quantum reports

IMPACT FACTOR 1.3

CITESCORE 3.0

Article

Quantum Simulation of Variable-Speed Multidimensional Wave Equations via Clifford-Assisted Pauli Decomposition

Boris Arseniev and Igor Zacharov

MDPI

https://doi.org/10.3390/quantum7040047

quantum reports

MDPI

Article

Quantum Simulation of Variable-Speed Multidimensional Wave Equations via Clifford-Assisted Pauli Decomposition

Boris Arseniev * and Igor Zacharov

Independent Researcher, Moscow 121205, Russia
* Correspondence: barseniev@gmail.com

Abstract

The simulation of multidimensional wave propagation with variable material parameters is a computationally intensive task, with applications from seismology to electromagnetics. While quantum computers offer a promising path forward, their algorithms are often analyzed in the abstract oracle model, which can mask the high gate-level complexity of implementing those oracles. We present a framework for constructing a quantum algorithm for the multidimensional wave equation with a variable speed profile. The core of our method is a decomposition of the system Hamiltonian into sets of mutually commuting Pauli strings, paired with a dedicated diagonalization procedure that uses Clifford gates to minimize simulation cost. Within this framework, we derive explicit bounds on the number of quantum gates required for Trotter–Suzuki-based simulation. Our analysis reveals significant computational savings for structured block-model speed profiles compared to general cases. Numerical experiments in three dimensions confirm the practical viability and performance of our approach. Beyond providing a concrete, gate-level algorithm for an important class of wave problems, the techniques introduced here for Hamiltonian decomposition and diagonalization enrich the general toolbox of quantum simulation.

Keywords: quantum algorithms; Hamiltonian simulation; Pauli decomposition; wave equation; mutual diagonalization; band matrices; commuting Pauli sets

Academic Editor: Vladimir M. Stojanović

Received: 12 September 2025
Revised: 2 October 2025
Accepted: 10 October 2025
Published: 13 October 2025

Citation: Arseniev, B.; Zacharov, I. Quantum Simulation of Variable-Speed Multidimensional Wave Equations via Clifford-Assisted Pauli Decomposition. *Quantum Rep.* **2025**, *7*, 47. https://doi.org/10.3390/quantum7040047

1. Introduction

Wave equations are fundamental for modeling a wide range of physical phenomena, from acoustic and seismic wave propagation to electromagnetic fields and quantum mechanics [1,2]. Numerically simulating these equations, especially in high dimensions and in the presence of heterogeneous material properties, remains computationally demanding for classical computers. While finite-difference and finite-element methods are widely used, they often suffer from the "curse of dimensionality", where computational costs increase exponentially with the number of spatial dimensions [2,3]. This scaling imposes a significant bottleneck on applications requiring high-resolution models in three or more dimensions, such as full-waveform inversion in geophysics [4,5] or detailed optical simulations [6].

Quantum computing presents a promising avenue for overcoming these limitations. By exploiting the inherent parallelism of quantum states, quantum algorithms can achieve exponential speedups for certain linear algebra tasks and differential equation solvers [7,8]. Recent advancements have made significant strides in creating quantum algorithms tailored for partial differential equations (PDEs), notably the wave equation. Previous studies have demonstrated the efficacy of quantum algorithms employing an oracle to probe the wave speed profile [9,10]. In contrast, different studies have concentrated on more

practical applications concerning one-dimensional scenarios where the speed remains constant [11–13]. An important challenge remains in creating clear, non-oracle methods for high-dimensional wave equations with changing speed coefficients, which are vital for accurate physical modeling. A recent study introduced an effective oracle-based solution for this task [14], employing an oracle technique that, while scalable, necessitates more ancilla qubits and incurs further overhead.

Our approach addresses this gap and builds on prior methods for Pauli decomposition of sparse matrices [15,16], extending them to arbitrary multidimensional wave problems with general and block-structured variable speed profiles. We derive explicit expressions for the number of Pauli strings, the structure of mutually commuting sets, and the corresponding diagonalization circuits. Exploiting these structures, we provide upper bounds on the number of one- and two-qubit gates required for first-order and higher-order Trotter approximations. Furthermore, by incorporating block-structured velocity profiles—common in practical applications such as layered media [5]—we demonstrate substantial reductions in circuit complexity, highlighting the advantage of leveraging problem-specific structure.

To support the theoretical scaling, we offer numerical simulations for a 3D wave equation featuring block-structured speed profiles. The results illustrate the effectiveness of our decomposition method and confirm the predicted scaling of Pauli term counts. Our analysis shows that the proposed quantum algorithm alleviates the classical "curse of dimensionality", providing a practical approach to modeling wave dynamics in high-dimensional spaces.

The article begins by outlining the general problem of the multidimensional wave equation as framed for quantum algorithms in Section 2, including essential definitions. The main results are detailed in Section 3, with Section 3.1 covering Pauli decomposition, Section 3.2 focusing on diagonalization, Section 3.3 addressing the scaling of the multi-dimensional problem, and Section 3.4 examining the block speed scenario. Numerical simulations for the three-dimensional wave equation are presented in Section 3.5, followed by a discussion in Section 4 and concluding remarks in Section 5. Details on discretization and vectorization are provided in Appendix A; Appendix B contains the example of quantum algorithm construction, while Appendix C compares the classical finite difference approach with a developed algorithm in case of standing wave in terms of number of operations. Finally proofs of propositions are in Appendix D.

2. Materials and Methods

The wave equation set in D dimensions, characterized by variable speed and adhering to zero boundary conditions, is expressed as follows

$$\begin{aligned}
&\frac{\partial^2 u(t,\vec{x})}{\partial t^2} = \sum_{j=1}^{D} \frac{\partial}{\partial x_j}\left(c^2(\vec{x})\frac{\partial u(t,\vec{x})}{\partial x_j}\right),\\
&u(t=0,\vec{x}) = f(\vec{x}),\\
&\frac{\partial u(t=0,\vec{x})}{\partial t} = g(\vec{x}),\\
&u(t,x_j=0) = 0, \quad j=1,\ldots,D,\\
&u(t,x_j=l_j) = 0, \quad j=1,\ldots,D.
\end{aligned} \tag{1}$$

where $\vec{x} = (x_1,\ldots,x_D)$ and l_j is the length of the corresponding dimension.

After discretization and vectorization denoted as $\mathrm{vec}(\cdot)$ (details are shown in Appendix A), we can write this in matrix–vector format as

$$\begin{aligned} &\frac{\partial^2}{\partial t^2}\mathrm{vec}(U(t)) = \tilde{L}_D(c(\vec{x}))\mathrm{vec}(U(t)), \\ &\mathrm{vec}(U(t=0)) = \mathrm{vec}(F), \\ &\frac{\partial}{\partial t}\mathrm{vec}(U(t=0)) = \mathrm{vec}(G), \end{aligned} \tag{2}$$

where $U(t)$ is the tensor for the function $u(t, \vec{x})$ in a given domain, F is the tensor for the function $f(\vec{x})$ in a given domain, and G is the tensor for the function $g(\vec{x})$ in a given domain. The discrete operator is

$$\tilde{L}_D(c(\vec{x})) = -\sum_{j=1}^{D}(I_1 \otimes \cdots \otimes B_j \otimes \ldots I_D)S^2(I_1 \otimes \cdots \otimes B_j^T \otimes \ldots I_D), \tag{3}$$

where S is the diagonal matrix with diagonal given by vectorized speed profile tensor C, which is discrete representation of $c(\vec{x})$ in a given domain, that is, $S = \mathrm{diag}(\mathrm{vec}(C))$, and B_j is the first order forward first derivative approximation operator with explicit boundary conditions (first and last rows are zeros, as well as first and last columns). With this choice of B_j, the boundary conditions are incorporated into $\tilde{L}_D$.

2.1. Formulation as Schrödinger Equation

Following [9,16], consider the Schrödinger equation $i\partial_t\psi = H_D\psi$ using the Hamiltonian

$$H_D = \begin{pmatrix} 0 & \tilde{B}_1 & \tilde{B}_2 & \tilde{B}_3 & \ldots & \tilde{B}_D \\ \tilde{B}_1^\dagger & 0 & 0 & 0 & \ldots & 0 \\ \tilde{B}_2^\dagger & 0 & 0 & 0 & \ldots & 0 \\ \tilde{B}_3^\dagger & 0 & 0 & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \tilde{B}_D^\dagger & 0 & 0 & 0 & \ldots & 0 \end{pmatrix}, \tag{4}$$

where $\tilde{B}_j = I_1 \otimes \cdots \otimes B_j \otimes \cdots \otimes I_D$, with $B_j \in \mathbb{R}^{N\times N}$, $N = 2^n$, being the matrix approximation of the first derivative with incorporated zero boundary conditions in a manner consistent with [16]. It is important to mention that typically, the size of matrix B_j may vary across dimensions; however, for the present discussion, we assume it is the same. The matrices $\tilde{B}$ have the dimensions $\mathbb{R}^{N^D\times N^D}$, while the Hamiltonian H_D is sized $\mathbb{R}^{(D+1)N^D\times(D+1)N^D}$. At present, we set $D+1 = 2^{n_D}$; extending this setup is straightforward since H_D can be supplemented with zero matrices, which have no effect on the Pauli decomposition.

In order to incorporate a variable speed profile in this Hamiltonian, we consider

$$\tilde{H}_D = \begin{pmatrix} 0 & \tilde{B}_1S & \tilde{B}_2S & \tilde{B}_3S & \ldots & \tilde{B}_DS \\ S\tilde{B}_1^\dagger & 0 & 0 & 0 & \ldots & 0 \\ S\tilde{B}_2^\dagger & 0 & 0 & 0 & \ldots & 0 \\ S\tilde{B}_3^\dagger & 0 & 0 & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ S\tilde{B}_D^\dagger & 0 & 0 & 0 & \ldots & 0 \end{pmatrix} = \tilde{S}H_D\tilde{S}, \quad \tilde{S} = \mathrm{diag}(I, \underbrace{S, \ldots, S}_{D}), \tag{5}$$

where $S = \mathrm{diag}(\mathrm{vec}(C))$, and I is the identity matrix of the same size.

Differentiating the Schrödinger equation over time, given the Hamiltonian in (5), we get

$$\frac{d^2}{dt^2}\begin{pmatrix}\mathrm{vec}(U_V)\\ \mathrm{vec}(G_1)\\ \vdots\\ \mathrm{vec}(G_D)\end{pmatrix} = -\begin{pmatrix}-\tilde{L}_D & 0 & \dots & 0\\ 0 & A_{11} & \dots & A_{1D}\\ \vdots & \vdots & \dots & \vdots\\ 0 & A_{D1} & \dots & A_{DD}\end{pmatrix}\begin{pmatrix}\mathrm{vec}(U_V)\\ \mathrm{vec}(G_1)\\ \vdots\\ \mathrm{vec}(G_D)\end{pmatrix}, \tag{6}$$

where $\tilde{L}_D = -\sum_{j=1}^{D}\tilde{B}_j S^2 \tilde{B}_j^\dagger$, $\mathrm{vec}(G_1), \dots, \mathrm{vec}(G_D)$ are some additional components in the wavefunction and $A_{ij} = S\tilde{B}_i^\dagger \tilde{B}_j S$. Note that if matrix B is a finite difference approximation of the first derivative, then its negative transpose, and $-B^\dagger$ is also an approximation. Specifically, $-B^\dagger$ employs the mirror scheme: it yields a backward scheme if B uses a forward scheme, a forward scheme if B uses a backward scheme, and remains a central scheme if B is central. We can see now that the first component $\mathrm{vec}(U_V)$ indeed evolves according to (2). One way to set the initial condition of the Schrödinger equation is

$$\begin{aligned}&\mathrm{vec}(U_V(t=0)) = \mathrm{vec}(F),\\ &\mathrm{vec}\big(G_j(t=0)\big) = \frac{1}{D} i(\tilde{B}_j S)^{-1}\mathrm{vec}(G), \quad j = 1, \dots D.\end{aligned} \tag{7}$$

Putting this together, the quantum algorithm for solving (1) reduces to preparing and evolving the state

$$|\psi(t)\rangle = \exp\big(-it\tilde{H}_D\big)|\psi(0)\rangle, \quad |\psi(t)\rangle = \begin{pmatrix}\mathrm{vec}(U_V)(t)\\ \mathrm{vec}(G_1)(t)\\ \vdots\\ \mathrm{vec}(G_D)(t)\end{pmatrix}. \tag{8}$$

We then postselect on the first component of ($|\psi(t)\rangle$). While postselection, initialization, and the design of a suitable cost function are all very challenging tasks, in this work we focus on implementing the propagator $\exp\big(-it\tilde{H}_D\big)$ using one- and two-qubit gates.

Remark 1. *If the right hand side of the wave Equation (1) is given by* $c^2(\vec{x})\sum_{j=1}^{D}\left(\frac{\partial^2 u(t,\vec{x})}{\partial x_j^2}\right)$, *one can use* $\tilde{S} = \mathrm{diag}(S, \underbrace{I, \dots, I}_{D})$ *and consider the first component in the wavefunction as* $\mathrm{vec}(U_V) = S^{-1}\mathrm{vec}(U_W)$. *This way the component* $\mathrm{vec}(U_W)$ *changes over time according to* $L_D = -S^2\sum_{j=1}^{D}\tilde{B}_j\tilde{B}_j^\dagger$, *which is effectively a form of* $c^2(\vec{x})\sum_{j=1}^{D}\left(\frac{\partial^2 u(t,\vec{x})}{\partial x_j^2}\right)$. *Initial conditions in this case can be set as* $\mathrm{vec}(U_V(t=0)) = S^{-1}\mathrm{vec}(F)$, *and* $\mathrm{vec}\big(G_j(t=0)\big) = \frac{1}{D}i(S^2\tilde{B}_j)^{-1}\mathrm{vec}(G), j = 1, \dots, D$. *This does not affect the results presented here since they are based on decompositions of* S *and* H_D.

2.2. Propagator Implementation Technique

To execute the propagator $\exp\big(-it\tilde{H}_D\big)$ on a quantum device, it must be broken down into operations that can be performed on such a device. This can be achieved by expressing $\tilde{H}_D$ as a sum of Pauli strings, which are tensor products of Pauli matrices, and then applying the Trotter formula [17,18]. For comprehensive definitions, the reader is directed to Section 2 of [15]; here, we will only present the key concepts.

To express Pauli strings with X and Z matrices, we use bit strings $\mathbf{z}, \mathbf{x} \in \mathbb{B}^n$; with them, an arbitrary Pauli string can be defined as the image of the extended Pauli string operator (Walsh function) $\hat{W} : \mathbb{B}^n \times \mathbb{B}^n \to \mathcal{P}_n$ as follows

$$\hat{W}(\mathbf{x}, \mathbf{z}) = \imath^{\mathbf{x}\cdot\mathbf{z}} X^{\mathbf{x}} Z^{\mathbf{z}} = \bigotimes_{j=1}^{n} \imath^{x_j z_j} X^{x_j} Z^{z_j}, \tag{9}$$

with the matrix product between $X^{\mathbf{x}}$ and $Z^{\mathbf{z}}$ and $\mathbf{x} \cdot \mathbf{z} = \sum_{l=1}^{n} x_l z_l$ denoting the inner product of two bitstrings. It can be seen that the Walsh function is bijective. Thus, each Pauli string can be encoded with a unique pair $(\mathbf{x}, \mathbf{z})$, and decomposition in the Pauli basis can be rewritten as

$$\tilde{H}_D = \frac{1}{2^n} \sum_{\mathbf{x},\mathbf{z} \in \mathbb{B}^n} \beta_{\mathbf{x},\mathbf{z}} \hat{W}(\mathbf{x}, \mathbf{z}), \tag{10}$$

where $\beta_{\mathbf{x},\mathbf{z}} \in \mathbb{C}$, and as shown in the proof of Proposition 4 of [15], the coefficients can be expressed in the form

$$\beta_{\mathbf{x},\mathbf{z}} = \imath^{\mathbf{x}\cdot\mathbf{z}} \sum_{\mathbf{p} \in \mathbb{B}} (-1)^{\mathbf{z}\cdot\mathbf{p}} \cdot h_{\mathbf{p},\mathbf{p}\oplus\mathbf{x}}, \tag{11}$$

where $\oplus$ is a binary XOR (addition modulo 2) operation and $h_{\mathbf{p},\mathbf{p}\oplus\mathbf{x}}$ are elements of $\tilde{H}_D$, which makes it possible to compute coefficients in decomposition with $O(n2^n)$ operations.

As the strings in the decomposition do not all commute with each other, the Trotter formula can be utilized, offering different accuracy levels depending on the formula's order. This approach is employed to manage the non-commutative characteristics of the operators involved in the decomposition. Building upon prior studies [15,16], we express $\tilde{H}_D$ as $\sum_{\gamma=1}^{\Gamma} H_\gamma$, where each H_γ consists of mutually commuting Pauli strings. The Trotter formulas for orders 1, 2, and higher even orders $p = 2k$, where $k = 2, 3, 4, \ldots$, are represented as

$$S_1(t) = e^{-itH_\Gamma} \ldots e^{-itH_1}, \tag{12}$$

$$S_2(t) = e^{\frac{-it}{2}H_1} \ldots e^{\frac{-it}{2}H_\Gamma} e^{\frac{-it}{2}H_\Gamma} \ldots e^{\frac{-it}{2}H_1}, \tag{13}$$

$$S_{2k}(t) = S_{2k-2}^2(s_k t) S_{2k-2}((1-4s_k)t) S_{2k-2}^2(s_k t), \tag{14}$$

where $s_k = 1/(4 - 4^{1/(2k-1)})$. The approximation accuracy is determined by the number of Trotterization steps r, the evolution time t, the order p of the Trotter formula, and the Hamiltonian norm.

3. Results

This section outlines our findings in the form of propositions and corollaries. We express the propagator for the multidimensional wave equation $\exp(-it\tilde{H}_D)$ using single and two qubit operations as follows:

1. We expand on the decomposition of Pauli strings as described in [16] to apply it to the wave equation in D dimensions (Proposition 1).
2. We present a method for efficiently diagonalizing mutually commuting groups that emerge during decomposition (Proposition 2).
3. We establish an upper bound and scaling on the complexity involved in solving the multidimensional wave equation with a variable speed profile using the Trotterization algorithm in terms of single- and two-qubit operations (Corollaries 2 and 3).
4. We examine the practical scenario of a wave equation in D dimensions, characterized by a variable speed profile with a block structure and establish its upper bound and scaling (Corollary 4).

5. We demonstrate the algorithm's numerical performance on the three-dimensional wave equation with a block-structured speed profile (Section 3.5). We also compare the finite-difference method (FDM) with the quantum algorithm presented here for a three-dimensional standing wave, reporting the number of operations required to achieve the same accuracy (Appendix C).

3.1. Pauli String Decomposition

We extend the results presented in Proposition 1 from [16], particularly focusing on the case of a block Hamiltonian H with dimensions $(D+1)N \times (D+1)N$.

Proposition 1 (Decomposition of a block Hamiltonian)**.** *The only Pauli strings that can have non-zero coefficients in the decomposition of matrix*

$$H = \begin{pmatrix} 0 & B_1 & B_2 & B_3 & \dots & B_D \\ B_1^\dagger & 0 & 0 & 0 & \dots & 0 \\ B_2^\dagger & 0 & 0 & 0 & \dots & 0 \\ B_3^\dagger & 0 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ B_D^\dagger & 0 & 0 & 0 & \dots & 0 \end{pmatrix}, \tag{15}$$

where $B_p \in \mathbb{C}^{N\times N}$, $p = 1, \dots, D$, are d-diagonal matrices with $N = 2^n$, described by $W(\mathbf{x}_{p,k,j}, \mathbf{z})$, where $\mathbf{z} \in \mathbb{B}^{n+\left|\hat{\mathbb{B}}(D)\right|}$ and

$$\mathbf{x}_{p,k,j} = \hat{\mathbb{B}}_l\left(\left|\hat{\mathbb{B}}(D)\right|, p\right) * \hat{\mathbb{B}}_l\left(n-s, 2^j-1\right) * \hat{\mathbb{B}}_l(s, 2^s-k)\,, \tag{16}$$

where $p \in \{1, \dots, D\}$, $k \in \{1, \dots, d\}$, $j \in \{1, \dots, n-s\}$, $s = \lceil \log_2(k) \rceil$, $d \leq 2^{n-1}$. Moreover, the main diagonals of B_p are represented by $W(\mathbf{x}_{p,0}, \mathbf{z})$, with

$$\mathbf{x}_{p,0} = \hat{\mathbb{B}}_l\left(\left|\hat{\mathbb{B}}(D)\right|, p\right) * \hat{\mathbb{B}}_l(n, 0), \tag{17}$$

where $p \in \{1, \dots, D\}$.

Here, function $\left|\hat{\mathbb{B}}(D)\right|$ returns the number of bits necessary to represent D in binary form. Meanwhile, $\hat{\mathbb{B}}_l(a, b)$ provides the binary form of b with a specified length a, padding with leading zeros if necessary. The symbol $*$ denotes the concatenation of binary strings.

It is noteworthy that if $D+1$ is not a power of 2, this proposition can still be utilized by appending matrices $B_p = 0$ for $p = D+1, \dots, D'$, where D' is defined as $2^{\lceil \log_2(D) \rceil}$. The d-diagonal matrices referenced in the proposition allow for enhanced accuracy of differential operators. Specifically, the three-diagonal matrices with d=1 (where d denotes the count of diagonals above the main diagonal) will represent the first-order accurate differential operators.

All Pauli strings with non-zero coefficients in the decomposition are part of the sets labeled by $\mathbf{x}_{p,k,j}$ and $\mathbf{x}_{p,0}$, as described below:

$$\begin{aligned} S_{p,k,j} &= \{W(\mathbf{x}_{p,k,j}, \mathbf{z}) \mid \mathbf{z} \in \mathbb{B}^{n+\left|\hat{\mathbb{B}}(D)\right|}\}, \\ S_{p,0} &= \{W(\mathbf{x}_{p,0}, \mathbf{z}) \mid \mathbf{z} \in \mathbb{B}^{n+\left|\hat{\mathbb{B}}(D)\right|}\}. \end{aligned} \tag{18}$$

In each set represented by $\mathbf{x}_{p,k,j}$ and $\mathbf{x}_0$, the elements are generated with $\mathbf{z} \in \mathbb{B}^{n+\left|\hat{\mathbb{B}}(D)\right|}$, which are the binary representation of numbers from 0 to $2^{n+\left|\hat{\mathbb{B}}(D)\right|} - 1$ of length $n + \left|\hat{\mathbb{B}}(D)\right|$. Therefore, the cardinality of each set is $2^{n+\left|\hat{\mathbb{B}}(D)\right|}$. Furthermore, if $D+1$ is a power of 2,

the cardinality can also be expressed as $(D+1)2^n$. Utilizing this proposition alongside Proposition 2 from [16], we formulate the ensuing corollary.

Corollary 1 (Number of sets in a block Hamiltonian)**.** *The total count of sets $S_{p,k,j}$ (including $S_{p,0}$) in the decomposition of Hamiltonian (15) with a d-band matrices $B_p \in \mathbb{C}^{N\times N}$, $p = 1,\ldots,D$, where $N = 2^n$ is given by*

$$s(D,d,n) = D\left(2^{\left|\hat{\mathbb{B}}(d)\right|} + \left[n - \left|\hat{\mathbb{B}}(d)\right|\right]d\right), \tag{19}$$

where $\left|\hat{\mathbb{B}}(d)\right|$ is the binary length of d.

We group the sets from Equation (18) into mutually commuting subsets, a result that aligns with previous findings [15,16].The assignment to a subset is determined by whether a Pauli string contains an odd or even number of Y matrices (i.e., the value of $\mathbf{x}\cdot\mathbf{z} \mod 2$), as stated in Corollary 2 of [16]. In the specific case of a real-valued matrix, all Pauli strings have an even number of Y matrices, which leads to $s(D,d,n)$ mutually commuting sets.

3.2. Mutual Diagonalization

In Proposition 1 we have provided strings $\mathbf{x}_{p,k,j}$, which generate mutually commuting sets based on parity of Y operators (value of $\mathbf{x}\cdot\mathbf{z} \pmod 2$). We outline the diagonalization process for these sets, drawing upon [19,20], and present it as the subsequent proposition.

Proposition 2 (Diagonalization of sets)**.** *Given the set of mutually commuting operators, characterized by a string $\mathbf{x} \in \mathbb{B}^n$, that is, $\{W(\mathbf{x},\mathbf{z}) \mid \mathbf{z} \in \mathbb{B}^n, \mathbf{x}\cdot\mathbf{z} = 0 \pmod 2\}$, or $\{W(\mathbf{x},\mathbf{z}) \mid \mathbf{z} \in \mathbb{B}^n, \mathbf{x}\cdot\mathbf{z} = 1 \pmod 2\}$, one can construct diagonalization operator $\hat{D}$, such that $W(\mathbf{x},\mathbf{z}) = (-1)^r \hat{D}^\dagger W(\mathbf{0},\tilde{\mathbf{z}})\hat{D}$, with $r = \left(\frac{\mathbf{x}\cdot\mathbf{z}+(\mathbf{x}\cdot\mathbf{z} \bmod 2)}{2}\right)$ as*

$$\begin{aligned} \hat{D} &= H_{k_1}\prod_{j=2}^{M} CNOT(k_1,k_j), \quad \text{if } \mathbf{x}\cdot\mathbf{z} = 0 \pmod 2, \\ \hat{D} &= H_{k_1}S_{k_1}\prod_{j=2}^{M} CNOT(k_1,k_j), \quad \text{if } \mathbf{x}\cdot\mathbf{z} = 1 \pmod 2, \end{aligned} \tag{20}$$

where $CNOT(c,t)$ is a controlled-NOT operation with control on the c-th qubit and target on the t-th qubit; H_k and S_k are Hadamard and phase gates acting on the k-th qubit, respectively. The product runs over all indices k_j of the nonzero bits in $\mathbf{x}$, where M is the total number of these bits and k_1 is the index of the first nonzero bit. This procedure yields the string $\tilde{\mathbf{z}}$, which is identical to $\mathbf{z}$ except that its k_1-th bit is set to 1.

This proposition enables the diagonalization of any set of mutually commuting operators using only the corresponding string $\mathbf{x}$. The number of one- and two-qubit gates required for this diagonalization is determined by the parity of $\mathbf{x}\cdot\mathbf{z}$: it is M if $\mathbf{x}\cdot\mathbf{z} = 0 \mod 2$ and $M+1$ if $\mathbf{x}\cdot\mathbf{z} = 1 \mod 2$, where M is the number of nonzero bits (Hamming weight) in $\mathbf{x}$. Additionally, the proposition provides the outcome of the diagonalization, that is, string $\tilde{\mathbf{z}}$.

3.3. Scaling of Multidimensional Wave Equation Quantum Algorithm

Using general results presented in previous sections, we formulate the following proposition made specifically for the multidimensional wave equation.

Proposition 3 (Decomposition of multidimensional wave equation Hamiltonian)**.** *The Hamiltonian for the D-dimensional wave problem stated as* (1) *can be written as*

$$\tilde{H}_D = \begin{pmatrix} 0 & \tilde{B}_1 S & \tilde{B}_2 S & \tilde{B}_3 S & \dots & \tilde{B}_D S \\ S\tilde{B}_1^\dagger & 0 & 0 & 0 & \dots & 0 \\ S\tilde{B}_2^\dagger & 0 & 0 & 0 & \dots & 0 \\ S\tilde{B}_3^\dagger & 0 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ S\tilde{B}_D^\dagger & 0 & 0 & 0 & \dots & 0 \end{pmatrix} = \tilde{S} H_D \tilde{S}, \quad \tilde{S} = \mathrm{diag}(I, \underbrace{S, \dots, S}_{D}), \tag{21}$$

where $\tilde{B}_j = I_1 \otimes \cdots \otimes B_j \otimes \dots I_D$ *with* $B_j \in \mathbb{R}^{N \times N}$, $j = 1, \dots, D$ *- d-diagonal matrices, with* $N = 2^n$*, and diagonal matrix* $S \in \mathbb{R}^{N^D \times N^D}$*. Given Pauli decomposition* $\tilde{S} = \sum_{j=1}^{T} \alpha_j Z^{\mathbf{z}_j}$*, where* $Z^{\mathbf{z}} = \bigotimes_{k=1}^{Dn+\left|\hat{\mathbb{B}}(D)\right|} Z^{z_k}$*, with* z_k *being the k-th bit in* $\mathbf{z}$*, the number of Pauli strings that can have non-zero coefficients in the decomposition of matrix* $\tilde{H}_D$ *is bounded by* $g_H \leq \Gamma \min(T^2 K, (D+1)N^D/2)$*, where* Γ *is the number of mutually commuting sets in* H_D*, and* K *is the number of Pauli strings in the single set.* T *is the number of Pauli strings in the decomposition of* $\tilde{S}$*.*

The Hamiltonian H_D can be directly subjected to Proposition 1 and Corollary 1 with no alterations. The form of the Pauli strings in the decomposition remains essentially the same. To convert H into H_D, one simply needs to insert extra identity (I) matrices at the qubit locations that align with the structure of the $\tilde{B}_j$ operators. In the binary representation of the Pauli strings using vectors $\mathbf{x}$ and $\mathbf{z}$, this operation corresponds to adding zeros at the appropriate bit locations.

It is also observed that the number of mutually commuting sets in $\tilde{H}_D$ is identical to that in H_D because $\tilde{S}$ represents a diagonal matrix. According to Corollary 1, this count is denoted by $\Gamma = s(D, d, n)$, owing to the real-valued nature of H_D. If the number of diagonals d above the main is significantly less than N, a situation common with finite difference approximation matrices B_j, the number of commuting sets can be approximated by $\Gamma = O(Ddn)$. The count of Pauli strings K contained within a single commuting set can be bounded by $K \leq 2^{n+\left|\hat{\mathbb{B}}(D)\right|-1} = (D+1)N/2$ since only strings with an even number of Y Pauli matrices participate in the decomposition. Consequently, the overall number of Pauli strings that appear in $\tilde{H}_D = \tilde{S} H_D \tilde{S}$ can be expressed as $O(D^2 dnN \min(T^2, N^{D-1}))$, where T denotes the number of Pauli strings used in the decomposition of $\tilde{S}$. Furthermore, in cases where $T^2 \leq N^{D-1}$, the estimation may be modified to

$$g_H = O(D^2 dn T^2 N). \tag{22}$$

From this result, we can estimate the number of one- and two-qubit gates required for a single Trotter step that realizes the time evolution operator $\exp\left(-i\tilde{H}_D t\right)$. The construction of one Trotter step requires the diagonalization of each of the $s(D, d, n)$ mutually commuting sets of Pauli strings. Recall that a Pauli string $\mathbf{x}$ can contain non-zero bits solely within the initial $\left|\hat{\mathbb{B}}(D)\right|$ places and the n locations specified by j in $\tilde{B}_j = I_1 \otimes \cdots \otimes B_j \otimes \dots I_D$. Based on Proposition 2 and given the condition $\mathbf{x} \cdot \mathbf{z} \equiv 0 \pmod 2$, the count of one- and two-qubit gates necessary for the diagonalization and reverse diagonalization of a single set is limited by

$$g_d \leq 2(n + \left|\hat{\mathbb{B}}(D)\right|). \tag{23}$$

Diagonalized Pauli strings are composed solely of Z operators. These can be executed using multi-qubit rotations denoted by $R_{\tilde{\mathbf{z}}}(\theta) \equiv \exp\left(-i\frac{\theta}{2} Z^{\tilde{\mathbf{z}}}\right)$, where $\tilde{\mathbf{z}}$ represents the modified string $\mathbf{z}$ in each respective diagonal group. The sequence of these rotations is organized in such a way as to optimize their implementation by reducing the number

of CNOT gates required, achieved through minimizing the Hamming distance between successive $\tilde{\mathbf{z}}$ strings [21] (see a circuit example for a single group in Appendix B). Without particular assumptions about the configuration of $\tilde{S}$, we employ a worst-case scenario in which every $\tilde{\mathbf{z}}$ sequence is composed entirely of ones, with no cancellations occurring. This provides an upper limit of $2(Dn + \left|\hat{\mathbb{B}}(D)\right| - 1)$ CNOT gates for each rotation since the maximum number of CNOTs needed for an m-qubit rotation is $2(m-1)$ [21]. Hence, the total number of gates necessary for realizing all rotations within a single diagonal configuration is bounded by $g_z \leq 2\Big(Dn + \left|\hat{\mathbb{B}}(D)\right|\Big) \min(T^2K, (D+1)N^D/2)$.

By bringing these components together, the total number g_1 of one- and two-qubit gates required for each single step of the first-order Trotter–Suzuki decomposition to approximate $\exp\left(-i\tilde{H}_D t\right)$ is constrained by

$$g_1 \leq \Gamma \cdot (g_d + g_z), \tag{24}$$

where

- $\Gamma = s(D,d,n) = D\Big(2^{|\hat{\mathbb{B}}(d)|} + \Big(n - \left|\hat{\mathbb{B}}(d)\right|\Big)d\Big)$ represents the number of sets of Pauli strings that internally mutually commute;
- $g_d \leq 2(n + \left|\hat{\mathbb{B}}(D)\right|)$ is the gate count for diagonalization operators per set;
- $g_z \leq (Dn + \left|\hat{\mathbb{B}}(D)\right|)(D+1)N\min(T^2, N^{D-1})$ represents the gate count for implementing the diagonal rotations per set. We relied on the estimation $K \leq (D+1)N/2$ as suggested by Proposition 1.

Therefore , the explicit bound in case $T^2 < N^{D-1}$ for one step of first order Trotter formula is

$$g_1 \leq s(D,d,n)\Big[2\Big(n + \left|\hat{\mathbb{B}}(D)\right|\Big) + \Big(Dn + \left|\hat{\mathbb{B}}(D)\right|\Big)N(D+1)T^2\Big]. \tag{25}$$

For higher-order Trotter–Suzuki formulas of even order p [17,18], the gate count scales as

$$g_p = 2 \cdot 5^{\lfloor p/2 \rfloor - 1} g_1, \quad \text{for } p = 2,4,6,\ldots \tag{26}$$

We formalize the scaling for a single Trotter step in the following corollary.

Corollary 2 (One Trotter step scaling)**.** *Given the Hamiltonian*

$$\tilde{H}_D = \begin{pmatrix} 0 & \tilde{B}_1 S & \tilde{B}_2 S & \tilde{B}_3 S & \ldots & \tilde{B}_D S \\ S\tilde{B}_1^\dagger & 0 & 0 & 0 & \ldots & 0 \\ S\tilde{B}_2^\dagger & 0 & 0 & 0 & \ldots & 0 \\ S\tilde{B}_3^\dagger & 0 & 0 & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ S\tilde{B}_D^\dagger & 0 & 0 & 0 & \ldots & 0 \end{pmatrix} = \tilde{S} H_D \tilde{S}, \quad \tilde{S} = \mathrm{diag}(I, \underbrace{S, \ldots, S}_{D}), \tag{27}$$

where $\tilde{B}_j = I_1 \otimes \cdots \otimes B_j \otimes \ldots I_D$ *with* $B_j \in \mathbb{R}^{N\times N}$, $j = 1,\ldots,D$ *- d-diagonal matrices, with* $N = 2^n$, *and diagonal matrix* $\tilde{S} = \sum_{j=1}^{T} \alpha_j Z^{\mathbf{z}_j} \in \mathbb{R}^{N^D \times N^D}$ *with T strings in decomposition. The number of one- and two-qubit gates for one step of first order Trotter formula for* $\exp\left(-i\tilde{H}_D t\right)$ *with assumptions* $d \ll N$ *and* $T^2 < N^{D-1}$ *scales as*

$$\begin{aligned} g_1 &= O\Big(D^3 d n^2 N T^2\Big) \\ g_p &= O\Big(5^{\lfloor p/2 \rfloor} D^3 d n^2 N T^2\Big), \quad p = 2,4,6\ldots. \end{aligned} \tag{28}$$

This process can be made more efficient with a few methods. To start, the Hamming distance between neighboring bitstrings $\tilde{\mathbf{z}}$ should be reduced. This configuration aids in eliminating CNOT gates used for executing the multi-qubit $R_{\tilde{\mathbf{z}}}$ rotations. Secondly, an analogous improvement is realized by tactically reordering the sets, which facilitates the elimination of CNOT gates produced during the diagonalization process. Moreover, the arrangement of these sets constituting $\tilde{S}$ can be utilized to lessen the number of gates, as illustrated in Section 3.4.

Finally, by employing Corollary 2, we shall illustrate the scaling of one- and two-qubit gates required for executing the propagator $\exp\left(-i\tilde{H}_D t\right)$ using the Trotter formula of order p. The scaling can be expressed as $g_{tr} = g_p r$, where g_p denotes the count of gates per trotter step of p-th order, and r represents the total number of steps needed to achieve an error of $\epsilon = \left\|\exp\left(-i\tilde{H}_D t\right) - U(t,r,p)\right\|_2$. In this context, $U(t,r,p)$ serves as the Trotter approximation for $\exp\left(-i\tilde{H}_D t\right)$. To determine r, we utilize the scaling outlined by the authors in [17], specifically

$$r = O\left(\left(2\Gamma 5^{\lfloor p/2\rfloor - 1}\left\|\tilde{H}_D\right\| t\right)^{1+1/p}(1/\epsilon)^{1/p}\right), \tag{29}$$

where Γ denotes the quantity of easy to implement Hamiltonians (in our case, we choose it to be mutually commuting sets) within $\tilde{H}_D$. We can approximate the Hamiltonian norm by $\left\|\tilde{H}_D\right\| = O(dN)$ because the matrices B_j in $\tilde{H}_D$ function as finite difference operators, which inherently contain the inverse of the discretization step size $1/h = O(N)$. The subsequent corollary outlines the scaling requirements essential for constructing a quantum circuit that sets up the propagator for the D-dimensional wave equation, denoted as $\exp\left(-i\tilde{H}_D t\right)$, with a margin of error ϵ.

Corollary 3 (Scaling for multidimensional wave equation quantum algorithm)**.** *Given the Hamiltonian for D-dimensional wave equation as*

$$\tilde{H}_D = \begin{pmatrix} 0 & \tilde{B}_1 S & \tilde{B}_2 S & \tilde{B}_3 S & \dots & \tilde{B}_D S \\ S\tilde{B}_1^\dagger & 0 & 0 & 0 & \dots & 0 \\ S\tilde{B}_2^\dagger & 0 & 0 & 0 & \dots & 0 \\ S\tilde{B}_3^\dagger & 0 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ S\tilde{B}_D^\dagger & 0 & 0 & 0 & \dots & 0 \end{pmatrix} = \tilde{S} H_D \tilde{S}, \quad \tilde{S} = \mathrm{diag}(I, \underbrace{S, \dots, S}_{D}), \tag{30}$$

where $\tilde{B}_j = I_1 \otimes \cdots \otimes B_j \otimes \dots I_D$ *with* $B_j \in \mathbb{R}^{N\times N}, j = 1, \dots, D$ *- d-diagonal matrices, with* $N = 2^n$*, and diagonal matrix* $\tilde{S} = \sum_{j=1}^{T} \alpha_j Z^{\mathbf{z}_j} \in \mathbb{R}^{N^D \times N^D}$ *with T strings in decomposition, the number of one- and two-qubit gates required to implement a p-th order Trotter approximation* $U(t,r,p)$ *with r steps, satisfying* $\left\|e^{-i\tilde{H}_D t} - U(t,r,p)\right\|_2 \leq \epsilon$ *scales as*

$$g_{tr} = O\left(t 5^p D^4 d^3 n^3 N^2 T^2 \left(\frac{2}{5} D d^2 n N t/\epsilon\right)^{1/p}\right), \tag{31}$$

assuming $d \ll N$ *and* $T^2 < N^{D-1}$.

The suggested approach mitigates the "curse of dimensionality" in a manner similar to the oracle-based method discussed in [9]. According to Corollary 2, the gate complexity for one Trotter step predominantly scales with the NT^2 factor, which varies linearly with the system size, where $N = 2^n$. Nevertheless, the total complexity is primarily determined by the number of steps needed to reach the desired precision. When utilizing a p-th order Trotter formula, the dominant scaling factor is $N^{2+1/p}T^2$. The extra scaling related to N

stems from the finite difference approximation, which requires a step size that is contingent upon N.

3.4. Scaling of Multidimensional Wave Equation Quantum Algorithm with Block-Model Speed Profile

In this section, we adopt a block-model representation in which the domain is partitioned into regions of constant wave speed (acoustic approximation). This assumption is widely used because it simplifies the forward problem, provides clear correspondence between model parameters and geological interfaces, and enables efficient numerical implementation. However, it should be noted that real Earth materials rarely consist of perfectly homogeneous blocks. In practice, velocity and density typically vary smoothly due to compaction, mineral composition, and fluid content, and fine-scale heterogeneity introduces scattering and waveform complexity that are not captured in block models [22]. As a result, the block approximation can overestimate reflection amplitudes at sharp interfaces and underestimate energy redistribution into scattered or coda waves. Nevertheless, when seismic wavelengths are large relative to heterogeneity scales, block models offer a reasonable first-order description of wave kinematics, particularly for traveltime and phase analyses.

Consider a computational domain of size $2^{n_x} \times 2^{n_y} \times 2^{n_z}$, discretized uniformly into 2^{m_x}, 2^{m_y}, and 2^{m_z} segments along the x, y, and z axes, respectively. The total number of distinct material blocks, and thus the number of independent variables in the system, is given by $T = 2^{m_x+m_y+m_z}$. Moreover, the vectorized velocity profile $S = \mathrm{diag}(\mathrm{vec}(C))$, with C being a tensor representation of $c(\vec{x})$, can be expressed as a linear combination of Pauli operators as follows:

$$S = \sum_{\substack{\mathbf{z}_x \in \mathbb{B}^{m_x}, \\ \mathbf{z}_y \in \mathbb{B}^{m_y}, \\ \mathbf{z}_z \in \mathbb{B}^{m_z}}} \alpha_{\mathbf{z}_x,\mathbf{z}_y,\mathbf{z}_z} \underbrace{Z^{\mathbf{z}_x} \otimes I^{\otimes n_x - m_x}}_{n_x} \otimes \underbrace{Z^{\mathbf{z}_y} \otimes I^{\otimes n_y - m_y}}_{n_y} \otimes \underbrace{Z^{\mathbf{z}_z} \otimes I^{\otimes n_z - m_z}}_{n_z}, \tag{32}$$

where $Z^{\mathbf{z}} = \bigotimes_{k=1} Z^{z_k}$, with z_k being the k-th bit in $\mathbf{z}$. This formulation demonstrates that the operator S is decomposed into a sum of T Pauli terms. Each term corresponds to a diagonal Pauli string (composed of I and Z operators) acting on the $m = m_x + m_y + m_z$ qubits that encode the model parameterization, while identity operators act on the remaining $(n_x + n_y + n_z - m)$ qubits that correspond to a size of block in particular direction.

The number of terms in this decomposition is precisely equal to the number of variables, T, and the generalization to the multidimensional case is straightforward. To estimate the gate complexity of implementing the operator $\tilde{S}$, one must account for an ancillary register of size $\lceil \log_2(D+1) \rceil = \left| \hat{\mathbb{B}}(D) \right|$ qubits. Crucially, the block structure of S given by Equation (32) provides a significant reduction in computational complexity. We leverage this by reformulating Corollaries 2 and 3 for the block-model speed profile.

Corollary 4 (Scaling for block-model speed profile)**.** *Given the Hamiltonian*

$$\tilde{H}_D = \begin{pmatrix} 0 & \tilde{B}_1 S & \tilde{B}_2 S & \tilde{B}_3 S & \dots & \tilde{B}_D S \\ S\tilde{B}_1^\dagger & 0 & 0 & 0 & \dots & 0 \\ S\tilde{B}_2^\dagger & 0 & 0 & 0 & \dots & 0 \\ S\tilde{B}_3^\dagger & 0 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ S\tilde{B}_D^\dagger & 0 & 0 & 0 & \dots & 0 \end{pmatrix} = \tilde{S} H_D \tilde{S}, \quad \tilde{S} = \mathrm{diag}(I, \underbrace{S, \dots, S}_{D}), \tag{33}$$

where $\tilde{B}_j = I_1 \otimes \cdots \otimes B_j \otimes \ldots I_D$ *with* $B_j \in \mathbb{R}^{N\times N}$, $j = 1, \ldots, D$ *- d-diagonal matrices, with* $N = 2^n$, *and a real diagonal matrix* S, *given as*

$$S = \sum_{\mathbf{z}_j \in \mathbb{B}^m, j=1,\ldots,D} \alpha_{\mathbf{z}_j,\ldots,\mathbf{z}_D} \underbrace{Z^{\mathbf{z}_1} \otimes I^{\otimes n-m}}_{n} \otimes \cdots \otimes \underbrace{Z^{\mathbf{z}_D} \otimes I^{\otimes n-m}}_{n}. \tag{34}$$

The number of one- and two-qubit gates for one step of first order Trotter formula for $\exp\left(-i\tilde{H}_D t\right)$ *with assumption* $d \ll N$ *scales as*

$$\begin{aligned} g_1 &= O\left(D^2 dnN 2^{(D-1)m}\right) \\ g_p &= O\left(5^{\lfloor p/2 \rfloor} D^2 dnN 2^{(D-1)m}\right), \quad p = 2, 4, 6 \ldots. \end{aligned} \tag{35}$$

Consequently, under the same assumptions the number of one- and two-qubit gates required to implement a p-th order Trotter approximation $U(t, r, p)$ *with r steps, satisfying* $\left\| e^{-i\tilde{H}_D t} - U(t, r, p) \right\|_2 \leq \epsilon$, *scales as*

$$g_{tr} = O\left(t 5^p D^3 d^3 n^2 N^2 2^{(D-1)m} \left(\frac{2}{5} D d^2 n N t/\epsilon \right)^{1/p} \right). \tag{36}$$

The manner in which the system scales is dictated by the configuration of the resulting sets of Pauli operators. According to Proposition 1, one can systematically generate the entire set of possible bit strings $\mathbf{x}$.

Given a binary string $\mathbf{x}$, the strings $\mathbf{z}$ are formulated in the following manner:

- In the case where $m = 0$, the non-zero elements of the $\mathbf{z}$ strings are constrained to the positions directly specified by the structure of the matrix $\tilde{B}_j$. For example, given $\tilde{B}_1 = B \otimes I \otimes \cdots \otimes I$, the Pauli strings associated with $\tilde{B}_1$ are the same as in decomposition of B but are padded with identity operators (I). This corresponds to appending zeros to both the $\mathbf{x}$ and $\mathbf{z}$ strings.
- For a general parameter $m \geq 0$, if the velocity profile adheres to the structure defined in Equation (34), the number of positions in the $\mathbf{z}$ strings that can support non-zero values increases. Specifically, an additional Dm bits must be considered. These correspond to the first m bits in each of the D spatial directions within the $\mathbf{z}$ string.

Therefore, the number of Pauli strings in a single set can be bounded as $K' \leq 2^{n+\left|\hat{\mathbb{B}}(D)\right|+(D-1)m-1}$, where the term $(D-1)m$ appears because, for a single dimension in the considered set, all bits are already accounted for in the implementation of B. Moreover, in this scenario, Gray code ordering can be applied within each mutually commuting set, allowing the elimination of certain CNOT gates and thus reducing the number of gates required to implement rotations for a single group to $g_z \leq 2K'$.

To illustrate this we present the full set of strings $\mathbf{x}$ and corresponding masks for $\mathbf{z}$ as described above for a three-dimensional wave equation—discretized with 2 qubits per dimension and with finite difference operators approximated by tridiagonal matrices ($d = 1$) in Table 1 for different values of m.

The number of $\mathbf{z}$ strings that must be considered can be reduced by leveraging the decomposition of the specific matrix H_D. However, in this work, we consider an arbitrary diagonal structure as a more general approach that enables the explicit encoding of boundary conditions, as demonstrated in [16].

Table 1. Example of all possible Pauli strings for the three-dimensional wave equation, discretized with 2 qubits per dimension and with finite difference operators approximated by tridiagonal matrices ($d = 1$). The strings $\mathbf{x}_{p,k,j}$ (where $k = 1$ due to $d = 1$) are generated according to Proposition 1. The corresponding masks for the $\mathbf{z}$ strings are shown for different values of m in the block speed model; a "$*$" denotes a position that can be either 0 or 1. Bits are grouped for readability: the first $\left|\hat{\mathbb{B}}(D)\right| = 2$ bits are used for the Hamiltonian construction, and each subsequent block of $n = 2$ bits encodes a spatial direction.

$\mathbf{x}_{p,1,j}$	$\mathbf{z}$, $m = 0$	$\mathbf{z}$, $m = 1$	$\mathbf{z}$, $m = 2$
$\mathbf{x}_{1,0} = 01\|00\|00\|00$	$\mathbf{z} = **\|**\|00\|00$	$\mathbf{z} = **\|**\|*0\|*0$	$\mathbf{z} = **\|**\|**\|**$
$\mathbf{x}_{1,1,1} = 01\|01\|00\|00$	$\mathbf{z} = **\|**\|00\|00$	$\mathbf{z} = **\|**\|*0\|*0$	$\mathbf{z} = **\|**\|**\|**$
$\mathbf{x}_{1,1,2} = 01\|11\|00\|00$	$\mathbf{z} = **\|**\|00\|00$	$\mathbf{z} = **\|**\|*0\|*0$	$\mathbf{z} = **\|**\|**\|**$
$\mathbf{x}_{2,0} = 10\|00\|00\|00$	$\mathbf{z} = **\|00\|**\|00$	$\mathbf{z} = **\|*0\|**\|*0$	$\mathbf{z} = **\|**\|**\|**$
$\mathbf{x}_{2,1,1} = 10\|00\|01\|00$	$\mathbf{z} = **\|00\|**\|00$	$\mathbf{z} = **\|*0\|**\|*0$	$\mathbf{z} = **\|**\|**\|**$
$\mathbf{x}_{2,1,2} = 10\|00\|11\|00$	$\mathbf{z} = **\|00\|**\|00$	$\mathbf{z} = **\|*0\|**\|*0$	$\mathbf{z} = **\|**\|**\|**$
$\mathbf{x}_{3,0} = 11\|00\|00\|00$	$\mathbf{z} = **\|00\|00\|**$	$\mathbf{z} = **\|*0\|*0\|**$	$\mathbf{z} = **\|**\|**\|**$
$\mathbf{x}_{3,1,1} = 11\|00\|00\|01$	$\mathbf{z} = **\|00\|00\|**$	$\mathbf{z} = **\|*0\|*0\|**$	$\mathbf{z} = **\|**\|**\|**$
$\mathbf{x}_{3,1,2} = 11\|00\|00\|11$	$\mathbf{z} = **\|00\|00\|**$	$\mathbf{z} = **\|*0\|*0\|**$	$\mathbf{z} = **\|**\|**\|**$

3.5. A Three-Dimensional Wave Equation in a Numerical Example

As a practical example of our approach, we conducted a numerical simulation focusing on the three-dimensional wave Equation ($D = 3$). A finite-difference scheme is employed to discretize the problem on a regular grid, with each spatial dimension depicted by n qubits, leading to $N = 2^n$ grid points for each dimension. The tridiagonal ($d = 1$) matrices provide an approximation for the finite-difference operators, and we utilize the block speed model described in Equation (32). The task at hand involves computing the implementation of the propagator $\exp\left(-it\tilde{H}_3\right)$, where the Hamiltonian $\tilde{H}_3$ is defined as

$$\tilde{H}_3 = \begin{pmatrix} 0 & \tilde{B}_x S & \tilde{B}_y S & \tilde{B}_z S \\ S\tilde{B}_x^T & 0 & 0 & 0 \\ S\tilde{B}_y^T & 0 & 0 & 0 \\ S\tilde{B}_z^T & 0 & 0 & 0 \end{pmatrix} = \tilde{S} H_3 \tilde{S}, \quad \tilde{S} = \begin{pmatrix} I & 0 & 0 & 0 \\ 0 & S & 0 & 0 \\ 0 & 0 & S & 0 \\ 0 & 0 & 0 & S \end{pmatrix}, \tag{37}$$

with $\tilde{B}_x = B \otimes I \otimes I$, $\tilde{B}_y = I \otimes B \otimes I$, $\tilde{B}_z = I \otimes I \otimes B$. Here, $B, I \in \mathbb{R}^{N \times N}$, and B is a tridiagonal matrix incorporating zero boundary conditions [15].

Direct classical simulation of $\exp\left(-it\tilde{H}_3\right)$ is computationally hard. For instance, a discretization with $n = 6$ qubits ($N = 64$ points) per dimension requires exponentiating a matrix of size $4N^3 = 2^{3n+2} = 2^{20} \approx 10^6$. For realistic scenarios requiring high resolution (e.g., $n \geq 10$, or $N \geq 1000$ points per dimension), direct classical computation of the propagator becomes intractable.

For this physically meaningful resolutions—e.g., $n \geq 10$ so $N = 2^{10} \approx 10^3$ grid points per axis—the state dimension is $4N^3 \sim 4 \times 10^9$, and even a single sparse matvec becomes impractical on commodity hardware. These constraints motivate our validation approach: rather than attempting end-to-end classical time evolution at large sizes, we (i) normalize the geometry and evolution time ($l_x = l_y = l_z = 1$, $t = 1$) to isolate algorithmic—not unit—scaling; (ii) adopt a randomized block speed field on a $2^m \times 2^m \times 2^m$ partition so that $\left\|\tilde{H}_3\right\|$ is representative (avoiding degenerate cases that would understate Trotter costs); and (iii) evaluate quantities that directly determine quantum resource scaling—namely, the number of Pauli terms and the gate count per first-order Trotter step—without

reconstructing the full propagator. Where feasible (e.g., $n \leq 5$), we still perform a functional check against a classical reference using `expm_multiply`, but the principal evidence comes from structural counts that (by Propositions 3 and Corollary 4) control the asymptotic behavior. This strategy preserves fidelity to the "hard part" of the problem—captured by the term structure and $\left\|\tilde{H}_3\right\|$—while remaining computationally tractable across the range of (n, m) needed to reveal the predicted scaling trends.

Therefore, to efficiently yet rigorously examine the scaling characteristics of our quantum algorithm, we employ a representative parameterization that encapsulates the essential complexity:

- Time of evolution $t = 1$.
- Domain lengths $l_x = l_y = l_z = 1$.
- A block speed model with random elements from the range $(0, 1]$, encoded with an equal number of qubits m per direction.
- An equal number of qubits per dimension n.

For the Hamiltonian $\tilde{H}_3$, the maximum possible number of gates (Proposition 3) is represented by $g_H \leq \Gamma K'$, where Γ is the number of sets containing mutually commuting Pauli strings (Equation (19)), and K' is the number of Pauli strings per set. In the block speed model (Section 3.4), we find that $K' \leq (D+1)2^{n+2m-1} = 2^{n+2m+1} = 2N4^m$. Thus, the limit turns into

$$g_H \leq 3(n+1)2^{n+2m+1} = 6(n+1)N4^m. \tag{38}$$

Figure 1 illustrates the findings of the numerical simulation. We expressed the Hamiltonian $\tilde{H}_3$ as a sum of Pauli strings and identified how many of these terms had non-zero coefficients. This decomposition was achieved by forming terms as outlined in Proposition 1 and evaluating their respective coefficients. For systems with $n < 6$ qubits per dimension, we fully reconstructed the Hamiltonian from its Pauli decomposition and verified its equivalence to the original matrix to validate our method. For $n \geq 6$, direct reconstruction becomes computationally prohibitive; therefore, we relied solely on the coefficient computation routine without full matrix reconstruction. The plot displays the counts of the resulting Pauli terms as crosses, and the theoretical upper limit from Equation (38) is represented by a dotted line.

The empirical results reveal that the calculated upper bound closely estimates the actual count of Pauli strings. This difference arises from the distinctive configuration of the finite-difference matrix B used in our simulation. While the theoretical limit is formulated for any tridiagonal matrix, our approach uses a typical first-order stencil with matrix B having -1 on the main diagonal and 1 on the upper diagonal. This specific arrangement results in a more efficient Pauli decomposition, needing about 2^{n+1} strings for each operator, as opposed to the general maximum of $(n+1)2^{n-1}$.

The number of gates required for a single iteration of a p-th order Trotterization is based on the first-order equation as presented in Equation (26). Consequently, our analysis prioritizes the numerical validation of the first-order estimate. For the Hamiltonian $\tilde{H}_3$, Equation (24) provides the upper limit on the gate count for one step of a first-order Trotterization, expressed as $g_1 \leq \Gamma(g_d + g_z)$. Here, Γ denotes the number of sets containing commuting Pauli strings, g_d represents the gate count required to diagonalize an individual set, and g_z signifies the gate count to execute the diagonal rotations in a diagonalized set.

For the block speed model, it is possible to utilize Gray code ordering within each mutually commuting set. This approach allows for further elimination of CNOT gates, thereby reducing the limit to $g_z \leq 2K'$. Previously, in Corollary 4 it was established that the block speed model restricts the number of Pauli strings per set by $K' \leq 2N4^m$. Thus, the limit on the number of one- and two-qubit gates for a first-order step is

$$g_1 \leq 3(n+1)[2(n+2)+4N4^m] = 6(n+1)[n+2+2N4^m]. \tag{39}$$

Figure 2 illustrates the results of the numerical simulation, showing how the gate counts vary with the number of qubits per dimension. We created a software program designed to produce a circuit corresponding to one Trotter step and for each collection of commuting Pauli strings. Each group's circuit is output in QASM format for later integration into several Trotter steps. Following this, the overall gate count was ascertained from these output files. In systems where the qubit count is small, specifically $n \leq 4$, we confirmed that the circuit constructed for a single group of mutually commuting operators accurately realizes $\exp(-itH_\gamma)$. Here, H_γ represents a Hamiltonian made up exclusively of Pauli strings from that group.

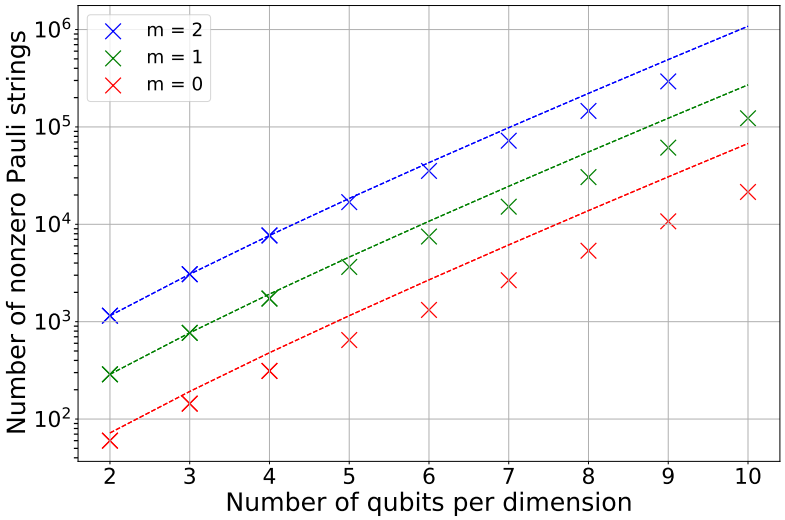

Figure 1. Number of non-zero Pauli terms in the decomposition of the Hamiltonian $\tilde{H}_3$ given by Equation (37). Empirical data (crosses), obtained with randomized velocity profiles for various block sizes m, are compared against the theoretical upper bound from Equation (38) (dotted line). The results demonstrate the scaling of the term count with the number of discretization qubits n for different values of m.

In Figure 2, the experimental results, indicated by crosses, are slightly beneath the theoretical limit outlined in Equation (39), represented by the dotted line. This is attributed to circuit optimization techniques applied during compilation. The CNOT gates in the diagonalization circuits of each set can cancel each other out if placed in a specific order, providing a slight optimization benefit. However, the main impact comes from performing the diagonal component.

Additionally, to validate the functional correctness of the algorithm, we examined a specific instance with fixed parameters $n = 5$ and $m = 2$. This corresponds to solving the wave equation on a $32 \times 32 \times 32$ grid ($N = 2^5 = 32$), partitioned into $2^m = 4$ velocity blocks per dimension, resulting in a total of 64 individual blocks, each of size $8 \times 8 \times 8$ and assigned a unique velocity value. For this configuration, we verified that the implemented quantum circuit accurately approximates the action of the target unitary $\exp\left(-i\tilde{H}_3 t\right)$.

We assessed the accuracy by setting the initial condition to a random wavefunction $|\psi(0)\rangle$ and comparing the statevector produced by our algorithm against a classical reference computed using the "scipy.sparse.linalg.expm_multiply" function in Python. Specifically, we compute the error

$$\epsilon = \left\| U(t,r,p)|\psi(0)\rangle - \exp\left(-it\tilde{H}_3\right)|\psi(0)\rangle \right\|_2, \tag{40}$$

where $U(t, r, p)|\psi(0)\rangle$ is the state obtained from our quantum algorithm using r Trotter steps of order p.

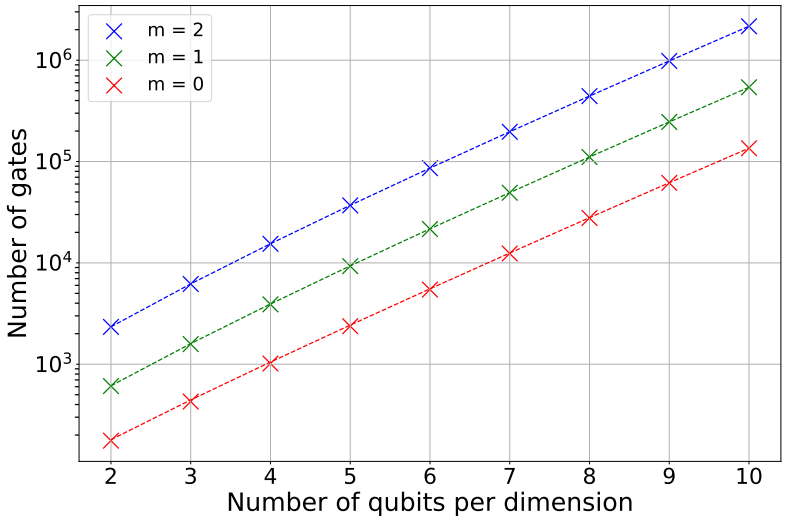

Figure 2. Gate count for a single first-order Trotter step of the Hamiltonian $\tilde{H}_3$. Empirical data (crosses), obtained from compiled QASM circuits, are compared against the theoretical upper bound from Equation (39) (dotted line). The results demonstrate the scaling of the gate count with the number of discretization qubits n for different velocity block sizes m.

Figure 3 illustrates how the error varies with both the quantity of Trotter steps and the Trotter order. As anticipated, the error diminishes when the number of steps or the formula's order is raised. This outcome confirms our implementation and offers a pragmatic calculation of the Trotter steps needed to reach a specified error threshold.

Conducting a more thorough validation requires calculating the operator norm error $\left\|U(t, r, p) - \exp\left(-it\tilde{H}_3\right)\right\|_2$ and assessing it against the bound in Corollary 4. Nevertheless, this calculation turns out to be computationally infeasible using classical methods for the matrix dimensions we are dealing with, even when n and m are relatively small.

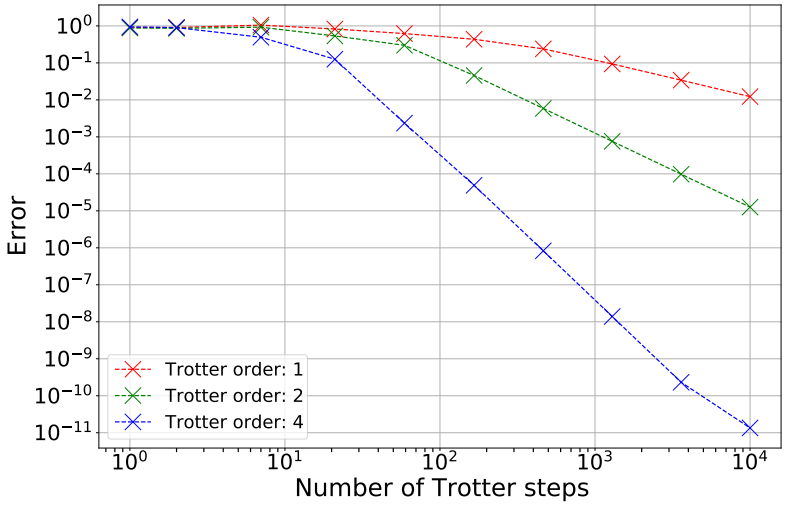

Figure 3. Trotter error scaling for the Hamiltonian $\tilde{H}_3$ with fixed parameters $n = 5$, $m = 2$. The error ϵ given by (40) is plotted against the number of Trotter steps r for different Trotter orders p. The results validate the algorithm and demonstrate the expected convergence with increasing r and p.

4. Discussion

We introduced a structured quantum algorithm for simulating the multidimensional variable-coefficient wave equation. The method begins with a Hamiltonian formulation and a decomposition into Pauli strings—tensor products of I, X, Y, Z. This representation affords a clear physical interpretation: Z terms encode diagonal energy shifts, whereas X and Y terms generate off-diagonal transitions between basis states.

Our main contribution is a systematic extension of bounds on only relevant Pauli strings and their generation for Hamiltonians with a special block structure, together with a partition of these strings into commuting sets that admit efficient diagonalization (Propositions 1 and 2). This customizes and generalizes prior decomposition strategies [15,16] to the setting of variable-coefficient, multi-dimensional wave equations.

A second contribution is the establishment of precise gate complexity limits for calculating time evolution operators expressed as $\exp\left(-i\tilde{H}_D t\right)$. Our analysis indicates that, given assumptions applicable to physical models, the gate complexity for a single Trotter step increases linearly with the size of one dimension N (Corollary 2). Nevertheless, employing Trotter–Suzuki formulas adds extra overhead: the gate complexity for the complete algorithm utilizing a p-th order Trotter formula scales as $N^{2+1/p}$ with respect to the size of a single dimension (Corollary 3). This result is significant because it tackles the challenge posed by the "curse of dimensionality", which hinders the functionality of conventional wave equation solvers that typically scale as N^D [1,2].

In contrast to oracle-based strategies [9] (with a 1D implementation in [12]), our algorithm specifies deterministic circuits in terms of one- and two-qubit gates. Although oracle-based methods may exhibit superior asymptotic query complexity, their constant factors and ancilla requirements can be substantial in regimes relevant to near-term problems [12]. Consequently, despite worse asymptotics, our explicit construction is often more practical at modest scales and enables precise resource estimation. Related oracle-dependent works [10,14] demonstrate scalability but at the cost of extra ancillas. For constant-speed models, ref [13] employs the Bell-basis approach to efficiently diagonalize each term of the Hamiltonian, yielding the same diagonalization as our method, while [11] demonstrates hardware experiments using Fourier transforms; however, these techniques do not directly extend to variable-speed profiles. In summary, relative to these lines of work, our algorithm provides explicit higher-dimensional resource bounds while accommodating variable coefficients.

For block-structured speed profiles, our analysis (Corollary 4) shows that respecting physical structure can materially reduce cost: the complexity simplifies to $2^{(D-1)m}$ when m qubits encode each block parameter (for a total of 2^{Dm} distinct speeds). Such piecewise-homogeneous models are common in seismic and acoustic imaging [5], making this specialization particularly relevant.

Three-dimensional simulations corroborate the predicted trends and indicate that rotation scheduling within each diagonal group can further reduce cost [21]. These optimizations are compatible with our commuting-set framework and offer practical gains without altering asymptotics.

Our use of Trotter–Suzuki decomposition entails the usual trade-off between accuracy and depth. Techniques such as qubitization [23] and linear combinations of unitaries [24] may improve asymptotic performance but generally require additional ancillas and more elaborate control logic. Moreover, our present estimates omit measurement overhead, cost-function construction, noise, and limited connectivity, all of which are consequential in the NISQ regime [25,26]. Closing these gaps is a necessary step toward end-to-end application benchmarks.

Beyond PDEs, the proposed Pauli decomposition and commuting-set diagonalization extend naturally to Hamiltonians with short-range interactions, where matrix elements cluster near the diagonal. Reducing non-commuting terms directly lowers measurement cost in variational and hybrid schemes [27,28]. The framework is also complementary to advances in randomized simulation [29], qubitization [23], and Schrödingerization [30]. We anticipate fruitful combinations that retain the structural advantages of our decomposition while improving asymptotic scaling.

Overall, the results advance quantum simulation of the wave equation in higher dimensions with variable coefficients, delivering explicit gate counts, deterministic circuit structure, and a pathway to exploiting block heterogeneity. We expect the practical advantage over oracle-based techniques observed in lower-dimensional, structured instances [15] to persist in 3D, though a like-for-like quantitative comparison remains open due to the absence of clear oracle constructions in this setting. Future work will integrate advanced simulation primitives and incorporate hardware-aware compilation and error mitigation to translate the present complexity gains into experimentally validated performance.

5. Conclusions

We have developed a general framework for quantum simulation of the multidimensional wave equation using Pauli string decomposition, diagonalization of commuting operator sets, and Trotterized time evolution. The method provides rigorous gate-complexity bounds and shows favorable scaling, particularly for the block speed model. Numerical validation in three dimensions confirms both the theoretical predictions and the potential for additional reductions through circuit-level optimization.

These results demonstrate that quantum computing can offer practical advantages for simulating high-dimensional wave dynamics, a class of problems that remain computationally demanding for classical solvers. Future directions include integrating qubitization and other advanced simulation techniques as well as designing hardware-aware circuit optimizations. The presented approach establishes a foundation for scalable quantum algorithms with applications in geophysics, nondestructive testing, and materials science.

Author Contributions: Conceptualization, B.A. and I.Z.; software, B.A.; validation, B.A. and I.Z.; formal analysis, B.A.; writing, review and editing, B.A. and I.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original data presented in the study are openly available in GitHub repository at https://github.com/barseniev/d_dim_quantum_wave_solver (accessed on 9 October 2025).

Acknowledgments: The authors acknowledge the use of Skoltech's Zhores supercomputer [31] for obtaining the numerical results presented in this paper.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

PDE	Partial differential equation
FDM	Finite difference method

Appendix A. Discretization and Vectorization Conventions

This appendix details the spatial discretization scheme and vectorization procedure employed to transform the continuous three dimensional wave equation, Equation (1)

($D = 3$), into a discrete system suitable for numerical computation and quantum algorithm construction. While the focus is on three dimensions, the methodology generalizes straightforwardly to other cases.

The continuous scalar field $u(t, x, y, z)$ is discretized onto a structured grid with N_x, N_y, and N_z points in the x, y, and z directions, respectively. We adopt the convention where the discrete solution is stored as a three-dimensional tensor $U(t) \in \mathbb{R}^{N_z \times N_y \times N_x}$, such that its elements are defined by

$$u_{i,j,k}(t) \equiv u(t, x_k, y_j, z_i),$$

where the tensor indices (i, j, k) correspond to the spatial coordinates (z, y, x). This specific index ordering is chosen to yield a canonical and efficient structure for the resulting discrete differential operators. A schematic representation of this data structure is provided in Figure A1.

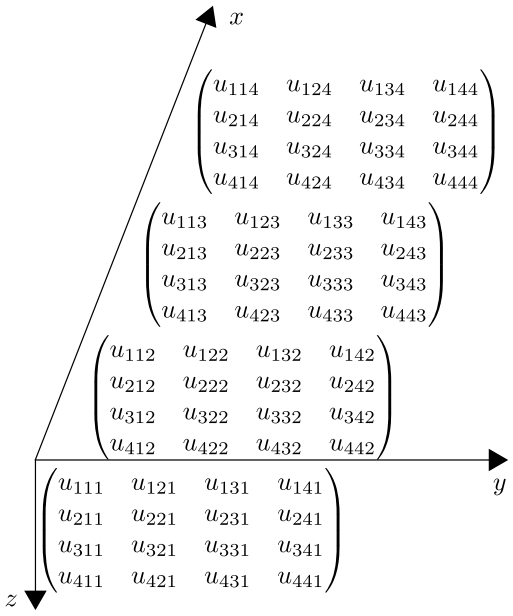

Figure A1. Schematic example of a three dimensional data tensor $U \in \mathbb{R}^{Z,Y,X}$, i.e., storing points corresponding to different coordinates as an array. Thus, an element of the array $u_{i,j,k}$ corresponds to a point with coordinates $u(x_k, y_j, z_i)$. For discretization, 4 points per direction are used.

To interface with standard numerical linear algebra routines, which typically operate on vectors, the tensor U is transformed into a state vector via a vectorization operation, denoted $\mathrm{vec}(U)$. We adopt the Fortran-style (column-major) vectorization convention. This process maps the multi-dimensional array into a column vector $\mathrm{vec}(U(t)) \in \mathbb{R}^{N_x N_y N_z}$. The sequence of this mapping is illustrated schematically in Figure A2.

This chosen discretization and vectorization ordering induces a highly structured form for the discrete Laplacian operator. The right-hand side of Equation (1) is approximated as

$$\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) \approx L_3 \,\mathrm{vec}(U), \tag{A1}$$

where the discrete 3D Laplacian L_{3d} is given by the sum of Kronecker products:

$$L_3 = L_x \otimes I_y \otimes I_z + I_x \otimes L_y \otimes I_z + I_x \otimes I_y \otimes L_z. \tag{A2}$$

Here, L_x, L_y, L_z are matrix representations of one-dimensional second-derivative operators (e.g., from a finite-difference method) incorporating the desired boundary conditions, and I_x, I_y, I_z are identity matrices of appropriate sizes. Crucially, the order of the Kronecker

product factors (x, y, z) is a direct consequence of the chosen vectorization convention, ensuring consistent and efficient application of the operator.

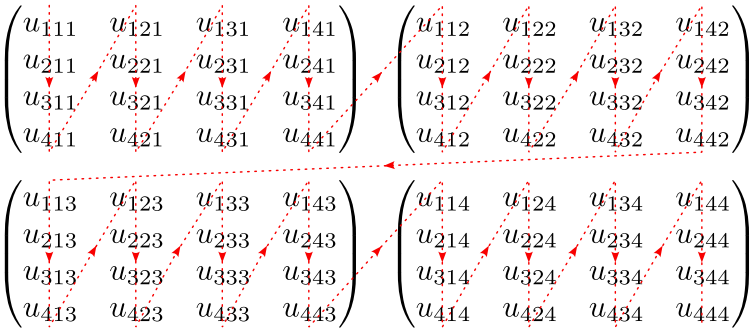

Figure A2. Illustration of the vectorization process for the tensor U from Figure A1. The red dashed lines indicate the order in which elements are sequenced into the resulting column vector (from top to bottom).

Consequently, the discrete form of the wave equation initial value problem, incorporating a spatially varying wave speed $c(\vec{x})$, is expressed as the following system of ordinary differential equations:

$$\begin{aligned} &\frac{\partial^2}{\partial t^2}\mathrm{vec}(U(t)) = L_3(c(\vec{x}))\,\mathrm{vec}(U(t)), \\ &\mathrm{vec}(U(0)) = \mathrm{vec}(F), \\ &\frac{\partial}{\partial t}\mathrm{vec}(U(0)) = \mathrm{vec}(G), \end{aligned} \tag{A3}$$

where $\mathrm{vec}(U(t))$ is the state vector, and F and G are the tensor representations of the initial conditions $f(\vec{x})$ and $g(\vec{x})$, respectively. The dependence of the operator L_3 on the wave speed profile $c(\vec{x})$ is implied.

Appendix B. Decomposition Example

This section provides a detailed, step-by-step example of the proposed decomposition method. We consider the three-dimensional ($D = 3$) wave equation with a block velocity profile:

$$\begin{aligned} &\frac{\partial^2 u(t,\vec{x})}{\partial t^2} = \frac{\partial}{\partial x}\left(c^2(\vec{x})\frac{\partial u(t,\vec{x})}{\partial x}\right) + \frac{\partial}{\partial y}\left(c^2(\vec{x})\frac{\partial u(t,\vec{x})}{\partial y}\right) + \frac{\partial}{\partial z}\left(c^2(\vec{x})\frac{\partial u(t,\vec{x})}{\partial z}\right), \\ &u(t=0,\vec{x}) = f(\vec{x}), \\ &\frac{\partial u(t=0,\vec{x})}{\partial t} = g(\vec{x}), \\ &u(t,x=0) = u(t,y=0) = u(t,z=0) = 0, \\ &u(t,x=l_x) = u(t,y=l_y) = u(t,z=l_z) = 0. \end{aligned} \tag{A4}$$

The velocity profile is defined as

$$c(x,y,z) = \begin{cases} c_1 & \text{if } z \le l_z/2, \\ c_2 & \text{if } z > l_z/2. \end{cases} \tag{A5}$$

This equation can be transformed into the Schrödinger form, $i\partial_t\psi = \tilde{H}3\psi$, with the Hamiltonian:

$$\tilde{H}_3 = \begin{pmatrix} 0 & \tilde{B}_x S & \tilde{B}_y S & \tilde{B}_z S \\ S\tilde{B}_x^T & 0 & 0 & 0 \\ S\tilde{B}_y^T & 0 & 0 & 0 \\ S\tilde{B}_z^T & 0 & 0 & 0 \end{pmatrix} = \tilde{S} H_3 \tilde{S}, \quad \tilde{S} = \begin{pmatrix} I & 0 & 0 & 0 \\ 0 & S & 0 & 0 \\ 0 & 0 & S & 0 \\ 0 & 0 & 0 & S \end{pmatrix}, \tag{A6}$$

where $\tilde{B}_x = B \otimes I \otimes I$, $\tilde{B}_y = I \otimes B \otimes I$, and $\tilde{B}_z = I \otimes I \otimes B$. The matrices $B, I \in \mathbb{R}^{N\times N}$, with $N = 2^n$, arise from discretizing each spatial dimension, and B is a tridiagonal matrix incorporating the zero boundary conditions.

The next step is to decompose this Hamiltonian for quantum simulation. We set $n = 2$ qubits per dimension. Following Proposition 1, the Pauli strings $\mathbf{x}$ that may appear in the decomposition are listed in the first column of Table A1. The second step involves analyzing the block velocity profile. In this case, the profile is split only along the z-direction, implying $m_x = 0$, $m_y = 0$, and $m_z = 1$. Consequently, the matrix S can be written as

$$S = \mathrm{diag}(\mathrm{vec}(C))) = \frac{c_1 + c_2}{2} I \otimes I \otimes I \otimes I \otimes I \otimes I + \frac{c_1 - c_2}{2} I \otimes I \otimes I \otimes I \otimes Z \otimes I. \tag{A7}$$

From this, we derive the masks for the $\mathbf{z}$ strings, shown in the second column of Table A1.

Table A1. Pauli strings for the three-dimensional wave equation, discretized with $n = 2$ qubits per dimension and finite-difference operators approximated by tridiagonal matrices ($d = 1$). The strings $\mathbf{x}_{p,k,j}$ (with $k = 1$ for $d = 1$) are generated per Proposition 1. The corresponding masks for the $\mathbf{z}$ strings are shown for a two-block velocity model; a "$*$" denotes a position that can be 0 or 1. Bits are grouped for readability: the first $\left|\hat{\mathbb{B}}(D)\right| = 2$ bits are for the Hamiltonian, and each subsequent block of $n = 2$ bits encodes a spatial direction. The third column lists the diagonalization operators for case where $\mathbf{x} \cdot \mathbf{z} = 0 \pmod 2$, according to Proposition 2. Here, H_k denotes a Hadamard gate on qubit k, and $CNOT(c, t)$ is a controlled-NOT gate with control c and target t.

$\mathbf{x}_{p,1,j}$	$\mathbf{z}$	**Diagonalization Operator** D
$\mathbf{x}_{1,0} = 01\|00\|00\|00$	$\mathbf{z} = **\|**\|00\|*0$	H_2
$\mathbf{x}_{1,1,1} = 01\|01\|00\|00$	$\mathbf{z} = **\|**\|00\|*0$	$H_2 CNOT(2,4)$
$\mathbf{x}_{1,1,2} = 01\|11\|00\|00$	$\mathbf{z} = **\|**\|00\|*0$	$H_2 CNOT(2,3) CNOT(2,4)$
$\mathbf{x}_{2,0} = 10\|00\|00\|00$	$\mathbf{z} = **\|00\|**\|*0$	H_1
$\mathbf{x}_{2,1,1} = 10\|00\|01\|00$	$\mathbf{z} = **\|00\|**\|*0$	$H_1 CNOT(1,6)$
$\mathbf{x}_{2,1,2} = 10\|00\|11\|00$	$\mathbf{z} = **\|00\|**\|*0$	$H_1 CNOT(1,5) CNOT(1,6)$
$\mathbf{x}_{3,0} = 11\|00\|00\|00$	$\mathbf{z} = **\|00\|00\|**$	$H_1 CNOT(1,2)$
$\mathbf{x}_{3,1,1} = 11\|00\|00\|01$	$\mathbf{z} = **\|00\|00\|**$	$H_1 CNOT(1,2) CNOT(1,8)$
$\mathbf{x}_{3,1,2} = 11\|00\|00\|11$	$\mathbf{z} = **\|00\|00\|**$	$H_1 CNOT(1,2) CNOT(1,7) CNOT(1,8)$

To illustrate, consider the row for $\mathbf{x}_{3,1,1} = 11|00|00|01$ with the mask $\mathbf{z} = **|00|00|**$. This mask indicates that only the Pauli strings listed in the second column of Table A2 may appear in the decomposition. Since these Pauli strings all commute, we can apply Proposition 2 to construct a diagonalization circuit. The required operator D is given in the last column of Table A1, and the result of applying this diagonalization is shown in the final column of Table A2.

Table A2. Pauli strings generated for $\mathbf{x}_{3,1,1} = 11|00|00|01$ and its corresponding mask $\mathbf{z} = **|00|00|**$, where "$*$" denotes a position that can be 0 or 1. Only strings with an even number of Y operators ($\mathbf{x} \cdot \mathbf{z} = 0 \pmod 2$) are retained as the Hamiltonian $\tilde{H}_3$ is real. The first column lists the specific $\mathbf{z}$ strings generated from the mask. The second column shows the corresponding Pauli strings in the decomposition. The third column lists the $\tilde{\mathbf{z}}$ strings after diagonalization per Proposition 2, and the last column shows the resulting diagonalized Pauli strings with diagonalization operator $D = H_1 CNOT(1,2) CNOT(1,8)$, and its conjugate transpose $D^\dagger = CNOT(1,8) CNOT(1,2) H_1$, since both the CNOT and Hadamard gates are Hermitian. Bits and Pauli matrices are grouped for readability.

$\mathbf{z} = **\|00\|00\|**$	$W(\mathbf{x},\mathbf{z})$	$\tilde{\mathbf{z}} = 1*\|00\|00\|**$	$W(0,\tilde{\mathbf{z}}) = (-1)^{(\mathbf{x}\cdot\mathbf{z})/2} D W(\mathbf{x},\mathbf{z}) D^\dagger$
00|00|00|00	$XX\|II\|II\|IX$	10|00|00|00	$+ZI\|II\|II\|II$
00|00|00|10	$XX\|II\|II\|ZX$	10|00|00|10	$+ZI\|II\|II\|ZI$
01|00|00|01	$XY\|II\|II\|IY$	11|00|00|01	$-ZZ\|II\|II\|IZ$
01|00|00|11	$XY\|II\|II\|ZY$	11|00|00|11	$-ZZ\|II\|II\|ZZ$
10|00|00|01	$YX\|II\|II\|IY$	10|00|00|01	$-ZI\|II\|II\|IZ$
10|00|00|11	$YX\|II\|II\|ZY$	10|00|00|11	$-ZI\|II\|II\|ZZ$
11|00|00|00	$YY\|II\|II\|IX$	11|00|00|00	$-ZZ\|II\|II\|II$
11|00|00|10	$YY\|II\|II\|ZX$	11|00|00|10	$-ZZ\|II\|II\|ZI$

We express the total Hamiltonian as a sum $\tilde{H}_D = \sum_{\gamma=1}^{\Gamma} H_\gamma$, where each H_γ is a mutually commuting group of Pauli strings characterized by a single $\mathbf{x}$. Since these groups do not all commute with each other, we employ a Trotter–Suzuki decomposition to approximate the time evolution.

A single Trotter step can be implemented using methods from [21]. The quantum circuit for the propagator $\exp(-iH_\gamma t)$, corresponding to the group characterized by $\mathbf{x}_{3,1,1}$, is shown in Figure A3. The circuit emphasizes the gate arrangement; the specific rotation angles for the R_z gates (which are defined in our implementation) are omitted for visual clarity. Using a Gray code ordering minimizes the number of CNOT gates in the diagonalization circuit.

The circuits for each group $\mathbf{x}$ are combined to form a single Trotter step. These steps are then repeated according to the chosen Trotter formula order to achieve the desired simulation accuracy.

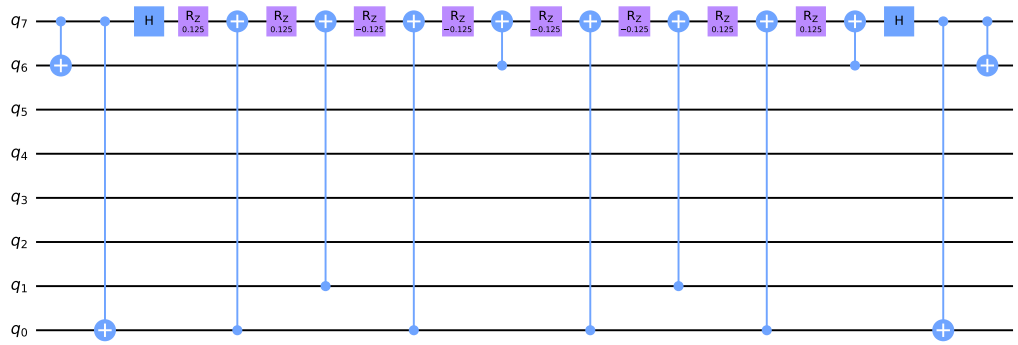

Figure A3. Quantum circuit for the propagator $\exp(-iH_\gamma t)$, where H_γ is characterized by $\mathbf{x}_{3,1,1}$. The $\mathbf{z}$ strings are processed in the following order (left to right): 00000000, 10000001, 10000011, 00000010, 11000010, 01000011, 01000001, 11000000. That is, in the the Gray code order for $\tilde{\mathbf{z}}$. The rotation angles are $\theta = 2\alpha t_{tr}$, where α is the coefficient of the corresponding Pauli string and t_{tr} is the scaled time from the Trotter decomposition (e.g., for a first-order formula, $t_{tr} = t/r$, with r being the number of steps). The figure shows gate arrangement; the numerical values inside the R_z gates are not relevant.

Appendix C. Application to a 3D Standing Wave Problem

This section demonstrates the application of our proposed quantum algorithm to a three-dimensional standing wave problem. We consider a constant wave velocity $c = 1$ and compare the computational complexity of our method against a standard finite-difference method (FDM) for the accuracy level. The problem is defined by the following wave equation and initial/boundary conditions:

$$\begin{aligned}
&\frac{\partial^2 u(t,\vec{x})}{\partial t^2} = \frac{\partial^2 u(t,\vec{x})}{\partial x^2} + \frac{\partial^2 u(t,\vec{x})}{\partial y^2} + \frac{\partial^2 u(t,\vec{x})}{\partial z^2},\\
&u(0,\vec{x}) = \sin(\pi x)\sin(\pi y)\sin(\pi z),\\
&\frac{\partial u(0,\vec{x})}{\partial t} = 0,\\
&u(t,0,y,z) = u(t,x,0,z) = u(t,x,y,0) = 0,\\
&u(t,1,y,z) = u(t,x,1,z) = u(t,x,y,1) = 0.
\end{aligned} \tag{A8}$$

An exact analytical solution is known for this problem:

$$u_{\text{exact}}(x,y,z,t) = \sin(\pi x)\sin(\pi y)\sin(\pi z)\cos\left(\sqrt{3}\pi t\right). \tag{A9}$$

We compare the solution from our quantum method against a classical second-order FDM using an identical spatial discretization. The classical FDM solution is constructed iteratively for $\mathbf{u}$—the discretized and vectorized version of $u(t,\vec{x})$:

$$\begin{aligned}
&\mathbf{u}(0) = \mathbf{u}(\Delta t) = \mathbf{f},\\
&\mathbf{u}(k\Delta t) = \Delta t^2 L_3 \mathbf{u}((k-1)\Delta t) + 2\mathbf{u}((k-1)\Delta t) - \mathbf{u}((k-2)\Delta t), \quad k \geq 2.
\end{aligned} \tag{A10}$$

Here, $\mathbf{f}$ is the discretized and vectorized initial condition $u(0,\vec{x})$. The discrete Laplacian L_3 is defined as

$$L_3 = L_x \otimes I_y \otimes I_z + I_x \otimes L_y \otimes I_z + I_x \otimes I_y \otimes L_z, \tag{A11}$$

where $L_x = -B_x^T B_x$, $L_y = -B_y^T B_y$, and $L_z = -B_z^T B_z$. The matrix B_x (and similarly B_y, B_z) is a first-order finite-difference operator (same as in Hamiltonian) with zero boundary conditions. For example, for $n = 2$ we have $B_x = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$. For the classical FDM simulation, the number of time steps N_t is chosen to satisfy the Courant–Friedrichs–Lewy (CFL) condition for numerical stability, specifically $N_t = \lceil \sqrt{3} N_x \rceil$, where $N_x = N_y = N_z$.

To quantify accuracy, we use the discrete L^2-norm error:

$$\epsilon_{num}(\mathbf{u},\mathbf{v}) = \sqrt{\Delta V \sum_j (u_j - v_j)^2}, \quad \Delta V = \Delta x \Delta y \Delta z. \tag{A12}$$

After computing the error ϵ_{num} between the FDM and analytical solutions, we estimate the number of Trotter steps r required for the quantum algorithm to achieve the same accuracy. This estimate is derived from the data plotted in Figure A4, where the crosses represent the quantum algorithm's error (error between the quantum and analytical solutions) as a function of r. For each spatial discretization level (i.e., for each number of qubits per dimension), we locate the fixed FDM error level on the y-axis. We then determine the point where a horizontal line at this FDM error value intersects the interpolated trend (dashed line) of the quantum error data on the log–log plot. The x-coordinate of this intersection

gives the estimated number of Trotter steps r needed for the quantum solution to match the classical FDM's precision. These intersection points are marked with dots in the figure.

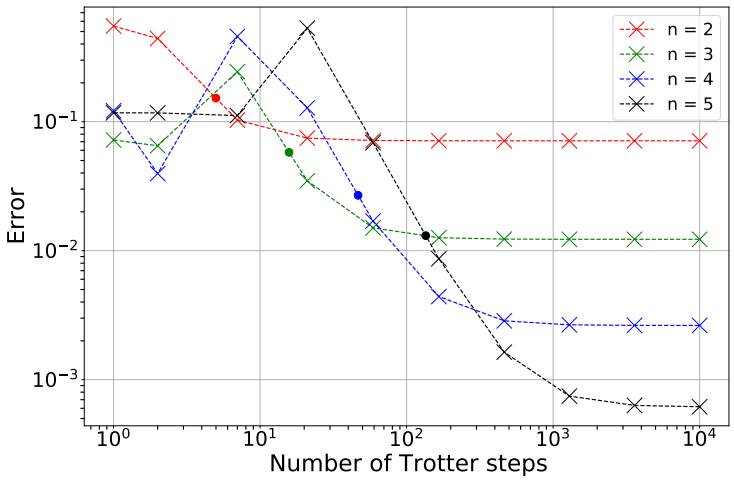

Figure A4. Estimation of the quantum Trotter steps required for the second order formula to match classical FDM accuracy. The crosses show ϵ_{num}— error between quantum algorithm and analytical solution — plotted against the number of Trotter steps r for varying qubits per dimension n, corresponding to discretization points $N = 2^n$. For a fixed FDM error, the required Trotter steps are determined by the intersection with the error trend (dashed line). These intersections, marked by dots, provide the estimated r required for the quantum algorithm to achieve the same accuracy as the FDM.

Finally, we compare the computational cost, measured in the total number of operations. For the classical FDM, the cost is estimated as $g_{cl} = 3N_t N^3 = 3\sqrt{3}N^4$ as the matrix L_3 has 3 non-zero entries per row, and the solution is evolved over N_t time steps. For the quantum algorithm, the cost is the total number of one- and two-qubit gates, estimated as $g_q = r \cdot g_1$, where r is the number of Trotter steps and g_1 is the gate count for a single step (can be evaluate as in Equation (39)). This comparison is shown in Figure A5, where it can be seen that the quantum algorithm requires fewer operations than the standard FDM for this problem.

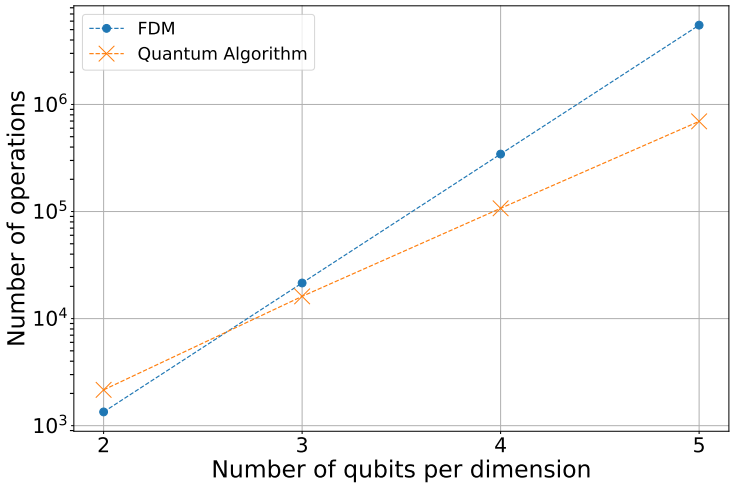

Figure A5. Comparison of the total number of operations (gate count for quantum, floating-point operations for classical) required by the quantum algorithm with the second order formula versus the classical FDM.

Appendix D. Proofs

Proof of Proposition 1. Without loss of generality, we assume $D+1 = 2^{n_d}$ for some integer n_d. If this is not the case, one can extend the Hamiltonian by adding matrices $B_p = 0$ for $p = D+1, \ldots, D'$, where $D' = 2^{\lceil \log_2(D) \rceil}$, ensuring D' is a power of two. This padding does not alter the structure or Pauli decomposition of the original H_D.

To prove the proposition, we utilize the standard basis matrices $E^n_{i,j}$, defined as the $2^n \times 2^n$ matrices that are zero everywhere except for a 1 at position (i,j). These matrices possess a crucial tensor product structure:

$$E^n_{i,j} = E^1_{i_0,j_0} \otimes \cdots \otimes E^1_{i_{n-1},j_{n-1}}, \tag{A13}$$

where i_k and j_k are the bits in the n-bit binary representations of i and j, respectively, i.e., $\hat{\mathbb{B}}_l(n,i)$ and $\hat{\mathbb{B}}_l(n,j)$. Using this basis, the Hamiltonian H_D can be expressed as

$$H_D = \sum_{p=1}^{D} \left(E^{n_d}_{0,p} \otimes B_p + E^{n_d}_{p,0} \otimes B_p^\dagger \right). \tag{A14}$$

Since the Pauli strings constituting B_p and $B_p^\dagger$ are identical (the difference is only in coefficients of these matrices), we can denote their combined Pauli support simply as P_p. Moreover, according to Proposition 1 in [16], the matrix B_p admits a decomposition into Pauli strings characterized by binary vectors of the form

$$\mathbf{x}_{k,j} = \hat{\mathbb{B}}_l\left(n-s, 2^j - 1\right) * \hat{\mathbb{B}}_l(s, 2^s - k), \tag{A15}$$

for $k \in \{1, \ldots, d\}$, $j \in \{1, \ldots, n-s\}$, $s = \lceil \log_2(k) \rceil$, and $d \leq 2^{n-1}$, plus an additional diagonal string $\mathbf{x}_0 = \hat{\mathbb{B}}_l(n, 0)$.

The single-qubit basis matrices have well-known Pauli decompositions:

$$E^1_{0,0} = (I+Z)/2, \qquad E^1_{1,1} = (I-Z)/2,$$
$$E^1_{0,1} = (X+iY)/2, \qquad E^1_{1,0} = (X-iY)/2.$$

Crucially, $E^1_{0,0}$ and $E^1_{1,1}$ are diagonal and are represented by the binary string $\mathbf{x} = 0$ (indicating I or Z), while $E^1_{0,1}$ and $E^1_{1,0}$ are off-diagonal and are represented by $\mathbf{x} = 1$ (indicating X or Y).

Now, applying the tensor product property (A13), we analyze the term $E^{n_d}_{0,p}$:

$$E^{n_d}_{0,p} = E^1_{0,p_0} \otimes \cdots \otimes E^1_{0,p_{n_d-1}}, \tag{A16}$$

where $p_0 \ldots p_{n_d-1} \equiv \hat{\mathbb{B}}_l(n_d, p)$ is the n_d-bit binary representation of p. The Pauli string representation of this matrix is determined by the $\mathbf{x}$-component of each single-qubit factor. Since each factor E^1_{0,p_k} is represented by the binary value p_k, the full string for $E^{n_d}_{0,p}$ is simply $\mathbf{x} = \hat{\mathbb{B}}_l(n_d, p)$. An identical argument shows that $E^{n_d}_{p,0}$ is also represented by the same string, $\mathbf{x} = \hat{\mathbb{B}}_l(n_d, p)$.

Therefore, the overall Pauli string representation for a term $(E^{n_d}_{0,p} + E^{n_d}_{p,0}) \otimes P_p$ in H_D is given by the concatenation of the string for the first subsystem and the string for P_p. This yields the general form

$$\mathbf{x}_{p,k,j} = \hat{\mathbb{B}}_l(n_d, p) * \hat{\mathbb{B}}_l\left(n-s, 2^j - 1\right) * \hat{\mathbb{B}}_l(s, 2^s - k), \tag{A17}$$

for $p \in \{1, \ldots, D\}$, $k \in \{1, \ldots, d\}$, $j \in \{1, \ldots, n-s\}$, $s = \lceil \log_2(k) \rceil$. Additionally, the diagonal contributions from P_p lead to strings of the form

$$\mathbf{x}_{p,0} = \hat{\mathbb{B}}_l(n_d, p) * \hat{\mathbb{B}}_l(n, 0). \tag{A18}$$

This completes the characterization of all Pauli strings present in the decomposition of H_D. □

Proof of Proposition 2. We employ the standard tableau method for the simultaneous diagonalization of sets of commuting Pauli strings, as detailed in [19,20]. For a set of m commuting Pauli operators, each acting on n qubits, the tableau is a data structure of size $m \times 2n$ bits. Each row k of the tableau corresponds to a Pauli string P_k and is represented as a pair $(\mathbf{x}_k, \mathbf{z}_k)$, where the binary vectors $\mathbf{x}_k, \mathbf{z}_k \in \mathbb{B}^n$ encode the corresponding Pauli string. The tableau for our set, characterized by a single $\mathbf{x}$, is structured as follows:

$$\left[\begin{array}{ccc|ccc} x_{1,1} & \cdots & x_{1,n} & z_{1,1} & \cdots & z_{1,n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ x_{1,1} & \cdots & x_{1,n} & z_{m,1} & \cdots & z_{m,n} \end{array}\right].$$

Our goal is to construct a Clifford circuit that maps these operators to strings consisting solely of Z and I operators. This is equivalent to transforming the tableau such that the $\mathbf{x}$-block becomes zero.

The transformation rules for the tableau under elementary Clifford gates are well-established [19,20]:

- A Hadamard gate on qubit j swaps the j-th columns of the $\mathbf{x}$ and $\mathbf{z}$ blocks for all rows.
- A CNOT gate with control c and target t performs the following for all rows:

  $$x_{k,t} \leftarrow x_{k,c} \oplus x_{k,t}$$
  $$z_{k,c} \leftarrow z_{k,c} \oplus z_{k,t}$$

- A Phase gate on qubit j performs

  $$z_{k,j} \leftarrow z_{k,j} \oplus x_{k,j}$$

The diagonalization procedure processes can be described as follows:

1. Identify Pivot: Find the first column j (the pivot column) for which $x_{k,j} = 1$.
2. Eliminate Other Entries: For every other columns $l > j$ with $x_{k,l} = 1$, apply a CNOT gate with control j and target l. This operation adds column j to column l in the $\mathbf{x}$-block, zeroing out the entry $x_{k,l}$.
3. Clear the Pivot: After step 2, only the pivot column j has a 1 in $\mathbf{x}$-block. At the same time only j-th column of the $\mathbf{z}$-block has changed. Moreover its values are equal to $z_{k,j} = \bigoplus_{l=1}^{M} z_{k,l}$, where l runs over all non zero bit in the initially given $\mathbf{x}$. It can be seen that $\bigoplus_{l=1}^{M} z_{k,l} = \mathbf{x} \cdot \mathbf{z}_k$.
   - If we are given a set, characterized by $\mathbf{x} \cdot \mathbf{z}_k = 0 \pmod 2$, apply a Hadamard gate on qubit j. This swaps the 1 in the $\mathbf{x}$-block with the 0 in the $\mathbf{z}$-block, completing the zeroing of $\mathbf{x}$ block.
   - If we are given a set, characterized by $\mathbf{x} \cdot \mathbf{z}_k = 1 \pmod 2$, first apply a Phase gate on qubit j. This sets $z_{k,j} \leftarrow z_{k,j} \oplus x_{k,j} = 0$ (since $z_{k,j} = 1$ and $x_{k,j} = 1$). Then, apply a Hadamard gate on qubit j to swap the 1 in $\mathbf{x}$ with the 0 in $\mathbf{z}$.

The final result is a tableau with $\mathbf{x} = \mathbf{0}$, meaning all Pauli strings have been diagonalized. Additionally, it is easy to see that after the diagonalization procedure only a single bit is changed in $\mathbf{z}_k$ strings. That is, we set pivot column bit j to 1 in each $\mathbf{z}_k$.

We now analyze the cumulative sign change from the specific CNOT operations applied in Step 2 of the diagonalization procedure. In this step, for a fixed target qubit t, a CNOT gate with control c (the pivot) and target t is applied to every other column k in the set for which $x_{t,k} = 1$.

Substituting these values simplifies the update rule significantly:

$$r_k \leftarrow r_k \oplus (1 \cdot z_{k,t} \cdot (1 \oplus z_{k,c} \oplus 1)) = r_k \oplus z_{k,t} z_{k,c}. \tag{A19}$$

Using it directly based on step 2 in the diagonalization procedure, one can obtain following result

$$r = \bigoplus_{i=1}^{M} z_i \left(\bigoplus_{j=0}^{i-1} z_j \right) = \sum_{0 \le j < i \le M} z_i z_j \pmod 2 = C_t^2 \bmod 2, \tag{A20}$$

where z_i denotes i-th index in $\mathbf{z}$ (that is i-th column in table), and C_t^2 is number of combinations with $t = \mathbf{x} \cdot \mathbf{z}$ being the number of bits in $\mathbf{z}$ equal to 1. Simplifying it further we can get

$$r = C_t^2 \bmod 2 = \frac{t(t-1)}{2} \bmod 2 = \lfloor \frac{t}{2} \bmod 2 \rfloor, \tag{A21}$$

Additionally, the sign bit r_k remains unchanged after the application of a Hadamard gate. The sign update rule for a Hadamard gate on qubit t is $r_k \leftarrow r_k \oplus (x_{k,t} \cdot z_{k,t})$. This update is trivial because a Hadamard gate is only applied to a qubit t when its state in the tableau is prepared such that either $x_{k,t} = 0$ or $z_{k,t} = 0$, ensuring the product $x_{k,t} \cdot z_{k,t}$ is always zero.

In contrast, the sign update rule for a Phase (S) gate is identical: $r_k \leftarrow r_k \oplus (x_{k,t} \cdot z_{k,t})$. However, an S gate is applied under precisely the opposite condition when both $x_{k,t} = 1$ and $z_{k,t} = 1$. Consequently, the product $x_{k,t} \cdot z_{k,t}$ equals 1, and the sign bit r_k is flipped.

That is the sign that appears in diagonalization is $(-1)^r$, where

$$r = \begin{cases} \frac{t}{2} \bmod 2 & \text{if } t = \mathbf{x} \cdot \mathbf{z} = 0 \pmod 2, \\ \frac{t+1}{2} \bmod 2 & \text{if } t = \mathbf{x} \cdot \mathbf{z} = 1 \pmod 2. \end{cases} \tag{A22}$$

or,

$$r = \left(\frac{t + (t \bmod 2)}{2} \right) \bmod 2. \tag{A23}$$

Finally the sign that appears during diagonalization is given in the form

$$(-1)^r = (-1)^{\left(\frac{t+(t \bmod 2)}{2}\right) \bmod 2} = (-1)^{\left(\frac{t+(t \bmod 2)}{2}\right)} \tag{A24}$$

□

Proof of Proposition 3. Let the velocity-model operator be decomposed as $\tilde{S} = \sum_{j=1}^{T} \alpha_j Z^{\mathbf{z}_j}$, where $Z^{\mathbf{z}} = \bigotimes_{k=1}^{Dn+\lceil \log_2(D+1) \rceil} Z^{z_k}$ is a Pauli string composed solely of I and Z matrices, and the Hamiltonian as $H_D = \sum_{j=1}^{K} \beta_j P_j$, where P_j are Pauli strings.

We note that the operators $\tilde{B}_j = I \otimes \cdots \otimes B_j \otimes \cdots \otimes I$ are constructed by embedding B_j into a larger tensor product. Consequently, the Pauli strings constituting $\tilde{B}_j$ are those of B_j, padded with zeros in the $\mathbf{x}$ and $\mathbf{z}$ vectors at positions corresponding to the identity matrices. Therefore, the structure of the commuting sets in H_D, as established by Proposition 1 and

Corollary 1, remains valid. The Hamiltonian H_D can be expressed as a sum over Γ mutually commuting sets:

$$H_D = \sum_{\gamma=1}^{\Gamma} X^{\mathbf{x}_\gamma} \left(\sum_{k=1}^{K_\gamma} i^{\mathbf{x}_\gamma \cdot \mathbf{z}_k} \beta_k Z^{\mathbf{z}_k} \right), \tag{A25}$$

where the phase factor $i^{\mathbf{x}_\gamma \cdot \mathbf{z}_k}$ arises from the Walsh operator definition (Equation (9)). According to Corollary 1, and since H_D is real-valued, the number of sets is $\Gamma = s(D,d,n) = D\left(2^{\lceil \log_2 d \rceil} + (n - \lceil \log_2 d \rceil)d\right)$, and the number of strings per set is $K_\gamma = 2^{n+\lceil \log_2(D+1) \rceil - 1}$, where the term -1 accounts for the constraint of an even number of Y operators.

We now analyze the transformed Hamiltonian $\tilde{H}_D = \tilde{S} H_D \tilde{S}$:

$$\begin{aligned} \tilde{S} H_D \tilde{S} &= \left(\sum_{j=1}^{T} \alpha_j Z^{\mathbf{z}_j} \right) \left[\sum_{\gamma=1}^{\Gamma} X^{\mathbf{x}_\gamma} \left(\sum_{k=1}^{K_\gamma} i^{\mathbf{x}_\gamma \cdot \mathbf{z}_k} \beta_k Z^{\mathbf{z}_k} \right) \right] \left(\sum_{l=1}^{T} \alpha_l Z^{\mathbf{z}_l} \right) \\ &= \sum_{\gamma=1}^{\Gamma} \sum_{j,l=1}^{T} \alpha_j \alpha_l Z^{\mathbf{z}_j} X^{\mathbf{x}_\gamma} \left(\sum_{k=1}^{K_\gamma} i^{\mathbf{x}_\gamma \cdot \mathbf{z}_k} \beta_k Z^{\mathbf{z}_k} \right) Z^{\mathbf{z}_l}. \end{aligned} \tag{A26}$$

Using the Pauli commutation relation $Z^{\mathbf{z}} X^{\mathbf{x}} = (-1)^{\mathbf{x} \cdot \mathbf{z}} X^{\mathbf{x}} Z^{\mathbf{z}}$ and the fact that Z operators commute, we simplify:

$$\begin{aligned} \tilde{S} H_D \tilde{S} &= \sum_{\gamma=1}^{\Gamma} \sum_{j,l=1}^{T} \sum_{k=1}^{K_\gamma} i^{\mathbf{x}_\gamma \cdot \mathbf{z}_k} (-1)^{\mathbf{x}_\gamma \cdot \mathbf{z}_j} \alpha_j \alpha_l \beta_k X^{\mathbf{x}_\gamma} Z^{\mathbf{z}_j \oplus \mathbf{z}_k \oplus \mathbf{z}_l} \\ &= \sum_{\gamma=1}^{\Gamma} X^{\mathbf{x}_\gamma} \left(\sum_{p=1}^{K'_\gamma} \nu_p Z^{\mathbf{z}_p} \right). \end{aligned} \tag{A27}$$

In the final expression, the index p runs over all unique binary vectors $\mathbf{z}_p = \mathbf{z}_j \oplus \mathbf{z}_k \oplus \mathbf{z}_l$ generated by the indices $j, l \in \{1, \ldots, T\}$ and $k \in \{1, \ldots, K_\gamma\}$. The coefficient ν_p is the sum of all terms $i^{\mathbf{x}_\gamma \cdot \mathbf{z}_k} (-1)^{\mathbf{x}_\gamma \cdot \mathbf{z}_j} \alpha_j \alpha_l \beta_k$ for which $\mathbf{z}_j \oplus \mathbf{z}_k \oplus \mathbf{z}_l = \mathbf{z}_p$.

Two key observations follow from Equation (A27):

1. The commuting sets of $\tilde{S} H_D \tilde{S}$ are identical to those of H_D as they are characterized by the same $\mathbf{x}_\gamma$ strings.
2. The number of Pauli strings K'_γ within each set γ is bounded by the number of unique $\mathbf{z}_p$ vectors that can be generated. This number is at most the minimum of two values:
   - The total number of combinations: $T^2 K_\gamma$.
   - The total number of all possible diagonal Pauli strings of length $L = Dn + \lceil \log_2(D+1) \rceil$, which is $(D+1)N^D$. However, since $\tilde{S} H_D \tilde{S}$ remains a real-valued matrix, its Pauli decomposition can only contain strings with an even number of Y operators. For a fixed $\mathbf{x}_\gamma$, this restricts the associated $\mathbf{z}_p$ vectors, effectively halving the number of possibilities to $(D+1)N^D/2$.

Thus, $K'_\gamma \leq \min\left(T^2 K_\gamma, (D+1)N^D/2\right)$, and the total number of Pauli strings in the decomposition of $\tilde{H}_D$ is bounded by

$$g_H \leq \Gamma \cdot \min\left(T^2 K_\gamma, (D+1)N^D/2\right). \tag{A28}$$

Substituting the expressions $\Gamma \approx Ddn$ (for $d \ll N$), $K_\gamma = (D+1)N/2$, and $N^D = 2^{Dn}$, the scaling of g_H can be written as

$$g_H = O\left(D^2 dn N \min\left(T^2, N^{D-1}\right)\right). \tag{A29}$$

□

References

1. Virieux, J.; Operto, S. An overview of full-waveform inversion in exploration geophysics. *Geophysics* **2009**, *74*, WCC1–WCC26. [CrossRef]
2. Taflove, A.; Hagness, S.C.; Piket-May, M. Computational electromagnetics: The finite-difference time-domain method. In *The Electrical Engineering Handbook*; Elsevier: Amsterdam, The Netherlands, 2005; Volume 3, p. 15.
3. LeVeque, R.J. *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*; Society for Industrial and Applied Mathematics (SIAM): Philadelphia, PA, USA, 2007.
4. Tarantola, A. *Inverse Problem Theory and Methods for Model Parameter Estimation*; Society for Industrial and Applied Mathematics (SIAM): Philadelphia, PA, USA, 2005.
5. Pratt, R.G. Seismic waveform inversion in the frequency domain, Part 1: Theory and verification in a physical scale model. *Geophysics* **1999**, *64*, 888–901. [CrossRef]
6. Yee, K. Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media. *IEEE Trans. Antennas Propag.* **1966**, *14*, 302–307.
7. Harrow, A.W.; Hassidim, A.; Lloyd, S. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.* **2009**, *103*, 150502. [CrossRef]
8. Berry, D.W. High-order quantum algorithm for solving linear differential equations. *J. Phys. A Math. Theor.* **2014**, *47*, 105301. [CrossRef]
9. Costa, P.C.S.; Jordan, S.; Ostrander, A. Quantum algorithm for simulating the wave equation. *Phys. Rev. A* **2019**, *99*, 012323. [CrossRef]
10. Bösch, C.; Schade, M.; Aloisi, G.; Keating, S.D.; Fichtner, A. Quantum wave simulation with sources and loss functions. *Phys. Rev. Res.* **2025**, *7*, 033225. [CrossRef]
11. Wright, L.; Mc Keever, C.; First, J.T.; Johnston, R.; Tillay, J.; Chaney, S.; Rosenkranz, M.; Lubasch, M. Noisy intermediate-scale quantum simulation of the one-dimensional wave equation. *Phys. Rev. Res.* **2024**, *6*, 043169. [CrossRef]
12. Suau, A.; Staffelbach, G.; Calandra, H. Practical quantum computing: Solving the wave equation using a quantum approach. *ACM Trans. Quantum Comput.* **2021**, *2*, 1–35. [CrossRef]
13. Sato, Y.; Kondo, R.; Hamamura, I.; Onodera, T.; Yamamoto, N. Hamiltonian simulation for hyperbolic partial differential equations by scalable quantum circuits. *Phys. Rev. Res.* **2024**, *6*, 033246. [CrossRef]
14. Sato, Y.; Tezuka, H.; Kondo, R.; Yamamoto, N. Quantum algorithm for partial differential equations of nonconservative systems with spatially varying parameters. *Phys. Rev. Appl.* **2025**, *23*, 014063. [CrossRef]
15. Arseniev, B.; Guskov, D.; Sengupta, R.; Biamonte, J.; Zacharov, I. Tridiagonal matrix decomposition for Hamiltonian simulation on a quantum computer. *Phys. Rev. A* **2024**, *109*, 052629. [CrossRef]
16. Arseniev, B.; Guskov, D.; Sengupta, R.; Zacharov, I. High-order schemes for solving partial differential equations on a quantum computer. *Phys. Rev. A* **2025**, *111*, 042625. [CrossRef]
17. Berry, D.W.; Ahokas, G.; Cleve, R.; Sanders, B.C. Efficient quantum algorithms for simulating sparse Hamiltonians. *Commun. Math. Phys.* **2007**, *270*, 359–371. [CrossRef]
18. Childs, A.M.; Su, Y.; Tran, M.C.; Wiebe, N.; Zhu, S. Theory of trotter error with commutator scaling. *Phys. Rev. X* **2021**, *11*, 011020. [CrossRef]
19. Kawase, Y.; Fujii, K. Fast classical simulation of Hamiltonian dynamics by simultaneous diagonalization using Clifford transformation with parallel computation. *Comput. Phys. Commun.* **2023**, *288*, 108720. [CrossRef]
20. Van Den Berg, E.; Temme, K. Circuit optimization of Hamiltonian simulation by simultaneous diagonalization of Pauli clusters. *Quantum* **2020**, *4*, 322. [CrossRef]
21. Welch, J.; Greenbaum, D.; Mostame, S.; Aspuru-Guzik, A. Efficient quantum circuits for diagonal unitaries without ancillas. *New J. Phys.* **2014**, *16*, 033040. [CrossRef]
22. Sato, H.; Fehler, M.C.; Maeda, T. *Seismic Wave Propagation and Scattering in the Heterogeneous Earth*; Springer: Berlin/Heidelberg, Germany, 2012; Volume 496.
23. Low, G.H.; Chuang, I.L. Optimal Hamiltonian simulation by quantum signal processing. *Phys. Rev. Lett.* **2017**, *118*, 010501. [CrossRef]
24. Berry, D.W.; Childs, A.M.; Cleve, R.; Kothari, R.; Somma, R.D. Simulating Hamiltonian dynamics with a truncated Taylor series. *Phys. Rev. Lett.* **2015**, *114*, 090502. [CrossRef]
25. Kandala, A.; Temme, K.; Córcoles, A.D.; Mezzacapo, A.; Chow, J.M.; Gambetta, J.M. Error mitigation extends the computational reach of a noisy quantum processor. *Nature* **2019**, *567*, 491–495. [CrossRef]
26. Preskill, J. Quantum computing in the NISQ era and beyond. *Quantum* **2018**, *2*, 79. [CrossRef]
27. Cerezo, M.; Arrasmith, A.; Babbush, R.; Benjamin, S.C.; Endo, S.; Fujii, K.; McClean, J.R.; Mitarai, K.; Yuan, X.; Cincio, L.; et al. Variational quantum algorithms. *Nat. Rev. Phys.* **2021**, *3*, 625–644. [CrossRef]

28. Peruzzo, A.; McClean, J.; Shadbolt, P.; Yung, M.H.; Zhou, X.Q.; Love, P.J.; Aspuru-Guzik, A.; O'brien, J.L. A variational eigenvalue solver on a photonic quantum processor. *Nat. Commun.* **2014**, *5*, 4213. [CrossRef] [PubMed]
29. Childs, A.M.; Ostrander, A.; Su, Y. Faster quantum simulation by randomization. *Quantum* **2019**, *3*, 182. [CrossRef]
30. Jin, S.; Liu, N.; Yu, Y. Quantum simulation of partial differential equations via Schrödingerization. *Phys. Rev. Lett.* **2024**, *133*, 230602. [CrossRef]
31. Zacharov, I.; Arslanov, R.; Gunin, M.; Stefonishin, D.; Bykov, A.; Pavlov, S.; Panarin, O.; Maliutin, A.; Rykovanov, S.; Fedorov, M. "Zhores"—Petaflops supercomputer for data—Driven modeling, machine learning and artificial intelligence installed in Skolkovo Institute of Science and Technology. *Open Eng.* **2019**, *9*, 512–520. [CrossRef]