Supplementary Information for: Hybrid Quantum Algorithm for Simulating Real-Time Thermal Correlation Functions

Elliot C. Eklund[1,2,a] and Nandini Ananth[2b]
[1]*School of Chemistry, University of Sydney, NSW 2006, Australia and*
[2]*Department of Chemistry and Chemical Biology,*
Cornell University, Ithaca, New York 14853, USA

[a] Email: elliot.eklund@sydney.edu.au
[b] Email: ananth@cornell.edu

I. BOUNDING THERMAL CORRELATION FUNCTION ERROR

In this section we prove that the error made from approximating the TCF using $\tilde{U}(t_c)$ can be bound by

$$\epsilon_C \leq \frac{2}{Z}\|A\|_1\|B\|_1\|e^{-\beta H/2}\|_1\epsilon_U, \tag{1}$$

where $\|M\|_1 = \mathrm{Tr}\sqrt{MM^\dagger}$ is the matrix 1-norm or trace norm, and we assume that $\|U(t_c) - \tilde{U}(t_c)\|_1 \leq \epsilon_U$. We start by defining the error as $\epsilon_C = |C_{AB}(t_c) - \tilde{C}_{AB}(t_c)|$, where $\tilde{C}_{AB}(t)$ is the TCF approximated by $\tilde{U}(t_c)$. Taking the difference and using that the trace is linear gives

$$\begin{aligned}
\epsilon_C &= \frac{1}{Z}\left|\mathrm{Tr}\left(U^\dagger(t_c)AU(t_c)B - \tilde{U}^\dagger(t_c)A\tilde{U}(t_c)B\right)\right| \\
&\leq \frac{1}{Z}\mathrm{Tr}\left|U^\dagger(t_c)AU(t_c)B - \tilde{U}^\dagger(t_c)A\tilde{U}(t_c)B\ \right| \\
&\leq \frac{1}{Z}\|U^\dagger(t_c)AU(t_c)B - \tilde{U}^\dagger(t_c)A\tilde{U}(t_c)B\|_1.
\end{aligned} \tag{2}$$

Using that $\|\cdot\|_1$ is sub-multiplicative and sub-linear, we find

$$\begin{aligned}
\epsilon_C &\leq \frac{1}{Z}\|U^\dagger(t_c)AU(t_c) - \tilde{U}^\dagger(t_c)A\tilde{U}(t_c)\|_1\|B\|_1 \\
&= \frac{1}{Z}\|U^\dagger(t_c)AU(t_c) - \tilde{U}^\dagger(t_c)A\tilde{U}(t_c) + \tilde{U}^\dagger(t_c)AU(t_c) - \tilde{U}^\dagger(t_c)AU(t_c)\|_1\|B\|_1 \\
&\leq \frac{1}{Z}\left(\|(U^\dagger(t_c) - \tilde{U}^\dagger(t_c))AU(t_c)\|_1 + \|\tilde{U}^\dagger(t_c)A(U(t_c) - \tilde{U}(t_c))\|_1\right)\|B\|_1 \\
&\leq \frac{1}{Z}\left(\|A\|_1\|U(t_c)\|_1 + \|\tilde{U}^\dagger(t_c)A\|_1\right)\epsilon_U\|B\|_1,
\end{aligned} \tag{3}$$

where we use that $\|U(t_c) - \tilde{U}(t_c)\|_1 = \|U^\dagger(t_c) - \tilde{U}^\dagger(t_c)\|_1$. Proceeding,

$$\begin{aligned}
\epsilon_C &\leq \frac{1}{Z}\left(\|A\|_1\|U(t_c)\|_1 + \|\tilde{U}^\dagger(t_c)A + U^\dagger(t_c)A - U^\dagger(t_c)A\|_1\right)\epsilon_U\|B\|_1 \\
&\leq \frac{1}{Z}\left(\|A\|_1\|U(t_c)\|_1 + \epsilon_U\|A\|_1 + \|U^\dagger(t_c)\|_1\|A\|_1\right)\epsilon_U\|B\|_1 \\
&= \frac{1}{Z}\left(\|A\|_1\|e^{-\beta H/2}\|_1 + \epsilon_U\|A\|_1 + \|e^{-\beta H/2}\|_1\|A\|_1\right)\epsilon_U\|B\|_1,
\end{aligned} \tag{4}$$

where the last line follows from splitting $U(t_c)$ and $U^\dagger(t_c)$ into a product of real and imaginary time evolution operators and recognizing that the real time contributions cancel in $\|\cdot\|_1$. We arrive at our final result by keeping only terms first order in ϵ_U,

$$\epsilon_C \leq \frac{2}{Z}\|A\|_1\|B\|_1\|e^{-\beta H/2}\|_1\epsilon_U. \tag{5}$$

II. BOUNDING DVR APPROXIMATION ERROR

Here, we prove bounds on the error between the exact one-dimensional DVR kinetic energy matrix and an approximation in which a set of diagonals are neglected. We start by assigning to each diagonal of DVR_α an index, $\nu \in [1, D]$, where $D = 2^n$, such that the main diagonal is $\nu = 1$, the adjacent upper and lower diagonals are $\nu = 2$, and so on. Let $\widetilde{DVR}_\alpha(\ell)$ be an approximation to DVR_α such that all diagonals $\nu \geq \ell$ are neglected. We define the error between DVR_α and $\widetilde{DVR}_\alpha(\ell)$ as

$$\delta = \|\Delta DVR_\alpha(\ell)\|_2 \tag{6}$$

where $\Delta DVR_\alpha(\ell) = DVR_\alpha - \widetilde{DVR}_\alpha(\ell)$ and $\|M\|_2 = \sqrt{\mathrm{Tr}\,(MM^\dagger)}$ is the matrix 2-norm or Frobenius norm. Computing the trace and using the fact that $\Delta DVR_\alpha(\ell)$ is real and symmetric produces a sum $S(l)$ given by

$$S(\ell) = D(\Delta DVR_{\alpha,11}(\ell))^2 + 2\sum_{\nu=2}^{D}\nu\left(\Delta DVR_{\alpha,\nu\nu}(\ell)\right)^2, \tag{7}$$

where the sum runs over each diagonal of the upper triangular block. The only diagonals of $\Delta DVR_\alpha(\ell)$ with non-zero elements are when $\nu < \ell$. Combining this fact with the definition

$$DVR_{\alpha,ij} = \frac{(-1)^{i-j}}{2m_\alpha(\Delta x)^2} \begin{cases} \frac{\pi^2}{3}, & i = j \\ \frac{2}{(i-j)^2}, & i \neq j \end{cases} \tag{8}$$

yields

$$S(\ell) = 2K^2 \sum_{\nu=\ell+1}^{D} \frac{\nu}{(\nu-1)^4} \tag{9}$$

where $K = \hbar^2/m_\alpha(\Delta x)^2$ and we have dropped the $\nu = 1$ term since it only contributes when $\Delta DVR_\alpha = \mathbf{0}$. As this corresponds to approximating DVR_α as a matrix of zeros, it is physically unmeaningful.

A. Error Upper Bound

We now prove that the error is upper bound by $\delta < \frac{2\sqrt{2}K}{\ell}$. To do this, we first prove that $S(\ell)$ is upper bound by $S(\ell) < 2K^2\left(\frac{1}{\ell} + \frac{1}{\ell^{3/2}}\right)^2$. Note that $S(\ell)$ is equivalent to the area comprised by a set of rectangles with heights $\frac{\nu}{(\nu-1)^4}$ centered at $x = \nu$ and edges specified by the points $\nu - 1/2$ and $\nu + 1/2$. This area is always be less than

$$A(\ell) = 2K^2\left(\frac{\ell+1}{\ell^4} + \int_{\ell+3/2}^{D+1/2} dx \frac{x-1/2}{(x-3/2)^4}\right), \tag{10}$$

where the first term gives the area of the rectangle for $\nu = \ell + 1$. The integral gives the area of the remaining terms and will always be larger than the associated sum because we shift the integrand to the right by $x = 1/2$ so that it intersects the right corner of each rectangle rather than its midpoint. Because the first term covers the domain $[\ell+1/2, \ell+3/2]$, the integral runs from $[\ell+3/2, D+1/2]$. In the limit of large D and upon integration, the difference between the upper bound and $A(\ell)$ becomes

$$2K^2\left(\frac{1}{\ell} + \frac{1}{\ell^{3/2}}\right)^2 - A(\ell) = \frac{2K^2}{\ell^{5/2}}\left(\ell^{1/2} + 2 - \left(\frac{1}{3\ell^{1/2}} + \frac{1}{\ell^{3/2}}\right)\right) \tag{11}$$

Because the positive terms on the RHS are monotonically increasing in magnitude and the negative terms are monotonically decreasing in magnitude, if the expression is greater than zero for a given ℓ, it must also be greater than zero for all larger ℓ. For $\ell = 1$, we find $14K^2/7$. This completes the proof.

We arrive at our final expression for the error upper bound by taking the square root,

$$\begin{aligned} \delta &< \frac{\sqrt{2}K}{\ell^{3/2}}\left(\ell^{1/2} + 1\right) \\ &< \frac{\sqrt{2}K}{\ell^{3/2}}\left(2\ell^{1/2}\right). \end{aligned} \tag{12}$$

B. Error Lower Bound

We now prove that the error is lower bound by $\delta > \frac{2K}{\ell^{3/2}}$. To do this, we first prove that $S(\ell) > \frac{4K^2}{\ell^3}$. This can be seen by showing that the sum of the first three terms of $S(\ell)$ is always larger than $\frac{4K^2}{\ell^3}$. When the remaining terms are included in $S(\ell)$, the inequality still holds because each term is positive. To show that the first three terms of $S(\ell)$ are always larger than $\frac{4K^2}{\ell^3}$, we take the difference between them and note that after expanding, all positive terms are monotonically increasing in magnitude while all negative terms are monotonically decreasing in magnitude. Following the same logic as for the error upper bound proof, we substitute $\ell = 1$ into the expression and find that it yields a positive number, proving the lower bound. Note that $D \geq \ell + 3$ must be true for this argument to hold. Finally, taking the square root yields

$$\delta > \frac{2K}{\ell^{3/2}}. \tag{13}$$

C. Error Bound Comparison

In Fig. 1 we compare the exact error of approximating DVR_α, given by $\sqrt{S(l)}$, with the upper and lower bounds derived above. We can see from Fig. 1 that the lower bound closely approximates the exact error and does a better job doing so than the upper bound. For this reason and because Eq. (13) is easier to work with and gives more insight than Eq. (9), we take the error lower bound as our expression for the error of approximating DVR_α.

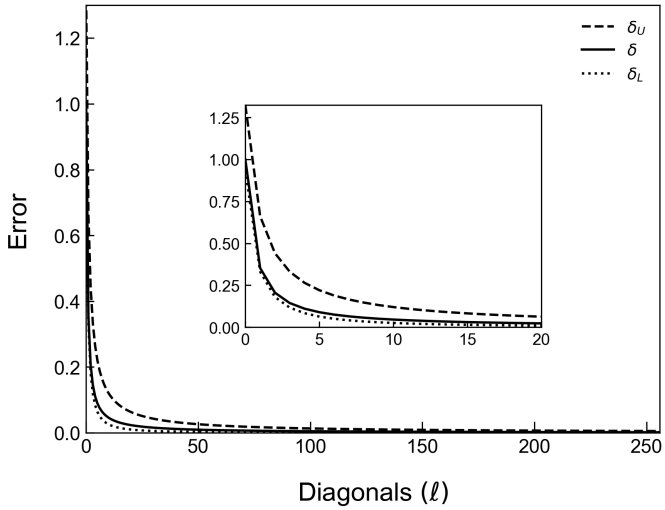


FIG. 1: Error from approximating DVR_α by neglecting all diagonals $\nu \geq \ell$. Here, δ_U and δ_L are the upper and lower error bounds, respectively, and δ is the exact error. We set $K = 1$ and use $D = 512$, however, we note similar results are obtained for K ranging several orders of magnitude. Each curve is normalized by the maximum error of δ.

III. 1-SPARSE DECOMPOSITION

In this section we prove that any matrix $\text{diag}(DVR_\alpha, \nu)$ can be decomposed into a sum of at most two 1-sparse, Hermitian matrices. Here, $\text{diag}(\cdot, \nu)$ returns the input matrix with all elements neglected except those on the νth upper and lower diagonals. A matrix is said to be k-sparse if each row and column have at most k elements. Thus, a 1-sparse matrix has at most one element in every row and column. We note that because $\text{diag}(DVR_\alpha, \nu)$ consists of at most two diagonals (except $\nu = 1$, which is the main diagonal), then $\text{diag}(DVR_\alpha, \nu)$ is at most 2-sparse. Additionally, $\text{diag}(DVR_\alpha, \nu)$ is always Hermitian since DVR_α is Hermitian as well.

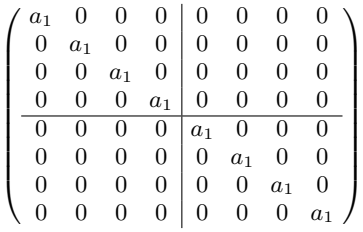
Let $h_U(\nu)$ and $h_L(\nu)$ be the list of pairs of indices corresponding to the upper and lower diagonals of $\text{diag}(DVR_\alpha, \nu)$, respectively. For example, $h_U(\nu) = \{(r_1, c_1), (r_2, c_2) \dots\}$, where r_j and c_j are the row and column index, respectively, and $h_L(\nu)$ is defined similarly. Given $h_U(\nu)$ and $h_L(\nu)$, we can determine if $\text{diag}(DVR_\alpha, \nu)$ is 1-sparse by checking that none of the row indices in $h_U(\nu)$ are the same as $h_L(\nu)$ and that none of the column indices in $h_U(\nu)$ are the same as $h_L(\nu)$. If this is true, then $\text{diag}(DVR_\alpha, \nu)$ is 1-sparse.

Each $\text{diag}(DVR_\alpha(\nu))$ can be grouped into one of four cases based on the value of ν. We prove the above claim on a case-by-cases basis. Fig. 2 gives an example matrix for each of the four cases.

Case 1, $\nu = 1$: This is automatically 1-sparse and Hermitian as $\text{diag}(DVR_\alpha, 1)$ corresponds to the main diagonal.

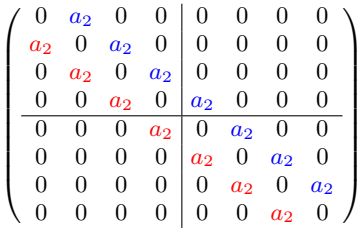
Case 2, $\nu > D/2$: This is automatically 1-sparse and Hermitian as well. To see this, note that the jth index pair of $h_U(\nu)$ and $h_L(\nu)$ are given by $(j, \nu + j - 1)_U$ and $(\nu + j - 1, j)_L$, respectively. We can write $\nu = D/2 + q$ with $q \in [1, D/2]$. Plugging this into the index pairs, we find $(j, D/2 + q + j - 1)_U$ and $(D/2 + q + j - 1, j)_L$. For a given q, j ranges from $[1, D/2 - q]$. From this, we can see that all column indices in the upper and lower diagonals will never be the same. This is also true for all row indices. Thus, $\text{diag}(DVR_\alpha, \nu)$ is 1-sparse. Hermiticity follows form DVR_α being Hermitian.

Case 3a, $1 < \nu \leq D/2$ and ν is even: Here, each $\text{diag}(DVR_\alpha, \nu)$ is 2-sparse and can be decomposed into a sum of two 1-sparse matrices in the following way. Starting from the leftmost element of the upper and lower diagonals, group this pair into a matrix, M_1. Group the adjacent element to the right of the upper and lower diagonals into a matrix, M_2. Continue alternating between grouping pairs into M_1 and M_2 until all pairs have been grouped. Both M_1 and M_2 are 1-sparse and Hermitian, and the sum of both matrices is $\text{diag}(DVR_\alpha, \nu)$.

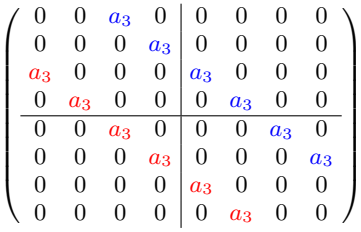


(a) Case 1: $\nu = 1$.

(b) Case 2: $\nu = 5$.



(c) Case 3a: $\nu = 2$.



(d) Case 3b: $\nu = 3$.

FIG. 2: Examples of various diag(DVR_α, ν) matrices for different values of ν when $D = 8$. The upper and lower diagonals are shown in blue and red, respectively. a_i is the constant matrix element corresponding to diag(DVR_α, ν).

To see this, we write $\nu = D/2 - q$ where q is even and $q \in [0, D/2 - 2]$. Inserting this into the index pairs we find $(j, D/2 - q + j - 1)_U$ and $(D/2 - q + j - 1, j)_L$. Note that all pairs where j is odd are grouped into M_1. Then, because q must be even, the opposite index will always be even as well. That is, $(j, D/2 - q + j - 1)_U \to (odd, even)_U$ and $(D/2 - q + j - 1, j)_L \to (even, odd)_L$. Because the row indices in the upper and lower diagonals are always odd and even, respectively, they can never be the same. Similarly, because the column indices in the upper and lower diagonals are always even and odd, respectively, they too can never be the same. Hence, M_1 is 1-sparse.

All index pairs where j is even are grouped into M_2. Here, the opposite index is always odd. That is, $(j, D/2 - q + j - 1)_U \to (even, odd)_U$ and $(D/2 - q + j - 1, j)_L \to (odd, even)_L$. Following the same logic as for M_1, we see that M_2 is also 1-sparse. Because M_1 and M_2 are both symmetric and all matrix elements are the same, they are both Hermitian as well.

Case 3b, $1 < \nu \leq D/2$ and ν is odd: Here, each diag(DVR_α, ν) is 2-sparse and can be decomposed into a sum of two 1-sparse matrices in the following way. Let p and b be a pair of positive integers such that

$$b = \left(\frac{\nu - 1}{2^p} \right), \tag{14}$$

and b is odd. We note that such a pair can be shown to always exist and be unique. Starting from the leftmost element of the upper and lower diagonals, partition the matrix into sets of 2^p contiguous elements. Then, starting from the leftmost *set* of both upper and lower diagonals, group every other set into a matrix, M_1. Group all remaining sets into another matrix, M_2. Both M_1 and M_2 are 1-sparse and Hermitian, and the sum of both matrices is diag(DVR_α, ν).

To see this, we assign each set of contiguous elements an integer index q where $q \in [1, m]$ and $m = b(D - \nu + 1)/(\nu - 1)$. Assign the leftmost set $q = 1$. Moving to the right, assign the adjacent set $q = 2$. Continue assigning indices in this way until the rightmost set is reached. For M_1, the upper diagonal row indices and lower diagonal column indices of the qth set of contiguous elements range from $[(q-1)2^p + 1, q2^p]$ while the upper diagonal column indices and lower diagonal row indices of the qth set of contiguous elements range from $[\nu + (q-1)2^p, \nu - 1 + q2^p]$. Note that q is always odd for M_1. Using the relationship $\nu - 1 = 2^p$ gives $[b2^p(q-1)2^p + 1, b2^p + q2^p]$ for the range of upper diagonal column indices and lower diagonal row indices. We summarize this in Table I below.

	row indices	column indices
upper diagonal	$[q - 1 + r, q] \cdot 2^p$	$[q + b - 1 + r, q + b] \cdot 2^p$
lower diagonal	$[q + b - 1 + r, q + b] \cdot 2^p$	$[q - 1 + r, q] \cdot 2^p$

TABLE I: Upper and lower diagonal range of indices for qth set of contiguous elements of M_1 for *Case 3b*. Each range is multiplied by 2^p and $r = 1/2^p$.

Consider the range of row indices for both upper and lower diagonals. Because $q-1$ is always even and $q+b-1$ is always odd, the range of row indices will never overlap for any given q. We can see this immediately holds for the range of column indices as well. Thus, M_1 is 1-sparse. We can see that M_2 is 1-sparse by repeating the same steps as for M_1 and noting that the only difference is that q is now even. The sum of both M_1 and M_2 is $\text{diag}(DVR_\alpha, \nu)$. Because M_1 and M_2 are both symmetric and all matrix elements are the same, they are both Hermitian as well. This completes the proof.

IV. QUANTUM CIRCUIT PRIMITIVES

In this section, we review a few primitive quantum circuits that form the building blocks of more complicated circuits used throughout the paper.

A. Controlled Rotation about z-axis: $C(R_z(\theta))$

We can implement $C(R_z(\theta))$ using two CNOT gates and two $R_z(\theta)$ gates as demonstrated in Fig. 3. The circuit depth and CNOT count of $C(R_z(\theta))$ is two.

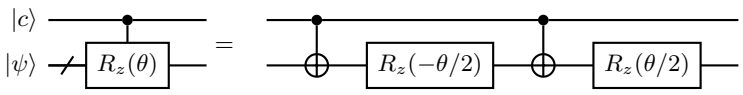


FIG. 3: $C(R_z(\theta))$ circuit. Here, $|c\rangle$ acts as the control.

B. Multiply-Controlled Rotation about z-axis: $C^m(R_z(\theta))$

We denote a unitary U with m controls as C^mU. A $C^mR_z(\theta)$ gate can be implemented using $2(m-1)$ Toffoli gates and $m-1$ ancilla qubits [1]. Each Toffoli gate can be implemented using six CNOT gates [1]. The total depth and CNOT count of $C^mR_z(\theta)$ is $12(m-1)+2$. A demonstration for $m=4$ is given in Fig. 4. We note that given

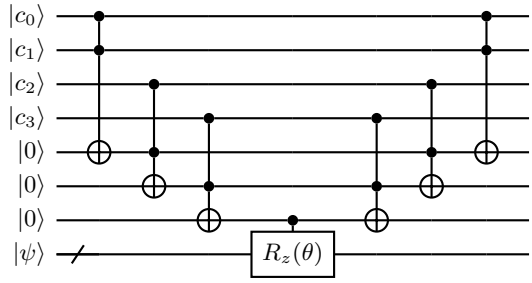


FIG. 4: Demonstration of $C^4R_z(\theta)$ circuit.

a circuit that implements CU, where U is an arbitrary unitary, C^mU can be constructed by replacing $CR_z(\theta)$ with CU in Fig. 4.

C. Greater-than Comparator: $O_>(a)$

The greater-than comparator $O_>(a)$ determines if a computational basis state is greater than a given input integer, a. Its action is given by

$$O_>(a)|x\rangle|0\rangle = |x\rangle|x > a\rangle. \tag{15}$$

Suppose n qubits are used to store $|x\rangle$. We can evaluate $x > a$ by adding $r = 2^n - a$ to x and checking the highest bit of the sum. $x > a$ is true if the highest bit is one, and false otherwise. We give a circuit that implements $O_>(a)$ in Fig. 5. The unitary U^{n+1}_{ADD} adds two n-bit integers together and returns the resulting $n+1$ bit sum, while U^n_{ADD} adds n-bit integers together returns only the first n bits of the sum. Hence, U^n_{ADD} ignores the highest carry bit of the sum.

Both U^{n+1}_{ADD} and U^n_{ADD} can be implemented using the method outlined in [2]. We note that the implementation of U^n_{ADD} is the same as U^{n+1}_{ADD}, except the controlled gates that target $|s_{n+1}\rangle$ in Fig. 5 are removed. The depth of U^{n+1}_{ADD} and U^n_{ADD} is $O(\log^2 n)$, and the CNOT count is $O(n\log(n))$ [2]. Hence, the circuit depth and CNOT count of $O_>(a)$ is also $O(\log^2 n)$ and $O(n \log n)$. Unfortunately, only the asymptotic complexity of the adder circuits is given in Remaud and Vandaele [2]. Thus, when an adder circuit is used in a more complex circuit, we denote the resource contribution form the adder asymptotically. We follow the convention of dropping terms that are polylogarithmic in n.

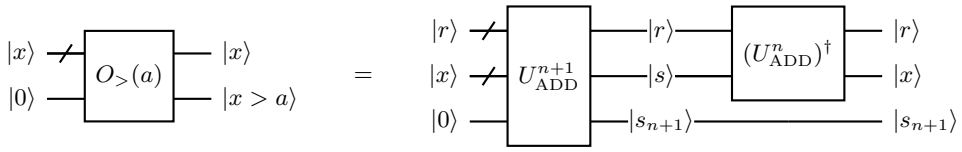


FIG. 5: Greater-than comparator circuit. Here, $|x\rangle$ contains an n-bit computational basis state and $|r\rangle$ stores n-bit integer that is added to $|x\rangle$. The state $|s\rangle$ contains the first n bits of the sum $|x + r\rangle$, and $|s_{n+1}\rangle$ is the $n+1$th bit.

We can construct a less-than comparator $O_<(a)$ from the greater-than comparator by conjugating $O_>(a)$ with X gates as shown in Fig. 6. The depth and CNOT count of $O_>(a)$ is the same as $O_>(a)$.

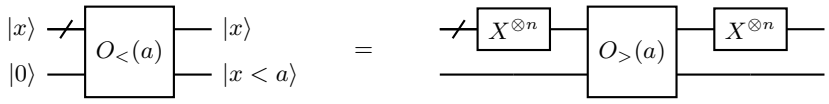


FIG. 6: Less-than comparator circuit. $O_<(a)$ is constructed by conjugating $O_>(a)$ with $X^{\otimes n}$ about the top n wires.

D. Controlled Greater-than Comparator: $CO_>(a)$

A controlled greater-than comparator can be implemented using the same construction as $O_>(a)$, except the adder circuits now have controls. We give a circuit that implement $CO_>(a)$ Fig. 7. Controlled-adder circuits CU^{n+1}_{ADD} and CU^n_{ADD} can be constructed using the method given in Remaud and Vandaele [2]. The circuit depth and CNOT count of CU^{n+1}_{ADD} and CU^n_{ADD} remains $O(\log^2(n))$ and $O(n \log n)$, respectively. Hence, the circuit depth and CNOT count of $CO_>(a)$ also remains the same.

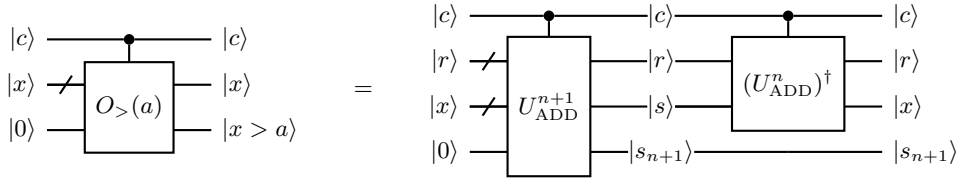


FIG. 7: Controlled greater-than comparator circuit constructed from controlled adder circuits.

To construct the controlled less-than comparator $CO_<(a)$, we conjugate $CO > (a)$ with X gates as shown in Fig. 8. The depth of $CO < (a)$ is the same as $CO > (a)$.

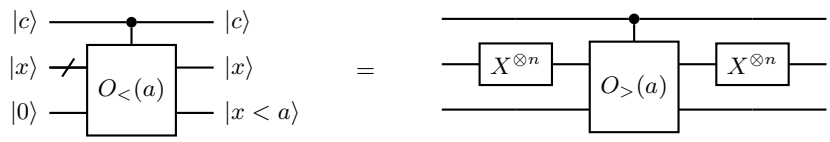


FIG. 8: Controlled less-than comparator circuit. $CO_<(a)$ is constructed by conjugating $CO_>(a)$ with $X^{\otimes n}$.

E. Controlled SWAP: CSWAP

The controlled SWAP gate CSWAP performs the following operation:

$$\begin{aligned} C\text{SWAP}|0\rangle|x_i\rangle|y_i\rangle &= |0\rangle|x_i\rangle|y_i\rangle \\ C\text{SWAP}|1\rangle|x_i\rangle|y_i\rangle &= |1\rangle|y_i\rangle|x_i\rangle, \end{aligned} \tag{16}$$

where $|x_i\rangle$ and $|y_i\rangle$ are single qubit states. A circuit that implements CSWAP is given in Fig. 9. To perform a controlled SWAP on two n-qubit states $|x\rangle$ and $|y\rangle$, we perform $C\text{SWAP}^{\otimes n}$. That is,

$$\begin{aligned} C\text{SWAP}^{\otimes n}|0\rangle|x\rangle|y\rangle &= |0\rangle|x\rangle|y\rangle \\ C\text{SWAP}^{\otimes n}|1\rangle|x\rangle|y\rangle &= |1\rangle|y\rangle|x\rangle. \end{aligned} \tag{17}$$

A circuit that implements $C\text{SWAP}^{\otimes n}$ is given in Fig. 10. We note that only a single ancilla is needed to implement $C\text{SWAP}^{\otimes n}$. This is because the ancilla qubit needed for a single CSWAP is reset to zero after the gate and can be reused for the remaining CSWAPs. The depth and CNOT count of $C\text{SWAP}^{\otimes n}$ is $18n$.

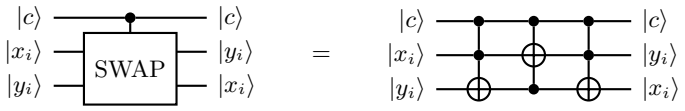


FIG. 9: Controlled SWAP circuit for a single qubit state.

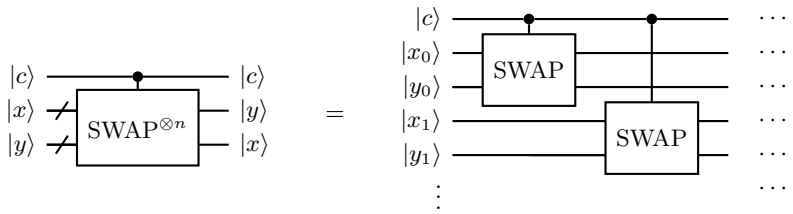


FIG. 10: Controlled SWAP circuit on n-qubit state. The single-qubit controlled SWAP gate is simply repeated on all n pairs $|x_i\rangle$ and $|y_i\rangle$.

V. DVR EVOLUTION CIRCUIT

In this section we give an explicit construction of a quantum circuit that implements time evolution under the DVR approximation to the kinetic energy operator. We focus on constructing the circuit for a one-dimensional system. However, the generalization to higher dimensions is straightforward.

Recall that the DVR matrix elements for a 1D system are given by

$$DVR_{ij} = \gamma(-1)^{i-j} \begin{cases} \lambda, & i = j \\ \frac{1}{(i-j)^2}, & i \neq j \end{cases}, \tag{18}$$

where $\gamma = \frac{1}{m\Delta x^2}$, and $\lambda = \frac{\pi^2}{6}$. We approximate time evolution under the DVR kinetic energy operator as

$$e^{-iDVR\Delta t} \approx \prod_{\nu=1}^{\ell} e^{-i\,\mathrm{diag}(DVR,\nu)\Delta t}, \tag{19}$$

where $\mathrm{diag}(\cdot,\nu)$ gives the input matrix with all elements neglected except those on the νth diagonals. As an example, the 4×4 DVR matrix is decomposed as

$$DVR = \underbrace{\begin{pmatrix} a_1 & 0 & 0 & 0 \\ 0 & a_1 & 0 & 0 \\ 0 & 0 & a_1 & 0 \\ 0 & 0 & 0 & a_1 \end{pmatrix}}_{\mathrm{diag}(DVR,1)} + \underbrace{\begin{pmatrix} 0 & a_2 & 0 & 0 \\ a_2 & 0 & a_2 & 0 \\ 0 & a_2 & 0 & a_2 \\ 0 & 0 & a_2 & 0 \end{pmatrix}}_{\mathrm{diag}(DVR,2)} + \underbrace{\begin{pmatrix} 0 & 0 & a_3 & 0 \\ 0 & 0 & 0 & a_3 \\ a_3 & 0 & 0 & 0 \\ 0 & a_3 & 0 & 0 \end{pmatrix}}_{\mathrm{diag}(DVR,3)} + \underbrace{\begin{pmatrix} 0 & 0 & 0 & a_4 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ a_4 & 0 & 0 & 0 \end{pmatrix}}_{\mathrm{diag}(DVR,4)}. \tag{20}$$

Here $a_\nu = DVR_{ij}$, such that $|i-j+1| = \nu$.

In SI Section III, we proved that $\mathrm{diag}(DVR,\nu)$ is either 1-sparse or 2-sparse. If (DVR,ν) is 2-sparse, we proved that it can always be decomposed as a sum of two 1-sparse Hermitian matrices and gave a method for finding such a decomposition. We can therefore approximation time evolution under DVR as

$$e^{-iDVR\Delta t} \approx \prod_{\nu=1}^{\ell}\prod_{\sigma=1}^{2} e^{-i\,\mathrm{diag}(DVR,\nu)_\sigma \Delta t}, \tag{21}$$

where σ indexes the 1-sparse matrices resulting from the decomposition. For example, the decomposition of $\mathrm{diag}(DVR,2)$ in Eq. (20) is given by

$$\underbrace{\begin{pmatrix} 0 & a_2 & 0 & 0 \\ a_2 & 0 & a_2 & 0 \\ 0 & a_2 & 0 & a_2 \\ 0 & 0 & a_2 & 0 \end{pmatrix}}_{\mathrm{diag}(DVR,2)} = \underbrace{\begin{pmatrix} 0 & a_2 & 0 & 0 \\ a_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_2 \\ 0 & 0 & a_2 & 0 \end{pmatrix}}_{\mathrm{diag}(DVR,2)_1} + \underbrace{\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & a_2 & 0 \\ 0 & a_2 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}}_{\mathrm{diag}(DVR,2)_2}. \tag{22}$$

Our strategy is to perform time evolution under each individual 1-sparse matrix using the methods for sparse Hamiltonian simulation developed by Aharonov and Ta-Shma [3]. To use this technique, we define a few oracles.

Active Subspace Oracle, $O_{\mathcal{S}(\nu,\sigma)}$: Let $\mathcal{H}$ be the Hilbert space that $|x\rangle$ lives in. For a given 1-sparse matrix $\mathrm{diag}(DVR,\nu)_\sigma$, let $\mathcal{S}(\nu,\sigma)$ be the subspace of $\mathcal{H}$ that $\mathrm{diag}(DVR,\nu)_\sigma$ acts on. We refer to $\mathcal{S}(\nu,\sigma)$ as the active subspace. We also define the inactive subspace $\bar{\mathcal{S}}(\nu,\sigma) = \mathcal{H} - \mathcal{S}(\nu,\sigma)$. This is the subspace that $(DVR,\nu)_\sigma$ has no action on. The active subspace oracle has following action:

$$O_{\mathcal{S}(\nu,\sigma)}|x\rangle|0\rangle = \begin{cases} |x\rangle|0\rangle, & \text{if } |x\rangle \notin \mathcal{S}(\nu,\sigma) \\ |x\rangle|1\rangle, & \text{if } |x\rangle \in \mathcal{S}(\nu,\sigma). \end{cases} \tag{23}$$

Thus, $O_{\mathcal{S}(\nu,\sigma)}$ flags whether or not $|x\rangle$ is in the active subspace.

The active subspace is easily found by identifying all rows that contain non-zero elements of a given $\mathrm{diag}(DVR,\nu)_\sigma$. Examples of active and inactive subspaces for several matrices are illustrated in Fig. 11.

Conjugate Pair Oracle, $O_{f(x)}$: Because there is only one non-zero matrix element for every row and column in a 1-sparse matrix, we can always determine the column index given the row index. Let $[\mathrm{diag}(DVR,\nu)_\sigma]_{xf(x)}$ be the matrix element at the xth row and $f(x)$th column of $\mathrm{diag}(DVR,\nu)_\sigma$. We refer to x and $f(x)$ as conjugate pairs. For a given 1-sparse matrix and row index x, $O_{f(x)}$ has the following action:

$$O_{f(x)}|x\rangle|0\rangle = |x\rangle|f(x)\rangle. \tag{24}$$

Thus, $O_{f(x)}$ acts on a row index and returns the conjugate pair into a new register. We note that for any $\mathrm{diag}(DVR,\nu)_\sigma$, the conjugate pair is given by

$$f(x) = \begin{cases} x-\nu+1, & \text{if upper diagonal} \\ x+\nu-1, & \text{if lower diagonal.} \end{cases} \tag{25}$$

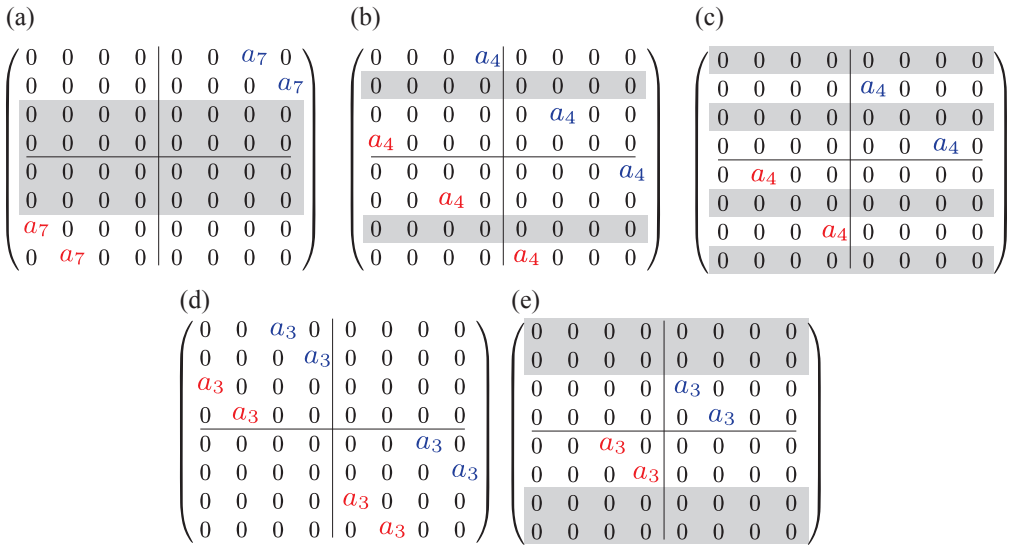


FIG. 11: Subspace characterization for $D = 8$. Grey rows indicate the inactive subspace of a given $\text{diag}(DVR, \nu)$; all other rows belong to the active subspace. (a) $\nu = 7$. (b) $\nu = 4$, $\sigma = 1$. (c) $\nu = 4$, $\sigma = 2$. (d) $\nu = 3$, $\sigma = 1$. (e) $\nu = 3$, $\sigma = 2$.

Comparator Oracle O_{CMP}: Given two n-qubit computational basis states, $|x\rangle$ and$|y\rangle$, O_{CMP} computes whether $x \geq y$ and stores the result in a new register. That is

$$O_{\text{CMP}}|x\rangle|y\rangle|0\rangle = |x\rangle|y\rangle|x \geq y\rangle. \tag{26}$$

We note that O_{CMP} is different than $O_{>}(a)$ defined in SI Section IV because O_{CMP} compares two computational basis states, while $O_{>}(a)$ compares one computational basis state to a constant a.

Given access to these oracles, the circuit that performs time evolution under $\text{diag}(DVR, \nu)_\sigma$ is given in Fig. 12 [2]. Here, $R_x(\theta)$ is a rotation about the x-axis of the Bloch sphere by an angle θ. We note that a $CR_x(\theta)$ gate is easily implemented by conjugating $CR_z(\theta)$ with Hadamard gates.

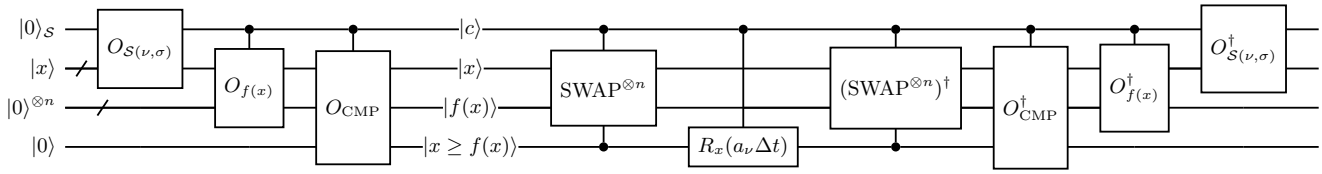


FIG. 12: Circuit that implements time evolution under $\text{diag}(DVR, \nu)_\sigma$ for an amount of time Δt.

Let the circuit in Fig. 12 be denoted by $U_{DVR}(\nu, \sigma)$. Let $U_{DVR}(l)$ be the circuit that approximates time evolution under DVR using l diagonals. This circuit is given by

$$U_{DVR}(l) \approx \prod_{\nu=1}^{l} \prod_{\sigma=1}^{2} U_{DVR}(\nu, \sigma). \tag{27}$$

In Section III, we classified $\text{diag}(DVR, \nu)$ into four different cases based on the structure of each matrix. We now demonstrate how to construct $O_{S(\nu,\sigma)}$, $O_{f(x)}$, and O_{CMP} for each of these cases.

A. Case 1: $\nu = 1$

We do not need to consider time evolution under $\text{diag}(DVR, \nu)$ because it only contributes a global phase.

B. Case 2: $\nu > D/2$

In this case, $\text{diag}(DVR, \nu)$ is automatically 1-sparse and does not need to be decomposed any further. Therefore, we define $\text{diag}(DVR, \nu)_1 = \text{diag}(DVR, \nu)$ and $\text{diag}(DVR, \nu)_1 = I$ when $\nu > D/2$.

To build the active subspace oracle, we first note from Fig. 11 (a) that that the inactive subspace is always a set of rows in the center of the matrix. The size of the inactive subspace is given by

$$|\bar{\mathcal{S}}(\nu, 1)| = 2(\nu - D/2 - 1). \tag{28}$$

If $|\bar{\mathcal{S}}(\nu, 1)| = 0$, then $\mathcal{S}(\nu, 1) = \mathcal{H}$. If this is the case, then we do not need $O_{\mathcal{S}(\nu,1)}$ and $U_{DVR}(\nu, 1)$ can be implemented without using the control qubit on the top wire of Fig. 12. Otherwise, the elements of the active subspace are given by

$$\mathcal{S}(\nu, 1) = \{|x\rangle : x < A_{\text{up}} \text{ or } A_{\text{low}} > x\}, \tag{29}$$

where $A_{\text{up}} = \frac{D}{2} - \frac{|\bar{\mathcal{S}}|}{2}$ and $A_{\text{low}} = \frac{D}{2} + \frac{|\bar{\mathcal{S}}(\nu,1)|}{2}$. We can construct the active subspace oracle as

$$O_{S(\nu,1)} = O_<(A_{\text{up}})\, O_>(A_{\text{low}}). \tag{30}$$

A circuit the implements $O_{S\nu,1)}$ is given in Fig. 13.

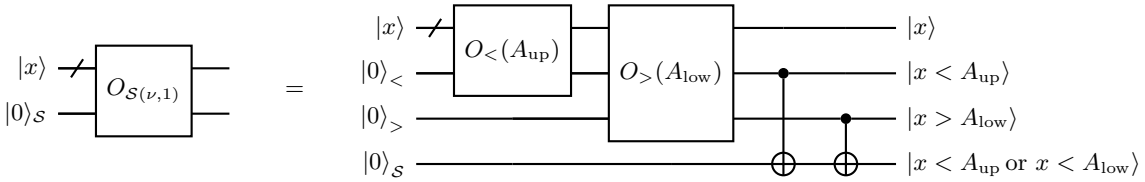


FIG. 13: Active subspace oracle for $\nu > D/2$. Here, $O_<(A_{\text{up}})$ acts on the first and second registers, and $O_>(A_{\text{low}})$ acts on the first and third registers. The bottom register flags the active subspace.

To implement $O_{f(x)}$ we need a way to determine if x is on the upper or lower diagonal. For the case $\nu > D/2$, if $x < A_{\text{up}}$, it belongs to the upper diagonal, and if $x > A_{\text{low}}$, it belongs to the lower diagonal. Thus, the second and third registers of Fig. 13 flag the upper and lower diagonals.

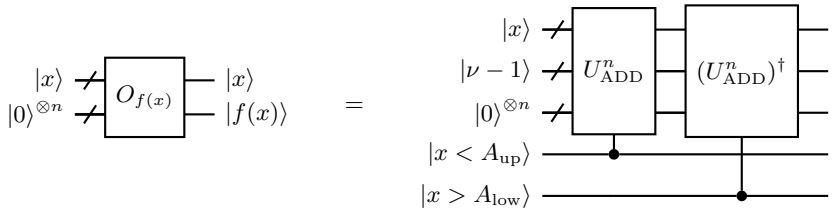


FIG. 14: Conjugate pair oracle for $\nu > D/2$. Here, $(U^n_{\text{ADD}})^\dagger$ is used to perform integer subtraction. The second register of the circuit on the right stores the value of $\nu - 1$ that is either added or subtracted to $|x\rangle$.

We do not need to construct a comparator oracle for the present case because the information returned by the comparator is already given by the $|x < A_{\text{up}}\rangle$ and $|x > A_{\text{low}}\rangle$ states. If $|x\rangle$ belongs to the $|x < A_{\text{up}}\rangle$ branch, then we know from Eq. (25) that $|x \geq f(x)\rangle$ is false. Similarly, if $|x\rangle$ belongs to the $|x < A_{\text{low}}\rangle$ branch, we know $|x \geq f(x)\rangle$ is true. Thus, instead of implementing a comparator oracle, we condition the SWAP gate in Fig. 12 on the $|x < A_{\text{up}}\rangle$ and $|x > A_{\text{low}}\rangle$ states.

Using Section IV, we estimate the resources needed to implement U_{DVR} and give the results in Table II. We note that the CNOT Count is a lower bound estimate because we do not know the constant factor for performing addition using Remaud and Vandaele's method [2].

	Ancilla	CNOT Count	Depth
U_{DVR}	$2n+3$	$12n\log n + 36n + 34$	$36n+34$
C^2U_{DVR}	$2n+5$	$12n\log n + 36n + 86$	$36nn+86$

TABLE II: Lower bound resource estimates to implement U_{DVR} and C^2U_{DVR} for Case 2.

C. Case 3a: $1 < \nu \leq D/2$, and ν is even

For this case, $\text{diag}(DVR,\nu)$ is 2-sparse and is decomposed into a sum of 1-sparse matrices using the method in SI Section III. From Fig. 11 (b) and (c) we see that there is always a region of $\mathcal{S}$ coming from the center of the matrix. Additionally, $\mathcal{S}$ contains every-other row before and after the center region.

For $\text{diag}(DVR,\nu)_1$, the active subspace is given by

$$\mathcal{S}(\nu,1) = \mathcal{P}_1(\nu,1) \cup \mathcal{P}_2(\nu,1) \cup \mathcal{P}_3(\nu,1), \tag{31}$$

where

$$\begin{aligned}
\mathcal{P}_1(\nu,1) &= \{|x\rangle \mid \nu - 2 \leq x \leq D - \nu + 1\} \\
\mathcal{P}_2(\nu,1) &= \{|x\rangle \mid x < \nu - 2, \text{ and } x \text{ is even}\} \\
\mathcal{P}_3(\nu,1)) &= \{|x\rangle \mid D - \nu + 1 > x, \text{ and } x \text{ is odd}\}.
\end{aligned} \tag{32}$$

A circuit that implements $O_{\mathcal{S}(\nu,1)}$ is given in Fig. 15. We note that the first two elements of this circuit identify $\mathcal{P}_2(\nu,1)$, the next two elements identify $\mathcal{P}_3(\nu,1)$, and the final element identifies $\mathcal{P}_1(\nu,1)$.

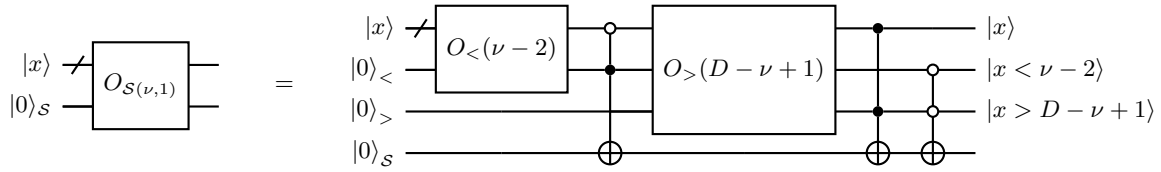


FIG. 15: Active subspace oracle for $1 < \nu \leq D/2$, ν is even, and $\sigma = 1$. Here, $O_{<}(\nu-2)$ acts on the first and second registers. The second register contains the result $|x < \nu - 2\rangle$. The second gate performs an X operation on the fourth register only if $x < \nu - 2$ and x is even is true. The next gate $O_{>}(D-\nu+1)$ acts on the first and third registers. The third register contains the result $|x > D - \nu + 1\rangle$. The fourth gate performs an X operation on the fourth register only if $x > D - \nu + 1$ and x is odd is true. Finally, the last gate applies X to the fourth register only if $\nu - 2 \leq x \leq D - \nu + 1$.

For $\text{diag}(DVR,\nu)_2$, the active subspace is

$$\mathcal{S}(\nu,2) = \mathcal{P}_1(\nu,2) \cup \mathcal{P}_2(\nu,2) \cup \mathcal{P}_3(\nu,2), \tag{33}$$

where

$$\begin{aligned}
\mathcal{P}_1(\nu,2) &= \{|x\rangle \mid \nu - 1 \leq x \leq D - v\} \\
\mathcal{P}_2(\nu,2) &= \{|x\rangle \mid x < \nu - 1, \text{ and } x \text{ is odd}\} \\
\mathcal{P}_3(\nu,2) &= \{|x\rangle \mid D - \nu > x, \text{ and } x \text{ is even}\}.
\end{aligned} \tag{34}$$

A circuit that implements $O_{\mathcal{S}(\nu,2)}$ is given in Fig. 16.

We can construct $O_{f(x)}$ by determining if $|x\rangle$ belongs to the upper or lower diagonal of $\text{diag}(DVR,\nu)_\sigma$. For $\sigma = 1$, the upper and lower diagonal elements correspond to $|x\rangle$ being even or odd, respectively. For $\sigma = 2$, the upper and lower diagonal elements correspond to $|x\rangle$ being odd or even, respectively. A circuit for the conjugate pair oracle for $\sigma = 1$ is given in Fig. 17. The circuit that implements the the conjugate pair oracle for $\sigma = 1$ is exactly the same as that in Fig. 17, except U^n_{ADD} is controlled by the odd register and $(U^n_{\text{ADD}})^\dagger$ is controlled by the even register.

We do not need to explicitly construct a comparator oracle for this case. This is because we can determine if $x \geq f(x)$ from the even and odd registers in Fig. 17. For $\sigma = 1$, if the even(odd) register is $|1\rangle$, then $x \geq (f)$ is false(true). For $\sigma = 2$, if the even(odd) register is $|1\rangle$, then $x \geq (f)$ is true(false). Hence, the CSWAP operations are implemented by controlling on the appropriate even(odd) registers. We provide resource estimate lower bounds for U_{DVR} for Case 3a in Table III.

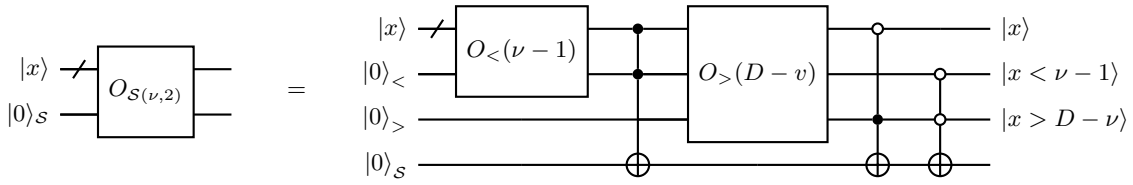


FIG. 16: Active subspace oracle for $1 < \nu \leq D/2$, ν is even, and $\sigma = 2$. Here, $O_{<}(\nu-1)$ acts on the first and second registers. The second register contains the result $|x < \nu-1\rangle$. The second gate performs an X operation on the fourth register only if $x < \nu - 1$ and x is odd is true. The next gate $O_{>}(D-\nu)$ acts on the first and third registers. The third register contains the result $|x > D-\nu\rangle$. The fourth gate performs an X operation on the fourth register only if $x > D-\nu$ and x is even is true. Finally, the last gate applies X to the fourth register only if $\nu - 1 \leq x \leq D-\nu$.

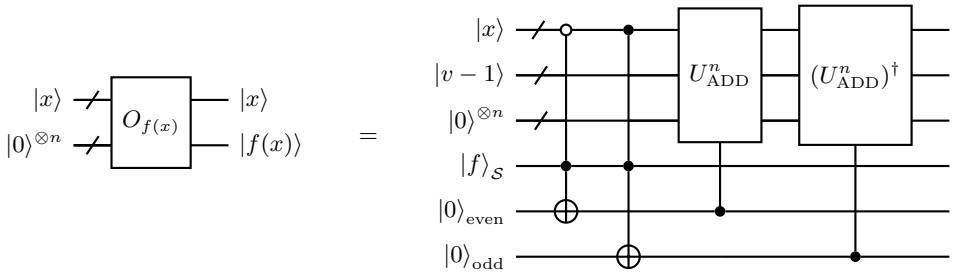


FIG. 17: Conjugate pair oracle for $1 < \nu \leq D/2$, ν is even and $\sigma = 1$. Here, $|f\rangle_{\mathcal{S}}$ is a register that flags if $|x\rangle \in \mathcal{S}(\nu, 1)$ is true. The bottom two registers track if x is an even or odd integer.

D. Case 3b: $1 < \nu \leq D/2$, and ν is odd

For this case, $\text{diag}(DVR, \nu)$ is also 2-sparse and is decomposed into a sum of 1-sparse matrices using SI Section III. We illustrate $\mathcal{S}$ in Fig. 11 (d) and (e). In Fig. 11 (d), $\mathcal{S} = \mathcal{H}$. In Fig. 11 (e), the rows in the center of the matrix are part of $\mathcal{S}$, while 2^p rows at the top and bottom of the matrix are part of $\bar{\mathcal{S}}$. Here, p is a integer such that following is true:

$$b = \frac{\nu - 1}{2^p}. \tag{35}$$

where b must be an odd integer. In general, $\bar{S}$ will always be a pattern containing blocks of 2^p rows (see SI Section III.

To construct the active subspace oracle for $\sigma = 1$, we first note that if $b = 1$, then it turns out that $\mathcal{S}(\nu, 1) = \mathcal{H}$. In this situation, we do not need to apply the active subspace oracle. For $b \neq 1$, the active subspace is given by

$$\mathcal{S}(\nu, 1) = \mathcal{P}_1(\nu, 1) \cup \mathcal{P}_2(\nu, 1) \cup \mathcal{P}_3(\nu, 1), \tag{36}$$

where

$$\begin{aligned}
\mathcal{P}_1(\nu, 1) &= \{|x\rangle \mid A_{\text{up}} \leq x \leq A_{\text{low}}\} \\
\mathcal{P}_2(\nu, 1) &= \{|x\rangle \mid n2^p \leq x < (n+1)2^p \text{ and } x < A_{\text{up}}, \text{ for } n \in \mathbb{Z}, \text{is even}\} \\
\mathcal{P}_3(\nu, 1) &= \{|x\rangle \mid n2^p \leq x < (n+1)2^p \text{ and } A_{\text{low}} < x, \text{ for } n \in \mathbb{Z}, \text{is odd}\},
\end{aligned} \tag{37}$$

where $A_{\text{up}} = (b-1)2^p$ and $A_{\text{low}} = D - (b-1)2^p - 1$. A circuit diagram that implements $O_{\mathcal{S}(\nu,1)}$ is given in Fig. 18. Note that we are able to selectively apply operations to a set of computational basis states of size 2^p by using the pth bit of $|x\rangle$ as a control.

For $\sigma = 2$, if $b = 1$, the active subspace is given by

$$\mathcal{S}(\nu, 2) = \mathcal{P}_4(\nu) \tag{38}$$

where

$$\mathcal{P}_4(\nu) = \{|x\rangle \mid B_{\text{up}} < x < B_{\text{low}}\}. \tag{39}$$

	Ancilla	CNOT Count	Depth
U_{DVR}	$2n+5$	$12n\log n + 36n + 60$	$36n+60$
C^2U_{DVR}	$2n+9$	$12n\log n + 36n + 196$	$36n+196$

TABLE III: Lower bound resource estimates to implement U_{DVR} and C^2U_{DVR} for Case 3a. The cost is the same for both $\sigma = 1$ and $\sigma = 2$

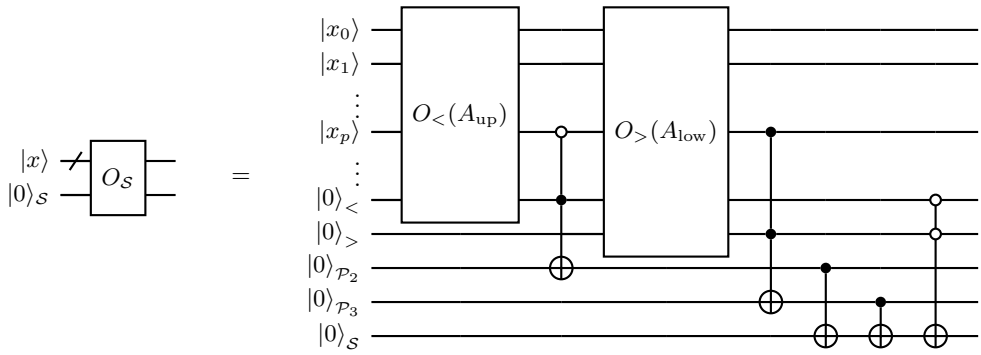


FIG. 18: Active subspace oracle for $1 < \nu \leq D/2$, ν is odd, and $\sigma = 1$. Here, $O_<(A_{\rm up})$ acts on the first $n+1$ registers, and $O_>(A_{\rm low})$ acts on the first n and $(n+2)$th registers. The first two gates determine if $|x\rangle \in \mathcal{P}_2(\nu, 1)$ and the next two gates determine if $|x\rangle \in \mathcal{P}_3(\nu, 1)$. The remaining three gates determine if $|x\rangle \in \mathcal{S}(\nu, 1)$.

where $B_{\rm u} = 2^p - 1$ and $B_{\rm l} = D - 2^p$. If $b \neq 1$, then the active subspace is given by

$$\mathcal{S}(\nu, 2) = (\mathcal{P}_1(\nu, 2) \cup \mathcal{P}_2(\nu, 2) \cup \mathcal{P}_3(\nu, 2)) \cap \mathcal{P}_4(\nu), \tag{40}$$

where

$$\begin{aligned}
\mathcal{P}_1(\nu, 2) &= \{|x\rangle \mid F_{\rm up} \leq x \leq F_{\rm low}\} \\
\mathcal{P}_2(\nu, 2) &= \{|x\rangle \mid n2^p \leq x < (n+1)2^p \text{ and } x < F_{\rm up}, \text{ for } n \in \mathbb{Z}, \text{is odd}\} \\
\mathcal{P}_3(\nu, 2) &= \{|x\rangle \mid n2^p \leq x < (n+1)2^p \text{ and } F_{\rm low} > x, \text{ for } n \in \mathbb{Z}, \text{is even}\}
\end{aligned} \tag{41}$$

where $F_{\rm up} = b2^p$ and $F_{\rm low} = D - b2^p - 1$. A circuit diagram for $O_{\mathcal{S}(\nu,2)}$ is given in Fig. 19.

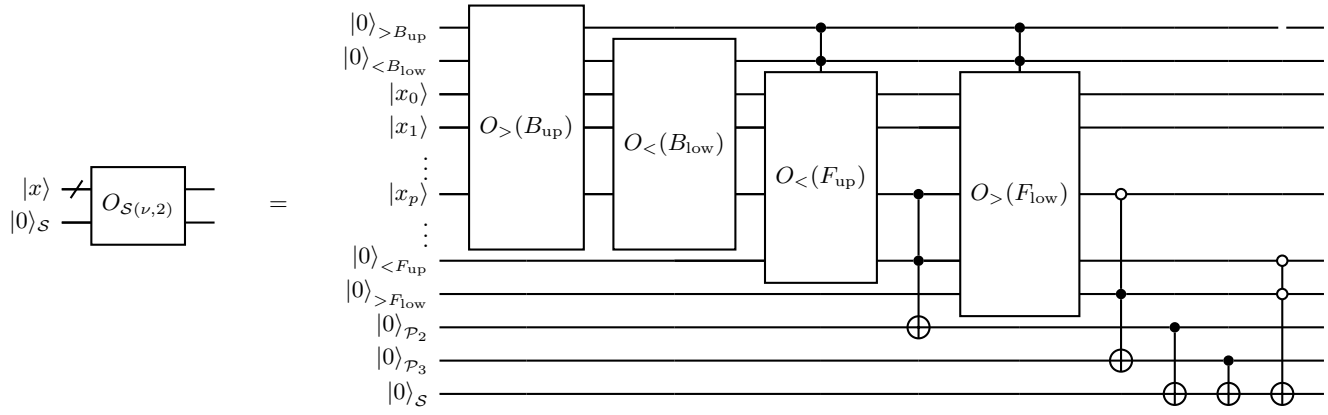


FIG. 19: Active subspace oracle for $1 < \nu \leq D/2$, ν is odd, and $\sigma = 2$. The first two gates identify the $\mathcal{P}_4$ subspace. The remainder of the circuit is the same as for $O_{\mathcal{S}(\nu,1)}$, except that the $O_<(F_{\rm up})$ and $O_<(F_{\rm low})$ gates are controlled by the top two register. This ensures that $|x\rangle$ remains within $\mathcal{P}_4$ throughout the circuit.

For $\sigma = 1$ the conjugate pair oracle is given by Fig. 20. The conjugate pair oracle for $\sigma = 2$ is the same as for $\sigma = 1$, except that $U^n_{\rm ADD}$ is controlled from the odd register and $(U^n_{\rm ADD})^\dagger$ is controlled from the even register .

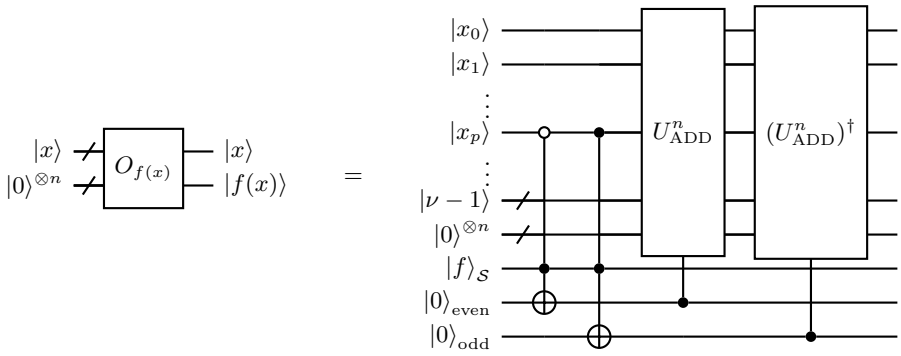


FIG. 20: Conjugate pair oracle for $1 < \nu \leq D/2$, ν is odd and $\sigma = 1$. Here, $|f\rangle_{\mathcal{S}}$ is a register that flags if $|x\rangle \in \mathcal{S}(\nu, 1)$ is true. The bottom two registers track if the pth bit of $|x\rangle$ is an even or odd integer.

Once again, the information returned by comparator oracle is given from the even and odd registers in Fig. 20 and we do not need to construct O_{CMP}. For $\sigma = 1$, if the even register is in the $|1\rangle$ state, then $x \geq f(x)$ is false, and if the odd register is in the $|1\rangle$ state, then $x \geq f(x)$ is true. For $\sigma = 2$, the situation is the reverse. If the even register is in the $|1\rangle$ state, then $x \geq f(x)$ is true, and if the odd register is in the $|1\rangle$ state, then $x \geq f(x)$ is false.

We provide a resource estimate lower bound to implement U_{DVR} for $\sigma = 1$ and $\sigma = 2$ in Table IV.

	Ancilla	CNOT Count	Depth
$U_{DVR}(\sigma = 1)$	$2n + 7$	$12n \log n + 36n + 84$	$36n + 84$
$C^2U_{DVR}(\sigma = 1$	$2n + 10$	$12n \log n + 36n + 344$	$36n + 344$
$U_{DVR}(\sigma = 2)$	$2n + 9$	$16n \log n + 36n + 132$	$36n + 132$
$C^2U_{DVR}(\sigma = 2$	$2n + 12$	$16n \log n + 36n + 358$	$36n + 358$

TABLE IV: Lower bound resource estimates to implement U_{DVR} and C^2U_{DVR} for Case 3b.

VI. DOUBLE-WELL COST ESTIMATION

In this section, we estimate the cost to perform time evolution under a double-well potential given by

$$V(x) = ax^2 + bx^4, \tag{42}$$

where $a, b \in \mathbb{R}$. The time evolution operator of $V(x)$ for a single time step Δt is given by

$$e^{-iV(x)\Delta t} = e^{-iax^2\Delta t}e^{-ibx^4\Delta t}. \tag{43}$$

The quadratic term can be simulated with a circuit depth and CNOT gate count of $O(n^2)$ [4, 5]. We now briefly review how this circuit is constructed.

For a given integer l, its binary expansion using n bits is

$$l = \sum_{j=0}^{n-1} k_j 2^j, \tag{44}$$

where $k_j \in \{0, 1\}$. Recall that the value of the lth grid point in position space is given by

$$x_l = \Delta x \sum_{j=1}^{n-1} \left(k_j 2^j + \alpha\right) \tag{45}$$

where $\Delta x = L/D$, and $\alpha = -D/2$. Inserting this expansion into the quadratic term yields

$$e^{-iax^2\Delta t} = \prod_{j,p=0}^{n-1} e^{-i\delta\left(k_j 2^j + \alpha\right)\left(k_p 2^p + \alpha\right)}, \tag{46}$$

where $\delta = a\Delta t$. Expanding within the exponential and grouping terms by order of k gives

$$e^{-iV(x)\Delta t} = T_0 T_1 T_2, \tag{47}$$

where

$$\begin{aligned} T_0 &= e^{-i\delta\alpha^2} \\ T_1 &= \prod_{j=0}^{n-1} e^{-i2\delta\alpha k_j 2^j} \\ T_2 &= \prod_{j,p=0}^{n-1} e^{-i\delta\alpha k_j 2^j \cdot k_p 2^p}. \end{aligned} \tag{48}$$

We can implement T_0 with two rotation gates, and T_1 with n rotation gates [5]. Implementing T_2 requires $n^2 - n$ controlled rotation gates and n rotation gates, but we approximate this cost as n^2 controlled rotation gates [5].

To time evolve the quartic operator, we begin with

$$e^{-ibx^4\Delta t} = \prod_{j_1,j_2,j_3,j_4=0}^{n-1} e^{-i\delta\left(k_{j_1}2^{j_1}+\alpha\right)\left(k_{j_2}2^{j_2}+\alpha\right)\left(k_{j_3}2^{j_3}+\alpha\right)\left(k_{j_4}2^{j_4}+\alpha\right)}, \tag{49}$$

where now $\delta = b\Delta t$, and expand the exponential. Grouping terms by order of k yields

$$e^{-ibx^4\Delta t} = T_0 T_1 T_2 T_3 T_4, \tag{50}$$

where

$$\begin{aligned} T_0 &= e^{-i\delta\alpha^2} \\ T_1 &= \prod_{j_1=0}^{n-1} e^{-i4\delta\alpha^3 k_{j_1} 2^{j_1}} \\ T_2 &= \prod_{j_1,j_2=0}^{n-1} e^{-i6\delta\alpha^2 k_{j_1} 2^{j_1} \cdot k_{j_2} 2^{j_2}} \\ T_3 &= \prod_{j_1,j_2,j_3=0}^{n-1} e^{-i4\delta\alpha k_{j_1} 2^{j_1} \cdot k_{j_2} 2^{j_2} \cdot k_{j_3} 2^{j_3}} \\ T_4 &= \prod_{j_1,j_2,j_3,j_4=0}^{n-1} e^{-i\delta\alpha k_{j_1} 2^{j_1} \cdot k_{j_2} 2^{j_2} \cdot k_{j_3} 2^{j_3} \cdot k_{j_4} 2^{j_4}}. \end{aligned} \tag{51}$$

Extending the circuit used to time evolve quadratic potentials, we find that the T_3 and T_4 operators can be implemented with less than n^3 doubly controlled rotation gates and n^4 triply controlled rotation gates, respectively [5]. Because every term in $e^{-iax^2\Delta t}$ commutes with every term in $e^{-ibx^4\Delta t}$, we can group the constant, linear, and quadratic terms together.

Let U_V be the unitary that performs time evolution under $V(x)$. In our proposed circuit to compute TEO matrix elements, we use controlled and doubly controlled version of U_V. The cost of implementing controlled versions of U_V can be determined using Section IV, and are given in Table V.

	Ancilla	CNOT Count/Depth
CU_V	4	$50n^4 + 38n^3 + 26n^2 + 14n + 4$
C^2U_V	5	$50n^4 + 38n^3 + 26n^2 + 14n + 16$

TABLE V: Resource upper bound estimates to implement CU_V and CU_V.

[1] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*, Cambridge University Press, 2010.

[2] M. Remaud and V. Vandaele, Reversible Computation, Cham, 2025, pp. 137–154.
[3] D. Aharonov and A. Ta-Shma, Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing, 2003, p. 20–29.
[4] G. Beneti and G. Strini, *Am. J. Phys.*, 2008, **76**, 657–662.
[5] P. J. Ollitrault, G. Mazzola and I. Tavernelli, *Phys. Rev. Lett.*, 2020, **125**, 260511.