

A Quantum Inspired Approach to Exploit Turbulence Structures

Nikita Gourianov^{1,*}, Michael Lubasch², Sergey Dolgov³, Quincy Y. van den Berg¹, Hessam Babaei⁴, Peyman Givi⁴, Martin Kiffner^{5,1}, and Dieter Jaksch^{1,5,6†}

¹Clarendon Laboratory, University of Oxford, Oxford, UK

²Cambridge Quantum Computing Limited, London, UK

³Department of Mathematical Sciences, University of Bath, Bath, UK

⁴Department of Mechanical Engineering and Materials Science, University of Pittsburgh, Pittsburgh, PA, USA

⁵Centre for Quantum Technologies, National University of Singapore, Singapore and

⁶Insitut für Laserphysik, Universität Hamburg, Hamburg, Germany

Understanding turbulence is the key to our comprehension of many natural and technological flow processes. At the heart of this phenomenon lies its intricate multi-scale nature, describing the coupling between different-sized eddies in space and time. Here we analyse the structure of turbulent flows by quantifying correlations between different length scales using methods inspired from quantum many-body physics. We present results for interscale correlations of two paradigmatic flow examples, and use these insights along with tensor network theory to design a structure-resolving algorithm for simulating turbulent flows. With this algorithm, we find that the incompressible Navier-Stokes equations can be accurately solved even when reducing the number of parameters required to represent the velocity field by more than one order of magnitude compared to direct numerical simulation. Our quantum-inspired approach provides a pathway towards conducting computational fluid dynamics on quantum computers.

Turbulence can phenomenologically be described by mutually interacting eddies stretching across an extremely broad range of length and time scales [1, 2]. These spatial scales range from the largest size of the energy containing eddies (known as the integral scale, ℓ), to the smallest ones, known as the Kolmogorov microscale, η . The separation between these scales is $\ell/\eta \sim \text{Re}^{3/4}$ where Re is the Reynolds number. A salient feature of turbulent flows is the scale-locality of the turbulent energy cascade [3–6], in the sense that eddies at a given length scale predominantly interact with other eddies of similar scale.

Since the pioneering work of Orszag and Patterson [7], direct numerical simulation (DNS) of the Navier-Stokes equations has been widely regarded as the computational method with the highest fidelity in capturing the dynamics of turbulent flows. Virtually all classical DNS methods such as spectral polynomial/Fourier, finite volume and finite element are *scale-resolving*, where increasing the number of variables (e.g. grid points M) resolves finer and finer scales. However, the wide separation of turbulent flow scales limits the range of Reynolds numbers that can be computationally considered. Straightforward estimates indicate that simulation of an incompressible flow inside a three-dimensional volume $\sim (10\ell)^3$ with $\text{Re} \sim 10^5$ would require *decades* of CPU time on a 1 teraflop computer.

For mitigating this huge numerical complexity, the importance of exploiting so-called coherent structures of turbulence [8] has long been recognised. This eventually led to the rise of *structure-resolving* methodologies (e.g., proper orthogonal decomposition) [9] that extract and exploit correlated structures of the solution. They represent the flow field down to the Kolmogorov microscale

through a superposition of modes, but with their number being much smaller than the total number of grid points in DNS. Up to now, the majority of the developed techniques have been used for *diagnostic* purposes. Using reduced order models for *predictive* purposes is hampered by difficulties in identifying suitable modes and remains an active area of research [10].

A similar challenge has been successfully tackled in a completely different area of physics. Quantum many-body systems are described by elements of a vector space whose dimension grows *exponentially* with the number of particles [11, 12]. This makes direct simulations quickly impossible with increasing system size. Tensor network methods [13, 14] made a revolutionary advance in simulating quantum systems with local interactions by removing unrealised long-distance correlations, thus enabling the simulation of physical systems that are otherwise intractable [15, 16]. The correlations of interest for quantum systems are known as quantum entanglement [17], and weakly entangled systems or those where quantum correlations are concentrated at the boundaries between different parts of the system (i.e., they are structured to follow a so-called area law [18]) can be efficiently simulated with tensor network methods.

Here we adapt these successful strategies for treating quantum many-body systems towards exploiting the scale-locality of turbulence. This quantum inspired approach allows us to develop structure-resolving methods for both diagnostic and predictive purposes. We first introduce tools from tensor network theory to analyse different length scales of flow in a manner different from both the traditional wavenumber approach [19–22] and more recent investigations in real space [5]. Then, we apply these diagnostic tools to study DNS solutions of two

paradigmatic flow configurations: a 2-D temporally developing jet (TDJ) [23] and the 3-D Taylor-Green vortex (TGV) [24]. Our study reveals that the scale-locality of turbulence restricts the amount of correlation present between different length scales. This motivates us to encode turbulence in a simple tensor network called matrix product state (MPS). The connectivity of the MPS network is well adapted to describing scale-local flows and hence exploiting the structures of turbulence. We design an algorithm for simulating the incompressible Navier-Stokes equations (INSE) in the compressed MPS format. This algorithm remains accurate even when reducing the number of variables parametrising the solution (NVPS) by more than one order of magnitude. The conceptual similarity between the tensor network algorithm presented here and those used in quantum physics opens the possibility of conducting computational fluid dynamics on a quantum computer.

RESULTS

Quantifying interscale correlations. Throughout this work we follow the standard approach in computational fluid dynamics and discretise the computational domain. Each spatial dimension is discretised by 2^N grid points, where N is a positive integer. In this way, the velocity field

$$\mathbf{V}(t, \mathbf{r}_q) = \sum_{i=1}^K u_i(t, \mathbf{r}_q) \hat{\mathbf{e}}_i \quad (1)$$

and its Cartesian components u_i are discrete functions of the grid points \mathbf{r}_q , where K is the spatial dimension and $\hat{\mathbf{e}}_i$ are Cartesian unit vectors. We measure the interscale correlations by using the Schmidt (singular value) decomposition to systematically divide the computational grid into sub-grids, as illustrated in Fig. 1a for $K = 2$. We decompose (for details, see the *Schmidt decomposition* section in *Methods*) each component u_i on this $2^N \times 2^N$ grid into functions R and f on a coarse and a fine subgrid, respectively,

$$u_i(t, \mathbf{r}_q) = \sum_{\alpha=1}^{d(n)} \lambda_{\alpha}(t) R_{\alpha}(t, \mathbf{X}_k) f_{\alpha}(t, \mathbf{x}_l), \quad \mathbf{r}_q = \mathbf{X}_k + \mathbf{x}_l. \quad (2)$$

Positions \mathbf{X}_k correspond to a quadratic grid with $2^n \times 2^n$ points (coarse grid), and \mathbf{x}_l correspond to a fine sub-grid with $2^{N-n} \times 2^{N-n}$ grid points. The functions R_{α} and f_{α} obey the orthonormality condition

$$\sum_k R_{\alpha}(t, \mathbf{X}_k) R_{\beta}(t, \mathbf{X}_k) = \sum_l f_{\alpha}(t, \mathbf{x}_l) f_{\beta}(t, \mathbf{x}_l) = \delta_{\alpha\beta}, \quad (3)$$

where $\delta_{\alpha\beta}$ is the Kronecker delta. The parameter $n = 1, \dots, N-1$ labels the possible bipartitions of the square lattice in coarse and fine grids [see Fig. 1a for $N = 10$ and $n = 2$]. The Schmidt number $d(n)$ denotes the number of retained terms in the summation in Eq. (2), and each product $R_{\alpha} f_{\alpha}$ is weighted by a Schmidt coefficient $\lambda_{\alpha} \geq 0$. These coefficients appear in descending order $\lambda_1 \geq \lambda_2 \dots \geq \lambda_{d(n)}$, thus varying $d(n)$ will only add or remove the least important of the orthonormal basis functions. Here we take $d(n)$ as a quantitative measure for the interscale correlations of turbulent flows at a given bipartition n of the lattice: $d(n) = 1$ corresponds to an uncorrelated product state, and with increasing $d(n)$ the flow becomes more strongly correlated between the coarse and the fine grid. Note that while the $d(n) = 1$ product state exhibits no interscale correlations, it is still highly correlated in space because the fine grid dependence is repeated.

Truncating the Schmidt decomposition in Eq. (2) approximates u_i in an orthonormal time-dependent basis that evolves with the fluid flow to optimally capture spatially correlated structures. This is in contrast to classical scientific computing techniques (implemented through e.g. finite difference or spectral methods) where the bases are structure-agnostic, i.e., they are chosen a priori and disregard any structure in the solution.

We first apply the decomposition in Eq. (2) to DNS solutions of the INSE [see Eq. (7)] for the TDJ shown in the top row of Fig. 2a. The TDJ comprises a central jet flow along the x -direction, and Kelvin-Helmholtz instabilities in the boundary layer of the jet eventually cause it to collapse (see Eqs. (9) through (15) in *Methods* for initial flow conditions). We decompose each velocity component according to Eq. (2), which is an exact representation if $d(n) = \Gamma^{2-D}(n)$ with (for details, see Supplementary Sec. 2)

$$\Gamma^{2-D}(n) = \min(4^n, 4^{N-n}). \quad (4)$$

Fig. 1b shows the Schmidt numbers $d_{99}(n, t)$ such that Eq. (2) represents the DNS solutions for the velocity fields with 99% accuracy in the L2 norm (more details on the Schmidt coefficients can be found in Supplementary Sec. 1). We find that $d_{99}(n, t)$ are well below their maximal values $\Gamma^{2-D}(n)$ for $n > 1$. More specifically, we define $\chi_{99} = \max d_{99}(n, t)$ as the maximal value of d_{99} for all n and time steps. We obtain $\chi_{99} = 25$, and the interscale correlations captured by Eq. (2) with $d(n) = \min(\Gamma^{2-D}(n), 25)$ are shown by the blue-shaded area \mathcal{M} in Fig. 1b. $d_{99}(n, t)$ is entirely contained within this blue area. Note that the Schmidt numbers are shown on a logarithmic scale in Fig. 1b, and thus the area \mathcal{M} is much smaller than the area \mathcal{D} corresponding to DNS.

We obtain qualitatively similar results for the DNS solutions to the TGV in 3-D, where vortex stretching causes a single, large, ordered fluctuation to collapse into a turbulent flurry of small scale structures (see top row in

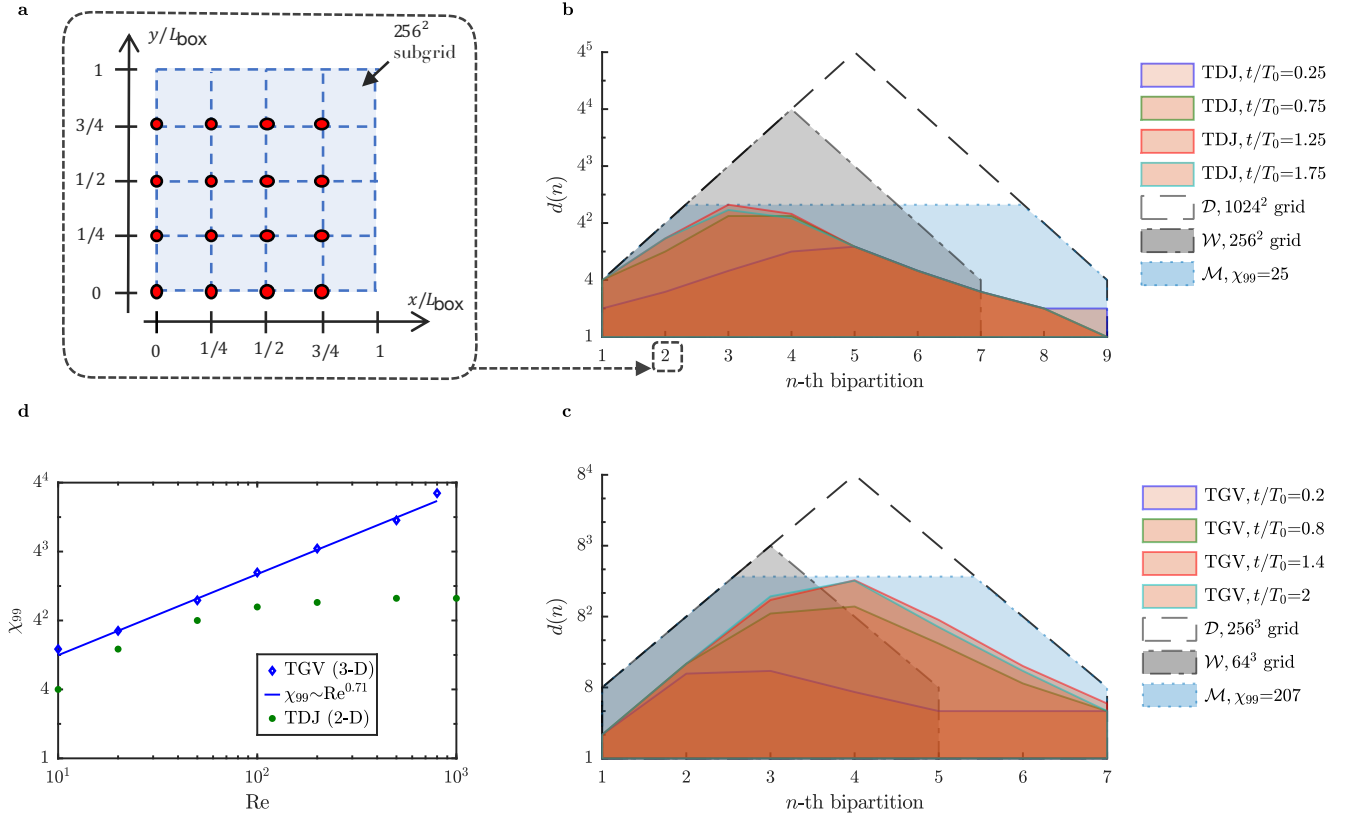


FIG. 1: **Interscale correlations of turbulent fluid flows.** (a) and (b) correspond to a square with edge length L_{box} on a $2^{10} \times 2^{10}$ grid. (a) illustrates the subgrid structure when decomposing a function u_i according to Eq. (2) for $n = 2$. Red dots are the $2^2 \times 2^2$ grid points \mathbf{X}_k of the coarse grid. Each blue square attached to the \mathbf{X}_k indicates the quadratic subgrid with the $2^8 \times 2^8$ grid points \mathbf{x}_k of the fine grid. (b) shows the Schmidt numbers $d_{99}(n, t)$ on a logarithmic scale such that the decomposition in Eq. (2) results in a 99% accurate representation of DNS solutions to the INSE at four different times [see Fig. 2]. The domain \mathcal{D} indicated by the black dashed line corresponds to DNS. The grey shaded area \mathcal{W} is for solutions on a $2^8 \times 2^8$ grid. The blue shaded area \mathcal{M} is for $d(n) \leq 25$ in Eq. (2). (c) Same as in (b) but for the 3-D simulations shown in Fig. 3. In (b) and (c), T_0 is the characteristic time scale on which the quickest particles in the initial flow fields can traverse the box (see *Set-up of numerical experiments* section in *Methods*). (d) Scaling of $\chi_{99} = \max d_{99}(n, t)$ with the Reynolds number for the 2-D and 3-D systems in (b) and (c), respectively.

Fig. 3a for visualisation, and (16) in *Methods* for initial flow conditions). In three spatial dimensions, the representation in Eq. (2) is exact if $d(n)$ equals (for details, see Supplementary Sec. 2)

$$\Gamma^{3-D}(n) = \min(8^n, 8^{N-n}). \quad (5)$$

The Schmidt numbers $d_{99}(n, t)$ resulting in a 99% accurate representation of the DNS solutions are shown in Fig. 1c. We find $\chi_{99} = 207$ such that all values of d_{99} in Fig. 1c are contained within the blue-shaded area \mathcal{M} corresponding to $d(n) = \min(\Gamma^{3-D}(n), 207)$ in Eq. (2). Since χ_{99} is much smaller than the upper vertex of the area \mathcal{D} at $\Gamma^{3-D}(4) = 2^{12}$, the interscale correlations of DNS solutions are far from being saturated (more details on the Schmidt coefficients can be found in Supplementary Sec. 1).

Next we investigate how the maximal Schmidt number χ_{99} scales with the Reynolds number Re , and the results are shown in Fig. 1d. We find that χ_{99} saturates in the 2-D case for $Re \gtrsim 200$. This suggests that interscale correlations of 2-D flows are bounded in analogy to quantum correlations in gapped 1-D quantum systems with local interactions [18]. In the 3-D case, χ_{99} increases according to a power law. The NVPS for $d(n) = \min(\Gamma^{3-D}(n), \chi_{99})$ scales as $\sim \chi_{99}^2 \log M$ (see Supplementary Sec. 2). Kolmogorov's theory [3] states that the number of grid points $M = 8^N$ must scale with the Reynolds number like $M \sim (\ell/\eta)^3 \sim Re^{9/4}$ to resolve all spatial scales. This implies that the NVPS of \mathcal{M} only scales as $\sim Re^{1.42} \log Re$, which is a substantially slower increase with Re compared to the NVPS of DNS, $\sim M \sim Re^{9/4}$.

Tensor network algorithm. The previous results demonstrate that it is beneficial to find a representation of flow fields where limiting the amount of interscale correlations directly translates into a reduction of the NVPS. This can be achieved by expressing each velocity component in a compressed tensor network format known as matrix product state (MPS) or tensor train decomposition [13, 14, 25, 26]. Our MPS encoding of function values is chosen such that it is consistent with the decomposition in Eq. (2) (see Supplementary Sec. 2). It comprises products of N matrices A^{ω_n} with dimension $d(n-1) \times d(n)$, where 2^N is the number of grid points in each spatial direction, $n = 1, \dots, N$ and $d(0) = d(N) = 1$ [27, 28]. The matrix A^{ω_n} is associated with a length scale $L_{\text{box}}/2^n$, and its dimension $d(n)$ controls the maximum amount of correlations allowed between neighbouring scales. The nearest neighbour correlations are mediated directly by each matrix product, while correlations between further distant length scales can only be captured indirectly by traversing several matrix products. These properties make MPS well suited for the description of scale-local turbulent flows where correlations between vastly different length scales are expected to be small.

Here we consider MPSs of bond dimension χ where we set $d(n) = \min(\Gamma^{2\text{-D}}(n), \chi)$ in 2-D and $d(n) = \min(\Gamma^{3\text{-D}}(n), \chi)$ in 3-D. The bond dimension χ controls the level of compression in the MPS format. For example, the interscale correlations captured by an MPS with bond dimension $\chi = 25$ ($\chi = 207$) are represented by the blue-shaded area in Fig. 1b (Fig. 1c). If χ is kept constant as N increases, the number of MPS parameters scales logarithmically with the total number of grid points, resulting in an exponential reduction of the NVPS compared to DNS. However, we emphasise that this reduction does not truncate the range of length scales covered by the MPS ansatz, it only limits the amount of interscale correlations.

In order to fully utilise this dimensionality reduction for numerical simulations on large grids, we devise an algorithm for solving the INSE without leaving the compressed MPS manifold \mathcal{M} (see the *Matrix product state algorithm* section in *Methods*). We use a second-order Runge-Kutta time-stepping scheme and discretise spatial derivatives in the same way as the DNS solver, which utilises an 8th-order finite difference stencil.

Validation of the tensor network algorithm. We now investigate how well the dynamics of turbulent flow are captured inside the MPS manifold \mathcal{M} by comparing our algorithm against DNS for different compressions. Reducing the bond dimension χ reduces the NVPS. The analogue of reducing the bond dimension in traditional DNS is to perform underresolved DNS (URDNS) where the simulation is carried out on a coarse grid not covering all relevant length scales. URDNS can be considered as the most basic form of large eddy simulations [29–

31], where no explicit models are employed to account for the disregarded subgrid scales (see the *Direct numerical simulation algorithm* section in *Methods*). For a fair comparison between MPS and URDNS, we choose for every bond dimension χ a corresponding URDNS grid such that the NVPS is approximately equal for both methods.

The results for the TDJ and for the different solvers are shown in Fig. 2. The top row in Fig. 2a corresponds to DNS and illustrates how the background perturbations in the shear-layer are amplified ($t/T_0 = 0.25$) until the layer rolls up into vortices which in turn pair-up and merge into progressively larger vortices ($t/T_0 = 0.75, 1.25$) until $t/T_0 = 1.75$, when pairing is terminated. The stress exerted upon the mean flow by turbulent fluctuations is given by the Reynolds stress tensor, one of whose components is plotted in Fig. 2b. These results are in accord with the Boussinesq approximation which indicates the Reynolds stresses are of the opposite sign of the mean streamwise velocity gradients along the cross-stream direction. The exception is when the vortex pairing is terminated, and the growth of the shear-layer is temporarily paused. Correctly resolving τ_{12} is important for the physical validity of the simulation.

We now evaluate the accuracy of the MPS and URDNS simulations. Rows 2-4 in Fig. 2 show MPS results for $\chi = 33, 74$ and 118 , corresponding to compression ratios of approximately $1 : 64$, $1 : 16$ and $1 : 8$ compared to DNS, respectively. These results show that the large-scale dynamics of the jet are correctly captured by all the MPS simulations, with $\chi = 74$ and 118 practically indistinguishable from DNS. The bottom three rows in Fig. 2 show the URDNS results for different grid sizes. We find that the 362^2 grid (corresponding to $\chi = 118$) is only accurate until $t/T_0 \approx 1.2$ while the lower-resolution URDNS already fails at $t/T_0 \approx 0.75$. This is observed in both the instantaneous vorticity dynamics results and the Reynolds-stresses. The MPS algorithm achieves a much higher accuracy than URDNS for the same NVPS (see Table 1 for quantitative results). This result can also be understood in terms of the interscale correlations shown in Fig. 1b, where the domain corresponding to URDNS on the 256^2 grid is shown by the grey-shaded area \mathcal{W} . A significant amount of correlations present in the DNS solutions are outside of \mathcal{W} , which is consistent with our finding that URDNS cannot accurately represent the DNS solutions.

Next we discuss the corresponding results for the dynamics of the TGV. The top row in Fig. 3a corresponds to DNS and illustrates how the original vortex collapses ($t/T_0 = 0.2$) into turbulent worm-like structures ($t/T_0 = 0.8$) which become progressively more turbulent ($t/T_0 = 1.4$) until viscosity eventually dissipates these vortical structures ($t/T_0 = 2$). Rows 2-4 and 5-7 in Fig. 3a correspond to the results of our MPS algorithm and to URDNS, respectively. The bond dimension and grid sizes have been chosen such that the compression

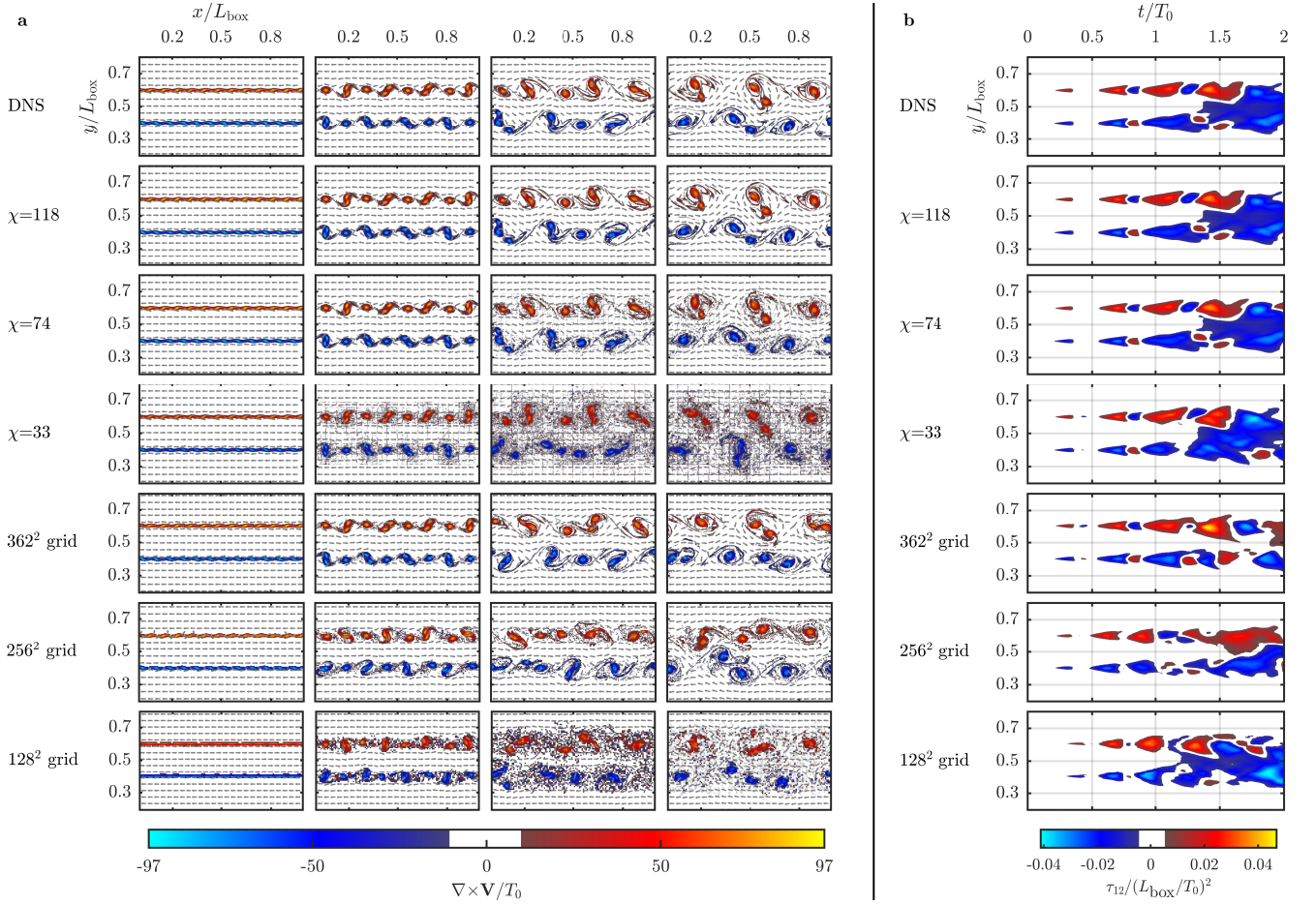


FIG. 2: **2-D Temporally developing jet.** Dynamical simulation of the INSE in 2-D for a planar jet streaming along x with $\text{Re} = 1000$, as defined in the *Set-up of numerical experiments* section in *Methods*. (a) shows snapshots of the vorticity and velocity fields taken at $t/T_0 = 0.25, 0.75, 1.25, 1.75$ (left to right). Red corresponds to positive vorticity (counter-clockwise flow) and blue to negative (clockwise). Top row corresponds to DNS results on a quadratic $2^{10} \times 2^{10}$ grid (cf. Fig. 1a). Rows 2-4 are MPS results with different maximal bond dimensions χ . Bottom three rows are for URDNS on quadratic grids as indicated. (b) Reynolds stress τ_{12} [see Eq. (14)] between the streamwise and cross-stream directions as a function of time and y coordinate. Red (blue) corresponds to positive (negative) stress.

ratios compared to DNS are approximately 1 : 25 (rows 2 and 5), 1 : 49 (rows 3 and 6) and 1 : 78 (rows 4 and 7). While MPS produces a solution comparable to DNS for a compression ratio of 1 : 49 ($\chi = 128$), the corresponding URDNS results clearly deviate from DNS. Discrepancies between URDNS and DNS are even visible for the largest URDNS grid (compression 1 : 25).

A more quantitative analysis of the performance of MPS vs. URDNS is shown in Fig. 3b. In the non-DNS simulations, a portion of the energy is erroneously lost to numerical diffusion. The amount of numerical diffusion can be measured by comparing the physical global dissipation (enstrophy) to the global kinetic energy dissipation [see Eqs. (17) and (18)]. The MPS predictions at $\chi \geq 128$ are consistent with both the DNS results here

and in previous work [24]. We find that the MPS simulations with $\chi = 128$ and 192 dissipate the energy more accurately than any of the URDNS results, especially for $t/T_0 \geq 1.4$, see Table 1. As in the 2-D case, this outcome is in line with the interscale correlations shown in Fig. 1c, where the domain corresponding to URDNS on the 64^3 grid is shown by the grey-shaded area \mathcal{W} . The bipartitions at $n = 5, 6, 7$ are associated with comparatively large Schmidt numbers for $t/T_0 \geq 0.8$, and hence these interscale correlations cannot be captured by URDNS.

DISCUSSION

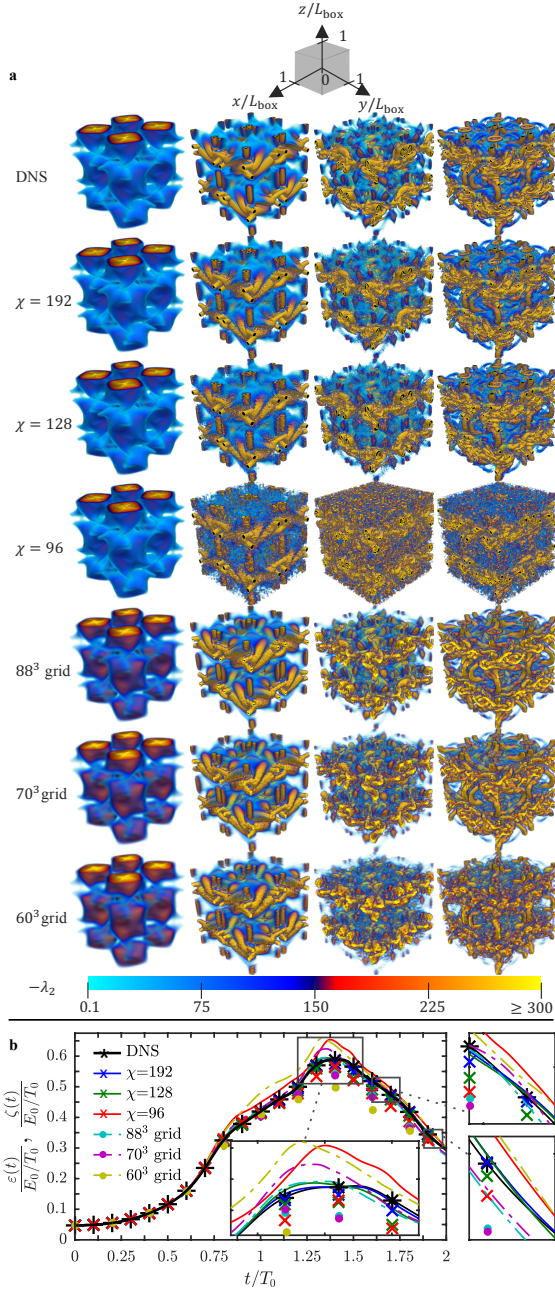


FIG. 3: **3-D Taylor-Green vortex.** Dynamical simulation of the INSE in 3-D for the Taylor-Green vortex and $\text{Re} = 800$, as defined in the *Set-up of numerical experiments* section in *Methods*. (a) Vortical structures rendered using the standard λ_2 method [51] are shown at times $t/T_0 = 0.2, 0.8, 1.4, 2$ (left to right). Top row is for DNS on a $2^8 \times 2^8 \times 2^8$ grid. Rows 2-4 are for MPS simulations with different χ , and bottom three rows are for URDNS on cubic grids as indicated. (b) shows the enstrophy $\zeta(t)$ (asterisks/crosses/circles) and the energy dissipation $\epsilon(t)$ (lines) as a function of time, with E_0 being the total energy at $t = 0$.

The structure-resolving properties of MPSs can lead to a reduced computational cost. The computational complexity of our MPS algorithm is $\sim \chi^4 \log M$, as explained in Supplementary Sec. 4. Resolving down to the Kolmogorov microscale η requires $M \sim (\ell/\eta)^K \sim \text{Re}^{3K/4}$ grid points. Assuming $\chi \sim \text{Re}^\gamma$, the overall scaling of the MPS algorithm becomes $\sim \text{Re}^{4\gamma} \log \text{Re}$. Comparing this to the scaling of DNS, which is $\sim M \log M \sim \text{Re}^{3K/4} \log \text{Re}$ (see the *Direct numerical simulation algorithm* section in *Methods*), we see that the MPS algorithm outperforms DNS when $\gamma < 3K/16$. For our 2-D example, Fig. 1d suggests $\gamma \approx 0$ which leads to an exponential speedup of the MPS algorithm over DNS for sufficiently large Re . It would be interesting to investigate whether this saturation of the Schmidt number with Re is a unique case for just the TDJ flow, or if it is a more general property of 2-D turbulence. If this is indeed general, it would have significant practical consequences for e.g. the simulation of atmospheric flows. For the TGV flow $\gamma \approx 0.71$, which is larger than $3K/16$. However, we note that numerical methods for manipulating high-dimensional tensors are an active field of research [32], potentially enabling an improved scaling of MPS algorithms with χ in the future. We also remark that MPS can be exponentially faster than DNS at simulating shock waves, as illustrated in Supplementary Sec. 3 through analytical studies of the 1-D Burgers' equation.

Our initial choice of MPS networks was motivated by the scale-locality of turbulent flows. However, MPS might be numerically inefficient when correlations between distant scales are relevant. One then needs a very large bond dimension χ to maintain an accurate description of the flow. Other tensor network geometries like Tree Tensor Networks (TTNs) or multi-scale entanglement renormalisation ansatz (MERA) and its derivatives [33, 34] might then be worthwhile considering. These network geometries (see Supplementary Sec. 6) have direct bonds between further distant length scales and might require smaller bond dimensions. However, TTNs maintain numerical efficiency by abandoning direct bonds between neighbouring length scales which are important for exploiting scale-locality. MERA is numerically challenging because of loops in the network.

The utility of tensor networks in fluid dynamics goes beyond the INSE. Future avenues of investigation for MPS include compressible flows, in which the Mach number is an important parameter, and transport of scalar quantities under both passive and chemically reactive conditions where the effects of Prandtl, Peclet and Damkohler numbers [35] must be taken into account. It would be interesting to examine how these parameters affect the fidelity of low χ MPS simulations. Furthermore, as tensor network methods are naturally suited

to tackle high-dimensional problems, their applicability to the transported probability density function (PDF) of turbulent reactive flows [36] should be considered. In these flows, in addition to temporal and spatial variations, the PDF is a function of the three-dimensional velocity field and all of the pertinent scalar variables (energy, pressure and species mass fractions) [37]. With just 10 species (a very simple chemical kinetics model), the unsteady PDF must be resolved in a 17-dimensional space. High-fidelity modelling and simulation of such complex flows can potentially be enabled through a well-chosen tensor network ansatz.

The close connection of our tensor network-based approach to quantum physics points towards the prospect of solving the Navier-Stokes equations on a quantum computer. Recently, several algorithms for solving nonlinear partial differential equations on quantum computers have been proposed [38–40]. In particular, the work in [38] introduces tensor networks as a programming paradigm for quantum computers, which makes our approach especially well-suited for quantum hardware implementations (see Supplementary Sec. 5 for details). Replacing classical floating point operations by quantum gates reduces the scaling with bond dimension to $\sim \chi^2$ (see Supplementary Sec. 5). In addition, potentially exponential speedups are possible by choosing an optimised quantum network that goes beyond the MPS ansatz for encoding the solution [38, 41, 42]. In this way, our work holds the promise of enabling large-scale computational fluid dynamics calculations that are well beyond the scope of current approaches.

METHODS

Schmidt decomposition. We consider a 1-D system and scale all lengths with its spatial dimension L_{box} . We discretise the spatial domain $[0, 1]$ of the velocity u with N bits into 2^N grid points $r_q = q/2^N$ with $q = 0, 1, \dots, 2^N - 1$. Next, we introduce $n = 1, \dots, N-1$ bipartitions of this grid into coarse and fine sub-grids. For a given n , the coarse sub-grid comprises the points $X_k = k/2^n$ with $k = 0, \dots, 2^n - 1$. The spacing between neighbouring points is thus 2^{-n} and this defines the coarse length scale. To each coarse grid point X_k is attached a fine sub-grid with points $x_l = l/2^N$ with $l = 0, 1, \dots, 2^{(N-n)} - 1$, and adjacent points are separated by the fine length scale 2^{-N} . In this way, any point r_q of the 1-D grid can be written as $r_q = X_k + x_l$. Finally, we arrange the function values $u(r_q) = u(X_k + x_l)$ into a $2^n \times 2^{N-n}$ matrix where the rows and columns correspond to increments along the coarse and fine grids, respectively. Performing a singular value decomposition (SVD) on this matrix [12, 13] gives the desired Schmidt

decomposition of $u(r_q)$ at bipartition n ,

$$u(r_q) = \sum_{\alpha=1}^{d(n)} \lambda_{\alpha} R_{\alpha}(X_k) f_{\alpha}(x_l). \quad (6)$$

This is the 1-D result corresponding to Eq. (2). For a full SVD the Schmidt number takes its maximal value $d(n) = \Gamma^{1-D}(n)$, where $\Gamma^{1-D}(n) = \min(2^n, 2^{N-n})$. If instead a truncated SVD is performed by keeping only the $d(n) = \chi$ largest singular values, the error in the L2 norm due to this approximation is $\sqrt{\sum_{\alpha=\chi+1}^{\Gamma^{1-D}(n)} \lambda_{\alpha}^2}$.

This procedure can be straightforwardly generalised by replacing bits with quaternaries (2-D) or octals (3-D), i.e., by replacing 1-D line segments with squares (2-D) or cubes (3-D). The maximal Schmidt numbers are then given by Eq. (4) in 2-D and Eq. (5) in 3-D.

Matrix product state algorithm. The INSE are a coupled set of partial differential equations for the velocity field \mathbf{V} and the pressure p ,

$$\begin{aligned} \nabla \cdot \mathbf{V} &= 0 \\ \frac{\partial \mathbf{V}}{\partial t} + (\mathbf{V} \cdot \nabla) \mathbf{V} &= -\nabla p + \nu \nabla^2 \mathbf{V}, \end{aligned} \quad (7)$$

where ν is the kinematic viscosity and ∇ is the nabla operator. After discretising the computational domain as described in the *Introduction*, we solve Eq. (7) in time via a second-order Runge-Kutta method by a variational scheme. Furthermore, we use the penalty method [43, 44] to satisfy the incompressibility condition $\nabla \cdot \mathbf{V} = 0$.

We illustrate the principle of our method by considering a simple Euler time step. To advance \mathbf{V} from time t_s to $t_s + \Delta t$, we minimise the cost function

$$\begin{aligned} \Theta(\mathbf{V}^*) &= \mu \|\bar{\nabla} \cdot \mathbf{V}^*\|_2^2 + \\ &\left\| \frac{\mathbf{V}^* - \mathbf{V}}{\Delta t} + (\mathbf{V} \cdot \bar{\nabla}) \mathbf{V} - \nu \bar{\nabla}^2 \mathbf{V} \right\|_2^2, \end{aligned} \quad (8)$$

where $\|\cdot\|_2$ is the L2 norm, $\bar{\nabla}$ is the nabla operator in finite difference form, \mathbf{V}^* is the trial solution at time $t_s + \Delta t$ and \mathbf{V} denotes the solution at the previous time step t_s . The term $\mu \|\bar{\nabla} \cdot \mathbf{V}^*\|_2^2$ in Eq. (8) enforces $\nabla \cdot \mathbf{V} = 0$ for sufficiently large values of the penalty coefficient μ . Note that the penalty method for enforcing the incompressibility condition removes the pressure p from Eq. (8). It can be calculated from the velocity fields via its Poisson equation [45].

We represent the flow field \mathbf{V} in terms of the MPS ansatz at all time steps t_s , and all operations on \mathbf{V} like differentiation are realised via standard matrix product operators acting on the MPS [27, p. 591] and [28, p. 22]. In this way, the entire computation is carried out in the MPS manifold $\mathcal{M} \in \mathcal{D}$. A derivation of our minimisation scheme is provided in Supplementary Sec. 4.

Direct numerical simulation algorithm Our DNS scheme is based upon a second-order Runge Kutta temporal discretisation combined with an eighth-order central finite difference discretisation of the spatial derivatives [46] on a Cartesian grid. The incompressibility condition is enforced through the projection method of Chorin [47] at every substep of each full Runge-Kutta time-step.

The computational complexity of the DNS scheme is $\sim M \log M$. This is because it is dominated by the projection step, which is performed through repeated fast Fourier transform and inverse fast Fourier transforms that scale as $\sim M \log M$.

If there are just enough gridpoints to resolve all scales from l to η , the scheme is DNS and solves the Navier-Stokes equations *exactly* within the sufficiently large \mathcal{D} of Figs. 1b and 1c. If the finest scales are removed however (without invoking any subgrid scale model) such that the smallest remaining resolved scale is significantly larger than η , then the scheme becomes an URDNS operating within the scale-restricted $\mathcal{W} \subset \mathcal{D}$. The Navier-Stokes equations cannot be solved exactly within \mathcal{W} due to the finest scales being subject to unphysical numerical dissipation. In comparison, the MPS algorithm operates within the MPS manifold of \mathcal{M} where the interscale correlations are limited while all the scales between l and η are still present.

Set-up of numerical experiments. For the TDJ simulations, we consider a square with edge length L_{box} with periodic boundary conditions and the initial conditions

$$\mathbf{V}(x, y, t = 0) = \mathbf{J}(y) + \mathbf{D}(x, y), \quad (9)$$

where $\mathbf{J}(y)$ is the initial jet profile

$$\mathbf{J}(y) = \hat{\mathbf{e}}_1 \frac{u_0}{2} \left[\tanh\left(\frac{y - y_{\min}}{h}\right) - \tanh\left(\frac{y - y_{\max}}{h}\right) - 1 \right] \quad (10)$$

with the streamwise direction along $\hat{\mathbf{e}}_1$. u_0 is the magnitude of the velocity differential between the jet and its surroundings, y_{\min} and y_{\max} describe the extent of the jet and h is the initial thickness of the vortex sheet. These parameters define the Reynolds number $\text{Re} = u_0 h / \nu$ and the time scale $T_0 = L_{\text{box}} / u_0$. The function

$$\mathbf{D} = \delta(\hat{\mathbf{e}}_1 d_1 + \hat{\mathbf{e}}_2 d_2) \quad (11)$$

in Eq. (9) is a small disturbance of miscellaneous wave modes needed to initiate the roll-up of the jet, statisti-

cally homogeneous along $\hat{\mathbf{e}}_1$ and divergence free, with

$$d_1(x, y) = 2 \frac{L_{\text{box}}}{h^2} \left[(y - y_{\max}) e^{-(y - y_{\max})^2 / h^2} + (y - y_{\min}) e^{-(y - y_{\min})^2 / h^2} \right] [\sin(8\pi x / L_{\text{box}}) + \sin(24\pi x / L_{\text{box}}) + \sin(6\pi x / L_{\text{box}})], \quad (12)$$

$$d_2(x, y) = \pi \left[e^{-(y - y_{\max})^2 / h^2} + e^{-(y - y_{\min})^2 / h^2} \right] \times [8 \cos(8\pi x / L_{\text{box}}) + 24 \cos(24\pi x / L_{\text{box}}) + 6 \cos(6\pi x / L_{\text{box}})], \quad (13)$$

$\delta = u_0 / (40A)$ and $A = \max_{x,y} \sqrt{d_1^2 + d_2^2}$. The components of the Reynolds stress-tensor shown in Fig. 2b are defined as

$$\tau_{ij}(y, t) = \overline{u'_i u'_j}, \quad (14)$$

where $u'_i = u_i - \bar{u}_i$ is the fluctuating part of the velocity component. The overbar denotes the ensemble-average across the statistically homogeneous streamwise direction,

$$\bar{u}_i = \frac{1}{L_{\text{box}}} \int_0^{L_{\text{box}}} u_i(x, y, t) dx. \quad (15)$$

Scaling all lengths with L_{box} , velocities with u_0 and time with T_0 , we set $y_{\min} = 0.4$, $y_{\max} = 0.6$, $h = 1/200$, and the penalty coefficient is $\mu = 2.5 \times 10^5$ in all MPS simulations.

The TGV simulations in 3-D are conducted on a cube with edge length L_{box} with periodic boundary conditions. We consider the initial flow field

$$\begin{aligned} u_1(\mathbf{r}, 0) &= -u_0 \sin(k_0 x) \cos(k_0 y) \cos(k_0 z), \\ u_2(\mathbf{r}, 0) &= u_0 \cos(k_0 x) \sin(k_0 y) \cos(k_0 z), \\ u_3(\mathbf{r}, 0) &= 0, \end{aligned} \quad (16)$$

where u_0 is the velocity amplitude of the initial vortex and its wavenumber is $k_0 = 2\pi / L_{\text{box}}$. The corresponding energy at $t = 0$ is $E_0 = u_0^2 / 2$, and the Reynolds number is defined using the integral scale as $\text{Re} = u_0 / (k_0 \nu)$.

In Fig. 3b we show the total kinetic energy dissipation

$$\varepsilon(t) = -\frac{1}{2} \frac{d}{dt} \int_{\mathcal{V}} |\mathbf{V}(\mathbf{r}, t)|^2 d\mathbf{r} \quad (17)$$

and enstrophy

$$\zeta(t) = \nu \int_{\mathcal{V}} |\nabla \times \mathbf{V}(\mathbf{r}, t)|^2 d\mathbf{r}, \quad (18)$$

where we integrate over the whole space \mathcal{V} . ζ is related to the viscous dissipation of kinetic energy [48]. For incompressible flows with periodic boundary conditions, the

INSE imply $\varepsilon(t) = \zeta(t)$. However, restricting the NVPS results in numerical diffusion violating this equality. In all TGV simulations, we set $T_0 = L_{\text{box}}/u_0$ and scale lengths and velocities with L_{box} and u_0 , respectively. In these units, the dimensionless penalty coefficient for MPS simulations is $\mu = 6.25 \times 10^4$.

Quantitative comparison between simulations

The accuracy of the MPS and URDNS simulations are gauged by comparing the *ensemble-aggregated* quantities of Figs 2b and 3b against DNS. Statistical quantities such as those must always be used when comparing different simulations due to the chaotic nature of turbulence. To aid the reader, we here provide a quantitative measure of the accuracies of MPS and URDNS by integrating the discrepancy DNS has to MPS and URDNS in Figs 2b and 3b.

We numerically calculate the *visual* difference between the Fig. 2b Reynolds stress of DNS (row 1) and that of MPS (rows 2-4) and URDNS (rows 5-7) as

$$\sigma(s, c) = \sqrt{\frac{\int_{L_{\text{box}}/5}^{4L_{\text{box}}/5} \int_0^{2T_0} [\tau_{12}^{\text{DNS}}(y, t) - \tau_{12}^{s,c}(y, t)]^2 dy dt}{\frac{6}{5} L_{\text{box}} T_0 \left[\max_{y,t} (\tau_{12}^{\text{DNS}}(y, t)) - \min_{y,t} (\tau_{12}^{\text{DNS}}(y, t)) \right]^2}}, \quad (19)$$

with s being the scheme in question and c the compression ratio of said scheme compared to DNS (as defined in *Results*). $\sigma(s, c)$ quantifies the root-mean-square of the *visual* difference between the subplots in Fig. 2b, and the closer $\sigma(s, c)$ is to 0, the nearer the Reynolds stress in question is to that of DNS. $\sigma(s, c)$ is tabulated in Table 1.

We also quantify the accuracy of URDNS and MPS against DNS for the TGV flow by integrating the numerical diffusion $|\zeta(t) - \epsilon(t)|$ of Fig. 3b. This is equivalent to

$$e(s, c) = \frac{1}{E_0} \int_0^{2T_0} |\zeta_{s,c}(t) - \epsilon_{s,c}(t)| dt, \quad (20)$$

when normalised by the initial total kinetic energy E_0 . The lower $e(s, c)$ is, the better is the accuracy of the relevant simulation. $e(s, c)$ is tabulated in Table 1. For comparison, we note that the corresponding (miniscule) error of DNS is $e(\text{DNS}) = 0.002$.

DATA AVAILABILITY

Our Code Ocean capsule [49] contains the raw output data from our MPS simulations. This data was generated using the C-functions `tntMpsBoxTurbulence2DTimeEvolutionRK2(...)` and `tntMpsBoxTurbulence3DTimeEvolutionRK2(...)`, using the initial conditions and parameters defined in the *Set-up of numerical experiments* section in *Methods*. Source Data for Figures 1, 2 and 3 is available with this manuscript.

Case	Compression	Scheme	Inaccuracy
TDJ	1:1	DNS	0
TDJ	1:8	MPS	0.0119
TDJ	1:8	URDNS	0.2612
TDJ	1:16	MPS	0.0485
TDJ	1:16	URDNS	0.3333
TDJ	1:64	MPS	0.2404
TDJ	1:64	URDNS	0.3201
TGV	1:1	DNS	0.002
TGV	1:25	MPS	0.0385
TGV	1:25	URDNS	0.1599
TGV	1:49	MPS	0.0844
TGV	1:49	URDNS	0.2133
TGV	1:78	MPS	0.2618
TGV	1:78	URDNS	0.4563

TABLE 1: **Quantitative comparison between MPS and URDNS simulations for the TDJ and TGV flow cases.** The rows corresponding to the TDJ flow tabulate $\sigma(s, c)$, as defined in Eq. (19). $\sigma(s, c)$ measures the discrepancy of the URDNS and MPS Reynolds stresses to that of DNS in Fig. 2b. The TGV flow case rows tabulate $e(s, c)$, which is defined in Eq. (20) and represents the total numerical diffusion in Fig. 3b. The nearer $\sigma(s, c)$ or $e(s, c)$ is to 0, the more accurate the simulation in question is.

CODE AVAILABILITY

The MATLAB code required to reproduce Figs. 1, 2 and 3 is available via Code Ocean [49]. The MPS simulations were done using the Tensor Network Theory Library [50].

ACKNOWLEDGEMENTS

The work at the University of Oxford was supported by the EPSRC Programme Grant DesOEQ (EP/P009565/1), and MK and DJ acknowledge financial support from the National Research Foundation, Prime Ministers Office, Singapore, and the Ministry of Education, Singapore, under the Research Centres of Excellence program, and furthermore, DJ acknowledges support by the excellence cluster 'The Hamburg Centre for Ultrafast Imaging - Structure, Dynamics and Control of Matter at the Atomic Scale' of the Deutsche Forschungsgemeinschaft. Current work at the University of Pittsburgh is supported by the National Science Foundation under Grant CBET-2042918. SD is thankful for the support from EPSRC New Investigator Award (EP/T031255/1) and New Horizons grant (EP/V04771X/1). The funders had no role in study design, data collection and analysis, decision to publish

or preparation of the manuscript. The authors would like to acknowledge the use of the University of Oxford Advanced Research Computing (ARC) facility in carrying out this work (<http://dx.doi.org/10.5281/zenodo.22558>). Finally, we thank the scientists and engineers at BAE Systems, Bristol for fruitful discussions and advice.

AUTHOR CONTRIBUTIONS

DJ conceived the research project and NG, ML, PG and DJ jointly planned it. NG, ML, SD, MK and DJ developed the quantum-inspired measure for interscale correlations based on Schmidt decompositions and hierarchical lattices. NG, ML and SD formulated the matrix product state algorithm and did the analytical calculations. NG, QYB and MK wrote the software. NG, HB and PG designed the numerical experiments for comparing MPS, URDNS and DNS. NG performed the numerical experiments. NG, ML, SD, HB, PG, MK and DJ analysed and interpreted the numerical results. NG, MK and DJ wrote the manuscript with contributions from ML, SD, HB, PG, and QYB helped revise the manuscript. The supplemental information was written by NG, ML, SD and MK. DJ supervised the project.

COMPETING INTERESTS

The authors declare no competing interests.

* nikgourianov@icloud.com

† dieter.jaksch@uni-hamburg.de

- [1] A. S. Monin and A. M. Yaglom, *Statistical Fluid Dynamics: Mechanics of Turbulence, vol. I* (Dover, New York, 2007).
- [2] A. S. Monin and A. M. Yaglom, *Statistical Fluid Dynamics: Mechanics of Turbulence, vol. II* (Dover, New York, 2007).
- [3] A. N. Kolmogorov, “The local structure of turbulence in incompressible viscous fluids for very large Reynolds’ numbers,” *Dokl. Akad. Nauk SSSR* **30**, 301–305 (1941).
- [4] G. L. Eyink, “Locality of turbulent cascades,” *Physica D* **207**, 91–116 (2005).
- [5] J. I. Cardesa, A. Vela-Martín, and J. Jiménez, “The turbulent cascade in five dimensions,” *Science* **357**, 782–784 (2017).
- [6] S. Chen, R. E. Ecke, G. L. Eyink, M. Rivera, M. Wan, and Z. Xiao, “Physical mechanism of the two-dimensional inverse energy cascade,” *Phys. Rev. Lett.* **96**, 084502 (2006).
- [7] S. A. Orszag and G. S. Patterson, “Numerical simulation of three-dimensional homogeneous isotropic turbulence,” *Phys. Rev. Lett.* **28**, 76–79 (1972).

- [8] A. K. M. F. Hussain, “Coherent structures—reality and myth,” *Phys. Fluids* **26**, 2816–2850 (1983).
- [9] P. Holmes, J. L. Lumley, G. Berkooz, and C. W. Rowley, *Turbulence, Coherent Structures, Dynamical Systems and Symmetry* (Cambridge University Press, Cambridge, 1996).
- [10] K. Taira, S. L. Brunton, S. T. M. Dawson, C. W. Rowley, T. Colonius, B. J. McKeon, O. T. Schmidt, S. Gordeyev, V. Theofilis, and L. S. Ukeiley, “Modal analysis of fluid flows: an overview,” *AIAA J.* **55**, 4013–4041 (2017).
- [11] D. Poulin, A. Qarry, R. Somma, and F. Verstraete, “Quantum simulation of time-dependent hamiltonians and the convenient illusion of hilbert space,” *Phys. Rev. Lett.* **106**, 170501 (2011).
- [12] R. Orús, “A practical introduction to tensor networks: Matrix product states and projected entangled pair states,” *Ann. Phys.* **349**, 117 – 158 (2014).
- [13] U. Schollwöck, “The density-matrix renormalization group in the age of matrix product states,” *Ann. Phys.* **326**, 96–192 (2011).
- [14] F. Verstraete, V. Murg, and J. I. Cirac, “Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems,” *Adv. Phys.* **57**, 143 (2008).
- [15] S. R. Clark and D. Jaksch, “Dynamics of the superfluid to Mott-insulator transition in one dimension,” *Phys. Rev. A* **70**, 043612 (2004).
- [16] Y. Simeng, D. A. Huse, and S. R. White, “Spin-liquid ground state of the $s = 1/2$ Kagome Heisenberg antiferromagnet,” *Science* **332**, 1173–1176 (2011).
- [17] R. Horodecki, P. Horodecki, M. Horodecki, and K. Horodecki, “Quantum entanglement,” *Rev. Mod. Phys.* **81**, 865–942 (2009).
- [18] J. Eisert, M. Cramer, and M. B. Plenio, “Colloquium: area laws for the entanglement entropy,” *Rev. Mod. Phys.* **82**, 277–306 (2010).
- [19] L. Onsager, “Statistical hydrodynamics,” *Nuovo Cim.* **6**, 279–287 (1949).
- [20] C. F. von Weizsäcker, “Das Spektrum der Turbulenz bei großen Reynoldsschen Zahlen,” *Z. Phys.* **124**, 614–627 (1948).
- [21] W. Heisenberg, “Zur statistischen Theorie der Turbulenz,” *Z. Phys.* **124**, 628–657 (1948).
- [22] R. Mathis, N. Hutchins, and I. Marusic, “Large-scale amplitude modulation of the small-scale structures in turbulent boundary layers,” *J. Fluid Mech.* **628**, 311 (2009).
- [23] P. Givi, “Model-free simulations of turbulent reactive flows,” *Prog. Energ. Combust. Sci.* **15**, 1–107 (1989).
- [24] M. E. Brachet, D. I. Meiron, S. A. Orszag, B. G. Nickel, R. H. Morf, and U. Frisch, “Small-scale structure of the Taylor–Green vortex,” *J. Fluid. Mech.* **130**, 411–452 (1983).
- [25] I. V. Oseledets, “Tensor-train decomposition,” *SIAM J. Sci. Comput.* **33**, 2295 (2011).
- [26] S. Holtz, T. Rohwedder, and R. Schneider, “On manifolds of tensors of fixed TT-rank,” *Numer. Math.* **120**, 701–731 (2012).
- [27] M. Lubasch, P. Moinier, and D. Jaksch, “Multigrid renormalization,” *J. Comput. Phys.* **372**, 587–602 (2018).
- [28] J. J. Garcia-Ripoll, “Quantum-inspired algorithms for multivariate analysis: from interpolation to partial differential equations,” *Quantum* **5**, 431 (2021).
- [29] P. Sagaut, *Large Eddy Simulation for Incompressible Flows: An Introduction* (Springer Berlin Heidelberg,

- 2006).
- [30] S. B. Pope, “Ten questions concerning the large-eddy simulation of turbulent flows,” *New J. Phys.* **6**, 35 (2004).
 - [31] Y. Zhiyin, “Large-eddy simulation: past, present and the future,” *Chin. J. Aeronaut.* **28**, 11–24 (2015).
 - [32] M. V. Rakhuba and I. V. Oseledets, “Fast multidimensional convolution in low-rank tensor formats via cross approximation,” *SIAM J. Sci. Comput.* **37**, A565–A582 (2015).
 - [33] G. Vidal, “Entanglement renormalization,” *Phys. Rev. Lett.* **99**, 220405 (2007).
 - [34] G. Evenbly and G. Vidal, “Class of highly entangled many-body states that can be efficiently simulated,” *Phys. Rev. Lett.* **112**, 240502 (2014).
 - [35] P. A. Libby and F. A. Williams, eds., *Turbulent Reacting Flows* (Academic Press, London, UK, 1994).
 - [36] S. B. Pope, “PDF Methods for Turbulent Reactive Flows,” *Prog. Energ. Combust.* **11**, 119–192 (1985).
 - [37] A. G. Nouri, M. B. Nik, P. Givi, D. Livescu, and S. B. Pope, “Self-contained filtered density function,” *Phys. Rev. Fluids* **2**, 094603 (2017).
 - [38] M. Lubasch, J. Joo, P. Moinier, M. Kiffner, and D. Jaksch, “Variational quantum algorithms for nonlinear problems,” *Phys. Rev. A* **101**, 010301(R) (2020).
 - [39] S. Lloyd, G. De Palma, C. Gokler, B. Kiani, Z.-W. Liu, M. Marvian, F. Tennie, and T. Palmer, “Quantum algorithm for nonlinear differential equations,” Preprint: arXiv:2011.06571 (2020).
 - [40] J.-P. Liu, H. Ø. Kolden, H. K. Krovi, N. F. Loureiro, K. Trivisa, and A. M. Childs, “Efficient quantum algorithm for dissipative nonlinear differential equations,” *Proc. Natl. Acad. Sci. U.S.A.* **118** (2021).
 - [41] M. P. Zaletel and F. Pollmann, “Isometric tensor network states in two dimensions,” *Phys. Rev. Lett.* **124**, 037201 (2020).
 - [42] S. H. Lin, R. Dilip, A. G. Green, A. Smith, and F. Pollmann, “Real- and imaginary-time evolution with compressed quantum circuits,” *PRX Quantum* **2**, 010342 (2021).
 - [43] A. V. Fiacco and G. P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques* (John Wiley, New York, 1968).
 - [44] A. V. Fiacco, “Penalty methods for mathematical programming in e^n with general constraint sets,” *J. Optim. Theor. Appl.* **6**, 252–268 (1970).
 - [45] A. J. Chorin, “Numerical solution of the Navier-Stokes equations,” *Math. Comp.* **22**, 745–762 (1968).
 - [46] B. Fornberg, “Generation of finite difference formulas on arbitrarily spaced grids,” *Math. Comput.* **51**, 699–706 (1988).
 - [47] A. J. Chorin, “The numerical solution of the Navier-Stokes equations for an incompressible fluid,” *Bull. Am. Math. Soc.* **73**, 928–931 (1967).
 - [48] C. Foias, O. Manley, R. Rosa, and R. Temam, *Navier-Stokes Equations and Turbulence*, Vol. 83 (Cambridge University Press, 2001).
 - [49] N. Goussier, *A Quantum Inspired Approach to Exploit Turbulence Structures [source code]*, *Code Ocean* <https://doi.org/10.24433/CO.4124213.v1> (2021).
 - [50] S. Al-Assam, S. R. Clark, and D. Jaksch, “The tensor network theory library,” *J. Stat. Mech. Theor. Exp.* **2017**, 093102 (2017).
 - [51] J. Jeong and F. Hussain, “On the identification of a vortex,” *J. Fluid. Mech.* **285**, 69–94 (1995).

Supplementary Information: A Quantum Inspired Approach to Exploit Turbulence Structures

Nikita Gourianov^{1,*}, Michael Lubasch², Sergey Dolgov³, Quincy Y. van den Berg¹, Hessam Babae⁴, Peyman Givi⁴, Martin Kiffner^{5,1}, and Dieter Jaksch^{1,5,6}

¹*Clarendon Laboratory, University of Oxford, Oxford, UK*

²*Cambridge Quantum Computing Limited, London, UK*

³*Department of Mathematical Sciences, University of Bath, Bath, UK*

⁴*Department of Mechanical Engineering and Materials Science,
University of Pittsburgh, Pittsburgh, PA, USA*

⁵*Centre for Quantum Technologies, National University of Singapore, Singapore and*

⁶*Insitut für Laserphysik, Universität Hamburg, Hamburg, Germany*

* nikgourianov@icloud.com

CONTENTS

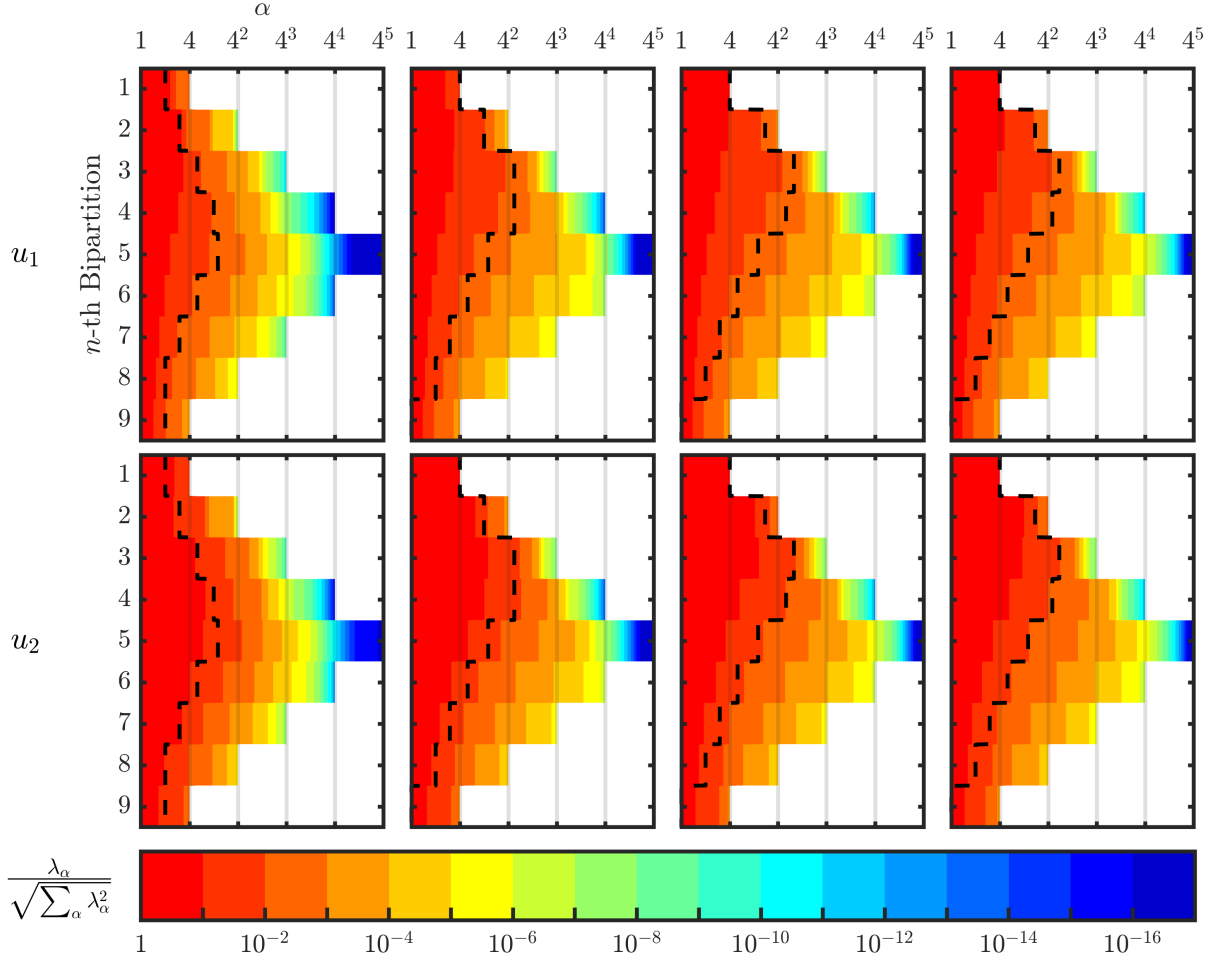
1. Additional results	3
A. Schmidt spectra	3
B. Entanglement entropy	5
2. Matrix product state representation	7
A. Encoding a flow component as a matrix product state	7
B. Number of physical variables	7
C. Schmidt decomposition	8
D. Encoding vector fields	8
E. Encoding quantum states	9
F. Quantum Schmidt decomposition	9
3. Matrix product state representation of shock waves	11
A. Burgers' equation and its analytical solution	11
B. Mathematical solution for initial δ function	11
C. Matrix product state representation of general solution	13
D. Triangular waves as matrix product states	13
4. Matrix product state algorithm	16
A. The variational problem	16
B. Local minimisation	17
C. Global minimisation	18
D. Theoretical computational scaling	18
E. Demonstration of computational scaling	19
F. Arithmetic intensity	19
5. Quantum algorithm	21
A. Variational quantum algorithm	21
B. Comparison to alternative proposals	21
6. Graphical notation	23
References	25

1. ADDITIONAL RESULTS

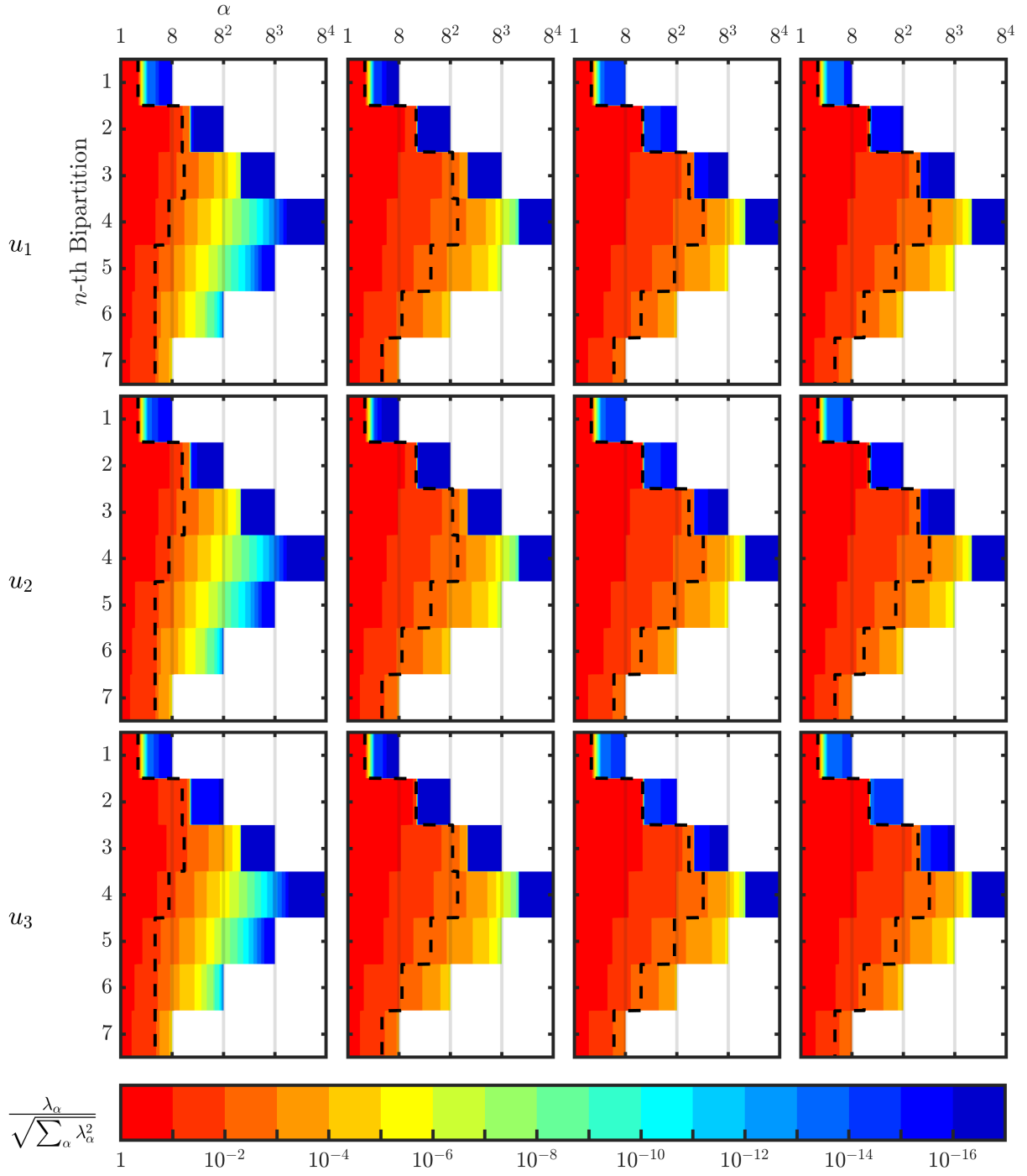
This section expands upon the results of the main text. In particular, we provide further details on interscale correlations obtained from Schmidt decompositions of the DNS results of the TDJ (Re=1000) and TGV (Re=800) flows studied in the main text. In Sec. 1 A we examine the Schmidt spectra λ_α and in Sec. 1 B we calculate the von Neumann entanglement entropy $H(n, t)$ following from these spectra.

A. Schmidt spectra

Supp. Figs. 1 and 2 show the Schmidt spectra of the 2-D TDJ and 3-D TGV flows for different times and bipartitions n . We have also included in both figures contours corresponding to $d_{99}(n, t)$ used as a single figure of merit for the overall accuracy of the velocity field in the main text. This shows that truncating the Schmidt spectra at these values does not discard any relevant interscale correlations of the flow.



Supplementary Figure 1: **Schmidt spectrum of the TDJ.** The normalised Schmidt coefficients obtained from the DNS of each velocity component of the TDJ flow at Re=1000 are shown at times $t/T_0 = 0.25, 0.75, 1.25, 1.75$ (left to right), for each of the 9 bipartitions available on the 1024×1024 DNS grid. The Schmidt coefficients are sorted in descending order and are normalised such that the sum of their squares equals 1. The black, dashed lines denote the $d_{99}(n, t)$ as used in the main text.



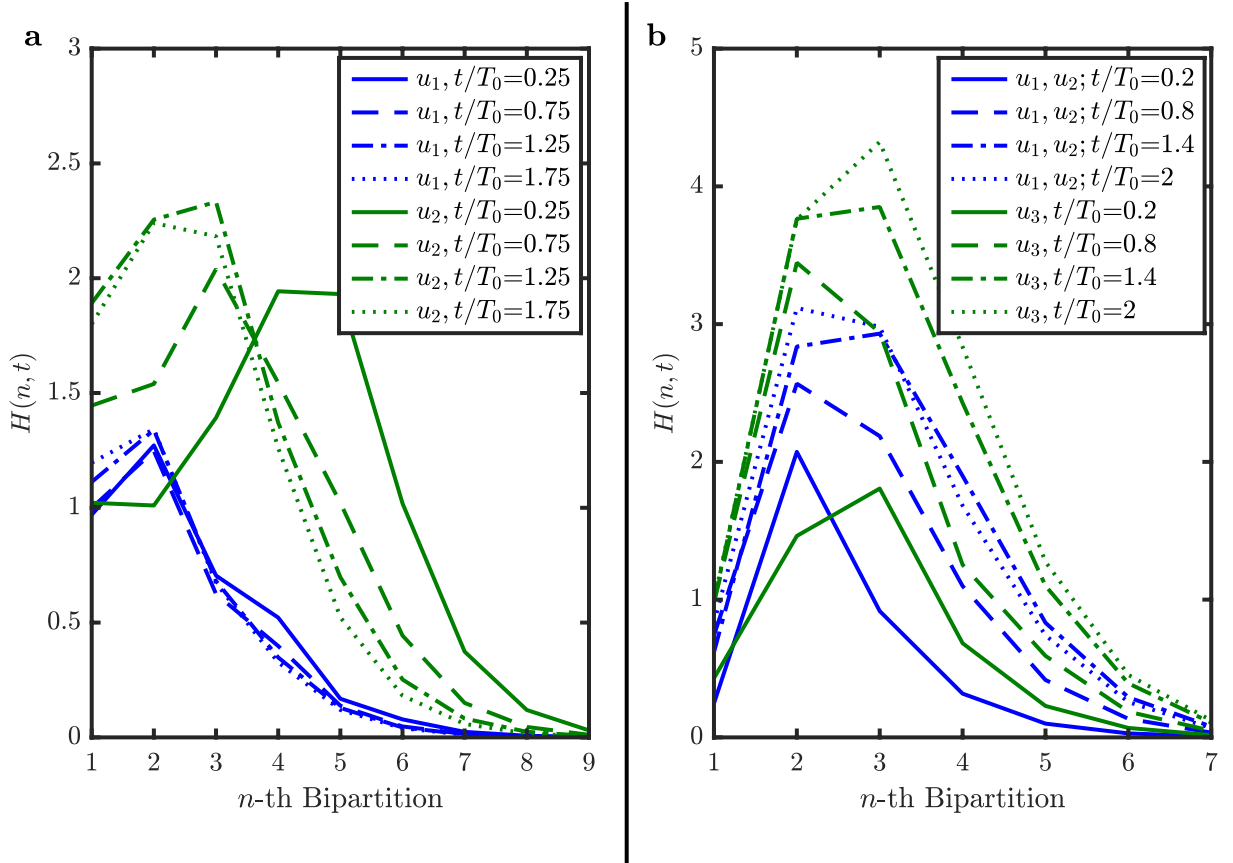
Supplementary Figure 2: **Schmidt spectrum of the TGV.** The normalised Schmidt coefficients obtained from the DNS of each velocity component of the TGV flow for $\text{Re}=800$ are shown at times $t/T_0 = 0.2, 0.8, 1.4, 2$ (left to right) for each of the 7 bipartitions available on the $256 \times 256 \times 256$ DNS grid. The Schmidt coefficients are sorted in descending order and are normalised such that the sum of their squares equals 1. The black, dashed lines denote the $d_{99}(n, t)$ as used in the main text.

B. Entanglement entropy

We adopt the standard definition of the von Neumann entanglement entropy $H(n, t)$ from quantum information theory. For a given spectrum $\lambda_\alpha = \lambda_\alpha(n, t)$ this is defined as

$$H(n, t) = \frac{-1}{E(t)} \sum_{\alpha=1}^{d(n)} \lambda_\alpha(n, t)^2 \log [\lambda_\alpha(n, t)^2 / E(t)], \quad (\text{S1})$$

with the normalisation factor $E(t) = \sum_{\alpha=1}^{d(n)} \lambda_\alpha(n, t)^2$. The entanglement entropy for the 2-D TDJ flow shown in Supp. Fig. 3a shifts towards bipartitions between coarser length scales (i.e. lower n) with increasing time. This behaviour is consistent with a 2-D inverse energy cascade [1] where energy is carried from fine to coarse length scales as time progresses, e.g. through vortex merging. These dynamics are particularly pronounced for the cross-stream u_2 velocity component, for which a large number of fine scale disturbances become energised by the shear and grow in size ($t/T_0 \approx 0.25, 0.75$) until the eventual collapse of the jet and the saturation of the shear layer ($t/T_0 \approx 1.25, 1.75$). At later times no further growth of the disturbances occur. Remarkably, these physics are visible in the dynamics of the entanglement entropy. At $t/T_0 = 0.25$ the entanglement entropy is large for all bipartitions between $n = 1$ and $n = 7$, indicating significant correlations between all length scales. At later times ($t/T_0 = 1.25, 1.75$), when the energy increasingly flows towards coarser length scales, also the entanglement entropy shifts towards lower n bipartitions as shown in Supp. Fig. 3a.



Supplementary Figure 3: **Von Neumann entanglement entropy between length scales in the TDJ and TGV flows.** The entanglement entropy at each of the bipartitions of TDJ (a) and TGV (b) calculated from the DNS solutions (i.e. $d(n)$ is maximal for all n) are illustrated at various times. The 1024×1024 grid of the DNS of the TDJ are bipartitioned along 9 length scales, while 7 bipartitions are used for the $256 \times 256 \times 256$ grid DNS of the TGV. A symmetry exists between the u_1 and u_2 components of the TGV velocity field, which makes them overlap within (b).

For the 3-D TGV flow shown in Supp. Fig. 3b the opposite happens. There, fine length scales become energised with increasing time. Correspondingly, the entanglement entropy increases at larger values of n with increasing time. This increase is consistent with the hypothesis of a direct energy cascade in 3-D turbulent flows [2], where energy is transported to progressively finer and finer length scales until the Kolmogorov microscale is reached and the energy starts being dissipated by viscosity. However, unlike in the 2-D TDJ case, the outflow of energy is not accompanied by a corresponding reduction of interscale correlations. Instead, the entanglement entropy increases with time for all bipartitions. This is a result of the disorder due to the collapse of the TGV into the various worm-like vortical structures discussed in the main text. We finally note that both of these behaviours for the 2-D TDJ and the 3-D TGV are consistent with the dynamics of d_{99} studied in the main text.

2. MATRIX PRODUCT STATE REPRESENTATION

Here we describe how the matrix product state (MPS) formalism can be used to encode scalar functions, vector fields as well as quantum wavefunctions. The encoding of scalar functions is outlined in Sec. 2 A. A general equation is provided for the number of parameters available in MPSs in Sec. 2 B, and Sec. 2 C derives Eq. (2) of the main text by performing a Schmidt decomposition in the MPS format. We describe in Sec. 2 D our strategy for encoding vector fields into MPS. Finally, Secs. 2 E and 2 F demonstrate the MPS ansatz and Schmidt decomposition in the context of quantum mechanics for comparison.

A. Encoding a flow component as a matrix product state

Consider a flow in a K -dimensional cube with edge length L_{box} where each spatial dimension is discretised by 2^N grid points. The whole K -dimensional grid thus comprises $M = 2^{KN}$ equally spaced points \mathbf{r}_q . First, we introduce a one-to-one mapping between the grid point vectors \mathbf{r}_q and a tuple of positive integers,

$$\mathbf{r}_q \leftrightarrow (q^1, q^2, \dots, q^K), \quad (\text{S2})$$

where $q^i \in \{0, \dots, 2^N - 1\}$ is the index of the grid point in the direction $\hat{\mathbf{e}}_i$. The binary representation $(\dots)_2$ of these indices q^i requires N bits,

$$q^i = (\sigma_1^i, \sigma_2^i, \dots, \sigma_N^i)_2, \quad (\text{S3})$$

where $\sigma_n^i \in \{0, 1\}$, $n = 1, \dots, N$, and σ_1^i and σ_N^i are the most and least significant bits, respectively.

Now consider a single Cartesian velocity component u and ignore its time dependence. The discretisation renders the velocity components into functions of the grid points \mathbf{r}_q . Due to the one-to-one mapping in Eq. (S2), they can also be regarded as functions of the indices q^i , $u(\mathbf{r}_q) \equiv u(q^1, \dots, q^K)$, and hence ultimately of σ_n^i . However, there is a freedom in the order of mapping σ_n^i into \mathbf{r}_q . We group all indices associated with the same bit, i.e. length scale:

$$\omega_n = (\sigma_n^1, \sigma_n^2, \dots, \sigma_n^K)_2, \quad (\text{S4})$$

such that $\omega_n \in \{0, \dots, 2^K - 1\}$. We approximate the discrete function values $u(\mathbf{r}_q)$ by an MPS $v(\mathbf{r}_q, \chi)$ defined as follows:

$$u(\mathbf{r}_q) \approx v(\mathbf{r}_q, \chi) = A^{\omega_1} A^{\omega_2} \dots A^{\omega_N}, \quad (\text{S5})$$

where the matrices A^{ω_n} have dimensions $d(n-1) \times d(n)$ with $n = 1, \dots, N$,

$$d(n) = \min(\Gamma^{K-D}(n), \chi) \quad (\text{S6})$$

and $\Gamma^{K-D}(n) = \min(2^{Kn}, 2^{K(N-n)})$, cf. $\Gamma^{3-D}(n)$ in the main text. Note that $d(0) = d(N) = 1$, and thus the MPS evaluates to a number. Each matrix A^{ω_n} in this MPS represents a particular length scale, and the maximum amount of interscale correlations that can be captured by the MPS is controlled via the bond dimension χ . If χ is set to $\chi = 2^{K\lfloor N/2 \rfloor}$, the approximation in Eq. (S5) becomes exact.

B. Number of physical variables

The number of physical parameters of $v(\mathbf{r}_q, \chi)$ is given by

$$Q = 2^K \sum_{n=1}^N d(n-1)d(n) - \sum_{n=1}^{N-1} d(n)^2. \quad (\text{S7})$$

The first sum is the total number of parameters in Eq. (S5) while the second sum represents the intrinsic gauge degrees of freedom of the MPS format [3]. When χ is maximal, i.e. $\chi = 2^{K\lfloor N/2 \rfloor}$, we get $Q = 2^{KN} = M$ and that any function on the grid can be perfectly captured in MPS form.

C. Schmidt decomposition

Next we show that the MPS representation of u in Eq. (S5) is consistent with the representation in Eq. (2) of the main text. To this end, we note that the gauge degrees of freedom allow one to bring v of Eq. (S5) to mixed canonical form [4],

$$v(\mathbf{r}_q, \chi) = \sum_{\alpha=1}^{d(n)} \lambda_{\alpha} \left[\hat{A}^{\omega_1} \dots \hat{A}^{\omega_n} \right]_{\alpha} \left[\hat{B}^{\omega_{n+1}} \dots \hat{B}^{\omega_N} \right]_{\alpha}, \quad (\text{S8})$$

where $\lambda_1 \geq \lambda_2 \dots \geq \lambda_{d(n)}$ are the Schmidt coefficients. The matrices \hat{A}^{ω_n} and \hat{B}^{ω_n} have the same dimensions as A^{ω_n} , and satisfy the relations

$$\begin{aligned} \sum_{\omega_n=0}^{2^K-1} \left(\hat{A}^{\omega_n} \right)^t \hat{A}^{\omega_n} &= \mathbb{1}, \\ \sum_{\omega_n=0}^{2^K-1} \hat{B}^{\omega_n} \left(\hat{B}^{\omega_n} \right)^t &= \mathbb{1}, \end{aligned} \quad (\text{S9})$$

where $(\cdot)^t$ denotes the matrix transpose and $\mathbb{1}$ is the identity matrix. Defining

$$\begin{aligned} R_{\alpha}(\mathbf{X}_k) &= \left[\hat{A}^{\omega_1} \dots \hat{A}^{\omega_n} \right]_{\alpha}, \\ f_{\alpha}(\mathbf{x}_l) &= \left[\hat{B}^{\omega_{n+1}} \dots \hat{B}^{\omega_N} \right]_{\alpha}, \end{aligned} \quad (\text{S10})$$

allows us to cast Eq. (S8) into the form

$$u(\mathbf{r}_q) = v(\mathbf{r}_q, \chi) = \sum_{\alpha=1}^{d(n)} \lambda_{\alpha} R_{\alpha}(\mathbf{X}_k) f_{\alpha}(\mathbf{x}_l), \quad (\text{S11})$$

when χ is maximal, i.e. $\chi = \Gamma^{K-D}(n)$. The grid points \mathbf{X}_k (coarse grid) and \mathbf{x}_l (fine grid) are defined through the mapping in Eq. (S2) and the integer values

$$\mathbf{X}_k: \quad q^i = (\sigma_1^i, \dots, \sigma_n^i, 0, \dots, 0)_2, \quad (\text{S12})$$

and

$$\mathbf{x}_l: \quad q^i = (0, \dots, 0, \sigma_{n+1}^i, \dots, \sigma_N^i)_2, \quad (\text{S13})$$

respectively. By definition we have $\mathbf{r}_q = \mathbf{X}_k + \mathbf{x}_l$, and Eq. (S9) implies the orthonormality conditions

$$\sum_k R_{\alpha}(\mathbf{X}_k) R_{\beta}(\mathbf{X}_k) = \sum_l f_{\alpha}(\mathbf{x}_l) f_{\beta}(\mathbf{x}_l) = \delta_{\alpha\beta}, \quad (\text{S14})$$

where $\delta_{\alpha\beta}$ is the Kronecker delta. Writing Eq. (S11) for all three Cartesian velocity components and explicitly re-introducing the time dependence of u_i , λ_{α} , $R(\alpha)$ and $f(\alpha)$, we obtain Eq. (2) of the main text.

D. Encoding vector fields

Let $\mathbf{V}(t, \mathbf{r}_q)$ be the K -D vector field of Eq. (1) of the main text. We now simply follow the recipe outlined in the previous Sec. 2 A for each velocity component u_i of \mathbf{V} , while again ignoring t . This will result in each component individually being represented as a MPS of bond dimension χ , giving the MPS vector field $\mathbf{W}(\mathbf{r}_q)$:

$$\mathbf{V}(\mathbf{r}_q) \approx \mathbf{W}(\mathbf{r}_q, \chi) = \sum_{i=1}^K \hat{e}_i A_i^{\omega_1} A_i^{\omega_2} \dots A_i^{\omega_N}, \quad (\text{S15})$$

with the matrices $A_i^{\omega_n}$ being of identical dimension to the A^{ω_n} matrices of Eq. (S5).

E. Encoding quantum states

In the above sections we describe how MPS can be used to encode scalar functions and vector fields. For the sake of comparison, we here outline how MPS are traditionally used to encode 1-D quantum many-body states.

Assume we have N spin-1/2 particles organised along a spin chain with open boundary conditions. The pure-state wavefunction of this system is given by

$$|\Psi\rangle = \sum_{\{i_n\}=0,1} C_{i_1 i_2 \dots i_N} |i_1\rangle |i_2\rangle \dots |i_N\rangle, \quad (\text{S16})$$

and is an element of the N particle Hilbert space $\mathcal{H} = \mathcal{H}_0^{\otimes N}$, with $|i_n\rangle \in \mathcal{H}_0$ and $\text{span}(\mathcal{H}_0) = \{|0\rangle, |1\rangle\}$. The amplitude of the basis states constituting $|\Psi\rangle$ is given by the order- N complex tensor C of dimension 2^N , and the wavefunction is normalised such that $\langle\Psi|\Psi\rangle = 1$. It is here possible to represent $|0\rangle$ and $|1\rangle$ as respectively the vectors $(1, 0)^\dagger$ and $(0, 1)^\dagger$, with $(\cdot)^\dagger$ being the conjugate-transpose.

The indices i_1, i_2, \dots, i_N are analogous to ω_n of Eq. (S4) for $K = 1$. This means C can be decomposed into a MPS analogously to $u(\mathbf{r}_q)$:

$$C_{i_1 i_2 \dots i_N} \approx \tilde{C}_{i_1 i_2 \dots i_N}(\chi) = A^{i_1} A^{i_2} \dots A^{i_N} \quad (\text{S17})$$

where the matrices A^{i_n} have dimensions $d(n-1) \times d(n)$ with $n = 1, \dots, N$,

$$d(n) = \min(\Gamma(n), \chi) \quad (\text{S18})$$

and $\Gamma(n) = \min(2^n, 2^{N-n})$. Hence if χ is set to $\chi = 2^{\lfloor N/2 \rfloor}$, the relationship in Eq. (S17) becomes exact. Due to the open boundary conditions, $d(0) = d(D) = 1$ and thus the MPS evaluates to a number. For a detailed step-by-step guide on how the MPS decomposition can be performed using SVDs, see [4]. Each matrix A^{i_n} of the above MPS is associated with a particular spin-1/2 particle, and the maximum amount of entanglement of the system that can be captured by the MPS is controlled by the bond dimension χ . Compare these properties with those of the MPS of the scalar function in Sec. 2 A for $K = 1$.

The above MPS representation of C allows us to rewrite the wavefunction as

$$|\Psi\rangle \approx |\tilde{\Psi}(\chi)\rangle = \sum_{\{i_n\}=0,1} A^{i_1} A^{i_2} \dots A^{i_N} |i_1\rangle |i_2\rangle \dots |i_N\rangle. \quad (\text{S19})$$

$|\tilde{\Psi}(\chi)\rangle$ is still a vector within the exponentially large \mathcal{H} , but the MPS decomposition has parameterised it with only a *polynomial* number of variables (when χ is limited). For many quantum systems, in particular area-law following ones, using the MPS decomposition does not cause a significant loss in accuracy [5].

F. Quantum Schmidt decomposition

The Schmidt decomposition can be employed on $|\tilde{\Psi}(\chi)\rangle$ just as it was employed on $u(\mathbf{r}_q, \chi)$ in Sec. 2 C. Putting the MPS into mixed canonical form with the canonical centre at the n -th bond yields

$$|\tilde{\Psi}(\chi)\rangle = \sum_{\alpha=1}^{d(n)} \lambda_\alpha \left[\hat{A}^{i_1} \dots \hat{A}^{i_n} \right]_\alpha \left[\hat{B}^{i_{n+1}} \dots \hat{B}^{i_N} \right]_\alpha |i_1\rangle |i_2\rangle \dots |i_N\rangle. \quad (\text{S20})$$

This is nothing else than the Schmidt decomposition, as can straightforwardly be seen by inserting

$$\begin{aligned} |\psi_\alpha^{1:n}\rangle &= \left[\hat{A}^{i_1} \dots \hat{A}^{i_n} \right]_\alpha |i_1\rangle \dots |i_n\rangle, \\ |\psi_\alpha^{n+1:N}\rangle &= \left[\hat{B}^{i_{n+1}} \dots \hat{B}^{i_N} \right]_\alpha |i_{n+1}\rangle \dots |i_N\rangle \end{aligned} \quad (\text{S21})$$

into Eq. (S20), yielding the Schmidt decomposition between two quantum sub-systems in its traditional form:

$$|\tilde{\Psi}(\chi)\rangle = \sum_{\alpha=1}^{d(n)} \lambda_\alpha |\psi_\alpha^{1:n}\rangle |\psi_\alpha^{n+1:N}\rangle. \quad (\text{S22})$$

The difference between this Schmidt decomposition and the one in Eq. (S8), is that the Schmidt decomposition of Eq. (S8) exposes the interscale correlations between a set of coarse and fine length scales, whilst the quantum Schmidt decomposition reveals the entanglement between two bipartitions of the spin chain

For a detailed description of how the entanglement of a quantum system might be studied in the context of MPS, see [4]. The tools provided there can also be used to study the interscale correlations of turbulent flows, like how we calculated the von Neumann entanglement entropy of the TDJ and TGV in Sec. 1 B.

3. MATRIX PRODUCT STATE REPRESENTATION OF SHOCK WAVES

In the following, we analyse how well MPS approximate hump-solutions of the one-dimensional Burgers' equation. First we introduce Burgers' equation and present its known mathematical solution in Sec. 3 A. Then in Sec. 3 B we derive the solution of Burgers' equation for an initial δ function. We show that this so-called hump-solution has, in general, an exponentially accurate MPS representation in Sec. 3 C. In the special case of vanishing viscosity, this hump solution becomes a triangular wave, i.e. a prototypical shock-wave. We provide the exact MPS description for this shock-wave in Sec. 3 D.

A. Burgers' equation and its analytical solution

Burgers' equation [6, 7] is given by

$$\frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial x^2} - u \frac{\partial u}{\partial x} \quad (\text{S23})$$

where $u = u(x, t)$. This equation simplifies by defining a function $w = w(x, t)$ via

$$\begin{aligned} u(x, t) &= -2\nu \frac{1}{w} \frac{\partial w}{\partial x} \\ &= -2\nu \frac{\partial \log(w)}{\partial x}, \end{aligned} \quad (\text{S24})$$

also known as the Hopf-Cole transformation [8, 9], which leads to the heat equation

$$\frac{\partial w}{\partial t} = \nu \frac{\partial^2 w}{\partial x^2}. \quad (\text{S25})$$

The heat equation has the general mathematical solution

$$w(x, t) = \frac{1}{2\sqrt{\pi\nu t}} \int_{-\infty}^{\infty} w_0(\alpha) e^{-\frac{(x-\alpha)^2}{4\nu t}} d\alpha \quad (\text{S26})$$

where w_0 denotes the initial function at time $t = 0$. We obtain w_0 from the initial function to Burgers' equation $u_0(x) = u(x, 0)$ by inverting Eq. (S24) for $t = 0$:

$$\begin{aligned} w_0(x) &= w(x, 0) \\ &= e^{-\frac{1}{2\nu} \int_a^x u_0(y) dy} \end{aligned}$$

where a can be chosen freely. Using Eq. (S24) we obtain the following solution to Burgers' equation

$$u(x, t) = \frac{1}{t} \frac{\int_{-\infty}^{\infty} w_0(\alpha) (x - \alpha) e^{-\frac{(x-\alpha)^2}{4\nu t}} d\alpha}{\int_{-\infty}^{\infty} w_0(\alpha) e^{-\frac{(x-\alpha)^2}{4\nu t}} d\alpha} \quad (\text{S28})$$

where $w_0(\alpha)$ represents the initial function defined in Eq. (S27).

B. Mathematical solution for initial δ function

Following [10], we investigate the so-called hump solution to the Burgers' equation. It is produced when using the δ function initial condition of

$$u_0(x) = Z\delta(x - x_0), \quad (\text{S29})$$

where Z is a normalising constant. Plugging this into Eq. (S27) gives

$$w_0(x) = e^{-\frac{Z}{2\nu} \int_a^x \delta(y - x_0) dy}. \quad (\text{S30})$$

We choose $a = x_0 + \epsilon$, where $\epsilon \rightarrow 0$ denotes an infinitesimally small number, so that

$$\begin{aligned} w_0(x) &= e^{\frac{Z}{2\nu}} & x \leq x_0, \\ w_0(x) &= 1 & x > x_0. \end{aligned} \quad (\text{S31})$$

We split the integration into two parts

$$\int_{-\infty}^{\infty} w_0(\alpha) \dots d\alpha = \int_{-\infty}^{x_0} e^{\frac{Z}{2\nu}} \dots d\alpha + \int_{x_0+\epsilon}^{\infty} \dots d\alpha. \quad (\text{S32})$$

To simplify our notation, we do not explicitly include ϵ from now on. The evaluation of the integral in the numerator of Eq. (S28) is straightforward and to evaluate the integral in the denominator we make use of the substitution $\beta = (x - \alpha)/(2\sqrt{\nu t})$. Ultimately these calculations lead to the result

$$u(x, t) = \sqrt{\frac{\nu}{t}} \frac{(e^{\frac{Z}{2\nu}} - 1) e^{-\frac{(x-x_0)^2}{4\nu t}}}{\sqrt{\pi} + \frac{\sqrt{\pi}}{2} (e^{\frac{Z}{2\nu}} - 1) \text{erfc}(\frac{x-x_0}{2\sqrt{\nu t}})}, \quad (\text{S33})$$

where $\text{erfc}(x) = (2/\sqrt{\pi}) \int_x^{\infty} \exp(-\alpha^2) d\alpha$ is the so-called complementary error function.

To analyse the solution (S33) in the limit of $\nu \rightarrow 0$, we first consider $x \leq x_0$ for which

$$\begin{aligned} \lim_{\nu \rightarrow 0} \text{erfc}\left(\frac{x-x_0}{2\sqrt{\nu t}}\right) &= 2, & x < x_0 \\ \lim_{\nu \rightarrow 0} \text{erfc}\left(\frac{x-x_0}{2\sqrt{\nu t}}\right) &= 1, & \text{for } x = x_0 \end{aligned} \quad (\text{S34})$$

and therefore Eq. (S33) becomes

$$\lim_{\nu \rightarrow 0} u(x, t) = 0, \quad x \leq x_0. \quad (\text{S35})$$

For $x > x_0$ we use the $x \rightarrow \infty$ asymptotic expansion of the complementary error function

$$\text{erfc}(x) = \frac{e^{-x^2}}{\sqrt{\pi}x} + O(x^{-3}e^{-x^2}) \quad (\text{S36})$$

so that for $\nu \rightarrow 0$

$$\text{erfc}\left(\frac{x-x_0}{2\sqrt{\nu t}}\right) \approx 2\sqrt{\nu t} \frac{e^{-\frac{(x-x_0)^2}{4\nu t}}}{\sqrt{\pi}(x-x_0)}, \quad x > x_0, \quad (\text{S37})$$

which transforms Eq. (S33) into

$$u(x, t) = \sqrt{\frac{\nu}{t}} \frac{(e^{\frac{Z}{2\nu}} - 1) e^{-\frac{(x-x_0)^2}{4\nu t}}}{\sqrt{\pi} + \sqrt{\nu t} (e^{\frac{Z}{2\nu}} - 1) \frac{e^{-\frac{(x-x_0)^2}{4\nu t}}}{x-x_0}}. \quad (\text{S38})$$

We observe that

$$\begin{aligned} \lim_{\nu \rightarrow 0} e^{-\frac{(x-x_0)^2}{4\nu t}} &= 0, & x > x_0, \\ \lim_{\nu \rightarrow 0} \left(e^{\frac{Z}{2\nu} - \frac{(x-x_0)^2}{4\nu t}} \right) &= \infty, & x_0 < x < x_0 + \sqrt{2Zt}, \\ \lim_{\nu \rightarrow 0} \left(e^{\frac{Z}{2\nu} - \frac{(x-x_0)^2}{4\nu t}} \right) &= 1, & \text{for } x = x_0 + \sqrt{2Zt}, \\ \lim_{\nu \rightarrow 0} \left(e^{\frac{Z}{2\nu} - \frac{(x-x_0)^2}{4\nu t}} \right) &= 0, & x > x_0 + \sqrt{2Zt}. \end{aligned} \quad (\text{S39})$$

Using these solutions along with Eq. (S38) gives in the limit $\nu \rightarrow 0$:

$$\begin{aligned} u(x, t) &= 0, & x \leq x_0 \wedge x > x_0 + \sqrt{2Zt}, \\ u(x, t) &= \frac{x-x_0}{t}, & x_0 < x < x_0 + \sqrt{2Zt}. \end{aligned} \quad (\text{S40})$$

As ν grows large, it is straightforward to show that Eq. (S33) approaches the Gaussian

$$u(x, t) = \frac{Z}{2\sqrt{\pi\nu t}} e^{-\frac{(x-x_0)^2}{4\nu t}}. \quad (\text{S41})$$

And for $\nu \rightarrow \infty$, this Gaussian will approach a uniform function with an amplitude tending towards zero.

C. Matrix product state representation of general solution

Here we show that MPS accurately represent the previously derived mathematical solution for the propagation of an initial δ function with Burgers' equation. For $K = 1$ the MPS digit of Eq. (S4) collapses to the range $\omega_n \equiv \sigma_n^1 \in \{0, 1\}$. In turn, every grid point can be defined as $\mathbf{r}_q = x_q = q/2^N$, where $q = (\omega_1, \dots, \omega_N)_2$.

The discretised δ -function $u_0(x_q) = Z\delta(q - j)$ with a normalising constant Z and a peak position $j = (b_1, \dots, b_N)_2$, $b_n \in \{0, 1\}$, can be represented as an exact MPS (see Eq. (S5)) of bond dimension $\chi = 1$ and factors

$$A^{\omega_1} = \begin{cases} Z, & \omega_1 = b_1, \\ 0, & \text{otherwise,} \end{cases} \quad \text{and} \quad A^{\omega_n} = \begin{cases} 1, & \omega_n = b_n, \\ 0, & \text{otherwise,} \end{cases} \quad (\text{S42})$$

for $n = 2, \dots, N$.

Time evolution of an initial δ function with Burgers' equation has the solution of Eq. (S33). The solution for $\nu \rightarrow 0$ is a triangular wave which has an exact MPS representation of bond dimension $\chi = 3$, as shown in the following section. The solution for very large ν is a Gaussian (tending towards a uniform function for $\nu \rightarrow \infty$), which has an exponentially convergent MPS approximation [11]. For $0 < \nu < \infty$ it can be shown that the function of Eq. (S28) remains holomorphic in x , and hence regularity arguments similar to those in [12] can be used to prove an exponential convergence of the polynomial approximation of Eq. (S28). In turn, polynomials of degree p sampled on an equidistant grid admit an exact MPS representation [13, Thm. 6] with $\chi \leq p + 1$. Therefore, the $0 < \nu < \infty$ MPS approximation of Eq. (S28) is also exponentially convergent.

Therefore we conclude that the MPS description of solutions of the initial values problem considered here are exponentially convergent in the number of variables used. In other words, an MPS scheme would require exponentially fewer variables than e.g. a standard finite differences scheme. For the sake of concreteness, we illustrate this for the case of $\nu \rightarrow 0$ in the following section.

D. Triangular waves as matrix product states

Let us analytically derive the MPS representation for the $\nu \rightarrow 0$ triangular-wave solution of Eq. (S33). This limit solution is a prototypical shock-wave and (as all shock-waves) it is discontinuous, which slows down the convergence of both polynomial and Fourier approximations. In contrast, an MPS with a bond dimension of just 3 can represent this function exactly, as we now demonstrate in this section.

Definition. A Heaviside vector of length $J \in \mathbb{N}$ with step position $j \in \mathbb{Z}$ is defined element-wise as

$$\theta_q^j := \begin{cases} 1, & q \leq j, \\ 0, & \text{otherwise,} \end{cases} \quad (\text{S43})$$

for $q \in \{0, \dots, J-1\}$.

Definition. A unit vector of length $J \in \mathbb{N}$ at position $j \in \mathbb{Z}$ is defined element-wise as

$$e_q^j := \begin{cases} 1, & q = j, \\ 0, & \text{otherwise,} \end{cases} \quad (\text{S44})$$

for $q \in \{0, \dots, J-1\}$.

Lemma. Let $j = (b_1, \dots, b_N)_2$ and $q = (\omega_1, \dots, \omega_N)_2$, with $b_n, \omega_n \in \{0, 1\}$. Then the j -th Heaviside vector of length $J = 2^N$ can be written as the MPS

$$\theta_{\omega_1 \dots \omega_N}^{b_1 \dots b_N} = T^{\omega_1} \dots T^{\omega_N} \quad (\text{S45})$$

with bond dimensions $d(n) = 2$, $n = 1, \dots, N-1$, where

$$T^{\omega_1} = \begin{bmatrix} e_{\omega_1}^{b_1} & \theta_{\omega_1}^{b_1-1} \end{bmatrix}, \quad (\text{S46})$$

$$T^{\omega_n} = \begin{bmatrix} e_{\omega_n}^{b_n} & \theta_{\omega_n}^{b_n-1} \\ 0 & 1 \end{bmatrix}, \quad (\text{S47})$$

for $n = 2, \dots, N-1$, and

$$T^{\omega_N} = \begin{bmatrix} \theta_{\omega_N}^{b_N} \\ 1 \end{bmatrix}. \quad (\text{S48})$$

Proof. Consider two indices first, and prove that $\theta_{\omega_1\omega_2}^{b_1,b_2} = \theta_{\omega_1}^{b_1-1} + \theta_{\omega_2}^{b_2} e_{\omega_1}^{b_1}$.

- If $\omega_1 > b_1$, we obtain 0, as expected from (S43).
- If $\omega_1 = b_1$, we get $\theta_{\omega_2}^{b_2}$. This becomes 0 when $\omega_2 > b_2$, and 1 when $\omega_2 \leq b_2$, which, together with $\omega_1 = b_1$, gives $q = 2\omega_1 + \omega_2 \leq 2b_1 + b_2 = j$, as expected.
- If $\omega_1 \leq b_1 - 1$, we are left with $\theta_{\omega_1}^{b_1-1} = 1$, but in this case we also have that $q = 2\omega_1 + \omega_2 < 2b_1 + b_2 = j$.

All cases are thus in agreement with (S43). Multiplying the last two factors of the Heaviside MPS gives

$$T^{\omega_{N-1}} T^{\omega_N} = \begin{bmatrix} \theta_{\omega_{N-1}}^{b_{N-1}-1} + \theta_{\omega_N}^{b_N} e_{\omega_{N-1}}^{b_{N-1}} \\ 1 \end{bmatrix} = \begin{bmatrix} \theta_{\omega_{N-1}, \omega_N}^{b_{N-1}, b_N} \\ 1 \end{bmatrix}. \quad (\text{S49})$$

Similarly, assuming that

$$T^{\omega_n} \dots T^{\omega_N} = \begin{bmatrix} \theta_{\omega_n \dots \omega_N}^{b_n \dots b_N} \\ 1 \end{bmatrix}, \quad (\text{S50})$$

gives the induction step for $T^{\omega_{n-1}} \dots T^{\omega_N}$, and eventually, since T^{ω_1} is just one row,

$$T^{\omega_1} \dots T^{\omega_N} = \theta_{\omega_1 \dots \omega_N}^{b_1 \dots b_N} \quad (\text{S51})$$

as expected.

Definition. A vector whose elements are $x_q = q = (\omega_1, \dots, \omega_N)_2$ can be expressed by the MPS

$$X_q \equiv X_{\omega_1 \dots \omega_N} = \begin{bmatrix} 1 & 2^{N-1} \omega_1 \end{bmatrix} \dots \begin{bmatrix} 1 & 2^{N-n} \omega_n \\ 0 & 1 \end{bmatrix} \dots \begin{bmatrix} \omega_N \\ 1 \end{bmatrix} \quad (\text{S52})$$

of bond dimension 2, per [13].

Theorem. A triangular wave vector with elements

$$w_{\omega_1 \dots \omega_N}^{b_1 \dots b_N} := X_{\omega_1 \dots \omega_N} \cdot \theta_{\omega_1 \dots \omega_N}^{b_1 \dots b_N} \quad (\text{S53})$$

can be written as an MPS of bond dimension 3.

Proof. Multiplying the MPS $X_{q_1 \dots q_N}$ with $\theta_{q_1 \dots q_N}^{b_1 \dots b_N}$ tensor by tensor will result in an MPS $\hat{W}^{\omega_1} \dots \hat{W}^{\omega_N}$ of bond dimension 4. However, this decomposition is redundant. For example, the first factor reads

$$\hat{W}^{\omega_1} = \begin{bmatrix} e_{\omega_1}^{b_1} & e_{\omega_1}^{b_1} 2^{N-1} \omega_1 & \theta_{\omega_1}^{b_1-1} & \theta_{\omega_1}^{b_1-1} 2^{N-1} \omega_1 \end{bmatrix}, \quad (\text{S54})$$

albeit $e_{\omega_1}^{b_1} \omega_1 = b_1 e_{\omega_1}^{b_1}$ and this means that

$$\hat{W}^{\omega_1} = \underbrace{\begin{bmatrix} e_{\omega_1}^{b_1} & \theta_{\omega_1}^{b_1-1} & \theta_{\omega_1}^{b_1-1} 2^{N-1} \omega_1 \end{bmatrix}}_{W^{\omega_1}} \underbrace{\begin{bmatrix} 1 & b_1 2^{N-1} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{R^1}. \quad (\text{S55})$$

W^{ω_1} (with bond dimension $d(1) = 3$) can be considered the first non-redundant MPS tensor of $w_{\omega_1 \dots \omega_N}^{b_1 \dots b_N}$. Multiplying R^1 with \hat{W}^{ω_2} will continue the reduction and produce W^{ω_2} . Assuming that $W^{\omega_1}, \dots, W^{\omega_{n-1}}$ have already been obtained, the next step gives

$$R^{n-1} \hat{W}^{\omega_n} := \begin{bmatrix} 1 & c_{n-1} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} e_{\omega_n}^{b_n} & 2^{N-n} \omega_n e_{\omega_n}^{b_n} & \theta_{\omega_n}^{b_n-1} & 2^{N-n} \omega_n \theta_{\omega_n}^{b_n-1} \\ 0 & e_{\omega_n}^{b_n} & 0 & \theta_{\omega_n}^{b_n-1} \\ 0 & 0 & 1 & 2^{N-n} \omega_n \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (\text{S56})$$

where c_{n-1} is a scalar, with $c_1 = b_1 2^{N-1}$. This gives

$$R^{n-1} \hat{W}^{\omega_n} = \begin{bmatrix} e_{\omega_n}^{b_n} & (2^{N-n} b_n + c_{n-1}) e_{\omega_n}^{b_n} & \theta_{\omega_n}^{b_n-1} & (2^{N-n} \omega_n + c_{n-1}) \theta_{\omega_n}^{b_n-1} \\ 0 & 0 & 1 & 2^{N-n} \omega_n \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (\text{S57})$$

Notice that the first two columns are linearly dependent, allowing us to rewrite the above expression into

$$R^{n-1} \hat{W}^{\omega_n} = \underbrace{\begin{bmatrix} e_{\omega_n}^{b_n} & \theta_{\omega_n}^{b_n-1} & (2^{N-n} \omega_n + c_{n-1}) \theta_{\omega_n}^{b_n-1} \\ 0 & 1 & 2^{N-n} \omega_n \\ 0 & 0 & 1 \end{bmatrix}}_{W^{\omega_n}} \underbrace{\begin{bmatrix} 1 & 2^{N-n} b_n + c_{n-1} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{R^n}. \quad (\text{S58})$$

Since R^n is identical to R^{n-1} except for the element $c_n = 2^{N-n} b_n + c_{n-1}$ at position $(1, 2)$, the recursion can continue all the way until \hat{W}^{ω_N} . This will result in the factors W^{ω_n} whose bond dimensions are all $d(n) = 3$, and these factors together form the non-redundant representation of $w_{\omega_1 \dots \omega_N}^{b_1 \dots b_N}$.

4. MATRIX PRODUCT STATE ALGORITHM

In this section we present our MPS algorithm for minimising the cost function of Eq. (8) in the main text in 2-D (i.e. $K = 2$). Generalising this scheme to other dimensions is straightforward. Sec. 4 A explains how Eq. (8) can be considered a variational problem to be solved within the MPS manifold. The minimisation can be done by repeatedly minimising for each of the matrices of the MPS representing the variational flow field. Our procedure for this local minimisation is outlined in Sec. 4 B. Sec. 4 C explains how repeatedly performing many such local minimisations allows us to converge to the global minimum of the cost function. The full computational complexity of our algorithm is derived in Sec. 4 D. The computational complexity is also demonstrated in practice in Sec. 4 E. Finally, in Sec. 4 F we discuss the arithmetic intensity of the MPS algorithm.

A. The variational problem

Let us begin by expanding upon main text Eq. (8). In 2-D, the finite difference del operator is given as

$$\bar{\nabla} = \hat{e}_1 \frac{\Delta}{\Delta x_1} + \hat{e}_2 \frac{\Delta}{\Delta x_2}, \quad (\text{S59})$$

with $\frac{\Delta}{\Delta x_k}$ being the derivative along unit vector \hat{e}_k , and the Laplace operator by

$$\bar{\nabla}^2 = \frac{\Delta^2}{\Delta x_1^2} + \frac{\Delta^2}{\Delta x_2^2}, \quad (\text{S60})$$

with $\frac{\Delta^2}{\Delta x_k^2}$ the second derivative along the k -th direction. We represent both $\frac{\Delta}{\Delta x_k}$ and $\frac{\Delta^2}{\Delta x_k^2}$ using eight-order central finite difference stencils [14] in matrix product operator (MPO) form. For details on how the method of finite differences can be implemented in MPO form, see e.g. [15, p. 591] or [16, p. 22]. Further, note that the main text's variational field, $\mathbf{V}^*(\mathbf{r}_q) = \hat{e}_1 u_1^*(\mathbf{r}_q) + \hat{e}_2 u_2^*(\mathbf{r}_q)$, along with the previous field, $\mathbf{V}(\mathbf{r}_q) = \hat{e}_1 u_1(\mathbf{r}_q) + \hat{e}_2 u_2(\mathbf{r}_q)$, both lie within the MPS manifold \mathcal{M} restricted at bond dimension χ (see Eq. (S15)). For the sake of convenience, we employ linear algebra notation in the rest of this section. We introduce the vector $\mathbf{u}_i = \text{vec}(u_i(\mathbf{r}_q))$ populated by the values of u_i at all grid points \mathbf{r}_q , and similarly for $\mathbf{u}_i^* = \text{vec}(u_i^*(\mathbf{r}_q))$, with $\mathbf{u}_i, \mathbf{u}_i^* \in \mathbb{R}^{2^{NK} \times 1}$. Using this, we rewrite Eq. (8) into

$$\begin{aligned} \Theta(\mathbf{V}^*) = & \sum_{i,j=1}^2 \left\{ \mu \left(\frac{\Delta \mathbf{u}_i^*}{\Delta x_i} \right)^t \frac{\Delta \mathbf{u}_j^*}{\Delta x_j} \right\} + \sum_{i=1}^2 \left\{ \frac{(\mathbf{u}_i^*)^t \mathbf{u}_i^*}{\Delta t^2} + \frac{(\mathbf{u}_i^*)^t}{\Delta t} \left(\frac{-\mathbf{u}_i}{\Delta t} + \sum_{j=1}^2 \left\{ \mathbf{u}_j \frac{\Delta \mathbf{u}_i}{\Delta x_j} - \nu \frac{\Delta^2 \mathbf{u}_i}{\Delta x_j^2} \right\} \right) \right. \\ & \left. + \left(\frac{-\mathbf{u}_i}{\Delta t} + \sum_{j=1}^2 \left\{ \mathbf{u}_j \frac{\Delta \mathbf{u}_i}{\Delta x_j} - \nu \frac{\Delta^2 \mathbf{u}_i}{\Delta x_j^2} \right\} \right)^t \frac{\mathbf{u}_i^*}{\Delta t} \right\} + [\dots], \end{aligned} \quad (\text{S61})$$

with [...] representing the remaining part of Θ which is independent of \mathbf{u}_i^* (it will vanish after differentiation), and $(\cdot)^t$ being the transpose. For the sake of simplicity, the nonlinear term is here represented in convective form, albeit it should be put into skew-symmetric form during actual numerical simulations [17].

Let us now write out \mathbf{V}^* . Assuming \mathbf{V}^* is put into mixed canonical form with canonical centre at site n , it can be written as

$$\mathbf{V}^*(\mathbf{r}_q) = \sum_{i=1}^2 \hat{e}_i \hat{A}_i^{*\omega_1} \hat{A}_i^{*\omega_2} \dots \hat{A}_i^{*\omega_{n-1}} C_i^{*\omega_n} \hat{B}_i^{*\omega_{n+1}} \hat{B}_i^{*\omega_{n+2}} \dots \hat{B}_i^{*\omega_N}, \quad (\text{S62})$$

with the left unitary $\hat{A}_i^{*\omega_n}$ and right unitary $\hat{B}_i^{*\omega_n}$ matrices satisfying orthonormality conditions like in Eq. (S9). For ease of notation, we now define row and column vectors

$$\begin{aligned} \Phi_i^{*\omega_1 \omega_2 \dots \omega_{n-1}} &= \hat{A}_i^{*\omega_1} \hat{A}_i^{*\omega_2} \dots \hat{A}_i^{*\omega_{n-1}} \in \mathbb{R}^{1 \times d(n-1)}, \\ \Psi_i^{*\omega_{n+1} \omega_{n+2} \dots \omega_N} &= \hat{B}_i^{*\omega_{n+1}} \hat{B}_i^{*\omega_{n+2}} \dots \hat{B}_i^{*\omega_N} \in \mathbb{R}^{d(n) \times 1}, \end{aligned} \quad (\text{S63})$$

as well as matrices $\Phi_i^{*n} \in \mathbb{R}^{2^{K(n-1)} \times d(n-1)}$ and $\Psi_i^{*n} \in \mathbb{R}^{d(n) \times 2^{K(N-n)}}$, obtained by stacking the vectors $\Phi_i^{*\omega_1\omega_2\ldots\omega_{n-1}}$ and $\Psi_i^{*\omega_{n+1}\omega_{n+2}\ldots\omega_N}$ corresponding to all values of $\omega_1, \ldots, \omega_N$. Note that the unitarity of $\hat{A}_i^{*\omega_n}$ and $\hat{B}_i^{*\omega_n}$ leads also to Φ_i^{*n} and Ψ_i^{*n} being unitary:

$$\begin{aligned} (\Phi_i^{*n})^t \Phi_i^{*n} &= \mathbb{1}, \\ \Psi_i^{*n} (\Psi_i^{*n})^t &= \mathbb{1}. \end{aligned} \quad (\text{S64})$$

Moreover, let us define a vector $\mathbf{c}_i^{*n} = \text{vec}(C_i^{*\omega_n})$ of all values of $C_i^{*\omega_n}$. Following a straightforward calculation, we can establish a linear map representation of the MPS:

$$\mathbf{u}_i^* = U_i^{*n} \mathbf{c}_i^{*n}, \quad \text{where} \quad U_i^{*n} = \Phi_i^{*n} \otimes \mathbb{1} \otimes (\Psi_i^{*n})^t, \quad (\text{S65})$$

which holds for all n . Lastly, note that $\frac{\Delta}{\Delta x_i}$ is a skew-symmetric operator, $\left(\frac{\Delta}{\Delta x_i}\right)^t = -\frac{\Delta}{\Delta x_i}$.

The above definitions allow us to rewrite Eq. (S61) into

$$\begin{aligned} \Theta(\mathbf{V}^*) &= \sum_{i,j=1}^2 \left\{ -\mu (\mathbf{c}_i^{*n})^t (U_i^{*n})^t \frac{\Delta}{\Delta x_i} \frac{\Delta}{\Delta x_j} U_j^{*n} \mathbf{c}_j^{*n} \right\} \\ &+ \sum_{i=1}^2 \left\{ \frac{(\mathbf{c}_i^{*n})^t (U_i^{*n})^t U_i^{*n} \mathbf{c}_i^{*n}}{\Delta t} + \frac{(\mathbf{c}_i^{*n})^t (U_i^{*n})^t}{\Delta t} \left(\frac{-\mathbf{u}_i}{\Delta t} + \sum_{j=1}^2 \left\{ \mathbf{u}_j \frac{\Delta \mathbf{u}_i}{\Delta x_j} - \nu \frac{\Delta^2 \mathbf{u}_i}{\Delta x_j^2} \right\} \right) \right. \\ &\left. \left(\frac{-\mathbf{u}_i}{\Delta t} + \sum_{j=1}^2 \left\{ \mathbf{u}_j \frac{\Delta \mathbf{u}_i}{\Delta x_j} - \nu \frac{\Delta^2 \mathbf{u}_i}{\Delta x_j^2} \right\} \right)^t \frac{U_i^{*n} \mathbf{c}_i^{*n}}{\Delta t} \right\} + [\dots], \end{aligned} \quad (\text{S66})$$

with the canonical centre of \mathbf{u}_i^* and its transpose set at site n . Now, the minimum of $\Theta(\mathbf{V}^*)$ is found at the stationary point where the gradient of Θ with regards to its variational variables vanishes. Here the variational variables in question are the matrices of $\mathbf{V}^*(\mathbf{r}_q)$. In other words, we require that

$$\frac{\partial \Theta(\mathbf{V}^*)}{\partial C_k^{*\omega_n}} = 0 \quad (\text{S67})$$

simultaneously for all n and k . The solution to Eq. (S67) is given by

$$(U_k^{*n})^t \left(\sum_{j=1}^2 \left\{ -\mu \Delta t \frac{\Delta}{\Delta x_k} \frac{\Delta}{\Delta x_j} U_j^{*n} \mathbf{c}_j^{*n} \right\} + \frac{U_k^{*n} \mathbf{c}_k^{*n}}{\Delta t} - \frac{\mathbf{u}_k}{\Delta t} + \sum_{j=1}^2 \left\{ \mathbf{u}_j \frac{\Delta \mathbf{u}_k}{\Delta x_j} - \nu \frac{\Delta^2 \mathbf{u}_k}{\Delta x_j^2} \right\} \right), \quad (\text{S68})$$

which can be rearranged into

$$\mathbf{c}_k^{*n} - (U_k^{*n})^t \sum_{j=1}^2 \left\{ \mu \Delta t^2 \frac{\Delta}{\Delta x_k} \frac{\Delta}{\Delta x_j} U_j^{*n} \mathbf{c}_j^{*n} \right\} = (U_k^{*n})^t \left(\mathbf{u}_k - \Delta t \sum_{j=1}^2 \left\{ \mathbf{u}_j \frac{\Delta \mathbf{u}_k}{\Delta x_j} - \nu \frac{\Delta^2 \mathbf{u}_k}{\Delta x_j^2} \right\} \right), \quad (\text{S69})$$

after using Eqs. (S64) and (S65) to set $(U_k^{*n})^t U_k^{*n} \mathbf{c}_k^{*n} = \mathbb{1} \mathbf{c}_k^{*n} = \mathbf{c}_k^{*n}$.

B. Local minimisation

Solving Eq. (S69) will locally minimise $\Theta(\mathbf{V}^*)$ with respect to the n -th tensor of the k -th flow component. We define

$$\begin{aligned} H_{kj} &= (U_k^{*n})^t \frac{\Delta}{\Delta x_k} \frac{\Delta}{\Delta x_j} U_j^{*n}, \\ \boldsymbol{\alpha}_k &= \mathbf{c}_k^{*n}, \\ \boldsymbol{\beta}_k &= (U_k^{*n})^t \left(\mathbf{u}_k - \Delta t \sum_{j=1}^2 \left\{ \mathbf{u}_j \frac{\Delta \mathbf{u}_k}{\Delta x_j} - \nu \frac{\Delta^2 \mathbf{u}_k}{\Delta x_j^2} \right\} \right), \end{aligned} \quad (\text{S70})$$

and collect these into vectors $\alpha = \text{vec}(\alpha_k), \beta = \text{vec}(\beta_k)$ while constructing the matrix H from elements H_{kj} . Inserting α, β and H into Eq. (S69) gives a set of linear equations for α

$$(\mathbb{1} - \mu \Delta t^2 H) \alpha = \beta, \quad (\text{S71})$$

with $-H$ being a positive semi-definite matrix. This can be seen by considering

$$\alpha^t H \alpha = \sum_{i,j=1}^2 (\mathbf{u}_i^*)^t \frac{\Delta}{\Delta x_i} \frac{\Delta}{\Delta x_j} \mathbf{u}_j^* = - \sum_{i,j=1}^2 \left(\frac{\Delta \mathbf{u}_i^*}{\Delta x_i} \right)^t \frac{\Delta \mathbf{u}_j^*}{\Delta x_j}. \quad (\text{S72})$$

Because $\sum_{i,j=1}^2 \left(\frac{\Delta \mathbf{u}_i^*}{\Delta x_i} \right)^t \frac{\Delta \mathbf{u}_j^*}{\Delta x_j} = \|\bar{\nabla} \cdot \mathbf{V}\|_2^2 \geq 0$, H must be negative semi-definite, making $(\mathbb{1} - \mu \Delta t^2 H)$ as a whole positive definite (recall that $\Delta t, \mu > 0$). Furthermore, it can be shown that constructing H costs $\mathcal{O}(N\chi^4)$, while computing its action on α costs just $\mathcal{O}(N\chi^3)$. We therefore employed the iterative method of conjugate gradient descent (CGD) [18] to solve the linear problem of Eq. (S71).

C. Global minimisation

The global minimum of Θ can be obtained by adapting well-established techniques for calculating the ground state energy of quantum many-body systems. We start with \mathbf{V}^* in right-canonical form and minimise Θ with respect to matrix $n = 1$ using Eq. (S71). Similarly to the density matrix renormalisation group (DMRG) algorithm [4], we subsequently perform a QR decomposition on site $n = 1$ and shift the canonical centre to site $n = 2$, and optimise Θ with respect to $n = 2$. We iterate this procedure and sweep through the sites of \mathbf{V}^* until convergence is achieved.

D. Theoretical computational scaling

The computational complexity of our algorithm for each individual time-step is a product of the computational cost of the global sweeps multiplied with that of the local minimisations. Let \bar{m}_1 be the average number of sweeps required for the minimisation of Θ to converge, and q be the cost associated with shifting the canonical centre as described in Sec. 4 C, and c the computational cost of a local minimisation. Then the scaling will be $\mathcal{O}[\bar{m}_1 N(q + c)]$ because the cost of the sweeps scales with the number of matrices swept, which is N .

Shifting the canonical centre rightwards from matrix n to $n + 1$ (or leftwards, to $n - 1$) is done in two parts. First, a QR decomposition is performed, followed by a tensor contraction between the upper triangular matrix and the next site. It is straightforward to show [4] that the cost of both goes as $\mathcal{O}(\chi^3)$, which in turn implies $q \sim \mathcal{O}(\chi^3)$.

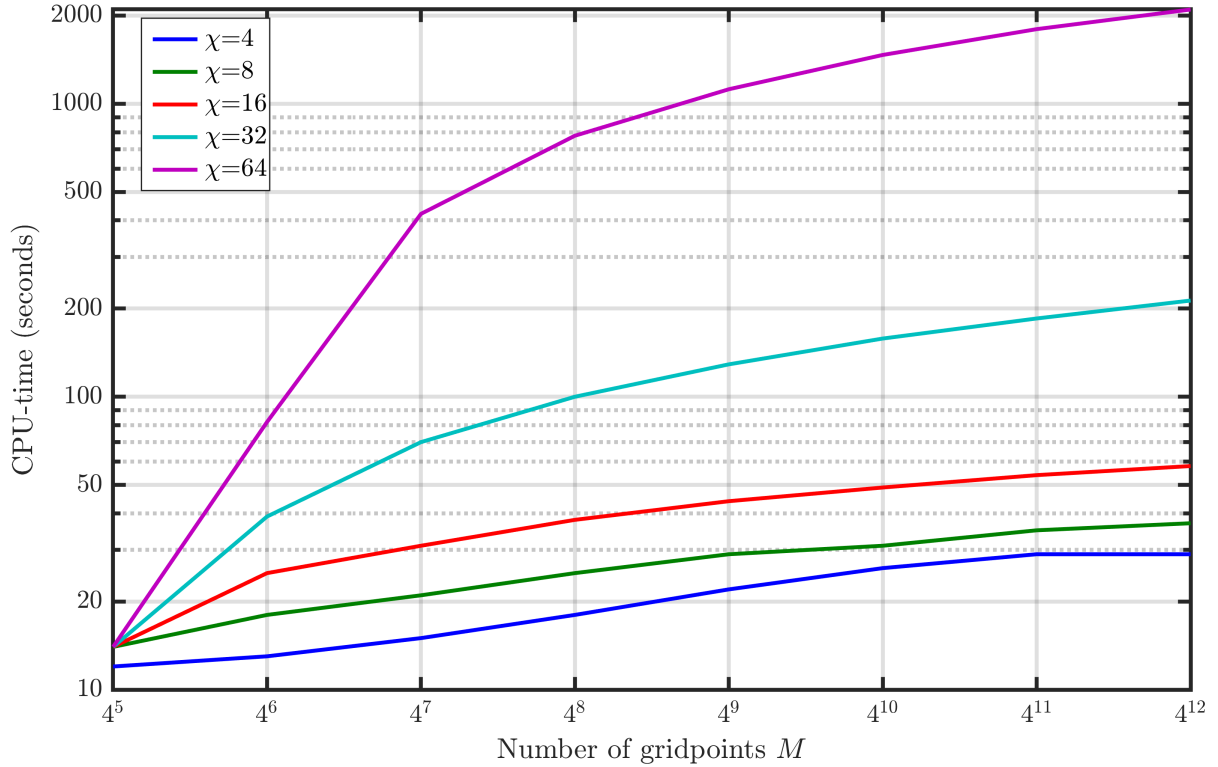
The computational cost of the local minimisation equals the cost needed to first explicitly calculate β in Eq. (S70) plus the subsequent cost associated with the CGD iterations needed to solve Eq. (S71). Regarding the latter, let \bar{m}_2 be the mean number of iterations CGD requires to converge to the solution of Eq. (S71) within the desired precision. The cost of these iterations is then on average a product of \bar{m}_2 and the cost of computing the action of H on α . The last operation can be executed as a tensor contraction where the matrix H is never explicitly formulated, but, instead, the tensor network of $H\alpha$ is contracted using standard tensor contraction techniques [5] at $\mathcal{O}(\chi^3)$ cost. Calculating β is however more expensive due to each β_i containing the nonlinear $\sum_{j=1}^2 \mathbf{u}_j \frac{\Delta \mathbf{u}_i}{\Delta x_j}$. We (variationally) construct it using exact Hadamard products [15, p. 593], which involves tensor contractions scaling as $\mathcal{O}(\chi^4)$. This makes the cost of each local minimisation step go as $c \sim \chi^4 + \bar{m}_2 \chi^3$.

The above implies the computational complexity of each time-step to be

$$\mathcal{O}[\bar{m}_1 N(q + c)] = \mathcal{O}[\bar{m}_1 N(\chi^4 + \chi^3(\text{Const} + \bar{m}_2))]. \quad (\text{S73})$$

The numerical precision is controlled by \bar{m}_1 and \bar{m}_2 . However, in our experience, the number of sweeps or CGD iterations required to achieve a given precision does not change with increasing system size. Dropping these prefactors along with the non-dominant terms leads to the total computational complexity of

$$\mathcal{O}[\bar{m}_1 N(\chi^4 + \chi^3(\text{Const} + \bar{m}_2))] = \mathcal{O}(N\chi^4) \quad (\text{S74})$$



Supplementary Figure 4: **Demonstration of MPS computational scaling.** 10 time-steps of the 2-D Navier-Stokes equations were simulated with our MPS algorithm. The CPU time (in seconds) required to perform these 10 time-steps is plotted for five values of χ against eight different grid-sizes.

per timestep. This scaling is equivalent to $\mathcal{O}(\chi^4 \log M)$, as the number of grid points M is related with N through $M = 2^{KN}$.

E. Demonstration of computational scaling

Among the most interesting aspects of the MPS algorithm is how the computational complexity of the above Sec. 4D scales linearly with N , i.e., logarithmically with M . To demonstrate, we have plotted in Supp. Fig. 4 the actual CPU time our algorithm requires to perform 10 time-steps simulating the 2-D Navier-Stokes equation at various values of M and χ . The simulations were all carried out on a MacBook Pro (Retina, 15-inch, Mid 2015) running macOS v. 11.6 on a 2,5 GHz Quad-Core i7 CPU with 16 GB 1600 MHz DDR3 ram. Note how all the curves in the figure saturate for sufficiently many gridpoints; this implies the computational performance scales *sub-polynomially* with M in a manner consistent with the computational complexity derived in Eq. (S74).

F. Arithmetic intensity

For completeness we also briefly discuss how memory-efficient our algorithm is. This can be done by studying a quantity known as the arithmetic intensity I . I is the ratio between the amount of arithmetic operations performed (in units of FLOPs) to the required memory traffic (in bytes) of an algorithm. Many algorithms running on modern computing systems are memory bound in the sense that their performance is limited by memory bandwidth rather than computing power. Thus, algorithms with a high value of I can more optimally utilize modern hardware than those with a low arithmetic intensity.

As discussed in Sec. 4D, our algorithm is dominated by the need to calculate the nonlinear term. This in practice boils down to repeatedly performing matrix-matrix multiplications between pairs of non-square matrices of type $(\text{Const} \cdot \chi^2) \times \chi$ and $\chi \times (\text{Const} \cdot \chi)$. Theoretically, the arithmetic intensity of such

operations goes as

$$I \sim \frac{\mathcal{O}(\chi^4)}{\mathcal{O}(\chi^2(\chi + \text{Const}))} = \mathcal{O}(\chi) \quad (\text{S75})$$

for large χ when memory is efficiently utilised. If, however, the memory is *not* efficiently utilised due to shortcomings in either hardware (e.g. inadequate cache size) or software (e.g. failure to use cache blocking), the scaling in Eq. (S75) will not hold.

Fortunately, the problem of matrix-matrix multiplication is among the most important and thoroughly studied problems of numerical linear algebra. Highly optimised hardware (along with the associated software) beyond CPUs exist for such operations, ranging from GPUs [19] to even TPUs [20]. Thus, the dominant operation of our algorithm is characterised by a high I . This in turn makes our algorithm capable of efficiently utilising the power of modern parallelised computing hardware.

5. QUANTUM ALGORITHM

Here we explain how to port our algorithm to a quantum computer. Our approach is based on the formalism of variational quantum algorithms (VQAs) [21] and summarized in Sec. 5 A. We compare this strategy to the alternative quantum algorithmic approaches [22, 23] in Sec. 5 B.

A. Variational quantum algorithm

The main difference between the classical MPS algorithm presented in the main text and a corresponding VQA is that the latter encodes the solution in 2^{KN} probability amplitudes of a quantum state $|u(\phi)\rangle$ of KN qubits. This state is created from a fixed initial state $|0\rangle^{\otimes KN} = |\mathbf{0}\rangle$ by a network of quantum gates $\hat{U}(\phi)$ that are parameterised by classical variational parameters ϕ , i.e. $|u(\phi)\rangle = \hat{U}(\phi)|\mathbf{0}\rangle$. Problem dependent gate operations and measurements are then applied to evaluate a cost function [21, 24].

We sketch these problem dependent operations for the minimization of the cost function Eq. (S61) for the special case of $K = 1$ and $\mu = 0$, i.e. Burgers' equation, as the more general Navier-Stokes case follows straightforwardly. The quantum version of the cost function Eq. (S61) is written as

$$\Theta(|u^*\rangle) = \left\| \frac{|u^*\rangle - |u\rangle}{\Delta t} + (|u\rangle \cdot \bar{\nabla})|u\rangle - \nu \bar{\nabla}^2 |u\rangle \right\|_2^2. \quad (\text{S76})$$

Here $|u^*\rangle = |u(\phi^*)\rangle$ is the trial solution for the current time step and $|u\rangle = |u(\phi)\rangle$ is the solution from the previous time step. Efficient VQA methods for the minimization of Eq. (S76) use gradient-based optimizers [21] which require evaluation of only those scalar products in Eq. (S76) that contain $|u^*\rangle$. The gradient reads

$$\frac{\partial \Theta(|u^*\rangle)}{\partial \phi_k^*} = \frac{\partial}{\partial \phi_k^*} \left(\frac{(\phi_0^*)^2}{\Delta t^2} - \frac{2\phi_0\phi_0^*}{\Delta t} \Re \left\{ \frac{1}{\Delta t} \langle u|u^*\rangle + \langle u|(|u\rangle \cdot \bar{\nabla})^\dagger |u^*\rangle - \nu \langle u|\bar{\nabla}^2 |u^*\rangle \right\} \right) \quad (\text{S77})$$

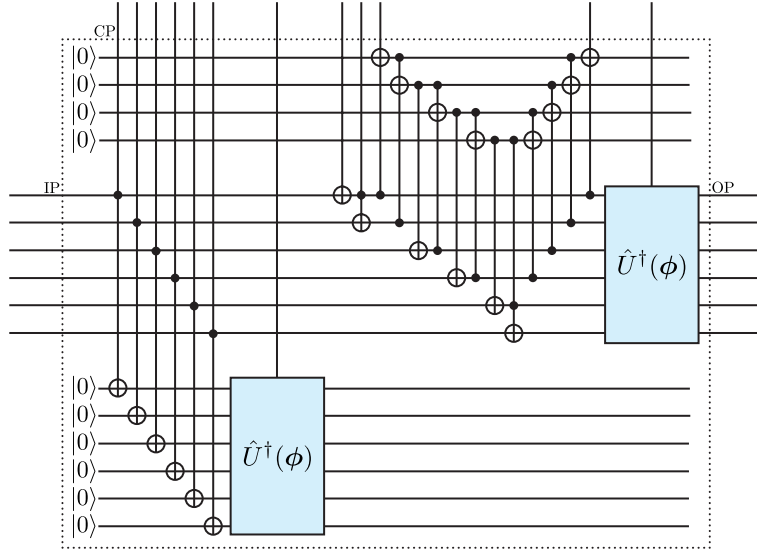
where $\Re\{\cdot\}$ is the real part, $(\cdot)^\dagger$ the adjoint and ϕ_0 and ϕ_0^* keep track of the changing normalization [24]. The quantum circuit representations of all terms in Eq. (S77) are given in [24, Supplementary Sec. III] together with a comprehensive derivation. Importantly, by making use of two copies of $|u\rangle$ we can straightforwardly handle the nonlinear term. The core part of the quantum network for evaluating the nonlinear term is shown in Supp. Fig. 5.

A distinguishing feature of the quantum version of our algorithm is that the variational states encoded in $|u(\phi)\rangle$ are not limited to MPS and can thus be more expressive. Therefore VQA optimization makes use of a variational manifold that is more general than the MPS manifold with the potential to lead to a quantum advantage [24]. Even when remaining within the MPS manifold the VQA approach immediately improves the scaling with bond dimension from $\mathcal{O}(\chi^4)$ to $\mathcal{O}(\chi^2)$ [24].

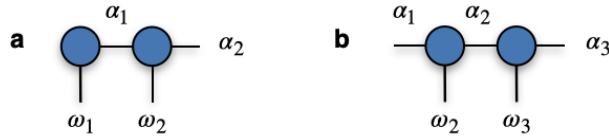
B. Comparison to alternative proposals

The VQA approach in Sec. 5 A and the alternative proposals [22, 23] have in common that they require a number of qubits that scales logarithmically with the number of grid points. That is why all three approaches have the potential to achieve an exponential quantum speedup over some standard classical computational methods.

The main difference between the three strategies is that our approach concentrates on solving general nonlinear problems whereas the approaches [22, 23] focus on being efficient in the number of time steps. The proposal [22] is based on the derivation of the nonlinear Schrödinger equation from the linear Schrödinger equation for quantum many-body systems, i.e. a mean-field approximation, which is accurate only in the limit of weak interactions. The approach [23] uses the well-known technique of Carleman linearization to map a specific set of weakly nonlinear ordinary differential equations with dissipation to a higher-dimensional linear problem. Both methods [22, 23] solve their linear problems using the quantum linear systems algorithm [25–27]. This allows them to be efficient in the number of time steps and leads to mathematically rigorous convergence guarantees that, however, hold only under the strong restrictions of their derivation. In contrast, our approach can be applied to a wide range of nonlinear partial differential equations with arbitrary interaction strengths.



Supplementary Figure 5: **Central part of the quantum circuit for the nonlinear term in Eq. (S77).** Gate $\hat{U}^\dagger(\phi)$ is the adjoint of the quantum network creating the previous solution $|u\rangle$. The variational state $|u^*\rangle$ is connected to the input port IP. The control port CP leads to the ancillary quantum control logic and the output port OP connects to the evaluation quantum logic discussed in detail in [24]. Here the quantum state is realized with $N = 6$ qubits.



Supplementary Figure 6: **Tensor contractions.** Sub-figure (a) illustrates a tensor contraction between an order-2 tensor (left) with an order-3 tensor (right). The contraction is performed along the closed bond α_1 , while the open bonds ω_1, ω_2 and α_2 are not summed over. A similar tensor contraction is shown in (b), except now both the tensors are order-3 and the summed-over bond is α_2 .

6. GRAPHICAL NOTATION

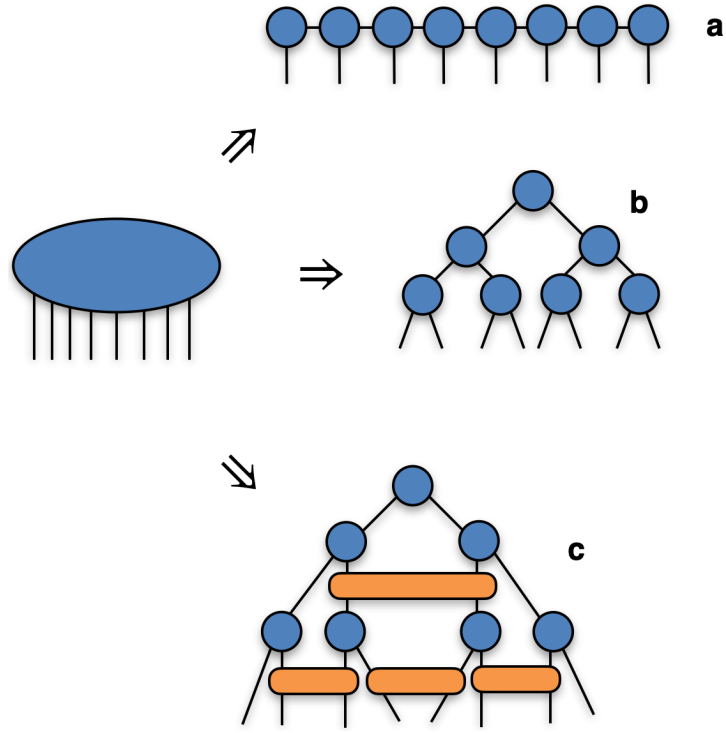
In this section we introduce graphical tensor network notation and use it to sketch MPS, tree tensor networks (TTNs) and the multiscale entanglement renormalisation ansatz (MERA).

Describing tensor networks algebraically, as we did in Sec. 2 and beyond, is unpractical for geometries more complicated than MPS. Such tensor networks are more legibly described *graphically*. To illustrate this graphical notation, consider the MPS in Eq. (S5). The matrix-matrix multiplications there can be written out as

$$v(\mathbf{r}_q, \chi) = \sum_{\{\alpha_n\}=1}^{d(n)} A_{\alpha_1}^{\omega_1} A_{\alpha_1 \alpha_2}^{\omega_2} \cdots A_{\alpha_{N-2} \alpha_{N-1}}^{\omega_{N-1}} A_{\alpha_{N-1}}^{\omega_N}. \quad (\text{S78})$$

This equation emphasises that the matrix-matrix multiplication between A^{ω_n} and $A^{\omega_{n+1}}$ is equivalent to a summation over the internal index α_n , which is equivalent to a tensor contraction. The tensor contraction can be graphically represented using the standard diagrammatic notation of tensor network theory (see e.g. [4, 5]), as we do in Supp. Figs. 6a and 6b for $n = 1$ and $n = 2$, respectively. This notation allows us to *draw* the MPS decomposition of Eq. S5, as shown for $N = 8$ in Supp. Fig. 7a.

Supp. Fig. 7a illustrates that the connectivity of MPS is solely between neighbouring sites. It is possible to connect more distant sites by using alternative tensor network geometries. One such geometry is the TTN [28–30] that is illustrated in Supp. Fig. 7b. While TTNs connect distant sites, they also have fewer direct connections between neighbouring sites than MPS. The MERA network [31] is a generalisation of TTNs that maintains connectivity between nearby sites, and is drawn in Supp. Fig 7c.



Supplementary Figure 7: **Three possible decompositions from one tensor.** In (a), an order-8 tensor has been decomposed into an 8-site MPS where only the nearest neighbours are connected. In (b), the same tensor has been decomposed into a TTN, which connects distant sites. The MERA decomposition is shown in (c). Note both the increased connectivity compared to MPS and TTN, as well as the presence of loops.

-
- [1] S. Chen, R. E. Ecke, G. L. Eyink, M. Rivera, M. Wan, and Z. Xiao, “Physical mechanism of the two-dimensional inverse energy cascade,” *Phys. Rev. Lett.* **96**, 084502 (2006).
 - [2] A. N. Kolmogorov, “The local structure of turbulence in incompressible viscous fluids for very large Reynolds’ numbers,” *Dokl. Akad. Nauk SSSR* **30**, 301–305 (1941).
 - [3] S. Holtz, T. Rohwedder, and R. Schneider, “On manifolds of tensors of fixed TT-rank,” *Numer. Math.* **120**, 701–731 (2012).
 - [4] U. Schollwöck, “The density-matrix renormalization group in the age of matrix product states,” *Ann. Phys.* **326**, 96–192 (2011).
 - [5] R. Orús, “A practical introduction to tensor networks: Matrix product states and projected entangled pair states,” *Ann. Phys.* **349**, 117 – 158 (2014).
 - [6] H. Bateman, “Some recent researches on the motion of fluids,” *Mon. Weather Rev.* **43**, 163 – 170 (1915).
 - [7] J. M. Burgers, “A mathematical model illustrating the theory of turbulence,” *Adv. Appl. Mech.* **1**, 171–199 (1948).
 - [8] E. Hopf, “The partial differential equation $u_t + uu_x = \mu_{xx}$,” *Commun. Pur. Appl. Math.* **3**, 201–230 (1950).
 - [9] J. D. Cole, “On a quasi-linear parabolic equation occurring in aerodynamics,” *Quart. Appl. Math.* **9**, 225–236 (1951).
 - [10] G. B. Whitham, *Linear and Nonlinear Waves* (John Wiley, New York, 1974).
 - [11] S. V. Dolgov, B. N. Khoromskij, and I. V. Oseledets, “Fast solution of multi-dimensional parabolic problems in the tensor train/quantized tensor train-format with initial application to the Fokker-Planck equation,” *SIAM J. Sci. Comput.* **34**, A3016–A3038 (2012).
 - [12] L. Herrmann, C. Schwab, and J. Zech, “Deep neural network expression of posterior expectations in Bayesian PDE inversion,” *Inverse Probl.* **36**, 125011 (2020).
 - [13] I. V. Oseledets, “Constructive representation of functions in low-rank tensor formats,” *Constr. Approx.* **37**, 1–18 (2013).
 - [14] B. Fornberg, “Generation of finite difference formulas on arbitrarily spaced grids,” *Math. Comput.* **51**, 699–706 (1988).
 - [15] M. Lubasch, P. Moinier, and D. Jaksch, “Multigrid renormalization,” *J. Comput. Phys.* **372**, 587–602 (2018).
 - [16] J. J. García-Ripoll, “Quantum-inspired algorithms for multivariate analysis: from interpolation to partial differential equations,” *Quantum* **5**, 431 (2021).
 - [17] Thomas A. Zang, “On the rotation and skew-symmetric forms for incompressible flow simulations,” *Appl. Numer. Math.* **7**, 27–40 (1991).
 - [18] J. Shewchuk, *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain*, Tech. Rep. (Carnegie Mellon University, Pittsburgh, Pennsylvania 15213, United States, 1994).
 - [19] D. Ernst, G. Hager, J. Thies, and G. Wellein, “Performance engineering for real and complex tall & skinny matrix multiplication kernels on gpus,” *Int. J. High Perform. Comput. Appl.* **35**, 5–19 (2021).
 - [20] N. Jouppi, C. Young, N. Patil, and D. Patterson, “Motivation for and evaluation of the first tensor processing unit,” *IEEE Micro* **38**, 10–19 (2018).
 - [21] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, “Variational quantum algorithms,” *Nat. Rev. Phys.* , 1–20 (2021).
 - [22] S. Lloyd, G. De Palma, C. Gokler, B. Kiani, Z.-W. Liu, M. Marvian, F. Tennie, and T. Palmer, “Quantum algorithm for nonlinear differential equations,” Preprint: arXiv:2011.06571 (2020).
 - [23] J.-P. Liu, H. Ø. Kolden, H. K. Krovi, N. F. Loureiro, K. Trivisa, and A. M. Childs, “Efficient quantum algorithm for dissipative nonlinear differential equations,” *Proc. Natl. Acad. Sci. U.S.A.* **118** (2021).
 - [24] M. Lubasch, J. Joo, P. Moinier, M. Kiffner, and D. Jaksch, “Variational quantum algorithms for nonlinear problems,” *Phys. Rev. A* **101**, 010301(R) (2020).
 - [25] A. W. Harrow, A. Hassidim, and S. Lloyd, “Quantum algorithm for linear systems of equations,” *Phys. Rev. Lett.* **103**, 150502 (2009).
 - [26] D. W. Berry, “High-order quantum algorithm for solving linear differential equations,” *J. Phys. A* **47**, 105301 (2014).
 - [27] D. W. Berry, A. M. Childs, A. Ostrander, and G. Wang, “Quantum algorithm for linear differential equations with exponentially improved dependence on precision,” *Commun. Math. Phys.* **356**, 1057–1081 (2017).
 - [28] Y.-Y. Shi, L.-M. Duan, and G. Vidal, “Classical simulation of quantum many-body systems with a tree tensor network,” *Phys. Rev. A* **74**, 022320 (2006).
 - [29] L. Tagliacozzo, G. Evenbly, and G. Vidal, “Simulation of two-dimensional quantum systems using a tree tensor network that exploits the entropic area law,” *Phys. Rev. B* **80**, 235127 (2009).
 - [30] V. Murg, F. Verstraete, Ö. Legeza, and R. M. Noack, “Simulating strongly correlated quantum systems with tree tensor networks,” *Phys. Rev. B* **82**, 205105 (2010).
 - [31] G. Vidal, “Entanglement renormalization,” *Phys. Rev. Lett.* **99**, 220405 (2007).