Supporting Information For: Quantum computation of stopping power for inertial fusion target design

Nicholas C. Rubin,[1,*] Dominic W. Berry,[2,†] Alina Kononov,[3] Fionn D. Malone,[1] Tanuj Khattar,[1] Alec White,[4] Joonho Lee,[1,5] Hartmut Neven,[1] Ryan Babbush,[1,‡] and Andrew D. Baczewski[3,§]

[1] *Google Quantum AI, Venice, CA, USA*
[2] *School of Mathematical and Physical Sciences, Macquarie University, Sydney, NSW, AUS*
[3] *Quantum Algorithms and Applications Collaboratory, Sandia National Laboratories, Albuquerque, NM, USA*
[4] *Quantum Simulation Technologies Inc., Boston, MA, USA*
[5] *Department of Chemistry and Chemical Biology, Harvard University, Cambridge, MA, USA*

CONTENTS

I. INITIAL STATE PREPARATION

The WDM regime is typified by an electronic temperature comparable to the electronic chemical potential [1, 2], thus it is particularly important to capture the impact of temperature on the initial state. Ideally we would initialize the system in a product state between the electrons and projectile, in which the electrons are in their exact thermal equilibrium state and the projectile is in a state with a sharply defined velocity, $\rho = \exp(-\beta H_0)/\mathrm{Tr}\left[\exp(-\beta H_0)\right] \otimes |\psi_{\text{proj}}(t=0)\rangle\langle\psi_{\text{proj}}(t=0)|$.

* Corresponding author: nickrubin@google.com
† Corresponding author: dominic.berry@mq.edu.au
‡ Corresponding author: babbush@google.com
§ Corresponding author: adbacze@sandia.gov

Unfortunately preparing the thermal ensemble on the electronic subsystem is believed to be exponentially hard for generic local Hamiltonians, even on a quantum computer. Here we assume that the initial thermal distribution of the system is well described by a finite-temperature mean-field approach such as thermal Hartree–Fock or Mermin Kohn-Sham DFT [3, 4], which is easy to prepare on a quantum computer [5]. Previous path integral Monte Carlo calculations of the static properties of WDM [6–8] suggest that Mermin Kohn-Sham DFT is often an excellent approximation where the domains of feasibility overlap, and it is by far the most popular *ab-initio* approach used in the field. The initial state of the calculation is thus chosen to be a Slater determinant drawn from the solution of a Mermin Kohn-Sham DFT calculation, noting that we do not expect that quantum resource estimates will be sensitive to the choice of exchange-correlation functional. The use of a thermal Hartree–Fock reference state may also be appropriate, consistent with its use as the conventional starting point for building wave function methods. The liquid-like ordering common in WDM along with the long-range excitations occurring in stopping power simulations often necessitate supercells large enough that a single-point reciprocal-space quadrature is justifiable (e.g., the Baldereschi mean-value point [9] or Γ point). We assume our initial state to have been drawn from such a single-point calculation.

To prepare a sample from this ensemble we use the improved Slater determinant state preparation protocol described in Ref. [5]. The initial orbitals

$$|\psi_i\rangle = \sum_{p\in G} c_{i,p}|p\rangle \tag{1}$$

are used to generate the Slater determinant

$$|\psi_{SD}\rangle = |\psi_1 \wedge \ldots \wedge \psi_\eta\rangle, \tag{2}$$

where p indexes the wavenumbers in a plane-wave basis and $\wedge$ is the antisymmetric tensor product. This state is prepared with $\tilde{O}(N\eta)$ cost using the Givens rotation protocol, which is applied sequentially to a second-quantized representation of the initial state. This cost is a negligible additive contribution to the time-evolution cost and not accounted for in the resource estimates in Section **??**.

However, $|\psi_{SD}\rangle$ is only a single sample from the initial mean-field density matrix associated with the Mermin Kohn-Sham solution. Each Mermin Kohn-Sham orbital $|\psi_i\rangle$ has a temperature-dependent occupation according to the Fermi-Dirac distribution, and thanks to the mean-field nature of the reference state, the probability associated with a particular Slater determinant is proportional to the product of probabilities that the corresponding set of single-particle orbitals is occupied (and its complement is unoccupied). We are careful to note that we are able to circumvent the prohibitive growth in the number of partially occupied Mermin Kohn-Sham orbitals with temperature, which limits the feasible system sizes and temperatures in many classical mean-field approaches to WDM. This is because any given Slater determinant has support on only η orbitals, though we will still need a classical representation of thermally occupied high-energy orbitals. The state preparation, time evolution, and measurement steps must be repeated N_s times to sample a thermal distribution over initial electronic conditions, and a (potentially) different Slater determinant will be prepared for each initial sample. This is accounted for in the total sampling requirements and it contributes multiplicatively to the total resource estimates. Numerical tests indicate that sampling from the attendant canonical and grand canonical ensembles have similar overheads, so we choose to develop our protocol for the canonical ensemble to avoid the need for preparing states with different particle number. Beyond numerical tests, we generally expect this to be a good choice for the WDM regime due to its low compressibility.

The projectile state $|\psi_{\text{proj}}(t=0)\rangle$ describes the quantum projectile nucleus starting at the same position as in the classical Mermin Kohn-Sham state used to intialize the electronic subsystem, but moving with a velocity v_{proj}. Preparing the projectile register involves two steps: (1) replacing the corresponding point charge with a Gaussian charge distribution with real-space standard deviation $\sigma_r = \sigma_k^{-1}$ and (2) translating the average momentum of that charge distribution to the initial momentum of the projectile, k_{proj}. In momentum space the resulting initial condition on the projectile wave packet is

$$\psi_{\text{proj}}(k, t=0) = \sqrt{\frac{1}{(2\pi)^{3/2}\sigma_k^3}}\, e^{ik\cdot R_{\text{proj}}}\, e^{-\|k-k_{\text{proj}}\|^2/4\sigma_k^2}. \tag{3}$$

There is no temperature dependence in the initial nuclear wave packet because it is far from equilibrium in a state with a relatively sharply peaked velocity ($\sigma_k \ll k_{\text{proj}}$). Subsequent non-BO dynamics of the coupled electron-projectile system will cause this wave packet to slow down such that its mean velocity will decay linearly, on average, with the average deceleration proportional to the stopping power. This initial Gaussian wave packet can be prepared using a method in Ref. [10] that contributes an additive $\mathcal{O}(\log N_n)$ cost that is negligible relative to the cost of time evolution.

II. CHOOSING THE PROJECTILE'S INITIAL VARIANCE

The projectile wave packet will have support on a range of momenta and will disperse (i.e., stop at different rates) if that range is too large. As long as the momentum gradient of the stopping power is relatively small over the dominant momentum components of the wave packet, this dispersion will be negligible over the relevant time evolution and we can treat the standard deviation as approximately fixed to facilitate resource estimation. Thus the value of σ_k, a free parameter, should be set to facilitate efficient sampling in the nuclear momentum (computational) basis to realize the fewest circuit repetitions and shortest run time.

While it might appear that we can make the sampling problem arbitrarily efficient by reducing σ_k, there is a trade-off in replacing the nuclear point charge in the BO problem with an explicit wave packet in the non-BO problem. Making the wave packet too narrow in momentum space will spread out the nuclear charge to an unphysical extent in real space, such that the resulting response will no longer represent a physical nuclear projectile traversing the medium. One way to understand this is in terms of the equivalent electrostatic potential of the projectile wave packet,

$$\frac{\zeta_{\text{proj}}\,\text{erf}\left(\|r-R_{\text{proj}}\|/\sqrt{2}\sigma_r\right)}{\|r-R_{\text{proj}}\|}, \tag{4}$$

where the error function approaches 1 for $\|r-R_{\text{proj}}\| \gg \sigma_r$, and the potential appears to be equivalent to that of a point charge. However, for relatively small values of $\|r-R_{\text{proj}}\|$ the effective nuclear charge is reduced along with the strength of the interaction between the projectile and both the electronic and nuclear degrees of freedom. One might then be tempted to set σ_r to be consistent with the physical extent of the projectile nucleus (e.g., $\sim 10^{-5}\,a_0$ for a proton or alpha particle, where a_0 is the Bohr radius), but the corresponding σ_k would then be $\sim 10^5$ a.u. and the sampling efficiency would be vastly degraded.

To quantify the error associated with replacing the point-like BO projectile nucleus with an explicit non-BO degree of freedom with a finite σ_k we can consider the difference between the BO electron-projectile force and the non-BO electron-projectile force projected onto a fixed form for the wave packet,

$$\Delta S_e = -\frac{4\pi}{\Omega}\sum_{i=1}^{\eta}\sum_{\substack{p,q\in G\\(p-q)\in G_0}}\frac{\zeta_{\text{proj}}(ik_{p-q}\cdot\hat{v}_{\text{proj}})}{\|k_{p-q}\|^2}\,|p\rangle\langle q|_i\left(e^{ik_{q-p}\cdot R_{\text{proj}}}-\sum_{\substack{\nu\in G\\ \nu-(p-q)\in G}}\psi^*_{\text{proj}}(k_\nu-k_{p-q})\psi_{\text{proj}}(k_\nu)\right). \tag{5}$$

It is straightforward to see that this difference vanishes for a projectile wave packet of the form in Eq. (3) in the $\sigma_k \to \infty$ limit. While we are using non-BO dynamics to simulate the stopping process, we aim to design an initial projectile state that remains a good approximation to a point nucleus throughout the subsequent dynamics. This is because we are primarily using non-BO dynamics to avoid the overheads of simulation with an explicitly time-dependent BO Hamiltonian and to allow the electrons to be excited by the nucleus, even while the projectile remains essentially classical in response.

One can think about the difference in Eq. (5) as quantifying a particular bias in the simulation. Ideally, we would be able to bound this difference to relate σ_k to the error that this introduces in the estimate of the stopping power. However, numerical tests suggest that convenient analytical bounds are too loose to be useful and we instead turn to classical TDDFT calculations for guidance. In many plane-wave TDDFT calculations the electrostatic potential of the point-like projectile ($\sigma_r \to 0$) is included in the total Hartree potential, which is itself represented in terms of a plane-wave basis set and (not quite) dual real-space grid. Thus these calculations are themselves subject to a similar source of error in so far as the real-space grid only approximately captures the point-like nature of the projectile, even in the absence of pseudization. We expect that setting σ_r to be less than or equal to the real-space grid spacing ensuring a particular degree of convergence in TDDFT forces (typically an order of magnitude or more below the target precision in the stopping power estimate) will introduce a bias that is consistent with the degree of convergence.

We estimate the size of this bias from TDDFT calculations implemented using an extension of the Vienna *ab initio* simulation package (VASP) [11–13] described and applied in Refs. [14–17]. Proton stopping power calculations for a deuterium plasma at a density of 10 g/cm^3 and temperature of 1 eV were analyzed for plane-wave cutoffs ranging from 500 eV to 5000 eV and a velocity of 1 a.u. A cutoff of 1000 eV suffices to converge the estimated stopping power to within 7×10^{-3} a.u. of the 5000 eV calculation, and the former cutoff corresponds to a real-space grid spacing of $1.75\times10^{-1}\,a_0$. Thus we expect that $\sigma_k \approx 5.7$ a.u. suffices to achieve a comparable bias in the force. A plane-wave cutoff of 4000 eV corresponds to $\sigma_k \approx 11.4$ a.u. and reduces this estimated bias by almost two orders of magnitude, and thus we expect $\sigma_k = 5-10$ a.u. to be a good rule of thumb for low-Z projectiles/targets and densities between 1 and 10 g/cm^3, relevant to WDM conditions that occur in ICF targets on their way to ignition. We note that we expect these estimates to be somewhat pessimistic because the supporting TDDFT calculations with different cutoffs each start from a different initial electronic state and contain additional convergence errors from different discretizations of the electronic system.

III. GRID RESOLUTION FOR THE PROJECTILE WAVE PACKET

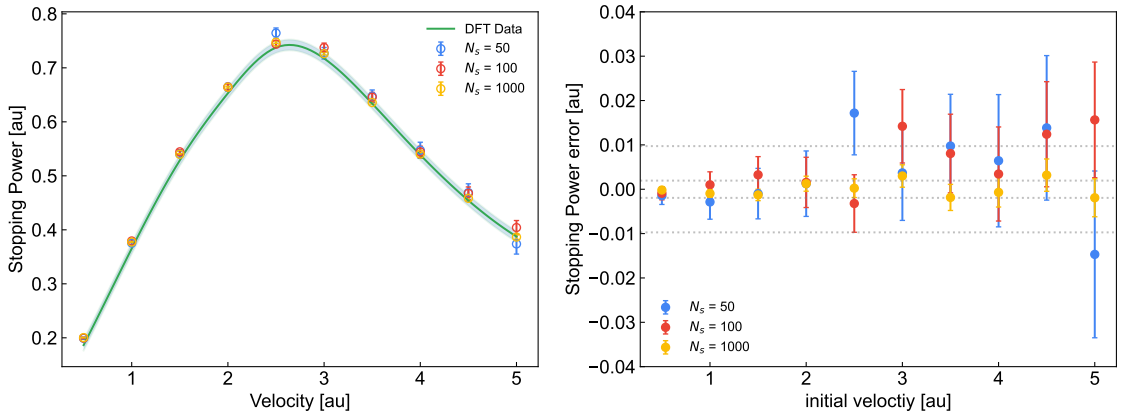

Figure 1. (Left) Comparison between the TDDFT stopping power and the stopping power computed through random sampling of the expected value of a Gaussian wave packet's kinetic energy. The shaded area represents an error of 0.01 a.u. demonstrating that the main features of the stopping curve can be resolved with a higher error threshold. (Right) Error in stopping power as a function of the projectile's initial velocity and number of samples (N_s) used to estimate the Gaussian wave packet's kinetic energy. Horizontal lines represent an accuracy in the stopping power of 0.1 eV/Å ($\approx$ 0.002 a.u.) and 0.5 eV/Å ($\approx$ 0.01 a.u.).

Given a value for σ_k, we next need to determine the number of plane waves required to resolve the kinetic energy of the projectile to within a specified accuracy. The larger the value of σ_k, the more plane waves will be required. We can numerically estimate the number of plane waves by computing the kinetic energy of the wave packet as a function of the kinetic energy cutoff. Taking our initial wave packet as

$$|\psi_{\text{proj}}\rangle = \frac{1}{\sqrt{\mathcal{N}}} \sum_{p\in\tilde{G}} e^{-\frac{\|k_p\|^2}{4\sigma_k^2}} |p\rangle_{\text{proj}}, \tag{6}$$

where the normalization factor is $\mathcal{N} = \sum_{p\in\tilde{G}} e^{-\frac{\|k_p\|^2}{2\sigma_k^2}}$, then the kinetic energy of the projectile is given by

$$\langle T_{\text{proj}}\rangle = \frac{1}{\mathcal{N}} \sum_{p\in\tilde{G}} \frac{\|k_p - k_{\text{proj}}\|^2}{2M_{\text{proj}}} e^{-\frac{\|k_p\|^2}{2\sigma_k^2}}. \tag{7}$$

Another concern with representing a continuous Gaussian wave packet on a grid is the discretization error which should vanish as $\Omega \to \infty$. We can monitor both of these convergence issues by computing the error in the kinetic energy

$$\epsilon_T = \frac{1}{\mathcal{N}} \sum_{p\in\tilde{G}} \frac{\|k_p\|^2}{2M_{\text{proj}}} e^{-\frac{\|k_p\|^2}{2\sigma_k^2}} - \frac{1}{\mathcal{N}_\infty} \int dk^3 \frac{\|k\|^2}{2M_{\text{proj}}} e^{-\frac{\|k\|^2}{2\sigma_k^2}}, \tag{8}$$

where the $\|k_{\text{proj}}\|^2$ and $k_p \cdot k_{\text{proj}}$ terms contributing to the $\|k - k_{\text{proj}}\|^2/(2M_{\text{proj}})$ factors cancel and vanish, respectively, and $\mathcal{N}_\infty = (2\pi)^{3/2}\sigma_k^3$. The integral expression comprising the second term is proportional to the second moment of the normal distribution and evaluates to $3\sigma_k^2/(2M_{\text{proj}})$.

For the box sizes considered here ($\Omega^{1/3} \approx 15\, a_0$) and $\sigma_k > 1$ a.u. as required by Section II, we find that the number of plane waves required to achieve a low ϵ_T can substantially exceed the number that suffices for convergence in classical TDDFT simulations. Fig. 2 shows that obtaining a kinetic energy error below 10^{-3} Ha with $\sigma_k = 10$ a.u. for a proton projectile requires a plane-wave cutoff of approximately 10^5 eV, 100 times greater than the TDDFT cutoff described in Section II. This cutoff corresponds to $N \approx 1.7 \times 10^7$ plane waves and would require $n_n = 9$ bits to store each component of the projectile's momenta. In Table I we list the resources required for different choices of σ_k.

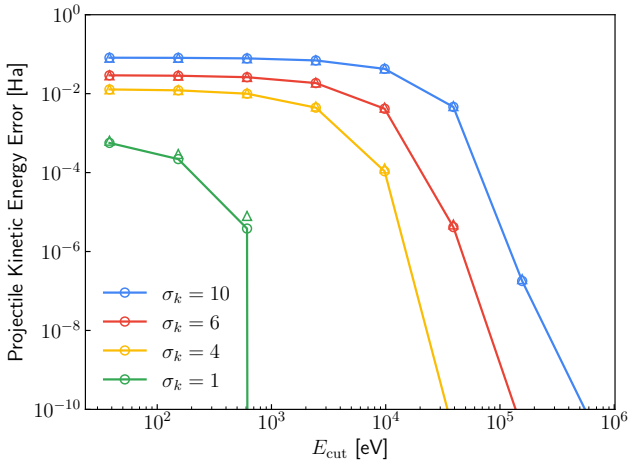

Figure 2. Kinetic energy convergence for a proton projectile with respect to the plane-wave cutoff for different values of the Gaussian wave packet standard deviation σ_k in a.u. Circles indicate errors evaluated according to Eq. (8). Triangles indicate errors calculated by replacing the sum in Eq. (8) with a truncated integral, demonstrating that the discretization error for the values of σ_k chosen in this work is very small.

σ_k [a.u.]	E_{cut} [eV]	N_n	n_n	ϵ_T [Ha]
1	3.8×10^{1}	6.4×10^{1}	3	5.6×10^{-4}
4	9.8×10^{3}	2.6×10^{5}	7	1.1×10^{-4}
6	3.9×10^{4}	2.1×10^{6}	8	4.2×10^{-6}
10	1.6×10^{5}	1.7×10^{7}	9	1.8×10^{-7}

Table I. Number of plane waves N_n and qubits n_n needed to converge the kinetic energy of the projectile wave packet (in 3D) to $\epsilon_T < 10^{-3}$ Ha for different values of the standard deviation σ_k and cutoff energy E_{cut}. Here we assumed a simulation volume of $\Omega^{1/3} = 15\, a_0$.

IV. ACCOUNTING FOR A LARGER NUMBER OF PLANE WAVES FOR THE NUCLEUS

We allow a different (larger) set of momenta for the projectile than for the electron. The complete Hamiltonian is therefore slightly modified from that in [18] to

$$H = T_{\text{elec}} + T_{\text{proj}} + U_{\text{elec}} + U_{\text{proj}} + V_{\text{elec}} + V_{\text{elec-proj}} \tag{9}$$

$$T_{\text{elec}} = \sum_{i=1}^{\eta} \sum_{p \in G} \frac{\|k_p\|^2}{2} |p\rangle\langle p|_i \tag{10}$$

$$T_{\text{proj}} = \sum_{p \in \tilde{G}} \frac{\|k_p - k_{\text{proj}}\|^2}{2\, M_{\text{proj}}} |p\rangle\langle p|_{\text{proj}} \tag{11}$$

$$U_{\text{elec}} = -\frac{4\pi}{\Omega} \sum_{\ell=1}^{L} \sum_{i=1}^{\eta} \sum_{\substack{p,q \in G \\ p \neq q}} \left(\zeta_\ell \frac{e^{ik_{q-p}\cdot R_\ell}}{\|k_{q-p}\|^2} \right) |p\rangle\langle q|_i \tag{12}$$

$$U_{\text{proj}} = \frac{4\pi}{\Omega} \sum_{\ell=1}^{L} \sum_{\substack{p,q \in \tilde{G} \\ p \neq q}} \left(\zeta_\ell \zeta_{\text{proj}} \frac{e^{ik_{q-p}\cdot R_\ell}}{\|k_{p-q}\|^2} \right) |p\rangle\langle q|_{\text{proj}} \tag{13}$$

$$V_{\text{elec}} = \frac{2\pi}{\Omega} \sum_{i \neq j}^{\eta} \sum_{p,q \in G} \sum_{\substack{\nu \in G_0 \\ (p+\nu) \in G \\ (p-\nu) \in G}} \frac{1}{\|k_\nu\|^2} |p+\nu\rangle\langle p|_i |q-\nu\rangle\langle q|_j \tag{14}$$

$$V_{\text{elec-proj}} = -\frac{4\pi}{\Omega} \sum_{i=1}^{\eta} \sum_{\substack{p \in G \\ q \in \tilde{G}}} \sum_{\substack{\nu \in G_0 \\ (p+\nu) \in G \\ (q-\nu) \in \tilde{G}}} \frac{\zeta_{\text{proj}}}{\|k_\nu\|^2} |p+\nu\rangle\langle p|_i |q-\nu\rangle\langle q|_{\text{proj}} \tag{15}$$

where the subscript proj is used to indicate quantities for the projectile nucleus treated quantum mechanically and ℓ indexes the nuclei treated within the BO approximation. The set $\tilde{G}$ is now the momenta for the projectile, U_{proj} is used for the potential energy between the single projectile treated quantum mechanically and the other nuclei, and $V_{\text{elec-proj}}$ is used for the potential energy between the projectile and the electrons.

There will need to be a different preparation of a $1/\|\nu\|$ state for U_{proj} than for the other potential operators, because it will need differences over the full range of projectile momentum. For V_{proj} we need to check that $(q-\nu) \in \tilde{G}$, which means that the projectile momentum has not been shifted outside the range of its allowed values. The sum over ν here is still over G_0, because that includes all allowable shifts of momenta for electrons. Furthermore, we still have the condition that $(p+\nu) \in G$ for the electron momentum to not be shifted outside the allowable range. The projectile kinetic energy T_{proj} differs from that for electrons in that it has division by M_{proj} for the mass of the projectile (with units chosen such that the electron mass is 1), as well as the sum over $\tilde{G}$. The terms U_{proj} and V_{proj} have factors for the charge of the projectile, ζ_{proj}.

To improve the efficiency, we consider the case where the projectile momentum is centred around some offset. Then instead of $\|k_p\|^2$, we would have $\|k_p - k_{\text{proj}}\|^2$. We need to add $\sum_{w \in \{x,y,z\}} [(k^w_{\text{proj}})^2 - 2k^w_p k^w_{\text{proj}}]$ where the superscript w is indicating the Euler direction. Since $(k^x_{\text{proj}})^2$ is a classically chosen number, it just gives an undetectable global phase shift which can be ignored, and we just need $-2k^w_p k^w_{\text{proj}}$ (the w-components of k_p multiplied by a constant). This means that we are effectively adding an extra term to the Hamiltonian

$$T_{\text{mean}} = -\sum_{w \in \{x,y,z\}} \sum_{p \in \tilde{G}} \frac{k^w_p k^w_{\text{proj}}}{M_{\text{proj}}} |p\rangle\langle p|_{\text{proj}} \,. \tag{16}$$

We will now analyse the complexity for the block encoding of this Hamiltonian. We will not analyse the interaction picture approach, which is likely to be more costly.

A. Value of λ

First, we will define a new value corresponding to the sum of $1/\|\nu\|^2$ over the wider range as

$$\lambda_\nu^{\rm proj} = \sum_{\nu\in\tilde{G}_0} \frac{1}{\|\nu\|^2}. \tag{17}$$

Here $\tilde{G}_0$ is the equivalent of G_0, except for differences between elements of $\tilde{G}$ for the projectile momentum. The contributions to λ from $U_{\rm elec}$ and $U_{\rm proj}$ are

$$\begin{aligned}\lambda_U^{\rm elec} &= \frac{\eta\lambda_\zeta}{\pi\Omega^{1/3}}\lambda_\nu\,,\\ \lambda_U^{\rm proj} &= \frac{\zeta_{\rm proj}\lambda_\zeta}{\pi\Omega^{1/3}}\lambda_\nu^{\rm proj}\,,\end{aligned} \tag{18}$$

respectively. For $U_{\rm proj}$, we need ν summed over $\tilde{G}_0$ for differences of projectile momentum. The contributions to λ from V and $V_{\rm proj}$ are

$$\begin{aligned}\lambda_V^{\rm elec} &= \frac{\eta(\eta-1)}{2\pi\Omega^{1/3}}\lambda_\nu\,,\\ \lambda_V^{\rm proj} &= \frac{\eta\zeta_{\rm proj}}{\pi\Omega^{1/3}}\lambda_\nu\,,\end{aligned} \tag{19}$$

respectively.

The contribution to λ from the electron component of the kinetic energy is

$$\lambda_T = \frac{6\eta\pi^2}{\Omega^{2/3}}2^{2(n_p-1)}. \tag{20}$$

As discussed in [18], the reason why there is the square of 2^{n_p-1} rather than $2^{n_p-1}-1$ is because there is a simplification in the state preparation for the registers selecting the bits of the momentum. The component of λ for the projectile, but ignoring the mean is

$$\lambda_T^{\rm proj} = \frac{6\pi^2}{M_{\rm proj}\Omega^{2/3}}2^{2(n_n-1)}. \tag{21}$$

The component of λ for the product of the offset and the mean will then be obtained from $k_{\max}k_{\rm proj}^w$ for component w, where the factor of 2 from squaring and the factor of 1/2 for kinetic energy cancel. Because $k = 2\pi p/\Omega^{1/3}$, that gives

$$\frac{2\pi\sum_{w\in\{x,y,z\}}|k_{\rm proj}^w|}{M_{\rm proj}\Omega^{1/3}}2^{n_n-1}. \tag{22}$$

Here we have accounted for the state preparation giving an effective 2^{n_n-1} rather than $2^{n_n-1}-1$. This will also be needed for implementing $k_{\rm proj}^w$ (because it will effectively correspond to all ones classically), so the cost will need to be adjusted by a factor of $2^{n_n-1}/(2^{n_n-1}-1)$. That gives

$$\lambda_T^{\rm mean} = \frac{2\pi\sum_{w\in\{x,y,z\}}|k_{\rm proj}^w|}{M_{\rm proj}\Omega^{1/3}}\frac{2^{2(n_n-1)}}{2^{(n_n-1)}-1}. \tag{23}$$

Now consider the value of λ as given in Eq. (119) of [18], which is

$$\lambda = \max\left[\lambda_T+\lambda_U+\lambda_V, [\lambda_U+\lambda_V/(1-1/\eta)]/p_\nu\right]. \tag{24}$$

The reason for this equation is that, in the case where the inequality test $i\neq j$ fails, or the preparation of the $1/\|\nu\|$ state fails, one can simply apply the kinetic energy component of the Hamiltonian. In the case where that would yield a larger contribution to T than the actual size, that would imply you need to perform an AND with a qubit flagging T, and flag a result of 0 as 'failure' (removing that contribution to the block encoding). When considering the effective λ values with failures of state preparation, it is divided by the probability of success, so we would have

$[\lambda_U + \lambda_V/(1-1/\eta)]/p_\nu$. For further explanation see Ref. [18] or Appendix IV B 3. In the case where it would not yield a sufficient contribution to T, there would need to be application of T based on an OR with a qubit flagging T. That would imply that $\lambda_T + \lambda_U + \lambda_V$ is the correct value of λ to use.

In our case, the only contribution to the Hamiltonian where we would apply $T_{\rm elec}$ (or $T_{\rm proj}$ or $T_{\rm mean}$) if $i = j$ is if we were otherwise applying $V_{\rm elec}$, which corresponds to $\lambda_V^{\rm elec}$. Then for the preparation of the $1/\|\nu\|$ state, we have a distinct preparation for $U_{\rm proj}$ (corresponding to $\lambda_U^{\rm proj}$) as for the other contributions to the potential energy. We would therefore have $1/(1-1/\eta)$ for $\lambda_V^{\rm elec}$ alone, and $1/p_\nu$ for most potential terms, except $1/p_\nu^{\rm proj}$ for $U_{\rm proj}$. Therefore the new expression for λ is

$$\lambda = \max\left[\lambda_T^{\rm elec} + \lambda_T^{\rm proj} + \lambda_T^{\rm mean} + \lambda_U^{\rm elec} + \lambda_U^{\rm proj} + \lambda_V^{\rm elec} + \lambda_V^{\rm proj}, [\lambda_U^{\rm elec} + \lambda_V^{\rm elec}/(1-1/\eta) + \lambda_V^{\rm proj}]/p_\nu + \lambda_U^{\rm proj}/p_{\nu,\rm proj}\right]. \tag{25}$$

This expression will be discussed in more detail below where we analyse the state preparation. In the case where amplitude amplification is used for the $1/\|\nu\|$ state preparation, there will be a similar expression with p_ν and $p_{\nu,\rm proj}$ replaced with the corresponding probabilities with the amplitude amplification.

B. Preparation cost

We now need to have a separate superposition over ν prepared for $U_{\rm proj}$ than for all other potential terms, and there will need to be a different preparation over the bits of T for the projectile and electron momenta. We will also need to adjust the preparation of the registers for selecting between the different terms in the Hamiltonian.

1. *Preparation of ν state*

First note that the most difficult part of the preparation is that we need different superpositions over ν depending on whether we have $U_{\rm proj}$ or any other part of the Hamiltonian. Referring to Eq. (77) of [18], the first step in the preparation via nested boxes is to prepare a state of the form

$$\frac{1}{\sqrt{2^{n_p+2}}}\sum_{\mu=2}^{n_p+1}\sqrt{2^\mu}\,|\mu\rangle\,, \tag{26}$$

where $|\mu\rangle$ is encoded in unary. In our case we will need the equivalent state except with n_p replaced with n_n for the case of $U_{\rm proj}$. Because the state is prepared by a sequence of controlled Hadamards, one can control between preparing the two states by making $n_n - n_p$ of the Hadamards also controlled by the qubit selecting $U_{\rm proj}$. This just increases the cost of the controlled Hadamards by 1 each for an extra cost of $n_n - n_p$ Toffolis.

The useful feature of this approach is that *no* further amendment to the preparation scheme is needed to make it controlled. The rest of the state preparation for ν can proceed exactly as before, with the only extra Toffoli cost being $n_n - n_p$ at the beginning for preparing the nested boxes state.

To explain the controlled preparation scheme in more detail, the unary encoding is of the form

$$\frac{1}{\sqrt{2^{n_p+2}}}\sum_{\mu=2}^{n_p+1}\sqrt{2^\mu}\,|\mu\rangle = \frac{1}{\sqrt{2^{n_p+2}}}\sum_{\mu=2}^{n_p+1}\sqrt{2^\mu}\,|0\cdots0\underbrace{1\cdots1}_{\mu}\rangle\,. \tag{27}$$

The unary basis state corresponding to $\mu = n_p + 1$ corresponds to $|1\cdots1\rangle$. The start of the state preparation is to perform a Hadamard on the first qubit, then use that to control a Hadamard on the second qubit, and so forth. At the end we would perform a controlled Hadamard on the second-last qubit. This would give an equal superposition between $\mu = 2$ and $\mu = 1$, but because we do not allow $\mu < 2$, the case $\mu = 1$ would be flagged as a failure. The final qubit depicted here can be omitted, because it would always be 1 in this encoding. There will be n_p qubits, and $n_p - 1$ controlled Hadamards.

In our case here, we would want to either prepare this state or

$$\frac{1}{\sqrt{2^{n_n+2}}}\sum_{\mu=2}^{n_n+1}\sqrt{2^\mu}\,|0\cdots0\underbrace{1\cdots1}_{\mu}\rangle\,, \tag{28}$$

where n_p has been replaced with n_n. Because these states need to be represented on the same qubits, we would have $n_n - n_p$ leading zeros when preparing the first state. Therefore, for the first Hadamard on the first qubit, it would

need to be controlled on the qubit selecting between the two states. Then for the next qubit, provided $n_n - n_p > 1$, we would perform a doubly-controlled Hadamard. That is, the Hadamard on the second qubit would be controlled by the first qubit and the qubit selecting between the two states. This will be true for all following qubits that need to be zero for the n_p state. Making the Hadamard on the first of these qubits controlled, and the remaining Hadamards doubly controlled, gives an extra Toffoli complexity of $n_n - n_p$.

For the first qubit that is non-zero for the n_p state, we would need to perform a Hadmard for that state, or a controlled Hadmard for the n_n state. This selection does not require any further non-Clifford gates. One can simply use the qubit selecting the n_p state as the control for a CNOT on the preceding qubit. That ensures it is 1 for the n_p state. Then perform the controlled Hadamard as before. For the n_n case this is just part of the sequence of controlled Hadamards, but for n_p it is ensuring the Hadamard is performed on the qubit. Then just perform another CNOT to erase the preceding qubit. Then the sequence of controlled Hadamards can proceed in the same way as when not preparing this state in a controlled way. By this procedure one can control between preparing the state with n_p or n_n with an extra Toffoli cost of only $n_n - n_p$.

2. *Preparation of momentum control qubits*

Next consider how to prepare superpositions over control qubits for the bits of the momentum. This preparation will need to be controlled by a qubit selecting between the electron and projectile momentum. The preparation is described in Eqs. (67) to (69) of [18], and again it proceeds by a sequence of Hadamard gates, except this time it needs to be done twice for two states (giving r and s). We can make this controlled in exactly the same way as for the preparation of ν. The only difference is that this time there are two states, so the extra Toffoli cost is $2(n_n - n_p)$.

We will also need a preparation for a state selecting between the components of $k_{\rm proj}$ for the product $k_p^w k_{\rm proj}^w$. In practice, the direction of $k_{\rm proj}$ does not need to be taken to be very precise, and we can just consider a rounded direction. We will therefore just use 8 bits for selecting between the components. The exact value chosen has very little effect on the overall cost. The method is to use an 8-qubit equal superposition state (prepared with Hadamards). There are then two inequality tests to prepare the qubits for selecting between x, y, z, which a total cost of 16 Toffolis. These can be inverted with Cliffords for the inverse state preparation provided the temporary qubits are retained.

We will also need to use the qubit selecting the product of the mean momentum and offset to control a swap of these qubits and those that are used for selecting x, y, z for the square of the momentum. That will cost another 4 Toffolis, including 2 for the controlled swap and another 2 for the inversion.

3. *Preparation of state selecting term in Hamiltonian*

In practice we are applying the kinetic component of the Hamiltonian in the case of failure of state preparation for the potential terms. In particular, there are two scenarios, corresponding to whether the first or second expression in Eq. (25) gives the maximum. When the first is larger, this implies that only applying the kinetic term in the case of state preparation failure will not give sufficient weight on that term. You need to apply the kinetic term if there is failure OR a qubit flagging the kinetic term is in the $|1\rangle$ state. That can be computed with a Toffoli. In the case where the second expression in Eq. (25) is larger, that means that applying the kinetic term in the case of preparation failure would give too large a weight on the kinetic term. Then one needs to apply the kinetic term if there is failure AND the qubit flagging the kinetic term is in the $|1\rangle$ state. That logical AND is again something that can be computed with a single Toffoli.

But, unlike in [18] there are three kinetic components to account for. This means that, in the case of failure of the state preparation we also need a register to select between the three kinetic components. To achieve this, we will prepare a state of the form

$$(\sqrt{\alpha_{UV}}\,|0\rangle + \sqrt{\alpha_T}\,|1\rangle)\left(\sqrt{\mu_T^{\rm elec}}\,|0\rangle + \sqrt{\mu_T^{\rm proj}}\,|1\rangle + \sqrt{\mu_T^{\rm mean}}\,|2\rangle\right)\left(\sqrt{\mu_U^{\rm elec}}\,|0\rangle + \sqrt{\mu_U^{\rm proj}}\,|1\rangle + \sqrt{\mu_V^{\rm elec}}\,|2\rangle + \sqrt{\mu_V^{\rm proj}}\,|3\rangle\right), \tag{29}$$

where the first qubit is used to select the kinetic component, the second register is to select between the different kinetic energy components, and the third register is used to select between the potential energy components.

Now, in the case where the first expression in Eq. (25) gives the maximum, we would perform an OR between the result of state preparation and the first qubit, and use the second register to select between the components of T. To describe this state preparation in a simplified way, we will describe it as a rotated qubit flagging success. It will, of course, be entangled with the prepared state, but we are ignoring that for the simplicity of the explanation here. The

state can then be written as

$$(\sqrt{\alpha_{UV}}\,|0\rangle + \sqrt{\alpha_T}\,|1\rangle)\left(\sqrt{\mu_T^{\rm elec}}\,|0\rangle + \sqrt{\mu_T^{\rm proj}}\,|1\rangle + \sqrt{\mu_T^{\rm mean}}\,|2\rangle\right)\left[\left(\sqrt{\mu_U^{\rm elec}}\,|0\rangle + \sqrt{\mu_V^{\rm proj}}\,|3\rangle\right)\left(\sqrt{p_\nu}\,|0\rangle + \sqrt{1-p_\nu}\,|1\rangle\right)\right.$$
$$\left. + \sqrt{\mu_U^{\rm proj}}\,|1\rangle\left(\sqrt{p_{\nu,{\rm proj}}}\,|0\rangle + \sqrt{1-p_{\nu,{\rm proj}}}\,|1\rangle\right) + \sqrt{\mu_V^{\rm elec}}\,|2\rangle\left(\sqrt{(1-1/\eta)p_\nu}\,|0\rangle + \sqrt{1-(1-1/\eta)p_\nu}\,|1\rangle\right)\right]. \tag{30}$$

This corresponds to a probability of p_ν for success with U or $V_{\rm proj}$, since we only need to prepare the $1/\|\nu\|$ state with n_p qubits. Then there is $p_{\nu,{\rm proj}}$ for $U_{\rm proj}$ since there is preparation with n_n qubits for the projectile. Lastly, for V there is $(1-1/\eta)p_\nu$ since we need preparation of the $1/\|\nu\|$ state and $i \neq j$ in preparing the equal superposition state.

To take account of the case where there is amplitude amplification performed for the state preparation for ν, there will be separate boosted probabilities $p_\nu^{\rm amp}$ and $p_{\nu,{\rm proj}}^{\rm amp}$ for the electron and projectile parts. This is because the state preparation and amplitude amplification is performed entirely controlled by the register selecting between electron and projectile components. (A different expression would be obtained if there were amplitude amplification involving the selection between components as well.) We give the reasoning below using the expressions for the un-amplified probabilities, but exactly the same reasoning applies with the amplified probabilities.

The overall squared amplitude for $|0\rangle$ on the ancilla flag qubit is then

$$\mu_{UV} := p_\nu(\mu_U^{\rm elec} + (1-1/\eta)\mu_V^{\rm elec} + \mu_V^{\rm proj}) + p_{\nu,{\rm proj}}\mu_U^{\rm proj}\,. \tag{31}$$

We would *only* apply a potential component of the Hamiltonian if we have $|0\rangle$ on this qubit and $|0\rangle$ on the first qubit, which has a squared amplitude α_{UV}. Therefore the squared amplitude for performing the kinetic component of the Hamiltonian at all is $1-\alpha_{UV}\mu_{UV}$. The squared amplitudes for applying the kinetic components will correspond to this factor times the squared amplitudes in the second register. So, for example, the squared amplitude for $T_{\rm elec}$ is $(1-\alpha_{UV}\mu_{UV})\mu_T^{\rm elec}$.

In the overall block encoding, the block encoding for the Hamiltonian gives H/λ, which is how λ is defined. Here we would have a squared amplitude for $T_{\rm elec}$, then block encode $T_{\rm elec}$ with a factor of $1/\lambda_T^{\rm elec}$. This means that the factor of $1/\lambda$ in the overall block encoding needs to be the same as $(1-\alpha_{UV}\mu_{UV})\mu_T^{\rm elec}/\lambda_T^{\rm elec}$. Solving for $\lambda_T^{\rm elec}$ then gives

$$\lambda_T^{\rm elec} = \lambda\,(1-\alpha_{UV}\mu_{UV})\,\mu_T^{\rm elec}\,. \tag{32}$$

In exactly the same way, for the other kinetic components we obtain

$$\lambda_T^{\rm proj} = \lambda\,(1-\alpha_{UV}\mu_{UV})\,\mu_T^{\rm proj}\,, \qquad \lambda_T^{\rm mean} = \lambda\,(1-\alpha_{UV}\mu_{UV})\,\mu_T^{\rm mean}\,. \tag{33}$$

Then, for the potential components, we need to multiply the squared amplitude for that component in the third register by the squared amplitudes for $|0\rangle$ on the first register and flag qubit. For example, for $U_{\rm elec}$ we have

$$\lambda_U^{\rm elec} = \lambda\,\alpha_{UV}p_\nu\mu_U^{\rm elec}\,, \tag{34}$$

where the factor of λ arises from exactly the same reasoning as for the kinetic components. Similarly, for the other potential components we have

$$\lambda_U^{\rm proj} = \lambda\,\alpha_{UV}p_{\nu,{\rm proj}}\mu_U^{\rm proj}\,, \qquad \lambda_V^{\rm elec} = \lambda\,\alpha_{UV}p_\nu(1-1/\eta)\mu_V^{\rm elec}\,, \qquad \lambda_V^{\rm proj} = \lambda\,\alpha_{UV}p_\nu\mu_V^{\rm proj}\,. \tag{35}$$

Next, note that when we sum $\lambda_T^{\rm elec}, \lambda_U^{\rm elec}, \lambda_V^{\rm elec}, \lambda_T^{\rm proj}, \lambda_U^{\rm proj}, \lambda_V^{\rm proj}, \lambda_T^{\rm mean}$, we just get λ. That is what is expected, because there is no case here where no component of the Hamiltonian is implemented. This means that the individual λ values should correspond to sums of weights in linear combinations of unitaries for the components of the Hamiltonian, with their sum corresponding to the sum of weights for the complete Hamiltonian. This is also consistent with the first expression in Eq. (25).

Let us prepare the registers using inequality tests using numbers of qubits n_{VT}, n_T, and n_{UV}, for the respective registers. There will then need to be six inequality tests for the preparation. First, the possible error in α_T or α_{UV} will be $\lambda/2^{n_{VT}+1}$. The total contribution to the error in $\lambda_T^{\rm elec}, \lambda_T^{\rm proj}, \lambda_T^{\rm mean}$ will come from the sum of that in the expressions for those three quantities. The contribution to the error should be no more than

$$\frac{\lambda}{2^{n_{VT}+1}}\mu_{UV}\,. \tag{36}$$

We get this exact same expression if we sum the possible contributions to the error from $\lambda_U^{\rm elec}, \lambda_U^{\rm proj}, \lambda_V, \lambda_V^{\rm proj}$. We therefore find that the contribution to the error from this source (preparation of the first qubit) is no larger than

$$\frac{\lambda}{2^{n_{VT}}}\mu_{UV} \leq \frac{\lambda}{2^{n_{VT}}}\,. \tag{37}$$

Next, consider the contribution to the error from the imprecision in the inequality tests for the second register. The error in $\mu_T^{\rm elec}, \mu_T^{\rm proj}, \mu_T^{\rm mean}$ can be $1/2^{n_T+1}$ for two, and $1/2^{n_T}$ for the third, for a total of $2/2^{n_T}$. This error is multiplied by $\lambda(1-\alpha_{UV}\mu_{UV})$, to give

$$\frac{2\lambda}{2^{n_T}}\left(1-\alpha_{UV}\mu_{UV}\right)\,. \tag{38}$$

Then, the contributions to the error from the inequality tests for the third register are $1/2^{n_{UV}+1}$ for two, and $1/2^{n_{UV}}$ for two. The total contribution to the error will then be smaller than

$$\frac{\lambda}{2^{n_{UV}}}\alpha_{UV}\mu_{UV}\,. \tag{39}$$

Now, if we take $n_{UV} = n_T - 1$, then the total of the error from the two sources (preparation of the second and third registers) is upper bounded by

$$\frac{2\lambda}{2^{n_T}}. \tag{40}$$

If we take $n_{VT} = n_T$, then the total error from the three sources is upper bounded as

$$\frac{3\lambda}{2^{n_T}}. \tag{41}$$

The other case is that when the second expression in Eq. (25) gives the maximum. That corresponds to the failure cases of the state preparation giving a weighting that is too large for T, so an AND is performed with the qubit flagging T to reduce the weight to the correct value. In that case, the same three-register state is used, and the result of the state preparation success flag state is the same. But, we apply kinetic components only if there is a failure of the state preparation (which happens with squared amplitude $1-\mu_{UV}$, and there is $|1\rangle$ on the first qubit, with squared amplitude α_T. By exactly the same reasoning as before, we obtain the individual λ-values as the squared amplitudes multiplied by λ. That gives the kinetic λ-values as

$$\lambda_T^{\rm elec} = \lambda\,\alpha_T\,(1-\mu_{UV})\,\mu_T^{\rm elec}\,, \qquad \lambda_T^{\rm proj} = \lambda\,\alpha_T\,(1-\mu_{UV})\,\mu_T^{\rm proj}\,, \qquad \lambda_T^{\rm mean} = \lambda\,\alpha_T\,(1-\mu_{UV})\,\mu_T^{\rm mean}\,. \tag{42}$$

For the potential energy components, we apply them *whenever* there is success of the state preparation, regardless of the state of the first qubit. This means that the λ-values are

$$\lambda_U^{\rm elec} = \lambda\,p_\nu\mu_U^{\rm elec}\,, \qquad \lambda_U^{\rm proj} = \lambda\,p_{\nu,{\rm proj}}\mu_U^{\rm proj}\,, \qquad \lambda_V^{\rm elec} = \lambda\,p_\nu(1-1/\eta)\mu_V^{\rm elec}\,, \qquad \lambda_V^{\rm proj} = \lambda\,p_\nu\mu_V^{\rm proj}\,. \tag{43}$$

It is then easy to check that

$$[\lambda_U^{\rm elec} + \lambda_V^{\rm elec}/(1-1/\eta) + \lambda_V^{\rm proj}]/p_\nu + \lambda_U^{\rm proj}/p_{\nu,{\rm proj}} = \lambda\,, \tag{44}$$

so the second expression in Eq. (25) gives λ as expected. Note that this is the expression without amplitude amplification for ν. When amplitude amplification is used the probabilities p_ν and $p_{\nu,{\rm proj}}$ are replaced with the amplified probabilities $p_\nu^{\rm amp}$ and $p_{\nu,{\rm proj}}^{\rm amp}$.

The error in α_T will only affect $\lambda_T^{\rm elec}, \lambda_T^{\rm proj}, \lambda_T^{\rm mean}$, because the implementation of the potential energy components is independent of the first qubit. Given that the error in α_T is upper bounded as $1/2^{n_{VT}+1}$, the contribution to the error in the three kinetic terms is upper bounded as

$$\frac{\lambda}{2^{n_{VT}+1}}\left(1-\mu_{UV}\right)\,. \tag{45}$$

Next consider the contribution to the error from the preparation of the second register. The total error in $\mu_T^{\rm elec}, \mu_T^{\rm proj}, \mu_T^{\rm mean}$ can be upper bounded as $2/2^{n_T}$, to give an upper bound on the contribution to the error in $\lambda_T^{\rm elec}, \lambda_T^{\rm proj}, \lambda_T^{\rm mean}$ as

$$\frac{2\lambda}{2^{n_T}}\alpha_T\left(1-\mu_{UV}\right)\,. \tag{46}$$

The contributions to the error from the inequality tests for the third register will then be

$$\frac{\lambda}{2^{n_{UV}}}[p_\nu(\mu_U^{\rm elec}+(1-1/\eta)\mu_V^{\rm elec}+\mu_V^{\rm proj})+p_{\nu,{\rm proj}}\mu_U^{\rm proj}]=\frac{\lambda}{2^{n_{UV}}}\mu_{UV}\,. \tag{47}$$

The only difference from the first case (using an AND) is that we have removed the factor of α_{UV}. That is because the potential energy component does not depend on the first qubit. Provided we take $n_{UV}=n_{VT}+1$, adding this to the contribution to the error from α_T gives

$$\frac{\lambda}{2^{n_{UV}}}. \tag{48}$$

If we take $n_{UV}=n_T$, then the total error from the three sources (preparation on the three registers) is again upper bounded as

$$\frac{3\lambda}{2^{n_T}}. \tag{49}$$

When performing the inequality tests for the state preparation on each register, we naturally obtain a result encoded in unary. This is we will have one alternative where none of the inequalities are satisfied, giving 000, another with one satisfied to give 001, another with two satisfied giving 011, and so forth. The encoding of the registers in unary is convenient because we have separate qubits flagging each of the component of the Hamiltonian (after converting to one-hot unary).

In our implementation we will need a qubit selecting between electron and projectile components (so, for example, between $V_{\rm elec}$ and $V_{\rm proj}$). It is trivial to prepare such qubits separately for the kinetic and potential registers without Toffolis. But, we will also need to select between these qubits based on a qubit we prepare (discussed further below) selecting between the kinetic and potential components. That can be performed via a single controlled swap, with another controlled swap in the inverse preparation for a total of two Toffolis.

C. Selection cost

For the storage of the state, a larger number of qubits $3n_n$ are used for the projectile, but the electron momenta are stored in the same way as in [18]. The projectile state is stored in a given location, with the antisymmetrization only for the electron registers. Then, when we are performing the controlled swap of the momentum register to an ancilla, there will be two, one controlled by i and the other by j. For the second we only swap electron momenta into the ancilla, so the procedure is identical to that in [18]. For the first, we also control on a qubit for selecting the projectile momentum, and the ancilla will include extra qubits to allow for storage of the projectile momentum. The controlled swaps can just be performed as before, except in the case that the qubit flagging the momentum component is set we swap the n_n qubits for the projectile momentum. We are performing operations on the ancilla register in superposition (over the registers selecting the electron register or projectile register). The qubits in this ancilla will all be zeroed, so when we swap in an electron momentum the extra qubits that would be used in the case of the projectile momentum are all zero. This means it is possible to perform most operations on the momentum register in common between projectile and electron momenta.

Next we consider the cost. In our implementation, we use i,j to index electron registers, with an extra qubit to select between electron and projectile components. We will perform the controlled swaps of the projectile momentum into only *one* of the temporary registers. This means the cost of the controlled swaps is $2(\eta-2)$ for the unary iteration for the electron registers, then $2[(\eta+1)-2]$ for iteration over the electron and projectile registers. That gives a total of $4\eta-6$. Then cost of the controlled swaps themselves is

$$12\eta n_p+6n_n. \tag{50}$$

This is just an extra cost of $6n_n$ for the projectile momentum, for two controlled swaps on $3n_n$ qubits. That gives a cost

$$12\eta n_p+6n_n+4\eta-6. \tag{51}$$

The other main change to the select operations is that the operations need to be on n_n qubits rather than n_p. This will impact the selection cost of the components of the Hamiltonian in different ways, detailed in the next subsection. There also needs to be selection between the square of the momentum and the product of the momentum with the momentum offset for the kinetic energy.

To see how to modify the procedure to perform this selection, recall how the kinetic energy is computed as in Eq. (65) of [18]. There the kinetic energy is written as

$$T = \frac{2\pi^2}{\Omega^{2/3}} \sum_{j=1}^{\eta} \sum_{p\in G} \sum_{w\in\{x,y,z\}} \sum_{r=0}^{n_p-2} \sum_{s=0}^{n_p-2} 2^{r+s} p_{w,r} p_{w,s} \left|p\right\rangle\!\left\langle p\right|_j \ . \tag{52}$$

Here the two bits $p_{w,r}p_{w,s}$ correspond to products of bits in a component of p in order to obtain the square. The method is very simply modified to obtain the product of the form $k_p^x k_{\text{mean}}^x$, simply by selecting w as x, and replacing bit $p_{x,s}$ with the corresponding bit for k_{mean}^x. The way we encode k_{mean}^x is that it is represented by all bits equal to 1, with its actual value (the multiplying factor times that integer) being governed by the state preparation.

In particular, we need to modify step 4 in the list of steps in the left column of page 17 of [18]. This step involves performing a Toffoli controlled by the qubits storing $p_{w,r}$ and $p_{w,s}$. Here we would make the NOT controlled on $p_{w,r} \wedge (p_{w,s} \vee b)$, for b the flag for the T_{mean} component of the Hamiltonian.

There is also the selection of the component of the momentum depending on whether we want the square or product with the mean. As explained above, there is a cost of 4 Toffolis. The register to select the x, y, z component can be given in binary or unary, since we can convert between binary and unary with Cliffords for three. We can therefore assume that the w registers for the square and product are given using 2 qubits each (for binary). The qubit selecting T_{mean} can be used to control a swap between these two registers, with a cost of 2 Toffolis, then there are another 2 Toffolis to invert the swap for the inverse state preparation.

There is no extra non-Clifford cost for the minus sign in T_{mean}, because it is just a (Clifford) controlled phase gate. We may also perform controlled phase gates to apply signs of the components of k_{mean} with no extra Toffoli cost. The condition that $(p+\nu) \in G$ versus $(q-\nu) \in \tilde{G}$ needs no modification. The operation $p+\nu$ is for an electron momentum, so $(p+\nu) \notin G$ the extra $n_n - n_p$ qubits will not be all zero. Then those qubits are not swapped back into the momentum register in the controlled swap, so nonzero qubits are remaining to flag a 'fail' and remove that part in the block encoding. Similarly, if $(q-\nu) \notin \tilde{G}$ for the projectile momentum, then there will be an extra ancilla qubit flagging 'fail' resulting from the subtraction. There will be no need to treat these registers differently apart from the controlled swaps addressed above.

D. Total costs

To be consistent with Reference [18] we provide updated costs for each block encoding subroutine.

C1. The cost of the preparation of the registers selecting between the components of the Hamiltonian is changed to $6n_T - 1$, where n_T is the number of qubits used in inequality tests. That is the cost for the six inequality tests in the preparation of the three registers used, and accounting for one of those inequality tests using one fewer qubit (see SI I). We are also assuming that the temporary ancillas are retained so these inequality tests may be inverted without Toffolis. There will be another Toffoli for the AND or OR in the preparation of the qubit flagging whether the kinetic or potential energy is applied. There are another two Toffolis for controlled swaps for selecting the qubit that selects between electron and projectile components. That gives a total of $6n_T + 2$.

C2. The cost of preparing the superposition over η values of i, j is unchanged from that in Ref. [18]. There it is given as $14n_\eta + 8b_r - 36$ (where $n_\eta = \lceil \log(\eta) \rceil$) for preparing a superposition and flagging $i \neq j$.

C3. The state preparation of the w, r, s registers for the kinetic energy operators will have cost $2(2n_n + 9) + 2(n_n - n_p) + 20$. The first amendment over the $2(2n_p + 9)$ expression in Ref. [18] is to replace n_p with n_n because we need to account for the larger basis for the projectile. The second amendment is to include a $2(n_n - n_p)$ term to account for the extra control of the Hadamards between the electron and projectile parts. The third amendment is to include the cost of the inequality test for preparing a second w register (20 Toffolis) and the controlled swaps (4 Toffolis).

C4. The cost of swaps into the working registers is $12\eta n_p + 6n_n + 4\eta - 6$. Because we are now selecting between the η electron registers and the projectile register for i but *not* j, we need another two Toffolis, which changes -8 to -6. The extra $6n_n$ term is for the controlled swaps in the case when we are selecting the projectile register for i. There is a factor of 6 rather than 12 because it is only for i, not i and j as for the electron registers.

C5. The select cost of the kinetic energy terms is increased from $5(n_p - 1) + 2$ to $5(n_n - 1) + 2$ due to the need to account for the larger number of qubits for the projectile. There is another Toffoli for the selection between the square and the product of the momentum with the momentum offset for the projectile, for a total of $5n_n - 2$.

C6. For the cost of preparing the $1/\|\nu\|$ state, we first need to replace n_p with n_n. The other amendment is that we need to introduce a cost of $n_n - n_p + 1$ for the extra controls on the Hadamards. There are $n_n - n_p - 1$ double controls, where the double control may be applied with a Toffoli that can be inverted with Clifford gates. There is also an extra singly-controlled Hadamard, which needs another Toffoli for inversion. That gives the extra cost $(n_n - n_p - 1) + 2 = n_n - n_p + 1$. That gives a total cost of $3n_n^2 + 16n_n - n_p - 6 + 4n_{\mathcal{M}}(n_n - 1)$ for preparing the $1/\|\nu\|$ state.

C7. The QROM cost or R_ℓ is unchanged at $\lambda_\zeta + \mathrm{Er}(\lambda_\zeta)$.

C8. For the addition and subtraction cost of ν into momentum registers, only half of it is n_p replaced with n_n, because we are only allowing the projectile momentum in one temporary register. That gives a Toffoli cost $12(n_n + n_p)$.

C9. For the phasing cost we simply replace n_p with n_n to account for the projectile momentum, to give $6n_n n_R$.

V. COMPLEXITY FOR PRODUCT-FORMULA SIMULATIONS

A. An efficient implementation of the inverse square-root

To give the costing for this procedure more precisely, we first need to compute the sum of squares of the difference between each of electron's xyz-components. This arithmetic has a cost of three squares. According to Lemma 8 of Ref. [18], the complexity is $3n^2 - n - 1$ when each of the three numbers has n bits. The resulting number has no more than $2n+2$ bits. The Toffoli complexity of the QROM to output interpolation parameters is then $4n+2$, using a variable spacing QROM as in Ref. [19]. As an example, say we were performing the variable spacing QROM on 8 qubits. The integer ranges that would be used would be

$$0, 1, 2, 3, [4, 5], [6, 7], [8, 11], [12, 15], [16, 23], [24, 31], [32, 47], [48, 63], [64, 95], [96, 127], [128, 191], [192, 255], \tag{53}$$

where e.g. $[16, 23]$ is used to indicate the range of integers.

Then for the interpolation, we can describe it by the interpolation for the regions $[1, 3/2]$ and $[3/2, 2]$. For example, $[16, 23]$ is 16 times a range within $[1, 3/2]$, so we can use the parameters chosen for $[1, 3/2]$ appropriately scaled for this multiplying factor. For the region $[1, 3/2]$ we can use the interpolation

$$\frac{1}{\sqrt{x}} \approx a_0 - a_1(x-1) + a_2(x-1)^2 - a_3(x-1)^3\,, \tag{54}$$

with

$$a_0 = 0.99994132489119882162, \tag{55}$$
$$a_1 = 0.49609891915903542303, \tag{56}$$
$$a_2 = 0.33261112772430493331, \tag{57}$$
$$a_3 = 0.14876762006038398086. \tag{58}$$

This is followed by a step of Newton-Raphson as

$$y \mapsto \frac{1}{2} y(3 + \delta - y^2 x), \tag{59}$$

where $\delta = 5.1642030908180720584 \times 10^{-9}$. These parameters were found by numerically minimising the maximum relative error over the region. It was then found that the relative error is no more than about 2.5821×10^{-9} within this interval. The constants a_j are appropriately scaled for x in the range $[2^m, (3/2)2^m]$ as $a_0 \mapsto a_0/2^{m/2}$, $a_1 \mapsto a_1/2^{3m/2}$, $a_2 \mapsto a_2/2^{5m/2}$, $a_3 \mapsto a_3/2^{7m/2}$, with $x - 1$ replaced with $x - 2^m$. (The constant δ is unchanged.)

For x in the range $[3/2, 2]$, one can use

$$\frac{1}{\sqrt{x}} \approx a_0 - a_1(x-3/2) + a_2(x-3/2)^2 - a_3(x-3/2)^3\,, \tag{60}$$

with

$$a_0 = 0.81648515205385221995, \tag{61}$$
$$a_1 = 0.27136515484240234115, \tag{62}$$
$$a_2 = 0.12756148214815175348, \tag{63}$$
$$a_3 = 0.044753028579153842218, \tag{64}$$
$$\delta = 3.6279794522852781448 \times 10^{-10}. \tag{65}$$

These parameters were similarly found by a numerical optimisation to reduce the error, and give relative error no more than about 1.8140×10^{-10} in this region.

Writing the polynomial in this way makes it appear as if many powers and multiplications are needed. The computation can be significantly simplified by rewriting it as (for the case of the range $[1, 3/2]$)

$$a_0 - (x-1)\{a_1 - (x-1)[a_2 - a_3(x-1)]\} = a_0 - a_1(x-1) + a_2(x-1)^2 - a_3(x-1)^3. \tag{66}$$

Thus, only three multiplications are needed to compute the polynomial, and the multiplications are the most costly part. We have also written it with a polynomial of $x-1$ rather than x. This requires one subtraction, but reduces the number of bits needed to represent $x-1$ (versus x) by 1, and reduces the number of bits for each of a_0, a_1, a_2 by 1. This subtraction can be performed with Clifford gates because removing a leading 1 from x can be performed with a CNOT gate. We have a similar form for the polynomial in the range $[3/2, 2]$.

It is also found that the initial polynomial interpolation may be given to only 15 bits of precision, and the resulting accuracy of the approximation after the step of Newton-Raphson is still about one part in 10^8. For a rough estimate of the complexity of the arithmetic, we can assume it is performed with no more than 15 bits at this step. As discussed in [19], the complexity of multiplying two real numbers is approximately the square of the number of bits. This is because less significant bits can be omitted in the calculation. If we are using 15 bits for each multiplication in the interpolation here, the complexity of the three multiplications is about $3 \times 15^2 = 675$. This is the dominant cost in the arithmetic, and the subtractions have significantly lower cost. Three subtractions on 15 bits have Toffoli cost about 45 (computing $x-1$ or $x-3/2$ is not included here because it can be performed with Clifford gates).

For the step of Newton-Raphson, we can estimate the complexity of the square of a 15-bit number as 15^2 Toffolis. Then for the other two multiplications, if we aim for, for example, 24 bits of accuracy, then the complexity is 2×24^2. With another 24 Toffolis for the subtraction, the overall complexity is about 2136 (excluding the complexity of the sum of squares and QROM). With the number of bits in each direction being $n = 6$, the sum of squares has complexity 101, and the QROM has complexity 26 Toffolis. Those are trivial complexities compared to the other arithmetic for the inverse square root, and would bring the total to about 2263.

There are a couple of additional considerations for the complexity not discussed in the simplified discussion above. First, for small x the inverse square root is large, so in the multiplication of x by the approximation of $1/\sqrt{x}$ the assumptions in the estimate of the complexity do not hold. In order to avoid needing to use additional bits of precision in the multiplications to account for that, we can instead use bit shifts. First, we strip pairs of leading zeros from x. Since x has $2n+2$ bits, the complexity is $n(n+1)$, which is 42 for $n = 6$.

In this example, one may remove 2, 4, 6, 8, 10, or 12 leading zeros. (There cannot be all zeros which would correspond to two electrons at the same location.) These alternatives would require moving a number of bits which is 14 minus the number of leading zeros. The Toffoli cost corresponds to the number of bits moved, which gives a total of $12 + 10 + 8 + 6 + 4 + 2 = 42$ Toffolis. At the end one would need to shift the approximation of the inverse square root back again. This can be performed on the result of the QROM interpolation before multiplying in the Newton-Raphson iteration. That would have a Toffoli complexity no more than $15n$ bits giving we are computing the QROM interpolation to 15 bits. In the example with $n = 6$ it is 90 Toffolis. These two costs are relatively trivial compared to the overall cost of the step, and would bring it to about 2395 Toffolis. In this cost we have taken the specific example of $n = 6$. The n-dependent cost can be given as

$$2136 + (3n^2 - n - 1) + (4n + 2) + n(n+1) + 15n = 2137 + 4n^2 + 19n\,. \tag{67}$$

A further consideration is the need to uncompute the arithmetic. If we were to compute all the pairwise Coulomb potentials, sum then phase by the sum, we would then need to uncompute the sum. The number of Toffolis makes it infeasible to uncompute by retaining qubits. However, we can phase by each individual pairwise Coulomb potential and uncompute the arithmetic with Clifford gates by retaining about 2000 qubits used in the calculation. This approach would be reasonable given the algorithm is likely to need a large number of logical qubits already. A difficulty then is that a the potential may need to be multiplied by a constant before phasing. If we were to compute the complete Coulomb potential before phasing then that complexity would be trivial, but it will be significant if we need to do it for each pairwise potential.

If we were to use just the standard Lie-Trotter product formula then the length of the time step could be chosen such that no multiplication were needed. Higher-order product formulae would need time steps with irrational ratios, so multiplications would be needed. However, that can be avoided if we instead compute the factor as part of the QROM interpolation and Newton-Raphson. If we are aiming to compute $b/\sqrt{x}$, then we can simply multiply the constants in the QROM interpolation by b, and replace 3 with $2+b^2$ in the Newton-Raphson step. That will give the desired factor with the same Toffoli complexity as before.

B. Estimating number of Trotter steps

According to Theorem 4 of Ref. [20], for a real-space grid Hamiltonian defined for orbital indices $\{j,k,l,m\}$ and spin indices $\{\sigma,\tau\}$ of the form

$$H = \sum_{j,k,\sigma} \tau_{j,k} a^\dagger_{j,\sigma} a_{k,\sigma} + \sum_{l,m,\sigma,\tau} \nu_{l,m} a^\dagger_{l,\sigma} a_{l,\sigma} a^\dagger_{m,\tau} a_{m,\tau} \tag{68}$$

the spectral norm error in a fixed particle manifold for an order-k product formula $S_k(t)$ can be estimated as

$$\left\| S_k(t) - e^{-itH} \right\|_{W_\eta} = \mathcal{O}\left((\|\tau\|_1 + \|\nu\|_{1,[\eta]})^{k-1} \|\tau\|_1 \|\nu\|_{1,[\eta]} \eta\, t^{k+1} \right). \tag{69}$$

Here, $a^\dagger_{j,\sigma}$ and $a_{j,\sigma}$ are creation and annihilation operators, and the norms are defined as

$$\|\tau\|_1 = \max_j \sum_k |\tau_{j,k}| \tag{70}$$

$$\|\nu\|_{1,[\eta]} = \max_j \max_{k_1<\ldots<k_\eta} \left(|v_{j,k_1}| + \ldots + |v_{j,k_\eta}| \right). \tag{71}$$

If the constant of proportionality is ξ, then breaking longer evolution time t into r intervals gives error

$$\approx \xi (\|\tau\|_1 + \|\nu\|_{1,[\eta]})^{k-1} \|\tau\|_1 \|\nu\|_{1,[\eta]} \eta\, t^{k+1}/r^k . \tag{72}$$

In order to provide a simulation to within error ϵ, the number of time steps is then

$$r \approx t^{1+1/k} (\|\tau\|_1 + \|\nu\|_{1,[\eta]})^{1-1/k} (\xi \|\tau\|_1 \|\nu\|_{1,[\eta]} \eta/\epsilon)^{1/k}. \tag{73}$$

Our goal is to numerically determine the constant factor ξ for a high-order product formula. We first provide the constant factors of the norms used in the error scaling Eq. (69) and describe their convergence to asymptotic values. The scaling of the norms is

$$\|\tau\|_1 = \mathcal{O}\left(\frac{N^{2/3}}{\Omega^{2/3}} \right) \tag{74}$$

$$\|\nu\|_{1,[\eta]} = \mathcal{O}\left(\frac{\eta^{2/3} N^{1/3}}{\Omega^{1/3}} \right). \tag{75}$$

To estimate the constant factor for $\|\tau\|_1$, note that $\|\tau\|$ corresponds to the kinetic energy of a single electron. That fact easily follows from considering τ in the basis where it is diagonal, and noting that $\|\tau\|$ is invariant under basis transforms. Therefore $\|\tau\|_1$ can be lower bounded as

$$\|\tau\|_1 \geq \|\tau\| = \max_p \frac{\|k_p\|^2}{2} = \max_p \frac{4\pi^2 \|p\|^2}{2\Omega^{2/3}} = \frac{4\pi^2}{2\Omega^{2/3}} 3[(N^{1/3}-1)/2]^2 \approx \frac{3\pi^2 N^{2/3}}{2\Omega^{2/3}}. \tag{76}$$

This lower bound is an accurate approximation of $\|\tau\|_1$ in the limit of large N; see Appendix IX. The norm $\|\nu\|_{1,[\eta]}$ comes from Eq. (K4) of [18] which uses the the potential operator

$$V = \frac{N^{1/3}}{2\Omega^{1/3}} \sum_{i\neq j} \sum_{p,q} \frac{1}{\|p-q\|} |p\rangle\langle p|_i |q\rangle\langle q|_j . \tag{77}$$

and corresponds to the potential energy for a single electron with the other electrons packed around it as closely as possible. For p, q where there is a unit grid spacing, the volume is $\eta \approx (4/3)\pi R^3$, giving a radius of $R \approx [(3/4)\eta/\pi]^{1/3}$. The potential energy is the integral of $1/r$ over a sphere from 0 to R. That gives

$$\int_0^R 4\pi r^2/r\, dr = 4\pi \int_0^R r\, dr = 2\pi R^2 = 2\pi[(3/4)\eta/\pi]^{2/3}. \tag{78}$$

Thus, $\|\nu\|_{1,[\eta]}$ including constant factors is approximately

$$\|\nu\|_{1,[\eta]} \approx \pi^{1/3}(3/4)^{2/3}\frac{\eta^{2/3}N^{1/3}}{\Omega^{1/3}}. \tag{79}$$

To explore how quickly the norms converge to their asymptotic values (Eq. (79) and Eq. (76)) we plot each norm as a function of basis size and number of particles in Figure 3. We find that $\|\tau\|_1$ converges relatively quickly with respect to the grid spacing $(N^{1/3}/\Omega^{1/3})$ but $\|\nu\|_{1,[\eta]}$ does not converge until approximately 50 particles.

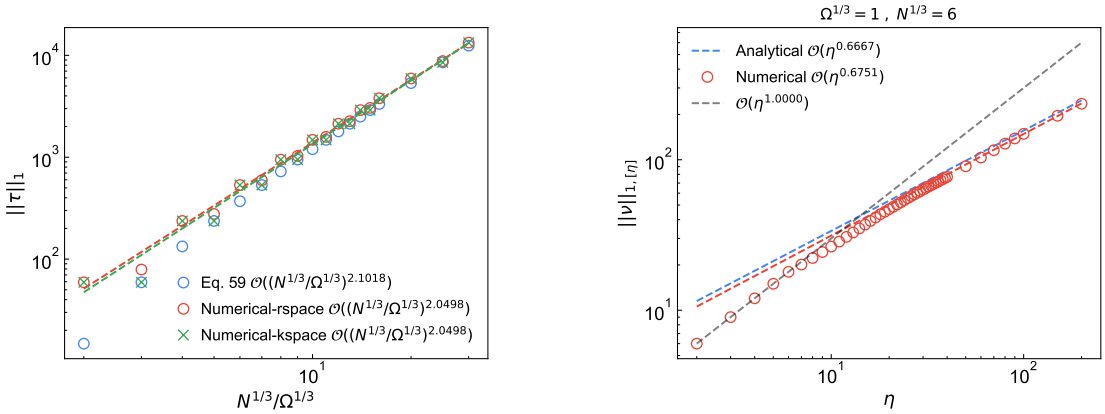

Figure 3. Comparison between the analytical asymptotic value of the the norms $\|\tau\|_1$ and $\|\nu\|_{1,[\eta]}$. For a given reciprocal lattice sampling defined as $k_\nu = \frac{2\pi\nu}{\Omega^{1/3}}$, $\nu \in G$, $G = \{\frac{-N^{1/3}-1}{2}, \frac{N^{1/3}-1}{2}\}$, the real space grid is defined as $r_p = \frac{p\Omega^{1/3}}{N^{1/3}}$, $p \in G$ satisfying shifted (centered) discrete Fourier transform $|r_p\rangle = \frac{1}{\sqrt{N}}\sum_\nu e^{ik_\nu \cdot r_p}|k_\nu\rangle$ and thus $k_\nu \cdot r_p = \frac{2\pi}{N^{1/3}}\nu \cdot p$. The grey dashed line for the $\|\nu\|_{1,[\eta]}$ norm plot describes the scaling of the rightmost four red (numerical) points. The red dashed line corresponds to a fit to the rightmost five red (numerical points). Similarly in the $\|\tau\|_1$ norm plot the N scalings are determined by fitting the rightmost five points.

In order to determine the constant ξ in Eq. (72) we numerically determine the spectral norm of Eq. (69) for a variety of product formulas for a variety of systems scaling in N and η. To avoid building an exponentially large matrix, we adapt the power method to determine the spectral norm of $\Delta(t) = S_k(t) - e^{-itH}$ as the square root of the maximal eigenvalue of $\Delta(-t)\Delta(t)$. Numerically taking the square root would halve the precision, so we instead use the power iteration to estimate the spectral norm of half an application of $\Delta(-t)\Delta(t)$. The full algorithm is outlined in Algorithm 1 which is implemented in the Fermionic Quantum Emulator (FQE) [21]. Using the FQE we can target a particular particle number sector, projected spin s_z sector, and use fast time evolution routines based on the structure of the Hamiltonian in Eq. (68). Our numerics involved 64 orbital (128 qubit) systems involving 2-4 particles.

In the Figure 4 we determined ξ by explicitly calculating the spectral norm of the difference between the exact unitary and a bespoke 8^{th}-order product formula described in Appendix VII. The 'prefactor' variable corresponds to $(\|\tau\|_1 + \|\nu\|_{1,[\eta]})^{k-1}\|\tau\|_1\|\nu\|_{1,[\eta]}\eta\, t^{k+1}$ for $t = 0.65$. For $N = 64$ $\eta = 4$ we estimate a $\xi = 3.4 \times 10^{-8}$ which is the value of the rightmost point.

For the 8^{th}-order product formula, each step requires 17 exponentials. Each exponential has a complexity on the order of $2395\eta(\eta-1)/2$. Combining the constant factors, norm computation, and number of Toffolis required per exponential allowed us to calculate the Toffoli and qubit complexities for time evolution via product formula. We provide comparative costs to QSP in Section **??**.

Algorithm 1: Power iteration algorithm to compute the spectral norm $\|\Delta(t)\|_{\mathcal{W}_\eta} = \Gamma$

Data: ϵ, $\Delta(t) = S_k(t) - e^{-itH}$
$\psi_{i=0} \leftarrow \frac{1}{\sqrt{|\mathcal{H}|}} \sum_j |j\rangle$;
$\delta \leftarrow \infty$;
$\Gamma_- \leftarrow 0$;
$\Gamma \leftarrow 0$;
while $\delta \geq \epsilon$ **do**
  $\psi_{i+\frac{1}{2}} \leftarrow \Delta(t)\psi_i$;
  $\Gamma \leftarrow \|\psi_{i+\frac{1}{2}}\|$;
  $\psi_{i+1} \leftarrow \Delta(-t)\psi_{i+\frac{1}{2}}$;
  $\psi_{i+1} \leftarrow \psi_{i+1}/\|\psi_{i+1}\|$;
  $\delta \leftarrow |\Gamma_- - \Gamma|$;
  $\Gamma_- \leftarrow \Gamma$;
end

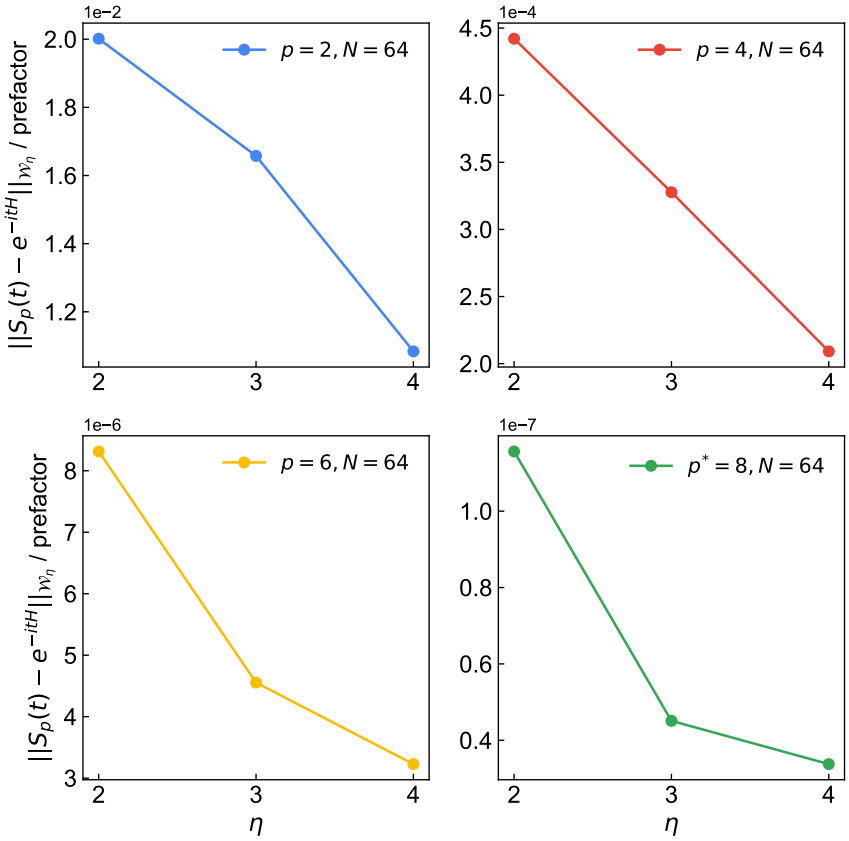

Figure 4. $N = 64$, $\Omega = 5$ grid based Hamiltonian convergence of the prefactor ξ with respect to particle number. $\xi = 3.4 \times 10^{-8}$ is used as an upper bound for the prefactor in all subsequent product formula resource estimates. p indicates the product formula order (star on the lower right plot indicates a numerically determined product formula). The blue points (upper left) are for the Strang product formula, red points (upper right) are for 4^{th}-order Suzuki-Trotter, yellow (lower left) are for 6^{th}-order Suzuki-Trotter, and green (lower right) corresponds to a custom 8^{th}-order formula described in Appendix VII.

VI. CONSTANT FACTOR ESTIMATES FOR PROJECTILE KINETIC ENERGY MEASUREMENT USING THE KNOCKOUT ALGORITHM

For a fixed standard error ϵ, Monte Carlo sampling provides the optimal bound on the number of samples needed to estimate the expected value of an observable. This *standard* sampling limit states the number of samples to estimate the observable to ϵ precision goes as $\mathcal{O}(\sigma^2/\epsilon^2)$ where σ^2 is the variance of the observable. Recently, a quantum algorithm was developed by Kothari and O'Donnell (KO) [22] that allows one to estimate the expected value of an observable to ϵ precision with the number of samples going as $\mathcal{O}(\sigma/\epsilon)$; a quadratic improvement over the standard limit. The main protocol in this algorithm is the use of phase estimation on a unitary that is the composition of a reflection around the prepared state–called the synthesizer–and a phasing operation that phases basis states according the value the random variable takes on those basis states. This protocol allows one to solve a decision problem that identifies if the expected value, encoded as the eigenvalue of the unitary, is to the left or right of a gapped range, thereby knocking out part of the possible range (Theorem 1.3 in Ref. [22]). Using additional classical reductions allows one to boost this decision problem to the mean estimation problem (Theorem 1.1 in Ref. [22]). In this section we describe a rough estimate of the constant factors associated with performing the KO algorithm to estimate the kinetic energy of the projectile. We do not detail all classical reductions necessary for the task but instead focus on the primary decision problem to obtain an estimate of when ϵ is small enough such that the KO algorithm has a computational advantage over Monte Carlo sampling. The unitary that needs to be phase estimated in the KO algorithm is the composition of a reflection analogous to the Grover diffusion operator REFL and a phase oracle ROT_y

$$U = \mathrm{REFL} \cdot \mathrm{ROT}_y. \tag{80}$$

Given a circuit that prepares the desired probability distribution

$$P|0\rangle = \sum_{\ell} \sqrt{p(\ell)}|\ell\rangle \tag{81}$$

the reflection is defined as

$$\mathrm{REFL} = P\left(2|0\rangle\langle 0| - \mathbb{I}\right)P^\dagger \tag{82}$$

and the ROT_y is

$$\mathrm{ROT}_y|\ell\rangle = e^{i\alpha_\ell}|\ell\rangle \tag{83}$$

where $\alpha_\ell = -2\arctan(y_\ell)$ and y_ℓ is the value of the random variable for the event indexed by ℓ. In order to perform phase estimation on the KO algorithm U we need controlled forms of REFL and ROT_y along with controlled forms of their inverses. For a non-abridged version of the algorithm and details surrounding allowed ancilla registers see Section 3 of Ref. [22].

For the synthesizer (P) in REFL we use the time-evolution operator. Therefore, building the reflection operator requires two calls to the state previously described state preparation circuit. For the ROT_y operator we first encode the kinetic energy of the projectile into an ancilla register through a series of multiplications and additions on the ancilla and the projectile register. Second, we calculate the arctan on this register which is linear complexity in the ancilla register size. Finally, controlled phase gates are used to accomplish the correct action defined by the ROT_y unitary. The remaining task to get order of magnitude estimates of the quantum resources required for the KO algorithm is to derive a circuit for writing the random variable value to the ancilla register used to compute α_ℓ.

Recall that the kinetic energy operator on the projectile

$$T = \sum_{p\in\tilde{G}} \frac{\|k_p - k_{\mathrm{proj}}\|^2}{2M_{\mathrm{proj}}}|p\rangle\langle p| \tag{84}$$

where $\|k_p - k_{\mathrm{proj}}\|^2 = \sum_{w\in\{x,y,z\}}\left((k_p^w)^2 + (k_{\mathrm{mean}}^w)^2 - 2k_p^w k_{\mathrm{proj}}^w\right)$ where $\sum_{w\in\{x,y,z\}}(k_{\mathrm{proj}}^w)^2$ is a constant term which we will add with an addition circuit. We can rewrite the coefficients as

$$\frac{\|k_p - k_{\mathrm{proj}}\|^2}{2M_{\mathrm{proj}}} = \frac{1}{2M_{\mathrm{proj}}}\left(\frac{2\pi}{\Omega^{1/3}}\right)^2\left(\boldsymbol{p}^T\boldsymbol{p} - 2\boldsymbol{p}^T\boldsymbol{p}_{\mathrm{proj}} + \boldsymbol{p}_{\mathrm{proj}}^T\boldsymbol{p}_{\mathrm{proj}}\right) \tag{85}$$

where $\boldsymbol{p} = (p_x, p_y, p_z)$. n_{mean} is the number of bits needed to represent the central momentum of the projectile where $n_{\mathrm{mean}} > n_{\mathrm{proj}}$. If we ignore the constant involving the mass and the volume element we are left with a series of integer products and summations. The Toffoli cost of each of the three terms can be derived from protocols described in the Appendix of Ref. [18] and are as follows:

1. The $\boldsymbol{p}^T\boldsymbol{p}$ term involves three n_{proj} registers as thus has $3n_{\text{proj}}^2 - n_{\text{proj}} - 1$ Toffoli complexity.

2. The $\boldsymbol{p}^T\boldsymbol{p}_{\text{proj}}$ involves the sum-product of three integer pairs of sizes $(n_{\text{proj}}, n_{\text{mean}})$. The products each take $2n_{\text{mean}}n_{\text{proj}} - n_{\text{mean}}$ and the three sums involved cost $3n_{\text{mean}}^2 - n_{\text{mean}} - 1$.

3. The final addition requires $\boldsymbol{p}_{\text{mean}}^T\boldsymbol{p}_{\text{proj}}$ stored in $2n_{\text{mean}} - 1$ bits to be added to the results from steps 1 and 2. if we pad out the results from 1 and 2 up to $2n_{\text{mean}} - 1 = n_f$ then we need $3n_f^2 - n_f - 1$ Toffolis for this operation.

4. We leave off the constant to be included in multiplying the variance for the mean-estimation algorithm. In the last step we must subtract an estimate of $\mu_0 = \langle \boldsymbol{p}^T\boldsymbol{p} - 2\boldsymbol{p}^T\boldsymbol{p}_{\text{proj}} + \boldsymbol{p}_{\text{proj}}^T\boldsymbol{p}_{\text{proj}} \rangle$ which we know from classical data. This additional step can be included with the previous step by modifying the subtracted value of $\boldsymbol{p}_{\text{proj}}^T\boldsymbol{p}_{\text{proj}}$ to $\boldsymbol{p}_{\text{proj}}^T\boldsymbol{p}_{\text{proj}} - \mu_0$.

Thus the total Toffoli complexity of performing the integer encoding step is

$$C_{\text{encoding}} = 3n_{\text{proj}}^2 - n_{\text{proj}} - 1 + 3(2n_{\text{mean}}n_{\text{proj}} - n_{\text{mean}}) + 3n_{\text{mean}}^2 - n_{\text{mean}} - 1 + 3n_f^2 - n_f - 1 \tag{86}$$

These costs combined with the costs associated with two calls to the synthesizer, reflection, and computing the arctan are combined to produce a total Toffoli complexity.

VII. BESPOKE 8^{th}-ORDER PRODUCT FORMULA

The 8^{th}-order formula we use was numerically determined by solving equations as described in Reference [23]. We solved for over 100,000 product formulae, and selected the one with the smallest constant factor via testing with random Hamiltonians of matrix dimension 6×6. Further refinement was performed by minimising the Taylor expansion up to 9^{th}-order (so including the error term) and then solving for the 8^{th}-order formula. The method for performing the Taylor expansion is described in Reference [24]. The symmetric product formula has the form

$$S_{\text{prod}} = S_2(w_{10}t)S_2(w_9t)....S_2(w_2t)S_2(w_1t)S_2(w_0t)S_2(w_1t)S_2(w_2t)....S_2(w_9t)S_2(w_{10}t) \tag{87}$$

with $w_0 = 1 - 2\sum_{i=1}^{10} w_i$ and $S_2(t) = e^{-itH_0/2}e^{-itH_1}e^{-itH_0/2}$ for two non-commuting Hamiltonians H_0 and H_1. The following formula was numerically determined

$$\begin{aligned} w = [&5.935806040085031 \times 10^{-1}, \\ &-4.691601234700394 \times 10^{-1}, \\ &2.743566425898439 \times 10^{-1}, \\ &1.719387948465702 \times 10^{-1}, \\ &2.343987448254160 \times 10^{-1}, \\ &-4.861642448032533 \times 10^{-1}, \\ &4.961736738811380 \times 10^{-1}, \\ &-3.266021894843879 \times 10^{-1}, \\ &2.327167934936900 \times 10^{-1}, \\ &9.824955741471075 \times 10^{-2}] \end{aligned}$$

and is the bespoke 8^{th}-order formula we use in this work.

VIII. PRECISION REQUIREMENTS FOR STOPPING POWER

In Fig. 5 we compare the precision in the kinetic energy of the projectile to the resultant precision in the stopping power. We find that for a precision of 0.1 eV/Å in the stopping power we require a precision in the kinetic energy of approximately 0.02 Ha at the highest velocity which corresponds to approximately 10^4 samples. If the precision is lowered to 0.1 eV / Å this sampling overhead drops to around $\mathcal{O}(10^2)$.

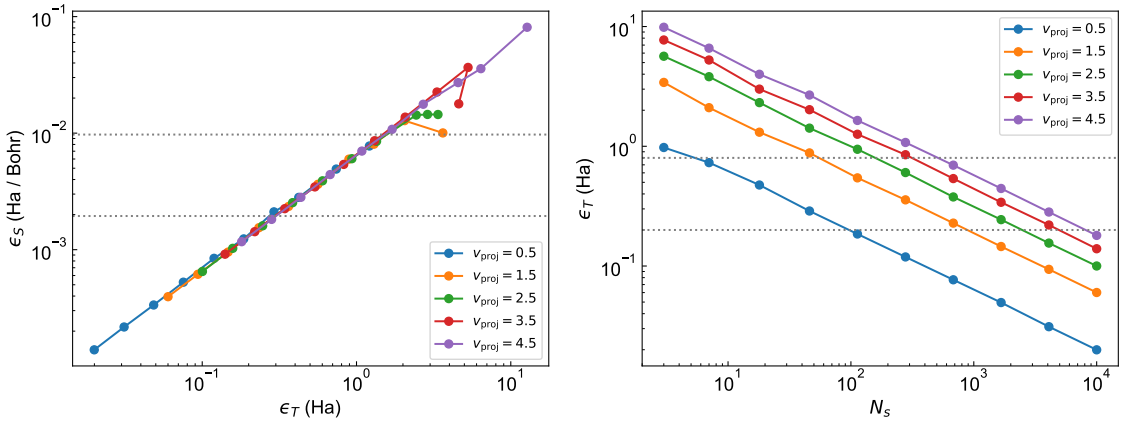

Figure 5. (left) Dependence of precision in the stopping power ϵ_S estimate on the precision in the individual kinetic energy data points ϵ_T for different values of the projectiles initial velocity ($v_{\rm proj}$). We extracted the stopping power using 10 equally spaced points and chose $\sigma_k = 4$. Dashed lines represent a desired target precision of 0.1 and 0.05 eV/Å. (right) Mean precision in the kinetic energy as a function of the number of samples N_s. Dashed lines correspond to the values of ϵ_T which yield the desired ϵ_S in the right hand panel.

IX. SCALING OF 1-NORM

Here we derive the explicit form of the 1-norm of τ for plane waves in a cubic region, showing that the constant factor in Eq. (76) for $\|\tau\|$ is also accurate for $\|\tau\|_1$. An explicit formula for τ is given in Ref. [25]. From Eq. (C4) of that work,

$$\tau_{pq} = \frac{1}{2N}\sum_{\nu} k_\nu^2 \cos(k_\nu \cdot r_{q-p})\,. \tag{88}$$

Using this expression, the 1-norm is given by

$$\begin{aligned}\|\tau\|_1 &= \max_p \sum_q |\tau_{pq}| \\ &= \frac{1}{2N}\max_p \sum_q \left|\sum_\nu k_\nu^2 \cos(k_\nu \cdot r_{q-p})\right| \\ &= \frac{1}{2N}\sum_q \left|\sum_\nu k_\nu^2 \cos(k_\nu \cdot r_q)\right|\,.\end{aligned} \tag{89}$$

In the third line we have used symmetry to find that the expression is independent of p, so the maximum is not needed. Then, we can re-express the sum over ν as

$$\begin{aligned}\sum_\nu k_\nu^2 \cos(k_\nu \cdot r_q) &= \sum_\nu k_\nu^2 \exp(i\, k_\nu \cdot r_q) \\ &= \sum_\nu k_\nu^2 \exp\left(\frac{2\pi i}{\Omega^{1/3}}\nu \cdot r_q\right) \\ &= -\left(\frac{\partial^2}{\partial r_x^2}+\frac{\partial^2}{\partial r_y^2}+\frac{\partial^2}{\partial r_z^2}\right)\sum_\nu \exp\left(\frac{2\pi i}{\Omega^{1/3}}\nu \cdot r_q\right)\,.\end{aligned} \tag{90}$$

This reasoning is similar to Eq. (F11) in Ref. [25]. In the first line we used the symmetry over ν to find that the imaginary part of the exponential cancels. The second line is the conversion between ν and k_ν for plane waves. In the third line we are using r_x, r_y, r_z to denote the components of r_q.

Each component of ν is summed over the range between plus and minus $(N^{1/3}-1)/2$ to give, for r_x, r_y, r_z nonzero

$$\sum_\nu \exp\left(\frac{2\pi i}{\Omega^{1/3}}\nu \cdot r_q\right) = \frac{\sin(\pi r_x (N/\Omega)^{1/3})}{\sin(\pi r_x/\Omega^{1/3})}\frac{\sin(\pi r_y (N/\Omega)^{1/3})}{\sin(\pi r_y/\Omega^{1/3})}\frac{\sin(\pi r_z (N/\Omega)^{1/3})}{\sin(\pi r_z/\Omega^{1/3})}\,. \tag{91}$$

The components of r take values that are integer multiples of $(\Omega/N)^{1/3}$, which implies that the numerators are equal to zero. This means that the numerators are equal to zero, except where r_x, r_y, r_z are zero. We also need to take account of nonzero values because we are taking derivatives. In the case where r_x is nonzero but $r_y, r_z = 0$, we obtain

$$\sum_\nu \exp\left(\frac{2\pi i}{\Omega^{1/3}}\nu\cdot r_q\right) = \frac{\sin(\pi r_x(N/\Omega)^{1/3})}{\sin(\pi r_x/\Omega^{1/3})}N^{2/3}\,, \tag{92}$$

where the factor of $N^{2/3}$ comes from two factors of $N^{1/3}$ from the sums over ν_y and ν_z. Then taking the second derivative with respect to r_x gives

$$-\frac{\partial^2}{\partial r_x^2}\sum_\nu \exp\left(\frac{2\pi i}{\Omega^{1/3}}\nu\cdot r_q\right) = (-1)^{q_x}\frac{2\pi^2 N^{1/3}\cos(\pi q_x/N^{1/3})}{\Omega^{2/3}\sin^2(\pi q_x/N^{1/3})}N^{2/3}. \tag{93}$$

The sum of the absolute value of this expression over nonzero q_x then gives

$$\sum_{q_x\neq 0}\left|(-1)^{q_x}\frac{2\pi^2 N^{1/3}\cos(\pi q_x/N^{1/3})}{\Omega^{2/3}\sin^2(\pi q_x/N^{1/3})}N^{2/3}\right| = \sum_{q_x=1}^{(N^{1/3}-1)/2}\frac{4\pi^2 N\cos(\pi q_x/N^{1/3})}{\Omega^{2/3}\sin^2(\pi q_x/N^{1/3})}. \tag{94}$$

The majority of the contribution to this sum is from small values of q_x, so we can approximate the sum by

$$\begin{aligned}\frac{4\pi^2 N}{\Omega^{2/3}}\sum_{q_x=1}^{(N^{1/3}-1)/2}\frac{1}{(\pi q_x/N^{1/3})^2} &= N\frac{4N^{2/3}}{\Omega^{2/3}}\sum_{q_x=1}^{(N^{1/3}-1)/2}\frac{1}{q_x^2}\\ &\approx N\frac{4N^{2/3}}{\Omega^{2/3}}\frac{\pi^2}{6}\,.\end{aligned} \tag{95}$$

It can be shown that a higher-order approximation is

$$\sum_{q_x=1}^{(N^{1/3}-1)/2}\frac{\cos(\pi q_x/N^{1/3})}{\sin^2(\pi q_x/N^{1/3})} \approx \frac{N^{2/3}}{6} - \frac{N^{1/3}}{\pi} + \frac{1}{12} + \frac{\pi}{24N^{1/3}}\,. \tag{96}$$

In the case $r_x = r_y = r_z = 0$,

$$\begin{aligned}\sum_\nu k_{\nu_x}^2\cos(k_\nu\cdot r_q) &= \sum_{\nu_x=-(N^{1/3}-1)/2}^{(N^{1/3}-1)/2}\left(\frac{2\pi\nu_x}{\Omega^{1/3}}\right)^2\times N^{2/3}\\ &= \frac{\pi^2}{3\Omega^{2/3}}(N^{2/3}-1)N^{1/3}\times N^{2/3}\\ &\approx N\frac{\pi^2 N^{2/3}}{3\Omega^{2/3}}\,.\end{aligned} \tag{97}$$

In the first two lines we have shown $\times N^{2/3}$ separately at the end to indicate the result of the sums over ν_y and ν_z. Adding this expression to that for the sum over $r_x \neq 0$ gives

$$\sum_{q_x}\left|\sum_\nu k_{\nu_x}^2\cos(k_\nu\cdot r_q)\right| = N\frac{\pi^2 N^{2/3}}{\Omega^{2/3}}. \tag{98}$$

We therefore obtain

$$
\begin{aligned}
\|\tau\|_1 &= \frac{1}{2N}\sum_q \left|\sum_\nu k_\nu^2 \cos(k_\nu \cdot r_q)\right| \\
&= \frac{1}{2N}\sum_q \left|\sum_\nu (k_{\nu_x}^2 + k_{\nu_y}^2 + k_{\nu_z}^2)\cos(k_\nu \cdot r_q)\right| \\
&= \frac{1}{2N}\left|\sum_\nu (k_{\nu_x}^2 + k_{\nu_y}^2 + k_{\nu_z}^2)\cos(k_\nu \cdot r_q)\right|_{q=0} + \frac{1}{2N}\sum_{\substack{q_x\neq 0\\ q_y=q_z=0}}\left|\sum_\nu k_{\nu_x}^2 \cos(k_\nu \cdot r_q)\right| \\
&\quad + \frac{1}{2N}\sum_{\substack{q_y\neq 0\\ q_x=q_z=0}}\left|\sum_\nu k_{\nu_y}^2 \cos(k_\nu \cdot r_q)\right| + \frac{1}{2N}\sum_{\substack{q_z\neq 0\\ q_x=q_y=0}}\left|\sum_\nu k_{\nu_z}^2 \cos(k_\nu \cdot r_q)\right| \\
&= \frac{3}{2N}\sum_{\substack{q_x\\ q_y=q_z=0}}\left|\sum_\nu k_{\nu_x}^2 \cos(k_\nu \cdot r_q)\right| \\
&\approx \frac{3}{2N} N \frac{\pi^2 N^{2/3}}{\Omega^{2/3}} \\
&= \frac{3\pi^2 N^{2/3}}{2\Omega^{2/3}}.
\end{aligned}
\tag{99}
$$

We have used the symmetry between the x, y, z directions, and used the property that the sums only give nonzero results if at least two of r_x, r_y, r_z are zero.

[1] F. Graziani, M. P. Desjarlais, R. Redmer, and S. B. Trickey, *Frontiers and challenges in warm dense matter*, Vol. 96 (Springer Science & Business, 2014).

[2] T. Dornheim, S. Groth, and M. Bonitz, The uniform electron gas at warm dense matter conditions, Physics Reports **744**, 1 (2018).

[3] W. Kohn and L. J. Sham, Self-consistent equations including exchange and correlation effects, Physical Review **140**, A1133 (1965).

[4] N. D. Mermin, Thermal properties of the inhomogeneous electron gas, Physical Review **137**, A1441 (1965).

[5] R. Babbush, W. J. Huggins, D. W. Berry, S. F. Ung, A. Zhao, D. R. Reichman, H. Neven, A. D. Baczewski, and J. Lee, Quantum simulation of exact electron dynamics can be more efficient than classical mean-field methods, Nature Communications **14**, 4058 (2023).

[6] B. Militzer, Path integral Monte Carlo and density functional molecular dynamics simulations of hot, dense helium, Physical Review B **79**, 155105 (2009).

[7] K. P. Driver and B. Militzer, All-electron path integral Monte Carlo simulations of warm dense matter: Application to water and carbon plasmas, Physical Review Letters **108**, 115502 (2012).

[8] B. Militzer, F. González-Cataldo, S. Zhang, K. P. Driver, and F. Soubiran, First-principles equation of state database for warm dense matter computation, Physical Review E **103**, 013203 (2021).

[9] A. Baldereschi, Mean-value point in the Brillouin zone, Physical Review B **7**, 5212 (1973).

[10] M. Bagherimehrab, Y. R. Sanders, D. W. Berry, G. K. Brennen, and B. C. Sanders, Nearly optimal quantum algorithm for generating the ground state of a free quantum field theory, PRX Quantum **3**, 020364 (2022).

[11] G. Kresse and J. Furthmüller, Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set, Physical Review B **54**, 11169 (1996).

[12] G. Kresse and J. Furthmüller, Efficiency of ab-initio total energy calculations for metals and semiconductors using a plane-wave basis set, Comp. Mater. Sci. **6**, 15 (1996).

[13] G. Kresse and D. Joubert, From ultrasoft pseudopotentials to the projector augmented-wave method, Physical Review B **59**, 1758 (1999).

[14] A. D. Baczewski, L. Shulenburger, M. Desjarlais, S. Hansen, and R. Magyar, X-ray thomson scattering in warm dense matter without the Chihara decomposition, Physical Review Letters **116**, 115004 (2016).

[15] R. J. Magyar, L. Shulenburger, and A. D. Baczewski, Stopping of deuterium in warm dense deuterium from Ehrenfest time-dependent density functional theory, Contrib. Plasm. Phys. **56**, 459 (2016).

[16] T. W. Hentschel, A. Kononov, A. Olmstead, A. Cangi, A. D. Baczewski, and S. B. Hansen, Improving dynamic collision frequencies: impacts on dynamic structure factors and stopping powers in warm dense matter, Physics of Plasmas **30**, 062703 (2023).

[17] A. Kononov, T. Hentschel, S. B. Hansen, and A. D. Baczewski, Trajectory sampling and finite-size effects in first-principles stopping power calculations, arXiv:2307.03213 (2023).
[18] Y. Su, D. Berry, N. Wiebe, N. Rubin, and R. Babbush, Fault-tolerant quantum simulations of chemistry in first quantization, PRX Quantum **4**, 040332 (2021).
[19] Y. R. Sanders, D. W. Berry, P. C. S. Costa, L. W. Tessler, N. Wiebe, C. Gidney, H. Neven, and R. Babbush, Compilation of Fault-Tolerant Quantum Heuristics for Combinatorial Optimization, PRX Quantum **1**, 020312 (2020).
[20] G. H. Low, Y. Su, Y. Tong, and M. C. Tran, Complexity of implementing Trotter steps, PRX Quantum **4**, 020323 (2023).
[21] N. C. Rubin, K. Gunst, A. White, L. Freitag, K. Throssell, G. K.-L. Chan, R. Babbush, and T. Shiozaki, The fermionic quantum emulator, Quantum **5**, 568 (2021).
[22] R. Kothari and R. O'Donnell, Mean estimation when you have the source code; or, quantum Monte Carlo methods, in *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)* (SIAM, 2023) pp. 1186–1215.
[23] H. Yoshida, Construction of higher order symplectic integrators, Physics Letters A **150**, 262 (1990).
[24] M. E. S. Morales, P. Costa, D. K. Burgarth, Y. R. Sanders, and D. W. Berry, Greatly improved higher-order product formulae for quantum simulation, arXiv:2210.15817 (2022).
[25] R. Babbush, N. Wiebe, J. McClean, J. McClain, H. Neven, and G. K.-L. Chan, Low-Depth Quantum Simulation of Materials, Physical Review X **8**, 011044 (2018).