ is 1. Therefore, if we minimize the associated objective function:

$$v'(\eta, r) = r^\dagger E_{U'}^U E_{U'}^U r + |\eta|^2 r^\dagger r - 2|r^\dagger E_{U'}^U r|, \quad (23)$$

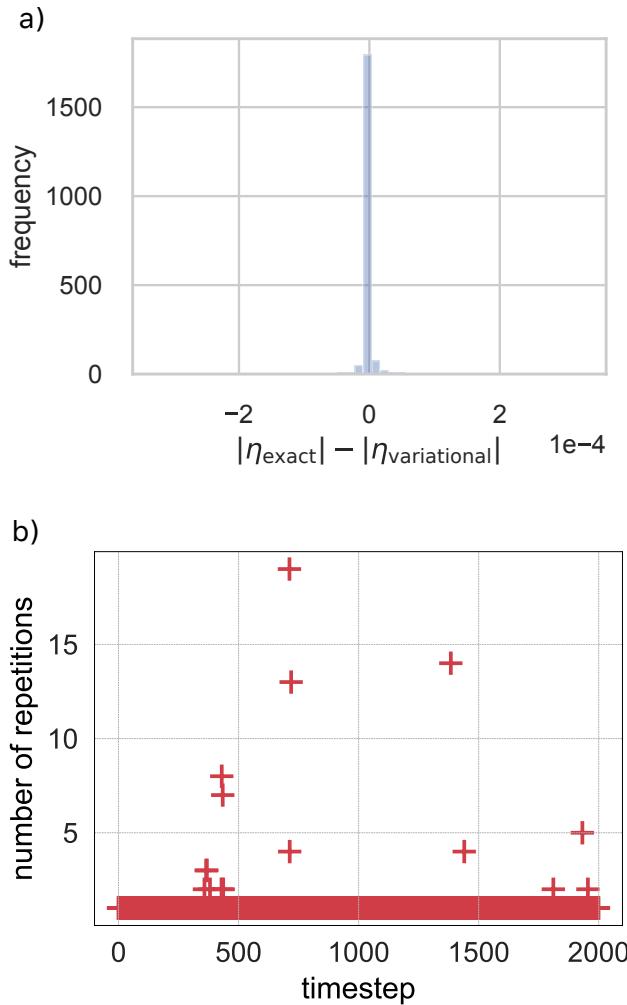
by taking the square root of the corresponding probabilities (Supplementary Fig. 3 b and e), the argmin will coincide with that of Supplementary Eq. (21) (to $O(dt^2)$), and we can determine the eigenvalue and eigenvector. In practise, we often achieve an accuracy much greater than $O(dt^2)$. Supplementary Fig. (2) gives data demonstrating the accuracy and reliability of this method of determining the overlap.

3. Ansätze

The implementation of iMPS algorithms on quantum computers require factorisations of the large unitaries in

which the tensors are embedded. Different ansätze make different tradeoffs between gate depth, gate availability on typical QC implementations, and expressibility. The different unitaries (U and V , containing the state tensor and the environment tensor respectively), can be factorised in different ways, to take into account their different structures. At low bond dimension, full parametrisations of the important matrix spaces are possible.

The state unitary U : For this work, we used the generic ansatz for U reported in Fig. 4a in the main body of the paper. We found it to be a workable tradeoff between expressibility and depth in the generic case. However, ground state problems often have deeper structure in the form of symmetries, and that structure can be exploited



Supplementary Figure 2. Performance of finding the right environment with Supplementary Eq. (23) a) Distribution of results of minimizing Supplementary Eq. (23): a comparison of the exact and variational largest eigenvalue of the mixed transfer matrices for 2000 steps along a TDVP trajectory, with $\delta t = 0.01$, using the BFGS optimizer provided in `scipy`. b) Number of repetitions vs. time: Number of repetitions required at each timestep in (a).

for more efficient circuit ansätze, and more effective optimization.

As an example, the transverse field Ising model used as an example in the main text exhibits time reversal invariance, parity symmetry, and has a conserved quantity (the total magnetization). While in Fig. 4b in the main text we use the general ansatz of Fig. 4a,

These properties constrain the form of the true ground state of the system, and its MPS approximation. The transverse field Ising model Hamiltonian is real, and thus the model exhibits time reversal invariance. The MPS form of the ground state should be made up of real tensors [5]. In Supplementary Fig. 4, we demonstrate the resulting speedups for two different real ansätze. Note in particular that for the ansatz of Supplementary Fig. 4, the optimization produces optimal results (i.e. equivalent to $D=2$ classical), for a state tensor of only depth 4.

Representing the environment: The simulations presented in the main paper were performed at $D = 2$. At $D = 2$, the depth required for a full parametrisation of the environment is easily accessible, and a minimal parametrisation is available. The resulting circuits allow us to calculate important quantities (the Schmidt coefficients, for example) classically, and point the way to natural generalisations for $D > 2$.

A full parametrisation of the right environment at $D = 2$: The environment – the right fixed point, r , of the MPS transfer matrix [3] – is an Hermitian, positive definite matrix, with trace 1. V embeds the Cholesky decomposition of the right environment in a unitary. To obtain the right environment, we parametrise V as shown in Supplementary Fig. 5.

Since the environment r has trace 1, the Cholesky factor $r^{\frac{1}{2}}$ has Frobenius norm 1. First consider the matrix in Supplementary Fig. 5a. This spans all possible diagonal environment square roots, since $\text{tr}(\Lambda^\dagger \Lambda) = 1$, $\Lambda^\dagger \Lambda \geq 0$, and $\cos(\gamma)$ is surjective onto $[-1, 1]$.

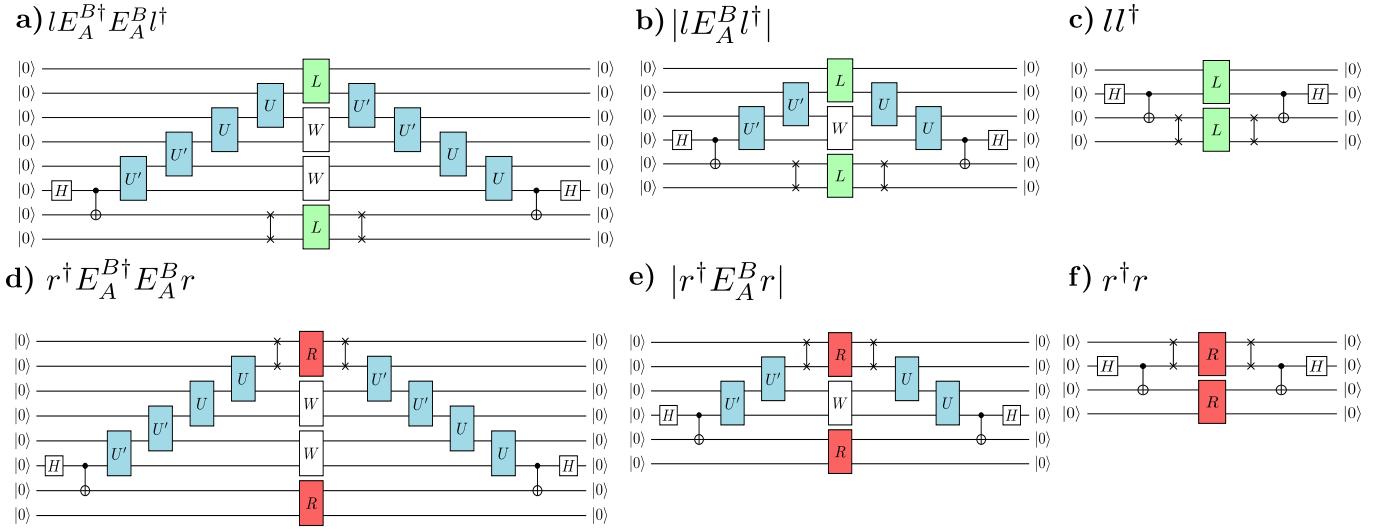
This parametrisation has some advantages:

- i. It is minimal, depending upon only 3 real parameters;
- ii. The eigenvalues of the environment are classically accessible. The purity, Von Neumann entanglement entropy etc. are all accessible with simple calculations if we know the parameters of the ansatz.

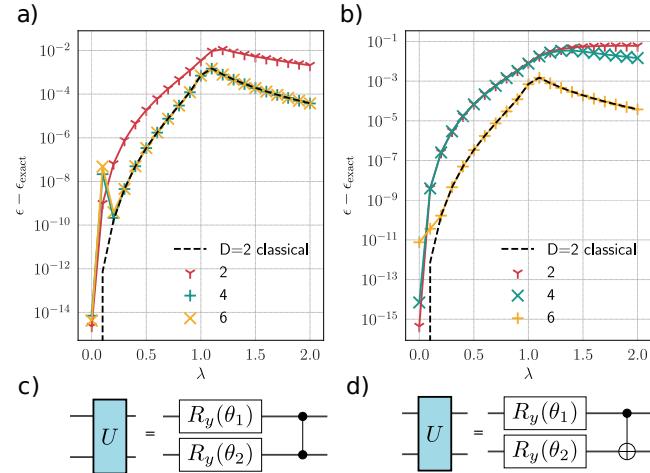
Parametrising the right environment at $D > 2$: It is not obvious how best to parametrise the right environment for higher bond dimension, for which we will not be able to efficiently access the full space of matrices. We have, however, found a natural generalisation of Supplementary Fig. 5 to be surprisingly effective in practise. The environment at any bond dimension can be unitarily diagonalised, with real, positive eigenvalues:

$$r = W \Lambda^2 W^\dagger. \quad (24)$$

We can assume that the unitary W can be absorbed by changing the parameters of the state tensor unitary U .



Supplementary Figure 3. **Diagrams required to determine the mixed environment:** Panels a), b) and c) show the circuits used to construct the left mixed environment and panels d), e) and f) those used to construct the right mixed environment. The probability of measuring the all 0s bit string is measured then the real square root is taken to calculate the labelled expression.



Supplementary Figure 4. **Time Reversal Symmetry in optimization of the transverse-field Ising model:** a) and b) Deviation from analytical ground state energy density of the transverse-field Ising model as a function of λ . Dashed line is the result of classical iDMRG at $D=2$, other lines/markers are the result of quantum variational optimization at different ansatz depths, given in the legend. c) and d) Ansätze used to produce figures a) and b), respectively.

This amounts to diagonalising the environment, an important step in the development of classical MPS algorithms [1]. The task that remains is to express the positive diagonal matrix Λ within a shallow quantum circuit, by embedding it across the legs of a unitary V :

$$(V|0\rangle)_{ij} = \Lambda_{ij}. \quad (25)$$

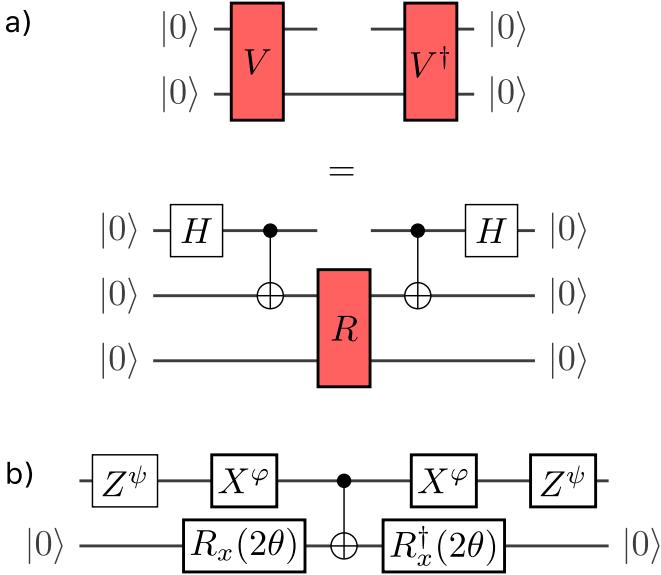
where the indices on the vector $V|0\rangle$ should be understood as denoting the elements of that vector when reshaped into a matrix.

$$\begin{aligned} \text{a)} \quad & \Lambda_{ij} = \begin{pmatrix} \cos(\gamma) & 0 \\ 0 & i \sin(\gamma) \end{pmatrix} = \begin{array}{c} |0\rangle \\ |0\rangle \end{array} \begin{array}{c} Y \\ Y^\gamma \end{array} \begin{array}{c} i \\ j \end{array} \\ \text{b)} \quad & W_{ij} = i \begin{array}{c} Z^\gamma \\ X^\psi \\ Z^\phi \end{array} \begin{array}{c} i \\ j \end{array} \\ \text{c)} \quad & [r^{\frac{1}{2}}]_{ij} = \begin{array}{c} |0\rangle \\ |0\rangle \end{array} \begin{array}{c} V \\ j \end{array} = \begin{array}{c} |0\rangle \\ |0\rangle \end{array} \begin{array}{c} YY^\gamma \\ X^\psi \\ Z^\phi \end{array} \begin{array}{c} i \\ j \end{array} \end{aligned}$$

Supplementary Figure 5. **Parametrisation of the environment V , used throughout:** a) shallow quantum circuit to represent the positive diagonal matrix used in Supplementary Eqs. (24) and (25). b) The right environment can be constructed from this using Supplementary Eq. (24) in which W is represented on the circuit as shown. c) The unitary V is constructed as shown.

For bond dimension 2 we have made use of the unitary $V = YY^\gamma$ to express this diagonal. At higher bond dimensions, V generates a map from qubits $(1, \dots, n) \rightarrow (n+1, \dots, 2n)$, where $D = 2^n$. We can apply the ansatz of Supplementary Fig. 5a to the all pairs of qubits $(i, n+i)$, $i \in [1, \dots, n]$, each with a different parameter γ_k . In doing so we will parametrise a subset of the possible diagonal matrices, of the form $\Lambda(\vec{\gamma}) = \otimes_{i=1}^n \Lambda(\gamma_k)$, where Λ_{ij} is the matrix of Supplementary Fig. 5a. Whilst this parametrisation no longer explores the full space of diagonal matrices, certain properties – the entanglement entropy, for example – can still be efficiently calculated from the parameters of the ansatz.

We have found such a constant depth ansatz for r to



Supplementary Figure 6. Alternative representation of the environment and its circuit realisation: a) The alternative representation of the environment, used in the TDVP circuits and b) the corresponding full parametrisation of the unitary R .

be effective for the problem of ground state finding, for all bond dimensions for which the algorithm can be effectively classically simulated. More work is required to produce theoretical guarantees that this should be the case in general.

Alternative representations of the environment: When considering the quantum circuits used to implement the time-dependent variational principle, it is convenient to use a different representation of the environment as indicated in Supplementary Fig. 6a. This restructuring allows us to trade depth for qubits, and makes manifest the time reversal symmetry of the TDVP circuits. It also allows us to explore non Hermitian tensors variationally. This is crucial in the calculation of the overlap. The parametrisation of the tensor R (Supplementary Fig. 6b) retains the benefits of the parametrisation of Supplementary Fig. 5c, in particular, the eigenvalues of the environment (the square of the Schmidt coefficients across each bond) are: $\lambda_1 = \cos^2(\theta)$, $\lambda_2 = \sin^2(\theta)$.

4. Optimization Methods

Variational algorithms require the use of effective optimization algorithms. In the main text, we have made use of several - the standard algorithms provided in `scipy`, and a coordinate optimization method named Rotosolve. The details of this algorithm — and its name — were presented in Ref. [6]. The same algorithm having been developed independently by several authors time[7–9] In

this section we detail a necessary modification to the Rotosolve algorithm, and compare the performance of each on our setup.

Doubled Rotosolve: Rotosolve is an optimization method for quantum circuits. As detailed in Ref. [6], it takes advantage of the fact that for quantum circuits $U(\vec{\theta})|0\rangle$, in which each θ_k parametrises a single exponential of a Pauli string ($e^{-i\theta_k S}$, $S^2 = I$), the expectation values of local observables must vary sinusoidally as a function of θ_k . The details of the resulting sinusoid can be determined by three measurements at different values of θ_k , and once determined, the minimum for that parameter (for the current values of the rest of $\vec{\theta}$) can be attained immediately. Analytic gradient methods like Rotosolve can lead to dramatic ($\sim 100\times$) reduction in the number of measurements required to optimize a circuit.

Rotosolve as introduced in Ref. [6] requires that each parameter control at most one gate. For circuits such as those shown in Figs. 1a-c), where each parameter controls more than one gate, a small modification to the Rotosolve algorithm is necessary (see also Ref.[9]).

Doubly Sinusoidal Expectation Values: A derivation of the original Rotosolve algorithm is given in the appendix of Ref. [6]. Here we present a brief extension of this derivation to the situation where a given parameter controls two gates. Consider a generic quantum circuit, in which each element is the parametrised exponential $U_i = e^{-i\theta_i S_i}$ of some Pauli string S_i , $i \in [1 \dots N]$, $S_i^2 = I$. Consider the expectation value of an operator \hat{O} in the output of this circuit, as a function of one of the θ_i , in a circuit consisting of N gates depending on $M < N$ parameters, where at most two identical gates U_k depend upon the same parameter θ_k . Using the expansion $U_k(\theta_k) = e^{-i\theta_k S_k} = \cos(\theta_k/2)\mathbb{I} + \sin(\theta_k/2)S_k$, and defining the quantities:

$$A = \langle \hat{O} \rangle_{\theta_k=0} + \langle \hat{O} \rangle_{\theta_k=\pi}, \quad (26)$$

$$B = \langle \hat{O} \rangle_{\theta_k=0} - \langle \hat{O} \rangle_{\theta_k=\pi}, \quad (27)$$

$$C = \langle \hat{O} \rangle_{\theta_k=\pi/2} + \langle \hat{O} \rangle_{\theta_k=-\pi/2}, \quad (28)$$

$$D = \langle \hat{O} \rangle_{\theta_k=\pi/2} - \langle \hat{O} \rangle_{\theta_k=-\pi/2}, \quad (29)$$

$$E = \langle \hat{O} \rangle_{\theta_k=\pi/4} - \langle \hat{O} \rangle_{\theta_k=-\pi/4}, \quad (30)$$

and the combinations:

$$a = \frac{1}{4} (2E - \sqrt{2}D), b = \frac{1}{4}(A - C), c = \frac{1}{2}D, d = \frac{1}{2}B,$$

one can show that:

$$\begin{aligned} \langle \hat{O} \rangle_{\theta_k} &= a \sin(2\theta_k) + b \cos(2\theta_k) + c \sin(\theta_k) + d \cos(\theta_k) \\ &= P \sin(2\theta_k + \phi) + Q \sin(\theta_k + \psi) \end{aligned} \quad (31)$$

where $P = \sqrt{a^2 + b^2}$, $\phi = \arctan_2(a, b)$, $Q = \sqrt{c^2 + d^2}$, $\psi = \arctan_2(c, d)$. Unlike the original Rotosolve algorithm, Supplementary Eq. (31) is not a single sinusoid but a sum of sinusoids with doubled frequencies and different amplitudes and phases. It can no longer be minimized analytically. However, the spirit of the Rotosolve algorithm remains, and one proceeds as follows:

1. Perform the 6 measurements required to determine A, B, C, D, E .
2. Perform a classical scalar minimization on the function $\langle \hat{O} \rangle_{\theta_k}$ (whose coefficients have now been determined). This can be done very quickly on the fly, or precomputed and interpolated for arbitrary P, Q, ϕ, ψ .
3. Set the variable θ_k to its corresponding minimum.

The resulting algorithm suffers no significant loss of performance over Rotosolve, and only requires 3 more measurements per iteration.

II. SUPPLEMENTARY DISCUSSION

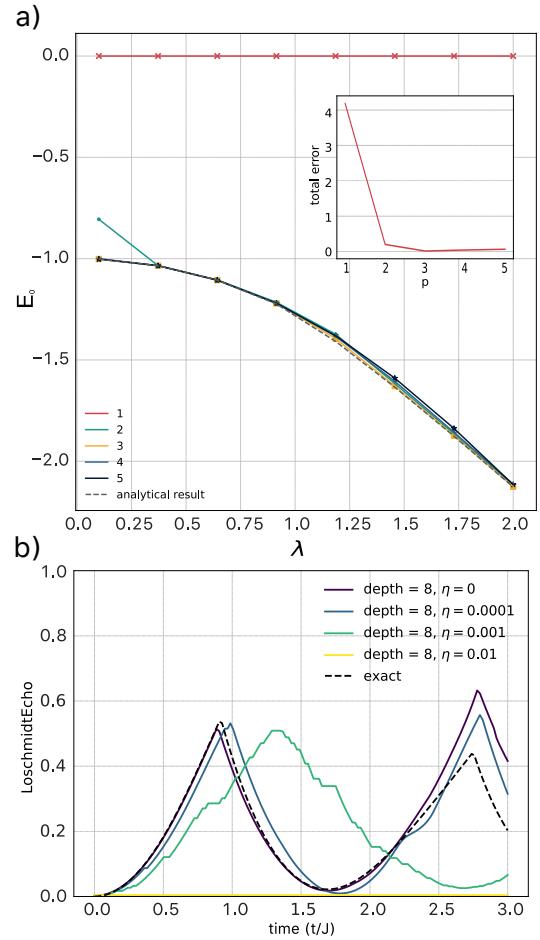
In the following, we provide some further technical details of our results

Finite Gate Fidelity and Restricting Circuit Depth: Finite gate-fidelity implies a maximum reliable depth of quantum circuit. As evident from Figs. 2 and 3, passing from the simplest task of representing a quantum state through the more complicated tasks of optimising and time-evolving states, the required circuits are broader and of greater depth. Time-evolution is therefore much more susceptible to gate errors than simply representing a state. Whilst it might be possible to represent and measure properties of a rather high bond-order state, time-evolution is inevitably restricted to lower bond order.

Supplementary Fig. 7a shows how the attainable groundstate fidelity of the transverse-field Ising model $\mathcal{H} = \sum_i \sigma_i^z \sigma_{i+1}^z + \lambda \sigma_i^x$ changes with λ for fixed noise strength $\eta = 1 \times 10^{-3}$. The same gate fidelity has a much more severe effect upon simulations of dynamics. Supplementary Fig. 7b shows how a simulation at $D = 2$ is degraded by this error rate.

This formalism allows us to make a variety of trade-offs between noise and performance, depending upon the resources available on the quantum computer. One can increase the expressibility of the circuit by increasing the depth of the ansatz. It is also possible to modify the variational space by using a shallow ansatz at a larger bond order, which requires a greater qubit count. It remains to be seen how each of these options will perform on a near term chip.

Construction of the Poincaré Map: To produce a Poincaré map, the parameters that define the quantum



Supplementary Figure 7. **Effect of finite gate fidelity upon optimisation and evolution:** a) Effect of noise upon time evolution, η is depolarising probability. Increasing noise deteriorates but does not destroy the appearance of the dynamical phase transition peaks. b) Optimization, for depolarising probability $\eta = 1 \times 10^{-3}$, and varying circuit depth (legend). Inset shows total deviation from exact curve as a function of depth (p). Note the tradeoff between circuit depth and sensitivity to noise. For computational expediency, expectation values are calculated exactly from noisy circuits.

state are recorded as they evolve over time. When a chosen parameter crosses a particular value in the positive direction the values of the other parameters are plotted. The values of these crossings indicated in Fig. 5c of the main text are obtained by polynomial interpolation of the values at discretised time intervals obtained from the quantum circuit. This proceeds as follows: Around the approximate crossing time, polynomial function interpolation is used to estimate the evolution of all parameters. A root finding algorithm is used on the interpolated polynomial function of the chosen parameter to get an estimated time of crossing. Finally this time is input into the other interpolated functions to get estimates of all the parameters at the crossing time.

-
- [1] Perez-Garcia, D., Verstraete, F., Wolf, M. M. & Cirac, J. I. Matrix product state representations. *Quantum Info. Comput.* **7**, 401–430 (2007).
- [2] Schollwöck, U. The density-matrix renormalization group in the age of matrix product states. *Ann. Phys. (NY)* **326**, 96 – 192 (2011). URL <http://www.sciencedirect.com/science/article/pii/S0003491610001752>. January 2011 Special Issue.
- [3] Orús, R. A practical introduction to tensor networks: Matrix product states and projected entangled pair states. *Ann. Phys. (NY)* **349**, 117 – 158 (2014). URL <http://www.sciencedirect.com/science/article/pii/S0003491614001596>.
- [4] Garcia-Escartin, J. C. & Chamorro-Posada, P. swap test and hong-ou-mandel effect are equivalent. *Phys. Rev. A* **87**, 052330 (2013). URL <https://link.aps.org/doi/10.1103/PhysRevA.87.052330>.
- [5] Pollmann, F. & Turner, A. M. Detection of symmetry-protected topological phases in one dimension. *Phys. Rev. B* **86**, 125441 (2012). URL <https://link.aps.org/doi/10.1103/PhysRevB.86.125441>.
- [6] Ostaszewski, M., Grant, E. & Benedetti, M. Structure optimization for parameterized quantum circuits. *Quantum* **5**, 391 (2021).
- [7] Vidal, J. G. & Theis, D. O. Calculus on parameterized quantum circuits. *arXiv preprint arXiv:1812.06323* (2018).
- [8] Parrish, R. M., Iosue, J. T., Ozaeta, A. & McMahon, P. L. A jacobi diagonalization and anderson acceleration algorithm for variational quantum algorithm parameter optimization. *arXiv preprint arXiv:1904.03206* (2019).
- [9] Nakanishi, K. M., Fujii, K. & Todo, S. Sequential minimal optimization for quantum-classical hybrid algorithms. *Physical Review Research* **2**, 043158 (2020).