

Efficient simulation of Bose-Einstein condensates in nontrivial topologies

A. Beregi , J.-B. Gerent , and N. Lundblad ^{*}*Department of Physics and Astronomy, Bates College, Lewiston, Maine 04240, USA*

(Received 4 March 2026; accepted 11 September 2026; published 15 October 2026)

Bubble-shaped Bose-Einstein condensates (BECs) constitute a unique class of quantum fluids with a hollow, thin-shell geometry that supports a wide variety of phenomena distinct from those of compact condensates. Numerical simulation of such systems is particularly challenging due to their inherently three-dimensional structure and extreme aspect ratios. We present an efficient finite-difference simulation framework for solving partial differential equations in nontrivial topologies, with a focus on the static and dynamical modeling of bubble-shaped BECs. By employing selective spatial sampling on a semistructured grid, our method substantially reduces memory usage and achieves more than an order-of-magnitude improvement in computational performance compared with conventional split-step Fourier solvers. The algorithm is naturally extendable to highly parallel execution on graphics processing units (GPUs), enabling large-scale, time-dependent simulations of thin-shell condensates. We apply this framework to simulate the formation of bubble BECs through a controlled hollowing-out protocol using *ab initio* trapping potentials relevant to the Cold Atom Laboratory aboard the International Space Station. From these simulations, we identify characteristic timescales and parameter ramps required to achieve adiabatic evolution, thereby assessing the feasibility of experimentally realizing bubble-shaped condensates in microgravity environments.

DOI: [10.1103/j2db-xh7h](https://doi.org/10.1103/j2db-xh7h)

I. INTRODUCTION

Bose-Einstein condensates (BECs), as routinely achieved in ultracold atomic physics experiments, provide a fertile platform for the study of quantum fluids and macroscopic quantum phenomena, often through the exploration of dimensionality [1–3], geometry [4,5], topology [6–8], spin-orbit coupling [9], and interactions [10,11] of the quantum-gas samples under study. In most experiments, BECs are confined in harmonic or box traps, resulting in compact, topologically equivalent systems that can be deformed continuously into one another. However, recent advancements in experimental techniques, both in microgravity environments such as those achieved in Earth-orbiting laboratories [12,13] and in terrestrial efforts [14–16], have enabled the realization of bubble-shaped systems, notable for being topologically distinct in real space from those produced using typical ultracold atomic trapping approaches. Bubble- or shell-like trapping geometries give rise to unique collective excitations [17–20], vortex phenomena [21–25], thermodynamic corrections due to curvature [26–31], and novel examples of nonlinear wave propagation [32,33]. More generally, these geometries offer a new framework for studying quantum-gas physics on curved manifolds [34–37].

Numerical simulation has been crucial to the development of the field, through studies of the time-independent and time-dependent Gross-Pitaevskii equations (GPEs); numerics are most commonly used to predict ground-state energies and wavefunctions in analytically intractable situations, and to reveal dynamics including collective modes and vortices. Recent technological advances such as the rapid growth in massively parallel graphics processing unit (GPU)-enabled computation have significantly expanded simulation capabilities. However, bubble-geometry quantum-gas systems remain challenging to simulate due to their unique hollow geometrical configuration with large spatial extent and strong local anisotropy. While strongly anisotropic compact systems can be efficiently simulated on anisotropic Cartesian grids, bubble-shaped systems (or any arrangement that evokes the embedding of one-dimensional or two-dimensional (2D) structures within a three-dimensional volume) result in a large fraction of the grid being wasted. In some special cases, such as radial symmetry or quasi-2D confinement, the problem is simplified through dimensionality reduction; in general, however, these simplifying limits are not realistically accessible in experiments, and the numerics of accurate BEC simulation remain challenging. As an example, an anisotropic bubble with micrometer-scale thickness and 100- μm -scale radius naively requires a Cartesian grid with 8×10^9 elements with corresponding ~ 120 GB of memory to store the wavefunction, which continues to be impractical outside of high-performance computing facilities, despite ongoing technological advances in the computational infrastructure of personal workstations.

In this work, we present a general scheme for addressing this issue without sacrificing accuracy; our approach can be

^{*}Contact author: nlundbla@bates.edu

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

used for simulations of realistic, anisotropic bubbles in situations (such as inflation and hollowing-out of a bubble) where a quasi-2D condition does not always hold. Our methods are applicable to both finding the ground state and evolving a wavefunction in an arbitrary time-dependent potential. Owing to their generality, they are not restricted to bubble-shaped traps, providing a speedup of simulations in any scenario where an exotic geometry is involved. Furthermore, the core idea of our method is not limited to the GPE and is easily adaptable for numerical studies of other differential equations.

This work is organized as follows: In Sec. II, we discuss the simulation methods developed, followed by their numerical verification in Sec. III; in Sec. IV, we evaluate the performance of the new approach; in Sec. V, we present applications of the technique to a physically relevant situation: the inflation of a bubble-geometry BEC; and in Sec. VI, we conclude and provide an outlook for future work.

II. SIMULATION METHODS

In a given potential, the equilibrium state and subsequent dynamics are described by the time-independent and time-dependent Gross-Pitaevskii equations

$$-\frac{\hbar^2}{2m}\nabla^2\psi + V(\mathbf{r})\psi + g|\psi|^2\psi = \mu\psi, \quad (1)$$

$$i\hbar\frac{\partial\psi}{\partial t} = -\frac{\hbar^2}{2m}\nabla^2\psi + V(\mathbf{r}, t)\psi + g|\psi|^2\psi, \quad (2)$$

where $\hbar$ is the reduced Planck constant, m is the particle mass, $V(\mathbf{r}, t)$ is the external potential, $g = 4\pi\hbar^2 a/m$ is the interaction constant for s -wave scattering length a , μ is the chemical potential, and ψ is the macroscopic wavefunction of the condensate [38]. The wavefunction obeys the normalization condition

$$\int |\psi|^2 dV = N, \quad (3)$$

where N is the number of particles. These equations arise from the mean-field treatment of the full quantum many-body problem and serve as the primary theoretical tool for the study of dilute-gas BECs.

The time-independent GPE is usually solved with the imaginary-time propagation technique by the transformation $t \rightarrow -it$ in Eq. (2) [39]. This results in nonunitary evolution, which attenuates the excited states at rates proportional to their energies. Therefore, by successive evolution for a short time δt followed by the renormalization of ψ by imposing Eq. (3), the wavefunction converges to the ground state. The main run-time-critical component, both in ground-state calculations and in time evolution, is the evaluation of the kinetic-energy term involving the Laplace operator. This can be achieved by transforming to the basis of momentum eigenstates in which the Laplace operator is diagonal; thus, the corresponding evolution operator is easy to evaluate. The change of basis is performed efficiently by discrete Fourier transforms, having favorable $\mathcal{O}(N \log N)$ scaling with input size. Since the kinetic-energy operator does not commute with the potential and interaction energy operators, the time evolution is typically implemented using a second-order Suzuki-Trotter decomposition. Within this scheme,

imaginary-time propagation is summarized as follows:

$$\psi e^{-\mathcal{H}_V \delta t / 2\hbar} \xrightarrow{\text{FFT}} \psi_k e^{-\mathcal{H}_T \delta t / \hbar} \xrightarrow{\text{iFFT}} \psi e^{-\mathcal{H}_V \delta t / 2\hbar}, \quad (4)$$

where $\mathcal{H}_{V,T}$ are the terms in the Hamiltonian diagonal in position and momentum space, respectively, and ψ_k is the wavefunction in the momentum basis. For simulating dynamics, one can employ the same method, but with the unitary time-evolution operators $e^{-i\mathcal{H}_V \delta t / 2\hbar}$ and $e^{-i\mathcal{H}_T \delta t / \hbar}$ in Eq. (4), where, in general, $\mathcal{H}_V$ is time dependent. As discrete Fourier transforms are an integral part of this approach, they are limited to simulation volumes that are rectangular cuboids.

However, bubble-shaped condensates typically occupy a small volume fraction of rectangular simulation grids, resulting in inefficient use of computational resources. Furthermore, the unique topology of a bubble results in the need for small spatial step sizes, as typical clouds are thin in all spatial directions. This is in stark contrast with the case of compact but highly anisotropic systems, for example, cigar- or pancake-shaped BECs, where choosing different spatial step sizes along the strongly and weakly confined directions improves the efficiency of simulation. A further limitation of the split-step Fourier scheme is the requirement for the differential equation to be homogeneous, which is not the case for an extended class of problems such as the stochastic GPE [40,41] or tunnel-coupled BECs [42,43]. A possible solution is to build a mesh around the relevant spatial region, as often done in finite-element simulations, used for instance in computational fluid dynamics, solid mechanics, or electromagnetism. This results in an unstructured grid, where the evaluation of the Laplacian is computationally expensive due to the irregular memory layout of neighboring grid points and nonconstant stencil weights. In our approach, we combine these two methods by using a semistructured Cartesian grid, defined only near the minimum of the potential. Within this scheme, we significantly exploit the local regularity of the grid, combined with the efficient use of simulation resources, by defining a region of interest (ROI). We developed two algorithms: a general scheme that is optimized for execution on CPU-based systems using a single thread and a highly parallel algorithm for execution on a GPU.

A. General algorithm

We start with a standard Cartesian grid and estimate the size of the condensate by the Thomas-Fermi (TF) approximation:

$$|\psi_{\text{TF}}|^2 = \max\left(\frac{\mu - V(\mathbf{r})}{g}, 0\right), \quad (5)$$

where μ is the chemical potential. This result is obtained by neglecting the kinetic-energy term in Eq. (1). We define the region of interest as the grid points at which the TF estimate yields nonzero density. To ensure that the simulated condensate is almost fully within the TF region, we set the value of the chemical potential to an unrealistically high value. As the value of such a chemical potential might not be known *a priori*, we calculate the number of atoms in the TF trial solution corresponding to an initial guess for μ and adjust its value to ensure that the trial solution corresponds to an unrealistically high atom number (e.g., an order of magnitude

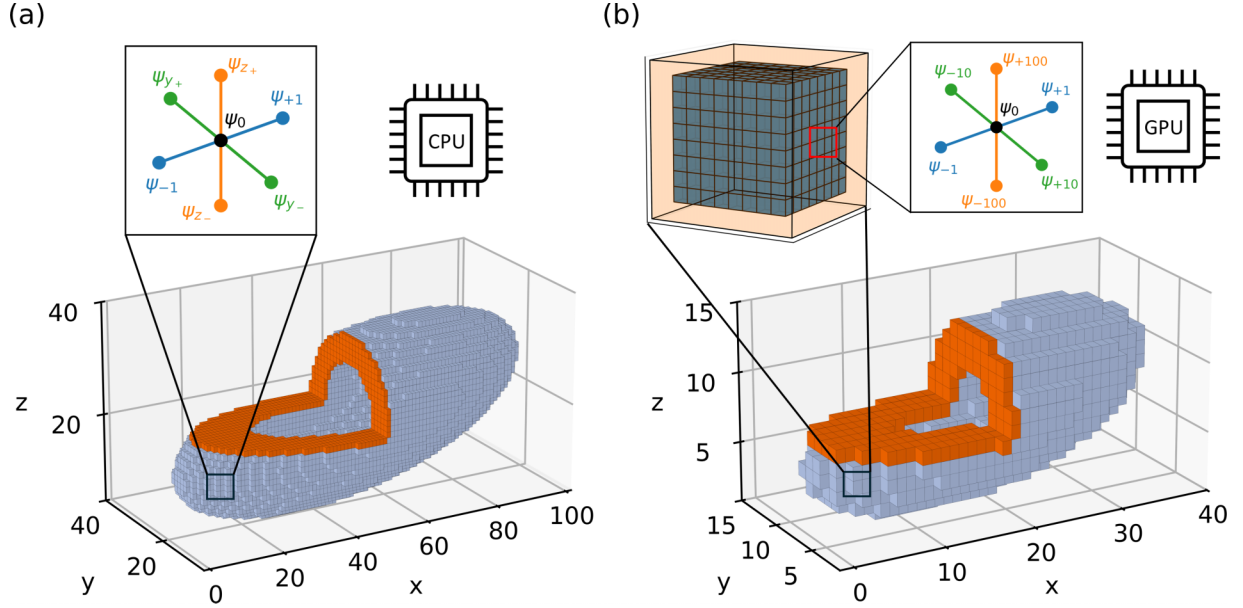


FIG. 1. Illustration of the semistructured grid using a Thomas-Fermi trial solution with $\mu/\hbar = 450$ Hz in a typical bubble trap potential, corresponding to $\sim 9 \times 10^5$ particles. For visualization purposes, a quarter of the grid is cut out and the resulting boundary is colored orange. (a) General scheme resulting in fine-grained postselection of grid points. The local structure of the grid is shown in the inset, with only the nearest neighbors along x at deterministic offsets (± 1) from the central point. The axes are labeled in grid-point units. (b) Parallel, GPU-optimized scheme resulting in coarse-grained postselection of 8^3 blocks of grid points supporting storage in shared memory. The axes are labeled in block units. The inset shows the structure of a single block, where the halo is shown with orange shading. Due to the coarser graining, the local structure of the grid is highly ordered, with all nearest neighbors at deterministic offsets ($\pm 1, \pm 10, \pm 100$) in shared memory.

higher than the atom number used in the actual simulation). An example of a simulation region is shown in Fig. 1(a). Next, the Laplace operator is constructed for the standard Cartesian grid using a finite-difference approach from outer products of identity matrices and tridiagonal matrices corresponding to second-order central approximations to second derivatives. Since the resulting Laplacian involves a seven-point stencil, for N grid points, the fraction of nonzero entries ignoring the edges is $7/N$, resulting in very sparse matrices. We then flatten the wavefunction and the potential array to a vector of N elements and identify the set of indices where the TF trial solution yields zero density, i.e., those lying outside the ROI, denoted by $\overline{\text{ROI}}$. The wavefunction and potential vectors are reduced by removing elements at all indices $i \in \overline{\text{ROI}}$, giving $\tilde{\psi}$ and $\tilde{V}$. Corresponding to the vector reduction operation, we remove all rows and columns of the Laplacian matrix with indices $i \in \overline{\text{ROI}}$ and denote the resulting operator by $\tilde{T}$, which includes the necessary prefactor $-\hbar^2/2m$. This operation is mathematically justified since we assume that the density is zero at these grid points and will stay zero during evolution in real or imaginary time. To calculate ground states, we numerically solve

$$\frac{\partial \tilde{\psi}}{\partial t} = (-1/\hbar)(\tilde{T} + \tilde{V} + g|\tilde{\psi}|^2)\tilde{\psi}, \quad (6)$$

with the renormalization of the wavefunction after each temporal step. Similar to the standard split-step Fourier approach, we first propagate for $\delta t/2$ using the operators that are diagonal in position space:

$$\tilde{\psi} \rightarrow \tilde{\psi} e^{-(\tilde{V} + g|\tilde{\psi}|^2)\delta t/2\hbar}. \quad (7)$$

This operation can be performed efficiently by precomputing $e^{-\tilde{V}\delta t/2\hbar}$. Furthermore, since typical bubble-shaped condensates are dilute, the interaction propagator can be approximated as $1 - g|\tilde{\psi}|^2\delta t/2\hbar$, avoiding the computationally expensive evaluation of the exponential function. Next, the kinetic-energy propagation is performed for a total time of δt in two steps using the forward Euler method [44]. Subsequently, we perform one more propagation step given by Eq. (7) and renormalize the wavefunction to the desired number of particles. Since imaginary time evolution under the Laplacian is treated in the finite-difference fashion, this approach does not require the wavefunction to be complex, providing an intrinsic speedup and a factor-of-2 improvement in memory efficiency compared to the conventional algorithm.

For time evolution, we numerically solve

$$\frac{\partial \tilde{\psi}}{\partial t} = (-i/\hbar)(\tilde{T} + \tilde{V}(t) + g|\tilde{\psi}|^2)\tilde{\psi}, \quad (8)$$

using various explicit Runge-Kutta methods, without employing the split-step procedure. In particular, we use Heun's third-order method (RKHE3), the classic fourth-order method (RK4) [45], and the fifth-order Cash-Karp method (RKCK) [46].

The algorithms are implemented using high-level libraries for sparse matrix-vector multiplication (SpMV) such as SciPy or CuPy and with a highly optimized custom SpMV kernel written in the C programming language. For all implementations, we employ the compressed sparse row format, which is generally faster for the required SpMV operation. The custom SpMV kernel exploits two properties of the Laplacian. First, for cubical grids, its matrix elements only have two unique

values, which reduces the required memory accesses. In fact, the diagonal part of the Laplacian can be encoded in the potential as a constant offset, which results in the elements of the remaining reduced matrix to be constant. Second, as the grid is semistructured, for the majority of the grid points, there are six nearest neighbors and we refer to these as normal grid points. Such grid points have irregular nearest-neighbor offsets along the two slower-changing directions, but along the fastest-changing direction they are deterministically ± 1 , resulting in better cache locality and memory access patterns. Moreover, a very large fraction of normal grid points also have normal neighbors, since the grid is locally regular. Therefore, for each normal grid point, if the next three are also normal, we process these together as the offsets along the two slower changing directions are the same for these points. For anisotropic grids, discussed in Appendix B, one can still exploit the local structure of the grid, but the stencil weights no longer remain constant.

B. Parallel algorithm

Similar to the previous algorithm, we start with a TF trial solution. However, instead of performing the postselection of individual grid points, we calculate the number of atoms in blocks of size 8^3 , as illustrated in Fig. 1(b). The choice of block size was motivated by the typical maximum number of threads per block on CUDA-compatible GPUs. If a block contains nonzero atoms, we add all of its grid points to the ROI and record their indices. For each block, we also record the indices of a shell of grid points around the block, which is often referred to as the halo; these points are required for consistent evaluation of the Laplacian at the boundaries of the block. This coarse-grained postselection procedure requires the grid to have $N_i \equiv 2 \pmod{8}$ points along directions $i = x, y, z$ and results in an $N_b \times 1000$ array of grid indices, denoted as M_{ind} . Note that by this coarse procedure, the ROI necessarily contains more grid points compared to the previous algorithm. This effect can be mitigated by choosing smaller blocks, but this results in a larger fraction of the grid points being reused as halo [47].

To improve global memory access patterns, we sort these grid indices in the following way. We ensure that elements of each block are stored contiguously in memory with the first index varying the fastest, such that the elements of the n th block are stored at offsets $[512 \times (n - 1), 512 \times n]$. This results in deterministic memory addresses for each array element in the blocks; however, block halo points remain at nondeterministic offsets. Therefore, for each block, we record an integer array of length 488 that contains the indices of its halo in an ordered fashion. The majority of these halo points are members of the other adjacent blocks; thus, they are stored in memory at offsets $[0, 512 \times N)$ for N blocks. However, some halo points of blocks at the boundary of the simulation region are not included; these are collectively moved to memory offsets above $512 \times N$. Next, we identify the unique indices in M_{ind} , perform the reduction of the wavefunction and the potential array, and finally remap the elements of M_{ind} to account for the ignored grid points and the grid reordering.

After these preparation steps, we proceed to solving the differential equation. The imaginary time propagation algo-

rithm is performed using three GPU kernels. The first kernel loads the 1000 values of the wavefunction for a given block and its halo into shared GPU memory. During the loading operation, each array element is multiplied by the propagator corresponding to diagonal operators given in Eq. (7), as well as a constant scaling factor that renormalizes the wavefunction from the previous iteration. The kernel then evaluates the first forward-Euler step in each central 8^3 block using the known constant offsets of the required Laplacian stencil. The halo allows the use of higher-order compact (HOC) stencils of size 3^3 , increasing the accuracy and isotropy of the evaluation of the Laplacian. Finally, the wavefunction in the central 8^3 block is written to an empty array in global memory, storing the temporary wavefunction. The second kernel loads the temporary wavefunction for the 1000 grid points in each block, applies the second forward-Euler step, and performs the second propagation step given in Eq. (7). Finally, the third kernel calculates the number of particles by reduction and the scaling factor that is used to renormalize the wavefunction in the next iteration.

Similar to the general algorithm, the time-evolution simulation is performed with explicit Runge-Kutta methods. However, due to the block-based postselection procedure, a modified, memory-optimized Runge-Kutta procedure is employed. In a standard Runge-Kutta procedure of n stages, the right-hand side of the differential equation would be evaluated n times according to

$$k_i = f(t + c_i \delta t, y^{(i)}), \quad (9)$$

where $y^{(i)} = y_0 + \delta t \sum_{j < i} a_{ij} k_j$ is the state of the system at the previous intermediate step, with a_{ij} being method-specific constants, and the results k_i are stored. After all stages have completed, the update of the system state at the next time step is calculated as a linear combination of k_i . For our block structure, this leads to poor usage of global memory, as for stage n , for the evaluation of the right-hand side at 512 grid points, we have to load and compute the linear combination of $y_0, k_1, k_2, \dots, k_{n-1}$, amounting to $n \times 1000$ values. To reduce global memory usage, we store the wavefunction at the intermediate stages $y^{(i)}$, which we compute from the linear combination of earlier stages $y^{(j < i)}$ and a single evaluation of the right-hand side at the current stage. The coefficients required for this are straightforward to derive from the coefficients of the standard approach (see Appendix A). In addition, the last kernel computes the appropriate linear combination of $\psi(t)$ and y_i to obtain $\psi(t + \delta t)$. The usage of this modified scheme provides 7.5%, 12%, and 44% less global memory traffic for the employed third-, fourth-, and fifth-order integrators, respectively.

III. ALGORITHM VERIFICATION

We verify the validity of our methods and the underlying approximations by comparing the numerical GPE ground-state solutions to those from a split-step Fourier solver. The simulation error has two sources: the finite-difference approximation of the derivatives and the irregular boundary of the ROI. To investigate the first source of error, we compare ground-state solutions calculated by the finite-difference method using the full, rectangular grid with those obtained

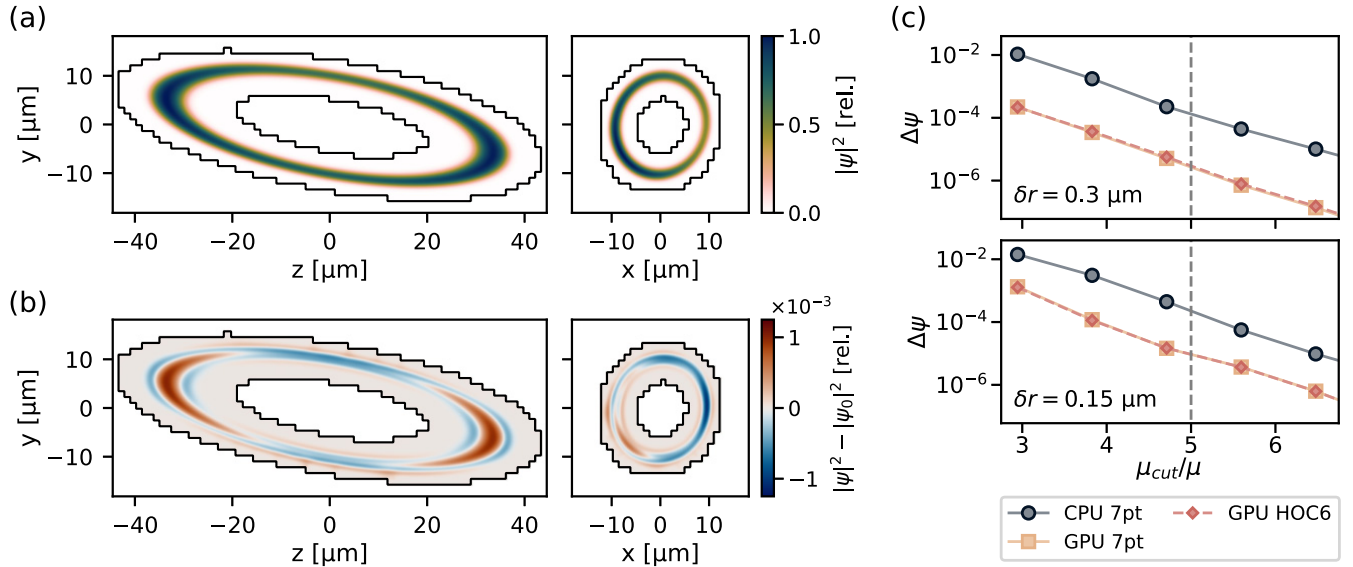


FIG. 2. Verification of the semistructured grid method with GPE ground states. (a) GPE ground-state density in a typical bubble potential, along the y-z and x-y planes. The solid black line shows the boundary of the region of interest. (b) Simulation errors for the semistructured grid method, shown along the y-z and x-y planes. (c) Scaling of simulation error with grid selection cutoff for the coarser grid with $\delta r = 0.3 \mu\text{m}$ (top) and the finer grid with $\delta r = 0.15 \mu\text{m}$ (bottom), for various Laplacian stencils. The gray dashed line shows the choice of chemical potential cutoff used in this work.

from the split-step Fourier solver. The potential was chosen to be a realistic radio frequency (rf)-dressed bubble potential as realized on the Cold Atom Laboratory (CAL) atom chip setup [48], and we chose the number of particles to be $N = 10^5$. We used a grid size of $120 \times 120 \times 300$, with grid spacing of $0.3 \mu\text{m}$ along all three axes, as well as a twice as dense, $240 \times 240 \times 600$ grid with spacing $0.15 \mu\text{m}$. The reference wavefunction was obtained using the split-step Fourier method until a relative tolerance of 10^{-9} was reached. The finite-difference wavefunction was computed using the same relative tolerance. The error was quantified as $\Delta\psi = 1/N \int (|\psi_0|^2 - |\psi|^2) dV$, where ψ_0 is the reference wavefunction and ψ is the wavefunction calculated by the finite-difference method. Using the 27-point stencil for the finite-difference Laplacian, the resulting error was 2.8×10^{-3} and 6.9×10^{-4} for the coarser and finer grids, respectively. The simpler, seven-point stencil yields 2×10^{-3} and 5.6×10^{-4} . This demonstrates that using the finite-difference method, the simulation error remains small for the employed grid sizes. Furthermore, from the scaling of the error with spatial step size, we conclude that the 27-point stencil performs better for progressively smaller values of the grid spacing.

Next, we compare the reference wavefunctions to those obtained from the general and parallelized semistructured grid methods. In Fig. 2(a), we show the ground-state density in a typical bubble potential obtained with the semistructured grid method along two orthogonal coordinate planes together with the boundary of the ROI. For this simulation, the ROI was set such that the ratio of the chemical potential cutoff and the chemical potential of the BEC is 5, providing sufficient clearance along all spatial directions. In Fig. 2(b), we show the simulation error along the same planes, which does not exceed the 10^{-3} level and shows the same general structure as the error for the finite-difference simulation performed over

the full grid. To investigate the effect of the boundary, the same calculation is repeated for a range of ROIs defined by different chemical potentials for the TF trial solution used for the grid postselection procedure. We plot the resulting errors for both the coarser and finer grids in Fig. 2(c) for various chemical potential cutoffs and Laplace operator stencils. For all cases considered, the data show rapid convergence as the ROI size is increased. For both spatial step sizes, the GPU algorithm performed better, which is due to the inherently larger simulation region. The simulation error is almost independent of the spatial step size, indicating that the source is predominantly the boundary. This can be easily mitigated by increasing the grid selection threshold μ_{cut} , which reduces the error exponentially [Fig. 2(c)]. For the ROI sizes used in this work, the simulation error is on the 10^{-5} level, which is negligible compared to the error due to the finite-difference approximations.

We evaluate the numerical performance of time-evolution simulations by propagating the ground-state solution obtained using the semistructured grid method for $t_{\text{max}} = 800 \text{ ms}$ —the longest evolution time considered in this work. We use an RK4 integrator with a time step of $2 \mu\text{s}$ and the 27-point stencil for the finite-difference evaluation of the Laplacian. The evolution is characterized by two figures of merit, conservation of particle number (unitarity) and fidelity between the time-evolved state and the ground state. For unitarity, we report $1 - N(t_{\text{max}})/N_0$, where N_0 is the initial atom number. Since the fidelity is oscillatory in time, we report the time average of $1 - |\langle \psi(t) | \psi_0 \rangle|^2$ in the $[0, t_{\text{max}}]$ interval. For both grid sizes considered, over a wide range of potentials across the hollowing-out transition, we observed violations of unitarity on the order of 10^{-8} . Similarly, the infidelities were on the order of 10^{-6} or smaller. From these results, we conclude that the employed numerical integration scheme and finite-difference stencil provide accurate results.

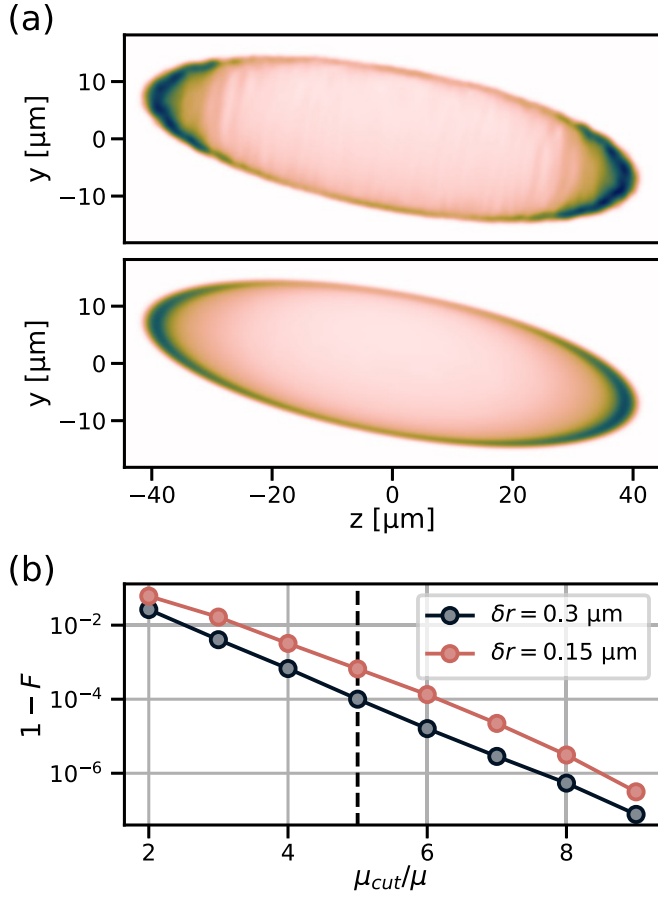


FIG. 3. Verification of the semistructured grid method with GPE time-dependent simulations. (a) Column-density images for a dynamically loaded bubble-shaped condensate (top) and the ground state in the same potential (bottom). The dynamically loaded condensate exhibits excess density near the poles along the z direction, as well as density ripples along the surface. (b) Scaling of the simulation infidelity with the grid selection cutoff for the two grid sizes. The solid lines connecting the data points serve as a guide to the eye. The dashed vertical line indicates the typical cutoff used in this work.

Finally, we investigate the effect of ROI selection for time evolution across a dynamical hollowing-out transition. We start with a compact, harmonically trapped cloud of approximate size $8 \mu\text{m}$ along the x and y directions, and $25 \mu\text{m}$ along the z direction. The potential is modified over 200 ms to arrive at the bubble trap shown in Fig. 2, in which the system is held for a further 200 ms. Due to the significant change in the potential and the timescale of the ramp, the evolution of the state is highly nonadiabatic, resulting in significant deviations from the ground state corresponding to the final potential, as shown in Fig. 3(a). The simulation is repeated with various ROI selection thresholds and we compute the time-averaged infidelity during the 200 ms hold time in the final potential. For the fidelity calculations, we use the selection threshold of $\mu_{\text{cut}}/\mu = 10$ to obtain the reference data. The typical threshold value of $\mu_{\text{cut}}/\mu = 5$ results in infidelities of 1.0×10^{-4} and 6.6×10^{-4} for grid spacings 0.3 and $0.15 \mu\text{m}$, respectively. We attribute the increased error for the finer grid to the smaller fraction of grid points in the ROIs. Neverthe-

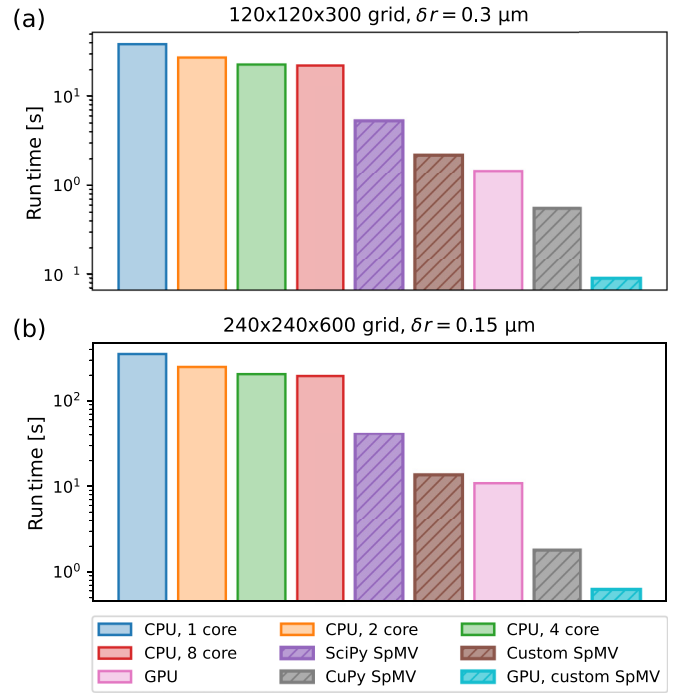


FIG. 4. Benchmark of GPE ground-state solvers for grid sizes $120 \times 120 \times 300$ (a) and $240 \times 240 \times 600$ (b). The bars with solid fill represent the conventional split-step Fourier method on CPU and GPU, while the hatched fill bars show the semistructured grid methods proposed in this work. The run times are averages over seven runs. The uncertainties of the mean values are negligible.

less, these results demonstrate the numerical accuracy of the semistructured grid method for time-evolution simulations of highly nonadiabatic dynamics.

IV. ALGORITHM BENCHMARKING

We benchmark the performance of our algorithms by comparing their run times with those of the conventional split-step Fourier scheme utilizing the highly efficient FFTW library [49] to perform the fast Fourier transforms, as well as GPU computation enabled by the CuPy library. The benchmarks for the split-step Fourier scheme were performed using the same spatial grids as the semistructured grid simulations before selecting the ROI [50]. Furthermore, all static quantities, such as the arrays of the evolution operators $e^{-(i)\mathcal{H}_V \delta t/2\hbar}$ and $e^{-(i)\mathcal{H}_T \delta t/\hbar}$ were precomputed and transferred to GPU memory before the simulation. The reported times reflect purely on the duration of the imaginary and real-time propagation, and exclude these preparation steps. The simulations for the general algorithm were run on an A2338 MacBook Pro equipped with an M1 processor and 16 GB memory. The GPU simulations were run on a Tesla V100 with 16 GB memory. All simulations used double-precision floating-point format.

For the ground-state solver, the simulation parameters employed were the same as in the earlier comparison apart from the relative tolerance of the solvers, which we set to 10^{-6} . In Fig. 4, we compare the run times of the split-step Fourier algorithm executed on CPU (at various levels of multithreading)

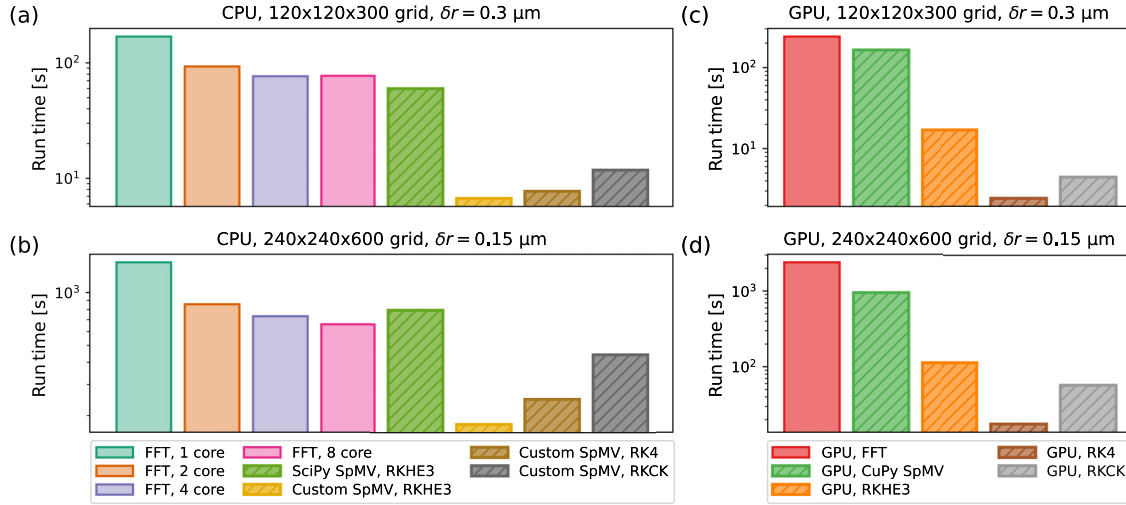


FIG. 5. Benchmarking the time-evolution algorithms and comparison with conventional split-step Fourier solvers for various grid sizes. The run times are averages over seven runs, with negligible variance. The hatched fill highlights the results from the algorithms from this work. (a), (b) CPU algorithm run times for a single-stage ramp between two potentials that correspond to inflating bubble-shaped clouds by $\approx 10\%$. (c), (d) GPU algorithm run times for a ten-stage ramp, which simulated the hollowing-out dynamics and subsequent inflation.

as well as on GPU, with those obtained using the semistructured grid method. The simple SciPy implementation of our algorithm already provided significant speedups compared to the CPU-based split-step Fourier algorithms. However, the highly optimized algorithm based on the custom SpMV kernel significantly outperformed all other CPU algorithms, with mean run times of 2.1 and 14.9 s on the smaller and larger grids, respectively. Furthermore, the custom SpMV algorithm performed marginally worse than the GPU implementation of the split-step Fourier solver. It is important to note that both semistructured grid algorithms executed on the CPU are single threaded; therefore, further speedup can be expected from multithreading.

Similar to the CPU results, the CuPy SpMV implementation of our algorithm resulted in substantial speedup compared to the GPU split-step Fourier solver. However, our highly optimized GPU algorithm outperformed both of these, resulting in run times of 0.09 and 0.6 s, representing a 17-fold speedup compared to the conventional GPU solver. Next, we benchmark the time-evolution algorithms. The CPU-based algorithms were tested on a single-stage ramp of the trapping potential, such that the bubble-shaped cloud changed in size by approximately 10%. This ramp was performed over 10 ms, after which the cloud was held in the final potential for a further 5 ms. For all methods considered, the simulation was initially run with various time steps to check the convergence of the solutions. These initial results were used to identify the time steps that yielded a mean relative global error of 10^{-6} . The benchmarking results are shown in Figs. 5(a) and 5(b) for two different grid sizes. The SciPy SpMV algorithm with an RKHE3 integrator yields similar run times when compared to the multicore split-step Fourier solver. The lack of speedup compared to the ground-state solvers is attributed to the higher order of the integrator as well as the necessity for the wavefunction to be complex. However, the custom SpMV algorithm yields significant speedups. For the problem considered, the RKHE3 integrator resulted in the best run

time, corresponding to a 25- and 8-fold increase in simulation speed for the smaller and larger grids, respectively.

The GPU-based algorithms were tested on a ten-stage ramp of the trapping potential, starting from a filled, compact cloud that was gradually expanded into a bubble-shaped configuration. The total duration of the ramp was 100 ms, after which the system was held in the final potential for 20 ms. Similar to the CPU-based case, the optimal time step was chosen with convergence tests by imposing the 10^{-6} mean relative global error condition. For the CuPy SpMV algorithm with an RKHE3 integrator, we observed a speedup of ≈ 2 over the FFT-based split-step solver. The highly optimized custom CUDA kernels, described in Sec. II B, significantly outperformed the previous two. For the specific problem, the RK4 integrator resulted in the best run time, corresponding to a 100- and 135-fold increase in simulation speed for the smaller and larger grids, respectively, despite using the 27-point HOC stencil.

V. SIMULATING INFLATION DYNAMICS

A common way to engineer trapping potentials for bubble-shaped quantum gases relies on the rf-dressing technique. The basis of this method is the irradiation of magnetically trapped atoms with a strong rf field; when this field is resonant with the energy splitting between Zeeman sublevels, the atoms experience an adiabatic potential that smoothly connects the two extremal (trapping and antitrapping) bare Zeeman states. By treating the interaction between the atoms and the rf field within the rotating-wave approximation, the dressed potential is given by

$$V(\mathbf{r}) = \hbar \tilde{m}_F \sqrt{\delta^2(\mathbf{r}) + \Omega^2(\mathbf{r})}, \quad (10)$$

where δ is the difference between the rf frequency and the Larmor frequency of the atoms, Ω is the Rabi frequency characterizing the strength of rf coupling between the Zeeman sublevels, and $\tilde{m}_F = 0, \pm 1, \dots$ is the magnetic quantum

number labeling the dressed eigenstates [51]. For the dressed states with $\tilde{m}_F > 0$, this results in the formation of a local minimum in the potential near rf resonance, corresponding to $\delta = 0$. The geometry of the trapped cloud is primarily controlled by the frequency of the rf field. If the rf frequency is well below the Zeeman splitting at the center of the trap, its effect is negligible, resulting in a conventional, compact cloud. In the opposite case, a trap minimum is formed near an ellipsoidal isomagnetic surface of the dc magnetic field; these surfaces are oblate spheroids for a quadrupole field and prolate spheroids for the Ioffe field used in this work. Therefore, a controlled ramp of the rf frequency provides a natural way to load a quantum gas from a tight, compact trap, where evaporative cooling is efficient, into a bubble trap. For Earth-based experiments, the effect of the additional gravitational potential results in sagged and partially filled shell-shaped clouds [52]. However, the application of this technique on experiments in microgravity environments, such as the Cold Atom Laboratory on the International Space Station, resulted in the first observation of ultracold quantum bubbles in an rf-dressed trap [12].

As the trapped gas undergoes a significant, topological change in its shape through the formation of the inner empty volume, it is vitally important to perform the ramp adiabatically to avoid the excitation of collective modes. This is especially important at the point of the hollowing-out transition when the interior boundary forms, as this is associated with a significant change in the excitation spectrum, with the presence of low-frequency modes [19,20]. Therefore, simulation of the dynamic loading of the shell trap is of primary importance for the development of optimal control protocols to guide ongoing experimental efforts.

Here, we use our highly efficient methods to simulate the dynamics that arise from various rf frequency ramps. We precompute rf-dressed adiabatic potentials for $\Delta = [-10 \text{ kHz}, 23 \text{ kHz}]$ in steps of 500 Hz, for a given dc and rf magnetic field configuration, where Δ is the detuning of the rf field from the Larmor frequency at the center of the dc magnetic trap. The resulting 67 potentials thus define a 66-stage ramp. The value of the starting detuning was chosen with the possible experimental implementation in mind: The dressing rf field must be turned on at a frequency below rf resonance, which does not significantly alter the potential, minimizing projective atom loss. This occurs when the rf detuning is around or less than the Rabi frequency used. The value of the final detuning was constrained by the size of the simulation grid.

To obtain the initial state for the dynamical simulation, we compute the ground-state wavefunction with 10^5 atoms of ^{87}Rb in the potential corresponding to $\Delta = -10 \text{ kHz}$. After this, we perform each of the 66 ramp stages, linearly interpolating between the starting and final potentials. At the end of each stage, the grid postselection procedure is repeated, and the wavefunction is remapped between the old and new grids to account for the varying size of the cloud. The threshold for grid postselection is chosen as $5 \times \max(\mu_i, \mu_{i+1})$, where μ_i is the chemical potential of the ground state with 10^5 particles at step i . To further ensure a suitable choice of simulation region, we use an extended ROI, defined as the union of the ROIs associated with the two potentials V_i and V_{i+1} . At the end of

the ramp, the cloud is held for 100 ms in the final potential. During this hold, we record the wavefunction every 1 ms and use the resulting data to quantify the excitations present in the bubble-shaped cloud.

As the dominant excitations are low-energy phonons that propagate along the surface of the bubble, we focus on the mean kinetic energy of these modes. To evaluate this quantity, we first extract the radius of the bubble $R(\theta, \phi)$ as a function of the polar and azimuthal angles θ and ϕ from the ground-state solution in the final $\Delta = 23 \text{ kHz}$ potential. We then numerically calculate the components of the metric tensor associated with the curved surface as $\gamma_{\theta\theta} = R_\theta^2 + R^2$, $\gamma_{\phi\phi} = R_\phi^2 + R^2 \sin^2 \theta$, and $\gamma_{\theta\phi} = R_\theta R_\phi$, where $R_{\theta,\phi} = \partial R / \partial (\theta, \phi)$. We interpolate the original wavefunction and extract its value along the bubble surface ψ_s , i.e., at the local minimum of the potential, using linearly spaced arrays of θ and ϕ . The surface wavefunction ψ_s is normalized to the original particle number by evaluating

$$N_s = \int |\psi_s|^2 \sqrt{\gamma} d\theta d\phi, \quad (11)$$

where γ is the metric determinant, and subsequently rescaling by $\sqrt{N/N_s}$. Finally, the surface kinetic energy is evaluated as

$$E_k[\psi_s] = \frac{\hbar^2}{2m} \int |\nabla_s \psi_s|^2 \sqrt{\gamma} d\theta d\phi, \quad (12)$$

where

$$|\nabla_s \psi_s|^2 = \frac{1}{\gamma} [\gamma_{\phi\phi} |\partial_\theta \psi_s|^2 + \gamma_{\theta\theta} |\partial_\phi \psi_s|^2 - \text{Re}(2\gamma_{\theta\phi} (\partial_\theta \psi_s^*)(\partial_\phi \psi_s))]. \quad (13)$$

The main figure of merit for adiabaticity was chosen to be $C = \langle E_k[\psi_s] \rangle_t$, the time average of the surface kinetic energy during the hold in the final potential. To find optimal control protocols, we perform Bayesian optimization of the rf detuning ramp using C as a cost function. We modify the ramp by changing the timing of the 66 individual stages. To reduce the dimensionality of the optimization space, we use 10 free parameters and linearly interpolate between them to get the duration of each of the 66 stages. The durations are scaled together such that the total length of the ramp t_{max} is fixed. The optimization was performed for $t_{\text{max}} = [100, 200, 400, 600, 800] \text{ ms}$.

In Fig. 6(a), we show the lowest cost achieved by the optimizer for each of the considered ramp durations and compare it with the results from two standard closed-form ramps of the same durations, linear and smoothstep (S_1). The optimized ramps performed significantly better than the standard ramps. To achieve the same degree of adiabaticity as the optimal ramp corresponding to $t_{\text{max}} = 200 \text{ ms}$, linear and smoothstep ramps would need to be a factor of ~ 3 and ~ 2 longer. This is particularly advantageous for an experimental implementation as long ramps suffer from the intrinsic heating and atom loss associated with technical noise in typical traps. In Fig. 6(b), we plot the optimized rf detuning ramps for two values of t_{max} along with the standard, closed-form ramps. The optimized ramps exhibit distinctly different behavior from the linear and smoothstep ramps. At the same time, for the two considered durations, the optimized ramps show common characteristics:

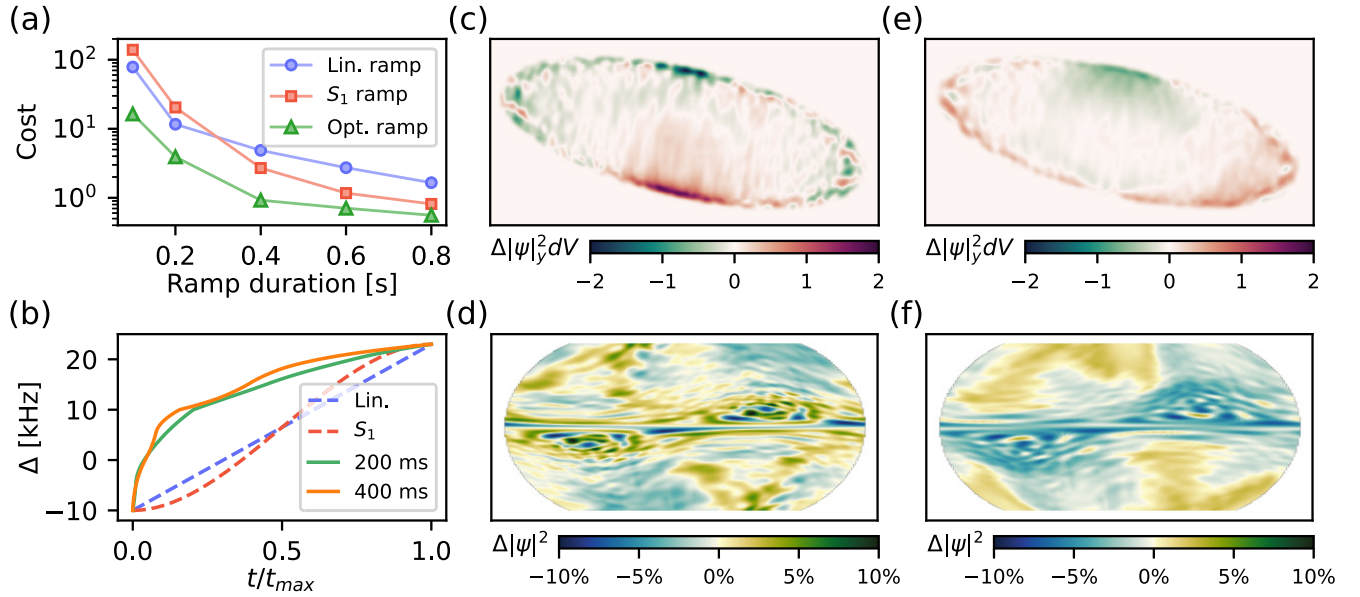


FIG. 6. Inflation dynamics of bubble-shaped BECs. (a) Evaluation of adiabaticity of various rf ramps with different durations. The lines connecting the markers serve as a guide to the eye. (b) Optimal rf detuning ramps of duration 200 and 400 ms, compared to linear and smoothstep (S_1) ramps. (c), (d) Column density and projected density difference between the ground state and the dynamically loaded bubble state for the 400 ms smoothstep ramp. (e), (f) Column density and projected density difference between the ground state and the dynamically loaded bubble state for the 400 ms optimal ramp. For the projected images, the area-preserving Equal Earth projection was used.

As the hollowing-out transition occurs in the $\Delta \gtrsim 0$ region, the rate of change of detuning significantly slows down, and the expansion of the bubble is performed very slowly. Next, we visualize the excitations present in the final state after the dynamical loading of the bubble trap and a subsequent 100 ms hold time in the final trapping potential for the smoothstep and optimized ramps with $t_{\max} = 400$ ms. In Figs. 6(c) and 6(e), we show the column density difference $\Delta|\psi|^2 dV$ between the dynamically loaded system and the ground state for these two ramps, respectively. As expected from the lower value of the cost function, the system undergoing the optimized loading ramp exhibits significantly smaller density fluctuations. The same comparison is performed in Figs. 6(d) and 6(f), where we use a projected view of the density mapping the entire bubble-shaped surface onto a 2D plane. The optimized ramp results in density fluctuations with smaller amplitude and longer wavelengths, demonstrating the suppression of higher-energy excitations.

VI. CONCLUSIONS AND OUTLOOK

In summary, we have developed a general, fast, and memory-efficient simulation method for the Gross-Pitaevskii equation modeling Bose-Einstein condensates in nontrivial topologies, which relies on a semistructured simulation grid. The simulation methods were implemented for execution on both CPUs and GPUs. For computing ground states and time evolution, we have observed speedups by factors of up to 17 and 135, respectively, when compared to the standard split-step Fourier approach. Finally, using the semistructured grid method, we simulated the loading and subsequent inflation dynamics of bubble traps and used machine-learning methods to identify optimal ramps that suppress the excitation

of collective modes. Compared to simple ramps, the optimal protocol allows the loading and inflation of the bubble trap twice as fast.

Our methods significantly extend current capabilities for simulating the dynamics of quantum gases in hollow geometries, thereby providing guidance for their experimental realization. In particular, we anticipate that our methods provide an interesting prospect for modeling bubble BECs, considering the growing interest in bubble-shaped quantum gases, with recent advances in experiments realizing bubble-shaped gases [12,16,17], experimental proposals [53], and experiments expected to come online in the near future [54]. Furthermore, the semistructured grid method is also extendable to a large class of simulations of Bose-Einstein condensates, such as soliton dynamics on curved manifolds [55,56], dipolar gases in bubble traps [57–59] or mixtures in bubble traps [33,60].

ACKNOWLEDGMENTS

We thank Jeffrey Oishi, Smitha Vishveshwara, Brendan Rhyno, and Matteo Sbroscia for useful discussion. This work was supported by the National Aeronautics and Space Administration (NASA) Science Mission Directorate, Division of Biological and Physical Sciences (BPS) through ROSES22 as well as multiple Jet Propulsion Laboratory (JPL) Research Support Agreements.

DATA AVAILABILITY

The data that support the findings of this article are not publicly available upon publication because it is not technically feasible and/or the cost of preparing, depositing, and

hosting the data would be prohibitive within the terms of this research project. The data are available from the authors upon reasonable request.

APPENDIX A: MEMORY-OPTIMIZED RUNGE-KUTTA PROCEDURE FOR GPUS

Within the standard Runge-Kutta procedure, each intermediate state of the system $y^{(i)}$ is related to the previous state of the system y_0 and the evaluations of the right-hand side at $y^{(j<i)}$ through

$$y^{(i)} = y_0 + \delta t \sum_{j<i} a_{ij} k_j. \quad (\text{A1})$$

This motivates defining vectors

$$\vec{Y} = (y^{(1)}, y^{(2)}, y^{(3)}, \dots, y^{(n)}) \quad (\text{A2})$$

and

$$\vec{K} = (y_0, k_1, k_2, \dots, k_{n-1}), \quad (\text{A3})$$

where $y^{(1)} \equiv y_0$. Then, we can write the relationship between $\vec{Y}$ and $\vec{K}$ as

$$\vec{Y} = \mathbf{M} \vec{K}, \quad (\text{A4})$$

where

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & \delta t a_{21} & 0 & \dots & 0 \\ 1 & \delta t a_{31} & \delta t a_{32} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \delta t a_{n1} & \delta t a_{n2} & \dots & \delta t a_{n,n-1} \end{bmatrix}, \quad (\text{A5})$$

with a_{ij} being the coefficients of the standard Butcher tableau. Inverting $\mathbf{M}$ allows us to express k_i as a linear combination of y_0 and $y^{(j<i)}$. Finally, at each step, we can use these linear combinations to express the intermediate state at the next step $y^{(i+1)}$ using the previous intermediate stages and k_i . This can be written as

$$y^{(i+1)} = \sum_{j=1}^i a_{ij}^* y^{(j)} + \delta t a_{i+1,i} k_i, \quad (\text{A6})$$

where the coefficients a_{ij}^* form the modified Butcher tableau. Here, the k_i term does not require additional memory accesses, as these values directly result from evaluating the right-hand side at $y^{(i)}$. The state of the system at the next time step is calculated as

$$y_1 = \sum_{j=1}^n b_j^* y^{(j)} + \delta t b_n k_n. \quad (\text{A7})$$

We now apply this protocol to Heun's third-order method (RKHE3). The standard implementation of this method uses the following iterations:

$$\begin{aligned} y_0 &= y(t), \\ k_1 &= f(t, y_0), \\ k_2 &= f(t + c_2 \delta t, y_0 + a_{21} \delta t k_1), \\ k_3 &= f(t + c_3 \delta t, y_0 + a_{32} \delta t k_2), \\ y(t + \delta t) &= y_0 + \delta t (b_1 k_1 + b_3 k_3), \end{aligned} \quad (\text{A8})$$

with coefficients

$$a_{ij} = \begin{bmatrix} 1/3 & 0 \\ 0 & 2/3 \end{bmatrix} \quad (\text{A9})$$

($i = 2, 3; j = 1, 2$),

$$b_i = [1/4 \quad 0 \quad 3/4] \quad (\text{A10})$$

($i = 1, 2, 3$), and

$$c_i = [1/3 \quad 2/3] \quad (\text{A11})$$

($i = 2, 3$). For an actual implementation, it is easy to see that y_0 , k_1 , and k_3 must be stored in memory before the update is finished, while k_2 is needed until the completion of the third step. On a GPU, each of the three steps is evaluated by a separate kernel to avoid race conditions. Furthermore, using a fast, shared-memory implementation of the algorithm, the second and third stages require loading both y_0 and $k_{1,2}$. For the block size used in this work, this amounts to loading 2000 values from global GPU memory, while only producing 512 output values for each block. The improved scheme addresses this memory inefficiency by changing which arrays are stored by each step. We start by defining $y^{(1)} \equiv y(t)$, and the first GPU kernel evaluates k_1 as in the case of the standard scheme [61]. However, instead of storing k_1 in global memory, we store the array required for the next stage,

$$y^{(2)} = y^{(1)} + \delta t a_{21} f(t, y^{(1)}). \quad (\text{A12})$$

This step requires loading 1000 and writing 512 values per block. The kernel evaluating the second step needs to load the 1000 values of $y^{(2)}$, followed by the evaluation of the right-hand side. Following the same logic, we wish to calculate and write into memory

$$y^{(3)} = y^{(1)} + \delta t a_{32} f(t + c_2 \delta t, y^{(2)}), \quad (\text{A13})$$

which requires loading an additional 512 values of $y^{(1)}$. However, this requires loading a total of 1512 elements instead of 2000 as in the standard scheme. Finally, we compute the last intermediate step $f(t + c_3 \delta t, y^{(3)})$ and compute the next state of the system as

$$y(t + \delta t) = y^{(1)} + b_1 \frac{y^{(2)} - y^{(1)}}{a_{21}} + \delta t b_3 f(t + c_3 \delta t, y^{(3)}). \quad (\text{A14})$$

The key step is expressing k_1 in terms of $y^{(1)}$ and $y^{(2)}$ using Eq. (A12). Inspecting the coefficients and gathering terms in Eqs. (A12)–(A14), we get the modified a and b coefficients

$$a_{ij} = \begin{bmatrix} 1/3 & 0 \\ 0 & 2/3 \end{bmatrix}, \quad (\text{A15})$$

$$a_{ij}^* = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \quad (\text{A16})$$

($i = 2, 3; j = 1, 2$), and

$$b^* = [1/4 \quad 3/4 \quad 0] \quad (\text{A17})$$

($i = 1, 2, 3$). The simple structure of the modified a coefficients results directly from a_{ij} being diagonal. The improvement in global memory traffic is summarized in Table I, revealing the 7.5% improvement. The standard implementation of the classic fourth-order Runge-Kutta method

TABLE I. Analysis of global memory traffic of normal and memory-optimized RKHE3 algorithms. The values indicate the number of array elements loaded/written per block. The bold symbols indicate arrays that must be loaded together with a halo, amounting to 1000 elements.

Normal method				
Step	Loading	Writing	N_{load}	N_{write}
1	$\mathbf{y_0}$	k_1	1000	512
2	$\mathbf{y_0, k_1}$	k_2	2000	512
3	$\mathbf{y_0, k_2}$	y_1	2000	512
Total			5000	1536
Optimized method				
Step	Loading	Writing	N_{load}	N_{write}
1	$\mathbf{y_0}$	$y^{(2)}$	1000	512
2	$y_0, \mathbf{y^{(2)}}$	$y^{(3)}$	1512	512
3	$y_0, y^{(2)}, \mathbf{y^{(3)}}$	y_1	2024	512
Total			4536	1536

uses the following iterations:

$$\begin{aligned}
 y_0 &= y(t), \\
 k_1 &= f(t, y_0), \\
 k_2 &= f(t + c_2 \delta t, y_0 + a_{21} \delta t k_1), \\
 k_3 &= f(t + c_3 \delta t, y_0 + a_{32} \delta t k_2), \\
 k_4 &= f(t + c_4 \delta t, y_0 + a_{43} \delta t k_3), \\
 y(t + \delta t) &= y_0 + \delta t (b_1 k_1 + b_2 k_2 + b_3 k_3 + b_4 k_4), \quad (\text{A18})
 \end{aligned}$$

with coefficients

$$a_{ij} = \begin{bmatrix} 1/2 & 0 & 0 \\ 0 & 1/2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{A19})$$

($i = 2, 3, 4; j = 1, 2, 3$),

$$b_i = [1/6 \quad 1/3 \quad 1/3 \quad 1/6] \quad (\text{A20})$$

($i = 1, 2, 3, 4$), and

$$c_i = [1/2 \quad 1/2 \quad 1] \quad (\text{A21})$$

($i = 2, 3, 4$). Since the a coefficient matrix is diagonal, the modified coefficients are given by

$$a^* = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}. \quad (\text{A22})$$

Furthermore, following the same procedure as for Heun's third-order method and expressing k_i in terms of $y^{(j)}$ in the formula for $y(t + \delta t)$ yields

$$b^* = [-1/3 \quad 1/3 \quad 2/3 \quad 1/3]. \quad (\text{A23})$$

The resulting scheme reduces global memory traffic by 12%.

For the RKCK method [46], as the a coefficient matrix is not diagonal, the modified coefficient matrix is more complicated, but straightforward to obtain using basic linear algebra. We obtain

$$a^* = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ \frac{5}{8} & \frac{3}{8} & 0 & 0 & 0 \\ 2 & 3 & -4 & 0 & 0 \\ \frac{-49}{81} & \frac{35}{27} & \frac{200}{81} & \frac{-175}{81} & 0 \\ \frac{253}{12288} & \frac{-3795}{4096} & \frac{1375}{1536} & \frac{2875}{4096} & \frac{1265}{4096} \end{bmatrix}, \quad (\text{A24})$$

and

$$b^* = \begin{bmatrix} \frac{-991}{15939} & \frac{-3865}{5313} & \frac{5000}{5313} & \frac{3125}{4554} & \frac{25}{154} \end{bmatrix}. \quad (\text{A25})$$

Due to the dependence of each stage on all the earlier stages in the standard implementation, the memory traffic analysis similar to Table I yields a 44% improvement.

APPENDIX B: FINITE-DIFFERENCE STENCILS FOR ANISOTROPIC GRIDS

For anisotropic grids ($d_x \neq d_y \neq d_z$), the standard seven-point Laplacian stencil can be written as

$$\nabla_{7\text{pt}}^2 = \delta_x^2 \otimes \mathbb{1} \otimes \mathbb{1} + \mathbb{1} \otimes \delta_y^2 \otimes \mathbb{1} + \mathbb{1} \otimes \mathbb{1} \otimes \delta_z^2, \quad (\text{B1})$$

where $\delta_i^2 = [1, -2, 1]/(d_i)^2$ with d_i being the step size along direction i and $\mathbb{1} = [0, 1, 0]$ is the identity operator. Following the derivation in Ref. [62], the 19-point HOC stencil for anisotropic grids is given by

$$\begin{aligned}
 \nabla_{19\text{pt}}^2 &= \nabla_{7\text{pt}}^2 + \frac{1}{12} [(d_x^2 + d_y^2) \delta_x^2 \otimes \delta_y^2 \otimes \mathbb{1} \\
 &\quad + (d_x^2 + d_z^2) \delta_x^2 \otimes \mathbb{1} \otimes \delta_z^2 + (d_y^2 + d_z^2) \mathbb{1} \otimes \delta_y^2 \otimes \delta_z^2]. \quad (\text{B2})
 \end{aligned}$$

- [1] M. A. Cazalilla, R. Citro, T. Giamarchi, E. Orignac, and M. Rigol, One dimensional bosons: From condensed matter systems to ultracold gases, *Rev. Mod. Phys.* **83**, 1405 (2011).
- [2] Z. Hadzibabic and J. Dalibard, Two-dimensional Bose fluids: An atomic physics perspective, *Riv. Nuovo Cimento* **34**, 389 (2011).
- [3] R. Shah, T. J. Barrett, A. Colcelli, F. Oručević, A. Trombettoni, and P. Krüger, Probing the degree of coherence through the full 1D to 3D crossover, *Phys. Rev. Lett.* **130**, 123401 (2023).

- [4] F. Schäfer, T. Fukuhara, S. Sugawa, Y. Takasu, and Y. Takahashi, Tools for quantum simulation with ultracold atoms in optical lattices, *Nat. Rev. Phys.* **2**, 411 (2020).
- [5] N. Navon, R. P. Smith, and Z. Hadzibabic, Quantum gases in optical boxes, *Nat. Phys.* **17**, 1334 (2021).
- [6] A. L. Fetter, Rotating trapped Bose-Einstein condensates, *Rev. Mod. Phys.* **81**, 647 (2009).

- [7] N. Goldman, J. C. Budich, and P. Zoller, Topological quantum matter with ultracold gases in optical lattices, *Nat. Phys.* **12**, 639 (2016).
- [8] A. Ramanathan, K. C. Wright, S. R. Muniz, M. Zelan, W. T. Hill, C. J. Lobb, K. Helmerson, W. D. Phillips, and G. K. Campbell, Superflow in a toroidal Bose-Einstein condensate: An atom circuit with a tunable weak link, *Phys. Rev. Lett.* **106**, 130401 (2011).
- [9] V. Galitski and I. B. Spielman, Spin-orbit coupling in quantum gases, *Nature (London)* **494**, 49 (2013).
- [10] C. Chin, R. Grimm, P. Julienne, and E. Tiesinga, Feshbach resonances in ultracold gases, *Rev. Mod. Phys.* **82**, 1225 (2010).
- [11] N. Defenu, T. Donner, T. Macrì, G. Pagano, S. Ruffo, and A. Trombettoni, Long-range interacting quantum systems, *Rev. Mod. Phys.* **95**, 035002 (2023).
- [12] R. A. Carollo, D. C. Aveline, B. Rhyno, S. Vishveshwara, C. Lannert, J. D. Murphree, E. R. Elliott, J. R. Williams, R. J. Thompson, and N. Lundblad, Observation of ultracold atomic bubbles in orbital microgravity, *Nature (London)* **606**, 281 (2022).
- [13] D. C. Aveline, J. R. Williams, E. R. Elliott, C. Dutenhoffer, J. R. Kellogg, J. M. Kohel, N. E. Lay, K. Oudhiri, R. F. Shotwell, N. Yu, and R. J. Thompson, Observation of Bose-Einstein condensates in an Earth-orbiting research lab, *Nature (London)* **582**, 193 (2020).
- [14] Y. Guo, E. Mercado Gutierrez, D. Rey, T. Badr, A. Perrin, L. Longchambon, V. S. Bagnato, H. Perrin, and R. Dubessy, Expansion of a quantum gas in a shell trap, *New J. Phys.* **24**, 093040 (2022).
- [15] A. Wolf, P. Boegel, M. Meister, A. Balaž, N. Gaaloul, and M. A. Efremov, Shell-shaped Bose-Einstein condensates based on dual-species mixtures, *Phys. Rev. A* **106**, 013309 (2022).
- [16] F. Jia, Z. Huang, L. Qiu, R. Zhou, Y. Yan, and D. Wang, Expansion dynamics of a shell-shaped Bose-Einstein condensate, *Phys. Rev. Lett.* **129**, 243402 (2022).
- [17] Z. Huang, K. Y. Lee, C. K. Wong, L. Qiu, B. Yang, Y. Yan, and D. Wang, Probing the hollowing transition of a shell-shaped Bose-Einstein condensate with collective excitation, *Phys. Rev. Res.* **7**, 033056 (2025).
- [18] C. Lannert, T.-C. Wei, and S. Vishveshwara, Dynamics of condensate shells: Collective modes and expansion, *Phys. Rev. A* **75**, 013611 (2007).
- [19] K. Padavić, K. Sun, C. Lannert, and S. Vishveshwara, Physics of hollow Bose-Einstein condensates, *Europhys. Lett.* **120**, 20004 (2017).
- [20] K. Sun, K. Padavić, F. Yang, S. Vishveshwara, and C. Lannert, Static and dynamic properties of shell-shaped condensates, *Phys. Rev. A* **98**, 013609 (2018).
- [21] Y. Xiong and X. Yu, Hydrodynamics of quantum vortices on a closed surface, *Phys. Rev. Res.* **6**, 013133 (2024).
- [22] A. Tononi, L. Salasnich, and A. Yakimenko, Quantum vortices in curved geometries, *AVS Quantum Sci.* **6**, 030502 (2024).
- [23] A. M. Turner, V. Vitelli, and D. R. Nelson, Vortices on curved surfaces, *Rev. Mod. Phys.* **82**, 1301 (2010).
- [24] K. Padavić, K. Sun, C. Lannert, and S. Vishveshwara, Vortex-antivortex physics in shell-shaped Bose-Einstein condensates, *Phys. Rev. A* **102**, 043305 (2020).
- [25] S. J. Bereta, M. A. Caracanhas, and A. L. Fetter, Superfluid vortex dynamics on a spherical film, *Phys. Rev. A* **103**, 053306 (2021).
- [26] A. Tononi and L. Salasnich, Bose-Einstein condensation on the surface of a sphere, *Phys. Rev. Lett.* **123**, 160403 (2019).
- [27] A. Tononi, F. Cinti, and L. Salasnich, Quantum bubbles in microgravity, *Phys. Rev. Lett.* **125**, 010402 (2020).
- [28] B. Rhyno, N. Lundblad, D. C. Aveline, C. Lannert, and S. Vishveshwara, Thermodynamics in expanding shell-shaped Bose-Einstein condensates, *Phys. Rev. A* **104**, 063310 (2021).
- [29] A. Tononi, A. Pelster, and L. Salasnich, Topological superfluid transition in bubble-trapped condensates, *Phys. Rev. Res.* **4**, 013122 (2022).
- [30] J. Yi, W. Zhang, R. Zhang, and T. Shi, Bosonic superfluid on ellipsoidal surface, *Europhys. Lett.* **150**, 55001 (2025).
- [31] S. M. de Oliveira and N. Salomé Móller, Geometric potential for a Bose-Einstein condensate on a curved surface, *AVS Quantum Sci.* **7**, 033203 (2025).
- [32] G. Li and D. K. Efimkin, Equatorial waves in rotating bubble-trapped superfluids, *Phys. Rev. A* **107**, 023319 (2023).
- [33] L. Brito, L. Tomio, and A. Gammal, Faraday waves on a bubble-trapped Bose-Einstein-condensed binary mixture, *Phys. Rev. A* **108**, 053315 (2023).
- [34] R. C. T. d. Costa, Quantum mechanics of a constrained particle, *Phys. Rev. A* **23**, 1982 (1981).
- [35] T. Grass, D. Bercioux, U. Bhattacharya, M. Lewenstein, H. S. Nguyen, and C. Weitenberg, *Colloquium*: Synthetic quantum matter in nonstandard geometries, *Rev. Mod. Phys.* **97**, 011001 (2025).
- [36] A. Tononi and L. Salasnich, Low-dimensional quantum gases in curved geometries, *Nat. Rev. Phys.* **5**, 398 (2023).
- [37] N. S. Móller, F. E. A. dos Santos, V. S. Bagnato, and A. Pelster, Bose-Einstein condensation on curved manifolds, *New J. Phys.* **22**, 063059 (2020).
- [38] A. J. Leggett, Bose-Einstein condensation in the alkali gases: Some fundamental concepts, *Rev. Mod. Phys.* **73**, 307 (2001).
- [39] J. Javanainen and J. Ruostekoski, Symbolic calculation in development of algorithms: Split-step methods for the Gross-Pitaevskii equation, *J. Phys. A: Math. Gen.* **39**, L179 (2006).
- [40] A. S. Bradley, C. W. Gardiner, and M. J. Davis, Bose-Einstein condensation from a rotating thermal cloud: Vortex nucleation and lattice formation, *Phys. Rev. A* **77**, 033616 (2008).
- [41] C. N. Weiler, T. W. Neely, D. R. Scherer, A. S. Bradley, M. J. Davis, and B. P. Anderson, Spontaneous vortices in the formation of Bose-Einstein condensates, *Nature (London)* **455**, 948 (2008).
- [42] E. Rydow, V. P. Singh, A. Beregi, E. Chang, L. Mathey, C. J. Foot, and S. Sunami, Observation of a bilayer superfluid with interlayer coherence, *Nat. Commun.* **16**, 7201 (2025).
- [43] E. Chang, V. P. Singh, A. Beregi, E. Rydow, L. Mathey, C. J. Foot, and S. Sunami, Coupling-induced universal dynamics in bilayer two-dimensional Bose gases, [arXiv:2510.23600](https://arxiv.org/abs/2510.23600).
- [44] The kinetic-energy propagation can be performed in a single step using higher-order explicit methods. However, it was found that for typical values of δt , this scheme was numerically stable while only requiring two evaluations of the right-hand side.
- [45] The iteration formulas for the RKHE3 and RK4 methods are provided in Appendix A.
- [46] J. R. Cash and A. H. Karp, A variable order Runge-Kutta method for initial value problems with rapidly varying right-hand sides, *ACM Trans. Math. Software* **16**, 201 (1990).

- [47] For a block size of $8 \times 8 \times 8$, the halo contains approximately the same number of points as the block itself. For a block size of $4 \times 4 \times 4$, the halo contains approximately three times more points.
- [48] N. Lundblad, R. A. Carollo, C. Lannert, M. J. Gold, X. Jiang, D. Paseltiner, N. Sergay, and D. C. Aveline, Shell potentials for microgravity Bose–Einstein condensates, *npj Microgravity* **5**, 30 (2019).
- [49] M. Frigo and S. Johnson, The design and implementation of FFTW3, *Proc. IEEE* **93**, 216 (2005).
- [50] The split-step Fourier method simulations were also run with similarly sized radix-2 grids ($128 \times 128 \times 256$ and $256 \times 256 \times 512$). These simulations produced runtimes that were at most 5% faster compared to the actual grid sizes.
- [51] B. M. Garraway and H. Perrin, Recent developments in trapping and manipulation of atoms with adiabatic potentials, *J. Phys. B: At. Mol. Opt. Phys.* **49**, 172001 (2016).
- [52] Y. Colombe, E. Knyazchyan, O. Morizot, B. Mercier, V. Lorent, and H. Perrin, Ultracold atoms confined in rf-induced two-dimensional trapping potentials, *Europhys. Lett.* **67**, 593 (2004).
- [53] R. Veyron, C. Métayer, J. B. Gerent, R. Huang, E. Beraud, B. M. Garraway, S. Bernon, and B. Battelier, All-optical bubble trap for ultracold atoms in microgravity, *AVS Quantum Sci.* **8**, 013203 (2026).
- [54] K. Frye, S. Abend, W. Bartosch, A. Bawamia, D. Becker, H. Blume, C. Braxmaier, S.-W. Chiow, M. A. Efremov, W. Ertmer, *et al.*, The Bose-Einstein condensate and cold atom laboratory, *EPJ Quantum Technol.* **8**, 1 (2021).
- [55] H.-B. Luo, G. Li, F.-Q. Dou, B. Liu, B. A. Malomed, J. Batle, and Y. Li, Solitons in Bose-Einstein condensates with attractive self-interaction on a Möbius strip, *Phys. Rev. A* **111**, 023325 (2025).
- [56] R. W. Sallatti, L. Tomio, D. E. Pelinovsky, and A. Gammal, Stability of dark solitons in a bubble Bose-Einstein condensate, *Phys. Rev. A* **113**, L041303 (2026).
- [57] M. Arazo, R. Mayol, and M. Guilleumas, Shell-shaped condensates with gravitational sag: Contact and dipolar interactions, *New J. Phys.* **23**, 113040 (2021).
- [58] H. S. Ghosh, S. Halder, S. Das, and S. Majumder, Induced supersolidity and hypersonic flow of a dipolar Bose-Einstein condensate in a rotating bubble trap, *Phys. Rev. A* **110**, 033322 (2024).
- [59] M. Ciardi, F. Cinti, G. Pellicane, and S. Prestipino, Supersolid phases of Bosonic particles in a bubble trap, *Phys. Rev. Lett.* **132**, 026001 (2024).
- [60] A. Andriati, L. Brito, L. Tomio, and A. Gammal, Stability of a Bose-condensed mixture on a bubble trap, *Phys. Rev. A* **104**, 033318 (2021).
- [61] In our naming convention, $y^{(i)}$ is the input of the i th intermediate stage.
- [62] W. F. Spotz and G. F. Carey, A high-order compact formulation for the 3D Poisson equation, *Numer. Methods Partial Differ. Equations* **12**, 235 (1996).