

Quantum Discrete Variable Representations

Szymon Pliś and Emil Zak

BEIT sp. z o.o., Mogilska 43, 31-545 Kraków, Poland

We present a fault-tolerant quantum algorithm for implementing the Discrete Variable Representation (DVR) transformation, a technique widely used in simulations of quantum-mechanical Hamiltonians. DVR provides a diagonal representation of local operators and enables sparse Hamiltonian structures, making it a powerful alternative to the finite basis representation (FBR), particularly in high-dimensional problems. While DVR has been extensively used in classical simulations, its quantum implementation, particularly using Gaussian quadrature grids, remains underexplored. We develop a quantum circuit that efficiently transforms FBR into DVR by following a recursive construction based on quantum arithmetic operations, and we compare this approach with methods that directly load DVR matrix elements using quantum read-only memory (QROM). We analyze the quantum resources, including T-gate and qubit counts, required for implementing the DVR unitary and discuss preferable choices of QROM-based and recursive-based methods for a given matrix size and precision. This study lays the groundwork for utilizing DVR Hamiltonians in quantum algorithms such as quantum phase estimation with block encoding.

1 Introduction

The Discrete Variable Representation (DVR) technique [1, 2, 3, 4] formulates partial differential equations using a localized basis associated with an underlying spatial grid. DVR is constructed as a linear combination of variational basis functions such that, in the spatial representation, each basis function is localized around the grid point it is associated with.

DVRs are widely used in computational physics, simulations, and engineering, with successful applications in nuclear motion theory [5, 6, 7], scattering calculations [8], nuclear physics [9], electronic structure theory [10], attosecond physics [11], and quantum circuit simulations [12].

In this work, we focus on Gaussian discrete variable representations in the context of quantum simulations of Hamiltonians. A commonly used approach for solving the associated Schrödinger equation is the finite basis representation (FBR), in which the Hamiltonian operator is represented in a truncated variational basis, and matrix elements are evaluated via quadrature [4]. The DVR can be obtained via a unitary transformation of the FBR and offers several advantages.

First, DVR yields diagonal representations for local quantum-mechanical operators such as multi-dimensional potential energy surfaces (PES) [4]. For this reason, DVR is

Szymon Pliś: szymonzdziawplis@gmail.com

Emil Zak: emil@beit.tech, <https://www.beit.tech>

often preferred over variational bases, particularly when evaluating many-body potential energy matrix elements is computationally intensive. Typically, the kinetic energy operator, being a second-order differential operator, has a simpler structure than the potential. When expressed in a DVR basis, the potential energy becomes a diagonal matrix with entries corresponding to the potential evaluated at DVR grid points. The kinetic energy operator is first evaluated analytically in a variational basis and then transformed to the DVR basis. Such an approach often results in sparse Hamiltonian matrices, making DVR especially valuable in molecular physics, where the curse of dimensionality imposes severe computational limits. Another notable advantage of DVR is the ability to selectively prune basis functions based on physical insight, thanks to their spatial localization. Finally, in contrast to finite-difference grid-based methods, DVRs exhibit exponential convergence in accuracy of approximating bound states of many-body Hamiltonians [13].

For multi-dimensional problems, DVR can be extended using a direct-product basis [4]. On classical computers, the memory required for this transformation scales exponentially with the number of dimensions, as N^D , where N is the number of basis functions per dimension and D is the dimensionality. Quantum computers however can reduce this memory scaling to $\mathcal{O}(ND)$. Beyond memory advantages, DVR may also offer quantum-computational speedups due to the sparsity of the DVR Hamiltonian [14]. We therefore argue that, given an efficient circuit implementation of the FBR-to-DVR transformation, Hamiltonian simulation problems can be solved on quantum computers at a lower gate cost in the DVR basis than in the FBR basis.

A particularly appealing application of DVR is in computing Hamiltonian eigenvalues. In this context, fault-tolerant quantum algorithms such as Quantum Phase Estimation (QPE) often rely on block encoding of the Hamiltonian [15, 16, 17, 18, 19]. The norm of the Hamiltonian, which proportionally scales the T-gate complexity of QPE, can be reduced compared to variational methods through appropriate truncations, such as removing irrelevant DVR grid points. Moreover, the Hamiltonian can often be decomposed into a diagonal PES and a structured kinetic energy operator, where the diagonal part can be efficiently block-encoded using standard techniques [14, 20, 21].

Several quantum algorithms have been developed for grid-based physical simulations [22, 23, 24, 25] and for solving differential equations [26, 27], with applications in electronic structure [28, 25] and nuclear motion calculations [19, 24, 29]. A quantum algorithm for wavelet-based representations was also proposed in Ref. [30]. For Hamiltonian simulation in general, the widely used quantum Fourier-transformed representation based on equidistant grids provides several advantages [28, 19]. However, quantum simulations using non-equidistant grids, particularly those defined by Gaussian quadrature, which allows exact integration of polynomials up to a given degree, have not yet been explored in detail. In particular, a quantum circuit implementing the FBR-to-DVR unitary transformation could enable more efficient simulations in DVR.

Below we present a quantum algorithm for implementing the FBR-to-DVR transformation and for loading DVR matrix elements into qubit states. In Section 2, we review the fundamentals of Gaussian DVRs, which are used throughout the paper. In Section 3.1, we describe the construction of a quantum DVR oracle used to implement the FBR-to-DVR transformation, which is detailed in Section 4. For each construction, we estimate fault-tolerant quantum resources, including T-gate counts, circuit depth, and qubit requirements. Finally, we briefly discuss potential applications of DVRs on quantum devices.

2 Gaussian DVRs

Gaussian DVR matrix elements can be defined as:

$$T_{pq} = N_q \sqrt{w_p} p_q(x_p), \text{ for } p, q = 0, \dots, N-1 \quad (1)$$

where $N_q = \|p_q\|_{L^2(\mu)}^{-1}$ is the normalization constant, w_p are the Gaussian quadrature weights, $p_q(x_p)$ is the value of the degree- q orthogonal polynomial defining the Gaussian quadrature evaluated at the Gaussian quadrature node x_p [31]. Gaussian quadrature nodes and weights are chosen such that the integral:

$$\int f d\mu \approx \sum_{k=0}^{N-1} w_k f(x_k) \quad (2)$$

is exact for f chosen as polynomials of degree less or equal $2N-1$. The Gaussian quadrature nodes x_0, x_1, \dots, x_{N-1} are chosen as zeroes of p_N . The weights are then determined from the equation:

$$w_k = \frac{a_N}{a_{N-1}} \frac{\|p_{N-1}\|_{L^2(\mu)}}{p'_N(x_k) p_{N-1}(x_k)} \geq 0, \quad k = 0, \dots, N-1,$$

Orthogonal polynomials associated with Gaussian quadratures satisfy the three-term recurrence relations:

$$p_q(x) = (a_q + b_q x) p_{q-1}(x) + c_q p_{q-2}(x) \quad (3)$$

which gives the following recurrence for the columns of the DVR matrix

$$T_{pq} = (A_q + B_q x_p) T_{pq-1} + C_q T_{pq-2}. \quad (4)$$

In eq. 4, $A_q = \frac{N_q}{N_{q-1}} a_q$, $B_q = \frac{N_q}{N_{q-1}} b_q$ and $C_q = \frac{N_q}{N_{q-2}} c_q$.

For simulations of quantum-mechanical Hamiltonians states can be represented using Finite Basis Representation (FBR), given by a linear combination of basis functions:

$$\psi(x) = \sum_{j=0}^{N-1} c_j \phi_j(x) \quad (5)$$

where the coefficients c_j are determined variationally. The DVR basis functions $d_p(x)$ can be written as a linear combination of FBR basis functions:

$$d_p(x) = \sum_{q=0}^{N-1} T_{pq} \phi_q(x) \quad (6)$$

Note that DVR functions vanish in all quadrature grid points except one, which can be shown by evaluating the DVR wavefunction at k -th grid point:

$$d_l(x_k) = \sum_{j=0}^{N-1} \sqrt{w_l \omega(x_k)} p_j(x_l) p_j(x_k). \quad (7)$$

From orthogonality of $p_j(x)$ we find

$$d_l(x_k) = \sqrt{\frac{\omega(x_k)}{w_l}} \delta_{lk} \quad (8)$$

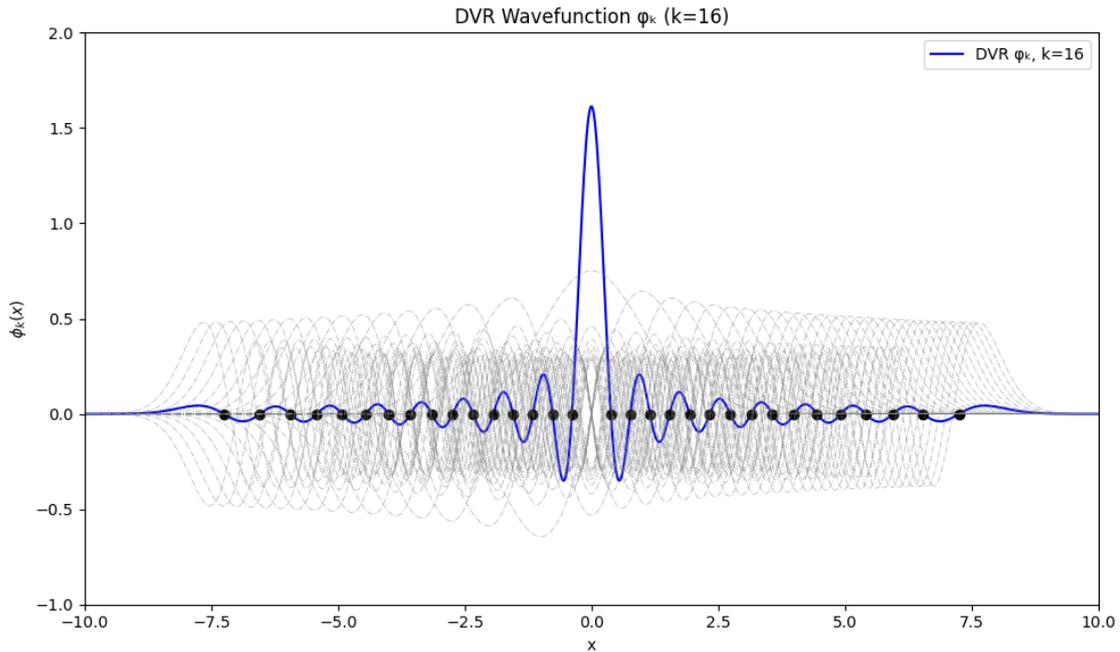


Figure 1: Example DVR basis function (blue solid line) constructed from 33 Harmonic oscillator wavefunctions (grey dashed lines). Zeros of the DVR basis function marking the Gauss-Hermite quadrature nodes are marked.

The DVR-to-FBR transformation \mathbf{T} is unitary because the Gaussian quadrature approximation to the overlap integral: $\int p_i(x)p_j(x)dx \approx (\mathbf{T}^\dagger\mathbf{T})_{ij} = \sum_{k=1}^N w_k p_i(x_k)p_j(x_k)$ is exact for i and j both smaller or equal $N - 1$, i.e. their combined degree is maximally $2N - 2$. For this reason $\mathbf{T}^\dagger\mathbf{T} = \mathbf{T}\mathbf{T}^\dagger = \mathbf{1}$. Note that the left index p in the DVR matrix given in eq.1 refers to grid-point (physical space) while the right index q refers to variational basis function index. Thus the DVR transformation connects delocalized (global) basis functions space with localized functions associated with a grid.

For solving quantum-mechanical problems, DVR is often constructed based on orthogonal polynomials that solve the Schrödinger equation for specific model systems: Hermite polynomials for the harmonic oscillator, Legendre polynomials for spherically symmetric problems, Laguerre polynomials for the hydrogen atom, associated Laguerre polynomials for the Morse oscillator, Chebyshev polynomials for a particle in a square potential well, and Lobatto polynomials for problems with fixed boundary conditions [32, 4]. An example set of DVR basis functions defined for the Hermite polynomials is shown in Figure 1. Further details about Gaussian DVRs and methods for generating them are discussed in Appendix 7.

3 Quantum DVR Oracle

3.1 Constructing the DVR oracle with the recursive method

For implementing the FBR-DVR unitary circuit we first construct a unitary, which we call the DVR oracle. The DVR oracle is defined as quantum state transformation encoding the elements of the DVR matrix T in qubit registers, written as:

$$\mathcal{T} |p\rangle |q\rangle |0\rangle = |p\rangle |q\rangle |T_{pq}\rangle \quad (9)$$

where T_{pq} are the DVR matrix elements, $|q\rangle$ is argument state representing column index (basis state index) of the DVR matrix, $|p\rangle$ is argument state representing row index (grid point index). We represent the column index as $q = wF + v$, where $w = 0, 1, 2, \dots, \frac{N}{F} - 1$ and $v = 0, 1, 2, \dots, F - 1$ for the purpose of splitting the column space in the DVR matrix into $\frac{N}{F}$ segments. Let us call F the *segmentation parameter*. The overall construction is divided into several steps, described in the following paragraphs.

Segment initialization. The construction begins with initialization unitary that loads $2\frac{N}{F}$ columns of the DVR matrix using quantum random-access memory (QROM) written as:

$$\hat{U}^{(init)}|p\rangle|w\rangle|0\rangle|z\rangle|z\rangle = |p\rangle|w\rangle|x_p\rangle|z \oplus T_{p\tilde{q}-1}\rangle|z \oplus T_{p\tilde{q}}\rangle \quad (10)$$

where $\tilde{q} = wF + \frac{F}{2}$ is the midpoint index in each of the $\frac{N}{F}$ segments dividing the DVR matrix. State $|x_p\rangle$ keeps nodes of the N -th orthogonal polynomial defining the DVR. This step requires loading $\frac{N^2}{F}$ pairs of numbers $(T_{p\tilde{q}-1}, T_{p\tilde{q}})$, given by m -bits each, leading to the T-gate cost of

$$C(\hat{U}_{init}) = 2\frac{N\sqrt{m}}{\sqrt{F}} + \sqrt{Nm} \quad (11)$$

in case of using the SELSWAP algorithm [33] and

$$C(\hat{U}_{init}) = \frac{N^2}{F} + N \quad (12)$$

for the SELECT QROM.

Recursive construction. Following the initialization step, the algorithm performs a sequence of operations $\hat{U}_{2c}, \hat{U}_{2c+1}$ controlled by the running index values $c = 1, 2, \dots, \frac{F}{4} - 1$ enumerating the range for the segmentation parameter F , as depicted in Figure 3. The recursive block of the circuit is wrapped with SWAP operations applied to two output registers, m -qubit each. The SWAP operations are controlled on the most significant bit v_0 of $v = v_02^{f-1} + v_12^{f-2} + \dots + v_{f-1}2^0$, where $v = 0, 1, 2, \dots, F - 1$ and the column index is given by $q = wF + v$. Here $F = 2^f$. The output registers keep the intermediate values in the recursive construction procedure, whereas the role of the SWAP operations is to move the output to the $|\cdot\rangle_g$ register, regardless of the parity of the column index queried. The cost of the single controlled-SWAP operation is m Toffoli gates.

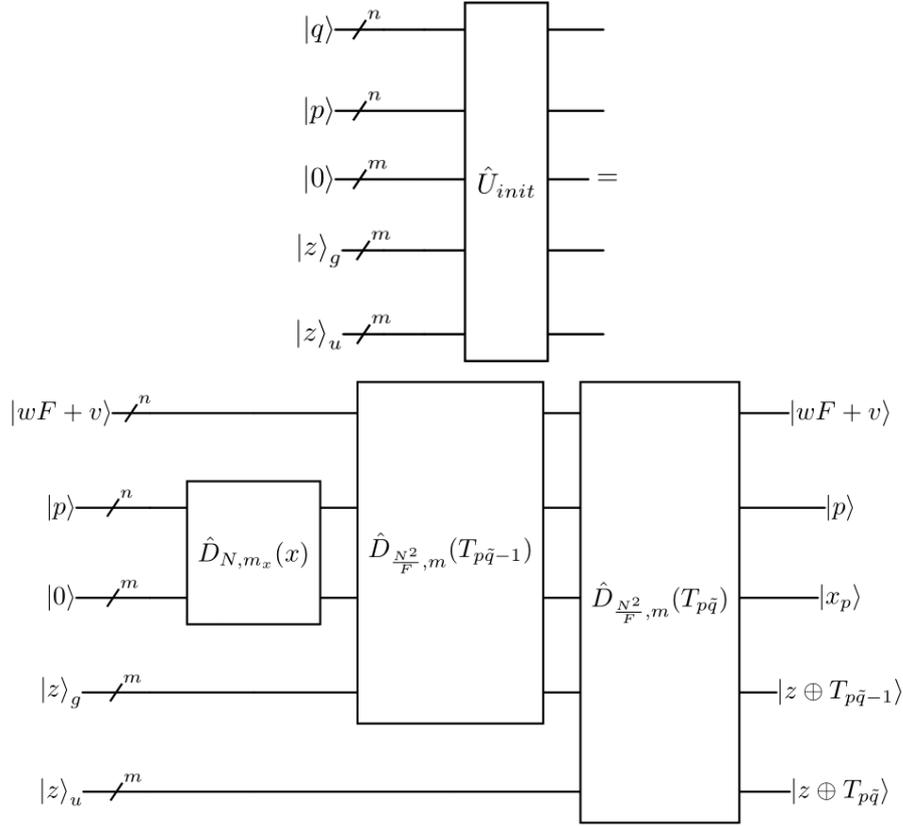


Figure 2: Quantum circuit representing initialization unitary for the DVR oracle defined in eq. 10. $\hat{D}_{N,m}(A)$ denotes QROM encoding data represented by function A for N arguments and output stored in m qubits. Here $T_{p\tilde{q}}$ denotes the DVR matrix element for the \tilde{q} 'th column, where $\tilde{q} = wF + \frac{F}{2}$. The column index q is represented as $q = wF + v$.

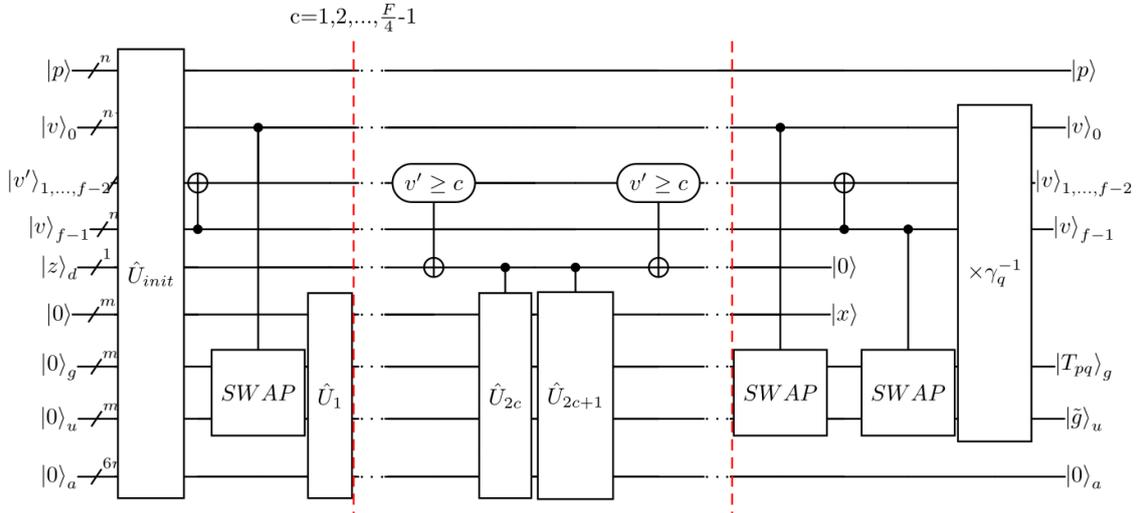


Figure 3: Quantum circuit representing the DVR oracle defined in eq. 9. \hat{U}_{init} represents state initialization oracle shown in Fig. 2. \hat{U}_c are iteration unitaries shown in Fig. 4. γ_q^{-1} is a gate multiplying the result by the appropriate scaling factor defined in eq. 16. The result is returned in m -qubit register $|T_{pq}\rangle_g$.

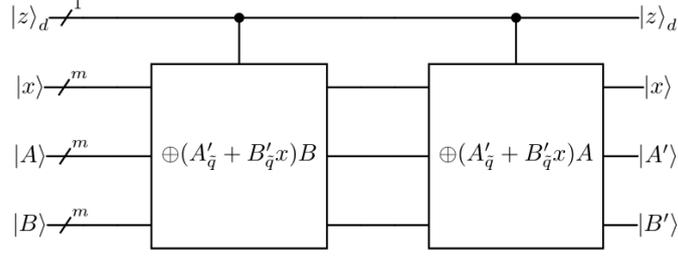


Figure 4: Quantum circuit representing \hat{U}_c iteration unitary in the the DVR oracle. $|z\rangle_d$ is the *dump* qubit register shown also in Figure 3.

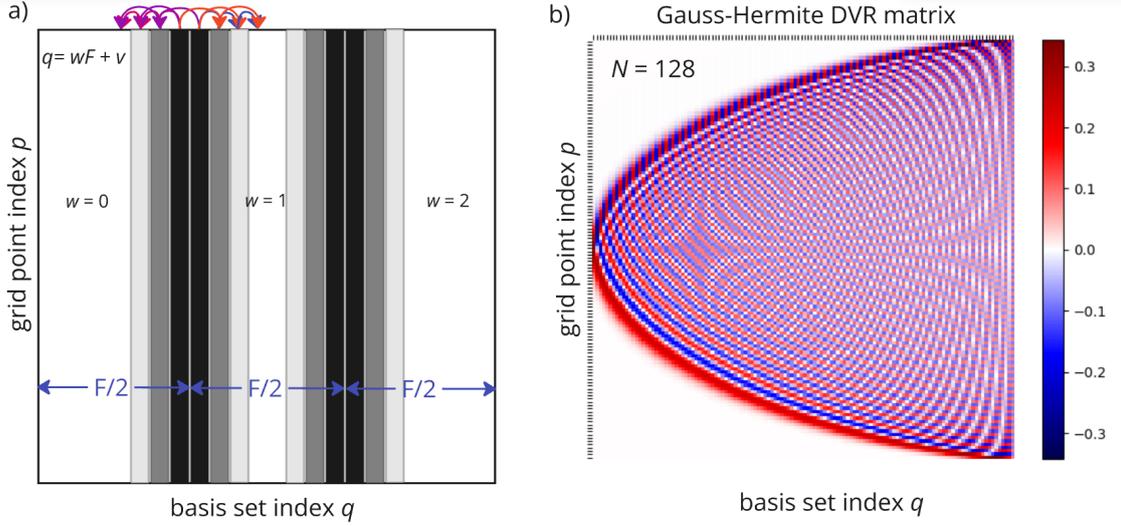


Figure 5: a) Schematic representation for the segment construction of the DVR oracle matrix. Basis set (column) index is represented as $q = wF + v$, where $w = 0, 1, 2, \dots, \frac{N}{F} - 1$ and $v = 0, 1, 2, \dots, F - 1$ for the purpose of splitting the column space in the DVR matrix into $\frac{N}{F}$ segments. Columns in each segment are constructed simultaneously in the ascending and descending horizontal direction following the three-step recursion given in eq. 13; b) Color map encoding values of the DVR transformation matrix for the Gauss-Hermite quadrature with $N = 128$ elements.

The recursive block is completed upon $\frac{F}{4} - 1$ steps (marked between red vertical dashed lines in Figure 3), where in each step two columns per segment are constructed. The columns are successively built moving in the descending and ascending order of q values simultaneously, starting from the midpoint columns indexed by $\tilde{q} = wF + \frac{F}{2}$. Schematic for this procedure is shown in Figure 5a. In constructing \hat{U}_{2c} and \hat{U}_{2c+1} we utilize the three-term recurrence relation satisfied by the DVR matrix elements:

$$T_{p,q+2} = (A_q + B_q x_p)T_{p,q+1} + C_q T_{p,q}. \quad (13)$$

where the specific values of coefficients A_q, B_q, C_q depend on the underlying orthogonal polynomial. The number of effective parameters in eq. 13 can be reduced by introducing scaled matrix elements: $T'_{p,q+2} = \gamma_q T_{p,q+2}$, giving the following recursions for the ascending and descending directions, respectively:

$$T'_{pq} = (A'_q + B'_q x_p)T'_{p,q-1} + T'_{p,q-2} \quad \text{for } q = wF + \frac{F}{2} + k \quad (14)$$

$$T'_{pq} = (A'_q + B'_q x_p)T'_{p,q+1} + T'_{p,q+2} \quad \text{for } q = wF + \frac{F}{2} - 1 - k \quad (15)$$

where $k = 1, \dots, \frac{F}{2} - 1$. For the two initial columns in each segment we set $\gamma_q = 1$ (for $q = wF + \frac{F}{2} - 1$ and $q = wF + \frac{F}{2}$). For the ascending direction of $q = wF + \frac{F}{2} + k$ the scaling constant is given by

$$\gamma_q = \gamma_{q-2} C_{q-2}^{-1} = (C_{q-2} C_{q-4} \dots C_i)^{-1} \quad (16)$$

where $i \in \{wF + \frac{F}{2} - 1, wF + \frac{F}{2}\}$ has the same parity as q . Then $(A'_q, B'_q) = \frac{\gamma_q}{\gamma_{q-1}} (A_{q-2}, B_{q-2})$. For the descending direction of $q = wF + \frac{F}{2} - 1 - k$, the scaling constant is given by the formula $\gamma_q = \gamma_{q+2} C_q = C_q C_{q+2} \dots C_i$, where again $i \in \{wF + \frac{F}{2} - 1, wF + \frac{F}{2}\}$ has the same parity as q . In this case $(A'_q, B'_q) = -\frac{\gamma_{q+2}}{\gamma_{q+1}} (A_q, B_q)$. Each unitary iteration \hat{U}_{2c} and \hat{U}_{2c+1} involves two arithmetic blocks, two multiplications and one addition each, as shown in Figure 4. The appropriate qubit state transformation for \hat{U}_{2c} and \hat{U}_{2c+1} can be written as:

$$U_{2c} |q\rangle |x\rangle |A\rangle |B\rangle = |q\rangle |x\rangle |A\rangle |B \oplus (A'_q + B'_q x)A\rangle \quad (17)$$

$$U_{2c+1} |q\rangle |x\rangle |A\rangle |B\rangle = |q\rangle |x\rangle |A \oplus (A'_q + B'_q x)B\rangle |B\rangle \quad (18)$$

for $|v - \frac{F-1}{2}| > 2c$. The indices for recursion constants appearing in eqs. 17, 18 are given by

$$\hat{q} = \hat{q}(v_0, w, c) = \begin{cases} wF + F/2 + 2c & \text{for } v_0 = 1 \\ wF + F/2 - 1 - 2c & \text{for } v_0 = 0 \end{cases} \quad (19)$$

and

$$\check{q} = \check{q}(v_0, w, c) = \begin{cases} wF + F/2 + 2c + 1 & \text{for } v_0 = 1 \\ wF + F/2 - 2 - 2c & \text{for } v_0 = 0. \end{cases} \quad (20)$$

where $q = wF + v$. Further details of the quantum arithmetics are given in Appendix B.

Cost estimation for the recursive block. The recursive block given by eqs. 17,18 and shown between the red dashed lines in Figure 3 requires loading $2N/F$ constants A'_q, B'_q with QROM, associated with the $2N/F$ Toffoli gate cost. The total Toffoli cost for the U_c iteration shown in Figure 4 consists of multiplication and addition costs (cf. eq. 1718). Circuit performing multiplication of numbers B'_q and x , denoted $MUL(B'_q, x)$, involves $2m^2$ Toffoli gates. The result of multiplication is rounded to m -bits. Circuit performing addition of A'_q and $B'_q x$, denoted as $ADD[A'_q, B'_q x]$, costs $4m$ Toffoli gates. All the above arithmetic operations must be uncomputed, which doubles the Toffoli cost. Finally, the bitwise addition in eq. 17, $B \oplus (A'_q + B'_q x)A$ costs $2m$ Toffoli gates, giving the overall Toffoli count for the arithmetic part \hat{U}_c :

$$C(\hat{U}_c) = 2C(\hat{D}_{\frac{N}{F}, m}(A')) + 2C(\hat{D}_{\frac{N}{F}, m}(B')) + 2C(MUL(B'_q, x)) + 2C(ADD[A'_q, B'_q x]) + \quad (21)$$

$$+ 2C(MUL(A'_q + B'_q x, B)) + C(B \oplus (A'_q + B'_q x)A) = 4\frac{N}{F} + 8m^2 + 10m \quad (22)$$

There are $\frac{F}{4} - 1$ pairs of transformations $(\hat{U}_{2c}, \hat{U}_{2c+1})$ giving the total cost of the recursive block:

$$C_{rec} = \left(\frac{F}{4} - 1\right) \left(2C(\hat{U}_c) + 2C(v' \geq c)\right) \quad (23)$$

where the checking the condition $v' \geq c$ (on $f - 2$ qubits) costs $f - 2$ Toffoli gates, while SWAP operations cost $3m$ Toffolis in total. Here $v' = v_1 2^{f-2} + \dots + v_{f-2} 2^1$ is a bitmask

of v with the most and the least significant bit removed. The cost of loading scaling constants γ_q^{-1} and multiplication $C(MUL(A, \gamma_q^{-1}))$ shown in Figure 3 is $N + 2m^2$ Toffoli gates, assuming SELECT QROM.

Resource estimation summary. We summarize the quantum computing resources for implementing the DVR oracle circuit given in eq. 9, by first estimating the required number of qubits. The DVR oracle requires $2n + m$ (nonancilla) qubits, n per argument register keeping values of matrix row and column indices, respectively, plus m qubits to represent the output matrix elements. We use additional m qubits for the purpose of loading pairs of entries of the DVR matrix, as required by the recursive scheme adopted in eqs. 17,18. Quantum arithmetic performed in eqs. 17,18 requires $6m$ additional ancilla qubits. Thus the total qubit count is $2n + 9m$.

The total Toffoli gate cost for the DVR oracle consists of initialization cost $C(\hat{U}_{init})$, the cost of the recursive block C_{rec} plus the cost of the remaining operations: $C(\hat{U}_1)$, $C(SWAPS)$ and $C(MUL(A, \gamma_q^{-1}))$, giving:

$$C_{ODVR} = C(\hat{U}_{init}) + C(\hat{U}_1) + C_{rec} + C(MUL(A, \gamma_q^{-1})) + C(SWAPS) \quad (24)$$

The total cost is thus:

$$C_{ODVR} = \frac{N^2}{F} + N + 4\frac{N}{F} + 8m^2 + 10m + \left(\frac{F}{4} - 1\right) \left(16m^2 + 8m + 4\frac{N}{F} + 2(f - 2)\right) + 3m + N + 2m^2 \quad (25)$$

in which the dominant cost is given by

$$C_{ODVR} \approx \frac{N^2}{F} + 4Fm^2 \quad (26)$$

In eqs. 25-27 we assume SELECT QROM for initial data loading (\hat{U}_{init}), with the benefit of qubit saving compared to the SELSWAP approach. When one optimizes for balance between the number of qubits and T-gates, the SELSWAP version of QROM in \hat{U}_{init} can be chosen (cf. eq. 11) giving the asymptotic cost:

$$C_{ODVR} \approx 4Fm^2 + N \left(\sqrt{\frac{m}{F}} + 1\right) \quad (27)$$

The DVR oracle for parity-conserving orthogonal polynomials. When $A_q = 0$ for all q , parity symmetry emerges in the orthogonal polynomials defined by the recursive relations shown in Eq. 13. For example, Hermite polynomials and Jacobi polynomials with $\alpha = \beta$ (such as Legendre and Chebyshev) exhibit the symmetry $T_{pq} = (-1)^q T_{N-p,q}$. These represent a special class of orthogonal polynomials that allow for additional quantum computing cost savings.

In such cases, it is sufficient to load only the upper half of the DVR matrix, applying a sign flip to the lower half for odd-numbered columns, at no additional Toffoli cost. Consequently, the Toffoli cost of QROM loading $\hat{D}_{N/2,m}(x)$ is halved in the SELECT method and reduced by a factor of $\sqrt{2}$ for equal-split SELSWAP. The cost of loading the initial $2\frac{N}{F}$ columns in \hat{U}_{init} is reduced by the same factor. Additionally, the operations associated with adding the free term $A_{q'}$ in the recursion are removed, saving an additional $2m$ Toffoli gates per recursive step.

In summary, for parity-conserving polynomials, the total Toffoli gate cost of the DVR oracle is given by

$$C_{ODVR} = \frac{N^2}{2F} + \frac{N}{2} + 4\frac{N}{F} + 8m^2 + 10m + \left(\frac{F}{4} - 1\right) \left(16m^2 + 8m + 4\frac{N}{F} + 2(f-2)\right) + 3m + N + 2m^2 \quad (28)$$

3.2 DVR Oracle via SELSWAP QROM

An alternative construction of the DVR oracle, compared to the *recursive method* described in Section 3.1, employs the generic QROM technique. In this approach, matrix elements are loaded into the quantum computer using a combination of the SELECT and SWAP oracles introduced in Ref. [33]. Loading the DVR matrix into a quantum state via the SELSWAP method requires $N\sqrt{m}$ Toffoli gates and $N\sqrt{m}$ ancilla qubits.

For parity-conserving orthogonal polynomials, this cost is reduced to $\frac{1}{2}N\sqrt{m}$ Toffoli gates, i.e. approximately half that of the general case. The T -depth of the SELSWAP method is roughly equal to the T -gate count, with the SELECT operation dominating the circuit depth. A comparison of the direct SELSWAP and recursive approaches is presented in the following section.

3.3 Comparison of techniques for constructing the DVR oracle

The two approaches for constructing the DVR oracle discussed in sections 3.1 and 3.2 are summarized in Table 1.

Quantity	T-count	T-depth	Qubit count	Volume
LKS	$N\sqrt{m}$	$N\sqrt{m}$	$N\sqrt{m}$	N^2m
REC	$4Fm^2 + \frac{N^2}{F}$	$4Fm^2 + \frac{N^2}{F}$	$2n + 9m$	$36Fm^3 + 9m\frac{N^2}{F}$
REC-LKS	$4Fm^2 + N\left(\sqrt{\frac{m}{F}} + 1\right)$	$4Fm^2 + N\left(\sqrt{\frac{m}{F}} + 1\right)$	$N\sqrt{\frac{m}{F}} + \sqrt{Nm} + 2n + 6m$	$N^2\frac{m}{F} + 24Fm^3$

Table 1: Comparison of methods for constructing the DVR oracle defined in eq. 9. LKS stands for the direct QROM method using the SELSWAP technique from ref. [33], REC is our present method where the initialization unitary given in eq. 12 is constructed using SELECT QROM (saving ancilla qubits), while REC-LKS is the present method where the initial unitary given in eq. 11 is constructed using SELSWAP QROM (balancing T-gate cost with ancilla qubits). Only dominant terms are shown for the recursive method. The columns correspond to: T-count, T-depth, qubit count, and quantum volume (qubit count \times T-count).

The recursive method involves approximately $\mathcal{O}(4Fm^2 + \frac{N^2}{F})$ T-gates compared to $\mathcal{O}(N\sqrt{m})$. Choosing $F = \frac{N}{m}$ leads to $\mathcal{O}(Nm)$ T-gate cost for the recursive method, which is typically greater cost than that of the SELSWAP method. However the recursive method is more favourable in terms of qubit count: $2n + 9m$ compared do $N\sqrt{m}$. The quantum volume for the recursive method with $F = \frac{N}{m}$ is $\mathcal{O}(Nm^2)$ compared to $\mathcal{O}(N^2m)$ for the LKS method. The advantage of the recursive method is that it uses $\mathcal{O}(m)$ ancilla qubits and $\mathcal{O}(Nm^2)$ Clifford gates, compared to $\mathcal{O}(N\sqrt{m})$ ancilla qubits and $\mathcal{O}(N^2m)$ Clifford gates for the LKS method. For the recursive-LKS method the quantum volume is reduced with respect to the SELECT-based recursive method by a constant factor. Quantum volume comparison between the SELSWAP (LKS) method and the recursive method is shown in Figure 6. It follows that for DVRs of size greater than 2^7 and at appropriately low precisions m for representing the matrix elements, the recursive method is advantageous over the LKS method with respect to the quantum volume metric.

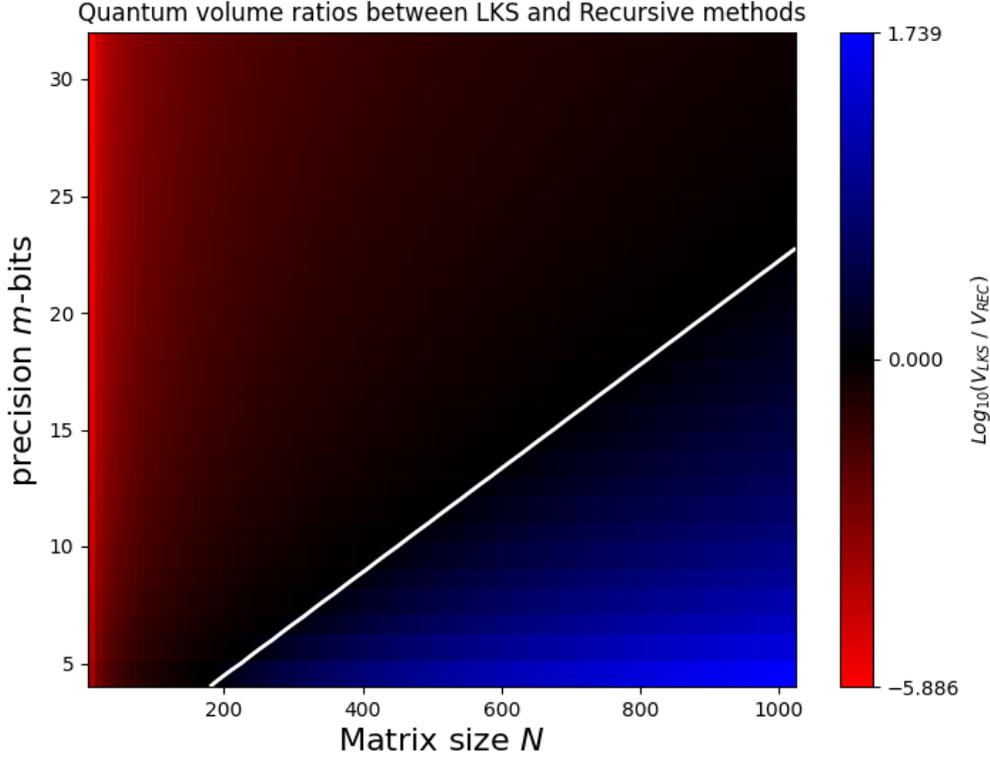


Figure 6: Comparison of quantum volumes for the LKS method and the recursive method for constructing the DVR oracle as a function of the number of basis functions N and the number of bits of precision m for representing the DVR matrix elements. White equal volume isoline is shown. Blue color denotes region of advantage of the recursive method.

We note that error-propagation when using arithmetics requires a qubit overhead of at least $2F$ additional qubits for representing x , T_{pq} (for precision of m -bits for the final results). It is thus a matter of choice and a specific use-case which method to choose. For optimizing the number of qubits used it is preferable to use the recursive method, whereas for high-precision and low matrix size computations we recommend using the LKS method.

4 The DVR unitary

4.1 Construction via reflections

The DVR unitary can be constructed from the DVR oracle using a technique given in ref. [33]. The resulting unitary circuit represents the following transformation:

$$\hat{D} = |0\rangle\langle 1| \hat{T} + |1\rangle\langle 0| \hat{T}^\dagger \quad (29)$$

The construction assumes providing an oracle returning the k -th column of the DVR unitary, $|u_k\rangle = \hat{T} |k\rangle$. Let U_k prepare the following state:

$$|w_k\rangle = |0\rangle_a |k\rangle - |1\rangle_a |u_k\rangle \quad (30)$$

where $|k\rangle$ represents index-state labeling columns of the DVR matrix. Then the product of N reflections:

$$\hat{R}_k = I - 2 |w_k\rangle\langle w_k| \quad (31)$$

gives $\hat{D} = \prod_{k=1}^N \hat{R}_k = |0\rangle \langle 1| \hat{T} + |1\rangle \langle 0| \hat{T}^\dagger$. Having execution of \hat{T} and \hat{T}^\dagger conditioned on the state of ancilla qubit can be useful in certain scenarios, as discussed further. The total cost for the \hat{D} unitary is $2N^2 + N(4m+1)n$, where m is the precision of matrix elements. This cost consists of the state preparation cost and the reflections cost. In the following paragraphs we give a more detailed resource estimation for the procedure sketched above for constructing \hat{D} .

State preparation. Let us consider the cost of preparing the state $|w_k\rangle$ given in eq. 30. For simplicity, consider the case of DVRs generated by polynomials with definite parity, i.e. $A_q = 0$ in eq. 4. For general polynomials the gate cost is roughly doubled due to twice the number of angles that must be loaded. Our present state preparation procedure follows the approach of Ref. [33] and references therein.

Step 1. First, initialize state $|0\rangle_a |0\rangle$, on which execute $HA_k \otimes I$ and get $|0\rangle_a |0\rangle \pm |1\rangle_a |0\rangle$, where $A_k \in \{I, X\}$ (if the first number in the k -th column of the \mathbf{T} matrix is nonnegative we choose $A_k = X$, otherwise $A_k = I$). Next, create the state $|0\rangle_a |k\rangle \pm |1\rangle_a |0\rangle$ with controlled- \hat{X} gates, with controls on the ancilla register $|0\rangle_a$ and $\log(N)$ target qubits for representing the integer index $|k\rangle$.

Step 2 Consider the n -qubit state $|\cdot\rangle = |\cdot\rangle_0 \otimes \dots \otimes |\cdot\rangle_{n-1}$ initialized for representing $|u_k\rangle$ given in eq. 30. We act on the $|\cdot\rangle_0$ qubit with controlled- H gate, where the condition is on the ancilla qubit $|\cdot\rangle_a$. Controlled Hadamard is equivalent to two T-gates.

Step 3 Apply CX^{n-1} on qubits $|\cdot\rangle_1, \dots, |\cdot\rangle_{t-1}$, with control on the 0-th qubit to be $|1\rangle_0$. This is required for accounting for the horizontal symmetry of the DVR matrix. We reflect the elements in the upper half of the DVR matrix.

Step 4 For a given k denoting column index, follow the recursive procedure: for $t = 1, \dots, n-1$ load suitable angles φ_l represented with $m+1$ -bits:

$$\hat{D}_{N,m+1}^{(t)}(\varphi) |l\rangle |0\rangle = |l\rangle |\varphi_l\rangle \quad (32)$$

where $l = l_1 l_2 \dots l_{t-1}$ is index

i.e. we load 2^{t-1} $(m+1)$ -bit numbers. The extra $m+1$ 'th bit stores information about the sign of the coefficient. The QROM given in eq. 32 is executed conditionally on the ancilla qubit $|0\rangle_a$ in state $|1\rangle_a$. Also, accounting for the horizontal symmetry in the DVR matrix we can ignore qubit $|\cdot\rangle_0$ - the appropriate reflection is implemented in step 3. The angles φ_l are chosen such that

$$\cos^2 \varphi_l = \frac{\sum_{s=0}^{2^{t-1}N-1} |T_{l2^{-t}N+s k}|^2}{\sum_{s=0}^{2^t N} |T_{l2^{-t}N+s k}|^2} \quad (33)$$

i.e. they reproduce the correct probabilities in the t 'th recursive step:

$$V_k^{(t)} |0\rangle_t = \cos |\varphi_l| |0\rangle_t + \sin |\varphi_l| |1\rangle_t \quad (34)$$

and the relative phase between the elements is retrieved by loading one bit informing about the sign of φ_l and executing controlled-Z operation on qubit $|\cdot\rangle_t$. We choose φ_l to be nonnegative for $T_{l2^{-t}Nk}$, $T_{l2^{-t}N+2^{t-1}Nk}$ both nonnegative or both nonpositive, and negative otherwise.

step 5. Uncompute the CX^{n-1} operation applied in step 3, If k is odd, apply Z to qubit $|\cdot\rangle_0$.

The Toffoli cost of the t -th recursive step in the procedure outlined above is 2^{t-1} , giving the total cost

$$\sum_{t=1}^{n-1} 2^{t-1} = 2^{n-1} - 1 = \frac{1}{2}N - 1 \quad (35)$$

in each recursive step the controlled single-qubit arbitrary rotation consumes $2m$ Toffoli gates (using phase-gradient technique). The circuit must be uncomputed adding another $\sum_{t=1}^{n-1} 2^{t-1} = \frac{1}{2}N$ Toffoli gates, to give the final cost of the state preparation

$$C(\hat{U}_k) = N + 2(n-1)m \quad (36)$$

Reflections. Implementing reflections $R_k = I - 2|w_k\rangle\langle w_k|$ requires loading $N/2$ angles that encode appropriate column of the DVR matrix $|u_k\rangle$. When u_k is non-symmetric with respect to the midpoint (for orthogonal polynomials of indefinite parity), N angles must be loaded. For the $|0\rangle|k\rangle$ part of the reflection no additional Toffoli gates are required, leaving $N/2 + 1$ Toffoli gates for state preparation with the SELECT-QROM method, plus equal number of Toffoli gates for uncomputation. This final Toffoli cost for the state preparation stage is thus $N + 2mn$, where $2mn$ is from controlled-adder in the phase-gradient technique [33]. Since $\hat{R}_k = U_k(I - 2|0\rangle\langle 0|)U_k^\dagger$ the cost implementing a single reflection operator is $2N + (4m + 1)n$. Finally, the Toffoli gate cost of the DVR unitary \hat{D} is

$$C_{REFL.}(\hat{D}) = \mathcal{O}\left(2N^2 + N(4m + 1)n\right). \quad (37)$$

4.2 Construction via block-encoding and QROM

The DVR unitary can be implemented via an construction alternative to the reflections method discussed in sec. 4.1. The other technique relies on block-encoding of an appropriate matrix loaded with the following oracle:

$$\tilde{T}|p\rangle|q\rangle|0\rangle = |p\rangle|q\rangle\left|\frac{2}{\pi}\arcsin T_{pq}\right\rangle \quad (38)$$

The DVR unitary can be synthesized using the technique from ref. [14], giving $\frac{1}{N}\hat{T}$, followed by amplitude amplification [34], associated with an N -fold increase in the Toffoli cost. The SELECT circuit for \hat{T} costs N Toffoli gates ($N^2/2$ for symmetric $T_{pq} = (-1)^q T_{N-pq}$ matrix elements). To get $\frac{1}{N^2}\hat{T}$ we construct the circuit implementing the following controlled-rotation:

$$P\left|\frac{2}{\pi}\arcsin a\right\rangle|0\rangle = \left|\frac{2}{\pi}\arcsin a\right\rangle(a|0\rangle + \sqrt{1-a^2}|1\rangle) \quad (39)$$

that involves m controlled single-qubit rotations and costs roughly $m+m^2(m-1)/2$ Toffoli gates. The gate count can be reduced at the expense of an increased number of qubits. Thus the Toffoli cost for implementing the DVR unitary scales as

$$C_{BE}(\hat{D}) = \mathcal{O}\left(N(N^2 + m^3)\right). \quad (40)$$

4.3 Construction via block-encoding and quantum arithmetics

An alternative construction of \tilde{T} uses quantum arithmetic operations to build matrix elements $\frac{2}{\pi}\arcsin T_{pq}$ defined in Eq. 38. The benefit relative to the direct loading of the matrix elements with QROM depends on the magnitude of the entries in \tilde{T} and their precision. For example, for the Gauss-Hermite DVR the largest T_{pq} entry is around 0.5, such that the Taylor series approximation $\arcsin(x) \approx x + x^3/6 + 3x^5/40 + 5x^7/102 + \frac{7!!}{8!! \cdot 9}x^9$ is sufficient for a 16-bit precision representation, with the associated cost of roughly $16m^2$ Toffoli gates.

In general, the construction of states storing values of a polynomial $P(x) = a_1x + a_3x^3 + \dots + a_{2p-1}x^{2p-1}$ with quantum arithmetic operations requires $3pm^2 + \mathcal{O}(pm)$ Toffoli gates and $\mathcal{O}(mp)$ qubits. The procedure begins by initializing qubit registers in the states: $|x\rangle, |x^2\rangle, |x^3\rangle, |x^5\rangle, \dots, |x^{2p-1}\rangle$, with the associated cost of $2pm^2$ Toffoli gates. Such a construction involves transforming the register $|x\rangle$ while keeping previously loaded numbers, by multiplying by x^2 to get $|x^3\rangle$ and so on. Each multiplication is associated with an approximate cost of m^2 Toffoli gates.

Next, using a separate qubit register, create $|a_{2k-1}x^{2k-1}\rangle$ for $k = 1, 2, \dots, p$ and sum the terms, giving an extra cost of $m^2 + m$ Toffoli gates, per term. Thus, overall, including the cost of QROM for loading the initial matrix elements T_{pq} , the total cost of the quantum arithmetic method is

$$C_{art} = N^2 + 3pm^2 + \mathcal{O}(pm) \quad (41)$$

where p depends on the desired accuracy and the type of Gaussian DVR. In practice, it can be net-beneficial to set a threshold value τ for the matrix elements T_{pq} , below which quantum arithmetic is used in constructing \tilde{T} , and QROM is used otherwise.

5 Summary

We developed a quantum algorithm that implements the discrete-variable representation (DVR) transformation. Our approach involves constructing a DVR oracle, which is then used to synthesize a unitary circuit implementing the DVR gate. We explore several synthesis strategies for the DVR gate, including a method based on reflections, block-encoding using QROM, and block-encoding with quantum arithmetic.

For the DVR oracle construction, we found that for matrix sizes of $N = 2^7$ and larger, our recursive procedure (described in Section 3.1) can be advantageous to QROM, particularly when high precision is not required (see Fig. 6 and Section 3.3). For instance, for a 10-qubit DVR matrix at 16 bits of precision, the recursive procedure yields a gate count roughly 50% lower than the QROM-based construction. However, it should be noted that this estimate does not account for numerical error propagation. Incorporating numerical error-propagation in the recursive construction of the DVR gate incurs an overhead of at least $2F$ qubits, which may offset the benefits over QROM. F is the segmentation parameter defined in section 3.1.

We also presented two methods for implementing the DVR unitary gate. The first leverages a modified DVR oracle \tilde{T} defined in eq.38, followed by appropriate quantum rotations. The second, based on reflections, is discussed in Section 4.1 and implements Eq. 29. The reflection-based method exhibits more favorable Toffoli gate count scaling with respect to matrix size and precision compared to the method using \tilde{T} : $\mathcal{O}(2N^2 + N(4m + 1)n)$ vs $\mathcal{O}(N(N^2 + m^3))$, respectively.

In summary, we examined multiple strategies for constructing both the DVR oracle and the unitary DVR-FBR transformation circuit. Among these, we proposed a construction exploiting the recursive properties of orthogonal polynomials underlying the Gaussian DVR, which can outperform SELECT-SWAP QROM for large matrices and low precision. For synthesizing the DVR unitary circuit, we recommend the reflection-based technique described in Section 4.1.

The advantages of implementing the DVR transformation directly within a quantum circuit are apparent in applications such as block-encoding many-body Hamiltonians, relevant to simulations of vibrational and vibronic dynamics in molecules and materials. In such Hamiltonians, the kinetic energy operator is naturally represented in a variational (FBR) basis, while the potential energy function is best described in a DVR basis, in

which it is diagonal. Performing the FBR-to-DVR transformation on a D -particle bosonic product basis incurs a cost that scales linearly with D , relative to the 1D DVR transformation discussed here. In contrast to grid-based and finite-difference methods, Gaussian DVRs exhibit exponential convergence of the approximation error with respect to basis set size. This arises from the exactness of the underlying N -point quadrature in integrating polynomials up to degree $2N - 1$. As a result, DVR methods are highly effective for representing smooth wavefunctions of bound states that are common in many-body systems such as molecular vibrations.

This favorable scaling in both solution accuracy and quantum gate complexity (due to the tensor-product structure of multidimensional DVRs and the typically sparse nature of DVR Hamiltonians) makes the approach particularly appealing for quantum computation.

6 Acknowledgments

We thank Witold Jarnicki and Konrad Deka for helpful discussions. This work is funded by the European Innovation Council accelerator grant COMFTQUA, no. 190183782.

7 Appendix A: Gaussian DVRs

In this appendix we discuss some of the useful properties of Gaussian DVRs [31], including the derivation of the diagonal form of local operators in DVR, a method of constructing the DVR matrix and the Schroedinger equation in multi-dimensional DVR [4].

The potential energy operator in DVR. The DVR basis functions generally take non-zero values between quadrature points, however they tend to be smaller with increasing distance from the point of localization of the DVR function, as shown in Figure 1. This observation intuitively supports the diagonal approximation to the potential matrix $V(x)$:

$$\begin{aligned} V_{ij}^{DVR} &= \langle d_i(x) | V(x) | d_j(x) \rangle = \int_a^b d_i(x) V(x) d_j(x) dx \approx \\ &\approx \sum_{k=0}^{N-1} \frac{w_k}{\omega(x_k)} d_i(x_k) V(x_k) d_j(x_k) = \sum_{k=0}^{N-1} \frac{w_k}{\omega(x_k)} \sqrt{\frac{\omega(x_k)}{w_i}} \delta_{ik} V(x_k) \sqrt{\frac{\omega(x_k)}{w_j}} \delta_{jk} = V(x_i) \delta_{ij} \end{aligned} \quad (42)$$

For the potential energy operators represented as polynomial functions of coordinates (quadratic or higher degrees), the Gaussian quadrature rule is not exact and the FBR/DVR approximation can no longer be considered variational.

DVR defined via the Position operator. A practical way of obtaining the DVR can be achieved through the so called *product approximation* [4], by means of diagonalisation of the position operator X matrix. In VBR the matrix elements of X are written as:

$$\left(\mathbf{X}^{\text{VBR}} \right)_{ij} = \int_a^b \phi_i(x) x \phi_j(x) dx \approx \sum_{k=0}^{N-1} \frac{w_k}{\omega(x_k)} \phi_i(x_k) x_k \phi_j(x_k) = \sum_{k=0}^{N-1} \mathbf{T}_{ik}^T \mathbf{X}_k \mathbf{T}_{kj} \quad (43)$$

As long the $(\mathbf{X}^{\text{VBR}})_{ij}$ matrix is truncated to size $N \times N$ and the basis functions are orthogonal polynomials of degree $N - 1$, the integrated function is of degree $2N - 1$, and can be evaluated exactly by a Gaussian quadrature. It means that the \mathbf{T} matrix diagonalizes the position operator matrix in the orthogonal polynomials basis. As a result, eigenvalues

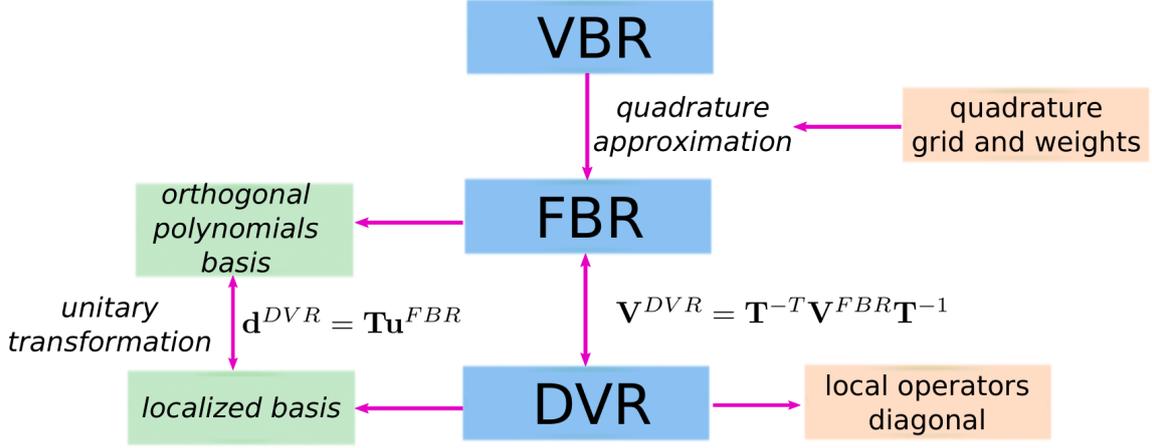


Figure 7: A general scheme for the VBR-FBR-DVR transformation. \mathbf{V}^{FBR} denotes FBR representation of a local operator and \mathbf{T} is the FBR-to-DVR transformation matrix.

of \mathbf{X} in this basis correspond to quadrature points and the diagonalising transformation matrix is related to the quadrature weights. Diagonalisation of \mathbf{X}^{VBR} unambiguously defines the DVR:

$$\mathbf{X}^{DVR} = \mathbf{T}\mathbf{X}^{VBR}\mathbf{T}^T \quad (44)$$

where we used the fact that for the Gaussian quadratures the FBR-DVR transformation matrix is unitary: $\mathbf{T}^{-1} = \mathbf{T}^T$. The VBR representation of the position matrix in the orthogonal polynomials basis is straightforward to derive, thanks to the three-term recurrence relations for orthogonal polynomials: $p_{q+1}(x) = (A_q + B_q x)p_q(x) + C_q p_{q-1}(x)$, where A_q, B_q, C_q are constants characteristic for a given class of polynomials. As a result the position operator matrix is tridiagonal. In practice, diagonalising the position operator matrix is the most efficient way of finding a DVR.

The Schroedinger equation in DVR. The Hamiltonian given in the variational basis (VBR) can be approximated with finite basis representation as follows

$$\mathbf{H}^{VBR} \approx \mathbf{H}^{FBR} = \mathbf{K}^{VBR} + \mathbf{T}^T \mathbf{V}^{diag} \mathbf{T} \quad (45)$$

where the kinetic energy operator \mathbf{K}^{VBR} remains in the VBR representation and quadrature approximation is applied to the potential energy matrix elements $\mathbf{V}^{VBR} \approx \mathbf{T}^T \mathbf{V}^{diag} \mathbf{T}$. Then the Schroedinger equation can be written in the matrix form as:

$$\left(\mathbf{K}^{VBR} + \mathbf{T}^T \mathbf{V}^{diag} \mathbf{T} \right) \mathbf{U} = \mathbf{T}^T \mathbf{T} \mathbf{E} \mathbf{U} \quad (46)$$

where overlap integral matrix S on the right-hand side is also given in quadrature approximation, i.e. $\mathbf{S} = \mathbf{T}^T \mathbf{T}$. Upon multiplication of eq. 46 from the left by \mathbf{T}^{-T} and defining a new basis by $\mathbf{Z} = \mathbf{T} \mathbf{U}$ we get the DVR form of the Schroedinger equation:

$$\left(\mathbf{T}^{-T} \mathbf{K}^{VBR} \mathbf{T}^{-1} + \mathbf{V}^{diag} \right) \mathbf{Z} = \mathbf{Z} \mathbf{E} \quad (47)$$

A schematic for the VBR-FBR-DVR transformation chain is depicted in Figure 7.

This scheme extends to many-dimensions by incorporating direct-product basis set representation of the multi-dimensional problem. The D -dimensional basis functions gain D indices as follows: $\{\phi_{n_1, n_2, \dots, n_D} := \phi_{n_1}(Q_1) \phi_{n_2}(Q_2) \dots \phi_{n_D}(Q_D)\}_{n_c=1, 2, \dots, N_c; c=1, \dots, D}$. Direct

product quadrature grid is a grid composed as a simple sum of 1-dimensional grids so that the indices of the multidimensional grid are independent. The multi-dimensional FBR Schroedinger equation then takes the form

$$\left(\mathbf{K}^{VBR} + \mathbf{T}^T \mathbf{V}^{diag} \mathbf{T}\right) \mathbf{U} = \mathbf{T}^T \mathbf{T} \mathbf{U} \mathbf{E} \quad (48)$$

where $\mathbf{T} = {}^{(1)}\mathbf{T} \otimes {}^{(2)}\mathbf{T} \otimes \dots \otimes {}^{(D)}\mathbf{T}$ with ${}^{(c)}\mathbf{T}_{p_c q_c} = N_{q_c} \sqrt{{}^{(c)}\mathbf{w}_{k_c n_c}} {}^{(c)}\mathbf{p}_{p_c q_c}$, where ${}^{(c)}\mathbf{p}_{p_c q_c} = p_{q_c}(Q_{p_c})$. The DVR representation is then obtained by transposing the Kronecker \mathbf{T} directly, at no additional cost compared to the one-dimensional problem. Note that the size of the wavefunction \mathbf{U} is N^D which quickly becomes a prohibitively large number with increasing D . For instance, if 10 basis functions per dimension are needed to converge the variational wavefunction, then for a 5-atom molecule ($D = 9$) e.g. CH_4 , as many as 10^9 basis functions are required. For molecules with more than 5 atoms, the amount of memory needed to store even a single wavefunction vector becomes problematic. To circumvent this *curse of dimensionality* several approaches have been developed to define FBR and DVR with non-direct product basis sets and non-direct product quadrature grids [35, 36, 37, 38]. With quantum computer qubit encoding of information, this memory requirement can be alleviated to $\mathcal{O}(ND)$ qubits, while the tensor structure of the multidimensional DVR transformation allows to execute this transformation at low depth (each dimension can be transformed simultaneously).

8 Appendix B: orthogonal polynomials

In this appendix we lay out properties of popular classes of orthogonal polynomials used in construction of Gaussian DVRs.

I. Generalized Laguerre polynomials For fixed $\alpha \in \mathbb{R}$ orthogonal polynomials with respect to $\mu = x^\alpha e^{-x} \chi_{[0, \infty)}$

We have the formula

$$L_n^{(\alpha)}(x) = \frac{x^{-\alpha} e^x}{n!} (x^{n+\alpha} e^{-x})^{(n)} = (-1)^n x^n / n! + \dots,$$

the recurrence relations

$$L_n^{(\alpha)}(x) = \frac{(-x + 2n + \alpha - 1)L_{n-1}^{(\alpha)}(x) - (n + \alpha - 1)L_{n-2}^{(\alpha)}(x)}{n}$$

and the norm

$$\|L_n^{(\alpha)}\|_{L^2(\mu)} = \sqrt{\frac{\Gamma(\alpha + n + 1)}{\Gamma(n + 1)}}.$$

Thus

$$t_{pq} = \frac{(\alpha + q)^{1/2} (-x + 2q + \alpha - 1) t_{pq-1}}{q^{3/2}} - \left(\frac{q + \alpha - 1}{q}\right)^{3/2} \sqrt{\frac{q + \alpha}{q - 1}} t_{pq-2},$$

Laguerre polynomials: $\alpha = 0$ Orthogonal polynomials with respect to $\mu = e^{-x} \chi_{[0, \infty)}$
We have the formula

$$L_n(x) = L_n^{(0)}(x) = \frac{e^x}{n!} (x^n e^{-x})^{(n)} = (-1)^n x^n / n! + \dots,$$

the recurrence relations

$$L_n(x) = \frac{(-x + 2n - 1)L_{n-1}(x) - (n - 1)L_{n-2}}{n}$$

and the norm

$$\|L_n\|_{L^2(\mu)} = 1.$$

Thus

$$t_{pq} = \frac{(-x_p + 2q - 1)}{q} t_{pq-1} - \frac{(q-1)}{q} t_{pq-2},$$

II. Jacobi polynomials For fixed $\alpha, \beta > -1$ orthogonal polynomials with respect to $\mu = (1-x)^\alpha(1+x)^\beta \chi_{[-1,1]}$.

We have the formula

$$\begin{aligned} P_n^{(\alpha,\beta)}(x) &= \frac{(-1)^n}{2^n n!} (1-x)^{-\alpha} (1+x)^{-\beta} \left((1-x)^\alpha (1+x)^\beta (1-x^2)^n \right)^{(n)} \\ &= \sum_{s=0}^n \binom{n+\alpha}{n-s} \binom{n+\beta}{s} \left(\frac{x-1}{2} \right)^s \left(\frac{x+1}{2} \right)^{n-s}, \end{aligned}$$

the recurrence relations

$$\begin{aligned} &2n(n+\alpha+\beta)(2n+\alpha+\beta-2)P_n^{(\alpha,\beta)}(x) \\ &= (2n+\alpha+\beta-1) \left((2n+\alpha+\beta)(2n+\alpha+\beta-2)x + \alpha^2 - \beta^2 \right) P_{n-1}^{(\alpha,\beta)}(x) \\ &\quad - 2(n+\alpha-1)(n+\beta-1)(2n+\alpha+\beta)P_{n-2}^{(\alpha,\beta)}(x) \end{aligned}$$

and the norm

$$\|P_n^{(\alpha,\beta)}(x)\|_{L^2(\mu)} = \sqrt{\frac{2^{\alpha+\beta+1}}{2n+\alpha+\beta+1} \frac{\Gamma(\alpha+n+1)\Gamma(\beta+n+1)}{\Gamma(\alpha+\beta+n+1)n!}}.$$

Thus recurrence relations for t_{pq} are complicated (but we can write it without Γ function).

It seems hard to deal with it but let us look at some special cases.

Legendre polynomials: $\alpha = \beta = 0$ Orthogonal polynomials with respect to $\mu = \chi_{[-1,1]}$

We have the formula

$$P_n = P_n^{(0,0)}(x) = \frac{1}{2^n n!} \left((x^2 - 1)^n \right)^{(n)}$$

the recurrence relations

$$P_n(x) = (2 - 1/n)xP_{n-1}(x) - (1 - 1/n)P_{n-2}$$

and the norm

$$\|P_n\|_{L^2(\mu)} = \sqrt{\frac{2}{2n+1}}.$$

Thus

$$t_{pq} = \frac{\sqrt{4q^2-1}}{q} x_p t_{pq-1} - \frac{(q-1)}{q} \sqrt{\frac{2q+1}{2q-3}} t_{pq-2},$$

Chebyshev polynomials (second kind): $\alpha = \beta = 1/2$ Orthogonal polynomials with respect to $\mu = \sqrt{1-x^2} \chi_{[-1,1]}$

We have the formula

$$U_n(x) = P_n^{(1/2,1/2)}(x) = \sum_{k=0}^{[n/2]} (-1)^k \binom{n-k}{k} (2x)^{n-2k}, \quad U_n(\cos t) = \frac{\sin((n+1)t)}{\sin t},$$

the recurrence relations

$$U_n(x) = 2xU_{n-1}(x) - U_{n-2}$$

and the norm

$$\|U_n\|_{L^2(\mu)} = \sqrt{\pi/2}.$$

Thus

$$\begin{aligned} t_{pq} &= 2x_p t_{pq-1} - t_{pq-2}, \\ \gamma_q &= (-1)^{\lfloor (q+1)/2 \rfloor} \end{aligned}$$

and

$$t'_{pq} = (-1)^q 2x_p t'_{pq-1} + t'_{pq-2}$$

thus we do not need to multiply by γ_q .

Chebyshev polynomials (first kind): $\alpha = \beta = -1/2$ Orthogonal polynomials with respect to $\mu = (1 - x^2)^{-1/2} \chi_{[-1,1]}$

We have the formula

$$T_n(x) = P_n^{(-1/2, -1/2)}(x) = \sum_{k=0}^{\lfloor n/2 \rfloor} \binom{n}{2k} (x^2 - 1)^k x^{n-2k}, \quad T_n(\cos t) = \cos(nt),$$

the recurrence relations

$$T_0 = 1, \quad T_1(x) = x, \quad T_n(x) = 2xT_{n-1}(x) - T_{n-2}$$

and the norm

$$\|T_0\|_{L^2(\mu)} = \sqrt{\pi} \text{ and } \|T_n\|_{L^2(\mu)} = \sqrt{\pi/2} \text{ for } n > 0.$$

Thus

$$\begin{aligned} t_{p2} &= 2x_p t_{p1} - \sqrt{1/2} t_{p0} \text{ and } t_{pq} = 2x_p t_{pq-1} - t_{pq-2} \text{ for } q > 2, \\ \gamma_q &= (-1)^{\lfloor (q+1)/2 \rfloor} \sqrt{(3 + (-1)^q)/2} \end{aligned}$$

and

$$t'_{pq} = (-1)^q 2x_p t'_{pq-1} + t'_{pq-2}$$

thus again we do not need to multiply by γ_q .

9 Appendix C: Quantum arithmetic for Quantum DVR (QDVR)

In this appendix, we discuss details of quantum arithmetic required for the implementation of the unitary blocks U_c in the recursive scheme for constructing the DVR oracle shown in Figure 3. The dominant cost arises from the multiplication of numbers, which requires $\mathcal{O}(m^2)$ gates in each U_c .

Recalling eq. 18:

$$U_{2c} |q\rangle |x\rangle |A\rangle |B\rangle = |q\rangle |x\rangle |A\rangle |B \oplus (A'_q + B'_q x)A\rangle \quad (49)$$

we need to multiply the numbers x , A , and B'_q . Multiplication by B'_q is cheaper than by A because B'_q is a fixed number (not a quantum input). These numbers are multiplied as integers written in binary system. The cost of computing xA is approximately $2 \text{length}(x) \text{length}(A)$ Toffoli gates, plus additional Clifford gates. Here $\text{length}(x)$ corresponds to binary precision, commonly denoted with m throughout the text. The cost of multiplying xA by B'_q is $\text{length}(xA) \cdot \text{length}(B'_q)$ Toffoli gates, plus Clifford gates. Of

course, the exact cost depends on the multiplication algorithm; here, we assume the most basic textbook method, i.e. without using the Quantum Fourier Transform.

Next, we add $(xAB'_q)_m$, which denotes the m most significant bits of xAB'_q , to A'_q , conditioned on the ancilla qubit that encodes $v' \geq c$ (see Figure 4). This controlled addition costs approximately $2m$ Toffoli gates, plus $\mathcal{O}(m)$ Clifford gates. Finally, we uncompute xAB'_q .

The basic algorithm we utilize for multiplying two numbers a and b is as follows. Let a and b be m -bit and m' -bit numbers, respectively, and let $m'' = m + m' + 1$. We perform m controlled additions: the i -th Adder A_i is controlled on the bit a_i in the decomposition $a = \sum a_i 2^i$, for $i = 0, \dots, m - 1$:

$$A_i |a\rangle |b\rangle |c\rangle = \begin{cases} |a\rangle |b\rangle |c\rangle & \text{if } a_i = 0 \\ |a\rangle |b\rangle |c + 2^i b\rangle & \text{otherwise.} \end{cases}$$

Each such adder is an addition of two m -bit numbers: b and $\lceil c/2^i \rceil$, and costs approximately $2m$ Toffoli gates. If we multiply by a fixed number, the operation costs (depending on the specific value) $2\times$ fewer Toffoli gates, since the Adders are then controlled by classical bits. We note that multiple possibilities for optimizing this cost may exist.

The multiplication operation $Mult_x |A\rangle = |xA\rangle$ can be implemented in the following way: if x is an m -bit (nonzero) number and a is m' -bit, then we can perform

$$Mult_x |A\rangle_{m'} |0\rangle_{m'+1} = |xA\rangle_{m''}.$$

which requires m' additional ancilla qubits. Note that if $|x| \neq 1$, we cannot simply approximate xA by truncating bits, since $Mult_x$ is a unitary operation and must therefore be injective. For example, if $x = 2^{-m}$, we require at least $2^{-(m+m')}$ precision. Consequently, if we perform K such multiplications as in our construction, we need $\mathcal{O}(Nm)$ additional ancilla qubits.

References

- [1] David O. Harris, Gail G. Engerholm, and William D. Gwinn. Calculation of matrix elements for one-dimensional quantum-mechanical problems and the application to anharmonic oscillators. *The Journal of Chemical Physics*, 43(5):1515–1517, September 1965.
- [2] J. C. Light and Z. Bačić. Adiabatic approximation and nonadiabatic corrections in the discrete variable representation: Highly excited vibrational states of triatomic molecules. *The Journal of Chemical Physics*, 87(7):4008–4019, October 1987.
- [3] J. C. Light, R. M. Whitnell, T. J. Park, and S. E. Choi. *Quantum Dynamics of Small Systems using Discrete Variable Representations*, page 187–213. Springer Netherlands, 1989.
- [4] John C. Light and Tucker Carrington. Discrete-variable representations and their utilization, January 2000.
- [5] J. Tennyson, M. A. Kostin, P. Barletta, G. J. Harris, O. L. Polyansky, J. Ramanlal, and N. F. Zobov. DVR3D: a program suite for the calculation of rotation-vibration spectra of triatomic molecules. 163:85–116, 2004.
- [6] Xiao-Gang Wang and Tucker Carrington. A discrete variable representation method for studying the rovibrational quantum dynamics of molecules with more than three atoms. *The Journal of Chemical Physics*, 130(9), March 2009.

- [7] Csaba Fábri, Martin Quack, and Attila G. Császár. On the use of nonrigid-molecular symmetry in nuclear motion computations employing a discrete variable representation: A case study of the bending energy levels of CH_5^+ . *The Journal of Chemical Physics*, 147(13), October 2017.
- [8] Daniel T. Colbert and William H. Miller. A novel discrete variable representation for quantum mechanical reactive scattering via the s-matrix kohn method. *The Journal of Chemical Physics*, 96(3):1982–1991, February 1992.
- [9] Aurel Bulgac and Michael McNeil Forbes. Use of the discrete variable representation basis in nuclear physics. *Physical Review C*, 87(5), May 2013.
- [10] Robert M. Parrish, Edward G. Hohenstein, Todd J. Martínez, and C. David Sherrill. Discrete variable representation in electronic structure theory: Quadrature grids for least-squares tensor hypercontraction. *The Journal of Chemical Physics*, 138(19), May 2013.
- [11] Philipp V. Demekhin, David Hochstuhl, and Lorenz S. Cederbaum. Photoionization of hydrogen atoms by coherent intense high-frequency short laser pulses: Direct propagation of electron wave packets on large spatial grids. *Physical Review A*, 88(2), August 2013.
- [12] Brittany Richman, C. J. Lobb, and Jacob M. Taylor. Utilizing discrete variable representations for decoherence-accurate numerical simulation of superconducting circuits, 2025.
- [13] Robert G. Littlejohn, Matthew Cargo, Tucker Carrington, Kevin A. Mitchell, and Bill Poirier. A general framework for discrete variable representation basis sets. *The Journal of Chemical Physics*, 116(20):8691–8703, May 2002.
- [14] Daan Camps, Lin Lin, Roel Van Beeumen, and Chao Yang. Explicit quantum circuits for block encodings of certain sparse matrices, 2022.
- [15] Ryan Babbush, Craig Gidney, Dominic W. Berry, Nathan Wiebe, Jarrod McClean, Alexandru Paler, Austin Fowler, and Hartmut Neven. Encoding electronic spectra in quantum circuits with linear t complexity. *Phys. Rev. X*, 8:041015, Oct 2018.
- [16] Joonho Lee, Dominic W. Berry, Craig Gidney, William J. Huggins, Jarrod R. McClean, Nathan Wiebe, and Ryan Babbush. Even more efficient quantum computations of chemistry through tensor hypercontraction. *PRX Quantum*, 2:030305, Jul 2021.
- [17] Guang Hao Low and Isaac L Chuang. Hamiltonian simulation by qubitization. *Quantum*, 3:163, 2019.
- [18] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, page 193–204, New York, NY, USA, 2019. Association for Computing Machinery.
- [19] Danial Motlagh, Robert A. Lang, Jorge A. Campos-Gonzalez-Angulo, Tao Zeng, Alan Aspuru-Guzik, and Juan Miguel Arrazola. Quantum algorithm for vibronic dynamics: Case study on singlet fission solar cell design, 2024.
- [20] Jan Tułowiecki, Łukasz Czerwiński, Konrad Deka, Jan Gwinner, Witold Jarnicki, and Adam Szady. Diagonal operator decomposition on restricted topologies via enumeration of quantum state subsets, 2024.

- [21] David Gosset, Robin Kothari, and Kewen Wu. Quantum state preparation with optimal t-count, 2024.
- [22] Alexandru Macridin, Panagiotis Spentzouris, James Amundson, and Roni Harnik. Electron-phonon systems on a universal quantum computer. *Physical Review Letters*, 121(11), September 2018.
- [23] Alexandru Macridin, Panagiotis Spentzouris, James Amundson, and Roni Harnik. Digital quantum computation of fermion-boson interacting systems. *Physical Review A*, 98(4), October 2018.
- [24] Pauline J Ollitrault, Sven Jandura, Alexander Miessen, Irene Burghardt, Rocco Martinazzo, Francesco Tacchino, and Ivano Tavernelli. Quantum algorithms for grid-based variational time evolution. *Quantum*, 7:1139, October 2023.
- [25] Hans Hon Sang Chan, Richard Meister, Tyson Jones, David P. Tew, and Simon C. Benjamin. Grid-based methods for chemistry simulations on a quantum computer. *Science Advances*, 9(9), March 2023.
- [26] Andrew M. Childs, Jin-Peng Liu, and Aaron Ostrander. High-precision quantum algorithms for partial differential equations. *Quantum*, 5:574, November 2021.
- [27] Giorgio Tosti Balducci, Boyang Chen, Matthias Möller, Marc Gerritsma, and Roeland De Breuker. Review and perspectives in quantum computing for partial differential equations in structural mechanics. *Frontiers in Mechanical Engineering*, 8, September 2022.
- [28] Yuan Su, Dominic W. Berry, Nathan Wiebe, Nicholas Rubin, and Ryan Babbush. Fault-tolerant quantum simulations of chemistry in first quantization. *PRX Quantum*, 2(4), November 2021.
- [29] Chee-Kong Lee, Chang-Yu Hsieh, Shengyu Zhang, and Liang Shi. Variational quantum simulation of chemical dynamics with quantum computers. *Journal of Chemical Theory and Computation*, 18(4):2105–2113, March 2022.
- [30] Mohsen Bagherimehrab and Alán Aspuru-Guzik. Efficient quantum algorithm for all quantum wavelet transforms. *Quantum Science and Technology*, 9(3):035010, April 2024.
- [31] J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis*. Springer New York, 1980.
- [32] Milton Abramowitz and Irene A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover, New York, ninth dover printing, tenth gpo printing edition, 1964.
- [33] Guang Hao Low, Vadym Kliuchnikov, and Luke Schaeffer. Trading t gates for dirty qubits in state preparation and unitary synthesis. *Quantum*, 8:1375, June 2024.
- [34] Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation, 2002.
- [35] Gustavo Avila and Tucker Carrington. Nonproduct quadrature grids for solving the vibrational schrödinger equation. *J. Chem. Phys.*, 131(17):174103, nov 2009.
- [36] Gustavo Avila and Tucker Carrington. A multi-dimensional smolyak collocation method in curvilinear coordinates for computing vibrational spectra. *J. Chem. Phys.*, 143(21):214108, dec 2015.

- [37] Emil J. Zak and Tucker Carrington. Using collocation and a hierarchical basis to solve the vibrational schrödinger equation. *The Journal of Chemical Physics*, 150(20):204108, May 2019.
- [38] Gustavo Avila and Tucker Carrington. Reducing the cost of using collocation to compute vibrational energy levels: Results for ch_2nh . *J. Chem. Phys.*, 147(6):064103, aug 2017.