

Grid-based methods for chemistry modelling on a quantum computer

Hans Hon Sang Chan,^{1,*} Richard Meister,¹ Tyson Jones,¹ David P. Tew,² and Simon C. Benjamin^{1,3}

¹*Department of Materials, University of Oxford, Oxford OX1 3PH, United Kingdom*

²*Department of Chemistry, University of Oxford, Oxford OX1 3TA, United Kingdom*

³*Quantum Motion, 9 Sterling Way, London N7 9HJ, United Kingdom*

We explore grid-based techniques for simulating multi-electron dynamics using a quantum computer, an approach sometimes called *first quantized chemistry*. Using a split-operator Fourier transform (SO-FT) method on exactly-emulated quantum computers with up to 36 qubits, we model the dynamics of 2D and 3D atoms with single and paired particles interacting via the Coulomb potential. The grid-based method enables direct visualisation of quantum dynamics, including the behaviour of bound states, electron ionization by an applied electric field, and electron-electron scattering. We exploit ancilla-qubit measurements for iterative phase estimation, state preparation and for implementing complex (absorbing) potentials. Ancilla-based methods for obtaining observables such as binding energies are found to be far more efficient and robust than direct sampling. We also introduce an augmented split-operator (ASO) method where the usual kinetic and potential operators are supplemented by a core-stabilisation operator that dramatically improves wavepacket propagation near a Coulomb singularity. Overall, the grid-based method scales well in terms of the quantum processor's size and execution time; we extrapolate the resource requirements for accurate modelling of classically-intractable systems from our numerical results, and we describe suitable quantum architectures. We conclude by noting that the approach should be successful in far broader settings, from high energy physics to financial portfolio modelling.

I. INTRODUCTION AND BACKGROUND

A. Purpose and scope of this paper

Computer simulations provide invaluable insights into the properties of molecules and materials, informing and complementing laboratory investigations. But while the time and resource costs of simulation may be lower than experimental efforts, those costs are nevertheless often high. Moreover there are systems of great interest that are beyond the capabilities of conventional computers to comprehensively simulate [1].

Quantum computing promises to be very significant for molecular simulation. *Ab initio* simulation of quantum molecular dynamics, one of the most accurate and powerful approaches, is understood to scale only polynomially on quantum computers [2, 3], but is limited by exponential scaling on classical computers. In spite of their formidable classical computation cost, and cheaper methods (e.g. electronic structure theory [4, 5] and semi-classical quantum dynamics theory [6, 7]) being very practical approaches for gaining good chemical insight [3], small-molecule quantum dynamics simulations have proliferated in research [1, 8–13], a testament to their practical value. As the era of exponentially increasing classical computing power ends however, one notes that the task of simulating the quantum dynamics of larger molecular systems remains out of reach. Many hope that the accelerating development of quantum hardware [14, 15] will ultimately lead to a generation of novel materials or medically significant molecules that, while

they are created in the laboratory, are first discovered using quantum computers.

Here we are concerned with extending and adapting the ideas from the era of conventional computing into the coming era of quantum-assisted modelling. There is of course a substantial literature addressing this exciting possibility; there are numerous analytic studies targeting both the noisy, intermediate scale quantum (NISQ) era and the longer term era of fully fault-tolerant machines. Moreover, demonstrations of the former have been realised on diverse prototype systems (e.g. [16]). The specific approach examined in this paper involves exploiting quantum computers' ability to efficiently store digital grid representations of molecular (or other multi-particle) wavefunctions in the qubit equivalent of pixels, and efficient time-propagation of the stored state using an implementation of the split-operator method [17, 18] based on quantum logic gates. We refer to this approach as grid-based quantum simulation, while it has also been called '*first quantized chemistry*' as it requires explicit encoding of particle symmetry [19]. It has been considered in a number of previous works [20–24] but is less commonly explored than '*second quantized chemistry*' which involves tracking the occupancy of orbitals. The latter has the merit that small scale demonstrations can operate using only a few qubits. The former may scale better long-term, but require more qubits to represent (several) coordinate registers for a meaningful realisation. Earlier works have investigated the asymptotic scaling [2, 20, 25] as well as exact costs [22] of first-quantised chemistry simulation methods through detailed paper analyses. The complementary approach we take here is to actually numerically perform and assess grid-based simulations, highlighting that analytic error bounds of Trotter-based methods such as the split-operator are typ-

* hans.chan@materials.ox.ac.uk

ically loose compared with methods investigated in the aforementioned works. We thus establish the exact resource costs for simulating systems of modest size to given levels of accuracy, employing established techniques as well as methods that are, to our knowledge, novel. We extrapolate our data to estimate the cost of larger-scale simulations. The results are encouraging, and support the view of several previous works [22, 25, 26] that grid-based methods may indeed be optimal for a majority of post-classical quantum chemistry simulations.

All the numerical results in this paper are simulations of simulations: we emulate a quantum computer which itself is modelling the dynamics of an atomic or atom-like system. Real quantum processors are currently subject to significant noise and too small to offer fault tolerance, therefore we instead exactly-emulate ideal quantum processors using conventional computers from laptops, through GPUs, to high performance computing clusters with up to 1 TB of total memory. The present paper is, to the authors' knowledge, the first to report a full first quantized, real-space grid simulation of the quantum split-operator method propagating electron wavepackets under the Coulomb potential. We do note that simulations of this kind using parabolic harmonic potentials [21, 27], as well as second quantized split-operator propagation of states under the Coulomb potential [26, 28], have recently been performed. Ours is also the first work that presents the visualisations of such atom and atom-like systems which a quantum computer would enable: we can watch as superpositions of eigenstates evolve, and particles are driven by applied electric fields or collide with one another. Since our quantum computers are classically-emulated, obviously the targets of our simulations can only have limited complexity. But the user of a real quantum processor would have access to analogous images (derived by repeated sampling) for far more complex systems: The prospect of directly seeing the quantum dynamics in a molecule unfold is certainly exciting.

In the remainder of this section we provide a brief summary of the relevant techniques and literature from the long history of quantum molecular dynamics simulation with conventional computers. In Section II, we summarise the key ideas used in the grid-based method; these ideas are not novel with respect to prior works but are set out here to provide a self-contained description with consistent notation. Expert readers may care to skip directly to Section III where we set out the specific methods we will explore in the present paper, including approaches that to our knowledge are not reported in prior literature. Then in Section IV we present a range of results from applying those techniques to 2D and 3D systems with single and paired particles. Section V then reflects on those results to discuss the quantum resources required for simulations beyond the reach of emulation, and there we also present a suitable quantum hardware architecture. In Section VI we conclude and remark that the quantum split-operator may also be advantageous in

applications well beyond molecular dynamics.

B. Background to the techniques employed

In seeking to use a controlled quantum machine as our simulator, one may adapt the rich set of numerical methods for solving the time dependent Schrödinger equation on ‘classical’ (pre-quantum) computing platforms. Though many state representation and time propagation schemes exist [8, 29–34], the one of particular interest here is the dynamic Fourier or split-operator Fourier transform (SO-FT) method, which dates back at least as far as the work of Feit and Fleck in 1976 [17, 35, 36]. It uses the Lie-Suzuki-Trotter product formula approach to split the chemical Hamiltonian and time propagator into the two non-commuting kinetic and potential operators, which can respectively be calculated efficiently in two complementary grid-representations of the molecular wavefunction (namely the position-space and momentum-space or plane wave basis). These two representations are related by the discrete Fourier transform. Time propagation of a quantum state is thereby approximated by discretising the total time evolution into small time steps, and rapidly alternating between application of a small time kinetic propagation exactly in the momentum-basis, and a small time potential propagation approximately in the position-basis (often referred to as the discrete variable representation [32]). The scheme is popular and often employed in small-molecule applications, predominantly in propagating nuclear wavepackets on pre-calculated static or time-dependent electronic potential energy surfaces. It has found uses in reactive scattering [37, 38] and excited-state chemistry [13, 39–42], as well as modelling quantum processes like tunnelling [43, 44] and superfluid turbulence [45], and indeed quantum control for designing quantum devices [46, 47]. Application to the relativistic Klein-Gordon equation [48] and non-linear Schrödinger equation [49–51] extends the method’s scope. Its success is partly attributed to the simplicity of its construction, as it only involves updating the vector describing the state, and partly to the low computational price paid for obtaining efficiently converging results [52].

Building on the works of Zalka [53] and Wiesner [54] from the mid 1990’s, it was observed just over a decade ago by Kassal *et al.* [18] that it will be advantageous to store high-dimensional spatial grid representations of molecular wavefunctions on digital quantum computers using qubits that scale only linearly with the degrees of freedom. They also showed that, by exploiting the remarkably efficient quantum Fourier transform (QFT), a quantum version of the SO method will be able to simulate the dynamics of a chemical Hamiltonian in polynomial time. Indeed, the authors challenged the *status quo* of limiting the split-operator method to propagating nuclei on electronic potential energy surfaces. They suggested that, in the fault-tolerant era where plenty

error corrected qubits are available, it would be more efficient to explicitly include both the nuclei and electrons on an equal footing, interacting simply via pairwise Coulomb potentials, thereby completely avoiding the Born-Oppenheimer approximation. This makes it a simple and elegant alternative for studying reactions where full quantum treatment of nuclear and electronic motion is necessary to describe the correct physics [13, 55], advancing the application of the SO method beyond low-dimensional problems.

Although the quantum SO method is oriented towards fault-tolerant quantum computing, it is nonetheless gaining increasing interest. The method has been investigated in the context of propagating one-dimensional Gaussian wavepackets [56], quantum harmonic oscillator [57], non-adiabatic dynamics across coupled, pre-computed potential energy surfaces [21], imaginary-time evolution for molecular geometry optimisation [58] as well as ground state, Gibbs state and partition function determination [24]. We highlight works geared towards understanding the scaling of the quantum SO at mid-to-large scale, specifically that of Jones *et al.* which looked at the resource necessary for a fault-tolerant implementation [25], and the extensive investigation into first-quantized basis-set simulations for chemistry of Su *et al.*, which also included block-encoding the SO Hamiltonian for simulation using qubitization and interaction picture techniques [22]. Other attempts at direct quantum simulations in real-space grids include using finite difference stencils to approximate the kinetic operator in second quantization [26], and in first quantization with propagation based on the truncated Taylor series algorithm [20], as well as a Cartesian component-separated approach [23].

In this work, we explore use of the SO time propagation for capturing the dynamics of single and paired electrons under the influence of the Coulomb potential in 2D and 3D, at a scale that can be numerically simulated exactly using classical high-performance computing resources.

II. THEORETICAL FRAMEWORK

The non-relativistic time evolution of a quantum state is governed by the time-dependent Schrödinger equation (TDSE)

$$\frac{\partial}{\partial t} \Psi(\mathbf{r}, \mathbf{s}, t) = -i\hat{H}\Psi(\mathbf{r}, \mathbf{s}, t), \quad (1)$$

where Ψ is the normalised, complex-valued, many-body wavefunction defined by the spatial \mathbf{r} and spin \mathbf{s} coordinates of the constituent particles (note that throughout this work we are using atomic units where $\hbar = 1$). For the systems of interest here, the Hamiltonian \hat{H} is:

$$\hat{H}_{\text{tot}} = \hat{H}_{\text{kin}} + \hat{H}_{\text{int}}, \quad (2)$$

where

$$\hat{H}_{\text{kin}} = -\sum_{p=1}^P \frac{1}{2m_p} \nabla_p^2 \quad (3)$$

represents the kinetic energy of each of the P particles present, and \hat{H}_{int} encompasses all interactions. In many cases it is convenient to further resolve according to

$$\hat{H}_{\text{int}} = \hat{H}_{\text{U}} + \hat{H}_{\text{V}}, \quad (4)$$

where \hat{H}_{U} represents single-particle interactions with e.g. classical fields, while \hat{H}_{V} represents particle-particle interactions. For interacting charged particles, we would write

$$\hat{H}_{\text{V}} = \sum_{p,q=1; p \neq q}^P \frac{q_{p,q}}{|\mathbf{r}_p - \mathbf{r}_q|} \quad (5)$$

for suitable constants q . In the case of atomic and molecular systems, we could opt to model each nucleus as a full quantum particle in its own right, in which case we may set $\hat{H}_{\text{U}} = 0$ unless there are external e.g. electric or magnetic fields. In their 2008 paper, Kassal *et al.* argue that this is the natural choice given the relatively modest additional resources needed [18]. However, if the nuclei are not treated explicitly within the model, we then opt to model only the electrons and employ classical fields to represent the M nuclear potentials originating at fixed locations \mathbf{R}_m according to

$$\begin{aligned} \hat{H}_{\text{U}} &= \sum_{m=1}^M \sum_{p=1}^P \frac{Q_{p,m}}{|\mathbf{r}_p - \mathbf{R}_m|} \\ &= \sum_{p=1}^P \frac{Q_p}{|\mathbf{r}_p|} \quad \text{for } M = 1 \text{ nucleus at } \mathbf{R}_0 = \mathbf{0} \end{aligned} \quad (6)$$

for suitable values of the electron-nuclear Coulomb energies Q_p or $Q_{p,m}$. In the 2D and 3D atomic simulations we report presently, we consider $M = 1$ nucleus as in the equation above. However the techniques generalise naturally to $M > 1$. External and possibly time-dependent potentials can also be included in the Hamiltonian. Presently we will consider the case of uniform electric field \mathbf{E} , by including within \hat{H}_{U} a term

$$\hat{H}_{\text{E}} = \sum_{p=1}^P Q_p \mathbf{r}_p \cdot \mathbf{E}.$$

Formally the solution to the TDSE is

$$\Psi(\mathbf{r}, \mathbf{s}, t) = e^{-i\hat{H}t} \Psi(\mathbf{r}, \mathbf{s}, 0). \quad (7)$$

For the multi-electron Hamiltonian with more than two particles, it is not possible to analytically evaluate the action of the time evolution operator $e^{-i\hat{H}t}$, and one has to resort to numerical techniques to solve the initial value problem. A practical time propagation method therefore

involves selecting a representation of the state and then applying (some approximation of) the time evolution operator.

We begin by discussing the real-space and momentum-space (k-space) grid representations of a many-body wavefunction, suitably encoded on a quantum computer. We then discuss the SO method for propagating the state forward in time. As explained in Section 1B this method of exploiting quantum computers, explored by earlier authors in [18, 21, 22, 25, 27, 56–58], is an adaption of the classical computing methods developed in [17, 35, 36, 59].

A. Representations in k-space and in real-space

We wish to model the dynamics of a multi-particle problem with Hamiltonian \hat{H}_{tot} . We divide the qubits of our quantum computer into registers, each associated with one of our particles. Further, we divide each register into sub-registers corresponding to the dimensions of the model; in the present work we consider 2D and 3D models. In general the registers corresponding to different particles need not be the same size, nor do the sub-registers need to match in size. Indeed, many scenarios would naturally suggest variations, e.g. a nuclear particle may be adequately modelled within a smaller ‘box’ than its bound electrons, or we might model e.g. a solid-state interface where the z -direction is restricted versus the x - y plane. However in the present work we confine ourselves to considering cases where all sub-registers are of equal size, so that the computer has size of order $O(dPn_r)$ (neglecting ancillas used in temporary roles) where d is the spatial dimensionality of the problem (2 or 3 for us), P is the number of particles (1 or 2 in our simulations), and n_r is the number of qubits in each sub-register. For this last parameter we will consider a range of choices $6 \leq n_r \leq 14$ to determine the required resolution for accurate modelling in various scenarios.

Suppose that we wish to model a 1D system which is well-localised within a region $-\frac{L}{2} < x < \frac{L}{2}$. We refer to this region as the ‘simulation box’, and thus L is the box width. For now we will assume that the real system’s state has negligible amplitude outside this box, both initially and throughout the anticipated simulation. This condition is relaxed presently when we consider scattering and ionization. We choose a k -space (or ‘spectral’) representation in which a plane wave basis state of the modelled system is represented by a state of the computer’s sub-register as follows

$$\phi_k(x) = L^{-\frac{1}{2}} \exp\left(\frac{i2\pi kx}{L}\right) \leftrightarrow |k\rangle. \quad (8)$$

Here k is an integer and $|k\rangle$ refers to the computational basis state which, regarded as a binary string, corresponds to k . A negative value of k implies the two’s complement binary representation. The same meaning is intended whenever we write a Latin letter in a ket.

Defining $\rho = 2^{n_r-1}$ and noting that we have 2ρ basis states in our computer’s sub-register, a natural choice¹ for the allowed k is to run from $-\rho$ through zero to $\rho-1$. Therefore state Ψ of a 1D modelled system would be represented by our sub-register $|\psi\rangle_x$ according to

$$\Psi = L^{-\frac{1}{2}} \sum_{k=-\rho}^{\rho-1} a_k e^{i2\pi kx/L} \leftrightarrow |\psi\rangle_x^{\text{KS}} = \sum_{k=-\rho}^{\rho-1} a_k |k\rangle. \quad (9)$$

Formally this is a bijective encoding of the Fourier components of the modelled state as computational basis states. Throughout this paper we will use the double-headed arrow \leftrightarrow to indicate such a mapping: encoding of Fourier components (modelled system) on the left, to the quantum computer’s state on the right. Practically, the meaning of \leftrightarrow is that when we apply operations to the qubit register, we will do so with our mapping to the modelled system in mind; this will be clear presently. We use the superscript KS for k -space. For simplicity we do not put ‘ $p = 1$ ’ in the subscript, but we understand this is for a given particle.

It would be possible to exclusively use the k -space representation in grid-based modelling, and indeed there may be advantages to doing so [2, 22]. However, in the present study we employ the approach in which the registers are periodically transformed into a ‘dual’ representation [26]. Specifically, we apply to each sub-register a quantum Fourier transform (QFT) denoted by U_{QFT} and defined by the circuit shown in Appendix A. The sub-register as a whole will be transformed as

$$|\psi\rangle_x^{\text{RS}} = U_{\text{QFT}} |\psi\rangle_x^{\text{KS}} = \sum_{n=-\rho}^{\rho-1} b_n |n\rangle, \quad (10)$$

where

$$b_n = \frac{1}{\sqrt{2\rho}} \sum_{k=-\rho}^{\rho-1} \exp\left(i \frac{n\pi}{\rho} k\right) a_k. \quad (11)$$

The superscript RS indicates that we call this the real-space mode of representation; we will justify this label momentarily. When we wish to return to the original, k -space representation we employ the inverse QFT:

$$|\psi\rangle_x^{\text{KS}} = U_{\text{QFT}}^{-1} |\psi\rangle_x^{\text{RS}} = \sum_{k=-\rho}^{\rho-1} a_k |k\rangle, \quad (12)$$

¹ With this choice subsequent expressions have a compact form; but the mapping from plane wave to computational basis state can employ a shift so that k runs from (say) 0 to $2\rho - 1$. Indeed, in the simulations performed presently, underlying code using both conventions is tested. The eventual choice will be a matter of optimising circuit depths in the real quantum processor. As a further aside, note that an interesting alternative to Eqn. 8 is to use $2\pi(k + 1/2)$ rather than $2\pi k$. Then any state of the model must satisfy $\Psi(x = L/2) = -\Psi(x = -L/2)$.

where of course

$$a_k = \frac{1}{\sqrt{2\rho}} \sum_{n=-\rho}^{\rho-1} \exp\left(-i\frac{k\pi}{\rho}n\right) b_n. \quad (13)$$

Note that the number of gates required for the QFT scales only as the square of the number of qubits.

We can now ask, what wavefunction must be represented by each computational basis state $|n\rangle$ appearing in Eqn. (10)? We can answer by setting all b coefficients to zero except for b_n (i.e. $b_m = \delta_{m,n}$), transforming to k -space using Eqns. (12) and (13), and finally referring back to our original declared mapping in Eqn. (8). We find that each $|n\rangle$ maps to a wavefunction peaked at (but not strictly localised around) the spatial point $x_n = \frac{nL}{\rho^2}$. The upper left panel of Fig. 1 shows a plot for the case that $n_r = 6$, $n = 0$. Specifically, the inferred mapping is

$$\phi_n(x) = P_{n_r}^n(x) \leftrightarrow |n\rangle \quad (14)$$

with

$$P_{n_r}^n(x) = \exp\left(-\frac{i\pi x'}{L}\right) \sqrt{\frac{2}{\rho L}} \sum_{j=1}^{\rho} \cos \frac{\pi(2j-1)x'}{L} \quad (15)$$

where $x' = x - x_n$ with $x_n = \frac{nL}{2\rho}$, and $\rho = 2^{n_r-1}$.

Within our simulation box $-\frac{L}{2} < x < \frac{L}{2}$, the function $P_{n_r}^n(x)$ serves the role of an approximation or ‘smear’ of the Dirac delta function $\Delta(x-x_n)$, the sharpness increasing as 2^{n_r} tends to infinity.² Shortly we will motivate the informal term ‘pixel functions’ for our $P_{n_r}^n(x)$. The separation between the peaks of adjacent pixel functions, e.g. $x_2 - x_1$, is $\delta r = L/2^{n_r}$, and we now define the model’s *spatial resolution* as the reciprocal of this quantity, i.e. as the number of pixel functions per unit distance:

$$\delta r^{-1} = 2^{n_r}/L. \quad (16)$$

The duality between the k -space and real-space representations under the Fourier transform has been explored in multiple studies including Refs. [8, 26, 60, 61]. Whereas our k -space representation maps qubit basis states to wavefunctions with sharp values of k , in the dual representation the mapping is to wavefunctions that are not perfectly sharp around points in real-space. Their Dirac-like character is their key property; indeed one can analyse techniques and protocols *as if* they were Dirac functions, secure in the knowledge that in the high-resolution limit this becomes exact. This insight is employed in several grid-based quantum simulation papers [18, 21, 24].

² Strictly speaking it tends to a Dirac comb since the function is periodic beyond the simulation box. The function can also be recognised as a difference of two Dirichlet kernels; e.g. for $L = \pi$

$$P_{n_r}^0(x) \propto D_{2\rho}(x) - D_\rho(2x).$$

For the present paper, we note that there are four key observations regarding the set of spatial wavefunctions $P_{n_r}^n(x)$ which are true for any given qubit count n_r . They are mutually orthogonal, each is peaked at x_n , and importantly each is zero at all points $x_{m \neq n}$. That is to say, where a given spatial wavefunction has its primary peak, all other wavefunctions are strictly zero. Finally, we note that at its peak each function has the value

$$P_{n_r}^n(x = x_n) = \frac{1}{\sqrt{\delta r}}.$$

These properties mean that a suitable decomposition to represent any 1D wavefunction $\Psi(x)$ in our quantum register is very intuitive:

$$\Psi(x) \approx \frac{C}{\sqrt{\delta r}} \sum_{n=-\rho}^{\rho-1} \Psi(x_n) P_{n_r}^n(x) \leftrightarrow \frac{C}{\sqrt{\delta r}} \sum_{n=-\rho}^{\rho-1} \Psi(x_n) |n\rangle \quad (17)$$

i.e. the required amplitude of $|n\rangle$, the state representing the wavefunction peaked at point x_n , is found simply by sampling the target wavefunction at that point. Here C is a normalisation constant that will be close to unity providing that (a) the target wavefunction has negligible amplitude outside of the simulation box and (b) the target wavefunction varies slowly with respect to δr . If C differs significantly from unity then the model does not properly capture $\Psi(x)$, however C can be above or below unity and $C = 1$ does not imply optimality.

So far we have considered the role of one sub-register, which corresponds to one dimension of a given particle. The generalisation to 2D or 3D is the natural one: the sub-registers tensored together to form the complete representation of a given particle. The 3D analogue of Eqn. (17) is

$$\begin{aligned} \Psi(\mathbf{r}) &\approx C\delta r^{-\frac{3}{2}} \sum_{n,m,l=-\rho}^{\rho-1} \Psi(x_n, y_m, z_l) P_{n_r}^n(x) P_{n_r}^m(y) P_{n_r}^l(z) \\ &\leftrightarrow C\delta r^{-\frac{3}{2}} \sum_{n,m,l=-\rho}^{\rho-1} \Psi(x_n, y_m, z_l) |n\rangle |m\rangle |l\rangle. \end{aligned} \quad (18)$$

(Our numerics will include models of 2D systems where of course we omit the third sub-register.) When we generalise to represent a P -particle wavefunction $\Psi(\mathbf{r}_1, \dots, \mathbf{r}_P)$ we need only extend in the natural fashion,

$$C\delta r^{-\frac{3P}{2}} \sum_{\{n_1 \dots l_P\}=-\rho}^{\rho-1} \Psi(x_{n_1}, y_{m_1}, \dots, z_{l_P}) |n_1\rangle |m_1\rangle \dots |l_P\rangle. \quad (19)$$

Figure 1 shows the ground state of 2D hydrogen plotted from the analytic solution (lower left side, cyan) and constructed as an approximation using the 2D variant of Eqn. 18 for the case of $n_r = 6$ qubits per sub-register (right side, orange). This wavefunction will be discussed in more detail presently, and is one of the initial states we will use in our numerical simulations. It is apparent by inspection that the approximation is good over

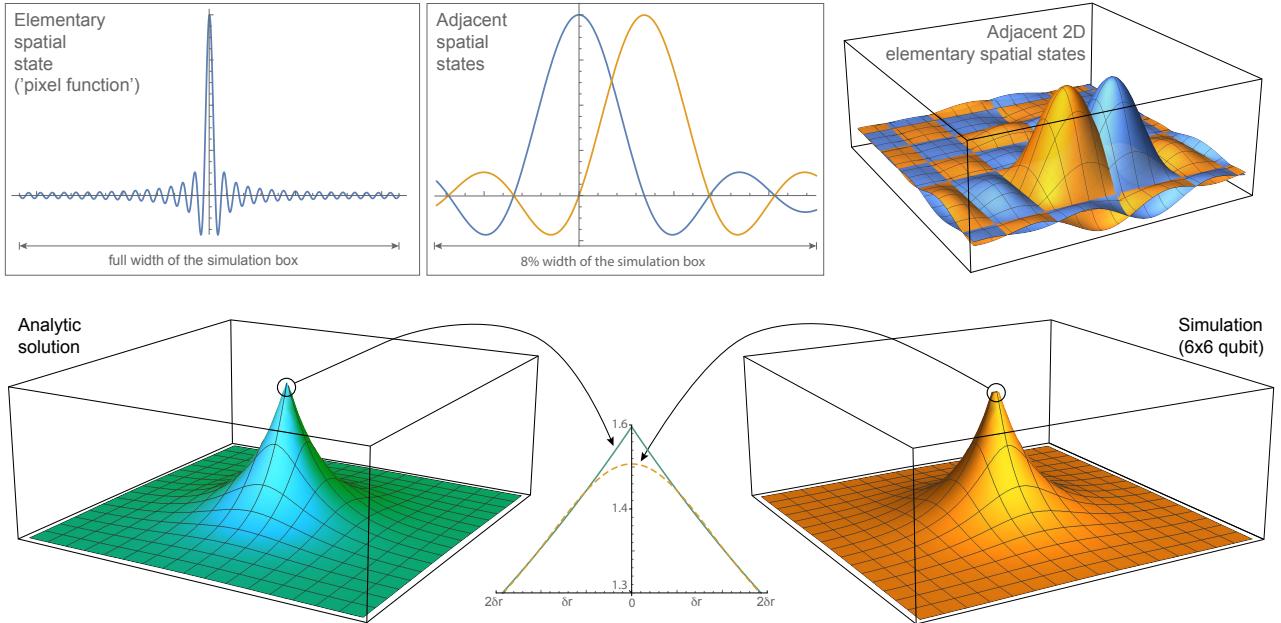


FIG. 1. Spatial wavefunctions modelled by a quantum computer. Upper panels show the component states: Left plot is ‘pixel function’ $P_{n_r}^n(x)$ with $n_r = 6$ qubits, and $n = 0$ (neglecting its complex phase, see Eqn. 15). The central upper panel shows the adjacent pixel functions $n = 0$ and $n = 1$; where a function has its primary peak, all others are zero. Upper right panel shows 2D pixel functions formed from products of the 1D cases. Lower panels illustrate the accuracy with which a state can be represented. The lower left (blue-green) figure shows the exact ground state $\Psi_{0,0}$ of 2D hydrogen. The lower right (orange) plot is the same state as modelled by a $6 + 6 = 12$ qubit quantum computer. The fidelity is very high as noted in the text; the significant discrepancy is a $\approx 6\%$ variation localised at the very centre, as highlighted by the central plot.

the majority of the simulation box. The analytic state of course extends beyond the box, where our modelled state is undefined; taking the modelled state to be zero there, the fidelity of the modelled state with respect to the analytic state (integrating over all space) is 0.99946. If instead we project the analytic state into the box and renormalize it, then the infidelity between the model and the analytic solution falls below 10^{-4} . The discrepancy is primarily localised at the Coulomb singularity, where the analytic expression has a gradient discontinuity. Of course, the reconstruction must have a continuous gradient (since all constituent functions have this property). The sharpness of the approximation is characterised by δr^{-1} and so increases exponentially with n_r . The upper panels in Fig. 1 show examples of $P_{n_r}^n(x)$ for the same case of $n_r = 6$ (but with the phase component removed so that the plots can be seen without an imaginary axis). Recall that in our two’s complement notation, integer indices can denote negative or positive coordinates; the register state $|00..0\rangle = |n = 0\rangle$ corresponds to the wavefunction peaked at the origin (the blue lines in the left and middle panels). The orthogonality property is clear from the middle panel, showing $|n = 0\rangle$ in blue and $|n = 1\rangle$ in orange.

Intuitively, one can think of these spatial wavefunctions analogously to pixels as used in conventional digital photographs: A pixel has a single colour corresponding to the (real, continuous) scene sampled at that point.

Similarly each spatial basis state $|n, m, l\rangle$ in Eqn. 18 is associated with a single complex number corresponding to the continuous wavefunction sampled at that point. A photo should have sufficient pixels that all features of interest to the viewer are fully captured, and our simulation should have sufficiently many spatial basis states that all features of the wavefunction are adequately captured. Because of this close analogy, we sometimes refer to the $P_{n_r}^n$ wavefunctions as ‘pixel functions’. As we will see presently, a moderate resolution δr^{-1} can prove to be sufficient and thus the number of qubits required can be remarkably low.

From Fig. 1 it is obvious that pixel wave functions resemble standard sinc functions; indeed if we increase the number of qubits n_r while commensurately increasing L so that the resolution δr^{-1} is held constant, then the pixel function $P_{n_r}^0$ tends to a sinc function in the interval $[-\frac{L}{2}, \frac{L}{2}]$. This is reminiscent of Whittaker-Shannon interpolation and indeed there is a close association between the grid-based quantum model and the theory of signal processing.

The representation described above involves simple Cartesian grids in momentum-space and position-space, with the conversion effected by a QFT; in principle, other coordinate systems could be used (cylindrical, spherical etc) or alternative transformations [60]. In the present paper we will consider the hybrid coordinates needed to evaluate particle-particle interactions.

A spin degree of freedom is trivial to add into the simulation: for spin one-half particles we would simply require a single additional qubit per complete register, while a spin- S particle would require $\ln(S)$ qubits. However in the numerics we describe presently, we do not consider any Hamiltonians with a spin-dependent term and therefore we do not model the particle's spin. Importantly, particle symmetry is properly preserved by the split-operator method that we now discuss; symmetry can be confirmed at any stage using e.g. a SWAP test [62] on the registers representing the relevant pair of particles.

B. Split-Operator Propagation

We now consider the means to simulate the dynamics induced by our multi-particle Hamiltonian. The split-operator scheme first discretises the global time $[0, t]$ evolution into short time intervals $t = N\delta t$, i.e.

$$e^{-i\hat{H}t} = \underbrace{e^{-i\hat{H}\delta t} e^{-i\hat{H}\delta t} \dots e^{-i\hat{H}\delta t}}_{N \text{ times}}. \quad (20)$$

The method then approximates the short time propagator by Trotterisation, thus splitting the Hamiltonian into kinetic and interacting parts

$$\begin{aligned} e^{-i\hat{H}\delta t} &= e^{-i(\hat{H}_{\text{kin}} + \hat{H}_{\text{int}})\delta t} \\ &= e^{-i\hat{H}_{\text{kin}}\delta t} e^{-i\hat{H}_{\text{int}}\delta t} + O(\delta t^2) \\ &= e^{-i\hat{H}_{\text{kin}}\delta t/2} e^{-i\hat{H}_{\text{int}}\delta t} e^{-i\hat{H}_{\text{kin}}\delta t/2} + O(\delta t^3) \end{aligned} \quad (21)$$

Both the first-order ($O(\delta t^2)$) and second-order ($O(\delta t^3)$) Trotterisation split-operator have well-documented numerical advantages [8, 29, 63, 64], and higher order Lie-Trotter-Suzuki sequences can also be relevant [65, 66]. For simplicity, and in order to compare different techniques on an equal basis, in this work we focus on the first-order split-operator – however the methods we will employ are equally compatible with any Trotter sequence. Our interaction Hamiltonian \hat{H}_{int} involves Coulomb potential(s) and, in one study, an additional external electric field. In any SO simulation where $[\hat{H}_{\text{kin}}, \hat{H}_{\text{int}}] \neq 0$, the dynamics will be imperfectly modelled and gives rise to the Trotter error terms in Eqn. (21); we will discuss the accumulation of errors in Section III D.

The real and k -space Cartesian grid dual representations (detailed in Section II A) are natural options for state representation when computing the approximate time evolution operator of Eqn. (21). In the k -space representation, the kinetic part of the Hamiltonian \hat{H}_{kin} is separable and exactly local (diagonal). In the real-space representation the interaction part of the Hamiltonian \hat{H}_{int} , is approximately diagonal.

On a quantum computer it is relatively efficient to switch between the two representations using the QFT; the number of gates required per sub-register is quadratic in its number of qubits n_r , and all sub-registers can be transformed independently. Thus we can implement the

two parts of the SO cycle each in their preferred, diagonal basis. We perform the following to approximate a time step δt using our quantum computer:

$$|\psi(t + \delta t)\rangle^{\text{RS}} = U_{\text{SO}}(\delta t) |\psi(t)\rangle^{\text{RS}}$$

where we recall that the superscript RS denotes the real space representation, and

$$U_{\text{SO}}(\delta t) = e^{-iD_{\text{int}}\delta t} \left(U_{\text{QFT}}^\dagger e^{-iD_{\text{kin}}\delta t} U_{\text{QFT}} \right). \quad (22)$$

Here D_{kin} and D_{int} are diagonal real matrices as explained above, and U_{QFT} is the quantum Fourier transform applied to all sub-registers.

We now discuss the evaluation of these operators on quantum computers in detail. From Eqn (3) we observe that (regardless of basis) any period of evolution under purely the kinetic part \hat{H}_{kin} , i.e. any operator

$$U_{\text{kin}}(\delta t) = e^{-i\hat{H}_{\text{kin}}\delta t}$$

will separate exactly into a product of operators acting independently on each particle and in each dimension:

$$U_{\text{kin}}(\delta t) = \prod_{p=1}^P \prod_{q \in \{x,y,z\}} \exp\left(i \frac{\hbar^2 \delta t}{2m_p} \frac{\partial^2}{\partial q^2}\right)$$

because these components commute. Now suppose that a given quantum sub-register is currently in the k -space representation defined by Eqn (9), and recall Eqn. (8) for the meaning of each computational basis state, viz.

$$\phi_k(x) = L^{-\frac{1}{2}} \exp\left(\frac{i2\pi kx}{L}\right) \leftrightarrow |k\rangle.$$

We see that the proper way to account for the action of $U_{\text{kin}}(\delta t)$ on the molecule is to introduce phases onto our computational basis states according to

$$|k\rangle \Rightarrow e^{iC\delta t k^2} |k\rangle \quad (23)$$

with constant $C = 2\hbar^2\pi^2/(L^2 m_p)$. Thus our task is simply to apply phase operations to our quantum computer, independently for each sub-register (and in parallel if our hardware has that capability). Each element of the matrix D_{kin} appearing in Eqn. (22) is just the sum of $C(k_x^2 + k_y^2 + k_z^2)$ over all particles. It is trivial to implement the required phases as, for example, a sequence of single- and two-qubit phase gates (see for example [21, 25]). The number of gates required goes as the square of the number of qubits in the register.

We now consider the challenge of modelling evolution under the potential parts of \hat{H}_{tot} . Consider first \hat{H}_U , which represents the energy of each particle in classical fields. We are primarily interested in an attractive Coulomb potential representing a nucleus (although we discuss a variation including a static electric field presently). For the Coulombic case we write

$$\hat{H}_U = \sum_{p=1}^P \frac{Q_p}{|\mathbf{r}|_p} = Q \sum_{p=1}^P \frac{1}{\sqrt{x_p^2 + y_p^2 + z_p^2}}. \quad (24)$$

Here we took Q outside of the sum since all Q_p are the same in our atomic and molecular systems of interest; however there would be no difficulty in retaining distinct values. As before, we can write an time evolution operator

$$U_U(\delta t) = \prod_{p=1}^P \exp\left(-i \frac{Q \delta t}{\sqrt{x^2 + y^2 + z^2}}\right).$$

The operations are independent between the registers corresponding to different particles, but not independent between sub-registers assigned to a given particle.

We will be in the real space representation when we implement the corresponding dynamics. Recall that the computational basis states of each sub-register will thus correspond to locally-peaked 1D single-particle wavefunctions specified in Eqn. (14), viz.

$$\phi_n(x) = P_{nr}^n(x) \leftrightarrow |n\rangle$$

with each such ‘pixel function’ having its primary peak at $x_n = nL/2^{n_r} = n\delta r$. Similar expressions describe the roles of the y sub-register states $|m\rangle$, peaked at y_m , and the z sub-register states $|l\rangle$ peaked at z_l . These pixel functions were depicted in Fig. 1.

We proceed to apply a period of time evolution under \hat{H}_U using an approximation that *would* be exact in the limit of infinite spatial resolution. In that case we could apply a series of phase changes to our quantum registers to properly describe any time step δt .

$$\begin{aligned} |n\rangle |m\rangle |l\rangle &\Rightarrow \exp\left(\frac{-iQ \delta t}{\sqrt{x_n^2 + y_m^2 + z_l^2}}\right) |n\rangle |m\rangle |l\rangle \\ &= \exp\left(\frac{-iQ \delta t}{\delta r \sqrt{n^2 + m^2 + l^2}}\right) |n\rangle |m\rangle |l\rangle \quad (25) \end{aligned}$$

Comparing Eqn. 23 and Eqn. 25, we see that the task of implementing the latter is only modestly more complex than the former: Now, we need to compute on an entire register of all three sub-registers and the phases required are inverse-square-root instead of merely square. Efficient evaluation of functions such as the inverse-square-root on quantum computers is an active area of development. In particular, Kassal *et al.* [18], Jones *et al.* [25] and Häner *et al.* [67] proposed procedures for reversible computation of the inverse-square-root using fixed-point arithmetic and Newton-Raphson iteration. Polynomial function interpolation using quantum read-only memory (QROM) type circuits is another promising approach [68]. Recent work from Poirier *et al.* [23, 69] combined with other formulations of the Coulomb potential may also ease the efforts of computing the inverse-square-root. We refer to [25, 67] for resource estimates, but here it suffices to note that the number of qubits required can potentially scale merely quadratically with the number of qubits n_r . We discuss this further in Appendix B.

We motivated the expression above by noting that this would be the exactly correct process if our spatial states

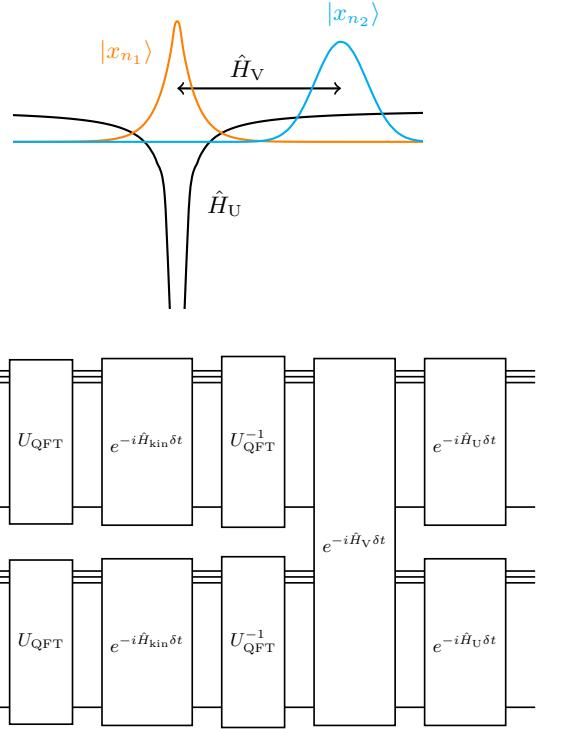


FIG. 2. Above: Illustrating the propagation of an atomic system in real-space on a quantum register. One electron is modelled as a Gaussian wavepacket, and the other as the bound ground state hydrogenic solution. The two interact via a repulsive Coulomb potential, and the atomic nucleus is modelled as a classical attractive Coulomb potential. Note that the two electrons will have to be antisymmetrised. Below: Schematic of the corresponding quantum circuit for a single split-operator step propagation.

were Dirac delta functions. Those states in fact have finite spatial extent but we can expect that, for any sufficiently short time δt and given a smoothly varying potential, an adequate spatial resolution δr^{-1} will result in dynamics that converge to the exact behaviour. However the Coulomb potential is singular at $\mathbf{r} = \mathbf{0}$ and consequently we will need to investigate the behaviour in this region carefully. As we presently explain, it is possible to augment the basic SO cycle of kinetic and potential evolution with a third phase, whose purpose is to stabilise behaviour at the core singularity.

Finally we must consider the part of the Hamiltonian corresponding to particle-particle interactions, \hat{H}_V as defined in Eqn. 5. This is straightforward to handle using the same approximation above, i.e. by updating our registers *as if* the states they represent are Dirac delta functions. Then the operator,

$$U_V(\delta t) = \exp\left(i\delta t \sum_{p,q=1; p \neq q}^P \frac{q_{p,q}}{|\mathbf{r}_p - \mathbf{r}_q|}\right) \quad (26)$$

will be approximated as having diagonal form (and will therefore commute with the $U_U(\delta t)$ operator). Consider the two registers, each composed of three sub-registers, which represent a given pair of particles $p = 1$ and $p = 2$. Then basis states will be updated as

$$|n_1 m_1 l_1\rangle |n_2 m_2 l_2\rangle \Rightarrow e^{-i\Theta} |n_1 m_1 l_1\rangle |n_2 m_2 l_2\rangle$$

where

$$\begin{aligned} \Theta &= \frac{Q \delta t}{\sqrt{(x_{n_1} - x_{n_2})^2 + (y_{m_1} - y_{m_2})^2 + (z_{l_1} - z_{l_2})^2}} \\ &= \frac{Q \delta t}{\delta r \sqrt{(n_1 - n_2)^2 + (m_1 - m_2)^2 + (l_1 - l_2)^2}} \end{aligned} \quad (27)$$

One means of computing the required phases is discussed in Section V B. We would pair every particle with a partner and perform the following process (which can occur simultaneously over each pair of registers), before repeating with a different pairing, and so on until all pairings are considered. For each pair, we perform the computation

$$\begin{aligned} &|n_1, m_1, l_1\rangle |n_2, m_2, l_2\rangle \Rightarrow \\ &|n_1 - n_2, m_1 - m_2, l_1 - l_2\rangle |n_2, m_2, l_2\rangle \end{aligned} \quad (28)$$

which is a straightforward instance of quantum arithmetic on pairs of sub-registers. One suitable method uses the QFT which of course our machine will already have been optimised for [70]. This approach is frugal in terms of the qubit count: we need only a single additional qubit to fully represent each difference $n_1 - n_2$, etc (as the maximum magnitude is within a factor of two of the maximum magnitudes of n_1 and n_2). With the registers in this form, we can simply use the same procedure employed for the single-particle phases as specified in Eqn. (25). Once the desired phases are applied, we perform the reverse of the computation Eqn. (28) and then repeat the whole process with another set of particle pairings. For a system of P particles, the total number of particle pairings is $\frac{1}{2}P(P - 1)$, however as we discuss in Section V B there is a natural parallelism so that the time cost should scale only linearly with P . Moreover, there are methods to trade greater hardware resources to reduce the time cost; Jones *et al.* introduced a parallelised scheme for speeding up the computation of the particle pairings using only $O(\log P)$, or even $O(1)$, circuit depths [25]. Regardless of the implementation method used, the elements of the diagonal matrix D_{int} are thus simply sums of phases appearing in Eqns. (25) and (27).

In the approach just described, we are finding the relative coordinates of each particle pairing. There was no need to compute the complete transformation by mapping the second particle's register to $|n_1 + n_2, m_1 + m_2, l_1 + l_2\rangle$. However, this would have been possible. A motivation for completing this full transformation is that general, non-diagonal operations could be performed if we wished; this observation relates to the augmented split-operator concept developed presently.

III. TECHNIQUES FOR QUANTUM SPLIT-OPERATOR MODELLING

We now explore techniques and important considerations for the split-operator time propagation scheme. We discuss some well-established classical methods recast into the quantum setting, some that are only convenient for classical simulation, and some which have no classical analogue.

A. Observables and Phase Estimation

Here we review the standard technique of measuring energy via phase estimation; the experienced reader may care to skip this and the following section.

The energy expectation is the most ubiquitous observable in quantum chemistry. The first method to extract it from a dynamics simulation is to split the Hamiltonian and calculate the kinetic energy expectation in k -space, followed by the potential energy expectation in real space

$$\langle E \rangle = \langle \Psi | U_{\text{QFT}}^\dagger \hat{H}_{\text{kin}} U_{\text{QFT}} | \Psi \rangle + \langle \Psi | \hat{H}_{\text{int}} | \Psi \rangle \quad (29)$$

where here $|\Psi\rangle$ is understood to be the spatial representation. In principle, it is efficient to compute this value ‘on-the-fly’ in a simulation on a traditional computer because the full state is readable from memory, but inefficient on a quantum computer because the state will have to be extracted through repeated sampling measurements of the state register. These efficiency concerns are in practice irrelevant because, as we will demonstrate in Section IV C, the aforementioned non-conservation of energy due to the Trotter error renders the method unsuitable for efficiently extracting the correct physics.

An alternative is to extract the energy of a state from the autocorrelation function, which is a discrete-time series of the inner product of a propagated state with the initial state. The phase of the signal is in fact the global phase acquired by the initial state during time propagation, and thus also contains the eigenenergies of the state as follows: Suppose we time propagate an eigenstate Ψ_n of \hat{H}_{tot} with energy E_n . The corresponding autocorrelation signal is

$$\langle \Psi_n(t) | \Psi_n(0) \rangle = e^{-iE_n t}. \quad (30)$$

In the classical SO, we can compute the inner product of the current state with a copy of the initial state at regular time steps, then numerically fit a sinusoidal signal to the autocorrelation to find the phase. The absolute value of the autocorrelation of an eigenstate propagation

$$C(t) = |\langle \Psi(t=0) | \Psi(t) \rangle| \quad (31)$$

should be close to unity and in this work we use it as one measure of a simulation’s veracity.

The same information can be efficiently extracted in the quantum setting using phase estimation. Phase estimation through the use of ancilla qubits [71] is one of the

fundamental techniques employed in diverse applications of quantum computing, and its utility in the context of split-operator and first quantized simulation is well recognised (see e.g. [18]). Even with a single ancilla [57, 72] it permits us to learn the otherwise-unobservable global phase that an eigenstate acquires as it propagates in time.

The estimation scheme that we will employ is summarised in Figure 3. We conditionally apply N split-operator steps $U^N(\delta t)$ controlled by an ancillary qubit in the $|+\rangle$ state. At the N^{th} step, we measure the ancillary quit in the $|+\rangle$ basis, at which point the state is discarded (but cf. the state preparation variant discussed presently). We see that, if the initial state is a pure eigenstate, the global phase information is encoded in the relative phase between the $|0\rangle$ and $|1\rangle$ state of the ancillary qubit

$$\begin{aligned} |+\rangle |\Psi_n(t=0)\rangle &\Rightarrow \frac{1}{\sqrt{2}}(|0\rangle |\Psi_n(0)\rangle + |1\rangle |\Psi_n(t)\rangle) \\ &= \frac{1}{\sqrt{2}}(|0\rangle + e^{-iE_n t}|1\rangle) |\Psi_n(0)\rangle. \end{aligned} \quad (32)$$

By resetting the time propagation and repeating the ancilla measurement at different time steps, the probability of finding the ancillary qubit in state $|+\rangle$ fluctuates as the phase. We use this to extract a periodic time signal $a(t)$ where the frequency is proportional to the energy of the simulated wavefunction

$$a(t) = \cos^2\left(\frac{E_n}{2}t\right). \quad (33)$$

Because the ancilla projection probability is statistical in nature, in practice the time propagation and measurement will have to be repeated multiple times in this single-ancilla method. In the numerical results we discuss presently, we show the exact evolution of $a(t)$ plotted for every time point – this is straightforward since we use classically-emulated quantum processors. In a physical implementation, the sampling nature of the procedure gives an energy error

$$a(t) = \cos^2\left(\frac{E_n \pm \epsilon}{2}t\right), \quad (34)$$

where ϵ should shrink as $1/\sqrt{m}$ where m is the number of repetitions [72]. With the real device, one would design a protocol for extracting information in the most frugal manner, perhaps using a Bayesian approach to target the data points which will yield the most information the curve. Moreover, for certain purposes such as optimising an initial state to most closely approximate a given eigenstate, it may suffice to characterise only the initial part of the $a(t)$ curve.

The numerical studies we report presently only use this single-ancilla method, and only in cases where there are at most two eigenstates present. For completeness we describe the standard extension to full phase estimation in Appendix C where we also note the use of classical Fourier analysis to extract features if indeed the hardware is limited to a single ancilla.

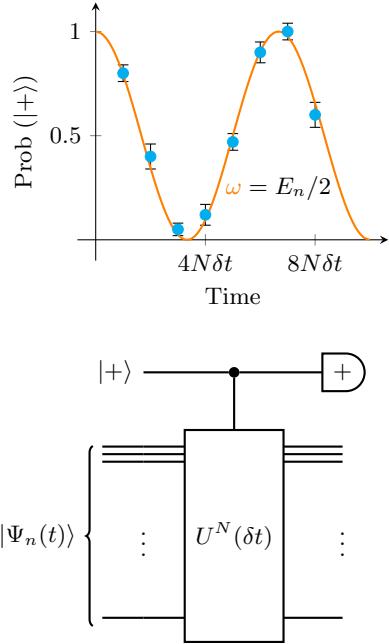


FIG. 3. The single-ancilla iterative phase estimation scheme, employed in this work for extracting the global phase acquired by an initial eigenstate during its (emulated) quantum split-operator time propagation. Above: the global phase is encoded in the probability of measuring an entangled ancilla qubit, which controls the application of N split-operator cycles, in the $|+\rangle$ state. To obtain this probability one must repeat the propagation and measurement at each point in time where we wish to sample the signal. Below: Circuit for the single-ancilla iterative phase estimation.

B. State Preparation

Here we review the challenge of state preparation, including the preparation (and editing) of a state via phase estimation; the experienced reader may care to skip to the following section.

Direct quantum simulation solves an initial value problem. We therefore require a method for preparing a desired initial state on the quantum computer with no more than polynomial complexity. The state we wish to prepare is that given by Eqns. 18 and 19. Those simple expressions mean that, if we can analytically describe the state we wish to prepare, the preparation is quite trivial for the emulated quantum processors used in the present work. However for real devices the problem of state preparation is challenging even given an analytic expression, especially when one considers the demands of overall particle exchange symmetry.

In Appendix D we present an overview of the ideas in the literature for preparing a known state on a quantum simulator, including grid-based representations, as well as generation of states that are antisymmetric under exchange of particle registers. In the present paper we will simply load known, analytic states into our emulated

quantum computers.

A still-greater challenge is that of preparing a state which we cannot even describe classically, but whose properties we can list (e.g. it is the ground state or has a given excited energy). There are various relevant ideas in the literature, including adiabatic evolution from an initial, easy to prepare, state to the target [73], as well as probabilistic imaginary time evolution [24]. In the present paper we instead demonstrate a single-ancilla variant of state preparation by phase estimation [71, 74, 75]. Our means of tracking accumulated phase has been described in Section III A, and the generalisation here is simply to regard the ancilla measurement as a mid-point rather than an end-goal.

Suppose we can prepare an initial state which contains a superposition of eigenstates of differing energies including our desired state. Let us assume that we know the energy E of the desired state; if not, the method of III A can reveal the spectrum of energies by classical Fourier analysis (Appendix C). We prepare our ancilla in the $|+\rangle$ state and then perform condition evolution previously described, choosing a total time $T = N\delta t = 2n\pi/E$ for some integer n . We then measure the ancilla. If the initial state had been exactly the desired state then the ancilla would now certainly be found in the $|+\rangle$. An initial state with a different energy $E_k \neq E$ would in general not yield the $|+\rangle$ with certainty but rather with probability $\cos^2(TE_k)$. We therefore demand a $|+\rangle$ outcome, and restart if we do not see it. On success, the *post-selected* state on the main register has a boosted amplitude associated with the desired state because other states have been suppressed by the factor $\sin^2(TE_k)$. We repeat the process to probabilistically purify the desired state (or if degeneracy is present, a state in the degenerate subspace with energy E). Our probability of success is precisely the total probability associated with the target state's component of the initial state, therefore it is important to prepare an initial state having reasonable overlap with the target. The total time required will depend on the inverse of the energy gap $E_k - E$ between the desired state and the nearest unwanted component.

An interesting variant of the above, is the case that we wish to *remove* a known state from a superposition. This might be useful in analysing a state after some (possibly non-unitary) evolution has occurred. This is the variant that we demonstrated with emulated quantum computers in Section IV D. To remove state Ψ_κ with eigenvalue E_κ , we again execute a controlled time propagation of the initial state as before but require measurement at T_κ , such that $e^{-iE_\kappa T_\kappa} = -1$, thus

$$t_\kappa = N_\kappa \delta t = \frac{\pi}{E_\kappa} \quad (35)$$

Now the undesired component has zero probability of yielding a $|+\rangle$ from the ancilla measurement, so that post-selecting on that outcome will entirely eliminate it from the register. This is the variant of state preparation (or more generally, state editing) which we explore

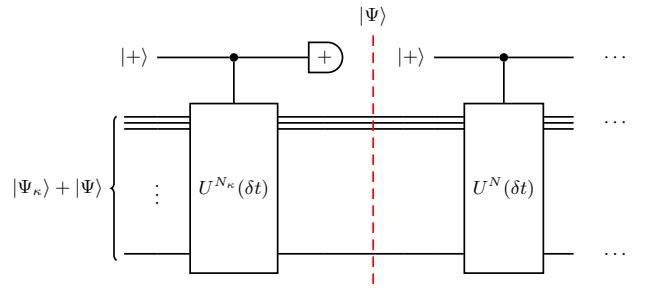


FIG. 4. Circuit for using a single-ancilla phase measurement to selectively remove eigenstate contributions in a superposition. If it is known *a priori* that the phase of an eigenstate $|\Psi_\kappa\rangle$ goes to a minimum at time $N_\kappa\delta t$, executing an ancilla measurement at that time should remove its contribution.

numerically in the following section. Figure 4 illustrates this case.

C. Attenuation and scattering

Here we introduce an analogue of complex absorbing potentials (CAPs) which we have developed. We presently test the technique in the context of ionisation and scattering in Section IV E.

As discussed in Section II A, the use of the k -space mapping imposes a periodic boundary condition in real space. For non-periodic systems, this introduces artificial interactions and state interference through the boundary and one of the requirements on the simulation box is that it is sufficiently large that these have negligible impact on the dynamics. Given that, for fixed spatial resolution, the width can be increased with only a logarithmic cost both in the number of additional qubits and the ‘wall clock’ execution time (see Section V), this does not usually pose a problem. However, there are simulation tasks where simply enlarging the box may not be an ideal solution: for example, a study of a scattering process where the quantum states of scattered particles have contributions that travel at speed towards the edges of the simulation box.

One alternative is to attenuate wavefunction amplitude near the boundary. Any such attenuation would be a non-unitary process involving measurement and implying some probabilistic element; however the cost associated with any need for repetition may be preferable to the increase in the number of qubits required in the former approach. Furthermore, as we demonstrate presently, from the rate of failure of the probabilistic process one can infer scattering probabilities.

The attenuation approach we now describe is an analogue of a well-established technique in classical simulation: complex absorbing potentials (CAPs) were introduced in the context of time-dependent quantum simulations by Kosloff and Kosloff in 1986 [59]. The interaction potential part of the Hamiltonian \hat{H}_{pot} is modified to

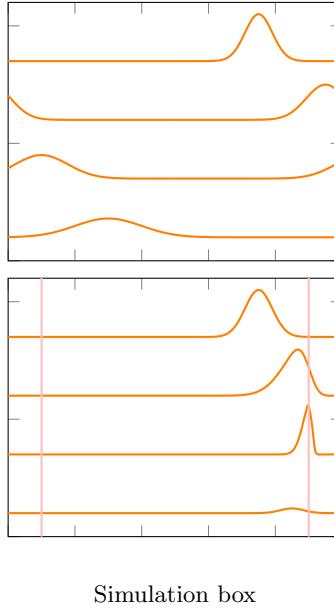


FIG. 5. Above: Dispersion of a Gaussian wavepacket across the periodic boundary, undesirable in scattering simulations. Below: Addition of complex absorbing region attenuates the scattered wavepacket by reducing its norm, cancelling the transmitted image while negating the need for a long simulation box. In the illustrated case, the process is not quite perfect: there is some reflection caused by the attenuation being too severe.

contain both a real and (negative) imaginary part

$$\hat{H}_{\text{int}} = \hat{H}_U + \hat{H}_V - i\hat{V} \quad (36)$$

The time evolution operator is thus

$$\Psi(\mathbf{r}, t) = e^{-i\hat{H}_t} e^{-\hat{V}t} \Psi(\mathbf{r}, 0) \quad (37)$$

The absorbing potential $\hat{V} = V(\mathbf{r})$ should be balanced between sharp, abrupt attenuation which reduces the size of the attenuation region while suppressing transmission, and smooth, gradual attenuation which suppresses reflection. For brevity, we will only assert that it is possible to construct complex absorbing potentials which are completely reflection and transmission free [76], and refer the reader to the rich body of classical quantum dynamics literature on complex potentials [34, 77–81].

Now considering simulation via a quantum computer, the objective is to achieve the following non-unitary transformation in (though not limited to) the real space representation at every split-operator step

$$\sum a_n |n\rangle \Rightarrow \sum a_n e^{-V(x_n)\delta t} |n\rangle \quad (38)$$

followed of course by the necessary renormalization. Here x_n is the location of the peak of the pixel function $P_{nr}^n(x)$ as per Eqn. 14. Of course δt is merely a constant, corresponding to the temporal granularity of the SO simulation. The analogous mapping should occur for y and z ,

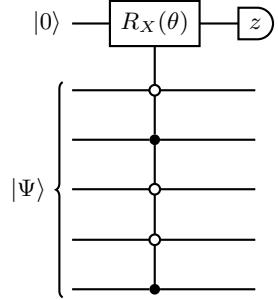


FIG. 6. Using single-ancilla measurement to attenuate the amplitude associated to a single pixel. The filled circles indicate ‘control by 1’ and the open circles ‘control by 0’. The state selected for attenuation is $|01001\rangle$.

and for each particle in the system – ideally this should be done in a fashion that perfectly preserves any exchange symmetry relevant to the system.

To achieve this, we could proceed as follows for each x_n that lies within our attenuating region (i.e. where $V(x_n) \neq 0$). At the end of each spatial part of our SO cycle we prepare an ancilla qubit in state $|0\rangle$ and we perform a multi-controlled rotation $\exp(i\sigma_x\theta)$ on that ancilla. The rotation occurs if and only if the sub-register is exactly in state $|n\rangle$ which corresponds to the pixel function peaked at x_n . We then measure the ancilla. If we obtain outcome $|1\rangle$ we say that the particle in question has been measured to be that point, and we may either halt the simulation or proceed with the remaining particles – both scenarios are examined in the numerical emulations presented later. However, if we obtain outcome $|0\rangle$ then the simulation simply proceeds, except that the amplitude of state $|n\rangle$ has been attenuated from a_n to $a_n \cos(\theta)$ (and the state has been renormalised). By repeating this process for all x_n within the attenuating region, selecting the proper $\theta = \arccos(\exp(-V(x_n)\delta t))$ in each case, we implement Eqn. 38. In parallel the same process can be applied for other sub-registers and other particles.

If the attenuation region is narrow, corresponding to only a few values of x_n , then the above protocol can be relatively efficient. Remarkably frugal constructions for multi-controlled Paulis do exist [82], and of course a controlled Pauli can implement the general θ rotation described above with the use of additional single-qubit rotations. However if the attenuation region is a significant portion of the simulation box then the method would be exponentially inefficient as it requires action for each element of the spatial superposition. Fortunately however, one can simply set a uniform attenuation strength V over a range of x_n values and implement it in one step using an appropriate subset of register qubits in the control process. The simplest example occurs when the width of the attenuating region is $2^{-m}L$ for some integer m (L being the width of the simulation box). In this case, only the most significant m qubits from the sub-register

control the rotation. This is the approach taken in the numerical emulations presented later in this work.

By repeating the entire simulation multiple times and keeping track of the probability that the ancillary qubit yields $|1\rangle$, we are also recording the time-dependent probability that a given particle has ‘escaped’ the simulation box. This allows us to directly measure particle scattering and ionisation probabilities. We presently explore two scenarios: ionisation of a single bound electron by a strong applied electric field, and two-particle scattering.

D. Demands on spatial and temporal resolution

Here we discuss some general considerations regarding the granularity of the simulation’s spatial and temporal resolutions.

The standard SO method described so far involves three critical approximations: (a) a discretized representation of the state (Section II A), (b) propagation of the potential phase assuming the interaction operator \hat{H}_{int} is exactly diagonal (Section II B), and (c) the Trotter error from discretization of $e^{-i\hat{H}_{\text{tot}}\delta t}$ into the two non-commuting $e^{-i\hat{H}_{\text{kin}}\delta t}$ and $e^{-i\hat{H}_{\text{int}}\delta t}$ phases (Section II B). Both (a) and (b) become exact in the limit of infinite spatial resolution, while (c) becomes exact in the limit of infinitesimally small time steps.

It is worth stressing that the first of these, the finite resolution, has consequences for both the wavefunction representation and moreover for the operators forming the SO cycle: the representation of these continuous operators as a discrete matrix is an approximation, and moreover the nature of the off-diagonal terms in the potential part will depend on the resolution. The severity of the impact depends on the nature of the modelled interactions; if we were using quadratic³ particle interactions $V(\mathbf{r}) \propto r^2$, then a modest resolution could ensure that the potential is almost constant over interval between one spatial pixel function and the next, $x_{n+1} - x_n = \delta r$. However the results presented in this paper concern the far more challenging Coulomb potentials which are singular at $\mathbf{r} = \mathbf{0}$. Our data from exploring variations in temporal and spatial resolution are presented in Section IV B.

A relatively modest resolution can still suffice if the wavefunctions involved have near-zero amplitude at the origin; for example in 2D hydrogen, states with quantum number $m \neq 0$ have this property (see Section IV A), and this motivates our use of such states in several of the numerical studies presented in Section IV.

When simulating the natural Coulomb potential, a general simulation protocol must be capable of handling

states that are peaked at the origin. It may not be intuitively clear that such states can be satisfactorily modelled with *any* finite resolution, since increasing the resolution introduces new pixel functions that are closer to the origin and for whom the diagonal approximation is ‘worse’. However, it is well-established in the literature that any reasonable observable property of our system will converge properly as the spatial resolution δr^{-1} is increased (provided that other parameters are suitably updated; notably δt must decrease as explained below). The most direct way to see that this is the case is by considering the unitary nature of the simulation; the SO method, while approximate, is nevertheless strictly unitary being formed of a sequence of unitary steps. As we increase resolution, the pixels that are closest to the origin may indeed be more and more imperfectly updated by our simulation steps because of the approximation in the Coulomb operator, but they constitute an exponentially vanishing component of the entire simulation, both in terms of the Hilbert space and (crucially) the total amplitude associated with them. Thus it ultimately becomes impossible for any observable to efficiently distinguish between a state evolved under the ideal $e^{-i\hat{H}_{\text{tot}}\delta t}$ and the same state evolved under our protocol.

We can therefore be confident that there is *some* value of the resolution δr that will suffice to capture even the most challenging wavefunctions, and moreover adequately approximate the Coulomb operator, addressing the first two sources of error. But this comes with an important caveat that relates to the Trotter error. The problematic cost is not the qubit count but rather the time required for our simulation; as δr decreases we must reduce the time step δt proportionately and thus more SO cycles will be needed to simulate a given period of dynamics. The intuitive reason is clear: the effect of the kinetic operator is to change the shape of the wavepacket in real space, but if δt is too large the change is significant on the scale of multiple pixel widths δr . Amplitude can then ‘magically’ pass through features such as the Coulomb singularity. Instead, δt should be small enough that the kinetic part of the split-operator step only modestly alters the amplitudes associated with each pixel function.

Optimising our approach would involve selecting an appropriate Trotter sequence [8, 65, 66]. In this paper we use only the most simple 1st order sequence where the error $\epsilon = \frac{1}{2}[\hat{H}_{\text{kin}}, \hat{H}_{\text{int}}]\delta t^2$, however any order Trotter sequence will ultimately have an error involving on the same commutator (in nested form), which depends on the second order derivative of the interaction potentials.

In our finite discretization it is therefore the elements very close to the Coulomb singularity that are the most problematic, and the more so as the resolution increases. Thus we should anticipate that the time increment δt must be commensurately decreased and this is what we presently report (albeit the severity of the decrease will vary with the order of the Trotter expansion). Ultimately this cost will not prevent successful simulation of interesting molecules on a quantum computer; as will be dis-

³ As an aside, we remark that the preliminary testing and debugging of our emulation code was performed on ‘easy’ scenarios of this kind.

cussed in Section V, the proper choice of δr^{-1} does not scale with system size but only with the highest nuclear charge. Nevertheless it is an impactful cost.

We conclude this section by noting that various methods for handling the Coulomb interaction in the real-space context have been explored in the classical literature. One approach is to cap or truncate it [83, 84] in some fashion, either by defining it to be constant within some radius or smoothing the singularity. Another is to describe it with a Fourier expansion so that it cannot be more highly curved than the highest Fourier component (which might be chosen to match the highest k planewaves in the state’s representation) [26, 85]. We can also replace the nuclear potential with an effective potential representing both the nucleus and the inner shell electrons [86]. Methods of this kind may ultimately prove to be the optimal approach for quantum computer enabled simulations too, but for the present paper we attempt to deal with the Coulomb potential in its ‘natural’ form.

E. Augmented split-operator

We now describe an alternative means to tackle simulation of states that may be peaked at the Coulomb singularity, instead of a ‘brute force’ increase in spatial and temporal resolution. We call the approach the Augmented split-operator (ASO) as it involves introducing an additional step in the SO cycle. It is compatible with, and complementary to, the choice of optimal Trotter sequence [8, 65, 66] but we will describe it in the context of the simplest 1st order SO cycle. The results of applying the protocol below with an emulated quantum computer are presented in Section IV F.

Following the implementation of the kinetic and interaction parts, and while still in the real-space representation, we apply a *core stabilization* process. This is intended to ‘fix’ the most severe deviations of the applied SO unitary from the ideal unitary $e^{-i\hat{H}_{\text{tot}}\delta t}$. We emphasise that the fix is at the operator level, and is agnostic to the specific state that is being modelled – however the motivation is to make the dynamics of core-peaked states more accurate. We aim to reduce the overall time cost of a simulation with a given accuracy, without significantly increasing the physical qubit count.

We describe the process for a simulation of a single particle and comment on the generalisation to N particle systems further below. The following is performed on a classical computer, as a one-off preparation for our simulation. We make a fixed choice of δt , and we must repeat the analysis for each value of δt we wish to use.

1. We compute the ideal iterate $U_{\text{ideal}} = e^{-i\hat{H}_{\text{tot}}\delta t}$ and the actual iterate defined by Eqn. 22 viz.

$$U_{\text{SO}}(\delta t) = e^{-iD_{\text{int}}\delta t} \left(U_{\text{QFT}}^\dagger e^{-iD_{\text{kin}}\delta t} U_{\text{QFT}} \right).$$

These objects are computed in the full basis of pixel

functions for the particle (and are therefore far from diagonal).

2. We examine U_{repair} which is defined by

$$U_{\text{ideal}} = U_{\text{repair}} U_{\text{SO}}(\delta t)$$

and we will find that U_{repair} is close to the identity except for elements corresponding to pixel functions that are near to the Coulomb singularity. Therefore we select a set of Q such pixels, and derive a small $Q \times Q$ unitary U_{core} that closely matches U_{repair} in that subspace.

3. We define augmentation U_{aug} as the identity operator except for the Q -state subspace, where it corresponds to U_{core} .
4. We find a circuit C_{aug} that can implement U_{aug} to a good approximation. This circuit is not merely a set of phase operations since U_{aug} is not diagonal in the Q -state subspace; C_{aug} will cause a flow between the amplitudes of those states.

This classical analysis is non-trivial but tractable since the unitaries involved are ‘only’ of size $2^{n_r D}$ for a D -dimensional problem where n_r is the number of qubits per sub-register. Moreover there are helpful symmetries in the analysis, and notwithstanding the description in Step 1 we need not compute U_{ideal} and U_{SO} entirely but only in the subspace where we are likely to identify the Q core pixels. With these steps completed, we are in a position to perform simulations with the augmented SO iterate:

$$\begin{aligned} U_{\text{ASO}}(\delta t) &\equiv U_{\text{aug}} U_{\text{SO}}(\delta t) \\ &= U_{\text{aug}} e^{-iD_{\text{int}}\delta t} \left(U_{\text{QFT}}^\dagger e^{-iD_{\text{kin}}\delta t} U_{\text{QFT}} \right). \end{aligned} \quad (39)$$

In the Results section we report on the performance of this method for two cases, 2×2 and 4×4 pixels. As demonstrated by those examples, even with such small corrections the method is quite effective, and the cost of the augmentation can be modest: The circuit C_{aug} is compact and need only target a small subset of qubits, with the others acting as controls.

The generalisation of the ASO method to the P -particle case is straightforward except for a caveat. As discussed in more detail in the sections concerning resource costs and architectures, the main time cost of the SO method lies in the implementation of the $e^{-i\hat{H}_{\text{int}}\delta t}$ part, because of the $P(P - 1)/2$ pairings involved. The ASO method can be efficiently implemented by applying an U_{aug} augmentation after each such pairing, subsequent to implementing the $e^{-iC\delta t/|\mathbf{r}|}$ and while the registers are still representing relative coordinates $x_i - x_j$, $y_i - y_j$ etc. Because U_{aug} is not diagonal, ensuring the proper evolution requires implementing the full orthogonal transformation to relative and central coordinates:

$\{x_i, x_j\} \Rightarrow \{x_i - x_j, x_i + x_j\}$ rather than $\{x_i - x_j, x_i\}$ and similarly for y and z . Fortunately the additional arithmetic step that this implies, is trivial.

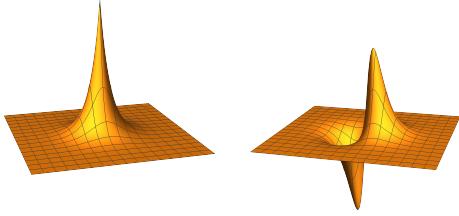


FIG. 7. The ground $\Psi_{0,0}$ state (left) and a first excited $\Psi_{1,1}$ state (right) of two-dimensional hydrogen. Note that the plots here do not reflect the choice of simulation box size and are not to scale.

IV. RESULTS

We now present our numerical results. Data were obtained from exactly-emulated quantum processors, implemented by the open source tools QuEST [87], QuESTlink [88] and pyQuEST [89]. Results are reported in Hartree atomic units, where the reduced Planck constant, electron mass, elementary charge and Bohr radius are treated to be unity $\hbar = m_e = e = a_0 = 1$. Details of hardware employed, as well as significant configuration choices including the alignment between the grid of pixel functions and the nuclear potential, are given in Appendix B.

A. Commonly used Hamiltonian and initial states

We will frequently use the 2D hydrogenic system described by the Hamiltonian

$$\hat{H}_{\text{tot}} = -\frac{\hbar^2}{2m_e} \nabla_{\mathbf{r}} - \frac{e^2}{4\pi\epsilon_0 |\mathbf{R} - \mathbf{r}|}, \quad (40)$$

where we model the atomic nucleus as a clamped classical Coulomb potential. Analytic solutions to Eqn. 40 have been reported by Parfitt *et. al.* [90] and Yang *et. al.* [91]. We use an equation from the former,

$$\Psi_{n,m}(r, \theta) = \sqrt{\frac{q_0^3(n-|m|)!}{\pi(n+|m|)!}} \times (2q_0r)^{|m|} \times e^{-q_0r} \times L_{n-|m|}^{2|m|}(2q_0r) \times e^{im\theta}, \quad (41)$$

where $q_0 = \frac{1}{n+1/2}$ and L are the generalised Laguerre polynomials. The quantum numbers $n = 0, 1, 2, \dots$, and there are $(2n+1)$ values of m . The energy eigenvalues are

$$E_n = -\frac{1}{2(n + \frac{1}{2})^2}. \quad (42)$$

We show two of the eigenstate used in this work in Figure 7.

In a 36 qubit simulation we will use the well-known solution to the 3D hydrogen-like atom

$$\begin{aligned} \Psi_{n,l,m}(r, \theta, \phi) = & \sqrt{\left(\frac{2Z}{n}\right)^3 \frac{(n-l-1)!}{2n(n+l)!}} \times \left(\frac{2Zr}{n}\right)^l \\ & \times e^{-Zr/n} \times L_{n-l-1}^{2l+1}\left(\frac{2Zr}{n}\right) \\ & \times Y_l^m(\theta, \phi), \end{aligned} \quad (43)$$

where $Y_l^m(\theta, \phi)$ is a spherical harmonic and Z is the central nuclear charge.

B. Spatial and Temporal Resolution

We investigated the demands on spatial resolution δr^{-1} and temporal resolution δt^{-1} using emulated quantum computers with various values of n_r , the number of qubits per sub-register. We performed a series of state propagation experiments having initialised to the ground state $\Psi_{0,0}$ of 2D hydrogen, and second series using the excited state $\Psi_{1,1}$. States were propagated for 1.5 atomic time units but at different time step resolutions. We employed the single-ancilla phase estimation procedure documented in Section III A to extract the autocorrelation signal from these simulations.

Figure 8 presents the final absolute values of the autocorrelation in each propagation sequence, specifically their deviation from unity, as fidelity metrics. The figure suggests that when a high spatial resolution is used, very fine time steps are needed to suppress Trotter error and conserve the fidelity (Section III D); indeed, for the largest n_r the fidelity only approaches unity at time steps below $\delta t = 0.0005$ a.u. (totalling 3000 split-operator steps). When we raise the spatial resolution, we must make a commensurate increase in the temporal resolution in order to produce a superior fidelity. The fidelity of the ground $\Psi_{0,0}$ state is much more sensitive to adequate time resolution than the $\Psi_{1,1}$ state; the former can drop to 90% when large time steps are used, whereas that of the latter never deviates more than 0.0005 from unity even at the crudest time resolution of $\delta t = 0.1$ a.u.

Figure 8 correlates the drop in fidelity of propagated eigenstates with corresponding errors in energy predictions from the single-ancilla phase estimation. As anticipated, our results also indicate that the best achievable energy prediction is limited by the spatial resolution δr used to represent the state and the operator. For the ground $\Psi_{0,0}$ state, the error in the most accurate energy prediction (attained with the smallest time step propagation) halves when we increment n_r . At the highest spatial resolution and finest time step investigated, the energy inferred from phase estimation is not within chemical accuracy of the exact energy. We anticipate that an extra 4 qubits per sub-register would be necessary, but we also note that the choice of simulation box size was generous and could have been further reduced without ‘clipping’

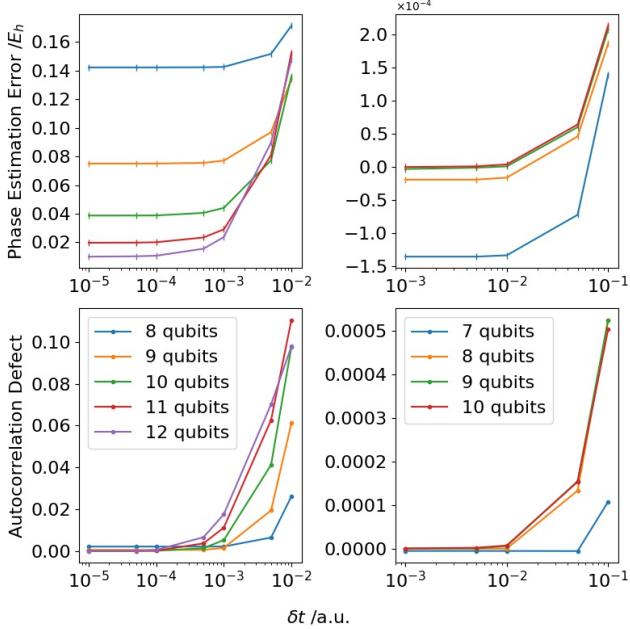


FIG. 8. We initialise the ground state $\Psi_{0,0}$ of 2D hydrogen centred in a simulation box with sides of 10 a.u., such that the origin of the Coulomb singularity lies half way between two central grid points. Each sub-register has a budget of $8 \leq n_r \leq 12$ qubits to store the wavefunction, corresponding to spatial resolutions of $0.039 \geq \delta r \geq 0.002$ a.u.. We also initialise the $\Psi_{1,1}$ excited state in a simulation box with sides of length 40 a.u., with budgets of $7 \leq n_r \leq 10$ qubits per sub-register and corresponding resolutions of $0.313 \geq \delta r \geq 0.039$ a.u.. These two states, represented at different spatial resolutions, are all time propagated using the first-order SO for 1.5 atomic time units. We used time steps between $0.00001 \leq \delta t \leq 0.01$ (150000 to 150 SO steps) for the $\Psi_{0,0}$ state, and between $0.001 \leq \delta t \leq 0.1$ (1500 to 15 SO steps) for the $\Psi_{1,1}$ state. Above: Difference between the energy from phase estimation and the analytic energy. The phase estimation error of the ground state $\Psi_{0,0}$ (left pane), while converging, does not reach chemical accuracy in the set of numerical experiments conducted in this work. Below: Deviation of the fidelity from unity at the end of the propagation.

the wavefunction. Commensurately we would require at least another order of magnitude finer time resolution for convergence.

The smallest time step investigated here, propagation for only 1.5 a.u. (≈ 36.2 attoseconds) already demands 150000 SO cycles and, as discussed in Section III D, the more qubits used the longer each SO cycle will take to execute. While these time requirements for accurate simulation of core-peaked states like $\Psi_{0,0}$ may seem daunting, it is worth reiterating that the approach taken here is not optimised for a compact simulation box size and we employed only the 1st Trotter sequence. This choice was made to set a baseline for comparison between the various techniques we explore, but higher order sequences will almost certainly significantly reduce the required cycle count. Moreover, as we show in Section IV F, it is

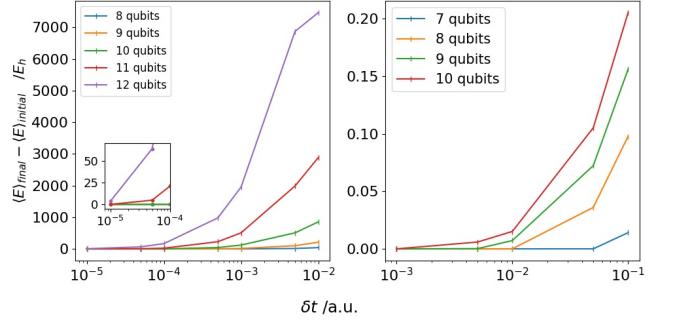


FIG. 9. A ‘bad’ observable as discussed in the main text. Plots show the difference between the final and initial energy expectation value of the ground state (left) and the first excited state (right), propagated at different spatial and time resolutions. The left inset plot zooms in on the energy error at high temporal resolutions for the ground state, highlighting the error (on the order of $1 E_h$) the highest spatial and temporal resolution tested.

possible to augment the SO cycle itself such that we can obtain accurate phase estimation with far fewer qubits and lower time resolution than the brute force method reported here. Leaving aside these remarks, it is also noteworthy that accurate modelling of non core-peaked state $\Psi_{1,1}$ is remarkably tolerant of low resolutions.

Finally, we remark that placement of the nuclear origin between grid points does not significantly affect the dynamics. Refer to Appendix E for details.

C. Cautionary tale: A ‘bad’ energy observable

Here we examine a sampling-based method of estimating the system’s energy which proves to be highly sensitive to inevitable imperfections in the model. Ultimately it will converge to give the correct expected energy once spatial and temporal resolution are sufficiently high. However it is profoundly inaccurate at resolutions where other methods can already provide reliable, well-converged results.

We are referring to the energy expectation as given by Eqn. 29, viz. $\langle E \rangle = \langle \hat{H}_{\text{kin}} \rangle + \langle \hat{H}_{\text{pot}} \rangle$, and supposing that this would be obtained from our quantum computer as follows: Generate the desired state at time t , measure in the k -space representation, and repeat this many times in order to estimate the first term. Apply the same process but measuring in the real-space representation to estimate the second term. This method is inefficient in terms of the sampling cost, and is therefore already unattractive compared to phase estimation, but more problematically it is very sensitive to the Trotter error. As shown in Fig. 9 the error between the initial and final expected energy grows exponentially with the size of the elementary time step δt : for the ground state $\Psi_{0,0}$, the energy difference was more than $7000 E_h$ accumulated over less than 40 attoseconds of simulated

time in the worst offending case. In the $\Psi_{1,1}$ simulation, the non-conservation of expected energy is also apparent when the temporal resolution does not match the spatial resolution, but is more contained relative to the ground state (in the worst case, it goes up to $0.20 E_h$). It is evident that the core-peaked nature of the $\Psi_{0,0}$ state is significant in the issue.

It is the kinetic energy term $\langle \hat{H}_{\text{kin}} \rangle$ that exhibits this diverging behaviour. The explanation is as follows: Imbalance between the extreme potential and extreme kinetic energy near the Coulomb origin, inevitable in our discrete grid representation, can allow a small amount of the amplitude to diffuse towards high frequency states in the plane wave representation. The extent of this diffusion may be small relative to the initial state so that the fidelity of the state remains high (see Figure 8) and thus phase estimation methods can perform well. However, simply sampling $|k\rangle$ and squaring it to estimate $\langle \hat{H}_{\text{kin}} \rangle$ gives direct weight to this error that actually worsens as we improve the fineness of our resolution. Reducing δr , the separation of our spatial pixels, can reduce the leakage of amplitude (increasing the state's fidelity) but the maximum kinetic energy that the model can represent goes as $1/(\delta r)^2$. Amplitude leakage declines less rapidly than the rate at which the energy of the leaked-states increases: thus the problem worsens. One must use extremely high time resolution to control the effect.

Moving to a higher order Trotter process would presumably ameliorate the effect, in the sense that a more modest time resolution could control the leakage. Nevertheless we anticipate that this will always be an inferior approach to estimating energy compared to phase estimation.

D. Phase estimation for state preparation

In Section III A we described the single-qubit ancilla method that can suffice to learn an eigenstate's energy or to prepare an eigenstate from an initial superposition. Earlier plots (Figure 8) have presented the results of this method of energy estimation for a variety of scenarios. Here we demonstrate the state preparation aspect; it is probabilistic (similar to e.g. [24]), but proper choice of parameters will maximise success rates.

Figure 10 shows the results of a simulation where the initial state (shown lower left and inset (i)) is a superposition of two eigenstates of 2D hydrogen:

$$\frac{1}{\sqrt{2}} (\Psi_{1,1} + \Psi_{2,2}).$$

The state's applitude is not symmetric about the nucleus, and when we apply our SO cycles we observe a rotation of the state due to the different rates at which the two superposed eigenstates acquire phase. At the time marked (ii), the total phase acquired by more tightly bound state $\Psi_{1,1}$ (having a more negative energy) has reached π and

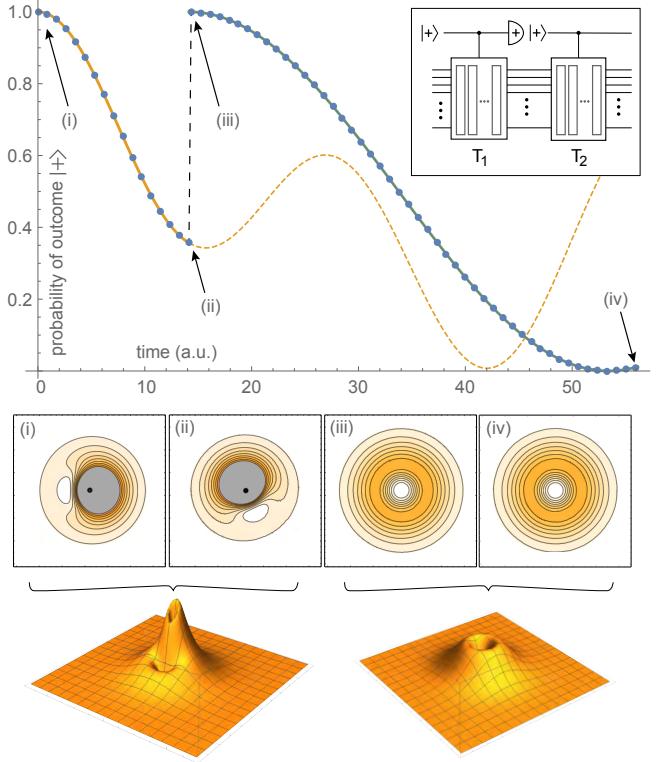


FIG. 10. Demonstrating preparation by phase measurement using $1+2 \times 8$ qubits in an emulated quantum computer. The initial state is a superposition of non-degenerate eigenstates of 2D hydrogen, as described in the main text. The representation uses 8 qubits for each of the two dimensions, with a total simulation box width of $L = 56$ a.u. The state evolves, conditional on a controlling ancilla, for time T_1 chosen such that $T_1 E_1 = \pi$; in this period the conditional evolution is a rotation of the state due to the accumulated phase difference between states (lower left plots). At T_1 the controlling ancilla is measured in the $|+\rangle$ as shown in the upper right inset. This projects the state into the $\Psi_{2,2}$ state.

thus if this state were the sole one present, a control ancilla initially in state $|+\rangle$ would now certainly be in state $|-\rangle$. We indeed measure the ancilla at this point, but post-select on seeing the outcome $|+\rangle$. The probability of the desired outcome depends on the probability associated with the target $\Psi_{2,2}$ within the initial superposition (which was 0.5) and the probability that this state, had it been prepared alone, would yield a $|+\rangle$ outcome at this point. The latter is 0.713 in the present case. In a scenario where the states to be distinguished are closer in energy, it may be optimal to simulate for several complete cycles of the undesired state before measuring.

In our numerical emulation, we indeed assume that the desired $|+\rangle$ state is obtained and we continue our ancilla-conditional SO cycles: The new evolution of the ancilla state (green curve in the figure) is exactly that of the pure $\Psi_{2,2}$ state. The contour plots of the simulated state (insets (iii) and (iv)) confirm that we have prepared that pure state. Fidelity with respect to $\Psi_{2,2}$ was essentially

identical to an initial state prepared directly in that state.

This is a demonstration of the practically-useful capability to take an initial state that is not fully understood, and remove from it the components corresponding to states with energies that *are* understood. More generally, one could use Fourier analysis (see Appendix C) to identify the components in the plot of $\text{Prob}(|+\rangle)$ for the full state, and then use the post-selection method to (probabilistically) isolate given components.

E. Electric Field Ionization and Two Particle Scattering

As introduced in Section III C, it is possible to monitor the escape of a particle from the simulation box as a result of, for example, a scattering or ionisation event. This can be achieved by repeated weak measurement of the spatial pixels, which closely approximates the ‘complex potential’ technique used in classical computing models as explained in that earlier section. Here we describe the results of two numerical simulations of the approach: the first concerns a single particle in a strong electric field, and the second is a two-particle scattering simulation. The corresponding data are shown in Figs. 11 and 12.

In Figure 11 we report the performance of a 19 qubit emulated quantum computer, modelling a single 2D particle (9 + 9 qubits represent the state, one qubit is used for the weak measurements). The modelled system at $t = 0$ is a state within the first excited manifold of 2D hydrogen, specifically

$$\frac{1}{\sqrt{2}}(\Psi_{1,1} + \Psi_{1,-1}).$$

A term E_x in the Hamiltonian corresponds to a strong, static electric field is applied in the horizontal direction; the total potential of the Coulomb field and the electric field is shown in an inset in the top right panel. Because of the electric component, the initial state is no longer an eigenstate and the simulation can determine whether the electron will indeed ionize away from the nucleus.

The initial state occupies only a small central region in the simulation box, as illustrated in the top left the figure. The contour plots in the main part of the figure show the evolution, both in the centre part of the simulation box (the box bordered in green) and a larger region encompassing the centre and the region to the right (the box bordered in blue). The pink region, constituting the outermost 50% of the simulation box in both the x and y directions, is the region that is monitored by weak measurements.

The plot in the upper right of Fig. 11 shows the cumulative probability that the particle will indeed have ‘escaped’ i.e. that it will have been measured to be in the pink region. The curve ultimately approaches unity, indicating that the particle will eventually ionize with certainty. Interestingly, we observe oscillations in this

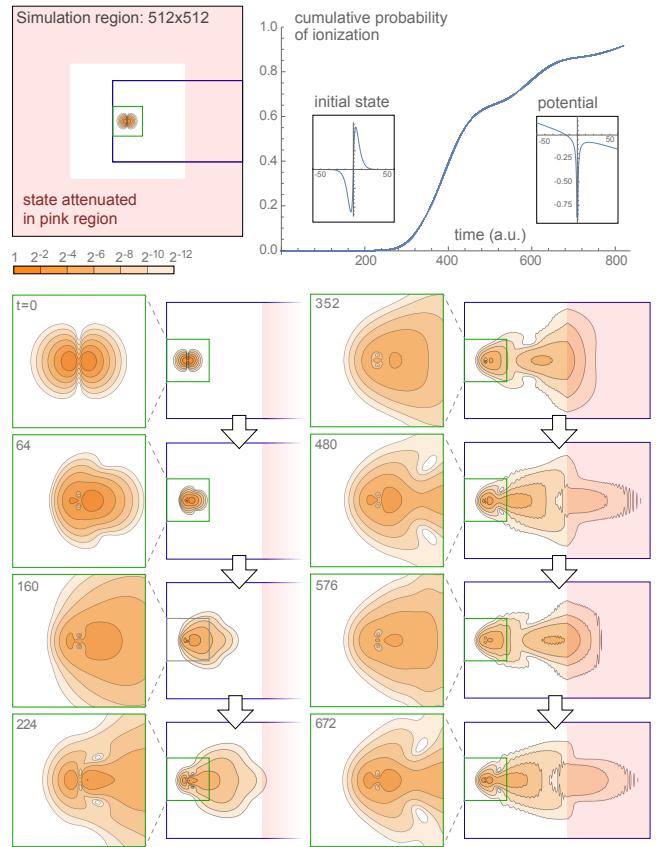


FIG. 11. Figures showing a 2D simulation of a single particle ionized by a strong electric field, executed on a $1+2 \times 9$ -qubit quantum computer. The initial state is a low-lying bound state of 2D Hydrogen ($\Psi_{1,1} + \Psi_{1,-1}$). The panel in the upper left shows the initial state as a contour plot; it occupies a small region in the centre of the entire simulation box (black boundary). The regions marked with green and blue borders are the zones into which we zoom in the lower panels, which are a series of contour plots with the time index shown in the corner of each plot. The pink region corresponds to attenuation (a complex potential) as explained in the main text. The total probability of the particle having been found in the outer, pink region is shown by the upper right plot; it approaches unity indicating that the particle will certainly ionize. The plot exhibits undulations that are correlated to dipolar oscillations, as discussed in the main text.

curve, which we can account for by examining the contour plots of particle density shown below.

Note that the contour plots show the particle’s probability density *post-selected* on it not having yet escaped; therefore the normalisation is unity in each case. Focusing on the green boxes, those zoomed in close to the nucleus, we observe that the part of the state that remains close to the nuclear core actually oscillates in a dipole-like fashion. Note that the sequence of green panels on the right, labelled 352 to 672, exhibit this: panel 352 is similar to 576 while 480 is similar to 672. Moreover, examining the corresponding blue regions we observe waves

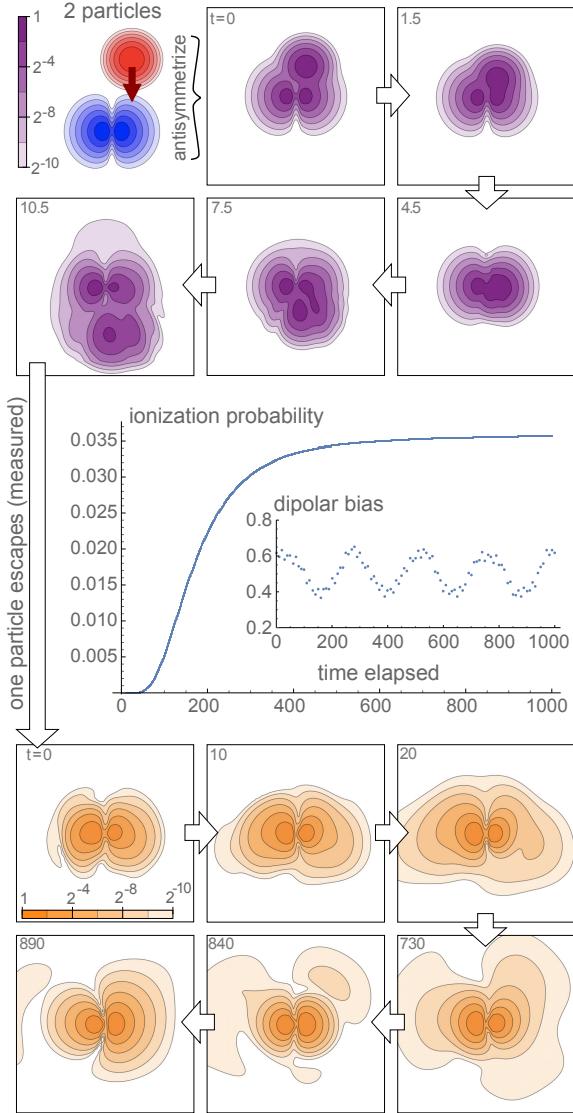


FIG. 12. Figures showing a 2D simulation of two particle scattering using a 25 qubit emulated quantum computer. The initial state is the antisymmetrized version of the following: an electron (blue) in a low-lying bound state of 2D Hydrogen while a second electron (red) is in a Gaussian state and moving in the negative y -direction (downward in the plots). The simulation reveals the scattering event (purple plots) until we deem one particle to have been measured. A second phase of simulation shows the evolution of the state of the remaining electron (orange) which has a probability of about 3.5% of ultimately ionizing (escaping the nuclear potential). Interestingly, in the event that the remaining electron does not ionise it nevertheless remains in a state where it oscillates back and forth; the inset plot shows the quantity $\langle x_L \rangle$ as defined in the text.

of density propagating away from the nucleus, synchronised with the dipolar oscillation – in essence, whenever the oscillation favours the ‘down stream’ right side, there is an enhanced probability that the particle will escape; in due course this leads to a fluctuation in the probability

of observing the particle in the pink, attenuating region. It is interesting to note that this fluctuation probability is reminiscent of the bond-breaking of sodium iodide observed with femto-second pulsed lasers [92], an experiment recognised by the Nobel Prize in Chemistry in 1999.

If we were working with a real quantum processor rather than an emulator, then the plot in the upper right of Fig. 11 could be obtained by repeatedly executing the simulation. The contour plots below could also be obtained by repeated sampling, simply by measuring the state of the particle’s register at a given time t . While obtaining these outputs through ‘brute force’ sampling would obviously represent a multiplicative cost depending on the accuracy with which we require the plots, it would not be a function of the complexity of the system (dimensionality, particle count, etc).

Figure 12 shows the results of a two-particle scattering simulation. There are two simulation phases. In the initial phase, we have the both interacting particles present: an electron initially in a bound state of 2D hydrogen corresponding to $(\Psi_{1,1} + \Psi_{1,-1})/\sqrt{2}$, and an incident electron in a Gaussian state (but with the total state properly antisymmetrized). This simulation runs until one of the particles is measured to have ‘escaped’ our simulation box. We then proceed to a second phase of simulation where we study the dynamics of the surviving particle. We find that it has a small ($\sim 3.5\%$) probability of ionizing due to the perturbation of the prior ‘impact’ with the incident particle.

In the first phase we use a $25 = 1 + 4 \times 6$ qubit emulated quantum processor, where the x and y coordinates of each particle are represented with $2^6 = 64$ states. As in the case of the electric field ionization described earlier, we monitor with weak measurement a set of spatial pixels near the boundary of the simulation box. In the initial phase of this two-particle simulation the width of that region is 25% rather than the 50% width used in the electric field case. The contour plot panels in the upper part of Fig. 12 show the central, non-attenuated region. The state is antisymmetrized so that the probability density plots do not distinguish one particle from the other, but informally we can say that the incident particle interacts with the bound particle before passing on, away from the nucleus. In the case shown in the figure, we deem that a particle has been detected in the attenuation region exactly at the point when the cumulative probability of detection reaches 50% (this is an arbitrary choice; in with a real quantum processor the operator would of course be unable to specify this). This event occurs shortly after the last of the panels in the upper part of the figure.

The simulation to that point would not teach us very much about the nature of the scattering event. We could in principle measure, e.g., any deflection in the trajectory of the incident particle, and we can confirm (from longer simulations) that there is near-100% probability of a particle exiting the simulation region: the incident particle does not become bound. However, it is more interesting

to study the subsequent behaviour of the remaining particle. Because this particle is represented by a register in the quantum processor that has not ‘collapsed’, we can simply continue to simulate its evolution; the component U_V in our split-operator cycle, corresponding to particle-particle interaction, will no longer be applied. We can now choose to vary other parameters such as δt , anticipating that further dynamics are on a slower timescale since the high energy particle has exited the simulation. Moreover, we can reallocate some of the qubits that were previously used to model the now-exited particle, repurposing them to model a larger simulation box. Given that the outer regions of such a box have zero amplitude associated with them at the moment the first particle exits, there is no difficulty in simply introducing those qubits. (In the case that we employ two’s complement, so that the coordinate $(0, 0)$ is in the centre of the simulation box, we simply append the new qubits to the high-order end of each sub-register and perform a CNOT gate on each new qubit controlled by the prior highest-order qubit.) In the simulation shown in Fig. 12 we do indeed do this, and the second phase of the simulation uses $1 + 2 \times 8$ qubits to provide a much larger attenuation region.

In this second phase we observe that the remaining particle has been significantly perturbed by the passage of the incident particle: its distribution at $t = 0$ (now measuring time relative to the exit of the other particle) is noticeably more irregular than the simple symmetric initial form. The probability distribution is lopsided, favouring the left side. Over the remaining period of simulation, the particle exhibits mild dipolar oscillation between a left-favoured and a right-favoured distribution. Defining $P_{\text{left}}(t)$ as the probability that the particle would be found left-of-centre, we observe the oscillation shown in the inset to the plot in Fig. 12. Moreover, as the particle oscillates it sheds probability – i.e. there is a finite probability that the particle will escape the simulation region. In contrast to the electric field simulation described earlier, this probability is shed symmetrically (both left and right) and moreover it asymptotes to a small cumulative probability of about 3.5% (see main plot), whereas the electric field scenario was certain to cause ionization.

F. Augmented Split-Operator

The concept of the augmented split-operator was described in Section III E. As explained there, we introduce additional elements to the basic SO cycle with the purpose of stabilizing the behaviour of states that have significant amplitude at a Coulomb singularity. The intent is to increase the fidelity simulation without resort to very high spatial or temporal resolutions; the costs of the augmentation should therefore be lower than the cost of such ‘brute force’ resolution increase. As explained in Section III E the method acts at the operator level and is independent of the state being simulated, however the measure of its success is established by examining the

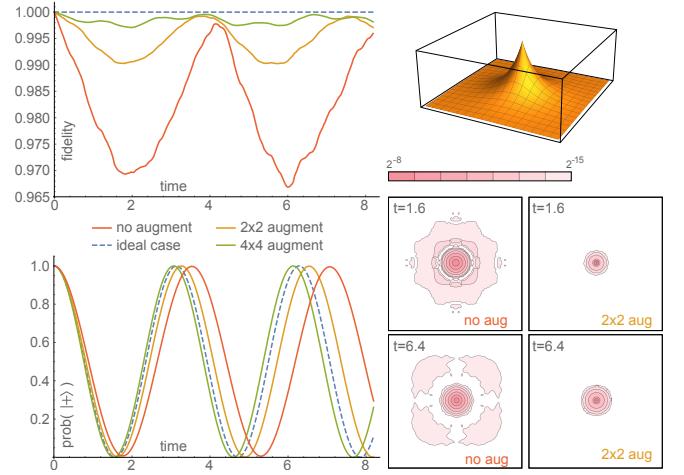


FIG. 13. The performance of the augmented split-operator (ASO) technique on a $13 = 1 + 6 \times 6$ qubit emulated quantum computer. Upper right: The eigenstate $\Psi_{0,0}$ of 2D hydrogen that is used in this experiment, within its simulation box. Upper left: The autocorrelation of the state at time t with respect to the initial state, which ideally remains at unity (blue dashed line). The red, orange and green lines are respectively the cases of no augmentation, a 2×2 augmentation and a 4×4 augmentation. The lower left plot shows the result of phase estimation for the same three cases, with the ideal again shown with a blue dashed line. The contour plots on the lower right show the absolute difference in probability density, with respect to the initial state, for the case of no augmentation (left) and the 2×2 augmentation (right).

most challenging states: Those that are peaked at the singularity.

In Figure 13 we present numerical results from our study of the ASO method. We consider the ground state of 2D hydrogen $\Psi_{0,0}$ which is peaked (with discontinuous gradient) at the origin of the classical Coulomb potential. We use a relatively modest resolution corresponding to $n_r = 6$ qubits per register (i.e. a 64×64 grid of spatial pixels) to represent the state, and set the simulation box to be optimal (see upper left graphic in the Figure).

Step 2 in Section III E states that, having decided that our core patch will involve a subspace of Q pixels, we ‘derive a small $Q \times Q$ unitary U_{core} that closely matches U_{repair} in that subspace’. We recall that the operator U_{repair} was simply the matrix that maps from the SO cycle as actually applied, to the ideal time increment operator. We therefore begin by calculating these matrices, each an $2^{2n_r} \times 2^{2n_r}$ object, explicitly using Mathematica. Having thus obtained U_{repair} we can proceed.

In simulations reported in Fig. 13 we consider two cases: $Q = 2 \times 2$ and $Q = 4 \times 4$ pixels. For each case, we write down the $Q \times Q$ matrix M_{core} which is composed of the elements of U_{repair} that lie in the Q -pixel subspace. M_{core} will not be unitary and therefore cannot be implemented deterministically on our emulated quantum computer; we need a unitary matrix U_{core} that is close to M_{core} . This was obtained by performing a stan-

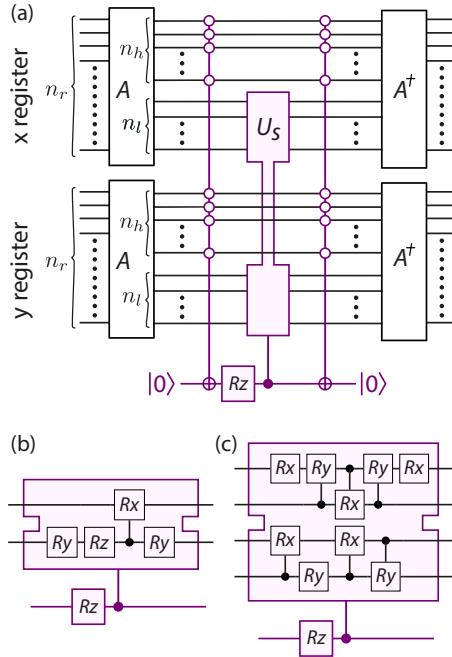


FIG. 14. Figures showing the circuits involved in core stabilisation. Part (a) shows the generic circuit, while (b) and (c) specify the particular circuits used for our 2×2 and 4×4 augmentations, respectively. The A operators simply increments the indexing of the spatial ‘pixels’ so that the lower right pixel is $(0,0)$. In the case of the 2×2 augmentation, the increment used is $G = 1$ so that A maps the indices of the pixels of interest, i.e. $(-1, -1), (-1, 0), (0, -1), (0, 0)$, to $(0, 0), (0, 1), (1, 0), (1, 1)$ respectively. These indices are now exactly those states for which the most-significant $n_r - 1$ qubits are zero, facilitating the application of the small augmentation circuit to *only* the four target states.

dard singular value decomposition of M_{core} and using the components to construct U_{core} :

$$M_{\text{core}} = u \Sigma v^\dagger \quad \text{then} \quad U_{\text{core}} \equiv u v^\dagger.$$

Note that Σ matrix would be the identity if M_{core} were unitary; it is not, but by omitting Σ it we generate our unitary approximation. The final step 4 of finding a circuit to implement the stabilization, was performed using a circuit synthesis tool (for the sizes used here, it is a simple task).

We perform a series of standard and augmented SO simulations, in all cases fixing the time resolution at $\delta t = 0.004$. We monitor the absolute value of the autocorrelation as per Eqn. 31, which should be unity in the case of an ideal simulation (since the initial state is an eigenstate). The graph in the upper left of Fig. 13 shows the result for three cases: The simple SO protocol (red), and protocols with a small (orange) and a medium (green) scale core stabilization augmentation. The small augmentation involves a circuit that modifies only the amplitudes associated with the 2×2 spatial pixels that are closest to the singularity (note that, as detailed in

Appendix B, we align the spatial pixel lattice such that the singularity is mid-way between the four central pixels). The medium augmentation involves a larger set of 4×4 spatial pixels. We observe that there is a dramatic improvement in the autocorrelation, by an order of magnitude between the red and green plots.

The plots on the lower left of Fig. 13 show the result of the single-qubit phase-estimation method described in Section III A. Ideally, the plot would match the dashed blue curve, which corresponds to phase acquisition exactly according to the analytically derived energy. We observe that the red, orange and green lines are again progressively closer to the ideal. We should emphasise that the ASO method is agnostic to the state simulated; therefore, while it would be trivial to ‘cheat’ and apply an exactly compensating phase to obtain the blue dashed curve, in fact the circuits we have employed are derived without foreknowledge of the specific simulation task (i.e. $\Psi_{0,0}$ in no way enters the derivation of U_{core}).

Finally, a third lens on the simulation fidelity is provided by evaluating the difference between the $t = 0$ probability density (over the grid of 64^2 spatial pixels) and the density at a later time. Ideally, of course, this difference would be zero. In the contour plots on the lower right of Fig. 13, we contrast the case with no augmentation with the small augmentation scenario. While there is still a discrepancy, it is far more localised and stable (the un-augmented simulation involves wider, more dramatic fluctuations in the probability distribution).

The circuits that implement both the small and the medium augmentations are shown in Figure 14. They were derived through the process explained in Section III E. Note the preparatory step of applying a unitary A that simply adds an integer to all states in the spatial representation, i.e.

$$A |n\rangle = |n + G\rangle$$

where the addition is understood to be modulo 2^{n_r} . The main figure shows the implementation of this for a general ‘size’ of the augmentation patch n_l , where in our numerical studies $n_l = 1$ for the small case and $n_l = 2$ for the medium case. The figure caption defines the shift for the case of the 2×2 augmentation, where we will use $G = 1$. The A operators can be avoided entirely if we simply define the origin of our spatial coordinates to a bounding corner of our core stabilisation region. The implementation of the spatial part of the SO would need to be corrected for such shift, but that may prove to have lower total cost. Regardless, we include the A operators in the figure in order to make the it directly consistent with the other circuits and expressions in the present paper.

Finally we note that the multi-controlled NOT operation appearing in the circuits of Fig. 14 can be compactly realised by recently discovered circuits [82] that involve $4n - 6$ T-gates (single-qubit phase gates) and a comparable number of control-NOTs, together with an ancilla that is measured during the process.

We anticipate that the time cost of moving from the simple SO approach to the ASO method should be far less than the cost of the ‘brute force’ increase to the temporal resolution needed to assure proper behaviour of core-peaked states (Section III D). For the present demonstration, the performance of the $n_r = 6$ qubit registers was able to approximately match that of the $n_r = 11$ qubit registers under a ‘brute force’ approach (Fig. 8, upper left panel) and moreover this was achieved with a time step δt an order of magnitude greater, so facilitating more rapid runtimes. The ASO method is therefore highly relevant when one operates with the strict Coulomb interaction, as in all the numerical studies in the present paper. It remains to be seen if it would also be useful in scenarios where the Coulomb singularity is approximated by some of the other means listed in Section III D. The ASO method is distinct from, and compatible with, the use of higher order Trotter sequences. Here as throughout the paper we restrict to the lowest order Trotter pattern, but extending this is an interesting direction for further work.

G. 3D Helium simulation

Finally, we extend the low-dimensional demonstrations to simulating the dynamics of a helium atom: two electrons interacting via a repulsive Coulomb interaction, both bound by a central attractive Coulomb potential, in three spatial dimensions. As the true electron eigenstates of the helium atom cannot be solved exactly, we approximate the 2-electron initial state by combining two single-electron solutions of the 3D hydrogen-like Schrödinger equation (Eqn. 43), with a central nuclear charge of $Z = 2$. Note that this would be an exact eigenstate if there were no electron-electron interaction. The two sets of quantum numbers (n, l, m) we used were $(2, 1, 0)$ and $(2, 1, -1)$, of which the former is the $2p_z$ -orbital and the latter is the atomic orbital $2p_{-1} = (2p_x - i2p_y)/\sqrt{2}$. The complete initial (triplet) state is then the antisymmetric wave function

$$\Psi_{\text{init}}(\vec{r}_1, \vec{r}_2) = \Psi_{2,1,0}(\vec{r}_1) \Psi_{2,1,-1}(\vec{r}_2) - \Psi_{2,1,-1}(\vec{r}_1) \Psi_{2,1,0}(\vec{r}_2). \quad (44)$$

As our simulation box we use a cube with sides lengths of 25 a.u. and discretise the initial function Ψ_{init} as per Eqn 19 using $n_r = 6$ qubits per register providing 64 divisions per axis and per particle. We then propagate the full 36 qubit state forwards in time for 500 SO steps, where each step is of length 0.05 a.u., thus a total evolution of 25 a.u. It is worth noting that this relatively simple initial state has symmetries that could be exploited for a more compact representation; however we wished to test the full 3D, two-particle grid representation and therefore we did not exploit any such properties.

The single-ancilla phase estimation method was used in previous sections (see e.g. Fig. 10) to track the evolution of the simulated system’s state. However this requires our emulator to use double the memory that would be needed simply to represent and propagate the state. Since the resource costs for the present emulation are already considerable (see Appendix B 1), we opted instead to compute and record, at every SO time step, the probability density of one of the elec-

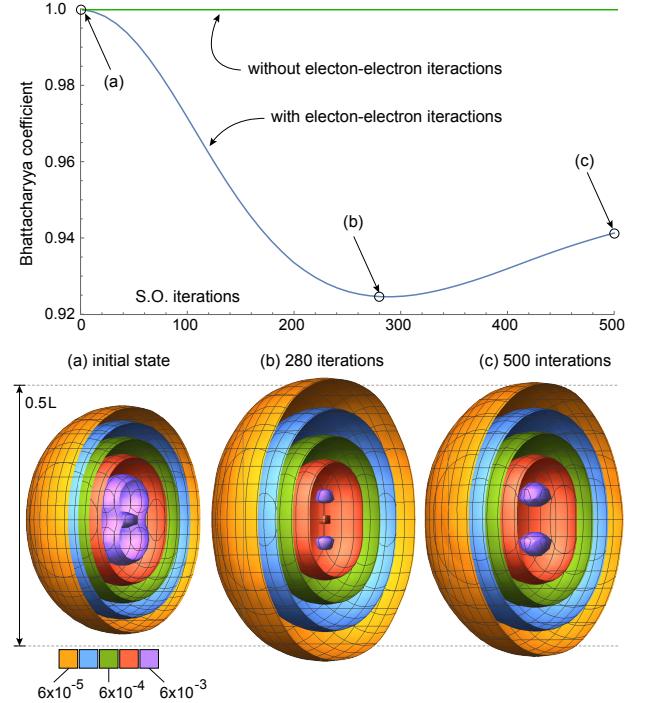


FIG. 15. Bhattacharyya coefficients (with and without enabled $e\text{-}e$ -interactions) and real-space electron density distributions during the time evolution of the He simulation. Coloured shells are electron probability density isosurfaces, within a simulation box with $L = 25$ a.u. Distributions are shown for the initial state, at the time where it is maximally spread out, and at the end of the simulation. The 500 SO cycles correspond to propagation of 25 a.u. (≈ 0.6 femtoseconds).

trons⁴. Specifically, we record the probability associated with each computational basis state of the three registers corresponding to one of the particles. Using these probability distributions we can compute the Bhattacharyya coefficient [93] of the distribution at time t with respect to the distribution of the initial state Ψ_{init} . This quantity is

$$\sum_i \sqrt{p_i} \sqrt{q_i}$$

for our two discrete probability distributions p and q . It can be thought of as a classical analogue of the usual inner product fidelity. It is plotted in Fig. 15.

As the initial state is not an eigenstate, we expect the distribution to vary over time. As shown in Figure 15, the electron density is initially distributed with rotational symmetry around the vertical z -axis, with charge accumulations in the positive and negative z -directions. Because the electrons partly shield each other from the core, the chosen central charge of the initial state $Z = 2$ is too large, and thus the electron orbitals are too close to the core. The time evolution shows that the charge initially spreads out away from the core

⁴ Because they are indistinguishable, the probability density is equal for both electrons.

in every direction, then returns slightly, but not to its original distribution. To confirm that the interaction between the electrons is the cause for this behaviour, we also performed an identical calculation, but with the $e\text{-}e$ -interaction disabled. Figure 15 shows that in this case the probability distribution simply stays constant. We thus confirm that our simulation directly shows the effect of electron shielding in this hypothetical electron configuration.

One could repeat the experiment to find the value for the effective nuclear charge that allows our analytic initial state to most closely approximate a true helium eigenstate; moreover, one could use the method of Section IV D or that proposed in Ref. [24] to actually prepare the eigenstate from such an initial approximation. These are interesting tasks for further study.

V. QUANTUM COMPUTER RESOURCES AND ARCHITECTURES

In the following two subsections we assess on the resource requirements for undertaking modelling beyond the reach of classical algorithms, and the related question of suitable quantum architectures.

A. Resources for post-classical modelling

In this section we reflect upon the implications of our numerical results for the resource demands of post-classical chemistry modelling. We will not undertake a formal resource scaling analysis, noting instead that analyses have been made in the literature since 2012 [25] and there is a highly relevant paper within the last year [22]. There the authors present an extensive and purely analytic study of optimal approaches for mid-to-large scale 1st quantized quantum simulation. Complementing those analyses, our present work implemented grid-based simulations using emulated quantum computers that have proven large enough to elucidate practical issues, such as the number of basis functions required for given levels of accuracy in specific observables. Data of this kind will be of use in estimating the constants that must appear in any resource scaling analysis.

The lowest possible qubit count (for a given simulation accuracy) results from selecting the smallest adequate ‘simulation box’ length L , and setting the spatial resolution δr (Eqn. 16) to be just sufficient to capture the most curved elements of the wavefunction (discussed in Section III D). The number of qubits per particle, per spatial dimension, is then $n_r = \log(L/\delta r)$. The quantity L simply specifies the region of space within our multi-particle wavefunction can be considered well-confined, in the sense that only negligible amplitude is ‘clipped’ by imposing the boundary. Moreover as we explain in the attenuation Section III C we can study processes that would, over the simulation time, go beyond the simulation box due to scattering or ionization. It is reasonable to assume, as in [24] for example, that the L^3 goes linearly with the number of particles P ; a molecule with twice the number of particles requires (of order) twice the volume. Meanwhile, the severity of the wavefunction’s curvature should scale directly with Z_{\max} , the maximum nuclear charge of any of the nuclei in the system (Section III D). Increasing the molecule’s size without increasing Z_{\max} should not require any adjustment in δr .

Note that as we increase the size of the simulation box to accommodate more particles (a more complex molecule, say) then for a given tightly bound state, the simulated box will now be far larger than needed. Importantly, the accuracy with which we model it is essentially unaffected by this change. We have confirmed this numerically for an eigenstate of 2D hydrogen (Figure 8), by scaling the linear size of the simulation box by a factor of 64: we moved from a 13 qubit ($1+6\times 2$) scenario to a more demanding 25 qubit ($1+12\times 2$) scenario while holding constant other parameters, notably the time increment δt . The observed phase evolution and the autocorrelation were practically unchanged. We emphasise that in this process of scaling via additional qubits, the total number of SO steps required to evolve the dynamics a given total time T did not alter (albeit, the circuits required to apply each SO cycle would deepen with the qubit count).

In view of the above observations we can expect that, for suitable constants C_i ,

$$\begin{aligned} n_r &= \log_2 \left(\frac{L}{\delta r} \right) \approx \log_2 \left(\frac{C_1 P^{\frac{1}{3}}}{C_2 Z_{\max}^{-1}} \right) \\ &= C_3 + \log_2(Z_{\max}) + \frac{1}{3} \log_2(P). \end{aligned} \quad (45)$$

The total qubit count will be $3Pn_r$ for P particles in 3D. Note that this simple expression for n_r does not account for the potentially helpful fact that atomic radii have a highly non-linear (and sub-linear) dependence on the number of electrons [94]. In Appendix F we make an estimate of the root term C_3 from our numerics; we argue that $C_3 = 10$ is optimistic but not unreasonable. Using some further assumptions we make a rough estimate of the number of qubits necessary to model two interesting molecules. A complex molecule of great interest is adenosine triphosphate (ATP), which is core to energy cycles in all known forms of life. In the Appendix we suggest that grid-based modelling of ATP may require of order 16,000 computational qubits including a frugal use of ancillas. As a contrast, we consider H₂O as one of the smallest interesting molecules and we make relatively optimistic assumptions to reach an estimate of 300 qubits.

It is difficult to provide any meaningful estimate of total ‘wall clock’ time for a significant simulation, even to within an order of magnitude. For any but the smallest molecules, quantum computation will almost certainly require a fault tolerant machine. For such machines the most relevant metric may be the T-gate count. This is a measure of how many steps in the algorithm correspond to the costly non-Clifford operations that cannot be directly performed in stabilizer codes, and the effort to minimise the count for standard subroutines is ongoing [95]. For example a recent note has reduced the number of such gates need for multi-controlled rotations to a remarkably frugal level [82]; such rotations are key in both our attenuation and augmented SO techniques. More generally, the trade-off between time and qubit count is a research topic in its own right and recent papers have shown how dramatically this balance can be adjusted [96, 97].

Notwithstanding the above, in Appendix F we use various observations and assumptions, and focus on the QFT as indicative of the more costly elements of the algorithm, to speculate that the gate depth for modelling ATP may be a few billion (and this is accounting for parallelism; hundreds of billions of gates in total). Estimates of logical gate execution times vary dramatically depending on the native error rate (typically assumed to be 10^{-3} or 10^{-4}), the speed of a

stabilizer cycle (often estimated at $1\mu\text{s}$) and because one can trade computation times for higher qubit overheads [97]. For surface code based implementations a credible range [98] is $0.1 - 10\mu\text{s}$. This gives a clock time ranging from a hour to days for a meaningful ATP simulation run. However many of the plots in the present paper would require repeated execution, so that obtaining outputs of this kind could still require months with a single processor. Fortunately such repetitions are perfectly parallelisable over independent quantum processors, which need not have quantum interlinks or even be co-located.

B. Quantum computer architectures

The preceding section obtained back-of-the-envelope estimates of qubit counts ranging from several hundred to several thousand. While this sounds encouraging, we noted the very likely need for fault tolerance and the resulting multiplicative increase in physical qubit count. Presently the most well-understood codes can require many hundreds of physical qubits per logical qubit, assuming relatively deep algorithms and physical error rates comparable to today's best QC prototypes. Even if one makes the very optimistic assumption that some form of error mitigation can suffice in place of full fault tolerance, at least for small molecular simulations, the more powerful forms of mitigation can require a multiplicative increase in the number of physical qubits [99, 100]. Thus the number of physical qubits required for the modelling considered in the preceding section could easily reach the high thousands or millions. This raises the question of what kind of architecture would be needed.

In particular, we are motivated to explore whether some form of network architecture might be compatible with our SO (or ASO) approach. Such a network might involve quantum computers interlinked within a building, analogously to a conventional HPC facility, and relevant methods of linking processors have been experimentally realised. Alternatively, for suitably compact platforms the network might correspond to linked QC processors on a single chip, analogous to today's multicore CPUs – indeed multicore quantum computing has recently been explored [101]. In either case, it is realistic to assume that in a network of processor nodes the inter-node operations are slower than the intra-node ones.

Fortunately, the SO method is quite compatible with a network paradigm; there is a natural partitioning of the problem into nodes that each contain the registers (or sub-registers) associated with a given simulated particle. While data exchange between nodes is obviously required, it need not be a dominant component of the overall resource costing even if the physical links are slow.

In Figure 16 we show one possible partitioning – it is not the most granular, since one could assign individual sub-registers to cores, but it does strike a good balance between the intra- and inter-core operations. Note that the required connectivity between cores is merely linear and nearest-neighbour. We suppose that there are two forms of core: the compute nodes are responsible for all the processing that is associated with the SO method, while the simpler memory nodes only store registers transiently.

Each node includes communication resources which facilitate transfer of quantum information between nodes, for example through the use of teleportation enabled by high-

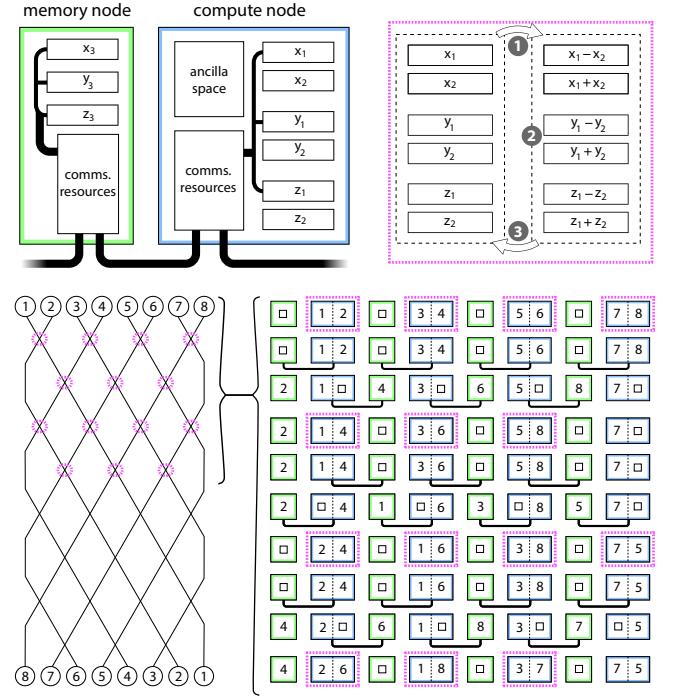


FIG. 16. Figures illustrating a possible ‘multicore’ architecture based on distributing the P -particle state over $P/2$ memory nodes ($3n_r$ qubits) and $P/2$ compute nodes ($6n_r$ qubits) which are interlinked either on-chip [101] or at the macro-scale. The upper-right panel indicates that three steps occur in a compute node. In step 1, quantum addition and subtraction are used to move to the relative and total coordinates (and step 3 will reverse this). In the middle step 2, the relative coordinate is used to apply phase shifts required by the SO cycle, and optionally we apply an augmentation step as described earlier. The lower panels show how parallelised phases alternatingly process and permute the data. The numbers within the circles and squares are the labels of particles; they correspond to the subscripts of the x , y and z symbols in the upper-right panel.

quality shared Bell pairs. The ‘comms resources’ would thus correspond to Bell pair distribution, purification, and buffering. Such processes can occur independently of the main computation and simultaneously with it, and need not involve a large number of qubits; see for example the analysis in Ref. [101, 102].

In the scheme illustrated in Fig. 16 the transfer of a register from one node to another involves writing into an ‘empty’ register where all qubits are in state $|0\rangle$. If the individual qubits are in fact encoded logical states formed of many physical qubits, then this would introduce a multiplicative factor in the Bell pair count, but it would remain linear in the register size. Ideally the ‘comms’ hardware would be capable of generating the required Bell pairs within the time that the compute node requires for a full implementation of the particle-particle interaction component of the SO for the current pair (and any augmentation, as per Section III E). Given that this computation will require a gate depth of at least n_r^2 , there is scope for a factor of $10 - 100$ in the relative speeds of the intra-core computations and the inter-code Bell preparation before the

latter would become a bottle neck.

VI. SUMMARY AND OUTLOOK

In this work, we numerically demonstrate for the first time the real-space grid propagation of atomistic electrons using the split-operator method on emulated quantum computers. We first consolidate the correspondence between wavefunctions in position- and momentum-space, and their representation as pixels on a quantum register’s computational basis states, in order to anticipate the challenges of propagating wavepackets under a grid representation of the singular Coulomb potential. We then use exact simulations to determine the resources needed for accurate time propagation of a 2D hydrogen atom (a single electron bound by a classical attractive Coulomb potential). We assess the quality of the model by simulating a probabilistic, single-ancilla qubit version of the phase estimation algorithm which measures the global phase acquired by a state during propagation, as well as computing the kinetic and potential energy expectation value directly from the state. We confirm that, when propagating states with cusps at the Coulomb singularity, accurate representation of the Coulomb potential near the singularity is required, which demands a high spatial resolution, either with a large number of qubits or a compact simulation box. Propagation of high resolution grids must also be matched with small timesteps if one wishes to achieve chemically accurate time series data, resulting in deep circuits. It suffices to represent and propagate states that do not have significant amplitudes at the origin with coarse space and time grids. We also note that extracting observables using ancilla-based methods is generally more robust and resource efficient than direct sampling of the state register.

We then incorporate additional quantum simulation techniques which enrich the split-operator approach; instead of using the single-ancilla phase estimation method as a means of measuring the global phase, we use the phase measurement to purify the contribution of an eigenstate in superposition and achieve a probabilistic state preparation. We also model two different particle escape events, first with a linear ionizing electric field, and then scattering by a second incident particle modelled as a Gaussian wavepacket. We monitor the escape probabilities and attenuate the periodic image of outgoing amplitudes by introducing an additional non-unitary propagation in the form of a complex absorbing potential at the boundaries of the simulation box. We achieve the non-unitary transformation via weak measurement of a single ancilla qubit entangled to selected grid pixels that we wish to attenuate. Motivated to address the high qubit count and deep circuits necessary for the accurate time propagation of states with cusps at the ‘raw’ Coulomb singularity, we augment the split-operator scheme by introducing an additional phase that stabilises the imperfect propagation at the core of the Coulomb potential. We achieve far superior energy estimates (by phase estimation) through this augmentation. Finally, we numerically demonstrate the generalisation of the quantum split-operator to many-body problems with a 36 qubit dynamics simulation of a helium atom.

From our numerical results, we extrapolate to molecular systems. We observe that, while choice of the minimum necessary spatial resolution does not scale with system size but only with the largest nuclear charge, achieving these resolutions

by a simple ‘brute-force’ increase in qubits may lead to prohibitive computational times even for small molecules. This motivates the use of techniques such as the augmented split-operator as well as alternative representations of the Coulomb potential which can reduce the qubit and time overhead. We conclude by arguing that the straightforward partitioning of particles into spatial sub-registers within the grid representation lends itself to the network paradigm of interconnected quantum processors.

The first-principles quantum molecular dynamics simulations enabled by the methods discussed in this work can be thought of as digital experiments that provide data for learning models, in keeping with the ideas presented in [3]. Moreover, recent work such as [103] speaks to a possible use case where large scale simulations can become part of a learning/prediction cycle which augments physical experiments. We believe the quantum split-operator may prove itself superior to classical quantum molecular dynamics simulations relatively early in the era of fault tolerant machines. We have already noted that the technique itself leads to robust methods for measuring observables such as phase estimation. We have also noted that we are free to check certain properties like particle symmetries quite cheaply at any time, analogous to stabilisers of a code, and their cost of evaluation can be low even if preparing the state was not. In the attenuation case, we have also used frequent, weak measurement to modify the evolution of the system. In light of this, it is interesting to ask whether the grid method can be inherently robust to errors. It may be possible to craft a version of the algorithm where the majority of harmful errors will cause the state to fail a validation check, while less damaging errors are mitigated by a non-unitary component in the dynamics.

A natural next step is to explore the use of multi-resolution grids as employed in e.g. MADNESS [104, 105], the highly successful classical computing program for real-space grid simulations. We are well-motivated to incorporate such ideas given that the present paper has revealed remarkable variation in resolution requirements – even across different states of a single system. Moreover, particles within many-body systems can often be considered as localized, presenting the opportunity for frugal representations based on that locality. However while multi-resolution grids might reduce the qubit count considerably, it would likely be at the cost of more sophisticated time propagators and basis transformations. Thus it is important to establish whether the methods in e.g. [104, 105] can indeed be translated successfully to the quantum context.

The split-operator method was first introduced as an explicit time stepping scheme for modelling lasers in the atmosphere, and the aforementioned MADNESS facility is used in a variety of settings. It is therefore obvious that the grid simulation techniques presented here can be generalised to modelling systems beyond quantum chemistry. The solution to any Cauchy type initial value problem

$$\frac{\partial}{\partial t} \Psi(x, t) = \hat{\mathcal{D}}(t) \Psi(x, t) \quad (46)$$

with a time-dependent differential operator that may be separated into operators that are respectively diagonal in position- and momentum-space

$$\hat{\mathcal{D}}(t) = \hat{\mathcal{D}}_1(t) + \hat{\mathcal{D}}_2(t) \quad (47)$$

can be approximated with the split-operator approach [106]. Many problems of interest can be modelled with Cauchy type

partial differential equations. For example, modelling of high energy particles requires reconciliation of quantum mechanics with special relativity. Models such as the Dirac and the Klein-Gordon equations are Cauchy type and may benefit from a quantum split-operator modelling. Another somewhat lucrative extension of the methods described in this work is application to financial engineering [107]. Quantum advantage is often promised for the modelling of how the prices of assets, such as options and derivatives, evolve over time [108, 109], which is key to executing purchases and sales that maximise the eventual payoff from trading such assets. These assets often have complex underlying dependence on random variables, and has in practice been numerically modelled using stochastic Monte Carlo methods with expensive computational resource. In the same manner as pixelating real-space wavefunctions and storing them in the computational basis states of a quantum computer, probability distributions corresponding to asset prices can be discretised and loaded into quantum registers. The distributions can then be evolved forward in time under the influence of the dependent variables. A very relevant model is the Black-Scholes-Merton equation [110, 111], a Cauchy type partial differential equation, used to model the price dynamics of European-style options [112]. Because of its form, the split-operator method can be directly applied for propagating distributions under the above model.

The use of repeated ancilla measurements during simulation, as employed in this work and others (e.g. [24]), extends the grid simulation techniques presented to non-unitary propagation processes. It remains to be seen whether the advantages in quantum molecular dynamics simulations using the split-operator translate to these other generalisations.

ACKNOWLEDGEMENTS

HHSC would like to thank J. Wu, D. Marti-Dafcik, P. Ollitrault, A. Lieberherr, A. Rattew, H. Jnane, J. P. Malhado, T. Carrington, J. Richardson and D. Manolopolous for helpful discussions. All the authors would like to thank B. Koczor and S. McArdle for their expert advice. HHSC is funded by the Croucher Foundation, Hong Kong. SCB acknowledges support from the EPSRC QCS Hub EP/T001062/1, from U.S. Army Research Office Grant No. W911NF-16-1-0070 (LOGIQ), and from EU H2020-FETFLAG-03-2018 under the grant agreement No 820495 (AQTION). The authors would like to acknowledge the use of the University of Oxford Advanced Research Computing (ARC) facility [113] in carrying out this work.

-
- [1] D. C. Clary, *Science*, 2008, **321**, 789–791.
 - [2] R. Babbush, D. W. Berry, J. R. McClean and H. Neven, *npj Quantum Information*, 2019, **5**, 1–7.
 - [3] J. R. McClean, N. C. Rubin, J. Lee, M. P. Harrigan, T. E. O’Brien, R. Babbush, W. J. Huggins and H.-Y. Huang, *The Journal of Chemical Physics*, 2021, **155**, 150901.
 - [4] T. Helgaker, P. Jorgensen and J. Olsen, *Molecular Electronic-Structure Theory*, Wiley, 2014.
 - [5] A. Szabo and N. Ostlund, *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*, Dover Publications, 1996.
 - [6] W. H. Miller, *Proceedings of the National Academy of Sciences*, 2005, **102**, 6660–6664.
 - [7] R. Conte and M. Ceotto, in *Semiclassical Molecular Dynamics for Spectroscopic Calculations*, John Wiley and Sons, Ltd, 2020, ch. 19, pp. 595–628.
 - [8] D. Tannor, *Introduction to Quantum Mechanics*, University Science Books, 2007.
 - [9] D. A. Micha and I. Burghardt, *Quantum dynamics of complex molecular systems*, Springer, 2007, vol. 83.
 - [10] E. R. Bittner, *Quantum dynamics: applications in biological and materials systems*, CRC press, 2009.
 - [11] F. Gatti, *Molecular quantum dynamics: from theory to applications*, Springer, 2014.
 - [12] F. Gatti, B. Lasorne, H.-D. Meyer and A. Nauts, *Applications of quantum dynamics in chemistry*, Springer, 2017, vol. 98.
 - [13] B. F. E. Curchod and T. J. Martínez, *Chemical Reviews*, 2018, **118**, 3305–3336.
 - [14] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell *et al.*, *Nature*, 2019, **574**, 505–510.
 - [15] H.-S. Zhong, H. Wang, Y.-H. Deng, M.-C. Chen, L.-C. Peng, Y.-H. Luo, J. Qin, D. Wu, X. Ding, Y. Hu, P. Hu, X.-Y. Yang, W.-J. Zhang, H. Li, Y. Li, X. Jiang, L. Gan, G. Yang, L. You, Z. Wang, L. Li, N.-L. Liu, C.-Y. Lu and J.-W. Pan, *Science*, 2020, **370**, 1460–1463.
 - [16] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik and J. L. O’brien, *Nature communications*, 2014, **5**, 1–7.
 - [17] J. A. Fleck, J. Morris and M. Feit, *Applied physics*, 1976, **10**, 129–160.
 - [18] I. Kassal, S. P. Jordan, P. J. Love, M. Mohseni and A. Aspuru-Guzik, *Proceedings of the National Academy of Sciences*, 2008, 18681–18686.
 - [19] D. S. Abrams and S. Lloyd, *Phys. Rev. Lett.*, 1997, **79**, 2586–2589.
 - [20] I. D. Kivlichan, N. Wiebe, R. Babbush and A. Aspuru-Guzik, *Journal of Physics A: Mathematical and Theoretical*, 2017, **50**, 305301.
 - [21] P. J. Ollitrault, G. Mazzola and I. Tavernelli, *Phys. Rev. Lett.*, 2020, **125**, 260511.
 - [22] Y. Su, D. W. Berry, N. Wiebe, N. Rubin and R. Babbush, *PRX Quantum*, 2021, **2**, 040332.
 - [23] B. Poirier and J. Jerke, *Full-Dimensional Schrödinger Wavefunction Calculations using Tensors and Quantum Computers: the Cartesian component-separated approach*, 2021.
 - [24] T. Kosugi, Y. Nishiya and Y. ichiro Matsushita, *Probabilistic imaginary-time evolution by using forward and backward real-time evolution with a single ancilla: first-quantized eigensolver of quantum chemistry for ground states*, 2021.
 - [25] N. C. Jones, J. D. Whitfield, P. L. McMahon, M.-H. Yung, R. V. Meter, A. Aspuru-Guzik and Y. Yamamoto, *New Journal of Physics*, 2012, **14**, 115023.

- [26] R. Babbush, N. Wiebe, J. McClean, J. McClain, H. Neven and G. K.-L. Chan, *Phys. Rev. X*, 2018, **8**, 011044.
- [27] T. Liu, J.-G. Liu and H. Fan, *Quantum Information Processing*, 2021, **20**, 1–21.
- [28] I. D. Kivlichan, C. Gidney, D. W. Berry, N. Wiebe, J. McClean, W. Sun, Z. Jiang, N. Rubin, A. Fowler, A. Aspuru-Guzik, H. Neven and R. Babbush, *Quantum*, 2020, **4**, 296.
- [29] R. Kosloff, *The Journal of Physical Chemistry*, 1988, **92**, 2087–2100.
- [30] R. Kosloff, *Annual Review of Physical Chemistry*, 1994, **45**, 145–178.
- [31] C. Leforestier, R. H. Bisseling, C. Cerjan, M. D. Feit, R. Friesner, A. Guldberg, A. Hammerich, G. Jolicard, W. Karrlein, H.-D. Meyer, N. Lipkin, O. Roncero and R. Kosloff, *J. Comput. Phys.*, 1991, **94**, 59–80.
- [32] J. C. Light and T. Carrington Jr., in *Discrete-Variable Representations and their Utilization*, John Wiley and Sons, Ltd, 2000, pp. 263–310.
- [33] B. I. Schneider, L. A. Collins and S. X. Hu, *Phys. Rev. E*, 2006, **73**, 036708.
- [34] F. Gatti, B. Lasorne, H.-D. Meyer and A. Nauts, in *Introduction to Numerical Methods*, Springer International Publishing, Cham, 2017, pp. 201–285.
- [35] M. Feit, J. Fleck and A. Steiger, *Journal of Computational Physics*, 1982, **47**, 412–433.
- [36] M. D. Feit and J. A. Fleck, *The Journal of Chemical Physics*, 1983, **78**, 301–308.
- [37] R. Sadeghi and R. T. Skodje, *The Journal of Chemical Physics*, 1993, **99**, 5126–5140.
- [38] D. H. Zhang and J. Z. H. Zhang, *The Journal of Chemical Physics*, 1995, **103**, 6512–6519.
- [39] V. N. Serov, V. B. Sovkov, V. S. Ivanov and O. Atabek, *The Journal of Chemical Physics*, 2001, **115**, 6450–6458.
- [40] X. Chen and V. S. Batista, *The Journal of Chemical Physics*, 2006, **125**, 124313.
- [41] A. D. Bandrauk and H. Shen, *The Journal of Chemical Physics*, 1993, **99**, 1185–1193.
- [42] S. M. Greene and V. S. Batista, *Journal of Chemical Theory and Computation*, 2017, **13**, 4034–4042.
- [43] Y. Wu and V. S. Batista, *The Journal of Chemical Physics*, 2003, **118**, 6720–6724.
- [44] Y. Wu and V. S. Batista, *The Journal of Chemical Physics*, 2004, **121**, 1676–1680.
- [45] V. A. Trofimov and N. Peskov, *Mathematical Modelling and Analysis*, 2009, **14**, 109–126.
- [46] C. Dion, A. Hashemloo and G. Rahali, *Computer Physics Communications*, 2014, **185**, 407–414.
- [47] B. Buonacorsi, B. Shaw and J. Baugh, *Phys. Rev. B*, 2020, **102**, 125406.
- [48] M. Ruf, H. Bauke and C. H. Keitel, *Journal of Computational Physics*, 2009, **228**, 9092–9106.
- [49] P. L. DeVries, *AIP Conference Proceedings*, 1987, **160**, 269–271.
- [50] J. A. C. Weideman and B. M. Herbst, *SIAM Journal on Numerical Analysis*, 1986, **23**, 485–507.
- [51] E. Faou, L. Gauckler and C. Lubich, *Forum of Mathematics, Sigma*, 2014, **2**, e5.
- [52] R. Kosloff, in *The Fourier Method*, ed. C. Cerjan, Springer Netherlands, Dordrecht, 1993, pp. 175–194.
- [53] C. Zalka, *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 1998, **454**, 313–322.
- [54] S. Wiesner, *Simulations of Many-Body Quantum Systems by a Quantum Computer*, 1996.
- [55] T. R. Nelson, A. J. White, J. A. Bjorgaard, A. E. Sifain, Y. Zhang, B. Nebgen, S. Fernandez-Alberti, D. Mozyrsky, A. E. Roitberg and S. Tretiak, *Chemical Reviews*, 2020, **120**, 2215–2287.
- [56] G. Benenti and G. Strini, *American Journal of Physics*, 2008, **76**, 657–662.
- [57] R. D. Somma, *Quantum simulations of one dimensional quantum systems*, 2016.
- [58] H. Hirai, T. Horiba, S. Shirai, K. Kanno, K. Arimitsu, Y. O. Nakagawa and S. Koh, *Molecular Structure Optimization based on Electrons-Nuclei Quantum Dynamics Computation*, 2021.
- [59] R. Kosloff and D. Kosloff, *Journal of Computational Physics*, 1986, **63**, 363–376.
- [60] J. Boyd, *Chebyshev and Fourier Spectral Methods: Second Revised Edition*, Dover Publications, 2001.
- [61] C. C. Marston and G. G. Balint-Kurti, *The Journal of Chemical Physics*, 1989, **91**, 3571–3576.
- [62] H. Buhrman, R. Cleve, J. Watrous and R. de Wolf, *Phys. Rev. Lett.*, 2001, **87**, 167902.
- [63] S. Choi and J. Vaníček, *The Journal of Chemical Physics*, 2019, **150**, 204112.
- [64] J. Roulet, S. Choi and J. Vaníček, *The Journal of Chemical Physics*, 2019, **150**, 204113.
- [65] M. Suzuki, *Journal of Mathematical Physics*, 1991, **32**, 400–407.
- [66] A. M. Childs, Y. Su, M. C. Tran, N. Wiebe and S. Zhu, *Phys. Rev. X*, 2021, **11**, 011020.
- [67] T. Häner, M. Roetteler and K. M. Svore, *Optimizing Quantum Circuits for Arithmetic*, 2018.
- [68] Y. R. Sanders, D. W. Berry, P. C. Costa, L. W. Tessler, N. Wiebe, C. Gidney, H. Neven and R. Babbush, *PRX Quantum*, 2020, **1**, 020312.
- [69] B. Poirier, *Efficient Evaluation of Exponential and Gaussian Functions on a Quantum Computer*, 2021.
- [70] T. G. Draper, *Addition on a Quantum Computer*, 2000.
- [71] A. Y. Kitaev, *Quantum measurements and the Abelian Stabilizer Problem*, 1995.
- [72] M. Dobšíček, G. Johansson, V. Shumeiko and G. Wendin, *Phys. Rev. A*, 2007, **76**, 030306.
- [73] E. Farhi, J. Goldstone, S. Gutmann and M. Sipser, *Quantum Computation by Adiabatic Evolution*, 2000.
- [74] D. Poulin and P. Wocjan, *Phys. Rev. Lett.*, 2009, **102**, 130503.
- [75] D. W. Berry, M. Kieferová, A. Scherer, Y. R. Sanders, G. H. Low, N. Wiebe, C. Gidney and R. Babbush, *npj Quantum Information*, 2018, **4**, 1–7.
- [76] U. V. Riss and H. D. Meyer, *Journal of Physics B: Atomic, Molecular and Optical Physics*, 1995, **28**, 1475–1493.
- [77] G. Jolicard and E. J. Austin, *Chemical Physics Letters*, 1985, **121**, 106–110.
- [78] D. Neuhauser and M. Baer, *The Journal of Chemical Physics*, 1989, **90**, 4351–4355.
- [79] U. V. Riss and H. D. Meyer, *Journal of Physics B: Atomic, Molecular and Optical Physics*, 1993, **26**, 4503–4535.
- [80] G. G. Balint-Kurti and Á. Vibók, in *Complex Absorbing Potentials in Time Dependent Quantum Dynamics*, ed. C. Cerjan, Springer Netherlands, Dordrecht, 1993, pp. 195–205.

- [81] S. Scheit, H.-D. Meyer, N. Moiseyev and L. S. Cederbaum, *The Journal of Chemical Physics*, 2006, **124**, 034102.
- [82] C. Gidney and N. C. Jones, *A CCCZ gate performed with 6 T gates*, 2021.
- [83] F. M. Fernandez, *Journal of Physics A: Mathematical and General*, 1991, **24**, 1351–1353.
- [84] R. L. Hall, N. Saad, K. D. Sen and H. Ciftci, *Phys. Rev. A*, 2009, **80**, 032507.
- [85] J. Spencer and A. Alavi, *Phys. Rev. B*, 2008, **77**, 193110.
- [86] F. J. Rogers, H. C. Grabske and D. J. Harwood, *Phys. Rev. A*, 1970, **1**, 1577–1586.
- [87] T. Jones, A. Brown, I. Bush and S. C. Benjamin, *Sci. Reports*, 2019, **9**, 10736.
- [88] T. Jones and S. C. Benjamin, *Quantum Sci. Techno.*, 2020, **5**, 034012.
- [89] R. Meister, *pyQuEST - A Python interface for the Quantum Exact Simulation Toolkit (QuEST)*, 2022, <https://github.com/rrmeister/pyQuEST>.
- [90] D. G. W. Parfitt and M. E. Portnoi, *Journal of Mathematical Physics*, 2002, **43**, 4681–4691.
- [91] X. L. Yang, S. H. Guo, F. T. Chan, K. W. Wong and W. Y. Ching, *Phys. Rev. A*, 1991, **43**, 1186–1196.
- [92] A. Mokhtari, P. Cong, J. Herek and A. Zewail, *Nature*, 1990, **348**, 225–227.
- [93] A. Bhattacharyya, *Bull. Calcutta Math. Soc.*, 1943, **35**, 99–109.
- [94] Wikipedia page ‘Atomic radii of the elements (data page)’ collates multiple data sets, <https://www.pnas.org/content/early/2008/11/24/080245105>.
- [95] M. Beverland, E. Campbell, M. Howard and V. Klyuchnikov, *Quantum Science and Technology*, 2020, **5**, 035009.
- [96] C. Gidney and A. G. Fowler, *Flexible layout of surface code computations using AutoCCZ states*, 2019.
- [97] D. Litinski, *Quantum*, 2019, **3**, 128.
- [98] A. G. Fowler, M. Mariantoni, J. M. Martinis and A. N. Cleland, *Phys. Rev. A*, 2012, **86**, 032324.
- [99] B. Koczor, *Phys. Rev. X*, 2021, **11**, 031057.
- [100] W. J. Huggins, S. McArdle, T. E. O’Brien, J. Lee, N. C. Rubin, S. Boixo, K. B. Whaley, R. Babbush and J. R. McClean, *Phys. Rev. X*, 2021, **11**, 041036.
- [101] H. Jnane, B. Undseth, Z. Cai, S. C. Benjamin and B. Koczor, *Multicore Quantum Computing*, 2022.
- [102] R. Nigmatullin, C. J. Ballance, N. de Beaudrap and S. C. Benjamin, *New Journal of Physics*, 2016, **18**, 103028.
- [103] A. Hosseiniزاده, N. Breckwoldt, R. Fung, R. Sepehr, M. Schmidt, P. Schwander, R. Santra and A. Ourmazd, *Nature*, 2021, **599**, 697–701.
- [104] R. J. Harrison, G. Beylkin, F. A. Bischoff, J. A. Calvin, G. I. Fann, J. Fosso-Tande, D. Galindo, J. R. Hammond, R. Hartman-Baker, J. C. Hill *et al.*, *SIAM Journal on Scientific Computing*, 2016, **38**, S123–S142.
- [105] N. Vence, R. Harrison and P. Krstić, *Phys. Rev. A*, 2012, **85**, 033403.
- [106] H. Bauke and C. H. Keitel, *Computer Physics Communications*, 2011, **182**, 2454–2463.
- [107] R. Orús, S. Mugel and E. Lizaso, *Reviews in Physics*, 2019, **4**, 100028.
- [108] N. Stamatopoulos, D. J. Egger, Y. Sun, C. Zoufal, R. Iten, N. Shen and S. Woerner, *Quantum*, 2020, **4**, 291.
- [109] S. Chakrabarti, R. Krishnakumar, G. Mazzola, N. Stamatopoulos, S. Woerner and W. J. Zeng, *Quantum*, 2021, **5**, 463.
- [110] F. Black and M. Scholes, *Journal of Political Economy*, 1973, **81**, 637–654.
- [111] R. C. Merton, *The Bell Journal of Economics and Management Science*, 1973, **4**, 141–183.
- [112] T. Kariya and R. Y. Liu, in *Options, Futures and Other Derivatives*, Springer US, Boston, MA, 2003, pp. 9–26.
- [113] A. Richards, *University of Oxford Advanced Research Computing*, 2015.
- [114] Wolfram Research Incorporated, *Wolfram Engine*, <https://www.wolfram.com/engine/>, Champaign, IL, 2021.
- [115] N. J. Ward, I. Kassal and A. Aspuru-Guzik, *The Journal of Chemical Physics*, 2009, **130**, 194105.
- [116] L. Grover and T. Rudolph, *Creating superpositions that correspond to efficiently integrable probability distributions*, 2002.
- [117] P. Kaye and M. Mosca, International Conference on Quantum Information, 2001, p. PB28.
- [118] M. Möttönen, J. J. Vartiainen, V. Bergholm and M. M. Salomaa, *Quantum Info. Comput.*, 2005, **5**, 467–473.
- [119] G. Marin-Sánchez, J. Gonzalez-Conde and M. Sanz, *Quantum algorithms for approximate function loading*, 2021.
- [120] A. Kitaev and W. A. Webb, *Wavefunction preparation and resampling using a quantum computer*, 2009.
- [121] A. Holmes and A. Y. Matsuura, 2020 IEEE International Conference on Quantum Computing and Engineering (QCE), Los Alamitos, CA, USA, 2020, pp. 169–179.
- [122] A. Carrera Vazquez and S. Woerner, *Phys. Rev. Applied*, 2021, **15**, 034027.
- [123] A. G. Rattew, Y. Sun, P. Minssen and M. Pistoia, *Quantum*, 2021, **5**, 609.
- [124] L. Lin and Y. Tong, *Quantum*, 2020, **4**, 372.

Appendix A: QFT circuit

Figure 17 shows the circuit used.

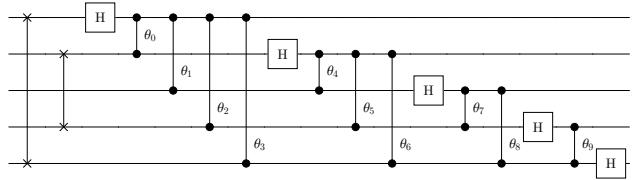


FIG. 17. The QFT circuit used to transform from the k -space to the real-space representation. The circuit shown is for a 5-qubit sub-register; the generalisation is the natural one.

Appendix B: Numerical modelling

Here we describe further details of the hardware, software, and configuration details relating to the numerical simulations presented in the main paper.

1. Emulation software and hardware

All numerical results in this manuscript were ultimately obtained through the Quantum Exact Simulation Toolkit (QuEST) [87], though interfaced through the Python pyQuEST [89] and Mathematica QuESTlink [88] software packages. The core QuEST simulator is written in C behind a hardware agnostic interface, allowing redeployment of the simulations in this manuscript between laptops, GPUs and distributed supercomputing facilities. This enabled multithreaded and GPU simulation of the modestly sized systems, such as the 25-qubit 2D Hydrogen scattering presented in Figure 12, directly within QuESTlink. While open-source, we note that the use of QuESTlink requires a Wolfram Engine environment, obtainable either through the commercial Mathematica product, or the recently released Wolfram Engine standalone [114].

Results for the largest scale systems we considered were generated with pyQuEST, which enables relatively low-level access to QuEST simulation primitives through a high-level Python interface. Importantly, pyQuEST inherits the capacity to run distributed tasks, where the numerical representation of a quantum state is partitioned between compute nodes cooperating over a network. This allows both the representation of states too large to fit into the memory of any single compute node, and their concurrent simulation – each multicore node is further able to parallelise its local simulation tasks through multithreading. In this manuscript, distributed pyQuEST was used to emulate 36-qubit quantum computers in our study of 3D helium, employing up to 32 compute nodes of the Oxford Advanced Research Computing (ARC) facility [113]. Each node contains 48 CPU cores, and took roughly 52 hours ($\approx 50\,000$ core hours) to process its 64 GiB partition of the full 1 TiB quantum state-vector.

2. Emulation of the split-operator

The QuEST family of emulators perform numerical simulation at the level of individual gates. Supported gates include all commonly used operations, and general unitaries of any number of control and target qubits, and therefore all operations required of the quantum algorithms presented in this manuscript. However, completing this work required the repeated simulation of circuits within which a series of contiguous gates could be more efficiently effected by a single invocation of a bespoke function. We implemented this optimisation to accelerate the simulations in this manuscript, which has since been integrated into the QuEST emulators. The interested reader may view these documented facilities [here](#). We now describe these optimisations, and where they were invoked, in detail.

Section II B described how our grid-based description of 3D multi-electron systems meant that particle-field and particle-particle interactions admit a unitary time evolution operator which can be split into step operators

$$U_U |n m l\rangle = \exp\left(-i \frac{Q \delta t}{\sqrt{n^2 + m^2 + l^2}}\right) |n m l\rangle, \quad (\text{B1})$$

$$U_V |n_1 m_1 l_1\rangle |n_2 m_2 l_2\rangle = \exp(-i\Theta) |n_1 m_1 l_1\rangle |n_2 m_2 l_2\rangle,$$

$$\Theta = \frac{2^{1-n_r} Q \delta t}{\sqrt{(n_1 - n_2)^2 + (m_1 - m_2)^2 + (l_1 - l_2)^2}},$$

Algorithm 1: Embarassingly parallel distributed simulation of the 3D U_U split-operator upon an N -qubit statevector $|\psi\rangle$, which is uniformly distributed between 2^k nodes. $\vec{\psi}$ is the 2^{N-k} vector of complex amplitudes stored in each node, with i -th element $\vec{\psi}[i]$, indexed from 0. Each node has a unique rank $0 \leq r < 2^k$. Integers q_l, q_m, q_n are the starting qubit indices of the contiguous n_r -qubit registers which together form substate $|n\rangle |m\rangle |l\rangle$ of Equation B1, via a two's complement signed binary encoding. Symbols & and \gg denote bit-wise AND and bit right-shift operators respectively. The outer **for** loop of our algorithm is trivially parallelised using multithreading or GPU acceleration.

```

1 apply $U_U(\vec{\psi}, Q, \delta t, q_l, q_m, q_n, n_r)$ 
2 // iterate every local basis state
3 for  $i$  in  $\{0, \dots, 2^{N-k} - 1\}$ 
4   // determine global index of basis state
5    $j = r 2^{N-k} + i$ 
6   // determine sub-register values of basis state
7    $l = \text{getRegVal}(j, q_l, n_r)$ 
8    $m = \text{getRegVal}(j, q_m, n_r)$ 
9    $n = \text{getRegVal}(j, q_n, n_r)$ 
10  // evaluate the phase
11   $\theta = Q \delta t / \sqrt{l^2 + m^2 + n^2}$ 
12  // modify the amplitude
13   $\vec{\psi}[i] = \exp(-i \theta) \vec{\psi}[i]$ 
14 // returns  $(b \geq 0)$ -th bit of integer  $j$ 
15 getBit( $j, b$ )
16 return  $(j \gg b) \& 1$ 
17 // returns the signed value of the  $n_r$  contiguous
  // qubits, starting at index  $q$ , in the  $j$ -th basis state
18 getRegVal( $j, q, n_r$ )
19  $v = 0$ 
20 for  $k$  in  $\{0, \dots, n_r - 2\}$ 
21    $v = v + 2^k \text{getBit}(j, q + k)$ 
22 if getBit( $j, q + n_r - 1$ ) is 1
23    $v = v - 2^{n_r - 1}$ 
24 return  $v$ 
```

where sub-register $|n m l\rangle$ encode (z, y, x) coordinates of a single particles, which may be negative (encoded with two's complement binary). An experimentalist must effect these $(d n_r)$ - and $(2 d n_r)$ -qubit operators through $\mathcal{O}(n_r^2)$ single-qubit and double-qubit phase gates. However, we can leverage that U_U and U_V are diagonal in the real space representation to numerically simulate them cheaper than even a *single* general unitary gate, in an *embarrassingly parallel* manner. We formally present such a scheme to effect U_U , the simplest of the split-operators, upon a distributed statevector in Algorithm 1. For clarity, we have excluded the pseudocode for additional functionality critical to our actual implementation, such as the ability to override the phases at particular l, m, n values to avoid phase divergences.

Use of Algorithm 1 and several similar strategies for the other split-operators was crucial in order to numerically study

the various simulation scenarios presented in the main paper without incurring impractical time costs.

We stress that the use of these high level functions, which do not involve specifying a circuit-level description, does not compromise the exact nature of our numerical emulation of a quantum processor. It is simply an expediency allowing us to arrive at that the exact behaviour with a more modest use of classical resources that would be required if we were to explicitly use gate-level operations throughout. In the real device, such circuit-level prescriptions must of course be used.

For the U_{kin} operator, which simply applies a phase dependent on the square of the binary number k in each of the system's sub-registers, this is straightforward: we require only a series of single- and two-qubit phase gates applied directly to the sub-registers [21, 57]. Of $O(n_r^2)$ such gates would be required for each sub-register of n_r qubits.

The implementation of U_U and U_V is more complex as it involves the Coulomb potential $1/|\mathbf{r}|$. In the main paper, Section II B, we describe the arithmetic that occurs in the case that we are applying U_V and therefore require sub-registers to represent $x_i - x_j$ etc. One could then apply a circuit C to compute the desired phase $1/r$ into an ancilla register (i.e. a binary representation of the desired phase to any given accuracy), followed by actually implementing that phase by a series of single-qubit phase gates applied to the ancilla qubits. Finally one would apply circuit C^\dagger to un-compute the ancilla's state, disentangling it from the main registers.

Any classical algorithm that computes $1/\sqrt{x^2 + y^2 + z^2}$ from inputs x, y, z will in principle suffice; such an algorithm might be expressed in terms of irreversible operations (AND, OR, XOR etc) but any such circuit can be re-expressed as classical reversible circuit and thus provide a suitable C . It is of course interesting to optimise this, especially since quantum gate sets are can potentially implement classically-reversible circuits more efficiently and we will wish to be as frugal as possible with the use of ancilla qubits. However even an inefficient implementation would suffice in the sense that it would not alter the overall scaling of the SO method: the computation of $1/r$ happens at the register level, and register size scales only very weakly with the complexity of the simulated system as argued in Section V.

Appendix C: Multi-qubit Phase Estimation and Fourier Analysis

In order that the present paper can be a useful resource for introducing the grid-based method, we now summarise the standard multi-qubit phase estimation method using the present paper's notation. This method is of course more powerful than the frugal single-ancilla technique described in the main paper.

The logarithmically-scaling phase estimation of Kitaev [71], trades higher qubit overhead for measuring the phase with far less sampling; in suitable cases, even in a single shot. Consider an ancillary register with S qubits and a working register containing the state of interest. We first conditionally apply $2^{S-1}N$ split-operator steps to the state, controlled by the most significant ancillary qubit. We then repeat by conditionally applying $2^{S-2}N$ split-operator cycles, now controlled by the next most significant ancillary qubit, and repeat again until every ancillary qubit has been used and the least significant qubit controls only N split-operator steps. At this point

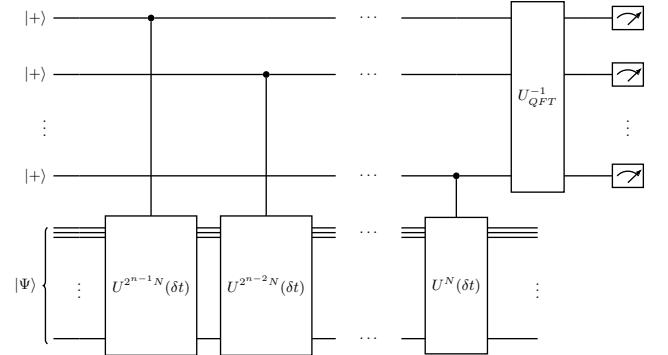


FIG. 18. Generalisation of the single-ancilla iterative phase estimation to the ‘canonical’ phase estimation using an n -qubit ancilla register. The i^{th} most significant ancilla qubit controls the propagation of $2^{n-i}N$ split-operator steps in the main register.

the quantum computer is in the state

$$\frac{1}{\sqrt{2^S}} \sum_{m=0}^{2^S-1} |m\rangle U^{mN}(\delta t) |\Psi\rangle. \quad (\text{C1})$$

If $|\Psi\rangle$ is an eigenstate $|\Psi_n\rangle$ with energy E_n , the state is thus

$$\begin{aligned} & \frac{1}{\sqrt{2^S}} |\Psi_n\rangle \sum_{m=0}^{2^S-1} e^{-iE_n m N \delta t} |m\rangle \\ & = \frac{1}{\sqrt{2^S}} |\Psi_n\rangle \sum_{m=0}^{2^S-1} e^{-i\phi m} |m\rangle \end{aligned} \quad (\text{C2})$$

where $\phi = E_n N \delta t$. The phase information ϕ is now encoded in the ancillary qubit register, and can be efficiently extracted using an inverse QFT on the ancillary register. Indeed, if $\phi = 2^k$ where $k \in \mathbb{N}$, then a QFT of the ancillary register should yield the binary representation of k upon measurement.

We also mention that classical Fourier analysis of the autocorrelation signal can also reveal the power spectrum, with resonance peaks that correspond to the stationary states of the system. This is in many ways the classical analogue of the canonical phase estimation method. Specifically, the power spectrum is peaked at the eigenvalues of \hat{H}_{tot} with heights corresponding to $|c_n|^2$

$$\frac{1}{\sqrt{2\pi}} \int \langle \Psi(t) | \Psi(0) \rangle e^{i\omega t} dt = \sqrt{2\pi} \sum_n |c_n|^2 \delta(\omega - E_n) \quad (\text{C3})$$

In practice, the discrete and finite time integration limits the resolution of the power spectrum, given by

$$\delta\omega = \frac{2\pi}{M_{\text{tot}} N \delta t} \quad (\text{C4})$$

where M_{tot} is the total number of time samples.

Appendix D: State preparation techniques

As discussed in the main paper, our use of emulated quantum computers means that we can side-step the challenge of

preparing a known, analytic initial state on our device – we simply load it into the emulator’s memory. For completeness, in this appendix we briefly review some of the literature that describes methods and costs for real quantum processors.

In 2009, Ward *et al.* [115] extended the work of Zalka [53] to develop such a method for generating physically relevant states in grid representations, which we summarise here. We begin with the task of loading suitable initial states into the particle quantum register. The objective is to distribute a single particle state ϕ centred about (x_0, y_0, z_0) within a simulation box of width L across the amplitudes of an N -qubit register

$$|0\rangle |0\rangle |0\rangle \Rightarrow \phi(\mathbf{r}) |n\rangle |l\rangle |m\rangle \quad (\text{D1})$$

where $\mathbf{r} = (n\delta r - x_0, l\delta r - y_0, m\delta r - z_0)$ and $\delta r = L/2^{nr}$. L must be sufficiently large to capture the significant parts of the wavefunction and the sum of all the amplitudes squared should be unity. In 1998, Zalka proposed a constructive algorithm for loading single particle distributions onto a register [53], which was subsequently either rediscovered or generalised, most notably by Grover and Rudolph in 2002 [116], in a number of contexts [117–120]. The method splits the norm of the single particle function N times across the N qubits by performing qubit rotations, the angles for which are determined by computing exponentially many integrals. Complex-valued functions will require additional procedures for including the phase in each amplitude. There are conflicting opinions on the efficiency of the method; some argue that the method has an exponential $O(2^N)$ overhead [119, 121], while others claim that it can be efficient [122], perhaps using quantum computers to solve the integrals [115]. In recent years, more novel approaches for initial state loading techniques have been proposed. Ollitrault *et al.* prepared Gaussian amplitudes using hybrid variational algorithms [21]. Holmes and Matsura used matrix product states to represent functions, which maps to state preparation circuits with linear-depth [121]. In particular, a noise-resistant technique for efficient preparation of normal distributions from Rattew *et al.* [123] shows promise for generalisation to the single particle functions required.

Loading multiple uncorrelated single particle states onto separate quantum registers is not sufficient to represent many-particle eigenstates of real molecular Hamiltonians. Ward *et al.* proposed loading occupied Hartree-Fock orbitals [5] into real-space particle registers using the aforementioned method by Zalka, followed by (anti)symmetrisation which we will discuss presently, to generate Slater-determinant type states. They further generalise this to preparing superpositions of Slater-determinant type states in the real-space particle registers, yielding exact molecular ground states through the creation of Full Configuration Interaction (FCI) states. Their method however demands the classical or quantum precomputation of the electronic structure in second-quantisation, loading the result in ancilla qubits, and achieving a transformation which loads the information from the second-quantised ancilla to the real-space particle registers. The authors did not detail how such a transformation can be achieved, and loading a factorial number of Slater determinants from the FCI state may pose challenges.

As noted in the main text, an alternative method is to drive an initial state that can be efficiently loaded onto a quantum computer towards the correlated molecular ground state. An advantage of this approach is that we need not have the ability to analytically describe the target state. Two common techniques include adiabatic evolution [73] and phase

estimation [71, 74, 75]. There are also protective methods, and the main paper presents numerics corresponding to one such technique.

Other techniques for ground state preparation which are compatible with the quantum split-operator method have been proposed. Similar to what we propose in the forthcoming Section III C for achieving propagation of a non-unitary complex attenuating potential using single-ancilla measurements, Kosugi *et al.* combines a single-ancilla measurement with forward and backward split-operator time propagation to implement a first quantized probabilistic imaginary time evolution [27] for preparing ground states in real space, which they have numerically emulated for parabolic potentials [24]. Lin *et al.* also proposed a near-optimal ground state preparation which does not require knowing the ground state energy in advance [124].

Finally, in the first quantized simulation framework considered here, we also need to explicitly enforce the right particle symmetries. Ward *et al.* improved upon work by Abrams and Lloyd from 1997 [19] to give a deterministic algorithm for this purpose. The protocol prepares a (anti)symmetrised state in an ancillary register containing $P(\log_2 P)$ qubits (where P is the number of particles in the system) and transfers the symmetry to the register storing the particle state, by simultaneously sorting the ancillary register and performing the same SWAP operations in the main register. Berry *et al.* further presented an improved poly-logarithmic algorithm for antisymmetrising initial states [75]. Because we can generate any initial state in classical emulations, where necessary we simply initialise spatially antisymmetrised states and load them into our emulation.

Appendix E: Placement of nuclear origin

We briefly report that placement of the nuclear origin between grid points does not significantly affect the dynamics (summarised in Table I).

When we time propagate the $\Psi_{1,1}$ state with the Coulomb potential centred at different fractions between two neighbouring grid points, the state visibly oscillates where grid points lie asymmetrically around the Coulomb origin. However, the magnitude of the loss in fidelity is limited. Most importantly, the energy from the phase estimation is not influenced at all by the skew. We conclude that discrepancies in the placement of grid points relative to the system can, if needed, be alleviated with a brute force increase in the spatial resolution and correspondingly the time resolution, but its influence on observables related to state fidelity can be negligible.

	0.5	0.4	0.3	0.2	0.1
Max loss	7×10^{-7}	0.00085	0.00021	0.0042	0.0022
Phase Est. Err. ($10^{-5} E_h$)	-1.84	-1.11	0.29	-1.66	5.98

TABLE I. Placement of the Coulomb nuclear potential between grid points and its influence on the autocorrelation as well as the energy predicted through phase estimation.

Appendix F: Resource scaling

In Section V we considered the way in which the grid-based method's resources can be expected to scale. We noted that there are detailed studies in the literature. The contribution to this topic made by the present paper is that its emulated algorithms may elucidate some of the constants that arise in any scaling analysis.

We argued that the qubit count should scale with particle count P according to $3n_r P$ with

$$n_r \approx C_3 + \log_2(Z_{\max}) + \frac{1}{3} \log_2(P) \quad (\text{reiterating Eqn. 45})$$

where Z_{\max} accounts for the highest nuclear charge in the modelled system. We noted that this crude expression does not account for the fact that a given atom's radius has a sub-linear dependence on the number of electrons; accounting for this would lower the outcome.

We can proceed to make an estimate of the root constant C_3 . We begin by noting that in the numerical modelling reported here, remarkably accurate and stable simulations can be achieved with as few as 6 qubits per x , y or z sub-register (thus, 18 qubits per 3D particle). Methods such as ancilla-based global phase tracking (Section III A), requiring only one additional qubit, can obtain then eigenenergies with accuracy up to 6 decimal places. Our results using the augmented SO method in Section IV F suggest that even core-peaked electronic states can be modelled without significant increase in resolution. The results of that section did suggest that more than $n_r = 6$ qubits per register would be needed, but only modestly so. For present purposes we therefore take an optimistic stance and assume that $n_r = 8$ can suffice for scenarios where the simulation box size is optimised for just the eigenstate in question.

Consider the task of modelling a single atom of substantially higher nuclear charge, say potassium ($Z = 19$) which has a particularly large covalent bond radius [94]. While innermost s-electrons might indeed be modelled with $n_r = 8$ qubits per register, to represent all electrons (properly antisymmetrized) within the same model would require a simulation box large enough to encompass the outermost shell. Potassium's outer shell radius is around 6 times that of hydrogen [94], while its innermost shell might be estimated to be a factor of $1/Z = 1/19$ times smaller than that of hydrogen. This implies an overall factor of 114, corresponding to 7 additional qubits for a total of 15. If we accept this as worse-than-typical (recognising that many atoms of a molecule will be lighter or more compact) we can use Eqn. 45 with $n_r = 15$, $P = Z_{\max} = 19$ to infer $C_3 = 10$ (rounding up).

Before using this expression to estimate the resources required for a relatively complex structure, it is interesting to consider the special case of H_2O . An optimistic resource audit can process as follows: Using standard sources for characteristic dimensions, we can assume that the simulation box needed for the entire molecule is about $4 = 2^2$ times that needed for the most high curved states (the oxygen s-orbitals). Taking $n_r = 7$ as adequate for each dimension of a simulation box adjusted for such a core state, we therefore require at least $n_r = 2 + 7 = 9$ qubits for each dimension, for each particle. The total (assuming we do not model the nuclear particles as quantum objects but rather use classical nuclear potentials) is $9 \times 3 \times 10 = 270$ qubits. We might round this to 300, recognising that multiple ancillas may be required even in a very frugal implementation.

A more challenging case is that of adenosine triphosphate ($\text{C}_{10}\text{H}_{16}\text{N}_5\text{O}_{13}\text{P}_3$) which is a profoundly important component of all living systems. The highest-charge nuclei are those of phosphorus, giving us $Z_{\max} = 15$. The total electron count P in neutral ATP is $10 \times 6 + 16 \times 1 + 5 \times 7 + 13 \times 8 + 3 \times 15 = 260$. Eqn. 45 then gives $n_r = 17$ rounding up. Recognising that ATP is a linear molecule, whereas Eqn. 45 was derived assuming a cubic simulation box, we might allow $n_r = 19$ for one of the directions. The total qubit count is then $260(19 + 2 \times 17) = 13,780$. We could call this 16,000 allowing for a frugal ancilla overhead.

Estimating the overall time cost for simulation is difficult but certainly interesting to speculate about. The overall 'wall clock' duration of a simulation is determined by the time required for each complete SO cycle, how much evolution each SO step represents (the time resolution δt), and the total duration of the dynamical process under investigation. The latter depends on the time and energy scales across which the events probed occur, which we postulate to be independent of system size. We assume that time resolution δt depends only on the δr sufficient to describe the tightly bound electrons of the element with highest nuclear charge, which also does not change with system size. Moreover the methods discussed in Sections III A and III B, and indeed the ingenious proposal in Ref. [24], indicate that energy estimation and state preparation could also be achieved using a number of time steps that does not scale significantly with the molecular size.

The time to complete each SO cycle does indeed scale with system size. There is a modest scaling due to the computation associated with introducing proper phases (equivalent to binary addition/subtraction, squaring, and inversion) which scales with the number of qubits in each register – although not, of course, with the number of qubits in total. Since the n_r scales only logarithmically with simulation box size L , this time factor should be poly-logarithmic in e.g. the number of particles $O(\log^k P)$. A more significant factor comes from the need to compute the electron-electron interaction for all possible pairings of electrons. The task can be divided into a series stages within each of which all electrons are paired-off and we can assume that computations relevant to electron-electron dynamics occur in parallel (see [25]). The number of such stages is $P - 1$ for P electrons, so that the total time for a complete SO implementation is of the order $O(P \log^k P)$. Considerations such as the time required to move from one pairing to another should also be borne in mind, but one finds that this should not scale with n even in devices with restricted connectivity (see Section VB).

Our QFT implementation used an explicit quantum circuit, while the conditional phases were applied directly using a bespoke algorithm (see Appendix B); if we focus on the QFT component and use the same implementation employed in our emulator (Fig 17) then $O(n_r^2)$ gates are required. A full SO cycle may involve operations equivalent to several QFTs, since they constitute an approach to the addition and squaring operations that we employ. Moreover other operations may require n_r^2 gates, such as the application of the $\propto k^2$ phases for the kinetic operator. If we therefore take $10n_r^2$ as a plausible speculation for the gate cost of each SO sub-step in a parallelised implementation (each row of the lower panel in Fig. 16), then for the ATP molecule for each SO cycle we have a total gate depth of $260 \times 10 \times 19^2$ which is of order one million (and $260/2 = 130$ such processes occurring in parallel throughout that cycle).

Many of the simulations presented in the main paper used

of order 1000 SO cycles in order to generate interesting dynamics with good precision. This count would diminish with the adoption of more sophisticated Trotter sequences than the lowest-order pattern employed in the paper; but conversely, the dynamics of many interesting systems would likely evolve on timescales far longer than those of the simple systems we

have modelled. We must conclude that several billion steps would be required the total simulation (and hundreds of billions of individual gates). In the main paper we indicate how such numbers may translate to clock time, finding a answer that varies from an hour to days depending on the nature of the hardware platform and the logical encoding.