# Quantum Algorithms for Chemistry and Differential Equations

by

Philipp Schleich

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy

Department of Computer Science
University of Toronto

© Copyright 2025 by Philipp Schleich

Quantum Algorithms for Chemistry and Differential Equations

Philipp Schleich
Doctor of Philosophy

Department of Computer Science
University of Toronto
2025

# Abstract

Quantum computers were first proposed to perform simulation – in particular, quantum simulation. The premise is that such computers can simulate quantum dynamics exponentially faster than a classical computer. Some of the early proposals of practical applications of quantum computers come from quantum chemistry. Algorithms for more general systems, described by ordinary and partial differential equations, emerged later on. This thesis grounds on an extensive amount of prior research and proposes a set of improvements as well as new algorithms or algorithmic tools, focusing on applications from quantum chemistry and differential equations.

First, we propose a technique relating to finding ground-states for electronic systems via the variational quantum eigensolver. The presented method enables solving the problem on early quantum hardware by a partitioning of the computational problem into smaller problems, using only classical resources to perform the partitioning. Next, the thesis considers simulation problems from chemistry in a more general sense. Following the idea that dynamical simulation of many chemical systems is efficient, whereas preparing ground-states is a 'hard' problem, our computational framework leverages simulation routines to prepare molecular states as an input to a computation, relying only on 'hard' input states for small problem instances. The preparation relies on dynamical scattering of initial fragments supported by a trap potential, allowing to estimate the success probability for certain bonds. A weak measurement scheme enables preparation of molecular states.

Furthermore, this dissertation investigates quantum algorithms for differential equations. More specifically, the the Carleman linearization approach is considered to solve nonlinear differential equations. This entails bounding the Carleman truncation error, re-scaling of the solution to avoid an exponentially vanishing success probability, and discussing higher-order discretization. Moreover, this thesis introduced a penalty-based method to enforce constraints on the solution of differential equations, with inspiration from complex absorbing potentials and the quantum Zeno effect. We bound the approximation errors and embed the penalty method into an efficient quantum algorithm, with computational overhead only logarithmic in the penalty's strength. Efficiency is enabled by an interaction-picture approach with fast-forwardable constraint projections.

# Acknowledgements

The beginning of my PhD was in the middle of the Covid-19 pandemic. When I landed in Toronto, the University of Toronto arranged for a shuttle to a hotel – for a two-week quarantine not being able to leave my room, where I also spent my birthday. This sounds like a crazy start, though, I enjoyed this little bubble at the beginning of my PhD journey. Even more thankful I am that both during the very quiet Covid times and the more buzzing "restart phase" once things opened, there were plenty of great humans that were essential for this thesis to have come into existence.

First, I am grateful for my supervisor Alán, for giving me freedom and encouragement to explore, several opportunities for travel, and a great lab environment. Thank you to Dominic W. Berry for hosting me in Sydney, it was a privilege to work with you and learn from you. Further thank you to Nathan Wiebe, for the whiteboard discussions and for frequently mitigating my inner pessimist. There were many things Dominic or Nathan explained to me, where months after I realized that they already told me the solution – I just did not see it yet. I was very fortunate to have Lukasz Cincio from LANL mentoring me throughout the Quantum Computing Summer School. Further, I thank Artur Izmaylov for being on my committee and Stephen P. Jordan for serving as external appraiser on my exam committee.

Many great postdocs and students have been supportive during my PhD. Without their mentorship and companionship, this would have been impossible. A monstrous thank you to Jakob S. Kottmann, Lasse B. Kristensen, and Abhinav Anand, for being my friends and mentors. Similarly, I am extremely grateful for the friendship of my lab-mates Cher-Tian Ser, Ella Rajaonson, and Marta Skreta. Many thanks to my office buddies in the asylum and the graveyard, Rodrigo Vargas-Hernández, Chong Sun, and Davide Avagliano. Abdul Aldossary deserves a special thanks for the cookies that fueled the submission of this document. I am grateful for the mentorship of Mohsen Bagherimehrab and Thi Ha Kyaw. Sincere thanks extend to many more members of the MatterLab; Luis Mantilla, Sean Park, Matthew Choi, Changhyeok Choi, Austin Cheng, Phillip W. K. Jensen, Marko Huang, Gary Tom, Sumner Alperin-Lea, Yeonghun Kang, Jorge Campos, Shi Xuan Leong, Sam Zhang, Manuel Drehwald, Zijian Zhang, Luca Thiede, Samantha Corapi, Jiaru Bai, Mohammad Ghazi-Vakili, Karthik Panicker, etc. etc., as well as Abhishek Rajput and the rest of the Wiebe group as well as Deepanshu Kush as hallway co-occupants. Specifically, anyone that went camping with me at some point, was part of the running group, or went climbing.

Beyond, I am fortunate to have met a lot of great people outside of the University of Toronto, which holds in particular for my time at the Institute for Pure and Applied Mathematics at UCLA, and the Sydney Quantum Academy and Macquarie University. In particular, Tyler Kharazi (who gave me the idea for the mixing time discussion related to DEs in the introduction), Alicia Negre, Jakob Huhn, Ahmad Alkadri, and Brandon Barton for the fun we had in LA and beyond. At this note, I'm also deeply indebted to Marius Junge, Peixue Wue, and Zhiyan Ding; for teaching me fancy math and giving me ideas for research I want to do in the future.

The MatterLab admin team deserves another special thank you; specifically, Irene Zuniga, Mukesh Dodain, Chris Crebolder, Angela Zhao, and Melissa Szopa.

Lastly, thanks to my family, especially to my mum, my dad and my brother, who accepted that I moved across the Atlantic and for a while further across the Pacific, and that I will not be living back in Germany for the foreseeable future.

# Contents

## Citations to previously published work

At the time of writing, Section 2.1 has appeared in publication whereas Sections 2.2 and 3.2 are actively undergoing peer review, Section 3.1 has been accepted for publication in npj Quantum Information.

- Philipp Schleich, Joseph Boen, Lukasz Cincio, Abhinav Anand, Jakob S Kottmann, Sergei Tretiak, Pavel A Dub, and Alán Aspuru-Guzik. "Partitioning quantum chemistry simulations with clifford circuits". In: *Journal of Chemical Theory and Computation* 19.15 (2023), pp. 4952–4964

- Philipp Schleich, Lasse Bjørn Kristensen, Jorge A Angulo Campos, Abdulrahman Aldossary, Davide Avagliano, Mohsen Bagherimehrab, Christoph Gorgulla, Joe Fitzsimons, and Alán Aspuru-Guzik. "Chemically Motivated Simulation Problems are Efficiently Solvable by a Quantum Computer". In: *arXiv preprint arXiv:2401.09268* (2024)

- Pedro Costa, Philipp Schleich, Mauro ES Morales, and Dominic W Berry. "Further improving quantum algorithms for nonlinear differential equations via higher-order methods and rescaling". In: *arXiv preprint arXiv:2312.09518* (2023)

- Philipp Schleich, Tyler Kharazi, Xiangyu Li, Jin-Peng Liu, Alán Aspuru-Guzik, and Nathan Wiebe. "Arbitrary Boundary Conditions and Constraints in Quantum Algorithms for Differential Equations via Penalty Projections". In: *arXiv preprint arXiv:2506.21751* (2025).

I have contributed to several other publications during the course of my PhD which did not make it into this thesis:

- Philipp Schleich*, Marta Skreta*, Lasse Bjørn Kristensen, Rodrigo Vargas-Hernandez, and Alan Aspuru-Guzik. "Quantum Deep Equilibrium Models". In: *Advances in Neural Information Processing Systems*. Vol. 37. 2024, pp. 31940–31967

- Zhiyan Ding, Marius Junge, Philipp Schleich, and Peixue Wu. "Lower bound for simulation cost of open quantum systems: Lipschitz continuity approach". In: *Communications in Mathematical Physics* 406.3 (2025), p. 60

- Philipp Schleich, Jakob S Kottmann, and Alán Aspuru-Guzik. "Improving the accuracy of the variational quantum eigensolver for molecular systems by the explicitly-correlated perturbative [2] R12-correction". In: *Physical Chemistry Chemical Physics* 24.22 (2022), pp. 13550–13564

- Abhinav Anand*, Philipp Schleich*, Sumner Alperin-Lea*, Phillip WK Jensen*, Sukin Sim, Manuel Díaz-Tinoco, Jakob S Kottmann, Matthias Degroote, Artur F Izmaylov, and Alán Aspuru-Guzik. "A quantum computing view on unitary coupled cluster theory". In: *Chemical Society Reviews* 51.5 (2022), pp. 1659–1684

- Mauro ES Morales, Lirandë Pira, Philipp Schleich, Kelvin Koor, Pedro Costa, Dong An, Alán Aspuru-Guzik, Lin Lin, Patrick Rebentrost, and Dominic W Berry. "Quantum linear system solvers: A survey of algorithms and applications". In: *arXiv preprint arXiv:2411.02522* (2024)

- Mohsen Bagherimehrab, Luis Mantilla Calderón, Dominic W. Berry, Philipp Schleich, Mohammad Ghazi Vakili, Abdulrahman Aldossary, Jorge A. Campos Gonzalez Angulo, Christoph Gorgulla, and Alán Aspuru-Guzik. "Faster algorithmic quantum and classical simulations by corrected product formulas". In: *arXiv preprint arXiv:2409.08265* (2024)
- Philipp Schleich, Luis Mantilla Calderón, Chong Sun, Mohsen Bagherimehrab, Abdulrahman Aldossary, Jakob S. Kottmann, and Alán Aspuru-Guzik. *Quantum Computing for Quantum Chemistry*. American Chemical Society, May 2025

# Chapter 1

# Introduction

## 1.1 Computation in Scientific Disciplines

Natural sciences strive to understand how the world around us works, e.g., by identifying causal and correlation structures. The process of discovery is enabled by observations – experiments. Then, the deduction of rules and patterns in these observations allows one to find and confirm theories, often with mathematics as the underlying language, which then can lead to putting forward new hypotheses. Be it the venture for fundamental understanding or for application: Scientific discovery by experiments and theory has long been supported by computers through simulation, namely the computational representation of physical phenomena. The capacity of computers to perform numerical experiments is of aid whenever experiments may be very costly, materials or forces involved potentially dangerous, and when the complexity of the problem at hand prohibits analytic treatment. Table 1.1 shows a list that was put forward as a selection of the ten most impactful algorithms of the

| **Year** | **Algorithm** |
|---|---|
| 1946 | The Metropolis Algorithm for Monte Carlo |
| 1947 | Simplex Method for Linear Programming |
| 1950 | Krylov Subspace Method |
| 1951 | The Decompositional Approach to Matrix Computations |
| 1957 | The Fortran Optimizing Compiler |
| 1959 | QR Algorithm for Eigenvalues |
| 1962 | Quicksort |
| 1965 | Fast Fourier Transform (FFT) |
| 1977 | Integer Relation Detection |
| 1987 | Fast Multipole Method |

Table 1.1: The Top 10 "most impactful" algorithms of the 20th century outlined in [SD00].

twentieth century. Many of these algorithms have become essential, often in a fundamental way that supports a multitude of applications; a small selection is mentioned in the following. For instance, the Fast-Fourier Transform is ubiquitous in signal processing and data analysis. Many applications that rely on physical models utilize spectral properties, that can be obtained by Krylov subspace methods such as Lanczos iterations for eigenvalues or the QR decomposition. Krylov subspace methods also find widespread application in the solution of linear systems of equations, for example through the conjugate gradients or the GMRES [SS86] algorithms.

These algorithms are based on the standard, 'classical' computational model, the framework typically understood when discussing computing. In the early 1980s, Benioff, Manin and Feynman [Ben80; Man80; Fey82] have independently put forward the idea of performing computation, or in particular simulation, using a quantum mechanical model. Soon after this was generalized to a universal model of computation [Deu85]. The simulation of the dynamics of quantum-mechanical systems was conjectured as a prime use-case for quantum computers already at the time and it remains one of the most promising candidates to date. Several additional breakthroughs, most famously by Peter Shor, were necessary so that quantum computing was able to gain wider traction. His algorithm for factoring composite numbers into their prime factors [Sho94] was the first algorithm demonstrating practical use. Furthermore, the noise and error susceptibility of quantum computers seemed even theoretically prohibitive, before Shor's foundational work [Sho95] laid the groundwork for quantum error correction. Other algorithms that supported increasing interest in quantum computers were, exemplarily: The Deutsch-Josza algorithm from 1992 can identify whether a function that maps $n$ bits to one bit either has a constant output, or is uniformly spread across the possible outputs [DJ92]. It requires only one quantum query compared to a number of classical queries that is exponential in $n$. The Bernstein-Vazirani algorithm builds upon the latter algorithm from 1997 and is able to identify a string encoded in a function output of length $n$ with a single query rather than at least $\Omega(n)$ classical queries [BV97]. Simon's algorithm (1994) can tell us whether a function mapping $n$ bits to $m$ bits, with $m \geq n$, is one-to-one or two-to-one, with an 'exponential speedup', meaning that the number of quantum queries necessary is exponentially smaller than the number of classical queries [Sim97]. Lastly, Grover's algorithm [Gro98] allows to find the input so that a black-box oracle attains a specific output, which enables search in unstructured databases of size $N$ with complexity $O(\sqrt{N})$ rather than $O(N)$ – this speedup is not as strong as the others presented in this paragraph, yet of universal use, in particular as a subroutine.

This thesis is centred around the question how quantum computing can help in addressing computational problems from (quantum) chemistry and physical sciences in a more general sense, that is, phenomena that are described by differential equations. When it comes to quantum chemistry, there is an intuitive rationale for a quantum advantage as we simulate quantum systems with another system that 'is quantum'; specific applications and innovative algorithms remain to be explored more thoroughly. This will be subject of study in Chapter 2. Furthermore, for the simulation of more general, non-unitary and potentially nonlinear dynamics, a rationale to expect a *general* quantum advantage in simulation is not immediate. The potential practical usefulness for the simulation of such systems is likely high though, as many numerical procedures in engineering, such as simulation of fluids and structural mechanics, are characterized by non-unitary dynamics. The work presented in Chapter 3 aims to help staking out the realm where quantum computers may be useful in simulating classical systems.

The following sections present a brief overview of the background and motivation for this thesis. Namely, computation in the context of general physical problems is introduced in Section 1.1 and more concretely for computational chemistry in Section 1.2.2. In particular, computational bottlenecks and areas where the novel computational model can help are identified. Subsequently, Section 1.2 covers existing quantum techniques to tackle these problems and in particular chances for quantum algorithms to assist in solving problems more efficiently.

**Scientific Computing** First, let us look more closely what scientific computing can mean, in particular if we want to use computation to investigate problems from mathematical sciences. Broadly, we can divide this into deterministic and probabilistic approaches. In deterministic computing, the goal is to devise an algorithm [CLRS22] $\mathcal{A}$ so that, after a sequence of finite computational steps, an input $x \in \mathcal{X}$ is mapped to its correct output $y \in \mathcal{Y}$, or, in the case of a numerical algorithm, up to some error $\varepsilon$. Thus, we require

$$\mathcal{A}_\varepsilon : x \mapsto y, \quad d(y, y_{\text{true}}) \leq \varepsilon. \tag{1.1.1}$$

Here we assume that the problem-at-hand admits a certain error metric $d(y, y') : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_0^+$. Now, if the algorithm is probabilistic, which we call $\mathcal{P}_\varepsilon$, then the result is meant to be sufficiently accurate with high probability $1 - \delta$,

$$\Pr\left(d(\mathcal{P}_\varepsilon(x), y_{\text{true}}) \leq \varepsilon\right) \geq 1 - \delta. \tag{1.1.2}$$

We continue by looking at a specific, illustrative example that appears in many areas of scientific computation, namely the task of integrating a function, $\int_0^1 f(x)\mathrm{d}x$. There are many numerical methods to solve this; one of the simplest deterministic approaches is the trapezoidal rule, and Monte-Carlo sampling can provide a simple probabilistic one. With the trapezoidal rule, we get a simple recipe that

$$\sum_{j=0}^{N-1} \frac{f(x_j) + f(x_{j+1})}{2}(x_{j+1} - x_j) \approx_\varepsilon \int_0^1 f(x)\mathrm{d}x, \tag{1.1.3}$$

where $\varepsilon \in O(\frac{1}{N^2})$, with constants depending on the derivatives of $f$. The notation $\approx_\varepsilon$ denotes up to a difference $\varepsilon > 0$, i.e., $a \approx_\varepsilon b$ means $|a - b| \leq \varepsilon$. The trapezoidal rule is a numerical algorithm that suits the language above, in the sense that the sum in Eq. (1.1.3) describes a sequence of arithmetic steps that, given the ability to compute the function $f(x)$ for a specific $x$, maps the $N + 1$ points $\{x_j\}_{j=0}^N$ to an approximation of the desired integral value. Intuitively, this method approximates an integral as the area under a curve via step-wise averages. Hence, more informed choices of such an average would lead to better error scaling, which is observed in more advanced techniques we do not go into here (see e.g. [DR07]).

If we use a so called Monte-Carlo sampler to compute the value of the integral, we note that we can write

$$\int_0^1 f(x)\mathrm{d}x = \int_0^1 f(x)p_{\text{unif}}(x)\mathrm{d}x, \tag{1.1.4}$$

where $p_{\text{unif}}(x) \sim U(0, 1)$ is the probability density function of the uniform distribution across the interval $[0, 1]$. Then, $\int_0^1 f(x)p_{\text{unif}}(x)\mathrm{d}x = \mathbb{E}_{x\sim U}[f]$, and we have an unbiased estimator for this expectation given by

$$\frac{1}{M}\sum_{j=1}^{M} f(x_j) \approx_\varepsilon \int_0^1 f(x)p_{\text{unif}}(x)\mathrm{d}x, \quad x_j \sim U(0, 1) \tag{1.1.5}$$

where one can show via Hoeffding's inequality that for $\varepsilon = \sqrt{\frac{\text{Var}(f)}{M}}$, the probability that the approximation error is larger than $\varepsilon$ decays exponentially to zero. In other words, we want to use $M = \frac{\text{Var}(f)}{\varepsilon^2}$ samples to observe $\varepsilon$ error with probability close to one. The associated probabilistic

algorithm then relies again on the ability to point-wise evaluate $f(x)$, and instead of combining a fixed set of points via a fixed rule, we randomly sample an $x_j$ from a uniform distribution and average $M$ of their values.

There are many more examples for such numeric routines that appear in several situations in computational science – eigenvalue problems, the solution of differential equations, linear and nonlinear systems of equations, etc. In computer science, the notion of such numerical methods being efficient is oftentimes that the complexity to prepare an output is polynomial in relevant system parameters. Here, complexity quantifies the computational cost, which can be the memory required to store information (space complexity) or the number of operations that need to occur (time complexity).

Numerical methods to date are very successful and still a very active field of research. Now, this raises the question what chances there are for quantum algorithms. Typically, quantum algorithms help by providing a 'speed-up' in the sense that if a classical algorithm has time or query complexity $O(n)$, a quantum algorithm has complexity $O(g(n))$ where we want that $g(n)$ grows slower than linear so that the quantum algorithm is faster. Exponential speed-up means $g(n) = \log(n)$, or polynomial speed-ups of order $p \geq 1$ have the form $g(n) = n^{1/p}$, $p > 1$. Query complexity means that the algorithm relies on a black-box 'oracle' that requires a realization. In the integration example above, the function $f(x)$ takes the role of such an oracle and the associated query complexity is how many of such function evaluations need to take place. Here, we can make an observation when comparing probabilistic and deterministic methods, in the context of the integration example above: Using Monte-Carlo methods in higher dimensions leads to a query complexity in terms of how often we evaluate $f(x)$ that does not exponentially increase in the dimensionality, as common for direct integration formulas like the trapezoidal rule. However, there are other aspects to consider, such as the variance of the underlying function. Within the same example, we already point out a certain quantum speed-up here: Using the amplitude estimation algorithm [BHMT02], a quantum computer can estimate the integral's value using only $M_Q = \frac{\sqrt{\mathrm{Var}(f)}}{\varepsilon}$ samples (meaning quantum realizations of $f(x)$). Furthermore, one can show that this matches a lower bound [Gro98; BHMT02] and is thus optimal. This means that there is no algorithm that has asymptotically better query complexity for this task in generic situations.

Some additional aspects that we ignore for now are: We need less quantum samples. But it is not clear whether a quantum sample, or a quantum query, is more or less costly than a single classical query. Quantum computers undergo more, and more complicated, errors than classical computers. Correcting for these errors comes at a cost – we do not go into details here and refer to literature (e.g., [Rof19]). It turns out that, according to recent estimates [Bab+21], quadratic speedups such as the one from amplitude estimation in our example may be lost within problem sizes of practical interest (i.e., computations that take at most several weeks or months rather than years).

## 1.2 Scientific Quantum Computing

Being familiar with some models of classical numerical computation, we continue to take a closer look at quantum computing.

### 1.2.1 Brief Introduction to Quantum Computation

Computers in general are devices that are able to represent, operate on and output information. Beyond analogue computers, this requires a fundamental unit of information. For classical devices, information is composed of 'bits', binary values $x \in \{0, 1\}$. This choice is motivated by the capabilities of the hardware: Using a semiconductor, we can distinguish whether the present signal is above a specific threshold. Bits are composed simply by concatenation, so that $x = (x_1 x_2 \cdots x_{N_{\text{bit}}}) \in \{0, 1\}^{N_{\text{bit}}}$.

Before tackling quantum computing, a useful model to consider is probabilistic computing. To that end, let $\vec{p} = (a,\ b)^T$ with $a, b > 0$, where $\|\vec{p}\|_{\ell_1} = a + b = 1$. Here, $a, b$ describe the probability of the probabilistic bit to be measured with outcome '0' or '1'. Sometimes, this is denoted as $\vec{p} = a|0) + b|1)$. We can generalize this to $N_{\text{p}}$-dimensional vectors by expanding the number of possible basis states to

$$\vec{p} = \sum_{j=0}^{N_{\text{p}}-1} p_j |j), \quad \sum_{j=0}^{N_{\text{p}}-1} p_j = 1. \tag{1.2.1}$$

Operations on probabilistic bits are given by matrices that map probabilistic bits to themselves, i.e., that map normalized non-negative vectors in a $2^{N_{\text{p}}}$-dimensional space to other normalized non-negative vectors. Normalization of vectors follows the $\ell_1$-norm. Permutations and (normalized) convex combinations of such satisfy this property, forming a subset of the orthogonal matrices.

Quantum computers assume access to so called qubits as the fundamental unit of information. This is a $\ell_2$-normalized, two-dimensional complex vector, which we may represent as

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \in \mathbb{C}^2, \quad |\alpha|^2 + |\beta|^2 = 1. \tag{1.2.2}$$

Physical instantiations mimic a two-level system – typically, not a true two-level system such as the spin of an electron but a system that is engineered so that a computational two-level subspace is well separated from the rest of the spectrum. E.g., in hardware realizations that rely on superconducting qubits, so called Josephson junctions take care of this [Kja+20]. For the same purpose, neutral atom arrays are operated at the isolated Rydberg frequency [Wu+21].

Apart from merely representing information, there needs to be a means to process it. Operations that map qubit states to qubit states are unitary operations, such that preserve distances under the $\ell_2$-norm. In the two-dimensional qubit case, $2 \times 2$ complex unitary matrices. In reality though it is rarely possible to have an immediate physical implementation of an arbitrary $U(2)$ representation. What then happens is that we divide each given unitary $U$ into a sequence of operations that we expect to be able to implement physically, and call these *gates* in alignment to the classical case. A quantum circuit then is a set of gates $\{U_j\}_{j=1}^m$ so that $U \approx_\varepsilon U_1 U_2 \cdots U_m$, where we measure $\varepsilon$ in the spectral norm,

$$\|U - (U_1 \cdots U_m)\| \leq \varepsilon. \tag{1.2.3}$$

Universal sets of quantum gates are able to satisfy Eq. (1.2.3) up to arbitrary $\varepsilon > 0$.

Multiple qubits are combined via the tensor product, so that an $N_{\text{q}}$-qubit state lives in $\mathbb{C}^{2^{N_{\text{q}}}} = \mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2$. This shows the first aspect of quantum computers that shows promise for power beyond the classical counterpart. The accessible state space here is exponentially larger, meaning that exponential memory is required to store a vector in $\mathbb{C}^{2^{N_{\text{q}}}}$ with classical bits. Operations also directly follow from the single-qubit model, so that any unitary $U$ that maps a $N_{\text{q}}$-qubit state

to another $N_\mathrm{q}$-qubit state comes from the unitary group $U(2^{N_\mathrm{q}})$, or $2^{N_\mathrm{q}} \times 2^{N_\mathrm{q}}$ complex unitary matrices. Now, this directly points at the quantum advantage through the ability to represent exponential information compared to deterministic algorithms, which is a direct consequence of the tensor product structure.

An important difference to the probabilistic bits of before is the normalization. The latter are $\ell_1$-normalized and non-negative. Qubits are $\ell_2$-normalized. This means that beyond the possibility for 'negative probabilities', a phenomenon that is the origin of many provable quantum speedups comes from the possibility of phase differences and thus the construction of constructive and destructive interference effects across the distribution induced by the quantum computers state. An example for this is 'forrelation', [GRZ20] – without going into details, this specific problem exhibits structure that can be exploited by Fourier transforms, which is something that quantum computers can do well.

The construction of quantum algorithms typically draws from a set of 'gates' as a set computational elements that are available to implement. Then, the task of devising an algorithm, beyond the state preparation and measurement steps, is *given initial state $|\psi(0)\rangle$, find a quantum circuit (formed by a composition of $m$ quantum gates) so that $U_1 U_2 \dots U_m |\psi(0)\rangle = |\psi(1)\rangle$, where $|\psi(1)\rangle$ is the sought-after state that encodes the solution to the underlying computational problem.*

What are typical quantum gates? A popular set is given by the Pauli matrices,

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -\mathrm{i} \\ \mathrm{i} & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \tag{1.2.4}$$

which together with the identity form a complete basis for $2 \times 2$ Hermitian matrices. Additionally, there are parametrized rotations $R_P(\theta) = \mathrm{e}^{-\mathrm{i}\frac{\theta}{2}P}$ with $P = X, Y, Z$ and the Hadamard-gate

$$\mathrm{Had} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \tag{1.2.5}$$

which coincides with the 2-dimensional discrete Fourier transformation. By the nature of measurement, as outlined below, quantum states are only distinguishable up to a global phase factor. However, local phase differences are relevant to computation, and are leveraged by 'controlled operations'. A local phase difference means that different components of a wavefunction inhibit relative phase differences. As an example, consider the controlled $Z$ or controlled-phase gates:

$$\mathrm{C}Z = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \mathrm{e}^{\mathrm{i}\pi} \end{pmatrix} = |0\rangle\langle 0| \otimes \mathbb{I} + |1\rangle\langle 1| \otimes Z. \tag{1.2.6}$$

If the first qubit is in state $|0\rangle$, the second remains unchanged; if the first is in $|1\rangle$, the second experiences a Pauli-$Z$ gate, i.e., a local $\mathrm{e}^{\mathrm{i}\pi} = -1$ phase difference is induced. Similarly, there is the controlled-NOT or CNOT gate, $\mathrm{CNOT} = |0\rangle\langle 0| \otimes \mathbb{I} + |1\rangle\langle 1| \otimes X$, where the second qubit experiences a bitflip on the part of the wavefunction where the first qubit is in the $|1\rangle$ state.

Furthermore, one can show that there are several discrete sets of gates that are 'universal'. That means they can achieve that they can approximate any unitary arbitrarily accurate, cf. Eq. (1.2.3).

One such gate set is formed by the Hadamard, the $T = \mathrm{e}^{-\mathrm{i}\frac{\pi}{8}Z}$ and the CNOT gate. Intuitively, Hadamard + T are able to generate any element of $U(2)$ as the angle coming from a single application of T is a transcendental number, i.e., there is no integer multiple of this angle that is a period, which would mean the set of attainable angles would be closed and would not allow arbitrary angles. Additionally, CNOT gates allow to 'extend' individual $U(2)$ approximations to arbitrary numbers of qubits.

So far, we left a very important question open: How do we extract information from a quantum computation? As it is a quantum system, we cannot 'look into the state' during the computation. Information is only accessible via measurement of observables. Observables are Hermitian operators acting on the system's state space (which is a Hilbert space). By that virtue, they enable an eigendecomposition into $\{\lambda, P_\lambda\}$, where $\lambda \in \mathbb{R}$ are eigenvalues of the observable and $P_\lambda$ projections onto the respective eigenspaces. Thus, we can express the expectation value of an observable $M$ with respect to a state $|\psi\rangle$ through projective measurement as

$$\langle\psi|M|\psi\rangle = \sum_\lambda \lambda \langle\psi|P_\lambda|\psi\rangle . \tag{1.2.7}$$

Such measurements are called projective measurements; a more general notion of measurement is given by positive-operator values measurements (POVM). We refer to the literature for details thereon, as projective measurements are sufficient for the considerations in this thesis. Further, note that typically this is not how measurement of an expectation value would be implemented in practice but is rather a theoretical tool.

### 1.2.2 Quantum Algorithms for Chemistry

The following paragraphs provide some high-level background of aspects from quantum chemistry in order to put the contributions of this thesis into context. After introducing the set of problems that is studied together with classical computational methods, we provide an overview of how they can be related to quantum methods for these problems. The goal here is to provide essential context and not to be comprehensive, as many other works have done so already; there are, among others, the reviews [Cao+19; McA+20; Dal+23; Cer+21; Bha+22; Til+22; Ana+22; Sch+25b].

#### Computational Quantum Chemistry and Electronic Structure

Fundamental questions in quantum chemistry revolve around the time-dependent Schrödinger equation,

$$\partial_t \psi(t, r) = -\mathrm{i} H \psi(t, r), \tag{1.2.8}$$

or the time-independent Schrödinger equation,

$$H\psi_n(r) = E_n \psi_n(r). \tag{1.2.9}$$

We can call the former a *simulation problem* where the task is to apply a unitary operation that signifies time-propagation induced by the Hamiltonian operator $H$, and the latter a *spectral problem* where we are interested in finding the eigenvalues $E_n$ and their associated eigenstates $\psi_n(r)$. For

molecules, the Hamiltonian in *first quantization* has typically the formulation

$$H = -\frac{1}{2}\sum_{k:\text{el}\vee\text{nuc}} \Delta_k - \sum_{k:\text{el},L:\text{nuc}} \frac{Z_L}{\|r_k - R_L\|} + \sum_{k\neq l:\text{el}} \frac{1}{\|r_k - r_l\|} + \sum_{K\neq L:\text{nuc}} \frac{Z_K Z_L}{\|R_K - R_L\|} \tag{1.2.10}$$

Being a quantum-mechanical observable, Hamiltonians are Hermitian ($H = H^\dagger$), thereby it is satisfied that their eigenvalues $E_n$ as energies (eigenenergies) are real-valued. In addition, molecular wavefunctions are typically real-valued as well as long as the Hamiltonian does not contain any explicit terms acting on spin degrees-of-freedom. The time-independent formulation is prominent in *electronic structure theory*. Of particular interest is often the *ground-state*, where the ground-state wavefunction $\psi_0$ is the wavefunction that attains the minimum eigenvalue $E_0$ among all 'physical wavefunctions'. This is defined via the variational principle

$$E_0 = \min_{\||\psi\rangle\|=1} \langle\psi|H|\psi\rangle. \tag{1.2.11}$$

Some additional assumptions are necessary here: Molecular systems consist of electrons and nuclei (protons, neutrons), sometimes it is convenient to also express e.g. photons, and these particles need to satisfy Fermionic and Bosonic statistics. Therefore, the Fermionic part of the wavefunction $\psi$ is required to be anti-symmetric with respect to exchanging two particles of the same type, and the Bosonic part is required to be symmetric. Broadly, that means that for two Fermions, $\psi_F(r_1, r_2) = -\psi_F(r_2, r_1)$. For two Bosons, $\psi_B(R_1, R_2) = \psi_B(R_2, R_1)$.

A simple functional form for antisymmetric functions is a determinant. Therefore, [Sla29; Hei26; Dir26] discuss the *Slater determinant* as a fundamental object in quantum chemistry. Exact wave functions are composed of a linear combination of Slater determinants. The Hartree-Fock model, which often serves as a starting point for many quantum chemical computations, relies on finding the Slater determinant that minimizes the energy [Har28; Foc30; Sla30b][Sla30a; Sla29]. This enables to introduce a concept called *second quantization*, where the exchange symmetry is embedded in the space the functions live in and in the rules about how they are acted on rather than in the function itself. More formally, the Fock space is defined as

$$\mathcal{F} = \bigoplus_{m=0}^{\infty} \mathcal{S}_\pm \left(\mathcal{H}_1\right)^{\otimes m}, \tag{1.2.12}$$

where $\mathcal{H}_1$ is a single-particle Hilbert space and $\mathcal{S}_\pm$ is a (anti)symmetrization operation. Then, naturally, (direct sums of) determinants formed by one-particle basis functions give a basis for this Fock space in the Fermionic case. This approach has become widely practical for quantum chemistry through the lens of second quantization, where a state in Fock space contains information about which basis element (which orbital spanning a one-particle Hilbert space) is 'used', i.e., is used in forming the respective Slater determinant. Then, operators in second quantization are expressed via the creation $a_p^\dagger$ and annihilation $a_p$ operators so that they satisfy certain anti-commutation rules that ensure exchange symmetry is not violated. The Hartree-Fock method above is able to give only one term of the Fock space expansion; it is also called a mean-field method, as the interactions for every single orbital are formed by the average field coming from the electrons at the other orbitals. In order to be more accurate, 'post Hartree-Fock' methods have been developed that add yield linear combinations of Slater determinants, thereby adding more correlation. We refer to the literature

for more details, such as the well-established text books [HJO00; SO67] or a recent primer by the author [Sch+25b].

The reason why second quantization gained a lot of practical attention compared to first-quantized approaches defined on real-space grids or a plane wave basis is that this approach is very amenable to using a well-informed one-particle basis through 'atomic orbitals'. The latter are functions that are designed to resemble the eigenfunctions of the Hydrogen atom which leads to a set of desirable properties like locality and asymptotical behaviour with respect to nuclei that approach each other. In this dissertation, we avoid re-stating the well-established definitions here as they are not essential to the central contributions, and again refer to the textbooks mentioned in the previous paragraph.

### Perspective for quantum algorithms in chemistry

Using classical hardware, quantum chemical problems that are represented via an exact model suffer from the curse of dimensionality – namely, an exponential increase of computational resources required when increasing the problem size. Hence, there is immediate intuition for using quantum computers. If the probabilistic quantum Monte Carlo method is used, the so called sign problem becomes a challenge. Also here, quantum computers seem to show promise [Hug+22]. Furthermore, approximate models were introduced, for instance for methods that optimize parametrized wavefunctions [HJO00], density functional theory [OPN09], and tensor-network based methods [Sch05]. These are all quite successful, however, there is plenty of room for improvement in high-accuracy regimes and in the representation of large systems and strong interactions where quantum computers can help. A set of quantum chemical methods along with the basic idea, drawbacks and opportunities for quantum computers in the context are summarized in Table 1.2. Based on the quantum

| **Class of Methods** | **Approach** | **Bottlenecks** | **Opportunities for Quantum Computation** |
|---|---|---|---|
| Wavefunction methods | parametrized wavefunction | dimensional cost when accuracy in model ↑ | Reduction of the dimensional cost |
| Density Functional Theory | exchange-correlation functional | finding good functionals | open question |
| Quantum Monte Carlo | parametrized wavefunction but sampling access | sign problem, sampling access | e.g. [Hug+22], quantum Metropolis algorithm[JI24] |
| Tensor-networks methods (DMRG etc.) | parametrized state via tensor network | operational cost ↑ with correlation ↑ | Complementary, e.g. [Ber+24] |

Table 1.2: Overview of computational methods for quantum chemistry

phase estimation algorithm that allows to extract eigenvalues of a unitary circuit [Kit95], [ADLH05] proposed to use this procedure to extract the ground-state energy based on a unitary circuit that represents the Hamiltonian simulation $\exp(-\mathrm{i}tH)$ of a quantum chemical Hamiltonian $H$. A key step here was proving that simulation of Hamiltonians of relevance is indeed efficient [Llo96; BACS07]. Algorithms that are based on general algorithmic Hamiltonian simulation, achievable through product formulas [Llo96; Chi+21], randomized product formulas [Cam19] or through 'qubitiziation' or quantum signal processing [LC16; Ber+19], typically fall into the category of fault-tolerant algorithms. A comprehensive overview for such a fault-tolerant algorithm is, e.g. in [Su+21]. A famous example for a class of near-term algorithms that do not require a fault-tolerant architecture are

variational algorithms like the variational quantum eigensolver [Per+14].

Considerations from complexity theory suggest that the ground-state problem will still be hard for quantum computers [KKR05], as determining the ground-state eigenvalue of a $k$-local Hamiltonian belongs to the complexity class Quantum Merlin Arthur, which we may see as quantum generalization to a randomized form of the famous class 'NP' (non-deterministic polynomial). This aligns with the study in [Lee+23] which points out a lack of efficient initial state preparation, whereas more recently, there has been more positive outlook in [Ber+24]. There, the authors claim very high overlap with the ground-state with an input state prepared by classical tensor network methods; however, the exact reference for certification is not available at the studied system size.

### 1.2.3 Quantum Algorithms for Differential Equations

The following section draws from Section VIII.A in [Mor+24], which was mostly written by the author of this thesis.

Similar to the problems outlined in quantum chemistry, the solution of differential equations is centred around stationary and evolutionary problems. The latter may look like

$$\partial_t u(t,x) - \mathcal{D}\left(u, \nabla u, \nabla^2 u, \ldots\right) = f(t,x). \tag{1.2.13}$$

with $u$ being the sought-after solution, $\mathcal{D}$ a differential operator and $f$ a source or forcing term, and $t$ a time parameter. For the purposes of this thesis, we assume that all differential equations in consideration are well-posed. Though, at this point, we have no further assumptions on $\mathcal{D}$, i.e., it may be unbounded and nonlinear. Stationary problems either follow from situations (like in quantum mechanics) when the time dependency can be decoupled from the differential operator $\mathcal{D}$, or when the problem admits a fixed-point. Then,

$$\mathcal{D}(u, \nabla u, \nabla^2 u, \ldots) = f, \tag{1.2.14}$$

follows from the $t \to \infty$ limit of Eq. (1.2.13), so that $\lim_{t\to\infty} \partial_t u(t,x) = 0$.

**Example** (Heat Equation)**.** *As an example for a linear PDE, we take the heat equation. Then, $\mathcal{D} = D\Delta$ is the Laplacian operator defined on a domain $\Omega \subseteq \mathbb{R}^d$ weighted with a diffusion coefficient $D > 0$, $u(t,x)$ is the temperature distribution at spatial coordinate $x \in \Omega$ and time $t > 0$. The problem can be defined as an evolution equation or as a stationary problem so that $u(x;t) \overset{t\to\infty}{\to} u_\infty(x)$ defined by Eq.* (1.2.14)*. The evolution equation as an initial value problem requires some initial values, also called initial data, at time $t = 0$ so that the evolved solution is defined. Here that would be some initial temperature distribution. Additionally, there may be boundary conditions so that the problem is well-defined in space; these can be Dirichlet conditions that restrict the temperature on $\Gamma_D \subset \mathbb{R}^d$, Neumann conditions that restrict the in/outflux of heat along the surface normal $\partial_\mathrm{n}$ of $\Gamma_N \subset \mathbb{R}^d$, or*

*a mix of these through Robin conditions. A full problem definition then looks like,*

$$\begin{aligned} \partial_t u(t,x) &= D\Delta u(t,x) + f(t,x), && x \in \Omega, t \geq 0 \\ u(t,x) &= g(t,x), && x \in \Gamma_D, t \geq 0 \\ \partial_n u(t,x) &= h(t,x), && x \in \Gamma_N, t \geq 0 \\ u(t=0,x) &= u(x), && x \in \Omega. \end{aligned} \tag{1.2.15}$$

**Example** (Schrödinger equation)**.** *The time-dependent Schrödinger equation from Eq.* (1.2.8)*, as discussed earlier in Section 1.2.2*

$$\partial_t \psi(t,r) = -\mathrm{i}H\psi(t,r). \tag{1.2.16}$$

*The Hamiltonian operator $H = -\frac{1}{2}\Delta + V$ is made up of the kinetic energy ($-\frac{1}{2}\Delta$) and potential energy ($V$); therefore, we can see this is a linear PDE.*

So far, the setting is nearly equivalent to the case of the heat equation (cf. example above). Therefore it might seem more natural to introduce the general case here first and then move onto the special case. However, the motivation here is the following: Natural operations on quantum computers are unitary, which means they are compliant with the Schrödinger equation. Realizing dynamics that differ from unitary thus may require additional treatment on quantum hardware, and may inherit some additional limitations, as previously described in [ALWZ25].

One requirement for solving differential equations, as on classical hardware, is to use a discretization technique to bring them to a form that allows digital processing. There is a wide range of methods for discretization. Examples are the finite difference method, which is likely the most simple method and what we use later on in quantum algorithms, the finite volume method and the discontinuous Galerkin method which are popular when solving hyberpolic PDEs that follow conservation laws, as well as finite element methods, wavelet discretizations, etc. We refer to the literature for more details, such as the following books, [But16; Ame14; EBY12; CKS12; DKO97]. The goal is to find an approximate $\bar{u}$ (for the sake of simplicity this may be either solution to a stationary or a evolutionary problem), so that there is $\varepsilon > 0$,

$$d(u, \bar{u}) \leq \varepsilon. \tag{1.2.17}$$

Suppose that $\mathbf{D}$ is a (possibly nonlinear) discretization of $\mathcal{D}$; then, we can find $\bar{u}$ by solving

$$\mathbf{D}[\bar{u}] = \bar{f} \text{ (stationary form)} \quad \text{or} \quad \frac{\mathrm{d}}{\mathrm{d}t}\bar{u} - \mathbf{D}[\bar{u}] = \bar{f} \text{ (evolutionary form).}$$

Based on the problem formulation, this means that stationary problems will lead to linear systems of equations (or a spectral problem) and evolutionary problems to the numerical solution of an ODE system. When solving this task with a quantum algorithm, s we seek to prepare a state $|\bar{u}\rangle$ such that

$$\| |u\rangle - |\bar{u}\rangle \| \leq \varepsilon. \tag{1.2.18}$$

As discussed prior, a practical algorithm would then require to extract information in form of the measurement of expectation values via an observable $M$, $\langle \bar{u}|M|\bar{u}\rangle$, so that this expectation value represents a meaningful quantity like the average energy, ..., of a part of the represented phenomenon.

For instance, [Bab+23] and [Liu+23] discuss average kinetic energy.

In the case of the heat equation, A simple example here could be the average heat flux across a part of the domain when discretizing the stationary heat equation. Though, we note that in particular the heat equation is not a candidate for an exponential quantum speedup as it can be reduced to a search problem that inhibits a lower bound (square-root speedup) [LMS22a].

A simple example here is to pick $M$ as a projection on a subset of the underlying domain, leading to an average solution across this subset. If the solution is encoded as a history state, one can also consider (weighted) time-averages. Such considerations are discussed e.g. in [Liu+23] and in form of the kinetic energy as observable in [Bab+23].

Before we continue with some more details, the notion of a block-encoding is defined as it will be ubiquitious in the differential equations algorithms.

**Definition 1.2.1** (Block-encoding)**.** *Let $A \in \mathbb{C}^{2^N \times 2^N}$ with $\alpha \geq \|A\|$ ('sub-normalization' or block-encoding factor). Then, $U$ is a $(\alpha, n_a, \varepsilon)$ block-encoding to $A$ if using $n_a$ ancillary qubits,*

$$\left\| (\langle 0|_{n_a} \otimes \mathbb{I}) U (|0\rangle_{n_a} \otimes \mathbb{I}) - \frac{1}{\alpha} A \right\| \leq \varepsilon. \tag{1.2.19}$$

*A block-encoding gives a probabilistic encoding of any possibly non-unitary matrix, where the success probability goes as $\alpha^{-1/2}$.*

**Stationary problems** The problem statement of stationary PDEs leads to a quite clear connection to a linear system problem after discretization. For elliptic PDEs such as the Poisson equation or the heat equation a set of quantum algorithms, based on finite difference or finite element discretization, has been proposed that relies on inverting a linear system [Cao+13; Mon16; CLO21]. We refer to [Mor+24] for a more detailed discussion.

**Evolutionary problems** The solution of evolutionary differential equations can be tackled via a a linear systems problem as well, such as done in [Ber14; BCOW17; CL20; BC24; Kro23]. There, the solution is encoded as a history state [Fey82], or 'Feynman clock',

$$|\bar{u}_{\text{hist}}\rangle = \frac{1}{\||\bar{u}_{\text{hist}}\rangle\|} \sum_{\tau=0}^{n_t-1} \left|\bar{u}\big(\tau \frac{\Delta t}{n_t}\big)\right\rangle \otimes |\tau\rangle \,. \tag{1.2.20}$$

Then, a matrix is defined so that it corresponds to a discretized time-evolution between time-steps $u(0), u(\Delta t), u(2\Delta t), \ldots$.

There are alternative approaches that carry out such a time-stepping 'in-place' rather than storing the entire history. While the qubit counts for the representation itself is lower here, dealing with non-unitary evolution is a challenge; to that end, [FLT23] make use of a technique called uniform singular value amplification to mitigate this effect.

More recently, several approaches have been introduced that map an underlying, non-unitary evolution to a combination of unitary evolutions, such as the Schrödingerization technique [JLY24] and the Linear Combination of Hamiltonian Simulation [ALL23; ACL23].

Typically, ODE algorithms rely on queries to a initial state preparation oracle that prepares $|\mathbf{u}_0\rangle$ and an implementation of the system matrix $\mathbf{D}$. Optimal scaling with respect to the initial state preparation is regarded to be the ratio of the norms of initial vs. final state, meaning, if

information/norm is 'lost', more repetitions are needed. The query complexity to an encoding of $\mathbf{D}$ is regarded to be optimal if it is linear in the encoding factor $\alpha(\mathbf{D})$, linear in the evolution time and logarithmic in the inverse error. Linear-systems based approaches have good complexity in the approximation error ($\alpha T \log \frac{1}{\varepsilon}$) but are have higher cost in terms of state-preparation (compared to competitive methods, this depends on $\alpha$ and $T$ as well). The time-marching approach of [FLT23] scales optimally with respect to the initial state preparation but not with respect to approximation error and time in the number of matrix queries. Methods that relate the evolution to Hamiltonian simulation are promising as they are near-optimal, i.e., optimal in the initial state preparation cost, however the error dependency in the query complexity to an encoding of $\mathbf{D}$ is at least $\log^2(\frac{1}{\varepsilon})$.

Exponential quantum advantage was achieved in [Bab+23] for the simulation of classical harmonic oscillators, which even though thoroughly analyzed stayed within a very 'protected' setting that can be directly mapped to unitary dynamics. The subject of study in [CJO19] is the wave equation, which as a problem that inherits interference effects is quite promising – and achieves a cubic speedup compared to classical.

**Remark** (Stationary vs. evolutionary solutions: Mixing times)**.** *When solving stationary problems, it is interesting to know whether the stationary formulation of the problem and associated computational procedures (eigenvalue problems, matrix inversion) are more efficient than simulating the evolutionary problem "long enough".*

*To that end, let us consider an example. Suppose we have a PDE*

$$\partial_t u(t,x) - \mathcal{D}u(t,x) = f(t,x,u(t,x)), \tag{1.2.21}$$

*so that it admits a unique stationary point $u_\infty = \lim_{t\to\infty} u(t,x)$. Then, the notion of a mixing time $t^\varepsilon_{\text{mix}}$ allows to quantify that at this mixing time, the solution has distance at most $\varepsilon$ from the true stationary state:*

$$t^\varepsilon_{\text{mix}} := \inf\{t > 0 \mid d(u_\infty, u(t)) < \varepsilon\}. \tag{1.2.22}$$

*That means that if we have an estimate for the mixing time, we can decide whether it is (asymptotically) more efficiently to solve the stationary problem or evolve at least until $t^\varepsilon_{\text{mix}}$.*

*In fact, time evolution that is not fast-forwardable [AA17; ALWZ25] has complexity $O(\alpha T \log(\frac{1}{\varepsilon})$ in terms of queries to the matrix oracle, whereas the stationary solution via a linear systems problem requires $O(\kappa \log(\frac{1}{\varepsilon}))$ [CABB23; Mor+24].*

*It is not necessarily easy to find such mixing times in general and it goes beyond the scope of this thesis to open this rabbithole. Therefore, we look again at the heat equation for a very simple example (the argument extends easily as well processes that can be easily related to Markov Chains, such as Fokker-Planck equations). Here, we can bound the mixing time by the inverse spectral gap [LP17]. Suppose the equation is discretized on a regular grid and there are periodic boundary conditions, then the spectral gap is equal to $\frac{1}{\lambda_{\min}}$, where $\lambda_{\min}$ is the smallest non-zero eigenvalue; generally, this depends on the specific domain and boundary conditions, which determines the corresponding graph). The block-encoding factor can be upper bounded by $\lambda_{\max}$, and hence $\alpha t^\varepsilon_{\text{mix}} \log(\frac{1}{\varepsilon}) \simeq \frac{\lambda_{\max}}{\lambda_{\min}} \log(\frac{1}{\varepsilon})$, where we can identify $\frac{\lambda_{\max}}{\lambda_{\min}} = \kappa$. We can conclude that asymptotically, the cost of determining the stationary state in such situations, from a dynamical or stationary problem perspective, scales equivalently. For future work, it would be interesting to look at this argument more thoroughly.*

*Note also that there is a direct relationship between evolution time and condition number for*

*approaches based on a history state representation using a linear systems solver [BC24]: the condition number is proportional to the number of time-steps weighted by the block-encoding factor at that time-step; therefore, one can bound this overall by* $\kappa \in O(\alpha T)$.

So far, this covers linear evolution equations. Nonlinear equations cannot be solved directly on a quantum computer and require a linearization method.

**Nonlinear Differential Equations** Nonlinear differential equations are problems that are classically hard to simulate, such as the famous Navier-Stokes equations which are part of the Millenium Prize Problems. Hence extensive efforts have been spent on developing algorithms for nonlinear DEs, as in the early work of [LO08] and more recently via Carleman linearization [Car32; Liu+21; Liu+23; Kro23; CSMB23; Pen+24] and homotopy perturbation [XWG21], which are both methods that map a nonlinear ODE to a large system of linear ODEs.

Popular equations of study come from fluids problems (see e.g. the perspective [SISS23]) or are of a reaction-diffusion type, such as considered in [Liu+21; Liu+23; Kro23; CSMB23],

$$\partial_t u(t,x) - D\Delta u(t,x) = f(t,x,u(t,x)), \tag{1.2.23}$$

where $D > 0$ is a diffusion constant and $\Delta$ the Laplacian operator. For the considered quantum algorithms, the function $f$ is typically replaced by a monomial so that Carleman linearization can be applied easily. In order for simulations using Carleman linearization to be stable, it is typically necessary that the nonlinearity is weak compared to dissipative components in the dynamics. This limits the applicability to strong linearity, though if these assumptions are fulfilled, the linearization error decays exponentially in a relevant truncation number[FP17; ASM21] (for details, see Section 3.1). Methods that mitigate the limitation on strong nonlinearities include the usage of different models, as the Lattice-Boltzmann-picture in [Li+25], or the analysis in [WWL25] that derive an alternative stability criterion based on eigenvalue gaps. More discussion on the limits of stability in the context of flow problems can be found in, e.g., [SISS23; Pen+24]. A more general discussion is also provided in [Lin+22].

Information-theoretic lower bounds additionally restrict the potential of quantum algorithms for nonlinear dynamics in general. E.g., in the context of turbulent and chaotic problems in [LENS24] as well as nonlinear unitary dynamics in [BW25], it was found that based on state-discrimination lower bounds, a quantum algorithm with, for the sake of simplicity, a nonlinearity quantified by $\beta > 0$ and evolution time $T$, must scale at least $\mathrm{e}^{\Omega(\beta T)}$ to prepare a state corresponding to time $T$. Therefore, in this setting, we may expect no more than polynomial speedups when using a quantum computer. The limitation here is similar to the one for dissipative systems, first discussed for the heat equation in [LMS22b] and in a more general way in [ALWZ25]. There, it it possible to circumvent the limitation by constructing smart observables rather than considering complexity with respect to the quantum state. Perhaps similar strategies can be found for the nonlinear case.

## 1.3 Discussion

### 1.3.1 What would be an ideal quantum algorithm?

There are a few requirements that we would like a quantum algorithm to satisfy; this list may not be comprehensive:

- Provable exponential (or high-order polynomial) quantum speedup. That is, the computational cost of running the quantum algorithm is (exponentially) lower than the one of the best known classical algorithm.
- End-to-end efficient complexity. That is, a guarantee that we can prepare the initialization of the computation efficiently and that the computation output can be obtained easily, while not compromising a speedup.
- Utility. There may be tasks that quantum computers can do much better than classical ones – take the 'supremacy' experiment in [Aru+19] performing random circuit sampling. However, going through the intense effort and cost of developing building quantum computers if there is no computational advantage that is useful to society may not be be broadly convincing, beyond academic interest.

One algorithm that satisfies these requirements is Shor's algorithm [Sho94]. For many routines that promise exponential speedup, such as Hamiltonian simulation or matrix inversion, it tends to be hard to show that the initial state preparation and solution extraction steps are efficient, too. Preparation of relevant initial states, instantiations of necessary oracles, and efficient extraction of information are one of the biggest bottlenecks for the efficiency of quantum algorithms. For the general case, these remain open problems. In the worst case, this comes at exponential time in the number of qubits; one can achieve linear time complexity if one uses an exponential amount of ancillary qubits [ZLY22], which seems equally impractical in general. Hence it is important to identify problem instances that admit a practical quantum speed-up – by theoretical guarantees or empirical validation.

### 1.3.2 What limitations are there to quantum computation?

Next, we discuss some limitations of quantum computation that are relevant to the problems studied in this dissertation. In particular, what we look at are *how fast and which systems can we simulate* and*how much information can we extract*, and discuss these limitations on some specific examples.

- **Simulation time.** Generally, we know that quantum computers can simulate quantum systems well, i.e., through circuit implementatons of $\exp(\mathrm{i}tH)$ with Hamiltonian $H$. However, not all possible Hamiltonians are efficiently simulatable – the restriction are so called 'row-computable' Hamiltonians, namely those where the nonzero entries can be efficiently retrieved. A relevant subclass of these is given by sparse Hamiltonians, such as $k$-local Hamiltonians $H = \sum_{l=1}^{L} H_l$ that often occur in physical problems, where each $H_l$ acts on at most $k$ sites. [BACS07][Ber+14] E.g., molecular Hamiltonians in quantum chemistry fall under this class. The time-evolution of general Hamiltonians that can be efficiently simulated has time complexity that is at least $\Omega(T)$ in evolution time $T > 0$. This result is also called 'no fast-forwarding'. [BACS07]

Certain types of operators however do allow 'fast-forwarding': Those where one a priori knows an efficient means of diagonalization [AA17; GSŞ21], or also, simulating dissipative dynamics with information about the stationary state [AOY24].

Solving a linear system of equations – differential equations can be discretized as such – must have complexity $\Omega(\kappa \log \frac{1}{\varepsilon})$ [HHL09; CABB23]. Suppose this represents an evolution ODE encoded as a history state [Fey82; Ber14]. Then, the condition number will be at least linear in the number of time steps (e.g., observed in [Ber14; BCOW17; BC24; Kro23]). Then, as the number of time steps is proportional to simulation time, the conclusion on overall complexity will be similar as for 'no fast-forwarding'.

So far, the above deliberations focussed on simulating unitary dynamics. Suppose we aim to prepare a state $|v(T)\rangle = U_{\exp(AT)} |v(0)\rangle$, where $U_{\exp(AT)}$ is a unitary implementation of time evolution up to time $T$ generated by a matrix $A$. As pointed out in [ALWZ25], 'the less unitary', quantified by the real parts of the eigenvalues of $A$, and 'the less normal' the matrix $A$ is, the more costly the simulation will be. An important tool that will also be mentioned below in the bullet point on information extraction is Holevo's theorem. In the simulation context, it allows to give lower bounds on the complexity of state discrimination; if we start with two very similar input states in terms of overlap, strongly dissipative dynamics would allow to exponentially reduce the overlap with simulation time. Thus, by the Holevo-Helstrøm bound, we know that then measurement would need to increase to match the lower bound. In implementations by block-encodings can show as a loss in success probability of measuring the sought-after solution. However, recall that we should view quantum algorithms not as a tool to measure the full state $v(T)$; choosing the right observable can mean that there is no such loss in success probability. E.g., [LMS22a] studied the heat equation, and their complexities that consider averages over the final state do not increase exponentially with time.

- **Information extraction.** A general aspect to keep in mind when designing quantum algorithms is that the quantum computer's state represents a distribution and we can extract information only by sampling from it, via measurement. On one hand, this introduces the necessity for efficient input state preparation and representing the sought-after information reasonably. On the other hand, there is a physical limitation to measurement: the Heisenberg limit, which follows from the uncertainty relation. Roughly speaking, information that is distributed across $N$ qubits requires a measurement effort linear in $N$.

  While the Heisenberg limit is a physical limitation, it is also possible to find an upper bound on the accessible information via Holevo's theorem from information theory. The overall conclusion is similar: Even though $N$ qubits can represent exponentially larger information, information of at most size $N$ can be accessed by a measurement. Measuring expectation values of general, sparse-access observables on a state of $N$ qubits requires $\Omega(N)$ measurements. This was shown in [Ala+22], where the authors also look specifically at states that come from the solution of linear systems of equations.

  E.g., if one estimates eigenenergies of chemical systems, the energy-time uncertainty relation is what limits the accuracy that can be attained in extracting eigenvalues – $N$ qubits lead to $\varepsilon \sim \frac{1}{N}$ error, so the effort goes as $\frac{1}{\varepsilon}$ [ADLH05; LT22].

At this point, we return to the main topic of this thesis: quantum algorithms for problems of practical interest, more specifically, for applications in chemistry and for the solution of differential equations.

### 1.3.3 The problems addressed in this thesis

Work in this thesis will fit into the context in the following way.

**Quantum Chemistry** Near-term quantum algorithms are limited by the amount of available quantum resources. Therefore, in Section 2.1, we are looking at a near-term quantum algorithm that has the goal to speed up early adaption of hardware by using classical computational resources to distribute the quantum computation and account for the thereby induced loss of correlation. In Section 2.2, we provide a dynamics-based state preparation scheme. The central idea is to use that dynamical simulation can be done efficiently, and we aims to circumvent the hardness of the ground-state problem using a heuristic state-preparation scheme.

**Differential Equations** Nonlinear differential equations are problems that are classically hard to simulate, such as the famous Navier-Stokes equations which are part of the Millenium Prize Problems. Hence extensive efforts have been spent on developing algorithms for nonlinear DEs, as in the early work of [LO08] and more recently via Carleman linearization [Car32; Liu+21; Liu+23; Kro23; CSMB23; Pen+24] and homotopy perturbation [XWG21], which are both methods that map a nonlinear ODE to a large system of linear ODEs. There are limitations to these methods though, see e.g. [LENS24], which typically show through a requirement that the strength of dissipation in the system is at least as large as the strength of nonlinearity. This thesis contributes to these efforts with the algorithm presented in Section 3.1. Beyond this, we consider a physics-inspired technique to enhance a quantum algorithm in Section 3.2: Inspired by concepts such as complex absorbing potentials and the quantum Zeno effect, we study the implementation of constraints and boundary conditions when solving differential equations on a quantum computer.

# Chapter 2

# Simulating Quantum Systems with Quantum Computers

Quantum algorithms can, roughly, be classified into two approaches. Those that aim to provide an early demonstration of computations on noisy, currently- or soon-to-be-available hardware, usually referred to as 'NISQ' (noisy-intermediate-scale quantum) [Pre18]. And fault-tolerant algorithms, that assume an abundance in available qubits, no limits on qubit-connectivity and the ability to correct for errors.

The first work presented in Section 2.1 targets early adoption on NISQ hardware; in fact, the goal is to 'split up' quantum computations by adding only classical computation. Moving on, we take a further look into the future, relying on fault-tolerant quantum algorithms. Section 2.2 introduces a heuristical simulation scheme that uses a physically-motivated scattering approach as dynamics-based state preparation with the intent to avoid explicit ground-state preparation – a notoriously hard problem, both quantumly and classically – when performing computations where the desired output comes from dynamical simulation.

## 2.1 Partitioning Quantum Chemistry Simulations with Clifford Circuits

This section originally appeared as:

Philipp Schleich, Joseph Boen, Lukasz Cincio, Abhinav Anand, Jakob S Kottmann, Sergei Tretiak, Pavel A Dub, and Alán Aspuru-Guzik. "Partitioning quantum chemistry simulations with clifford circuits". In: *Journal of Chemical Theory and Computation* 19.15 (2023), pp. 4952–4964.

### 2.1.1 Introduction

An important goal of computational chemistry is to understand the behavior of strongly correlated electronic systems, as they are key to many applications in materials science. Examples hereof are the development of more efficient catalysts, drugs or high-temperature superconductors. Since the computational complexity of many existing classical, exact numerical methods scales exponentially with system size owing to solving the Schrödinger equation, there has been increased interest in developing quantum algorithms for this task in recent years [ADLH05; Cao+19; McA+20]. One of the most popular approaches for finding the ground state [Per+14] and several low-lying excited states [HWB19; NMF19; MKCD17; Ast+23] is the Variational Quantum Eigensolver (VQE) [Cer+21; Til+22; Bha+22].

The VQE is a hybrid quantum-classical algorithm that utilizes the variational principle to find variational approximations to the eigenstates and associated eigenvalues of a molecular Hamiltonian. Given a molecular Hamiltonian encoded as a linear combination of tensor products of Pauli operators, a parameterized quantum circuit (PQC) is used to prepare a trial wavefunction (ansatz) and compute its energy. By optimizing the parameters of the quantum circuit, approximations to the ground and also excited states and their energies can be obtained. The VQE has been successfully applied to simulations of small molecules on quantum hardware [Kan+17]. One strategy to find trial wavefunctions is given by chemically-inspired ansatze, such as the popular family of Unitary Coupled Cluster (UCC) ansatze [Per+14; Ana+22]. UCC utilizes domain knowledge from quantum chemistry to prepare trial wavefunctions by exciting electrons from a reference state, which is usually the Hartree-Fock state. In contrast, hardware-efficient ansatze [Kan+17; OBRT20] construct circuits from a limited set of gates that exploit the characteristics of the underlying device, such as connectivity or coherence time. Instead of directly encoding chemical knowledge into the ansatz, hardware-efficient ansatze rather try to be maximally expressible given hardware constraints. Despite continuous development of quantum hardware, current devices are prone to significant errors and can only support a limited number of operations on a few qubits, which limits the complexity of the quantum circuits used to generate trial wavefunctions. Thus, the development of techniques for designing shallow and resource-efficient ansatze is not only academically significant, but also of considerable practical importance. High gate errors and low coherence times of devices that are available in the current and near future – so called noisy intermediate-scale quantum computers (NISQ) [Pre18] – pose a critical limit to the circuit depth. As a result, many techniques have been developed to find suitable shallow circuits, see [Bha+22; Cer+21; Ana+22] for a representative

overview.

In this work, we attempt to clusterize circuits to facilitate the usage of existing NISQ devices to run larger experiments. There has been recent work in this direction, as in [Zha+22; Edd+22]. Both studies attempt to discover clusters or reduce the number of qubits per circuit through the utilization of correlation metrics. While [Zha+22] makes use of mutual information to clusterize the circuit, [Edd+22] uses entanglement forging to divide the circuit into two pieces that are executed separately. In our method, we pre-define the clusters and generally "cut" the circuit into two equally large pieces. We consider two options in choosing the clusters. For a physically motivated choice, we use the separated-pair-ansatz from [KA22], which is classically tractable and provides a natural clustering into electron-pairs, made up of pair-natural orbitals [KSTA21; KBV20]. Beyond that, we also consider arbitrary clusters. Here, we expect that the sub-optimal choice of clusters can be accounted for by a Clifford circuit since permutation of qubits can be realized as a sequence of Clifford gates. Then, to re-gain lost correlation by assuming the product state, we search for a re-entangling circuit.In order to enable that the quantum computation can be carried out on hardware with a lower number of qubits, we use this circuit to similarity-transform the system Hamiltonian. In the literature[RLGI20], this transform is often called "folding" or "dressing" the Hamiltonian. This procedure then is similar to the virtual Heisenberg circuits from [Sha+21] or [Zha+22]. The latter optimizes, beyond the quantum and the virtual circuit, a classical neural network to enable error mitigation.The circuit that modifies the Hamiltonian is trained together with VQE circuits, minimizing the overall energy. This means that the (near-)Clifford circuit modifies the Hamiltonian so that its ground state is closer to a product state. This procedure is briefly outlined in Fig. 2.1. We emphasize here, that both the cluster circuits as well as the virtual circuits are to be optimized. This means that for each modification of the virtual circuit, the parameters in the cluster circuits will be adjusted.

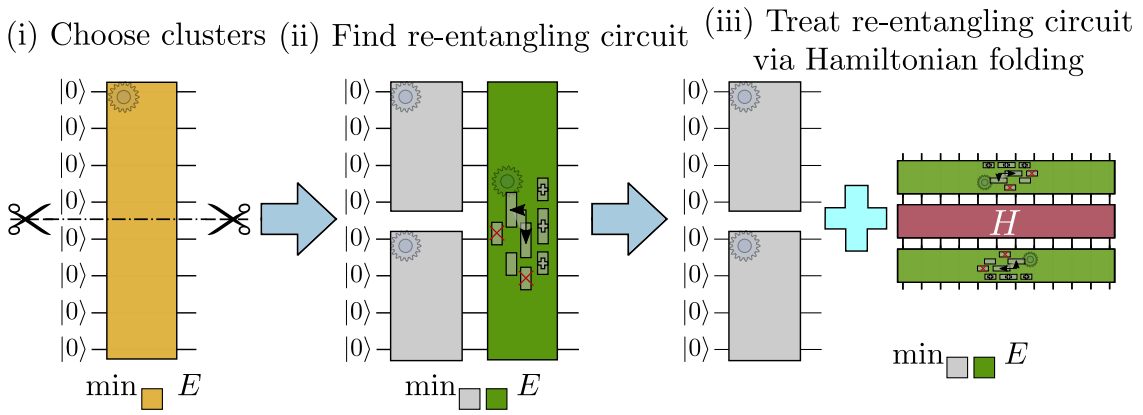



Figure 2.1: Overview of our proposed method of dividing large system into smaller pieces. The ansatz consists of two parts: (i) VQE-like circuits denoted by gray boxes and with continuous, variational parameters (ii) near-Clifford, classically simulable circuit (green box) whose architecture is to be found by the optimizer as well. Near-Clifford circuits are evaluated by modifying (or dressing) the Hamiltonian $H$.

For the re-entangling circuits, we will first focus on using Clifford circuits only. The latter describe sequences of unitaries composed of elements of the Clifford group. Thus, they leave the number of Pauli strings when transforming a Hamiltonian in the Pauli basis invariant. As a con-

sequence, the measurement complexity does not increase as opposed to folding general gates into the Hamiltonian. This will be extended by near-Clifford circuits, which introduce a small number of non-Clifford operations into these gate sequences so that the expressibility is increased but the operator complexity does not increase vastly. It is known that Clifford circuits are efficiently simulatable classically but do not form a universal set of quantum gates [Got98b; Got98a; BK05]. There have been recent efforts for the usage of Clifford circuits on early-stage quantum computers, such as the development of initialization strategies to find better initial circuits and guesses for gate parameters based on Clifford gates only [Mit+22; Che+22]. Other areas of application include classically efficient compression of quantum circuits [Ana+22] and a standalone ansatz for variational quantum algorithms [Rav+22]. Such a parametrization however is not able to reproduce the results obtained with universal gate sets due to limited expressibility of the Clifford group. In this study, we are exploring the limit to which classical post-processing (mostly relying on Clifford gates) can reduce the resource requirements in quantum chemistry calculations. Since Clifford gates form a discrete set, the optimization requires specialized techniques, as described in Section 2.1.2.

We start this work by outlining our proposed methodology in Section 2.1.2, including a motivation for the choice of clusters, looking at how to fold important circuit identities and the way Clifford and near-Clifford circuits are constructed. Then, we show results in Section 2.1.3 for simulating the ground-states of several molecules ($H_2$, $N_2$ and $BeH_2$) before concluding our work in Section 2.1.4.

### 2.1.2 Methodology

#### Outline

In a VQE calculation, one aims to find an approximation $E$ to the ground-state energy $E_0$

$$E_0 \leq E = \min_{\psi} \langle\psi|H|\psi\rangle \,. \tag{2.1.1}$$

The state $|\psi\rangle$ is usually prepared and optimized using a parametrized unitary operation $|\psi(\theta)\rangle = U(\theta)\,|0\rangle$, where $U(\theta)$ acts on $N_\mathrm{q}$ qubits.

In this work, we prepare the wavefunction $|\psi(\theta)\rangle$ through unitaries $U_j(\theta_j)$ that only have local support on qubit-cluster $j$ followed by a global operation $M(\tau)$ that is built using a (near-)Clifford circuit, which add some additional variational parameters $\tau$:

$$|\psi(\theta,\tau)\rangle = M(\tau) \prod_{j=1}^{N_\mathrm{cl}} \underbrace{U_j(\theta_j)\,|0\rangle}_{=:|\psi_j(\theta_j)\rangle} . \tag{2.1.2}$$

Here, $M(\tau)$ acts on the full space over $N_\mathrm{q}$ qubits, while $U_j$ act only on the respective clusters $j = 1, \ldots, N_\mathrm{cl}$ with ${N_\mathrm{q}}^{[j]}$ qubits. This way, we have a set of $N_\mathrm{cl}$ factorized states that can be evaluated independently and are then re-combined with $M(\tau)$. This circuit $M(\tau)$ will be folded into the Hamiltonian. Our method can be summarized as a Schrödinger evolution on a subsystems with a virtual Heisenberg circuit that alters the Hamiltonian so that its ground-state resembles a product state over the subsystems. We provide an outline of the procedure in comparison with the standard, Schrödinger-picture-like VQE, in Fig. 2.2.

In what follows, we will first discuss how one can choose the subsystems in Section 2.1.2. Then, as the main contribution of this work, we will outline our procedure in Section 2.1.2 to find the

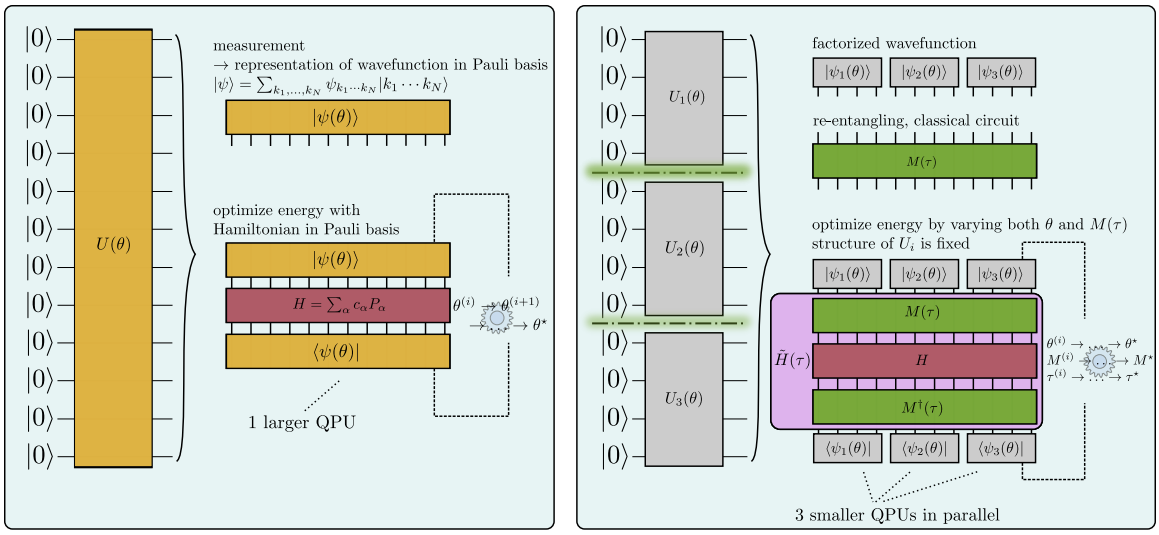



Figure 2.2: Comparison of standard VQE procedure (left) and our approach (right), depicted using an example of three clusters.

virtual circuit $M(\tau)$ so that one regains maximum correlation energy.

### Clustering into subsystems

We choose subsystems either in a naive, unformed manner or guided by physical intuition.

In the naive approach, we set clusters by simply cutting the circuit in $N_{\text{cl}}$ roughly equally large clusters without considering any additional information. This allows us to test the capabilities of the re-entangling circuit $M(\tau)$ to correct for shortcomings of the naive clustering compared to the physically motivated approach and experiment with more arbitrary clusterings - for example, within electron pairs. In particular, we will see that the inclusion of permutations [Tka+21] into the ansatz allows the approach to retrieve a more optimal clustering.

For the physically motivated splitting, we use the separated-pair ansatz (SPA) [KA22] with fixed pair-natural orbitals – i.e., we are not considering orbital-optimization as in [Kot22]. The SPA ansatz admits a natural decomposition into factorized clusters made up by $N_{\text{el}}/2$ electron pairs, where the full wavefunction reads (cp. Eq. (15) in [KA22])

$$|\psi_{\text{SPA}}\rangle = \prod_{j=1}^{N_{\text{el}}/2} U_j(\theta_j) U_{\text{HF}} \, |00\cdots0\rangle \ , \tag{2.1.3}$$

with pair-specific UCC-circuits $U_j$, number of electrons $N_{\text{el}}$ and a circuit to prepare the Hartree-Fock state $U_{\text{HF}}$.

The reasoning behind our method is that we, on one hand, would like to push the boundaries of the classically efficient SPA-ansatz. On the other hand, we also expect a self-assembly of clusters throughout our technique. The latter can be understood in the sense of the so-called permVQE [Tka+21], where qubits (or conversely, orbitals) are permuted in order to maximize mutual information leading to a reduction in circuit depth. Since permutations are Clifford operations, including them in the allowed set of operators for $M(\tau)$ does not increase complexity of the Hamiltonian. This means that the sub-optimal choice of clusters here is correctable by the virtual circuit

$M(\tau)$.

As we will discuss later in Section 2.1.2, any modification of the Hamiltonian beyond Clifford gates may increase its complexity in form of the number of independent terms in the Pauli basis significantly. While it is in general beneficial to use measurement reduction techniques to lighten the measurement load, it can particularly help here when non-Clifford gates in $M(\tau)$ increase the measurement complexity. These procedures typically effectuate a grouping of the Hamiltonian terms by various means so that multiple terms can be measured simultaneously [IYLV19; YVI20; VYI20; BBO20; YI21; Hug+21]. At this point, we note that most of the techniques find a unitary operation that is applied directly before measurement so that the grouping of terms can be achieved in a desired basis. This is the case in particular for the methodologies outlined in [YI21; Hug+21] that suppress the cost to at most $\mathcal{O}(N^2)$ measurements. Unfortunately, this unitary gate needs to be evaluated on the quantum hardware with support beyond clusters for the method to be effective, which is impractical when dividing the circuit into clusters with the objective to run each cluster on hardware independently. However, one can still make advantage of graph clique cover methods such as [VYI20] and especially the approach proposed in [BBO20], which can also compress the measurements into a number of partitions that is at best $O\left(N^2\right)$.

**Recovering correlation by folding a Heisenberg-circuit**

Here, we describe our approach to construct the re-entangling circuit $M(\tau)$ as in Eq. (2.1.2), carried out as a virtual Heisenberg circuit in the spirit of [Zha+22; Sha+21; Zha+21]. We first start by taking a closer look at folding typical circuit building blocks, not necessarily Clifford, in a generator formalism before introducing a procedure to build $M(\tau)$. Hereafter, we use the expression "to fold" a unitary circuit $U$ into a Hermitian operator $H$ to indicate a mapping according to a similarity transformation $H \xrightarrow{U} U^\dagger H U$. We will be using either purely-Clifford or near-Clifford circuits, which contain a small number of non-Clifford, parametrized gates to increase expressibility [SJA19] of the circuit.

**Folding of important circuit identities** We aim to approximate the ground-state energy of an electronic system using an ansatz specified in Eq. (2.1.2) with a virtual circuit $M(\tau)$, given a clustering ansatz $\prod_j U_j(\theta)$. The resulting cost function, as in Eq. (2.1.1), then takes the form

$$\langle 0|(\prod_{j=N_{\mathrm{cl}}}^{1} U_j^\dagger(\theta))\underbrace{M(\tau)^\dagger H M(\tau)}_{=:\tilde{H}(\tau)}(\prod_{j=1}^{N_{\mathrm{cl}}} U_j(\theta))|0\rangle\,. \tag{2.1.4}$$

We consider virtual circuits that are built by a sequence of individual unitary gates $M(\tau) = \prod_{l=1}^{N_M} V_l(\tau_l)$, where individual operations can be expressed as

$$V_l(\tau_l) = \exp\left(-\mathrm{i}\frac{\tau_l}{2} G_l\right) \tag{2.1.5}$$

for Hermitian generators $G_l = G_l^\dagger$. This allows us to implement the mapping $H \xrightarrow{M(\tau)} \tilde{H}(\tau)$ by sequentially folding individual operations $V_l(\tau_l)$.

Since folding operations to find $V_l^\dagger H V_l$ are an important part of our method (cp. to Fig. 2.2), we first look at efficient ways to find analytical implementations, using different classes of unitary gates.

This encompasses recent work [KAA21; ILY21; WIWL22; KE21] related to analytical gradients for parametrized quantum circuits using the parameter-shift rule [Sch+19], where in this work, we resort to [ILY21; KAA21] in particular. We first investigate rotation-like gates, whose generators are involutions, and gates that are generated by projections. Both have in common that their generators have exactly two eigenvalues, while the latter can be reformulated as the former by modifications so that their spectrum is symmetric around zero. Beyond that, we discuss excitation-like gates, whose spectrum differs from involutions by an additional zero-eigenvalue, and discuss a strategy for general gates that is inspired by [ILY21].

**Rotation-like gates.** For rotation-like gates such as $R_X, R_Y, R_Z$, the generators are involutions $G^2 = \mathbb{I}$. Hence we can write $V(\tau) = \cos\big(\frac{\tau}{2}\big)\,\mathbb{I} - \mathrm{i}\sin\big(\frac{\tau}{2}\big)G$. This gives us the identity

$$V^\dagger(\tau)HV(\tau) = \cos^2\left(\frac{\tau}{2}\right)H + \sin^2\left(\frac{\tau}{2}\right)GHG - \mathrm{i}\cos\left(\frac{\tau}{2}\right)\sin\left(\frac{\tau}{2}\right)[H,G]. \tag{2.1.6}$$

**Projector-generated gates.** Another important class of gates is generated by projection operators, i.e. $G^2 = G$. We can motivate this class in the following way: If one tries to generate an $X$-gate by using the rotation $R_X(\pi) = \exp\big(-\mathrm{i}\frac{\pi}{2}X\big)$, one obtains $-\mathrm{i}X$. A phase-true $X$ is generated by the projection $G = \frac{1}{2}(\mathbb{I} - X)$ and $\tau = 2\pi$, i.e. $X = \exp\big(-\mathrm{i}\frac{\pi}{2}(\mathbb{I} - X)\big)$. One can thus generally relate this to the class of rotation-like gates from the above section with self-inverse generators $\tilde{G}$ and associated angles $\tilde{\tau}$ by setting $\tau = 2\tilde{\tau}$ and $G = \frac{1}{2}(\mathbb{I} - \tilde{G})$:

$$V(\tau) = \exp\left(-\mathrm{i}\frac{\tau}{2}G\right) \equiv \exp\left(-\mathrm{i}\frac{\tilde{\tau}}{2}\tilde{G}\right). \tag{2.1.7}$$

This means that projector-generated gates are equivalent up to a $U(1)$ transformation to rotation-like gates since $\tilde{G}^2 = \mathbb{I}$. We still deem it worthy to mention them as a separate class to emphasize that a phase-true implementation of certain gates requires a projection as generator, which is of importance for controlled versions of these gates. Hence upon having done this substitution, the folding can be performed just as in Eq. (2.1.6).

**Excitation-like gates.** For quantum chemistry applications, circuit elements corresponding to fermionic excitations are of particular interest as they conserve particle-number symmetry. Beyond that, qubit excitations [RYGI18; RLGI20; XK20; Ana+22] or (multi)-controlled rotations belong to this class. These gates belong to a class of operations that are generated by $G = \mathbb{P}_+ - \mathbb{P}_- + \mathbb{P}_0$, projecting on eigenstates with symmetric eigenvalues $\pm\lambda$ and 0. One may express the generated unitary operator as [KAA21]

$$V(\tau) = \cos\left(\frac{\tau}{2}\right)\mathbb{I} - \mathrm{i}\sin\left(\frac{\tau}{2}\right)G + \left(1 - \cos\left(\frac{\tau}{2}\right)\right)\mathbb{P}_0. \tag{2.1.8}$$

The associated folded Hamiltonian is then

$$V^\dagger(\tau)HV(\tau) = \cos^2\left(\frac{\tau}{2}\right)H + \sin^2\left(\frac{\tau}{2}\right)GHG \tag{2.1.9}$$

$$+ \left((1-\cos\left(\frac{\tau}{2}\right)\right)^2 \mathbb{P}_0 H \mathbb{P}_0$$

$$- \mathrm{i}\cos\left(\frac{\tau}{2}\right)\sin\left(\frac{\tau}{2}\right)[H,G] \tag{2.1.10}$$

$$+ \cos\left(\frac{\tau}{2}\right)\left(1-\cos\left(\frac{\tau}{2}\right)\right)\{H,\mathbb{P}_0\}$$

$$+ \mathrm{i}\sin\left(\frac{\tau}{2}\right)\left(1-\cos\left(\frac{\tau}{2}\right)\right)(GH\mathbb{P}_0 - \mathbb{P}_0 HG)\,. \tag{2.1.11}$$

**General gates.** To obtain an exact expression for general gates, one can use the spectral decomposition of Hermitian generators $G = \sum_k \lambda_k \mathbb{P}_k$, with $\lambda_k$ eigenvalues and $\mathbb{P}_k$ projectors on the respective eigenspaces and then proceed similarly as for excitation-like gates, e.g. following the procedure outlined in section B.2. of [ILY21]. If the dimensionality of the generator prohibits a numerical eigendecomposition, the gate may be decomposed into lower-dimensional building blocks.

**Hamiltonian-dependent identities.** Whenever the transformed Hamiltonian has specific properties they might be exploited for similarity transformations with specific gates. A prominent example relevant in the context of this work are fermionic single-excitations with $G_{pq} = -i\kappa_{pq}a_p^\dagger a_q$ and anti-Hermitian matrix $\kappa$ in combination with Hamiltonians consisting of fermionic annihilation (creation) operators $a_p$ ($a_p^\dagger$) only. Here, the operators are transformed as $\tilde{a}_p^\dagger \equiv e^{iG}a_p e^{-iG} = \sum_q R_{pq}a_q^\dagger$ with the unitary matrix $R = e^{-\kappa}$. See the appendices of [Sok+20; Kot22] for details and applications, as well as [Miz+20; Yal+21; KA22].

**Clifford circuits** We now have a recipe for folding different kinds of circuits. Next, we explore the types of circuits that facilitate the recovery of correlation energy lost through the clustering of states. Using quantum-chemical encoding (see e.g. reviews [Cao+19; McA+20]), a molecular Hamiltonian can be written as a linear combination of Pauli strings

$$H = \sum_{\boldsymbol{\alpha}} c_{\boldsymbol{\alpha}} P_{\boldsymbol{\alpha}} = \sum_{\boldsymbol{\alpha}} c_{\boldsymbol{\alpha}} \sigma_{\alpha_1} \otimes \cdots \otimes \sigma_{\alpha_{N_q}}\ , \tag{2.1.12}$$

where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_{N_q})$ and $\sigma_{\alpha_j} \in \{\mathbb{I}, X, Y, Z\}$ is a Pauli matrix acting on site $\alpha_j$. We transform the original Hamiltonian as described above in Section 2.1.2 and measure expectation value of

$$\tilde{H} = M(\tau)^\dagger H M(\tau) = \sum_{\boldsymbol{\alpha}} c_{\boldsymbol{\alpha}} M(\tau)^\dagger P_{\boldsymbol{\alpha}} M(\tau). \tag{2.1.13}$$

The complexity of the measurement depends on the cardinality of the transformed Hamiltonian, i.e., the number of distinct Pauli strings in $\tilde{H}$ that require independent measurement.

At this point, we notice that the cardinality of the transformed Hamiltonian can significantly grow, depending on the structure of $M(\tau)$. For single-qubit gates $M(\tau) \in SU(2)$, one can write

$$M(\tau) = \exp(\mathrm{i}\tau\vec{n}\cdot\vec{\sigma}) = \cos(\tau)\,\mathbb{I} + \mathrm{i}\sin(\tau)\vec{n}\cdot\vec{\sigma}, \tag{2.1.14}$$

where $\|\vec{n}\| = 1$, $\vec{\sigma} = (X, Y, Z)$. We further note that the Pauli operators (together with phase

factors, see Eq. (2.1.15)) form a group. On one qubit, this means that due to the closure property, transformation of a Pauli operator by a single-qubit operation as in Eq. (2.1.14) can only produce terms with $\{\mathbb{I}, X, Y, Z\}$, up to the phase factors. Thus for each single-qubit operator that transforms a Pauli-string in the Hamiltonian as in Eq. (2.1.12), the number of terms grows at most by a factor of 4. Then, for a system of $N_\mathrm{q}$ qubits, a virtual circuit consisting of only single-qubit gates leads to a cardinality of at most $|\tilde{H}| \leq O\left(4^{N_\mathrm{q}}|H|\right)$ [Bia21]. This bound holds as well for multi-qubit gates.

Thus, dressing the Hamiltonian with general gates is very costly and should be avoided. As an example, we show the increase in number of terms for folding a UCCSD circuit into the Hamiltonians for $H_2$, $BeH_2$ and $N_2$ in Fig. 2.7a. The cost-effective solution is given by gates from the Clifford group $\mathcal{C}_n$, which is defined as the subgroup of the unitary group that normalizes the Pauli group. That is, the set of unitary operations that keep the Pauli group

$$\Pi_n = \left\{ e^{\mathrm{i}\theta\frac{\pi}{2}}\sigma_0 \otimes \cdots \otimes \sigma_n : \ \theta \in \{0,1,2,3\}, \ \sigma_j \in \{\mathbb{I}, X, Y, Z\} \right\} \tag{2.1.15}$$

invariant under similarity transformations [Tol18], i.e.

$$\mathcal{C}_n = \{U : U^\dagger \sigma U = \sigma'; \sigma, \sigma' \in \Pi_n\}. \tag{2.1.16}$$

Hence for $M(\tau) \in \mathcal{C}_n$, the cardinality of a Hamiltonian over $n$ qubits remains unchanged.

However, for Clifford gates only, we just introduce a certain, restricted type of entanglement between the clusters as Clifford circuits are not universal [Got98b; JN13]. This might already prove useful, but ultimately we considering *near-Clifford* circuits, so that we can control the additional cost and take benefit from more general transformations. Such circuits are composed mostly out of Clifford gates, however, also contain a limited number of non-Clifford gates in order to boost expressiblity.

[LRI20] also has investigated the growth of Hamiltonian terms for certain non-Clifford circuits as the qubit coupled cluster (QCC) approach can be formulated as a folding of the Hamiltonian. They found that, while exponential increase cannot be avoided, the coefficient $c$ for an increase that goes as $c^N$, is smaller for so-called involutory linear combinations of entanglers that lead to a involution as opposed to standard QCC entanglers. More recently, [LGI22] extended this work by exploiting the structure of the Clifford group to build a Hamiltonian that is folded by so-called normalizer Pauli strings rather than qubit excitations. Within their approach, they circumvent the complications that come with a discrete optimization problem, however, they do not address a Clifford-only circuit, which is the main focus of this work.

### Optimizing Clifford circuits using a genetic algorithm

In this work, optimization refers to the process of minimizing the energy expectation value by optimizing the parameters in the circuits in the following sense: For the clusters $U_j(\theta)$ with fixed architecture, the parameters $\theta$ are varied. At the same time, both structure and parameters need to be adjusted for the near-Clifford circuits, which are supposed to account lost correlation due to the enforced product ansatz in the $U_j$'s. For the case when $M \in \mathcal{C}_{N_\mathrm{q}}$, there are no continuous parameters available that would allow an adaptive scheme such as in the family of ADAPT-VQE

methods [GEBM19; Tan+21] and qubit coupled cluster techniques [RYGI18; RLGI20] as the Clifford group only forms a discrete set. One could think of Bayesian mappings from the discrete gate set to a continuous parameter such as in [BMTW22] or a simple simulated annealing strategy as in [Rav+22]. In our tests, a simulated annealing strategy was surpassed by a genetic algorithm, which is what we use and will describe in the following.

The genetic algorithm we choose resorts to an operator pool, which it uses as a reference set for the discrete optimization problem. Elements of the operator pool are used to modify the circuit by adding new operators and modifying existing gates in $M(\tau)$.

The operator pool we use can be found in Table 2.1. Aside from the generators of the Clifford group $(H, S, C_X)$, we also include the Pauli gates, controlled Pauli gates, a SWAP gate to mimic permutations. These have proven helpful in [Tka+21] to reorganize the correlation pattern in the state, so it is easier to describe with a cluster product ansatz. We also include gate blueprints that are inspired by fermionic excitations [Ana+22], where we fix the angle of the freely parametrized $R_Z$-rotation in the center aiming to obtain an approximate representation in form of a Clifford gate so that it is either one of $S, Z, S^\dagger$. These are supposed to mimic gate sequences that have proven useful in finding ground-states for quantum chemistry. Extending the gate pool may aid to speed up optimization, as these sequences can be picked from the pool and do not need to be found via optimization. Next, we explain the optimization procedure to find a Clifford circuit $M$, which is also described in algorithm 1.

The general procedure to generate the classical circuit is outlined in Fig. 2.3. First, we generate a set of initial populations by randomly sampling circuits built from the operator pool. Any initial gates that do not have support over multiple clusters can be seen as an extension/modification of the cluster circuits; thus, one can choose to avoid them in the optimization procedure. We discuss this further in Section 2.1.3.

Based on the initial circuits per population, we impose a set of modifications, drawn randomly from the set of allowed operations. These are comprised of the addition of gates from the pool, exchanging specific gates with others or re-arranging them and deleting gates. We choose a scheduling so that each modification occurs with a certain probability; e.g., we found that favoring changing over deletion and addition is advantageous, while during early iterations the rate of additions should initially be higher.

Then, for each population, we allow a number of off-springs/children with independent modifications. We pick the child with the lowest energy to continue the search. After a set number of iterations or when a convergence criterion is fulfilled – here, the change in energy between iterations – we end the optimization and pick the minimum over all populations as an approximation to the optimal solution. Within the search, we make use of an additional "patience" hyperparameter – this parameter is increased whenever the modifications does not lead to an improvement in energy. When the patience parameter reaches a certain value, the associated population "frustrates" and re-sets its circuit instructions to a previous stage. We further allow updates such as in simulated annealing, where we pick the best offspring per population with probability $e^{-\frac{\Delta E}{T}}$ and decrease temperature $T$ over the course of iterations. As a result, the best improvement in energy is picked most likely but there is still a chance that even an increase in energy would be allowed to avoid local minima; however, we do not allow increases in energy beyond the minmium energy that was found for the employed reference method $U$ with $M = \mathbb{I}$.

**Algorithm 1** Architecture search for $M$

**Input** operator pool, reference method / circuits, number of populations, offsprings per population and maximum iterations
**for all** populations **do**
    compute reference energy
    sample initial circuits $M$
    optimize reference to compute energy with initial circuits
**end for**
**while** not converged **do**
    **for all** populations **do**
        **for all** offsprings per population **do**
            propose modification to $M$
            optimize reference to compute energy
        **end for**
        pick offspring to continue with based with probability $e^{-\frac{\Delta E}{T}}$, update $T$
    **end for**
**end while**
**Output** circuit $M$, optimal reference, approximation to ground-state energy

After each such optimization step, the parameters in the cluster circuit $U_j(\theta)$ are adjusted. For the case of Clifford circuits, one may consider techniques proposed in [FD19; RLP22] to further optimize the depth of the resulting circuits.

#### Near-Clifford circuits to increase expressibility

It is well-known that circuits made up from Clifford gates only are not universal [Got98b]. We aim to counter-act this by adding few non-Clifford gates to the circuit $M(\tau)$. However, we also know that including more and more non-Cliffords likely leads to an immense increase in number of terms in the Hamiltonian, making measurement intractable. Thus what we understand as a near-Clifford circuit is a circuit that results in a "reasonable" increase in the sense that $|\tilde{H}| \leq C|H|$ for some moderately-sized constant $C$. In particular, we want to avoid a regime with exponential increase. An easy choice whose capabilities we will explore later on is modifying a single Clifford gate towards its parametrized version, i.e. for a Clifford gate $U_{\text{Cliff}}$ with generator form $U_{\text{Cliff}} = \exp(-\mathrm{i}\theta_{\text{Cliff}}G)$ and fixed $\theta_{\text{Cliff}}$, we use $U(\theta) = \exp(-\mathrm{i}\theta G)$ with a free parameter that can be used for optimization. Hence, for our experiments here, we will loop over all Clifford gates within the virtual circuits $M$, except for controlled gates and SWAP, and replace them with parametrized versions. We outline this strategy in algorithm 2. For example, we substitute $X \to R_X(\tau)$ or $\text{Exc}_{1;2}$ (cp. with Tab. 2.1) receive a $R_Z(\tau)$ as center gate.

### 2.1.3 Results

We will demonstrate our proposed technique on a set of small molecular examples to validate capabilities of Clifford and near-Clifford circuits to re-gain correlation due to the assumed product state. For the hydrogen molecule, we choose a basis set of Gaussians, whereas for beryllium hydride and the nitrogen molecule, we use a basis of pair-natural orbitals generated by multi-resolution analysis [KSTA21; KA22], as described before.

**Algorithm 2** Optimization of near-Clifford $M(\tau)$ with one single-qubit non-Clifford gate

**Input** non-Clifford version of operator pool, reference method / circuits, optimized Clifford circuit $M$ and maximum iterations
**for all** (single-qubit) Clifford gates $V$ in $M$ **do**
    replace $V$ with its non-Clifford version $g(\tau)$ and initial parameter set to correspond to Clifford
    **while** not converged **do**
        optimize reference
        optimize parameters in $V(\tau)$ with only few iterations
    **end while**
    re-update reference and then $\tau$ with smaller tolerance/more iterations
**end for**
set $V(\tau)$ with best performance as non-Clifford
**Output** circuit $M(\tau)$, optimal parameter $\tau^\star$, optimal reference and approximation to ground-state energy

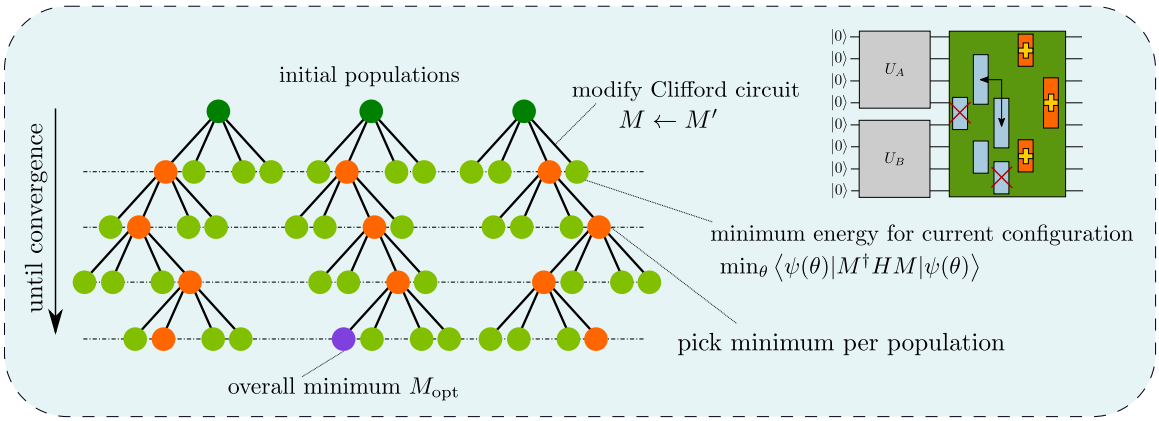



Figure 2.3: Outline of the genetic algorithm used to assemble the classical circuit $M$. Based on some initial circuits (initial populations) made up by random gates drawn from the operator pool in Tab. 2.1, random changes of this circuit in form of addition, deletion and change of gates as shown in the top-right corner, are proposed. The best change per circuit is used as a reference for the next iteration as long as this change does not perform worse than the reference method; this procedure is repeated until convergence or until a maximum number of iteration is reached.

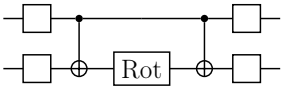
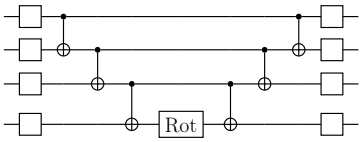
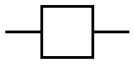
| | |
|---|---|
| Pool of Clifford gates | $\{H, S, X, Y, Z, C_X, C_Y, C_Z, \mathrm{SWAP}, \mathrm{Exc}_1, \mathrm{Exc}_2\}$ |
| $\mathrm{Exc}_1$ |   |
| $\mathrm{Exc}_2$ |   |
| $\mathrm{Rot} \in \{S, Z, S^\dagger\}$, |  $\in \{R_X(\pi/2), R_Y(\pi/2)\}$ |

Table 2.1: Operator pool used for Clifford optimization including excitation-like gates $\mathrm{Exc}_1, \mathrm{Exc}_2$. These consist of basis transformations, CNOT ladders and a central $Z$-rotation, which implements operations $e^{-\mathrm{i}\frac{\theta}{2}\sigma_1\sigma_2\cdots\sigma_{N_\mathrm{q}}}$, where the basis transformations allow $\sigma_i$ to be any of $\{X, Y, Z\}$. Further, $\mathrm{Rot} \in \{S, Z, S^\dagger\}$, $\theta \in \{\frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$.

The simulations are based on a pre-optimized product state

$$|\mathrm{ref}\rangle = \left(\prod_{j=1}^{N_\mathrm{cl}} U_j\right)|0\rangle \tag{2.1.17}$$

that we call reference method. That is, "reference" within this study decribes a product state over chosen qubit clusters and not describe the Hartree-Fock state. We choose the following reference method for all our systems: We use a power method-type procedure that enforces a product state in the wavefunction to find the ground-state. A brief overview of this method can be found in Appendix A.1.

**Simulation setup**

Simulation code was written with the TEQUILA [Kot+21] framework using the QULACS simulator [Suz+21] as simulation backend for the quantum circuits and the automatically differentiable framework of [KAA21]. Direct basis determination via pair-natural orbitals was performed with MADNESS [Har+16] as described in [KSTA21] with the diagonal approximation described in [KA22] and the MP2-PNO implementation of [KBV20] without cusp regularization. The latter interfaces mean-field implementations described in [Har+04; Bis14]. The Jordan-Wigner encoding corresponds to the implementation in OPENFERMION [McC+20b]. Orbitals in standard basis sets were computed with PSI4 [Smi+20]while orbital optimization was carried out in PYSCF [Sun+20; ZC22].

For the genetic algorithm as outlined in Fig. 2.3, we choose the number of populations between 10–15 and the number of offsprings per population as 8–10. The convergence criterion for the reference method is set to $10^{-5}$, and the maximum number of iterations (tree depth in Fig. 2.3) is set between 10–25. We observed that the performance of the algorithm is quite insensitive to these parameters (around $10^{-4}$ milli-Hartrees), which motivates our choice, cp. to Fig. A.2 in Appendix A.2.

For all the following results, we show the error of various employed methods with respect to the

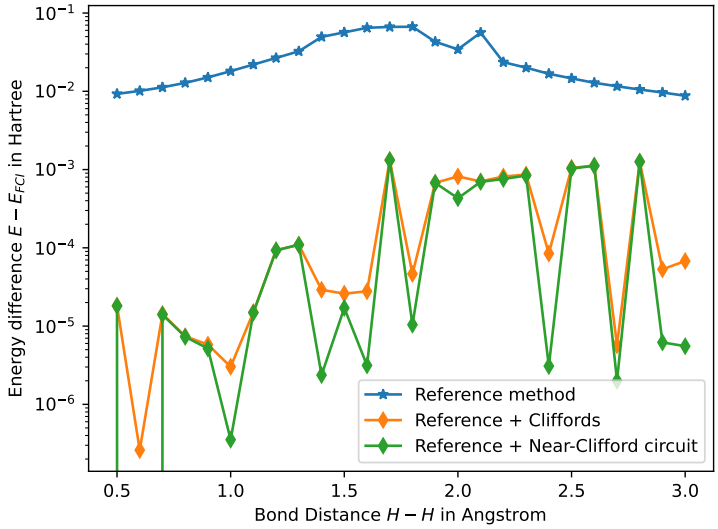



Figure 2.4: Simulating the ground-state of $H_2$ in the 6-31G basis set using an arbitrary active space of six qubits; here, we also use an uninformed clustering, where the first set of three and the second set of three qubits are enforced into a separable state."Reference method" refers to the power-method based reference from Eq. (2.1.17) or Appendix A.1.

exact ground state (Full Configuration Interaction) within the same basis set. The reference state as in Eq. (2.1.17) is obtained following the procedure in Appendix A.1. With this as a starting point, we assemble a virtual circuit $M$ built from Clifford gates only ("Reference + Cliffords") and subsequently search over all Clifford gates and check if the energy can be improved by optimizing their parameter when making them parametrized ("Reference + Near-Clifford circuit").

### Hydrogen molecule

We use the 6-31G basis set for $H_2$, where we choose an active space of six spin-orbitals to obtain a six-qubit toy example. As a reference, we enforce a product state of overall six qubits, made up by the product over a state containing the "first" and another one containing the "last" three orbitals. Here, first and last corresponds to the ordering with respect to orbital energies. As showcased in Fig. 2.4, this approach has a rather high error, of the order of hundreds of milli-Hartree. However, addition of a virtual Clifford circuit enables to suppress the error to the order of milli-Hartrees. In the case of small bond distances, the reduction is as much as three orders of magnitude. For large bond distances, the improvement by adding additional entanglement is lower but still accounts for roughly one order of magnitude. This is because the true state is close to a state preparable by a pure Clifford circuit supported only on the first four qubits as explained in [Kot22]. To sum up, we observe significant improvement for Clifford-gates only. While the addition of freely parametrized non-Clifford gates does not reliably lower the energy, if successful, it can yield up to another order of magnitude in accuracy.

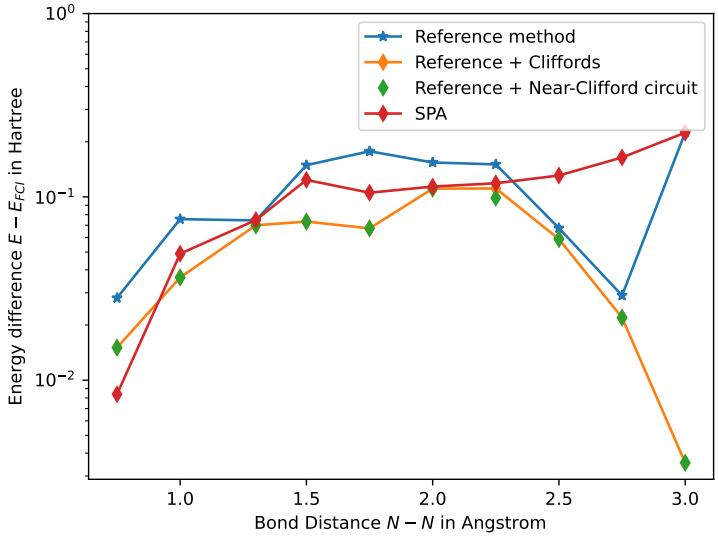



Figure 2.5: $N_2$ with two SPA pairs as clusters in frozen-core approximation, 12 qubit simulation. "Reference method" refers to the power-method based reference from Eq. (2.1.17) or Appendix A.1.

### Nitrogen molecule

For the $N_2$ molecule, as depicted in Fig. 2.5, the SPA outperforms our product state reference method in most regimes, while still not coming close to FCI level. The SPA performs worse here due to the surrogate model that is used to generate the PNOs which are used as a basis. They are generated using second-order Møller-Plesset perturbation theory and thus the PNOs inherit its shortcomings [KBV20; KSTA21; SKA22]. Even more powerful models, such as $k$-UpCCGSD, struggle here [LHHW18; KA22]. While our method makes use of the same PNO basis, it is, in contrast to the SPA, able to recover a part of this error.

Upon addition of a virtual Clifford circuit, we again observe a consistent improvement. It is noticeable that for short bond lengths, SPA and our method overall perform quite comparably. For longer bond lengths, despite using the same PNO basis as SPA, it seems to be able to recover part of the error due to the basis. Furthermore, we found that parametrizing a single Clifford gate of the virtual circuit for the most part improves the energies compared to the Clifford circuit only barely. As the parameter space for $N_2$ is already considerably large, convergence of the genetic algorithm for the Clifford-circuit already takes rather long. The increase in complexity by adding even a single non-Clifford gate already makes it hard for the algorithm to find optimal configurations.

### Beryllium hydride

We further consider $BeH_2$ as a system of interest in comparison with results from [KA22]. We use a basis of eight qubits in a frozen-core approximation using MRA-PNOs (PNOs determined by multi-resolution analysis), made up by two clusters of two spatial orbitals based on each bonding electron pair. As expected, we can see in Fig. 2.6a that the SPA-ansatz slightly outperforms our reference method. However, the addition of a virtual circuit made up by Cliffords only allows to improve beyond the SPA. This can be further boosted by parametrizing one of the Clifford gates. We note

that the improvement seems to produce a somewhat constant offset, while the general behaviour of the error curve remains unchanged. This behaviour is similar to orbital optimization on top of the SPA reference (OO-SPA), as done in [KA22]. However, we note that the improvement by the optimized Clifford circuits and especially near-Clifford circuits is generally slightly higher. This is despite of the orbital optimizations being non-Clifford operations, which are typically expected to be more expressible. We observed that the behaviour of the procedure is more favorable if power method-based reference method is used instead of the SPA as a reference. The outcome of the initial optimization is somewhat randomized and therefore has eased to find gate sequences that lower the energy in subsequent iterations. We expect SPA to be stuck in a more stable local minimum as compared to the iterative power-method solver, as for SPA, our procedure consistently either aborted early or only led to circuits outputting the same energy. Beyond that, we provide a discussion on fidelity computations for optimization outcomes with eigenstates obtained by direct diagonalization of the Hamiltonian in the eight qubit basis in Appendix A.3, Fig. A.3.

We next investigate the consequences of using non-optimal clustering, i.e. clusters from different bonding electron pairs. We mix pairs so that a spin-orbital of one pair is exchanged with one the other pair. Results for this case can be found in Figs. 2.6b and 2.6c as well as that the reference method behaves significantly worse. However, this does not seem to influence the outcome of the computations that include a virtual circuit. This is likely due to the inclusion of SWAP gates into the operator pool, which are able to retrieve the optimal clustering by permuting orbitals [Tka+21]. In fact, we notice here that the circuits built from Clifford gates only perform as well as the near-Clifford circuits for most geometries and performs slightly better as in the un-permuted case. Since the employed algorithm is random and the outcome of the genetic algorithm with a finite number of populations and offsprings has some variance, this is not surprising.

#### Increase in the number of Hamiltonian terms

For the results discussed above, we show the increase of the number of terms in the Hamiltonian when folding the classical circuit in Fig. 2.7b, compared to folding a full UCCSD circuit several non-Clifford gates in Fig. 2.7a. We show both the cardinality before ($|H|$) and after folding ($|\tilde{H}|$). Additionally we demonstrate the range of cardinalities that can be accessed by exchanging one Clifford gate for another one in the previously optimized Clifford circuits for each geometry. As expected, the upper bound of $4^{N_q}$ times the previous complexity is respected.

While it is clear that the measurement cost increases, we note that the increase when folding one gate should be manageable given that a single non-Clifford gate only increases the complexity by a constant factor. In fact, in the case of hydrogen, the cardinality after folding for the optimized circuit is almost at the lower bound of the range of circuits. However, this cannot be established as a trend for the larger systems.

#### Circuit structures

We further take a look at the characteristics of the learned circuits in order to check if there are some distinct circuit building blocks noticable that have been favored by the genetic algorithm used. We note that a recurring theme in the circuits discovered are $C_X$ ladder-like structures that entangle and exchange information across the qubits (Fig. 2.8). One can also see that in the way we set up the algorithm, there is an initial part of the circuit that does not have support beyond the clusters.

This means one can see this section of the virtual circuit rather as a part of $U$ than $M(\tau)$, and one may choose to not allow such gates if one assumes the reference circuit $U$ is already good enough. An argument for leaving the gates that could be incorporated into the cluster-unitaries $U_j$ is that the algorithm has identified a part of the description of the cluster-states $|\psi_j\rangle$ that can be described classically. This is beneficial when an ansatz for the circuits $U_j$ has been established that is evaluated on real quantum computer. Then, thanks to $M$, the circuit for $U_j$ could be shorter.

### 2.1.4 Conclusion and Outlook

In this work, we investigate whether a larger quantum chemical simulation can be run on smaller quantum computers without sacrificing the accuracy too much. This procedure is of practical interest because it allows to make use of smaller quantum computers that will be available in the near future. Furthermore, reducing the circuit size can allow to alleviate the problem of limited connectivity between qubits as when reduced to individual problems on less qubits, one can use better-connected parts of the architecture only.

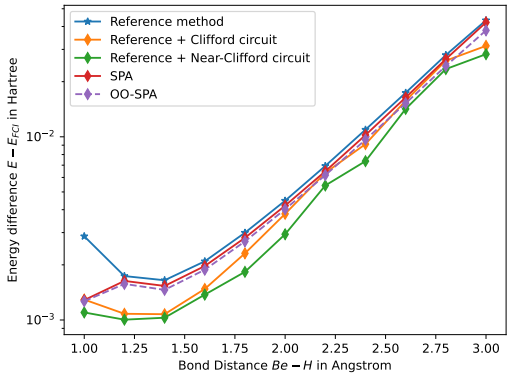
We explored the capabilities of Clifford and near-Clifford circuits to re-entangle a molecular wavefunction that has been enforced into the form of a product state. Within our proposed procedure, we use the energy as the only, global criterion for optimization; we deploy a genetic algorithm that mutates a set of random, initial virtual circuits using a Clifford operator pool. Empirical validation of our re-entangling method using Cliffords only shows that the partitioned ansatz, implemented on a smaller set of qubits, achieves similar levels of accuracy as the separable-pair ansatz. After successful optimization of the Clifford circuit, we "turn on" the parameter of one the Clifford gates in the virtual circuit to further improve the energy (while continuing to adjust the parameters in the cluster circuit). As we demonstrate, the increase in measurement cost using such near-Clifford circuits is still modest. A worthwhile direction to follow in the future would be to perform a more concise investigation of near-Clifford operations in this context, e.g. resorting to the proposed approach in [LGI22] to assess the Hamiltonian growth.

Future work might include investigating more sophisticated genetic algorithms. Aside of cross-population mutations that enable broader exploration of the space of Clifford circuits, it would be interesting to include populations with permuted orbital ordering at the beginning of the optimization procedure. This is motivated by our observations in the case of $BeH_2$ in Fig. 2.6b, where the permuted ordering is able to outperform the optimal ordering for some geometries.

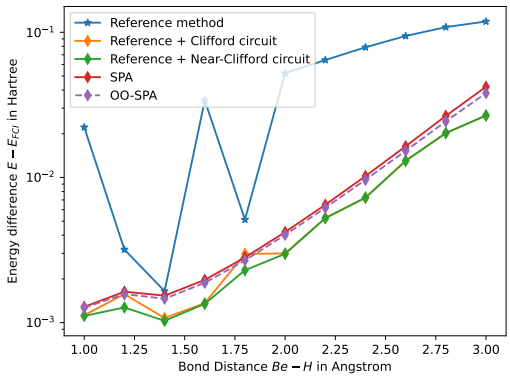
Despite yielding significant improvements for $H_2$ and $BeH_2$, the improvements we see by adding the non-Clifford gate is nearly negligible in the case of $N_2$, which suggests that one non-Clifford gate is not enough in this case. Thus further work needs to be done to explore near-Clifford circuits and in particular influence of few, specific gates on the expressibility of circuits.

## Code and Data Availability

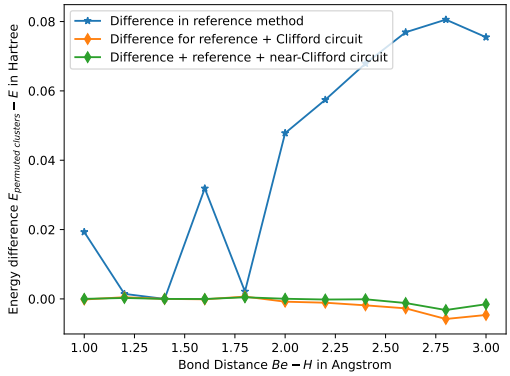
Results and code for the Clifford-partitioning can be found at `https://github.com/philipp-q/partitioning-with-cliffords`. Furthermore code for the power method based reference methods can be found at `https://github.com/philipp-q/power_method_for_product_states`.




(a) $BeH_2$ with clusters according to electron pairs, 8 qubit simulation

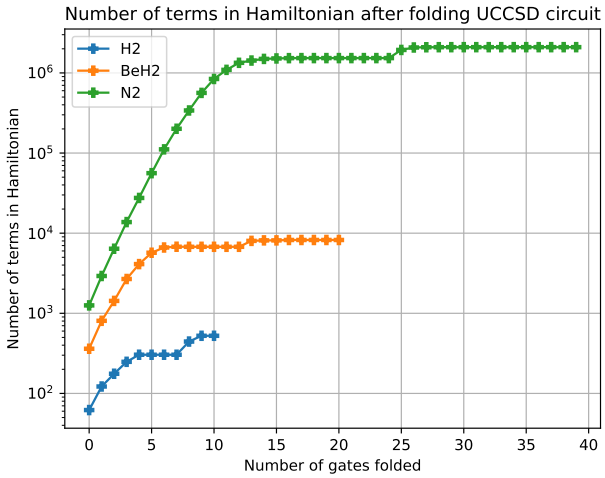



(b) $BeH_2$ with permuted reference clusters, also see Fig. 2.6c.

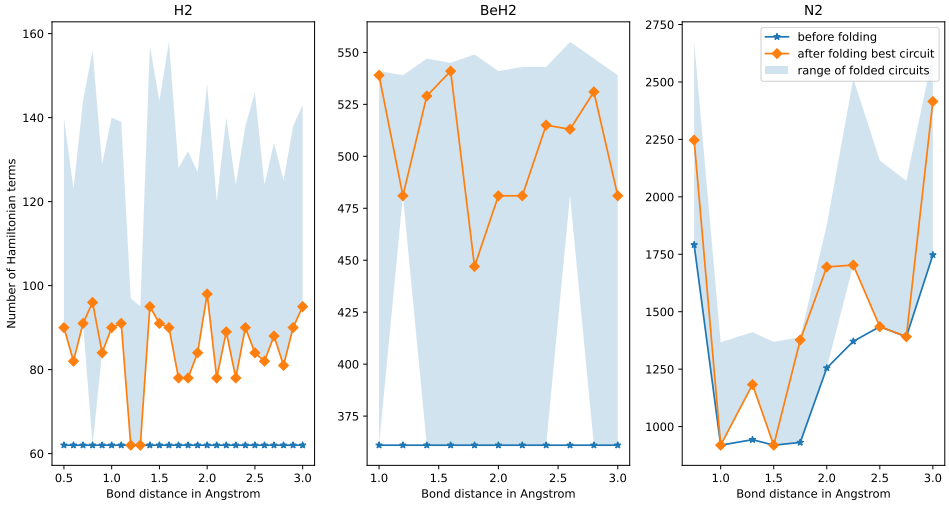



(c) Comparison of $BeH_2$ with pair-wise and modified clusters, showing energy difference between permuted clusters and clusters by electron pairs. For a configuration between clusters $A$ and $B$ as $(000_A, 111_B)$, where 0,1 shows association to an electron pair, we permute to $(100_A, 011_B)$.

Figure 2.6: Errors of optimization results with respect to FCI in corresponding bases. "Reference method" refers to the power-method based reference from Eq. (2.1.17) or Appendix A.1.




(a) Increase in number of terms in the Hamiltonian for $H_2$, $BeH_2$ and $N_2$ molecules when folding a vanilla UCCSD circuit [Ana+22]. The horizontal lines without term increase correspond to Clifford gates.




(b) Increase of the number of terms in the Hamiltonian after folding the optimized circuits when using only one gate of the circuit as a parametrized non-Clifford gate. The range of folded circuits depicts the envelope of maximum- and minimum number of terms that are possible when parametrizing a single gate of the previously optimized Clifford circuit, assuming the parametrization makes them non-Clifford. Otherwise the minimum coincides with the number of terms before folding (blue line). The theoretically maximum number of terms are as follows. $H_2$: $N_q = 6, 4^{N_q} = 4096$; $BeH_2$: $N_q = 8, 4^{N_q} = 65536$; $N_2$: $N_q = 12, 4^{N_q} = 16777216$.

Figure 2.7: Increase in Hamiltonian terms for non-Clifford (a) and near-Clifford (b) circuits

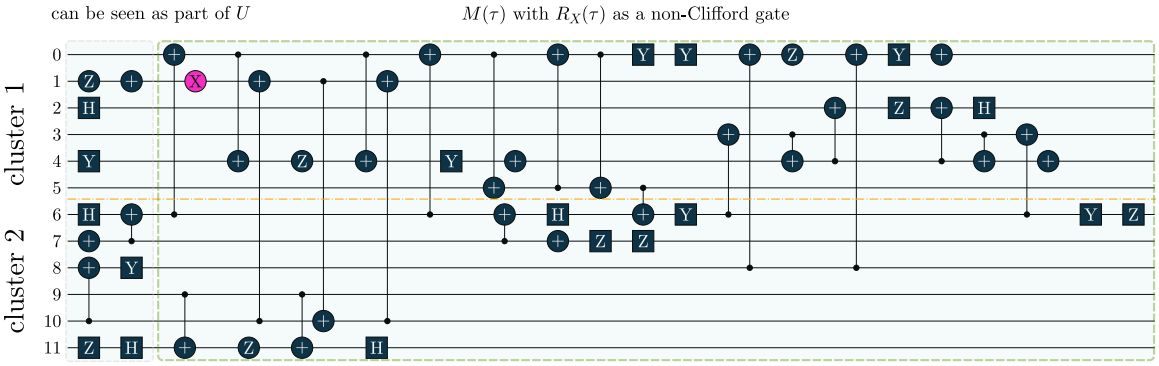



Figure 2.8: Optimized circuit architecture for $N_2$ at a bond distance of 2.0 Angstrom; pink gates denote a parametrized gate and are thus non-Clifford gates. Note that gates in the beginning until the first gate with support beyond two clusters can technically be seen as an extension of the quantum circuit as they do not add correlation between clusters. One may choose to allow or not allow such gates.

## 2.2 Efficient Quantum Chemistry via Quantum Dynamics

This is undergoing peer-review and available as a preprint as:

> Philipp Schleich, Lasse Bjørn Kristensen, Jorge A Angulo Campos, Abdulrahman Aldossary, Davide Avagliano, Mohsen Bagherimehrab, Christoph Gorgulla, Joe Fitzsimons, and Alán Aspuru-Guzik. "Chemically Motivated Simulation Problems are Efficiently Solvable by a Quantum Computer". In: *arXiv preprint arXiv:2401.09268* (2024).

### 2.2.1 Introduction

The idea of using quantum computers for the simulation of quantum systems goes back to the first proposals of quantum computing by Benioff, Feynman, and Manin [Ben80; Fey82; Man80]. This idea has generated substantial effort toward applying quantum computing to chemical problems, as conventional quantum many-body simulations for chemistry are inherently limited by the curse of dimensionality and constitute a significant portion of current supercomputing usage. This effort has largely focused on two problems: quantum simulation of dynamics (also known simply as Hamiltonian simulation or quantum simulation in the field of quantum computing) and the ground-state preparation problem. Quantum simulation [Zal98; Wie96; AL97; AT03; BACS07] describes the problem of time-evolving an initial state according to the Schrödinger equation under a Hamiltonian. The relevant Hamiltonians for chemical and most physical processes can be efficiently computed, and the corresponding time evolution, governed by the Schrödinger equation, is provably within the computational complexity class BQP on a quantum computer. This class encompasses decision problems that a quantum computer can solve in polynomial time with a bounded error probability [AL97; Llo96]. In contrast, the problem of determining the ground state – formulated as the local Hamiltonian problem in quantum complexity theory – is complete for the class Quantum Merlin-Arthur (QMA) [Llo96; AL99; KKR05], a larger class sometimes compared to the classical complexity class NP. More recently, it was shown that simulating Schrödinger operators of the form $-\Delta + V$ with some restrictions on $V$ and without particle exchange symmetries, a slight restriction compared to general local Hamiltonians, is also BQP-complete, and the ground-state problem for such operators is StoqMA-hard [ZLLW24]; namely Merlin-Arthur given Hamiltonians without sign problem (stoquastic Hamiltonians) [BBT06]. As a result, it is not clear whether ground state problems can be solved efficiently, even on a quantum computer.

Ground-state search and Hamiltonian simulation are often used jointly. Simulation algorithms serve as a subroutine to find the ground-state energy (e.g., via phase estimation [AL99; Kit95; ADLH05; WBA11]) or in the elucidation of reaction mechanisms described in [Rei+17]), while ground states can be input states for performing quantum dynamics. State-of-the-art implementations of Hamiltonian simulation can be categorized into several classes. One is the class of algorithms based on Trotter-Suzuki formulas [Llo96; Chi+21], which split the exponential of a sum into products of exponentials. Another class is based on "qubitization" [LC19; Ber+19], which makes use of quantum signal processing [MW24] to encode the simulation in a quantum walk. Additionally, there are randomized algorithms such as QDRIFT [Cam19] and its extensions [NBA24; Poc+24]. Each of these approaches has its advantages in different scenarios [RRW22; HW23], and methods

based on product formulas in particular can make the evolution of perturbative systems, as often present in chemistry and physics, even more efficient [Bag+24; Bos+24]. However, as they all lead to polynomially sized circuits for Hamiltonians of constant sparsity, we refrain from going into details here and refer to the relevant literature. A good overview can be found in [Su+21].

We propose a departure from the current mainstream approach of the quantum computing community to chemistry, moving beyond the 1950s computational chemist's way of thinking [Löw58; Löw95], which has been shaped by the limitations of classical computing to use the ground-state as the starting point of computations, to a new era. With the advent of fault-tolerant quantum computers, dynamical simulation of quantities that a practicing chemist might care about is within reach; after all, most relevant quantum chemistry problems are inherently addressable through dynamical evolution alone, leading to efficient quantum algorithms for these problems.

Our proposed framework employs a limited set of attainable atomic initial states and then dynamically prepares input states for a reaction of interest through a scattering process. Particularly, our main contribution is the approach to prepare molecular states by hierarchically 'combining' atomic ones – here specifically by scattering them. This allows for a state preparation heuristic guided by chemical intuition. Then, another dynamical evolution embodies a reaction, and a broad set of relevant quantities can be measured. Under certain assumptions on what constitutes relevant initial states, the ensuing algorithm is not limited by the QMA-hardness of preparing ground states and thermal states. Similarly, we can circumvent an orthogonality catastrophe, namely, a vanishing success probability to retrieve a ground state. As it has been recently shown [Lee+23], this effect is why exponential speedups for ground-state energy estimation of molecular Hamiltonians on quantum computers may be hard to attain as the (state-of-the-art) methods considered for state preparation fail to produce reliable overlaps for molecules of increasing size. However, a distinction to make is that our methodology is based on heuristical physical intuition, which makes it not directly comparable to methods like QPE from a complexity-theoretic perspective.

Two observations are at the foundation of our approach. First, the simulation of $k$-local Hamiltonians is achievable by polynomial-size quantum circuits. Hamiltonians that stem from chemical problems are 2-local due to the nature of the Coulomb interaction and, thus, also have finite locality when represented as quantum operations on qubits. Including photons to simulate light-matter interactions in the computation increases the maximum support of operations but not beyond a constant factor. The second observation is that we aim to simulate processes corresponding to experiments that can be performed in a lab in finite time, i.e., problems that are, in some sense, experimentally verifiable. Molecular ground states, as viewed from the perspective of computational chemistry, which implies a frozen molecular geometry at absolute-zero temperature, typically do not represent a system's thermal equilibrium state and hence do not belong to this class. We propose to simulate the process of producing reactants with a hierarchical scattering process. First, we prepare the ground states of atoms by quantum phase estimation – this does not fall under the orthogonality catastrophe as atoms are finite-sized and as long as we stay within such where efficient heuristics for initial state preparation are available – and then use a simple scattering process to produce molecular reactants. Our method integrates artificial potentials and photonic fields to induce the success of scatterings, leading to a lower-bounded probability of success. Specifically, mergo-association as discussed in Section 2.2.4 is a promising avenue to realize this. This means that a fixed number of repetitions of a weak measurement scheme – see Section 2.2.4 – to herald success will suffice to ensure

the success of intermediary scattering processes. Thus, meaningful molecular input states, which do not need to be ground states, are efficiently prepared. Our framework facilitates the modeling of complex chemical reactions by hierarchically operating the scattering with $N$ atoms to create $M$ reactants, which can then undergo a quantum simulation corresponding to a reaction, followed by measurements of reaction rates and time correlation functions.

### 2.2.2 Hamiltonian Simulation and Complexity Considerations

We continue by discussing relevant complexity classes for the problems we consider and the efficiency of their solutions. The complexity class BQP (bounded-error quantum polynomial-time) is oftentimes considered the quantum generalization of the complexity class P (polynomial time), or, more precisely, its probabilistic extension BPP (bounded-error probabilistic polynomial-time). Polynomial complexity is usually considered efficient as the increase in cost when scaling relevant parameters is somewhat moderate. Problems from QMA (Quantum Merlin-Arthur), in contrast, can be considered "hard"; QMA is the quantum analog of NP (non-deterministic polynomial time) or, more precisely, the probabilistic class MA (Merlin-Arthur). QMA describes promise problems with solutions that can be verified in polynomial time with bounded error probability. However, there is no guarantee of efficiency in finding their solution. [Amb14; WB21; BCW21] Our argument is that a large class of chemically relevant phenomena can be addressed by algorithms solving efficiently solvable problems. More precisely, the intuitive argument we make is that phenomena that occur in finite time in a chemical laboratory can be simulated in finite time. The gap between time in nature and time on a quantum computer is only polynomially sized, which is why we relate it to BQP. Systems in the world around us during chemical reactions typically do not cool down to the ground state; thus, it would not be part of the chemically relevant quantities we consider. In fact, it was recently shown that local minima are more favorable to reach in quantum systems than ground states [CHPZ24].

Here, we can draw a distinction between polynomially-sized problems, such as those from BQP, and QMA-hard problems, such as finding the ground state. In this way, we envision a fundamental change in the way chemistry problems on quantum computers are typically approached, and we display this in Fig. 2.9. We denote by Chem the set of computational problems of chemical relevance – including phenomena like ground states. Note that these are not necessarily decision problems; our argumentation is based on circuit fragments that stem from known complexity results rather than decision problems and the language of complexity theory. We call ChemPoly to the focal set of problems within this work, namely, problems of relevance to chemistry and that have polynomial complexity. Due to the BQP-completeness of Hamiltonian simulation [Fey85], perhaps one can reduce polynomially-sized problems to ones from the set of problems with chemical relevance. Irrespective of this point, our present study highlights a wide set of chemically interesting problems involving simulation as a subroutine yet avoiding subroutines and decision problems known to be QMA-hard. Many approaches in chemistry that involve searching for the ground state are QMA-, StoqMA or NP-hard [SV09; WLA12; OIWF22; ZLLW24]; examples of tasks that fall into this category are finding the universal functional in density functional theory or the Hartree-Fock problem. Significant progress was made on solving the latter numerically, to the point that it is mostly seen as "solved" despite the formal hardness. This view results in hope about tackling the ground-state problem as well. Yet still, at this point, we seem far from similar success on chemistry ground-state

search, and the biggest hope for solving practical chemical problems lies in dynamics.

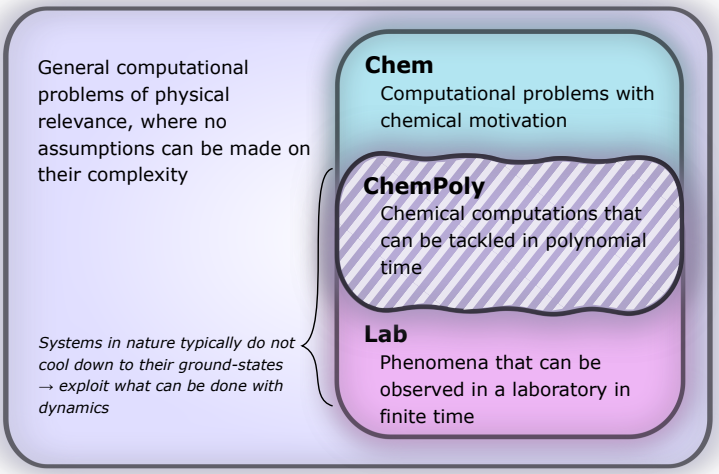



Figure 2.9: Complexity of solving chemical problems. We target the set of computational problems, ChemPoly, consisting of problems within chemistry with polynomial complexity when solved on a quantum computer. These problems have unknown overlap with those that correspond to observables in a laboratory; the sets Lab and ChemPoly may also coincide.

To represent chemical systems, we choose molecular Hamiltonians that are not necessarily restricted to the Born-Oppenheimer (BO) approximation. Non-BO was previously explored in the split-operator formalism on quantum computers, where kinetic and potential dynamics are factored [Kas+08; Cha+23]. While it was argued in [Kas+08] that the non-BO approach is more efficient than implementing the BO procedure, there have been recent advances that render BO more efficient thanks to a fully coherent implementation [Sim+24]. Hamiltonians occurring in chemistry are composed by the operators outlaid in Table 2.2, from which locality is an obvious consequence, as $k$-locality follows from the 2-body nature of the interactions.

| | |
|---|---|
| Charged particle kinetic energy | $\frac{\mathbf{p}^2}{2m}$ |
| Photon energy | $\omega\left(a^\dagger a + \frac{1}{2}\right)$ |
| Interparticle potential | $\frac{qq'}{\lvert\mathbf{r}-\mathbf{r}'\rvert}$ |
| Photon-particle interaction | $\frac{q}{m}\mathbf{A}(\mathbf{r})\left[\frac{q}{2m}\mathbf{A}(\mathbf{r}) - \mathbf{p}\right]$ |

Table 2.2: Hamiltonian components for molecular systems interacting with photons. For a given particle with mass $m$ and charge $q$, $\mathbf{p}$ and $\mathbf{r}$ are, respectively, the corresponding momentum and position operators. For a photon with frequency $\omega$, $a$ and $a^\dagger$ are the corresponding annihilation and creation operators. $\mathbf{A}(\mathbf{r}) \propto \mathbf{c}(\mathbf{k},\mathbf{r})a + \mathbf{c}(\mathbf{k},\mathbf{r})^* a^\dagger$, where $\mathbf{k}$ is the photon wave-vector, and $\mathbf{c}(\mathbf{k},\mathbf{r})$ is a polarization vector, is the vector potential corresponding to the photon field.

Our framework is independent of the choice of basis used to represent the Hamiltonian numerically and the choice between first and second quantization. Asymptotically, a first-quantized representation tends to be more efficient for fault-tolerant quantum algorithms with an abundant number of logical qubits, as the number of required qubits grows linearly with the number of particles and

logarithmically with the number of basis functions (since one stores the basis information for every particle), as opposed to the linear dependence in the number of basis functions for second quantization, where one stores occupation for each basis function. [Su+21; Bab+16; Kiv+18; Bab+18a; Bab+18b; McC+20a] In this work, we restrict ourselves to the first-quantized representation for concrete examples. For a circuit to have polynomially-sized complexity, it is further necessary that the operator norm of the simulated Hamiltonian, or rather, its individual terms acting on at most $k$ qubits, grow at most polynomially in the number of qubits used to represent them [AT03; BACS07; Llo96]. Energy is an extensive thermodynamic property, so it grows roughly linearly by increasing the system size (i.e., the number of particles). Therefore, the thermodynamic relation between the amount of matter and internal energy is linear.

### 2.2.3 Computational Framework

In what follows, we describe the computational framework in more detail – i.e., a dynamics-based state-preparation scheme that serves as input for a "main" quantum dynamics routine, preceding measurement. Our overall idea is based on the experiment in [Liu+18], where molecules were "built" using two atoms with finite success probability, as well as a mergo-association scheme that merges atoms, e.g. Rb and Cs, using a trap potential [Rut+23; BLH23; BH24]. The chemical dynamics we aim to simulate to extract chemical properties requires a sensible initial state for the time evolution. This state must reproduce a natural one faithfully extract the desired properties. Depending on the process under consideration, good candidates could be found among ground-states, or perturbed ground-states, such as those obtained by laser excitation. However, as previously argued, general molecular ground states are hard to access. Nevertheless, preparing the ground states of atoms is feasible – the lighter the atom, the easier – and can be done efficiently as constant overlap for heuristic input states is expected [Lee+23]. For this reason, we propose to follow a hierarchical approach, as outlined in Fig. 2.10. All processes generating the reactants and products are carried out through Hamiltonian simulation with local Hamiltonians, including molecular and external field components. After the using this technique to generate initial states, we aim to measure observables, e.g., reaction rates according to the scheme in [Kas+08; Ped+14] or auto-correlation functions [KLFK24].

We start with a state representing $N$ atoms, all of which are assumed to be in their ground states or another state of interest $\rho_i^{(\mathrm{atom})}$, such as a thermal state. The preparation of these states can be achieved by existing algorithms for ground-state [AL99; ADLH05; DCL24; Ber+24; HLSW24] and thermal-state preparation [CKBG23; HW25] followed by amplitude amplification to enhance the probability of obtaining the desired state. For generality, we represent both pure and mixed states using density-matrix notation. Since the atoms are all finitely sized, and we can prepare an initial state with a significant overlap with the desired state for each atom, any amplitude amplification costs are constant with respect to overall system size. The overall cost to prepare the atomic initial states is $O(N\ \mathrm{poly}(\frac{1}{\varepsilon}))$, with $\varepsilon$ being the accuracy in preparing the atomic states. Using these atoms encoded as a state $\rho_1^{(\mathrm{atom})}\rho_2^{(\mathrm{atom})}\cdots\rho_N^{(\mathrm{atom})}$, we seek to create $M$ molecular reactants $\rho_1^{(\mathrm{mol})}\rho_2^{(\mathrm{mol})}\cdots\rho_M^{(\mathrm{mol})}$.

Based on these initial states, we next prepare the reactants by a scattering process mimicking a physical experiment. In other words, we jointly evolve a set of atoms meant to form a reactant molecule.We discuss the modeling and treatment of a bath that allows exchanging energy with an environment further below.

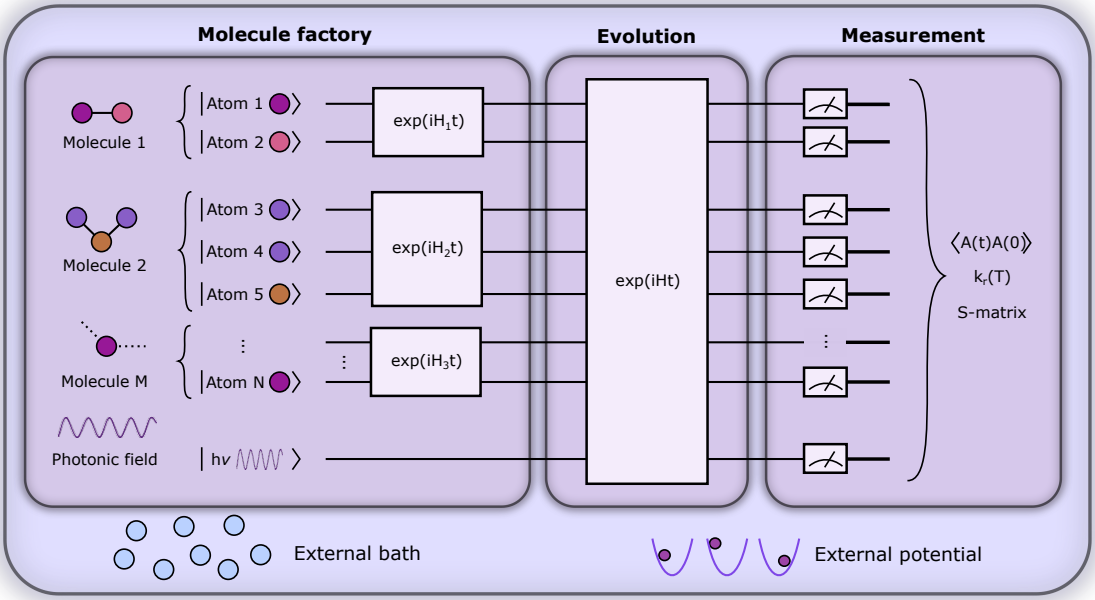



Figure 2.10: Computational framework. Our simulation framework can be separated into a state preparation procedure ("molecule factory"), the evolution of the reaction of interest, and a measurement step that extracts usable information. The molecule factory prepares a set of molecular input states for the reaction, which may resemble thermal or ground states. These states are produced in a tree-like fashion equipped with a weak measurement scheme to ensure that the target states are prepared with sufficient probability in a heralded way. A photonic field serves as a source of energy for reactions, and an external bath, either explicit or implicit, serves as an energy sink. Furthermore, we utilize external potentials in the spirit of optical tweezers tailored to the different Hamiltonians along the procedure to ensure sufficient success probability and to control positions in space.

Consider one of the reactant molecules to be prepared. We combine the constituent atoms by successively scattering in a tree-like manner, as in Fig. 2.10, until the desired molecular state is attained. In the worst case, each molecule is generated by combining only two participants at a time, leading to a binary scattering tree, and thus an overall number of scatterings that is quasi-linear in the number of input atoms. It is essential to ensure a high overlap with the desired intermediate state at each level in this procedure. Otherwise, the overlap would decrease at each combination step. For instance, with an initial overlap of $(1-\delta)$, the overlap would drop exponentially to $(1-\delta)^n$ after $n$ steps. A seemingly obvious choice for mitigating this problem would be to use oblivious amplitude amplification [Ber+14]. However, in our case, the open-system character of the simulation and the fact that the abstract angle to be amplified is not independent of the (unknown) input state are major roadblocks for this approach, which we leave up to future research. Instead, we propose the following approach towards bounded success probability. Inspired by the nanoreactor approach in [Wan+14] and the mergo-association scheme from [Rut+23; BLH23; BH24], we introduce artificial potentials that confine the products to be combined in each scattering step, as well as an additional photonic field as energy source and a bath as energy sink. Following the procedure we propose in Section 2.2.4, mergo-association shows more promise in terms of suitability for simulation on the molecular scale at this point as we can confine individual, small systems without the need for large ensembles in high-pressure and high-temperature regimes. Using this framework, we show that,

heuristically, we can expect a constant lower bound $P$ on the probability of success for certain types of bonds for the chemical formation of reactants. Such a lower bound is an important assumption in the procedure and also highlights its heuristic nature. For the example of mergo-association, we show that the existence of a constant lower bound on the success probability for certain types of bonds, such as covalent bonds, is reasonable; cf. Section 2.2.4. The intuition we observe there is that given a certain order of magnitude of bonding energy (say, the typical regime of covalent bonds of approximately $30 - 180\ \frac{\text{kcal}}{\text{mol}}$), then a requirement of simulating potentials of similar magnitude may lead to a substantial constant factor in the simulation cost, but the scaling with respect to system size of this cost would heuristically be only polynomial in the combined nuclear masses, not exponential.

Suppose the scattering is organized as a binary (or higher-order, ternary, ...) tree. For each node in the tree, there is a simulation channel $\mathcal{E}(\rho)$, parametrized by the subsystem size, dissipation model (bath), and confinement properties (artificial potentials). Then, we may assume that this channel produces a state of the kind

$$\mathcal{E}(\rho_0) = p_0\rho_{\text{suc}} + (1 - p_0)\rho_{\neg\text{suc}} + C, \tag{2.2.1}$$

with given input $\rho_0$ and probability of success $p_0$; $C$ describes any potential coherence between the subspace of successful and unsuccessful scatterings and comes from overlap terms such as $\text{tr}[\rho_{\text{suc}}^\dagger \rho_{\neg\text{suc}}]$. We may similarly assume we have an observable $\mathbb{O}_{\text{suc}}$ that allows us to distinguish between the subspaces spanned by $\rho_{\text{suc}}$ and $\rho_{\neg\text{suc}}$ by weak measurements, which enables the heralding of the scattering success and projecting the state into the successful subspace. The presence of a coherence $C$ does not influence measurement outcomes of $\text{tr}[\mathbb{O}_{\text{suc}}\mathcal{E}(\rho_0)]$; see Appendix B.2.3. Using these two building blocks, we will now outline the simulation strategy of each node, as depicted in Fig. 2.11. The first thing to note is that said measurement of $\mathbb{O}_{\text{suc}}$, even if carried out weakly, could disturb the state such that it does not describe a realistic state encountered in nature anymore after the measurement. However, we assume that the simulation channel $\mathcal{E}(\cdot)$, such as the one devised below through mergo-association, does resemble a process in nature. Thus, it tends to make states decay towards such physically meaningful states, and another application of the channel will, therefore, map us back into a state resembling those in nature,

$$\mathcal{E}(\rho_{\text{suc}}) = p_{\text{suc}}\rho_{\text{suc}}^{\text{nat}} + p_{\neg\text{suc}}\rho_{\neg\text{suc}}^{\text{nat}} + C^{\text{nat}}. \tag{2.2.2}$$

If the molecule is unstable, but the reacted state is desired, we may simply skip this reapplication and proceed. Hence, if measuring $\mathbb{O}_{\text{suc}}$ yields a positive outcome, generating meaningful states with a high success rate seems plausible. Further, we can use the following strategy to ensure success even if $\mathbb{O}_{\text{suc}}$ shows that scattering was unsuccessful. Although we are projected into the unsuccessful subspace, the failure is heralded. Therefore, we can apply another simulation channel $\mathcal{E}'(\cdot)$, which may be slightly modified from $\mathcal{E}(\cdot)$ (e.g., stronger confinement or dissipation), leading to

$$\mathcal{E}(\rho_{\neg\text{suc}}) = p_1\rho'_{\text{suc}} + (1 - p_1)\rho'_{\neg\text{suc}} + C'. \tag{2.2.3}$$

This assumption is similar to that in [DCL24] in that we expect a redistribution of probability toward the state of interest. We can then redo the measurement and iterate between measurements

of $\mathbb{O}_{\text{suc}}$ and (progressively more) modified simulation channels until success occurs. To summarize, we can use the above strategy to create a tree-like sequence of scatterings so that, at each step, the success probability at each of the nodes in the tree is bounded by, say, $1 \geq P > 0$. As we can herald success, at most $O(1/P)$ repetitions are required per node. Since $P$ is promised to be fixed, the number of repetitions per node is $O(1)$. Furthermore, since failure does not require a restart from the leaves of the tree, the complexities of individual nodes straightforwardly add up. Hence, the number of repetitions grows linearly with the number of nodes in the tree, which, with $N$ initial constituents, goes at most as $O(N \log(N))$. In Section 2.2.4 and Appendix B.2, we provide a more detailed discussion of the construction of the weak measurements.

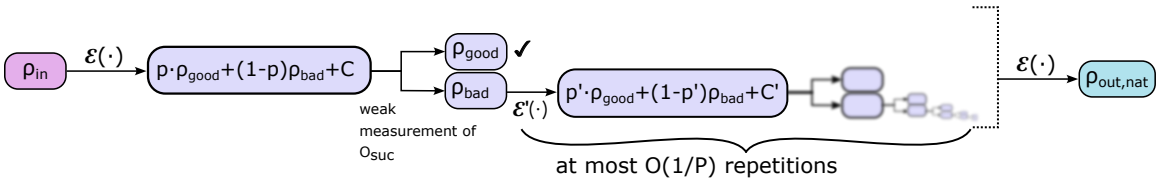



Figure 2.11: Procedure of a single scattering step. Evolution through the simulation channel $\mathcal{E}(\cdot)$ produces states overlapping successful and not successful spaces. To project the state onto one of these subspaces, a weak measurement is performed, yielding either success or failure. In the case of success, proceed and potentially apply another step of time evolution to ensure the state represents a natural state. If unsuccessful, perform an additional, possibly modified, time evolution, which again produces overlap in the successful subspace, then measure again. Repeat this until success, the expected number of repetitions scaling inversely in the lower bound $P$ on the success probability per step. Success can be quantified by a weak measurement of an observable $\mathbb{O}_{\text{suc}}$ that, e.g., signifies the success of forming a bond by capturing spatial proximity.

We leverage parallels to optical tweezers [NN00; MP02] or molecular beams [Cas00; MBM08] that are used in physical experiments to engineer the aforementioned artificial potentials to boost the success probability of scatterings. Before we go into details regarding a specific instance in terms of mergo-association in Section 2.2.4, we first outline more general approaches here. Molecular beams offer well-established success rates, though they require an abundance of particles and, hence, an abundance of available qubits. On the other hand, traps via artificial potentials – e.g., adding to the Hamiltonian a harmonic potential term confined to a specific region as done in Eq. (2.2.11) – can considerably boost success without high additional space requirement. Many of the applications we will discuss benefit from reactant states in the form resembling nuclear wave packets. To that end, consider the input states for the scattering process to be prepared as such. Then, the said process supported by artificial potentials would, in most physical cases, largely preserve the locality in configuration space and maintain the wave-packet like nature of the states. At this point, one may choose to induce a reaction by explicitly modeling a photonic reservoir to excite the reactants [CDG97]. Additionally, an explicit register for a bath can be used to absorb excess energy once the reaction has occurred, allowing the products to relax. Alternatively, instead of explicitly tracing out a subsystem here, post-selection on a 'successful' reaction, as described later in Section 2.2.4, can serve to model energy moving out of the system. Finally, energy dissipation can be implemented using Markovian open-system simulation methods, as discussed in more detail below.

Beyond the scattering approach, one could think of using a molecular Hartree-Fock (HF) state as an alternative heuristic for an initial state for the reaction dynamics under investigation. One can expect the efficacy of this approach to be limited to cases where the Hartree-Fock state as

input to a simulation channel is close enough to the manifold of physically meaningful states so that convergence to a desired state as input to a reaction happens in controllable time, such as in systems with low static correlation. While such systems tend to be amenable to classical treatment, they may be candidates for early experiments of our approach on quantum devices, as it is likely that using an HF initial state has considerably lower constant factors than the scattering approach. In contrast, the scattering approach will apply to more general states (such as those where Hartree-Fock does not provide a sufficient heuristic).

As mentioned above, the embedding processes into a larger environment play an important role in the framework during the molecular preparation stage. The ability to dissipate excess energy is essential for both the probability of successfully forming stable bound states and the ability of the dynamics to emulate the open-system evolution of chemical experiments. Recently proposed methods for simulating the weak coupling to a large, memory-less (i.e., Markovian) environment modelled by a Lindbladian [Lin76; MS72] can be used to model the presence of such a dissipative bath [CW17; CKBG23; DJSW25; PSW23; HW25].. This simulation is efficient in the size of the system and for some methods shown to converge to a thermal state [CKBG23; HW25]. The only parameter of this procedure that does not scale polynomially with system size is the thermalization time, which is difficult to predict and can, in principle, grow exponentially with system size. However, our observation is that slowly thermalizing systems in our simulation correspond to systems that also thermalize slowly in nature. Thus, for physically motivated open-system models, we would expect to produce either thermalized, metastable, or slowly thermalizing states, depending on which ones are prevalent in nature. A good example of these types of systems would be a glassy molecular mixture. From this perspective, we conjecture that polynomial-time simulations are sufficient to reach all chemically relevant states.

One thing to note is that the system on which the readout is to be performed may need to include some degrees of freedom surrounding the molecule, e.g., if solvent effects, photon or phonon couplings, or non-Markovian dynamics [DGB25] are important. The preparation of this more explicit bath follows the same framework as the main molecular degrees of freedom.

### 2.2.4 A Scattering-based State-preparation Step

With a conceptual framework in mind, in this section, we discuss a specific instance of a single scattering step. The approach is based on an external-potential assisted merging of molecular fragments and the construction of a success-heralding measurement oracle. In the first part, we discuss the coherent part of the simulation of the merging. Part of the open-system character of the approach in Fig. 2.10 is brought into play by means of the measurement oracle outlined in the second part of this section.

#### Molecular States by Mergo-Association

A promising approach to realize the assemblage of molecular wavefunctions in Fig. 2.10 is mergo-association, as considered in [Rut+23; BLH23; BH24]. Here, optical tweezers confine two fragments – e.g., two Hydrogen atoms – to form a bond.

To mimic this, we propose to carry out the simulation in the following way. We consider a real-space first-quantized representation the most convenient with oracles for the repeat-until-success

procedure. We closely follow the implementation put forward in [Su+21] to represent the Hamiltonian. However, note that the core building block is open-system simulation, e.g., implemented by combining the methods of [Su+21] with open-system techniques, as proposed in [CKBG23]. Then, we place the two participating systems at a reasonable estimate for a bonding distance. For Dihydrogen, this is well-known; for more involved systems, a heuristic guess needs to be found. Initially, we implement their dynamics according to two independent Hamiltonians, $H_A$ on subsystem $A$ and $H_B$ on subsystem $B$. Then, we slowly turn on inter-system Coulomb interactions $H_{AB} = V^{\text{int}}$ together with trapping potentials $V^{\text{trap}}$, according to

$$H(s) = H_A + H_B + f(s)H_{AB} + g(s)V^{\text{trap}}. \tag{2.2.4}$$

For the example of two nuclei, modeling the trap by a harmonic potential acting on nuclear coordinates $R_1, R_2$, its functional form is given simply by [Rut+23; BLH23; BH24]

$$V^{\text{trap}}(R_1, R_2) = V_1^{\text{trap}}(R_1) + V_2^{\text{trap}}(R_2), \tag{2.2.5}$$

$$V_j^{\text{trap}}(R_j) = \frac{m_j}{2}(R_j - R_{0,j})^T \omega_j^2 (R_j - R_{0,j}), \quad j = 1, 2. \tag{2.2.6}$$

More details on this are in Appendix B.1.1. A qualitative choice of $f, g$ is displayed in Fig. 2.12. After the merging ($s \geq s_0$), the trap is re-released ($s_0 \leq s \leq s_1$), while the interaction stays on. With a certain probability of success, the state will not undergo a diabatic transition into a higher-lying excited state, meaning a successful merger has occurred. The specific design of scheduling functions is left up to further research. However, in order to mimic mergo-association, we propose the use of an $f(s)$ that schedules the inter-species interactions that resembles the trajectory of a Coulomb potential. More details on a quantum encoding of the trap potential can be found in Appendices B.1 and B.1.1.

Equivalently to the adiabatic evolution in $s$, one may put system $A$ and system $B$ at initial distance $z^*$ and move their centers of masses together at a predetermined rate $\frac{\mathrm{d}\Delta z}{\mathrm{d}t}$ until the for bonding length is reached. Hence, the scheduling functions $f(s), g(s)$ are not present in the Hamiltonian anymore, and naturally the strength of the interactions is implicitly determined by distance – recovering the procedure of mergo-association.

The mergo-association scheme we follow relies on assumptions from scattering theory. The setup in scattering theory is that two scattered objects, situated beyond a certain scattering length, can be modeled as free particles. Within the scattering length, the scattered product is considered as 'one'. Conceptually, this is analogous to the 'molecule factory' in Fig. 2.10. The scattering length, or harmonic length, is defined as $\beta = \sqrt{\hbar/(\mu\omega)}$ with a frequency $\omega$ and relative mass $\mu = m_1 m_2/(m_1 + m_2)$ and induces a notion of a bonding energy via $\hbar\omega$. As we are interested in chemical bonds, a proxy for this scattering frequency can be the most weakly bound state or the highest-lying *bonded* vibrational excited state.

The key aspect of such a mergo-association is the choice of trap frequency $\omega$ and the consequences on the diabatic transition probability. Here, we work with an isotropic trap for simplicity, hence Eq. (2.2.6) is described by a single trap frequency $\omega$. Successful, in our case, means that a (e.g., covalent) bond has been formed and that there is no diabatic transition beyond a certain vibrational frequency of the *trapped* system, say $\omega_a$. Then, it is possible to approximate the probability of

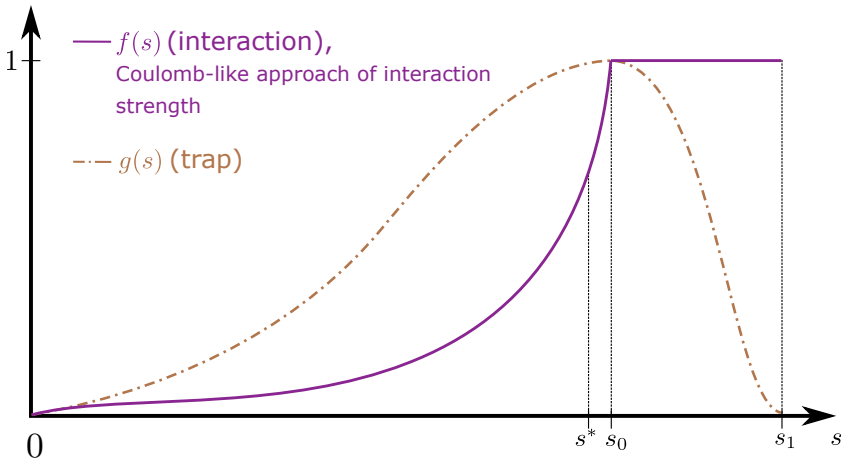



Figure 2.12: Scheduling of interactions. Inter-species interaction is described by $f(s)$ so that $f(0) = 0, f(s \geq s_0) = 1$ and is monotonically increasing until $s_0$ and then constant. The harmonic trap follows $g(s)$ with $g(0) = 0, g(s_0) = 1, g(s_1) = 0$. It is monotonically increasing until $s_0$ and then decreasing until $s_1$. Following adiabatic evolution, there is a 'point of contact' of states at $s^*$ a little earlier than $s_0$, which is the evolution parameter (with corresponding effective distance) used to evaluate the diabatic transition probability. Here, we assume the scenario that the constituents are already placed at closed distance and we 'slowly turn on' the Coulomb interaction, where with the trajectory of $f(s)$, we aim to mimick an evolution of interaction strength that is Coulomb-like if they were to approach each other. To that end, let $z_{0,1}, z_{0,2}$ be the centers of the traps. Then, in order to follow $V^{\text{int}}(R_1(s), R_2(s)) = \frac{q_1 q_2}{|\Delta R_{12}(s)|}$ with nuclear charge $q_j$, let the implemented interaction be $V^{\text{int}}(s) = f(s) H_{AB} = f(s) \frac{q_1 q_2}{|z_{0,1} - z_{0,2}|}$ where via $f(s) = \frac{|z_{0,1} - z_{0,2}|}{|\Delta R_{12}(s)|}$, the desired motion of the nuclei can be emulated. Towards $s \to s_0$, the fixed-strength potential assuming trap centers needs to be replaced by the actual strengths to account for fluctuations in the positions which will matter at that stage.

success by using the Landau-Zener probability [Wit05],

$$p_{\text{suc}} \geq 1 - p_{\text{LZ}}; \qquad p_{\text{LZ}} \approx \exp(-2\pi\Gamma) \quad \text{with} \quad \Gamma = \frac{\omega_{\text{eff}}^2/\hbar}{\left|\frac{\mathrm{d}}{\mathrm{d}s}(E_{\text{mol}} - E_{\text{atom}})\right|} \tag{2.2.7}$$

Here, $\omega_{\text{eff}}^2$ is the effective potential strength of the trapped system (see Appendix B.1.2). Then, we use that the scheduling function $f(s)$ can be related to a corresponding inter-nuclear motion $R(s)$ in a physical mergo-association experiment. Thus, we can rewrite the expression in terms of derivatives of this equivalent position, $\frac{\mathrm{d}E}{\mathrm{d}s} = \frac{\mathrm{d}E}{\mathrm{d}R}\frac{\mathrm{d}R}{\mathrm{d}s}$. More details are in the caption of Fig. 2.12. Defining the velocity $v = \frac{\mathrm{d}R}{\mathrm{d}s}$ and using the shorthand $\partial E_{\text{mol}}, \partial E_{\text{atom}}$ for the energy derivatives, this gives

$$p_{\text{LZ}} \approx \exp\left(-\frac{2\pi}{\hbar}\frac{\omega_{\text{eff}}^2}{|\partial E_{\text{mol}} - \partial E_{\text{atom}}|\, v}\right) \tag{2.2.8}$$

A more detailed discussion can be found in Appendix B.1.2, and we continue by presenting a simplified final expression. We start by the observation in [BLH23, Eq. (54)] that

$$\omega_{\text{eff}}^2 \propto \omega^2 \tilde{E}_a^{1/2} \exp\left(-\tilde{E}_a\right) \tag{2.2.9}$$

with $\tilde{E}_a = \frac{\hbar\omega_a}{\hbar\omega}$ as a 'relative bonding energy'. Together with expressions for energy derivatives from

Eq. (B.1.16) and an approximation to Eq. (2.2.9) we can obtain a bound on $p_{\mathrm{LZ}}$,

$$p_{\mathrm{LZ}} \lessapprox \exp\left(-4\left(\frac{\pi}{\mu}\right)^{1/2}\left(\hbar\omega\frac{\tilde{E}_a}{3+\tilde{E}_a}\right)^{1/2}\frac{\exp\left(-\frac{1}{2}\tilde{E}_a-\frac{3}{2}\right)}{v}\right) \tag{2.2.10}$$

Ideally, we want $p_{\mathrm{LZ}}$ to be close to zero so that $p_{\mathrm{suc}}$ is close to one. Then, up to the speed of evolution $v$ and effective mass $\mu$, this depends on the ratio of binding energy compared to trap energy, $\tilde{E}_a \sim \omega_a/\omega$. The inner exponential is closer to one (which means smaller $p_{\mathrm{LZ}}$) for small $\tilde{E}_a$ and thus for $\omega$ as large as possible relative to $\omega_a$. Beyond this, for fixed $\hbar\omega_a$, the other term depending on frequencies, $\left(\hbar\omega\frac{\tilde{E}_a}{3+\tilde{E}_a}\right)$, is also maximal for large trap strength $\omega$. However, there is a trade-off in maximizing this probability of success versus minimizing the cost factor of the block-encoding of the operator. The latter represents the cost of representing the potential in the block-encoding access model, and is quantified by the operator one-norm of the potential, $\alpha_{\mathrm{trap}}$. Generally, we would aim to keep the frequency no larger than necessary to keep the block-encoding cost low. Additionally, we can look at the system of interest in [Rut+23; BLH23; BH24], namely Rb and Cs. They use $\omega$ =150 kHz for all of their simulations and experiments. This order of magnitude is about the same as the appearing $\omega_a$, which is roughly 110 kHz; the scattering length (regarding binding energy) for this system *in a weakly bonded regime*, is approximately $645\,a_0$ [Tak+12]. This setup allows [Rut+23] to observe high probabilities of success (see [Rut+23, Fig. 4], close to 80 %).

Typical covalent bonds are of the order of 100 $\frac{\mathrm{kcal}}{\mathrm{mol}}$, which amounts to approximately a $35,000\,\frac{1}{\mathrm{cm}}$ frequency for $\omega_a$ (approx. $10^{12}$ kHz). Therefore, we expect the required $\omega$ to be, in general, vastly larger than 150 kHz; however, the hope is that the remaining behavior will carry over, and we only need to scale it according to the relative difference in bonding energy. Then, in fact, $E_a = \hbar\omega_a$ can remain bounded in the two-atom/molecule scatterings and effectively act as a constant in an asymptotic sense. This means for $p_{\mathrm{LZ}}$ to be bounded as well, $\omega$ incurs a (likely substantial) constant factor through $E_a$ and then scales as $O(\mu)$ for $\mu = \frac{m_1 m_2}{m_1+m_2}$. Therefore, since at the last scattering stage in the "molecule factory", $\mu \sim N_{\mathrm{nuc}}$ , we can roughly estimate $\omega$ in the expression of $\alpha_{\mathrm{trap}}$ by a linear factor $N_{\mathrm{nuc}}$ and a system-dependent constant factor for the binding energy of covalent bonds. Using this in our asymptotic expression for the block-encoding factor of the trap,

$$\alpha_{\mathrm{trap}} \approx O({N_{\mathrm{nuc}}}^3\Omega_{\mathrm{trap}}^{2/3}). \tag{2.2.11}$$

We want to compare this result with the corresponding sub-normalization factors of the kinetic energy, $\sim \frac{N_{\mathrm{el}}N^{2/3}}{\Omega^{2/3}}$, where $N$ is the number of particles and $\Omega$ the volume of the simulation box, and the Coulomb energies, $\sim \frac{{N_{\mathrm{el}}}^2 N^{1/3}}{\Omega^{1/3}}$ [Su+21]. If the number of grid points grows with the number of particles, then the encodings of these operators increase roughly with power $\frac{5}{3}$ or $\frac{7}{3}$, whereas the trap scales cubically. It also grows with volume $\Omega^{2/3}$ rather than with the inverse grid density. A more accurate representation will need a finer grid spacing, increasing the cost of representation for the Coulomb potentials. For the trap potential, the box size matters, which will need to increase for larger systems, although there is no increase in cost for increasing accuracy through a finer grid spacing.

### Measurement Oracles for Success Heralding

Suppose now that after a scattering step, we aim to herald success according to Eqs. (2.2.2) and (2.2.3). To that end, we need a measurement operator $\mathbb{O}_{\text{suc}}$ able to give the desired partitioning. A more detailed discussion of the following outline can be found in Appendix B.2.

**Success of merging** First, we start by briefly outlining the success heralding via weak measurements. We are given a quantum state that results from a previous 'scattering' event, such as by simulating the Hamiltonian in Section 2.2.4, which is supported by a trap potential. Such a state will be, following the notion of success that we used before, in a superposition of states that underwent a diabatic transition into higher-lying vibrational states or not. This notion of success does not correspond with one that we can directly observe in a non-destructive manner. Thus, we develop another notion of 'success' that we can use for efficient success heralding to ensure a successful merging. The state we assume is in a first-quantized real-space discretization; this may not be an optimal basis but is most convenient for our discussion, and efficient transformations to other bases can be thought of as well [Bab+18a; BA24]. Then, success can be formulated most straightforwardly using proximity as a geometrical criterion. E.g., suppose we collect a set of locations of interest where we want to test proximity across the nuclei, $\mathcal{I} \subseteq [N_{\text{nuc}}] \times [N_{\text{nuc}}]$, then

$$\tilde{\mathcal{C}}_{\text{geom},\Delta}(\{r,R\},\mathcal{I}) = \begin{cases} 1 & \|R_j - R_k\| \leq \Delta_{jk} \quad \forall (j,k) \in \mathcal{I} \\ 0 & \text{else.} \end{cases} \tag{2.2.12}$$

with $\Delta_{jk} > 0$. We mark success if the criterion $\mathcal{C}$ returns 1. Note that we choose to restrict the evaluation to nuclear coordinates as the relative position of nuclei within a bonded state is essentially stationary compared to the electrons. The definition in Eq. (2.2.12) is classical and assumes direct access to $\{R_j\}_{j\in[N_{\text{nuc}}]}$, whereas types of states we encode are superpositions of tensor products across basis state labels, $|\mathtt{r}_1, \mathtt{r}_2, \cdots, \mathtt{r}_{N_{\text{el}}}; \mathtt{R}_1, \cdots, \mathtt{R}_{N_{\text{nuc}}}\rangle$. More generally, we can think of what happens to a mixed state when going through the success heralding. Then, we consider states of the form

$$\rho = \sum_{p,q} \rho_{p,q} \, |\psi(\{r,R\}_p)\rangle\langle\psi(\{r,R\}_q)| \,, \quad |\psi(\{r,R\}_p)\rangle \sim |\mathtt{r}_1^{(p)}, \mathtt{r}_2^{(p)}, \cdots, \mathtt{r}_{N_{\text{el}}}^{(p)}; \mathtt{R}_1^{(p)}, \cdots, \mathtt{R}_{N_{\text{nuc}}}^{(p)}\rangle. \tag{2.2.13}$$

The set $\{r,R\} = \{r_j\}_{j=1}^{N_{\text{el}}} \cup \{R_j\}_{j=1}^{N_{\text{nuc}}}$ denotes a nuclear and electronic configuration, and $\{r,R\}_j$ denotes an instance as the $p$th term of the superposition. Configurations include a spin degree of freedom. Hence the range of $p,q$ is the number of possible basis states labeling the grid points.

Now, we assume that there is a quantum circuit that can achieve the action of $\mathcal{C}$ efficiently, namely a $U_{\mathcal{C}}$ that takes the state $\rho$ plus an ancilla and stores success in the ancilla, according to the following sets:

$$\begin{aligned} A &= \{j : \mathcal{C}(\{r,R\}_j) = 1\} \\ B &= \{j : \mathcal{C}(\{r,R\}_j) = 0\}. \end{aligned} \tag{2.2.14}$$

The crucial part here is that such a criterion induces a bi-partition of the Hilbert space, splitting the set of states into two groups. Specifically, every state of the form $|\psi(\{r,R\}_j)\rangle$ either corresponds to

a configuration with $\mathcal{C}(\{r, R\}_j) = 1$ or a configuration with $\mathcal{C}(\{r, R\}_j) = 0$. The set $A$ enumerates the states the oracle accepts as merged, and $B$ enumerates the states that the oracle rejects. For convenience we will denote $|\psi_j\rangle := |\psi(\{r, R\}_j)\rangle$ from now on. With this bi-partition, a general input state can now be written as

$$\rho = \sum_{j,k\in A} \rho_{j,k} |\psi_j\rangle\langle\psi_k| + \sum_{j,k\in B} \rho_{j,k} |\psi_j\rangle\langle\psi_k| + \sum_{j\in A, k\in B} (\rho_{j,k} |\psi_j\rangle\langle\psi_k| + \rho_{k,j} |\psi_k\rangle\langle\psi_j|) \tag{2.2.15}$$

Weak measurement of this oracle and thus flagging of a successful merging can be implemented following the subsequent operations based on work in [Lun11; AH22; Yan+22; Miz09]:

1. Append an ancilla qubit, $q_{a,1}$, and perform the unitary $U_{\mathcal{C}}$ on the joint system. This stores the 'success value' in the ancilla.

2. Append a second ancilla, $q_{a,2}$, and perform a controlled rotation $\mathrm{CR}_y(\delta)$ gate on $q_{a,2}$ conditioned on $q_{a,1}$.

3. Perform $U_{\mathcal{C}}$ on the input state and $q_{a,1}$, resetting $q_{a,1}$.

4. Measure $q_{a,2}$ in the computational basis and reset it for later use. The measurement on $q_{a,2}$ then is used as a result flag of a successful merger.

Finally, we can conclude that the measurement oracle $\mathbb{O}_{\mathrm{suc}}$ is composed by the circuit $U_{\mathcal{C}}$ and the weak measurement scheme above. More details are given in Appendix B.2.3.

An important aspect to consider in constructing these measurements is that the wavefunctions encoded, even when expressed as a mixed state, need to satisfy the necessary exchange symmetry (antisymmetric for Fermions, symmetric for Bosons). Ongoing measurement induces the risk of compromising such symmetries. Thus, the measurement needs to be adapted to satisfy that. A detailed discussion is provided in Appendix B.2.2.

**Spin** It is straightforward to append $\mathbb{O}_{\mathrm{suc}}$ by a check whether the obtained state is in the correct spin state. Suppose we have access to an implementation of the time-evolution of the (normalized) spin operator $S^2$. Then, one step of quantum phase estimation allows storing in an ancilla qubit whether there is a singlet (0) or triplet (1) state. Augmentation to higher-order spin states (such as doublets, quadruplets, etc.) follows simply by augmenting the ancilla register of the QPE circuit to represent the appropriate spin numbers. Based on this result, we can accept or reject based on the obtained result and thus effectively have a scheme that projects into the desired spin state. Therefore, this yields an implementation of a projection into the correct spin state; this requires nontrivial knowledge about the expected spin state. One can draw from ideas in [Jac+22] to deal with more complicated ensembles of spin states. A detailed procedure in case of lack of knowledge or intuition about the expected spin state is subject to further research.

We remark that construction of the $S^2$ operator is simple given an encoding as in Eq. (2.2.13), as then it simply translates to its form as Pauli operators acting on the spin degree of freedom.

### 2.2.5 Measuring Dynamical Quantities of Interest and a Review of Exemplary Applications

Simulating the dynamics of molecular processes provides access to meaningful information about the rearrangements of atomic nuclei and electrons and their interactions with electromagnetic fields as they unfold. Already with classical resources, time-dependent approaches offer numerous advantages over time-independent ones, as the former are more amenable to handling the continuum portion of the spectrum and grant access to the relevant elements of the scattering matrix [1] over a meaningful range of energies. We provide an overview of chemical problems, with time-dependent versus stationary quantities, solved on different hardware, in Fig. 2.13.

Measuring observables in a dynamical picture requires considering the time evolution of a wave packet, i.e., a superposition of solutions of the time-dependent Schrödinger equation (TDSE). Most measurements of dynamical quantities can be phrased in terms of wave packet correlation functions, whose calculation fits perfectly into our framework. Transition amplitudes are measured, e.g., using a Hadamard test [Kas+08; Ped+14; KLFK24]. We can follow the scheme introduced in [TCCG20], which is capable of obtaining two-point correlation functions, or the cumulative correlation function method in [LT22]. This scheme is extendable to the S-matrix and to $n$-point correlators by additional time-evolutions and block encodings, similarly to the linear response framework in [KLFK24]. If results at more than two times are required, we can use a history state encoding following the conditional time evolutions of the quantum circuit of quantum phase estimation, similar to the construction in [Dia+23], who also consider observables measured across various time-steps. To that end, we add a clock register $\sum_{\ell=0}^{n_t} |\ell \Delta t\rangle$ so that $n_t \Delta t = t$. Then, instead of a direct time evolution of the overall system after the molecule factory in Fig. 2.10 in the state $\rho_1^{\text{mol}} \rho_2^{\text{mol}} \cdots \rho_M^{\text{mol}}$, we split the evolution into chunks of $\Delta t$ and condition on the clock register to produce a superposition of the state at different evolution time steps. Furthermore, if needed, applying a Quantum Fourier Transform before measurement is straightforward and does not compromise the overall efficiency of the algorithm. While this framework could be interpreted as purely theoretical, we can easily show that this approach covers a vast range of contexts, extending from nature to chemical laboratories.

**Reaction rates.** Chemists are commonly interested in observed kinetics, i.e., reaction rates, a topic inherently suitable for quantum dynamics simulation. For instance, dynamical simulation for nonadiabatic processes such as charge transfer reactions falls into the class of quantum circuits of polynomial complexity [OMT20]. Our framework allows using the measurement schemes proposed in [Kas+08] in which the reaction rate can be computed from the degree of localization of the wave packet in nuclear configuration space corresponding to the formed products. In addition to making the simulation itself achievable, the quantum approach reduces the measurement to a binary search instead of the more complicated evaluation of a time correlation function [Liu+22]. If needed, the latter can also be computed from the dynamics via a rate constant calculated as a function of the scattering cross-sections.

**Photochemistry and photophysics.** Photochemical processes are triggered when a molecule absorbs a photon. In these transformations, electronically excited states become populated, giving

[1] The $S$-matrix has transition elements concerning a process represented by the *evolution* step in Fig. 2.10, independent of the state preparation scheme in the *molecule factory* part.

access to reactive channels that are thermally unachievable [Bhu+23]. The interaction between a molecule and an external photonic field alters the potential energy surface on which wave packets propagate. Therefore, the nuclear and electronic degrees of freedom cannot be decoupled, and the Born-Oppenheimer approximation breaks down [CM18]. The dynamics simulation of these systems on classical computers becomes prohibitive after including very few degrees of freedom. The inclusion of photonic fields in the procedure according to Fig. 2.10 is straightforward by explicitly adding a register for photonic degrees, an explicit bath, or a modified Hamiltonian as in Eq. (2.2.11). Further photophysical processes that can be simulated are, for instance, when spectroscopic measurements do not suffice, ultrafast spectroscopy, which tracks the system during a light-initiated transformation. Other examples are light-harvesting complexes, molecular machines, and photovoltaics; recent work in [Mot+24] also proposed a dynamics-based algorithm for singlet-fission solar cell design. Our approach would make it possible for quantum computers to simulate quantum systems, e.g., optoelectronic devices, where the quantum dynamics are far more important than the eigenstates of the Hamiltonian. In these devices, the relaxation pathways of the excitons are exploited for light generation and harvesting. Classical simulations suffer from the large space of excitons and phonon coupling, making current simulations beyond hopeless. Conversely, simulating the time evolution of thermal quantum states inherently captures all the necessary behavior.

**Linear and non-linear molecular spectroscopy.** Another class of problems that dynamics can solve is the in-laboratory characterization of molecular structures and properties. The absorption spectrum of a molecule is given by the Fourier transform of the wave packet autocorrelation function [Tan07]. By judiciously selecting an initial state, measuring the autocorrelation function can provide us with different spectra, including electronic, vibrational, and rotational [Vit+15]. Through the computation of emission and absorption spectra, our approach accommodates the simulation of fluorescent systems, such as those used in biomolecule marking or thermally activated delayed fluorescence, where forbidden relaxations are assisted by thermal coupling to the environment. Additionally, $n$-time correlation functions allow the exploration of linear-response molecular spectroscopy beyond UV-Vis and fluorescence, such as rotational or vibrational spectroscopy, in which contributions from excited states are typically significant at room temperature. Simulating spectroscopic measurements can be used not only to reproduce experiments but also to probe the simulated quantum system, e.g., the presence of an IR signal may indicate the formation of a molecule, as in the molecular factory in Fig. 2.10. Two-dimensional spectroscopy could also benefit enormously from extracting $n$-time correlation functions from dynamics by following our approach. For instance, two-dimensional infrared (2D IR) spectroscopy reveals second-order vibrational couplings, which characterize molecular interactions [T H09; Nod90]. Classical simulations of this state-of-the-art technique typically exclude anharmonicity and produce errors associated with the BO approximation and vibrational population transfer. In this example and many others, finding the ground state of the system is far removed from reproducing the experimental spectrum.

**Free Energy Simulations.** Free energies play a role in many naturally occurring physical processes, as they determine whether a process occurs spontaneously, such as whether a ligand binds to a protein, whether a material such as salt dissolves in water, or into what shape a protein folds. The type of free energy relevant to a specific system depends on the nature of the system. In the

case of an isothermal, isobaric system that only allows volumetric work, the relevant free energy is the Gibbs free energy. In applications, one is primarily interested in free-energy differences between two different states of the system, which can be characterized by two Hamiltonians, $H_1$ and $H_2$. There exist multiple methods for calculating the free-energy difference between these two states. One class of methods uses fluctuation relations, such as the Jarzinsky equation, that require ensembles of dynamics simulation of the systems of interest [Bas+22]. To estimate the free-energy difference, these methods require evaluating the total energy of each trajectory for both the initial and final states. A time-dependent Hamiltonian $H(\lambda(t))$ is used to alter the system from state 1 to state 2, where $\lambda$ is an externally controlled parameter such that $H(0) = H_1$ and $H(1) = H_2$ (the transition is generally non-adiabatic, and states do not have to be or remain in the ground state such as in adiabatic quantum computing). The number of energy values required depends on the type of problem considered, the speed at which the Hamiltonian is transformed, and the desired accuracy. There are two straightforward ways to calculate free energies in alignment with Fig. 2.10. First, one can create a sufficiently large number of identical systems and perform separate quantum simulations before measuring the total energies of the initial and final states. Alternatively, one can use the clock-register approach from above and simulate a single trajectory. To obtain a sufficiently large number of statistically independent energy values of both initial and final states, one can simulate each state long enough to take measurements with sufficiently large time intervals in between. Once individual energies are measured, the averages and final calculation of the free energy difference can be carried out on a classical computer. This approach is expected to be preferred in terms of quantum and classical resources compared to representing a complete ensemble as long as the variance of single-trajectory estimates is not significantly larger than the joint estimate.

**Quantum Machine Learning.** Instead of directly measuring interesting quantities, the output of the dynamical quantum simulation can be processed by quantum machine learning. This framework can be envisioned as machine learning with quantum input data, with possible classical or quantum outputs [Bia+17; Cer+22]. In particular, recent results indicate that there is a provable advantage in the efficiency of extracting information when given access to multiple copies of a state in a format that a quantum computer can manipulate compared to having access to only measurements performed on the state [Hua+22]. Thus, certain chemically relevant properties of states and quantum evolutions may be learnable more efficiently in our quantum computing framework than in a conventional experiment. Alternatively, it is also possible to measure spectroscopic quantities and replace physical experiments in machine learning pipelines that operate on spectroscopic results, potentially aiding further developments in molecular design [AYSA22; Jou+21].

### 2.2.6 Conclusion and Outlook

We have provided an algorithmic framework that, in principle, can solve a broad set of chemically relevant problems using inherently efficient building blocks. To that end, we considered that, while general ground states are hard to obtain, we may assume to be able to prepare atomic ones as a single-cost effort that can enable a building block library. A scattering process, implemented by simulating dynamics and boosted by artificial potentials, can produce a molecular input state for a subsequent dynamical simulation, which is then followed by the measurement of dynamical quantities. Preparing molecular states via mergo-association is a particularly promising candidate

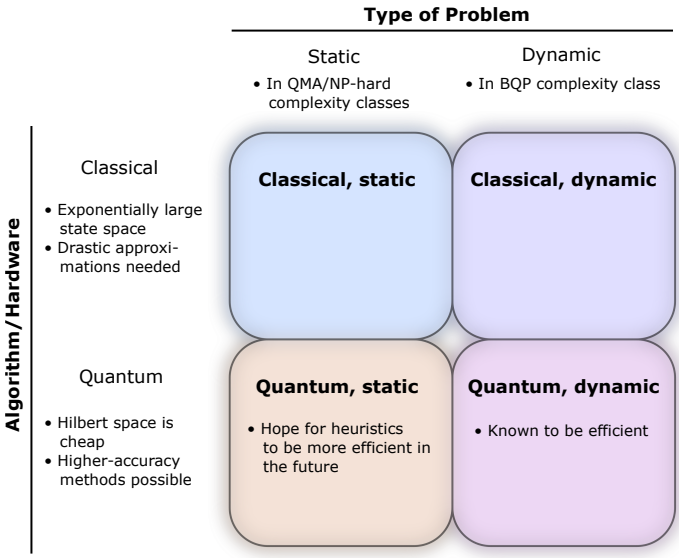



Figure 2.13: Classifying chemical problems related to their hardness and space complexity. Dynamical properties are quantumly efficient, whereas static properties are generally hard. As quantum computers do not suffer from the curse of dimensionality, one can expect the sweet spot of quantum simulations, up to constant factors in the cost, to lie in the evaluation of dynamic properties.

here, which we discussed in detail. We provide examples of applications from, e.g., spectroscopy, photochemistry, and beyond.

Future work remains to build upon this approach and provide a more general and detailed analysis of mergo-association and numerical experiments of this approach, e.g., to gauge the feasibility of the procedure with respect to constant factors and to investigate the more precise costs arising from choosing specific problem instances and methodologies. Another interesting extension would be the inclusion of other modeling tools used in classical simulation, such as the Nosé-Hoover thermostat [Hoo91]. Additionally, an interesting avenue would be to consider additional classical dynamical simulation such as more molecular dynamics (e.g., studied in the context of protein modeling in [LLWL24]), in particular in the case when conservation laws (such as energy conservation) allow direct mappings to Hamiltonian simulation [Bab+23].

A common focal point in quantum chemistry is finding ground state energies, a problem known to be QMA-complete for local Hamiltonians like those seen in molecular systems. We should not restrict ourselves to that perspective, which focuses on a problem known to be hard, and look for paths that allow using dynamics more directly. Additionally, for situations when the ground state is of interest, it may prove useful to give up the search for exact solutions to a hard problem and look at heuristics. The Hartree-Fock problem is known to be NP-hard yet practically efficient, thanks to approximations [WLA12]. Attempts based on open systems dynamics such as in [DCL24; LZL24] may be a promising path towards quantum heuristics for ground states. Nevertheless, we call upon what dynamical quantum simulation offers for chemistry.

# Chapter 3

# Simulating more General Systems with Quantum Computers

So far, we have been considering the application of quantum computers to answer questions related to quantum systems. The research presented in the following chapter relaxes that restriction in the sense that we will consider phenomena described by differential equations in a more general sense. Here, the considered dynamics are more general than linear, unitary dynamics typical for usual quantum systems. Non-unitarity means the norm of encoded solutions is not invariant during the evolution. This poses additional challenges on the algorithms regarding their stability as well as the quantum algorithm's success probability of obtaining the desired result. In addition to that, Section 3.1 considers nonlinear differential equations, based on Carleman linearization [Car32] which allows to approximate the nonlinear dynamics, that cannot be directly encoded into a quantum circuit, by a larger linear dynamical system. Subsequently, Section 3.2 proposes an algorithm that allows to implement constraints on differential equations, such as boundary values for discretized partial differential equations, through a penalty method. Fig. 3.1 sets the upcoming sections into context.

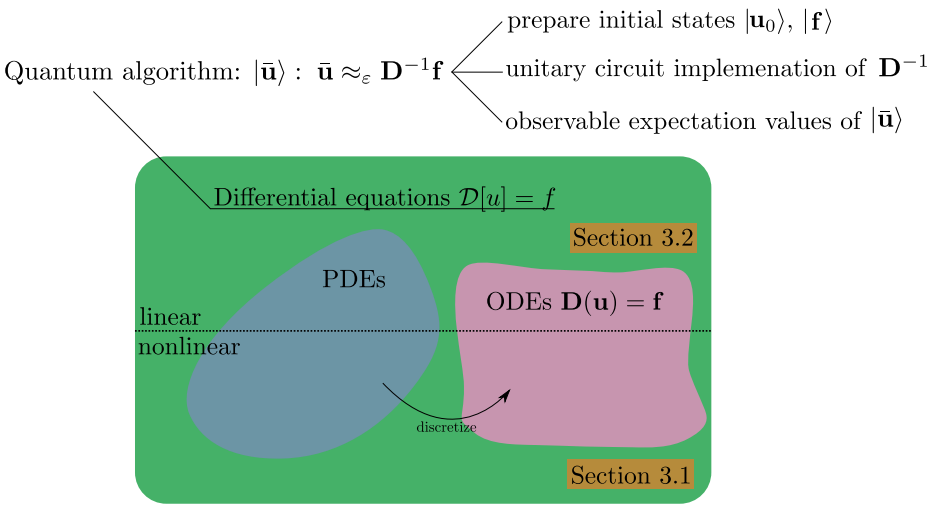



Figure 3.1: Chapter context

## 3.1 Further improving quantum algorithms for nonlinear differential equations via higher-order methods and rescaling

This section is based the following preprint which has been accepted for publication at npj Quantum Information:

> Pedro Costa, Philipp Schleich, Mauro ES Morales, and Dominic W Berry. "Further improving quantum algorithms for nonlinear differential equations via higher-order methods and rescaling". In: *arXiv preprint arXiv:2312.09518* (2023).

Together with my co-authors, I worked on conceptualization, higher-order methods, encoding and algorithm, as well as the writing of the manuscript.

### 3.1.1 Introduction

Many processes in nature exhibit nonlinear behaviour that is not sufficiently approximated by linear dynamics. Examples range from biological systems, chemical reactions, fluid flow, and population dynamics to problems in climate science. Because the Schrödinger equation is linear, quantum algorithms are more naturally designed for linear ordinary differential equations (ODEs), as in [Ber14; BCOW17; BC24; CL20; ALWZ25; FLT23; Kro23; ALL23; ACL23]. These algorithms are normally based on discretising time to encode the ODE in a system of linear equations, then using quantum linear system solvers [HHL09; Cos+22]. Others are based on a time-marching strategy, solving the ODE using a linear combination of unitary dynamics [ALL23; ACL23]. The advantage of these quantum algorithms is that they naturally provide an exponential speedup in the dimension (number of simultaneous equations), similar to the simulation of quantum systems, with the caveat that the solution is encoded in the amplitudes of a quantum state.

The most natural way to approximate quantum solutions of partial differential equations (PDEs) is to first discretise the PDE to construct an ODE, which can then be solved using a quantum ODE algorithm. Although one might expect an exponential speedup in the number of discretization points (which would give the dimension for the ODE), this is not realised. This approach to solve PDEs typically has a more modest polynomial speedup over classical methods due to the norm or condition number of the matrices resulting from the discretization. Reference [CJS13] suggested using preconditioners, though later work found that the preconditioners did not significantly reduce the condition number. Reference [CLO21] approached this problem by using higher-order finite difference stencils as well as a pseudo-spectral method. Alternatively, one can use a wavelet-based preconditioner to achieve scaling independent of the condition number in some cases [BNWA23]. References [JLY24; JLY23a] introduce a new method using a variable transformation which provides solutions of PDEs in an equivalent frame using quantum simulation techniques.

Quantum algorithms for nonlinear differential equations were addressed in early work which had very large complexity [LO08]. Later proposals were based on the nonlinear Schrödinger equation [Llo+20], or an exact mapping of the nonlinear Hamilton-Jacobi PDE into a linear PDE [JL24; JLY23b]. Possibly the most promising approach for the solution of nonlinear ODEs is based on Carleman linearization [Car32], which involves transforming the nonlinear differential equation into a linear differential equation on multiple copies of the vector. This approach can be realised par-

ticularly easily for differential equations with polynomial nonlinearities and has been applied to quantum algorithms in the case of a quadratic function as the nonlinear part of the ODEs [Liu+21], for a higher power of the function for a specific PDE [Liu+23], and for the notorious Navier-Stokes equations [Li+25]. The homotopy perturbation method to tackle quadratic nonlinear equations in Ref. [XXWG22] leads to similar equations as Carleman linearization.

However, most approaches to quantum Carleman linearization [Liu+21; Liu+23] applied to PDEs suffer from high error rates due to simple discretization schemes for the underlying PDE in time and space. One work [Kro23] does use an improved discretization in time via a truncated Taylor series [BCOW17]. Using a finer discretization to achieve a given accuracy results in higher complexity, typically due to the complexity depending on the matrix condition number. That can result in the complexity being the same or worse than that for classical solution. Another difficulty in the use of Carleman linearization in prior work is that the component with the solution may have low probability to be measured. In this study, we provide three improvements over prior work. First, we use higher-order methods in the time evolution as well as for the spatial discretization for PDEs. Second, we use rescaling in order to eliminate the problem of the low probability of the component with the solution for an intrinsic system of ODEs. (Reference [Kro23] mentioned a rescaling at one point, though their explanation is unclear and it is unclear if they are using it.) Third, we provide a tighter bound on the error in Carleman linearization by explicitly bounding repeated integrals. In the case of a PDE, the appropriate stability condition is in terms of the max-norm. However, the interaction of the requirement of the rescaling with the Carleman linearization and the stability requirement for the ODE solver means that a stronger stability criterion is needed to enable efficient solution.

It is important to note that in the case of PDEs the factor that is exponential in $N$ in prior work [Liu+23] would give a large power in the number of grid points. Since a simple classical algorithm would have complexity linear in the number of grid points, the quantum speedup would be eliminated. Our work demonstrates that quantum computers can provide a sublinear complexity in the number of grid points for nonlinear PDEs, as well as establishing the limitations to this type of approach. We present an overview of the general solution procedure of nonlinear differential equations on quantum computers in relation to the present work in Fig. 3.2.

This chapter is organized as follows. We summarize the contributions and significance of the current work in Section 3.1.2, then specify the problem and our method of solution in Section 3.1.3. We then describe the Carleman linearization techniques in Section 3.1.4, with a summary in Section 3.1.4. Section 3.1.4 explains how rescaling the solution vector in Carleman linearization can boost the amplitude for the solution. We provide improved error bounds for the rescaled case in Section 3.1.4. Subsequently, we apply the ODE solver from Ref. [BC24] in Section 3.1.5 to improve over the forward Euler method, providing logarithmic as opposed to linear dependency on the error, and linear as opposed to quadratic dependency on the simulation time. We also provide complexities adjusted to the proposed rescaling. We analyse the case of time-independent quantities, though the approach can also be applied for time-dependent equations. We then show how we can apply our results to obtain the solution of a specific case of nonlinear PDEs in Section 3.1.6, explaining the limitations. Finally, we conclude in Section 3.1.7.

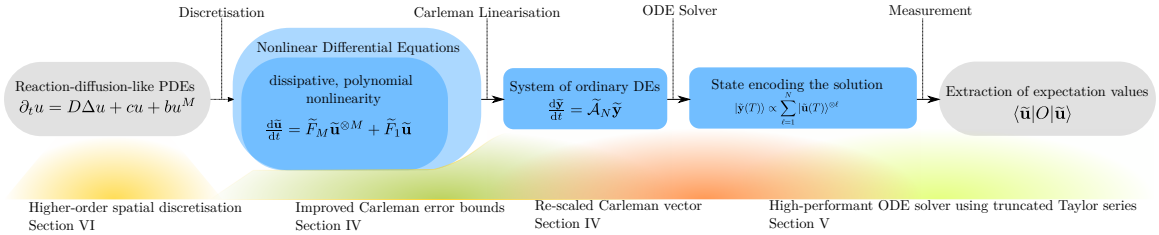



Figure 3.2: Overview of solution pipeline for nonlinear (partial) differential equations in relation to presented contributions and location in paper. The PDE is approximated by a vectorial ODE via discretization, $u \longrightarrow \mathbf{u}$ and we use a scaling $\mathbf{u} \xrightarrow{1/\gamma} \widetilde{\mathbf{u}}$ as described in Def. 3.1.1.

### 3.1.2 Contribution and significance

We give a number of improvements to the solution of nonlinear ODEs and PDEs.

1. We use a higher-order method for discretization of the PDE, which will be required in practice because the stability of the solution will require that the number of points is not too large.

2. We use rescaling of the components in the Carleman linearization in order to ensure that the first component containing the solution can be obtained with high probability. We show that the amount of rescaling that can be used is closely related to the stability of the equations.

3. We provide a much tighter analysis of the error due to the Carleman linearization for ODEs, and extend this analysis to PDEs. This analysis is dependent on the stability and the discretization of the PDE.

All these improvements are dependent on the stability of the equations, which is required for the quantum algorithm to give an efficient solution. The equations have a linear dissipative term and the nonlinear growth term. As the input is made larger, the nonlinear term will cause growth and make the solution unstable. Therefore, for the solution to be dissipative, the input needs to be sufficiently small that the dissipative term dominates. In the ODE case the input is a vector $\mathbf{u}_{\text{in}}$, and the stability criterion can be given in terms of the 2-norm of that vector. In the case of the PDE, it is more appropriate to give the stability criterion in terms of the max-norm, because the 2-norm will change depending on the number of discretization points.

Giving the stability criterion in terms of the max-norm then makes the analysis of higher-order discretizations challenging. The reason is that, while the first-order discretization of the Laplace operator is stable in terms of the max-norm, the higher-order discretizations no longer are. In the analysis of the Carleman linearization error it is required that the equations are stable. For the ODE this stability in terms of the 2-norm enables the 2-norm of the error to be bounded. For the PDE, stability in terms of the max-norm enables the max-norm to be bounded, but the higher-order discretization complicates the analysis and means slightly stronger stability is required.

The reason why rescaling is needed is that the Carleman method involves constructing a quantum state with a superposition of one copy of the initial vector, two copies, and so forth up to $N$ copies. If the initial vector is not normalised, then this means that there can be an exponentially large weight on the largest number of copies, whereas the first part of the superposition with a single copy is needed for the solution. In order to ensure the probability for obtaining that component is not exponentially small, the Carleman vector needs to be rescaled by (at least) the 2-norm so that there

is sufficient weight on that first component. Even if $N$ is small, this feature means that rescaling is essential in order to obtain any speedup over classical algorithms for PDEs. Without the rescaling, the complexity is superlinear in the number of grid points.

In order to ensure that the same equations are being solved, the components of the matrix need a matching rescaling, which can increase the weight of the nonlinear part (causing growth) as compared to the linear dissipative part. In the case of an ODE, we show that if the original nonlinear equation is dissipative then the linear ODE obtained from Carleman linearization is also stable. That stability is required for the quantum ODE solver to be efficient. If the ODE is not stable, then the condition number will be exponentially large (in time), which causes the linear equation solver to have exponential complexity.

Similarly for the discretised PDE, there needs to be rescaling by the 2-norm in order to ensure there is adequate weight on the first component of the solution. The key difference now is that the stability of the equations is given in terms of the max-norm, but the rescaling is by the 2-norm which is typically larger. That rescaling can give a linear ODE that is no longer stable, which in turn would mean an exponential complexity of the algorithm. That is perhaps surprising, because the original nonlinear equation is stable.

However, if the PDE is sufficiently dissipative, then the discretised equation will still satisfy the stability criterion in terms of the 2-norm, and there will still be an efficient quantum algorithm. Because the 2-norm will increase without limit with the number of discretization points, it is then crucial to minimise the number of discretization points used. That further motivates using the higher-order discretization of the PDE, because that minimises the number of discretization points.

### 3.1.3 Problem description and solution strategy

The main focus of this work is the treatment of nonlinear differential equations, when we have an arbitrary power $M$ in the nonlinear ODE problem on quantum computers, that is

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{u} = F_1\mathbf{u} + F_M\mathbf{u}^{\otimes M}\,, \tag{3.1.1}$$

followed by its application to the nonlinear reaction-diffusion PDE,

$$\partial_t u(\mathbf{x},t) = D\Delta u(\mathbf{x},t) + cu(\mathbf{x},t) + bu^M(\mathbf{x},t). \tag{3.1.2}$$

In the following we summarise the problem description and solution strategy for these two equations.

#### The ODE problem

Here, we present the problem of solving the nonlinear ODE, including the variable definitions and the dissipativity condition needed for an efficient quantum algorithm.

**Problem 1.** *We consider the solution of a system of nonlinear (vectorial) dissipative ODEs of the form*

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = F_1\mathbf{u} + F_M\mathbf{u}^{\otimes M}\,, \tag{3.1.3}$$

*with initial data*

$$\mathbf{u}(t=0) = \mathbf{u}_{\mathrm{in}}, \tag{3.1.4}$$

*where* $\mathbf{u} = (u_1 \cdots, u_n)^T \in \mathbb{R}^n$ *with time-dependent components* $u_j = u_j(t)$ *for* $t \in [0, T]$ *and* $j \in [n]$, *using the notation* $[n] = \{1, 2, \ldots, n\}$. *The matrices* $F_M \in \mathbb{R}^{n \times n^M}$, $F_1 \in \mathbb{R}^{n \times n}$ *are time-independent. We denote the eigenvalues of* $(F_1 + F_1^\dagger)/2$ *by* $\lambda_j$, *and the dissipativity condition means that* $\lambda_j < 0$. *Denoting the maximum eigenvalue by* $\lambda_0$, *we require that* $R < 1$, *where*

$$R \coloneqq \frac{\|F_M\| \cdot \|\mathbf{u}_{\text{in}}\|^{M-1}}{|\lambda_0|}. \tag{3.1.5}$$

*The task is to output a state* $|\mathbf{u}\rangle$ *encoding the solution to Eq.* (3.1.3) *at time* $T$.

To solve Def. 1, we first map the finite-dimensional system of nonlinear differential equations in Eq. (3.1.3) to an infinite-dimensional, linear set of ODEs that can be truncated to some order $N$. This mapping is the Carleman linearization technique [Car32], which has previously been applied to quantum algorithms in Refs. [Liu+21; Liu+23; Kro23]. Next, we show that by rescaling the linearised ODEs, we can reduce the complexity of the quantum algorithm. This is followed by improved error bounds due to Carleman linearization for the rescaled variable and an estimate of the overall complexity for obtaining the solution of the truncated linearised ODE.

In contrast to Refs. [Liu+21; Kro23], we do not consider the driving term; on the other hand, we explore arbitrary nonlinear powers in the ODE problem rather than constrained to the quadratic case as in Refs. [Liu+21; Kro23]. When we have an arbitrary power $M$ in the nonlinear ODE it is more challenging to include the driving term $F_0$, because $F_0$ will produce characteristics of a more general polynomial of order $M$ as opposed to just a single component. Therefore, to analyse the driving term we would also need to consider a general polynomial of order $M$ for the nonlinear part of the ODE problem. We leave that considerably more complicated analysis to future work.

The solution of a linearised form of Def. 1 relies on oracles for $F_1$, $F_M$, and the initial vector. We show in Section 3.1.5, that the complexity of the solution in terms of calls to oracles for $F_1$ and $F_M$ scales as

$$O\left(\frac{1}{\sqrt{1 - R^{2/(M-1)}}} \frac{\|\mathbf{u}_{\text{in}}\|}{\|\mathbf{u}(T)\|} \lambda_{F_1} T N \log\left(\frac{N}{\varepsilon}\right) \log\left(\frac{N \lambda_{F_1} T}{\varepsilon}\right)\right). \tag{3.1.6}$$

In this complexity, $\varepsilon$ is the allowable error, and $\lambda_{F_1}$ is the $\lambda$-value for block encoding $F_1$ (with an extra assumption on the efficiency of the block encoding of $F_M$). An important quantity here is the Carleman order $N$, which can be chosen logarithmically in the allowable error provided $R < 1$. For the complexity in terms of calls to the preparation of the initial vector, there is an extra factor of $N$, but the final log factor can be omitted, so the overall complexity is similar. Without the rescaling, there would be an extra factor in the complexity $O\left(\|\mathbf{u}_{\text{in}}\|^N\right)$ that is exponential in the aforementioned Carleman truncation number $N$. Even though $N$ can be chosen logarithmic in the other parameters, that would still result in large complexity.

The result as given in Ref. [Liu+23] has that problem. The complexity from Ref. [Liu+23] is (using Eq. (4.2) of that work and replacing $a$ in their notation with $c$ in our notation)

$$O\left(\frac{1}{G^2 \varepsilon} s T^2 D^2 d^2 n^{4/d} N^3 \|\mathbf{u}_{\text{in}}\|^{2N} \text{poly}\left(\log\left(\frac{cDdMn^{1/d} N s T}{G\varepsilon}\right)\right)\right), \tag{3.1.7}$$

where $G$ denotes the average $\ell_2$ solution norm of the history state, and $s$ is the maximum sparsity of $F_1$, $F_M$. The factor $\|\mathbf{u}_{\text{in}}\|^{2N}$ exponential in $N$ is due to the higher-order components of the Carleman vector without rescaling. They also have a factor of $T^2$ rather than $T$, which is due to

using a simple forward Euler scheme in time. We also give a further improvement in the polynomial factor of $N$, with our scaling being $N$ in comparison to their $N^3$.

**Carleman solver for the reaction-diffusion equation**

A large system of ODEs of the form in Eq. (3.1.3) may arise from discretization of partial differential equations. Specifically, we can derive the nonlinear differential equation resulting from the discretization of a nonlinear reaction-diffusion PDE similar to Ref. [Liu+23],

$$\partial_t u(\mathbf{x}, t) = D\Delta u(\mathbf{x}, t) + cu(\mathbf{x}, t) + bu^M(\mathbf{x}, t). \tag{3.1.8}$$

This equation will be stable according to a criterion that depends on the max-norm of $u(\mathbf{x}, t)$, in contrast to the condition for the ODE that is based on the 2-norm. Discretising this PDE into an ODE, the stability condition $R < 1$ would be stronger and depend on the number of discretization points. That condition is stronger than necessary for the PDE, but after we use Carleman linearization to give a linear ODE it requires $R < 1$ for stability. This means that the stability condition needed for the quantum algorithm is stronger than that for the original PDE.

We explore techniques of finite-difference methods with higher-order approximations for the spatial discretization of the PDEs. Our improved nonlinear ODE solver is then applied to the reaction-diffusion equation Eq. (3.1.8), with $F_1$ resulting from the Laplacian discretization and $F_M$ giving the non-linearity from the PDE. The overall procedure is illustrated in Fig. 3.2.

We then show in Corollary 3.1.6 that for this PDE, the overall cost for the solution in terms of calls to the oracles that block encode $F_1$ and $F_M$ is

$$O\left(\frac{1}{\sqrt{1-R^{2/(M-1)}}}\frac{\|\mathbf{u}_{\text{in}}\|}{\|\mathbf{u}(T)\|}(dDn^{2/d}+|c|)TN\log\left(\frac{N}{\varepsilon}\right)\log\left(\frac{N(dDn^{2/d}+|c|)T}{\varepsilon}\right)\right), \tag{3.1.9}$$

where we have used $n$ gridpoints in total for the spatial discretization of the $d$-dimensional PDE given in Eq. (3.1.8).

Classically, it is less useful to perform linearization by the Carleman procedure, because the system size grows exponentially with the truncation number $N$ making the simulation prohibitively costly. In general, explicit time-stepping methods like forward Euler or Runge-Kutta schemes do not rely on linearization of the underlying differential equations. However, (semi-)implicit schemes which exhibit more favourable numerical stability rely on inversion of the system. This either requires linearization (e.g., Carleman or Koopman-von-Neumann schemes) or methods to solve nonlinear systems, such as Newton-Raphson, which rely on a good initial guess and require inversion of a Jacobian matrix.

### 3.1.4 Quantum Carleman solver with rescaling and improved error bounds on Carleman truncation

**Background on Carleman linearization**

We start with the Carleman linearization for the initial value problem described by the $n$-dimensional equation with a nonlinearity of order $M$ as given in Eq. (3.1.3). We recall the dissipativity assumption on $F_1$, i.e., the eigenvalues of $(F_1+F_1^\dagger)/2$ are purely negative. The quantity $\lambda_0$, the eigenvalue closest

to zero, thus gives the weakest amount of dissipation. This way, $R$ in Eq. (3.1.5) can be used to quantify the strength of the nonlinearity of the problem. As shown in Ref. [Liu+21], there exists a quantum algorithm that can solve Eq. (3.1.3) efficiently whenever $R < 1$. Furthermore, for $R \geq \sqrt{2}$, the problem was shown to be intractable on quantum computers.

Next, we briefly outline the key idea of the Carleman linearization. First, notice that

$$\mathbf{u}^{\otimes M} = (u_1^M, u_1^{M-1}u_2, \ldots, u_1u_n^{M-1}, u_2u_1^{M-1}, \ldots, u_n^{M-1}u_{n-1}, u_n^M)^T \in \mathbb{R}^{n^M}. \tag{3.1.10}$$

In particular, for $M = 2$, the Kronecker product gives

$$\mathbf{u}^{\otimes 2} = (u_1^2, u_1u_2, \ldots, u_1u_n, u_2u_1, \ldots, u_nu_{n-1}, u_n^2)^T \in \mathbb{R}^{n^2}. \tag{3.1.11}$$

Now, define a new variable consisting of Kronecker powers of the solution vector

$$\mathbf{y}_1 = \mathbf{u},\ \mathbf{y}_2 = \mathbf{u}^{\otimes 2},\ \ldots,\ \mathbf{y}_N = \mathbf{u}^{\otimes N}, \ldots, \tag{3.1.12}$$

which we can summarise as a vector $\mathbf{y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N, \ldots]^T$. If we consider the time-derivative, we can identify the time-independent matrices $F_M \in \mathbb{R}^{n \times n^M}$ and $F_1 \in \mathbb{R}^{n \times n}$ as follows,

$$\begin{aligned}
\frac{\mathrm{d}\mathbf{y}_j}{\mathrm{d}t} &= \frac{\mathrm{d}\mathbf{u}^{\otimes j}}{\mathrm{d}t} \\
&= \frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} \otimes \mathbf{u} \otimes \cdots \otimes \mathbf{u} + \cdots + \mathbf{u} \otimes \mathbf{u} \otimes \cdots \otimes \frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} \\
&= (F_M\mathbf{u}^{\otimes M}) \otimes \mathbf{u} \otimes \cdots \otimes \mathbf{u} + \cdots + \mathbf{u} \otimes \mathbf{u} \otimes \cdots \otimes (F_M\mathbf{u}^{\otimes M}) \\
&\quad + (F_1\mathbf{u}) \otimes \mathbf{u} \otimes \cdots \otimes \mathbf{u} + \cdots + \mathbf{u} \otimes \mathbf{u} \otimes \cdots \otimes (F_1\mathbf{u}).
\end{aligned} \tag{3.1.13}$$

We can write this in compact form,

$$\frac{\mathrm{d}\mathbf{y}_j}{\mathrm{d}t} = A_{j+M-1}^{(M)}\mathbf{y}_{j+M-1} + A_j^{(1)}\mathbf{y}_j, \tag{3.1.14}$$

where $A_{j+M-1}^{(M)} \in \mathbb{R}^{n^j \times n^{j+M-1}}$ and $A_j^{(1)} \in \mathbb{R}^{n^j \times n^j}$ with

$$\begin{aligned}
A_{j+M-1}^{(M)} &= F_M \otimes \mathbb{I}^{\otimes(j-1)} + \mathbb{I} \otimes F_M \otimes \mathbb{I}^{\otimes(j-2)} + \cdots + \mathbb{I}^{\otimes(j-1)} \otimes F_M \\
A_j^{(1)} &= F_1 \otimes \mathbb{I}^{\otimes(j-1)} + \mathbb{I} \otimes F_1 \otimes \mathbb{I}^{\otimes(j-2)} + \cdots + \mathbb{I}^{\otimes(j-1)} \otimes F_1,
\end{aligned} \tag{3.1.15}$$

where the $\mathbb{I}$ operation is the identity with the same domain as $F_1$, i.e., $\mathbb{R}^{n \times n}$.

This results in an infinite-dimensional linear system, as there is no bound on the range of $j$. To make this computationally feasible, we restrict to $j \in [N]$ for some $\mathbb{N} \ni N > M$. Further, we can see that $N > M$ is a requirement in order to be able to capture any effects coming from a nonlinearity of order $M$. This allows one to write down a matrix form,

$$\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}t} = \mathcal{A}_N\mathbf{y}, \tag{3.1.16}$$

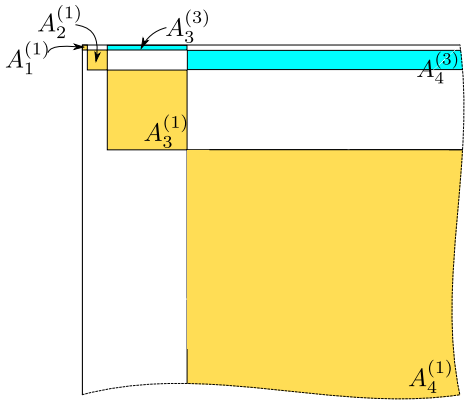



Figure 3.3: Depiction of a snippet of $\mathcal{A}_N$ for $M = 3$ until $N = 4$. Given the exponential increase in size, only a fraction of $N = 4$ is shown. The diagonal blocks correspond to linear terms of the ODE, the upper-diagonal blocks to a nonlinearity on the $(M-1)$st off-diagonal.

with

$$\mathcal{A}_N = \begin{bmatrix} A_1^{(1)} & 0 & \cdots & 0 & A_M^{(M)} & 0 & \cdots & 0 \\ 0 & A_2^{(1)} & \cdots & 0 & 0 & A_{M+1}^{(M)} & 0 & \vdots \\ \vdots & 0 & \ddots & & & 0 & \ddots & 0 \\ & & \ddots & \ddots & & & \ddots & A_N^{(M)} \\ & & & & \ddots & & & 0 \\ & & & & \ddots & \ddots & & \vdots \\ \vdots & & & & & 0 & A_{N-1}^{(1)} & 0 \\ 0 & 0 & \cdots & & & \cdots & 0 & A_N^{(1)} \end{bmatrix}. \tag{3.1.17}$$

The matrix $\mathcal{A}_N \in \mathbb{R}^{N_{\text{tot}} \times N_{\text{tot}}}$ is called the Carleman matrix with truncation order $N$, where $N_{\text{tot}} = \sum_{j=1}^{N} n^j = \frac{n(n^N-1)}{n-1}$. The non-truncated, infinitely large matrix we call $\mathcal{A}$. As the dimensionality of the system is exponential in the order of Carleman truncation (see Fig. 3.3), this technique tends to be intractable for practical applications on classical computers.

The simple block structure of the matrix $\mathcal{A}_N$ enables us to obtain the upper bound for $\|\mathcal{A}_N\|$ in terms of the norms of the submatrix of $\mathcal{A}_N$, that is

$$\begin{aligned} \|\mathcal{A}_N\| &\leq \max_{1\leq j\leq N} \|A_j^{(1)}\| + \max_{1\leq j\leq N+M-1} \|A_j^{(M)}\| \\ &= N\|F_1\| + (N-M+1)\|F_M\|\,. \end{aligned} \tag{3.1.18}$$

A similar relation holds for the $\lambda$-values, which is important for the estimation of the complexity of our quantum algorithm. In what follows, we present a lemma that allows us to quantify the total error involved in the Carleman truncation. Our lemma considers the error from the Carleman linearization for the rescaled nonlinear ODE problem when we have an arbitrary power $M$ for the function, as opposed to the quadratic case without the rescaling given in [Liu+21]. To that end, we will first present said rescaling.

### A rescaled Carleman solver

We will motivate this rescaling by looking at the measurement probabilities of components in the vector $\mathbf{y} = [\mathbf{u}, \mathbf{u}^{\otimes 2}, \ldots, \mathbf{u}^{\otimes N}]^T$. Recall that the sole entry we are interested in measuring will be $\mathbf{y}_1 \equiv \mathbf{u}$. The standard way to encode the solution $\mathbf{u}(t)$ in a computational basis $\{|j\rangle\}$ is

$$|\mathbf{u}(t)\rangle = \sum_{j=1}^{n} u_j(t)\,|j\rangle\,. \tag{3.1.19}$$

Analogously, components $|\mathbf{y}_m\rangle$ of $\mathbf{y}$ are written as a quantum state as

$$\begin{aligned} |\mathbf{y}_m(t)\rangle &= \sum_{j_1,\ldots,j_m=1}^{n} u_{j_1}(t)\cdots u_{j_m}(t)\,|m, j_1\cdots j_m, 0^{\otimes \log(n)(N-m)}\rangle \\ &= \sum_{j_1,\ldots,j_m=1}^{n} u_{j_1}(t)\cdots u_{j_m}(t)\,|y_m^{(j_1 j_2\cdots j_m)}\rangle\,, \end{aligned} \tag{3.1.20}$$

with

$$|y_m^{(j_1 j_2\cdots j_m)}\rangle \coloneqq |m, j_1\cdots j_m, 0^{\otimes \log(n)(N-m)}\rangle\,. \tag{3.1.21}$$

This follows the state encoding outlined in Appendix 3.C in Ref. [Liu+21], where in each step up to the largest order $N$, extra dimensions are padded in the form of $|0\rangle$'s to avoid the structure of a superposition over components of different size. The first register is set to $m$ so we can distinguish the order by measurement of a subsystem. Then, we can write the full vector $|\mathbf{y}(t)\rangle$ as follows:

$$|\mathbf{y}(t)\rangle = \sum_{j_1=1}^{n} u_{j_1}(t)\,|y_1^{(j_1)}\rangle + \sum_{j_1,j_2=1}^{n} u_{j_1}(t)u_{j_2}(t)\,|y_2^{(j_1 j_2)}\rangle + \cdots + \sum_{j_1,\ldots,j_N=1}^{n} u_{j_1}(t)\ldots\bar{u}_{j_N}(t)\,|y_N^{(j_1 j_2\cdots j_N)}\rangle\,. \tag{3.1.22}$$

For a normalised quantum state, the amplitudes $u_{j_l}(t)$ in Eq. (3.1.22) need to be normalised so that $\langle\mathbf{y}\rangle = 1$. We then have to consider the normalization factor $1/\sqrt{V_N}$ where

$$V_N = \|\mathbf{y}\|^2 = \sum_{\ell=1}^{N} \|\mathbf{u}(t)\|^{2\ell} = \|\mathbf{u}(t)\|^2 \frac{1-\|\mathbf{u}(t)\|^{2N}}{1-\|\mathbf{u}(t)\|^2}\,. \tag{3.1.23}$$

Note that this formula does not work in the case that $\|\mathbf{u}(t)\| = 1$. We therefore adopt the convention that wherever there appears a ratio of this form, for $\|\mathbf{u}(t)\| = 1$ it takes the value in the limit $\|\mathbf{u}(t)\| \to 1$, so

$$\|\mathbf{u}(t)\|^2 \frac{1-\|\mathbf{u}(t)\|^{2N}}{1-\|\mathbf{u}(t)\|^2} \to N\,. \tag{3.1.24}$$

The solution of the nonlinear ODE is given by the first component, where the probability is given by

$$P(\mathbf{y}_1(t)) = \sum_{j_1=1}^{n} \left|\langle y_1^{(j_1)}\rangle\,\mathbf{y}(t)\right|^2 = \frac{1}{V_N}\sum_{j_1=1}^{n} |u_{j_1}(t)|^2 = \frac{1-\|\mathbf{u}(t)\|^2}{1-\|\mathbf{u}(t)\|^{2N}}. \tag{3.1.25}$$

From this equation, we see that as we increase the Carleman truncation order we also increase $V_N$, which suppresses the probability of extracting the desired component. This brings an exponential cost in $N$ for the algorithm due to the $O\left(1/\sqrt{P(\mathbf{y}_1(t))}\right)$ rounds of amplitude amplification needed

at the end. To avoid this high cost in the algorithm, we propose the following rescaling, which can significantly reduce the cost of amplitude amplification.

**Definition 3.1.1** (Rescaled Carleman problem)**.** *Consider a nonlinear ODE system of the form* $\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = F_1\mathbf{u} + F_M\mathbf{u}^{\otimes M}$ *as in Def. 1. Then, using a variable transformation in the form of a rescaling* $\widetilde{\mathbf{u}} = \mathbf{u}/\gamma$ *with* $\gamma > 0$*, we obtain another system in the rescaled variable*

$$\frac{\mathrm{d}\widetilde{\mathbf{u}}}{\mathrm{d}t} = \widetilde{F}_1\widetilde{\mathbf{u}} + \widetilde{F}_M\widetilde{\mathbf{u}}^{\otimes M}, \tag{3.1.26}$$

*with* $\widetilde{F}_1 = F_1$ *and* $\widetilde{F}_M = \gamma^{M-1}F_M$*.*

This allows us to improve the measurement probability in the following sense.

**Lemma 3.1.2** (Measurement probability of the rescaled Carleman problem)**.** *Using the rescaling in Def. 3.1.1, using a scaling factor* $\gamma \geq \|\mathbf{u}_{\mathrm{in}}\|$ *and assuming dissipativity of the ODE, the probability to measure* $\widetilde{\mathbf{u}} = \widetilde{\mathbf{y}}_1$ *is given by*

$$P(\widetilde{\mathbf{y}}_1(t)) = \frac{1 - \frac{\|\mathbf{u}(t)\|^2}{\gamma^2}}{1 - \left(\frac{\|\mathbf{u}(t)\|}{\gamma}\right)^{2N}} \geq \frac{1}{N}. \tag{3.1.27}$$

*Proof.* Using the rescaling $\gamma > 0$, we obtain a new normalization

$$\widetilde{V}_N = \sum_{l=1}^{N}\left(\frac{\|\mathbf{u}(t)\|}{\gamma}\right)^{2l} = \|\widetilde{\mathbf{u}}(t)\|^2\frac{1 - \|\widetilde{\mathbf{u}}(t)\|^{2N}}{1 - \|\widetilde{\mathbf{u}}(t)\|^2}, \tag{3.1.28}$$

with $\|\widetilde{\mathbf{u}}(t)\| = \|\mathbf{u}(t)\|/\gamma$. Given dissipativity of the ODE, we have $\|\mathbf{u}(t)\| \leq \|\mathbf{u}_{\mathrm{in}}\|$, so $\|\widetilde{\mathbf{u}}(t)\| \leq 1$. In turn that implies

$$\frac{1 - \|\widetilde{\mathbf{u}}(t)\|^{2N}}{1 - \|\widetilde{\mathbf{u}}(t)\|^2} \leq N. \tag{3.1.29}$$

The measurement probability to obtain $\widetilde{\mathbf{y}}_1(t)$ is then

$$P(\widetilde{\mathbf{y}}_1(t)) = \sum_{j_1=1}^{n}\left|\langle y_1^{(j_1)}\rangle\,\widetilde{\mathbf{y}}(t)\right|^2 = \frac{\|\widetilde{\mathbf{u}}(t)\|^2}{\widetilde{V}_N} = \frac{1 - \|\widetilde{\mathbf{u}}(t)\|^2}{1 - \|\widetilde{\mathbf{u}}(t)\|^{2N}} \geq \frac{1}{N}. \tag{3.1.30}$$

□

Therefore, using the parameter $\gamma$, we can adjust the probability to obtain $\widetilde{\mathbf{y}}_1$. Here we have taken $\gamma \geq \|\mathbf{u}_{\mathrm{in}}\|$, though the first expression does not depend on this assumption. The probability is equal to $1/N$ if $\gamma = \|\mathbf{u}_{\mathrm{in}}\| = \|\mathbf{u}(t)\|$, and otherwise for $\gamma > \|\mathbf{u}(t)\|$ the probability is even better. Thus the rescaling avoids the exponential (in $N$) suppression of the probability of obtaining the component of interest of the ODE problem, which occurs for $\|\mathbf{u}(t)\| > 1$ without rescaling.

When we apply the rescaling above into Eq. (3.1.15) we obtain a linearised system in the rescaled solution vector with $\widetilde{A}_j^{(1)} = A_j^{(1)}$ and $\widetilde{A}_{j+M-1}^{(M)} = \gamma^{M-1}A_{j+M-1}^{(M)}$, and as a result we can write the rescaled Carleman linearization as

$$\frac{\mathrm{d}\widetilde{\mathbf{y}}}{\mathrm{d}t} = \widetilde{\mathcal{A}}_N\widetilde{\mathbf{y}}, \tag{3.1.31}$$

where

$$\widetilde{\mathcal{A}}_N = \begin{bmatrix} A_1^{(1)} & 0 & \cdots & 0 & \gamma^{M-1}A_M^{(M)} & 0 & \cdots & 0 \\ 0 & A_2^{(1)} & \cdots & 0 & 0 & \gamma^{M-1}A_{M+1}^{(M)} & 0 & \vdots \\ \vdots & 0 & \ddots & & & 0 & \ddots & 0 \\ & & \ddots & \ddots & & & \ddots & \gamma^{M-1}A_N^{(M)} \\ & & & & \ddots & & & 0 \\ & & & & \ddots & \ddots & & \vdots \\ \vdots & & & & & 0 & A_{N-1}^{(1)} & 0 \\ 0 & 0 & \cdots & & & \cdots & 0 & A_N^{(1)} \end{bmatrix}. \tag{3.1.32}$$

We discuss the cost of an implementation of the rescaled dynamics in Section 3.1.5.

### Error bounds on rescaled solution

Next, we present error bounds on the global and component-wise errors due to Carleman linearization in Lemma 3.1.3 and Lemma 3.1.4, where we make use of the rescaling technique outlined in the previous section. The first lemma provides a bound on the overall error in the Carleman vector. The error bounds we present here are based on the 2-norm.

**Lemma 3.1.3** (Global rescaled Carleman error)**.** *Consider the ODE from Eq.* (3.1.3) *with its Carleman linearization in Eq.* (3.1.13) *truncated at order $N$. Let $F_1$ be dissipative, so that for $\lambda_0 < 0$ with $|\lambda_0| > \|\mathbf{u}_{\text{in}}\|^{M-1}\|F_M\|$ and therefore $\|\widetilde{\mathbf{u}}_{\text{in}}\| \geq \|\widetilde{\mathbf{u}}(t)\|$ for $t > 0$. Then, the error in the rescaled solution as defined in Lemma 3.1.2 is given by $\eta_j = \widetilde{\mathbf{u}}^{\otimes j} - \widetilde{\mathbf{y}}_j$ at order $j \in [N]$ due to Carleman truncation $N > M \geq 2$ and a scaling factor $\gamma = \|\mathbf{u}_{\text{in}}\|$; $\widetilde{\mathbf{u}}$ denotes the exact solution to the underlying ODE whereas $\widetilde{\mathbf{y}}$ is the approximation due to Carleman truncation. Then, this error for any $j \in [N]$ is upper bounded by the overall error vector,*

$$\|\eta_j(t)\| \leq \|\eta(t)\| \leq (M-1)\|F_M\|\|\mathbf{u}_{\text{in}}\|^{M-1}\frac{1 - e^{N(\lambda_0+\gamma^{M-1}\|F_M\|)t}}{|\lambda_0 + \gamma^{M-1}\|F_M\||}. \tag{3.1.33}$$

The detailed proof is presented in Appendix C.2.1. Related results were given in Ref. [Liu+21] and Ref. [Kro23]. Neither included a general power for the nonlinearity, and were restricted to $M = 2$. Furthermore, we provide an exponential reduction in the Carleman order dependence due to the rescaling, i.e., $\|\eta\| \propto \|\mathbf{u}_{\text{in}}\|^M$ in opposed to $\|\mathbf{u}_{\text{in}}\|^N$. Although Ref. [Kro23] mentioned rescaling, it appears not to have been used in the error analysis. If the rescaled form was being used in that work, then it would imply that $\|\mathbf{u}_{\text{in}}\|$ would be equal to 1, so $\log(1/\|\mathbf{u}_{\text{in}}\|) = 0$ which results in $N$ being infinite in Eq. (7.23) of Ref. [Kro23].

A problem with using this form is that it does not go down with the Carleman order. We aim to show that the error may be made arbitrarily small with higher-order Carleman approximations. We can provide tighter bounds when we consider the individual components of the Carleman vector, as in the following lemma.

**Lemma 3.1.4** (Component-wise Carleman error)**.** *Under the same setting as in Lemma 3.1.3, and*

*$j \in [N]$, the Carleman error for each individual component of $\eta_j$ satisfies*

$$\|\eta_j(t)\| \leq \left(\frac{\|\mathbf{u}_{\text{in}}\|}{\gamma}\right)^j R^k f_{j,k,M}\left(|\lambda_0|t\right), \qquad j \in \Omega_k \tag{3.1.34}$$

*where*

$$f_{j,k,M}(\tau) = 1 - \frac{(M-1)\,\Gamma(k+j/(M-1))}{(k-1)!\,\Gamma(j/(M-1))} \sum_{\ell=0}^{k-1} (-1)^\ell \binom{k-1}{\ell} \frac{e^{-(\ell M-\ell+j)\tau}}{\ell M-\ell+j}\,, \tag{3.1.35}$$

*for $k \in \{1, 2, \cdots, \lceil \frac{N}{M-1} \rceil\}$ and $k$ is determined so that for any $j$, we have $k$ whenever $j$ falls into the index set $j \in \Omega_k$ with*

$$\Omega_k \coloneqq \{N - k(M-1) + 1, \ldots, N + (k-1)(1-M)\}. \tag{3.1.36}$$

*In particular, for $k = \lceil N/(M-1) \rceil$ we have*

$$\|\eta_1(t)\| \leq \frac{\|\mathbf{u}_{\text{in}}\|}{\gamma} R^{\lceil \frac{N}{M-1} \rceil} f_{1,\lceil N/(M-1)\rceil,M}\left(|\lambda_0|t\right). \tag{3.1.37}$$

The proof of Lemma 3.1.4 can be found in Appendix C.2.2. The function $f_{j,k,M}(\tau)$ is monotonically decreasing with $k$, and in particular $f_{j,k,M}(\tau) \leq f_{j,1,M}(\tau) = 1 - e^{-j\tau}$ (see Appendix C.2.2). This result does not depend on the choice of rescaling $\gamma$. There is a factor of $1/\gamma^j$ in the definition of $\eta_j$, so the result is effectively independent of the choice of rescaling. Moreover, $\|\eta_1(t)\|$ gives the error in the desired component at the end, and shows that the error in $\mathbf{u}$ is proportional to $\|\mathbf{u}_{\text{in}}\|$.

A similar result was provided in Ref. [Liu+23] without using the rescaling, though that does not affect the result for the error. We give a significant improvement over the result in Ref. [Liu+23] by evaluating the nested integrals to give the function $f_{j,k,M}(\tau)$, whereas the result in Ref. [Liu+23] just corresponds to replacing $f_{j,k,M}(\tau)$ with its upper bound of 1.

We can use Lemma 3.1.4 to solve for a lower bound on $N$ for a given allowable error. In practice, we are interested in the error in the solution relative to $\|\mathbf{u}_{\text{in}}\|$ rather than $\gamma$, so we aim to bound $\|\eta_1(t)\|\gamma/\|\mathbf{u}_{\text{in}}\|$. Given a maximum allowable error $\varepsilon$, we then require

$$\varepsilon \leq R^{\lceil \frac{N}{M-1} \rceil} f_{1,\lceil N/(M-1)\rceil,M}\left(|\lambda_0|t\right) \leq R^{\lceil \frac{N}{M-1} \rceil}. \tag{3.1.38}$$

It is therefore sufficient to choose $N$ as

$$\left\lceil \frac{N}{M-1} \right\rceil \leq \frac{\log\left(1/\varepsilon\right)}{\log\left(1/R\right)}, \tag{3.1.39}$$

or

$$N = (M-1)\left\lceil \frac{\log\left(1/\varepsilon\right)}{\log\left(1/R\right)} \right\rceil - (M-2)\,. \tag{3.1.40}$$

We can also numerically solve for $N$, by using the exact expression for $f_{j,k,M}(\tau)$ given in Eq. (3.1.35). That will give a tighter lower bound on $N$, but there is not a closed-form expression.

### 3.1.5 Solution of the linearised system of ordinary differential equations using a truncated Taylor series

Next, we describe how to solve the system of ODEs that results from the Carleman mapping applied onto the nonlinear system. The most simple way to solve the system of ODEs is to apply the first-order method for time discretization known as the explicit Euler method. Upon application of the Euler method, there is a linear system of equations that can be solved. Here, this is a quantum linear system problem (QLSP), as the solution is encoded in a quantum state. In what follows, we aim to solve the linear system by a more sophisticated method than explicit Euler. The main drawback of the forward Euler method is low accuracy since it is a first-order method, meaning finer time discretization is required to achieve a required precision. As a result, the dependence of the complexity for solving the QLSP is quadratic in the solution time, and there is a near-linear factor in the inverse error [Liu+21; Liu+23].

Here, we follow the procedure outlined in Ref. [BC24], which allows us to obtain an algorithm that has complexity near-linear in time and logarithmic in the inverse error. The solution of a time-independent ODE system

$$\frac{\mathrm{d}\mathbf{u}(t)}{\mathrm{d}t} = A\mathbf{u}(t), \tag{3.1.41}$$

may be approximated by $\mathbf{u}_K(t) = W_K(t,t_0)\mathbf{u}(t_0)$, with

$$W_K(t,t_0) := \sum_{\ell=0}^{K} \frac{(A\Delta t)^\ell}{\ell!}. \tag{3.1.42}$$

This is a Taylor series truncated at order $K$. The error in the solution due to time propagation can be bounded as

$$\|\mathbf{u}_K(t) - \mathbf{u}(t)\| \in \mathcal{O}\left(\frac{(\|A\|\Delta t)^{K+1}}{(K+1)!}\|\mathbf{u}(t_0)\|\right). \tag{3.1.43}$$

We aim to solve Eq. (3.1.16) where the vector $\mathbf{u}(t)$ is mapped to a rescaled vector $\widetilde{\mathbf{y}}(t)$ and $A$ is the rescaled Carleman matrix $\widetilde{\mathcal{A}}_N$ truncated at order $N$.

Following Theorem 2 in [BC24], there exists a quantum algorithm that can provide an approximation $|\hat{\mathbf{y}}\rangle$ of the solution $|\widetilde{\mathbf{y}}(T)\rangle$ satisfying $\||\hat{\mathbf{y}}\rangle - |\widetilde{\mathbf{y}}(T)\rangle\| \leq \varepsilon y_{\max}$. To do so, we require that $\mathcal{A}_N$ has non-positive logarithmic norm and we have the oracles $U_y$ to prepare the initial state and block encoding of $\mathcal{A}_N$ via $U_{\widetilde{\mathcal{A}}_N}$ with $\langle 0|\, U_{\widetilde{\mathcal{A}}_N}\, |0\rangle = \widetilde{\mathcal{A}}_N/\lambda_{\widetilde{\mathcal{A}}_N}$. Then, to achieve the desired accuracy, the average number of calls to $U_y$ and $U_{\widetilde{\mathcal{A}}_N}$ needed are

$$U_y: \quad O\left(\widetilde{\mathcal{R}}\lambda_{\widetilde{\mathcal{A}}_N} T \log\left(\frac{1}{\varepsilon}\right)\right) \tag{3.1.44}$$

$$U_{\widetilde{\mathcal{A}}_N}: \quad O\left(\widetilde{\mathcal{R}}\lambda_{\widetilde{\mathcal{A}}_N} T \log\left(\frac{1}{\varepsilon}\right) \log\left(\frac{\lambda_{\widetilde{\mathcal{A}}_N} T}{\varepsilon}\right)\right). \tag{3.1.45}$$

Furthermore, the number of additional elementary gates scales as

$$O\left(\widetilde{\mathcal{R}}\lambda_{\widetilde{\mathcal{A}}_N} T \log\left(\frac{1}{\varepsilon}\right) \log^2\left(\frac{\lambda_{\widetilde{\mathcal{A}}_N} T}{\varepsilon}\right)\right). \tag{3.1.46}$$

In these expressions

$$\widetilde{\mathcal{R}} \geq \frac{y_{\max}}{\|\widetilde{\mathbf{y}}(T)\|} \tag{3.1.47}$$

$$y_{\max} \geq \max_{t\in[0,T]} \|\widetilde{\mathbf{y}}(t)\| \,. \tag{3.1.48}$$

The stability requirement on the ODE to use the solver as in Ref. [BC24] is that the logarithmic norm of the matrix is non-positive (similar to Ref. [Kro23]). That norm is given by the eigenvalues of $(\widetilde{\mathcal{A}}_N + \widetilde{\mathcal{A}}_N^\dagger)/2$. The eigenvalues of that matrix can be bounded via the block form of the Gershgorin circle theorem. That is equivalent to the usual Gershgorin circle theorem, except using the spectral norms of the off-diagonal blocks. For example, see Theorem 2 of Ref. [FV62], or Ref. [van79].

For $(\widetilde{\mathcal{A}}_N + \widetilde{\mathcal{A}}_N^\dagger)/2$ we obtain rows with $A_j^{(1)}$ and $\gamma^{M-1}A_j^{(M)}/2$ (for $j \geq M$) and $\gamma^{M-1}A_{j+M-1}^{(M)}/2$ (for $j + M - 1 \leq N$). Now $\|A_{j+M-1}^{(M)}\| \leq j\|F_M\|$, so the sum of the norms of the off-diagonal blocks is at most, for $j \geq M$ and $j + M - 1 \leq N$,

$$\|A_{j+M-1}^{(M)}\| + \|A_j^{(M)}\| \leq j\|F_M\| + (j - M + 1)\|F_M\| = (2j - M + 1)\|F_M\| \,. \tag{3.1.49}$$

In the case $j < M$ but $j + M - 1 \leq N$ then we get $j\|F_M\|$. If $j + M - 1 > N$ but $j \geq M$ then we get $(j - M + 1)\|F_M\|$. Now the maximum eigenvalue of $[A_j^{(1)} + (A_j^{(1)})^\dagger]/2$ is $j\lambda_0$. In that case the eigenvalues of $(\widetilde{\mathcal{A}}_N + \widetilde{\mathcal{A}}_N^\dagger)/2$ can be at most

$$\begin{cases} j\lambda_0 + j\gamma^{M-1}\|F_M\|/2 \,, & 0 < j < M \\ j\lambda_0 + (2j - M + 1)\gamma^{M-1}\|F_M\|/2 \,, & j \geq M \text{ and } j \leq N - M + 1 \\ j\lambda_0 + (j - M + 1)\gamma^{M-1}\|F_M\|/2 \,, & N \geq j > N - M + 1 \end{cases} \tag{3.1.50}$$

We can then see that the eigenvalues will be non-positive given all three inequalities

$$\gamma^{M-1} \leq \frac{2|\lambda_0|}{\|F_M\|} \,, \tag{3.1.51}$$

$$\gamma^{M-1} \leq \frac{|\lambda_0|}{[1 - (M-1)/(2(N - M + 1))]\|F_M\|} \,, \tag{3.1.52}$$

$$\gamma^{M-1} \leq \frac{|\lambda_0|}{[1 - (M-1)/N]\|F_M\|} \,. \tag{3.1.53}$$

Provided $N \geq 2(M - 1)$ (as would normally be the case) the middle inequality would imply the other two. In all cases we can satisfy these inequalities using

$$\gamma^{M-1} \leq \frac{|\lambda_0|}{\|F_M\|} = \frac{\|\mathbf{u}_{\text{in}}\|^{M-1}}{R} \,, \tag{3.1.54}$$

or

$$\gamma \leq \frac{\|\mathbf{u}_{\text{in}}\|}{R^{1/(M-1)}}, \tag{3.1.55}$$

where we used the definition of $R$ from Eq. (3.1.5) in the equality above. Reference [BC24] argues that for cases where the solution does not decay significantly, $\mathcal{R} \in O(1)$. Here, we consider dissipative dynamics without driving, so $\mathcal{R}$ may be large. That is less of a problem for driven equations. We expect that our methods can be applied to driven equations as well, but the error analysis is

considerably more complicated so we leave it as a problem for future work.

We can construct the block encoding of the Carleman matrix $\widetilde{\mathcal{A}}_N$ in terms of the block encoding of $F_1$ and $F_M$, as discussed in Appendix C.7. Denoting the values of $\lambda$ for $F_1$ and $F_M$ by $\lambda_{F_1}$ and $\lambda_{F_M}$ respectively, the value of $\lambda$ for $\widetilde{\mathcal{A}}_N$ is

$$\lambda_{\widetilde{\mathcal{A}}_N} \leq N\lambda_{F_1} + (N-M+1)\gamma^{M-1}\lambda_{F_M}\,. \tag{3.1.56}$$

This expression easily follows from expressing $\widetilde{\mathcal{A}}_N$ as a sum, and the value of $\lambda$ being the sum of the values of $\lambda$ in the sum. Since $\widetilde{\mathcal{A}}_N$ includes $A_j^{(1)}$ up to $A_N^{(1)}$, and $A_N^{(1)}$ is a sum of $N$ operators with identity tensored with $F_1$, we obtain the term $N\lambda_{F_1}$ above. Similarly, we have $\gamma^{M-1}A_j^{(M)}$ up to $\gamma^{M-1}A_N^{(M)}$, and $A_N^{(M)}$ is a sum of $N-M+1$ operators with with $F_M$, giving the $(N-M+1)\gamma^{M-1}\lambda_{F_M}$ term.

If we choose $\gamma^{M-1} = |\lambda_0|/\|F_M\|$ as above, then

$$N\|F_1\| > (N-M+1)\gamma^{M-1}\|F_M\|\,. \tag{3.1.57}$$

In typical cases we would expect that $\lambda_{F_1} \propto \|F_1\|$ and $\lambda_{F_M} \propto \|F_M\|$. That would imply

$$\lambda_{\widetilde{\mathcal{A}}_N} \lesssim 2N\lambda_{F_1}\,. \tag{3.1.58}$$

Note that the scaling has not increased the value of $\lambda$ by more than a constant factor. Note that this is assuming that the $\lambda$-values and norms in the block encoding are comparable, so it is possible it could be violated if the block encoding of $F_M$ is inefficient, so $\lambda_{F_M}$ is much larger than $\|F_M\|$.

Now for $\mathcal{R}$ we have $y_{\max}$ which considers the maximum norm that the vector can assume along the entire time evolution. Since we are working with a dissipative problem the maximum occurs at $t = 0$. First we consider the case without the scaling for comparison. To compute the norm $\|\mathbf{y}(0)\|$, note that it is the vector resulting from the Carleman mapping, i.e., $\mathbf{y}(0) = [\mathbf{u}_{\text{in}}, \mathbf{u}_{\text{in}}^{\otimes 2}, \ldots, \mathbf{u}_{\text{in}}^{\otimes N}]^T$, so

$$\|\mathbf{y}(0)\|^2 = \|\mathbf{u}_{\text{in}}\|^2 \frac{1-\|\mathbf{u}_{\text{in}}\|^{2N}}{1-\|\mathbf{u}_{\text{in}}\|^2}, \tag{3.1.59}$$

as in Eq. (3.1.23). Similarly for the value of the norm at time $T$,

$$\|\mathbf{y}(T)\|^2 = \|\mathbf{u}(T)\|^2 \frac{1-\|\mathbf{u}(T)\|^{2N}}{1-\|\mathbf{u}(T)\|^2}. \tag{3.1.60}$$

Therefore

$$\begin{aligned}\mathcal{R} &\geq \frac{y_{\max}}{\|\mathbf{y}(T)\|} \\ &= \left[\frac{\left(1-\|\mathbf{u}_{\text{in}}\|^{2N}\right)\left(1-\|\mathbf{u}(T)\|^2\right)}{\left(1-\|\mathbf{u}_{\text{in}}\|^2\right)\left(1-\|\mathbf{u}(T)\|^{2N}\right)}\right]^{1/2} \frac{\|\mathbf{u}_{\text{in}}\|}{\|\mathbf{u}(T)\|}.\end{aligned} \tag{3.1.61}$$

Moreover, the above complexity is in order to obtain the full Carleman vector. The quantity $\mathcal{R}$ corresponds to an inverse amplitude for obtaining the state at the final time, so tells us how many steps of amplitude amplification are needed in the algorithm. In practice, we want only $\mathbf{u}(T)$ rather than the full vector. That implies a further factor in the complexity of $\|\mathbf{y}(T)\|/\|\mathbf{u}(T)\|$, corresponding

to the inverse amplitude for obtaining the component of the Carleman vector containing the solution. That gives a factor in the complexity of

$$\frac{\|\mathbf{y}(T)\|}{\|\mathbf{u}(T)\|}\mathcal{R} \geq \left[\frac{\left(1-\|\mathbf{u}_{\text{in}}\|^{2N}\right)}{(1-\|\mathbf{u}_{\text{in}}\|^2)}\right]^{1/2} \frac{\|\mathbf{u}_{\text{in}}\|}{\|\mathbf{u}(T)\|}. \tag{3.1.62}$$

From the equation above we can see how $\mathcal{R}$ grows exponentially in $N$ for $\|\mathbf{u}_{\text{in}}\| > 1$.

Now with the rescaling, we simply divide each $\mathbf{u}_{\text{in}}$ or $\mathbf{u}(T)$ by $\gamma$. That gives us

$$\frac{\|\widetilde{\mathbf{y}}(T)\|}{\|\widetilde{\mathbf{u}}(T)\|}\widetilde{\mathcal{R}} \geq \left[\frac{\left(1-\|\mathbf{u}_{\text{in}}\|^{2N}/\gamma^{2N}\right)}{(1-\|\mathbf{u}_{\text{in}}\|^2/\gamma^2)}\right]^{1/2} \frac{\|\mathbf{u}_{\text{in}}\|}{\|\mathbf{u}(T)\|}. \tag{3.1.63}$$

With the choice $\gamma^{M-1} = |\lambda_0|/\|F_M\|$, we obtain

$$\frac{\|\widetilde{\mathbf{y}}(T)\|}{\|\widetilde{\mathbf{u}}(T)\|}\widetilde{\mathcal{R}} \geq \frac{1}{\sqrt{1-R^{2/(M-1)}}} \frac{\|\mathbf{u}_{\text{in}}\|}{\|\mathbf{u}(T)\|}. \tag{3.1.64}$$

We then can see that the amplitude amplification cost can be exponentially reduced when $\|\mathbf{u}_{\text{in}}\| > 1$. We could also use $\gamma = \|\mathbf{u}_{\text{in}}\|$ to give

$$\frac{\|\widetilde{\mathbf{y}}(T)\|}{\|\widetilde{\mathbf{u}}(T)\|}\widetilde{\mathcal{R}} \geq \sqrt{N} \frac{\|\mathbf{u}_{\text{in}}\|}{\|\mathbf{u}(T)\|}\,, \tag{3.1.65}$$

but that bound is looser for realistic parameters.

A further consideration is the relation between the relative error in the solution for $\widetilde{\mathbf{y}}(T)$ and that for $\mathbf{u}(T)$. The complexity of the solution for the ODE solver is in terms of the former, whereas we need to bound the relative error in $\mathbf{u}(T)$. We have the error upper bounded by (with hats used to indicate results given by the linear equation solver)

$$\begin{aligned}
\|\hat{\mathbf{u}}(T) - \mathbf{u}(T)\| &\leq \gamma \|\hat{\mathbf{y}}(T) - \mathbf{y}(T)\| \\
&\leq \gamma\varepsilon y_{\max} \\
&\leq \gamma \left[\frac{\left(1-\|\mathbf{u}_{\text{in}}\|^{2N}/\gamma^{2N}\right)}{(1-\|\mathbf{u}_{\text{in}}\|^2/\gamma^2)}\right]^{1/2} \frac{\|\mathbf{u}_{\text{in}}\|}{\gamma} \\
&\leq \varepsilon \|\mathbf{u}_{\text{in}}\| \frac{1}{\sqrt{1-R^{2/(M-1)}}}\,.
\end{aligned} \tag{3.1.66}$$

In the second line we have assumed that the ODE solver has given the solution for $\mathbf{y}(T)$ to within error $\varepsilon y_{\max}$. This shows that the relative error in $\mathbf{u}(T)$ is the same as that for $\widetilde{\mathbf{y}}(T)$, up to a factor of $1/\sqrt{1-R^{2/(M-1)}}$ which should be close to 1. We can also use the simpler but looser upper bound

$$\|\hat{\mathbf{u}}(T) - \mathbf{u}(T)\| \leq \varepsilon \|\mathbf{u}_{\text{in}}\| \sqrt{N}\,, \tag{3.1.67}$$

which is obtained by noting that the expression in the square brackets in the third line of Eq. (3.1.66) is upper bounded by $N$.

We can now use the ODE solver given in Ref. [BC24] in combination with our rescaling technique to provide our quantum algorithm for Def. 1.

**Lemma 3.1.5** (Complexity of solving ODE)**.** *There is an algorithm to solve the nonlinear ODE from Eq.* (3.1.3) *i.e., to produce a quantum state* $|\hat{\mathbf{u}}(T)\rangle$ *encoding the solution such that* $\|\hat{\mathbf{u}}(T) - \mathbf{u}(T)\| \leq \varepsilon\|\mathbf{u}_{\text{in}}\|$*, using an average number*

$$O\left(\frac{1}{\sqrt{1-R^{2/(M-1)}}}\frac{\|\mathbf{u}_{\text{in}}\|}{\|\mathbf{u}(T)\|}\lambda_{F_1}TN\log\left(\frac{N}{\varepsilon}\right)\log\left(\frac{N\lambda_{F_1}T}{\varepsilon}\right)\right), \tag{3.1.68}$$

*of calls to oracles for* $F_1$ *and* $F_M$*,*

$$O\left(\frac{1}{\sqrt{1-R^{2/(M-1)}}}\frac{\|\mathbf{u}_{\text{in}}\|}{\|\mathbf{u}(T)\|}\lambda_{F_1}TN^2\log\left(\frac{N}{\varepsilon}\right)\right), \tag{3.1.69}$$

*calls to oracles for preparation of* $\mathbf{u}_{\text{in}}$*, and*

$$O\left(\frac{1}{\sqrt{1-R^{2/(M-1)}}}\frac{\|\mathbf{u}_{\text{in}}\|}{\|\mathbf{u}(T)\|}\lambda_{F_1}TN^2M\log\left(\frac{N}{\varepsilon}\right)\log\left(\frac{N\lambda_{F_1}T}{\varepsilon}\right)^2\log n\right), \tag{3.1.70}$$

*additional gates for dimension* $n$*, with*

$$N = O\left((M-1)\frac{\log(1/\varepsilon)}{\log(1/R)}\right). \tag{3.1.71}$$

*We require that* $R < 1$ *and assume that* $\lambda_{F_M}/\|F_M\| = \mathcal{O}(\lambda_{F_1}/\|F_1\|)$ *for the block encodings of* $F_1$ *and* $F_M$*.*

*Proof.* The main step to derive our quantum algorithm is first to apply the Carleman linearization in the rescaled nonlinear ODE problem, which is given in Eq. (3.1.26). We then have a linear ODE problem with the Carleman matrix of order $N$, denoted $\widetilde{\mathcal{A}}_N$. We can then apply the ODE solver given in Ref. [BC24] to this equation.

There are then a number of considerations needed to give the overall complexity.

- We need to multiply by a further factor of $\|\widetilde{\mathbf{y}}(T)\|/\|\widetilde{\mathbf{u}}(T)\|$ to obtain the correct component of the solution containing the approximation of $\mathbf{u}(T)$. The product of that with $\widetilde{\mathcal{R}}$ is given above in Eq. (3.1.64).

- The value of $\lambda_{\widetilde{\mathcal{A}}_N}$ is given above in Eq. (3.1.58) under the assumption $\lambda_{F_M}/\|F_M\| = \mathcal{O}(\lambda_{F_1}/\|F_1\|)$, which gives $\lambda_{\widetilde{\mathcal{A}}_N} = \mathcal{O}(N\lambda_{F_1})$.

- The matrix $\widetilde{\mathcal{A}}_N$ can be block encoded with $O(1)$ calls to the oracles for $F_1$, $F_M$. There is an extra $O(N)$ factor for the number of calls to $\mathbf{u}_{\text{in}}$. The implementation of the oracles is explained in Appendix C.7.

- The choice of the Carleman order $N$ in order to obtain a sufficiently accurate solution is given in Eq. (3.1.40). The error from the Carleman truncation can be chosen to be a fraction of the total allowable relative error $\varepsilon$ here, which is accounted for using the order notation for $N$.

- The solution for the ODE can be given to relative error $\varepsilon/\sqrt{N}$. According to Eq. (3.1.67) that will ensure that the relative error in $\mathbf{u}(T)$ obtained is $\varepsilon$ as required. It is for this reason that we have replaced the $1/\varepsilon$ in the complexity for the ODE solver with $N/\varepsilon$.

For the additional elementary gates, the block encoding as in Appendix C.7 requires a factor of $O(NM \log n)$ for swapping target registers into the appropriate location. That is a factor on the number of block encodings of $\widetilde{\mathcal{A}}_N$. Moreover, Ref. [BC24] gives a log factor to account for the complexity of correctly giving the weighting in the Taylor series. For simplicity we give the product of these factors, but these factors are for different contributions to the complexity and we could instead give a more complicated expression with the maximum of $NM \log n$ and the logarithm. □

We can compare our quantum algorithm performance with what is given in Theorem 8 of Ref. [Kro23] for the case $M = 2$. The complexity given in that theorem can be simplified to the situation we consider by removing the driving term and replacing $\|A\|$ with $\lambda_{\mathcal{A}_N}$. Then the complexity from [Kro23] is

$$O\left(\frac{\|\mathbf{u}_{\text{in}}\|}{\|\mathbf{u}(T)\|}\lambda_{F_1} TN \text{ poly}\left(N, \log\left(\frac{1}{\varepsilon}\right), \log\left(TN\lambda_{F_1}\right)\right)\right). \tag{3.1.72}$$

The speedup is unclear because the complexity in that work is given in terms of poly factors. That work appears to be assuming a rescaling in order to avoid complexity exponential in $N$, but by assuming $\|\mathbf{u}_{\text{in}}\| = 1$. The problem is that they give a formula for $N$ as

$$N = \left\lceil \frac{2\log\left(T\|F_2\|/\delta\|\mathbf{u}(T)\|\right)}{\log\left(1/\|\mathbf{u}_{\text{in}}\|\right)} \right\rceil. \tag{3.1.73}$$

Using $\|\mathbf{u}_{\text{in}}\| = 1$ in that formula gives infinite $N$. In contrast, here we have given the rescaling explicitly and given a working formula for $N$.

### 3.1.6 Application to the quantum nonlinear PDE problem

#### Complexity of the quantum algorithm

We now demonstrate our techniques applied to the nonlinear PDE [Liu+23]

$$\partial_t u(\mathbf{x}, t) = D\Delta u(\mathbf{x}, t) + cu(\mathbf{x}, t) + bu^M(\mathbf{x}, t), \tag{3.1.74}$$

for some diffusion coefficient $D \geq 0$ and constants $c, b \in \mathbb{R}$. As a simple means of discretization we consider finite differences with periodic boundary conditions, which leads to a vector-valued ODE that approximates the dynamics in Eq. (3.1.74). We go beyond the two-point stencil demonstrated in Ref. [Liu+23] and apply higher-order finite differences similar to Ref. [CLO21] for the linear case.

We discretise a $d$-dimensional space in each direction with uniformly equidistant grid points. As a result, we obtain a nonlinear system of ODEs as in Eq. (3.1.3) with $n$ grid points in total, or $n^{1/d}$ in each direction. Moreover, we consider the width of the simulation region to be 1 in each direction, so $x_j \in [0, 1]$, for simplicity.

The linear operator $F_1$ resulting from the spatial discretization of our PDE is given by

$$F_1 = DL_{k,d} + c\mathbb{I}^{\otimes d}, \tag{3.1.75}$$

where $\mathbb{I}$ is the $n^{1/d} \times n^{1/d}$ identity matrix, and

$$L_{k,d} = \sum_{\mu=1}^{d} \mathbb{I}^{\otimes(\mu-1)} \otimes L_k \otimes \mathbb{I}^{\otimes(d-\mu)} \,. \tag{3.1.76}$$

The operator $L_{k,d}$ above for the discretised Laplacian in dimension $d$ is constructed from the sum of the discretised Laplacians in one dimension, $L_k$. Here, $k$ is the order, so the truncation error scales as the inverse grid spacing to the power of $2k-1$, and $2k+1$ stencil points are used.

A Laplacian in one dimension with a $k$th order approximation and periodic boundary conditions can be expressed in terms of weights $a_j$ as (see Ref. [CLO21])

$$L_k = n^{2/d} \left( a_0 I + \sum_{j=1}^{k} a_j (S^j + S^{-j}) \right) , \tag{3.1.77}$$

where $S$ is a $n^{1/d} \times n^{1/d}$ matrix, where the entries are $S_{i,j} = \delta_{i,j+1 \mod n^{1/d}}$; $S$ is also known as a circulant matrix. Note that for a total of $n^{1/d}$ grid points and a region size of 1 in each direction, the grid spacing is $1/n^{1/d}$. The method to obtain the coefficients $a_j$ for the Laplacian operator given in Appendix C.5 guarantees that

$$a_0 + 2 \sum_{j=1}^{k} a_j = 0 \,, \tag{3.1.78}$$

and we provide the coefficients for $1 \leq k \leq 5$ in Table 3.1 (these are from [CJO19]). Moreover, this procedure leads to a truncation error in the representation of the Laplacian operator which scales as (for the 2-norm) [KWBA17; CLO21]

$$O\left( C(u,k) \sqrt{n} \left( \frac{e}{2} \right)^{2k} n^{(-2k+1)/d} \right) , \tag{3.1.79}$$

where $C(u,k)$ is a constant depending on the $(2k+1)$st spatial derivative in each direction

$$C(u,k) = \sum_{j=1}^{d} \left| \frac{d^{2k+1} u}{dx_j^{2k+1}} \right| . \tag{3.1.80}$$

This expression is obtained from that in Refs. [KWBA17; CLO21] by adding the errors for derivatives in each direction.

| Order $k$ | Coefficients $a_0$ to $a_k$ |
|---|---|
| 1 | -2, 1 |
| 2 | -5/2, 4/3, -1/12 |
| 3 | -49/18, 3/2, -3/20, 1/90 |
| 4 | -205/72, 8/5, -1/5, 8/315, -1/560 |
| 5 | -5269/1800, 5/3, -5/21, 5/126, -5/1008, 1/3150 |

Table 3.1: Central finite difference coefficients for approximating a second derivative in one dimension [CJO19].

We also have the matrix $F_M$ resulting from the spatial discretization of the nonlinear part

$bu^M(\mathbf{x},t)$ that is a rectangular matrix $F_M$,

$$F_M : \mathbb{R}^{n^M} \to \mathbb{R}\, R^n, \tag{3.1.81}$$

operating on the vector $\mathbf{u}^{\otimes M}$, as given in Eq. (3.1.10). Since we are only interested in the components $u_i^M$ from $\mathbf{u}^{\otimes M}$, where $i = 1, 2, \cdots n$, $F_M$ is a one sparse matrix with the non-zero components given by $b$. Hence $\|F_M\| = \|F_M\|_{\max} = |b|$. In the case $M = 2$, where

$$\mathbf{u}^{\otimes 2} = (u_1^2, u_1u_2, \ldots, u_1u_n, u_2u_1, u_2^2, \ldots, u_nu_{n-1}, u_{n^2})^T \in \mathbb{R}^{n^2}, \tag{3.1.82}$$

we can express $F_2$ as

$$[F_2]_{pq} = \begin{cases} b, & q = p + (p-1)n \\ 0, & \text{otherwise.} \end{cases} \tag{3.1.83}$$

Returning to the Laplacian operator with periodic boundary conditions, we see that Eq. (3.1.77) is a circulant matrix, so its eigenvalues are given by

$$\begin{aligned} \lambda_\ell(L_k) &= n^{2/d}\left[a_0 + \sum_{j=1}^{k} a_j\left(\omega^{\ell j} + \omega^{-\ell j}\right)\right] \\ &= n^{2/d}\left[a_0 + 2\sum_{j=1}^{k} a_j \cos\left(\frac{2\pi\ell j}{n^{1/d}}\right)\right], \quad \ell \in [n^{1/d} - 1] \end{aligned} \tag{3.1.84}$$

where $\omega = e^{i2\pi/n^{1/d}}$. Since the $a_j$, with $j = 0, 1, \cdots, k$, satisfy the condition in Eq. (3.1.78), we see that for $\ell = 0$, $\lambda_0 = 0$ gives the maximum eigenvalue and $\lambda_\ell < 0$ for $\ell \neq 0$. Moreover, using the triangle inequality in Eq. (3.1.77) we see that

$$\|L_k\| \leq n^{2/d}\left(|a_0| + 2\sum_{j=1}^{k} |a_j|\right). \tag{3.1.85}$$

By a simple application of Gershgorin's circle theorem, one may obtain the asymptotic bound (Lemma 2 in Ref. [CLO21], Lemma 6 in Ref. [KWBA17]),

$$\|L_k\| \leq n^{2/d}\frac{4\pi^2}{3}. \tag{3.1.86}$$

From the eigenvalues of $L_k$ we can then determine the eigenvalues of $F_1$ (which is symmetric so equal to $(F_1 + F_1^\dagger)/2$) as defined in Eq. (3.1.75) as

$$\lambda_{\vec{\ell}}(F_1) = c + Dn^{2/d}\sum_{p=1}^{d}\left[a_0 + 2\sum_{j=1}^{k} a_j \cos\left(\frac{2\pi\ell_p j}{n^{1/d}}\right)\right]. \tag{3.1.87}$$

As discussed above the maximum eigenvalue of $L_k$ is 0 with periodic boundary conditions (it can be negative for non-periodic boundary conditions). For Carleman linearization to be successful, we require $R < 1$ and, in particular, $\lambda_0 < 0$; see Section 3.1.4. Since the maximum eigenvalue of $F_1$ is $c$, we choose negative $c$ such that the overall dynamics becomes dissipative and satisfies the stability

condition $R < 1$. Moreover, we obtain the following bounds

$$\|F_1\| \leq |c| + dDn^{2/d}\left(|a_0| + \sum_{j=1}^{k}|a_j|\right) \leq |c| + dDn^{2/d}\frac{4\pi^2}{3}. \tag{3.1.88}$$

The value of $\lambda_{F_1}$ for the block encoding of $F_1$ can be determined in a similar way. The block encoding can be implemented by a linear combination of unitaries of the identity and powers of the circulant matrices $S$. The value of $\lambda_{F_1}$ is then exactly equal to the sum

$$\lambda_{F_1} = |c| + dDn^{2/d}\left(|a_0| + \sum_{j=1}^{k}|a_j|\right) \leq |c| + dDn^{2/d}\frac{4\pi^2}{3}. \tag{3.1.89}$$

Similarly, since $F_M$ is one-sparse it can be easily block encoded with a value of $\lambda_{F_M}$ equal to its norm of $|b|$. We then obtain the value of $\lambda$ for the complete block encoding as

$$\begin{aligned}\lambda_{\tilde{\mathcal{A}}_N} &= N\lambda_{F_1} + (N-M+1)\gamma^{M-1}\lambda_{F_M} \\ &\leq N\left(|c| + dDn^{2/d}\frac{4\pi^2}{3}\right) + (N-M+1)\gamma^{M-1}|b|\,.\end{aligned} \tag{3.1.90}$$

The condition on the dissipativity of the ODE $R < 1$ implies that

$$N\left(|c| + dDn^{2/d}\frac{4\pi^2}{3}\right) > (N-M+1)\gamma^{M-1}|b|\,. \tag{3.1.91}$$

Given these results for the discretization of the PDE, we can use Lemma 3.1.5 to provide the following corollary.

**Corollary 3.1.6** (Complexity of solving a dissipative reaction-diffusion PDE)**.** *There is a quantum algorithm to solve the nonlinear PDE in Eq.* (3.1.74) *i.e., to produce a quantum state* $|\hat{\mathbf{u}}(T)\rangle$ *encoding the solution such that* $\|\hat{\mathbf{u}}(T) - \mathbf{u}(T)\| \leq \varepsilon\|\mathbf{u}_{\text{in}}\|$*, using an average number*

$$O\left(\frac{1}{\sqrt{1-R^{2/(M-1)}}}\frac{\|\mathbf{u}_{\text{in}}\|}{\|\mathbf{u}(T)\|}(dDn^{2/d}+|c|)TN\log\left(\frac{N}{\varepsilon}\right)\log\left(\frac{N(dDn^{2/d}+|c|)T}{\varepsilon}\right)\right), \tag{3.1.92}$$

*of calls to oracles for* $F_1$ *and* $F_M$*, as defined in Eq.* (3.1.75) *and Eq.* (3.1.83) *respectively,*

$$O\left(\frac{1}{\sqrt{1-R^{2/(M-1)}}}\frac{\|\mathbf{u}_{\text{in}}\|}{\|\mathbf{u}(T)\|}(dDn^{2/d}+|c|)TN^2\log\left(\frac{N}{\varepsilon}\right)\right), \tag{3.1.93}$$

*calls to oracles for preparation of* $\mathbf{u}_{\text{in}}$*, and*

$$O\left(\frac{1}{\sqrt{1-R^{2/(M-1)}}}\frac{\|\mathbf{u}_{\text{in}}\|}{\|\mathbf{u}(T)\|}(dDn^{2/d}+|c|)TN^2M\log\left(\frac{N}{\varepsilon}\right)\log\left(\frac{N(dDn^{2/d}+|c|)T}{\varepsilon}\right)^2\log n\right), \tag{3.1.94}$$

*additional gates, with*

$$N = O\left((M-1)\frac{\log(1/\varepsilon)}{\log(1/R)}\right). \tag{3.1.95}$$

*We require that* $R < 1$*, where* $R$ *is computed from the discretised input vector* $\mathbf{u}_{\text{in}}$*.*

*Proof.* We first discretise the reaction-diffusion problem in Eq. (3.1.74) to the nonlinear ODE system with $n$ discretization points. We consider just the error in the solution of this ODE here, with the choice of $n$ to accurately approximate the solution of the PDE described below. For this ODE we have an explicit bound for $\lambda_{F_1}$ given in Eq. (3.1.89), and can use it in the expressions in Lemma 3.1.5. For this simple $F_M$ the values of $\lambda_{F_M}$ and $\|F_M\|$ are equal, so the condition $\lambda_{F_M}/\|F_M\| = \mathcal{O}(\lambda_{F_1}/\|F_1\|)$ is satisfied. □

Note that for this result the oracles for $F_1$ and $F_M$ can be easily implemented in terms of calls to elementary gates, with logarithmic complexity in $n$ and linear complexity in $M$. Powers of the circulant matrices can be implemented with modular addition, and $F_M$ can be implemented via equality tests between the copies it acts upon. Note also that, apart from the $R < 1$ condition, this complexity scales as $n^{2/d}$ up to logarithmic factors. For $d \geq 3$ this complexity is sublinear in $n$. This factor comes from the size of the discretised Laplacian, and is similar to that for quantum algorithms for linear PDEs. If we had the factor of $\|\mathbf{u}_{\text{in}}\|^{2N}$ as in Ref. [Liu+23], then because $\|\mathbf{u}_{\text{in}}\|^2 \propto n$ with the discretization there would be a further factor of $n^N$ for the scaling with $n$, making the complexity far worse than that for a simple classical solver.

**Stability and discretization**

Here we discuss conditions on nonlinear differential equations of the type in Eq. (3.1.74) so that numerical schemes based on Carleman linearization are stable. Recall that for the ODE we have the stability condition $R < 1$ with

$$R = \frac{\|F_M\| \cdot \|\mathbf{u}_{\text{in}}\|^{M-1}}{|\lambda_0|}. \tag{3.1.96}$$

That condition is not ideal here, because the 2-norm of the solution increases with the number of discretization points. Thus this condition for the stability depends not only on the underlying PDE and initial state but on its discretization.

Ideally we would aim for a condition on the max-norm of the solution. That can then be used in order to guarantee stability of the solution as well as to bound error. For example, Ref. [Liu+23] considers stability in their Lemma 2.1 and bounds error in their Theorem 3.3. A simple stability criterion can be given as

$$\|\mathbf{u}_{\text{in}}\|_{\max}^{M-1} \frac{b}{|c|} < 1\,. \tag{3.1.97}$$

Before discretization, the stability can be shown simply by considering the infinitesimal time interval $\mathrm{d}t$ and using

$$\begin{aligned}\|u(\mathbf{x},t) + \mathrm{d}t[D\Delta u(\mathbf{x},t) + cu(\mathbf{x},t) + bu^M(\mathbf{x},t)]\|_{\max} &= \|(\mathbb{I} + \mathrm{d}t\, D\Delta)\{u(\mathbf{x},t) + \mathrm{d}t[cu(\mathbf{x},t) + bu^M(\mathbf{x},t)]\}\|_{\max} \\ &\leq \|\,\mathbb{I} + \mathrm{d}t\, D\Delta\|_\infty \|u(\mathbf{x},t) + \mathrm{d}t[cu(\mathbf{x},t) + bu^M(\mathbf{x},t)]\|_{\max}.\end{aligned} \tag{3.1.98}$$

Now using the triangle inequality

$$\begin{aligned}\|u(\mathbf{x},t) + \mathrm{d}t[cu(\mathbf{x},t) + bu^M(\mathbf{x},t)]\|_{\max} &\leq \|u(\mathbf{x},t) + \mathrm{d}t\, cu(\mathbf{x},t)\|_{\max} + \mathrm{d}t\|bu^M(\mathbf{x},t)\|_{\max} \\ &= (1 + \mathrm{d}t\, c)\|u(\mathbf{x},t)\|_{\max} + \mathrm{d}t\, b\|u(\mathbf{x},t)\|_{\max}^M\,.\end{aligned} \tag{3.1.99}$$

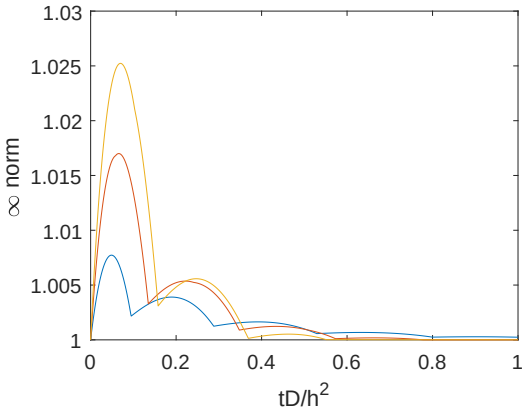



Figure 3.4: The (induced) $\infty$-norm of the exponential of $tD\Delta_h$ when using a discretization of the Laplacian of second order (blue), third order (red), and fourth order (orange).

Then if $(b/c)\,\|u(\mathbf{x},t)\|_{\max}^{M-1} \leq 1$ this expression is upper bounded by $\|u(\mathbf{x},t)\|_{\max}$.

Moreover, it is a standard result that $\|\,\mathbb{I} + \mathrm{d}t\, D\Delta\|_\infty = 1$. That is, the diffusion equation smooths out any peaks in the distribution. That expression also holds if we consider the discretised form, but only using the first-order discretization. Then for spatial grid spacing $h$, the discretised form in one dimension has $1 - 2\mathrm{d}t\, D/h^2$ on the diagonal, and $\mathrm{d}t\, D/h^2$ on the two off-diagonals. The sum of the absolute values along a row for this matrix is then exactly 1, giving an $\infty$-norm of 1. That means Eq. (3.1.98) implies $\|u(\mathbf{x},t)\|_{\max}$ is non-increasing for the PDE given the stability criterion in Eq. (3.1.97).

This result for the $\infty$-norm no longer holds for the discretised PDE when using higher-order discretizations. For example, for the second-order discretization, the $-1/12$ on the off-diagonals means that

$$\|\,\mathbb{I} + \mathrm{d}t\, DL_2\|_\infty = 1 + \mathrm{d}t\, D/(3h^2)\,. \tag{3.1.100}$$

That means that the max-norm is only upper bounded by the initial max-norm multiplied by a factor of $\exp\big(D/(3h^2)\big)$. In practice it is found that the max-norm is far better behaved. If we calculate the $\infty$-norm of $\exp(tDL_k)$, then we obtain the results shown in Fig. 3.4. For the second-order discretization the initial slope is $1/3$ as predicted using infinitesimal $t$, but the peak value is less than 1% above 1. For the higher-order discretizations this maximum increases, but it is still small for these orders. Therefore we find that if we consider the max-norm for just evolution under the discretised Laplacian then it is well-behaved, but that does not imply the result for the nonlinear discretised PDE.

To determine stability for the 2-norm, the equivalent of Eq. (3.1.98) gives

$$\begin{aligned}
\|u(\mathbf{x},t+\mathrm{d}t)\| &\leq \|\mathbb{I} + \mathrm{d}t\, D\Delta\| \times \big\|u(\mathbf{x},t) + \mathrm{d}t\,\big[cu(\mathbf{x},t) + bu^M(\mathbf{x},t)\big]\big\| \\
&\leq \big\|u(\mathbf{x},t) + \mathrm{d}t\,\big[cu(\mathbf{x},t) + bu^M(\mathbf{x},t)\big]\big\| \\
&\leq \|u(\mathbf{x},t)\| + \mathrm{d}t\,\big[c\|u(\mathbf{x},t)\| + b\|u^M(\mathbf{x},t)\|\big] \\
&\leq \|u(\mathbf{x},t)\| + \mathrm{d}t\,\big[c\|u(\mathbf{x},t)\| + b\|u(\mathbf{x},t)\|_{\max}^{M-1}\|u(\mathbf{x},t)\|\big]\,.
\end{aligned} \tag{3.1.101}$$

Therefore the 2-norm is non-increasing provided $(b/c)\,\|u(\mathbf{x},t)\|_{\max}^{M-1} \leq 1$. In the discretised case the

non-positive eigenvalues of the discretised Laplacian mean that the 2-norm is still stable given this condition, though as noted above it is possible for $\|\mathbf{u}\|_{\max}$ to increase above its initial value.

However, for the purpose of solving the ODE using Carleman linearization, what matters is not the stability of the nonlinear equation, but that of the linearised equation. That is because the stability of the linearised equation governs the condition number of the linear equations to solve, and in turn that is proportional to the complexity. For example, in Ref. [Liu+23] their Problem 1 assumes that $\|F_M\| \leq |\lambda_0|$ ($\lambda_1$ in the notation of that work) after some possible rescaling of the equation. Then Eq. (4.15) of that work gives $\|\mathbb{I} + Ah\| \leq 1$ (with $h$ the time discretization), using that condition from Problem 1. That is then used to provide the bound on the norm of $\|L^{-1}\|$ in Eq. (4.28) of that work, which is then used to give the bound on the condition number proportional to the number of time steps in Eq. (4.29) in Ref. [Liu+23].

According to our analysis above, the stability of the linearised system will be satisfied provided $\gamma^{M-1} \leq |\lambda_0|/\|F_M\|$. For the discretised PDE here we have $\lambda_0 = c$ and $\|F_M\| = b$. That means if $\gamma \leq \|\mathbf{u}_{\text{in}}\|_{\max}$, then the stability condition in Eq. (3.1.97) implies the stability of the matrix after Carleman linearization. That condition is needed in order to be able to use the ODE solver of Ref. [BC24], but it will mean that the rescaling gives a smaller probability of success for obtaining the correct component of the Carleman vector than if we had the stability condition $R < 1$.

However, if we have sufficiently small $b/|c|$, then the condition $R < 1$ would be satisfied, so

$$\|\mathbf{u}_{\text{in}}\|^{M-1} \frac{b}{|c|} < 1\,. \tag{3.1.102}$$

Because $\|\mathbf{u}_{\text{in}}\|$ increases with the number of discretization points as $\sqrt{n}$, this inequality can only be satisfied if the number of discretization points is made as small as possible. This gives a strong motivation for using the higher-order spatial discretization of the PDE. See Appendix C.4 for discussion of the number of points needed.

### Error Analysis

The overall error $\varepsilon$ comes from three different parts,

- the spatial discretization error of the semi-discrete dynamics $\varepsilon_{\text{disc}}$,
- the error $\varepsilon_{\text{Carl}}$ contributed by truncation in the Carleman linearization as bounded in Lemma 3.1.4, and
- the error in the time evolution $\varepsilon_{\text{time}}$ due to the Taylor series, as described in Section 3.1.5.

As usual in this type of analysis, we can simplify the discussion by taking the error to be $\varepsilon$ for each of these contributions. In reality, the contribution to the error from each source would need to be taken to be a fraction of $\varepsilon$ (e.g. $\varepsilon/3$), but because that fraction would at most give a constant factor to the complexity, it would not affect the complexities quoted using $\mathcal{O}$.

We have already considered $\varepsilon_{\text{time}}$ and $\varepsilon_{\text{Carl}}$ above in Corollary 3.1.6. The time discretization error will not be further considered here, but we will discuss how the Carleman error can be alternatively bounded in situations where the PDE is stable but $R \geq 1$. Above we show that the ODE needs $R < 1$ for the quantum solution to be efficient, but this bound on the Carleman error will be useful if that limitation can be circumvented.

In the case of higher-order discretised Laplacians we obtain a somewhat worse bound as derived in Appendix C.2.3,

$$\|\eta_j(t)\|_{\max} \lesssim G_\kappa^{dj} \left( \frac{\|F_M\|_\infty}{|c|} \|\mathbf{u}_{\text{in}}\|_{\max}^{M-1} G_\kappa^{dM} \right)^k f_{j,k,M}\left(|c|t\right) , \tag{3.1.103}$$

where

$$G_\kappa := \max_{\tau \geq 0} \|e^{L_\kappa \tau}\|_\infty . \tag{3.1.104}$$

The $\lesssim$ is because it is assuming that the max-norm of the solution is not increasing. We can use $\leq$ if $\|\mathbf{u}_{\text{in}}\|_{\max}$ is replaced with the maximum of $\|\mathbf{u}\|_{\max}$ over time. The quantity $G_\kappa$ is greater than 1 for higher-order discretised Laplacians, so this is a slightly larger upper bound than in the case of first-order discretised Laplacians. Nevertheless, the Carleman error may be made arbitrarily small with order provided

$$\|\mathbf{u}_{\text{in}}\|_{\max}^{M-1} \frac{b}{|c|} G_\kappa^{dM} < 1 . \tag{3.1.105}$$

This condition is slightly stronger than the condition for stability of the PDE by the factor of $G_\kappa^{dM}$, but that will typically be close to 1. Typically this will be a much weaker requirement than the stability condition $R < 1$.

Next, we consider the bound on the error due to the spatial discretization, which can be used to derive the appropriate number of discretization points $n$ to use. Using that in Corollary 3.1.6 then gives the complexity entirely in terms of the parameters of the problem instead of the value chosen for $n$. Our bound on the error is as given in the following Lemma, with the proof given in Appendix C.3.

**Lemma 3.1.7** (Nonlinear PDE solution error when discretising the Laplacian with higher-order finite differences)**.** *Using a higher-order finite difference discretization with $2k+1$ stencil points in each direction, the solution of the PDE*

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = (DL_{k,d} + c)\mathbf{u} + b\mathbf{u}^{\otimes M}, \tag{3.1.106}$$

*at time $T > 0$ has error due to spatial discretization when $c < 0$ and $|c| > M|b|\|\mathbf{u}_{\text{in}}\|^{M-1}$ bounded as*

$$\|\varepsilon_{\text{disc}}(T)\| = O\left( C(u,k)\sqrt{n} \left(\frac{e}{2}\right)^{2k} n^{-(2k-1)/d} \frac{1 - \exp\left\{ \left( c + M|b|\|\mathbf{u}_{\text{in}}\|_{\max}^{M-1} \right) t \right\}}{\left| c + M|b|\|\mathbf{u}_{\text{in}}\|_{\max}^{M-1} \right|} \right), \tag{3.1.107}$$

*where $C(u,k)$ given in Eq. (3.1.80) is a constant depending on the $(2k+1)$st spatial derivative of the solution assuming sufficient regularity, $n$ is the number of grid points used, and $d$ is the number of dimensions.*

In this result, we are considering the continuous time evolution. Note that for the stability of the discretization error we use the condition $|c| > M|b|\|\mathbf{u}_{\text{in}}\|_{\max}^{M-1}$, which is stronger than the condition $|c| > |b|\|\mathbf{u}_{\text{in}}\|_{\max}^{M-1}$ for PDE stability. This appears to be a fundamental condition due to the nonlinearity, because the derivative of the order-$M$ nonlinearity produces a factor of $M$.

Next, if we take $\varepsilon_{\text{disc}} \propto \varepsilon$, then solving Eq. (3.1.107) for $n$ gives

$$n = \Omega\left(\left[\frac{C(u,k)\left(\frac{e}{2}\right)^{2k}}{\left|c + M|b|\|\mathbf{u}_{\text{in}}\|_{\max}^{M-1}\right|}\frac{1}{\varepsilon}\right]^{\frac{2d}{2(2k-1)+d}}\right). \tag{3.1.108}$$

That is, this choice of $n$ is sufficient to give $\varepsilon_{\text{disc}}$ as some set fraction of $\varepsilon$. In practice we would choose the minimum $n$ needed to give the desired accuracy, so we would choose $n$ proportional to the expression in Eq. (3.1.108). For simplicity of the solution for $n$, we have used

$$1 - \exp\left\{\left(c + M|b|\|\mathbf{u}_{\text{in}}\|_{\max}^{M-1}\right)t\right\} \leq 1\,. \tag{3.1.109}$$

In Appendix C.4 we show how the discretization error is reduced with the number of grid points with different orders of discretization and a simple toy model.

The benefit of having fewer grid points comes with the drawback of having a less sparse operator. The block encoding of $F_1$ is performed by a linear combination of unitaries over $(2k+1)$ basis states with amplitudes given by a table, and so has complexity in terms of elementary gates proportional to $k$. That is not immediately obvious from Corollary 3.1.6, because it gives complexity in terms of block encodings of $F_1$ and $F_M$. It is also possible for $C(u,k)$ to increase with $k$. In a real implementation it would therefore be desirable to choose an optimal $k$ to minimise the complexity, instead of taking $k$ as large as possible.

### 3.1.7 Conclusion

In this study, we proposed a set of improvements to quantum algorithms for nonlinear differential equations via Carleman linearization, eliminating some of the exponential scalings seen in prior work. We have examined both ODEs, and a class of nonlinear PDEs corresponding to reaction-diffusion equations [Liu+23]. These improvements include

- rescaling the original dynamics,
- a truncated Taylor series for the time evolution,
- higher-order spatial discretization of the PDEs, and
- tighter bounds on the error of Carleman linearization.

The rescaling boosts the success probability, needing exponentially fewer steps for the amplitude amplification to obtain the solution component of interest in the linearised ODE system. That is vital to enable the complexity of the PDE solver to be sublinear in the number of discretization points. The solution approximation via the truncated Taylor method gives a near-linear dependence on $T$, the total evolution time. The higher-order spatial discretization greatly improves the complexity of the quantum solution of PDEs, because it reduces the number of discretization points needed, which is needed to avoid stability problems.

We show that the stability criterion for PDEs, rescaling, Carleman linearization, and stability criterion for ODE solvers all interact in a way that makes the solution of PDEs more challenging than was appreciated in prior work. In particular, the stability criterion for PDEs is in terms of a

max-norm, but rescaling by the 2-norm is required to obtain a reasonable probability for the correct component of the Carleman vector. But, rescaling by the 2-norm can make the resulting system of linearised equations unstable, which causes the ODE solver to have exponential complexity. If the discretised PDE is still stable in terms of the 2-norm, then the resulting quantum algorithm will still be efficient. Because the 2-norm increases as $\sqrt{n}$ in the number of discretization points, the number of those points should be made as small as possible, which is why it is crucial to use the higher-order discretization.

In future work, one could devise a less restricted quantum algorithm for solving nonlinear PDEs via some other approach. The feature that the linearised equations can be unstable even though the nonlinear equation is stable suggests that an alternative linear equation solver may be efficient. The reason why the condition number is large (causing the inefficiency) is that the solution can grow exponentially over time, but for an initial vector that is not of the Carleman form. A solver that is able to take advantage of the restricted form of states could potentially be efficient.

Furthermore, there are a number of important generalizations that can be made to the type of differential equations. Instead of just including a nonlinear term of order $M$, one could include all nonlinear orders up to $M$. That could also be used to analyse the effect of driving because the method used for quadratic nonlinearities would produce nonlinearities at a range of orders. A further generalization that could be considered is time-dependent differential equations. These generalizations can be made in a simple way in the quantum algorithm, but the analysis to bound the error would be considerably more complicated.

## 3.2 Efficient Quantum Algorithm for Differential Equations under Constraints and Boundary Conditions

This section covers the following preprint:

> Philipp Schleich, Tyler Kharazi, Xiangyu Li, Jin-Peng Liu, Alán Aspuru-Guzik, and Nathan Wiebe. "Arbitrary Boundary Conditions and Constraints in Quantum Algorithms for Differential Equations via Penalty Projections". In: *arXiv preprint arXiv:2506.21751* (2025).

### 3.2.1 Introduction

Ordinary and partial differential equations (ODEs, PDEs) describe many processes and phenomena in science and engineering. The modeling of these phenomena typically involves describing interactions of the system studied with its environment through 'boundaries', which imposes constraints on the solution of these equations. Hence, for a meaningful numerical solution it is crucial to represent such constraints and boundary conditions. In addition, solving DEs on classical computers is cursed by dimensionality. Quantum algorithms for differential equations have been developed in order to mitigate exponential cost in the dimension. Such algorithms are based on, e.g., encoding the solution in a linear system [Ber14; BCOW17; BC24; Kro23; CL20; CLO21], performing explicit 'time-marching' [FLT23; BS24], or making use of integral identities [ALWZ25; AT23; ALL23; ACL23; JLY24; LLLL25]. Recently, algorithms that map the ODE to a Lindblad evolution has been explored in [SGAZ24], drawing from quantum algorithms that are able to simulate open quantum systems. Particular systems studied are mostly focused around popular models from classical differential equations courses such as the Poisson equation [Cao+13] or more general elliptic equations [CLO21; BNWA23], the advection equation [NJ25] or the wave equation [CJO19] and also the Maxwell equations [JLM24] or nonlinear equations such as reaction diffusion equations [Liu+21; Liu+23; Kro23; CSMB23] or fluid dynamics [Li+25; BS25; Pen+24]. Most studies so far focus on input models that assume a structured geometry with no (or, periodic) boundary conditions which limits applicability. Imposing the boundaries in the encoding of the generator of the dynamics is possible for several techniques [CLO21; NDS23; JLLY24a; Liu+23], based on other classical techniques such as complex absorbing potentials, perfect matched layers or Dirichlet-to-Neumann maps [JLLY24b] or analytic continuation [Kha+24]. However, there are some practical limitations to this in the way it can affect the respective access model of the system matrix, e.g. the complexity of a circuit construction of a specific block-encoding. Additionally, it is not immediately obvious how to approach boundary conditions in the recently popular Linear Combination of Hamiltonian Simulations (LCHS) algorithm [ALL23; ACL23], which is promising due to (near-)optimal scaling both in the number of state-preparation queries as well as the system matrix queries. Furthermore, [MZLG24] considers the implementation of complex absorbing potentials, where the main underlying dynamics are unitary. Beyond the specific concept of boundary conditions, we may be interested in more general constraints.

In this work, we implement boundary values or constraints on evolution equations by separating the dynamics into unconstrained and constrained dynamics using perturbation theory. Our

method is also capable to model interface conditions, which was also explored in the context of quantum algorithms in [JLLY24a], as well as Robin boundary conditions. Conceptually, shows connections to a wide range of existing approaches that relate to perturbation methods and penalty methods [LYLY16]. Primarily, there are ties to 'complex absorbing potentials' for quantum dynamics [MPNE04], as previously used in quantum algorithms for non-unitary dynamics [ALL23; MZLR24]. Another interesting study in the simulation of acoustic waves uses counter-acting waves to annihilate the solution [Tsy03] – imagine numerical noise-cancelling headphones. Specifically, our approach adds a projection scaled by $\mathrm{i}\lambda$ to the generator of a dynamical system. A physical interpretation of this may be seen as measurement with (very fast) frequency $\lambda$ on the domain of the projector, similar to the quantum Zeno effect. Thereby, our bounds also closely resemble respective bounds related to the Zeno effect, generalized adiabatic dynamics and the rotating-wave approximation [Bur+19; BFGY22]. General results from perturbation theory suggest what we are looking for is a small signal-to-noise ratio [Van75]. For an ODE $\frac{\mathrm{d}}{\mathrm{d}t}v(t) = A_0 v(t)$, this can be envisioned as the systems 'energy' divided by the strength of the perturbation, $\frac{|\langle v(0), A_0 v(0)\rangle|}{\lambda}$; this gives us intuition about what necessary frequency in the perturbation to expect. Our bounds in Section 3.2.2 confirm this intuition. Using interaction-picture simulation, which is possible if the projection is chosen to be fast-forwardable as will be the case as we demonstrate later, we can expect a modest overhead of $O(\log(\lambda))$ to solving differential equations (DEs) when adding constraints in this manner.

Within this work, we focus on ordinary differential equations under constraints. This is equivalent to readily discretized partial differential equations under boundary conditions. We further focus on evolution equations of the kind $\frac{\mathrm{d}}{\mathrm{d}t}v(t) = A(t)v(t) + b(t)$ and a "time" parameter $t \geq 0$ and will consider stationary boundary value problems in future work. Fig. 3.5 depicts the general approach: When evolving an ODE, the addition of the penalty projection allows us to produce a final state that satisfies a set of constraints, as represented by the projection onto an infeasible space, up to some accuracy $\varepsilon$.

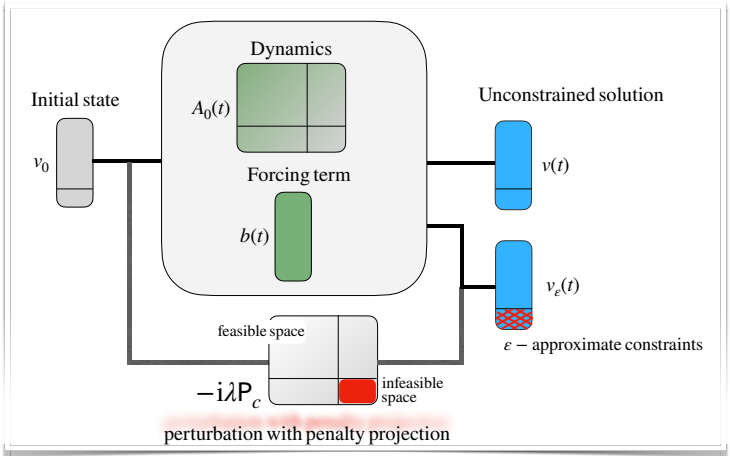



Figure 3.5: Enforcing constraints in solution $v(t)$ to an ordinary differential equation defined by $A_0(t)$ and source term $b(t)$, by adding a penalty function defined by a projection operation $\mathsf{P}_c$ that projects onto the infeasible, constraint space. By appropriately choosing the penalty strength $\lambda$, this approach leads can enforce constraints that are efficiently representable by a projection up to arbitrarily small accuracy $\varepsilon > 0$.

### Notation

Throughout this work we assume operators as finite-dimensional representations, typically as $n^d$-dimensional complex matrices, where $n$ is the spatial grid number in each coordinate and $d$ is the spatial dimension. Results reported pertaining to partial differential equations thus assume suitable numerical discretization. We often use bracket notation for inner products with a comma for non-normalized (non-quantum states) vectors, $\langle a, b\rangle \equiv a^\dagger b$. Whenever we do not specify the type of norm, we use the spectral or operator norm for matrices and $\|A\|$ and the $\ell_2$-norm for vectors $\|v\| = \|v\|_{\ell_2}$. Bold notation $\boldsymbol{j}$ denotes multi-indices so that $\boldsymbol{j} \in \mathbb{N}^d$, or quantities after spatial discretization. When we say "stable" in this work in the context of a differential equation, we mean that for an ODE $\frac{\mathrm{d}}{\mathrm{d}t}v = Av$, $\mathrm{Re}(A) \preceq 0$, which is a sufficient condition for numerical stability. We use square-bracket notation for ordered sets of integers: $[n, m] = \{n, n+1, \ldots, m\}$ for any $n < m \in \mathbb{Z}$. Most commonly, for $n > 0$, $[n] \equiv [1, n]$, $[n]_0 \equiv [0, n]$. With $\mathbb{I}$ we denote the identity matrix; if dimensionality is not clear from context, $\mathbb{I}_N$ acts on $N$ qubits, i.e., on $\mathbb{C}^{2^N}$. Some operators/matrices are denoted by sans-serif font, e.g., $\mathsf{P}$ for projections. The relations $\eqsim, \gtrsim, \lesssim$ are used to denote the relations up to a constant or asymptotically similar/greater/smaller.

### Problem Setting

Within this work, we consider the solution of ODEs given constraints, using a quantum algorithm. To motivate the setting, let $\mathsf{A}$ be an operator on continuously differentiable functions on some finite-dimensional vector space, let $v(t) : [0, T] \to \mathbb{C}^{n^d}$, and let $\mathsf{P} : \mathbb{C}^{\mathsf{n}^\mathsf{d}} \to \mathbb{C}^{\mathsf{n}^\mathsf{d}}$ be a linear constraint function, so that $\mathsf{P}(v_{\mathrm{bad}}) = \lambda v_{\mathrm{bad}}$ with some penalty $\lambda > 0$ if the constraint is not satisfied and $\mathsf{P}(v_{\mathrm{good}}) = 0$ if it is. Then, the kernel of $\mathsf{P}$ spanned by $v_{\mathrm{good}}$ makes up the constraint-admissible subspace a degenerate eigenspace of $\mathsf{P}$ with eigenvalue $\lambda > 0$ spanned by $v_{\mathrm{bad}}$ forms the constraint-inadmissible subspace. We seek a solution $v(t)$ to the constrained ODE such that $v(t)$ satisfies

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}v(t) &= \mathsf{A}\big(v(t)\big) \\ \mathsf{P}(v(t)) &= 0, \end{aligned} \tag{3.2.1}$$

which implies that we want $v_{\mathrm{bad}}(t) = 0$. Our nomenclature follows penalty methods in constrained optimization [LYLY16]: A penalty function is a function that satisfies (i) $\mathsf{P}$ is continuous, (ii) $\mathsf{P}(x) \geq 0$ for any valid input $x$, and (iii) $\mathsf{P}(x) = 0$ if and only if $x$ is in the feasible region, i.e., it satisfies the constraint. Then, an optimization problem via such a penalty function has as limit point the solution to the constrained optimization problem. Later on, we outline that the problem setup in Problem 3 indeed satisfies these requirements. Moreover, there is an equivalence between stationary solutions of a dynamical system and optimization. Quantum algorithms for this purpose exist, as in [CSW25], and could be extended by the penalty projections in our work to satisfy constraints.

We are particularly interested in settings where the constraint arises from boundary conditions on spatially discretized PDEs. In this case, $\mathsf{P}$ can take the form of a projector being applied to either the solution or its derivative to enforce Dirichlet or Neumann boundary conditions respectively on a subset of grid points. Dirichlet boundary conditions are conditions that constrain the solution to follow specific values, whereas Neumann conditions constrain the derivative (within domains in 3D space, typically the surface normal derivative). Linear combination of Dirichlet and Neumann

condition leads to what is typically called Robin boundary conditions.

**Definition 3.2.1** (Computational domain)**.** *We consider a problem over a $n^d$-dimensional finite space $\mathbb{C}^{n^d}$. This can expressed in terms of basis elements $\{\boldsymbol{j}\}$ where each label $\boldsymbol{j} = (j_1, \ldots, j_d) \in ([n-1]_0)^d$ is a d-dimensional tuple indexing a basis element; there are $n \in \mathbb{N}$ elements per dimension, such as representing each spatial dimension of the problem domain. The following definitions are used throughout the paper to describe the relevant subsets of the problem domain that appear in our work.*

1. *We first define the index set $\mathcal{I}_\Omega$ making up the unconstrained space, or "inside the domain", consisting of all basis elements*

$$\mathcal{I}_\Omega = \{\boldsymbol{j} \in [n-1]_0^d \mid \boldsymbol{j} \text{ is unconstrained}\}.$$

   *This corresponds to points inside a domain, i.e. the set of all points where the constraint operation in Eq.* (3.2.1) *acts trivially on.*

2. *Further, there is the set with value constraints*

$$\mathcal{I}_{\Gamma_D} = \{\boldsymbol{j} \in [n-1]_0^d \mid \boldsymbol{j} \text{ has a constraint on the value}\}$$

   *as the case for a Dirichlet boundary and the set spanning the derivate constraint*

$$\mathcal{I}_{\Gamma_N} = \{\boldsymbol{j} \in [n-1]_0^d \mid \boldsymbol{j} \text{ has a derivative constraint}\}$$

   *such as on a Neumann boundary.*

3. *For every $\boldsymbol{j} \in \mathcal{I}_{\Gamma_N}$, we have the neighbour set*

$$\zeta_{\boldsymbol{j}} = \{\boldsymbol{k} \in [n-1]_0 \mid \boldsymbol{k} \text{ neighbours } \boldsymbol{j} \text{ in a discretized directional derivative}\}.$$

   *By neighbouring we mean at most distance one per each coordinate $1 \leq l \leq d$.*

   *For now, we do not pose additional constraints on neighbouring points but will do so when we discuss implementing derivative constraints.*

4. *We call the constrained set, or boundary set, $\mathcal{I}_\Gamma = \mathcal{I}_{\Gamma_D} \sqcup \mathcal{I}_{\Gamma_N}$,*

5. *We require mutual satisfiability of the constraints; formally, $\mathcal{I}_{\Gamma_D} \cap \mathcal{I}_{\Gamma_N} = \emptyset, \mathcal{I}_\Omega \cap \mathcal{I}_\Gamma = \emptyset$. This ensures orthogonality of all subspaces spanned by the basis vectors coming from the index sets.*

**Definition 3.2.2** (Linear space associated to computational domain)**.** *Given bases indexed according to Definition 3.2.1, call the space associated to the index set $\mathcal{I}_\Omega$ feasible or unconstrained space*

$$S = \operatorname{span}\{|\boldsymbol{j}\rangle \mid \boldsymbol{j} \in \mathcal{I}_\Omega\} \subseteq \mathbb{C}^{n^d}$$

*and the constrained space associated to $\mathcal{I}_\Gamma$ we call $S_c \subseteq \mathbb{C}^{n^d}$, which again is split into the two orthogonal subspaces $S_c^D$ and $S_c^N$ for $\mathcal{I}_{\Gamma_D}$ and $\mathcal{I}_{\Gamma_N}$, respectively. The solutions we consider live in the space $\overline{S} = S \oplus S_c$.*

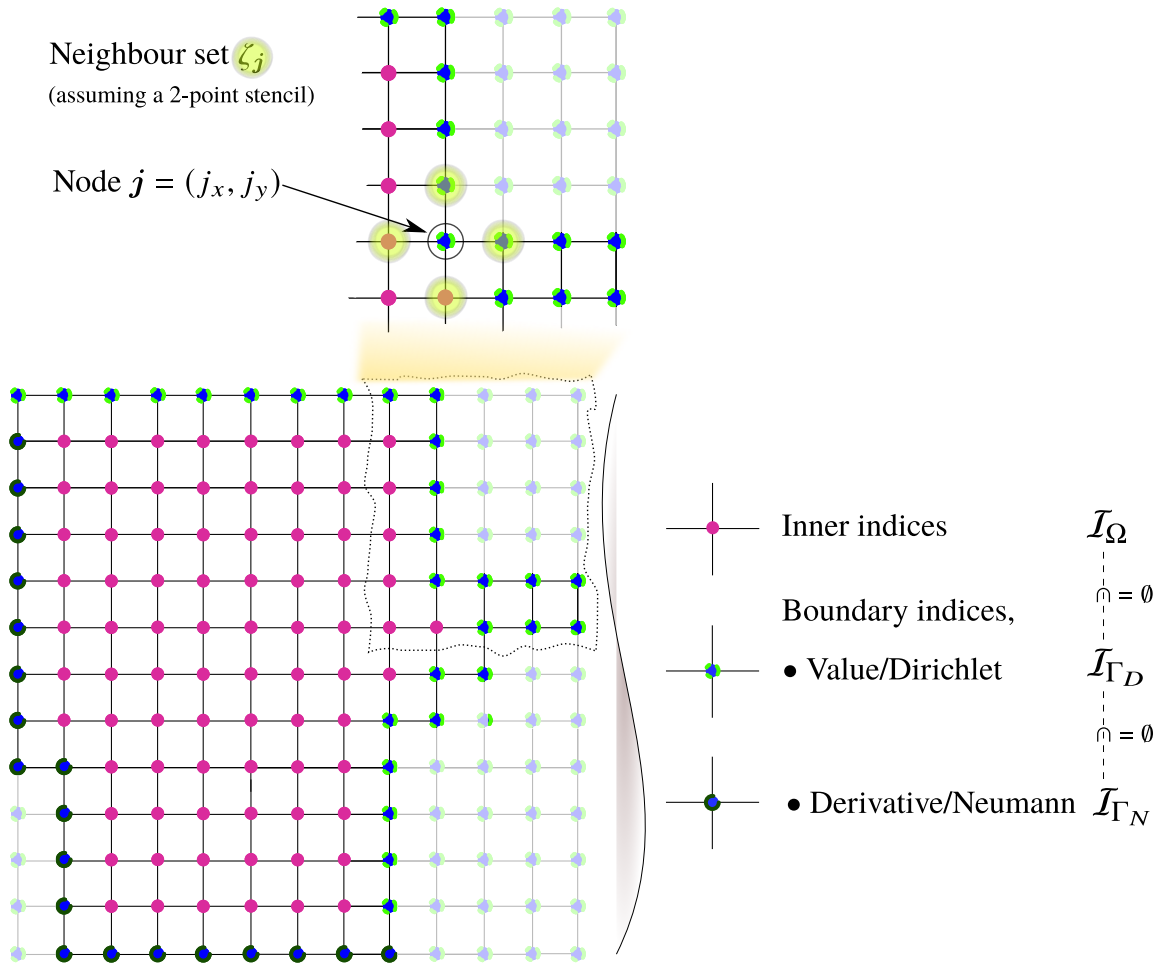



Figure 3.6: Visualization of relevant index sets as a graph depicted as a uniform grid. Recall that "inner" points span the unconstrained space $S$. The boundary, i.e., the constrained space, is composed by $S_c = S_c^D \oplus S_c^N$.

We further define the following projections.

**Definition 3.2.3** (Projection on constrained space)**.** *Consider a feasible (unconstrained) space $S$ and infeasible (constrained) spaces $S_c, S_c^D, S_c^N$ as defined in Definition 3.2.1. Then, for $v \in \overline{S}$, we have the following projections,*

$$\mathsf{P}_c v = \begin{cases} v, & v \in S_c \\ 0, & v \in S \end{cases}, \tag{3.2.2}$$

$$\mathsf{P}_c^{\perp} = \mathbb{I} - \mathsf{P}_c, \tag{3.2.3}$$

*so that $\mathsf{P}_c$ projects onto the constrained space $S_c$ and $\mathsf{P}_c$ to its complement $S$. Here, $\mathbb{I}$ is the identity on $\overline{S}$. We can further decompose $\mathsf{P}_c = \mathsf{P}_c^D + \mathsf{P}_c^N$ so that $(S_c^D)^{\perp} = S \oplus S_c^N = \ker(\mathsf{P}_c^D)$ and $(S_c^N)^{\perp} = S \oplus S_c^D = \ker(\mathsf{P}_c^N)$.*

**Definition 3.2.4** ($\ell_2$-norms over computational spaces)**.** *Given the computational spaces from Definition 3.2.2 and projectors onto the spaces from Definition 3.2.3, we define the $\ell_2$-norms over the feasible and infeasible spaces. Let $a \in \overline{S}$, then*

$$\|a\|_{\ell_2, S_c} = \left( \sum_{\boldsymbol{j} \in \mathcal{I}_\Gamma} a_{\boldsymbol{j}}^2 \right)^{1/2} = \|\mathsf{P}_c a\|_{\ell_2} \tag{3.2.4}$$

*This is equivalent to the notion of a $\mathsf{P}_c$-inner product so that*

$$\langle a, \mathsf{P}_c a \rangle = \langle a, {\mathsf{P}_c}^2 a \rangle = \langle \mathsf{P}_c a, \mathsf{P}_c a \rangle = \|a\|_{\ell_2, S_c}^2. \tag{3.2.5}$$

*A $\|\cdot\|_{\ell_2, S}$-norm can be defined equivalently through a $\mathsf{P}_c^{\perp}$-inner product.*

Next, we define the problem setting we generally consider in this work.

**Problem 2** (Constrained Discrete Initial Value Problem)**.** *We consider a finite-dimensional initial value problem with solution vector $v(t) : \mathbb{R}_+ \to \overline{S}$ and a matrix $A : \overline{S} \to \overline{S}$ as dynamical generator. We seek approximate solutions to the constrained dynamics*

$$\frac{\mathrm{d}}{\mathrm{d}t} v(t) = A v(t) + b(t), \quad t \geq 0 \tag{3.2.6}$$

$$\begin{aligned} \mathsf{P}_c^D v(t) &= g \quad \text{and } t \geq 0, \\ \mathsf{P}_c^N v(t) &= h \quad \text{and } t \geq 0, , \\ v(0) &= v_0, \quad \mathsf{P}_c^D v_0 = v_0 \text{ and } \mathsf{P}_c^N v_0 = v_0. \end{aligned} \tag{3.2.7}$$

*That means there is initial data $v(t=0)|_S = v_0$ on the feasible space $S$ and satisfies the constraints on the constrained space $S_c$, i.e., $v(t=0)|_{S_c^D} = g$ and $v(t=0)|_{S_c^N} = h$.*

**Remark** (Projection matrices as constraint functions are penalty functions)**.** *Recall that, following [LYLY16], penalty functions need to satisfy (i) continuity, (ii) non-negativity and (iii) they evaluate to zero if and only if the preimage is element of the feasible region. Therefore the penalty projections in Problem 2 satisfy these requirements, as (i) they are matrices (bounded linear operators are continuous), (ii) projections have eigenvalues $0, 1$ and (iii) by Definition 3.2.3.*

**Remark.** *For the derivative constraint in Eq.* (3.2.7)*,* $\mathsf{P}_c^N$ *already embodies the notion of a finite difference approximation of a derivative constraint. Given a proper construction of* $\mathsf{P}_c^D$ *and* $\mathsf{P}_c^N$*, we can expect that the treatment of Dirichlet and Neumann conditions will be mostly equivalent. Later on, we will assume* $\mathsf{P}_c^D$ *and* $\mathsf{P}_c^D$ *commute and are orthogonal projections for efficient simulation. One way to ensure is is by assuming that every point can only fall under one constraint,* $\mathcal{I}_{\Gamma_D} \cap \mathcal{I}_{\Gamma_N} = \emptyset$*. Then, the (nontrivial) domains of the corresponding projections commute by construction as they do not overlap.*

### 3.2.2 Approximating Boundary Conditions with a Projection

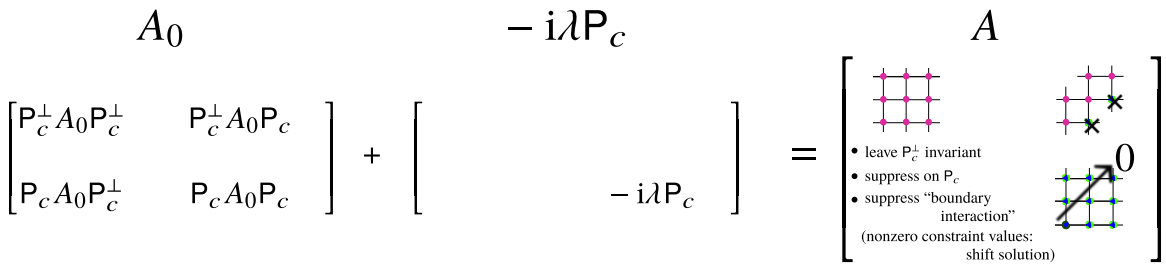



Figure 3.7: Inducing a constraint subspace via a penalty projection $-\mathrm{i}\lambda\mathsf{P}_c$ with $\lambda > 0$ and $\mathsf{P}_c$ the projection onto the constraint subspace. Then, dynamical evolution of $A = A_0 - \mathrm{i}\lambda\mathsf{P}_c$ leaves the $\mathsf{P}_c^\perp$-subspace invariant (pink vertices as in Fig. 3.6) and suppresses the solution on $\mathsf{P}_c$ (blue vertices).

#### Motivation

Now, we discuss a method to efficiently implement the solution of such boundary problems on a quantum computer. To that end, we take inspiration from the so-called complex absorbing potentials in quantum physics and chemistry [MPNE04]. Note that for the case when Eq. (3.2.6) has $A = -\mathrm{i}H$ anti-Hermitian and operator and a normalized initial state, the dynamics of $\psi(t, x)$ follows the Schrödinger equation, i.e.,

$$-\mathrm{i}\frac{\partial}{\partial t}\psi(t, x) = H\psi(t, x). \tag{3.2.8}$$

Then, a modification of the Hamiltonian with $\lambda > 0$

$$H \to H - \mathrm{i}\lambda\mathsf{P}_c \tag{3.2.9}$$

leads to enforcing that $\psi(t, x') \to 0$ for all $x' \in \Gamma$ on the boundary in the limit of $\lambda \to \infty$ if $\mathsf{P}_c = \int_\Gamma \mathrm{d}\,|x'\rangle\langle x'|$. In particular, we want $\lambda \gg \|H\|$. This is easy to see, as then in the solution

$$\psi(t, x) = \mathrm{e}^{-\mathrm{i}(H - \mathrm{i}\lambda\mathsf{P}_c)t}\psi(0, x), \tag{3.2.10}$$

components of the solution $\psi(t, x)$ on $\Gamma$ are suppressed to zero. The rest of $\psi(t, x)$ remains unchanged.

In what follows, we will show that the modification in Eq. (3.2.9) in the case for boundary conditions over more general, not necessarily quantum dynamics such as in Eq. (3.2.6) can work the same way. In a general setting, we consider the following modified dynamics governed by a

(potentially time-dependent) system matrix $A_0(t)$ and a forcing term $b(t)$,

$$\frac{\mathrm{d}}{\mathrm{d}t}v(t) = (A_0(t) - \mathrm{i}\lambda\mathsf{P}_c)v(t) + b(t). \tag{3.2.11}$$

Compared to the case of complex absorbing potentials in Eq. (3.2.9), $A_0(t)$ need not be Hermitian. As we will be working with orthogonal projections $\mathsf{P}_c, \mathsf{P}_c^\perp$ exclusively, this implies their Hermiticity; a more careful analysis could relax the assumptions to non-Hermitian penalty functions in future work. Additionally, a time-dependent constraint region can be thought of via a $\mathsf{P}_c(t)$.

We make the following observation here. In the case of quantum dynamics, Eqs. (3.2.8) and (3.2.9), a perturbation of the form $\mathrm{i}\lambda$ as in complex absorbing potentials "dissipates" the wavefunction on the boundaries. Our approach in Eq. (3.2.11) with positive $\lambda$ does not add dissipation per se. Rather, we can understand this e.g. under the realm of the rotating-wave approximation, where the evolution under a Hamiltonian $H = H_0 + H_1$ shows a separation of time-scales with respect to dynamics generated by $H_0$ and $H_1$, respectively. Then, one can say that the solution coming from the perturbative generator is negligible with respect to the initial dynamics in the sense that the on the unperturbed system's time-scale, the highly oscillatory perturbation averages to zero. In other words, the solution on the infeasible region then is perceived negligible compared to the solution on the feasible region.

Our analysis follows perturbation theoretic arguments, which is reasonable given the perturbation strength $\lambda$ is chosen to be much larger than the system $\|A_0\|$ and forcing $\|b(t)\|$. For the following arguments, we change the point-of-view to strong perturbation, in the sense that we consider the system dynamics generated by $A_0$ as a perturbation of the constraint projection $-\mathrm{i}\mathsf{P}_c$ with perturbation parameter $\frac{1}{\lambda}$. Then, a natural way to estimate the error is to consider 'transition elements' with respect to the projections $\mathsf{P}_c^\perp, \mathsf{P}_c$, as expectations $\langle v(t'), \mathsf{P}v(t)\rangle$. For $t = t'$, the Kubo formula [Kub57] gives a natural pathway to find an estimate. We are interested in a non-Hermitian formulation, which was developed in [SDM22] and further discussed in [GH22]. Building on this, we provide a generalization that also covers time-dependent generators for the unperturbed dynamics, and a forcing term. This enables the remainder of our analysis enforce the constraints.

Before stating the result, we introduce some context and notation. We are looking at a dynamical system $\frac{\mathrm{d}}{\mathrm{d}t}v(t) = A(t)v(t) + b(t)$, where $A(t) = H(t) + \zeta V(t)$ so that $V(t)$ is a perturbation of the original dynamics $H(t)$ in the sense that $\zeta\|V(t')\| \ll \|H(t')\|$ at all times. For now, $A(t), H(t), V(t)$ are arbitrary complex, square matrices and we take $1 \gg \zeta > 0$ and note that this can also be simply reformulated to cases of where the scaling term is time-dependent. The Kubo formula allows to estimate the difference between the expectation of an observable $P$ with respect to a perturbed compared to an unperturbed solution vector – in our more general case, the quadratic form of a matrix. We will use the following quantities:

- Let $\overline{V}(t, t')$ be the cumulative perturbation in the interval $[t'; t]$ for $t' \leq t$, $\overline{V}(t, t') := \int_{t'}^{t} \mathrm{d}\tau\, V(\tau)$.
- We denote by $T_t(\cdot)$ the time propagation of the argument (borrowing notation from dynamical semi-groups). In Proposition 3.2.5, this relates to the unperturbed dynamics, $T_t := \mathcal{T}\int_0^t \mathrm{d}s\, \exp\Big(\int_s^t \mathrm{d}s' H(s')\Big)$. Then, the solution to the inhomogeneous problem is given by $T_t(v(0)\delta(t) + b(t))$ with the Dirac-delta distribution $\delta(t)$.
- We use $\sigma(t', t)$ for the outer product of solutions at times $t', t$, i.e., $\sigma(t', t) := v(t')v(t)^\dagger$. This

notion is similar to the density matrix of pure quantum states, however with non-normalized complex vectors $v(t)$.

**Proposition 3.2.5** (A non-Hermitian, time-dependent, inhomogeneous Kubo formula)**.** *Let $A(t) = H(t) + \zeta V(t)$ be a perturbed dynamical generator with $\zeta\|V(t')\| \ll \|H(t')\|$ for any $t \geq t' \geq 0$ and $A(t), H(t), V(t)$ complex matrices. Let $v(t)$ be the solution to $\frac{\mathrm{d}}{\mathrm{d}t}v(t) = A(t)v(t) + b(t)$ with initial data $v(0)$. Further, suppose we are interested in measuring the expectation value of a matrix $P$. Then, the effect due to perturbation $\zeta V(t)$ on the expectation of $P$ up to first order in the strength of the perturbation*

$$\langle P(t)\rangle - \langle P\rangle_0 = \zeta \int_0^t \mathrm{d}t' \, \frac{\mathrm{tr}\left[\left\{P, \overline{V}(t,t')\sigma(t',t)\right\}_\sim\right]}{\mathrm{tr}\left[\sigma(t,t)\right]} + O\big(\zeta^2\big) \tag{3.2.12}$$

*where $\langle P(t)\rangle$ is the perturbed expectation and $\langle P\rangle_0$ is the expectation due to the unperturbed dynamics generated by $H(t)$. Further, there is the modified anticommutator $\{X, Y\}_\sim = XY + Y^\dagger X$, a density augmented by the forcing term $b(t)$ through $\sigma(t',t) = T_{t'}(v(0)\delta(t') + b(t'))\big(T_t(v(0)\delta(t) + b(t))\big)^\dagger$ where $T_t(u) = \int_0^t \mathrm{d}s\, \mathcal{T}\exp\Big(\int_s^t \mathrm{d}t'\, H(t')\Big)u(s)$ and $\overline{V}(t,t') = \int_{t'}^t \mathrm{d}\tau\, V(\tau)$.*

**Remark** (Original Kubo formula)**.** *For completeness, we briefly restate the original Kubo formula, as derived in [Kub57], with adapted notation. We are given Hermitian $H, V(t)$ as well as a Hermitian observable $P$, so the dynamics of a Schrödinger equation through the Hamiltonian $H$ is perturbed by time-dependent $V(t)$. Then, with $P(t)$ the observable evolved in the Heisenberg picture,*

$$\langle P\rangle_t - \langle P\rangle_0 = -\mathrm{i}\int_0^t \mathrm{d}s\, \langle[P(t), V(s)]\rangle_0, \tag{3.2.13}$$

*where $\langle\cdot\rangle_0$ denotes the expectation with respect to the unperturbed dynamics and $P(t)$ is the Heisenberg-evolved observable with respect to the perturbed dynamics.*

*We point out the following differences compared to Proposition 3.2.5. Non-Hermitian original dynamics lead to a necessary re-normalization (as observed in [SDM22; GH22]), as well as replacing the commutator by a (modified) anticommutator. Additionally, time-dependence of the original dynamics induce the time-cumulative perturbation $\overline{V}$ and the definition of $\sigma$ allows to add inhomogeneities to the solution.*

The proof of Proposition 3.2.5 is given in Appendix D.3. In the next section, we obtain tighter bounds on the error for special cases – such as time-independent $A_0$ or $b(t) = 0$ – with less general technique than Proposition 3.2.5. Therefore, we are looking into specific bounds under different sets of assumptions.

### Approximation Guarantees

In the upcoming section, we provide error bounds on the approximation of constraints such as given in Problem 2. We will structure this argumentation as follows. First, we consider the case of only one type of constraint, i.e., we do not split $S_c$ into $S_c^D$ and $S_c^N$ using only the projective properties of $\mathsf{P}_c$. This is sufficient, as we will show how to construct the projections so that the constraints on the different boundary strips will not affect each other, and $[\mathsf{P}_c^D, \mathsf{P}_c^N]$, aligning with our assumption in Definition 3.2.2 that $S_c^D \perp S_c^N$. Once we have derived guarantees for general constraints, we then

proceed to connect this to the case of Dirichlet and Neumann boundary conditions in discretized PDEs in Section 3.2.2.

First, we consider the more general problem of

**Problem 3** (Projection-Constrained Discrete Initial Value Problem)**.** *Under the same assumptions as in Problem 2, we define*

$$\frac{\mathrm{d}}{\mathrm{d}t}v(t) = (A_0(t) - \mathrm{i}\mathsf{P}_c)v(t) + b(t), \quad t \geq 0, \tag{3.2.14}$$

*where $\mathsf{P}_c$ is a Hermitian projection and $\lambda > 0$ is a penalty term that needs to be chosen. Then, given $\varepsilon > 0$, we seek $\lambda$ such that*

$$\|v(t)\|_{\ell_2, S_c} \leq \varepsilon, \tag{3.2.15}$$

*where $v(t)$ is understood to satisfy a constraint with respect to a projection $\mathsf{P}_c$ if $\mathsf{P}_c v(t) = 0$ (Definition 3.2.3).*

To verify that Problem 3 using the modified dynamics shows the sought-after behaviour, we need to show the following:

1. The solution error within the feasible space $S$ is small.
2. We can find $\lambda$ so that the approximation error on the boundary quantified by Eq. (3.2.4) is small.

Let us start by showing that Item 1 holds true under the problem setup in Problem 3.

**Lemma 3.2.6** (Error within the feasible space)**.** *The solution $v(t)$ to Problem 3 within the feasible space $S$ and the one to Problem 2 are equivalent.*

*Proof.* Let $u(t)$ follow the unconstrained dynamics $\frac{\mathrm{d}}{\mathrm{d}t}u = Au(t)$ and $v(t)$ the constrained dynamics, $\frac{\mathrm{d}}{\mathrm{d}t}v(t) = (A - \mathrm{i}\lambda\mathsf{P}_c)v(t)$ with the same initial conditions, $u(0) = v(0) = w$. Then, consider

$$\|u(t) - v(t)\|^2_{\ell_2, S} = \|(\mathrm{e}^{At} - \mathrm{e}^{A - \mathrm{i}\lambda\mathsf{P}_c)t})w\|^2_{\ell_2, S}. \tag{3.2.16}$$

Now we use the definition of $\|\cdot\|_{\ell_2, S}$ from Definition 3.2.4 to see that $\mathsf{P}_c^\perp$ leaves the norm over this space invariant and

$$\|(\mathrm{e}^{At} - \mathrm{e}^{A - \mathrm{i}\lambda\mathsf{P}_c)t})w\|^2_{\ell_2, S} \leq \|\mathsf{P}_c^\perp \mathrm{e}^{At} - \mathsf{P}_c^\perp \mathrm{e}^{A - \mathrm{i}\lambda\mathsf{P}_c)t}\|^2_{\ell_2, S} \|w\|^2_{\ell_2, S}. \tag{3.2.17}$$

We want to show that the difference between the time propagators in the $\|\cdot\|_{\ell_2, S}$-norm vanishes. To that end, use a Taylor series expansion of the matrix exponential and use that $\mathsf{P}_c^\perp$ is a projection to see that

$$\mathsf{P}_c^\perp \mathrm{e}^{(A - \mathrm{i}\lambda\mathsf{P}_c)t} = \mathsf{P}_c^\perp \left( \sum_{k \geq 0} \frac{((A - \mathrm{i}\lambda\mathsf{P}_c)t)^k}{k!} \right) = \sum_{k \geq 0} \frac{t^k}{k!} \left( \mathsf{P}_c^\perp A - \mathrm{i}\lambda \mathsf{P}_c^\perp \mathsf{P}_c \right)^k. \tag{3.2.18}$$

Observe that $\mathsf{P}_c^\perp \mathsf{P}_c = 0$ by definition, so $\mathsf{P}_c^\perp \mathrm{e}^{At} = \mathsf{P}_c^\perp \mathrm{e}^{A - \mathrm{i}\lambda\mathsf{P}_c t}$. This means Eq. (3.2.17) can be upper-bounded by zero and $\|u(t) - v(t)\|_{\ell_2, S} = 0$ for any $t \geq 0$. □

**Remark.** *For time-independent $\mathsf{P}_c$, Lemma 3.2.6 holds equivalently for time-dependent $A(t)$ and constant $\mathsf{P}_c$ by expanding a time-ordered exponential in Eq.* (3.2.18) *with a Dyson series. Furthermore, this statement is independent of whether the initial condition satisfies the constraint exactly or $\exists 1 > \varsigma \geq 0, \|\mathsf{P}_c w\|_{\ell_2} < \varsigma$ (implying that $\|\mathsf{P}_c^{\perp} w\|_{\ell_2} \geq 1 - \varsigma$).*

In what follows, we look at the approximation of different constraints in order to show Item 2. More specifically,

- $A$ is stable, i.e., its real part is negative semi-definite: $\mathrm{Re}(A) \preceq 0$. Often, this is also denoted as $A$ having non-positive logarithmic norm (discussion in [Kro23]).
- Given final time $T > 0$, $\exists C_T > 0$, $\sup_{t\in[0;T]} \|\exp(At)\| \leq C_T$ (a similar assumption as [Kro23] makes). This can be generally achieved by scaling $\lambda$ by an additional factor of $C_T$.
- Time-dependent $A(t)$.

Generally, we assume that initial data satisfies the constraints. For the quantum implementation this can be ensured by the approach outlined in Eq. (3.2.154).

### Constant dynamics

**Lemma 3.2.7** (Error in the infeasible space under stable dynamics)**.** *We consider a system as in Problem 2 with dynamics generated by $A = A_0 - \mathrm{i}\lambda\mathsf{P}_c$, the real part of $A_0$ is negative semi-definite, $\lambda > 0$ and $\mathsf{P}_c$ the projector on the infeasible space as defined in Definition 3.2.3. Further, we require that the initial data $v(0)$ satisfies $\|\mathsf{P}_c v(0)\|_{\ell_2}^2 = 0$. Then, the solution error at time $t > 0$ in the infeasible space is bounded as follows,*

$$\|v(t)\|_{\ell_2, S_c}^2 \leq \varepsilon \tag{3.2.19}$$

*for $\lambda \geq \frac{2v_{\max}^2 \|A_0\|}{\varepsilon}$.*

*Proof.* We have ODE and solution as

$$\frac{\mathrm{d}}{\mathrm{d}t} v(t) = Av(t), \quad v(t) = \exp(At)v(0) \tag{3.2.20}$$

with initial condition $v(0)$. Moreover,

$$|\langle v(0), Av(0)\rangle| \leq |\langle v(0), A_0 v(0)\rangle| + \lambda |\langle v(0), \mathsf{P}_c v(0)\rangle| \leq \|v(0)\|^2 \|A_0\|. \tag{3.2.21}$$

We thus define $E_0 = \|v(0)\|^2\|A_0\|$. Using the assumptions that $\mathrm{Re}(A) \preceq 0$, we have that $\|v(0)\| = \max_{0\leq t'\leq t}\|v(t')\| =: v_{\max}$, then, for all times $t > 0$,

$$\langle v(t), Av(t)\rangle = \left\langle v(0), \exp(At)^{\dagger} A \exp(At) v(0)\right\rangle = \left\langle v(0), \exp(At)^{\dagger} \exp(At) Av(0)\right\rangle, \tag{3.2.22}$$

and we want to argue that

$$\left|\left\langle v(0), \exp(At)^{\dagger} \exp(At) Av(0)\right\rangle\right| \leq E_0 \tag{3.2.23}$$

under the assumption that $A_0$ is stable, i.e., $\mathrm{Re}(A_0) \preceq 0$. First, notice that $\left|\left\langle v(0), \exp(At)^\dagger \exp(At) A v(0)\right\rangle\right| \leq \left\|\exp(At)^\dagger \exp(At)\right\| |\langle v(0), Av(0)\rangle|$. Then, $\|\exp(At)^\dagger\| \|\exp(At)\| \leq \exp(\|\mathrm{Re}(A_0)\|t) \exp(\|\mathrm{Re}(A_0)\|t) = \exp(2\|\mathrm{Re}(A_0)\|t) \leq 1$. This shows the boundedness of the "initial energy" $E_0$ in the stable case.

Carrying on with

$$|\langle v(t), A_0 v(t)\rangle - \mathrm{i}\lambda \langle v(t), \mathsf{P}_c v(t)\rangle| \geq 0, \tag{3.2.24}$$

we can apply the reverse triangle inequality to obtain a bound on the other side

$$|\langle v(t), A_0 v(t)\rangle - \mathrm{i}\lambda \langle v(t), \mathsf{P}_c v(t)\rangle| \geq ||\langle v(t), A_0 v(t)\rangle| - |\lambda \langle v(t), \mathsf{P}_c v(t)\rangle|| \tag{3.2.25}$$

$$\geq -|\langle v(t), A_0 v(t)\rangle| + \lambda \langle v(t), \mathsf{P}_c v(t)\rangle . \tag{3.2.26}$$

Thus we can conclude, under the assumption of $\mathrm{Re}(A_0) \preceq 0$, that

$$|\langle v(t), \mathsf{P}_c v(t)\rangle| \leq \frac{E_0 + |\langle v(t), A_0 v(t)\rangle|}{\lambda} \leq v_{\max}^2 \frac{2\|A_0\|}{\lambda}, \tag{3.2.27}$$

where we used that $\mathsf{P}_c \succeq 0$. Note that $\langle v(t), \mathsf{P}_c v(t)\rangle$ corresponds to the 2-norm of $v(t)$ over the region $S_c$. This means that for stable DEs, we can choose $\lambda$ so that the solution norm in the infeasible space is small. □

**Lemma 3.2.8** (Error in the infeasible space under dissipative and normal dynamics with an inhomogeneity)**.** *We consider a system as in Problem 2 with dynamics generated by $A = A_0 - \mathrm{i}\lambda \mathsf{P}_c$, where the real parts of the eigenvalues of $A_0$ are non-positive, $\lambda > 0$ and $\mathsf{P}_c$ the projector on the infeasible space as defined in Definition 3.2.3. Here, we further consider an inhomogeneity $b(t') : [0, t] \to \mathbb{C}^{n^d}$ so that $\max_{0\leq t'\leq t} \|b(t)\|_{\ell_2} \leq B$. We require the initial condition $v(0)$ and $b(t')$ to be adapted to the constraints at all times $0 \leq t' \leq t$, so that vanish under the action of $\mathsf{P}_c$. Then, the solution error at time $t > 0$ in the infeasible space with Dirichlet condition $g = 0$ is bounded as follows,*

$$\|v(t)\|_{\ell_2, S_c}^2 \leq \varepsilon \tag{3.2.28}$$

*for $\lambda \geq \frac{2\|A_0\|}{\varepsilon}\left(v_{\max}^2 + 2v_{\max} tB + t^2 B^2\right)$ and where $v_{\max} \geq \max_{0\leq t'\leq t} \|v(t')\| = \|v(0)\|$.*

*Proof.* We have ODE and solution as

$$\frac{\mathrm{d}}{\mathrm{d}t} v(t) = Av(t) + b(t), \quad v(t) = \underbrace{\mathrm{e}^{At} v(0)}_{v_{\mathrm{h}}(t)} + \underbrace{\int_0^t \mathrm{d}s\, \mathrm{e}^{As} b(t-s)}_{v_{\mathrm{p}}(t)} \tag{3.2.29}$$

with initial condition $v(0)$. This allows us to express a quantity $\langle v(t), Av(t)\rangle$ $t > 0$ as a sum of those expressed by the homogeneous and particular solution,

$$\langle v(t), Av(t)\rangle = \underbrace{\langle v_{\mathrm{h}}(t), Av_{\mathrm{h}}(t)\rangle}_{=:(i)} + \underbrace{\langle v_{\mathrm{h}}(t), Av_{\mathrm{p}}(t)\rangle + \langle v_{\mathrm{p}}(t), Av_{\mathrm{h}}(t)\rangle}_{=:(ii)} + \underbrace{\langle v_{\mathrm{p}}(t), Av_{\mathrm{p}}(t)\rangle}_{=:(iii)} . \tag{3.2.30}$$

The purely homogeneous term follows from Lemma 3.2.7. We briefly recall that, thanks to the negative-semidefiniteness of $\mathrm{Re}(A_0)$,

$$|(i)| = |\langle v_{\mathrm{h}}(t), Av_{\mathrm{h}}(t)\rangle| \leq v_{\max}^2 \|A_0\|. \tag{3.2.31}$$

Now, we bound the second and third term for the case of time-dependent $b(t)$. Recall that the particular solution is

$$v_{\mathrm{p}}(t) = \int_0^t \mathrm{d}s\ \mathrm{e}^{As} b(t-s).$$

By assumption, $\exists B > 0, \max_t \|b(t)\|_{\ell_2} \leq B$ which implies that $|\langle b(t), Ab(t)\rangle| \leq B^2\|A\|\ \forall t > 0$. Next, we consider the third term that comes only from the particular solution,

$$\begin{aligned}
|(iii)| = |\langle v_{\mathrm{p}}(t), Av_{\mathrm{p}}(t)\rangle| =& \left|\left\langle \int_0^t \mathrm{d}s_1 \mathrm{e}^{As_1} b(t-s_1), A\int_0^t \mathrm{d}s_2 \mathrm{e}^{As_2} b(t-s_2)\right\rangle\right| \\
=& \left|\int_0^t \mathrm{d}s_1 \int_0^t \mathrm{d}s_2 \left\langle b(t-s_1), \mathrm{e}^{A^\dagger s_1}\mathrm{e}^{As_1 + A(s_2-s_1)} Ab(t-s_2)\right\rangle\right| \\
\leq& B^2\|A_0\| \int_0^t \mathrm{d}s_1 \int_0^t \mathrm{d}s_2 \left\|\mathrm{e}^{A^\dagger s_1}\mathrm{e}^{As_1 + A(s_2-s_1)}\right\| \\
\leq& B^2\|A_0\| \int_0^t \mathrm{d}s_1 \left\|\mathrm{e}^{As_1}\right\|^2 \int_0^t \mathrm{d}s_2 \left\|\mathrm{e}^{A(s_2-s_1)}\right\|
\end{aligned} \tag{3.2.32}$$

The appearance of only $\|A_0\|$ in the second-last inequality is due to $\mathsf{P}_c b(s) = 0\,\forall s$. We can start by bounding the norms of the exponentials through their maximum eigenvalues:

$$\left\|\mathrm{e}^{As}\right\| = \left\|\mathrm{e}^{\mathrm{Re}(A)s}\right\| \leq \mathrm{e}^{\hat{\mu}_R(A)s}, \tag{3.2.33}$$

with $\hat{\mu}_R(A) = \max_j \mathrm{Re}(\mu_j(A))$; and we recall that $\hat{\mu}_R(A) \leq 0$. This can be used to bound both exponentials in Eq. (3.2.32) and obtain

$$|\langle v_{\mathrm{p}}(t), Av_{\mathrm{p}}(t)\rangle| \leq B^2\|A_0\| \int_0^t \mathrm{d}s_1\ \mathrm{e}^{2\hat{\mu}_R(A)s_1} \int_0^t \mathrm{d}s_2\ \mathrm{e}^{\hat{\mu}_R(A)(s_2-s_1)}. \tag{3.2.34}$$

The solution to an integral of the form above for any non-zero $L$ is

$$\int_0^t \mathrm{d}s_1\ \mathrm{e}^{2Ls_1} \int_0^t \mathrm{d}s_2\ \mathrm{e}^{L(s_2-s_1)} = \frac{(1-\mathrm{e}^{Lt})^2}{L^2}. \tag{3.2.35}$$

Thus, we make a case distinction here regarding $\hat{\mu}_R(A) = 0$ or $\hat{\mu}_R(A) < 0$. If $\hat{\mu}_R(A) = 0$, the upper bound is $|\langle v_{\mathrm{p}}(t), Av_{\mathrm{p}}(t)\rangle| \leq B^2\|A_0\|$. If $\hat{\mu}_R(A) < 0$, we choose $L = \hat{\mu}_R(A)$ in Eq. (3.2.35), and overall

$$|\langle v_{\mathrm{p}}(t), Av_{\mathrm{p}}(t)\rangle| \leq B^2\|A_0\| \cdot \begin{cases} \frac{(1-\mathrm{e}^{\hat{\mu}_R(A)t})^2}{(\hat{\mu}_R(A))^2}, & \hat{\mu}_R(A) < 0, \\ t^2, & \hat{\mu}_R(A) = 0. \end{cases} \tag{3.2.36}$$

For the second term, which depends both on the homogeneous and the particular solution, we obtain

using the same techniques as for the previous terms,

$$\begin{aligned}
|(ii)| &= |\langle v_{\mathrm{h}}(t), Av_{\mathrm{p}}(t)\rangle + \langle v_{\mathrm{p}}(t), Av_{\mathrm{h}}(t)\rangle| \\
&= \left| \int_0^t \mathrm{d}s \, \left[ \left\langle \mathrm{e}^{At} v(0), A\mathrm{e}^{As} b(t-s) \right\rangle + \left\langle \mathrm{e}^{As} b(t-s), A\mathrm{e}^{At} v(0) \right\rangle \right] \right| \\
&= \left| \int_0^t \mathrm{d}s \, \left[ \left\langle v(0), \mathrm{e}^{A^\dagger t} \mathrm{e}^{As} A_0 b(t-s) \right\rangle + \left\langle b(t-s), \mathrm{e}^{A^\dagger s} \mathrm{e}^{At} A_0 v(0) \right\rangle \right] \right| \\
&\leq \|A_0\| \int_0^t \mathrm{d}s \, \left( \left| \left\langle \mathrm{e}^{At} v(0), \mathrm{e}^{As} b(t-s) \right\rangle \right| + \left| \left\langle \mathrm{e}^{As} b(t-s), \mathrm{e}^{At} v(0) \right\rangle \right| \right) \\
&\leq \|A_0\| \mathrm{e}^{\hat{\mu}_R(A)t} \int_0^t \mathrm{d}s \, \left( \left| \left\langle v(0), \mathrm{e}^{As} b(t-s) \right\rangle \right| + \left| \left\langle \mathrm{e}^{As} b(t-s), v(0) \right\rangle \right| \right) \\
&\leq \|A_0\| \mathrm{e}^{\hat{\mu}_R(A)t} \int_0^t \mathrm{d}s \, \left( 2 v_{\max} B \mathrm{e}^{\hat{\mu}_R(A)s} \right) \\
&\leq 2 v_{\max} B \|A_0\| \cdot \begin{cases} \frac{\mathrm{e}^{\hat{\mu}_R(A)t} - 1}{\hat{\mu}_R(A)}, & \hat{\mu}_R(A) < 0 \\ t, & \hat{\mu}_R(A) = 0. \end{cases}
\end{aligned} \tag{3.2.37}$$

Now we proceed as in Lemma 3.2.7 where we applied the reverse triangle inequality in Eq. (3.2.26) to obtain a situation where we can express a bound on $|\langle v(t), \mathsf{P}_c v(t)\rangle|$,

$$\lambda \left| \langle v(t), \mathsf{P}_c v(t)\rangle \right| - \left| \langle v(t), A_0 v(t)\rangle \right| \leq \left| \langle v(t), Av(t)\rangle \right|. \tag{3.2.38}$$

Moreover, we can identify that the upper bound we obtained on the $|\langle v(t), Av(t)\rangle|$ holds equivalently for the terms without the constraint projection, $|\langle v(t), A_0 v(t)\rangle|$, which is easy to see as all the matrix norm dependencies in the bound on the former simplify to $\|A_0\|$ thanks to initial conditions and forcing terms satisfying the constraint and the constraint term being purely imaginary, thus $\hat{\mu}_R(A) = \hat{\mu}_R(A_0)$. Hence, we can assemble the final bound using Eqs. (3.2.31), (3.2.36) and (3.2.37) and get

$$|\langle v(t), \mathsf{P}_c v(t)\rangle| \leq \frac{2\|A_0\|}{\lambda} \left( v_{\max}^2 + 2 v_{\max} B \left\{ \begin{matrix} \frac{\mathrm{e}^{\hat{\mu}_R(A)t} - 1}{\hat{\mu}_R(A)}, & \hat{\mu}_R(A) < 0 \\ t, & \hat{\mu}_R(A) = 0 \end{matrix} \right\} + B^2 \left\{ \begin{matrix} \frac{(\mathrm{e}^{\hat{\mu}_R(A)t} - 1)^2}{(\hat{\mu}_R(A))^2}, & \hat{\mu}_R(A) < 0 \\ t^2, & \hat{\mu}_R(A) = 0 \end{matrix} \right\} \right) \tag{3.2.39}$$

We can simplify this by using the bound considering $\hat{\mu}_R(A) = 0$, as $\frac{e^{tL} - 1}{L}$ is monotonously increasing on $L \in (-\infty, 0)$ and $\lim_{L \nearrow 0} \frac{e^{tL} - 1}{L} = t$. □

Furthermore, we study the case of $A_0$ that are not stable, i.e., the eigenvalues of $A_0$ are not all non-positive. Here, we consider two cases. Case (I), $A_0$ has positive eigenvalues but $A_0 v(0) v(0)^\dagger$ does not (as discussed in [ACL23, Eq. (10)]). Then, the analysis follows equivalently to the other Lemmas presented.
For Case (II), we have the following Lemma.

**Lemma 3.2.9** (Error in the infeasible space under non-stable dynamics)**.** *Under the same assumptions as Lemma 3.2.7, however there is a $0 < \hat{\mu}_{R,0} < \infty$ as the maximum real part eigenvalue of $A_0$, $\hat{\mu}_{R,0} = \max_j \mathrm{Re}(\mu_j(A_0))$. Then, we have*

$$\|v(t)\|_{\ell_2, S_c}^2 \leq \epsilon \tag{3.2.40}$$

*for* $\lambda \geq v_{\max}^2 \|A_0\| \frac{1+\exp(2\hat{\mu}_{R,0}t)}{\epsilon}$.

*Proof.* The proof follows the same structure as the one for Lemma 3.2.7. We can define an initial "energy" as in Eq. (3.2.21). Further on in Eq. (3.2.27), instead of bounding $\langle v(t), A_0 v(t)\rangle$ by the bound from Eq. (3.2.21), we use that

$$\left\| \mathrm{e}^{A^\dagger t} \mathrm{e}^{At} \right\| \leq \mathrm{e}^{2\hat{\mu}_{R,0}t}. \tag{3.2.41}$$

Hence we conclude with

$$|\langle v(t), \mathsf{P}_c v(t)\rangle| \leq \|A_0\| \frac{1+\mathrm{e}^{2\hat{\mu}_{R,0}t}}{\lambda}. \tag{3.2.42}$$

□

**Time-dependent dynamics** Next, we look into enforcing constraints under time-dependent dynamics. These considerations are mostly based on the Kubo formula [Kub57], which allows us to estimate the difference in an expectation value – such as the projection onto the infeasible domain as a notion of constraint error – under a perturbation. As mentioned previously, we consider a strong perturbation, so that $\mathsf{i}\mathsf{P}_c \to \mathsf{i}\mathsf{P}_c + \frac{1}{\lambda}A_0(t)$ with $\frac{\|A_0(t)\|}{\lambda} \ll 1$. We consider the homogeneous case with $b = 0$ in Lemma 3.2.10, where we can make use of existing results from [SDM22; GH22] and an inhomogeneous case, where we use the result in Proposition 3.2.5. We note that while our penalty projections are both constant in time and Hermitian, Proposition 3.2.5 is able to cover time-dependent and non-Hermitian projections as well.

**Lemma 3.2.10** (Error in the infeasible space under stable dynamics with a time-dependent generator)**.** *Here, we have the same assumptions as in Lemma 3.2.7, however with a time-dependent generator of the dynamics in the sense that* $A(t) = A_0(t) - \mathsf{i}\lambda\mathsf{P}_c$*. Furthermore, the solution to Problem 2 in case of a time-dependent generator is given by the time-ordered operator exponential using the time-ordering operator* $\mathcal{T}$*,*

$$v(t) = \mathcal{T}\mathrm{e}^{\int_0^t A(\tau)\mathrm{d}\tau} v(0). \tag{3.2.43}$$

*This leads to the following error bound in the infeasible space,*

$$\|v(t)\|_{\ell_2, S_c}^2 \leq \varepsilon \tag{3.2.44}$$

*for* $\lambda \geq \frac{t v_{\max}^2}{\varepsilon} \max_{0\leq t'\leq t} \|[\mathsf{P}_c, A_0(t')]_\sim\|$*, where* $[P, Q]_\sim \equiv PQ - Q^\dagger P$*.*

*Proof.* For the time-dependent case, we make use of Kubo's formula. Kubo's formula was first introduced for Hamiltonian dynamics as a linear response result given an expectation value of an operator $P$,

$$\delta\langle P(\tau)\rangle = -\int_{\tau_0}^{\tau} \mathrm{d}\tau' \, \langle [P(\tau), V(\tau')]\rangle_0 \tag{3.2.45}$$

for an a constant term $H_0 = H(0)$ and a time-dependent perturbation $V(t)$ so that the time-dependent generator is

$$\mathrm{i}H(t) = \mathrm{i}(H_0 + V(t)). \tag{3.2.46}$$

Expectations are defined with respect to the current state (referring to the time of the operator), $\langle\cdot\rangle_t \sim \langle v(t), (\cdot)v(t)\rangle$ or the unperturbed expectation $\langle\cdot\rangle_0 \sim \langle (\mathrm{e}^{-\mathsf{i}\mathsf{P}_c t} v(0)), (\cdot)(\mathrm{e}^{-\mathsf{i}\mathsf{P}_c t} v(0))\rangle$ and are

normalized with respect to the unperturbed evolution $v^{(0)}(t) = \mathrm{e}^{-\mathrm{i}\mathsf{P}_c t} v(0)$), so that

$$\langle P(t)\rangle \approx \langle P(t)\rangle_0 + \delta\langle P(t)\rangle. \tag{3.2.47}$$

Using a correction as in Eq. (3.2.45), this is an identity due to the fundamental theorem of calculus and not a first-order approximation. The dynamics we are interested in stem from

$$A(\tau) = -\mathrm{i}\lambda\mathsf{P}_c + A_0(\tau); \tag{3.2.48}$$

note that $\mathsf{P}_c$ takes the role of $H_0$ above in Eq. (3.2.46). We have a constant constraint projection and a time-dependent $A_0(t)$. There exist generalizations of Kubo's result for non-Hermitian systems [PCCZ20] and including time-dependency [SDM22]. Then, we need to express Eq. (3.2.45) more generally using a linear response function $\chi_{PQ}(t, t')$ for an operator $P$ whose expectation we are interested in and a perturbation term $Q$,

$$\delta\langle P(\tau)\rangle = -\mathrm{i}\int_{\tau_0}^{\tau} \mathrm{d}\tau'\, \chi_{PQ}(\tau, \tau'), \tag{3.2.49}$$

and neither the original generator nor the perturbation need to be Hermitian. Note that compared to [SDM22], we do not separate the perturbation into constant operator and time-dependent forcing that can be pulled out of the response function. In our case, the constraint projection is Hermitian however, which will lead to a simplification of the general response function form [SDM22, Eq. (4)] that resembles Eq. (3.2.45) up to a modified commutator $[P, Q]_\sim := PQ - Q^\dagger P$, and does not require constant re-normalization. Note that in [SDM22], the generator corresponding to Eq. (3.2.48) is expressed as $-i\lambda\mathsf{P}_c + \mathrm{i}A_0(\tau)$, leading to the commutator expression. Hence, in our case and corresponding to what we will be seeing in Proposition 3.2.5, the commutator needs to be replaced with an anticommutator $\{P, Q\}_\sim := PQ + Q^\dagger P$. This results in $\chi_{PQ}(\tau, \tau') = -\mathrm{i}\ \mathbf{1}_{[\tau\geq\tau']}\langle\{P(\tau), Q(\tau')\}_\sim\rangle_0$, or when the dynamics are defined without the imaginary unit in front of $A_0(t)$, then $\chi_{PQ}(\tau, \tau') = \mathbf{1}_{[\tau\geq\tau']}\langle\{P(\tau), Q(\tau')\}_\sim\rangle_0$ so that

$$\delta\langle P(\tau)\rangle = \int_{\tau_0}^{\tau} \mathrm{d}\tau'\ \langle\{P(\tau), Q(\tau')]\}_\sim\rangle_0 \tag{3.2.50}$$

Then, we rephrase the setup as strong perturbation with small $\zeta = \frac{1}{\lambda}$,

$$A(t) = -\mathrm{i}\mathsf{P}_c + \zeta A_0(t), \tag{3.2.51}$$

that we can treat with the Kubo formula. We look for

$$|\langle P(t)\rangle_t - \langle P(t)\rangle_0| = \left|\int_0^t \mathrm{d}\tau\ \langle[P(t), \zeta A_0(\tau)]_\sim\rangle_0\right|. \tag{3.2.52}$$

Now, recall that expectations are normalized with respect to unperturbed *unitary* evolution $\mathrm{e}^{-\mathrm{i}\mathsf{P}_c\tau} v(0)$, so that

$$\langle P(\tau)\rangle_\tau = \frac{\left\langle v^{(0)}(\tau), P(\tau)\, v^{(0)}(\tau)\right\rangle}{\left\langle v^{(0)}(\tau), v^{(0)}(\tau)\right\rangle} = \frac{\left\langle e^{-\mathrm{i}\mathsf{P}_c\tau} v(0), P(\tau) e^{-\mathrm{i}\mathsf{P}_c\tau} v(0)\right\rangle}{\langle e^{-\mathrm{i}\mathsf{P}_c\tau} v(0), e^{-\mathrm{i}\mathsf{P}_c\tau} v(0)\rangle}. \tag{3.2.53}$$

This is less straightforward if the original dynamics are not unitary, as e.g. would be the case if $\mathsf{P}_c$

is *not* orthogonal. We can use this to continue with Eq. (3.2.52),

$$|\langle P(t)\rangle_t - \langle P(t)\rangle_0| = \text{Eq. (3.2.52)} \leq \zeta \int_0^t \mathrm{d}\tau \left| \frac{\left\langle \mathrm{e}^{-\mathrm{i}\mathsf{P}_c t} v(0), \{P(t), A_0(\tau)\}_\sim \mathrm{e}^{-\mathrm{i}\mathsf{P}_c t} v(0)\right\rangle}{\langle e^{-\mathrm{i}\mathsf{P}_c t} v(0), e^{-\mathrm{i}\mathsf{P}_c t} v(0)\rangle} \right|$$

$$\text{by Eq. (3.2.53)} \quad \leq \zeta \int_0^t \mathrm{d}\tau \, \|\{P(t), A_0(\tau)\}_\sim\| \,. \tag{3.2.54}$$

which satisfies the inequality,

$$|\langle P(t)\rangle_t - \langle P(t)\rangle_0| \leq \zeta t \max_{0\leq t'\leq t} \|\{P(t), A_0(t')\}_\sim\|. \tag{3.2.55}$$

Next, we can use that the observable of choice is $P(t) = \mathsf{P}_c$, which is constant over time. Furthermore, the initial conditions are adapted to the constraint and $\langle \mathsf{P}_c\rangle_0 = 0$ consequently. The expectation $\langle \mathsf{P}_c\rangle$ is normalized with respect unperturbed dynamics, thus

$$|\langle \mathsf{P}_c(t)\rangle - \langle \mathsf{P}_c\rangle_0| = \left| \frac{\langle v(t), \mathsf{P}_c v(t)\rangle}{\langle v^{(0)}(t), v^{(0)}(t)\rangle} \right| \leq (\cdot) \Leftrightarrow |\langle v(t), \mathsf{P}_c v(t)\rangle| \leq (\cdot) v_{\max}^2.$$

Remembering that $\zeta = \lambda^{-1}$, we can conclude that the final error is given by

$$|\langle v(t), \mathsf{P}_c v(t)\rangle| \leq \frac{1}{\lambda} t v_{\max}^2 \max_{0\leq t'\leq t} \|[\mathsf{P}_c, A_0(t')]_\sim\|. \tag{3.2.56}$$

□

We point out that the framework of Kubo's formula would also allow us to look at time-dependent strengths of the projection, which may allow to make the overall choice more optimal, and also time-dependent forms of the constraint projection $\mathsf{P}_c$. This is left up to future research.

**Lemma 3.2.11** (Error in the infeasible space under stable dynamics with a time-dependent generator and inhomogeneity with a generalized non-Hermitian, inhomogeneous Kubo formula formula)**.** *Under the same assumptions as in Lemma 3.2.10, however with a time-dependent generator of the dynamics $A(t) = A_0(t) - \mathrm{i}\lambda \mathsf{P}_c$ and a time-dependent inhomogeneous term $b(t)$ so that $\frac{\mathrm{d}}{\mathrm{d}t} v(t) = A(t)v(t) + b(t)$. We assume that for finite time $t \geq 0$, $-\infty \prec \mathrm{Re}(A_0(t)) \preceq 0$ as well as $\exists B > 0$, $\max_{0\leq t'\leq t} \|b(t')\|_{\ell_2}$. The general solution to Problem 2 in this setup follows,*

$$v(t) = \mathcal{T} \mathrm{e}^{\int_0^t \mathrm{d}s\, A(s)} v(0) + \int_0^t \mathrm{d}s\, \mathcal{T} \mathrm{e}^{\int_s^t \mathrm{d}s'\, A(s')} b(s). \tag{3.2.57}$$

*This leads to an error in the infeasible space of at time $t \geq 0$*

$$\|v(t)\|_{\ell_2, S_c}^2 \leq \varepsilon \tag{3.2.58}$$

*for $\lambda \geq \frac{1}{\varepsilon}\left(v_{\max}^2 + v_{\max} B_{L^1} + B_{L^1}^2\right) \frac{t^2}{2} \max_{0\leq t'\leq t} \|\{\mathsf{P}_c, A_0(t')\}_\sim\|$.*

*Proof.* We can directly apply Proposition 3.2.5 in this setting. Recall that

$$\langle P(t)\rangle - \langle P\rangle_0 \eqsim \zeta \int_0^t \mathrm{d}t' \, \frac{\mathrm{tr}\left[\left\{P, \overline{V}(t,t')\sigma(t',t)\right\}_\sim\right]}{\mathrm{tr}\left[\sigma(t,t)\right]} \tag{3.2.59}$$

with $\{X,Y\}_\sim = XY + Y^\dagger X$, $\sigma(t',t) = T_{t'}(v(0)\delta(t') + b(t'))\big(T_t(v(0)\delta(t)+b(t))\big)^\dagger$ where $T_t(u) = \int_0^t \mathrm{d}s\, \mathcal{T}\exp\Big(\int_s^t \mathrm{d}t'\, H(t')\Big)u(s)$ and $\overline{V}(t,t') = \int_{t'}^t \mathrm{d}\tau\, V(\tau)$. The expectation $\langle\cdot\rangle_0$ means expectation with respect to the solution of the unperturbed dynamics.

The quantity we are looking for is the unnormalized squared error, $|\langle v(t), \mathsf{P}_c v(t)\rangle|$. Therefore, we bound

$$|\langle v(t), \mathsf{P}_c v(t)\rangle| = \mathrm{tr}[\sigma(t,t)]\cdot|\langle P(t)\rangle - \langle P\rangle_0|\,. \tag{3.2.60}$$

In our setting, we have that $H(t) = -\mathrm{i}\mathsf{P}_c$ and $\zeta V(t) = \frac{1}{\lambda}A_0(t)$. Then, $T_t(u) = \int_0^t \mathrm{d}s\, \mathrm{e}^{-\mathrm{i}\mathsf{P}_c(t-s)}u(s) = \mathrm{e}^{-\mathrm{i}\mathsf{P}_c t}\left(\int_0^t \mathrm{d}s\, \mathrm{e}^{\mathrm{i}\mathsf{P}_c s}u(s)\right)$, which means that the unperturbed evolution is unitary and has a time-independent generator which simplifies the analysis. Furthermore, we also use $\mathsf{P}_c$ as the observable to measure the $(\ell_2, S_c)$-error. As now, the unperturbed dynamics are unitary, $T_t^\dagger T_t = \mathbb{I}$, we get

$$\begin{aligned}\sigma(t',t) =& \mathrm{e}^{-\mathrm{i}\mathsf{P}_c t'}\int_0^{t'}\mathrm{d}\tau'\int_0^t \mathrm{d}\tau\, \mathrm{e}^{\mathrm{i}\mathsf{P}_c\tau'}(v(0)+b(\tau'))(v^\dagger(0)+b^\dagger(\tau))\mathrm{e}^{-\mathrm{i}\mathsf{P}_c\tau}\mathrm{e}^{\mathrm{i}\mathsf{P}_c t}\\ =:& \mathrm{e}^{-\mathrm{i}\mathsf{P}_c t'}\tilde{v}(t',t)\mathrm{e}^{\mathrm{i}\mathsf{P}_c t},\end{aligned} \tag{3.2.61}$$

and therefore we have that

$$\overline{V}(t,t')\sigma(t',t) = \int_{t'}^t \mathrm{d}s\, A_0(s)\mathrm{e}^{-\mathrm{i}\mathsf{P}_c t'}\tilde{v}(t',t)\mathrm{e}^{\mathrm{i}\mathsf{P}_c t}. \tag{3.2.62}$$

Then, the anti-commutator $\big\{\mathsf{P}_c, \overline{V}(t,t')\nu(t',t)\big\}_\sim$ becomes

$$\int_{t'}^t \mathrm{d}s\, \Big[\mathsf{P}_c A_0(s)\mathrm{e}^{-\mathrm{i}\mathsf{P}_c t'}\tilde{v}(t',t)\mathrm{e}^{\mathrm{i}\mathsf{P}_c t} + \mathrm{e}^{-\mathrm{i}\mathsf{P}_c t}\tilde{v}^\dagger(t',t)\mathrm{e}^{\mathrm{i}\mathsf{P}_c t'}A_0^\dagger(s)\mathsf{P}_c\Big]\,. \tag{3.2.63}$$

Computing the trace of this expression, we get for the left term,

$$\int_{t'}^t \mathrm{d}s\,\mathrm{tr}\left[\mathsf{P}_c A_0(s)\mathrm{e}^{-\mathrm{i}\mathsf{P}_c t'}\tilde{v}(t',t)\mathrm{e}^{\mathrm{i}\mathsf{P}_c t}\right] = \int_{t'}^t \mathrm{d}s\int_0^t \mathrm{d}\tau\int_0^{t'}\mathrm{d}\tau' w^\dagger(\tau)\mathrm{e}^{\mathrm{i}\mathsf{P}_c(t-\tau)}\mathsf{P}_c A_0\mathrm{e}^{\mathrm{i}\mathsf{P}_c(\tau'-t')}w(\tau') \tag{3.2.64}$$

end for the right term

$$\int_{t'}^t \mathrm{d}s\,\mathrm{tr}\left[\mathrm{e}^{-\mathrm{i}\mathsf{P}_c t}\tilde{v}^\dagger(t',t)\mathrm{e}^{\mathrm{i}\mathsf{P}_c t'}A_0^\dagger(s)\mathsf{P}_c\right] = \int_{t'}^t \mathrm{d}s\int_0^t \mathrm{d}\tau\int_0^{t'}\mathrm{d}\tau' w^\dagger(\tau')\mathrm{e}^{\mathrm{i}\mathsf{P}_c(\tau'-t')}A_0^\dagger(s)\mathsf{P}_c\mathrm{e}^{\mathrm{i}\mathsf{P}_c(t-\tau)}w(\tau) \tag{3.2.65}$$

When we add the two terms together, we can re-label $(\tau,t)\leftrightarrow(\tau',t')$ to add them directly and use that $\mathsf{P}_c$ commutes with its time evolution,

$$\int_{t'}^t \mathrm{d}s\int_0^t \mathrm{d}\tau\int_0^{t'}\mathrm{d}\tau' w^\dagger(\tau)\left(\mathsf{P}_c\mathrm{e}^{\mathrm{i}\mathsf{P}_c(t-\tau)}A_0(s)\mathrm{e}^{-\mathrm{i}\mathsf{P}_c(t'-\tau')} + \mathrm{e}^{-\mathrm{i}\mathsf{P}_c(t-\tau)}A_0^\dagger(s)\mathrm{e}^{\mathrm{i}\mathsf{P}_c(t'-\tau')}\mathsf{P}_c\right)w(\tau'). \tag{3.2.66}$$

Next, we additionally add the integral over $t'$,

$$\int_0^t \mathrm{d}t' \int_{t'}^t \mathrm{d}s \int_0^t \mathrm{d}\tau \int_0^{t'} \mathrm{d}\tau' w^\dagger(\tau) \left( \mathsf{P}_c \mathrm{e}^{\mathrm{i}\mathsf{P}_c(t-\tau)} A_0(s) \mathrm{e}^{-\mathrm{i}\mathsf{P}_c(t'-\tau')} + \mathrm{e}^{-\mathrm{i}\mathsf{P}_c(t-\tau)} A_0^\dagger(s) \mathrm{e}^{\mathrm{i}\mathsf{P}_c(t'-\tau')} \mathsf{P}_c \right) w(\tau'). \tag{3.2.67}$$

We continue as follows: We apply the Cauchy-Schwartz-inequality to Eq. (3.2.67) after taking the absolute value. Then, for each summand, we commute the $\tau, \tau'$-$\mathsf{P}_c$-evolutions 'outwards' to the $w^\dagger(\tau), w(\tau')$. The order here does not matter insofar as the $\ell_2$-norm that arises from the inner product is unitarily invariant. Indeed, if we have two terms $\mathrm{e}^{\mathrm{i}\mathsf{P}_c\tau'} w(\tau')$ and $\mathrm{e}^{-\mathrm{i}\mathsf{P}_c\tau'} w(\tau')$ (there are always $\pm$-opposite pairs) that are associated in the Cauchy-Schwartz inequality and are not directly compliant, the corresponding vectors are perfectly aligned except for a complex phase of $\mathrm{e}^{\pm\mathrm{i}\tau'}$ on the $\mathsf{P}_c$-subspace. Hence this does not pose a complication in application of the Cauchy-Schwartz inequality as the obtained upper bound holds. Additionally, we use a similar argument regarding mis-aligned complex phases to re-express the matrix norm through the modified anti-commutator $\{X, Y\}_\sim = XY + Y^\dagger X$. Hence,

$$\begin{aligned}
\text{Eq. (3.2.67)} &\leq \int_0^t \mathrm{d}t' \left\| \int_0^t \mathrm{d}\tau\, w(\tau) \right\|_{\ell_2} \left\| \int_0^{t'} \mathrm{d}\tau'\, w(\tau') \right\|_{\ell_2} \left\| \int_{t'}^t \mathrm{d}s \, \left( \mathsf{P}_c \mathrm{e}^{\mathrm{i}\mathsf{P}_c t} A_0(s) \mathrm{e}^{-\mathrm{i}\mathsf{P}_c t'} + \mathrm{e}^{-\mathrm{i}\mathsf{P}_c t} A_0^\dagger(s) \mathrm{e}^{\mathrm{i}\mathsf{P}_c t'} \mathsf{P}_c \right) \right\| \\
&\leq \int_0^t \mathrm{d}t' \left\| \int_0^t \mathrm{d}\tau\, w(\tau) \right\|_{\ell_2} \left\| \int_0^{t'} \mathrm{d}\tau'\, w(\tau') \right\|_{\ell_2} \left\| \left\{ \mathsf{P}_c, \int_{t'}^t \mathrm{d}s\, A_0(s) \right\}_\sim \right\| \\
&\overset{t' \leq t}{\leq} \left\| \int_0^t \mathrm{d}\tau\, w(\tau) \right\|_{\ell_2}^2 \int_0^t \mathrm{d}s \left\| \left\{ \mathsf{P}_c, \int_0^s \mathrm{d}t'\, A_0(s) \right\}_\sim \right\|
\end{aligned} \tag{3.2.68}$$

Recall that $w(t) = v(0)\delta(t) + b(t)$. Thus, we want to estimate the norm of

$$\int_0^t \mathrm{d}s\, w(s) = v(0) + \int_0^t \mathrm{d}s\, b(s). \tag{3.2.69}$$

We obtain

$$\begin{aligned}
\left\| \int_0^t \mathrm{d}s\, w(s) \right\|_{\ell_2}^2 &= \langle v(0), v(0) \rangle + 2\,\mathrm{Re} \left\langle v(0), \int_0^t \mathrm{d}s\, b(s) \right\rangle + \left\| \int_0^t \mathrm{d}s\, b(s) \right\|_{\ell_2}^2 \\
&\leq \|v(0)\|_{\ell_2}^2 + 2\|v(0)\|_{\ell_2} \|b(s)\|_{L^1([0,t])} + \|b(s)\|_{L^1([0,t])}^2,
\end{aligned} \tag{3.2.70}$$

where $\|f\|_{L^1(W)} = \int_W \mathrm{d}s\, |f(s)|$ for $W \subseteq \mathbb{R}$, which implies that $\| \int_W \mathrm{d}s\, f(s) \|_{\ell_2}^2 \leq \|f(s)\|_{L^1(W)}^2$. To make notation shorter, we recall that $\|v(0)\|_{\ell_2} =: v_{\max}$ (by $\mathrm{Re}(A) \preceq 0$) and define $\|b(s)\|_{L^1([0,t])} =: B_{L^1}$. Therefore,

$$\begin{aligned}
\text{Eq. (3.2.70)} &\leq \left( v_{\max}^2 + v_{\max} B_{L^1} + B_{L^1}^2 \right) \int_0^t \mathrm{d}s\, s \, \|\{\mathsf{P}_c, A_0(s)\}_\sim\| \\
&\leq \left( v_{\max}^2 + v_{\max} B_{L^1} + B_{L^1}^2 \right) \frac{t^2}{2} \max_{0 \leq t' \leq t} \|\{\mathsf{P}_c, A_0(t')\}_\sim\|.
\end{aligned} \tag{3.2.71}$$

To conclude, this means that

$$|\langle v(t), \mathsf{P}_c v(t)\rangle| \leq \frac{1}{\lambda}\left(v_{\max}^2 + v_{\max} B_{L^1} + B_{L^1}^2\right) \frac{t^2}{2} \max_{0\leq t'\leq t} \|\{\mathsf{P}_c, A_0(t')\}_\sim\|. \tag{3.2.72}$$

□

**Application to Discretized Partial Differential Equations**

The type of constraints we are mainly interested in in this work are constraints coming from boundary conditions in the discretization of linear, evolutionary partial differential equations. Namely, for $\mathcal{L}$ a linear differential operator, then for $t \in \mathbb{R}_+$ and $r \in \Omega \subseteq \mathbb{R}^d$, and $v(t,r)$ comes from a function space so that the PDE with respect to $\mathcal{L}$ is well-posed,

$$\begin{aligned} \partial_t v(t,r) - \mathcal{L}v(t,r) &= f(t,r), && r \in \Omega \\ v(t,r) &= g(r,t), && r \in \Gamma_D \qquad (3.2.73) \\ \langle \partial_n,\, v(t,r)\rangle &= h(r,t), && r \in \Gamma_N \end{aligned}$$

$$v(0,r) = v_0(r). \tag{3.2.74}$$

The solution lives on a domain $\Omega$ and constrained to a specific value on the Dirichlet boundary $\Gamma_D$ or a normal derivative on the Neumann boundary $\Gamma_N$. Robin boundary conditions, which specify a linear combination of derivative and value at the boundary can be obtained as a corollary, see Section 3.2.2.

In the context of Problem 3, we have that on a discretized version of $v(t,r)$, $\mathsf{P}_c^D$ projects onto the domain $\Gamma_D$, $\mathsf{P}_c^N$ applies a backwards difference stencil along the surface normal on $\Gamma_N$ (for details see Section 3.2.2) and the differential operator $\mathcal{L}$ becomes $A_0(t)$. There are no further assumptions beyond linearity on $\mathcal{L}$ so far, though most of our error bounds in the previous section require that the for the discretized operator, $\mathrm{Re}(A_0) \preceq 0$.

The methodology we developed so far is able to enforce $g_c(t) = 0$ and $h_(t) = 0$, assuming access to boundary projections $\mathsf{P}_c^D, \mathsf{P}_c^N$. Thus, we next consider how to deal with non-zero values as this is an essential aspect of the utility of our method.

**Non-zero Dirichlet boundaries** We start by considering the problem in Eq. (3.2.73), assuming the condition on $\Gamma_D$. Let us further assume for simplicity that $\overline{\Omega} = [0, L]^d$. We will construct a function $u(r,t)$ that satisfies the constraint on $\Gamma_D$ in (3.2.73). For the one-dimensional case, $d = 1$, this can be found quite easily:

$$u(r,t) = \frac{1}{2}\frac{r}{L} g(L,t) + \frac{1}{2}\left(1 - \frac{r}{L}\right) g(-L,t). \tag{3.2.75}$$

Extending this to arbitrary dimension $d \geq 1$ follows naturally

$$u(r,t) = \frac{1}{2}\sum_{i=0}^{d-1} \frac{r_i}{L} g(r,t)\big|_{r_i = L} + \left(1 - \frac{r_i}{L}\right) g(r,t)\big|_{r_i = -L}. \tag{3.2.76}$$

Now, let us consider a new function

$$v'(r,t) = v(r,t) - u(r,t), \tag{3.2.77}$$

and compute the following,

$$\begin{aligned}\partial_t v'(r,t) - \mathcal{L}v'(r,t) &= \partial_t v(r,t) - \mathcal{L}v(r,t) - \partial_t u(r,t) + \mathcal{L}u(r,t)\\ &= f(r,t) + \mathcal{L}u(r,t) - \partial_t u(r,t) &(3.2.78)\\ &=: \widetilde{f}(r,t) &(3.2.79)\end{aligned}$$

where

$$\partial_t u(r,t) = \frac{1}{2}\sum_{i=0}^{d-1}\frac{r_i}{L}\partial_t g(r,t)\big|_{r_i=L} + \left(1-\frac{r_i}{L}\right)\partial_t g(r,t)\big|_{r_i=-L}. \tag{3.2.80}$$

The initial condition becomes

$$v'(r,0) = v(r,0) - u(r,0) = v_0(r) - u(r,0) = v_0(r) \tag{3.2.81}$$

and the boundary condition

$$v'(r,t)\big|_{r\in\Gamma_D} = g(r,t) - \underbrace{v'(r,t)\big|_{x\in\Gamma_D}}_{=g(r,t)} = 0. \tag{3.2.82}$$

Therefore, the differential equation for $v'$ is given by

$$\begin{aligned}\partial_t v'(r,t) - \mathcal{L}(t)v'(r,t) = \widetilde{f}(r,t), \quad r\in\Omega, t>0\\ v'(r,t) = 0 \quad r\in\Gamma_D, t\geq 0,\\ v'(r,0) = v_0'(r) \quad r\in\overline{\Omega}, t=0.\end{aligned} \tag{3.2.83}$$

That means that the PDE in $u(r,t)$ now has Dirichlet boundary conditions of value zero. The solution $v(r,t)$ to the original problem can then be obtained by

$$v(r,t) = v'(r,t) + u(r,t). \tag{3.2.84}$$

Therefore we easily obtain the following corollary to Lemma 3.2.8 and Lemma 3.2.11 for the case of a suitable discretization with error at most $O(\varepsilon)$, where then the differential operator $\mathcal{L}$ has discrete representation $A_0$.

**Corollary 3.2.12** (Error in the infeasible space under stable dynamics with non-zero Dirichlet Boundary Conditions.)**.** *We have the same assumptions as in Lemma 3.2.7 and additionally assume that the domain of interest is a regular box of length $L>0$ in $d$ dimensions, $\overline{\Omega} = [0,L]^d$. We are looking for a solution to non-zero value constraints as in*

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}v(t) &= A_0 v(t) + b(t),\\ \mathsf{P}_c v(t) &= g_c(t) \in S_c, t\geq 0.\end{aligned} \tag{3.2.85}$$

*Then, let* $v'(t) = v(t) - g_c(t)$*, which implies that for* $v(t)|_{S_c} = g_c(t)$*, the modified variable satisfies* $v'(t)|_{S_c} = 0$*. Then, with* $b'(t) = -A_0 g_c(t) - \frac{\mathrm{d}}{\mathrm{d}t} g_c(t) + b(t)$ *we get an equivalent homogeneous system*

$$\begin{aligned} &\frac{\mathrm{d}}{\mathrm{d}t} v'(t) = A_0(t) v'(t) + b'(t), \\ &\mathsf{P}_c v'(t) = 0, \quad t \geq 0. \end{aligned} \tag{3.2.86}$$

*This means, the penalized form of Eq.* (3.2.86)*,* $\frac{\mathrm{d}}{\mathrm{d}t} v'(t) = (A_0(t) - \mathrm{i}\lambda \mathsf{P}_c) v'(t) + b'(t)$*, will satisfy the desired boundary condition from Eq.* (3.2.85) *as per Lemmas 3.2.8 and 3.2.11 using an inhomogeneous term* $b'(t) = A_0(t) g_c(t) - \frac{\mathrm{d}}{\mathrm{d}t} g_c(t) + b(t)$*. The necessary* $\lambda$ *to achieve an error of at most* $\varepsilon$ *is*

$$\lambda = \frac{2\|A_0\|}{\varepsilon} \left( v_{\max}^2 + 2 t v_{\max} B' + t^2 B'^2 \right) \tag{3.2.87}$$

*$A_0$ time-indep.*

$$\lambda = \frac{\left( v_{\max}^2 + v_{\max} B'_{L^1} + (B'_{L^1})^2 \right)}{\varepsilon} \frac{t^2}{2} \max_{0 \leq t' \leq t} \|\{\mathsf{P}_c, A_0(t')\}_\sim\| \tag{3.2.88}$$

*$A_0(t)$ time-dep.*

*where* $B' \geq \max_{0 \leq t' \leq t} \|A_0 g_c(t') - \dot{g}_c(t') + b(t')\|_{\ell_2}$*,* $B'_{L^1} = \int_0^t \mathrm{d}t' \, |A_0(t') g_c(t') - \dot{g}_c(t') + b(t')|$*, and the modified anti-commutator as before* $\{X, Y\}_\sim = XY + Y^\dagger X$*.*

Assuming a suitable discretization technique [Ame14; EBY12] using a finite-dimensional basis of $n$ basis functions per dimension, we obtain a semi-continuous solution vector $v(t) \in \mathbb{R}^{n^d}$. In alignment with Definitions 3.2.1 and 3.2.2 we call the discrete domain $S$, the value boundary $S_c^D$ and the derivative boundary $S_c^N$. Then, we retrieve the formulation readily posed in Problem 3.

**Neumann boundary conditions** Within this work, we will not worry about the approximation error with respect to the continuum limit and assume that we are provided with an ODE that follows from a sensible discretization, achieving a target discretization error $\varepsilon > 0$. Thus, we will continue only with bounding the error in representing a difference formula, denoted by $\mathsf{D}$, using our projection method. The difference formula is implicitly defined by the sets $\zeta_{\boldsymbol{j}}$ as given in Definition 3.2.1; a weighted combination of differences with neighbours which is then implemented through a SWAP network.

Errors from enforcing the contraint via the penalty projection follow similar as in the case for Dirichlet conditions. This is because fundamentally what we do is suppress the overlap on the subspace given by the penalty projection. Then, if the penalty projection approximates a derivative reasonably well, as in Eq. (3.2.103), the approximation of a derivative follows. We give an example using zero Neumann boundary conditions as the simplest case later in Section 3.2.2.

Moreover, the same approach of 'shifting the data' can be applied to the case of Neumann boundary conditions, i.e., constraints on the derivative such as $\langle \partial_{\boldsymbol{n}}, v(t, r) \rangle = h(r, t)$ for $r \in \Gamma_N$ in Eq. (3.2.73). Here, $\boldsymbol{n}$ is the outward facing normal vector on the Neumann boundary of the domain, $\Gamma_N$. This problem has a solution only if the following consistency condition is satisfied,

$$\int_{\Gamma_N} \langle \partial_r v(r, t), \boldsymbol{n}(r) \rangle \mathrm{d}S = \int_{\Gamma_N} h(r, t) \mathrm{d}S \implies \int_\Omega \Delta v(r, t) \mathrm{d}V = \int_{\Gamma_N} h(r, t) \mathrm{d}S, \tag{3.2.89}$$

where the implication follows from the divergence theorem and $\mathrm{d}V$ and $\mathrm{d}S$ are volume and surface elements respectively. We assume this condition is satisfied so that the Neumann-value problem is

well-posed.

Similar to the Dirichlet case, we seek a function $u(r,t)$ that complies with the constraint,

$$\langle \partial_r u(r,t),\, \boldsymbol{n}(r)\rangle = h(r,t) \quad \text{for any } r \in \Gamma_N. \tag{3.2.90}$$

This function $u(r,t)$ also should satisfy the consistency condition Eq. (3.2.89). Now, let

$$v'(r,t) = v(r,t) - u(r,t). \tag{3.2.91}$$

The respective PDE in $v'$ then is

$$\begin{aligned}\partial_t v'(r,t) - \mathcal{L}v'(r,t) &= \partial_t v(r,t) - \mathcal{L}v(r,t) - \partial_t u(r,t) + \mathcal{L}u(r,t)\\ &= f(r,t) + \mathcal{L}u(r,t) - \partial_t u(r,t) =: \widetilde{f}(r,t),\end{aligned} \tag{3.2.92}$$

with initial condition $v'(r,0) = v_0(r) - u(r,0) =: v_0(r)$, and boundary condition $\langle \partial_r u(r,t),\, \boldsymbol{n}(r)\rangle = 0$. Therefore, in total, $v'$ will satisfy the homogeneous Neumann problem

$$\begin{aligned}\partial_t v'(r,t) - \mathcal{L}v'(r,t) &= \widetilde{f}(r,t), && r \in \Omega,\ t > 0\\ v'(r,0) &= v_0'(r), && r \in \overline{\Omega}, t = 0\\ \langle \partial_r v'(r,t),\, \boldsymbol{n}(r)\rangle &= 0, && r \in \Gamma_N, t \geq 0.\end{aligned} \tag{3.2.93}$$

Analogous to the Dirichlet case, this gives rise to a means to solve (3.2.93) for $v'$ and adding to it the function $u(r,t)$. Then, $v' + u$ satisfies the Neumann problem as in Eq. (3.2.73).

Notice that the consistency condition for $v'$ implies

$$\int_\Omega \Delta v'(r,t) \mathrm{d}V = 0. \tag{3.2.94}$$

An alternative approach to enforcing the derivative constraint that we do not follow within this work thus would be to express a penalty term in the modified ODE Eq. (3.2.11) so that $\mathsf{P}_c^N$ projects onto second derivatives within the unconstrained domain $S$, corresponding to Eq. (3.2.94). The error can be quantified according to Corollary 3.2.12 by using the difference-projection $\mathsf{P}_c^N$ instead of $\mathsf{P}_c^D$.

### Input Model of Constraint Projections for Quantum Implementation

We continue by discussing constructing the relevant projections and their quantum implementations. The quantity we want to encode is the solution vector $v(t)$, which is expressed with $n_l$ points per dimension $l \in [d]$; recall the definitions of $\mathcal{I}_\Omega, \mathcal{I}_{\Gamma_D}, \mathcal{I}_{\Gamma_N}$ from Definition 3.2.1. This vector is represented in terms of an amplitude encoding:

$$|v(t)\rangle = \frac{1}{\|v(t)\|} \sum_{\boldsymbol{j} \in \mathcal{I}_{\Omega\cup\Gamma}} v_{\boldsymbol{j}}(t) \,|\boldsymbol{j}\rangle\,. \tag{3.2.95}$$

Here, $\boldsymbol{j} = (j_1, j_2, \ldots, j_d)$ is a multi-index over $d$ dimensions so that $j_l \in [n_l - 1]_0$ for all $l \in [d]$.

In order to specify boundaries, we assume access to an oracles $\mathrm{O}_{\mathrm{bdry}}$ so that

$$\mathrm{O}_{\mathrm{bdry}} : |\boldsymbol{j}\rangle |0\rangle_{\mathrm{bdry}} \to \begin{cases} |\boldsymbol{j}\rangle |0\rangle_{\mathrm{bdry}}, & \boldsymbol{j} \in \mathcal{I}_{\Omega} : \text{ internal point} \\ |\boldsymbol{j}\rangle |1\rangle_{\mathrm{bdry}}, & \boldsymbol{j} \in \mathcal{I}_{\Gamma_D} : \text{ Dirichlet condition} \\ |\boldsymbol{j}\rangle |2\rangle_{\mathrm{bdry}}, & \boldsymbol{j} \in \mathcal{I}_{\Gamma_N} : \text{ Neumann condition} \\ |\boldsymbol{j}\rangle |3\rangle_{\mathrm{bdry}}, & \boldsymbol{j} \in \mathcal{I}_{\Gamma_R} : \text{ Robin condition} \end{cases} \tag{3.2.96}$$

The algorithm we present in Section 3.2.4 will consider the dynamics we study exclusively in an interaction picture with respect to $\mathsf{P}_c$. Therefore, the type of access we require to a projection is through Hamiltonian simulation of it, which we study alongside. An important point then so that the interaction picture simulation algorithm in [LW18] is efficient is that the projection's Hamiltonian simulation can be fast-forwarded [AA17, Definition 1], so that the complexity of implementation depends at most logarithmically on $\lambda$.

**Projections for Dirichlet conditions and value constraints** Dirichlet boundary conditions in PDEs are point-wise value constraints. On a grid, this naturally induces a projection matrix by projecting directly onto basis elements. We can define $\mathsf{P}_c^D$ to be

$$\mathsf{P}_c^D = \sum_{\boldsymbol{j} \in \mathcal{I}_{\Gamma_D}} |\boldsymbol{j}\rangle\langle\boldsymbol{j}|, \tag{3.2.97}$$

which is equivalent to conditioning on the boundary flag from the boundary oracle $\mathrm{O}_{\mathrm{bdry}}$ in Eq. (3.2.96) being in 1-state, so $\boldsymbol{j} \in \mathcal{I}_{\Gamma_D}$. Given this oracle $\mathrm{O}_{\mathrm{bdry}}$ it is straightforward to construct a circuit for a value constraint projector,

$$U_{\mathsf{P}_c^D} : |\boldsymbol{j}\rangle |0\rangle_{\mathrm{bdry}} \mapsto \mathrm{O}_{\mathrm{bdry}}^{\dagger} \left( \mathbb{I} \otimes |1\rangle\langle 1|_{\mathrm{bdry}} \right) \mathrm{O}_{\mathrm{bdry}} |\boldsymbol{j}\rangle |0\rangle_{\mathrm{bdry}}. \tag{3.2.98}$$

In terms of LCU, we can see $\mathrm{O}_{\mathrm{bdry}}$ as PREP and $\mathbb{I} \otimes |1\rangle\langle 1|_{\mathrm{bdry}}$ as (a part of) SEL.

**Hamiltonian simulation of the value constraint projection.** From [AA17, Theorem 3], we have that for any *commuting* Hamiltonian $H$ so that $\|H\|_{\infty} = 1$, $H \in \mathbb{C}^{2^n \times 2^n}$ and $\mathrm{supp}(H) = k$ with $k \in O(\log n)$, Hamiltonian simulation of $H$ can be $(T, \alpha)$ fast-forwarded with $T \in 2^{O(n)}$ and $\alpha > 0$. In the case of value constraints, locality is easy to show as $k = 1$ and as we deal with projections, it follows that $\|\mathsf{P}_c^D\|_{\infty} = 1$.

We notice that $[|\boldsymbol{j}\rangle\langle\boldsymbol{j}|, |\boldsymbol{k}\rangle\langle\boldsymbol{k}|] = 0$ for any $\boldsymbol{j}, \boldsymbol{k}$ in $\mathcal{I}_{\overline{\Omega}}$. This means for the evolution that

$$\mathrm{e}^{-\mathrm{i}\lambda t \mathsf{P}_c^D} = \prod_{\boldsymbol{j} \in \mathcal{I}_{\Gamma_D}} \mathrm{e}^{-\mathrm{i}\lambda t |\boldsymbol{j}\rangle\langle\boldsymbol{j}|}. \tag{3.2.99}$$

So for time-independent constraints, this can be implemented simply by phase gates parametrized by the penalty $\lambda$, controlled on $|\boldsymbol{j}\rangle$.

$$\mathrm{HamSim}(\mathsf{P}_c^D, \lambda t) |\boldsymbol{j}\rangle |0\rangle_{\mathrm{bdry}} = \mathrm{O}_{\mathrm{bdry}}^{\dagger} \left( \mathrm{e}^{-\mathrm{i}\lambda t} \otimes |1\rangle\langle 1|_{\mathrm{bdry}} \right) \mathrm{O}_{\mathrm{bdry}} |\boldsymbol{j}\rangle |0\rangle_{\mathrm{bdry}} \tag{3.2.100}$$

Therefore, Eq. (3.2.100) is fast-forwarded by this construction as the implementation cost does not directly depend on the parameter $\lambda t$.

**Projectors for Neumann conditions and derivative constraints** To obtain a finite-dimensional approximation to a Neumann projection, we employ a discrete finite-difference derivative $\mathsf{D}$. As observed in the statement of Problem 2, a crucial ingredient for our method to work is that the projection $\mathsf{P}_c^N$ for the approximate derivative condition is orthogonal to, and thereby commutes with, the projection onto the feasible region and the projection onto the value constraint region. We discuss next how these requirements are satisfied by a simple finite difference stencil evaluated on two points assuming a grid discretization. The motivation is that this allows us to enforce that the value on the boundary has the same value as the next-closest value within the domain in the direction of interest.

First, we look at a one-dimensional example. Take $v = \sum_{l=1}^{L} v_l e_l$ with the canonical basis $\{e_l\}_{l=1}^{L}$. Consider $\partial_n \cdot v \mid_L = 0$ and $v$ representing a line split into $L$ points, then $v_L$ falls into the derivative constraint. Then, a "inward" finite difference stencil would be $\frac{v_L - v_{L-1}}{h}$. For the sake of a projection, we only care about the difference, and may omit the scaling by the grid spacing $h$. Then, a discrete derivative is given by $\mathsf{D}v|_{\Gamma_N} = v_L - v_{L-1}$. Consequently, what we want to enforce is that $v_L = v_{L-1}$. The feasible and infeasible spaces then are

$$S = \mathrm{span}\{v : v_L = v_{L-1}\}, \quad S_c^N = \mathrm{span}\{v : v_L \neq v_{L-1}\}. \tag{3.2.101}$$

We will design an algorithm that allows to attain an approximate $S_c^N$ via

$$\mathrm{span}\{v : |v_L \neq v_{L-1}| < \varepsilon\} \tag{3.2.102}$$

for $\varepsilon > 0$. A swap operation $\mathsf{S}$ so that $\mathsf{S}v = \sum_{l=1}^{L-2} v_l e_l + v_L e_{L-1} + v_{L-1} e_L$ has the property that it leaves $S$ invariant. Then, the projection $\frac{1}{2}(\mathbb{I} - \mathsf{S}) \equiv \mathsf{P}$ has the desired properties that $\ker(\mathsf{P}) = S$ and $\mathrm{image}(\mathsf{P}) = S_c^N$. Note that the canonical basis to $S_c$ are not eigenvectors to $\mathsf{P}$ but its image is closed in $S_c^N$. That means it will allow us to project onto elements that are unacceptable and we can annihilate them. As we want orthogonal projections as per assumptions in Definition 3.2.3, we need to choose Hermitian projections. The 'bad' space projection already satisfies this, for the 'good' projection we can use $\mathsf{P}^{\perp} = \frac{1}{2}(\mathbb{I} + \mathsf{S})$.

We continue by discussing how to create the projector for $|\mathcal{I}_{\Gamma_N}| > 1$. A general solution vector $v$ represents a spatial discretization of the continuous solution over a domain $\Omega \subseteq \mathbb{R}^d$, with overall $n = \prod_{l=1}^{d} n_l$ basis elements. For every point $\boldsymbol{j} \in \mathcal{I}_{\Gamma_N}$ within the computational domain that lies on the derivative constraint region or Neumann boundary $\Gamma_N$, we have a set of neighbours $\zeta_{\boldsymbol{j}}$ that represent the set of basis elements that we use for the respective finite difference formula. This describes the set of mutual swaps that need to be done for every $\boldsymbol{j}$ on $\Gamma_N$. It is reasonable to assume this discretization is $\zeta$-local in the sense that every neighbour set has at most $|\zeta_{\boldsymbol{j}}| = \zeta \in \mathbb{N}$ elements. In fact, for a regular grid discretization, every node has at most $(n-1)^d < n^d - 1$ neighbours of which only a subset will be interesting for the derivative constraints. Therefore we only need to store information specifically what a node's respective neighbours are rather than which others every node neighbours. This leads to the following as a formulation for the sought after projector,

$$\mathsf{P}_c^N = \frac{1}{2}\left(\mathbb{I} - \prod_{\boldsymbol{j} \in \mathcal{I}_{\Gamma_N}, \boldsymbol{k} \in \zeta_{\boldsymbol{j}}} \mathrm{SWAP}(\boldsymbol{j}, \boldsymbol{k})\right) = \frac{1}{2}(\mathbb{I} - \mathsf{S}). \tag{3.2.103}$$

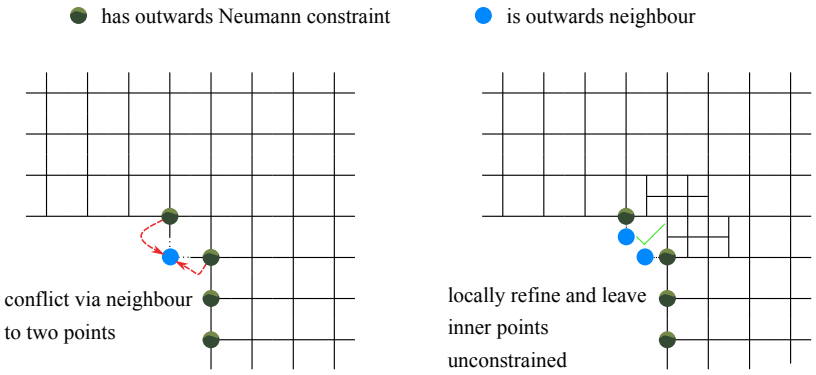
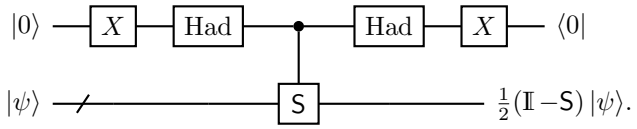



Figure 3.8: Conflicts arising in derivative constraints on a regular grid (left). Mitigation is possible by either only allowing one of the constraints, or a local grid refinement and ignoring the constraint on the finer level (right).

Direct implementation by a linear combination or unitaries yields,




(3.2.104)

Circuit constructions for the SWAP's $\mathsf{S}$ are discussed in the upcoming paragraphs.

Later on, we will discuss fast-forwarded Hamiltonian simulation of the constraint projections. For this to be possible, we will require the following conditions on the problem setup:

- Any point that is neighbour to a constraint point for the sake of a derivative condition (i.e., part of the finite difference formula) cannot be affected by a derivative constraint itself.

$$\forall \boldsymbol{j} \in \mathcal{I}_{\Gamma_N}, \zeta_{\boldsymbol{j}} \cap \mathcal{I}_{\Gamma_N} = \emptyset \tag{3.2.105}$$

- Any point can be neighbour in a finite difference formula for a derivative constraint to at most one point on $\Gamma_N$.

$$\forall \boldsymbol{j}, \boldsymbol{k} \in \mathcal{I}_{\Gamma_N}, \zeta_{\boldsymbol{j}} \cap \zeta_{\boldsymbol{k}} = \emptyset \tag{3.2.106}$$

For discretized PDEs, this scenario can be realized easily. Suppose a grid discretization would not satisfy these conditions (see left-hand side of Fig. 3.8), then a refinement of the discretization is sufficient (right-hand side of Fig. 3.8) and comes at little cost in extra qubits. If the problem-at-hand is a more general constrained ODE, one can think of the same strategy together with introducing additional dummy variables that are trivially coupled with their neighbours instead of a global 'refinement'.

**Example** (Error on a derivative constraint for stable dynamics)**.** *We consider a system as in Problem 2 with dynamics generated by* $A = A_0 - \mathrm{i}\lambda \mathsf{P}_c^N$*, with* $\mathsf{P}_c^N$ *from Eq.* (3.2.103)*,* $A_0$ *and thus also* $A$ *are normal and the real parts of the eigenvalues of* $A_0$ *are non-positive,* $\lambda > 0$ *and* $\mathsf{P}_c^N$ *the projector on the infeasible space as defined in Definition 3.2.3. Then, the error at time* $t > 0$ *along the*

*boundary with respect to the finite difference formula embodied by* $\mathsf{P}_c^N$ *is bounded as follows,*

$$\|\mathsf{D}v(t)\|^2_{\ell_2,S_c^N} \leq \epsilon \tag{3.2.107}$$

*for* $\lambda = \frac{2v_{\max}^2\|A_0\|}{\varepsilon}$.

*Proof.* Using Eq. (3.2.103), we have that ${\mathsf{P}_c^N}^2 = \mathsf{P}_c^N = (\mathsf{P}_c^N)^\dagger$. We now determine the approximation error of the projective boundary conditions with respect to the finite-dimensional approximation of a derivative constraint through $\mathsf{D}v(t)$, $\|\mathsf{D}v(t) - h\|$, while we consider $h = 0$ in this Lemma. But this then allows us to say that

$$\|\mathsf{D}v(t)\|^2_{\ell_2,S_c^N} = \left|\left\langle \mathsf{P}_c^N v(t), \mathsf{P}_c^N v(t)\right\rangle\right| = \left|\left\langle v(t), \mathsf{P}_c^N v(t)\right\rangle\right|, \tag{3.2.108}$$

Now, we can bound $\left|\left\langle v(t), \mathsf{P}_c^N v(t)\right\rangle\right|$ similar to the Dirichlet case. We recall that $v(t) = \exp(At)v(0)$, $A = A_0 - \mathrm{i}\lambda\mathsf{P}_c^N$, $\mathrm{Re}(A_0) \preceq 0$. First, observe that

$$|\langle v(0), Av(0)\rangle| = E_0 < \infty, \tag{3.2.109}$$

as we assume enough sufficient smoothness in the initial condition to satisfy the boundary condition. The rest of the proof then follows equivalently the proof of Lemma 3.2.7. Thus, to summarize, we can say that

$$\left|\left\langle v(t), \mathsf{P}_c^N v(t)\right\rangle\right| \leq 2v_{\max}^2 \frac{\|A_0\|}{\lambda}. \tag{3.2.110}$$

□

**Remark** (Alternative approach to Value Constraints: Example for time-independent case)**.** *Derivative constraints provide an alternative to the implementation of non-zero value constraints compared to homogenizing the ODE as in Section 3.2.2. We can introduce dummy variables (similar to 'ghost points'), which are not coupled with the other variables through* $A_0(t)$*, and store the desired values in them. Then, enforce a derivative constraint with the respective nodes, so that the effect of the constraint is that the affected boundary points will obtain the same value as stored in the dummy variables. That is,*

$$\tilde{v} = \begin{bmatrix} v \\ v_{\text{dummy}} \end{bmatrix}, \quad \widetilde{A_0} = \begin{bmatrix} A_0 & 0 \\ 0 & 0 \end{bmatrix}, \tag{3.2.111}$$

*where* $v_{\text{dummy}} \in \mathbb{C}^{|\mathcal{I}_{\Gamma_N}|}$*. Then, adding the constraint projection*

$$\mathsf{P}_c^N = \frac{1}{2}\big(\mathbb{I} - \mathrm{SWAP}(v|_{\Gamma_N}, v_{\text{dummy}})\big) \tag{3.2.112}$$

*to the dynamics through* $\frac{\mathrm{d}}{\mathrm{d}t}\tilde{v} = (\widetilde{A_0} - \mathrm{i}\mathsf{P}_c^N)\tilde{v}$ *leads to* $v = \mathsf{P}_{S\oplus S_c^N}\tilde{v}$ *so that a value/Dirichlet constraint is enforced up to accuracy* $\varepsilon$ *for* $\lambda$ *chosen according to the example above.*

*Note that this approach leads to a lower bound on* $\lambda$ *necessary for accuracy* $\varepsilon$ *follows, as* $v_{max}^2 + B^2 > \tilde{v}_{\max}^2 = v_{\max}^2 + \|v_{\text{dummy}}\|^2_{\ell_2}$ *with* $B = \|A_0 g\|_{\ell_2}$ *and* $\|v_{\text{dummy}}\|_{\ell_2} = \|g\|_{\ell_2}$*; see Corollary 3.2.12 vs. Lemma 3.2.7. On the other hand, the implementation cost for the derivative constraint is slightly higher due to the SWAP construction compared to the value constraints as we will observe in the following.*

**Input model for** SWAP **circuit.** This section discusses a circuit construction for the swap circuit $\mathsf{S}$ in Eq. (3.2.103). The first ingredient to that is an oracle $\mathrm{O}_\zeta$ that prepares a superposition encoding the distances $\boldsymbol{k}-\boldsymbol{j}$ over all neighbours $\boldsymbol{k}$ from a neighbour set $\zeta_{\boldsymbol{j}} \subseteq \mathcal{I}_{\Omega\cup\Gamma}$ that contains the neighbours to node $\boldsymbol{j}$. Then, we define the neighbour oracle by its action on a state $v_{\boldsymbol{j}} |\boldsymbol{j}\rangle$ as

$$\begin{aligned}\mathrm{O}_\zeta : v_{j_1\cdots j_d} |j_1\cdots j_d\rangle |0\rangle \\ \to v_{j_1\cdots j_d} |j_1\cdots j_d\rangle \left( \frac{1}{\sqrt{|\mathcal{I}_{\Gamma_N}||\zeta_{\boldsymbol{j}}|}} \sum_{(k_1\cdots k_d)\in\zeta_{\boldsymbol{j}}} |k_1-j_1\rangle |k_2-j_2\rangle\cdots|k_d-j_d\rangle \right). \end{aligned} \tag{3.2.113}$$

The action of $\mathrm{SWAP}(\boldsymbol{j},\boldsymbol{k})$ for a set of neighbour shifts $(k_1-j_1,\dots k_d-j_d)\in\zeta_{\boldsymbol{j}}$ is given as follows:

$$v_{j_1\cdots j_d} |j_1\cdots j_d\rangle \mapsto v_{j_1\cdots j_d} |k_1\rangle |k_2\rangle\cdots|k_d\rangle . \tag{3.2.114}$$

Namely, it effects a shift by distances $k_w-j_w$, $w\in[d]$, where the distances are given by the neighbour oracle $\mathrm{O}_\zeta$. Whenever the constraints are given by Neumann boundary conditions, we can also expect that most of these differences are zero. In the case of a two-point stencils to approximate the derivative, there is only one index $k_w$ that needs to be shifted in $d'$ dimensions, where $d'<d$ is the dimension of the boundary. Then, Eq. (3.2.114) becomes

$$v_{j_1\cdots j_d} |j_1\cdots j_d\rangle \mapsto v_{j_1\cdots j_d} |j_1\rangle |j_2\rangle\cdots|k_w\rangle\cdots|j_d\rangle \tag{3.2.115}$$

Therefore, these swaps can be implemented a sequence of controlled additions of $j_w$ with the distances $k_w-j_w$ and call this subroutine. The arithmetic that needs to be done here is fairly simple as we only need to add bit-strings. Recall that CADD is defined as

$$\begin{aligned}\mathrm{CADD} : |\boldsymbol{j}\rangle |\boldsymbol{k}\rangle \to |\boldsymbol{j}+\boldsymbol{k}\rangle |\boldsymbol{k}\rangle \\ \equiv |j_1\rangle |j_2\rangle\cdots|j_d\rangle |k_1\rangle |k_2\rangle\cdots|k_d\rangle \to |j_1\oplus k_1\rangle |j_2\oplus k_2\rangle\cdots|j_d\oplus k_d\rangle |k_1\rangle |k_2\rangle\cdots|k_d\rangle , \end{aligned} \tag{3.2.116}$$

where $j_1,\dots$ are bitstrings with length $\lceil\log_2(n)\rceil$ and $\oplus$ is addition modulo 2. Note that for the sake of our input model with added distances $\boldsymbol{k}-\boldsymbol{j}$, the addition will never exceed the representable range of grid point indices.

Moreover, for each $\boldsymbol{j}\in\mathcal{I}_{\Gamma_N}$, we have $|\zeta_{\boldsymbol{j}}|$ neighbours with $d$ dimensions. Thus, we need to do at most $O\left(d|\mathcal{I}_{\Gamma_N}|\cdot\max_{\boldsymbol{j}}|\zeta_{\boldsymbol{j}}|\right)$ additions. Assuming we have at most $n$ basis elements per dimension to index and thereby each summand has at most $\lceil\log_2 n\rceil$ bits, then the overall cost to perform the additions will be $O\left(d\lceil\log_2 n\rceil|\mathcal{I}_{\Gamma_N}|\cdot\max_{\boldsymbol{j}}|\zeta_{\boldsymbol{j}}|\right)$ [Gid18]. Applying the CADD circuit to Eq. (3.2.113), we obtain

$$(3.2.113) \overset{\mathrm{CADD}}{\to} \sum_{j_1\cdots j_d} v_{j_1\cdots j_d} |k_1(j_1)\rangle |k_2(j_2)\rangle\cdots|k_d(j_d)\rangle \left( \frac{1}{\sqrt{|\mathcal{I}_{\Gamma_N}||\zeta_{\boldsymbol{j}}|}} \sum_{\boldsymbol{k}\in\zeta_{\boldsymbol{j}}} |\boldsymbol{k}-\boldsymbol{j}\rangle \right) \tag{3.2.117}$$

$$\overset{\mathrm{O}_\zeta^{-1}}{\to} \left( \sum_{j_1\cdots j_d} v_{j_1\cdots j_d} |k_1(j_1)\rangle |k_2(j_2)\rangle\cdots|k_d(j_d)\rangle \right) |0\rangle , \tag{3.2.118}$$

where we also uncompute the register of boundary differences by $\mathrm{O}_\zeta^{-1}$. This means that $\prod_{\boldsymbol{j},\boldsymbol{k}}\mathrm{SWAP}(\boldsymbol{j},\boldsymbol{k})$

is implemented by

$$U_{\text{SWAPS}}\left(v_{\boldsymbol{j}}\,|\boldsymbol{j}\rangle_v\,|0\rangle_{\text{bdry},\zeta}\right) = \mathrm{O}_{\text{bdry}}^{-1}(\mathrm{O}_\zeta^{-1}\otimes|2\rangle\langle 2|_{\text{bdry}}) \cdot \text{CADD}\,(\mathrm{O}_\zeta\otimes|2\rangle\langle 2|_{\text{bdry}})\mathrm{O}_{\text{bdry}}\left(v_{\boldsymbol{j}}\,|\boldsymbol{j}\rangle_v\,|0\rangle_{\text{bdry},\zeta}\right). \tag{3.2.119}$$

Overall, we have the following registers,

- a state register $|\boldsymbol{j}\rangle_v$, consisting of $d\lceil\log_2(n)\rceil$ qubits for $n$ basis elements per dimension with binary representation;
- ancillas $|\cdot\rangle_{\text{bdry}}$ with 2 qubits to encode the type of basis element (inner, Dirichlet boundary, Neumann boundary);
- ancillas $|\cdot\rangle_\zeta$ to hold the neighbour set values coming from $\mathrm{O}_\zeta$, with $d\lceil\log_2\left(\max_{\boldsymbol{j}}|\zeta_{\boldsymbol{j}}|\right)\rceil$ qubits.

**Hamiltonian simulation of the derivative constraint projection.** In order to time-evolve a derivative constraint, we consider Lemma 3.2.13 that allow to avoid compiling $\mathrm{e}^{\mathrm{i}\lambda t\mathsf{P}_c^N}$ using Hamiltonian simulation techniques of the Hermitian projector $\mathsf{P}_c^N$. The following formula illustrates how this can be performed easily in the case of orthogonal projections.

**Lemma 3.2.13** (Exponential of an orthogonal projection.)**.** *Let* $\mathsf{P},\mathsf{Q}$ *be orthogonal and thus Hermitian projections on a vector space so that* $\mathsf{P}+\mathsf{Q}=\mathbb{I}$*. Then, for any* $\xi\in\mathbb{C}$*,*

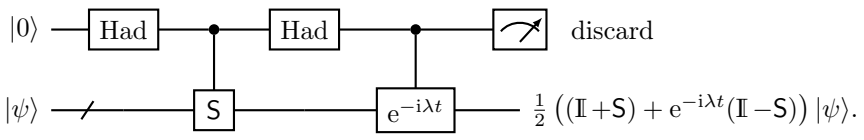
$$\mathrm{e}^{\xi\mathsf{P}} = \mathsf{Q} + \mathrm{e}^{\xi}\mathsf{P}. \tag{3.2.120}$$

*The case of* $\xi\in\mathrm{i}\,\mathbb{R}$ *recovers Hamiltonian simulation of an orthogonal projection.*

Hence, access to $\mathsf{P}_c^N$ and its orthogonal complement is sufficient for simulation in this context as only the image of the penalty projection is meant to experience a phase. Recall that the situation was the same for value constraints before, however, the complement there was simply the identity on the unconstrained domain and thus does not require an additional circuit implementation. Then, Lemma 3.2.13 permits us to implement $\mathsf{P}_c^{N\perp}+\mathsf{P}_c^N\mathrm{e}^{\mathrm{i}\lambda t}$ rather than $\mathrm{e}^{-\mathrm{i}\lambda t\mathsf{P}_c^N}$. More specifically, this is

$$\text{HamSim}(\mathsf{P}_c^N,\lambda t) = \frac{1}{2}(\mathbb{I}+\mathsf{S}) + \frac{\mathrm{e}^{-\mathrm{i}\lambda t}}{2}(\mathbb{I}-\mathsf{S}). \tag{3.2.121}$$

Indeed, this form is now equivalent to the value constraints up to modified projectors, where in the former case we can identify $\mathsf{Q}$ with $\sum_{\boldsymbol{j}\in\Omega\setminus\Gamma_D}|\boldsymbol{j}\rangle\langle\boldsymbol{j}|$ and $\mathsf{P}$ with $\sum_{\boldsymbol{j}\in\Gamma_D}|\boldsymbol{j}\rangle\langle\boldsymbol{j}|$. A circuit construction for $\mathsf{S}$ was given above in Eq. (3.2.119). Then, constructing a circuit for Eq. (3.2.121) is simple. Instead of packing things into extra LCUs, which would come at a constant factor loss in the success probability, we can do the following:




(3.2.122)

For the ancilla qubit, instead of projecting onto one of the basis states as before, we measure and discard the result (i.e., there is no post-selection necessary). The controlled phase gate $\mathrm{e}^{-\mathrm{i}\lambda t}$ is the

same as in the value constraint evolution in Eq. (3.2.100). This allows us to effectively fast-forward the evolution of $\mathsf{P}_c^N$, as the complexity of the circuit in Eq. (3.2.122) is independent of $\lambda t$. The ability to fast-forward also follows from [GSŞ21, Theorem 3.1 and Fig. 2].

**Interface conditions via 'derivative projection'.** The finite difference construction to approximate Neumann conditions is also useful to represent interface conditions, i.e., conditions between regimes that are governed by different PDEs. Suppose there are $\eta$ subdomains which represent a different physical model (expressed by a different linear PDE)

$$(\partial_t - \mathcal{L}_1)u_1(x;t) = 0,\ x \in \Omega_1, \quad \ldots, (\partial_t - \mathcal{L}_\eta)u_\eta(x;t) = 0,\ x \in \Omega_\eta, \tag{3.2.123}$$

then one can define interface conditions describe relationships at any intersections of the domains $k \neq l \in [\eta]$,

$$u_k(x;t) = \gamma\, u_l(x;t), \quad x \in \Omega_k \cap \Omega_l, \tag{3.2.124}$$

with $\gamma \in \mathbb{C}$; one can also think of more general relationships. The numerical treatment of interface conditions in classical numerical methods is described e.g. in [CZ15]. Our approach somewhat resembles the 'penalty method' described there, in the sense that it also introduces a notion virtual work that is minimized by admissible solutions.

In order to use 'Neumann projections' as in Eq. (3.2.103) we need an index set $\mathcal{I}_{\text{interface}}$ and neighbour sets $\bigcup_{\boldsymbol{j}\in\mathcal{I}_{\text{interface}}} \zeta_{\boldsymbol{j}}$. Using a grid discretization of the computational domain, then there are no more 'true' boundary points that are in $\Omega_k \cap \Omega_l$ but every point is in either $k$ or $l$; therefore, what we get are bipartite sets that give immediate rise to the necessary boundary point—neighbour relation. Upon proper definition of these sets and the according boundary oracles following Eq. (3.2.96), the treatment of interface conditions then follows immediately from how $\mathsf{P}_c^N$ is treated.

**Input Model for Robin Boundary Conditions** The ability to implement value constraints and derivative constraints allows us to extend the applicability to another class of boundary conditions. Constraints described by a superposition of Dirichlet and Neumann conditions are called Robin boundary conditions [GA98]; they can be expressed as

$$\alpha v(x;t) + \beta\, \partial_n \cdot v(x;t) = f(x;t), \quad x \in \Gamma_R, \tag{3.2.125}$$

with constants $\alpha, \beta$. Therefore, we can choose a projection

$$\mathsf{P}_c^R = \alpha \sum_{\boldsymbol{j}\in\mathcal{I}_{\Gamma_R}} |\boldsymbol{j}\rangle\langle\boldsymbol{j}| + \frac{\beta}{2}\left(\mathbb{I} - \prod_{\boldsymbol{j}\in\mathcal{I}_{\Gamma_R},\boldsymbol{k}\in\zeta_{\boldsymbol{j}}} \text{SWAP}(\boldsymbol{j},\boldsymbol{k})\right), \tag{3.2.126}$$

and additionally assume that $\alpha, \beta \in \mathbb{R}$ so that $\mathsf{P}_c^R$ is Hermitian. The point-wise projection $\sum_{\boldsymbol{j}\in\mathcal{I}_R} |\boldsymbol{j}\rangle\langle\boldsymbol{j}|$ maps any bitstrings from $\mathcal{I}_{\Omega\cup\Gamma}$ to those inside the Robin-boundary $\mathcal{I}_{\Gamma_R}$, whereas the SWAP-circuit takes any from $\mathcal{I}_{\Omega\cup\Gamma}$ to those that are considered neighbours for the Robin boundary $\mathcal{I}_{\Gamma_R\cap\Omega} = \bigcup_{\boldsymbol{j}\in\mathcal{I}_{\Gamma_R}} \zeta_{\boldsymbol{j}}$. Therefore, the commutativity of these operations depends on how the sets are defined. One option is to use 'outside' ghost points for the finite differences in the Neumann projections for the rather than points inside the domain, and include these to the image of the Dirichlet projection. Then, the operations commute restricted to the domain of interest (that is,

disregarding the ghost points) and one can compile $\exp\big(-\mathrm{i}\lambda t \mathsf{P}_c^R\big)$ with a product formula

$$\exp\big(-\mathrm{i}\lambda t \mathsf{P}_c^R\big) = \exp\left(-\mathrm{i}\alpha\lambda t \sum_{\boldsymbol{j}\in\mathcal{I}_{\Gamma_R}} |\boldsymbol{j}\rangle\langle\boldsymbol{j}|\right) \exp\left(-\mathrm{i}\frac{\beta}{2}\lambda t(\mathbb{I}-\mathsf{S}_R)\right), \tag{3.2.127}$$

or equivalently

$$\mathrm{HamSim}(\mathsf{P}_c^R, \lambda t) = \mathrm{HamSim}(\mathsf{P}_c^D|_{\mathcal{I}_{\Gamma_R}}, \alpha\lambda t)\mathrm{HamSim}(\mathsf{P}_c^N|_{\mathcal{I}_{\Gamma_R}}, \beta\lambda t), \tag{3.2.128}$$

where we know how to implement both from above, Eqs. (3.2.100) and (3.2.121).

**Hamiltonian simulation of combined projections** Using the results from the previous two sections, we can express the Hamiltonian simulation of orthogonal value and derivative projections as follows,

$$\begin{aligned}\mathrm{HamSim}(\mathsf{P}_c, \lambda t) = \mathrm{O}_{\mathrm{bdry}}^{-1} \Big(&\mathrm{HamSim}(\mathsf{P}_c^D, \lambda t) \otimes |1\rangle\langle 1|_{\mathrm{bdry}} \\ &+\mathrm{HamSim}(\mathsf{P}_c^N, \lambda t) \otimes |2\rangle\langle 2|_{\mathrm{bdry}} \\ &+\mathrm{HamSim}(\mathsf{P}_c^D, \alpha\lambda t)\mathrm{HamSim}(\mathsf{P}_c^N, \beta\lambda t) \otimes |3\rangle\langle 3|_{\mathrm{bdry}} \Big)\mathrm{O}_{\mathrm{bdry}},\end{aligned} \tag{3.2.129}$$

where $\mathrm{HamSim}(\mathsf{P}_c^D, \lambda)$ requires a call to controlled $\mathrm{e}^{-\mathrm{i}\lambda t}$ and $\mathrm{HamSim}(\mathsf{P}_c^N)$ needs additionally a controlled implementation of the swap $\mathsf{S}$. Recall that $|1\rangle_{\mathrm{bdry}}$ marks Dirichlet points, $|2\rangle_{\mathrm{bdry}}$ Neumann points and $|3\rangle_{\mathrm{bdry}}$ Robin boundary points.

### 3.2.3 Numerical Experiments

In the following, we are presenting numerical experiments as proof-of-concept validation of the penalty projections method via interaction picture based on classical simulation of the dynamics. In addition, we are interested in tightness of our bounds on $\lambda$ derived in Section 3.2.2. To that end, we consider the heat equation and the wave equation as two of the canonical examples for 'simple' PDEs. While for the heat equation, a quantum speedup beyond a quadratic one coming from amplitude amplification cannot be expected in general when evaluating an expectation value with respect to the final state (see [LMS22b]), it still serves as an illustrative example. However, as shown in [CJO19] and recovered as the isotropic case in [Bab+23], the wave equation, which in contrast to the heat equation is not dissipative and obeys conservation of energy, allows for an at least quartic speedup.

#### Setup

In the computations below, we restricted the possible range of $\lambda$ to 'small' values (up to $10^6$) to avoid aliasing in the numerical solution of the highly oscillatory system in interaction picture. Note that this is not as much of a problem in the quantum implementations, as we will outline below, as the necessary time discretization only grows with $\log(\lambda)$. For spatial discretization, we use a simple central three-point finite difference formula in periodic boundary conditions form to represent the Laplacian. We use the Runge-Kutta scheme RK23 [BS89] to simulate the time evolution. Even though this is not high-accuracy, it proves sufficient for the regimes we are looking at; in order

to capture high-frequency effects of the interaction picture simulation of $A_I(t)$, a relatively small time-step is already required, hence the accuracy is sufficient here to ensure that the penalty error is the dominant source of error. We are solving the following discretized PDEs by time-stepping the interaction picture equations from Eq. (3.2.143) with respect to the discretized differential operators.

**Simulation results**

In what follows, we present numerical experiments for dimensionless heat and wave equations, discretized by finite differences.

**Heat equation with Dirichlet and Neumann boundary conditions** We consider the following 2D-problem for the heat equation:

$$\partial_t u(x,y;t) = D\Delta u(x,y;t) + f(x,y;t) \tag{3.2.130}$$

$$u(x,y;t)|_{x,y\in\Gamma_D} = g(x,y;t) \tag{3.2.131}$$

$$\partial_n \cdot u(x,y;t)|_{x,y\in\Gamma_N} = h(x,y;t) \tag{3.2.132}$$

$$u(x,y;t=0) = u_0(x,y). \tag{3.2.133}$$

with the temperature distribution $u(x,y;t)$ and diffusion coefficient $D>0$; for the sake of simplicity, we choose isotropic diffusion here. The initial conditions are described by $u_0$, $f$ is a forcing term (heat source) and $g$ describes the temperature along the boundary $\Gamma_D$, where $h$ describes the temperature flux across the boundary $\Gamma_N$. Then, we introduce an equidistant uniform grid for space to obtain $\mathbf{u}_h(t) \in \mathbb{R}^{N^2}$ using $N$ points to represent $x$ and $y$, and a fourth-order finite difference stencil to approximate $\Delta$ via $\mathbf{L}_h \in \mathbb{R}^{N^2\times N^2}$ with truncation error $O(N^{-4})$. Then, we have the discretized ODE system

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{u}_h(t) = D\mathbf{L}_h\mathbf{u}_h(t) + \mathbf{f}(t), \tag{3.2.134}$$

and use the constraint projections as defined earlier in Eqs. (3.2.97) and (3.2.103) and use the homogenization strategies outlined in Section 3.2.2 for non-zero boundary conditions.

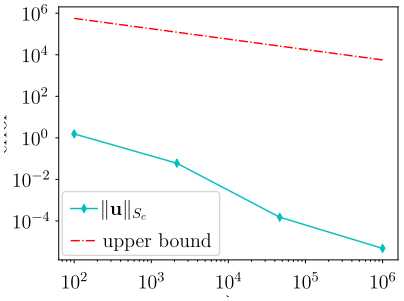
We examine the following choices of boundary conditions:

- Vanishing boundary conditions, $\mathsf{P}_c^D\mathbf{u}_h(t) = 0$. See Figs. 3.9 and 3.11.
- Non-zero boundary conditions, $\mathsf{P}_c^D\mathbf{u}_h(t) = g(\mathbf{x},\mathbf{y})$

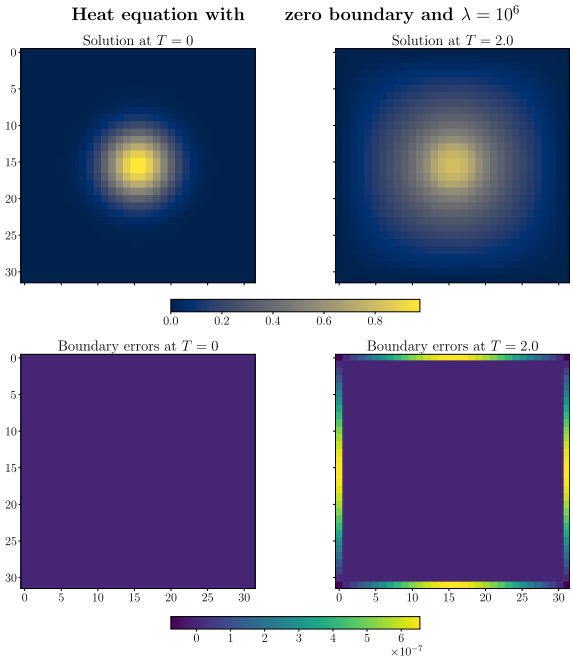
$$\mathbf{g} = (1-\mathbf{y}) \odot \begin{cases} \mathbf{x}, & \mathbf{x} \le \frac{1}{2} \\ (1-\mathbf{x}), & \mathbf{x} > \frac{1}{2}, \end{cases} \tag{3.2.135}$$

  where $\mathbf{x} = [0, \Delta x, 2\Delta x, \ldots,]^T$ and $\mathbf{y} = [0, \Delta y, 2\Delta y, \ldots,]^T$ are the grid representation of the $x,y$ domain and $\odot$ denotes element-wise multiplication. This is implemented via the approach outlined in Section 3.2.2. Results are shown in Fig. 3.10.

- No heat in/out-flux: $\mathsf{P}_c^N\mathbf{u}_h(t) = 0$. The boundary values are zero in compliance to the initial condition. If the initial condition is equal to zero on the boundary indices, then this scenario can test both the usage of the Neumann projectors to enforce a value as well as the gradient. The numerical results are depicted in Fig. 3.12.




(a) Boundary error $\|\mathbf{u}_h\|_{S_c}$ vs penalty parameter; upper bound follows Lemma 3.2.8




(b) Solution and boundary values at initial vs final time.

Figure 3.9: Numerical simulations of heat equation with $\mathsf{P}_c^D \mathbf{u}_h = 0$, $D = 4$, $N = 2^5$, $t = 1$, and $\Delta t = 10^{-5}$. The boundary is defined as the outer grid points ('wall').

All simulations are dimensionless, and as initial state we use a centred, isotropic Gaussian with height 1 and define a cut-off so that the initial condition respects the boundary constraints. Additionally, we use a point-source $\mathbf{f}(t) = 298$ at the centre element.

**Wave equation with Dirichlet boundary conditions** The 2-D wave equation is defined as

$$\partial_{tt} u(x, y; t) = c^2 \Delta u(x, y; t), \tag{3.2.136}$$

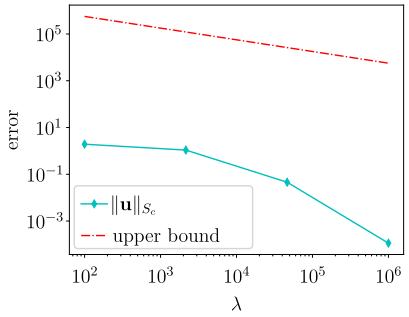
with a speed-of-sound parameter $c^2 > 0$. In order to solve the wave equation, let $w(x, y; t) = \partial_t u(x, y; t)$. Then, Eq. (3.2.136) is equivalent to the system

$$\partial_t \mathbf{v}(t) = \partial_t \begin{bmatrix} u(t) \\ w(t) \end{bmatrix} = \begin{bmatrix} 0 & \mathbb{I} \\ c^2\Delta & 0 \end{bmatrix} \begin{bmatrix} u(t) \\ w(t) \end{bmatrix} = \mathbf{A}\mathbf{v}(t), \tag{3.2.137}$$

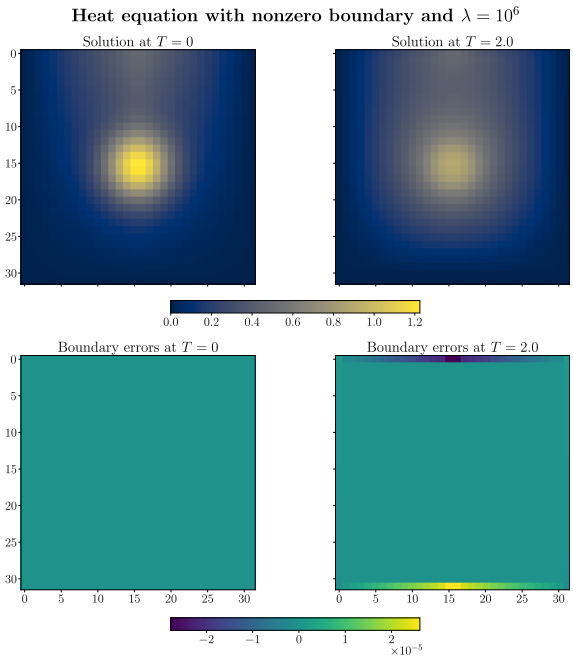
with initial data $\mathbf{v}(t) = \begin{bmatrix} u(x,y;t=0) \\ \partial_t u(x,y;t=0) \end{bmatrix}$. Upon discretizing space on a uniform, regular grid with $N$ points per spatial dimension, we obtain a matrix $\mathbf{L}_h \in \mathbb{R}^{N^2 \times N^2}$ and overall system matrix $\mathbf{A}_h \in \mathbb{R}^{2N^2 \times 2N^2}$, $\mathbf{v}_h(t) \in \mathbb{R}^{2N^2}$. For the constrained system,

$$\frac{\mathrm{d}}{\mathrm{d}t} \mathbf{v}_h(t) = (\mathbf{A}_h - \mathrm{i}\lambda \mathbf{P}_c)\, \mathbf{v}_h(t)$$

we apply a penalty projection on both the original variable $\mathbf{u}$ and $\mathbf{w} = \partial_t \mathbf{u}$, $\mathbf{P}_c = \begin{bmatrix} \mathsf{P}_c & 0 \\ 0 & \mathsf{P}_c \end{bmatrix}$. As the value of interest is only $\mathbf{u}(t)$, we measure the error via $\|\mathbf{u}\|^2_{\ell_2, S_c} = \langle \mathbf{v}, \widetilde{\mathbf{P}}_c \mathbf{v} \rangle$ so that $\widetilde{\mathbf{P}}_c = \begin{bmatrix} \mathsf{P}_c & 0 \\ 0 & \mathbb{I} \end{bmatrix}$.

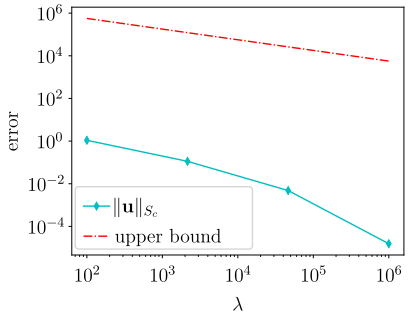



(a) Boundary error $\|\mathbf{u}_h\|_{S_c}$ for $N = 2^5, \Delta t = 10^{-5}$, $D = 4$ and boundary along the four walls upper bound follows Lemma 3.2.8.

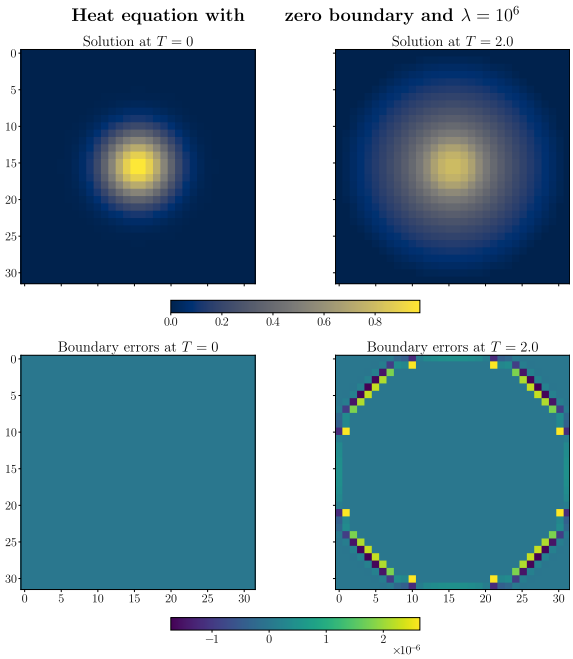



(b) Solution and boundary values at initial vs final time.

Figure 3.10: Numerical simulations of heat equation with $\mathsf{P}_c^D \mathbf{u}_h = \mathbf{g}$ from Eq. (3.2.135), $D = 4$, $N = 2^5$, $t = 1$, and $\Delta t = 10^{-5}$. The boundary is defined to be the outer grid points ('wall').

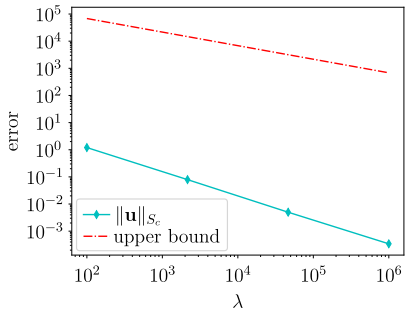



(a) Boundary error $\|\mathbf{u}_h\|_{S_c}$ for $N = 2^5, \Delta t = 10^{-5}$, $D = 4$ and boundary along a circle boundary Lemma 3.2.7

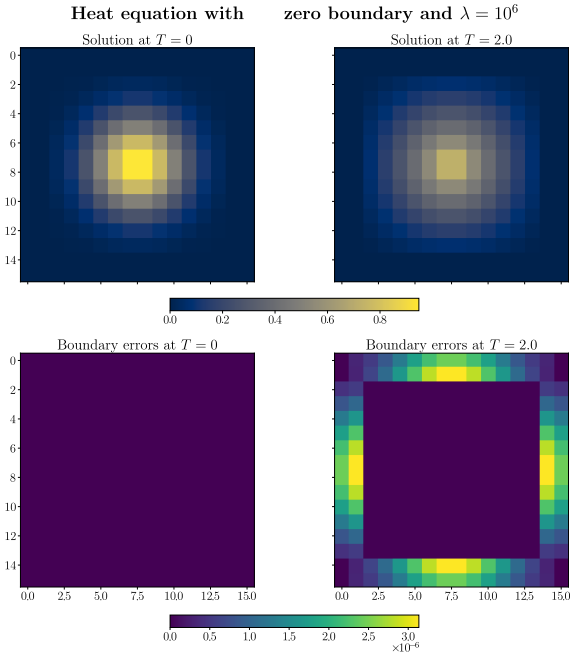



(b) Solution and boundary values at initial vs final time.

Figure 3.11: Numerical simulation of heat equation with $\mathsf{P}_c^D \mathbf{u} = \mathbf{g}$ from Eq. (3.2.135), $D = 4$, $N = 2^5$, $t = 1$, and $\Delta t = 10^{-5}$. The boundary is defined to be on a circle of radius $\frac{1}{2}$.




(a) Neumann boundary error $\|\mathbf{u_h}\|_{S_c}$ on wall boundary with respect to 'derivative constraint' $\mathsf{P}_c^N$




(b) Solution and boundary values at initial vs final time.

Figure 3.12: Numerical simulations of heat equation with $\mathsf{P}_c^N \mathbf{u}_h = 0$ from Eq. (3.2.135), $D = 4$, $N = 2^4$, $t = 2$, and $\Delta t = 10^{-3}$. The constraint region is formed by the differences of the outer points ('wall') with their closest neighbours inwards. The corner points as well as one of the points next to the corner are skipped so that every inner point is only neighbour to at most one corner point.

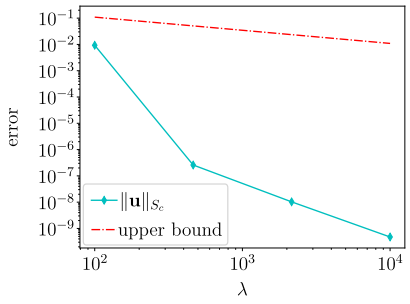
Recall that we generally assumed that $\mathrm{Re}(A) \preceq 0$. Looking for the eigenvalues $\chi$ of the discrete operator,

$$\begin{bmatrix} 0 & \mathbb{I} \\ c^2\mathbf{L}_h & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{w} \end{bmatrix} = \chi \begin{bmatrix} \mathbf{u} \\ \mathbf{w} \end{bmatrix}, \tag{3.2.138}$$

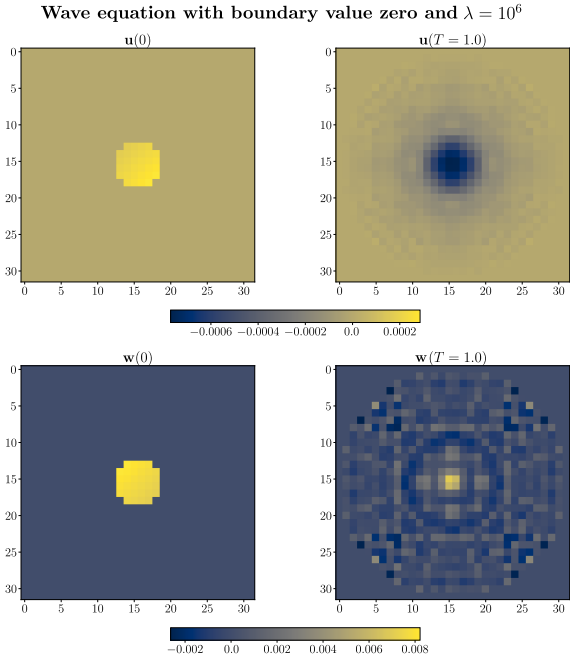
we obtain that $\mathbf{w} = \chi\mathbf{u}$ and $c^2\mathbf{L}_h\mathbf{u} = \chi\mathbf{w}$. Therefore, $\mathbf{L}_h\mathbf{u} = (\chi/c)^2\,\mathbf{u}$. Both the abstract Laplacian $\Delta$ as well as for the discretized $\mathbf{L}_h^{\mathrm{per}}$ with periodic boundary conditions have a spectrum (contained in) $(-\infty, 0]$. Therefore, $\chi$ as in Eq. (3.2.138) is purely imaginary and $\mathrm{Re}\left(\begin{bmatrix} 0 & \mathbb{I} \\ c^2\mathbf{L}_h & 0 \end{bmatrix}\right) = 0$. Thus, as long as there is no interaction with an environment (i.e., periodic boundary conditions), there is no dissipation. The initial condition is given by

$$\begin{bmatrix} \mathbf{v}_0 \\ \mathbf{w}_0 \end{bmatrix} = \begin{bmatrix} 0.001\,\sin(\mathbf{x})\sin(\mathbf{y}) \\ 0.01\,\cos(\mathbf{x})\cos(\mathbf{y}) \end{bmatrix} \text{ if } \mathbf{x}^2 + \mathbf{y}^2 \leq 0.1, \text{ else } 0. \tag{3.2.139}$$

The simulation results are shown in Fig. 3.13.




(a) Boundary error $\|\mathbf{u}\|_{S_c}$, upper bound follows Lemma 3.2.8.




(b) $\mathbf{u}_h$ and $\mathbf{w}_h$ at $t = 0$ and final time $t = 1$.

Figure 3.13: Numerical experiments for wave equation with zero boundary conditions at radius $\frac{1}{2}$. $N = 2^5$, $\Delta t = 10^{-4}$, $t = 1$, $c^2 = 1$.

**Discussion** The results to our simulations are shown in Figs. 3.9 to 3.12 and Fig. 3.13. For all simulations, we can observe that the errors that we witness decrease quadratically, or slightly faster, with $\lambda$, i.e. $\|\mathbf{u}\|_{S_c} \in O(\lambda^{-1/2})$. This aligns very well with the upper bounds derived previously. The measured errors are several orders of magnitude smaller than the upper bounds that we derived. Even though the asymptotic behaviour seems to be well-captured in the bounds, they appear to be quite loose. This is not necessarily surprising as they rely on first-order perturbation theory, and within the regimes we studied numerically, the ratio of perturbation $\lambda$ vs. 'system energy' $|\langle v, A_0 v\rangle|$ is not in a clearly perturbative regime yet.

The results in Fig. 3.10 validates the efficacy of the procedure of homogenizing/shifting the solution in order to solve for non-zero constraints. Moreover, we tested a circle boundary in Fig. 3.11 as a simple example for a non-trivial geometry. In particular, we notice that the solution does not penetrate further than two grid points into the constraint region within the chosen discretization. Additionally note that the constraint error decays faster in the case of the wave equation in Fig. 3.13; roughly linear if not including the first point at $\lambda = 100$. In the case of the heat equation, we observe that error performance is more reliable and more favourable in the case of derivative/Neumann constraints (Fig. 3.12). Therefore, using this setup in combination with 'ghost points' that store a desired value and are out of the range of the generator $A_0$ could be an interesting alternative to directly enforcing the value constraints.

### 3.2.4 Quantum Algorithm via Interaction-Picture simulation

Recall that the constrained ODEs we consider in this work have the form

$$\frac{\mathrm{d}}{\mathrm{d}t}v(t) = A_0(t)v(t) - \mathrm{i}\lambda\mathsf{P}_c v(t) + b(t) \equiv A(t)v(t) + b(t), \tag{3.2.140}$$

with initial data $v(t=0) = v_0$ and an orthogonal projection $\mathsf{P}_c$. The general solution to this via the variation of parameters formula is

$$v(t) = \mathcal{T}\mathrm{e}^{\int_0^t A(s)\mathrm{d}s}v_0 + \int_0^t \mathcal{T}\mathrm{e}^{\int_s^t A(s')\mathrm{d}s'}b(s)\mathrm{d}s. \tag{3.2.141}$$

There exist many quantum DE solvers that we could employ here; a common trait among all of them is that they depend (in the optimal case, linearly) on the block-encoding or subnormalization factor $\alpha_A \geq \sup_{0\leq t\leq T}\|A(t)\|$, which itself, by Eq. (3.2.140), is linear in $\lambda$. As we saw before in Section 3.2.2, a necessary penalty factor is at least of the size $\lambda \gtrsim v_{\max}^2 \frac{\|A\|}{\varepsilon}$, which would render implementations of the constraint projection impractical.

We can overcome this limitation by simulating Eq. (3.2.140) in the interaction frame of the projection. Then, the overhead in $\lambda$ compared to unconstrained evolution is only logarithmic. The reason for this is that an interaction picture transformation is unitary, thereby $\alpha_{A_I} = \alpha_{A_0}$, where $A_I(t) = \mathrm{e}^{-\mathrm{i}\lambda\mathsf{P}_c t}A_0(t)\mathrm{e}^{\mathrm{i}\lambda\mathsf{P}_c t}$. The logarithmic overhead in the gate complexity is expected as the number of grid-points to simulate the interaction picture dynamics depends on the maximum frequency [LW18, Lemma 5], as $\|\partial_t A_I(t)\| = O(\lambda)$. A necessary condition of this to work efficiently is that the cost of the interaction picture transformation itself, i.e., $\mathrm{e}^{-\mathrm{i}\lambda\mathsf{P}_c t}$, does not depend on $\lambda$. We know from Section 3.2.2 that $\mathsf{P}_c$ is fast-forwardable by construction – which makes the setup suited for such an interaction picture implementation [LW18].

Therefore, let $v_I(t) = \mathrm{e}^{\mathrm{i}\lambda\mathsf{P}_c t}v(t)$. A quick calculation shows that

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{e}^{\mathrm{i}\lambda\mathsf{P}_c t}v(t) &= \mathrm{e}^{\mathrm{i}\lambda\mathsf{P}_c t}\mathrm{i}\lambda\mathsf{P}_c v(t) + \mathrm{e}^{\mathrm{i}\lambda\mathsf{P}_c t}\frac{\mathrm{d}}{\mathrm{d}t}v(t) &(3.2.142)\\ &= \mathrm{e}^{\mathrm{i}\lambda\mathsf{P}_c t}\mathrm{i}\lambda\mathsf{P}_c v(t) + \mathrm{e}^{\mathrm{i}\lambda\mathsf{P}_c t}(A - \mathrm{i}\lambda\mathsf{P}_c)\mathrm{e}^{-\mathrm{i}\lambda\mathsf{P}_c t}\mathrm{e}^{\mathrm{i}\lambda\mathsf{P}_c t}v(t)\\ &= \underbrace{\mathrm{e}^{\mathrm{i}\lambda\mathsf{P}_c t}A\mathrm{e}^{-\mathrm{i}\lambda\mathsf{P}_c t}}_{=A_I(t)}\mathrm{e}^{\mathrm{i}\lambda\mathsf{P}_c t}v(t) = A_I(t)v_I(t). &(3.2.143)\end{aligned}$$

This means that if we solve for

$$v_I(t) = \mathcal{T}\mathrm{e}^{\int_0^t A_I(s)\mathrm{d}s} v_0 + \int_0^t \mathcal{T}\mathrm{e}^{\int_s^t A_I(s')\mathrm{d}s'} b_I(s)\mathrm{d}s =: S_I(t) v_I(0), \tag{3.2.144}$$

we can retrieve the sought-after solution $v(t)$ through implementing

$$\mathrm{HamSim}(\mathsf{P}_c, -\lambda) \quad \underbrace{S_I(t)}_{\text{ODE solver}} \quad \mathrm{HamSim}(\mathsf{P}_c, +\lambda). \tag{3.2.145}$$

**Time Evolution by LCHS**

We use the near-optimal Linear Combination of Hamiltonian simulations approach [ALL23; ACL23] as a quantum ODE solver. For our purposes, this choice is more natural to tackle ODEs or discretized PDEs compared to Schrödingerisation [JLY24; JLMY25], has better complexity with respect to approximation error and time compared to time-marching approaches [FLT23], and has better complexity with respect to state preparation compared to the linear-systems based approaches. The LCHS theorem in [ALL23] allows to write the (time-ordered) exponential of a general matrix $A(t)$ as an integral of a unitary time evolution with a specific kernel function $\frac{f(k)}{1-\mathrm{i}k}$, namely

$$\mathcal{T}\mathrm{e}^{\int_0^t A(s)\mathrm{d}s} = \int_{\mathbb{R}} \frac{f(k)}{1-\mathrm{i}k} \mathcal{T}\mathrm{e}^{\int_0^t \mathrm{i}(kA_{\mathrm{Re}}+A_{\mathrm{Im}})(s)\mathrm{d}s}\mathrm{d}k, \tag{3.2.146}$$

where the real and imaginary part of a matrix $A$ respectively are

$$A_{\mathrm{Re}} = \frac{A + A^\dagger}{2}, \quad A_{\mathrm{Im}} = \frac{A - A^\dagger}{2\mathrm{i}}. \tag{3.2.147}$$

Duhamel's formula gives an extension to particular solutions given a source term [ALL23]. In our case, the ODE system is defined by the interaction picture matrix

$$A_I(t) = \mathrm{e}^{\mathrm{i}\lambda\mathsf{P}_c t}(A_0)\mathrm{e}^{-\mathrm{i}\lambda\mathsf{P}_c t} = \mathrm{e}^{\mathrm{i}\lambda\mathsf{P}_c t}(A_{0,\mathrm{Re}} + A_{0,\mathrm{Im}})\mathrm{e}^{-\mathrm{i}\lambda\mathsf{P}_c t}. \tag{3.2.148}$$

We note that [AT23] provides an algorithm of LCHS with interaction picture, however in a different setting. In our case, the operator for the interaction frame itself is a Hamiltonian simulation and does not require treatment by the LCHS formula. Furthermore, in [AT23], the authors are using LCHS with the non-optimal kernel function from [ACL23] which does not achieve high accuracy. Choosing $\lambda \in \mathrm{i}\,\mathbb{R}$ would recover the approach in [AT23, Lemma 17] which could easily be extended to the higher-order quadratures and improved kernel function of [ACL23]. However, this would lead to higher simulation cost as we would need to simulate the linear combination $\mathrm{e}^{-|\lambda|\mathsf{P}_c t\cdot k_j}$ for all $k_j = -K, \ldots, K$ with $[-K, K]$ the discretization of the Fourier domain in the LCHS formula Eq. (3.2.146).

The solution of ODEs using LCHS requires access to a HAM-T oracle (Hermitian block-encoding access 'HAM' with a time parameter 'T') for $A_I(t)$ and a fast-forwarded Hamiltonian simulation $\mathrm{e}^{-\mathrm{i}\lambda t\mathsf{P}_c}$. For completeness, we briefly recall the LCHS algorithm from [ACL23] applied to $\mathcal{T}\mathrm{e}^{\int_0^t A_I(s)\mathrm{d}s}$. Given $k \in \mathbb{R}$, consider the Hamiltonian simulation integrand in (3.2.146),

$$U(k;t) = \mathcal{T}\mathrm{e}^{\mathrm{i}\int_0^t kA_{I,\mathrm{Re}}+A_{I,\mathrm{Im}}\mathrm{d}s}. \tag{3.2.149}$$

This unitary satisfies

$$\frac{\mathrm{d}}{\mathrm{d}t}U(k;t) = \mathrm{i}(kA_{I,\mathrm{Re}}(t) + A_{I,\mathrm{Im}}(t))U(k;t), \quad U(k;0) = \mathbb{I}, \tag{3.2.150}$$

where

$$A_{I,\mathrm{Re}}(t) = \mathrm{e}^{\mathrm{i}\lambda\mathsf{P}_c t}A_{0,\mathrm{Re}}(t)\mathrm{e}^{-\mathrm{i}\lambda\mathsf{P}_c t}, \quad A_{I,\mathrm{Im}}(t) = \mathrm{e}^{\mathrm{i}\lambda\mathsf{P}_c t}A_{0,\mathrm{Im}}(t)\mathrm{e}^{-\mathrm{i}\lambda\mathsf{P}_c t}. \tag{3.2.151}$$

Simulating $U(k;t)$ is achieved by a truncated Dyson series algorithm [LW18]. Then, the integral over $k$ in Eq. (3.2.146) is truncated to some finite interval $[-K;K]$ and evaluated with a higher-order quadrature; these steps amount to a linear combination the unitaries $U(k;t)$. These routines thus require HAM-Taccess to $A_{0,\mathrm{Re}}(t)$, $A_{0,\mathrm{Im}}(t)$ and access to $\mathrm{e}^{-\mathrm{i}\lambda t\mathsf{P}_c}$.

### Oracles

**HAM-T oracles** The HAM-T-oracles follow the convention from [ACL23; ALL23]. The truncated Dyson series is implemented across short-time integrators with length $\Delta t$ using $M_D$ time-steps. Let $q \in [M_D]$ and $|0\rangle_a$ a register of ancillas, then

$$\langle 0|_a \,\text{HAM-T}_{A_I,q}\, |0\rangle_a = \sum_{l=0}^{M_D-1} |l\rangle\langle l| \otimes \frac{A_I(q\Delta t + l\frac{\Delta t}{M})}{\alpha_A}. \tag{3.2.152}$$

This leads to the oracle for long-time integration with Hamiltonian $kA_{I,\mathrm{Re}} + A_{I,\mathrm{Im}}$,

$$\langle 0|_a \,\text{HAM-T}_{kA_{I,\mathrm{Re}}+A_{I,\mathrm{Im}},q}\, |0\rangle_a = \sum_{j=0}^{M-1}\sum_{l=0}^{M_D-1} |j\rangle\langle j| \otimes |l\rangle\langle l| \otimes \frac{k_j A_{I,\mathrm{Re}}(q\Delta t + l\frac{\Delta t}{M_D}) + A_{I,\mathrm{Im}}(q\Delta t + l\frac{\Delta t}{M_D})}{\kappa\alpha_{A_{I,\mathrm{Re}}} + \alpha_{A_{I,\mathrm{Im}}}}, \tag{3.2.153}$$

where $\kappa = \|[-K,\ldots,K]\|_{\ell_1}$ and $M$ time steps is for $q \in [M_D]$. Every oracle call to this HAM-T oracle will require one call to Hamiltonian simulation as in Eq. (3.2.129) and one to its inverse.

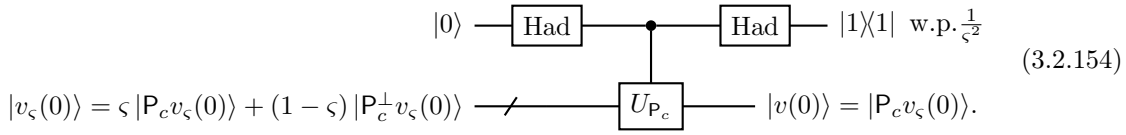
**Further oracles** Additionally, we require the following oracles for the LCHS algorithm:

- Circuits for the HamSim routines in Eq. (3.2.129), within the HAM-T construction and to, if desired, transform back from the interaction picture to the "Schrödinger picture" of the untransformed solution state.
- A state preparation oracle $\mathrm{O}_v$: $|0\rangle \to |v(0)\rangle$.
- An oracle to prepare the forcing term $\mathrm{O}_b$: $|0\rangle|\tau\rangle \to |b(\tau)\rangle|\tau\rangle$ for all time-steps $\tau \in [n_t - 1]_0$.

For cases when constructing oracles to initial states that approximately satisfy the constraint is significantly more efficient, or constructing a state that follows the constraint exactly is not possible, consider the following Lemma 3.2.14.

**Lemma 3.2.14** (Ensuring constraint-satisfying input states)**.** *Suppose we have an input oracle* $\mathrm{O}_{v,\varsigma}$ *so that* $\mathrm{O}_{v,\varsigma} : |0\rangle \mapsto |v_\varsigma(0)\rangle$ *so that* $\|\mathsf{P}_c v_\varsigma(0)\| \leq \varsigma$ *is not necessarily zero. Furthermore, we assume access to a circuit implementation of* $\mathsf{P}_c$ *via* $U_{\mathsf{P}_c}$*, such as Eqs.* (3.2.98) *and* (3.2.104)*. Then, one can produce a state* $|\mathsf{P}_c v_\varsigma(0)\rangle$ *with at most* $O\left(\frac{1}{\varsigma}\right)$ *calls to (controlled-)*$U_{\mathsf{P}_c}$*.*

*Proof.* Note that $U_{\mathsf{P}_c}$ unitarily implements the effect of the projection. Now, applying quantum phase estimation of $U_{\mathsf{P}_c}$ to the initial state $|v_\varsigma(0)\rangle$ has the following effect,




(3.2.154)

What this means is that we can trade off the error in overlap with success probability as we post-select on the $\mathsf{P}_c$-subspace, given by the 1-eigenvalue of $U_{\mathsf{P}_c}$. Then, this can be boosted using fixed-point amplitude amplification to cost $O(1/\varsigma)$ [GSLW19, Theorem 27]. □

### Complexity

In this section, we summarize the overall complexity. We first provide the final result and subsequently walk through the analysis.

**Final Result** In Table 3.2a, we summarize the necessary strengths of the penalty through lower bounds on $\lambda$. Then, as we show in the subsequent section after stating the main complexity result, this allows us to derive the smoothness parameters $\Lambda_I, \Xi_I$ as collected in Table 3.2b, which are part of the following theorem.

**Theorem 3.2.15** (Complexity of linear constrained ODE solution using LCHS, adapted from Theorem 14 in [ACL23])**.** *Consider the inhomogeneous ODE system in Problem 3. Suppose that $A_{I,\mathrm{Re}}(\tau) \preceq 0$ on $[0,t]$, and we are given the oracles described in Section 3.2.4. Let $\|A(\tau)\| \leq \alpha_A$ and define $\Lambda_I = \sup_{p\geq 0,\tau\in[0,t]} \|(\partial_\tau)^p A_I(\tau)\|^{\frac{1}{p+1}}$ and $\Xi_I = \sup_{p\geq 0,\tau\in[0,t]} \|(\partial_\tau)^p b_I(\tau)\|^{\frac{1}{p+1}}$. Then we can prepare an $\varepsilon$-approximation of the normalized solution $|v(t)\rangle$ with constant probability and a flag indicating success, by choosing*

$$M \in O\left(\alpha_A t \left(\log\left(\frac{\|v_0\| + B_{L^1}}{\|v(T)\|\varepsilon}\right)\right)^{1+1/\beta}\right), \quad M' \in \widetilde{O}\left(t\,\frac{\Gamma \cdot v_{\max}^2 \|A_0\|}{\varepsilon}\left(\log\left(\frac{1+B_{L^1}}{\|v(T)\|\varepsilon}\right)\right)^{1+1/\beta}\right), \tag{3.2.155}$$

*where $\Gamma \frac{v_{\max}^2\|A_0\|}{\varepsilon} = \Lambda_I + \Xi_I$. Different cases for $\Gamma$ are summarized in Table 3.2b, and $\Lambda_I, \Xi_I$ are at most $O\left(\frac{t^2}{\varepsilon}(v_{\max}^2 + B_{L^1}^2)\|A_0\|\right)$. Further, this requires*

$$\widetilde{O}\left(\frac{\|v(0)\| + B_{L^1}}{\|v(t)\|}\alpha_{A_0} t \left(\log\left(\frac{1}{\varepsilon}\right)\right)^{1+1/\beta}\right) \tag{3.2.156}$$

*queries to the HAM-T oracle and*

$$O\left(\frac{\|v(0)\| + B_{L^1}}{\|v(t)\|}\right) \tag{3.2.157}$$

*queries to the state preparation oracles $O_v$ and $O_b$.*

Note that the penalty affects the complexity *only* through the number of discretization points necessary for the inhomogeneous solution, $M'$. Thereby, it does enter the final gate complexity and is not visible in the query complexity in Eq. (3.2.156).

**Analysis of smoothness factors** Before we can do so, we look at the impact of the penalty projection. Intuitively, simulating $A_I(t)$ in the interaction picture with simulation parameter $\lambda t$ means that we simulate a highly oscillatory system, as $\lambda$ is large. Therefore, this will have impact on the size of the time steps to avoid aliasing. To that end, consider the smoothness parameters $\Lambda_I \geq \sup_{p\geq 0, t\in[0,T]} \|(\partial_t)^p A_I(t)\|^{\frac{1}{p+1}}, \Xi_I \geq \sup_{p\geq 0, t\in[0,T]} \|(\partial_t)^p b_I(t)\|^{\frac{1}{p+1}}$ from [ACL23], which need to be adjusted to the interaction picture simulation. Then, we obtain

$$(\partial_t)^p A_I(t) = (\partial_t)^p \left[\mathrm{e}^{\mathrm{i}\lambda \mathsf{P}_c t} A(t) \mathrm{e}^{-\mathrm{i}\lambda \mathsf{P}_c t}\right] \tag{3.2.158}$$

$$(\partial_t)^p b_I(t) = (\partial_t)^p \left[\mathrm{e}^{-\mathrm{i}\lambda \mathsf{P}_c t} b(t)\right] \tag{3.2.159}$$

We can expand this as

$$\begin{aligned}(\partial_t)^p A_I(t) &= (\partial_t)^{p-1}\left((\dot{A}(t)_I + \mathrm{i}\lambda[\mathsf{P}_c, A_I(t)]\right) = (\partial_t)^{p-2}\left((\mathrm{i}\lambda)^2[\mathsf{P}_c,[\mathsf{P}_c,A_I]] + 2\mathrm{i}\lambda[\mathsf{P}_c,(\dot{A})_I] + (\ddot{A})_I\right) \\ &= \ldots = \sum_{q\leq p}\binom{p}{q}\mathrm{ad}^q_{\mathrm{i}\lambda\mathsf{P}_c}((\partial_t^{p-q}A(t))_I) \end{aligned} \tag{3.2.160}$$

$$(\partial_t)^p b_I(t) = (\partial_t)^{p-1}\left(\mathrm{i}\lambda \mathsf{P}_c b_I(t) + (\dot{b})_I(t)\right) = \sum_{q\leq p}\binom{p}{q}(\mathrm{i}\lambda)^q \mathsf{P}_c(\partial_t^{p-q} b(t))_I \tag{3.2.161}$$

where we use the notation $\mathrm{ad}_y(x) = [y,x]$, $\mathrm{ad}_y^m(x) = \underbrace{[y,[y,\ldots,[y}_{m \text{ times}},x]]\cdots] = \underbrace{\mathrm{ad}_y \circ \cdots \circ \mathrm{ad}_y}_{m \text{ times}}(x)$ and $\mathrm{ad}^0$ is identity. We motivate the derivation for the identities Eqs. (3.2.160) and (3.2.161) in Appendix D.2. Then, we get for the first quantity $\Lambda_I$,

$$\Lambda_I = \sup_{p\geq 0, t\in[0,T]} \|(\partial_t)^p A_I(t)\|^{\frac{1}{p+1}} = \sup_{p\geq 0, t}\left\|\sum_{q\leq p}\binom{p}{q}\mathrm{ad}^q_{\mathrm{i}\lambda\mathsf{P}_c}((\partial_t^{p-q}A(t))_I)\right\|^{1/p+1}. \tag{3.2.162}$$

Moving along, we look time-independent and time-dependent $A$ separately. For the case of $A$ being constant in time,

$$\Lambda_I = \sup_{p\geq 0, t} \lambda^{\frac{p}{p+1}} \|A\|^{\frac{1}{p+1}}. \tag{3.2.163}$$

Then, note from Lemma 3.2.7 that we can take from before that $\lambda = 2\frac{v_{\max}^2\|A\|}{\varepsilon}$ for target error $\varepsilon > 0$. This means that, up to the constant $2^{1-o(1)}$,

$$\Lambda_I \eqsim \sup_{p\geq 0} (v_{\max}^2)^{\frac{p}{p+1}} \|A\|^{\cancel{\frac{p+1}{p+1}}} \frac{1}{\varepsilon^{\frac{p}{p+1}}} = \frac{v_{\max}^{2(1-o(1))}\|A\|.}{\varepsilon^{1-o(1)}} \tag{3.2.164}$$

Next, we look at the situation when $A(t)$ is time-dependent, as treated in Lemma 3.2.10 and Lemma 3.2.11. We start this by simplifying the expression for $\Lambda_I$ further using that the spectral

norm stays invariant under interaction picture transformations and that $\|\mathsf{P}_c\| = 1$.

$$\sup_{p\geq 0,t}\left\|\sum_{q\leq p}\binom{p}{q}\mathrm{ad}^q_{\mathrm{i}\lambda\mathsf{P}_c}((\partial_t^{p-q}A(t))_I)\right\|^{1/p+1} = \sup_{p\geq 0,t}\left\|\sum_{q\leq p}\binom{p}{q}\lambda^q\mathrm{ad}^q_{\mathsf{P}_c}(\partial_t^{p-q}A(t))\right\|^{1/p+1}$$
$$\leq \sup_{p\geq 0,t}\left(\sum_{q\leq p}\binom{p}{q}\lambda^q\left\|(\partial_t^{p-q}A(t))\right\|\right)^{1/p+1} \tag{3.2.165}$$

We continue by first identifying that the spectral norm is upper-bounded by the Frobenius norm and then that Plancherel's theorem holds on matrix-valued functions under the Frobenius norm. Then, we can use that in Fourier-space, derivatives correspond to multiplication and we can introduce another bound with the maximum occurring frequency in $A$ – this is valid as we assume that $A(t)$ is smooth and all its derivatives are bounded.

$$\begin{aligned}\text{Eq. (3.2.165)} &\leq \sup_{p\geq 0,t}\left(\sum_{q\leq p}\binom{p}{q}\lambda^q\left\|(\partial_t^{p-q}A(t))\right\|_F\right)^{1/p+1}\\ &= \sup_{p\geq 0,t}\left(\sum_{q\leq p}\binom{p}{q}\lambda^q\left\|\omega^{p-q}\hat{A}(\omega)\right\|_F\right)^{1/p+1}\\ &\leq \sup_{p\geq 0,t}\left(\sum_{q\leq p}\binom{p}{q}\lambda^q\omega_{\max}^{p-q}\left\|\hat{A}(\omega)\right\|_F\right)^{1/p+1}\\ &\leq \sup_{p\geq 0,t}\|A(t)\|_F^{1/p+1}\left(\sum_{q\leq p}\binom{p}{q}\lambda^q\omega_{\max}^{p-q}\right)^{1/p+1}\end{aligned} \tag{3.2.166}$$

This expression already suggests, aligning with what one would expect, that the smoothness factor $\Lambda_I$ is related to the maximum frequency of the constraint and the original dynamics. By the binomial theorem, we can conclude

$$\Lambda_I \lesssim \sup_{p\geq 0,t}\|A(t)\|_F^{1/p+1}(\lambda+\omega_{\max})^{\frac{p}{p+1}} = \max_{t'}\|A(t')\|_F^{o(1)}(\lambda+\omega_{\max})^{1-o(1)}. \tag{3.2.167}$$

Moreover, by Lemma 3.2.10, we have that we need $\lambda \approx \frac{tv_{\max}^2}{\varepsilon}\max_{t'}\|[\mathsf{P}_c, A(t')]_\sim\| \leq \frac{tv_{\max}^2}{\varepsilon}\max_{t'}\|A(t')\|$. Using this in our expression above,

$$\begin{aligned}\sup_{p\geq 0,t}\|A(t)\|_F^{1/p+1}(\lambda+\omega_{\max})^{\frac{p}{p+1}} &\leq \sup_{p\geq 0,t}\|A(t)\|_F\left(\frac{tv_{\max}^2}{\varepsilon}\|A(t)\| + \frac{\omega_{\max}}{\|A(t)\|}\right)^{\frac{p}{p+1}}\\ &\leq \sup_{p\geq 0,t}\|A(t)\|_F\left(\frac{tv_{\max}^2}{\varepsilon} + \frac{|\mu_{\max}[\mathrm{Im}(A)]|}{|\mu_{\min}[\mathrm{Re}(A)]|}\right)^{\frac{p}{p+1}}\\ &= \|A(t)\|_F\left(\frac{tv_{\max}^2}{\varepsilon} + \frac{|\mu_{\max}[\mathrm{Im}(A)]|}{|\mu_{\min}[\mathrm{Re}(A)]|}\right)^{(1-o(1))}\end{aligned} \tag{3.2.168}$$

With $\mu[\cdot]$, we denote eigenvalues. Thus, the ratio $\frac{|\mu_{\max}[\mathrm{Im}(A(t))]|}{|\mu_{\min}[\mathrm{Re}(A(t))]|}$ describes a notion of oscillation strength versus dissipation and is sometimes also called 'stiffness ratio' [Lam91]. This term is a

consequence of time-dependent simulation, whereas $tv_{\max}^2/\varepsilon$ is due to the additional interaction picture simulation of the constraint.

The remaining term that needs to be discussed is the smoothness parameter due to the inhomogeneous solution, $\Xi_I$,

$$\Xi_I = \sup_{p\geq 0,t} \|(\partial_t)^p b_I(t)\|^{\frac{1}{p+1}} = \sup_{p\geq 0,t} \left\| \sum_{q\leq p} \binom{p}{q} \lambda^q \mathsf{P}_c (\partial_t^{p-q} b(t))_I \right\|^{\frac{1}{p+1}} \tag{3.2.169}$$

Then, we consider the two cases whether $b$ depends on time or not – for both, we have a time-independent generator $A$. If $b(t)$ is constant, then

$$\Xi_I = \sup_{p\geq 0,t} \lambda^{\frac{p}{p+1}} \|b\|_{\ell_2}^{\frac{1}{p+1}} = \sup_{p\geq 0,t} \lambda^{\frac{p}{p+1}} B^{\frac{1}{p+1}}, \tag{3.2.170}$$

where in alignment with Lemma 3.2.8 we introduced $B = \max_{t'} \|b(t')\|_{\ell_2} = \|b\|_{\ell_2}$. Further, we have that Table 3.2a $\lambda = 2\frac{\|A\|}{\varepsilon}(v_{\max}^2 + 2v_{\max}B + B^2)$. We can insert this into Eq. (3.2.170) to obtain, again up to a factor $2^{1-o(1)}$,

$$\Xi_I \simeq \frac{\|A\|^{1-o(1)}}{\varepsilon^{1-o(1)}} \left( v_{\max}^{2(1-o(1))} B^{o(1)} + v_{\max}^{1-o(1)} B + B^{1+o(1)} \right). \tag{3.2.171}$$

Now if we can say that $B$ is within the convex hull of of $v_{\max}$ and $\|A\|$, then we know that $\Xi_I$ is within the convex hull of $O(\frac{\|A\|}{\varepsilon} v_{\max}^2)$ and $O(\frac{\|A\|^2}{\varepsilon} v_{\max}^2)$. That is to say, we can simplify the expression for the complexity this way if the forcing term grows with the size of the initial data or the system matrix. For time-dependent $b(t)$, we can use a similar approach via the binomial theorem as previously for $\Lambda_I$. That is,

$$\begin{aligned}
\sup_{p\geq 0,t} \left\| \sum_{q\leq p} \binom{p}{q} \lambda^q \mathsf{P}_c (\partial_t^{p-q} b(t))_I \right\|^{\frac{1}{p+1}} &\leq \sup_{p\geq 0,t} \left( \sum_{q\leq p} \binom{p}{q} \lambda^q \left\| \partial_t^{p-q} b(t) \right\|_{\ell_2} \right)^{\frac{1}{p+1}} \\
&= \sup_{p\geq 0,t} \left( \sum_{q\leq p} \binom{p}{q} \lambda^q \left\| \omega^{p-q} \hat{b}(\omega) \right\|_{\ell_2} \right)^{\frac{1}{p+1}} \\
&\leq \sup_{p\geq 0,t} B^{1/p+1} \left( \sum_{q\leq p} \binom{p}{q} \lambda^q \omega_{\max}^{p-q} \right)^{\frac{1}{p+1}} \\
&= \sup_{p\geq 0,t} B^{1/p+1} \left( \lambda + \omega_{\max} \right)^{\frac{p}{p+1}}.
\end{aligned} \tag{3.2.172}$$

Now we can again use substitute $\lambda$ with $2\frac{\|A\|}{\varepsilon}(v_{\max}^2 + 2v_{\max}B + B^2)$,

$$\begin{aligned}
&\leq B \sup_{p\geq 0,t} \left( \frac{\|A\|v_{\max}^2}{\varepsilon} \left( \frac{1}{B} + \frac{2}{v_{\max}} + \frac{B}{v_{\max}^2} \right) + \frac{\omega_{\max}}{B} \right)^{p/p+1} \\
&= B \left( \frac{\|A\|v_{\max}^2}{\varepsilon} \left( \frac{1}{B} + \frac{2}{v_{\max}} + \frac{B}{v_{\max}^2} \right) + \frac{\omega_{\max}}{B} \right)^{1-o(1)} \\
&\leq \frac{\|A\|v_{\max}^2}{\varepsilon} \left( 1 + 2\frac{B}{v_{\max}} + \left( \frac{B}{v_{\max}} \right)^2 \right) + \omega_{\max}. \qquad (3.2.173)
\end{aligned}$$

What all of these expressions have in common is the following – similar to the pattern that we have some case-specific term, $G$ times $v_{\max}^2\|A\|/\varepsilon$ for $\lambda$ as showcased in Table 3.2a: Both $\Lambda_I, \Xi_I$ end up roughly linear in $\lambda$ (more precisely, $\lambda^{1-o(1)}$), and sublinearly in $\|A\|$ through $\|A\|^{o(1)}$. Replacing $\lambda$ with expressions from Table 3.2a, we can conclude that it is typically $O(\tilde{G} \cdot \frac{\|A\|v_{\max}^2}{\varepsilon})$, with $\tilde{G}$ a modified case-specific factor as in Table 3.2b.

**Implications for the solution of discretized PDEs with boundary conditions** Typically, when quantifying the cost of numerical methods, we are interested in the effect of the particular choice of discretization given a target error on the final cost ( with a 'finer' discretization meaning more basis functions and thus more accurate representation). In our work, this 'refinement' is expressed by a number of grid points $n^d$. Upon choosing a suitable numerical method, there is also a relationship of approximation error with respect to the discretization. The example below estimates the complexity through the number of discretization points $M'$ for the case of inhomogeneous solutions in Theorem 3.2.15

**Example** (Overhead due to constraint for discrete heat equation)**.** *As an example, we take the three-point finite difference stencil as was used in Section 3.2.3. Then, the approximation error for the second derivative here goes as $O(n^{-d/2})$ [CLO21; KWBA17], which we can take as a rudimentary estimate for the time-evolved error.*

*By Theorem 3.2.15, overhead due to the constraint arises from $M'$, through a gate complexity $O(\log(M'))$, and $M' \in \widetilde{O}\left( \frac{t^3}{\varepsilon}(v_{\max}^2 + B_{L^1}^2)\|A_0\| \left( \log\left( \frac{1+B_{L^1}}{\|v(t)\|_{\ell_2}\varepsilon} \right) \right)^{1+1/\beta} \right)$, by Eq. (3.2.155) and Table 3.2b, where $v_{\max} = \|v(0)\|_{\ell_2}$ and $B_{L^1} = \|b\|_{L^1([0;t])}$. We know that the vector norms grow according to $O(n^{d/2})$ and $B_{L^1} \in O(tn^{d/2})$. Furthermore, the spectral norm of the discrete Laplacian with a three-point stencil follows $\|A_0\| = \|D\mathbf{L}_h\| \in O(n^{2d})$, taking the diffusion coefficient $D$ as a constant. For the discrete Laplacian, this also means that $-\infty < \mu_{\min} \preceq \mathbf{L}_h$, where $-\mu_{\min} \in O(n^{2d})$. By negative-semidefiniteness of $\mathbf{L}_h$, $\|v(t)\|_{\ell_2} \leq v_{\max} + B_{L^1} \in O(tn^{d/2})$. From Appendix D.4, Lemma D.4.1, we have that $\frac{1}{\|v(t)\|_{\ell_2}^2} \leq \mathrm{e}^{-D\mu_{\min}t} \frac{1}{2\|v(0)\|_{\ell_2}\|b\|_{L^1[0;t]}}$.*

*Then, together with $\varepsilon \in O(n^{-d/2})$, we get that*

$$\begin{aligned}
M' &\in \widetilde{O}\left( (t^3 n^{7d/2} + t^4 n^{3d}) \left( d \log\left( \frac{n^2(1+B_{L^1})}{\|v(0)\|_{\ell_2} B_{L^1}} \right) \right)^{1+1/\beta} \right) \\
&\in \widetilde{O}\left( (t^3 n^{7d/2} + t^4 n^{3d}) \left( d \log\left( \frac{n^2}{\|v(0)\|_{\ell_2}} \right) \right)^{1+1/\beta} \right), \qquad (3.2.174)
\end{aligned}$$

*and note that* const. $\lesssim \|v(0)\|_{\ell_2} \lesssim n^{d/2}$*. Finally, the overhead in terms of gate complexity through*

$\log(M')$ *becomes*

$$\widetilde{O}\big(d\log(n)+\log(t)\big), \tag{3.2.175}$$

*if the discretization is chosen to have* $n^d$ *grid-points. Moreover, note that an overhead using the penalty projections is only present for inhomogeneous solutions Eq.* (3.2.155) *when using the quadrature presented in [ACL23]. However, we expect this to be the more likely case in practice, as non-zero boundary conditions expressed through the penalty method via value constraints lead to forcing terms in the ODE following the homogenization techniques outlined in Section 3.2.2. It is not immediately obvious whether introducing ghost points for the non-zero constraint values and using the derivative constraint projections (see Section 3.2.2) would be more efficient here and we leave this for further study.*

### 3.2.5 Conclusion and Outlook

We present a quantum algorithm that uses a penalty projection to enforce constraints such as boundary conditions in evolutionary discretized partial differential equations. Particularly, our approach enables arbitrary constraints and boundary conditions for solving differential equations via the LCHS algorithm. The penalty projections have the advantage that they do not rely on adjustments of the block-encoding of the system matrix, which have the potential to make the algorithm more complicated when compiled to gates. Moreover, our approach is able to tackle a very general class of constraints and boundary conditions, as well as interface conditions. Assuming that the constraint projections are orthogonal, our approach only requires a few steps of fast-forwardable Hamiltonian simulation in addition to the usual ODE solution and interaction picture simulation of the constraint comes at a modest logarithmic overhead in the gate complexity when using the LCHS algorithm [ALL23; ACL23] as ODE solver. The necessary penalty strength goes at most as $O\left(\frac{t^2}{\varepsilon}(\max_{t'}\|v(t')\|_{\ell_2}^2+\|b\|_{L^1}^2)\right)$; for the example of the heat equation, we can show that the final gate complexity overhead grows at most as $\widetilde{O}(d\log n+\log t)$ for $n^d$ degrees of freedom. In a simple scenario of a uniform grid, directly enforcing constraints in the system matrix also attains a simple form for linear-systems based and time marching DE solvers. Yet, the algorithmic consequences, especially for LCHS, are not immediately obvious.

Looking ahead, it would be interesting to work out specific costings involving also constant factors of the presented method as compared to other approaches to enforce constraints presented in the literature. Beyond this, extensions to other constraints that admit decomposition into orthogonal subspaces, such as more general symmetries, seem like a straightforward extension. Less obvious but even more interesting are symmetries with non-local support or situations that lead to non-orthogonal projections, where it is not clear if fast-forwarding in the interaction frame is possible. One could envision this within the framework of scattering theory and build on, e.g., [RM93]. Furthermore, evaluating the performance on numerical examples of higher practical relevance would be an interesting subject of study. In particular, we found that our analytical error bounds were somewhat loose compared to the witnessed error in the simulation results of the heat and wave equations. Hence, one could look into exploring analytical methods that would enable tighter bounds, or perform a more thorough analysis of constant factors involved.

## Code and Data Availability

For the simulations in Section 3.2.3, we used Python with NumPy and SciPy as well as the `findiff` [Bae18] package for finite difference discretization. The code used to generate the results below is available at `https://github.com/philipp-q/q-abs`.

Table 3.2: Complexity-related parameters due to the constraint projection for different sets of assumptions

| **Result** | **Requirement on $\lambda$ to achieve error $\varepsilon > 0$** | **Reformulation as $C\frac{v_{\max}^2\|A_0\|}{\varepsilon} \cdot G$** |
|---|---|---|
| Lemma 3.2.7 | $2\frac{v_{\max}^2\|A_0\|}{\varepsilon}$ | $1$ |
| Lemma 3.2.8 | $2\frac{\|A_0\|}{\varepsilon}\left(v_{\max}^2 + 2v_{\max}tB + t^2B^2\right)$ | $1 + 2\frac{tB}{v_{\max}} + \left(\frac{tB}{v_{\max}}\right)^2$ |
| Lemma 3.2.9 | $(1 + \mathrm{e}^{2\lvert\mu_{\max}(\mathrm{Re}(A_0))\rvert t})\frac{v_{\max}^2\|A_0\|}{\varepsilon}$ | $1 + \mathrm{e}^{2\lvert\mu_{\max}(\mathrm{Re}(A_0))\rvert t}$ |
| Lemma 3.2.10 | $\frac{tv_{\max}^2}{\varepsilon}\max_{0\le t'\le t}\|\{\mathsf{P}_c, A_0(t')\}_\sim\|$ | $t\,\frac{\max_{0\le t'\le t}\left\|\{\mathsf{P}_c, A_0(t')\}_\sim\right\|}{\|A_0\|}$ |
| Lemma 3.2.11 | $\frac{t^2}{2\varepsilon}\left(v_{\max}^2 + 2v_{\max}B_{L^1} + B_{L^1}^2\right)\max_{0\le t'\le t}\|\{\mathsf{P}_c, A_0(t')\}_\sim\|$ | $\frac{t^2}{2}\left(1 + 2\frac{B_{L^1}}{v_{\max}} + \left(\frac{B_{L^1}}{v_{\max}}\right)^2\right)\frac{\max_{0\le t'\le t}\left\|\{\mathsf{P}_c, A_0(t')\}_\sim\right\|}{\|A_0\|}$ |

(a) Asymptotic requirements on penalty factor per class of assumptions. Recall Lemma 3.2.7 assumes $\mathrm{Re}(A_0) \preceq 0, b = 0$, Lemma 3.2.8 has nonzero $b(t)$; Lemma 3.2.9 allows the spectrum of $\mathrm{Re}(A_0)$ to be positive within finite time (with maximum eigenvalue $\mu_{\max}$); Lemma 3.2.10 allows time-dependent $A_0(t)$ but no forcing and Lemma 3.2.11 allows time-dependency in $A_0(t)$ and non-zero $b(t)$.

| | **Smoothness of dynamics $\Lambda_I$** | | **Smoothness of forcing $\Xi_I$** |
|---|---|---|---|
| $A_0$ const.: | $\frac{v_{\max}^2\|A_0\|}{\varepsilon}$ | $b \neq 0$ const.: | $\frac{v_{\max}^2\|A_0\|}{\varepsilon}\left(1 + 2\frac{B}{v_{\max}} + \left(\frac{B}{v_{\max}}\right)^2\right)$ |
| $A_0(t)$: | $\max_{0\le t'\le t}\|A_0(t')\|_F\left(\frac{tv_{\max}^2}{\varepsilon} + \frac{\lvert\mu_{\max}(\mathrm{Im}(A_0(t)))\rvert}{\lvert\mu_{\min}(\mathrm{Re}(A_0(t)))\rvert}\right)$ | $b(t)$: | $\frac{v_{\max}^2\|A_0\|}{\varepsilon}\left(1 + 2\frac{B}{v_{\max}} + \left(\frac{B}{v_{\max}}\right)^2\right) + \omega_{\max}$ |

(b) Asymptotic upper bounds for factors $\Lambda_I, \Xi_I$ used in Theorem 3.2.15, with $\Gamma v_{\max}^2\|A_0\|/\varepsilon = \Lambda_I + \Xi_I$. $\mu_{\max}$ and $\mu_{\min}$ denote the largest/smallest eigenvalues, and $\omega_{\max}$ the highest frequency component (in absolute value) of the forcing term that one aims to represent. Table entries as $O(\cdot)$.

# Chapter 4

# Reflections 💡 and Projections 🔭

In this dissertation, we investigated a set of quantum algorithms and related considerations when solving problems from quantum chemistry and differential equations with quantum computers.

The first half of the thesis is dedicated to problems originating in quantum chemistry.

**Section 2.1**

- 💡 In Section 2.1, we used an evolutionary algorithm to assemble a classically simulatable circuit composed of Clifford gates to split up the circuit coming from a variational quantum eigensolver into subsets that can be run simultaneously on multiple quantum computers. This was quite successful for small models, however, did not scale as well for larger problems.
- 🔭 It would be interesting to see what the rapid developments in generative modelling through large-language models can do here, such as using the recent generative quantum eigensolver [Nak+24]. Furthermore, looking at the performance when extending the Clifford circuits to simulatable non-Clifford circuits, e.g. as in [RLGI20], seems worthwhile.

**Section 2.2** :

- 💡 We considered a state-preparation scheme based on scattering particles originating from easy-to-prepare states into molecular states that can be used for dynamical simulation of chemical processes. Analysis based on mergo-association suggests that as long as the necessary potentials can be carefully engineered, such an approach may admit a finite lower bound on the success probability, which in combination with a weak-measurement scheme for success heralding leads to an efficient, albeit heuristic, process.
- 🔭 One interesting aspect to study here is a more thorough investigation of the physical process at hand, in order to better understand the success of scattering for more general configurations. Furthermore, a detailed cost analysis, supported by numerical simulations, including constant factors would give insight into the feasibility and scalability of the approach.

Beyond quantum chemistry Section 3.1.4 and Section 3.2 studied the solution of more general differential equations.

### Section 3.1

- Building up on prior work on the linearization of non-linear differential equations using Carleman linearization, we carried out analysis of rescaling which is necessary to avoid exponentially vanishing success probability and included higher-order discretizations in space and in time. Part of these considerations was an extended stability analysis considering the impacts of rescaling and the higher-order finite-difference discretizations. Moreover, tighter bounds on Carleman linearization in the case of arbitrary mononomial non-linearities were provided.

- To date, most quantum algorithms that linearize differential equations struggle with stability (as do the classical counterparts). Given one of the motivations to use quantum computers were the hardships in solving unstable dynamics, mild mitigations such as the Lattice-Boltzmann method in [Li+25] could be helpful to maintain a quantum advantage while relaxing the model. Furthermore, specific exact transformations such as in [Zyl+22] may be promising. While the latter may suggest analytical and efficient classical solutions, perhaps there is interesting structure to be found that can be exploited.

### Section 3.2

- Finally, we considered enforcing constraints and boundary conditions in a quantum algorithm for differential equations using perturbation with a projection as penalty, with physical motivations from complex absorbing potentials and the Zeno effect. The advantage here is that there is only a logarithmic overhead in the gate complexity compared to the unconstrained solution. Our analysis is able to cope with general, non-unitary dynamics and forcing terms within value and derivative constraints as long as the resulting projection operation representing them is orthogonal and fast-forwardable. Then, the quantum circuits that are needed for the simulation of the constraint projector are very simple.

- Grounding on the analysis we provide, it would be interesting to know whether our perturbation-theoretical bounds are tight. This could be studied via more extensive numerical experiments. As usual, a constant factor analysis and thorough comparison with other methods that implement boundary conditions by other means can give insight into which methods is more useful in different scenarios. In addition to that, it should prove useful to look into what can be done with situations when there are non-orthogonal constraint projectors or nonlinear underlying dynamics.

## A Look into the Future

Based on the findings in this dissertation, we can identify some directions for future work. In chemistry, improving initial state preparation remains a fundamental bottleneck. Additionally, extending the extensive work by the community on resource estimation of certain algorithms to more algorithms, e.g. the one presented in Section 2.2. Moreover, less works has been done in regards to applications in (partial) differential equations; for applications in fluid dynamics, there is e.g., [Pen+24].

In the beginning of the thesis, the introduction emphasized that one should view quantum algorithms holistically, by including initial state preparation and measurement in the analysis. However, Chapter 3 came short of that and focused on the simulation aspects. Therefore, we propose a closer investigation of instantiations of oracles for initial state preparation of certain problem classes and how to choose observables optimally.

An open problem in the context of solving nonlinear differential equations with a quantum algorithm is the role of dissipation, or rather, whether strongly nonlinear systems are attainable with efficient algorithms. There is recent work that can avoid the dissipativity requirement for Carleman linearization, however though the gap requirements present non-trivial assumptions that deserve further analysis. An open problem in the context of solving nonlinear differential equations with a quantum algorithm is the role of dissipation, or rather, whether strongly nonlinear systems are attainable with efficient algorithms. There is recent work that can avoid the dissipativity requirement for Carleman linearization, however introduced additional non-trivial assumptions on the spacing of the eigenvalues [WWL25].

Quantum Gibb's sampling has emerged as a popular and promising algorithmic tool in the more recent past. There are some more obvious applications for answering questions related to quantum systems and optimization problems. What could be interesting is investigating whether this tool is also useful for differential equations in general.

More generally, the following question is what the author is most interested in: What limitations are there for the simulation of problems from physics on quantum (or classical) computers, from the lens of the physics that is displayed in a given problem.

# Bibliography


[AA17] Yosi Atia and Dorit Aharonov. "Fast-forwarding of Hamiltonians and exponentially precise measurements". In: *Nature communications* 8.1 (2017), p. 1572.

[ACL23] Dong An, Andrew M Childs, and Lin Lin. "Quantum algorithm for linear non-unitary dynamics with near-optimal dependence on all parameters". In: *arXiv preprint arXiv:2312.03916* (2023).

[ADLH05] Alán Aspuru-Guzik, Anthony D. Dutoi, Peter J. Love, and Martin Head-Gordon. "Simulated Quantum Computation of Molecular Energies". In: *Science* 309.5741 (2005), pp. 1704–1707.

[AH22] Pablo Andrés-Martínez and Chris Heunen. "Weakly measured while loops: peeking at quantum states". In: *Quantum Science and Technology* 7.2 (2022), p. 025007.

[AL97] Daniel S. Abrams and Seth Lloyd. "Simulation of Many-Body Fermi Systems on a Universal Quantum Computer". In: *Physical Review Letters* 79.13 (1997), pp. 2586–2589.

[AL99] Daniel S. Abrams and Seth Lloyd. "Quantum algorithm providing exponential speed increase for finding eigenvalues and eigenvectors". In: *Physical Review Letters* 83.24 (1999), pp. 5162–5165.

[Ala+22] Abhijeet Alase, Robert R Nerem, Mohsen Bagherimehrab, Peter Høyer, and Barry C Sanders. "Tight bound for estimating expectation values from a system of linear equations". In: *Physical Review Research* 4.2 (2022), p. 023237.

[ALL23] Dong An, Jin-Peng Liu, and Lin Lin. "Linear combination of Hamiltonian simulation for non-unitary dynamics with optimal state preparation cost". In: *Physical Review Letters* 131 (15 2023), p. 150603.

[ALWZ25] Dong An, Jin-Peng Liu, Daochen Wang, and Qi Zhao. "Quantum Differential Equation Solvers: Limitations and Fast-Forwarding: D. An, J.-P. Liu, D. Wang, Q. Zhao". In: *Communications in Mathematical Physics* 406.8 (2025), p. 189.

[Amb14] Andris Ambainis. "On Physical Problems that are Slightly More Difficult than QMA". In: *2014 IEEE 29th Conference on Computational Complexity (CCC)*. ISSN: 1093-0159. 2014, pp. 32–43.

[Ame14] W.F. Ames. *Numerical Methods for Partial Differential Equations*. Computer Science and Scientific Computing. Elsevier Science, 2014. ISBN: 9780080571300.

[Ana+22] Abhinav Anand*, Philipp Schleich*, Sumner Alperin-Lea*, Phillip WK Jensen*, Sukin Sim, Manuel Díaz-Tinoco, Jakob S Kottmann, Matthias Degroote, Artur F Izmaylov, and Alán Aspuru-Guzik. "A quantum computing view on unitary coupled cluster theory". In: *Chemical Society Reviews* 51.5 (2022), pp. 1659–1684.

[AOY24] Dong An, Akwum Onwunta, and Gengzhi Yang. "Fast-forwarding quantum algorithms for linear dissipative differential equations". In: *arXiv:2410.13189* (2024).

[Aru+19] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G S L Brandao, David A Buell, Brian Burkett, Yu Chen, Zijun Chen, Ben Chiaro, Roberto Collins, William Courtney, Andrew Dunsworth, Edward Farhi, Brooks Foxen, Austin Fowler, Craig Gidney, Marissa Giustina, Rob Graff, Keith Guerin, Steve Habegger, Matthew P Harrigan, Michael J Hartmann, Alan Ho, Markus Hoffmann, Trent Huang, Travis S Humble, Sergei V Isakov, Evan Jeffrey, Zhang Jiang, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Paul V Klimov, Sergey Knysh, Alexander Korotkov, Fedor Kostritsa, David Landhuis, Mike Lindmark, Erik Lucero, Dmitry Lyakh, Salvatore Mandrà, Jarrod R McClean, Matthew McEwen, Anthony Megrant, Xiao Mi, Kristel Michielsen, Masoud Mohseni, Josh Mutus, Ofer Naaman, Matthew Neeley, Charles Neill, Murphy Yuezhen Niu, Eric Ostby, Andre Petukhov, John C Platt, Chris Quintana, Eleanor G Rieffel, Pedram Roushan, Nicholas C Rubin, Daniel Sank, Kevin J Satzinger, Vadim Smelyanskiy, Kevin J Sung, Matthew D Trevithick, Amit Vainsencher, Benjamin Villalonga, Theodore White, Z Jamie Yao, Ping Yeh, Adam Zalcman, Hartmut Neven, and John M Martinis. "Quantum supremacy using a programmable superconducting processor". In: *Nature* 574.7779 (2019), pp. 505–510.

[ASM21] Arash Amini, Qiyu Sun, and Nader Motee. "Error Bounds for Carleman Linearization of General Nonlinear Systems". In: *2021 Proceedings of the Conference on Control and its Applications*. Proceedings. Society for Industrial and Applied Mathematics, 2021, pp. 1–8.

[Ast+23] Ayush Asthana, Ashutosh Kumar, Abraham Vibin, Harper Grimsley, Yu Zhang, Lukasz Cincio, Sergei Tretiak, Pavel A Dub, Sophia E Economou, Edwin Barnes, et al. "Quantum self-consistent equation-of-motion method for computing molecular excitation energies, ionization potentials, and electron affinities on a quantum computer". In: *Chemical Science* (2023).

[AT03] Dorit Aharonov and Amnon Ta-Shma. "Adiabatic quantum state generation and statistical zero knowledge". In: *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*. STOC '03. New York, NY, USA: Association for Computing Machinery, 2003, pp. 20–29. ISBN: 978-1-58113-674-6.

[AT23] Dong An and Konstantina Trivisa. "Quantum algorithms for linear and non-linear fractional reaction-diffusion equations". In: *preprint* (2023).

[AYSA22] Andrea Angulo, Lankun Yang, Eray S. Aydil, and Miguel A. Modestino. "Machine learning enhanced spectroscopic analysis: towards autonomous chemical mixture characterization for rapid process optimization". en. In: *Digital Discovery* 1.1 (2022), pp. 35–44.

[BA24] Mohsen Bagherimehrab and Alán Aspuru-Guzik. “Efficient quantum algorithm for all quantum wavelet transforms”. In: *Quantum Science and Technology* 9.3 (2024), p. 035010.

[Bab+16] Ryan Babbush, Dominic W. Berry, Ian D. Kivlichan, Annie Y. Wei, Peter J. Love, and Alán Aspuru-Guzik. “Exponentially more precise quantum simulation of fermions in second quantization”. en. In: *New Journal of Physics* 18.3 (2016), p. 033032. ISSN: 1367-2630.

[Bab+18a] Ryan Babbush, Craig Gidney, Dominic W Berry, Nathan Wiebe, Jarrod McClean, Alexandru Paler, Austin Fowler, and Hartmut Neven. “Encoding Electronic Spectra in Quantum Circuits with Linear T Complexity”. In: *Physical Review X* 8.4 (2018), p. 041015.

[Bab+18b] Ryan Babbush, Nathan Wiebe, Jarrod McClean, James McClain, Hartmut Neven, and Garnet Kin-Lic Chan. “Low-Depth Quantum Simulation of Materials”. In: *Physical Review X* 8.1 (2018), p. 011044.

[Bab+21] Ryan Babbush, Jarrod R McClean, Michael Newman, Craig Gidney, Sergio Boixo, and Hartmut Neven. “Focus beyond quadratic speedups for error-corrected quantum advantage”. In: *PRX quantum* 2.1 (2021), p. 010103.

[Bab+23] Ryan Babbush, Dominic W Berry, Robin Kothari, Rolando D Somma, and Nathan Wiebe. “Exponential quantum speedup in simulating coupled classical oscillators”. In: *Physical Review X* 13.4 (2023), p. 041041.

[BACS07] Dominic W. Berry, Graeme Ahokas, Richard Cleve, and Barry C. Sanders. “Efficient Quantum Algorithms for Simulating Sparse Hamiltonians”. en. In: *Communications in Mathematical Physics* 270.2 (2007), pp. 359–371. ISSN: 1432-0916.

[Bae18] M. Baer. *findiff Software Package*. `https://github.com/maroba/findiff`. 2018.

[Bag+24] Mohsen Bagherimehrab, Luis Mantilla Calderón, Dominic W. Berry, Philipp Schleich, Mohammad Ghazi Vakili, Abdulrahman Aldossary, Jorge A. Campos Gonzalez Angulo, Christoph Gorgulla, and Alán Aspuru-Guzik. “Faster algorithmic quantum and classical simulations by corrected product formulas”. In: *arXiv preprint arXiv:2409.08265* (2024).

[Bas+22] Lindsay Bassman Oftelie, Katherine Klymko, Diyi Liu, Norm M. Tubman, and Wibe A. de Jong. “Computing Free Energies with Fluctuation Relations on Quantum Computers”. In: *Physical Review Letters* 129.13 (2022), p. 130603.

[BBO20] Xavier Bonet-Monroig, Ryan Babbush, and Thomas E. O’Brien. “Nearly Optimal Measurement Scheduling for Partial Tomography of Quantum States”. In: *Physical Review X* 10.3 (2020).

[BBT06] Sergey Bravyi, Arvid J Bessen, and Barbara M Terhal. “Merlin-Arthur games and stoquastic complexity”. In: *arXiv preprint quant-ph/0611021* (2006).

[BC24] Dominic W Berry and Pedro Costa. “Quantum algorithm for time-dependent differential equations using Dyson series”. In: *Quantum* 8 (2024), p. 1369.

[BCOW17] Dominic W. Berry, Andrew M. Childs, Aaron Ostrander, and Guoming Wang. "Quantum Algorithm for Linear Differential Equations with Exponentially Improved Dependence on Precision". en. In: *Communications in Mathematical Physics* 356.3 (2017), pp. 1057–1081.

[BCW21] Johannes Bausch, Toby S. Cubitt, and James D. Watson. "Uncomputability of phase diagrams". en. In: *Nature Communications* 12.1 (2021), p. 452. ISSN: 2041-1723.

[Ben80] Paul Benioff. "The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines". In: *Journal of Statistical Physics* 22.5 (1980), pp. 563–591. ISSN: 1572-9613.

[Ber+14] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. "Exponential improvement in precision for simulating sparse Hamiltonians". In: *Proceedings of the 46th Annual ACM Symposium on Theory of Computing - STOC '14*. New York, New York, USA: ACM Press, 2014, pp. 283–292. ISBN: 978-1-4503-2710-7.

[Ber+18] Dominic W. Berry, Mária Kieferová, Artur Scherer, Yuval R. Sanders, Guang Hao Low, Nathan Wiebe, Craig Gidney, and Ryan Babbush. "Improved techniques for preparing eigenstates of fermionic Hamiltonians". en. In: *npj Quantum Information* 4.1 (2018), pp. 1–7. ISSN: 2056-6387.

[Ber+19] Dominic W. Berry, Craig Gidney, Mario Motta, Jarrod R. McClean, and Ryan Babbush. "Qubitization of Arbitrary Basis Quantum Chemistry Leveraging Sparsity and Low Rank Factorization". en-GB. In: *Quantum* 3 (2019), p. 208.

[Ber+24] Dominic W Berry, Yu Tong, Tanuj Khattar, Alec White, Tae In Kim, Sergio Boixo, Lin Lin, Seunghoon Lee, Garnet Kin Chan, Ryan Babbush, et al. "Rapid initial state preparation for the quantum simulation of strongly correlated molecules". In: *arXiv preprint arXiv:2409.11748* (2024).

[Ber14] Dominic W. Berry. "High-order quantum algorithm for solving linear differential equations". en. In: *Journal of Physics A: Mathematical and Theoretical* 47.10 (2014), p. 105301.

[BFGY22] Daniel Burgarth, Paolo Facchi, Giovanni Gramegna, and Kazuya Yuasa. "One bound to rule them all: from Adiabatic to Zeno". In: *Quantum* 6 (2022), p. 737.

[BH24] Robert C Bird and Jeremy M Hutson. "Making molecules by mergoassociation: the role of center-of-mass motion". In: *arXiv preprint arXiv:2411.13393* (2024).

[Bha+22] Kishor Bharti, Alba Cervera-Lierta, Thi Ha Kyaw, Tobias Haug, Sumner Alperin-Lea, Abhinav Anand, Matthias Degroote, Hermanni Heimonen, Jakob S Kottmann, Tim Menke, et al. "Noisy intermediate-scale quantum algorithms". In: *Reviews of Modern Physics* 94.1 (2022), p. 015004.

[BHMT02] Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp. "Quantum amplitude amplification and estimation". In: *Contemporary Mathematics* 305 (2002), pp. 53–74.

[Bhu+23] Rahul Bhuyan, Jürgen Mony, Oleg Kotov, Gabriel W Castellanos, Jaime Gómez Rivas, Timur O Shegai, and Karl Börjesson. "The rise and current status of polaritonic photochemistry and photophysics". In: *Chemical Reviews* 123.18 (2023), pp. 10877–10919.

[Bia+17] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. "Quantum machine learning". en. In: *Nature* 549.7671 (2017), pp. 195–202. ISSN: 1476-4687.

[Bia21] Jacob Biamonte. "Universal variational quantum computation". In: *Physical Review A* 103.3 (2021), p. L030401.

[Bis14] Florian A. Bischoff. "Regularizing the molecular potential in electronic structure calculations. II. Many-body methods". In: *The Journal of Chemical Physics* 141.18 (2014), pp. 184106–184106.

[BK05] Sergey Bravyi and Alexei Kitaev. "Universal quantum computation with ideal Clifford gates and noisy ancillas". In: *Physical Review A* 71.2 (2005), p. 022316.

[BLH23] Robert C Bird, C Ruth Le Sueur, and Jeremy M Hutson. "Making molecules by mergoassociation: Two atoms in adjacent nonspherical optical traps". In: *Physical Review Research* 5.4 (2023), p. 043086.

[BMTW22] Hugh GA Burton, Daniel Marti-Dafcik, David P Tew, and David J Wales. "Exact electronic states with shallow quantum circuits through global optimisation". In: *arXiv preprint arXiv:2207.00085* (2022).

[BNWA23] Mohsen Bagherimehrab, Kouhei Nakaji, Nathan Wiebe, and Alán Aspuru-Guzik. "Fast quantum algorithm for differential equations". In: *arXiv preprint arXiv:2306.11802* (2023).

[Bos+24] Jan Lukas Bosse, Andrew M Childs, Charles Derby, Filippo Maria Gambetta, Ashley Montanaro, and Raul A Santos. "Efficient and practical Hamiltonian simulation from time-dependent product formulas". In: *arXiv preprint arXiv:2403.08729* (2024).

[Bro13] Dorje C Brody. "Biorthogonal quantum mechanics". In: *Journal of Physics A: Mathematical and Theoretical* 47.3 (2013), p. 035305.

[BS24] Sachin S Bharadwaj and Katepalli R Sreenivasan. "Compact quantum algorithms for time-dependent differential equations". In: *arXiv preprint arXiv:2405.09767* (2024).

[BS25] Sachin S Bharadwaj and Katepalli R Sreenivasan. "Towards simulating fluid flows with quantum computing". In: *Sādhanā* 50.2 (2025), pp. 1–19.

[BS89] Przemyslaw Bogacki and Lawrence F Shampine. "A 3 (2) pair of Runge-Kutta formulas". In: *Applied Mathematics Letters* 2.4 (1989), pp. 321–325.

[Bur+19] Daniel Burgarth, Paolo Facchi, Hiromichi Nakazato, Saverio Pascazio, and Kazuya Yuasa. "Generalized adiabatic theorem and strong-coupling limits". In: *Quantum* 3 (2019), p. 152.

[But16] J.C. Butcher. *Numerical Methods for Ordinary Differential Equations*. Wiley, 2016. ISBN: 9781119121503.

[BV97] Ethan Bernstein and Umesh Vazirani. "Quantum Complexity Theory". In: *SIAM Journal on Computing* 26.5 (1997), pp. 1411–1473.

[BW25] Noah Brustle and Nathan Wiebe. "Quantum and classical algorithms for nonlinear unitary dynamics". In: *Quantum* 9 (2025), p. 1741.

[CABB23] Pedro Costa, Dong An, Ryan Babbush, and Dominic Berry. "The discrete adiabatic quantum linear system solver has lower constant factors than the randomized adiabatic solver". In: *arXiv preprint arXiv:2312.07690* (2023).

[Cam19] Earl Campbell. "Random Compiler for Fast Hamiltonian Simulation". In: *Physical Review Letters* 123.7 (2019), p. 070503.

[Cao+13] Yudong Cao, Anargyros Papageorgiou, Iasonas Petras, Joseph Traub, and Sabre Kais. "Quantum algorithm and circuit design solving the Poisson equation". en. In: *New Journal of Physics* 15.1 (2013). Publisher: IOP Publishing, p. 013021.

[Cao+19] Yudong Cao, Jonathan Romero, Jonathan P Olson, Matthias Degroote, Peter D Johnson, Mária Kieferová, Ian D Kivlichan, Tim Menke, Borja Peropadre, Nicolas PD Sawaya, et al. "Quantum chemistry in the age of quantum computing". In: *Chemical reviews* 119.19 (2019), pp. 10856–10915.

[Car32] Torsten Carleman. "Application de la théorie des équations intégrales linéaires aux systèmes d'équations différentielles non linéaires". In: *Acta Mathematica* 59 (1932), pp. 63–87.

[Cas00] Piergiorgio Casavecchia. "Chemical reaction dynamics with molecular beams". en. In: *Reports on Progress in Physics* 63.3 (2000), p. 355. ISSN: 0034-4885.

[CDG97] Claude Cohen-Tannoudji, Jacques Dupont-Roc, and Gilbert Grynberg. *Photons and Atoms: Introduction to Quantum Electrodynamics*. 1st ed. Wiley, 1997.

[Cer+21] Marco Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, et al. "Variational quantum algorithms". In: *Nature Reviews Physics* 3.9 (2021), pp. 625–644.

[Cer+22] M. Cerezo, Guillaume Verdon, Hsin-Yuan Huang, Lukasz Cincio, and Patrick J. Coles. "Challenges and opportunities in quantum machine learning". en. In: *Nature Computational Science* 2.9 (2022), pp. 567–576. ISSN: 2662-8457.

[Cha+23] Hans Hon Sang Chan, Richard Meister, Tyson Jones, David P. Tew, and Simon C. Benjamin. "Grid-based methods for chemistry simulations on a quantum computer". In: *Science Advances* 9.9 (2023), eabo7484.

[Che+22] MH Cheng, KE Khosla, CN Self, M Lin, BX Li, AC Medina, and MS Kim. "Clifford circuit initialisation for variational quantum algorithms". In: *arXiv preprint arXiv:2207.01539* (2022).

[Chi+21] Andrew M Childs, Yuan Su, Minh C Tran, Nathan Wiebe, and Shuchen Zhu. "Theory of trotter error with commutator scaling". In: *Physical Review X* 11.1 (2021), p. 011020.

[CHPZ24] Chi-Fang Chen, Hsin-Yuan Huang, John Preskill, and Leo Zhou. "Local minima in quantum systems". In: *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*. 2024, pp. 1323–1330.

[CJO19] Pedro CS Costa, Stephen Jordan, and Aaron Ostrander. "Quantum algorithm for simulating the wave equation". In: *Physical Review A* 99.1 (2019), p. 012323.

[CJS13] B. D. Clader, B. C. Jacobs, and C. R. Sprouse. "Preconditioned Quantum Linear System Algorithm". In: *Physical Review Letters* 110.25 (2013), p. 250504. ISSN: 1079-7114.

[CKBG23] Chi-Fang Chen, Michael J Kastoryano, Fernando GSL Brandão, and András Gilyén. "Quantum thermal state preparation". In: *arXiv preprint arXiv:2303.18224* (2023).

[CKS12] B. Cockburn, G.E. Karniadakis, and C.W. Shu. *Discontinuous Galerkin Methods: Theory, Computation and Applications*. Lecture Notes in Computational Science and Engineering. Springer Berlin Heidelberg, 2012. ISBN: 9783642597213.

[CL20] Andrew M. Childs and Jin-Peng Liu. "Quantum Spectral Methods for Differential Equations". In: *Communications in Mathematical Physics* 375.2 (2020), pp. 1427–1457.

[CLO21] Andrew M. Childs, Jin-Peng Liu, and Aaron Ostrander. "High-precision quantum algorithms for partial differential equations". In: *Quantum* 5 (2021), p. 574.

[CLRS22] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2022.

[CM18] Basile F. E. Curchod and Todd J. Martínez. "Ab Initio Nonadiabatic Quantum Molecular Dynamics". In: *Chemical Reviews* 118.7 (2018), pp. 3305–3336. ISSN: 0009-2665.

[Cos+22] Pedro CS Costa, Dong An, Yuval R Sanders, Yuan Su, Ryan Babbush, and Dominic W Berry. "Optimal Scaling Quantum Linear-Systems Solver via Discrete Adiabatic Theorem". In: *PRX Quantum* 3.4 (2022), p. 040303.

[CSMB23] Pedro Costa, Philipp Schleich, Mauro ES Morales, and Dominic W Berry. "Further improving quantum algorithms for nonlinear differential equations via higher-order methods and rescaling". In: *arXiv preprint arXiv:2312.09518* (2023).

[CSW25] Ahmet Burak Catli, Sophia Simon, and Nathan Wiebe. "Exponentially Better Bounds for Quantum Optimization via Dynamical Simulation". In: *arXiv preprint arXiv:2502.04285* (2025).

[CW17] Richard Cleve and Chunhao Wang. "Efficient Quantum Algorithms for Simulating Lindblad Evolution". In: *44th International Colloquium on Automata, Languages, and Programming (ICALP 2017)*. Schloss-Dagstuhl-Leibniz Zentrum für Informatik. 2017.

[CZ15] Song Cen and Qun Zhang. *Multiphysics Modeling: Numerical Methods and Engineering Applications*. Academic Press, 2015.

[Dal+23] Alexander M Dalzell, Sam McArdle, Mario Berta, Przemyslaw Bienias, Chi-Fang Chen, András Gilyén, Connor T Hann, Michael J Kastoryano, Emil T Khabiboulline, Aleksander Kubica, et al. "Quantum algorithms: A survey of applications and end-to-end complexities". In: *arXiv preprint arXiv:2310.03011* (2023).

[DCL24] Zhiyan Ding, Chi-Fang Chen, and Lin Lin. "Single-ancilla ground state preparation via Lindbladians". In: *Physical Review Research* 6.3 (2024), p. 033147.

[Deu85] David Deutsch. "Quantum theory, the Church–Turing principle and the universal quantum computer". In: *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* 400.1818 (1985), pp. 97–117.

[DGB25] Xiaohan Dan, Eitan Geva, and Victor S Batista. "Simulating Non-Markovian Quantum Dynamics on NISQ Computers Using the Hierarchical Equations of Motion". In: *Journal of Chemical Theory and Computation* (2025).

[Dia+23] NL Diaz, Paolo Braccia, Martin Larocca, JM Matera, R Rossignoli, and M Cerezo. "Parallel-in-time quantum simulation via Page and Wootters quantum time". In: *arXiv preprint arXiv:2308.12944* (2023).

[Dir26] Paul Adrien Maurice Dirac. "On the theory of quantum mechanics". In: *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* 112.762 (1926), pp. 661–677.

[DJ92] David Deutsch and Richard Jozsa. "Rapid solution of problems by quantum computation". In: *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences* 439.1907 (1992), pp. 553–558.

[DJSW25] Zhiyan Ding, Marius Junge, Philipp Schleich, and Peixue Wu. "Lower bound for simulation cost of open quantum systems: Lipschitz continuity approach". In: *Communications in Mathematical Physics* 406.3 (2025), p. 60.

[DKO97] W. Dahmen, A. Kurdila, and P. Oswald. *Multiscale Wavelet Methods for Partial Differential Equations*. ISSN. Elsevier Science, 1997. ISBN: 9780080537146.

[DR07] Philip J Davis and Philip Rabinowitz. *Methods of numerical integration*. Courier Corporation, 2007.

[DTM90] G Dattoli, A Torre, and R Mignani. "Non-Hermitian evolution of two-level quantum systems". In: *Physical Review A* 42.3 (1990), p. 1467.

[EBY12] G. Evans, J. Blackledge, and P. Yardley. *Numerical Methods for Partial Differential Equations*. Springer Undergraduate Mathematics Series. Springer London, 2012. ISBN: 9781447103776.

[Edd+22] Andrew Eddins, Mario Motta, Tanvi P Gujarati, Sergey Bravyi, Antonio Mezzacapo, Charles Hadfield, and Sarah Sheldon. "Doubling the size of quantum simulators by entanglement forging". In: *PRX Quantum* 3.1 (2022), p. 010309.

[FD19] Andrew Fagan and Ross Duncan. "Optimising Clifford Circuits with Quantomatic". In: *Electronic Proceedings in Theoretical Computer Science* 287 (2019), pp. 85–105.

[Fey82] Richard P. Feynman. "Simulating physics with computers". In: *International Journal of Theoretical Physics* 21.6-7 (1982), pp. 467–488.

[Fey85] Richard P. Feynman. "Quantum Mechanical Computers". EN. In: *Optics News* 11.2 (1985), pp. 11–20.

[FLT23] Di Fang, Lin Lin, and Yu Tong. "Time-marching based quantum solvers for time-dependent linear differential equations". In: *Quantum* 7 (2023), p. 955.

[Foc30] V. Fock. "Näherungsmethode zur Lösung des quantenmechanischen Mehrkörperproblems". In: *Zeitschrift für Physik* 61.1-2 (1930), pp. 126–148.

[FP17] Marcelo Forets and Amaury Pouly. "Explicit error bounds for Carleman linearization". In: *arXiv preprint arXiv:1711.02552* (2017).

[FV62] David G. Feingold and Richard S. Varga. "Block diagonally dominant matrices and generalizations of the Gerschgorin circle theorem". In: *Pacific Journal of Mathematics* 12 (1962), pp. 1241–1250.

[GA98] Karl Gustafson and Takehisa Abe. "The third boundary condition–was it Robin's?" In: *Mathematical Intelligencer* 20.1 (1998).

[GEBM19] Harper R Grimsley, Sophia E Economou, Edwin Barnes, and Nicholas J Mayhall. "An adaptive variational algorithm for exact molecular simulations on a quantum computer". In: *Nat. commun.* 10.1 (2019), pp. 1–9.

[GH22] Kevin T Geier and Philipp Hauke. "From non-Hermitian linear response to dynamical correlations and fluctuation-dissipation relations in quantum many-body systems". In: *PRX Quantum* 3.3 (2022), p. 030308.

[Gid18] Craig Gidney. "Halving the cost of quantum addition". In: *Quantum* 2 (2018), p. 74.

[Got98a] Daniel Gottesman. "The Heisenberg representation of quantum computers". In: *arXiv preprint quant-ph/9807006* (1998).

[Got98b] Daniel Gottesman. "Theory of fault-tolerant quantum computation". In: *Physical Review A* 57.1 (1998), p. 127.

[Gro98] Lov K Grover. "How fast can a quantum computer search?" In: *arXiv preprint quant-ph/9809029* (1998).

[GRZ20] Uma Girish, Ran Raz, and Wei Zhan. "Lower bounds for XOR of forrelations". In: *arXiv preprint arXiv:2007.03631* (2020).

[GSLW19] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. "Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics". In: *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*. 2019, pp. 193–204.

[GSŞ21] Shouzhen Gu, Rolando D Somma, and Burak Şahinoğlu. "Fast-forwarding quantum evolution". In: *Quantum* 5 (2021), p. 577.

[Har+04] Robert J. Harrison, George I. Fann, Takeshi Yanai, Zhengting Gan, and Gregory Beylkin. "Multiresolution quantum chemistry: Basic theory and initial applications". In: *Journal of Chemical Physics* 121.23 (2004), pp. 11587–11598.

[Har+16] Robert J Harrison, Gregory Beylkin, Florian A Bischoff, Justus A Calvin, George I Fann, Jacob Fosso-Tande, Diego Galindo, Jeff R Hammond, Rebecca Hartman-Baker, Judith C Hill, et al. "MADNESS: A multiresolution, adaptive numerical environment for scientific simulation". In: *SIAM Journal on Scientific Computing* 38.5 (2016), S123–S142.

[Har28] D. R. Hartree. "The Wave Mechanics of an Atom with a Non-Coulomb Central Field Part I Theory and Methods". In: *Mathematical Proceedings of the Cambridge Philosophical Society* 24.1 (1928), pp. 89–110.

[Hei26] Werner Heisenberg. "Mehrkörperproblem und Resonanz in der Quantenmechanik". In: *Zeitschrift für Physik* 38.6 (1926), pp. 411–426.

[HHL09] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. "Quantum Algorithm for Linear Systems of Equations". In: *Physical Review Letters* 103 (15 2009), p. 150502.

[HJ12] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge University Press, 2012.

[HJO00] Trygve Helgaker, Poul Jørgensen, and Jeppe Olsen. *Molecular Electronic-Structure Theory*. Publication Title: Molecular Electronic-Structure Theory. Chichester, UK: John Wiley & Sons, Ltd, 2000. ISBN: 978-1-119-01957-2.

[HLP34] Godfrey H Hardy, John E Littlewood, and George Pólya. *Inequalities (Cambridge mathematical library)*. Cambridge university press, 1934.

[HLSW24] William J Huggins, Oskar Leimkuhler, Torin F Stetina, and K Birgitta Whaley. "Efficient state preparation for the quantum simulation of molecules in first quantization". In: *arXiv preprint arXiv:2407.00249* (2024).

[Hoo91] William Graham Hoover. *Computational Statistical Mechanics*. 1st ed. Elsevier, 1991.

[Hua+22] Hsin-Yuan Huang, Michael Broughton, Jordan Cotler, Sitan Chen, Jerry Li, Masoud Mohseni, Hartmut Neven, Ryan Babbush, Richard Kueng, John Preskill, and Jarrod R. McClean. "Quantum advantage in learning from experiments". In: *Science* 376.6598 (2022), pp. 1182–1186.

[Hug+21] William J Huggins, Jarrod R McClean, Nicholas C Rubin, Zhang Jiang, Nathan Wiebe, K Birgitta Whaley, and Ryan Babbush. "Efficient and noise resilient measurements for quantum chemistry on near-term quantum computers". In: *npj Quantum Information* 7.1 (2021), pp. 1–9.

[Hug+22] William J. Huggins, Bryan A. O'Gorman, Nicholas C. Rubin, David R. Reichman, Ryan Babbush, and Joonho Lee. "Unbiasing fermionic quantum Monte Carlo with a quantum computer". en. In: *Nature* 603.7901 (2022). Number: 7901 Publisher: Nature Publishing Group, pp. 416–420. ISSN: 1476-4687.

[HW23] Matthew Hagan and Nathan Wiebe. "Composite quantum simulations". In: *Quantum* 7 (2023), p. 1181.

[HW25] Matthew Hagan and Nathan Wiebe. "The Thermodynamic Cost of Ignorance: Thermal State Preparation with One Ancilla Qubit". In: *arXiv preprint arXiv:2502.03410* (2025).

[HWB19] Oscar Higgott, Daochen Wang, and Stephen Brierley. "Variational quantum computation of excited states". In: *Quantum* 3 (2019), p. 156.

[ILY21] Artur F Izmaylov, Robert A Lang, and Tzu-Ching Yen. "Analytic gradients in variational quantum algorithms: Algebraic extensions of the parameter-shift rule to general unitary transformations". In: *Physical Review A* 104.6 (2021), p. 062443.

[IYLV19] Artur F Izmaylov, Tzu-Ching Yen, Robert A Lang, and Vladyslav Verteletskyi. "Unitary partitioning approach to the measurement problem in the variational quantum eigensolver method". In: *Journal of chemical theory and computation* 16.1 (2019), pp. 190–195.

[Jac+22] Daniel M Jackson, Urs Haeusler, Leon Zaporski, Jonathan H Bodey, Noah Shofer, Edmund Clarke, Maxime Hugues, Mete Atatüre, Claire Le Gall, and Dorian A Gangloff. "Optimal purification of a spin ensemble by quantum-algorithmic feedback". In: *Physical Review X* 12.3 (2022), p. 031014.

[JI24] Jiaqing Jiang and Sandy Irani. "Quantum Metropolis sampling via weak measurement". In: *arXiv preprint arXiv:2406.16023* (2024).

[JL24] Shi Jin and Nana Liu. "Quantum algorithms for nonlinear partial differential equations". In: *Bulletin des Sciences Mathématiques* 194 (2024), p. 103457.

[JLLY24a] Shi Jin, Xiantao Li, Nana Liu, and Yue Yu. "Quantum simulation for partial differential equations with physical boundary or interface conditions". In: *Journal of Computational Physics* 498 (2024), p. 112707.

[JLLY24b] Shi Jin, Xiantao Li, Nana Liu, and Yue Yu. "Quantum simulation for quantum dynamics with artificial boundary conditions". In: *SIAM Journal on Scientific Computing* 46.4 (2024), B403–B421.

[JLM24] Shi Jin, Nana Liu, and Chuwen Ma. "Quantum simulation of Maxwell's equations via Schrödingerisation". In: *ESAIM: Mathematical Modelling and Numerical Analysis* 58.5 (2024), pp. 1853–1879.

[JLMY25] Shi Jin, Nana Liu, Chuwen Ma, and Yue Yu. "On the Schr\" odingerization method for linear non-unitary dynamics with optimal dependence on matrix queries". In: *arXiv preprint arXiv:2505.00370* (2025).

[JLY23a] Shi Jin, Nana Liu, and Yue Yu. "Quantum simulation of partial differential equations: Applications and detailed analysis". In: *Physical Review A* 108.3 (2023), p. 032603.

[JLY23b] Shi Jin, Nana Liu, and Yue Yu. "Time complexity analysis of quantum algorithms via linear representations for nonlinear ordinary and partial differential equations". In: *Journal of Computational Physics* 487 (2023), p. 112149.

[JLY24] Shi Jin, Nana Liu, and Yue Yu. "Quantum Simulation of Partial Differential Equations via Schrödingerization". In: *Physical Review Letters* 133.23 (2024), p. 230602.

[JN13] Richard Jozsa and Maarten Van den Nest. "Classical simulation complexity of extended Clifford circuits". In: *arXiv preprint arXiv:1305.6190* (2013).

[Jou+21] Joonyoung F. Joung, Minhi Han, Jinhyo Hwang, Minseok Jeong, Dong Hoon Choi, and Sungnam Park. "Deep Learning Optical Spectroscopy Based on Experimental Database: Potential Applications to Molecular Design". In: *JACS Au* 1.4 (2021), pp. 427–438.

[KA22] Jakob S. Kottmann and Alán Aspuru-Guzik. "Optimized low-depth quantum circuits for molecular electronic structure using a separable-pair approximation". In: *Phys. Rev. A* 105 (3 2022), p. 032449.

[KAA21] Jakob S Kottmann, Abhinav Anand, and Alán Aspuru-Guzik. "A feasible approach for automatically differentiable unitary coupled-cluster on quantum computers". In: *Chemical Science* 12.10 (2021), pp. 3497–3508.

[Kan+17] Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M Chow, and Jay M Gambetta. "Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets". In: *Nature* 549.7671 (2017), pp. 242–246.

[Kas+08] Ivan Kassal, Stephen P. Jordan, Peter J. Love, Masoud Mohseni, and Alán Aspuru-Guzik. "Polynomial-time quantum algorithm for the simulation of chemical dynamics". In: *Proceedings of the National Academy of Sciences* 105.48 (2008), pp. 18681–18686.

[KBV20] Jakob S Kottmann, Florian A Bischoff, and Edward F Valeev. "Direct determination of optimal pair-natural orbitals in a real-space representation: the second-order Moller–Plesset energy". In: *The Journal of Chemical Physics* 152.7 (2020), p. 074105.

[KE21] Oleksandr Kyriienko and Vincent E Elfving. "Generalized quantum circuit differentiation rules". In: *Physical Review A* 104.5 (2021), p. 052417.

[Kha+24] Tyler Kharazi, Ahmad M Alkadri, Jin-Peng Liu, Kranthi K Mandadapu, and K Birgitta Whaley. "Explicit block encodings of boundary value problems for many-body elliptic operators". In: *arXiv preprint arXiv:2407.18347* (2024).

[Kit95] A Yu Kitaev. "Quantum measurements and the Abelian stabilizer problem". In: *arXiv preprint quant-ph/9511026* (1995).

[Kiv+18] Ian D. Kivlichan, Jarrod McClean, Nathan Wiebe, Craig Gidney, Alán Aspuru-Guzik, Garnet Kin-Lic Chan, and Ryan Babbush. "Quantum Simulation of Electronic Structure with Linear Depth and Connectivity". In: *Physical Review Letters* 120.11 (2018), p. 110501.

[Kja+20] Morten Kjaergaard, Mollie E Schwartz, Jochen Braumüller, Philip Krantz, Joel I-J Wang, Simon Gustavsson, and William D Oliver. "Superconducting qubits: Current state of play". In: *Annual Review of Condensed Matter Physics* 11.1 (2020), pp. 369–395.

[KKR05] Julia Kempe, Alexei Kitaev, and Oded Regev. "The Complexity of the Local Hamiltonian Problem". In: *FSTTCS 2004: Foundations of Software Technology and Theoretical Computer Science*. Ed. by Kamal Lodaya and Meena Mahajan. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2005, pp. 372–383. ISBN: 978-3-540-30538-5.

[KLFK24] Efekan Kökcü, Heba A Labib, JK Freericks, and Alexander F Kemper. "A linear response framework for quantum simulation of bosonic and fermionic correlation functions". In: *Nature Communications* 15.1 (2024), p. 3881.

[Kot+21] Jakob S Kottmann, Sumner Alperin-Lea, Teresa Tamayo-Mendoza, Alba Cervera-Lierta, Cyrille Lavigne, Tzu-Ching Yen, Vladyslav Verteletskyi, Philipp Schleich, Abhinav Anand, Matthias Degroote, et al. "Tequila: A platform for rapid development of quantum algorithms". In: *Quantum Science and Technology* 6.2 (2021), p. 024009.

[Kot22] Jakob S Kottmann. "Molecular Quantum Circuit Design: A Graph-Based Approach". In: *arXiv preprint arXiv:2207.12421* (2022).

[Kro23] Hari Krovi. "Improved quantum algorithms for linear and nonlinear differential equations". In: *Quantum* 7 (2023), p. 913.

[KSTA21] Jakob S Kottmann, Philipp Schleich, Teresa Tamayo-Mendoza, and Alán Aspuru-Guzik. "Reducing qubit requirements while maintaining numerical precision for the variational quantum eigensolver: A basis-set-free approach". In: *The Journal of Physical Chemistry Letters* 12.1 (2021), pp. 663–673.

[Kub57] Ryogo Kubo. "Statistical-mechanical theory of irreversible processes. I. General theory and simple applications to magnetic and conduction problems". In: *Journal of the physical society of Japan* 12.6 (1957), pp. 570–586.

[KWBA17] Ian D Kivlichan, Nathan Wiebe, Ryan Babbush, and Alán Aspuru-Guzik. "Bounding the costs of quantum simulation of many-body physics in real space". In: *Journal of Physics A: Mathematical and Theoretical* 50.30 (2017), p. 305301.

[Lam91] John Denholm Lambert. *Numerical methods for ordinary differential systems: the initial value problem*. John Wiley & Sons, Inc., 1991.

[LC16] Guang Hao Low and Isaac L. Chuang. "Hamiltonian Simulation by Qubitization". In: (2016).

[LC19] Guang Hao Low and Isaac L. Chuang. "Hamiltonian Simulation by Qubitization". en-GB. In: *Quantum* 3 (2019), p. 163.

[LCG24] Zhenning Liu, Andrew M Childs, and Daniel Gottesman. "Low-depth quantum symmetrization". In: *arXiv preprint arXiv:2411.04019* (2024).

[Lee+23] Seunghoon Lee, Joonho Lee, Huanchen Zhai, Yu Tong, Alexander M. Dalzell, Ashutosh Kumar, Phillip Helms, Johnnie Gray, Zhi-Hao Cui, Wenyuan Liu, Michael Kastoryano, Ryan Babbush, John Preskill, David R. Reichman, Earl T. Campbell, Edward F. Valeev, Lin Lin, and Garnet Kin-Lic Chan. "Evaluating the evidence for exponential quantum advantage in ground-state quantum chemistry". In: *Nature Communications* 14.1 (2023), p. 1952. ISSN: 2041-1723.

[LENS24] Dylan Lewis, Stephan Eidenbenz, Balasubramanya Nadiga, and Yiğit Subaşi. "Limitations for quantum algorithms to solve turbulent and chaotic systems". In: *Quantum* 8 (2024), p. 1509.

[LGI22] Robert A Lang, Aadithya Ganeshram, and Artur F Izmaylov. "Growth reduction of similarity transformed electronic Hamiltonians in qubit space". In: *arXiv preprint arXiv:2210.03875* (2022).

[LHHW18] Joonho Lee, William J Huggins, Martin Head-Gordon, and K Birgitta Whaley. "Generalized unitary coupled cluster wave functions for quantum computation". In: *Journal of Chemical Theory and Computation* 15.1 (2018), pp. 311–324.

[Li+25] Xiangyu Li, Xiaolong Yin, Nathan Wiebe, Jaehun Chun, Gregory K Schenter, Margaret S Cheung, and Johannes Mülmenstädt. "Potential quantum advantage for simulation of fluid dynamics". In: *Physical Review Research* 7.1 (2025), p. 013036.

[Li05] Jianping Li. "General explicit difference formulas for numerical differentiation". In: *Journal of Computational and Applied Mathematics* 183.1 (2005), pp. 29–52.

[Lin+22] Yen Ting Lin, Robert B Lowrie, Denis Aslangil, Yiğit Subaşi, and Andrew T Sornborger. "Challenges for quantum computation of nonlinear dynamical systems using linear representations". In: *arXiv preprint arXiv:2202.02188* (2022).

[Lin76] Goran Lindblad. "On the generators of quantum dynamical semigroups". In: *Communications in mathematical physics* 48 (1976), pp. 119–130.

[Liu+18] L. R. Liu, J. D. Hood, Y. Yu, J. T. Zhang, N. R. Hutzler, T. Rosenband, and K.-K. Ni. "Building one molecule from a reservoir of two atoms". In: *Science* 360.6391 (2018), pp. 900–903.

[Liu+21] Jin-Peng Liu, Herman Øie Kolden, Hari K Krovi, Nuno F Loureiro, Konstantina Trivisa, and Andrew M Childs. "Efficient quantum algorithm for dissipative nonlinear differential equations". In: *Proceedings of the National Academy of Sciences* 118.35 (2021), e2026805118.

[Liu+22] Hongbin Liu, Guang Hao Low, Damian S. Steiger, Thomas Häner, Markus Reiher, and Matthias Troyer. "Prospects of quantum computing for molecular sciences". In: *Materials Theory* 6.1 (2022), p. 11. ISSN: 2509-8012.

[Liu+23] Jin-Peng Liu, Dong An, Di Fang, Jiasu Wang, Guang Hao Low, and Stephen Jordan. "Efficient Quantum Algorithm for Nonlinear Reaction–Diffusion Equations and Energy Estimation". In: *Communications in Mathematical Physics* 404.2 (2023), pp. 963–1020.

[LLLL25] Rundi Lu, Hao-En Li, Zhengwei Liu, and Jin-Peng Liu. "Infinite-dimensional Extension of the Linear Combination of Hamiltonian Simulation: Theorems and Applications". In: *arXiv preprint arXiv:2502.19688* (2025).

[Llo+20] Seth Lloyd, Giacomo De Palma, Can Gokler, Bobak Kiani, Zi-Wen Liu, Milad Marvian, Felix Tennie, and Tim Palmer. "Quantum algorithm for nonlinear differential equations". In: *arXiv:2011.06571* (2020).

[Llo96] Seth Lloyd. "Universal quantum simulators". In: *Science* 273.5278 (1996), pp. 1073–1078.

[LLWL24] Zhenning Liu, Xiantao Li, Chunhao Wang, and Jin-Peng Liu. "Toward end-to-end quantum simulation for protein dynamics". In: *arXiv preprint arXiv:2411.03972* (2024).

[LMS22a] Noah Linden, Ashley Montanaro, and Changpeng Shao. "Quantum vs. Classical Algorithms for Solving the Heat Equation". In: *Communications in Mathematical Physics* 395.2 (2022), pp. 601–641. ISSN: 1432-0916.

[LMS22b] Noah Linden, Ashley Montanaro, and Changpeng Shao. "Quantum vs. classical algorithms for solving the heat equation". In: *Communications in Mathematical Physics* 395.2 (2022), pp. 601–641.

[LO08] Sarah K Leyton and Tobias J Osborne. "A quantum algorithm to solve nonlinear differential equations". In: *arXiv:0812.4423* (2008).

[Löw58] Per-Olov Löwdin. "Correlation Problem in Many-Electron Quantum Mechanics I. Review of Different Approaches and Discussion of Some Current Ideas". In: *Advances in Chemical Physics*. John Wiley & Sons, Ltd, 1958, pp. 207–322.

[Löw95] Per-Olov Löwdin. "The historical development of the electron correlation problem". de. In: *International Journal of Quantum Chemistry* 55.2 (1995), pp. 77–102.

[LP17] David A Levin and Yuval Peres. *Markov chains and mixing times*. Vol. 107. American Mathematical Soc., 2017.

[LRI20] Robert A Lang, Ilya G Ryabinkin, and Artur F Izmaylov. "Unitary transformation of the electronic hamiltonian with an exact quadratic truncation of the baker-campbell-hausdorff expansion". In: *Journal of Chemical Theory and Computation* 17.1 (2020), pp. 66–78.

[LT22] Lin Lin and Yu Tong. "Heisenberg-Limited Ground-State Energy Estimation for Early Fault-Tolerant Quantum Computers". In: *PRX Quantum* 3.1 (2022), p. 010318.

[Lun11] AP Lund. "Efficient quantum computing with weak measurements". In: *New Journal of Physics* 13.5 (2011), p. 053024.

[LW18] Guang Hao Low and Nathan Wiebe. "Hamiltonian simulation in the interaction picture". In: *arXiv preprint arXiv:1805.00675* (2018).

[LYLY16] David G Luenberger, Yinyu Ye, David G Luenberger, and Yinyu Ye. "Penalty and barrier methods". In: *Linear and Nonlinear Programming* (2016), pp. 397–428.

[LZL24] Hao-En Li, Yongtao Zhan, and Lin Lin. "Dissipative ground state preparation in ab initio electronic structure theory". In: *arXiv preprint arXiv:2411.01470* (2024).

[Man80] Yuri Manin. "Computable and uncomputable". In: *Sovetskoye Radio, Moscow* 128 (1980), p. 28.

[MBM08] Sebastiaan Y. T. van de Meerakker, Hendrick L. Bethlem, and Gerard Meijer. "Taming molecular beams". en. In: *Nature Physics* 4.8 (2008), pp. 595–602. ISSN: 1745-2481.

[McA+20] Sam McArdle, Suguru Endo, Alán Aspuru-Guzik, Simon C Benjamin, and Xiao Yuan. "Quantum computational chemistry". In: *Reviews of Modern Physics* 92.1 (2020), p. 015003.

[McC+20a] Jarrod R McClean, Fabian M Faulstich, Qinyi Zhu, Bryan O'Gorman, Yiheng Qiu, Steven R White, Ryan Babbush, and Lin Lin. "Discontinuous Galerkin discretization for quantum simulation of chemistry". In: *New Journal of Physics* 22.9 (2020), p. 093015.

[McC+20b] Jarrod R McClean, Nicholas C Rubin, Kevin J Sung, Ian D Kivlichan, Xavier Bonet-Monroig, Yudong Cao, Chengyu Dai, E Schuyler Fried, Craig Gidney, Brendan Gimby, et al. "OpenFermion: the electronic structure package for quantum computers". In: *Quantum Science and Technology* 5.3 (2020), p. 034014.

[Mit+22] Kosuke Mitarai, Yasunari Suzuki, Wataru Mizukami, Yuya O Nakagawa, and Keisuke Fujii. "Quadratic Clifford expansion for efficient benchmarking and initialization of variational quantum algorithms". In: *Physical Review Research* 4.3 (2022), p. 033012.

[Miz+20] Wataru Mizukami, Kosuke Mitarai, Yuya O. Nakagawa, Takahiro Yamamoto, Tennin Yan, and Yu-ya Ohnishi. "Orbital optimized unitary coupled cluster theory for quantum computer". In: *Phys. Rev. Research* 2 (3 2020), p. 033421.

[Miz09] Ari Mizel. "Critically damped quantum search". In: *Physical review letters* 102.15 (2009), p. 150501.

[MKCD17] Jarrod R McClean, Mollie E Kimchi-Schwartz, Jonathan Carter, and Wibe A De Jong. "Hybrid quantum-classical hierarchy for mitigation of decoherence and determination of excited states". In: *Physical Review A* 95.4 (2017), p. 042308.

[Mon16] Ashley Montanaro. "Quantum algorithms: An overview". In: *npj Quantum Information* 2.1 (2016), pp. 1–17.

[Mor+24] Mauro ES Morales, Lirandë Pira, Philipp Schleich, Kelvin Koor, Pedro Costa, Dong An, Alán Aspuru-Guzik, Lin Lin, Patrick Rebentrost, and Dominic W Berry. "Quantum linear system solvers: A survey of algorithms and applications". In: *arXiv preprint arXiv:2411.02522* (2024).

[Mot+24] Danial Motlagh, Robert A Lang, Jorge A Campos-Gonzalez-Angulo, Tao Zeng, Alan Aspuru-Guzik, and Juan Miguel Arrazola. "Quantum Algorithm for Vibronic Dynamics: Case Study on Singlet Fission Solar Cell Design". In: *arXiv preprint arXiv:2411.13669* (2024).

[MP02] Justin E. Molloy and Miles J. Padgett. "Lights, action: Optical tweezers". In: *Contemporary Physics* 43.4 (2002), pp. 241–258. ISSN: 0010-7514.

[MPNE04] JG Muga, JP Palao, B Navarro, and IL Egusquiza. "Complex absorbing potentials". In: *Physics Reports* 395.6 (2004), pp. 357–426.

[MS72] J Mehra and ECG Sudarshan. "Some reflections on the nature of entropy, irreversibility and the second law of thermodynamics". In: *Il Nuovo Cimento B (1971-1996)* 11.2 (1972), pp. 215–256.

[MW24] Danial Motlagh and Nathan Wiebe. "Generalized quantum signal processing". In: *PRX Quantum* 5.2 (2024), p. 020368.

[MZLG24] Mariane Mangin-Brinet, Jing Zhang, Denis Lacroix, and Edgar Andres Ruiz Guzman. "Efficient solution of the non-unitary time-dependent schrodinger equation on a quantum computer with complex absorbing potential". In: *Quantum* 8 (2024), p. 1311.

[MZLR24] Mariane Mangin-Brinet, Jing Zhang, Denis Lacroix, and Edgar Andres Ruiz Guzman. "Efficient solution of the non-unitary time-dependent Schrodinger equation on a quantum computer with complex absorbing potential". In: *Quantum* 8 (2024), p. 1311.

[Nak+24] Kouhei Nakaji, Lasse Bjørn Kristensen, Jorge A Campos-Gonzalez-Angulo, Mohammad Ghazi Vakili, Haozhe Huang, Mohsen Bagherimehrab, Christoph Gorgulla, FuTe Wong, Alex McCaskey, Jin-Sung Kim, et al. "The generative quantum eigensolver (GQE) and its application for ground state search". In: *arXiv preprint arXiv:2401.09253* (2024).

[NBA24] Kouhei Nakaji, Mohsen Bagherimehrab, and Alán Aspuru-Guzik. "High-Order Randomized Compiler for Hamiltonian Simulation". In: *PRX Quantum* 5.2 (2024), p. 020330.

[NDS23] I. Novikau, I.Y. Dodin, and E.A. Startsev. "Simulation of Linear Non-Hermitian Boundary-Value Problems with Quantum Singular-Value Transformation". In: *Physical Review Applied* 19.5 (2023).

[NJ25] Ivan Novikau and Ilon Joseph. "Quantum algorithm for the advection-diffusion equation and the Koopman-von Neumann approach to nonlinear dynamical systems". In: *Computer Physics Communications* 309 (2025), p. 109498.

[NMF19] Ken M Nakanishi, Kosuke Mitarai, and Keisuke Fujii. "Subspace-search variational quantum eigensolver for excited states". In: *Physical Review Research* 1.3 (2019), p. 033062.

[NN00] P. A. Maia Neto and H. M. Nussenzveig. "Theory of optical tweezers". en. In: *Europhysics Letters* 50.5 (2000), p. 702. ISSN: 0295-5075.

[Nod90] Isao Noda. "Two-Dimensional Infrared (2D IR) Spectroscopy: Theory and Applications". In: *Applied Spectroscopy* 44.4 (1990), pp. 550–561. ISSN: 0003-7028.

[OBRT20] Pauline J Ollitrault, Alberto Baiardi, Markus Reiher, and Ivano Tavernelli. "Hardware efficient quantum algorithms for vibrational structure calculations". In: *Chemical science* 11.26 (2020), pp. 6842–6855.

[OIWF22] Bryan O'Gorman, Sandy Irani, James Whitfield, and Bill Fefferman. "Intractability of Electronic Structure in a Fixed Basis". In: *PRX Quantum* 3.2 (2022), p. 020322.

[OMT20] Pauline J. Ollitrault, Guglielmo Mazzola, and Ivano Tavernelli. "Nonadiabatic Molecular Quantum Dynamics with Quantum Computers". en. In: *Physical Review Letters* 125.26 (2020), p. 260511. ISSN: 0031-9007, 1079-7114.

[OPN09] Maylis Orio, Dimitrios A Pantazis, and Frank Neese. "Density functional theory". In: *Photosynthesis research* 102 (2009), pp. 443–453.

[Orú14] Román Orús. "A practical introduction to tensor networks: Matrix product states and projected entangled pair states". In: *Annals of physics* 349 (2014), pp. 117–158.

[PCCZ20] Lei Pan, Xin Chen, Yu Chen, and Hui Zhai. "Non-Hermitian linear response theory". In: *Nature Physics* 16.7 (2020), pp. 767–771.

[Ped+14] J. S. Pedernales, R. Di Candia, I. L. Egusquiza, J. Casanova, and E. Solano. "Efficient Quantum Algorithm for Computing n-time Correlation Functions". In: *Physical Review Letters* 113.2 (2014), p. 020505. ISSN: 0031-9007, 1079-7114.

[Pen+24] John Penuel, Amara Katabarwa, Peter D Johnson, Collin Farquhar, Yudong Cao, and Michael C Garrett. "Feasibility of accelerating incompressible computational fluid dynamics simulations with fault-tolerant quantum computers". In: *arXiv preprint arXiv:2406.06323* (2024).

[Per+14] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J Love, Alán Aspuru-Guzik, and Jeremy L O'brien. "A variational eigenvalue solver on a photonic quantum processor". In: *Nature communications* 5.1 (2014), pp. 1–7.

[Poc+24] Matthew Pocrnic, Matthew Hagan, Juan Carrasquilla, Dvira Segal, and Nathan Wiebe. "Composite Qdrift-product formulas for quantum and classical simulations in real and imaginary time". In: *Physical Review Research* 6.1 (2024), p. 013224.

[Pre18] John Preskill. "Quantum computing in the NISQ era and beyond". In: *Quantum* 2 (2018), p. 79.

[PSW23] Matthew Pocrnic, Dvira Segal, and Nathan Wiebe. "Quantum simulation of lindbladian dynamics via repeated interactions". In: *arXiv preprint arXiv:2312.05371* (2023).

[Rav+22] Gokul Subramanian Ravi, Pranav Gokhale, Yi Ding, William M Kirby, Kaitlin N Smith, Jonathan M Baker, Peter J Love, Henry Hoffmann, Kenneth R Brown, and Frederic T Chong. "CAFQA: Clifford Ansatz For Quantum Accuracy". In: *arXiv preprint arXiv:2202.12924* (2022).

[Rei+17] Markus Reiher, Nathan Wiebe, Krysta M. Svore, Dave Wecker, and Matthias Troyer. "Elucidating reaction mechanisms on quantum computers". In: *Proceedings of the National Academy of Sciences* 114.29 (2017), pp. 7555–7560.

[RLGI20] Ilya G Ryabinkin, Robert A Lang, Scott N Genin, and Artur F Izmaylov. "Iterative Qubit Coupled Cluster approach with efficient screening of generators". In: *J. Chem. Theory Comput.* 16.2 (2020), pp. 1055–1063.

[RLP22] Jonas Richter, Oliver Lunt, and Arijeet Pal. "Transport and entanglement growth in long-range random Clifford circuits". In: *arXiv preprint arXiv:2205.06309* (2022).

[RM93] UV Riss and H-D Meyer. "Calculation of resonance energies and widths using the complex absorbing potential method". In: *Journal of Physics B: Atomic, Molecular and Optical Physics* 26.23 (1993), p. 4503.

[Rof19] Joschka Roffe. "Quantum error correction: an introductory guide". In: *Contemporary Physics* 60.3 (2019), pp. 226–245.

[RRW22] Abhishek Rajput, Alessandro Roggero, and Nathan Wiebe. "Hybridized Methods for Quantum Simulation in the Interaction Picture". en-GB. In: *Quantum* 6 (2022), p. 780.

[Rut+23] Daniel K Ruttley, Alexander Guttridge, Stefan Spence, Robert C Bird, C Ruth Le Sueur, Jeremy M Hutson, and Simon L Cornish. "Formation of ultracold molecules by merging optical tweezers". In: *Physical Review Letters* 130.22 (2023), p. 223401.

[RYGI18] Ilya G Ryabinkin, Tzu-Ching Yen, Scott N Genin, and Artur F Izmaylov. "Qubit coupled cluster method: a systematic approach to quantum chemistry on a quantum computer". In: *J. Chem. Theory Comput.* 14.12 (2018), pp. 6317–6326.

[Sch+19] Maria Schuld, Ville Bergholm, Christian Gogolin, Josh Izaac, and Nathan Killoran. "Evaluating analytic gradients on quantum hardware". In: *Physical Review A* 99.3 (2019), p. 032331.

[Sch+23] Philipp Schleich, Joseph Boen, Lukasz Cincio, Abhinav Anand, Jakob S Kottmann, Sergei Tretiak, Pavel A Dub, and Alán Aspuru-Guzik. "Partitioning quantum chemistry simulations with clifford circuits". In: *Journal of Chemical Theory and Computation* 19.15 (2023), pp. 4952–4964.

[Sch+24a] Philipp Schleich, Lasse Bjørn Kristensen, Jorge A Angulo Campos, Abdulrahman Aldossary, Davide Avagliano, Mohsen Bagherimehrab, Christoph Gorgulla, Joe Fitzsimons, and Alán Aspuru-Guzik. "Chemically Motivated Simulation Problems are Efficiently Solvable by a Quantum Computer". In: *arXiv preprint arXiv:2401.09268* (2024).

[Sch+24b] Philipp Schleich*, Marta Skreta*, Lasse Bjørn Kristensen, Rodrigo Vargas-Hernandez, and Alan Aspuru-Guzik. "Quantum Deep Equilibrium Models". In: *Advances in Neural Information Processing Systems*. Vol. 37. 2024, pp. 31940–31967.

[Sch+25a] Philipp Schleich, Tyler Kharazi, Xiangyu Li, Jin-Peng Liu, Alán Aspuru-Guzik, and Nathan Wiebe. "Arbitrary Boundary Conditions and Constraints in Quantum Algorithms for Differential Equations via Penalty Projections". In: *arXiv preprint arXiv:2506.21751* (2025).

[Sch+25b] Philipp Schleich, Luis Mantilla Calderón, Chong Sun, Mohsen Bagherimehrab, Abdulrahman Aldossary, Jakob S. Kottmann, and Alán Aspuru-Guzik. *Quantum Computing for Quantum Chemistry*. American Chemical Society, 2025.

[Sch05] Ulrich Schollwöck. "The density-matrix renormalization group". In: *Reviews of modern physics* 77.1 (2005), pp. 259–315.

[SD00] Francis Sullivan and Jack Dongarra. "Guest Editors' Introduction: The Top 10 Algorithms". In: *Computing in Science & Engineering* 2.1 (2000), pp. 0022–23.

[SDM22] Doru Sticlet, Balázs Dóra, and Cătălin Paşcu Moca. "Kubo formula for non-Hermitian systems and tachyon optical conductivity". In: *Physical Review Letters* 128.1 (2022), p. 016802.

[SGAZ24] Zhong-Xia Shang, Naixu Guo, Dong An, and Qi Zhao. "Design nearly optimal quantum algorithm for linear differential equations via Lindbladians". In: *arXiv preprint arXiv:2410.19628* (2024).

[Sha+21] Zhong-Xia Shang, Ming-Cheng Chen, Xiao Yuan, Chao-Yang Lu, and Jian-Wei Pan. "Schrödinger-Heisenberg Variational Quantum Algorithms". In: *arXiv preprint arXiv:2112.07881* (2021).

[Sho94] Peter W Shor. "Algorithms for quantum computation: discrete logarithms and factoring". In: *Proceedings 35th annual symposium on foundations of computer science*. IEEE. 1994, pp. 124–134.

[Sho95] Peter W Shor. "Scheme for reducing decoherence in quantum computer memory". In: *Physical review A* 52.4 (1995), R2493.

[Sim+24] Sophia Simon, Raffaele Santagati, Matthias Degroote, Nikolaj Moll, Michael Streif, and Nathan Wiebe. "Improved precision scaling for simulating coupled quantum-classical dynamics". In: *PRX Quantum* 5.1 (2024), p. 010343.

[Sim97] Daniel R Simon. "On the power of quantum computation". In: *SIAM journal on computing* 26.5 (1997), pp. 1474–1483.

[SISS23] Sauro Succi, Wael Itani, Katepalli Sreenivasan, and René Steijl. "Quantum computing for fluids: Where do we stand?" In: *Europhysics Letters* 144.1 (2023), p. 10001.

[SJA19] Sukin Sim, Peter D Johnson, and Alán Aspuru-Guzik. "Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms". In: *Advanced Quantum Technologies* 2.12 (2019), p. 1900070.

[SKA22] Philipp Schleich, Jakob S Kottmann, and Alán Aspuru-Guzik. "Improving the accuracy of the variational quantum eigensolver for molecular systems by the explicitly-correlated perturbative [2] R12-correction". In: *Physical Chemistry Chemical Physics* 24.22 (2022), pp. 13550–13564.

[Sla29] J. C. Slater. "The theory of complex spectra". In: *Physical Review* 34.10 (1929), pp. 1293–1322.

[Sla30a] J. C. Slater. "Atomic shielding constants". In: *Physical Review* 36.1 (1930), pp. 57–64.

[Sla30b] J. C. Slater. "Note on hartree's method [5]". In: *Physical Review* 35.2 (1930), pp. 210–211.

[SLSB19] Yuval R Sanders, Guang Hao Low, Artur Scherer, and Dominic W Berry. "Black-Box Quantum State Preparation without Arithmetic". In: *Physical Review Letters* 122.2 (2019), p. 020502.

[Smi+20] Daniel GA Smith, Lori A Burns, Andrew C Simmonett, Robert M Parrish, Matthew C Schieber, Raimondas Galvelis, Peter Kraus, Holger Kruse, Roberto Di Remigio, Asem Alenaizan, et al. "PSI4 1.4: Open-source software for high-throughput quantum chemistry". In: *The Journal of chemical physics* 152.18 (2020), p. 184108.

[SN06] Magnus Svärd and Jan Nordström. "On the order of accuracy for difference approximations of initial-boundary value problems". In: *Journal of Computational Physics* 218.1 (2006), pp. 333–352.

[SO67] Attila Szabo and Neil S. Ostlund. *Modern Quantum Chemistry – Introduction to Advanced Electronic Structure Theory*. Vol. 35. Issue: 11 Pages: 1097. Dover Publications, 1967.

[Sok+20] Igor O Sokolov, Panagiotis Kl Barkoutsos, Pauline J Ollitrault, Donny Greenberg, Julia Rice, Marco Pistoia, and Ivano Tavernelli. "Quantum orbital-optimized unitary coupled cluster methods in the strongly correlated regime: Can quantum algorithms outperform their classical equivalents?" In: *The Journal of chemical physics* 152.12 (2020), p. 124107.

[SS86] Youcef Saad and Martin H Schultz. "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems". In: *SIAM Journal on scientific and statistical computing* 7.3 (1986), pp. 856–869.

[Su+21] Yuan Su, Dominic W. Berry, Nathan Wiebe, Nicholas Rubin, and Ryan Babbush. "Fault-Tolerant Quantum Simulations of Chemistry in First Quantization". In: *PRX Quantum* 2.4 (2021), p. 040332.

[Sun+20] Qiming Sun, Xing Zhang, Samragni Banerjee, Peng Bao, Marc Barbry, Nick S Blunt, Nikolay A Bogdanov, George H Booth, Jia Chen, Zhi-Hao Cui, et al. "Recent developments in the PySCF program package". In: *The Journal of chemical physics* 153.2 (2020).

[Sun+24] Zhu Sun, Gregory Boyd, Zhenyu Cai, Hamza Jnane, Balint Koczor, Richard Meister, Romy Minko, Benjamin Pring, Simon C Benjamin, and Nikitas Stamatopoulos. "Low Depth Phase Oracle Using a Parallel Piecewise Circuit". In: *arXiv preprint arXiv:2409.04587* (2024).

[Suz+21] Yasunari Suzuki, Yoshiaki Kawase, Yuya Masumura, Yuria Hiraga, Masahiro Nakadai, Jiabao Chen, Ken M Nakanishi, Kosuke Mitarai, Ryosuke Imai, Shiro Tamiya, et al. "Qulacs: a fast and versatile quantum circuit simulator for research purpose". In: *Quantum* 5 (2021), p. 559.

[SV09] Norbert Schuch and Frank Verstraete. "Computational complexity of interacting electrons and fundamental limitations of density functional theory". In: *Nature Physics* 5.10 (2009), pp. 732–735. ISSN: 1745-2481.

[T H09] Neil T. Hunt. "2D-IR spectroscopy : ultrafast insights into biomolecule structure and function". en. In: *Chemical Society Reviews* 38.7 (2009), pp. 1837–1848.

[Tak+12] Tetsu Takekoshi, Markus Debatin, Raffael Rameshan, Francesca Ferlaino, Rudolf Grimm, Hanns-Christoph Nägerl, C Ruth Le Sueur, Jeremy M Hutson, Paul S Julienne, Svetlana Kotochigova, et al. "Towards the production of ultracold ground-state RbCs molecules: Feshbach resonances, weakly bound states, and the coupled-channel model". In: *Physical Review A—Atomic, Molecular, and Optical Physics* 85.3 (2012), p. 032506.

[Tan+21] Ho Lun Tang, V.O. Shkolnikov, George S. Barron, Harper R. Grimsley, Nicholas J. Mayhall, Edwin Barnes, and Sophia E. Economou. "Qubit-ADAPT-VQE: An Adaptive Algorithm for Constructing Hardware-Efficient Ansätze on a Quantum Processor". In: *PRX Quantum* 2 (2 2021), p. 020310.

[Tan07] David J. Tannor. *Introduction to Quantum Mechanics*. University Science Books, 2007. ISBN: 978-1-891389-23-8.

[Tay16] Cameron R. Taylor. *Finite Difference Coefficients Calculator*. `https://web.media.mit.edu/~crtaylor/calculator.html`. 2016.

[TCCG20] Francesco Tacchino, Alessandro Chiesa, Stefano Carretta, and Dario Gerace. "Quantum Computers as Universal Quantum Simulators: State-of-the-Art and Perspectives". In: *Advanced Quantum Technologies* 3.3 (2020), p. 1900052. ISSN: 2511-9044.

[Til+22] Jules Tilly, Hongxiang Chen, Shuxiang Cao, Dario Picozzi, Kanav Setia, Ying Li, Edward Grant, Leonard Wossnig, Ivan Rungger, George H Booth, et al. "The variational quantum eigensolver: a review of methods and best practices". In: *Physics Reports* 986 (2022), pp. 1–128.

[Tka+21] Nikolay V Tkachenko, James Sud, Yu Zhang, Sergei Tretiak, Petr M Anisimov, Andrew T Arrasmith, Patrick J Coles, Lukasz Cincio, and Pavel A Dub. "Correlation-Informed Permutation of Qubits for Reducing Ansatz Depth in the Variational Quantum Eigensolver". In: *PRX Quantum* 2.2 (2021), p. 020337.

[Tol18] J Tolar. "On Clifford groups in quantum computing". In: *Journal of Physics: Conference Series*. Vol. 1071. IOP Publishing. 2018, p. 012022.

[Tsy03] Semyon Victor Tsynkov. "Artificial boundary conditions for the numerical simulation of unsteady acoustic waves". In: *Journal of Computational Physics* 189.2 (2003), pp. 626–650.

[Van75] A Van der Sluis. "Perturbations of eigenvalues of non-normal matrices". In: *Communications of the ACM* 18.1 (1975), pp. 30–36.

[van79] A. van der Sluis. "Gershgorin domains for partitioned matrices". In: *Linear Algebra and its Applications* 26 (1979), pp. 265–280.

[Vit+15] Valerio Vitale, Jacek Dziedzic, Simon M.-M. Dubois, Hans Fangohr, and Chris-Kriton Skylaris. "Anharmonic Infrared Spectroscopy through the Fourier Transform of Time Correlation Function Formalism in ONETEP". In: *Journal of Chemical Theory and Computation* 11.7 (2015), pp. 3321–3332. ISSN: 1549-9618.

[VYI20] Vladyslav Verteletskyi, Tzu-Ching Yen, and Artur F Izmaylov. "Measurement optimization in the variational quantum eigensolver using a minimum clique cover". In: *The Journal of chemical physics* 152.12 (2020), p. 124114.

[Wan+14] Lee-Ping Wang, Alexey Titov, Robert McGibbon, Fang Liu, Vijay S. Pande, and Todd J. Martínez. "Discovering chemistry with an ab initio nanoreactor". en. In: *Nature Chemistry* 6.12 (2014), pp. 1044–1048. ISSN: 1755-4349.

[WB21] James D Watson and Johannes Bausch. "The complexity of approximating critical points of quantum phase transitions". In: *arXiv preprint arXiv:2105.13350* (2021).

[WBA11] James D. Whitfield, Jacob Biamonte, and Alán Aspuru-Guzik. "Simulation of electronic structure Hamiltonians using quantum computers". In: *Molecular Physics* 109.5 (2011), pp. 735–750. ISSN: 0026-8976.

[Wie96] Stephen Wiesner. "Simulations of many-body quantum systems by a quantum computer". In: *arXiv preprint quant-ph/9603028* (1996).

[Wit05] Curt Wittig. "The landau- zener formula". In: *The Journal of Physical Chemistry B* 109.17 (2005), pp. 8428–8430.

[WIWL22] David Wierichs, Josh Izaac, Cody Wang, and Cedric Yen-Yu Lin. "General parameter-shift rules for quantum gradients". In: *Quantum* 6 (2022), p. 677.

[WLA12] James Daniel Whitfield, Peter John Love, and Alán Aspuru-Guzik. "Computational complexity in electronic structure". en. In: *Physical Chemistry Chemical Physics* 15.2 (2012), pp. 397–411. ISSN: 1463-9084.

[Wu+21] Xiaoling Wu, Xinhui Liang, Yaoqi Tian, Fan Yang, Cheng Chen, Yong-Chun Liu, Meng Khoon Tey, and Li You. "A concise review of Rydberg atom based quantum computation and quantum simulation". In: *Chinese Physics B* 30.2 (2021), p. 020305.

[WWL25] Hsuan-Cheng Wu, Jingyao Wang, and Xiantao Li. "Quantum algorithms for nonlinear dynamics: Revisiting carleman linearization with no dissipative conditions". In: *SIAM Journal on Scientific Computing* 47.2 (2025), A943–A970.

[XK20] Rongxin Xia and Sabre Kais. "Qubit coupled cluster singles and doubles variational quantum eigensolver ansatz for electronic structure calculations". In: *Quantum Science and Technology* 6.1 (2020), p. 015001.

[XWG21] Cheng Xue, Yu-Chun Wu, and Guo-Ping Guo. "Quantum homotopy perturbation method for nonlinear dissipative ordinary differential equations". In: *New Journal of Physics* 23.12 (2021), p. 123035.

[XXWG22] Cheng Xue, Xiao-Fan Xu, Yu-Chun Wu, and Guo-Ping Guo. "Quantum algorithm for solving a quadratic nonlinear system of equations". In: *Physical Review A* 106.3 (2022), p. 032427.

[Yal+21] Saad Yalouz, Bruno Senjean, Jakob Günther, Francesco Buda, Thomas E O'Brien, and Lucas Visscher. "A state-averaged orbital-optimized hybrid quantum–classical algorithm for a democratic description of ground and excited states". In: *Quantum Science and Technology* 6.2 (2021), p. 024004.

[Yan+22] Bao Yan, Shijie Wei, Haocong Jiang, Hong Wang, Qianheng Duan, Zhi Ma, and Gui-Lu Long. "Fixed-point oblivious quantum amplitude-amplification algorithm". In: *Scientific Reports* 12.1 (2022), p. 14339.

[YI21] Tzu-Ching Yen and Artur F Izmaylov. "Cartan Subalgebra Approach to Efficient Measurements of Quantum Observables". In: *PRX Quantum* 2.4 (2021), p. 040320.

[YVI20] Tzu-Ching Yen, Vladyslav Verteletskyi, and Artur F Izmaylov. "Measuring all compatible operators in one series of single-qubit measurements using unitary transformations". In: *Journal of chemical theory and computation* 16.4 (2020), pp. 2400–2409.

[Zal98] Christof Zalka. "Simulating quantum systems on a quantum computer". In: *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 454.1969 (1998), pp. 313–322.

[ZC22] Xing Zhang and Garnet Kin-Lic Chan. "Differentiable quantum chemistry with PySCF for molecules and materials at the mean-field level and beyond". In: *The Journal of Chemical Physics* 157.20 (2022), p. 204801.

[Zha+21] Shi-Xin Zhang, Zhou-Quan Wan, Chang-Yu Hsieh, Hong Yao, and Shengyu Zhang. "Variational quantum-neural hybrid error mitigation". In: *arXiv preprint arXiv:2112.10380* (2021).

[Zha+22] Yu Zhang, Lukasz Cincio, Christian FA Negre, Piotr Czarnik, Patrick J Coles, Petr M Anisimov, Susan M Mniszewski, Sergei Tretiak, and Pavel A Dub. "Variational quantum eigensolver with reduced circuit complexity". In: *npj Quantum Information* 8.1 (2022), p. 96.

[ZLLW24] Yufan Zheng, Jiaqi Leng, Yizhou Liu, and Xiaodi Wu. "On the Computational Complexity of Schrödinger Operators". In: *arXiv preprint arXiv:2411.05120* (2024).

[ZLY22] Xiao-Ming Zhang, Tongyang Li, and Xiao Yuan. "Quantum state preparation with optimal circuit depth: Implementations and applications". In: *Physical Review Letters* 129.23 (2022), p. 230504.

[ZY24] Xiao-Ming Zhang and Xiao Yuan. "Circuit complexity of quantum access models for encoding classical data". In: *npj Quantum Information* 10.1 (2024), p. 42.

[Zyl+22] Julien Zylberman, Giuseppe Di Molfetta, Marc Brachet, Nuno F Loureiro, and Fabrice Debbasch. "Quantum simulations of hydrodynamics via the Madelung transformation". In: *Physical Review A* 106.3 (2022), p. 032408.

# Appendix A

# Supplementary Material for Section 2.1

## A.1 Product-state enforcing reference method

Given a fermionic Hamiltonian $H_{\text{ferm}}$, we use a quantum-chemical encoding [Cao+19; McA+20] to obtain a qubit Hamiltonian $H_{\text{qc}}$ as a linear combination of Pauli strings. Our code for the following method is availabe at `https://github.com/philipp-q/power_method_for_product_states`.

Then, we use a generic wavefunction as a tensor over the qubit space, assuming a product state. For the sake of clarity, we will only show the case of two clusters and note that this can be arbitrarily extended. Then, for $N_{\text{q}}$ qubits, this gives $|\psi\rangle \in \mathbb{C}^{2^{N_{\text{q}}/2}} \otimes \mathbb{C}^{2^{N_{\text{q}}/2}}$. We use a random, initial wavefunction factorized over subsystems $A, B$ with uniformly sampled and normalized coefficients in a Pauli basis.

Further, we define the reduced Hamiltonians of the subsystems $A, B$ as

$$H_A = \text{tr}_B(H) = \langle(\cdot)\psi_B|H|(\cdot)\psi_B\rangle \tag{A.1.1}$$

$$H_B = \text{tr}_A(H) = \langle\psi_A(\cdot)|H|\psi_A(\cdot)\rangle\,, \tag{A.1.2}$$

where $H_A$ is obtained by tracing out all subsystems except for $A$; see also Fig. A.1.

To find the ground-state, we perform the following power method-type iterations using the reduced Hamiltonians for each systems, where $|\psi^{(i)}\rangle = |\psi_A^{(i)}\rangle \otimes |\psi_B^{(i)}\rangle$:

$$|\psi_A^{(i+1)}\rangle = H_A\,|\psi_A^{(i)}\rangle - \gamma\,|\psi_A^{(i)}\rangle \tag{A.1.3}$$

$$|\psi_B^{(i+1)}\rangle = H_B\,|\psi_B^{(i)}\rangle - \gamma\,|\psi_B^{(i)}\rangle\,. \tag{A.1.4}$$

For our computations, we use $\gamma = 1$ and use a convergence criterion of

$$|E^{(i)} - E^{(i+1)}| \leq \text{TOL} \quad \text{where} \quad E^{(i)} = \langle\psi_A^{(i)}\psi_B^{(i)}|H|\psi_A^{(i)}\psi_B^{(i)}\rangle, \tag{A.1.5}$$

with $\text{TOL} = 10^{-5}$ milli-Hartree. We note that we keep the wavevectors as full vectors. For computationally more involved experiments than presented in this work, one could think of a representation using tensor network methods [Orú14].

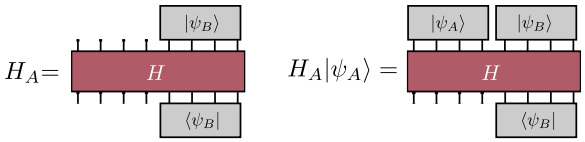



Figure A.1: Hamiltonian that has been reduced over subspace $B$ and now acts only on subspace $A$.

## A.2 Parameter choice for genetic algorithm

Here, we show an exemplary plot used for the hyperparameter search for the number of populations and offsprings per population for the geometric algorithm used in our procedure, as mentioned in the simulation setup in Section 2.1.3. As we can see in Fig. A.2, the sensitivity to the parameters is around the order of $10^{-4}$ milli-Hartrees, which justifies our choice of 10–15 populations for 8–10 offsprings.

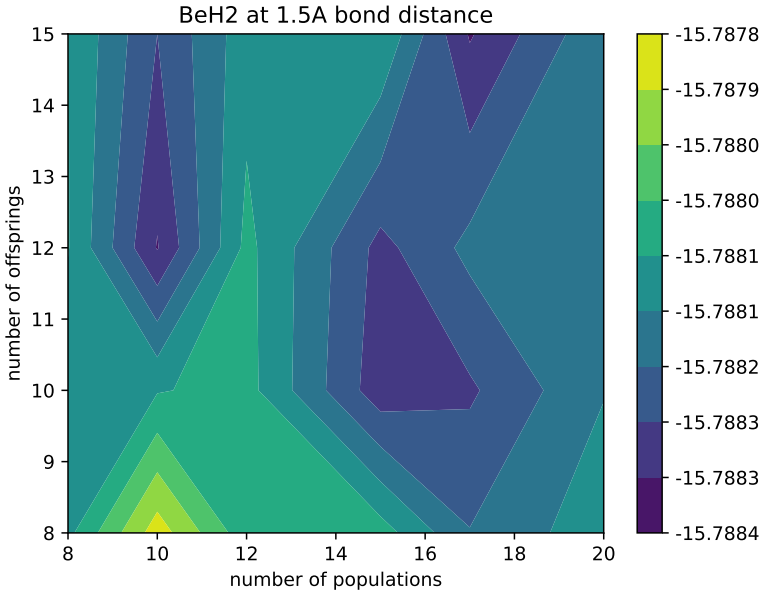



Figure A.2: Parameter search for setup of genetic algorithm, exemplarity for $BeH_2$ at 1.5 Angstrom bond distance. The color gradient depicts the obtained energy in Hartree.

## A.3 Fidelities before and after optimization

Here, we provide fidelities between the FCI state obtained from exact diagonalization of the Hamiltonian with the product-state reference method in Appendix A.1, optimization with Clifford gates and the SPA state [KA22].

We carried out the calculation exemplarily for $BeH_2$ in three geometries and show the fidelities in Fig. A.3 and associated energies in Table A.1. In the case of degenerate eigenspaces, we sum up individual overlaps for all degenerate states. At a bond distance of 1.5 Angstrom, all methods show similar performance, while SPA is the most cost-effective solution thanks to the very short circuit depth in the hardcore-boson encoding [KA22]. For 3.0 Angstrom, the factorized states in "Reference" and SPA perform similarly. The addition of a optimized Clifford circuit led to improvements, which for this system size were small but may accumulate for larger systems. It is not clear here whether the addition of Clifford circuits would pay off in a cost-benefit analysis. Looking at even larger dissociation at a bond distance of 4.5 Angstrom with a triplet ground-state and a near-degenerate singlet state, the situation is different. Since the PNOs are a bad orbital choice here (construction via MP2), standard SPA does not perform well. It is also restricted from penetrating into the triplet space (see low overlap with the ground-state in Fig. A.3). Orbital-optimization however allows to recover a lot of this error and is nearly exact. The "reference" performs better than SPA and the addition of Clifford gates allows to improve significantly upon that, however does not quite reach FCI. Both methods have overlap with inexact symmetry states and are also able to reach the triplet space. From Fig. A.3 (bond distance 3 A) we see, that while the reference has some yet significant overlap with the triplet and the subsequent singlet state, the addition of Cliffords attains high overlap with the triplet ground-state.

| Bond Distance (A) | Reference | Reference + Cliffords | SPA | FCI |
|---|---|---|---|---|
| 1.5 | -15.7877 | -15.7881 | -15.7878 | -15.7895 |
| 3.0 | -15.5318 | -15.5397 | -15.5328 | -15.5749 |
| 4.5 | -15.4137 | -15.5415 | -15.3658 | -15.5636 |

Table A.1: Energies in Hartree associated to overlap computation. Orbital-optimized SPA for $R = 4.5$ A attains an almost exact energy of $-15.5632$ Hartree.

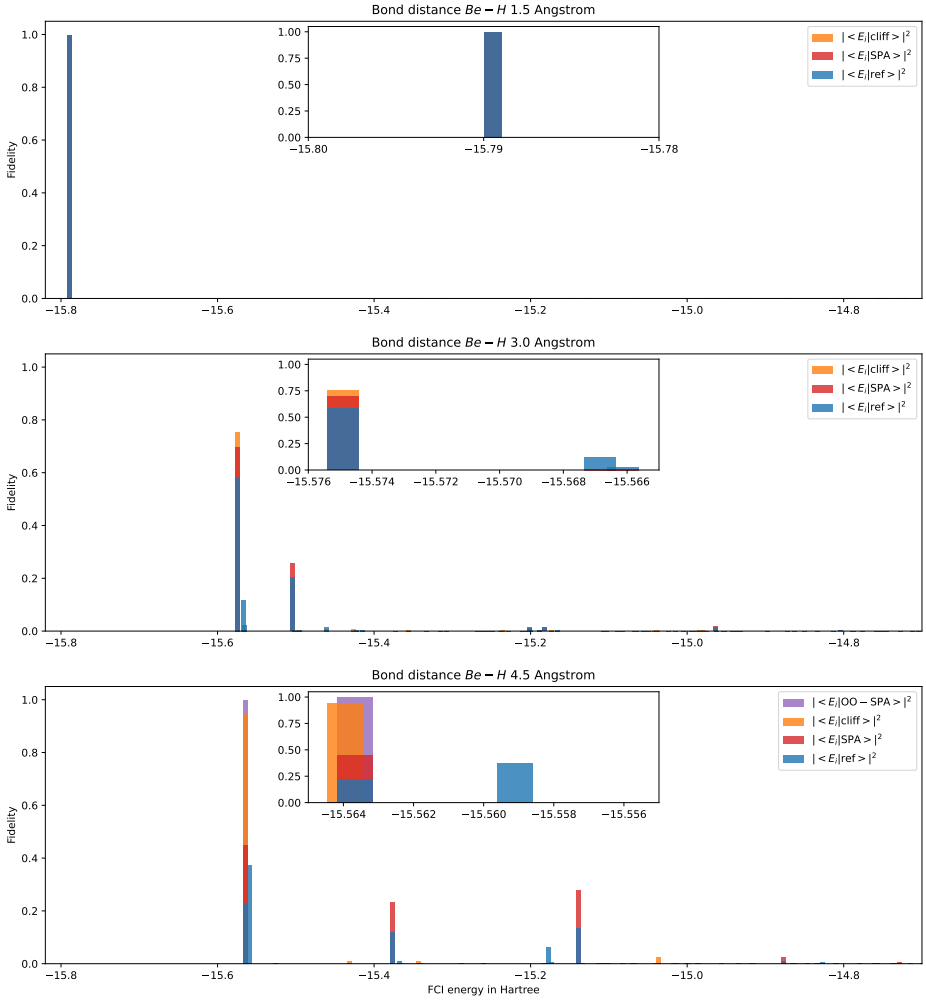



Figure A.3: Computation of fidelities between eigenstates obtained by direct diagonalization of the $BeH_2$ Hamiltonian in the PNO basis of 8 qubits in a frozen core approximation. ref$\sim$ power-method based reference, cliff$\sim$ref with optimized Clifford circuit. For $R = 4.5$ Angstrom, we also provide results for orbital-optimized SPA.

## A.4 Examples for optimized circuits

Below, we show some exemplary circuits for each molecule that showed higher/lower performance; this data is also part of our github repository at
`https://github.com/philipp-q/partitioning-with-cliffords`. We note that we only show the near-Clifford version with the best-performing non-Clifford gate (pink) as the respective Clifford circuit is simply given by a fixed parametrization. We provide the optimal angles in Table A.2, where we notice that for nitrogen, two of the computations yield (effectively) Clifford circuits. Beyond that, we note that in almost every case except for once, a parametrized $Z$ rotation yields the best configuration.

| | Geometry/A | Angle/rad | Rotation gate |
|---|---|---|---|
| $BeH_2$ | 1.0 | -0.00466767 | $R_Z$ |
| | 3.0 | 0.00155535 | $R_Z$ |
| $N_2$ | 3.0 | – (effectvely Clifford) | $R_Z$ |
| | 1.5 | – (Clifford) | – |
| | 2.25 | 0.03207715 | $R_Z$ |
| $H_2$ | 1.4 | $2.1953 \cdot 10^{-5}$ | $R_X$ |
| | 2.8 | 0.00329714 | $R_Z$ |

Table A.2: Angles for non-Clifford gates in example circuits.

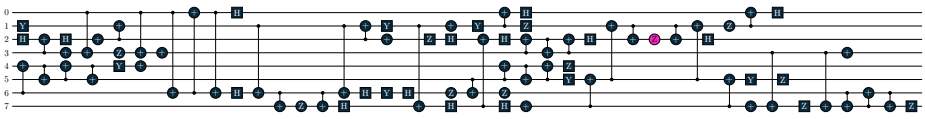


Figure A.4: Better-performing near-Clifford circuit for $BeH_2$ at a bond distance of 1.0 Angstrom.

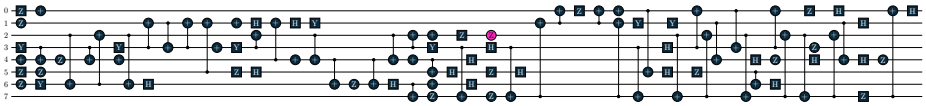


Figure A.5: Near-Clifford circuit for $BeH_2$ at 3.0 A; for $BeH_2$, the improvement is rather constant and less fluctuating than for $N_2$.

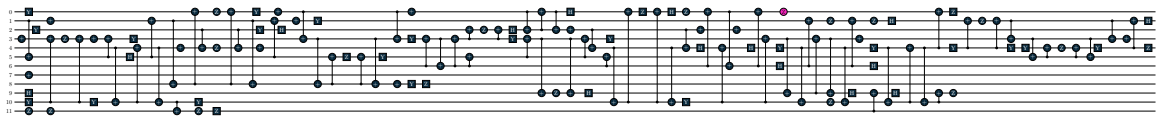


Figure A.6: Near-Clifford circuit for $N_2$ at bond distance 3.0; here, substantial improvement beyond the reference and the SPA is noticed (compare to Fig. 2.5), although the final, optimal circuit turned out to be effectively Clifford (see Table A.2).

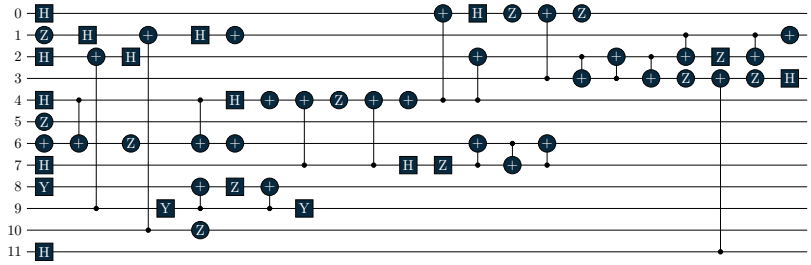


Figure A.7: Poorly performing near-Clifford optimization for $N_2$ at bond distance 1.5; here, when attempting to parametrize non-Clifford gates, no improvement could have been made out, so the result remains a Clifford circuit.

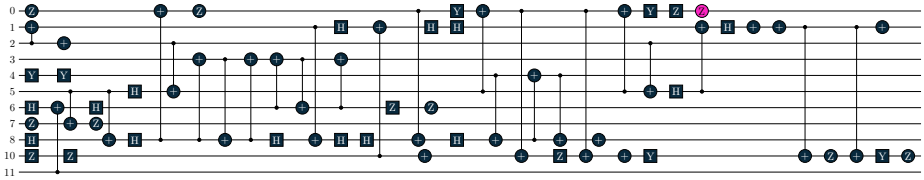


Figure A.8: Well-performing near-Clifford optimization for $N_2$ at bond distance 2.25; for this configuration, no actions have been performed on qubit 9.

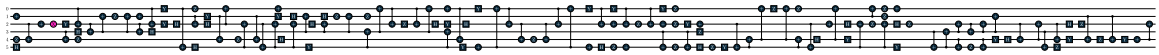


Figure A.9: Near-Clifford circuit for $H_2$ at bond distance 1.4; adding and optimizing a non-Clifford gate improves error by roughly an order or magnitude here.

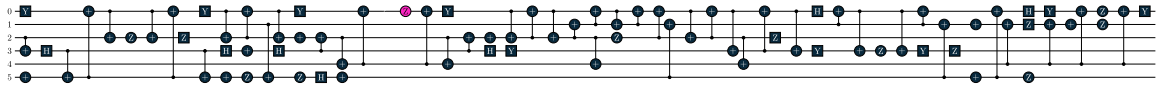


Figure A.10: Near-Clifford circuit for $H_2$ at bond distance 2.8; little improvement by adding a non-Clifford gate was observed for this configuration.

# Appendix B

# Supplementary Material for Section 2.2

## B.1 Scattering Molecules via Mergo-Association

This section includes more details on merging molecules by mergo-association as described in Section 2.2.4.

### B.1.1 Outline of Quantum Computational Encoding

First, we describe the quantum computational encoding of the approach. A grid $G$ of $N$ points labels an integer lattice, where for convenience, we assume that $N$ has an integer cubic root so that $\exists m \in \mathbb{N}$, $m^3 = N$,

$$G = \left[ -\frac{N^{1/3}-1}{2}, \frac{N^{1/3}-1}{2} \right]^3 . \tag{B.1.1}$$

The associated volume box $\Omega := [-\frac{L}{2}, \frac{L}{2}]^3$. Exemplarity for Dihyodrogen with $N_{\text{el}} = 2$, $N_{\text{nuc}} = 2$, an estimated volume of $|\Omega| = L^3 = (10a_0)^3$ is expected to suffice, where $a_0$ is the Bohr radius. With a grid spacing of $\sim 0.1\ a_0$, this volume would lead to $N \sim 10^6$. Then, upon defining the size of the simulation box, it is straightforward to map every $p \in G$ to a coordinate, $\mathtt{r}_p \in \Omega$ for electrons and $\mathtt{R}_p \in \Omega$ for nuclei.

The following is needed to describe the desired dynamics: Hamiltonians describing the isolated Hydrogen atoms, interactions between the Hydrogen atoms, and a 'trap potential' modeled by a harmonic oscillator potential. The system Hamiltonian without trap is given by

$$H_{\text{sys}} = T_{\text{el}} + T_{\text{nuc}} + U_{\text{nuc-el}} + V_{\text{ee}} + V_{\text{nn}}. \tag{B.1.2}$$

As the initial state to the simulation, we assume access to a product state of the ground states of the Hydrogen atom, respectively,

$$|\psi_{0,\text{H}}\rangle|\psi_{0,\text{H}}\rangle. \tag{B.1.3}$$

From [Su+21], we know the sub-normalization factors (which we will repeat further below) to encode these in real space, together with estimates for the cost of Hamiltonian simulation algorithms. The

placement of the individual $|\psi_{0,\mathrm{H}}\rangle$ should follow a guess for the desired molecular distance so that the relaxation through the trap potential induces a more accurate placement for the bonded state.

The harmonic trap potential from [BLH23; BH24] is

$$V^{\mathrm{trap}}(R_1, R_2) = V_1^{\mathrm{trap}}(R_1) + V_2^{\mathrm{trap}}(R_2), \tag{B.1.4}$$

which confines the nuclear motion per atom $j$:

$$V_j^{\mathrm{trap}}(R_j) = \frac{m_j}{2}(R_j - R_{0,j})^T \omega_j^2 (R_j - R_{0,j}). \tag{B.1.5}$$

We may neglect electronic mass and coordinates for the trap parameters given the large discrepancy between electronic and nuclear mass; the confinement is a heuristic approach nonetheless. The positions $R_{0,j}$ define the center of the two traps. For single-particle nuclei, such as Dihydrogen, the determination of trap centers is straightforward; in contrast, for larger systems, center-of-mass and relative coordinates need to be considered [BH24]. The frequencies $\omega_j^2$, in general, make up positive, diagonal matrices; the expression above describes the strength of the trap, which may not be spherically symmetric. For the sake of our discussion here, we will assume the trap to be isotropic, and then the trap frequency can be described by a scalar $\omega_j$. The role of these frequencies and relevant energy scales will be discussed further below in Appendix B.1.2.

Next, we discuss a quantum implementation of the trap potential. We reformulate Eq. (B.1.5) to

$$\begin{aligned} &V^{\mathrm{trap}}(R_1, \ldots, R_{N_{\mathrm{nuc}}}) = \\ &\quad = \sum_{j \in \mathrm{nuclei}} \frac{m_j}{2} \sum_{p \in G} \sum_{w \in \{x,y,z\}} \omega_{j,w}^2 (R_{p_j,w} - R_{0;w})^2 \, |p\rangle\langle p|_j \,, \end{aligned} \tag{B.1.6}$$

acting on position labeled by $p$. The state we encode is a superposition over $|p\rangle = |p_1\rangle \cdots |p_{N_{\mathrm{el}}+N_{\mathrm{nuc}}}\rangle$, which hold grid labels for each particle in a register, plus one extra qubit per particle for spin if desired. Synonymeously in this work, we use typewriter-font to denote the grid labels, $|\mathtt{r}_1\rangle \cdots |\mathtt{r}_{N_{\mathrm{el}}}\rangle \, |\mathtt{R}_1\rangle \cdots |\mathtt{R}_{N_{\mathrm{nuc}}}\rangle$. A state called $|r_j\rangle$ or $|R_k\rangle$ will hold the explicit coordinate information. This requires $\log(N)$ qubits per particle to represent its position on the grid. For Dihydrogen, overall $N_{\mathrm{nuc}} = N_{\mathrm{el}} = 2$, so we need $4 \times \log_2$(number of grid-points) qubits to encode the grid; with the estimated $10^6$ grid points from above, that makes roughly 80 qubits for the $|p\rangle$ register.

The procedure that is simulated is described using scheduling functions as mentioned above in Eq. (2.2.4), so that $f(s)$ introduces the inter-atomic Coulomb interactions and $g(s)$ guides the strength of the trap. As a consequence, implementing this scattering follows evolution across $\mathcal{T}\mathrm{e}^{-\mathrm{i}\int_0^{s_1} H(s)\mathrm{d}s}$, with $H(s)$ from Eq. (2.2.4). The truncated Dyson series algorithm in [LW18] can serve as a way to implement this. A necessary requirement is then to have access to the potentials as a HAM-T oracle, which we outline for $V^{\mathrm{trap}}$:

$$\sum_{s=0}^{n_t} \frac{V^{\mathrm{trap}}(s\frac{s_1}{n_t}) \otimes |s\rangle\langle s|}{\alpha_{\mathrm{trap}}}. \tag{B.1.7}$$

Using this, we can simulate over $s$ across the scheduling functions so that the inter-species interactions remain and the trap has fully decayed, i.e., until $s_1$. This operation can be constructed easily given

the LCU outlined in Eqs. (B.1.8) and (B.1.10) together with oracular access to the scheduling functions from Eq. (B.1.9) and an appropriate superposition over $|s\rangle$, $s = 0, \ldots, n_t$. To construct superpositions over where $n_t$ is not a power of two, see e.g. [Su+21, Appendix J]. The equivalent construction is necessary for the interaction term together with $f(s)$; see Eq. (2.2.4), and encodings for the Coulomb-term in the literature [Su+21, Appendix K]. There is an argument to make here about the (a)diabaticity of this evolution, which will be part of Appendix B.1.2.

We continue by outlining an LCU encoding of $V^{\text{trap}}(s')$ for a specific $0 \leq s' \leq s_1$. Coordinate directions are denoted by $\{x, y, z\} \sim \{0, 1, 2\}$ and we use the usual LCU notation of PREP and SEL operations so that $(\langle 0|_{\text{ancilla}} \otimes \mathbb{1}_{\text{system}})\text{PREP}^\dagger \cdot \text{SEL} \cdot \text{PREP}(|0\rangle_{\text{ancilla}} \otimes \mathbb{1}_{\text{system}})$ block-encodes the desired operator. The state we are acting on is assumed to be a superposition over $|p\rangle\,|s'\rangle$, with the state $|p\rangle$ representing a spatial part and $|s'\rangle$ the current step in the evolution of the merging. The PREP step involves preparing a superposition across nuclei and coordinate directions, weighted by the square root of the interaction strength at that point:

$$|0\rangle\,|p\rangle\,|s\rangle \mapsto \sum_{j:\text{nuclei}} \sum_{s=0}^{n_t} \sum_{w=0}^{2} \frac{\sqrt{\frac{m_j}{2}}\omega_{j,w}}{\sum_{j',w'} \sqrt{\frac{m_{j'}}{2}}\omega_{j',w'}} |j\rangle\,|w\rangle\,|p\rangle\,|s\rangle \tag{B.1.8}$$

Next, we illustrate the SEL operation, where we omit amplitudes and explicit sums for clarity. The goal here is to compute state-dependent interaction strengths on the fly. Namely, we aim to prepare a register that contains $(R_{p_{i,w}} - R_{0;w})^2$, and another containing $g(s)$ to multiply the latter, so that we can use e.g. [SLSB19] to move information from the state into the amplitude. Consequently, access to oracles $O_f, O_g$ that represent the scheduling is required,

$$O_{f/g} : |0\rangle\,|s\rangle \to |f/g(s)\rangle\,|s\rangle\,. \tag{B.1.9}$$

The oracular access to the scheduling function and the label-to-coordinate mapping may be realized with classical data encodings, e.g., a QROM or other techniques [Bab+18a; ZY24; Sun+24]. Using a number of ancillas given by $n_{\text{anc}}$ initialized in zero, and assuming a state that encodes the relative center of the trap $R_0$,

$$\begin{aligned}
&|j\rangle\,|w\rangle\,|s\rangle\,|0\rangle_{\text{a}}\,|p_1 \cdots p_{N_{\text{part}}}\rangle\,|R_{0;w}\rangle \\
\mapsto\ &|j\rangle\,|w\rangle\,|s\rangle\,|0\rangle_{\text{a}'}\,|g(s)\rangle\,|R_{p_{j,w}}\rangle\,|p\rangle\,|R_{0;w}\rangle \\
\mapsto\ &|j\rangle\,|w\rangle\,|s\rangle \qquad |g(s)(R_{p_{j,w}} - R_{0;w})^2\rangle\,|g(s)\rangle\,|R_{p_{j,w}}\rangle\,|p\rangle\,|R_{0;a,w}\rangle \\
\mapsto\ &|j\rangle\,|w\rangle\,|s\rangle\,|0\rangle_{\text{a}''}\,|g(s)|R_{p_j} - R_0|^2\rangle\,|p\rangle\,|R_{0;w}\rangle
\end{aligned} \tag{B.1.10}$$

The size of the necessary ancillary register $|0\rangle_{\text{a}}$ that is used to store intermediary and final results scales linearly in the bits of precision used for numerical representation (to store $g(s), R_{p_{j,w}}$ and perform multiplication and sum-of-squares). As mentioned above, to finalize the SEL operation, the state information in $|g(s)|R_{p_j} - R_0|^2\rangle$ needs to be moved into the amplitude. There are multiple ways to do so, such as the inequality-testing approach in [SLSB19], or, the conceptually simplest would be controlled rotations, conditioned on $|g(s)|R_{p_j} - R_0|^2\rangle$ and applied to the system register.

We refrain from analyzing exact Toffoli or gate counts as done in [Su+21], expecting that the addition of the trap potentials would not add a significant change here *per single query* to involved

oracles and circuits as the appearing arithmetic operations are comparable to the usual potentials in the Hamiltonian. The factor that may make the trap potentials more costly and needs to be studied more closely is its sub-normalization factor. This factor captures the number of necessary queries for a successful block-encoding. We recall the sub-normalization factors from [Su+21, Appendix K],

$$\alpha_T \in O\left(\frac{N_{\mathrm{el}} N^{2/3}}{|\Omega|^{2/3}}\right), \quad \alpha_V \in O\left(\frac{{N_{\mathrm{el}}}^2 N^{1/3}}{|\Omega|^{1/3}}\right), \quad \alpha_U \in O\left(\frac{{N_{\mathrm{el}}}^2 N^{1/3}}{|\Omega|^{1/3}}\right) \tag{B.1.11}$$

with $T$ kinetic energy, $V$ electron-electron interactions and $U$ nuclear-electron interactions.

The block-encoding factor of the trap potential may be upper-bounded by $\alpha_{\mathrm{trap}} \leq \|\sum_{j\in[N_{\mathrm{nuc}}]} V_j^{\mathrm{trap}}\|$. Then, $\alpha_{\mathrm{trap}} \leq N_{\mathrm{nuc}} m_{\max} \omega_{\max}^2 \max_{r,r'} |r-r'|^2$. In the present discretization, $\max_{r,r'\in\Omega_{\mathrm{trap}}} |r-r'|^2 \in O(|\Omega_{\mathrm{trap}}|^{2/3})$, where $\Omega_{\mathrm{trap}} \subseteq \Omega$ is the part of the domain the trap potential is defined on. Most of the mass of particles under such a potential will be incentivized to be close to the center; therefore, an upper bound that puts the maximum mass at a maximum distance will not be very tight in most scenarios, and a smaller $\alpha_{\mathrm{trap}}$ may be sufficient. One could further think about designing the shape of the trap potential to be decaying for larger distances to reduce the encoding cost or oscillatory as in optical tweezers, such that maximum amplitude is always within reach – we leave concrete specifications up to future work. The maximum nuclear mass $m_{\max} = \max_{1\leq j\leq N_{\mathrm{nuc}}} m_j$ can be treated as a constant factor attached to the nuclei. Then, we have that

$$\alpha_{\mathrm{trap}} \in O\left(\omega_{\max}^2 N_{\mathrm{nuc}} |\Omega_{\mathrm{trap}}|^{2/3}\right). \tag{B.1.12}$$

Up to the frequency $\omega_{\max}$, the sub-normalizations of the present procedure in Eq. (B.1.11) compared to the trap potential inhibits a few key differences. Because the strength of the trap depends on a maximum distance, there is no notion of 'grid density' in the cost. Therefore, the factors in Eq. (B.1.11) increase when a lower target accuracy is desired. The trap encoding is not directly dependent on accuracy, though it is more pronounced on system size. The system size of the trap may be upper-bounded with the overall system studied, as present at the last scattering stage before 'evolution' in Fig. 2.10.

Therefore, it is key to understand the role of $\omega_{\max}$ to the scattering and whether there is a way to relate this quantity to the system size.

### B.1.2 Probability of Success of Mergo-Association

To address the choice of trap frequency, we consider that the probability of a diabatic transition into $|a\rangle$ through the mergo-association from [BLH23; BH24] can be approximated by the Landau-Zener rule and is proportional to

$$p_{\mathrm{LZ}} \approxeq \exp\left(-\frac{2\pi}{\hbar} \frac{\omega_{\mathrm{eff}}^2}{|\frac{\partial}{\partial s}(E_{\mathrm{mol}} - E_{\mathrm{atom}})|}\right) \tag{B.1.13}$$

Here $\omega_{\mathrm{eff}}$ is an effective frequency we explain further below that models the strength of the potential. Then, we express the rate of change of the energy, $\frac{\partial}{\partial s}(E_{\mathrm{mol}} - \partial E_{\mathrm{atom}})$, by consequence of the chain rule, as $|\partial E_{\mathrm{mol}} - \partial E_{\mathrm{atom}}|\, v$. Here, $\partial E$ are energy derivatives near the initial configuration with

respect to the internuclear distance and $v$, the speed of the evolution along the scheduling function.

$$p_{\mathrm{LZ}} \cong \exp\left(-\frac{2\pi}{\hbar}\frac{\omega_{\mathrm{eff}}^2}{|\partial E_{\mathrm{mol}} - \partial E_{\mathrm{atom}}|\, v}\right) \tag{B.1.14}$$

Thus, the success probability of the mergo-association, so that the merging happens without transitioning into a higher vibrational state of the trapped system as desired, is

$$p_{\mathrm{suc}} \sim 1 - p_{\mathrm{LZ}}. \tag{B.1.15}$$

As we pointed out in the previous section, this means that the type of evolution we call successful here is adiabatic. It is important to note that this is not adiabatic ground-state preparation, which is explicitly beyond the scope of our work. Following another discussion in the preceding section, the threshold to success here considers molecular bound states, so the 'upper limit' for adiabaticity would be the highest-lying vibrational bound state. One way or another, a practical implementation will need to consider this. This still poses limits on the speed of evolution (given by $v$ in Eq. (B.1.14)). Then, $v = \max\{|\frac{\mathrm{d}z}{\mathrm{d}f}\frac{\mathrm{d}}{\mathrm{d}s}f(s^*)|, |\frac{\mathrm{d}z}{\mathrm{d}g}\frac{\mathrm{d}}{\mathrm{d}s}g(s^*)|\}$, where $z$ is the considered internuclear distance.

Looking at Eq. (B.1.14), we can identify that the stronger ("steeper") the potential as described by the effective frequency, the higher the success, and the faster the merging proceeds, the less likely. This conclusion is consistent with intuition.

We continue by breaking down the quantities that appear. Following [BLH23], $\partial E_{\mathrm{atom}} \approx 0$, as it is reasonable to assume the energy surface for separated atoms is flat. Taking a harmonic oscillator approximation to the relative motion of the nuclei, [BLH23, Eq. (37)] obtains an approximation to the gradient, $\partial E_{\mathrm{mol}} \approx \mu\omega^2 z^*$, where $z^*$ is the point of contact of the avoided crossing between the two states. In terms of the scheduling functions that we envision, this corresponds to the state at $s = s^*$, $0 < s^* < s_0 < s_1$ (Fig. 2.12): briefly before both interaction and trap are 'fully acting'. Assuming an isotropic potential, the spatial coordinate is approximated as $z^* = (\frac{\hbar}{\mu\omega})^{1/2}(3 + \frac{\omega_a}{\omega})^{1/2}$ so that [BLH23, Eq.(35)]

$$\partial E_{\mathrm{mol}} \approx (\hbar\mu)^{1/2}\omega(3\omega + \omega_a)^{1/2}, \tag{B.1.16}$$

where $\omega_a$ is defined as the frequency associated with the bonded vibrational excited state.

To estimate $\omega_{\mathrm{eff}}$, we can also follow [BLH23, Eq. (52)]:

$$\begin{aligned}\omega_{\mathrm{eff}}^2 &\cong \left(\frac{\langle a|(\mathbb{1} - |000\rangle\langle 000|)V^{\mathrm{int}}|000\rangle}{1 - |\langle a|000\rangle|^2}\right)^2 \\ &\approx \langle a|V^{\mathrm{int}}|000\rangle^2 = \frac{2\hbar^2}{\sqrt{\pi}}\omega_a^{1/2}\omega^{3/2}\exp\left[-\frac{1}{2}\left(3 + \frac{\omega_a}{\omega}\right)\right].\end{aligned} \tag{B.1.17}$$

States "000" describe the vibrational ground-state and "$a$" an excited vibrational state, $\omega$ is the frequency of the harmonic oscillator trap.The approximation to only look at the transition element itself is investigated in [BLH23, Fig. 6] – for a merely qualitative argument, this is a sufficient choice. The choice for $\omega_a$ in our case is an estimate for an upper threshold of the respective *bonded* vibrational energy subspace. Introducing a 'binding energy'-quantity $E_a = \hbar\omega_a$, we can rewrite $\omega_{\mathrm{eff}}^2$ to be approximately proportional to [BLH23, Eq. (54)],

$$\omega_{\mathrm{eff}}^2 \propto \omega^{3/2}(E_a/\hbar)^{1/2}\exp\left(-\frac{E_a}{\hbar\omega}\right) = \omega^2\tilde{E}_a^{1/2}\exp\left(-\tilde{E}_a\right). \tag{B.1.18}$$

In the latter equation, we introduce the relative binding energy $\tilde{E}_a = E_a/(\hbar\omega)$, defined in relation to the trap energy. Then,

$$\begin{aligned} p_{\mathrm{LZ}} &\approx \exp\left(-4\left(\frac{\hbar\pi}{\mu}\right)^{1/2}\left(\frac{\omega\omega_a}{3\omega+\omega_a}\right)^{1/2}\frac{e^{-\frac{1}{2}\left(3+\frac{\omega_a}{\omega}\right)}}{v}\right) \\ &= \exp\left(-4\left(\frac{\pi}{\mu}\right)^{1/2}\left(\frac{\hbar\omega E_a}{3\hbar\omega+E_a}\right)^{1/2}\frac{\exp\left(-\frac{1}{2}\frac{E_a}{\hbar\omega}-\frac{3}{2}\right)}{v}\right) \\ &= \exp\left(-4\left(\frac{\pi}{\mu}\right)^{1/2}\left(\hbar\omega\frac{\tilde{E}_a}{3+\tilde{E}_a}\right)^{1/2}\frac{\exp\left(-\frac{1}{2}\tilde{E}_a-\frac{3}{2}\right)}{v}\right). \end{aligned} \tag{B.1.19}$$

## B.2 Construction of Measurement Oracles for Certifying Reactions

For this oracle construction, we assume a first-quantized real space implementation. The choice of representation in first-quantization on real space makes the construction of the following oracle easier; of course, plane-wave or plane-wave dual bases [Bab+18b] are possible too and can be translated to with appropriate transformations. The encoded quantum states are linear combinations of wavefunctions with Fermionic (antisymmetric) and Bosonic (symmetric) parts. Within such a combination, a single component looks like a tensor product over electronic $\{r\} \sim \{r_j\}_{j=1}^{N_{\mathrm{el}}}$ and nuclear $\{R\} \sim \{R_k\}_{k=1}^{N_{\mathrm{nuc}}}$ grid labels (represented on the grid Eq. (B.1.1)),

$$|\psi(\{r,R\})\rangle = |\mathtt{r}_1 \cdots \mathtt{r}_{N_{\mathrm{el}}}\rangle_{\mathrm{el}} \, |\mathtt{r}_1 \cdots \mathtt{r}_{N_{\mathrm{nuc}}}\rangle_{\mathrm{nuc}} \, ; \tag{B.2.1}$$

We further remember that the above collapsed notation that also contain spin information in an extra qubit, i.e., $|\mathtt{r}_j\rangle \sim |\mathtt{r}_j\rangle \otimes |\sigma_j\rangle$ with $\sigma_j \in \{\uparrow,\downarrow\}$, and analogous for nuclei if desired. Then, a density matrix formulation of linear combinations of such a state is

$$\rho = \sum_{\{r,R\},\{r',R'\}} \rho_{\{r,R\},\{r',R'\}} \, |\psi(\{r,R\})\rangle\langle\psi(\{r',R'\})| \, , \tag{B.2.2}$$

with the entries of the density matrix in the position basis, $\rho_{\{r,R\},\{r',R'\}} \in \mathbb{C}$. The goal of this section is to construct an oracle that can distinguish the 'good', reacted components of such a state from the 'bad', non-reacted components. Given a state representing two molecular fragments that have evolved for some time, we want to distinguish the parts of the wavefunction that correspond to the fragments having reacted to form a single molecule from those where they are still separate unbound fragments.

As we consider linear transformations, it suffices to study the effect on the individual components from Eq. (B.2.1) to draw conclusions for a general state in Eq. (B.2.2). Therefore, we first focus on a single position-basis state $|\psi(\{r,R\})\rangle$ in the upcoming Appendix B.2.1 before we consider exchange symmetry in Appendix B.2.2 and more general superpositions in Appendix B.2.3.

### B.2.1 Geometrical Criteria

We now discuss an approach to test whether or not a molecular bond has formed for a state of the form $|\psi(\{r, R\})\rangle$. When atomic nuclei can be precisely located, a reasonable description for the molecular structure can be given in terms of a set of inter-nuclear distances $\{R_{jk}^{\text{react}}\}$. Using this information, a corresponding criterion $\mathcal{C}$ for whether a given configuration corresponds to a desired molecule can then be constructed,

$$\mathcal{C}_{\text{geom},\varepsilon}(\{r, R\})) = \begin{cases} 1, & \left| \|R_j - R_k\| - R_{jk}^{\text{react}} \right| \leq \varepsilon \ \forall j, k \\ 0, & \text{else.} \end{cases} \tag{B.2.3}$$

Note that evaluating this criterion is classically efficient with respect to the number of bits devoted to the constituent quantities, scaling no worse than $O(N_{\text{nuc}}^2)$ in the number of nuclei and no worse than quadratically in the bit-precision due to a product and a square root in the evaluation of the norm. In other words, evaluating this function is classically efficient in the size of the molecule as long as the bit-precision only grows polynomially, corresponding to the requirement that the number of grid points in the simulation should not grow faster than exponential in the involved particle number.

In some cases, it may be possible to devise more accurate criteria, such as by comparing the angles between inter-nuclear distances to desired bond angles. On the other hand, there may be cases where less is known about the target structure. Then, the best that one can hope for is to check that the two fragments are at least spatially adjacent, corresponding to the looser requirement

$$\tilde{\mathcal{C}}_{\text{geom},\Delta}(\{r, R\}, I) = \begin{cases} 1, & \|R_j - R_k\| \leq \Delta_{jk} \ \forall (j, k) \in I \subseteq [N_{\text{nuc}}] \times [N_{\text{nuc}}] \\ 0, & \text{else.} \end{cases} \tag{B.2.4}$$

for some suitable $\Delta_{jk} > 0$ and a set of locations of interest $\mathcal{I}$, in which we only check a subset of the encoded locations regarding proximity. There are many ways this type of geometric requirements could be mixed and matched, e.g., utilizing different levels of knowledge or different precisions $\varepsilon$ for different groups of atoms. Either way, common to these approaches is the fact that the classical evaluation of the criterion is efficient under the mild restrictions to the grid resolution outlined above.

A result of this classical efficiency is that a corresponding unitary to perform the computation can also be implemented efficiently on a quantum computer using reversible logic. Specifically, for a given efficient criteria $\mathcal{C}$ of the form considered here, the following unitary can be implemented:

$$U_{\mathcal{C}} \left| p_{N_{\text{el}}} \cdots p_{N_{\text{el}}+N_{\text{nuc}}} \right\rangle_{\text{nuc}} |0\rangle = \left| p_{N_{\text{el}}} \cdots p_{N_{\text{el}}+N_{\text{nuc}}} \right\rangle_{\text{nuc}} |\mathcal{C}(\{r, R\})\rangle \,. \tag{B.2.5}$$

Thus, extracting information about the criteria into an ancilla qubit for easy access is possible. Using a measurement of this ancilla then allows for a projection onto the good ($\mathcal{C} = 1$) subspace, heralded by an outcome "1" of the measurement. However, the outcome "0" similarly fully projects the state into the bad subspace ($\mathcal{C} = 0$). Furthermore, care needs to be taken to avoid either kind of projection destroying physically important symmetries of the state, including Fermionic and Bosonic particle-exchange symmetries. Mitigating each of these potential problems will be the topics of the following two sections.

### B.2.2 Preserving Exchange Symmetry Throughout Measurement

So far, we have been primarily considering a single component of the wavefunction, $|\psi(\{r, R\})\rangle$, in which the positions of all particles are fully specified. However, we typically deal with a collection of nuclei and electrons modeled through wavefunctions that need to be of the correct symmetry with respect to the exchange of particles. We focus on preserving exchange symmetry compared to other symmetries present in the system that may also be broken in the real physical processes. Hence, the goal is to evaluate reaction criteria in a manner that does not violate the symmetries of present states.

#### Symmetrization of the wavefunction

Recall that the wavefunctions we encode are given as superpositions of position-labelling basis states,

$$\left(|\mathtt{r}_1\rangle|\mathtt{r}_2\rangle\cdots|\mathtt{r}_{N_{\mathrm{el}}}\rangle\right)\left(|\mathtt{R}_1\rangle|\mathtt{R}_2\rangle\cdots|\mathtt{R}_{N_{\mathrm{nuc}}}\rangle\right) \tag{B.2.6}$$

We repeat that by $|\mathtt{r}_j\rangle, |\mathtt{R}_k\rangle$ we mean the labels for grid points and spin rather than the resolved coordinates and choose this notation as it is more illustrative in the context of (anti)symmetrization of the wavefunctions. For Fermionic (indistinguishable) particles, the wavefunction needs to be anti-symmetrized with respect to the indistinguishable degrees of freedom. Formally, this can be done by applying the antisymmetrization operator $\mathcal{A}$. Using the electronic degrees of freedom as an example, and with $\sigma$ a permutation from the permutation group over $N_{\mathrm{el}}$ elements, $S_{N_{\mathrm{el}}}$:

$$\mathcal{A}\left(|\mathtt{r}_1\rangle\,|\mathtt{r}_2\rangle\cdots|\mathtt{r}_{N_{\mathrm{el}}}\rangle\right) = \frac{1}{\sqrt{N_{\mathrm{el}}!}}\sum_{\sigma\in S_{N_{\mathrm{el}}}}\mathrm{sgn}(\sigma)\,|\mathtt{r}_{\sigma(1)}\rangle\,|\mathtt{r}_{\sigma(2)}\rangle\cdots|\mathtt{r}_{\sigma(N_{\mathrm{el}})}\rangle \tag{B.2.7}$$

Similarly, Bosonic degrees of freedom will be symmetric with respect to exchange, described by symmetrization $\mathcal{S}$ of the general form

$$\mathcal{S}\left(|\mathtt{R}_1\rangle\,|\mathtt{R}_1\rangle\cdots|\mathtt{R}_{N_{\mathrm{nuc}}}\rangle\right) = \frac{1}{\sqrt{k!}}\sum_{\sigma\in S_{N_{\mathrm{nuc}}}}|\mathtt{R}_{\sigma(1)}\rangle\,|\mathtt{R}_{\sigma(2)}\rangle\cdots|\mathtt{R}_{\sigma(N_{\mathrm{nuc}})}\rangle \tag{B.2.8}$$

In general, a set of nuclei to be modeled as indistinguishable will correspond to a subset of nuclear registers to be (anti)symmetrized. Thus, the desired symmetry properties are characterized by sets of registers to be symmetrized, $\{B_i\}_{i=1}^{N_B}$, and sets of registers to be anti-symmetrized, $\{F_i\}_{i=1}^{N_F}$. Denoting which basis states an operator acts on by a subscript, one gets the following anti-symmetrized state:

$$\mathcal{A}\left(|\mathtt{r}_1\rangle\,|\mathtt{r}_2\rangle\cdots|\mathtt{r}_{N_{\mathrm{el}}}\rangle\right)\otimes\left(\prod_{i=1}^{N_F}\mathcal{A}_{F_i}\prod_{j=1}^{N_B}\mathcal{S}_{B_j}\right)\left(|\mathtt{R}_1\rangle\,|\mathtt{R}_2\rangle\cdots|\mathtt{R}_{N_{\mathrm{nuc}}}\rangle\right). \tag{B.2.9}$$

#### Illustrative example: $H_2O_2$

To elucidate our notation, consider the molecule $H_2O_2$, and assume that the non-symmetrized positions are encoded as

$$|\mathtt{R}_{O,1}\rangle\,|\mathtt{R}_{O,2}\rangle\,|\mathtt{R}_{H,1}\rangle\,|\mathtt{R}_{H,2}\rangle\,. \tag{B.2.10}$$

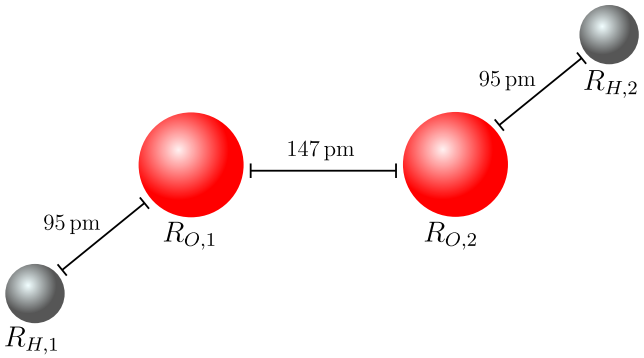



Figure B.1: Equilibrium configuration of a $H_2O_2$ molecule, with oxygen nuclei marked in red and hydrogen nuclei marked in grey.

In this case, the oxygen nuclei are spin-0 nuclei, meaning they should be symmetrized. On the other hand, the hydrogen nuclei are spin-1/2 particles, meaning they should be anti-symmetrized. We can represent this by the following two sets:

$$B_1 = \{1, 2\}, \quad F_1 = \{3, 4\}. \tag{B.2.11}$$

The first tells us to symmetrize the oxygen registers (1 and 2), and the second tells us to anti-symmetrize the hydrogen registers (3 and 4). The state after symmetrization is therefore

$$\begin{aligned}\mathcal{A}_{F_1}\mathcal{S}_{B_1}\left(|\mathtt{R}_{O,1}\rangle\,|\mathtt{R}_{O,2}\rangle\,|\mathtt{R}_{H,1}\rangle\,|\mathtt{R}_{H,2}\rangle\right) =& \frac{1}{2}\left(|\mathtt{R}_{O,1}\rangle\,|\mathtt{R}_{O,2}\rangle + |\mathtt{R}_{O,2}\rangle\,|\mathtt{R}_{O,1}\rangle\right) \\ & \otimes \left(|\mathtt{R}_{H,1}\rangle\,|\mathtt{R}_{H,2}\rangle - |\mathtt{R}_{H,2}\rangle\,|\mathtt{R}_{H,1}\rangle\right).\end{aligned} \tag{B.2.12}$$

The encoded molecular state will generally be a superposition of states of form in Eq. (B.2.9), i.e., the tensor products of an antisymmetric wavefunction, a symmetric wavefunction, and possibly a non-symmetrized component. Then, for the remainder, it suffices to discuss the oracle's effect on one term of the superposition. To see what symmetry implies for the construction of oracles, consider again the example of $H_2O_2$. Looking at the form of the wavefunction in Eq. (B.2.10) and the equilibrium geometry of the molecule, one might be tempted to define a criteria of the form

$$\begin{aligned}\mathcal{C}_\varepsilon(\{R\}) = \Big(&\left|\|R_1 - R_3\| - 95\,\mathrm{pm}\right| < \varepsilon) \wedge (\left|\|R_2 - R_4\| - 95\,\mathrm{pm}\right| < \varepsilon) \\ & \wedge\ (\left|\|R_1 - R_2\| - 147\,\mathrm{pm}\right| < \varepsilon)\Big)\end{aligned} \tag{B.2.13}$$

in suitable numerical units and with subscripts denoting the ordering of the registers. After all, if we achieve the equilibrium configuration (See Fig. B.1), an oracle based on this criteria will, by construction, certify the non-symmetrized state as reacted:

$$\begin{aligned}U_{\mathcal{C}}\,|\mathtt{R}_{O,1}\rangle\,|\mathtt{R}_{O,2}\rangle\,|\mathtt{R}_{H,1}\rangle\,|\mathtt{R}_{H,2}\rangle\,|0\rangle &= |\mathtt{R}_{O,1}\rangle\,|\mathtt{R}_{O,2}\rangle\,|\mathtt{R}_{H,1}\rangle\,|\mathtt{R}_{H,2}\rangle \\ &\quad \otimes |\mathcal{C}_\varepsilon(\{R_{O,1}, R_{O,2}, R_{H,1}, R_{H,2}\})\rangle \\ &= |\mathtt{R}_{O,1}\rangle\,|\mathtt{R}_{O,2}\rangle\,|\mathtt{R}_{H,1}\rangle\,|\mathtt{R}_{H,2}\rangle\,|1\rangle\end{aligned} \tag{B.2.14}$$

However, the same cannot be said for states where the positions are permuted. For instance, swap-

ping the oxygen positions, the evaluation of the oracle reads

$$\begin{aligned}\mathcal{C}_\varepsilon(\{R_{O,2}, R_{O,1}, R_{H,1}, R_{H,2}\}) = &(|\|R_{O,2} - R_{H,1}\| - 95\,\mathrm{pm}| < \varepsilon) \\ &\wedge (|\|R_{O,1} - R_{H,2}\| - 95\,\mathrm{pm}| < \varepsilon) \\ &\wedge\ (|\|R_{O,2} - R_{O,1}\| - 147\,\mathrm{pm}| < \varepsilon)\end{aligned} \tag{B.2.15}$$

and consulting the geometric configuration in Fig. B.1) shows that this Boolean formula evaluates to zero. Thus,

$$U_\mathcal{C} \left|\mathtt{R}_{O,2}\right\rangle \left|\mathtt{R}_{O,1}\right\rangle \left|\mathtt{R}_{H,1}\right\rangle \left|\mathtt{R}_{H,2}\right\rangle |0\rangle = \left|\mathtt{R}_{O,2}\right\rangle \left|\mathtt{R}_{O,1}\right\rangle \left|\mathtt{R}_{H,1}\right\rangle \left|\mathtt{R}_{H,2}\right\rangle |0\rangle\,. \tag{B.2.16}$$

Considering the full symmetrized wavefunction, one finds

$$\begin{aligned}U_\mathcal{C} &\left[\frac{1}{2}\left(\left|\mathtt{R}_{O,1}\right\rangle \left|\mathtt{R}_{O,2}\right\rangle + \left|\mathtt{R}_{O,2}\right\rangle \left|\mathtt{R}_{O,1}\right\rangle\right) \otimes \left(\left|\mathtt{R}_{H,1}\right\rangle \left|\mathtt{R}_{H,2}\right\rangle - \left|\mathtt{R}_{H,2}\right\rangle \left|\mathtt{R}_{H,1}\right\rangle\right)\right] |0\rangle \\ &= \frac{1}{2}\left[\left|\mathtt{R}_{O,1}\right\rangle \left|\mathtt{R}_{O,2}\right\rangle \left|\mathtt{R}_{H,1}\right\rangle \left|\mathtt{R}_{H,2}\right\rangle - \left|\mathtt{R}_{O,2}\right\rangle \left|\mathtt{R}_{O,1}\right\rangle \left|\mathtt{R}_{H,2}\right\rangle \left|\mathtt{R}_{H,1}\right\rangle\right] |1\rangle \\ &+ \frac{1}{2}\left[\left|\mathtt{R}_{O,2}\right\rangle \left|\mathtt{R}_{O,1}\right\rangle \left|\mathtt{R}_{H,1}\right\rangle \left|\mathtt{R}_{H,2}\right\rangle - \left|\mathtt{R}_{O,1}\right\rangle \left|\mathtt{R}_{O,2}\right\rangle \left|\mathtt{R}_{H,2}\right\rangle \left|\mathtt{R}_{H,1}\right\rangle\right] |0\rangle\end{aligned} \tag{B.2.17}$$

Despite all four components corresponding to the same molecule (just with particle labeling altered), only two of them have been certified by the oracle. Furthermore, if the ancilla is measured, the resulting projected state is no longer of the symmetric form described in Eq. (B.2.9), and direct inspection shows that it is neither symmetric nor anti-symmetric upon permutations among the nuclear coordinates.

One possible response to this problem might be simply re-applying an (anti-)symmetrization step whenever a projection occurs. In principle, this operation scales linearly (up to logarithmic factors) in the system size[Ber+18; LCG24]; thus, it would not change the polynomial runtime of the algorithm. However, existing algorithms require a particular structure on the input state, a requirement that is linked to unitarity [Ber+18] and thus likely challenging to lift. Furthermore, even with a hypothetical generalization of the algorithm, the present work proposes a quasi-continuous weak-measurement strategy for monitoring the reaction criteria. Since this entails frequent (weak) measurements during the simulation run, it is likely to lead to a high total number of measurements, scaling extensively with the simulation time. Restoring symmetry after each measurement could, therefore, add significant complexity to the algorithm, even if a fast and applicable symmetrization algorithm is available. Therefore, we next discuss a more comprehensive approach to address these shortcomings.

### Symmetry of states

To build certification oracles that are valid under the correct exchange symmetry, we will first formalize the symmetrization requirement for the states of the system. Consider first a single set of identical particles, represented by a set of indices $\mathcal{X}$. The group of permutations of this set is characterized by the symmetric group, $S_{|\mathcal{X}|}$, and for each permutation $\sigma \in S_{|\mathcal{X}|}$, we can define a

unitary consisting of SWAP gates that implements the permutation our registers:

$$U_\sigma \left|\mathtt{R}_1\right\rangle \left|\mathtt{R}_2\right\rangle \ldots \left|\mathtt{R}_{|\mathcal{X}|}\right\rangle = \left|\mathtt{R}_{\sigma(1)}\right\rangle \left|\mathtt{R}_{\sigma(2)}\right\rangle \ldots \left|\mathtt{R}_{\sigma(|\mathcal{X}|)}\right\rangle, \tag{B.2.18}$$

or, more compactly

$$U_\sigma \left|\psi\{r, R\}\right\rangle = \left|\psi\{\sigma[r, R]\}\right\rangle \tag{B.2.19}$$

$\sigma[\{r, R\}]$ denotes the (ordered) set of positions after the permutation has been applied. In a group-theory language, this corresponds to a representation of the group $S_{|\mathcal{X}|}$ onto the space of states. In terms of these constructions, the symmetrization requirement is

$$U_\sigma \left|\psi\right\rangle = \begin{cases} \mathrm{sgn}(\sigma)\left|\psi\right\rangle & \text{Fermions} \\ \left|\psi\right\rangle & \text{Bosons} \end{cases} \qquad \forall \sigma \in S_{|\mathcal{X}|} \tag{B.2.20}$$

where $\mathrm{sgn}(\sigma)$ is the sign of the permutation; the expression depends on whether the identical particles represented by the indices in $\mathcal{X}$ are Fermions or Bosons.

Generalizing this construction to multiple sets of Bosonic and Fermionic particles, the symmetry group takes the form of a product group of the individual Bosonic and Fermionic symmetries,

$$S = S_B \otimes S_F \tag{B.2.21}$$

$$S_B = \bigotimes_{j=1}^{N_B} S_{|B_j|}, \quad S_F = \bigotimes_{i=1}^{N_F} S_{|F_i|}, \tag{B.2.22}$$

with the group representation taking a similar product form. Note that the tensor product structure implies that every element in $S$ can be decomposed into the product of a Fermionic and Bosonic component, $\sigma = \sigma_B \otimes \sigma_F$. Using this notation, the symmetry requirement becomes

$$U_\sigma \left|\psi\right\rangle = \mathrm{sgn}(\sigma_F)\left|\psi\right\rangle \quad \forall \sigma \in S. \tag{B.2.23}$$

**Symmetry of oracles**

Using the notation introduced above, we can now state the requirement that a reaction criteria $\mathcal{C}$ needs to fulfill to respect the symmetry defined in the previous section. Specifically, we will say that a criterion $\mathcal{C}$ respects symmetry when a projective measurement of $\mathcal{C}$ on a symmetrized state produces post-measurement states that are still correctly symmetrized. Below, we will show that this property holds if and only if

$$\mathcal{C}(\sigma[\{r, R\}]) = \mathcal{C}(\{r, R\}) \quad \forall \sigma \in S,\ \forall \{r, R\}\ . \tag{B.2.24}$$

To see that this is sufficient criteria, consider the projection operators related to the readout of $\mathcal{C}$ using an ancilla qubit:

$$\Pi_\pm^{\mathcal{C}} = \frac{1}{2} U_{\mathcal{C}}^\dagger \left(\mathbb{1} \otimes (\mathbb{1} \pm Z)_{\mathrm{anc}}\right) U_{\mathcal{C}} \tag{B.2.25}$$

$$\tilde{\Pi}_\pm^{\mathcal{C}} = \left(\mathbb{1} \otimes \langle 0|_{\mathrm{anc}}\right) \Pi_\pm^{\mathcal{C}} \left(\mathbb{1} \otimes |0\rangle_{\mathrm{anc}}\right) \tag{B.2.26}$$

Looking at the constituent operators, we can note that

$$\begin{aligned} U_{\mathcal{C}}\left(U_\sigma \otimes \mathbb{1}\right)|\psi(\{r,R\})\rangle\,|0\rangle &= U_{\mathcal{C}}\,|\psi(\sigma[\{r,R\}])\rangle\,|0\rangle \\ &= |\psi(\sigma[\{r,R\}])\rangle\,|\mathcal{C}(\sigma[\{r,R\}])\rangle \qquad \text{(B.2.27)} \\ \left(U_\sigma \otimes \mathbb{1}\right)U_{\mathcal{C}}\,|\psi(\{r,R\})\rangle\,|0\rangle &= U_\sigma\,|\psi(\{r,R\})\rangle\,|\mathcal{C}(\{r,R\})\rangle \\ &= |\psi(\sigma[\{r,R\}])\rangle\,|\mathcal{C}(\{r,R\})\rangle\,. \qquad \text{(B.2.28)} \end{aligned}$$

Thus, if the relation in Eq. (B.2.24) holds, it implies that

$$[U_{\mathcal{C}}, (U_\sigma \otimes \mathbb{1})] = \left[U_{\mathcal{C}}^\dagger, (U_\sigma \otimes \mathbb{1})\right] = 0 \quad \forall\sigma \in S. \qquad \text{(B.2.29)}$$

Therefore, the projectors also commute with the permutations,

$$\left[\Pi_\pm^{\mathcal{C}}, (U_\sigma \otimes \mathbb{1})\right] = \left[\tilde{\Pi}_\pm^{\mathcal{C}}, U_\sigma\right] = 0 \quad \forall\sigma \in S, \qquad \text{(B.2.30)}$$

which in turn means the post-projection states of (anti)symmetrized states are also (anti)symmetrized

$$\begin{aligned} U_\sigma\left(\tilde{\Pi}_\pm^{\mathcal{C}}\,|\psi\rangle\right) &= \tilde{\Pi}_\pm^{\mathcal{C}}\,U_\sigma\,|\psi\rangle \\ &= \tilde{\Pi}_\pm^{\mathcal{C}}\,\mathrm{sgn}(\sigma_F)\,|\psi\rangle \\ &= \mathrm{sgn}(\sigma_F)\left(\tilde{\Pi}_\pm^{\mathcal{C}}\,|\psi\rangle\right). \qquad \text{(B.2.31)} \end{aligned}$$

Hence, adhering to Eq. (B.2.24) is sufficient for the measurements of the criteria to preserve (anti)symmetry. To see that it is also a necessary condition, assume that a configuration $\{r_0, R_0\}$ and permutation $\sigma_0$ exist so that

$$\mathcal{C}(\sigma[\{r_0,R_0\}]) \neq \mathcal{C}(\{r_0,R_0\}). \qquad \text{(B.2.32)}$$

However, suppose for contradiction that the projectors related to $\mathcal{C}$ still map (anti)symmetrized states to (anti)symmetrized states. Under this assumption, consider a state $|\psi\rangle$ formed by the (anti)symmetrization of $|\psi(\{r_0,R_0\})\rangle$. By definition, this state has a non-zero overlap with $|\psi(\{r_0,R_0\})\rangle$ Writing the decomposition of the state in terms of positions,

$$|\psi\rangle = \sum_{r,R} \alpha_{\{r,R\}}\,|\psi(\{r,R\})\rangle\,, \qquad \text{(B.2.33)}$$

this is equivalent to the statement that $\left|\alpha_{\{r_0,R_0\}}\right|^2 > 0$.

Assume now without loss of generality that $\mathcal{C}(\{r_0,R_0\}) = 0$, and consider

$$U_{\sigma_0}\tilde{\Pi}_+^{\mathcal{C}}\,|\psi\rangle = \mathrm{sgn}(\sigma_{0,F})\tilde{\Pi}_+^{\mathcal{C}}\,|\psi\rangle \qquad \text{(B.2.34)}$$

which is fulfilled by the assumption that the projectors preserve (anti)symmetry. Since $\left(\tilde{\Pi}_+^{\mathcal{C}}\right)^2 =$

$\tilde{\Pi}_+^{\mathcal{C}}$[1], this implies

$$\tilde{\Pi}_+^{\mathcal{C}} U_{\sigma_0} \tilde{\Pi}_+^{\mathcal{C}} |\psi\rangle = \mathrm{sgn}(\sigma_{0,F}) \tilde{\Pi}_+^{\mathcal{C}} |\psi\rangle . \tag{B.2.35}$$

Therefore, the norms of the left-hand and right-hand sides above are also equal. Using the following sets

$$A = \{\{r, R\} : C(\{r, R\}) = 0\} \tag{B.2.36}$$

$$B_{\sigma_0} = \{\{r, R\} : C(\sigma_0[\{r, R\}]) = 0\} \tag{B.2.37}$$

this equality of norms can be written as

$$\sum_{\{r,R\} \in A \cap B_{\sigma_0}} \left|\alpha_{\{r,R\}}\right|^2 = \sum_{\{r,R\} \in A} \left|\alpha_{\{r,R\}}\right|^2 \tag{B.2.38}$$

Note that basic set theory implies that $A \cap B_{\sigma_0} \subseteq A$, meaning every term in the left-hand sum is also present in the right-hand sum. Subtracting these common terms, this implies

$$\sum_{\{r,R\} \in A \backslash B_{\sigma_0}} \left|\alpha_{\{r,R\}}\right|^2 = 0 \tag{B.2.39}$$

There can be no nonzero amplitude for any configurations in $A \backslash B_{\sigma_0}$. However, by assumption $\mathcal{C}(\{r_0, R_0\}) = 0$, meaning $\{r_0, R_0\} \in A$. Furthermore, Eq. (B.2.32) implies that $C(\sigma_0[\{r, R\}]) \neq 0$, meaning $\{r_0, R_0\} \notin B_{\sigma_0}$. Thus, $\{r_0, R_0\} \in A \backslash B_{\sigma_0}$ and $\left|\alpha_{\{r_0,R_0\}}\right|^2 > 0$. We have, in other words, arrived at a contradiction. For a criteria $\mathcal{C}$ that does not fulfill Eq. (B.2.24), configurations necessarily exist where measuring $\mathcal{C}$ breaks symmetrization.

Note that all arguments above apply equally to other groups of spatial transformations under which the wave function should be invariant (up to a phase), for instance, SO(3). They also may be extended to other operators that commute with the Hamiltonian, such as total spin. In general, one should construct criteria that are maximally invariant under trivial transformations. One will otherwise discard components that correspond to desired configurations, yielding artificially low success probabilities (e.g., 50% in the case of fully formed $H_2O_2$).

### B.2.3 Effect of Oracle Measurement for General Superpositions

In this section, we cover the effect of weak measurement of an oracle on a general mixed state. The goal is to elucidate the measurement procedure, the effects of the measurements, and the role of the parameter $\delta$ describing the strength of the measurement.

We recall the form of a general input state,

$$\rho = \sum_{j,k} \rho_{j,k} |\psi(\{r, R\}_j)\rangle\langle\psi(\{r, R\}_k)| \tag{B.2.40}$$

as well as access to a classical criterion $\mathcal{C}$ and a unitary implementation $U_{\mathcal{C}}$, as discussed in Sec. B.2.1. Further, we recall the definition of the set of 'successful states' marked by $A$ and 'unsuccessful

[1] To see this in the case of the tilded operators, compare their effects on states from the complete basis $\{|\psi(\{r, R\})\rangle\}$.

states' marked by $B$ from the main text as:

$$\begin{aligned} A &= \{j : \mathcal{C}(\{r, R\}_j) = 1\} \\ B &= \{j : \mathcal{C}(\{r, R\}_j) = 0\} \\ |\psi_j\rangle &:= |\psi(\{r, R\}_j)\rangle \end{aligned} \tag{B.2.41}$$

Then, the input state from above can be rewritten to account for the 'successful' and 'unsuccessful' subspaces as

$$\begin{aligned} \rho = &\sum_{j,k\in A} \rho_{j,k} |\psi_j\rangle\langle\psi_k| + \sum_{j,k\in B} \rho_{j,k} |\psi_j\rangle\langle\psi_k| \\ &+ \sum_{j\in A, k\in B} \left(\rho_{j,k} |\psi_j\rangle\langle\psi_k| + \rho_{k,j} |\psi_k\rangle\langle\psi_j|\right). \end{aligned} \tag{B.2.42}$$

Below, we will go through each step outlined in the main text in Section 2.2.4, discussing motivation and relevant considerations.

**Step 1: First application of $U_{\mathcal{C}}$**

This step aims to extract the information represented by the oracle into an ancilla for easy access. Using the definition of the sets $A$ and $B$, the state resulting from this step can be explicitly written as

$$\begin{aligned} U_{\mathcal{C}} \left(\rho \otimes |0\rangle\langle 0|\right) U_{\mathcal{C}}^{\dagger} = &\sum_{j,k\in A} \rho_{j,k} |\psi_j\rangle\langle\psi_k| \otimes |1\rangle\langle 1| + \sum_{j,k\in B} \rho_{j,k} |\psi_j\rangle\langle\psi_k| \otimes |0\rangle\langle 0| \\ &+ \sum_{j\in A, k\in B} \left(\rho_{j,k} |\psi_j\rangle\langle\psi_k| \otimes |1\rangle\langle 0| + \rho_{k,j} |\psi_k\rangle\langle\psi_j| \otimes |0\rangle\langle 1|\right). \end{aligned} \tag{B.2.43}$$

At this stage, a measurement of the ancilla qubit would project the state onto either the first or the second term in this sum. In the case of projection onto the first term (i.e., a measurement outcome of '1'), this projection would correspond to a heralded projection onto the desired set of states, as identified by the oracle. However, such a measurement would come with a risk of a projection onto the space that does not correspond to success. Thus, a full measurement risks projecting the state out of the desired space, and repeated measurements would risk a Zeno-effect-like freezing of the dynamics in the $B$ subspace.

**Step 2+3: Preparation for weak measurement**

To avoid the problems discussed above, this step *partially* extracts the information stored in the first ancilla into a second ancilla, then resets the first ancilla using a second application of $U_{\mathcal{C}}$. For ease of notation, we combine these two steps here and omit the first ancilla (now unentangled and

in the state $|0\rangle\langle 0|$) from the expression:

$$\begin{aligned}\rho_\delta = &\left(\cos(\delta)^2 \sum_{j,k\in A} \rho_{j,k} |\psi_j\rangle\langle\psi_k| + \sum_{j,k\in B} \rho_{j,k} |\psi_j\rangle\langle\psi_k| + \cos(\delta) \sum_{j\in A,k\in B} (\rho_{j,k} |\psi_j\rangle\langle\psi_k| + \rho_{k,j} |\psi_k\rangle\langle\psi_j|)\right) \otimes |0\rangle\langle 0| \\ &+ \sin(\delta)\cos(\delta) \sum_{j,k\in A} \rho_{j,k} |\psi_j\rangle\langle\psi_k| \otimes (|0\rangle\langle 1| + |1\rangle\langle 0|) \\ &+ \sin(\delta) \sum_{j\in A,k\in B} (\rho_{j,k} |\psi_j\rangle\langle\psi_k| \otimes |1\rangle\langle 0| + \rho_{k,j} |\psi_k\rangle\langle\psi_j| \otimes |0\rangle\langle 1|) \\ &+ \sin(\delta)^2 \sum_{j,k\in A} \rho_{j,k} |\psi_j\rangle\langle\psi_k| \otimes |1\rangle\langle 1| . \end{aligned} \tag{B.2.44}$$

Note that only the first and last of these expressions will be relevant once the ancilla is measured.

**Step 4: Measurement and heralding**

We are now ready to see the effect of a measurement of the second ancilla and to understand the role played by the parameter $\delta$. Consider first the projectors corresponding to the two measurement outcomes of the projective measurement:

$$\Pi_A = \mathbb{1} \otimes \left(\frac{\mathbb{1} - Z}{2}\right) \tag{B.2.45}$$

$$\Pi_B = \mathbb{1} \otimes \left(\frac{\mathbb{1} + Z}{2}\right) \tag{B.2.46}$$

Assume now that the measurement yields a '1' outcome. The probability of this event is

$$\begin{aligned} p_1 &= \mathrm{Tr}(\rho_\delta \Pi_A) = \mathrm{Tr}(\Pi_A \rho_\delta \Pi_A) \\ &= \sin(\delta)^2 \,\mathrm{Tr}\left(\sum_{j,k\in A} \rho_{j,k} |\psi_j\rangle\langle\psi_k|\right) \\ &= \sin(\delta)^2 \sum_{j\in A} \rho_{j,j} \\ &= \sin(\delta)^2 \, p_{\mathrm{suc}} , \end{aligned} \tag{B.2.47}$$

where $p_{\mathrm{suc}} = \sum_{j\in A} \rho_{j,j}$ is the probability of the system having transitioned to a state labeled by the criteria as successfully merged. Assuming $p_1$ is nonzero, the state after the measurement is given by

$$\begin{aligned} \rho_1 &= \frac{1}{p_1} \Pi_A \rho_\delta \Pi_A \\ &= \frac{1}{p_{\mathrm{suc}}} \sum_{j,k\in A} \rho_{j,k} |\psi_j\rangle\langle\psi_k| . \end{aligned} \tag{B.2.48}$$

Thus, this measurement outcome is the desirable one: it heralds that the state has been projected onto the desired subspace. Furthermore, the probability of getting this outcome depends both on the overlap of the input state with the desired subspace and the parameter $\delta$, with values of $\delta \approx \frac{\pi}{2}$ maximizing the probability.

Consider now instead the '0' outcome. The probability for this outcome is

$$p_0 = 1 - p_1 = 1 - \sin(\delta)^2 p_{\text{suc}} \tag{B.2.49}$$

and the post-measurement state is

$$\begin{aligned}\rho_0 &= \frac{1}{p_0}\Pi_B \rho_\delta \Pi_B \\ &= \frac{1}{p_0}\left(\cos(\delta)^2 \sum_{j,k\in A} \rho_{j,k} |\psi_j\rangle\langle\psi_k| + \sum_{j,k\in B} \rho_{j,k} |\psi_j\rangle\langle\psi_k| + \cos(\delta) \sum_{j\in A, k\in B} (\rho_{j,k} |\psi_j\rangle\langle\psi_k| + \rho_{k,j} |\psi_k\rangle\langle\psi_j|)\right).\end{aligned} \tag{B.2.50}$$

Note that this state closely resembles the input state from Eq. (B.2.42), except the components related to the $A$ subspace have decreased in magnitude by a factor of $\cos(\delta)$ and a renormalization by $1/p_0$ has occurred. To better understand this measurement-induced perturbation, we can define the following normalized states and coefficients

$$\rho_A = \rho_1 = \frac{1}{p_{\text{suc}}} \sum_{j,k\in A} \rho_{j,k} |\psi_j\rangle\langle\psi_k| \quad \rho_B = \frac{1}{1-p_{\text{suc}}} \sum_{j,k\in B} \rho_{j,k} |\psi_j\rangle\langle\psi_k| \tag{B.2.51}$$

$$\Lambda_A(\delta, p_{\text{suc}}) = \frac{\sin(\delta)^2 (1-p_{\text{suc}})\, p_{\text{suc}}}{(1-p_{\text{suc}}) + \cos(\delta)^2 p_{\text{suc}}} \tag{B.2.52}$$

$$\Lambda_B(\delta, p_{\text{suc}}) = \frac{\sin(\delta)^2 (1-p_{\text{suc}})\, p_{\text{suc}}}{1 - \sin(\delta)^2 p_{\text{suc}}} \tag{B.2.53}$$

$$\Lambda_C(\delta, p_{\text{suc}}) = \frac{1 - \cos(\delta) - \sin(\delta)^2 p_{\text{suc}}}{1 - \sin(\delta)^2 p_{\text{suc}}} \tag{B.2.54}$$

to rewrite the state as

$$\rho_0 = \rho - \Lambda_A(\delta, p_{\text{suc}})\, \rho_A + \Lambda_B(\delta, p_{\text{suc}})\, \rho_B - \Lambda_C(\delta, p_{\text{suc}}) \sum_{j\in A, k\in B} (\rho_{j,k} |\psi_j\rangle\langle\psi_k| + \rho_{k,j} |\psi_k\rangle\langle\psi_j|) \tag{B.2.55}$$

$$\begin{aligned}&\simeq \rho - \delta^2 (1-p_{\text{suc}})\, p_{\text{suc}}\, \rho_A + \delta^2 (1-p_{\text{suc}})\, p_{\text{suc}}\, \rho_B \\ &\quad - \delta^2 \left(\frac{1}{2} - p_{\text{suc}}\right) \sum_{j\in A, k\in B} (\rho_{j,k} |\psi_j\rangle\langle\psi_k| + \rho_{k,j} |\psi_k\rangle\langle\psi_j|) + O(\delta^4).\end{aligned} \tag{B.2.56}$$

Thus, receiving a measurement result of '0' implies that the state has been left mostly unchanged, except primarily for two effects: The desired components related to $\rho_A$ have decreased by an amount $\Lambda_A\, \rho_A$, while the unwanted components related to $\rho_B$ have increased by an amount $\Lambda_B\, \rho_B$. In other words, the measurement has caused a shift towards the subspace of undesired states, with the magnitude of the shift depending on $\delta$ and $p_{\text{suc}}$. From this, we see a trade-off in play when picking the parameter $\delta$. As shown above, the largest probability of successful heralded projection requires $\delta \approxeq \frac{\pi}{2}$, but in this case the desired $A$ part of the state is completely lost whenever the measurement outcome '0' occurs (see Eq. (B.2.50)). On the other hand, a small value of $\delta$ implies a

small probability of successful heralded projection but also a small state perturbation in the case of the outcome '0', with both scaling as $\delta^2$ in the small-$\delta$ limit. In this sense, $\delta$ represents the power of the measurement, with strong measurements yielding a higher chance of detecting a successful molecular merger but also a higher disruptive impact of the measurement on the state. Picking good schedules for adjusting $\delta$ has previously been studied in the context of Grover search [AH22; Yan+22; Miz09], and will likely also play a significant role in determining the performance of the simulation approaches presented here.

# Appendix C

# Supplementary Material for Section 3.1

## C.1 Variable names and conventions

We generally denote scalars using lower case, e.g. $c$, vectors using lower-case bold, e.g. $\mathbf{u}$, and matrices/operators using upper-case letters, such as $A$. Unless noted otherwise, norms correspond to the spectral or Euclidean 2-norm. The notation $[N]$ describes the set $\{1, 2, \ldots, N\}$.

- $\mathbf{x}$ – Coordinate in space.
- $t$ – Coordinate in time.
- $T$ – Final time.
- $\mathbf{u}$ – Solution vector for nonlinear ODE.
- $\mathbf{y}$ – Solution vector of system arising from Carleman linearisation, $\mathbf{y} = [\mathbf{u}, \mathbf{u}^{\otimes 2}, \ldots, \mathbf{u}^{\otimes N}, \ldots]^N$.
- $F_1, F_M$ – Terms appearing in nonlinear ODE in Problem 1, where $F_1$ is an operator representing a linear term and $F_M$ corresponds to a nonlinearity of the order of the subscript $M$.
- $N$ – Truncation number of Carleman linearisation.
- $\mathcal{A}, \mathcal{A}_N$ – System matrix after Carleman linearisation, subscript $N$ denotes truncated system; see Eq. (3.1.17).
- $A_j^{(i)}$ – Components of block-structure of Carleman matrix ($j$th row, $(i-1)$th diagonal). Defined in Eq. (3.1.15).
- $\varepsilon$ – Solution error.
- $\eta$ – Error in Carleman vector, defined in Lemmas 3.1.3 and 3.1.4.
- $\Delta$ – Laplacian operator.
- $L$, $L_k$ – Finite difference approximation of the Laplacian, up to "order" $k$, so that there is a central finite difference stencil with $2k + 1$ points.

- $D, c, b$ – Coefficients from PDE in Eq. (3.1.74); $D$ - diffusion, $c$ decay, $b$ nonlinearity.
- $n$ – Total number of gridpoints in the discretisation of the PDE example in Eq. (3.1.74).
- $d$ – Dimensionality of the PDE problem in Eq. (3.1.74).
- $\mathbb{I}$ – Identity matrix, indexing may clarify the dimension rather than number of qubits.
- $\lambda_j(\cdot)$ – Eigenvalue $j$ of matrix $(\cdot)$, whereas $\lambda_{(\cdot)}$ is the subnormalization factor of the block-encoding of $(\cdot)$.
- $\gamma$ – Rescaling factor introduced in Definition 3.1.1, $\gamma > 0$.
- $\widetilde{(\cdot)}$ – Rescaled quantities as defined in Definition 3.1.1.
- $R$ – Ratio of strength of nonlinearity over decay, Eq. (3.1.5).
- $f_{j,k,M}$ – Function to express tighter bound for Carlemann errors, defined in Lemma 3.1.4.
- $\Omega_k$ – Carleman error intervals, used in Lemma 3.1.4.
- $\eta$ – Error due to Carleman linearisation, defined in Lemma 3.1.3.
- $s(\cdot)$ – Sparsity of quantity $(\cdot)$.

## C.2 Proofs of bounds on error due to Carleman linearisation

### C.2.1 Proof of bound on the full vector of errors due to Carleman linearisation

Here, we present the proof of Lemma 3.1.3.

*Proof.* We define the error due to Carleman truncation of order $N$, for any $j \in [N]$, including a rescaling by $\gamma > 0$, as

$$\begin{aligned} \eta_j(t) &\coloneqq \widetilde{\mathbf{u}}^{\otimes j} - \widetilde{\mathbf{y}}_j \\ &= \frac{\mathbf{u}^{\otimes j} - \mathbf{y}_j}{\gamma^j}. \end{aligned} \tag{C.2.1}$$

Here, $\mathbf{u}$ is the exact solution of the nonlinear ODE, and $\mathbf{y}_j$ are the components of the solution vector using the truncated Carleman approximation. Therefore, $\eta_j$ corresponds to the error arising from the Carlemann linearisation at finite order, but with the rescaling $\gamma$. This definition is similar to that in [Liu+21] (see the proof of Lemma 1 in the Supplementary Information of that work), except we are including the rescaling. By writing $\mathbf{U}$ as a vector that has the components $\widetilde{\mathbf{U}}_j = \widetilde{\mathbf{u}}^{\otimes j}$, we have from Eq. (C.2.1) that

$$\begin{aligned} \frac{\mathrm{d}\eta_j(t)}{\mathrm{d}t} &= \left(\widetilde{\mathcal{A}}\widetilde{\mathbf{U}}\right)_j - \left(\widetilde{\mathcal{A}}_N\widetilde{\mathbf{y}}\right)_j \\ &= A_j^{(1)}\widetilde{\mathbf{u}}^{\otimes j} + \widetilde{A}_{j+M-1}^{(M)}\widetilde{\mathbf{u}}^{\otimes j+M-1} - A_j^{(1)}\widetilde{\mathbf{y}}_j - \widetilde{A}_{j+M-1}^{(M)}\widetilde{\mathbf{y}}_{j+M-1}. \end{aligned} \tag{C.2.2}$$

Notice that in the equation above we have $\widetilde{\mathcal{A}}$ as the rescaled form of $\mathcal{A}$, the infinite-dimensional matrix generated from the Carleman linearisation, as explained in Section 3.1.4, and $\widetilde{\mathcal{A}}_N = \gamma^{M-1}\mathcal{A}_N$ as the truncated matrix as given in Eq. (3.1.32). Because we are taking the difference between the action of an infinite-dimensional matrix $\widetilde{\mathcal{A}}$ and a truncated one $\widetilde{\mathcal{A}}_N$, we need different treatment for values of $j$ below and above that truncation. For $j \leq N - M + 1$ we are below the truncation, and in Eq. (C.2.2) we can use $\eta_j(t) = \widetilde{\mathbf{u}}^{\otimes j} - \widetilde{\mathbf{y}}_j$ to give

$$\frac{\mathrm{d}\eta_j}{\mathrm{d}t} = A_j^{(1)}\eta_j + \widetilde{A}_{j+M-1}^{(M)}\eta_{j+M-1}. \tag{C.2.3}$$

Then, using $\widetilde{\mathcal{A}}_j = \gamma^{M-1}\mathcal{A}_j$ gives

$$\frac{\mathrm{d}\eta_j}{\mathrm{d}t} = A_j^{(1)}\eta_j + \gamma^{M-1}A_{j+M-1}^{(M)}\eta_{j+M-1}. \tag{C.2.4}$$

Now, for $j > N - M + 1$, we have $j + M - 1 > N$, and so $\widetilde{\mathbf{y}}_{j+M-1}$ is past the end of the vector and should be taken to be zero. This implies that we obtain

$$\frac{\mathrm{d}\eta_j}{\mathrm{d}t} = A_j^{(1)}\eta_j + \widetilde{A}_{j+M-1}^{(M)}\widetilde{\mathbf{u}}_{j+M-1}. \tag{C.2.5}$$

Now, using $\widetilde{\mathcal{A}}_j = \gamma^{M-1}\mathcal{A}_j$ and $\widetilde{\mathbf{u}}_{j+M-1} = \gamma^{-(j+M-1)}\mathbf{u}_{j+M-1}$ we get for $j > N - M + 1$

$$\begin{aligned}\frac{\mathrm{d}\eta_j}{\mathrm{d}t} &= A_j^{(1)}\eta_j + \gamma^{M-1}A_{j+M-1}^{(M)}\gamma^{-(j+M-1)}\mathbf{u}^{\otimes(j+M-1)} \\ &= A_j^{(1)}\eta_j + \gamma^{-j}A_{j+M-1}^{(M)}\mathbf{u}^{\otimes(j+M-1)}.\end{aligned} \tag{C.2.6}$$

Therefore, we end up with

$$\begin{aligned}\frac{\mathrm{d}\eta_j}{\mathrm{d}t} &= A_j^{(1)}\eta_j + \gamma^{M-1}A_{j+M-1}^{(M)}\eta_{j+M-1}, \quad j \in [N - M + 1] \\ \frac{\mathrm{d}\eta_j}{\mathrm{d}t} &= A_j^{(1)}\eta_j + \gamma^{-j}A_{j+M-1}^{(M)}\mathbf{u}^{\otimes(j+M-1)}, \quad j \in \{N - M + 2, \ldots, N\},\end{aligned} \tag{C.2.7}$$

Notice that in the final form above, we left the equations without rescaling notation to show the explicit dependence of $\gamma$ in the error equation. In a more compact form, the equation above can be written as

$$\frac{\mathrm{d}\eta}{\mathrm{d}t} = \widetilde{\mathcal{A}}_N\eta + \mathbf{b}, \tag{C.2.8}$$

which has the following block matrix structure

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\eta_1\\ \eta_2\\ \vdots\\ \eta_{N-1}\\ \eta_N\end{bmatrix} = \underbrace{\begin{bmatrix} A_1^{(1)} & 0 & \cdots & 0 & 0 & \gamma^{M-1}A_M^{(M)} & 0 & 0 \\ 0 & A_2^{(1)} & \cdots & 0 & 0 & 0 & \gamma^{M-1}A_{M+1}^{(M)} & 0 \\ \vdots & & & \vdots & \vdots & & \cdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & A_{N-1}^{(1)} & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & A_N^{(1)} \end{bmatrix}}_{\widetilde{\mathcal{A}}_N} \cdot \begin{bmatrix}\eta_1\\ \eta_2\\ \vdots\\ \eta_{N-1}\\ \eta_N\end{bmatrix} + \underbrace{\begin{bmatrix}\mathbf{b}_1\\ \mathbf{b}_2\\ \vdots\\ \mathbf{b}_{N-1}\\ \mathbf{b}_N\end{bmatrix}}_{\mathbf{b}}, \tag{C.2.9}$$

where

$$\begin{aligned}\mathbf{b}_j &= \mathbf{0}^{\otimes j}, \quad j \in [N-M+1] \\ \mathbf{b}_j &= \gamma^{-j} A^{(M)}_{j+M-1} \mathbf{u}^{\otimes(j+M-1)} \quad j \in \{N-M+2, \dots, N\}.\end{aligned} \tag{C.2.10}$$

and $\mathbf{0}$ represents a vector with all entries equal to zero with the same dimension as $\mathbf{u}$. The blocks $A^{(1)}_j$ and $A^{(M)}_{M+j-1}$ are given in Eq. (3.1.15). Similarly to Eq. (35) in the Supplementary Information of [Liu+21], we obtain by considering the derivative of $\|\eta\|$ that

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}(\eta^\dagger \eta) &= \frac{\mathrm{d}\eta^\dagger}{\mathrm{d}t}\eta + \eta^\dagger \frac{\mathrm{d}\eta}{\mathrm{d}t} \\ &= \eta^\dagger \left(\widetilde{\mathcal{A}}_N + \widetilde{\mathcal{A}}_N^\dagger\right)\eta + \eta^\dagger \mathbf{b} + \mathbf{b}^\dagger \eta.\end{aligned} \tag{C.2.11}$$

Using Gershgorin's circle theorem [HJ12; FV62; van79], the maximum eigenvalue of $\widetilde{\mathcal{A}}_N + \widetilde{\mathcal{A}}_N^\dagger$ is at most (see the discussion above for Eq. (3.1.50))

$$2j\lambda_0 + (2j - M + 1)\gamma^{M-1}\|F_M\|\,. \tag{C.2.12}$$

We then have,

$$\eta^\dagger \left(\widetilde{\mathcal{A}}_N + \widetilde{\mathcal{A}}_N^\dagger\right)\eta \leq \left(2N\lambda_0 + (2N - M + 1)\gamma^{M-1}\|F_M\|\right)\|\eta\|^2. \tag{C.2.13}$$

The remaining term to bound in Eq. (C.2.11) is $\eta^\dagger \mathbf{b} + \mathbf{b}^\dagger \eta$, which has the upper bound

$$\begin{aligned}\eta^\dagger \mathbf{b} + \mathbf{b}^\dagger \eta &\leq 2\|\mathbf{b}\|\|\eta\| \\ &= 2\sum_{j=N-M+2}^{N} \gamma^{-j}\|A^{(M)}_{j+M-1}\| \cdot \|\mathbf{u}^{\otimes(j+M-1)}\| \cdot \|\eta\| \\ &= 2\|F_M\| \sum_{j=N-M+2}^{N} j\gamma^{-j}\|\mathbf{u}^{\otimes(j+M-1)}\| \cdot \|\eta\|\,,\end{aligned} \tag{C.2.14}$$

where we used Eq. (C.2.10) for the vector norm of $\mathbf{b}$, followed by using Eq. (3.1.15). Now by taking the maximum value that $j$ can assume in the summation above gives

$$\eta^\dagger \mathbf{b} + \mathbf{b}^\dagger \eta \leq 2N\|F_M\| \sum_{j=N-M+2}^{N} \gamma^{-j}\|\mathbf{u}^{\otimes(j+M-1)}\| \cdot \|\eta\|. \tag{C.2.15}$$

As stated in the theorem, we are dealing with dissipative problems so $\|\mathbf{u}(t)\| \leq \|\mathbf{u}_{\mathrm{in}}\|$ for $t > 0$, and we also take $\gamma = \|\mathbf{u}_{\mathrm{in}}\|$. Therefore

$$\eta^\dagger \mathbf{b} + \mathbf{b}^\dagger \eta \leq 2N(M-1)\|F_M\|\|\mathbf{u}_{\mathrm{in}}\|^{M-1}\|\eta\|. \tag{C.2.16}$$

Returning to Eq. (C.2.11) together with the fact that $\frac{\mathrm{d}\|\eta\|}{\mathrm{d}t} = (2\|\eta\|)^{-1}\frac{\mathrm{d}\|\eta\|^2}{\mathrm{d}t}$, we have

$$\frac{\mathrm{d}\|\eta\|}{\mathrm{d}t} = (2\|\eta\|)^{-1}\left[\eta^\dagger \left(\mathcal{A}_N + \mathcal{A}_N^\dagger\right)\eta + \eta^\dagger \mathbf{b} + \mathbf{b}^\dagger \eta\right]. \tag{C.2.17}$$

We can now combine the results from Eq. (C.2.13) and Eq. (C.2.16) in the equation above to get

$$\frac{\mathrm{d}\|\eta\|}{\mathrm{d}t} \leq N\left(\lambda_0 + \gamma^{M-1}\|F_M\|\right)\|\eta\| + N(M-1)\|F_M\|\|\mathbf{u}_{\mathrm{in}}\|^{M-1}. \tag{C.2.18}$$

We know that if we have a differential equation given by

$$\frac{\mathrm{d}\|\eta\|}{\mathrm{d}t} = A\|\eta\| + B, \tag{C.2.19}$$

where $\|\eta(0)\| = 0$, its solution is given by

$$\|\eta(t)\| = \frac{B}{A}\left(1 - e^{At}\right). \tag{C.2.20}$$

Since in our case we have

$$\begin{aligned} A &= N\left(\lambda_0 + \gamma^{M-1}\|F_M\|\right) \\ B &= N(M-1)\|F_M\|\|\mathbf{u}_{\mathrm{in}}\|^{M-1}, \end{aligned} \tag{C.2.21}$$

we get

$$\|\eta(t)\| \leq (M-1)\|F_M\|\ \|\mathbf{u}_{\mathrm{in}}\|^{M-1}\frac{1 - e^{N(\lambda_0+\gamma^{M-1}\|F_M\|)t}}{|\lambda_0 + \gamma^{M-1}\|F_M\||}. \tag{C.2.22}$$

This concludes the proof as $\|\eta(t)\| \geq \|\eta_j(t)\|$ for all $t$. □

### C.2.2 Proof of bound on components of the vector of errors due to Carleman linearisation

In this section we present the proof to Lemma 3.1.4. For this proof we use the definition of $f_{j,k,M}(|\lambda_0|t)$ as

$$\begin{aligned} f_{j,k,M}(\tau) := \left[\prod_{\ell=0}^{k-1}(\ell M - \ell + j)\right]\int_0^{\tau} d\tau_1\, e^{-j(\tau-\tau_1)}\int_0^{\tau_1} d\tau_2\, e^{-[(M-1)+j](\tau_1-\tau_2)}\int_0^{\tau_2} d\tau_3\, e^{-[2(M-1)+j](\tau_2-\tau_3)}\cdots \\ \cdots\int_0^{\tau_{k-1}} d\tau_k\, e^{-[(k-1)(M-1)+j](\tau_{k-1}-\tau_k)}, \end{aligned} \tag{C.2.23}$$

This function has the closed-form expression given in Lemma 3.1.4, which is proven in Theorem C.2.1 below. We plot these formulae for the example $M = 2$ in Fig. C.1. The blue curve on the left is the simple upper bound, so we can see that our expression gives significantly tighter bounds.

**Theorem C.2.1.** *The function $f_{j,k,M}(\tau)$ defined by Eq.* (C.2.23) *for real $\tau \geq 0$ and natural numbers $j \geq 1$, $k \geq 1$, and $M \geq 2$, satisfies*

$$f_{j,k,M}(\tau) = 1 - \frac{(M-1)\,\Gamma(k + j/(M-1))}{(k-1)!\,\Gamma(j/(M-1))}\sum_{\ell=0}^{k-1}(-1)^{\ell}\binom{k-1}{\ell}\frac{e^{-(\ell M-\ell+j)\tau}}{\ell M - \ell + j}. \tag{C.2.24}$$

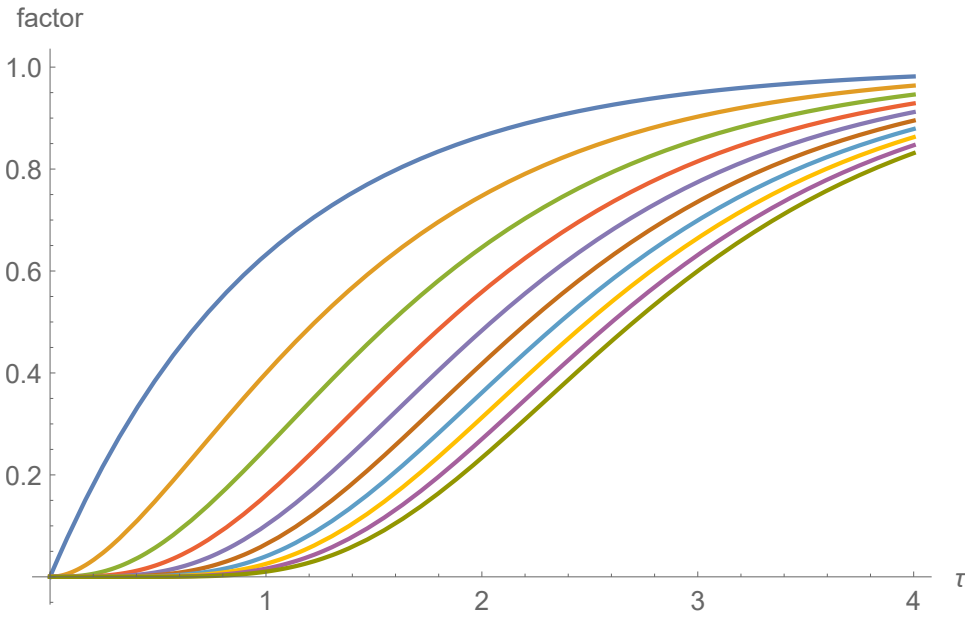



Figure C.1: The factors in the upper bound $f_{j,k,M}(\tau)$ with $j = 1$ and $M = 2$. The lines are for $k$ equal to 1 to 10 ordered from left to right. As we show, $f_{j,k,M}(\tau)$ is monotonically decreasing with $k$.

*Proof.* We first regroup the exponentials as

$$f_{j,k,M}(\tau) = \left[\prod_{\ell=0}^{k-1}(\ell M - \ell + j)\right] e^{-j\tau} \int_0^\tau d\tau_1\, e^{-(M-1)\tau_1} \int_0^{\tau_1} d\tau_2\, e^{-(M-1)\tau_2} \int_0^{\tau_2} d\tau_3\, e^{-(M-1)\tau_3} \cdots$$
$$\cdots \int_0^{\tau_{k-1}} d\tau_k\, e^{[(k-1)(M-1)+j]\tau_n}\,. \tag{C.2.25}$$

The integral is over values of $\tau_\ell$ satisfying $\tau \geq \tau_1 \geq \tau_2 \geq \cdots \geq \tau_k$. We can exchange the order of the integrals in the standard way so they are given as

$$f_{j,k,M}(\tau) = \left[\prod_{\ell=0}^{k-1}(\ell M - \ell + j)\right] e^{-j\tau} \int_0^\tau d\tau_k\, e^{[(k-1)(M-1)+j]\tau_k} \int_{\tau_k}^\tau d\tau_{k-1}\, e^{-(M-1)\tau_{k-1}} \cdots$$
$$\cdots \int_{\tau_3}^\tau d\tau_2\, e^{-(M-1)\tau_2} \int_{\tau_2}^\tau d\tau_1\, e^{-(M-1)\tau_1}\,. \tag{C.2.26}$$

The advantage of this form is that we have a sequence of integrals of the same form, so we can give a general expression that can be proven by induction. In particular, let us define the repeated integral

$$g_{k,M}(\tau_k, \tau) := \int_{\tau_k}^\tau d\tau_{k-1}\, e^{-(M-1)\tau_{k-1}} \cdots \int_{\tau_3}^\tau d\tau_2\, e^{-(M-1)\tau_2} \int_{\tau_2}^\tau d\tau_1\, e^{-(M-1)\tau_1}\,. \tag{C.2.27}$$

The closed-form expression will be shown to be, for $k \geq 1$,

$$g_{k,M}(\tau_k, \tau) = \frac{[e^{-(M-1)\tau_k} - e^{-(M-1)\tau}]^{k-1}}{(k-1)!(M-1)^{k-1}}\,. \tag{C.2.28}$$

In the case of $k = 1$ the definition of $g_{k,M}(\tau_k, \tau)$ can be just taken to be 1 (no integrals), and Eq. (C.2.28) holds. In the simple case of $k = 2$ we have, from the definition

$$\begin{aligned} g_{2,M}(\tau_2, \tau) &= \int_{\tau_2}^{\tau} d\tau_1 \, e^{-(M-1)\tau_1} \\ &= \frac{1}{M-1} \left[ e^{-(M-1)\tau_2} - e^{-(M-1)\tau} \right] . \end{aligned} \tag{C.2.29}$$

This also agrees with the claimed expression. More generally, if Eq. (C.2.28) holds for $k$, then we obtain

$$\begin{aligned} g_{k+1,M}(\tau_{k+1}, \tau) &= \int_{\tau_{k+1}}^{\tau} d\tau_k \, e^{-(M-1)\tau_k} g_{k,M}(\tau_k, \tau) \\ &= \int_{\tau_{k+1}}^{\tau} d\tau_k \, e^{-(M-1)\tau_k} \frac{\left[ e^{-(M-1)\tau_k} - e^{-(M-1)\tau} \right]^{k-1}}{(k-1)!(M-1)^{k-1}} \\ &= \left[ -\frac{\left[ e^{-(M-1)\tau_k} - e^{-(M-1)\tau} \right]^{k}}{k!(M-1)^{k}} \right]_{\tau_{k+1}}^{\tau} \\ &= \frac{\left[ e^{-(M-1)\tau_{k+1}} - e^{-(M-1)\tau} \right]^{k}}{k!(M-1)^{k}} . \end{aligned} \tag{C.2.30}$$

Thus Eq. (C.2.28) holds for $k + 1$, and must hold for all $k \geq 1$ by induction.

Now to obtain the expression for $f_{j,k,M}(\tau)$, we use

$$f_{j,k,M}(\tau) = \left[ \prod_{\ell=0}^{k-1} (\ell M - \ell + j) \right] e^{-j\tau} \int_0^{\tau} d\tau_k \, e^{[(k-1)(M-1)+j]\tau_n} g_{k,M}(\tau_k, \tau) . \tag{C.2.31}$$

Expanding the expression for $g_{k,M}(\tau_k, \tau)$ involving a power into a sum gives

$$\begin{aligned} f_{n,M}(\tau) &= \frac{\prod_{\ell=0}^{k-1}(\ell M - \ell + j)}{(k-1)!(M-1)^{k-1}} e^{-j\tau} \int_0^{\tau} d\tau_k \, e^{[(k-1)(M-1)+j]\tau_k} \sum_{\ell=0}^{k-1} (-1)^{\ell} \binom{k-1}{\ell} e^{-(k-1-\ell)(M-1)\tau_k} e^{-\ell(M-1)\tau} \\ &= \frac{\prod_{\ell=0}^{k-1}(\ell M - \ell + j)}{(k-1)!(M-1)^{k-1}} \int_0^{\tau} d\tau_n \sum_{\ell=0}^{k-1} (-1)^{\ell} \binom{k-1}{\ell} e^{-(\ell M - \ell + j)(\tau - \tau_k)} \\ &= \frac{\prod_{\ell=0}^{k-1}(\ell M - \ell + j)}{(k-1)!(M-1)^{k-1}} \sum_{\ell=0}^{k-1} (-1)^{\ell} \binom{k-1}{\ell} \left[ -\frac{e^{-(\ell M - \ell + j)(\tau - \tau_k)}}{(\ell M - \ell + j)} \right]_0^{\tau} \\ &= \frac{\prod_{\ell=0}^{k-1}(\ell M - \ell + j)}{(k-1)!(M-1)^{k-1}} \sum_{\ell=0}^{k-1} (-1)^{\ell} \binom{k-1}{\ell} \frac{1 - e^{-(\ell M - \ell + j)\tau}}{\ell M - \ell + j} . \end{aligned} \tag{C.2.32}$$

Next we aim to show that

$$\frac{\prod_{\ell=0}^{k-1}(\ell M - \ell + j)}{(M-1)^{k}} = \frac{\Gamma(k + j/(M-1))}{\Gamma(j/(M-1))} . \tag{C.2.33}$$

First, it is obvious that

$$\frac{\prod_{\ell=0}^{k-1}(\ell M - \ell + j)}{(M-1)^{k}} = \prod_{\ell=0}^{k-1} (\ell + j/(M-1)) . \tag{C.2.34}$$

Now for $k = 0$ both sides of Eq. (C.2.33) are equal to 1, because there are no factors in the product,

and $k + j/(M-1) = j/(M-1)$ so the result is trivial. If Eq. (C.2.33) is correct for some $k$, we obtain

$$\begin{aligned}\frac{\Gamma(k+1+j/(M-1))}{\Gamma(j/(M-1))} &= \frac{\Gamma(k+1+j/(M-1))}{\Gamma(k+j/(M-1))}\frac{\Gamma(k+j/(M-1))}{\Gamma(j/(M-1))} \\ &= (k+j/(M-1))\prod_{\ell=0}^{k-1}(\ell+j/(M-1)) \\ &= \prod_{\ell=0}^{k}(\ell+j/(M-1))\,. \end{aligned} \tag{C.2.35}$$

In the second line we have used $\Gamma(x+1) = x\Gamma(x)$ and assumed Eq. (C.2.33) holds for $k$. That means Eq. (C.2.33) holds for all $k \geq 0$ by induction. Thus we can replace the product with the expression in terms of gamma functions to show that

$$f_{j,k,M}(\tau) = \frac{(M-1)\,\Gamma(k+j/(M-1))}{(k-1)!\,\Gamma(j/(M-1))}\sum_{\ell=0}^{k-1}(-1)^{\ell}\binom{k-1}{\ell}\frac{1-e^{-(\ell M-\ell+j)\tau}}{\ell M-\ell+j}\,. \tag{C.2.36}$$

Next we show that $f_{j,k,M}(\tau)$ is equal to 1 in the limit $\tau \to \infty$. If we use shifted variables of integration $\tilde{\tau}_\ell = \tau_\ell - \tau$, then the definition of $f_{j,k,M}(\tau)$ gives

$$\begin{aligned} f_{j,k,M}(\tau) = \left[\prod_{\ell=0}^{k-1}(\ell M-\ell+j)\right]\int_{-\tau}^{0} d\tilde{\tau}_1\, e^{j\tilde{\tau}_1}\int_{-\tau}^{\tilde{\tau}_1} d\tilde{\tau}_2\, e^{-(M-1+j)(\tilde{\tau}_1-\tilde{\tau}_2)}\int_{-\tau}^{\tilde{\tau}_2} d\tilde{\tau}_3\, e^{-[2(M-1)+j](\tilde{\tau}_2-\tilde{\tau}_3)}\cdots \\ \cdots\int_{-\tau}^{\tilde{\tau}_{n-1}} d\tilde{\tau}_k\, e^{-[(k-1)(M-1)+j](\tilde{\tau}_{k-1}-\tilde{\tau}_k)}\,. \end{aligned} \tag{C.2.37}$$

The limit $\tau \to \infty$ then gives

$$\begin{aligned} \lim_{\tau\to\infty} f_{j,k,M}(\tau) = \left[\prod_{\ell=0}^{k-1}(\ell M-\ell+j)\right]\int_{-\infty}^{0} d\tilde{\tau}_1\, e^{j\tilde{\tau}_1}\int_{-\infty}^{\tilde{\tau}_1} d\tilde{\tau}_2\, e^{-(M-1+j)(\tilde{\tau}_1-\tilde{\tau}_2)}\int_{-\infty}^{\tilde{\tau}_2} d\tilde{\tau}_3\, e^{-[2(M-1)+j](\tilde{\tau}_2-\tilde{\tau}_3)}\cdots \\ \cdots\int_{-\infty}^{\tilde{\tau}_{k-1}} d\tilde{\tau}_k\, e^{-[(k-1)(M-1)+1](\tilde{\tau}_{k-1}-\tilde{\tau}_k)}\,. \end{aligned} \tag{C.2.38}$$

Then it is easy to see that each successive integral gives division by one of the factors in the product, so the overall expression is equal to 1.

Therefore, since the negative exponentials in our expression for $f_{j,k,M}(\tau)$ approach zero in the limit $\tau \to \infty$, the constant term must be equal to 1, and we can give $f_{j,k,M}(\tau)$ as

$$f_{j,k,M}(\tau) = 1 - \frac{(M-1)\Gamma(k+j/(M-1))}{(k-1)!\,\Gamma(j/(M-1))}\sum_{\ell=0}^{k-1}(-1)^{\ell}\binom{k-1}{\ell}\frac{e^{-(\ell M-\ell+j)\tau}}{\ell M-\ell+j}\,. \tag{C.2.39}$$

□

For the use in our derivations, note that $f_{j,k,M}(\tau)$ satisfies the recurrence relation (which is obvious from its definition)

$$f_{j,k,M}(\tau) = j\int_0^{\tau} d\tau_1\, e^{-j(\tau-\tau_1)} f_{M-1+j,k-1,M}(\tau_1)\,. \tag{C.2.40}$$

Another useful property is that $f_{j,k,M}(\tau) \le f_{j,k-1,M}(\tau)$, so $f_{j,k,M}(\tau)$ is monotonically decreasing with $k$. This can be seen using Eq. (C.2.37), and upper bounding it by taking the limit as the last integral goes to infinity as

$$\begin{aligned} f_{j,k,M}(\tau) &\le \left[\prod_{\ell=0}^{k-1}(\ell M - \ell + j)\right] \int_{-\tau}^{0} d\tilde{\tau}_1\, e^{j\tilde{\tau}_1} \int_{-\tau}^{\tilde{\tau}_1} d\tilde{\tau}_2\, e^{-(M-1+j)(\tilde{\tau}_1-\tilde{\tau}_2)} \int_{-\tau}^{\tilde{\tau}_2} d\tilde{\tau}_3\, e^{-[2(M-1)+j](\tilde{\tau}_2-\tilde{\tau}_3)} \dots \\ &\quad \dots \int_{-\infty}^{\tilde{\tau}_{k-1}} d\tilde{\tau}_k\, e^{-[(k-1)(M-1)+j](\tilde{\tau}_{k-1}-\tilde{\tau}_k)} \\ &\le \left[\prod_{\ell=0}^{k-2}(\ell M - \ell + j)\right] \int_{-\tau}^{0} d\tilde{\tau}_1\, e^{j\tilde{\tau}_1} \int_{-\tau}^{\tilde{\tau}_1} d\tilde{\tau}_2\, e^{-(M-1+j)(\tilde{\tau}_1-\tilde{\tau}_2)} \int_{-\tau}^{\tilde{\tau}_2} d\tilde{\tau}_3\, e^{-[2(M-1)+j](\tilde{\tau}_2-\tilde{\tau}_3)} \dots \\ &\quad \dots \int_{-\tau}^{\tilde{\tau}_{k-2}} d\tilde{\tau}_{k-1}\, e^{-[(k-2)(M-1)+j](\tilde{\tau}_{k-2}-\tilde{\tau}_{k-1})} \\ &= f_{j,k-1,M}(\tau)\,. \end{aligned} \tag{C.2.41}$$

Now we proceed to the proof of Lemma 3.1.4.

*Proof.* We recall from Eq. (C.2.7)

$$\frac{\mathrm{d}\eta_j}{\mathrm{d}t} = A_j^{(1)}\eta_j + \gamma^{M-1}A_{j+M-1}^{(M)}\eta_{j+M-1}, \quad j \in [N-M+1] \tag{C.2.42}$$

$$\frac{\mathrm{d}\eta_j}{\mathrm{d}t} = A_j^{(1)}\eta_j + \gamma^{-j}A_{j+M-1}^{(M)}\mathbf{u}^{\otimes(j+M-1)}, \quad j \in \{N-M+2,\dots,N\}. \tag{C.2.43}$$

The method to bound the norms of $\eta_j$ is to first use the second line (C.2.43) for $j \in \Omega_1$ with

$$\Omega_1 \coloneqq \{N-M+2,\dots,N\}. \tag{C.2.44}$$

These bounds can be derived just using that value of $j$, since the equation does not depend on $\eta_j$ for any other values of $j$. For smaller values of $j$ we need to use the first line in Eq. (C.2.42), but that depends on $\eta_{j+M-1}$. If we have $j \in \Omega_2$ for

$$\Omega_2 \coloneqq \{N-2M+3,\cdots,N-M+1\}, \tag{C.2.45}$$

then $j+M-1 \in \Omega_1$. We can therefore use the bound derived for $j \in \Omega_1$ to bound the second term in Eq. (C.2.42) and thereby derive the bound on $\eta_j$ for $j \in \Omega_1$.

Then for $j$ less than $N-2M+3$ we can use the bounds on $\eta_j$ for $j \in \Omega_2$, and so forth, to eventually derive a bound on $\eta_1$. In particular we define

$$\Omega_k \coloneqq \{N-k(M-1)+1,\dots,N-(k-1)(M-1)\}, \tag{C.2.46}$$

for $k = 1,2,\cdots,\lceil\frac{N}{M-1}\rceil$. Then we work backwards in steps of $M-1$ to derive bounds on $\eta_j$ for $j \in \Omega_k$ using the bound on $\eta_{j+M-1}$ for $j+M-1 \in \Omega_{k-1}$. In particular, $1 \in \Omega_k$ for $k = \lceil\frac{N}{M-1}\rceil$. This can be seen from the extremal cases; first, if $N$ is a multiple of $M-1$, then $k = N/(M-1)$, so

$$N-k(M-1)+1 = N-[N/(M-1)](M-1)+1 = 1\,. \tag{C.2.47}$$

The other extremal case is where $N-1$ is a multiple of $M-1$, so the ceiling function gives the maximal rounding up of $N/(M-1)$. Then $k-1=(N-1)/(M-1)$, and

$$N-(k-1)(M-1)=N-[(N-1)/(M-1)](M-1)=1\,. \tag{C.2.48}$$

By deriving bounds on $\eta_j$ in the sequence of $\Omega_k$ sets for $k$ from 1 to $\lceil\frac{N}{M-1}\rceil$ we are therefore able to provide the bound on $\eta_1$.

More specifically, let us start with $j\in\Omega_1$, in which case

$$\eta_j(t)=\gamma^{-j}\int_0^t e^{A_j^{(1)}(t-s_0)}A_{j+M-1}^{(M)}\mathbf{u}^{\otimes(j+M-1)}\frac{\mathrm{d}}{\mathrm{d}s_0}\,. \tag{C.2.49}$$

Now for purely dissipative problems, as displayed in Eq. (3.1.5), we have $\|e^{A_j^{(1)}t}\|\le e^{j\lambda_0 t}$. In addition, $\|A_{j+M-1}^{(M)}\|=j\|F_M\|$ and $\|\mathbf{u}(t)\|\le\|\mathbf{u}_{\rm in}\|$, so

$$\begin{aligned}\|\eta_j(t)\|&\le j\gamma^{-j}\|F_M\|\|\mathbf{u}_{\rm in}\|^{j+M-1}\int_0^t e^{j\lambda_0(t-s_0)}\frac{\mathrm{d}}{\mathrm{d}s_0}\\&=\gamma^{-j}\frac{\|F_M\|\|\mathbf{u}_{\rm in}\|^{j+M-1}}{|\lambda_0|}\left(1-e^{j\lambda_0 t}\right).\end{aligned} \tag{C.2.50}$$

For the case $k=1$, our formula for $f_{j,k,M}$ gives

$$f_{j,1,M}(\tau)=1-e^{-j\tau}, \tag{C.2.51}$$

and so we find that for $k=1$ we obtain

$$\|\eta_j(t)\|\le\left(\frac{\|\mathbf{u}_{\rm in}\|}{\gamma}\right)^j\frac{\|F_M\|\|\mathbf{u}_{\rm in}\|^{M-1}}{|\lambda_0|}f_{j,1,M}\left(|\lambda_0|t\right). \tag{C.2.52}$$

Now for more general $j\in\Omega_k$, we will show

$$\|\eta_j(t)\|\le\left(\frac{\|\mathbf{u}_{\rm in}\|}{\gamma}\right)^j\left(\frac{\|F_M\|\|\mathbf{u}_{\rm in}\|^{M-1}}{|\lambda_0|}\right)^k f_{j,k,M}\left(|\lambda_0|t\right) \tag{C.2.53}$$

We have already shown this for $k=1$ where $j\in\Omega_1$, so we just need to show the iteration step to show the general result by induction. For $j$ smaller than $N-M+2$, we use Eq. (C.2.42), and Eq. (C.2.53) for $\eta_{j+M-1}$. That is,

$$\frac{\mathrm{d}\eta_j}{\mathrm{d}t}=A_j^{(1)}\eta_j+\gamma^{M-1}A_{j+M-1}^{(M)}\eta_{j+M-1}\,, \tag{C.2.54}$$

leads to

$$\eta_j(t)=\gamma^{M-1}\int_0^t e^{A_j^{(1)}(t-s_1)}A_{j+M-1}^{(M)}\eta_{j+M-1}(s_1)\frac{\mathrm{d}}{\mathrm{d}s_1}\,. \tag{C.2.55}$$

To upper bound the component $\eta_j(t)$ with $j\in\Omega_k$, we use the bound in Eq. (C.2.53) for $j+M-1\in\Omega_{k-1}$. (That is, we are assuming the expression holds for $k-1$ in order to derive it for $k$.) That

yields the upper bound

$$\begin{aligned}
\|\eta_j(t)\| &\leq \gamma^{M-1} \int_0^t \|e^{A_j^{(1)}(t-s_1)}\| \cdot \|A_{j+M-1}^{(M)} \eta_{j+M-1}(s_1)\| \frac{\mathrm{d}}{\mathrm{d}s_1} \\
&\leq \gamma^{M-1} j \|F_M\| \int_0^t e^{j\lambda_0(t-s_1)} \|\eta_{j+M-1}(s_1)\| \frac{\mathrm{d}}{\mathrm{d}s_1} \\
&\leq \gamma^{M-1} j \|F_M\| \left(\frac{\|\mathbf{u}_{\mathrm{in}}\|}{\gamma}\right)^{j+M-1} \int_0^t e^{j\lambda_0(t-s_1)} \left(\frac{\|F_M\|\|\mathbf{u}_{\mathrm{in}}\|^{M-1}}{|\lambda_0|}\right)^{k-1} f_{j+M-1,k-1,M}\left(|\lambda_0| s_1\right) \\
&= \frac{\|F_M\|\|\mathbf{u}_{\mathrm{in}}\|^{M-1}}{|\lambda_0|} \left(\frac{\|\mathbf{u}_{\mathrm{in}}\|}{\gamma}\right)^{j} \left(\frac{\|F_M\|\|\mathbf{u}_{\mathrm{in}}\|^{M-1}}{|\lambda_0|}\right)^{k-1} f_{j,k,M}\left(|\lambda_0| t\right) \\
&= \left(\frac{\|\mathbf{u}_{\mathrm{in}}\|}{\gamma}\right)^{j} \left(\frac{\|F_M\|\|\mathbf{u}_{\mathrm{in}}\|^{M-1}}{|\lambda_0|}\right)^{k} f_{j,k,M}\left(|\lambda_0| t\right), \qquad \text{(C.2.56)}
\end{aligned}$$

where in the second-last line we used Eq. (C.2.40). This shows the result in Eq. (C.2.53) for $k-1$ implies the result for $k$, and therefore for all $k$ by induction.

Since

$$R = \frac{\|F_M\|}{|\lambda_0|} \|\mathbf{u}_{\mathrm{in}}\|^{M-1}, \qquad \text{(C.2.57)}$$

we obtain the result given in the theorem. Moreover, since $j=1$ is an element of $\Omega_k$ for $k = \lceil \frac{N}{M-1} \rceil$, we obtain the claimed upper bound on $\|\eta_1\|$. □

### C.2.3 Bound on the error in terms of max-norm

For the case where the ODE is obtained from discretising the PDE, then the above bound on the error for the ODE can be used, but $\|\mathbf{u}_{\mathrm{in}}\|$ will increase with the number of discretisation points. Instead we can derive a bound using the max-norm, which will be (approximately) independent of the number of discretisation points. However, we will find that the possibly increasing max-norm with the higher-order discrete Laplacians can cause the bound to be larger than for the first-order discrete Laplacian.

We can start from Eq. (C.2.49) above, and use the fact that for the discretised PDE $A_{j+M-1}^{(M)}$ acts only on the components of $\mathbf{u}^{\otimes(j+M-1)}$ with powers of the entries of $\mathbf{u}$. That means that

$$\|A_{j+M-1}^{(M)} \mathbf{u}^{\otimes(j+M-1)}\|_{\max} \leq \|A_{j+M-1}^{(M)}\|_\infty \|\mathbf{u}\|_{\max}^{j+M-1}. \qquad \text{(C.2.58)}$$

Hence we can obtain the equivalent of Eq. (C.2.50) as

$$\|\eta_j(t)\|_{\max} \lesssim j\gamma^{-j} \|F_M\|_\infty \|\mathbf{u}_{\mathrm{in}}\|_{\max}^{j+M-1} \int_0^t \|e^{A_j^{(1)}(t-s_0)}\|_\infty \frac{\mathrm{d}}{\mathrm{d}s_0}. \qquad \text{(C.2.59)}$$

This expression is obtained in much the same way as before, but using the $\infty$-norm as the induced matrix norm for the max-norm on the vectors. Here it has been assumed that $\|\mathbf{u}\|_{\max} \leq \|\mathbf{u}_{\mathrm{in}}\|_{\max}$, which is correct for the exact PDE, but is not strictly correct for the higher-order discretised PDE. But, provided the error to the discretisation is appropriately bounded the max-norms should be sufficiently close to that for the continuous PDE, and so $\|\mathbf{u}\|_{\max} \lesssim \|\mathbf{u}_{\mathrm{in}}\|_{\max}$ (which is why we use $\lesssim$ in the expression above).

A further difficulty now is that we need a bound on the $\infty$-norm of $e^{A_j^{(1)}(t-s_0)}$. For the discretised

PDE we consider, we have $F_1 = DL_{k,d} + c\mathbb{I}^{\otimes d}$. First, since $A_j^{(1)}$ is constructed from $j$ commuting terms with $F_1$ acting on different subsystems, we have

$$\|e^{A_j^{(1)}(t-s_0)}\|_\infty = \|e^{F_1(t-s_0)}\|_\infty^j \,. \tag{C.2.60}$$

Next, since the identity commutes with the discretised Laplacian, we have

$$\|e^{F_1(t-s_0)}\|_\infty^j = \|e^{DL_{\kappa,d}(t-s_0)}\|_\infty^j e^{jc(t-s_0)} \,. \tag{C.2.61}$$

There we are using $\kappa$ for the order of the discretisation to avoid confusion with $k$ used for iteration in the derivation. Since the derivatives in the different directions commute, we have

$$\|e^{DL_{\kappa,d}(t-s_0)}\|_\infty^j e^{jc(t-s_0)} = \|e^{DL_\kappa(t-s_0)}\|_\infty^{jd} e^{jc(t-s_0)} \,. \tag{C.2.62}$$

Ideally we should have that the $\infty$-norm of the discretised Laplacian is equal to 1, but it can be larger by a moderate amount as discussed in Sec. 3.1.6. If we define the maximum of this norm as

$$G_\kappa := \max_{\tau \geq 0} \|e^{L_\kappa \tau}\|_\infty \,, \tag{C.2.63}$$

then we have

$$\|e^{A_j^{(1)}(t-s_0)}\|_\infty \leq G_\kappa^{jd} e^{jc(t-s_0)} \,. \tag{C.2.64}$$

That then gives us

$$\begin{aligned} \|\eta_j(t)\|_{\max} &\lesssim j\gamma^{-j}\|F_M\|_\infty \|\mathbf{u}_{\text{in}}\|_{\max}^{j+M-1} G_\kappa^{jd} \int_0^t e^{jc(t-s_0)} \frac{\mathrm{d}}{\mathrm{d}s_0} \\ &= \frac{\|F_M\|_\infty \|\mathbf{u}_{\text{in}}\|_{\max}^{M-1} G_\kappa^{jd}}{|c|} \left(1 - e^{jct}\right) . \end{aligned} \tag{C.2.65}$$

The general form of the bound is then

$$\|\eta_j(t)\|_{\max} \lesssim \left( \frac{\|F_M\|_\infty}{|c|} \|\mathbf{u}_{\text{in}}\|_{\max}^{(M-1)} G_\kappa^{d[j+(k-1)(M-1)/2]} \right)^k f_{j,k,M}\left(|c|t\right) . \tag{C.2.66}$$

This can be shown using iteration as per Eq. (C.2.56) to give

$$\begin{aligned} \|\eta_j(t)\|_{\max} &\leq \gamma^{M-1} \int_0^t \|e^{A_j^{(1)}(t-s_1)}\|_\infty \cdot \|A_{j+M-1}^{(M)} \eta_{j+M-1}(s_1)\|_{\max} \frac{\mathrm{d}}{\mathrm{d}s_1} \\ &\leq \gamma^{M-1} j \|F_M\|_\infty G_\kappa^{jd} \int_0^t e^{jc(t-s_1)} \|\eta_{j+M-1}(s_1)\|_{\max} \frac{\mathrm{d}}{\mathrm{d}s_1} \\ &\lesssim \gamma^{M-1} j \|F_M\|_\infty G_\kappa^{jd} \int_0^t e^{jc(t-s_1)} \left( \frac{\|F_M\|_\infty}{|c|} \|\mathbf{u}_{\text{in}}\|_{\max}^{M-1} G_\kappa^{d[j+k(M-1)/2]} \right)^{k-1} f_{j+M-1,k-1,M}\left(|c|s_1\right) \\ &= \frac{\gamma^{M-1}\|F_M\|_\infty G_\kappa^{jd}}{|c|} \left( \frac{\|F_M\|_\infty}{|c|} \|\mathbf{u}_{\text{in}}\|_{\max}^{M-1} G_\kappa^{d[j+k(M-1)/2]} \right)^{k-1} f_{j,k,M}\left(|c|t\right) \\ &= \left( \frac{\|F_M\|_\infty}{|c|} \|\mathbf{u}_{\text{in}}\|_{\max}^{M-1} G_\kappa^{d[j+(k-1)(M-1)/2]} \right)^k f_{j,k,M}\left(|c|t\right) . \end{aligned} \tag{C.2.67}$$

This upper bound is problematic, because it has $G_\kappa$ to a power increasing with $k$ inside the power of $k$. This means it is not possible to show convergence with increasing order of the Carleman linearisation. As the order is increased, $k$ will increase so eventually the expression inside the power of $k$ here will be larger than 1.

A further improvement can be made if we note where the factor of $G_\kappa^{jd}$ is coming from in the second line. That is coming from $\|e^{A_j^{(1)}(t-s_1)}\|_\infty$, but $A_j^{(1)}$ is a sum of $j$ operators each acting on a subsystem. That means $e^{A_j^{(1)}(t-s_1)}$ can be written as a tensor product of exponentials of $F_1$ acting on each of the $j$ subsystems. In turn, $A_{j+M-1}^{(M)}\eta_{j+M-1}(s_1)$ is a sum of $j$ terms, and each of those $A_{j+M-1}^{(M)}$ has affected only one of the $j$ subsystems. On the remainder of the subsystems, $\eta_{j+M-1}(s_1)$ has been obtained by an exponential of the form $e^{A_j^{(1)}(s_1-s_2)}$. That means, if we consider the operators before using the product formulae for norms, we have subsystems where $e^{F_1(t-s_1)}e^{F_1(s_1-s_2)} = e^{F_1(t-s_2)}$. The product of these exponentials can be bounded by $G_\kappa^d$ instead of $G_\kappa^{2d}$.

What this means is that for the $j$ exponentials above, $j-1$ of them can be combined with the previous exponentials, where the norm has already been taken into account as $G_\kappa^d$. That means that the factor need only be $G_\kappa^d$ rather than $G_\kappa^{jd}$, and we can obtain the form

$$\|\eta_j(t)\|_{\max} \lesssim G_\kappa^{dj}\left(\frac{\|F_M\|_\infty}{|c|}\|\mathbf{u}_{\text{in}}\|_{\max}^{(M-1)}G_\kappa^{dM}\right)^k f_{j,k,M}\left(|c|t\right). \tag{C.2.68}$$

The iteration to show this is

$$\begin{aligned}
\|\eta_j(t)\|_{\max} &\leq \gamma^{M-1}j\|F_M\|_\infty G_\kappa^d\int_0^t e^{jc(t-s_1)}\|\eta_{j+M-1}(s_1)\|_{\max}\frac{\mathrm{d}}{\mathrm{d}s_1}\\
&\lesssim \gamma^{M-1}j\|F_M\|_\infty G_\kappa^d G_\kappa^{d(j+M-1)}\int_0^t e^{jc(t-s_1)}\left(\frac{\|F_M\|_\infty}{|c|}\|\mathbf{u}_{\text{in}}\|_{\max}^{M-1}G_\kappa^{dM}\right)^{k-1} f_{j+M-1,k-1,M}\left(|c|s_1\right)\\
&= \frac{\gamma^{M-1}\|F_M\|_\infty G_\kappa^{d(j+M)}}{|c|}\left(\frac{\|F_M\|_\infty}{|c|}\|\mathbf{u}_{\text{in}}\|_{\max}^{M-1}G_\kappa^{dM}\right)^{k-1} f_{j,k,M}\left(|c|t\right)\\
&= G_\kappa^{dj}\left(\frac{\|F_M\|_\infty}{|c|}\|\mathbf{u}_{\text{in}}\|_{\max}^{M-1}G_\kappa^{dM}\right)^k f_{j,k,M}\left(|c|t\right).
\end{aligned} \tag{C.2.69}$$

The advantage of this form is now that the expression inside the exponential is now independent of $k$, and so increasing the order of the Carleman linearisation can improve the accuracy provided the expression inside the exponential is below 1.

## C.3 Proof of semi-discrete error bound due to finite difference discretisation

The proof to Lemma 3.1.7 follows.

*Proof.* We want to take account of the error derived from the spatial discretisation of the following rescaled $d$-dimensional nonlinear PDE

$$\partial_t u(\mathbf{x},t) - \left(D\Delta u(\mathbf{x},t) + cu(\mathbf{x},t)\right) - bu^M(\mathbf{x},t) = 0. \tag{C.3.1}$$

When we discretise the equation above, taking $n$ grid points in total, we use the $k$th $d$-dimensional

Laplacian approximation $L_{k,d}$, as given in Eq. (3.1.76), such that

$$L\mathbf{u}(t) = L_{k,d}\mathbf{u}(t) + \mathbf{r}(t), \tag{C.3.2}$$

where $\mathbf{r}(t)$ is the remainder vector. Using the results given in [KWBA17; CLO21] for the Laplacian approximation, we can write the max-norm of the remainder vector as

$$\|\mathbf{r}\|_{\max} = \|(L - L_{k,d})\,\mathbf{u}\|_{\max} = O\left(C(u,k)\left(\frac{e}{2}\right)^{2k} n^{-(2k-1)/d}\right), \tag{C.3.3}$$

where $C(u,k)$ is a constant depending on the $(2k+1)$st spatial derivative on each direction of the Laplacian operator,

$$C(u,k) = \sum_{j=1}^{d}\left|\frac{d^{2k+1}u}{dx_j^{2k+1}}\right|. \tag{C.3.4}$$

The error in terms of the 2-norm can be obtained immediately from that for the max-norm by multiplying by the factor $\sqrt{n}$

$$\|\mathbf{r}\| = \|(L - L_{k,d})\,\mathbf{u}\| = O\left(C(u,k)\sqrt{n}\left(\frac{e}{2}\right)^{2k} n^{-(2k-1)/d}\right). \tag{C.3.5}$$

In the equation above we also introduce notions of the exact quantities evaluated on the grid points as $\mathbf{u}(t) = [u_1, u_2, \ldots, u_n]^T$ where the components are time dependent. The grid representation of the Laplacian $\Delta$ is denoted $L$, in the sense that $L\mathbf{u}$ is used to represent the Laplacian of $u$ evaluated at grid points.

To quantify the error due to spatial discretisation we write

$$\boldsymbol{\varepsilon}(t) = \mathbf{u}(t) - \mathbf{u}^h(t), \tag{C.3.6}$$

where $\mathbf{u}^h(t)$ are the approximate values of the function resulting from the spatial discretisation. From the definition above we then have

$$\begin{aligned}\frac{\mathrm{d}\boldsymbol{\varepsilon}}{\mathrm{d}t} &= D(L\mathbf{u} - L_{k,d}\mathbf{u}^h) + c(\mathbf{u} - \mathbf{u}^h) + F_M(\mathbf{u}^{\otimes M} - (\mathbf{u}^h)^{\otimes M}) \\ &= \left(DL_{k,d} + c\mathbb{I}^{\otimes d}\right)\boldsymbol{\varepsilon} + F_M(\mathbf{u}^{\otimes M} - (\mathbf{u}^h)^{\otimes M}) + \mathbf{r}\,,\end{aligned} \tag{C.3.7}$$

where we have used Eq. (C.3.6) followed by using the operator definitions as given in Eq. (3.1.75), i.e., $F_1 = DL_{k,d} + c\mathbb{I}^{\otimes d}$. Note that $F_M$ is one-sparse and has only entries equal to $b$ resulting from the spatial discretisation of Eq. (C.3.1).

We can then obtain the derivative of $\|\boldsymbol{\varepsilon}\|$ by considering first

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}(\boldsymbol{\varepsilon}^\dagger\boldsymbol{\varepsilon}) &= \frac{\mathrm{d}\boldsymbol{\varepsilon}^\dagger}{\mathrm{d}t}\boldsymbol{\varepsilon} + \boldsymbol{\varepsilon}^\dagger\frac{\mathrm{d}\boldsymbol{\varepsilon}}{\mathrm{d}t} \\ &= \boldsymbol{\varepsilon}^\dagger\left(F_1 + F_1^\dagger\right)\boldsymbol{\varepsilon} + \left(\mathbf{u}^{\otimes M} - (\mathbf{u}^h)^{\otimes M}\right)^\dagger F_M^\dagger\boldsymbol{\varepsilon} \\ &\quad + \boldsymbol{\varepsilon}^\dagger F_M\left(\mathbf{u}^{\otimes M} - (\mathbf{u}^h)^{\otimes M}\right) + \mathbf{r}^\dagger\boldsymbol{\varepsilon} + \boldsymbol{\varepsilon}^\dagger\mathbf{r}\,.\end{aligned} \tag{C.3.8}$$

From the quantity above we have for the first term

$$\boldsymbol{\varepsilon}^\dagger \left(F_1 + F_1^\dagger\right) \boldsymbol{\varepsilon} \leq 2c\|\boldsymbol{\varepsilon}\|^2, \tag{C.3.9}$$

where we have used the results for the eigenvalues of $F_1$ from Eq. (3.1.87), i.e.,

$$\lambda_\ell(F_1) = Dn^{2/d}a_0 + c + Dn^{2/d}\sum_{j=1}^{k}\left[a_j\left(\omega^{(\ell-1)j} + \omega^{(\ell-1)(n-j)}\right)\right], \tag{C.3.10}$$

by considering a $d$-dimensional system. The expression for the eigenvalues above $c < 0$ is chosen such that it has the maximum real part among all the eigenvalues for $F_1$ and is smaller than zero to guarantee the dissipativity of the PDE. Now for the two other terms, we can first derive the following bound

$$\begin{aligned}
\left\|F_M(\mathbf{u}^{\otimes M} - (\mathbf{u}^h)^{\otimes M})\right\|^2 &= \sum_\ell b^2(\mathbf{u}_\ell^M - (\mathbf{u}_\ell^h)^M)^2 \\
&\leq b^2\sum_\ell\left(\sum_{j=0}^{M-1}\left(\mathbf{u}_\ell^h\right)^j\left(\mathbf{u}_\ell - \mathbf{u}_\ell{}^h\right)\mathbf{u}_\ell{}^{(M-j-1)}\right)^2 \\
&\leq b^2\sum_\ell\left(\sum_{j=0}^{M-1}\left(\mathbf{u}_\ell - \mathbf{u}_\ell^h\right)\right)^2 \max\left(\|\widetilde{\mathbf{u}}\|_{\max}^{2(M-1)}, \left\|\widetilde{\mathbf{u}}^h\right\|_{\max}^{2(M-1)}\right) \\
&\leq b^2M^2\|\boldsymbol{\varepsilon}\|^2\max\left(\|\widetilde{\mathbf{u}}\|_{\max}^{2(M-1)}, \left\|\widetilde{\mathbf{u}}^h\right\|_{\max}^{2(M-1)}\right).
\end{aligned} \tag{C.3.11}$$

Provided $|c| > \|\mathbf{u}_{\text{in}}\|_{\max}^{M-1}|b|$, the PDE is stable, and we have $\|\mathbf{u}\|_{\max} \leq \|\mathbf{u}_{\text{in}}\|_{\max}$. Then using the approximation that $\|\mathbf{u}\|_{\max} \approx \left\|\mathbf{u}^h\right\|_{\max}$ (the solution is accurate) we obtain

$$\left\|F_M(\mathbf{u}^{\otimes M} - (\mathbf{u}^h)^{\otimes M})\right\| \leq bM\|\boldsymbol{\varepsilon}\|\max\left(\|\mathbf{u}\|_{\max}^{M-1}, \left\|\mathbf{u}^h\right\|_{\max}^{M-1}\right) \lesssim bM\|\boldsymbol{\varepsilon}\|\|\mathbf{u}_{\text{in}}\|_{\max}^{M-1}. \tag{C.3.12}$$

Using that expression for the two other terms in Eq. (C.3.8)

$$\begin{aligned}
\left(\mathbf{u}^{\otimes M} - (\mathbf{u}^h)^{\otimes M}\right)^\dagger F_M^\dagger\boldsymbol{\varepsilon} + \boldsymbol{\varepsilon}^\dagger F_M\left(\mathbf{u}^{\otimes M} - (\mathbf{u}^h)^{\otimes M}\right) &\leq 2\|F_M(\mathbf{u}^{\otimes M} - (\mathbf{u}^h)^{\otimes M})\|\|\boldsymbol{\varepsilon}\| \\
&\leq 2bM\|\boldsymbol{\varepsilon}\|^2\|\mathbf{u}_{\text{in}}\|_{\max}^{M-1}.
\end{aligned} \tag{C.3.13}$$

For the last two components in Eq. (C.3.8) we have

$$\mathbf{r}^\dagger\boldsymbol{\varepsilon} + \boldsymbol{\varepsilon}^\dagger\mathbf{r} \leq 2\|\mathbf{r}\|\|\boldsymbol{\varepsilon}\|. \tag{C.3.14}$$

Returning to Eq. (C.3.8) together with the fact that $\frac{\mathrm{d}\|\boldsymbol{\varepsilon}\|}{\mathrm{d}t} = (2\|\boldsymbol{\varepsilon}\|)^{-1}\frac{\mathrm{d}\|\boldsymbol{\varepsilon}\|^2}{\mathrm{d}t}$, we have

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\|\boldsymbol{\varepsilon}\| = (2\|\boldsymbol{\varepsilon}\|)^{-1}\Big[&\boldsymbol{\varepsilon}^\dagger\left(F_1 + F_1^\dagger\right)\boldsymbol{\varepsilon} + \left(\mathbf{u}^{\otimes M} - (\mathbf{u}^h)^{\otimes M}\right)^\dagger F_M^\dagger\boldsymbol{\varepsilon} \\
&+ \boldsymbol{\varepsilon}^\dagger F_M\left(\mathbf{u}^{\otimes M} - (\mathbf{u}^h)^{\otimes M}\right) + \mathbf{r}^\dagger\boldsymbol{\varepsilon} + \boldsymbol{\varepsilon}^\dagger\mathbf{r}\Big].
\end{aligned} \tag{C.3.15}$$

We can now combine Eq. (C.3.9), Eq. (C.3.13) and Eq. (C.3.14) in the equation above to get

$$\frac{\mathrm{d}}{\mathrm{d}t}\|\boldsymbol{\varepsilon}\| \leq \left(c + M|b|\|\mathbf{u}_{\text{in}}\|_{\max}^{M-1}\right)\|\boldsymbol{\varepsilon}\| + \|\mathbf{r}\|, \tag{C.3.16}$$

where we used the condition that $|c| > M|b|\|\mathbf{u}_{\text{in}}\|^{M-1}$. That is stronger than the condition $|c| > |b|\|\mathbf{u}_{\text{in}}\|^{M-1}$ which ensures that $\mathbf{u}(t)$ does not increase with time. We next apply the solution for the ODE as it is given in Eq. (C.2.20), with $\|\boldsymbol{\varepsilon}(0)\| = 0$

$$\|\boldsymbol{\varepsilon}(t)\| \leq \frac{1 - \exp\left\{\left(c + M|b|\|\mathbf{u}_{\text{in}}\|_{\max}^{M-1}\right)t\right\}}{\left|c + M|b|\|\mathbf{u}_{\text{in}}\|_{\max}^{M-1}\right)}\|\mathbf{r}\|, \tag{C.3.17}$$

We can finally conclude from Eq. (C.3.5) that

$$\|\boldsymbol{\varepsilon}(t)\| = O\left(C(u,k)\sqrt{n}\left(\frac{e}{2}\right)^{2k} n^{-(2k-1)/d}\frac{1 - \exp\left\{\left(c + M|b|\|\mathbf{u}_{\text{in}}\|_{\max}^{M-1}\right)t\right\}}{\left|c + M|b|\|\mathbf{u}_{\text{in}}\|_{\max}^{M-1}\right|}\right). \tag{C.3.18}$$

□

## C.4 Error when using high-order finite difference discretisations

From Eq. (C.3.3) we have, for $d = 1$, the spatial discretisation max-norm error

$$O\left(C(u,k)\left(\frac{e}{2}\right)^{2k} n^{-(2k-1)}\right). \tag{C.4.1}$$

Therefore, if we want equivalent error between first-order and order-$k$ discretisations, we would need

$$C(u,k)\left(\frac{e}{2}\right)^{2k} n_k^{-(2k-1)} \sim C(u,1)\left(\frac{e}{2}\right)^{2} n_1^{-1}. \tag{C.4.2}$$

That then gives

$$n_k = O\left(\left(\frac{C(u,k)\, n_1}{C(u,1)}\right)^{1/(2k-1)}\right). \tag{C.4.3}$$

This shows that, apart from the factor of $C(u,k)$, increasing the order provides a $(2k-1)$-root improvement in the number of discretisation points needed to achieve a given error. That is for equal max-norm error. If we were to instead consider 2-norm error, then there would be a $(4k-3)$-root improvement in the number of points needed. That is because the lower $n_k$ also reduces the 2-norm by reducing the length of the vector.

In Fig. C.2 we compare the scaling of the error for a first and second-order approximation of the Laplacian (i.e., $k = 1, 2$ in Eq. (3.1.77)) for a simple differential equation to illustrate the faster error convergence. The advantages of using a certain order of approximation $k$ over lower orders will depend on the specific PDE to be solved and the initial conditions. The value of $C(u,k)$ can increase with $k$ if there is high-frequency variation of $\mathbf{u}$. That is less of a problem with the Laplace

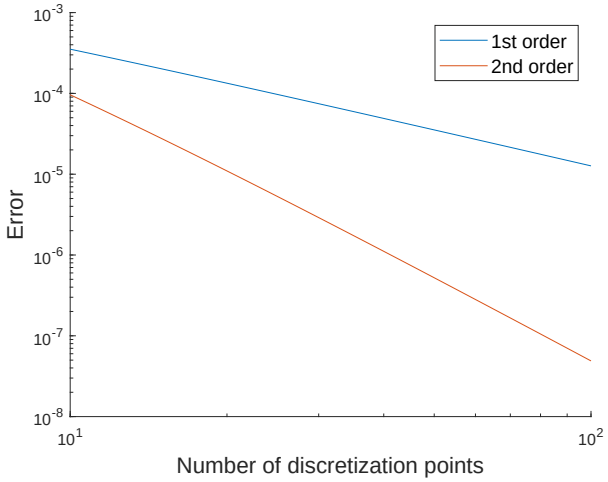



Figure C.2: Comparison of error in approximating the Laplacian when considering first ($k = 1$) and second ($k = 2$) order discretisations. To illustrate the scaling of the error we have solved the equation $u_{xx} = e^x$ in the interval $x \in [0, 1]$ with Dirichlet boundary conditions $u(0) = 0$ and $u(1) = 1$. The error is computed as $\left\| L_k^{-1} \exp[\mathbf{x}] - \mathbf{u}_{\text{sol}} \right\|$, where $\mathbf{x}$ is the vector of discrete points in space, $\exp[\mathbf{x}]$ is obtained by exponentiating each entry of $\mathbf{x}$ and $\mathbf{u}_{\text{sol}}$ is the vector containing the exact discretised solution evaluated at the grid points.

equation, because that smooths out high-frequency variation, but could be more of a problem for other PDEs. Moreover, the complexity of the block-encoding increases with $k$, so if $k$ is order $n$ then it would negate any quantum speedup in solving the PDE (unless there were a more efficient block-encoding). Therefore, one should take into consideration the specifics of the PDE to be solved and choose $k$ accordingly.

## C.5 Construction of finite-difference coefficients

For the construction of arbitrary finite-difference coefficients, we generally refer to standard methods, such as based on Taylor expansions or Lagrange polynomials [Tay16]. Given a grid discretisation of $x$ over some finite interval in $n$ points, a finite difference approximation or "stencil" to the $m$th derivative of a function $f(x)$ reads

$$f^{(m)}(x) \approx \sum_{j=0}^{n-1} a_j f(x_j), \tag{C.5.1}$$

where $\{a_j\}_{j=0}^{n-1}$ are the stencil points to the grid $\{x_j\}_{j=0}^{n-1}$. When finding these approximations, one may express $f(x)$ via its Taylor expansion or as a Lagrange polynomial to find the coefficients $\{a_j\}$ given a certain grid. Generally, the number of grid points used in approximating a certain derivative will influence its accuracy; we refer to Ref. [Li05] which provides a comprehensive analysis.

Within this study, we are mostly interested in the discretisation of second derivatives, where we focus on central stencils with periodic boundary conditions. To that end, one can make use of the formulae in Ref. [Li05], which as well as the stencil coefficients also provides an investigation of

the discretisation error. Whenever we talk about "order of discretisation" $k$ in this work, we have a central stencil with $2k+1$ points, corresponding to an error that goes roughly as $O\left(h^{2k-1}\right)$ in a uniform grid spacing $h$.

The formulae in Ref. [Li05] are also considered in other works that apply finite differences to quantum algorithms [KWBA17; CLO21]. This way, an approximation to the second derivative for a function $f(x)$ can be written as

$$f''(0) \approx \frac{1}{h^2} \sum_{j=-k}^{k} a_j f(jh), \quad \text{where} \tag{C.5.2}$$

$$a_j = \begin{cases} \frac{2(-1)^{j+1}(k!)^2}{j^2(k-j)!(k+j)!}, & 1 \leq j \leq k \\ -2\sum_{i=1}^{k} a_i, & j = 0 \\ a_{-j}, & -k \geq j \geq -1. \end{cases} \tag{C.5.3}$$

For $k = 1$ the usual stencil with coefficients $1, -2, 1$ is retrieved.

Using this construction, the discrete Laplacian matrix $L_k$ with a uniform grid spacing $h$ has entries

$$[L_k]_{pq} = \begin{cases} \frac{2(-1)^{\ell+1}(k!)^2}{\ell^2(k-\ell)!(k+\ell)!}, & q < p \leq q+k \quad \text{or} \\ & q-k \leq p < q, \quad \ell := |p-q| \\ -2\sum_{\ell=q+1}^{q+k}[L_k]_{\ell q}, & p = q \end{cases} \tag{C.5.4}$$

To treat boundaries, one can use the method in Ref. [SN06], which showns that if using a $k$th order finite difference scheme within the domain, it is sufficient to treat the boundary with a stencil of accuracy $k-2$, given that the solution is point-wise bounded. We need to make sure that only internal points are used for the creation of the difference stencils; thus, for rows with row-index smaller than $k$ or greater than $n-k$, we need to construct specific, non-centric stencils. The procedure for boundary treatment in Ref. [CJO19] may be used as well.

## C.6 Sparsity of Carleman matrix

Here we discuss the sparsity of the Carleman matrix applied to Eq. (3.1.74).

**Lemma C.6.1** (Sparsity of the Carleman matrix for nonlinear PDE problem)**.** *Let $\mathcal{A}_N$ be the Carleman matrix with truncation number $N$ applied to discretised PDE problem $\partial_t u = DL_k u + cu + bu^M$ in Eq.* (3.1.74)*, where $L_k$ is a central difference approximation of the Laplacian with $2k+1$ stencil points under periodic boundary conditions. Then, considering only one dimension in space, the sparsity $s(\mathcal{A}_N)$ of $\mathcal{A}_N$ is*

$$s(\mathcal{A}_N) = O\left(kN\right). \tag{C.6.1}$$

*Proof.* As individual diagonal blocks in $\mathcal{A}_N$ (see Eq. (3.1.17)) do not overlap, we can treat them independently and sum up the contributions. We first consider the linear term $F_1 = DL_k + c\mathbb{I}$. As the diagonal entries of $L_k$ are non-zero, we can equivalently only consider $L_k$. The central-difference stencil is of the form $\frac{1}{h^2}\sum_{\ell=-k}^{k} r_\ell f(\ell h)$, hence for each row of $L_k$, there are $2k+1$ non-zero contributions, where there is one diagonal entry and $2k$ off the diagonal. The diagonal blocks of $\mathcal{A}_N$

are of the form

$$A_j^{(1)} = F_1 \otimes \mathbb{I}^{\otimes(j-1)} + \mathbb{I} \otimes F_1 \otimes \mathbb{I}^{\otimes(j-2)} + \cdots + \mathbb{I}^{\otimes(j-1)} \otimes F_1 \,. \tag{C.6.2}$$

Each term is of the form of the tensor product of $F_1$ with identity matrices, which leaves the sparsity unchanged. The sparsity of $A_j^{(1)}$ is then upper bounded by the sum of the sparsities of the terms, and so is upper bounded by $j(2k+1)$. Since the maximum value of $j$ is $N$, that gives an upper bound of $N(2k+1)$. In fact, since each term has one on-diagonal and $2k$ off-diagonal entries in each row, the sparsity will be $2kN+1$.

Next, consider the off-diagonal blocks of $\mathcal{A}_N$, given by (see Eq. (3.1.15))

$$A_{j+M-1}^{(M)} = F_M \otimes \mathbb{I}^{\otimes(j-1)} + \mathbb{I} \otimes F_M \otimes \mathbb{I}^{(j-2)} + \cdots + \mathbb{I}^{\otimes(j-1)} \otimes F_M \,. \tag{C.6.3}$$

The $n \times n^M$ matrices $F_M$ have the only non-zero entries (see Eq. (3.1.83) for the case $M = 2$)

$$(F_M)_{i,i+(i-1)(\sum_{\ell=1}^{M-1} n^\ell)} = b \,. \tag{C.6.4}$$

Hence $F_M$ is one-sparse, and again taking the tensor product with the identity leaves the sparsity unchanged. The sparsity of $A_{j+M-1}^{(M)}$ is then no more than the sum of the sparsities of the terms, and so is no larger than $j$. The last ($N$th) column of $\mathcal{A}_N$ has $A_{j+M-1}^{(M)}$ with $j+M-1 = N$, and so the maximum value of $j$ is $N-M+1 < N$.

The overall sparsity of $\mathcal{A}_N$ is then upper bounded by the sum of the sparsities of $A_j^{(1)}$ (on-diagonal blocks) and $A_{j+M-1}^{(M)}$ (off-diagonal blocks). That gives the total sparsity upper bounded as

$$s(\mathcal{A}_N) < N(2k+1) + N \in O\left(kN\right) \,. \tag{C.6.5}$$

□

## C.7 Block-encoding of Carleman matrix $\mathcal{A}_N$

As usual in the block encoding of an operator written as a sum, the sum can be block encoded by using a register in superposition to select between the terms in the sum, and then a select operation controlled by that register to implement the terms in the sum. Here we have a sum of block diagonals with $A_j^{(1)}$ and $A_{j+M-1}^{(M)}$. We therefore have the following approach to the block encoding.

1. A qubit in superposition is used to select between $A_j^{(1)}$ and $A_{j+M-1}^{(M)}$.

2. We prepare ancilla registers in equal superpositions over $N$ and $N-M+1$ basis states; these register store $\ell$. These registers are for the block encodings of $A_j^{(1)}$ and $A_{j+M-1}^{(M)}$, respectively.

3. For the block encoding of $A_j^{(1)}$ we apply $F_1$ to target register $\ell$. Instead of using $N$ controlled block encodings of $F_1$, we can perform a controlled swap of register $\ell$ to a working register, use *one* controlled block encoding of $F_1$, then swap the result back. These controlled swaps have cost $O\left(N \log n\right)$ for dimension $n$ of the vector, in terms of elementary gates.

4. For the block encoding of $A_{j+M-1}^{(M)}$ we perform a controlled application of $F_M$ on target registers $\ell$ to $\ell+M-1$ (with $\ell$ starting from 1). Rather than using $N-M+1$ controlled block encodings of

$F_M$, we can swap the $M$ target registers to working registers, then perform one controlled block encoding of $F_M$, then swap the result back. These controlled swaps have cost $O(NM \log n)$.

5. For block encoding $A^{(M)}_{j+M-1}$ a further complication is that $F_M$ maps $M$ copies to 1, so we need to shift the target systems over so we have a contiguous block of copies. That is, if $\ell = 1$, we have the first term in the sum for $A^{(M)}_{j+M-1}$, the first $M$ copies are mapped to 1 and we have the remaining $M-1$ copies being zeroed, and then there are the rest of the copies. The $M-1$ zeroed copies need to be shifted to the end. These controlled swaps have a further cost $O(NM \log n)$ in elementary gates. We also subtract $M-1$ from $m$, to indicate that the number of copies has been reduced by $M-1$. That corresponds to the blocks containing $A^{(M)}_{j+M-1}$ being away from the main diagonal.

6. Lastly, we need to truncate the sums for $A^{(1)}_j$ and $A^{(M)}_{j+M-1}$ to $j$ terms. This can be performed by using an inequality test between $\ell$ and $m$, with a result of 1 removing the term in the block encoding. In particular, for $A^{(1)}_j$ we have $j$ corresponding to $m$, and $\ell > m$ indicates that we are past the end of the sum. Then, for $A^{(M)}_{j+M-1}$ we have $j + M - 1 = m$, so $\ell > m - M + 1$ indicates we are past the end of the sum.

It is also possible to prepare the initial Carleman vector with $O(N)$ calls to the state preparation for $\mathbf{u}_{\text{in}}$. We may first prepare a register in unary for the appropriate weightings between the components of the Carleman vector. Then simply use each qubit of that unary register to control preparation of the $\mathbf{u}_{\text{in}}$ state. Then we obtain a number of copies corresponding to the component of the Carleman vector as required.

# Appendix D

# Supplementary Material for Section 3.2

## D.1 Proof of Lemma 3.2.13

**Lemma** (Parametrized exponential of an orthogonal projection; Lemma 3.2.13 in main text)**.** *Let* $\mathsf{P}, \mathsf{Q}$ *be orthogonal projections on a vector space so that* $\mathsf{P} + \mathsf{Q} = \mathbb{I}$*. Then, for any* $\xi \in \mathbb{C}$*,*

$$\mathrm{e}^{\xi \mathsf{P}} = \mathsf{Q} + \mathrm{e}^{\xi} \mathsf{P}. \tag{D.1.1}$$

*For* $\xi \in \mathrm{i}\,\mathbb{R}$*, we retrieve Hamiltonian simulation.*

*Proof.* We start by using the Taylor series of the operator exponential,

$$\begin{aligned} \mathrm{e}^{\xi \mathsf{P}} &= \sum_{k \geq 0} \frac{\xi^k}{k!} (\mathsf{P})^k \\ &= \mathbb{I} + \sum_{k \geq 1} \frac{\xi^k}{k!} (\mathsf{P})^k \quad ; \mathsf{P}^k = \mathsf{P}\, \forall k \geq 1 \\ &= \mathbb{I} \pm \mathsf{P} + \mathsf{P} \sum_{k \geq 1} \frac{\xi^k}{k!} \\ &= \mathsf{Q} + \mathrm{e}^{\xi} \mathsf{P}. \end{aligned} \tag{D.1.2}$$

□

## D.2 Derivatives for conjugated generators

Here we derive the identities in Eq. (3.2.161), recalling that

$$\begin{aligned}\mathrm{ad}_y(x) &= [y,x]\\ \mathrm{ad}_y^m(x) &= \underbrace{[y,[y,\dots,[y}_{m\text{ times}},x]]\cdots] = \underbrace{\mathrm{ad}_y\circ\cdots\circ\mathrm{ad}_y}_{m\text{ times}}(x)\\ \mathrm{ad}^0(x) &= x.\end{aligned} \tag{D.2.1}$$

Let the conjugated operator be

$$A_C(t) = \mathrm{e}^{Bt}A(t)\mathrm{e}^{-Bt}, \tag{D.2.2}$$

with $A(t)$ smooth in time $t\in\mathbb{R}$. In the main text, we use $B=\mathsf{i}\mathsf{P}_c$ and thus call this interaction picture $(\cdot)_I$. Similarly, $b_C(t)=\mathrm{e}^{Bt}b(t)$ and smooth $b(t)$. We assume here that $B$ is constant in time. Therefore, the final expressions rely on a conjugation operation $T_t$ so that $T_tA(T_t)^{-1}$ satisfies $\partial_tT_t = BT_t$ and $[B,T_t]=0$, which restricts us to $\mathrm{e}^{Bt}$ with constant $B$. Then, we consider an arbitrary $p$th, $p\in\mathbb{N}$, partial derivative with respect to time

$$(\partial_t)^pA_C(t) = (\partial_t)^{p-1}\left((\dot{A}(t)_C + [B,A_C(t)]\right) = (\partial_t)^{p-2}\left([B,[B,A_C]] + 2[B,(\dot{A})_C] + (\ddot{A})_C\right) = \dots$$

Using the ad-notation introduced above,

$$(\partial_t)^pA_C(t) = (\partial_t)^{p-1}\left((\dot{A}(t))_C + \mathrm{ad}_B(A_C(t))\right) = (\partial_t)^{p-2}\left(\mathrm{ad}^2(A_C(t)) + 2\mathrm{ad}_B((\dot{A})_C) + (\ddot{A})_C\right) = \dots$$

Then, as a direct consequence of the product rule of differentiation and the aforementioned condition that $\partial_t\mathrm{e}^{Bt} = B\mathrm{e}^{Bt} = \mathrm{e}^{Bt}B$, we can recognize a binomial structure of

$$\partial_t^p(A_C(t)) = ((\partial_t + \mathrm{ad}_B)^p(A(t)))_C\,,\ p\in\mathbb{N}. \tag{D.2.3}$$

For binomials, we have the well-known identity that $(x+y)^p = \sum_{0\le q\le p}\binom{p}{q}x^qy^{p-q}$, hence

$$((\partial_t+\mathrm{ad}_B)^p(A(t)))_C = \left(\sum_{0\le q\le p}\binom{p}{q}\mathrm{ad}_B^q\,\partial_t^{(p-q)}A(t)\right)_C = \sum_{0\le q\le p}\binom{p}{q}\mathrm{ad}_B^q\left(\partial_t^{(p-q)}A(t)\right)_C \tag{D.2.4}$$

The same approach can be used for vectors $b(t)$, where the interaction picture rotation is only a left-multiplication by $\mathrm{e}^{Bt}$ rather than a conjugation. This means, simply replace $A(t)$ with $b(t)$ in Eq. (D.2.4) and replace $\mathrm{ad}_y^p(x)$ with left-multiplication instead of commutators, $y^px$.

## D.3 Proof of Proposition 3.2.5

**Proposition D.3.1** (Restatement of Proposition 3.2.5)**.** *Let $A(t) = H(t) + \zeta V(t)$ be a perturbed dynamical generator with $\zeta\|V(t')\| \ll \|H(t')\|$ for any $t \geq t' \geq 0$ and $A(t), H(t), V(t)$ complex matrices. Let $v(t)$ be the solution to $\frac{\mathrm{d}}{\mathrm{d}t}v(t) = A(t)v(t) + b(t)$ with initial data $v(0)$. Further, suppose we are interested in measuring the expectation value of a matrix $P$. Then, the effect due to perturbation $\zeta V(t)$ on the expectation of $P$ up to first order in the strength of the perturbation*

$$\langle P(t)\rangle - \langle P\rangle_0 = \zeta \int_0^t \mathrm{d}t' \, \frac{\mathrm{tr}\left[\left\{P, \overline{V}(t,t')\sigma(t',t)\right\}_\sim\right]}{\mathrm{tr}\left[\sigma(t,t)\right]} + O\big(\zeta^2\big) \tag{D.3.1}$$

*where $\langle P(t)\rangle$ is the perturbed expectation and $\langle P\rangle_0$ is the expectation due to the unperturbed dynamics generated by $H(t)$. Further, there is the modified anticommutator $\{X, Y\}_\sim = XY + Y^\dagger X$, a density augmented by the forcing term $b(t)$ through $\sigma(t',t) = T_{t'}(v(0)\delta(t') + b(t'))\big(T_t(v(0)\delta(t) + b(t))\big)^\dagger$ where $T_t(u) = \int_0^t \mathrm{d}s\, \mathcal{T} \exp\Big(\int_s^t \mathrm{d}t'\, H(t')\Big)u(s)$ and $\overline{V}(t,t') = \int_{t'}^t \mathrm{d}\tau\, V(\tau)$.*

*Proof.* Given dynamics

$$\frac{\mathrm{d}}{\mathrm{d}t}v(t) = A(t)v(t) + b(t) = \left(H(t) + \zeta V(t)\right)v(t) + b(t), \tag{D.3.2}$$

we seek to find a first-order estimate to the difference in an expectation of an operator $P$ with respect to the perturbed evolution $v_{H+\zeta V}(t)$ compared to the unperturbed evolution $v_H(t)$:

$$\frac{\langle v_{H+\zeta V}(t),\, P\, v_{H+\zeta V}(t)\rangle - \langle v_H(t),\, P\, v_H(t)\rangle}{\langle v_H(t), v_H(t)\rangle} =: \langle P(t)\rangle - \langle P\rangle_0 = \delta\langle P(t)\rangle, \tag{D.3.3}$$

This covers, beyond non-Hermitian operators, also a time-dependency in the original dynamics and a forcing term in the differential equation and thus goes beyond formulas provided in [SDM22; GH22]. As a reference on non-Hermitian dynamics, e.g. see [DTM90; Bro13]. We may express the solution, according to Eq. (3.2.57) in terms of homogeneous and particular solution as follows using Green's functions,

$$v(t) = v_\mathrm{h}(t) + v_\mathrm{p}(t) = (G \star v_0)(t) + (G \star b)(t) \tag{D.3.4}$$

where $v_0(t) = \delta(t)v(0)$ (where $\delta(t)$ is the delta distribution) and

$$(G \star x)(t) = (G \star x)(t, 0) = \int_0^t \mathrm{d}s\, G(t,s)x(s). \tag{D.3.5}$$

Note that we restrict ourselves to finite-dimensional, matrix-values $A(t)$, and thus also $G(t,s)$. This means that all convolutions will go across the time degree of freedom and the space degree of freedom follows matrix-matrix and matrix-vector multiplication. Of course one could think of generalizing this approach to infinite-dimensional objects, which we leave up for further research. For time-propagation according to Eq. (D.3.2), we can identify

$$G(t,s) = G_A(t,s) = \mathcal{T}\mathrm{e}^{\int_s^t \mathrm{d}s' A(s')}. \tag{D.3.6}$$

In the case of the homogeneous solution, the convolution simplifies to a matrix-vector product of the time-ordered exponential with the initial vector, and similarly for a constant forcing term. The

reason we choose this unconventional form for the homogeneous solution is that it allows us to treat the homogeneous $v_{\mathrm{h}}(t)$ and particular solution $v_{\mathrm{p}}(t)$ simultaneously. We denote $G = G_A$ with generator $A$, and $G_H$ and $G_V$ with the respective generators in the subscript. If we drop one of the time arguments, then $G(t) = G(t, 0)$.

The composition of multiple convolutions follows

$$(G_1 \star G_2 \star x)(t, 0) = \int_0^t \mathrm{d}t_1 \int_0^{t_1} \mathrm{d}t_2\, G_1(t, t_1) G_2(t_1, t_2) x(t_2), \tag{D.3.7}$$

and similar for more than two kernel functions; this combination preserves time-ordering. To make notation simpler, let us introduce $w(t) = v_0(t) + b(t) = \delta(t)v(0) + b(t)$ to collect homogeneous and inhomogeneous terms.

In order to analyze perturbative problems, the notion of interaction picture is often convenient in physics to isolate the effects due to perturbation in the analysis – this will also be the case here. Initially, let us assume that $H(t) \neq H^\dagger(t)$. Then, we define $T_t$ as evolution generated by $H(t)$ ('forward') and $\overline{T}_{-t}$ is generated by $-H^\dagger(t)$ ('backward', defined on the adjoints $u^\dagger(t)$), so that $T_t(u) = (G_H \star u)(t)$ and $\overline{T}_{-t}(u^\dagger) = (u^\dagger \star G_{-H^\dagger})(t)$. While these are *not* unitary and $\{T_t\}_t, \{\overline{T}_t\}_t$ form separate dynamical semigroups (over the right and left half-line respectively), the inverse elements for $\{T_t\}$ are in the adjoint elements from $\{\overline{T}_{-t}\}$ and vice versa, i.e., $T_t \overline{T}^\dagger_{-t} = \overline{T}^\dagger_{-t} T_t = \overline{T}_{-t} T_t^\dagger = \mathbb{I}$. This is a consequence of bi-orthogonality of the bases of the non-Hermitian generator [DTM90]. Then, $T_t, \overline{T}_t$ give us a means to introduce a non-Hermitian interaction picture transformation similar to what was done in [SDM22] where $H$ was chosen to be constant in time; then, such an interaction picture transformation does not change the expectation based on the (not necessarily unique) definitions,

$$\begin{aligned} v_I(t) &:= \overline{T}^\dagger_{-t} v(t) = (G^\dagger_{-H^\dagger} \star v)(t) \\ v_I^\dagger(t) &:= v^\dagger(t) \overline{T}_{-t} = (v^\dagger \star G_{-H^\dagger})(t) \\ P_I(t) &:= T_t^\dagger P T_t. \end{aligned} \tag{D.3.8}$$

Intuitively, observables experience forward evolution and states undergo the backward evolution. We verify by inserting into the expectation,

$$\begin{aligned} \langle P(t) \rangle := \frac{\langle v(t), \mathbb{I}\, P\, \mathbb{I}\, v(t) \rangle}{\langle v(t), v(t) \rangle} = \frac{\langle v(t), \overline{T}_{-t} T_t^\dagger P T_t \overline{T}^\dagger_{-t} v(t) \rangle}{\langle v(t), v(t) \rangle} &= \frac{\langle \overline{T}^\dagger_{-t} v(t), (T_t^\dagger P T_t)(\overline{T}^\dagger_{-t} v(t)) \rangle}{\langle v(t), v(t) \rangle} \\ &\equiv \frac{\langle v_I(t), P_I(t) v_I(t) \rangle}{\langle v(t), v(t) \rangle}. \end{aligned} \tag{D.3.9}$$

Notice that we need to be careful with the normalization term, as $\langle v(t), v(t) \rangle = \langle v(t), T_t \overline{T}^\dagger_{-t} v(t) \rangle \neq \langle v_I(t), v_I(t) \rangle$.

Now, Eq. (D.3.9) gives us a good starting point for the perturbative analysis. We start by looking at the nominator. First, express $v_I^\dagger, P_I, v_I$ up to first order in the perturbation, i.e., first order in $\zeta$. Then, upon inserting this into Eq. (D.3.9), we again continue by only keeping linear order. Finally, the denominator will receive similar treatment, to estimate impact of the perturbation onto the normalization.

We use notation $\mathcal{T} \exp\big(\int_0^\tau \mathrm{d}\tau' A(\tau')\big) =: T_\tau[A]$, where we add the generator in square brackets. Then, our goal now is to simplify the evolutions $\overline{T}^\dagger_{-t} T_t[A] = (G_{-H^\dagger} \star G_A \star \delta)(t)$ that is used to

express $v_I(t)$. Here it proves useful to consider [Chi+21, Lemma A.2]. Given $A(\tau) = H(\tau) + V(\tau)$, it holds that

$$T_\tau[A] = T_\tau[H]\, T_\tau\big[T_\tau^\dagger[-H^\dagger]\, V\, T_\tau[H]\big], \tag{D.3.10}$$

since we recall form above that the inverse to $T_\tau[H]$ is $\overline{T}^\dagger_{-t} = T^\dagger_\tau[-H^\dagger]$. Using notation as before, $T_\tau = T_\tau[H]$ and $\overline{T}_{-\tau} = T_\tau[-H^\dagger]$ and using a 'test function' $u$, Eq. (D.3.10) becomes

$$T_\tau[A]\, u = (G \star u)(\tau) = T_\tau\, T_\tau\big[\overline{T}^\dagger_{-\tau} V T_\tau\big]\, u = T_\tau\, T_\tau[V_I]\, u \tag{D.3.11}$$

Then,

$$\overline{T}^\dagger_{-t} T_t[A] = \underbrace{\overline{T}^\dagger_{-t} T_t}_{=\mathbb{I}}\, T_t[V_I] = T_t[V_I] \tag{D.3.12}$$

and consequently $v_I(t) = (G_{V_I} \star w)(t)$. The expression for the adjoint element, $v_I^\dagger(t) = (w^\dagger \star G^\dagger_{V_I})(t)$, follows immediately thanks to Eq. (D.3.8). Furthermore,

$$G_{V_I}(t,s) = \mathcal{T} \mathrm{e}^{\int_s^t \mathrm{d}\tau\, \zeta V_I(\tau)}. \tag{D.3.13}$$

We continue by approximating this propagator to linear order in $\zeta$ by expanding the time-ordered exponential,

$$\begin{aligned} G_{V_I}(t,s) &= \sum_{n=0}^\infty \zeta^n \int_s^t \mathrm{d}\tau_1 \int_s^{\tau_1} \mathrm{d}\tau_2 \cdots \int_s^{\tau_{n-1}} \mathrm{d}\tau_n\, V_I(\tau_1) V_I(\tau_2) \cdots V_I(\tau_n) \\ &= \delta(t-s) + \zeta \int_s^t \mathrm{d}\tau_1\, V_I(\tau_1) + O(\zeta^2). \end{aligned} \tag{D.3.14}$$

Moreover,

$$(G_{V_I} \star w)(t) = \int_0^t \mathrm{d}s\, G_{V_I}(t,s) w(s) \approx \int_0^t \mathrm{d}\tau_1 \left(\delta(t-\tau_1) + \zeta \int_{\tau_1}^t \mathrm{d}\tau_2\, V_I(\tau_2)\right) w(\tau_1) \tag{D.3.15}$$

$$= \underbrace{w(t)}_{=:w_0(t)} + \underbrace{\zeta \int_0^t \mathrm{d}\tau_1 \int_{\tau_1}^t \mathrm{d}\tau_2\, V_I(\tau_2) w(\tau_1)}_{=:\zeta w_1(t)}. \tag{D.3.16}$$

We remember that $w(t) = v_0\delta(t) + b(t)$. Now we can insert this in the expectation value $\langle v_I(t), P_I(t) v_I(t)\rangle$ as in Eq. (D.3.9),

$$\begin{aligned} &\langle (G_{V_I} \star w)(t), P_I(t)(G_{V_I} \star w)(t)\rangle \overset{\text{Eq. (D.3.16)}}{\approx} \langle w_0(t) + \zeta w_1(t), P_I(t)(w_0(t) + \zeta w_1(t))\rangle \\ &= \underbrace{\langle w_0(t), P_I(t) w_0(t)\rangle}_{=:(\square)} + \underbrace{\zeta \langle w_0(t), P_I(t) w_1(t)\rangle + \zeta \langle w_1(t), P_I(t) w_0(t)\rangle}_{=:(\#)} + \zeta^2 \langle w_1(t), P_I(t) w_1(t)\rangle. \end{aligned} \tag{D.3.17}$$

Up to normalization, we identify the unperturbed expectation and recall that $T_t w_0(t) = (G_H \star w)(t)$ is the unperturbed evolution,

$$\langle P\rangle_0 := \frac{\langle w_0(t), P_I(t) w_0(t)\rangle}{\langle (G_H \star w)(t), (G_H \star w)(t)\rangle} = \frac{\langle T_t w_0(t), P\, T_t w_0(t)\rangle}{\langle T_t w_0(t), T_t w_0(t)\rangle}, \quad (\square) = \langle P\rangle_0 \|T_t w_0(t)\|^2. \tag{D.3.18}$$

The linear-order terms lead to the following expression, where we use notation $\overline{V}_I(t,s) = \int_s^t \mathrm{d}\tau\, V_I(\tau)$.

$$\begin{aligned}(\#) =&\zeta\Bigg( \left\langle \int_0^t \mathrm{d}\tau_1\, \overline{V}_I(t,\tau_1) w(\tau_1), P_I(t) \int_0^t \mathrm{d}\tau_2\, \delta(t-\tau_2) w(\tau_2) \right\rangle \\ &+ \left\langle \int_0^t \mathrm{d}\tau_1\, \delta(t-\tau_1) w(\tau_1), P_I(t) \int_0^t \mathrm{d}\tau_2\, \overline{V}_I(t,\tau_2) w(\tau_2) \right\rangle \Bigg) \end{aligned} \tag{D.3.19}$$

The following relation will be convenient: $V_I(t) = \overline{T}^\dagger_{-t} V(t) T_t$. Consequently,

$$\begin{aligned}\overline{V}_I(t,s) &= \left( \int_s^t \mathrm{d}\tau\, \overline{T}^\dagger_{-\tau} V(\tau) \right) T_\tau = \overline{T}^\dagger_{-s} \left( \int_0^{t-s} \mathrm{d}\tau\, \overline{T}^\dagger_{-\tau} V(\tau+s) T_\tau \right) T_s \\ &= \overline{T}^\dagger_{-s} \left( \int_s^t \mathrm{d}\tau\, \overline{T}^\dagger_{-(\tau-s)} V(\tau) T_{\tau-s} \right) T_s = \overline{T}^\dagger_{-s} \overline{V}(t,s) T_s \end{aligned} \tag{D.3.20}$$

Then, we may conclude that

$$\begin{aligned}(\#) = \ldots =&\zeta\left( \int_0^t \mathrm{d}\tau\, \left\langle w(\tau), \overline{V}^\dagger_I(t,\tau) T_t^\dagger P T_t w(t) \right\rangle + \int_0^t \mathrm{d}\tau\, \left\langle w(t), T_t^\dagger P T_t \overline{V}_I(t,\tau) w(\tau) \right\rangle \right) \\ &= \zeta\left( \int_0^t \mathrm{d}\tau\, \left\langle T_\tau w(\tau), \overline{V}^\dagger(t,\tau) T^\dagger_{t-\tau} P (T_t w(t)) \right\rangle + \left\langle T_t w(t),\, P T_{t-\tau} \overline{V}(t,\tau) (T_\tau w(\tau)) \right\rangle \right) \end{aligned} \tag{D.3.21}$$

Moving along, we define a 'density'

$$\nu[A](t,\tau) := \left(T_t[A] v(t)\right) \left(T_\tau[A] v(\tau)\right)^\dagger . \tag{D.3.22}$$

If we omit the generator in square brackets of $\nu[H] = \nu$, we assume the unperturbed evolution through $H(t)$. That means that Eq. (D.3.21) becomes

$$\begin{aligned}&\zeta \int_0^t \mathrm{d}\tau\, \mathrm{tr}\left[ \nu(\tau,t)^\dagger \overline{V}^\dagger(t,\tau) T^\dagger_{t-\tau} P + P T_{t-\tau} \overline{V}(t,\tau) \nu(\tau,t) \right] \\ &\qquad = \zeta \int_0^t \mathrm{d}\tau\, \mathrm{tr}\left[ \left\{ P,\, T_{t-\tau} \overline{V}(t,\tau) \nu(\tau,t) \right\}_\sim \right] =: \zeta \mathbb{P}(t) \end{aligned} \tag{D.3.23}$$

with the modified anti-commutator $\{X,Y\}_\sim := XY + Y^\dagger X$.

Next, we consider the normalization term in Eq. (D.3.9), up to first order. Here it also proves useful to use Eq. (D.3.11) which implies that $v(t) = T_t v_I(t)$. Then,

$$\langle T_t v_I(t), T_t v_I(t) \rangle = \left\langle v_I(t), T_t^\dagger T_t v_I(t) \right\rangle \tag{D.3.24}$$

Thus, we can treat Eq. (D.3.24) analogously to the nominator $\langle v_I(t), P_I(t) v_I(t) \rangle$ but replacing $P_I(t)$

with $T_t^\dagger T_t$, up to keeping in mind that we have to divide by this term:

$$\begin{aligned}
&\Big\langle w_0(t) + \zeta w_1(t), T_t^\dagger T_t (w_0(t) + \zeta w_1(t)) \Big\rangle \\
&= \Big\langle w_0(t), T_t^\dagger T_t w_0(t) \Big\rangle + \zeta \left( \Big\langle w_1(t), T_t^\dagger T_t w_0(t) \Big\rangle + \Big\langle w_0(t), T_t^\dagger T_t w_1(t) \Big\rangle \right) + O\big(\zeta^2\big) \\
&= \|T_t w_0(t)\|_{\ell_2}^2 \left( 1 + \zeta \frac{\Big\langle w_1(t), T_t^\dagger T_t w_0(t) \Big\rangle + \Big\langle w_0(t), T_t^\dagger T_t w_1(t) \Big\rangle}{\|T_t w_0(t)\|_{\ell_2}^2} + O\big(\zeta^2\big) \right) \\
&= \|T_t w_0(t)\|_{\ell_2}^2 \left( 1 + \zeta \frac{\mathbb{T}(t)}{\|T_t w_0(t)\|_{\ell_2}^2} + O\big(\zeta^2\big) \right)
\end{aligned} \tag{D.3.25}$$

where $\mathbb{T}(t) = \int_0^t \mathrm{d}\tau \, \mathrm{tr}\left[ \left\{ \mathbb{I}, \overline{V}(t,\tau)\sigma(\tau,t) \right\}_\sim \right]$ by Eqs. (D.3.19) and (D.3.21) to (D.3.23). To find the inverse of the parentheses term in Eq. (D.3.25), consider the Taylor series of $\frac{1}{1+x}$ at $x = 0$ given by $\sum_{k=0}^{\infty} (-x)^k$. Then if we have small $x \ll 1$, we can use the approximation $\frac{1}{1+x} = 1 - x + O(x^2)$. This allows us to express the normalization factor in first order as

$$\frac{1}{\langle v(t), v(t) \rangle} = \frac{1}{\|T_t w_0\|_{\ell_2}^2} \left( 1 - \zeta \frac{\mathbb{T}(t)}{\|T_t w_0\|_{\ell_2}^2} + O(\zeta^2) \right). \tag{D.3.26}$$

The final step now is to assemble Eq. (D.3.9) based on previous results,

$$\begin{aligned}
\langle P(t) \rangle &= \frac{\langle v_I(t), P_I(t) v_I(t) \rangle}{1} \cdot \frac{1}{\langle v(t), v(t) \rangle} \\
&= \Big( \langle P \rangle_0 \cdot \|T_t w_0(t)\|^2 + \zeta \mathbb{P}(t) + O\big(\zeta\big)^2 \Big) \cdot \left( \frac{1}{\|T_t w_0\|^2} \left( 1 - \zeta \frac{\mathbb{T}(t)}{\|T_t w_0\|^2} + O(\zeta^2) \right) \right) \\
&\approx \langle P \rangle_0 + \zeta \frac{\mathbb{P}(t)}{\|T_t w_0(t)\|_{\ell_2}^2} - \zeta \langle P \rangle_0 \mathbb{T}(t) + O\big(\zeta^2\big)
\end{aligned} \tag{D.3.27}$$

And hence,

$$\langle P(t) \rangle - \langle P \rangle_0 = \zeta \frac{\mathbb{P}(t) - \langle P \rangle_0 \mathbb{T}(t) + O(\zeta)}{\|T_t w_0(t)\|^2} \tag{D.3.28}$$

Let us continue by expanding the definitions of $\mathbb{P}(t), \mathbb{W}(t)$,

$$\langle P(t) \rangle - \langle P \rangle_0 = \frac{\zeta}{\mathrm{tr}[\nu(t,t)]} \left( \int_0^t \mathrm{d}\tau \, \mathrm{tr}\left[ \left\{ (P - \langle P \rangle_0), \overline{V}(t,\tau)\sigma(\tau,t) \right\}_\sim \right] + O(\zeta) \right) \tag{D.3.29}$$

Here, we can identify a transfer or response function $\chi(t,t')$ similar to [SDM22, Eq. (1)] and [GH22, Eq. (2)]

$$\chi(t,t') = \mathbf{1}_{[t \geq t']} \frac{\mathrm{tr}\left[ \left\{ (P - \langle P \rangle_0), \overline{V}(t,t')\sigma(t',t) \right\}_\sim \right] + O(\zeta)}{\mathrm{tr}[\sigma(t,t)]} \tag{D.3.30}$$

so that $\delta \langle P(t) \rangle = \zeta \int_0^t \mathrm{d}t' \, \chi(t,t')$. □

## D.4 A lower bound on the final solution norm for the discrete heat equation

**Lemma D.4.1.** *Given* $\frac{\mathrm{d}}{\mathrm{d}t}v(t) = D\mathbf{L}_h v(t) + b(t)$ *with* $v, b \in \mathbb{R}^{n^d}$ *and* $\mathbf{L}_h \in \mathbb{R}^{n^d \times n^d}$ *a discrete Laplacian in periodic boundary conditions. Thus we also have that* $\exists\, \mu_{\min} < 0$ *so that* $-\infty \prec \mu_{\min}\mathbb{I} \preceq \mathbf{L}_h \preceq 0$*. Additionally, this means that* $\|v(s)\|_{\ell_2} \leq \|v(0)\|_{\ell_2}$ *for any* $0 \leq s \leq t$*. Furthermore, in this Lemma, we assume that* $b(s) \geq 0$ *for* $0 \leq s \leq t$*. Then,*

$$\|v(t)\|_{\ell_2}^2 \geq \|v(t)\|_{\ell_2}^2 \geq 2\mathrm{e}^{D\mu_{\min}t}\|v(0)\|_{\ell_2}\|b\|_{L^1[0;t]} \tag{D.4.1}$$

*Proof.* By the governing equation, we have

$$\begin{aligned}\|v(t)\|_{\ell_2}^2 &= \left\|\mathrm{e}^{D\mathbf{L}_h t}v(0) + \int_0^t \mathrm{d}s\, \mathrm{e}^{D\mathbf{L}_h s}b(s)\right\|_{\ell_2}^2 \\ &= \langle v(0), \mathrm{e}^{2D\mathbf{L}_h t}v(0)\rangle + 2\int_0^t \mathrm{d}s\, \left\langle v(0), \mathrm{e}^{D\mathbf{L}_h s}b(s)\right\rangle + \int_0^t \mathrm{d}s \int_0^t \mathrm{d}s'\, \left\langle b(s'), \mathrm{e}^{D\mathbf{L}_h(s+s')}b(s)\right\rangle \\ &:= (\mathrm{I}) + (\mathrm{II}) + (\mathrm{III})\end{aligned} \tag{D.4.2}$$

We can obtain a lower bound using a reverse Cauchy-Schwartz-type inequality due to Pólya and Szegő [HLP34, p. 62, §71], which says that for bounded vectors $x, y$ with positive entries,

$$\langle x, y\rangle^2 \geq C\langle x, x\rangle\langle y, y\rangle \tag{D.4.3}$$

with a constant

$$C = \tfrac{1}{4}\left(\sqrt{\tfrac{\max\{x\}\max\{y\}}{\min\{x\}\min\{y\}}} + \sqrt{\tfrac{\min\{x\}\min\{y\}}{\max\{x\}\max\{y\}}}\right)^2. \tag{D.4.4}$$

For the case of the discretized heat equation here with the solution signifying temperature, the positivity assumption on $v(t)$ holds, for the forcing term we invoke a positivity assumption for the sake of this argument. This is physically reasonable when we think of a heat source. In the case of strong heat sinks, we may expect the solution to fully decay to zero (and not become negative to remain physical); we can ensure non-negativity by shifting the solution by $\|b\|_{L^1[0;t]}$. Then, applying the inequality on term (II),

$$\left\langle \mathrm{e}^{D\mathbf{L}_h t}v(0), \int_0^t \mathrm{d}s\, \mathrm{e}^{D\mathbf{L}_h s}b(s)\right\rangle^2 \geq C\left\langle v(0), \mathrm{e}^{2D\mathbf{L}_h t}v(0)\right\rangle \int_0^t \mathrm{d}s \int_0^t \mathrm{d}s'\, \left\langle b(s'), \mathrm{e}^{D\mathbf{L}_h(s+s')}b(s)\right\rangle. \tag{D.4.5}$$

Using this expression in Eq. (D.4.2),

$$\|v(t)\|_{\ell_2}^2 \geq (\mathrm{I}) + C\sqrt{(\mathrm{I})}\sqrt{(\mathrm{III})} + (\mathrm{III}) \tag{D.4.6}$$

In addition,

$$\mathrm{e}^{\mu_{\min} t} \min_j [v(0)]_j \le [\mathrm{e}^{D\mathbf{L}_h t} v(0)]_j \qquad \le \max_j [v(0)]_j \quad \forall\, j \in [n^d] \tag{D.4.7}$$

$$\mathrm{e}^{\mu_{\min} t} \cdot t \min_{j,0\le s\le t} [b(s)]_j \le \int_0^t \mathrm{d}s\, \mathrm{e}^{D\mathbf{L}_h s} b(s) \qquad \le t \max_{j,0\le s\le t} [b(s)]_j \quad \forall\, j \in [n^d]. \tag{D.4.8}$$

For convenience, we define the amplification factor $Q := \frac{\max_j [v(0)]_j \max_{k,s} [b(s)]_k}{\min_j [v(0)]_j \min_{k,s} [b(s)]_k}$, noting that $Q \ge 1$, and obtain for the constant (where $\mu_{\min} < 0$)

$$C = \frac{1}{4}\left( \underbrace{\mathrm{e}^{-\mu_{\min} t}\sqrt{Q}}_{>1} + \underbrace{\mathrm{e}^{\mu_{\min} t}\sqrt{Q^{-1}}}_{<1} \right)^2 \ge \frac{\mathrm{e}^{2\mu_{\min} t}}{4Q}. \tag{D.4.9}$$

Now,

$$\begin{aligned}
\|v(t)\|_{\ell_2}^2 &\ge \text{(I)} + \sqrt{\frac{\mathrm{e}^{2\mu_{\min} t}}{4Q}\text{(I)}\,\text{(III)}} + \text{(III)} \\
&\ge \mathrm{e}^{2D\mu_{\min} t}\left( \|v_0\|_{\ell_2}^2 + \|b\|_{L^1[0;t]}^2 \right) + \sqrt{\frac{\mathrm{e}^{2D\mu_{\min} t}\|v_0\|_{\ell_2}^2 \|b\|_{L^1[0;t]}^2}{4Q}} \\
&\ge \mathrm{e}^{D\mu_{\min} t}\left( 2 + \frac{1}{\sqrt{Q}} \right) \|v_0\|_{\ell_2} \|b\|_{L^1[0;t]}
\end{aligned} \tag{D.4.10}$$

In the second inequality, we use the bounds $\text{(I)} \ge \mathrm{e}^{2D\mu_{\min} t}\|v(0)\|_{\ell_2}^2$ and $\text{(III)} \ge \mathrm{e}^{2D\mu_{\min} t}\|b\|_{L^1[0;t]}^2$. The third inequality follows from bounding the arithmetic with the geometric mean, $\frac{x+y}{2} \ge \sqrt{xy}$. Recall that $Q \ge 1$ to obtain

$$\|v(t)\|_{\ell_2}^2 \ge 2\mathrm{e}^{D\mu_{\min} t}\|v(0)\|_{\ell_2}\|b\|_{L^1[0;t]}. \tag{D.4.11}$$

□