

**PAPER****OPEN ACCESS****RECEIVED**  
20 January 2025**REVISED**  
21 May 2025**ACCEPTED FOR PUBLICATION**  
10 December 2025**PUBLISHED**  
26 February 2026

Original content from  
this work may be used  
under the terms of the  
[Creative Commons  
Attribution 4.0 licence](https://creativecommons.org/licenses/by/4.0/).

Any further distribution  
of this work must  
maintain attribution to  
the author(s) and the title  
of the work, journal  
citation and DOI.



# Tensor network methods for the Gross–Pitaevskii equation on fine grids

Ryan J J Connor<sup>1,\*</sup> , Callum W Duncan<sup>1,2</sup> and Andrew J Daley<sup>3,1</sup><sup>1</sup> Department of Physics and SUPA, University of Strathclyde, Glasgow G4 0NG, United Kingdom<sup>2</sup> Aegiq Ltd., Cooper Buildings, Arundel Street, Sheffield S1 2NS, United Kingdom<sup>3</sup> Department of Physics, University of Oxford, Clarendon Laboratory, OX1 3PU Oxford, United Kingdom

\* Author to whom any correspondence should be addressed.

**E-mail:** [ryan.connor.2019@uni.strath.ac.uk](mailto:ryan.connor.2019@uni.strath.ac.uk)**Keywords:** tensor network, Gross–Pitaevskii equation, dipolar gases, cold atoms, quantum Fourier transform

## Abstract

The Gross–Pitaevskii equation and its generalisations to dissipative and dipolar gases have been very useful in describing dynamics of cold atomic gases, as well as polaritons and other nonlinear systems. For some of these applications the numerically accessible grid spacing can become a limiting factor, especially in describing turbulent dynamics and short-range effects of dipole-dipole interactions. We explore the application of tensor networks to these systems, where (in analogy to related work in fluid and plasma dynamics), they allow for physically motivated data compression that makes simulations possible on large spatial grids which would be unfeasible with direct numerical simulations. Analysing different non-equilibrium cases involving vortex formation, we find that these methods are particularly efficient, especially in combination with a matrix product operator representation of the quantum Fourier transform, which enables a spectral approach to calculation of both equilibrium states and time-dependent dynamics. The efficiency of these methods has interesting physical implications for the structure in the states that are generated by these dynamics, and provides a path to describe cold gas experiments that are challenging for existing methods.

## 1. Introduction

Tensor networks are an indispensable tool used in the study of quantum many-body physics [1–3], allowing for the study of low lying energy states of many-body Hamiltonians, beginning with the density matrix renormalisation group [4, 5], and allowing for the study of out of equilibrium physics via time evolution [4, 6–8]. Tensor networks can allow for the study of large systems consisting of many particles by systematic data compression in the sense of reducing the number of parameters that must be stored, provided that the amount of entanglement present in the states they represent is appropriately restricted [9–11]. Furthermore, previous work has explored the application of tensor networks to the simulations of continuous quantum systems and beyond for applications to solutions of linear equations and even Poisson like differential equations [12, 13].

Recent work has begun to explore the possibility of applying tensor networks to non-linear partial differential equations (PDEs), such as modelling the motion of fluids and plasmas [14–20]. Numerically simulating non-linear PDEs remains a challenging task for problems of importance in these areas, especially when there is a substantial range of scales relevant to the dynamics. Solving PDEs with direct numerical simulation (DNS) involves discretising the solution space on which the quantity of interest is specified, at the expense of a computational cost that scales with the number of grid points. In contrast, TN approaches allow us to discard correlations which do not contribute to the overall physics, exploiting underlying correlation structures of the problem at hand, and the computational complexity can scale logarithmically with, e.g. the number of grid points, provided there is appropriate structure in the solution (specifically, limits in the types of correlations present).

Non-linearity appears in many physical systems, and this includes a variety of non-linear Schrödinger equations, e.g. in nonlinear optics. The theoretical description of the dynamics of cold gases, both with contact interactions and dipolar interactions are described by the Gross–Pitaevskii equation (GPE) [21–24]. The GPE and its extended versions have been useful for some time in describing experimentally observed dynamics and phenomena such as vortex formation in cold atomic gases. There is still significant ongoing interest in the GPE, especially in regards to quantum turbulence [25, 26], formation of quantised vortices [27–29] and exotic properties in dipolar gases [30–33], with many experimental groups currently involved in exploring such phenomena [27, 34–40].

There remain certain challenges in performing numerical simulations with the GPE however, such as being able to capture vastly different length scales within a simulation. To perform a physically accurate simulation one must be able to resolve the smallest size details present in the dynamics, which for turbulent dynamics, or systems with anisotropic short-range interactions, may be orders of magnitudes smaller than the size of the simulation domain. The memory requirements for performing a simulation over such large separations in length scales becomes a bottleneck, restricting the possible spatial resolution. As such, a natural question remains as to whether tensor networks can be used to provide data compression to overcome these current limitations.

In this work we extend the application of tensor networks, in the form of matrix product states (MPSs), to perform simulations of quantum fluids and turbulence through the GPE. We exploit novel techniques such as the implementation of the quantum Fourier transform [41] in matrix product operators (MPOs) to explore and compare various methods for studying dynamics of a trapped BEC in experimentally relevant settings. The remainder of this paper is structured as follows: In section 1, we outline the details of the GPE, before introducing tensor networks and discussing how one can use them to encode the GPE. We then outline how one can implement the commonly used split-step Fourier method with tensor networks in section 2, which are then illustrated by examples simulations in section 3. We highlight the use of tensor networks for modelling simple Soliton propagation, and for a more challenging test case of vortex formation in 2D and 3D. Finally, in section 4 we demonstrate how a dipolar gas can equally be simulated with tensor networks.

## 2. Simulating the GPE with tensor networks

### 2.1. The GPE

The GPE describes the dynamics of a dilute Bose gas via the classical field  $\Psi(\mathbf{r}; t)$  [21, 22]

$$i\hbar \frac{\partial \Psi(\mathbf{r}; t)}{\partial t} = \left[ -\frac{\hbar^2}{2m} \nabla^2 + V(\mathbf{r}) - \mu + g|\Psi(\mathbf{r}; t)|^2 \right] \Psi(\mathbf{r}; t), \quad (1)$$

with interaction parameter  $g$ , atomic mass  $m$ , chemical potential  $\mu$ , and a trapping potential  $V(\mathbf{r})$ . The non-linear term of the GPE is due to the assumption of a soft  $s$ -wave scattering of the atoms and appears in the typical self-consistent form of mean field equations [22]. The GPE is often a good approximation for the dynamics of a BEC in the weakly interacting regime, and also takes the same form as the non-linear Schrodinger equation [42].

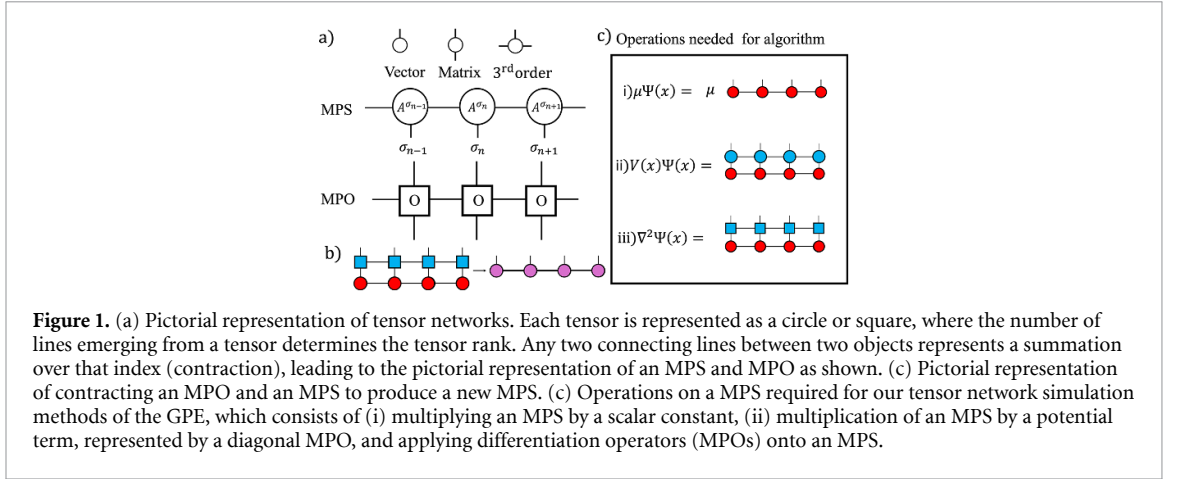
The GPE as outlined in equation (1) does not include any inherent mechanism for damping excitations, which is known to be present in a dilute BEC [43, 44]. We will consider the introduction of a phenomenological damping term to give the modified GPE [45, 46]

$$i\hbar \frac{\partial \Psi}{\partial t} = (1 - i\gamma) \left[ -\frac{\hbar^2}{2m} \nabla^2 + V(\mathbf{r}) - \mu + g|\Psi|^2 \right] \Psi, \quad (2)$$

with  $\gamma$  the dissipation parameter. Note that we have dropped the explicit space and time dependence of the state for ease of reading in equation (2). Throughout this paper we use non-dimensional units, where we set  $\hbar = 1$  and  $m = 1$ .

### 2.2. Representing the state as a tensor network

We will begin by considering the representation of  $\Psi(\mathbf{r}; t)$  as a tensor network. In this work we will focus on the case of encoding the state into a MPS (which is a specific 1D tensor network, and sometimes referred to as a tensor train). We note that other geometries could be considered for encoding the problem onto tensor networks and might better representation correlations in some scenarios (though often at the cost of how the algorithm scales with the size of the tensor network). This has been investigated, e.g. for PDEs from Plasma dynamics [18].



To encode  $\Psi$  onto an MPS we need to discretise the  $d$ -dimensional state  $\Psi(\mathbf{r}; t)$  into a grid of  $2^{dN}$  gridpoints  $\mathbf{r}_q$ . This is equivalent to that for direct numerical approaches that do not utilise the tensor network formalism. We will take the gridpoint index  $q^i$  to be defined via  $N$  bits

$$q^i = (\sigma_1^i, \sigma_2^i, \dots, \sigma_N^i), \quad (3)$$

with  $i$  the dimension ( $i \in \{1, d\}$ ) and  $\sigma_j^i \in \{0, 1\}$ . We can then write the discretised  $\Psi$  as a tensor

$$\Psi(\mathbf{r}_q; t) = A_{\prod_{i=1}^d \sigma_i^i}, \quad (4)$$

with  $A_{\prod_{i=1}^d \sigma_i^i}$  being a  $dN$ -order tensor. We can then apply the approach of singular value decomposition (SVD) on the  $dN$ -order tensor to reduce it to the combination of two smaller tensors, then by repeating this process  $dN - 1$  times we will arrive at the MPS representation of the tensor

$$A_{\prod_{i=1}^d \sigma_i^i} = \sum_{\alpha_1}^{D_1} \sum_{\alpha_2}^{D_2} \dots \sum_{\alpha_{dN-1}}^{D_{dN-1}} A_{\alpha_1}^{\tilde{\sigma}_1} A_{\alpha_1, \alpha_2}^{\tilde{\sigma}_2} \dots A_{\alpha_{dN-1}}^{\tilde{\sigma}_{dN}}, \quad (5)$$

where  $\alpha_j$  is known as a bond index and has the same dimensionality as the number of singular values obtained from the SVD. Note that we have introduced the following to simplify the notation of equation (5)

$$\tilde{\sigma}_n = \begin{cases} \sigma_n^1, & 1 \leq n \leq N, \\ \sigma_{n-N}^2, & N < n \leq 2N, \\ \text{etc.} & \end{cases} \quad (6)$$

i.e. the first  $N\tilde{\sigma}_n$  represent the first spatial dimension, the consecutive  $N$  then represents the second dimension, and so on. It should be noted that there are different ways to encode a high dimensional problem into MPS form [14, 16].

There exists a complementary pictorial representation of an MPS, illustrated in figure 1(a). Each  $A^{\sigma_n}$  tensor is represented by a circle, with corresponding indices shown as lines emerging from the body. Any two connecting lines then indicate a summation over the shared index, known as a contraction, i.e. matrix multiplication. This diagrammatic representation will become useful when we come to discuss the full algorithm implemented.

### 2.3. Truncation of a MPS

The number of singular values in between each matrix is referred to as the bond dimension, and it directly gives the size of the corresponding dimension of the matrices on either side. The maximum bond dimension after performing the SVDs for equation (5) exactly is given by  $D_n = \min(2^n, 2^{N-n})$ . However, many of the singular values can be effectively zero and can be discarded allowing for a controlled truncation of the bond dimension without a significant loss in accuracy of the numerical approach. To do so, we perform an SVD at each bond within the MPS to generate singular values  $S_i$ , which are normalised such that  $\sum_i S_i^2 = 1$ . We then retain the  $\chi$  largest singular values, whilst discarding the rest, thus performing a truncation of our state. Unless otherwise stated, we set a cutoff to determine  $\chi$ , where we

discard any singular value which when squared is below this cutoff, along with their corresponding elements in the left and right matrices of the SVD. Using a cutoff for truncation is often advantageous as our bond dimension can dynamically increase or decrease during a simulation, and is useful in ensuring convergence of our results.

If an MPS provides a good representation of  $\Psi$ , i.e. we can represent  $\Psi$  accurately with a small bond dimension, depends on the correlations in the data. By restricting the bond dimension in the MPS at a given bond  $n$ , we are effectively limiting the possible correlations between the two halves of the MPS on either side of bond  $n$ . Hence, the trick to the efficient encoding of a given state in an MPS is to find a representation such that correlations across all splits of the system at each given bond are minimal. We can then achieve physically motivated data compression through this truncation of the singular values. Additionally, by decomposing  $\Psi$  into a tensor network we can infer information about the structure of correlations which are present in our state.

## 2.4. Representing the GPE as a MPO

Just as one can encode classical functions into MPS form, one can express an operator in the form of a MPO with a limited bond dimension. An MPO acts on an MPS to produce another MPS, and is thus represented pictorially as shown in figure 1(a). To apply an MPO to an MPS, one contracts with the MPS as shown in figure 1(b). Note that after the application of an MPO, the bond dimension of the resultant MPS will grow and the truncation of the bond dimension will be applied if applicable.

The application of the GPE given in equation (2) requires the following operations on the MPS; (i) multiplication by a scalar, (ii) multiplication by a scalar function, (iii) the implementation of the kinetic term, (iv) squaring the MPS, and (v) the time-stepping procedure. These steps are illustrated in figure 1(c). The implementation of (i) is straightforward, with (ii) and (iv) being achieved by promoting the MPS to a diagonal MPO

$$\Psi_{\text{MPO}} = \sum_{\sigma} A^{\tilde{\sigma}_1} \delta_{\sigma_1}^{\tilde{\sigma}_1} A^{\tilde{\sigma}_2} \delta_{\sigma_2}^{\tilde{\sigma}_2} \dots A^{\tilde{\sigma}_{dN}} \delta_{\sigma_{dN}}^{\tilde{\sigma}_{dN}} |\tilde{\sigma}\rangle \langle \tilde{\sigma}'|, \quad (7)$$

where  $A^{\tilde{\sigma}_n}$  are the MPS tensors and  $\delta_b^a$  is the kronecker delta. Contraction of such a diagonal MPO onto an MPS results in performing an element wise multiplication of the encoded functions.

There are two main options for the implementation of the kinetic term in the GPE, more specifically, the implementation of the Laplace operator. The most direct approach is to encode the Laplace operator as an MPO [47]. This can be written out by first defining effective raising and lowering operators

$$s_i^- = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad s_i^+ = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, \quad I_i = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad (8)$$

at the  $i$ th site of the MPS. We can then write the central finite difference in terms of these operators on the MPS as

$$\begin{aligned} \hat{\partial}_x^2 = \frac{\partial^2}{\partial x^2} &= \frac{\hat{S}^+ + \hat{S}^- - 2\hat{I}}{\Delta x^2} = \frac{1}{\Delta x^2} \begin{pmatrix} s_1^+ & s_1^- & I_1 \end{pmatrix} \begin{pmatrix} s_2^- & 0 & 0 \\ 0 & s_2^+ & 0 \\ s_2^+ & s_2^- & I_2 \end{pmatrix} \begin{pmatrix} s_3^- & 0 & 0 \\ 0 & s_3^+ & 0 \\ s_3^+ & s_3^- & I_3 \end{pmatrix} \\ &\dots \begin{pmatrix} s_{N-1}^- & 0 & 0 \\ 0 & s_{N-1}^+ & 0 \\ s_{N-1}^+ & s_{N-1}^- & I_{N-1} \end{pmatrix} \begin{pmatrix} s_N^- \\ s_N^+ \\ s_N^+ + s_N^- - 2I_N \end{pmatrix}. \end{aligned} \quad (9)$$

One can identically encode higher-order finite-difference approximations, for example an 8th order central finite difference approximation [48]. Another option is to utilise so-called split-step methods, a common approach for the simulation of the GPE, and utilise the quantum Fourier transform which we will discuss in the next section.

## 3. The split-step method with tensor networks

An alternative to the finite difference representations of  $\nabla^2$  described above is to use the Fourier transform such that the differential terms become diagonal in momentum space. We can define the (discrete) Fourier transform  $\mathcal{F}$  as

$$\tilde{f}(\mathbf{k}) = \mathcal{F}[f(\mathbf{r})] = \frac{1}{\sqrt{M}} \sum_{\mathbf{r}} e^{i\mathbf{k}\cdot\mathbf{r}} f(\mathbf{r}), \quad (10)$$

where  $M$  is the total number of data points. We can therefore write the differentiation operator along  $x$  as

$$\hat{\partial}_x = -i \mathcal{F}^{-1} \sum_{\mathbf{k}} k_x |\mathbf{k}\rangle \langle \mathbf{k}| \mathcal{F}, \quad (11)$$

with  $k_x$  being the  $x$ -component of the, in general, three-dimensional quasimomentum  $\mathbf{k} = (k_x, k_y, k_z)$ . Then to obtain the kinetic term of the GPE, we can utilise the fact that

$$\nabla^2 = - \mathcal{F}^{-1} \sum_{\mathbf{k}} (k_x^2 + k_y^2 + k_z^2) |\mathbf{k}\rangle \langle \mathbf{k}| \mathcal{F}. \quad (12)$$

This approach of utilising the Fourier transform for the kinetic term forms the backbone of the split-step Fourier method to the simulation of the GPE [49, 50]. As the name suggests the split-step method breaks up the evolution into discrete steps in both time and between the kinetic and potential terms of the GPE. Here we will independently apply the kinetic and diagonal terms of the modified GPE of equation (2) such that evolution for a time step  $\Delta t$  under a second-order Trotter-Suzuki expansion would be given by

$$U^{(2)}(\Delta t) = e^{-i\Delta t H_d/2} \mathcal{F}^{-1} e^{-i\Delta t |\mathbf{k}|^2/2} \mathcal{F} e^{-i\Delta t H_d/2}, \quad (13)$$

with  $|\mathbf{k}|^2 = k_x^2 + k_y^2 + k_z^2$  and  $H_d = V(\mathbf{r}) - \mu + g|\Psi|^2$ . One can additionally construct a fourth-order Trotter decomposition from the second-order [51] to obtain

$$\hat{U}^{(4)}(\Delta t) = \hat{U}^{(2)}(\delta_1) \hat{U}^{(2)}(\delta_1) \hat{U}^{(2)}(\delta_2) \hat{U}^{(2)}(\delta_1) \hat{U}^{(2)}(\delta_1), \quad (14)$$

where  $\delta_1 = \frac{\Delta t}{4-\sqrt{4}}$  and  $\delta_2 = \Delta t - 4\delta_1$ , which can provide a more accurate simulation, but at a cost of five times the number of operations per time-step. The error of the second order split-step implementation scales as  $\mathcal{O}(\Delta t^2)$ , whilst that of the fourth order implementation scales as  $\mathcal{O}(\Delta t^4)$ .

### 3.1. The quantum Fourier transform

The QFT is a well established operation that forms the backbone of many quantum algorithms such as Shor's algorithm [52]. Given a quantum state  $|\Psi\rangle = \sum_{\vec{\sigma}} \Psi_{\vec{\sigma}} |\vec{\sigma}\rangle$ , where  $|\vec{\sigma}\rangle = |\sigma_1, \sigma_2, \sigma_3, \dots, \sigma_{dN}\rangle$ , one can apply the QFT operator  $\hat{\mathcal{F}}_{\text{QFT}}$

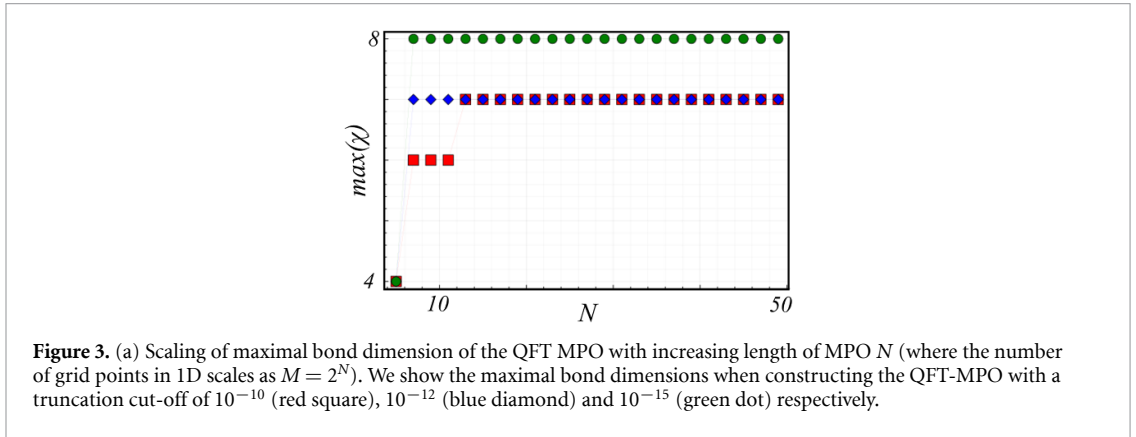
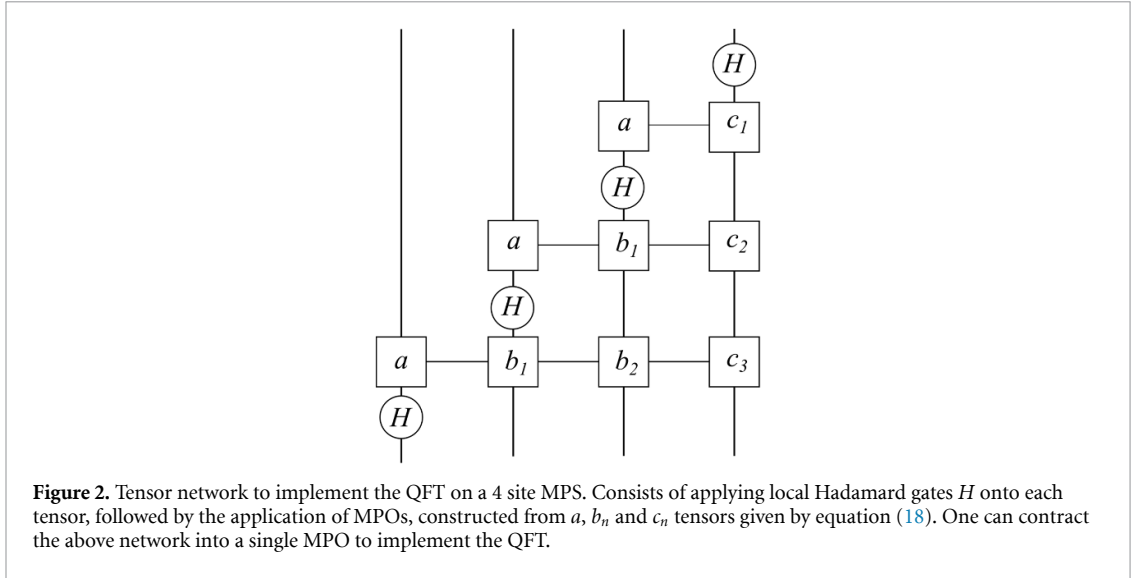
$$\hat{\mathcal{F}}_{\text{QFT}} = \frac{1}{\sqrt{2^{dN}}} \sum_{\sigma, \sigma'=0}^{2^{dN}-1} e^{\frac{2\pi i \sigma \sigma'}{2^{dN}}} |\vec{\sigma}\rangle \langle \vec{\sigma}'|. \quad (15)$$

In this way, the QFT is the quantum analogue of the discrete Fourier transform. One can construct a quantum circuit for the QFT, which consists of applying single qubit Hadamard gates ( $H$ ), and a series of controlled rotation gates  $R(\theta)$

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad R(\theta) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix}. \quad (16)$$

To exactly implement the QFT as shown in equation (15) one must additionally perform a final reversal of the ordering of the qubits. It has previously been shown that implementing this full operation, including the final qubit reversal, generates a large amount of entanglement, and the QFT corresponds to a maximally entangling operation [53, 54], which to implement as an MPO would require exponentially large bond dimensions.

However, it has recently been shown that if one instead implements the core part of the QFT operation and neglects the final reversal step, this then corresponds to an operator with exponentially decaying singular values, i.e. it generates little entanglement [41]. This key observation allows the QFT to be efficiently implemented on classical computers as an MPO with small bond dimension  $\chi_{\text{MPO}}$ .



### 3.2. MPO construction

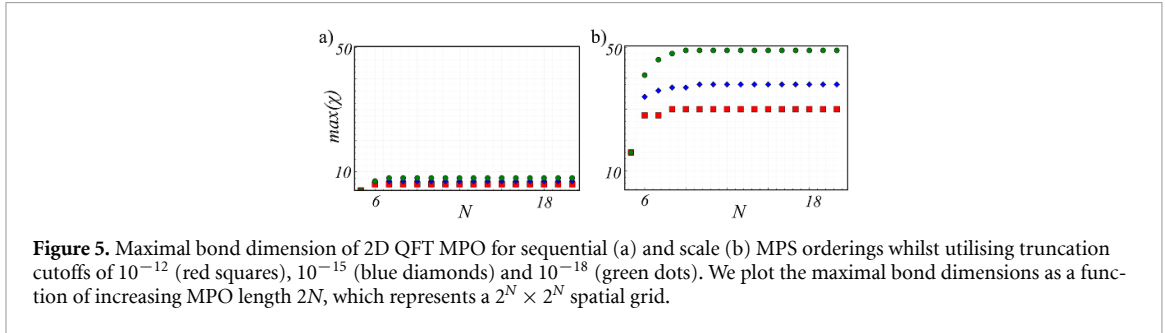
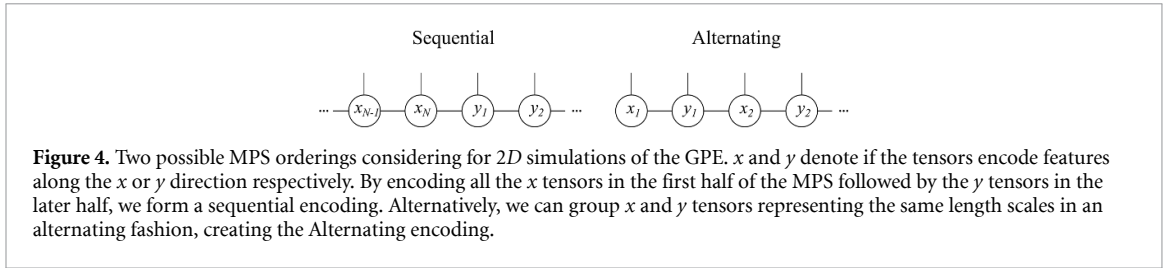
The QFT can be implemented as a tensor network as illustrated in figure 2, which consists of contracting single tensor  $H$  gates, along with a series of MPOs of bond dimension 2. Each MPO implements a series of controlled rotation gates, and are constructed from three types of tensors  $a$ ,  $b_n$  and  $c_n$  defined as

$$a = (|0\rangle\langle 0| \quad |1\rangle\langle 1|), \quad b_n = \begin{pmatrix} I & 0 \\ 0 & R\left(\frac{\pi}{2^n}\right) \end{pmatrix}$$

$$c_n = \begin{pmatrix} I \\ R\left(\frac{\pi}{2^n}\right) \end{pmatrix}. \quad (17)$$

By contracting the tensor network, we obtain an overall MPO which performs the QFT operation. We analyse how the maximal bond dimension of this QFT-MPO in 1D ( $d = 1$ ) scales with increasing grid density, i.e. the length of the MPO ( $N$ ). Note that an MPS or MPO of size  $N$  encodes  $2^N$  grid points as discussed previously. After contracting the network in figure 2, we truncate the MPO back to a smaller bond dimension using a pre-defined truncation cut-off, with the results illustrated in figure 3(a) for various cut-offs. We observe the bond dimension of the QFT-MPO quickly saturates for low grid density to a small bond dimension, even in the case of a cut-off of  $10^{-18}$ , i.e. effectively numerically exact, with the maximum bond dimension saturating at  $\chi_{\text{MPO}} = 10$ .

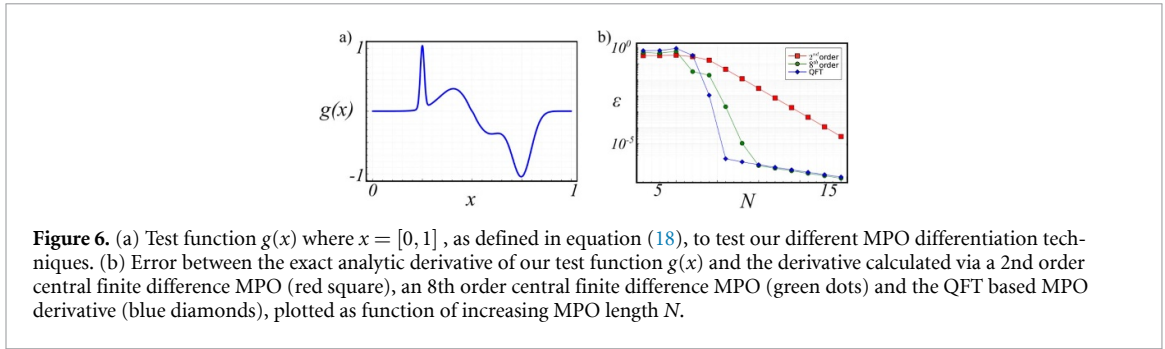
It was shown in [41] that the entanglement properties of the QFT circuit are identical to those of time evolving a chain of qubits under a  $z$ - $z$  interacting Hamiltonian, with exponential decaying interactions. Time evolution under such a Hamiltonian obeys a form of area law [41, 55], and for a chain of qubits, the entanglement generation becomes bounded to a constant, and does not increase with qubit number. This behaviour is thus exactly reflected in the saturating bond dimension of the QFT-MPO.



Above we analysed the QFT-MPO for a one-dimensional problem. We now analyse the behaviour of the QFT-MPO for higher dimensions. We will analyse the scaling of the bond dimension of the QFT-MPO for the case of two different orderings, sequential and alternate ordering, as illustrated in figure 4. Sequential ordering, as demonstrated in section 2.2, uses the first  $N$  tensors in the MPO to index the  $x$  positions, and the remaining  $N$  tensors for  $y$ . Alternate ordering groups together  $x$  and  $y$  tensors which encode the same length scale in an alternating pattern, i.e.  $x_1 y_1 x_2 y_2$ . While we are free to choose any ordering, we will gain the most benefit from our MPS approach if we choose the ordering such that correlations between distant tensors of the MPS are minimised.

The two-dimensional QFT-MPO was obtained by first constructing the QFT operator for  $x$  and  $y$  ( $\mathcal{F}_x, \mathcal{F}_y$ ) separately, and then contracting these two MPOs to form the full two-dimensional operator. The maximal bond dimensions of the 2D QFT show an identical scaling to the 1D case for both MPO orderings, with a bond dimension which quickly saturates and does not scale with MPO length, as shown in figure 5 for different truncation cutoffs and both MPS orderings. The choice of ordering, however, does have a significant impact on the resultant bond dimension. The Sequential ordering reaches a maximal bond dimension exactly equal to the 1D case of  $\chi_{\text{MPO}} = 10$  for a truncation cutoff of  $10^{-18}$ . However, the Alternate ordering saturates at  $\chi_{\text{MPO}} = 65$ , more than double compared to the Sequential order. To construct the two-dimensional QFT MPO, we apply  $\mathcal{F}_x$  and  $\mathcal{F}_y$  separately. The  $\mathcal{F}_x$  operator only generates entanglement between  $x$  tensors, and likewise for the  $\mathcal{F}_y$  operation. For the Sequential encoding,  $x$  and  $y$  are separated, and thus we are in effect simply applying a 1D QFT on the first  $N$  tensors, and then a QFT on the remaining  $N$  tensors independently. In contrast, for the Alternate ordering, correlations between  $x$  tensors have to be transferred through the MPO via intermediary  $y$  tensors, and vice-versa. Thus to perform the two-dimensional QFT, each bond in the MPO now needs to capture the generated correlations between the  $x$  and  $y$  tensors, i.e. effectively double the entanglement. As the entanglement entropy  $S$  generally scales as  $S \sim \log(\chi)$ , the required bond dimension for the alternate ordering should be much more than double the 1D QFT bond dimension, as is observed. Due to the significant reduction in bond dimension for the Sequential ordering, we use this ordering throughout unless otherwise stated.

We have illustrated how the Fourier transform can be implemented as an MPO of small bond dimension, and as such, we can perform a Fourier transform by contracting with an MPS at a computation cost scaling as  $\mathcal{O}(N\chi_{\text{MPS}}^2) = \mathcal{O}(\log_2(M)\chi_{\text{MPS}}^2)$ , where  $M$  is the number of gridpoints in the simulation. This contrasts with the computation scaling of the standard FFT, which scales as  $\mathcal{O}(M\log(M))$ , and points to the fact that performing the Fourier transform is no longer the most costly operation when exploiting MPS. Instead, calculation of the non-linear term  $|\Psi|^2$  dominates the computation, scaling as  $\mathcal{O}(N\chi_{\text{MPS}}^4)$ .



**Figure 6.** (a) Test function  $g(x)$  where  $x = [0, 1]$ , as defined in equation (18), to test our different MPO differentiation techniques. (b) Error between the exact analytic derivative of our test function  $g(x)$  and the derivative calculated via a 2nd order central finite difference MPO (red square), an 8th order central finite difference MPO (green dots) and the QFT based MPO derivative (blue diamonds), plotted as function of increasing MPO length  $N$ .

### 3.3. Analytical example

We consider a simple test case to demonstrate the ability of the QFT-MPO to perform differentiation on a chosen test function  $g(x)$

$$g(x) = \sin(2\pi x) \left[ e^{\frac{-(x-a)^2}{\delta_1}} + e^{\frac{-(x-\frac{a}{2})^2}{\delta_2}} - \frac{e^{\frac{-(x-a)^2}{\delta_2}}}{4} + e^{\frac{-(x-\frac{3a}{4})^2}{\delta_3}} \right]. \quad (18)$$

We will consider the case of periodic boundary conditions, which allows for a simpler implementation of the Fourier transform without the need to truncate low quasimomentum due to a boundary. We plot this function in  $x \in [0, 1]$  in figure 6(a), with  $\delta_1 = 0.1$ ,  $\delta_2 = 0.01$ ,  $\delta_3 = 0.05$  and  $a = 0.5$ .

For a direct comparison of the QFT and finite difference methods outlined previously, we will calculate the error  $\varepsilon$  for different methods as

$$\varepsilon_{\text{MPO}} = \frac{1}{2^N} \sqrt{\sum_x [g'_{\text{MPO}}(x) - g'_{\text{Exact}}(x)]^2}, \quad (19)$$

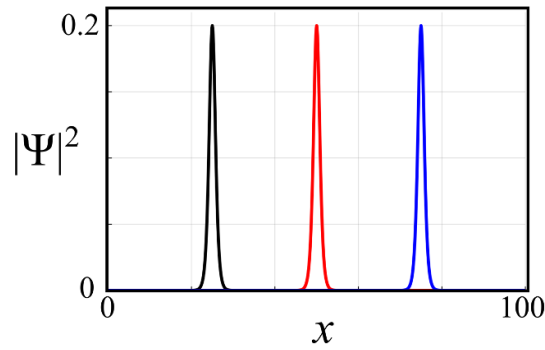
where  $g'_{\text{Exact}}(x)$  is the exact analytical derivative, and  $g'_{\text{MPO}}(x)$  is the derivative obtained from a given implementation as an MPO. In figure 6(b), we show  $\varepsilon_{\text{MPO}}$  using the QFT tensor network approach, a 2nd order central finite difference scheme, and an 8th order finite difference stencil [14] for an increasingly fine spatial resolution. The QFT tensor network approach shows small errors for dense grids similar to the 8th order finite difference and provides a much smaller error than the 2nd order finite difference. For all numerical differentiation schemes, we see that our errors tend to machine precision, i.e. exact agreement with the analytic result, as one increases the number of grid points.

## 4. Examples of simulating the GPE with tensor networks

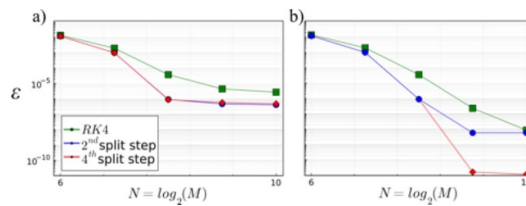
We will now consider a number of examples of problems commonly studied utilising the simulation of the GPE. We will utilise the tensor network approach introduced above using both finite difference and split-step methods for the application of the kinetic term. When utilising the finite-difference approach with tensor networks we use an 8th order finite difference stencil, and we perform time-evolution using a 4th order Runge–Kutta scheme (RK4) [56]. RK4 schemes are widely used in simulations of differential equations [56]. To perform one time-step, one first calculates  $\frac{\partial \Psi(t)}{\partial t}$ , which is then added to  $\Psi(t)$  to produce an intermediate increment of the solution. Four intermediate increments are calculated, which are then averaged over to obtain the next time-step,  $\Psi(t + \Delta t)$ . Further details are provided in appendix A.1. The split-step approach by contrast utilises the QFT and splits apart the evolution into real space and momentum space, as outlined in equations (13) and (14). We compare using both second and fourth order split-step methods.

### 4.1. Soliton propagation

As a first demonstration of performing time evolution, we consider the dynamics of a Soliton. A Soliton is an exact solution of the GPE, and consists of a localised wave packet in an otherwise uniform density ( $|\Psi|^2$ ), which travels without dissipation,  $\gamma = 0$  (see figure 7) [57, 58]. We initialise the soliton at  $t = 0$ , and use the different MPS simulation methods on offer to compare against the exact analytical solution for the soliton problem at later times.



**Figure 7.** Soliton propagation over a spatial domain  $x \in [0, 100]$ . A bright soliton is initialised at  $t = 0$  (black, leftmost pulse) and is left to propagate under the GPE with interaction strength  $g = -5$  and a velocity of  $v = 1$  and amplitude  $A = \frac{1}{\sqrt{|g|}}$  as prescribed in equation (20). We plot snapshots of  $|\Psi|^2$  at times of  $t = 10$  (red, middle pulse) and  $t = 20$  (blue, rightmost pulse).



**Figure 8.** Errors relative to analytic solution for bright Soliton propagation, with parameters outlined in figure 7, from tensor network simulations using a finite difference RK4 scheme (green squares), a second order split-step scheme (blue dots) and a fourth order split-step evolution scheme (red diamonds). We consider using an MPS truncation cutoff of (a)  $10^{-12}$  and b)  $10^{-28}$  (set to ensure we retain all non-zero singular values, i.e. untruncated) across a range of MPS lengths ( $N$ ), corresponding to  $2^N$  grid points. Results are calculated with a time step of  $\Delta t = 0.001$ , and compared at simulation time of  $t = 0.5$ .

We will consider the case of a bright soliton, which is a solution to the GPE for attractive interactions,  $g < 0$ . The analytical form the Soliton takes on is given as

$$\Psi(x, t) = A \operatorname{sech}(B(x - vt)) e^{i(vx - \frac{v^2 t}{2} - \mu t)}, \quad (20)$$

where  $A = \frac{B}{\sqrt{|g|}}$ ,  $\mu = -\frac{B^2}{2}$ , and  $v$  is the velocity of the soliton in motion. For the simulations presented, we consider  $g = -5$ ,  $A = \frac{1}{\sqrt{|g|}}$  and  $v = 1$ . We initialise  $\Psi(x, t = 0)$  in MPS form, and then use the above discussed time-evolution techniques to model the propagation of the soliton. We quantify the accuracy of each MPS simulation method by defining the error  $\epsilon_{t-e}$  as

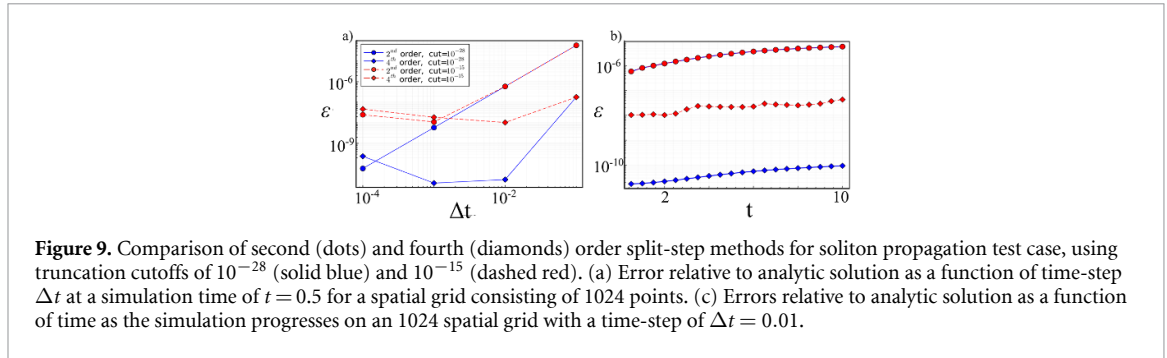
$$\epsilon_{t-e} = \sqrt{\frac{1}{2^N} \sum_j |\Psi_j^{\text{sim}} - \Psi_j^{\text{Exact}}|^2}, \quad (21)$$

where  $\Psi^{\text{sim}}$  is the results from our tensor network simulations, and  $\Psi^{\text{Exact}}$  is the analytical expression for the soliton.

We first explore the decay of errors as one increases the MPS length  $N$ , corresponding to resolving finer and finer grids, for the various simulation methods, shown in figure 8. We use both split-step methods and the finite difference RK4 scheme with a time-step of  $\Delta t = 0.001$ . We observe that across all resolutions, the split-step methods appear always to produce errors smaller than those from the RK4 scheme.

We consider a truncation cutoff of  $10^{-15}$  in figure 8(a), and perform no truncation in figure 8(b) i.e. retain all non-zero singular values. In the case of the larger truncation cutoff, both the second and fourth order split-steps schemes produce very similar errors, and there is no clear benefit from the fourth order scheme. It is only when one reduces the truncation cutoff further that one observes the benefit from the fourth order split-step scheme. This is also linked to the contribution of the time-stepping to the errors, as we will now discuss.

Increasing the spatial resolution of a simulation method alone will not necessarily reduce the errors to arbitrarily small values. One must also consider the time-step  $\Delta t$ . To this effect, we consider a fixed MPS length of  $N = 10$  and perform the above soliton simulation with a range of time-steps for the



**Figure 9.** Comparison of second (dots) and fourth (diamonds) order split-step methods for soliton propagation test case, using truncation cutoffs of  $10^{-28}$  (solid blue) and  $10^{-15}$  (dashed red). (a) Error relative to analytic solution as a function of time-step  $\Delta t$  at a simulation time of  $t = 0.5$  for a spatial grid consisting of 1024 points. (c) Errors relative to analytic solution as a function of time as the simulation progresses on an 1024 spatial grid with a time-step of  $\Delta t = 0.01$ .

second order split-step methods, shown in figure 9(a). For a truncation cutoff of  $10^{-28}$ , as one decreases the time-step from  $\Delta t = 0.1$  to  $\Delta t = 10^{-4}$ , the errors of the simulation decrease. If one instead uses a truncation cutoff of  $10^{-15}$ , the errors initial decay with decreasing time-step, before beginning to increase as one decreases the time-step below  $\Delta t = 10^{-3}$ .

In the case of performing MPS contractions exactly with zero truncation, as one decreases time-step, the global simulation error should in general decrease for a sufficiently large spatial grid, as it does when dealing with a DNS. However, performing truncation after each time step causes this logic to fail. More time-steps means performing more truncations to evolve to the same physical time in the simulation, and hence we end up discarding more information about the state and thus observe an increase in errors as one reduces the time-step. Thus, one needs to balance between time-step and truncation errors for a desired amount of data compression.

In addition, we compare how using the fourth order split-step method compares against its second order counterpart, also plotted in figure 9. Using the fourth order method, we notice a much stronger timestep-truncation interplay. This can be understood as to perform one time-step, the fourth order method performs 5 times the number of operations as the second order split-step approach. After each MPS operation, one performs a truncation of the bond dimension up to a set truncation cutoff, and thus to evolve one time step, we perform more truncations with the fourth order method, introducing larger truncation errors per time step and off-setting the benefit of the higher order method.

However, the power of the fourth order split-step method is that it allows one to choose a much larger time-step whilst retaining a high accuracy, as can be demonstrated in both figures 9(a) and (b). For a time-step of  $\Delta t = 0.1$ , using a second order split-step approach produces errors on the order of  $10^{-5}$ . Instead, using the fourth order scheme, we are able to achieve errors below the  $10^{-10}$  level.

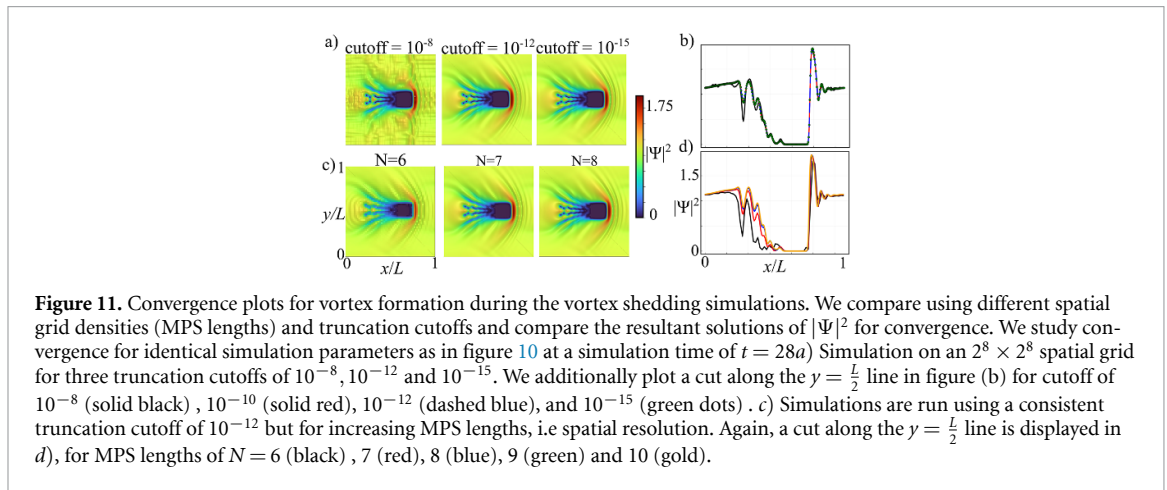
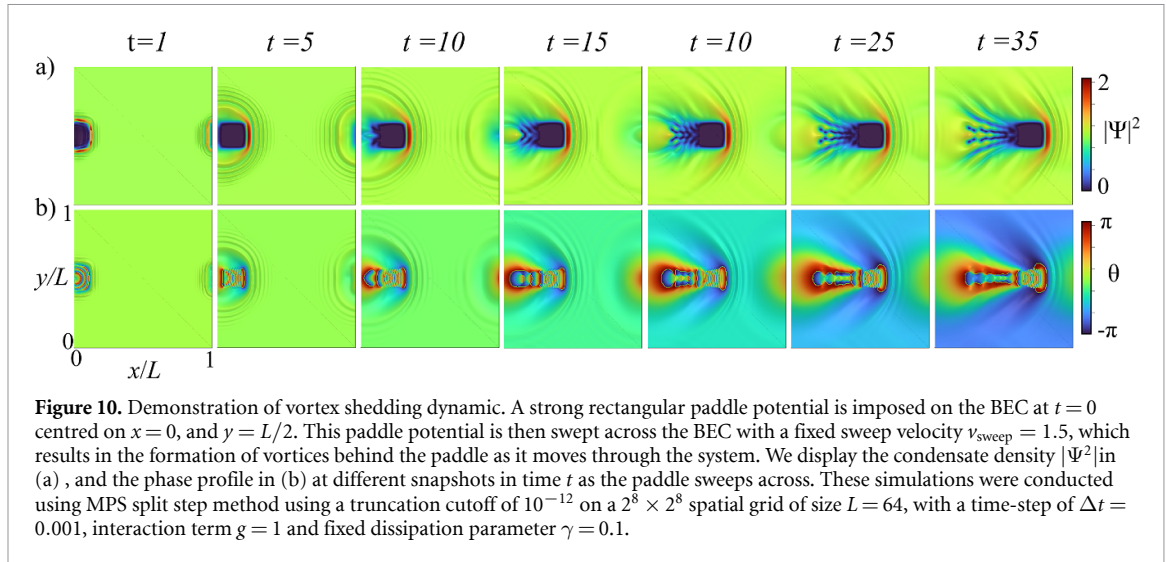
These results highlight the importance of balancing the time-step and desired truncation level correctly. For sufficiently small time-steps, the overall error of our tensor network simulations is due to an interplay between truncation level, time discretisation error and spatial discretisation error, which must all be balanced to obtain the most amount of data compression whilst minimising simulation errors.

## 4.2. Vortex shedding

We now consider a more challenging dynamical example of vortex generation, which does not have an exact analytical solution. There has been a range of experiments conducted in recent years studying the behaviour and dynamics of vortices generated within superfluids [59–63], and understanding the behaviour and dynamics of vortices is central to the field of quantum turbulence. For example, it has been shown that the ability of a superfluid to flow without viscous effects is broken above a critical velocity due to the formation of vortices, which induce an effective viscous force [64]. Vortices in a BEC can be a localised region with zero occupancy in the condensate density, i.e. where  $|\Psi|^2 = 0$ . In addition, vortices formed in a BEC must have a quantised circulation, where the phase  $\theta$ , defined as  $\Psi = |\Psi|e^{i\theta}$ , must change by an integer multiple of  $2\pi$  around the vortex core [65]. This quantisation arises from the fact that  $\Psi$  must be single valued and continuous. We will utilise the modified GPE with a non-zero dissipation term ( $\gamma \neq 0$ ), as most studies into the generation of vortices include damping to account for atom loss.

### 4.2.1. Single paddle

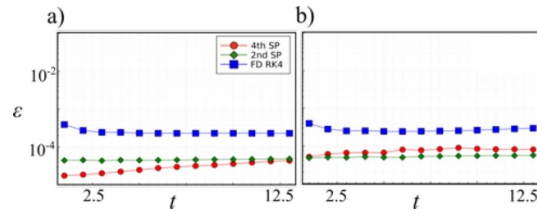
We use our MPS based methods to now study the dynamics of a BEC when a strong external potential is swept through the system. This external potential,  $V_{\text{pad}}(x, t)$ , which we call a ‘paddle’, has a fixed, rectangular shape, and is moved across the BEC with a sweep velocity  $v$  (see figure 10). To have a resemblance closer to that which occurs experimentally, we implement a form of rectangular paddle in MPS



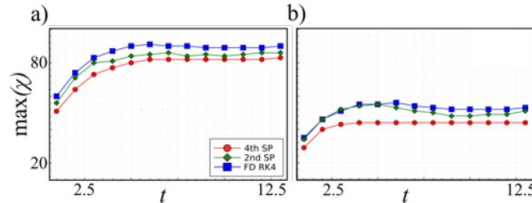
form with rounded edges (see appendix B for further details). Depending on the sweep velocity and the strength of dissipation, pairs of vortices will be generated from the rear of the paddle at regular intervals, which then trail behind the paddle as it moves across the system, akin to the formation of von-Karman vortices in classical fluid flows [63]. These vortices can be seen in figure 10(a). We additionally display the phase ( $\theta$ ) in figure 10(b). Such phase plots allow vortices to be clearly identified, as the phase will change by an integer multiple of  $2\pi$  around the vortex. One can also infer the direction of rotation of a vortex, depending on if the phase changes from  $-\pi$  to  $\pi$  in a clockwise or counter clockwise direction.

As with all MPS methods, one must choose a suitably small truncation cutoff to achieve convergence of the physics, however we must also take into consideration the effect of the spatial resolution. We must ensure that the simulation contains sufficiently many grid-points to fully resolve vortex formation. We perform a vortex shedding simulation using the second order split-step method for a range of truncations and system sizes, and plot the resultant density profiles in figure 11. We observe that a truncation cutoff of  $10^{-12}$  is more than sufficient to accurately converge the BEC dynamics, agreeing perfectly with truncation cutoffs of  $10^{-15}$  across all simulation times and system sizes considered. In addition, we find that we really require a spatial density above  $2^8 \times 2^8$  to perfectly converge the physics. Performing simulations at a spatial density of  $2^7 \times 2^7$  yields results which are qualitatively similar to the more resolved simulations, but there are noticeable deviations of the density around the 10% level where the vortices form.

Having implemented the vortex shedding dynamics with MPS methods, we now wish to quantify the errors in these simulations, comparing the error from using MPS with a finite differences RK4 scheme, and from MPS exploiting the split-step methods using the QFT-MPO. To quantify the error of our MPS based simulations, we perform a comparison with DNSs, for which we use the software package of XMDS2 [66]. XMDS2 solves ordinary and PDEs via a variety of DNS methods. For the results presented in this work, we setup XMDS2 such that it is 4th order accurate in time, and calculates our spatial



**Figure 12.** (a) Errors from MPS based vortex shedding simulations against XMDS2 direct simulation, for an  $256 \times 256$  spatial grid, with  $L = 64$ ,  $g = 1$ ,  $\gamma = 0.1$ ,  $\nu = 1.5$  and  $\Delta t = 0.001$ . We compare errors during time evolution using the finite difference RK4 evolution scheme (blue squares), a second order split-step scheme (green diamonds) and a fourth order split-step scheme (red dots). The MPS simulations are run for different truncation cutoffs of (a)  $10^{-15}$  and (b)  $10^{-12}$ .



**Figure 13.** (a) Maximal bond dimension of  $\Psi$  during vortex shedding simulations for an  $128 \times 128$  spatial grid, with  $L = 64$ ,  $g = 1$ ,  $\gamma = 1.0$ ,  $\nu = 1.5$  and  $\Delta t = 0.001$ . We plot the maximal bond dimension during time evolution using the finite difference RK4 evolution scheme (blue squares), a second order split-step scheme (green diamonds) and a fourth order split-step scheme (red dots). The MPS simulations are run for different truncation cutoffs of (a)  $10^{-15}$  and (b)  $10^{-12}$ .

derivates in a spectral fashion, making use of the fast Fourier transform. The computational costs of these methods scale as  $M \log M$  with the number of grid points  $M$ . For comparison, within our MPS we are able to represent a number of grid points that grows exponentially with the length of our MPS. We define the global error from the MPS simulation as

$$\varepsilon_{t-d} = \sqrt{\frac{1}{4^N} \sum_x |\Psi_{\text{MPS}}(x) - \Psi_{\text{DNS}}(x)|^2}, \quad (22)$$

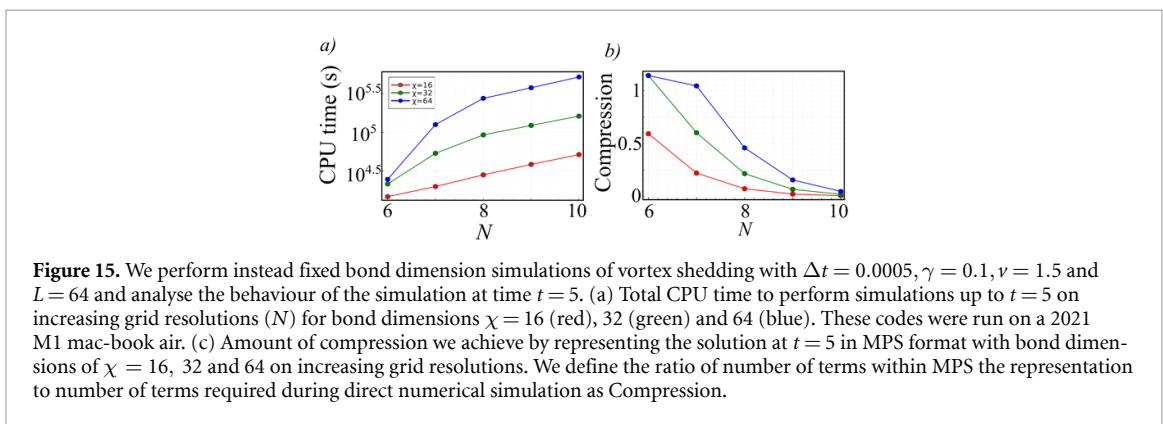
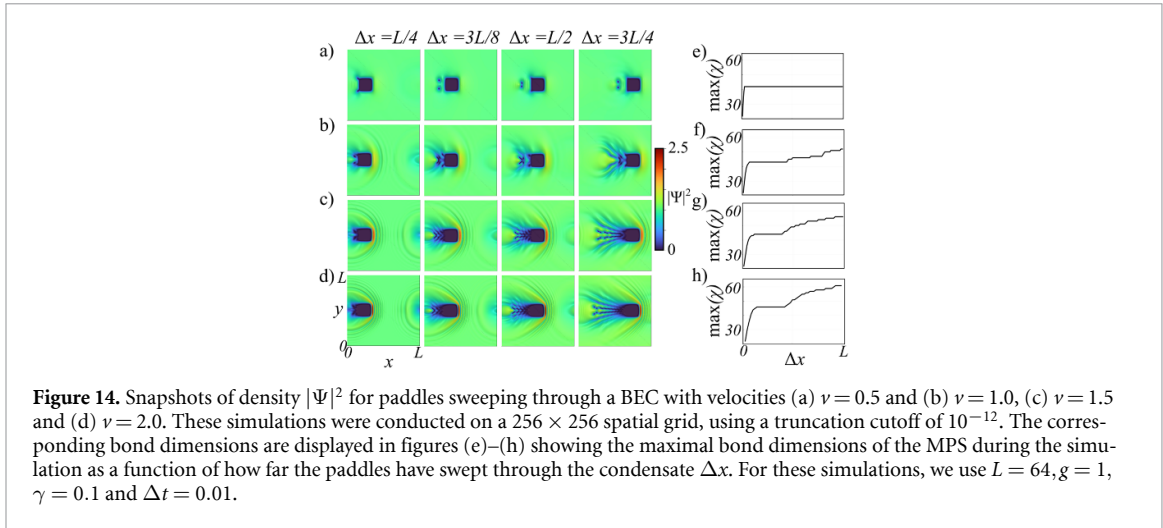
where  $\Psi_{\text{DNS}}$  is the simulation result from XMDS2.

We first conduct a simulation on an  $256 \times 256$  spatial grid, with a sweep velocity of  $\nu = 1.5$ , and a dissipation rate of  $\gamma = 0.1$ . We calculate the global error from different simulation methods in figure 12. This simulation is conducted using a time-step of  $\Delta t = 0.001$ , and we observe that at all truncation errors, both the second-order and fourth-order split-step methods outperform the finite-difference RK4 method, as these produce smaller global errors as in figure 12.

When performing the tensor network simulations with a truncation cutoff of  $10^{-15}$ , the fourth order split-step method produces the smallest errors on the order of  $2 \times 10^{-5}$ , which slowly grows across the simulation time-frame. For the same truncation level, the second order split-step also performs well, but with errors which appear to remain at a constant level of  $5 \times 10^{-5}$ . However an important note is that the second-order accurate split-step method produces smaller errors than the fourth-order split-step for the larger truncation cutoff of  $10^{-12}$ . This observation is linked to the fact that one must perform more truncation steps during the fourth order split-step scheme as compared to the second order, and thus when the truncation error dominated the second order method can produce lower errors.

We also investigated an alternative time evolution scheme widely used for MPS in the form of the time-dependent variational principle (TDVP). TDVP is a local time evolution scheme, which sweeps across the MPS, evolving each tensor individually under the action of the local effective Hamiltonian [8]. In appendix A.2, we further outline the TDVP approach and show that while TDVP can provide smaller errors than those of the finite difference and split-step method for large cutoffs, but larger errors for smaller cutoffs.

The maximal bond dimension to represent the function  $\Psi$  was also tracked throughout the simulations, and displayed in figure 13. Across all simulation methods, we observe modest bond dimensions which depend on the truncation cutoff chosen. When simulating with a truncation cutoff of  $10^{-15}$ , the observed maximal bond dimension appears to grow linearly with time as the dynamics unfold reaching a



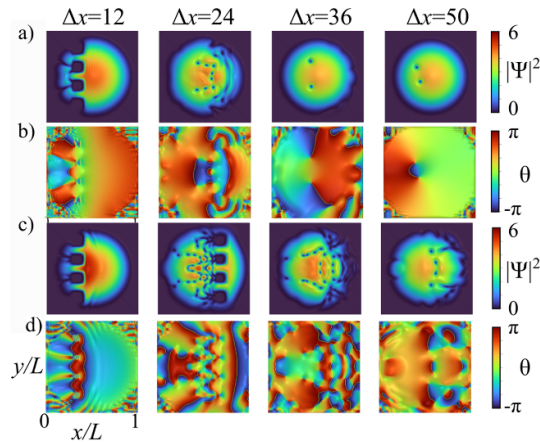
peak value of 80 at the end of the simulation. Using a smaller truncation cutoff results in lower maximal bond dimensions, which increases at a reduced rate.

We now consider how the speed of the paddles impacts the ability to represent the solution efficiently in MPS format. In general, it is observed that the faster the paddle sweeps through the BEC, the more vortices are generated, and as the velocity is increased further still, we begin to see the emergence of waves resembling shock waves, which can be observed in figure 14.

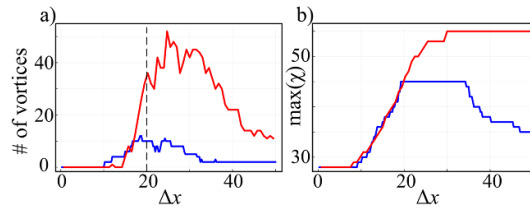
A series of paddle velocities were considered, and the corresponding maximal bond dimensions during the simulations are also shown in figure 14. It appears that increasing the velocity of the paddle in general requires a growth in the bond dimension to represent the resultant dynamics. Increasing the sweep velocity generates a lot more shock fronts and produces many more vortices behind, resulting in a much less smooth solution compared to a low sweep velocity, thus requiring a larger bond dimension to encode.

The ability to restrict tensor network simulations to low bond dimensions is ultimately what provides the benefit over other simulation techniques. For a fixed bond dimension, the computational complexity of the tensor network methods scales linearly in the length of the MPS/MPO, corresponding to a logarithmic scaling with number of grid points, and only polynomially in terms of the bond dimension. This yields an overall computational scaling as  $\mathcal{O}(\chi^4 \log M)$ , with the dominant contribution from the calculation of the non-linear term  $|\Psi|^2$ . This contrasts with the corresponding complexity scaling of DNSs, which scale according to  $\mathcal{O}(M \log M)$  with the dominant scaling via that of the fast Fourier transform. To demonstrate this scaling of the tensor network method, we calculated the required CPU time to perform the vortex shedding simulation outlined above across various grid-size, ranging from  $64 \times 64$  up to  $1024 \times 1024$ , for different fixed bond dimensions. Results are illustrated in figure 15(a). We find precisely a linear scaling in the CPU time required with the length of the MPS, ranging from  $10^4$ s on a  $64 \times 64$  spatial grid to  $10^5$ s on an  $1024 \times 1024$  spatial grid, with a bond dimension of  $\chi = 32$ .

Along with the speedup, there is also the data compression aspect. By restricting  $\chi$ , we reduce the number of terms one requires to represent the state  $\Psi$ , reducing the memory requirements to simulate the dynamics. One can calculate the ratio of the number of terms to encode  $\Psi$  as an MPS with given



**Figure 16.** Snapshots of the dynamics as four paddles are swept through a BEC confined in a harmonic trap with sweep velocity  $\nu = 0.5$  (a), (b) and  $\nu = 1.5$  (c), (d). After the paddles have swept through a distance of  $\Delta x = 20$  they begin to be linearly ramped down over a fixed time period of  $T_r = 5$ . We show snapshots of the dynamics after the paddles have swept through various distances  $\Delta x$ , where both (a), (c) show the density  $|\Psi|^2$  and (b), (d) the phase  $\theta$ . We use simulation parameters of  $L = 35, g = 1, \gamma = 0.1, \Delta t = 0.0005$  on a  $256 \times 256$  spatial grid, with a harmonic trap frequency of  $\omega = 0.15\sqrt{2}$ .



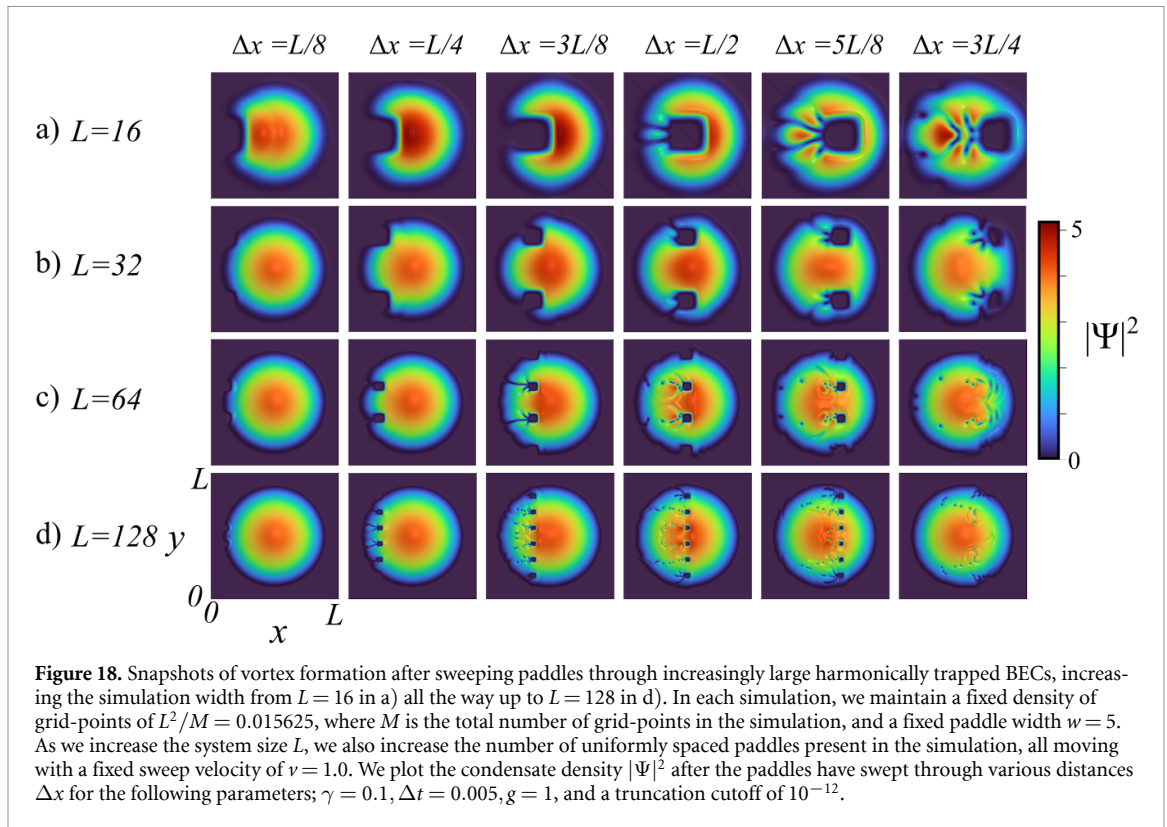
**Figure 17.** (a) Number of vortices present in simulation domain behind the paddles and (c) maximal bond dimension of  $\Psi$  as a function of the distance the paddles swept through ( $\Delta x$ ) for the simulation shown in figure 16. After the paddles sweep through a distance  $\Delta x = 20$  (black dashed line), they are ramped down to zero over a time of  $T_r = 5$ , and system left to freely evolve. Shown for a paddle sweep velocity of  $\nu = 0.5$  (blue) and  $\nu = 1.5$  (red).

bond dimension, to the number of terms required to represent  $\Psi$  during a DNS. This ratio decreases exponentially fast as one increases the length of MPS, as observed in figure 15(b). Combined, the data compression and scaling of tensor networks allows one to exploit these methods to simulate incredibly fine spatial resolutions, which would be otherwise unfeasible with direct numerical simulations.

#### 4.2.2. Multiple paddles

Having demonstrated and bench-marked the ability of one to use tensor networks to perform simulations of the GPE in the previous sections, we now use our tensor network approach to further probe vortex dynamics and interactions in the modified GPE. We consider an alternative stirring mechanism, replacing the one large paddle with four smaller paddles, all aligned along the  $y$  direction. This type of geometry has been considered previously in experiments [27], and allows more vortices to be injected into the BEC for a given sweep velocity at a given time. We introduce the paddles into the BEC, which are then swept across in the  $x$  direction with a constant sweep velocity  $\nu$  as was the case previously. Once the paddles have travelled through halfway across the BEC, we linearly ramp down the potential strength to zero over a fixed time  $T_r$ , after which the vortices are left to evolve freely. In addition, we introduce a fixed harmonic trap throughout the simulation to confine the BEC into a small region, allowing us to trap the created vortices. We consider a fixed dissipation rate of  $\gamma = 0.1$ , and two sweep velocities  $\nu = 0.5$  and  $\nu = 1.5$ , with snapshots of the resultant dynamics illustrated in figure 16.

One can observe vortices continuously shedding off the paddles as they sweep across the BEC, with more vortices generated behind the faster moving paddles. The number of vortices during the simulation can also be recorded [67], as shown in figure 17(a), where we clearly observe that the  $\nu = 1.5$  paddle produces around five times the number of vortices as compared to the  $\nu = 0.5$  simulation. For both sweep velocities, there is an initial increase in the number of vortices as the paddles move into the bulk of the harmonic trap. After the paddles have swept through a distance of  $\Delta x = 20$ , they are quickly ramped down, and more vortices are created during this stage.



**Figure 18.** Snapshots of vortex formation after sweeping paddles through increasingly large harmonically trapped BECs, increasing the simulation width from  $L = 16$  in a) all the way up to  $L = 128$  in d). In each simulation, we maintain a fixed density of grid-points of  $L^2/M = 0.015625$ , where  $M$  is the total number of grid-points in the simulation, and a fixed paddle width  $w = 5$ . As we increase the system size  $L$ , we also increase the number of uniformly spaced paddles present in the simulation, all moving with a fixed sweep velocity of  $v = 1.0$ . We plot the condensate density  $|\Psi|^2$  after the paddles have swept through various distances  $\Delta x$  for the following parameters;  $\gamma = 0.1, \Delta t = 0.005, g = 1$ , and a truncation cutoff of  $10^{-12}$ .

After a vortex is formed, it remains present in the simulation until it collides with another vortex with opposite rotational direction, where they annihilate one another. These vortex collisions give rise to a decay in the number of vortices after the paddles are ramped down. These simulations were conducted using a truncation cut-off of  $10^{-12}$ , and the resultant maximal bond dimensions are displayed in figure 17(b). Again it was found a small bond dimension was required to represent the resultant dynamics, where the bond dimension initially grows linearly in time as the paddles sweep across and generate vortices, before saturating for later times.

To illustrate the true benefit of using tensor networks, in figure 18 we consider performing simulations over increasingly large spatial sizes  $L$ , whilst maintaining a fixed paddle width ( $w$ ), each time doubling the length of the system  $L$  and hence the number of total grid points per dimension, whilst retaining a fixed spatial discretisation  $\Delta x$ . The following parameters are chosen,  $\Delta x = 0.125, \Delta t = 0.0005, g = 1, \omega = \frac{5.25\sqrt{2}}{L}, \gamma = 0.1, w = 5$  and  $v = 1.0$ , where the harmonic trap frequency is scaled with the system length.

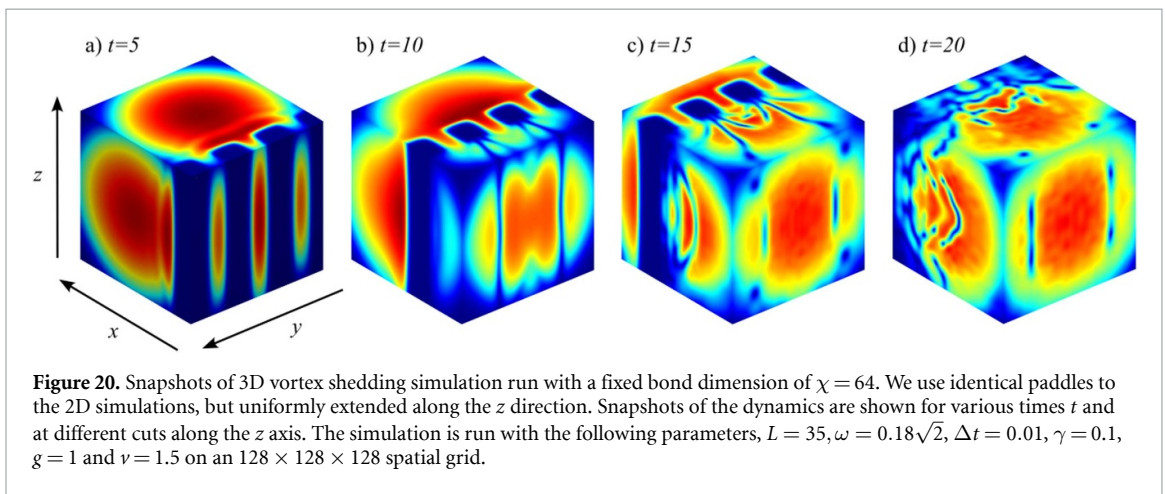
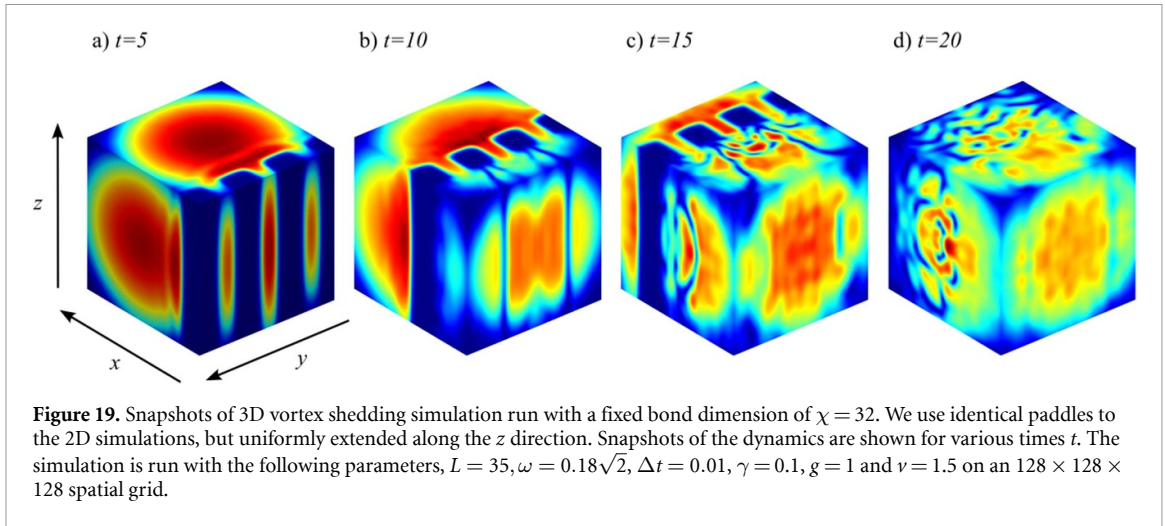
In this way we demonstrate the ability of tensor network simulations to resolve increasingly fine length scales without the exponential increase in memory requirements.

### 4.3. 3D simulation

As a final example of vortex shedding, we extend the 2D simulations to a full 3D simulation. Once again, we prepare a BEC trapped in a harmonic trap  $V(x, y, z) = \frac{1}{2}\omega^2(x^2 + y^2 + z^2)$ , where paddles are swept through with a fixed sweep velocity. For simplicity, we consider the case of translationally invariant paddles along the  $z$  axis, i.e.  $V_{3\text{Dpad}}(x, y, z, t) = V_{2\text{Dpad}}(x, y, t)$ , where  $V_{2\text{Dpad}}(x, y, t)$  is the 2D paddle potential as used in the previous section.

As a demonstration, we perform the 3D simulations on an  $128 \times 128 \times 128$  spatial grid with fixed bond dimensions of  $\chi = 16, 32$  and  $64$ , with snapshots of the dynamics illustrated in figures 19 and 20. We see that even for these modest bond dimensions, we are able to clearly resolve the formation of vortices in the 3D BEC. The true scope of the data compression achievable with MPS is most clearly illustrated in 3D. Using a bond dimension of  $\chi = 64$ , one requires only to store 4% of the number of variables parameterising the solution compared to DNS.

These simulations on an  $128^3$  spatial grids are beginning to push that which can be achieved easily with DNS [25, 68]. Performing a DNS on an  $256^3$  spatial grid would require a factor of 8 growth in the memory requirements, and a corresponding increase in computation time. In contrast, performing a



simulation on a  $256^3$  spatial grid and beyond, with MPS and a fixed bond dimension, requires only a small increase in computational complexity, which can allow for very dense simulations.

## 5. Simulating dipolar gases with tensor networks

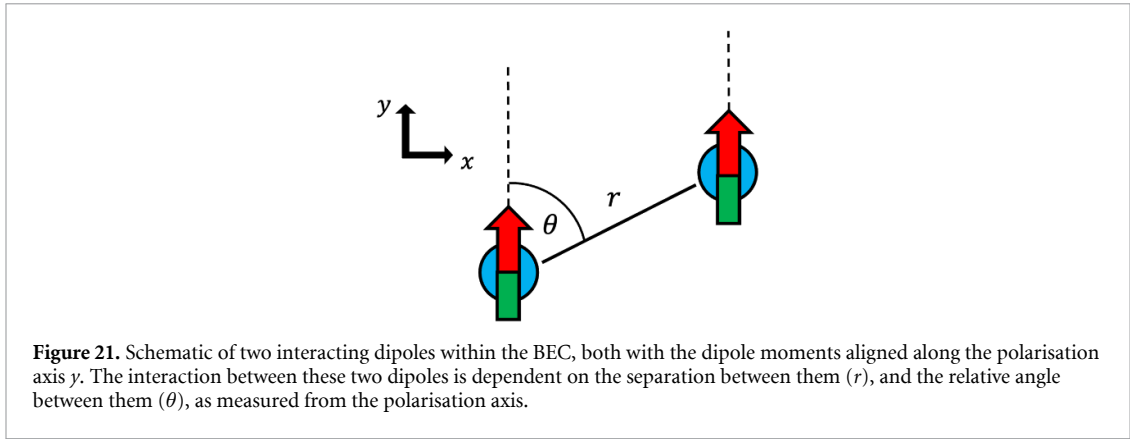
We have so far shown how one can efficiently use MPS methods to simulate the standard GPE. We will now introduce additional complexity into the GPE, in the form of long-range interactions of a dipolar gas. A dipolar gas is a BEC made from cold atoms with a dipole moment, where an additional interaction is introduced dependent on the relative separation and orientation of neighbouring dipole moments, as illustrated in figure 21. Specifically, we consider the case of a polarised dipolar gas, where the dipole moments of all the atoms point along the same direction, which we take throughout as along the  $y$  axis. When the dipoles are aligned (i.e. the two atoms separated along the polarisation axis), there will be an attractive interaction, but if separated against the polarisation angle there will be a repulsive interaction. There exists a magic angle  $\theta_m$  at which there is zero dipole-interaction between the atoms.

We can express the dipole-dipole interaction  $\Phi(x)$  at some point  $x$  as an integral over all space of the condensate density  $|\Psi|^2$  and the dipole interaction potential  $U_{dd}(r)$

$$\Phi(x) = \int dx' U_{dd}(x-x') |\Psi(x')|^2, \quad (23)$$

with the full 3D potential given by

$$U_{dd}(r) = \frac{3}{4\pi r^3} \left( 1 - \frac{3y^2}{r^2} \right). \quad (24)$$



This dipole-dipole interaction term is introduced into the GPE as follows

$$i\hbar \frac{\partial \Psi}{\partial t} = \left[ -\frac{\hbar^2}{2m} \nabla^2 + V(x) + g|\Psi|^2 + g_{\text{dd}}\Phi(x) \right] \Psi , \quad (25)$$

where  $g_{\text{dd}}$  is known as the dipolar interaction strength and controls the dipolar interactions within the system. One can then apply all the aforementioned MPS based schemes to simulate the dipolar GPE, provided that the dipole-dipole term  $\Phi(x)$  can be efficiently constructed in MPS form.

### 5.1. Constructing the interaction term

One possibility to construct  $\Phi(x)$  is to numerically integrate equation (23) directly. However, the dipole interaction potential is highly singular as it diverges  $r \rightarrow 0$ , creating difficulties for numerical integration techniques. There are some approaches to deal with this in real space [69], however, one may reformulate the interaction as a convolution [70]

$$\Phi(\mathbf{r}) = \mathcal{F}^{-1} \left\{ \tilde{U}_{\text{dd}}(\mathbf{k}) \mathcal{F} \left\{ |\Psi(\mathbf{r})|^2 \right\} \right\} , \quad (26)$$

where  $\tilde{U}_{\text{dd}}(k)$  is the k-space interaction kernel, found via the Fourier transform of the dipole interaction potential (equation (24)), and is given by

$$\tilde{U}_{\text{dd}}(k) = 3 \frac{k_y^2}{k^2} - 1 . \quad (27)$$

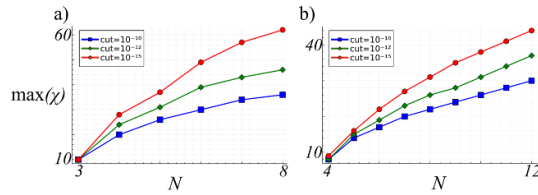
One can also obtain expressions for  $\tilde{U}_{\text{dd}}(k)$  for effective 2D dipolar systems, where one makes assumptions about the behaviour along the  $z$  axis. One could assume the condensate is uniform along the  $z$  direction, in which case  $U_{\text{dd}}(k)$  takes on the same form as equation (27). Alternatively one could also consider a condensate in a strong harmonic trap along the  $z$  direction, where all dynamics occur in the  $x$ - $y$  plane only. Details of the 2D dipolar interactions are provided in appendix C.

We note that calculation of the dipolar interactions via the Fourier transform implies the presence of periodic boundaries. This will have an unwanted effect of including in the dipolar terms the contribution from copies of the physical system outwith the simulation domain. To mitigate this, one will often apply a cutoff in the dipolar interaction kernel to avoid these unphysical contributions. We outline the details of this and its tensor network implementation in appendix D.

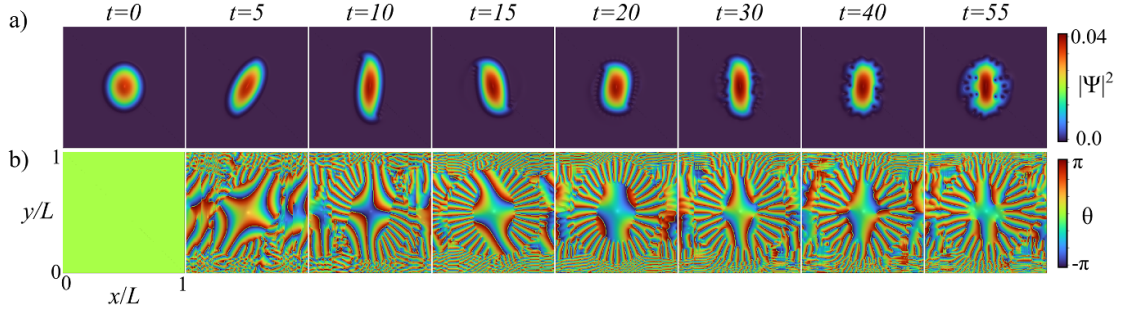
To simulate dipolar gases with the MPS approach,  $\tilde{U}_{\text{dd}}(\mathbf{k})$  is first created as an array for all  $\mathbf{k}$  points, and SVDs are then performed to encode into the correct MPS form, where we then truncate with some truncation cutoff. The resultant maximal bond dimensions are illustrated within figure 22. The maximal bond dimension does grow with increasing system size as the bond dimension appears to scale logarithmically with MPS length, and not exponentially. This would thus allow one still to push simulations to very fine spatial resolutions without a blow-up of bond dimension to represent the interaction kernel.

### 5.2. Vortex formation upon rotation

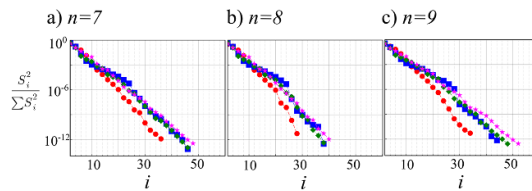
A further test case for MPS-based simulation methods is now considered where we study the formation of vortices as we begin to spin the condensate within a harmonic trap [71–76]. A dipolar BEC is trapped in a stationary, non-rotating harmonic trap. Then a sudden rotation at frequency  $\Omega$  is imparted on the trap and a small anisotropy is introduced to impart angular momentum onto the BEC. To account for



**Figure 22.** Maximal bond dimension scaling of  $U_{dd}$  (equation (27)) for increasing MPS size ( $N$ ) for truncation cutoffs of  $10^{-10}$  (blue squares),  $10^{-12}$  (green diamonds) and  $10^{-15}$  (red dots). Results are shown for the (a) 3D interaction kernel and (b) the 2D interaction kernel assuming a uniform  $z$  direction. Further details are provided in appendix C.



**Figure 23.** Snapshots of the resultant dynamics of a dipolar gas with parameters  $g = 250$ ,  $g_{dd} = 75$ ,  $\gamma = 0.03$  and  $L = 25$  as it begins to rotate around the  $z$  axis. The condensate is first prepared in the ground state of a harmonic trap with  $\omega = 1$  ( $t = 0$ ), before a sudden rotation at frequency  $\Omega = 0.7$  is imparted. The simulation is conducted on an  $256 \times 256$  spatial grid, using a time-step of  $dt = 0.001$  and truncation cutoff of  $10^{-15}$ . We display both  $|\Psi|^2$  and the phase at various simulation times  $t$ .



**Figure 24.** Squared singular values of the MPS  $\Psi$  in descending order during the simulation of dipole rotation as outlines in figure 23. We plot the singular values across the (a)  $n = 7$  (b)  $n = 8$  and (c)  $n = 9$  bond of the MPS at simulation times of  $t = 10$  (red dots),  $t = 25$  (blue square),  $t = 40$  (green diamond) and  $t = 55$  (pink stars). At all times and at all bonds the singular values decay linearly in the above plots, indicating the singular values decay exponentially fast.

this rotation, we introduce the angular momentum operator  $L_z = -i\hbar(x\partial_y - y\partial_x)$  into the dipolar GPE as follows

$$i\hbar \frac{\partial \Psi}{\partial t} = \left[ -\frac{\hbar^2}{2m} \nabla^2 + V(x) - \mu + g|\Psi|^2 + g_{dd}\Phi(x) - \Omega L_z \right] \Psi. \quad (28)$$

For the simulations which follow, we consider a strong conventional interaction term of  $g = 250$  and dipolar interaction strength of  $g_{dd} = 75$ , where we first use imaginary time-evolution to prepare the groundstate of the stationary trap, before the rotation is suddenly turned on to  $\Omega = 0.7$ . Snapshots of the resultant dynamics are displayed in figure 23, displaying both the density of the BEC and the associated phase.

The condensate starts in a rotationally symmetric state at  $t = 0$ , but as we begin to spin the trap, we observe the condensate morph into a more elliptical shape which rotates around. After an initial period, we begin to observe the formation of vortices at  $t \sim 40$ , which can again be identified via the quantised circulation of phase. The associated singular values for these dynamics are displayed in figure 24 across different bonds  $n$  of the MPS, and for various simulation times. There is a clear exponential decay of singular values across all times, which implies the ability of one to truncate the MPS bond dimension substantially, by discarding the least significant singular values.

In this paper we have demonstrated the ability to perform simulations of the GPE on fine grids making use of the tensor network formalism, where the main advantage is that of data compression. Note

that this is expected to work well for cases where those fine grids are necessary, but reformulating a direct simulation in terms of tensor networks introduces some computational overhead. Therefore if the grid sizes required for some problem can be handled with a more conventional direct solver, then conventional solvers would often be a better approach as opposed to using a tensor network formalism. We have noted several examples in the manuscript involving turbulence and dipolar interactions where fine grids may be required, and therefore tensor networks can give an advantage. Furthermore, even for those problems where tensor networks may be of benefit, the data compression one can achieve with tensor networks for a given problem will depend upon the physics of that instance, in the form of the structure of the correlations present. Therefore one must test that tensor networks can provide useful data compression for each instance of a problem.

In addition, we note that other methods have been developed which can further compete against tensor network approaches such as neural networks (NNs). NNs have also been applied to the study of the GPE, where they have been demonstrated to be able to solve a 1D time independent GPE for a variety of external potentials. The drawback with NNs as compared with tensor-network approaches is that they require sufficient training of the network to reproduce accurate results, which may be computationally intensive. But in general NNs have been used in the past for other PDEs [77], such as for simulations of fluid dynamics [78] and heat transfer problems, where a physics-informed neural network was used to qualitatively capture the dynamics of a convection flow past a cylinder, with errors on the order of a few per cent level as compared with a direct solver [79]. Such NN methods may also be able to be applied for dynamical simulations of the GPE and dipolar gases.

## 6. Conclusion

We have shown how MPS can be exploited to encode solutions of the GPE and achieve substantial data compression for dynamics where we are able to restrict the bond-dimension. Along with finite-difference MPOs to calculate derivative terms, the quantum Fourier transform provides a useful alternative way to perform time-evolution dynamics, often producing smaller errors for the same bond-dimension as compared with the finite-difference counterpart. Going beyond the GPE, we included effects of dipolar interactions, again relying on the quantum Fourier transform. We demonstrated how the required bond dimension of the interaction term scales linearly in MPS length, and thus one may still be able to achieve an exponential data compression as compared with direct simulations. This work opens up the possibility of performing extremely finely resolved numerical simulations of cold gases, beyond that which would be otherwise possible. Having access to very fine spatial resolution is key for studying and understanding quantum turbulence.

We have so far focused on time-evolution of single component Bose gases. However, future work may begin to explore the dynamics of multi-component systems, such as a binary condensate or a spin-1 system. By modifying the interactions between components, one can explore a variety of rich physics and explore phase transitions [29, 68, 80]. In addition, we have here demonstrated inclusion of dipolar interactions. An important future avenue of work may be a further detailed exploration of dipolar physics, such as the formation of super-solids [30–33], while exploiting MPS to perform simulations on very fine spatial grids.

## Acknowledgments

We thank Paula García-Molina, Preetma Soin, Neil Gaspar, Ashton Bradley, Blair Blakie and Dieter Jaksch for helpful discussions. This work was supported by the EPSRC through the Programme Grant QQQS (EP/Y01510X/1), and Grant EP/Y005058/2. The authors would like to acknowledge AWE plc for financial support of the work. UK Ministry of Defence © Crown owned copyright 2024/AWEs.

## Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: <https://doi.org/10.15129/74c84249-5aeb-477e-ac24-83d983687bd3> [85].

## Appendix A. Time evolution schemes

### A.1. 4th order Runge–Kutta

A frequently used numerical scheme to perform time evolution of differential equations is that of Runge–Kutta 4 (RK4). RK4 belongs to a class of Runge–Kutta solvers, however RK4 is most widely used for its good accuracy and stability for solving PDEs.

Given a PDE of the following form,

$$\frac{\partial \Psi(t)}{\partial t} = -\frac{i}{\hbar} \hat{H}(t) \Psi(t) , \quad (\text{A1})$$

one performs RK4 via evaluating the  $\frac{\partial \Psi(t)}{\partial t}$  at 4 intermediate points in order to produce  $\Psi(t + dt)$ . Specifically, one must calculate the following intermediary states during an RK4 time-step

$$K_1 = \frac{\partial \Psi(t)}{\partial t} , \quad (\text{A2})$$

$$\Psi_1 = \Psi(t) + \frac{dt}{2} K_1 , \quad (\text{A3})$$

$$K_2 = \frac{\partial \Psi_1(t + \frac{dt}{2})}{\partial t} , \quad (\text{A4})$$

$$\Psi_2 = \Psi(t) + \frac{dt}{2} K_2 , \quad (\text{A5})$$

$$K_3 = \frac{\partial \Psi_2(t + \frac{dt}{2})}{\partial t} , \quad (\text{A6})$$

$$\Psi_3 = \Psi(t) + dt K_3 , \quad (\text{A7})$$

$$K_4 = \frac{\partial \Psi_3(t + dt)}{\partial t} . \quad (\text{A8})$$

Having obtained the intermediate derivatives  $K_1, K_2, K_3$  and  $K_4$ , we produce the next time-step in the simulation via addition with the solution  $\Psi(t)$  at the current time,

$$\Psi(t + dt) = \Psi(t) + \frac{dt}{6} (K_1 + 2K_2 + 2K_3 + K_4) . \quad (\text{A9})$$

Specifically for our implementation with tensor networks, the derivatives  $K_i$  are obtained via an MPO-MPS contraction with the Hamiltonian MPO  $\hat{H}(t)$  and the MPS representing  $\Psi$ . One can then add a number of MPS in the usual manner [4] to produce the MPS for the evolved solution.

### A.2. TDVP evolution

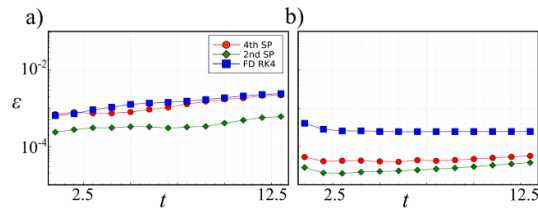
A further evolution scheme available to us is that of TDVP. TDVP is a local time-evolution method which evolves an MPS under the action of some Hamiltonian  $H$ . TDVP works in a similar fashion to the famous DMRG algorithm, where one evolves the first MPS tensor in time by a small amount  $dt$  under a local effective Hamiltonian, before sweeping right across the MPS evolving each tensor and bond in a similar fashion [6, 8]. We implement TDVP by using the code package ITensorTDVP [81], which works with the ITensors library to allow for efficient time-evolution.

To evolve the GPE using TDVP, we first must construct the Hamiltonian as an MPO. The Hamiltonian can be separated as ,

$$H = -\frac{1}{2} \nabla^2 + H_d , \quad (\text{A10})$$

where  $H_d = V(\mathbf{r}) - \mu + g|\Psi|^2$  is a diagonal MPO in real space. One can construct the Laplacian term using both finite difference methods or by exploiting the QFT as follows,

$$\nabla^2 = - \mathcal{F}^{-1} \sum_{\mathbf{k}} \left( k_x^2 + k_y^2 + k_z^2 \right) |\mathbf{k}\rangle \langle \mathbf{k}| \mathcal{F} . \quad (\text{A11})$$



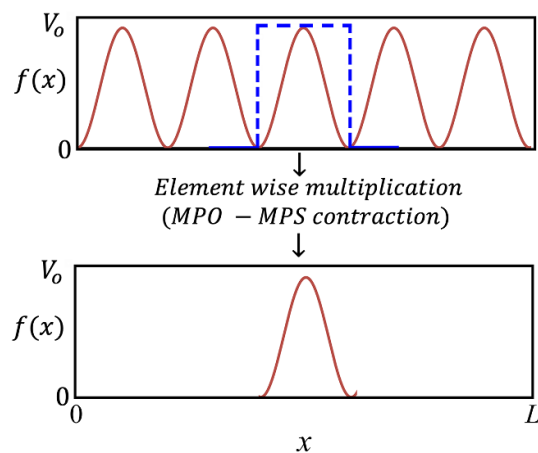
**Figure 25.** (a) Errors from MPS based vortex shedding simulations against XMDS2 direct simulation, for an  $256 \times 256$  spatial grid, with  $\gamma = 0.1$ ,  $\nu = 1.5$  and  $\Delta t = 0.001$ . We compare errors during time evolution using the finite difference RK4 evolution scheme (blue squares), a second order split-step scheme (red dots) and using TDVP (green diamonds). The MPS simulations are run for different truncation cutoffs of (a)  $10^{-12}$ , and (b)  $10^{-15}$ .

We found that TDVP performed using the QFT Laplacian produces smaller errors as compared to the finite difference Laplacian across all times in the simulation, and for all truncation cutoffs. For this reason, whenever using TDVP for full time evolution of the GPE, we constructed the Hamiltonian using the QFT based Laplacian.

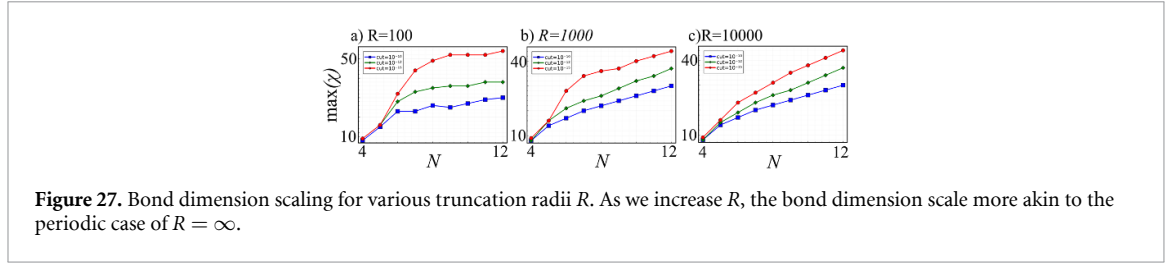
We perform an identical simulation as in section 4.2.1 but using TDVP for the full time evolution, with the resultant errors relative to XMDS2 illustrated in figure 25. We consistently find that we obtain very similar errors for truncation cutoffs of  $10^{-12}$  between TDVP and the split-step schemes, however for a truncation cutoff of  $10^{-15}$  performing a split-step evolution scheme produces a smaller error. Additionally, we always find that TDVP produces smaller errors compared to the finite difference RK4 scheme.

## Appendix B. Paddle generation

To generate the MPS representation of such paddles, we first construct an MPS representation of the top-hat function, and an MPS representation of an extended rounded potential, e.g. a sine wave. One can then multiply these functions together to create a rounded paddle shape directly in MPS form as depicted in figure 26. Note that while this allows for the simulation of the type of strong potentials utilised in recent experiments, we are not implementing any boundary conditions for the quantum fluid at the boundary of the strong potential, e.g. we do not strictly enforce the derivative to go to zero.



**Figure 26.** Method for construction of rounded paddles directly in MPS form. An extended function across the whole simulation domain, such as  $\sin(x)^2$ , is created directly as an MPS, shown in red in the top diagram. A top-hat function is then created as an MPO (shown as dashed blue line) centred on the desired region of the extended function. One then contracts the MPO with the MPS to produce a localised rounded paddle shape as shown in bottom in diagram.



## Appendix C. 2D Dipolar interaction

Given the full 3D dipolar interaction term, it is possible to construct a reduced 2 dimensional model via assumptions on the condensate along the  $z$  directions. The 3D dipolar interaction term is constructed via a convolution, exploiting the Fourier transform,

$$\Phi(\mathbf{r}) = \mathcal{F}^{-1} \left\{ \tilde{U}_{\text{dd}}(\mathbf{k}) \mathcal{F} \left\{ |\Psi(\mathbf{r})|^2 \right\} \right\}, \quad (\text{C1})$$

where  $\tilde{U}_{\text{dd}}(k_x, k_y, k_z)$  is the dipole interaction term in momentum space,

$$\tilde{U}_{\text{dd}}(k_x, k_y, k_z) = 3 \frac{k_y^2}{k^2} - 1. \quad (\text{C2})$$

For the following sections, it is important to realise one can split the Fourier transform separately between different dimensions,

$$\begin{aligned} \mathcal{F} &= \frac{1}{(2\pi)^3} \iiint dx dy dz e^{i(k_x x + k_y y + k_z z)} \\ &= \frac{1}{2\pi} \int dz e^{ik_z z} \mathcal{F}_{2\text{D}} \end{aligned} \quad (\text{C3})$$

### C.1. Translational invariance

If one assumes the condensate is uniform along the  $z$  direction, we can remove the spatial dependence along this direction as,

$$\Psi(x, y, z) = \psi(x, y). \quad (\text{C4})$$

Substituting into equation (C1), and separating the  $z$  components of the Fourier transforms, one obtains,

$$\Phi(\mathbf{r}) = \frac{1}{2\pi} \int dk_z e^{-ik_z z} \left[ \int e^{ik_z z'} dz' \mathcal{F}_{2\text{D}}^{-1} \left\{ \tilde{U}_{\text{dd}}(\mathbf{k}) \mathcal{F}_{2\text{D}} \left\{ |\psi(x', y')|^2 \right\} \right\} \right]. \quad (\text{C5})$$

The above expression can be reduced by means of the Dirac delta function, defined as,

$$\frac{1}{2\pi} \int dx e^{i(k-k_0)x} = \delta(k-k_0). \quad (\text{C6})$$

Substituting into the previous expression yields,

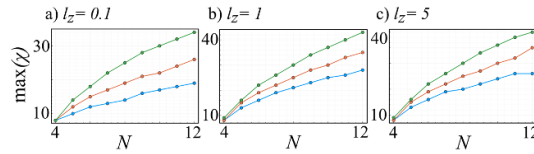
$$\Phi(\mathbf{r}) = \int dk_z \delta(k_z) e^{-ik_z z} \left[ \mathcal{F}_{2\text{D}}^{-1} \left\{ \tilde{U}_{\text{dd}}(\mathbf{k}) \mathcal{F}_{2\text{D}} \left\{ |\psi(x', y')|^2 \right\} \right\} \right], \quad (\text{C7})$$

which can be easily simplified to the reduced 2D dipolar model for a translationally invariant condensate along the  $z$  axis,

$$\Phi(\mathbf{r}) = \mathcal{F}_{2\text{D}}^{-1} \left\{ \tilde{U}_{\text{dd}}^{2\text{D}}(k_x, k_y) \mathcal{F}_{2\text{D}} \left\{ |\psi(x', y')|^2 \right\} \right\}. \quad (\text{C8})$$

The 2D dipolar interaction kernel  $\tilde{U}_{\text{dd}}^{2\text{D}}$  is thus defined as the usual 3D interaction kernel but evaluated only in the  $k_z = 0$  plane,

$$\tilde{U}_{\text{dd}}^{2\text{D}}(k_x, k_y) = \tilde{U}_{\text{dd}}(k_x, k_y, k_z = 0). \quad (\text{C9})$$



**Figure 28.** Maximal bond dimension scaling of  $U_{\text{dd}}$  (equation (27)) for increasing MPS size ( $N$ ) for truncation cutoffs of  $10^{-10}$  (blue),  $10^{-12}$  (orange) and  $10^{-15}$  (green). Results are shown for assuming a strong harmonic trap along the  $z$  direction and for  $l_z$  choices of (a)  $l_z = 0.1$ , (b)  $l_z = 1$  and (c)  $l_z = 5$ .

## C.2. Harmonically trapped

If instead a sufficiently strong harmonic trap is applied along the  $z$  axis, then one can assume that the condensate remains in the groundstate of the harmonic trap along the  $z$  axis, and the dynamics unfold along the  $x$ - $y$  plane [82]. One can express  $\Psi$  as

$$\Psi(x, y, z) = \psi(x, y) \phi(z) , \quad (\text{C10})$$

where  $\phi(z) = \left(\frac{m\omega_z}{\hbar\pi}\right)^{1/4} e^{-\frac{m\omega_z}{2\hbar}z^2}$  is the groundstate along the  $z$  direction.

Substituting into equation (C1), one obtains,

$$\Phi(\mathbf{r}) = \frac{1}{2\pi} \int dk_z e^{-ikz} \left[ \int e^{ik_z z'} \phi(z) dz' \mathcal{F}_{2D}^{-1} \left\{ \tilde{U}_{\text{dd}}(\mathbf{k}) \mathcal{F}_{2D} \left\{ |\psi(x', y')|^2 \right\} \right\} \right] . \quad (\text{C11})$$

Following on from this, it is possible to derive the reduced 2D interaction kernel for this problem [83, 84], given by,

$$U_{\text{dd}}^{2D}(k_x, k_y) = -1 + 3\sqrt{\pi} \frac{q_y^2}{q^2} e^{-q^2} \text{erfc}(q) \quad (\text{C12})$$

where  $\mathbf{q} = \mathbf{k} \frac{l_z}{\sqrt{2}}$ ,  $l_z = \sqrt{\frac{\hbar}{m\omega_z}}$ , and  $\text{erfc}(q)$  is the complimentary error function. Note that it is possible to extend this result to the case when the dipolar polarisation axis is aligned along the  $z$ - $y$  plane, at some angle  $\alpha$  to the  $z$  axis [83, 84],

$$U_{\text{dd}}^{2D} = \cos(\alpha)^2 F_{\perp} + \sin(\alpha)^2 F_{\parallel} , \quad (\text{C13})$$

where  $F_{\perp} = 2 - 3\sqrt{\pi} q e^{q^2} \text{erfc}(q)$  and  $F_{\parallel} = -1 + 3\sqrt{\pi} \frac{q_y^2}{q^2} e^{-q^2}$ .

We again construct the above interaction kernel in momentum space and decompose into MPS form, for various truncation cutoffs and  $l_z$  shown in figure 28. We again observe the same logarithmic scaling with MPS length as was observed in the main text.

## Appendix D. Dipolar interaction truncation

Constructing the interaction kernel as in equation (C1) with the Fourier transform results in a periodic interaction term, and hence the interaction term technically includes the effect of interactions with identical copies of the BEC outside of the original simulation domain. In most cases, this will not provide a substantial impact on the physics, however one may desire to truncate the long range interaction as follows,

$$U_{\text{dd}}^{\mathbf{R}}(\mathbf{r}) = \begin{cases} U_{\text{dd}}(\mathbf{r}), & |\mathbf{r}| < |\mathbf{R}|, \\ 0, & \text{else} \end{cases} \quad (\text{D1})$$

where  $\mathbf{R}$  can be viewed as the truncation radius, beyond which we ignore all interactions. We therefore expect that as  $\mathbf{R}$  tends to infinity, we recover the fully periodic treatment. The form of this truncation is presented in [70] for three dimensions. However, here we restrict to two dimensions by again assuming a uniform condensate along the  $z$  axis. The radius truncated kernel can then be given by

$$\tilde{U}_{\text{dd}}^{\mathbf{R}}(\mathbf{k}) = \frac{1}{2} + 3 \frac{k_y^2 - k_x^2}{k^2} \left[ \frac{1}{2} - \frac{J_1(\mathbf{kR})}{\mathbf{kR}} \right] , \quad (\text{D2})$$

with  $J_1(\cdot)$  being the Bessel function of the first kind.

We construct this interaction potential as an array in momentum space, before performing successive SVDs to put into MPS form. The resultant bond dimension scalings for various truncations and interaction Radii  $\mathbf{R}$  are illustrated in figure 27. We observe that the resultant bond dimensions are larger for smaller truncation radii, and still grow with system size. It can be seen that as  $R \rightarrow \infty$ , we recover the same scaling as the fully periodic treatment in figure 22.

## ORCID iD

Ryan J J Connor  0009-0001-5292-8391

## References

- [1] Biamonte J D and Bergholm V 2017 arXiv: Quantum physics (available at: <https://api.semanticscholar.org/CorpusID:119679246>)
- [2] Orús R 2019 *Nat. Rev. Phys.* **1** 538–50
- [3] Orús R 2014 *Ann. Phys., NY* **349** 117–58
- [4] Schollwöck U 2011 *Ann. Phys., NY* **326** 96–192
- [5] Chan G K-L, Keselman A, Nakatani N, Li Z, and White S R 2016 Matrix product operators, matrix product states, and ab initio density matrix renormalization group algorithms (arXiv:1605.02611 [physics.chem-ph])
- [6] Paeckel S, Köhler T, Swoboda A, Manmana S R, Schollwöck U and Hubig C 2019 *Ann. Phys., NY* **411** 167998
- [7] Vanderstraeten L, Haegeman J and Verstraete F 2019 *SciPost Phys. Lect. Notes* **7** 1–77
- [8] Haegeman J, Lubich C, Oseledets I, Vandereycken B and Verstraete F 2016 *Phys. Rev. B* **94** 165116
- [9] Evenbly G and Vidal G 2011 *J. Stat. Phys.* **145** 891–918
- [10] Cichocki A 2014 Tensor networks for big data analytics and large-scale optimization problems (arXiv:1407.3124)
- [11] Schuch N, Wolf M M, Verstraete F and Cirac J I 2008 *Phys. Rev. Lett.* **100** 030504
- [12] Iblisdir S, Orús R and Latorre J I 2007 *Phys. Rev. B* **75** 104305
- [13] Hong R, Xiao Y-X, Hu J, Ji A-C and Ran S-J 2022 *Phys. Rev. B* **105** 165116
- [14] Gourianov N, Lubasch M, Dolgov S, van den Berg Q Y, Babaee H, Givi P, Kiffner M and Jaksch D 2022 *Nat Comput Sci* **2** 30–37
- [15] Kiffner M and Jaksch D 2023 (arXiv:2303.03010)
- [16] Ye E and Loureiro N F G 2022 *Phys. Rev. E* **106** 035208
- [17] Lubasch M, Moinier P and Jaksch D 2018 *J. Comput. Phys.* **372** 587–602
- [18] Ye E and Loureiro N, 2024 Quantized tensor networks for solving the Vlasov-Maxwell equations (arXiv:2311.07756 [physics.comp-ph])
- [19] Gourianov N, Givi P, Jaksch D, and Pope S B 2024 Tensor networks enable the calculation of turbulence probability distributions (arXiv:2407.09169 [physics.flu-dyn])
- [20] Hölscher L, Rao P, Müller L, Klepsch J, Luckow A, Stollenwerk T, and Wilhelm F K 2024 Quantum-inspired fluid simulation of 2d turbulence with GPU acceleration (arXiv:2406.17823 [physics.flu-dyn])
- [21] Pethick C 2008 *Bose-Einstein Condensation in Dilute Gases* 2nd edn (Cambridge University Press)
- [22] Pitaevskii L and Stringari S 2016 *Bose-Einstein Condensation and Superfluidity* (Oxford University Press) pp 42–65
- [23] Proukakis N, Gardiner S, Davis M and Szymańska M 2013 *Quantum Gases* Imperial College Press (available at: <https://arxiv.org/abs/www.worldscientific.com/doi/pdf/10.1142/p817>)
- [24] Blakie P, Bradley A, Davis M, Ballagh R and Gardiner C 2008 *Adv. Phys.* **57** 363
- [25] Fischer T Z and Bradley A S 2024 Regimes of steady-state turbulence in a quantum fluid (arXiv:2409.03184 [cond-mat.quant-gas])
- [26] Kobayashi M, Parnaudeau P, Luddens F, Lothodé C, Danaila L, Brachet M and Danaila I 2021 *Comput. Phys. Commun.* **258** 107579
- [27] Gauthier G, Reeves M T, Yu X, Bradley A S, Baker M A, Bell T A, Rubinsztein-Dunlop H, Davis M J and Neely T W 2019 *Science* **364** 1264–7
- [28] Baggaley A W and Parker N G 2018 *Phys. Rev. A* **97** 053608
- [29] Hernáandez-Rajkov D et al 2024 *Nat. Phys.* **20** 939–44
- [30] Poli E, Baillie D, Ferlaino F, and Blakie P B 2024 Excitations of a two-dimensional supersolid (arXiv:2407.01072 [cond-mat.quant-gas])
- [31] Ripley B T E, Baillie D and Blakie P B 2023 *Phys. Rev. A* **108** 053321
- [32] Bland T, Lamporesi G, Mark M J and Ferlaino F 2024 *Comptes Rendus. Phys.* **24** 133–52
- [33] Sabari S, Kishor Kumar R and Tomio L 2024 *Phys. Rev. A* **109** 023313
- [34] Sachkou Y P et al 2019 *Science* **366** 1480–5
- [35] Lim Y L, Lee Y, Goo J, Bae D and Shin Y 2022 *New J. Phys.* **24** 083020
- [36] Navon N, Eigen C, Zhang J, Lopes R, Gaunt A L, Fujimoto K, Tsubota M, Smith R P and Hadzibabic Z 2019 *Science* **366** 382–5
- [37] Norcia M A, Politi C, Klaus L, Poli E, Sohmen M, Mark M J, Bisset R N, Santos L and Ferlaino F 2021 *Nature* **596** 357–61
- [38] Tanzi L, Maloberti J G, Biagioni G, Fioretti A, Gabbanini C and Modugno G 2021 *Science* **371** 1162–5
- [39] Cabrera C R, Tanzi L, Sanz J, Naylor B, Thomas P, Cheiney P and Tarruell L 2018 *Science* **359** 301–4
- [40] Bigagli N, Yuan W, Zhang S, Bulatovic B, Karman T, Stevenson I and Will S 2024 *Nature* **631** 289–93
- [41] Chen J, Stoudenmire E and White S R 2023 *PRX Quantum* **4** 040318
- [42] Sulem C and Sulem P-L 2007 *The Nonlinear Schrödinger Equation: Self-Focusing and Wave Collapse* vol 139 (Springer)
- [43] Mewes M-O, Andrews M R, van Druten N J, Kurn D M, Durfee D S, Townsend C G and Ketterle W 1996 *Phys. Rev. Lett.* **77** 988
- [44] Jin D S, Matthews M R, Ensher J R, Wieman C E and Cornell E A 1997 *Phys. Rev. Lett.* **78** 764
- [45] Choi S, Morgan S A and Burnett K 1998 *Phys. Rev. A* **57** 4057
- [46] Pitaevskii L 1959 *Sov. Phys. -JETP* **8** 282
- [47] Kazeev V A and Khoromskij B N 2012 *SIAM J. Matrix Anal. Appl.* **33** 742
- [48] Fornberg B 1988 *Math. Comput.* **51** 699
- [49] Javanainen J and Ruostekoski J 2006 *J. Phys. A: Math. Gen.* **39** L179
- [50] Feit M, Fleck Jr J and Steiger A 1982 *J. Comput. Phys.* **47** 412
- [51] Hatano N and Suzuki M 2005 Finding exponential product formulas of higher orders *Lecture Notes in Physics* (Springer) pp 37–68
- [52] Nielsen M A and Chuang I L 2001 *Quantum Computation and Quantum Information* vol 2 (Cambridge University Press)

- [53] Tyson J 2002 *J. Phys. A: Math. Gen.* **36** 6813
- [54] Nielsen M A, Dawson C M, Dodd J L, Gilchrist A, Mortimer D, Osborne T J, Bremner M J, Harrow A W and Hines A 2003 *Phys. Rev. A* **67** 052301
- [55] Gong Z-X, Foss-Feig M, Brandão F G S L and Gorshkov A V 2017 *Phys. Rev. Lett.* **119** 050501
- [56] Press W H, Teukolsky S A, Vetterling W T and Flannery B P 2007 *Numerical Recipes 3rd Edn: The Art of Scientific Computing* 3rd edn (Cambridge University Press) pp 907–10
- [57] Zakharov V E and Shabat A B 1970 *J. Exp. Theor. Phys.* **34** 62
- [58] Salasnich L 2017 *Opt. Quantum Electron.* **49** 409
- [59] Kwon W J, Moon G, Seo S W and Shin Y 2015 *Phys. Rev. A* **91** 053615
- [60] Onofrio R, Raman C, Vogels J M, Abo-Shaeer J R, Chikkatur A P and Ketterle W 2000 *Phys. Rev. Lett.* **85** 2228
- [61] Raman C, Köhl M, Onofrio R, Durfee D S, Kuklewicz C E, Hadzibabic Z and Ketterle W 1999 *Phys. Rev. Lett.* **83** 2502
- [62] Neely T W, Samson E C, Bradley A S, Davis M J and Anderson B P 2010 *Phys. Rev. Lett.* **104** 160401
- [63] Kwon W J, Kim J H, Seo S W and Shin Y 2016 *Phys. Rev. Lett.* **117** 245301
- [64] Reeves M T, Billam T P, Anderson B P and Bradley A S 2015 *Phys. Rev. Lett.* **114** 155302
- [65] Matthews M R, Anderson B P, Haljan P C, Hall D S, Wieman C E and Cornell E A 1999 *Phys. Rev. Lett.* **83** 2498
- [66] Dennis G R, Hope J J and Johnsson M T 2013 *Comput. Phys. Commun.* **184** 201
- [67] Reeves M T, Billam T P, Anderson B P and Bradley A S 2014 *Phys. Rev. A* **89** 053631
- [68] Smith B D, Cooke L W and LeBlanc L J 2022 *Comput. Phys. Commun.* **275** 108314
- [69] Tang Q, Zhang Y and Mauser N J 2017 *Comput. Phys. Commun.* **219** 223
- [70] Smith J C, Baillie D and Blakie P B 2023 *Phys. Rev. A* **107** 033301
- [71] Lobo C, Sinatra A and Castin Y 2004 *Phys. Rev. Lett.* **92** 020403
- [72] Tsubota M, Kasamatsu K and Ueda M 2002 *Phys. Rev. A* **65** 023603
- [73] Kasamatsu K, Tsubota M and Ueda M 2003 *Phys. Rev. A* **67** 033610
- [74] Tsuzuki S 2023 *Phys. Fluids* **35** 047102
- [75] Klaus L, Bland T, Poli E, Politi C, Lamporesi G, Casotti E, Bisset R N, Mark M J and Ferlaino F 2022 *Nat. Phys.* **18** 1453–8
- [76] Prasad S B, Bland T, Mulkerin B C, Parker N G and Martin A M 2019 *Phys. Rev. A* **100** 023625
- [77] Lagaris I, Likas A and Fotiadis D 1998 *IEEE Trans. Neural Netw.* **9** 987
- [78] Cai S, Mao Z, Wang Z, Yin M, and Karniadakis G E, 2021 Physics-informed neural networks (pinns) for fluid mechanics: a review (arXiv:2105.09506 [physics.flu-dyn])
- [79] Cai S, Wang Z, Wang S, Perdikaris P and Karniadakis G E 2021 *J. Heat Transfer* **143** 060801
- [80] Williamson L A and Blakie P B 2016 *Phys. Rev. Lett.* **116** 025301
- [81] Fishman M, White S R and Stoudenmire E M 2022 *SciPost Phys. Codebases* **4**
- [82] Cai Y, Rosenkranz M, Lei Z and Bao W 2010 *Phys. Rev. A* **82** 043623
- [83] Ticknor C, Wilson R M and Bohn J L 2011 *Phys. Rev. Lett.* **106** 065301
- [84] Mulkerin B C, van Bijnen R M W, O'Dell D H J, Martin A M and Parker N G 2013 *Phys. Rev. Lett.* **111** 170402
- [85] Connor R, Duncan C and Daley A 2025 Data for ‘Tensor network methods for the Gross-Pitaevskii equation on fine grids’ (<https://doi.org/10.15129/74c84249-5aeb-477e-ac24-83d983687bd3>)