Universiteit
Leiden
The Netherlands

Capturing dynamics with noisy quantum computers

Dechant, D.S.

Citation

Dechant, D. S. (2026, February 17). *Capturing dynamics with noisy quantum computers*. Retrieved from https://hdl.handle.net/1887/4290771

Version:	Publisher's Version
License:	Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden
Downloaded from:	https://hdl.handle.net/1887/4290771

Note: To cite this publication please use the final published version (if applicable).

CHAPTER 3

Error and Resource Estimates of Variational Quantum Algorithms for Solving Differential Equations Based on Runge-Kutta Methods

3.1 Introduction

For more than four decades, quantum computing has captivated the minds of researchers, but significant experimental advancements have only been achieved in recent years. We are living in the era of Noisy Intermediate-Scale Quantum (NISQ) devices [15, 107–109], which, while capable of certain super classical computations in principle, are also susceptible to noise and errors, preventing them from providing substantial computational advantages over classical computers. Efforts to tackle these challenges have resulted in proliferate research lines dedicated to error correction and noise mitigation across various quantum architectures [110]. One approach to ameliorating some of the issues involves hybrid quantum-classical algorithms that use quantum computers to process information, while classical computers handle the correction and optimization processes, helping to reduce errors and improve efficiency. For example, in variational quantum algorithms, such as the Variational Quantum Eigensolver (VQE) or the Quantum Approximate Optimization Algorithm (QAOA), the

The contents of this chapter have been published in Ref. [46].

quantum part evaluates the cost function depending on a given set of parameters, while the classical part optimizes these parameters to minimize the cost function. In physical terms, the cost function is typically the energy of a Hamiltonian that encodes the optimization task. Extracting the final result from such a hybrid device is challenging, as errors arise both from quantum and classical sources. A notable source of error is the measurement shot noise arising out of the discrete nature of quantum measurements and the limited number of measurements taken [111].

Solving differential equations (DEs) is a critical task in various scientific and engineering fields, and several quantum computing-based proposals have been developed to tackle this problem. First ideas [35, 36, 112] were built around the quantum linear system algorithm [37], but they require fault-tolerant quantum computers. Later, approaches based on variational quantum algorithms were introduced, e.g., in Refs. [26, 113–115]. In one of the latter approaches, the DE is mapped to an imaginary-time Schrödinger equation [116]. It is solved by a variational algorithm, where the time evolution of a quantum state is approximated by a variational quantum circuit and mapped to the time evolution of the parameters of this circuit. After classically computing the parameters at the final time step, one reinserts them into the variational quantum circuit to prepare the evolved quantum state. This is an approach originally proposed in Ref. [117] for quantum simulation and has attracted a lot of interest since [118–124].

In the approach of Refs. [116, 117], the resulting time evolution of the quantum circuit parameters has the form of an ordinary differential equation (ODE). Classically, ODEs can be solved by time-discretization methods such as the Euler method or the more general Runge-Kutta methods [125] (RKMs) that can be categorized by their order p. The first proposed variational quantum algorithm for solving differential equations based on the Euler method is shown in Ref. [113]. Those methods approximate the time evolution by calculating a truncated Taylor expansion at each time step, and the higher the order of the RKM, the lower is the resulting truncation error, but the higher is the required number of function evaluations. There are also different generalizations of the Euler methods, like the linear multistep methods or the general linear methods [125], which can lead to similar accuracies as RKMs with different requirements of the number of function evaluations.

Previous works have proposed that choosing RKMs, such as the widely used classical RKM [126] with the order $p = 4$ instead of the Euler method, will also be favorable in solving the time evolution of the quantum circuit parameters [117, 127, 128]. However, each function evaluation incorporates evaluations of quantum circuits, making higher-order RKMs more demanding on the quantum resources. Moreover, quantum circuit evaluations are affected by the "shot noise bottleneck," where the error scales as $\mathcal{O}(1/N_{meas}^2)$ with the number of measurements N_{meas}, introducing an additional source of error. The

number of measurements is a precious resource in quantum computing, as it is the most costly and time-consuming operation and as it is severely limited by the available runtime of the device before it requires recalibration [107, 129, 130]. That is why, for the practicality of the variational algorithm, it is crucial to minimize the total number of quantum circuit measurements.

In this chapter, we focus on solving differential equations based on the approach of Refs. [116, 117]. We investigate whether higher-order RKMs outperform the Euler method in solving the time evolution of the circuit parameters by analyzing the different sources of error and resource requirements. Specifically, we provide a detailed analysis of the total error of the variational quantum algorithm for solving differential equations, defined as the trace distance between the actual solution and the output of the variational algorithm. We explicitly consider the truncation error associated with the chosen RKM and the shot noise error. Other relevant sources of errors, such as circuit error (gate infidelity, bias, SPAM errors) and the representation error (the variational circuit being able to represent the solution at all time steps with its parameters), are assumed ideal as they depend among others on the chosen Ansatz and problem instance [131]. We establish rigorous error bounds and use them to estimate a sufficient number of circuit evaluations required by the algorithm for a given target error, based on the order of the RKM. Additionally, we perform an analysis of the RKMs under the assumption of no shot noise.

Further, we validate the resource estimates through benchmarking: the analysis without shot noise is demonstrated with a simple ODE, while the analysis of the variational algorithm is applied to option pricing, where the dynamics are described by the Black-Scholes equation [26, 81]. The latter is a partial differential equation that has attracted a lot of attention in the variational quantum computing community [26, 128, 132]. This shows that the application of our method is not restricted to ODEs but can also be applied to solving partial differential equations. We directly compare the total number of circuit evaluations required by the algorithm, depending on the order of the chosen RKM. In this chapter, we derive rigorous error bounds that are general, but may overestimate the true error in practice. Our resource estimates are based on optimizing resources with respect to these upper bounds and might therefore be overly conservative. Such an approach constitutes the optimal parameter selection strategy, providing a guaranteed success probability.

Similar error and resource estimates have been conducted in Refs. [116, 133–135]. However, our analysis is integrating the truncation errors of RKMs and shot noise errors, making it comprehensive, and offers direct comparisons of resource requirements between different RKMs and an a-priori analysis that leads to substantial savings in the cost of the algorithms.

Our work is relevant not only to quantum algorithms that use variational approaches but also to those that employ RKMs for solving DEs, such as in Ref. [112, 136]. Our analysis highlights the importance of studying the

sensitivity of classical numerical methods for ODEs to perturbations in the input function, which has hardly been explored so far.

From our results, we conclude that depending on the parameters distinct to the problem at hand, higher-order RKMs are decreasing the resource requirements. In the use case of option pricing via the Black-Scholes equation, we showed that an RKM of order $p = 2$ is requiring the minimal number of total quantum circuit evaluations. For other applications, even higher-order methods might be favorable. With our thorough analysis of the involved parameters, we provide a straightforward framework that can be applied to other use cases that can be tackled by solving a DE in the form of Eq. (3.11) and decrease the resource requirements of the variational algorithms by suggesting the most efficient RKM.

In the interest of making this chapter self-contained, we build up the chapter in the following way: We begin with an introduction to the RKM and the variational quantum algorithm from [116] and with the problem statement. In Sec. 3.3-3.4, we show estimates of the errors and minimal resources required for ODE solving without and with shot noise. In Sec. 3.5, we analyze parameters of the variational algorithm that the error and resource estimates depend on. Afterwards, we are performing numerical analyzes of a simple ODE without shot noise and of an option pricing use case in Sec. 3.6. In Sec. 3.7, we provide a discussion of the results and conclusions.

3.2 Preliminaries

In this chapter, we investigate the impact of different types of errors on variational quantum algorithms for solving DEs based on Runge-Kutta methods. These algorithms are motivated by the variational quantum algorithm for imaginary time evolution introduced in Ref. [116]. In the following, we firstly present the Runge-Kutta methods, which is a family of classical methods for solving ODEs. Secondly, we present the variational quantum algorithm for solving DEs that are based on the Runge-Kutta methods. And thirdly, we introduce the errors and resources that we analyze in this chapter.

3.2.1 Runge-Kutta methods

Let us consider the initial value problem, which is an ODE together with an initial condition:

$$\begin{aligned} \frac{dy(\tau)}{d\tau} &= f\left(\tau, y(\tau)\right) \\ y(\tau_0) &= y_0 \ , \end{aligned} \tag{3.1}$$

where τ is the time and $y(\tau)$ is an element of the image of an unknown function in a scalar or vector form that we want to determine, and where $f\left(\tau, y(\tau)\right)$ fulfills

the assumptions of the Picard-Lindelöf theorem, guaranteeing the existence of a unique solution to Eq. (3.1).

Since most of the time an analytically closed form is not viable, numerical methods are the only way to obtain an approximate solution. A common way of solving Eq. (3.1) is with the so-called Runge-Kutta methods (RKMs). The RKMs are a class of methods based on the Taylor expansion of y in order to approximate the numerical solution of the ODE at the future time step by using evaluations of $f(\tau, y(\tau))$.

A general blueprint of RKMs with s stages can be outlined as follows: For simplicity, let us set $\tau_0 = 0$. We divide the time interval $\tau \in [0, T]$, $T > 0$ into N_τ time steps denoted by τ_n, $n \in \{1, ..., N_\tau\}$. We assume the step size $\Delta\tau = \tau_n - \tau_{n-1} = T/N_\tau$ to be constant and denote the computed solution at the n-th time step by $y(\tau_n)$. Using Eq. (3.1) and the solution $y(\tau_n)$ at the n-th time step, we can compute $y(\tau_{n+1})$ in the following way (see Ref. [137, p.907]):

$$y(\tau_{n+1}) = y(\tau_n) + \Delta\tau \sum_{i=1}^{s} b_i k_i \ , \tag{3.2}$$

where the calculation of the latter function is done in s stages

$$\begin{aligned} k_1 &= f(\tau_n, y(\tau_n)), \\ k_2 &= f(\tau_n + c_2\Delta\tau, y(\tau_n) + a_{21}k_1\Delta\tau), \\ k_3 &= f(\tau_n + c_3\Delta\tau, y(\tau_n) + (a_{31}k_1 + a_{32}k_2)\Delta\tau), \\ &\vdots \\ k_s &= f(\tau_n + c_s\Delta\tau, y(\tau_n) + (a_{s1}k_1 + a_{s2}k_2 + \cdots + a_{s,s-1}k_{s-1})\Delta\tau). \end{aligned} \tag{3.3}$$

The constants $a_{i,j}$ $(1 \leq j < i \leq s)$, b_i $(1 \leq i \leq s\ )$ and c_i $(2 \leq i \leq s\ )$ are specific for each RKM. In order to be consistent, the constants have to satisfy

$$\sum_{i=1}^{s} b_i = 1, \quad \text{and} \quad \sum_{j=1}^{i-1} a_{i,j} = c_i, \quad \text{for} \quad 2 \leq i \leq s. \tag{3.4}$$

For later analysis, we define the maxima of these parameters for a specific RKM in the following way:

$$b_{max} = \max_i |b_i|, \tag{3.5}$$

$$a_{max} = \max_{i,j} |a_{i,j}| \ . \tag{3.6}$$

The simplest RKM is the Euler method ($s = 1$), which approximates the

function in one stage iteratively as follows:

$$y(\tau_i + \Delta\tau) = y(\tau_i) + \Delta\tau f\left(\tau_i, y(\tau_i)\right). \tag{3.7}$$

The estimation error ℓ_i induced at each time step τ_i due to the truncation of the Taylor series is referred to as the local truncation error (LTE) of the method. RKMs are classified according to the error scaling of their LTE. A RKM is said to have an order p if the LTE is bounded by an error that scales as $\mathcal{O}(\Delta\tau^{p+1})$. Generally, the order and the number of stages of an RKM are related by $s = p$ for $1 \leq p \leq 4$, and $s > p$ for $p \geq 5$. This discrepancy is due to the fact that finding the coefficients $a_{i,j}, b_i$ and c_i becomes increasingly difficult for higher values of p as it involves solving a system of non-linear equations that becomes more complicated for higher p. For higher-order methods this can only be achieved with an increasingly higher number of stages s. To the best of our knowledge, there is no closed formula for calculating the minimum number of stages required for a specific order. The relations up to order $p = 10$ are provided in Table 3.1.

Order p	**Number of stages s**
5	6
6	7
7	9
8	11
9	13
10	16

Table 3.1: Relation between order and the minimum number of stages of RKMs [125, 138, 139]

The following theorem provides an upper bound on the LTE of a p-th order RKM:

Theorem 3.1. *(See Ref. [140, p.180]) The LTE ℓ_n of the p-th order RKM at the step $n \in [1, N_\tau]$ is bounded by*

$$\|\ell_n\| = \|y(\tau_n) - y_n\| \leq \Delta\tau^{p+1} K L_{f\tau}^p M + \mathcal{O}(\Delta\tau^{p+2}) \,, \tag{3.8}$$

where $\|\cdot\|$ denotes a norm, which can be any norm on the state space (e.g., Euclidean, maximum, or 1-norm), as the bound is independent of the specific choice. Here, y_n is the RKM approximation of $y(\tau_n)$ calculated at step n, assumed that the value $y(\tau_{n-1})$ is exact. Further, $K > 0$ is a constant depending on the chosen RKM and $f(\tau_n, y(\tau_n))$ can be upper bounded by $M > 0$ and is Lipschitz-continuous with respect to the first variable, such that the following

bounds hold:

$$\|f(\tau_n, y(\tau_n))\| \leq M, \quad \left\| \frac{\partial f(\tau_n, y(\tau_n))}{\partial \tau} \right\| \leq L_{f\tau}, \ldots, \tag{3.9}$$

$$\left\| \frac{\partial^i f(\tau_n, y(\tau_n))}{\partial \tau^i} \right\| \leq \frac{L_{f\tau}^i}{M^{i-1}} \ , \tag{3.10}$$

for all $1 \leq i \leq p$, $0 < \tau_n < T$ *and* $f(\tau_n, y(\tau_n)) := \frac{\partial y(\tau)}{\partial \tau}\Big|_{\tau_n}$.

Unless stated otherwise, the same norm is used consistently throughout the analysis for states, errors, Lipschitz bounds, and the corresponding induced operator norms. Since all norms on finite-dimensional spaces are equivalent, the order of the error bounds remains unaffected by the specific choice of norm, with only the constants differing. In our setting, we use the trace norm for quantum states and the Euclidean (2-)norm, together with its induced operator norm, for all other quantities.

We are estimating the constants K, M and $L_{f\tau}$ in Secs. 3.6.1 and 3.6.2 for the specific examples of DEs that we cover numerically in Sec. 3.6.

3.2.2 Variational quantum algorithm for solving differential equations

In this section, we present the variational algorithm for solving linear differential equations of the following type:

$$\frac{dy(\tau)}{d\tau} = -\mathcal{H} \cdot y(\tau) \tag{3.11}$$

$$y(\tau_0) = y_0 \ , \tag{3.12}$$

where $y(\tau) \in \mathbb{C}^d$ is a vector and $\mathcal{H}$ a $d \times d$-dimensional matrix. We assume w.l.o.g. that d is a power of 2 and that $\mathcal{H}$ is Hermitian. This differential equation is a special case of the initial value problem as stated in Eq. (3.1).

It is possible to map a wide variety of DEs to Eq. (3.11), such as stochastic DEs (see Ref. [128]), the Black-Scholes partial DE (see our analysis in Sec. 3.6.2 and Ref. [26]), or other linear partial DEs (see Ref. [141]). Typically, this mapping involves a discretization of the underlying space and differential operators onto a grid [26, 128, 141], but it is also possible to encode the solution via spectral methods like Chebyshev polynomials or in Fourier basis (see for example Ref. [44], although they use a different quantum algorithm).

If d is not a power of 2, it is always possible to embed $\mathcal{H}$ and $y(\tau)$ into higher-dimensional spaces, such that the assumption holds. The resulting higher dimensional space into which $y(\tau)$ is embedded is less than a factor of 2 larger

than the original space.

If $\mathcal{H}$ is not Hermitian, several strategies can be applied: For some differential equations, it is possible to apply changes of variables in order to transform $\mathcal{H}$ to a Hermitian matrix, such as the transformation done in Ref. [26]. Alternatively, it is possible to divide any matrix $\mathcal{H}$ into a Hermitian and an anti-Hermitian part, which effectively leads to a doubling in the number of circuit evaluations, as demonstrated in Ref. [142]. It is also possible to use the technique shown in Ref. [143] that gives a mapping from Eq. (3.11) with a non-Hermitian $\mathcal{H}$ to the real time Schrödinger equation, which one can solve with the variational algorithm introduced in Ref. [117]. Further, there exists a generalization of the variational algorithm for imaginary time evolution that we present here which can be applied to linear differential equations with non-Hermitian matrices $\mathcal{H}$ without the need to embed them first into higher-dimensional Hilbert spaces [144]. Note that our analysis can easily be adapted to this generalized variational algorithm, as well as to the variational algorithm solving the real time Schrödinger equation.

The variational quantum algorithm that solves Eq. (3.11), is based on the variational quantum algorithm for imaginary time evolution introduced in Ref. [116]. Therefore, we firstly bring Eq. (3.11) into the form of quantum imaginary time evolution.

For the rest of the chapter, we assume that $\mathcal{H}$ is Hermitian. The matrix $\mathcal{H}$ can thus be decomposed in the following way:

$$\mathcal{H} = \sum_{m=1}^{N} \lambda_m \sigma_m \ . \tag{3.13}$$

Here, $\{\lambda_m\}_{m=1}^N \in \mathbb{R}$ are the decomposition coefficients and $\{\sigma_m\}_{m=1}^N$ are the Pauli strings, which are tensor products of single qubit Pauli matrices and the identity. This possible decomposition always exists and is unique, since the collection of d^2 Pauli strings form an orthogonal basis for Hermitian operators acting on a d dimensional Hilbert space. The operator $\mathcal{H}$ plays the role of the Hamiltonian of a quantum system. Effectively replacing the real time parameter t from the Schrödinger equation with $-i\tau$, where τ is a real number, Eq. (3.11) can then be considered as a Schrödinger equation in imaginary time.

Let us further realize the function $y(\tau)$ as a quantum state $|\psi(\tau)\rangle \in \mathbb{C}^d$ and the initial condition $y(\tau_0) = y_0$ as a state $|\psi(0)\rangle \in \mathbb{C}^d$. That can be done by normalizing the entries of the vector $y(\tau)$ and by subsequently encoding the resulting entries as the amplitudes of $|\psi(\tau)\rangle$:

$$|\psi(\tau)\rangle = \sum_{i=0}^{d-1} \frac{y(\tau)_i}{\sqrt{\|y(\tau)\|_2^2}} |i\rangle \ , \tag{3.14}$$

where $\{|i\rangle\}_{i=0}^{d-1}$ is the computational basis.

The problem of solving Eq. (3.11) then comes down to simulating the corresponding non-unitary time evolution $V(\tau) = e^{-\mathcal{H}\tau}$ that solves the following equation:

$$\frac{d\,|\psi(\tau)\rangle}{d\tau} = -\mathcal{H} \cdot |\psi(\tau)\rangle \tag{3.15}$$

with the initial state $|\psi(0)\rangle$ at time $\tau = 0$. The goal is then to calculate the evolved state $|\psi(T)\rangle$ at time T, which can be calculated from the initial state as:

$$|\psi(T)\rangle = \gamma(T)V(T)\,|\psi(0)\rangle\,, \tag{3.16}$$

$$\text{with the normalization} \quad \gamma(\tau) = \left(\langle\psi(0)|\,V(2T)\,|\psi(0)\rangle\right)^{-1/2}\,. \tag{3.17}$$

While reading out the amplitudes of $|\psi(T)\rangle$ itself will only give the ratio of each basis vector in the solution $y(T)$ of the DE in Eq. (3.11), keeping track of the initial normalization from the mapping of $y(\tau_0)$ to $|\psi(0)\rangle$ and the intermediate normalization factors $\gamma(\tau)$ will give the resulting renormalization factor. One can identify $\gamma(\tau)$ as one of the parameters that is updated at each step (see for example Refs. [113, 128]).

As proposed in Ref. [116], the evolution of the state $|\psi(\tau)\rangle$ can be simulated by using a parameterized quantum circuit to approximate the evolved state. Instead of $|\psi(\tau)\rangle$, the variational quantum "trial" state

$$|\phi(\boldsymbol{\theta}(\tau))\rangle := R(\boldsymbol{\theta}(\tau))\,|\overline{0}\rangle\,, \quad |\phi(\boldsymbol{\theta}(\tau))\rangle \in \mathbb{C}^d, \tag{3.18}$$

prepared by a variational circuit $R(\boldsymbol{\theta}(\tau))$ with a vector of time-dependent parameters $\boldsymbol{\theta}(\tau) = (\theta_1(\tau), \theta_2(\tau), \ldots, \theta_{N_V}(\tau))^T \in \mathbb{R}^{N_V}$ is taken. The trial state is often referred to as the *Ansatz*. The circuit is chosen in such a way that it consists of a sequence of N_V layers each depending on one variational parameter as follows:

$$R(\boldsymbol{\theta}(\tau)) = R_{N_V}(\theta_{N_V}(\tau))R_{N_V-1}(\theta_{N_V-1}(\tau))\ldots R_1(\theta_1(\tau))\,. \tag{3.19}$$

Each $R_k(\theta_k(\tau))$, $k \in (1, \ldots, N_V)$ is a unitary operator that can be written as

$$R_k(\theta_k(\tau)) = \exp\left\{\left(\theta_k(\tau)\sum_{i=1}^{N_d} f_{k,i}\sigma_{k,i}\right)\right\}\,, \tag{3.20}$$

with fixed complex parameters $f_{k,i}$ and Pauli strings $\sigma_{k,i}$. For simplicity, we keep N_d fixed. In general, only a small subspace of the Hilbert space can be reached with such an Ansatz, but as was shown in Ref. [145], this is sufficient for physically relevant states. Furthermore, this Ansatz captures a wide class of

possible implementations, such as the coupled cluster Ansatz [146] or hardware efficient methods [147, 148].

The idea of the method is to map the time evolution of the state $|\psi(\tau)\rangle$ according to the Hamiltonian $\mathcal{H}$ to a time evolution of the parameters $\boldsymbol{\theta}(\tau)$ of the state $|\phi(\boldsymbol{\theta}(\tau))\rangle$. To this end, one first finds the vector of parameters $\boldsymbol{\theta}(\tau = 0)$ such that it minimizes the distance $\| \, |\phi(\boldsymbol{\theta}(0))\rangle - |\psi(0)\rangle \, \|$. The McLachlan's variational principle [149, 150] as given by

$$\delta \| (d/d\tau + \mathcal{H}) \, |\phi(\boldsymbol{\theta}(\tau))\rangle \, \| = 0 \tag{3.21}$$

is fulfilled if Eq. (3.15) is valid for the trial state $|\phi(\boldsymbol{\theta}(\tau))\rangle$. Here $\| \, |\phi\rangle \, \| = \sqrt{\langle\phi|\phi\rangle}$ is the Hilbert space norm of a vector $|\phi\rangle$ and δ denotes an infinitesimal variation. However, if the chosen Ansatz is not expressive enough or too biased, Eq. (3.21) will not hold. The resulting errors are hard to control in practice, but there exist ways to estimate them. See for example Ref. [131, 134].

Eq. (3.21) is used to translate the time evolution of the trial state $|\phi(\boldsymbol{\theta}(\tau))\rangle$ to a time evolution of the vector of parameters $\boldsymbol{\theta}(\tau)$, given as an ODE (see Appendix 3.D):

$$\sum_{l=1}^{N_V} A_{kl} \left(\boldsymbol{\theta}(\tau)\right) \frac{\partial \theta_l(\tau)}{\partial \tau} = C_k \left(\boldsymbol{\theta}(\tau)\right) \quad \forall \tau \ . \tag{3.22}$$

Note that this ODE is acting on the N_V-dimensional vector $\boldsymbol{\theta}(\tau)$, where N_V is independent of the dimension d of the time evolution in Eq. (3.15), and is entirely determined by the number of parameters of the chosen Ansatz. In particular, N_V is not bound to be a power of 2. In principle, lower N_V will reduce the total number of circuit evaluations for our algorithm (see Sec. 3.5.1), but will also lead to a lower expressivity of the corresponding Ansatz.

The elements of the matrix A and the vector C are:

$$A_{kl} \left(\boldsymbol{\theta}(\tau)\right) = \mathfrak{Re} \left(\frac{\partial \, \langle\phi(\boldsymbol{\theta}(\tau))|}{\partial \theta_k} \frac{\partial \, |\phi(\boldsymbol{\theta}(\tau))\rangle}{\partial \theta_l} \right), \tag{3.23}$$

$$C_k \left(\boldsymbol{\theta}(\tau)\right) = \mathfrak{Re} \left(-\frac{\partial \, \langle\phi(\boldsymbol{\theta}(\tau))|}{\partial \theta_k} \mathcal{H} \, |\phi(\boldsymbol{\theta}(\tau))\rangle \right). \tag{3.24}$$

Taking the derivatives of Eq. (3.20) with respect to the parameters, one can calculate the derivative of the trial state in Eq. (3.18):

$$\frac{\partial \, |\phi(\boldsymbol{\theta}(\tau))\rangle}{\partial \theta_k} = \sum_{i=1}^{N_d} f_{k,i} R_{k,i} \, |\overline{0}\rangle \, , \tag{3.25}$$

$$R_{k,i} = R_{N_V} R_{N_V - 1} ... R_{k+1} R_k \sigma_{k,i} R_{k-1} \dots R_2 R_1, \tag{3.26}$$

where we omitted the dependencies of the R_ks on θ_ks for simplicity. With the chosen *Ansatz* in Eq. (3.26), the matrix elements in Eq. (3.23) and in Eq. (3.24) can be computed as:

$$A_{k,l}\left(\boldsymbol{\theta}(\tau)\right) = \sum_{i,j=1}^{N_d} \left(f^*_{k,i} f_{l,j} \left\langle \overline{0} \right| R^\dagger_{k,i} R_{l,j} \left| \overline{0} \right\rangle + h.c. \right), \tag{3.27}$$

$$C_k\left(\boldsymbol{\theta}(\tau)\right) = \sum_{i=1}^{N_d} \sum_{m=1}^{N} \left(f^*_{k,i} \lambda_m \left\langle \overline{0} \right| R^\dagger_{k,i} \sigma_m R \left| \overline{0} \right\rangle + h.c. \right), \tag{3.28}$$

by measuring circuits illustrated in Fig. 3.1. Alternatively, it is possible to calculate the matrix element with parameter-shift rules [92, 151], making the circuits shorter in depth and therefore potentially more suitable for NISQ applications. They are applied to similar algorithms in Ref. [135, 152].

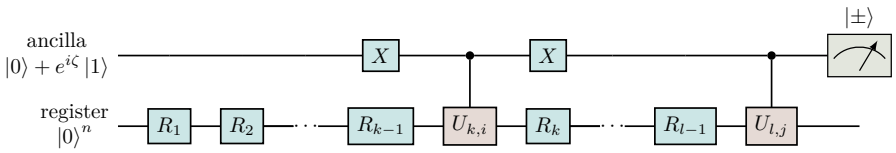

Figure 3.1: The quantum circuit evaluating the elements of A and C. The controlled unitary $U_{k,i}$ is one of the $\sigma_{k,i}$. Depending on if one is evaluating A or C, the controlled unitary $U_{l,j}$ is another $\sigma_{l,j}$ or one of the Pauli strings σ_m (in which case we take $l = N_V + 1$) that constitute the Hamiltonian, respectively.

If the matrix $A\left(\boldsymbol{\theta}(\tau)\right)$ is invertible for all $\boldsymbol{\theta}(\tau)$ in the relevant domain, then Eq. (3.22) can be written in the form of Eq. (3.1) by identifying $y(\tau)$ with $\boldsymbol{\theta}(\tau)$:

$$\frac{\partial \boldsymbol{\theta}(\tau)}{\partial \tau} = f\left(\boldsymbol{\theta}(\tau)\right) := A^{-1}\left(\boldsymbol{\theta}(\tau)\right) C\left(\boldsymbol{\theta}(\tau)\right), \tag{3.29}$$

and with the initial condition $\boldsymbol{\theta}(0)$ at time $\tau = 0$. The time evolution of the parameters $\boldsymbol{\theta}(\tau)$ is given as this ODE, and solving it until final time T gives the vector of parameters $\boldsymbol{\theta}(T)$ that yield the final state $|\phi(\boldsymbol{\theta}(T))\rangle$, which serves as an approximation of the evolved state $|\psi(T)\rangle$. If A is not invertible, regularization introduces an additional error (see Secs. 3.2.3 and 3.5.2). Similarly, the inversion itself will inevitably also include an additional error that has to be taken into account. As long as the errors stemming from regularization and inversion of A fulfill the error bounds in this chapter (such as Eq. (3.75)), our estimates are still applicable. We assume therefore for the remainder of this chapter that A is invertible.

We can thus solve a DE of the form in Eq. (3.11) with a hybrid quantum-classical algorithm by solving the ODE in Eq. (3.22) defining the parameters

$\boldsymbol{\theta}(\tau)$ of an Ansatz state.

In the next section, we introduce the errors and resources required for solving the DEs in Eqs. (3.1) and (3.11) with the Runge-Kutta methods and with the variational quantum algorithm, respectively.

3.2.3 Errors and resources

In this chapter, we analyze errors and resource requirements of the Runge-Kutta methods and the variational algorithm as described above. For this, we focus on analyzing the errors involved in solving the ODE using the RKM described in Sec. 3.2.1, and for the variational algorithm additionally the total error arising from preparing the trial state $|\phi(\boldsymbol{\theta}(T))\rangle$ defined in Eq. (3.18) using the method described in Sec. 3.2.2. In this chapter, we are not considering representation errors stemming from the chosen Ansatz state not being expressive enough, as it proves very challenging to estimate these errors in general [153]. However, they have an influence on both errors that we estimate, by playing a role in the approximation of the time evolution of $\boldsymbol{\theta}(\tau)$ and in the approximation of the final state $|\psi(T)\rangle$. They are specific to the task at hand, it is possible to derive a posteriori error bounds and they become less relevant for deep Ansätze, see for example Refs. [131, 134]. Also, we assume circuit error such as gate infidelity, bias and SPAM errors to be negligible. We further assume the matrices A as defined in Eq. (3.27) to be invertible (see our discussion in Sec. 3.5). In practice, it might be necessary to introduce matrix regularizations for the cases where A is not invertible (see techniques in Refs. [26, 116, 128, 152]), which would lead to additional errors.

We denote the total error arising while approximating the solution of the ODE (Eq. (3.1)) for noiseless evaluations of the functions $f(\tau, y(\tau))$:

$$\epsilon_{ODE}^{(0)} := \|y(\tau_{N_\tau}) - y_{N_\tau}\|_2 \ , \tag{3.30}$$

and analyze it in Sec. 3.3 (Thm. 3.2). For the variational quantum algorithm, the parameters calculated via an RKM from Eq. (3.29) additionally are influenced by shot noise in the evaluations of $f(\boldsymbol{\theta}(\tau))$, which we denote by the superscript δ:

$$\epsilon_{ODE}^{(\delta)} := \|\boldsymbol{\theta}(\tau_{N_\tau}) - \hat{\boldsymbol{\theta}}_{N_\tau}\|_2 \ . \tag{3.31}$$

We added the hat to the approximation of $\boldsymbol{\theta}_{N_\tau}$ of the parameters calculated via an RKM, in order to show that they may be perturbed (e.g., from shot noise). We analyze $\epsilon_{ODE}^{(\delta)}$ in Sec. 3.4.

Whenever the norm is not specified, we are using the 2-norm:

$$\|\boldsymbol{\theta}(\tau_{N_\tau}) - \hat{\boldsymbol{\theta}}_{N_\tau}\|_2 := \left(\sum_{i=1}^{N_V} |\boldsymbol{\theta}(\tau_{N_\tau})_i - \hat{\boldsymbol{\theta}}_{N_\tau,i}|^2\right)^{1/2} , \tag{3.32}$$

where N_V is the length of the vectors $\boldsymbol{\theta}(\tau_{N_\tau})$ and $\hat{\boldsymbol{\theta}}_{N_\tau}$.

We define the total error arising from applying the variational algorithm to approximate the state $|\psi(T)\rangle$ with the trial state $|\phi(\hat{\boldsymbol{\theta}}_{N_\tau})\rangle$ as:

$$\epsilon_{total} := \|\psi(T) - \phi(\hat{\boldsymbol{\theta}}_{N_\tau})\|_1 , \tag{3.33}$$

where $\psi(T) = |\psi(T)\rangle\langle\psi(T)|$ and $\phi(\hat{\boldsymbol{\theta}}_{N_\tau}) = |\phi(\hat{\boldsymbol{\theta}}_{N_\tau})\rangle\langle\phi(\hat{\boldsymbol{\theta}}_{N_\tau})|$, where $|\psi(T)\rangle$ and $|\phi(\hat{\boldsymbol{\theta}}_{N_\tau})\rangle$ are as defined in Eqs. (3.14) and (3.18). For ϵ_{total}, we use the trace distance as the most convenient distance for quantum states, which for two pure states is defined as:

$$\|\psi(T) - \phi(\hat{\boldsymbol{\theta}}_{N_\tau})\|_1 := \sqrt{1 - |\langle\psi(T)|\phi(\hat{\boldsymbol{\theta}}_{N_\tau})\rangle|^2} . \tag{3.34}$$

We analyze the error ϵ_{total} in Sec. 3.5.1.

Based on the error estimates, we are estimating the resources needed in order for executing the RKMs and the variational algorithm. We define the cost of solving an ODE with an RKM as the total number of times that the function $f(\boldsymbol{\theta}(\tau))$ has to be evaluated for the whole RKM:

$$C(N_\tau, N_r, s, p) := sN_\tau(s,p)N_r(s,p) , \tag{3.35}$$

where N_r is the number of measurements of the function $f(\boldsymbol{\theta}(\tau))$ at a single stage of one RKM time step. In the absence of shot noise, the cost in Eq. (3.35) reduces to

$$C(N_\tau, s, p) := sN_\tau(s,p) , \tag{3.36}$$

as at each stage and time step, one needs only one evaluation of $f(\tau, y(\tau))$.

We are estimating the minima of the costs in Eq. (3.35) and in Eq. (3.36) required to reach a specified target error $\epsilon^{(\delta)}_{target}$ that upper bounds the error $\epsilon^{(\delta)}_{ODE} \leq \epsilon^{(\delta)}_{target}$ in Sec. 3.3 and 3.4.

For the variational algorithm described in Sec. 3.2.2, we also determine the total number of quantum circuit evaluations required. As we will demonstrate, calculating $f(\boldsymbol{\theta}(\tau))$ for a given input $\boldsymbol{\theta}(\tau)$ requires the preparation and measurement of several quantum circuits. This significantly increases the total number of quantum circuit evaluations beyond the cost in Eq. (3.35). We refer to this as N_{circ}, and we estimate it in Sec. 3.5.1.

3.3 Runge-Kutta methods without shot noise

For ease of exposition, in this section we analyze the total error arising from solving an ODE with an RKM without the presence of shot noise in the evaluations of $f(\tau, y(\tau))$ in Eq. (3.1). We also determine the minimal number $N_\tau^{(0)}$ of RKM steps required to achieve a prescribed accuracy.

In Thm. 3.2, we establish an upper bound on the total error, denoted $\epsilon_{ODE}^{(0)}$ in Eq. (3.30), by analyzing the error propagation due to the truncation error of the RKM (see Thm. 3.1).

We select the upper bound of the error $\epsilon_{ODE}^{(0)}$ as a target error, meaning the maximal error we can expect. This target error is the used to determine the minimal number of RKM steps needed to ensure that the resulting error remains within the target. The result is formalized in Thm. 3.3, and the corresponding minimal cost, as defined in Eq. (3.36), is derived in Corollary 1.

Theorem 3.2. *Let $y(\tau_{N_\tau})$ be the solution of Eq. (3.1). We assume that the assumptions of Thm. 3.1 hold. Let us further assume that there exists a Lipschitz constant L_{fy}, such that $\forall y_1(\tau_n), y_2(\tau_n)$ in the spaces $\{y(\tau_n) : \tau_n \in [0,T]\}$ and $\{y_n : n \in [1, N_\tau]\}$, the following holds:*

$$\|f(\tau_n, y_1(\tau_n)) - f(\tau_n, y_2(\tau_n))\| \leq L_{fy}\|y_1(\tau_n) - y_2(\tau_n)\| \ , \forall \tau_n \in [0,T]. \quad (3.37)$$

Then, the approximation y_{N_τ} of the solution $y(\tau_{N_\tau})$ at time τ_{N_τ} calculated via a p-th order RKM with s stages in N_τ time steps has the error defined in Eq. (3.30) bounded by:

$$\epsilon_{ODE}^{(0)} \leq \frac{(1 + F(N_\tau, s))^{N_\tau} - 1}{F(N_\tau, s)} \left(\frac{T}{N_\tau}\right)^{p+1} K L_{f\tau}^p M \ , \quad (3.38)$$

where we used the notation

$$F(N_\tau, s) := \frac{b_{max}}{a_{max}} \left(\left(1 + L_{fy} a_{max} \frac{T}{N_\tau}\right)^s - 1 \right). \quad (3.39)$$

For a definition of a_{max} and b_{max} see Eqs. (3.5) and (3.6). We define the upper bound in Eq. (3.38) as the target error, i.e., the theoretical maximum error that can occur when applying this method:

$$\epsilon_{target}^{(0)} := \frac{(1 + F(N_\tau, s))^{N_\tau} - 1}{F(N_\tau, s)} \left(\frac{T}{N_\tau}\right)^{p+1} K L_{f\tau}^p M \ . \quad (3.40)$$

Based on this result, we estimate the minimal number of time steps that are needed to guarantee a particular target error $\epsilon_{target}^{(0)}$ while solving Eq. (3.1):

Theorem 3.3. *Let the assumptions of Thms. 3.1-3.2 hold. Then, the minimal number of RKM steps $N_\tau^{(0)}$ required to solve Eq. (3.1) with a target error $\epsilon_{target}^{(0)}$ is*

$$N_\tau^{(0)} = L_{f\tau} T \left(\frac{KM \left(e^{b_{max} T L_{fy} s} - 1 \right)}{\epsilon_{target}^{(0)} b_{max} s L_{fy}} \right)^{1/p} . \tag{3.41}$$

Using the latter results, we get:

Corollary 1. *For a target error ϵ_{target}, the minimal value of the cost function (3.36) is*

$$C(N_\tau^{(0)}, s, p) = s L_{f\tau} T \left(\frac{KM \left(e^{b_{max} T L_{fy} s} - 1 \right)}{\epsilon_{target}^{(0)} b_{max} s L_{fy}} \right)^{1/p} , \tag{3.42}$$

where s is the number of the RKM stages, p is the RKMs order and $N_\tau^{(0)}$ is the minimal number of time steps of the chosen RKM.

The proofs of Thms. 3.2 and 3.3 are provided in Appendices 3.A and 3.B, respectively. In the following section, we analyze the RKM in the presence of the shot noise in the evaluations of the differential $f(\boldsymbol{\theta}(\tau))$.

3.4 Runge-Kutta methods under the presence of shot noise

In the variational algorithm that we presented in Sec. 3.2.2, we are solving the ODE given in Eq. (3.29) by using RKMs. The function $f(\boldsymbol{\theta}(\tau))$ from Eq. (3.29) is given by expectation values estimated via sampling quantum circuits. Therefore, we assume that instead of $f(\boldsymbol{\theta}(\tau))$, we have access to its approximation $\hat{f}(\boldsymbol{\theta}(\tau))$. We need to take into account the statistical errors arising while computing $\hat{f}(\boldsymbol{\theta}(\tau))$ based on the measurement results. The analysis of this section is valid for all differential equations as given in Eq. (3.1) that have a noisy $\hat{f}(\boldsymbol{\theta}(\tau))$.

By virtue of the central limit theorem, let us also assume that each measurement is drawn from a random Gaussian distribution with mean $f(\boldsymbol{\theta}(\tau))$ and standard deviation σ_{single}. Calculating the average of these measurements gives the estimate $\hat{f}(\boldsymbol{\theta}(\tau))$:

$$\hat{f}(\boldsymbol{\theta}(\tau)) = \frac{1}{N_r} \sum_{i=1}^{N_r} \hat{f}_i(\boldsymbol{\theta}(\tau)) , \tag{3.43}$$

where $\hat{f}_i(\boldsymbol{\theta}(\tau))$ is a single measurement result. By the central limit theorem as $N_r \to \infty$, the estimate $\hat{f}(\boldsymbol{\theta}(\tau))$ behaves as a Gaussian distribution:

$$\sqrt{N_r}(\hat{f}(\boldsymbol{\theta}(\tau)) - f(\boldsymbol{\theta}(\tau))) \to^d N(0, \sigma^2_{single}). \tag{3.44}$$

From Chebyshev's inequality we get the following bound for a $\delta > 0$:

$$P(\|\hat{f}(\boldsymbol{\theta}(\tau)) - f(\boldsymbol{\theta}(\tau))\| \geq \delta) \leq \frac{\sigma^2_{single}}{\delta^2 N_r}. \tag{3.45}$$

Alternatively, a Chernoff bound would be tighter, but more cumbersome to apply here and probably not lead to a different qualitative analysis.

If we take the probability $P(\|\hat{f}(\boldsymbol{\theta}(\tau)) - f(\boldsymbol{\theta}(\tau))\| \geq \delta)$ to be equal to $\eta \in (0, 1)$, then with probability of $1 - \eta$, the following bound holds:

$$\|\hat{f}(\boldsymbol{\theta}(\tau)) - f(\boldsymbol{\theta}(\tau))\| \leq \delta = \frac{\Sigma}{\sqrt{N_r}} \tag{3.46}$$

In the above, we defined $\Sigma = \frac{\sigma_{single}}{\sqrt{\eta}}$.

Assuming access to a noisy estimate of $f(\boldsymbol{\theta}(\tau))$, as described above, we derive error and resource estimates for solving ODEs using RKMs under the presence of shot noise.

In Thm. 3.4, we establish an upper bound on the error $\epsilon^{(\delta)}_{ODE}$ as defined in Eq. (3.31) by analyzing how both the truncation error of the RKM (see Thm. 3.1) and the shot noise in the estimation of $f(\boldsymbol{\theta}(\tau))$ (see Eq. (3.46)) contribute to error propagation.

We select the upper bound of the error $\epsilon^{(\delta)}_{ODE}$ as a target error, meaning the maximal error we can expect. This target error is the used to determine the minimal number of RKM steps and the minimal number of measurements of each $f(\boldsymbol{\theta}(\tau))$ required to keep the resulting error within the target. The result is formalized in Thm. 3.5, and the corresponding minimal cost, as defined in Eq. (3.35), is derived in Corollary 2.

Theorem 3.4. *Let us assume that the assumptions of Thm. 3.1 hold for Eq. (3.29). Under the conditions in Eq. (3.10), Eq. (3.37), the approximation $\hat{\boldsymbol{\theta}}_{N_\tau}$ at time τ_{N_τ} calculated via a p-th order RKM with s stages in N_τ time steps has the error defined in Eq. (3.31) upper bounded by:*

$$\epsilon^{(\delta)}_{ODE} \leq \frac{(1 + F(N_\tau, s))^{N_\tau} - 1}{F(N_\tau, s)} \left(\frac{3\delta}{L_{fy}} F(N_\tau, s) + \left(\frac{T}{N_\tau} \right)^{p+1} K L^p_{f\tau} M \right), \tag{3.47}$$

where we used the notation introduced in Eq. (3.39).

Let us further denote the latter upper bound as

$$\epsilon_{target}^{(\delta)} := \frac{(1+F(N_\tau,s))^{N_\tau}-1}{F(N_\tau,s)}\left(\frac{3\delta}{L_{fy}}F(N_\tau,s)+\left(\frac{T}{N_\tau}\right)^{p+1}KL_{f\tau}^{p}M\right). \tag{3.48}$$

Given this error estimate, we obtain the following resource minimization and the directly following minimal cost:

Theorem 3.5. *Let us assume that $\hat{f}(\boldsymbol{\theta}(\tau))$ is an approximation for $f(\boldsymbol{\theta}(\tau))$ calculated from N_r measurements, with the error scaling $\delta = \frac{\Sigma}{\sqrt{N_r}}$, where $\Sigma > 0$ is a constant. Then, calculating the approximated solution $\hat{\boldsymbol{\theta}}_{N_\tau}$ to the ODE in Eq. (3.29) by a p-th order RKM with s stages and with a target error $\epsilon_{target}^{(\delta)}$ defined in Eq. (3.48), the number of time steps required must be at least*

$$N_\tau^{(\delta)} = TL_{f\tau}\left(\frac{KM\left(e^{b_{max}sL_{fy}T}-1\right)(2p+1)}{\epsilon_{target}^{(\delta)}b_{max}sL_{fy}}\right)^{1/p}, \tag{3.49}$$

and at least

$$N_r^{(\delta)} = \frac{9\Sigma^2}{L_{fy}^2}\left(\frac{\epsilon_{target}^{(\delta)}}{\left(1+F(N_\tau^{(\delta)},s)\right)^{N_\tau^{(\delta)}}-1}-\left(\frac{T}{N_\tau^{(\delta)}}\right)^{p+1}\frac{KL_{ft}^{p}M}{F(N_\tau^{(\delta)},s)}\right)^{-2} \tag{3.50}$$

many measurements for the estimation of $f(\boldsymbol{\theta}(\tau))$ for each set of inputs $\boldsymbol{\theta}(\tau)$.

The proofs of Thms. 3.4 and 3.5 are given in Appendices 3.A and 3.C, respectively.

Corollary 2. *For a target error $\epsilon_{target}^{(\delta)}$, the minimal value of the cost function (3.35) is*

$$C(N_\tau^{(\delta)},N_r^{(\delta)},s,p) = sN_\tau^{(\delta)}(s,p)N_r^{(\delta)}(s,p), \tag{3.51}$$

where s is the number of stages, p is the order, $N_\tau^{(\delta)}$ and $N_r^{(\delta)}$ are the minimal number of time steps of the RKM; and the minimal number samples in calculating $\hat{f}(\boldsymbol{\theta}(\tau))$ for each set of inputs according to Thm. 3.5.

In the following section, we further analyze the latter error and resource estimates. By applying it to the variational quantum algorithm discussed in

Sec. 3.2.2, we show by numerical estimations in Sec. 3.6, how they evaluate for specific DEs such as given in Eq. (3.29). We shall see that with these results, we can choose a Runge-Kutta method such that the required total number of circuit evaluations will be minimal.

3.5 Analysis of the variational quantum algorithm

In this section, we analyze the total error as defined in Eq. (3.33) of the algorithm provided in Sec. 3.2.2. We show its relation to Eq. (3.31) and derive the resource requirements of the algorithm for a given target error, based on the estimates in Sec. 3.4. Afterwards, we continue with an estimation of the shot noise and further quantities based on a toy model we introduce. These further estimates are necessary for the numerical analysis of the following Sec. 3.6.

3.5.1 Error and resource estimate

We are interested in estimating the total error in Eq. (3.33). Using the triangle inequality, we get

$$\epsilon_{total} \leq \|\psi(T) - \phi(\boldsymbol{\theta}(T))\|_1 + \|\phi(\boldsymbol{\theta}(T)) - \phi(\hat{\boldsymbol{\theta}}_{N_\tau})\|_1 = \epsilon_{PQC} + \epsilon_{par} \ . \tag{3.52}$$

Here, we define the distance between the trial functions with the different input parameters as follows:

$$\epsilon_{par} := \|\phi(\boldsymbol{\theta}(T)) - \phi(\hat{\boldsymbol{\theta}}_{N_\tau})\|_1. \tag{3.53}$$

The representation error coming from approximating the function $\psi(T)$ with the trial function $\phi(\boldsymbol{\theta}(T))$ is defined as

$$\epsilon_{PQC} := \|\psi(T) - \phi(\boldsymbol{\theta}(T))\|_1. \tag{3.54}$$

Furthermore, using the multi-dimensional mean-value theorem [1], the second term of Eq. (3.52) can be bounded by

$$\epsilon_{par} \leq \sup_{\boldsymbol{\theta_0} \in \Xi} \|\nabla_{\boldsymbol{\theta}} \phi(\boldsymbol{\theta_0}(T))\|_{(1,2)} \epsilon_{ODE}^{(\delta)}, \tag{3.56}$$

[1] Multivariate Mean Value Theorem: for $x, y \in R^n$

$$\|f(x) - f(y)\|_q \leq \sup_{z \in [x,y]} \|f'(z)\|_{(q,p)} \|x - y\|_p, \tag{3.55}$$

Where $z \in [x, y]$ denotes a vector z contained in the set of points between $x, y \in R^n$ and $\|f'(z)\|_{(q,p)}$ is the $L_{(p,q)}$ norm of the derivative matrix of f evaluated at z.

where $\Xi = \{w\hat{\boldsymbol{\theta}}_{N_\tau} + (1-w)\boldsymbol{\theta}(T) | w \in [0,1]\}$. The norm of the Jacobian is defined as

$$\sup_{\boldsymbol{\theta_0}\in\Xi} \|\nabla_{\boldsymbol{\theta}}\phi(\boldsymbol{\theta_0}(T))\|_{(1,2)} := \sup_{\boldsymbol{\theta}_0\in\Xi} \sup_{\boldsymbol{\theta}^*(T)\in\Xi, \boldsymbol{\theta}^*(T)\neq 0} \frac{\|\nabla_{\boldsymbol{\theta}}\phi(\boldsymbol{\theta_0}(T))\cdot\boldsymbol{\theta}^*(T)\|_1}{\|\boldsymbol{\theta}^*(T)\|_2} . \tag{3.57}$$

The following lemma shows us an upper bound to this expression.

Lemma 2. *The norm of the Jacobian with the chosen circuit (Eq. (3.19) and Eq. (3.20)) is bounded by*

$$\sup_{\boldsymbol{\theta}_0\in\Xi} \sup_{\boldsymbol{\theta}^*(T)\in\Xi, \boldsymbol{\theta}^*(T)\neq 0} \frac{\|\nabla_{\boldsymbol{\theta}}\phi(\boldsymbol{\theta_0}(T))\cdot\boldsymbol{\theta}^*(T)\|_1}{\|\boldsymbol{\theta}^*(T)\|_2} \tag{3.58}$$

$$\leq \sup_{\boldsymbol{\theta}^*(T)\in\Xi, \boldsymbol{\theta}^*(T)\neq 0} \frac{\sum_k \left(\sum_j 2|f_{k,j}|\right)|\theta_k^*(T)|}{\|\boldsymbol{\theta}^*(T)\|_2} . \tag{3.59}$$

We provide the proof in Appendix 3.F.

According to this lemma, Eq. (3.57) is upper bounded by

$$S := \left(\sup_{\boldsymbol{\theta}^*(T)\in\Xi, \boldsymbol{\theta}^*(T)\neq 0} \frac{\sum_{k=1}^{N_V}\left(\sum_{j=1}^{N_d} 2|f_{k,j}|\right)|\theta_k^*(T)|}{\|\boldsymbol{\theta}^*(T)\|_1} \right) . \tag{3.60}$$

Thus, the total error is bounded by

$$\epsilon_{total} \leq \epsilon_{PQC} + S\epsilon_{ODE}^{(\delta)}. \tag{3.61}$$

As described in Sec. 3.2.3, there are errors which we assume to be negligible. In particular, we are disregarding the representation error ϵ_{PQC}. We are thus left with estimating S for the specific Ansatz at hand (see Sec. 3.6) and the error in calculating the parameters, $\epsilon_{ODE}^{(\delta)}$, which we analyzed in Sec. 3.4.

The resource that we would like to minimize for the application of the variational algorithm is the total number of circuit evaluations N_{circ} which is needed for running the algorithm. In comparison to the cost that was estimated in Sec. 3.4, we have to add an additional factor in order to get N_{circ}. The derivation of this factor follows.

Denote with N_A and N_C the numbers of circuits that are needed to calculate the matrix A and the vector C for specific input parameters. Each matrix A or vector C consists of N_V^2 or N_V elements, respectively where N_V is the number

of parameters $\boldsymbol{\theta}$. Each element of A and C is calculated by the evaluation of N_d^2 or $N_d N$ different circuits, respectively. Here N is the number of terms in the Hamiltonian and N_d is an upper bound on the number of Pauli strings in the unitary operator R_k of the variational circuits, as defined in Eq. (3.20). Thus, they are bounded above by $N_A \leq \overline{N_A} = N_V^2 N_d^2$ and $N_C \leq \overline{N_C} = N_V N_d N$. Let us assume that these upper bounds are always reached. We denote by $N_\tau^{(\delta)}$ the minimum number of time steps in the ODE solving algorithm and by $N_r^{(\delta)}$ the minimum number of times, each stage value $f(\boldsymbol{\theta}(\tau))$ is measured; both have been estimated in Thm. 3.5. By s, we denote the number of stages that are calculated at each time step of the RKM. Therefore, the number of circuit evaluation N_{circ} needed to calculate $\vec{\theta}(T)$ is equal to

$$N_{circ} = N_\tau^{(\delta)} s N_r^{(\delta)} (\overline{N_A} + \overline{N_C}) \tag{3.62}$$

$$= N_\tau^{(\delta)} s N_r^{(\delta)} (N_V^2 N_d^2 + N_V N_d N) \tag{3.63}$$

$$= N_\tau^{(\delta)} s N_r^{(\delta)} N_V N_d (N_V N_d + N) \, . \tag{3.64}$$

One can see that the number of circuit evaluations in the whole circuit scales quadratically in the number of circuit parameters N_V.

Note that, if instead of counting the total number of circuit evaluations we care about the total run time of the algorithm, and allow parallel evaluations of $N_V^2 N_d^2 + N_V N_d N$ circuits, the cost comes down to $N_\tau^{(\delta)} s N_r^{(\delta)}$.

In the next subsection, we will further estimate the dependence of $N_r^{(\delta)}$ on the condition number of A, and give a comprehensive overview of the resource estimate in Eqs. (3.77)- (3.80).

3.5.2 Estimation of the shot noise

In Sec. 3.4, we presented error and resource estimates for RKMs with an error in the evaluations of $f(\tau, \boldsymbol{\theta}(\boldsymbol{\tau}))$ stemming from shot noise. We derive our estimates on the basis of the bound in Eq. (3.46). Since we would like to incorporate the results of Sec. 3.4 into analyzes of the required resources for the variational quantum algorithm, we are studying the shot noise inherent to this algorithm in this section. In the end of this section, we will give an estimate of Σ as defined in Eq. (3.46).

Each term $\left(f_{k,i}^* f_{l,j} \langle\overline{0}| R_{k,i}^\dagger R_{l,j} |\overline{0}\rangle + h.c.\right)$ or $\left(f_{k,i}^* \lambda_m \langle\overline{0}| R_{k,i}^\dagger \sigma_m R |\overline{0}\rangle + h.c.\right)$ in the matrix A and the vector C as given in Eqs. (3.27) and (3.28) can be written in the following form:

$$q = a \mathrm{Re}\left(e^{i\zeta} \langle 0|^{\otimes n} U |0\rangle^{\otimes n}\right), \tag{3.65}$$

where the amplitude a and the phase ζ are determined by either $f_{k,i}^* f_{l,j}$ or

$f^*_{k,i}\lambda_m$. The unitary operator U is equal to either $R^\dagger_{k,i}R_{l,j}$ or $R^\dagger_{k,i}\sigma_m R$.

The terms q can be obtained by the parameterized quantum circuits shown in Fig. 3.1. Note that these circuits are different from the circuit which prepares the final state $|\phi(\hat{\boldsymbol{\theta}}_{N_\tau})\rangle$. An ancillary qubit of the circuit is initialized as $(|0\rangle + e^{i\zeta}|1\rangle)/\sqrt{2}$, while the remaining qubits are initialized in the state $|0\rangle^{\otimes n}$. A projective measurement of the ancillary qubit in the $\{|+\rangle, |-\rangle\}$ basis has following probability to find the qubit in the state $|+\rangle$:

$$\hat{P}_+ = \frac{\mathrm{Re}\big(e^{i\zeta}\langle 0|^{\otimes n} U |0\rangle^{\otimes n}\big) + 1}{2}. \tag{3.66}$$

To estimate the probability $\hat{P}_+$ and subsequently the terms q, we need to implement and measure the circuit N_r times. Let us assume that the measurements correspond to independent Bernoulli distributed random variables, which have the variance $\sigma^2(\hat{P}_+) = \hat{P}_+(1-\hat{P}_+)/N_r$.

Thus, the variances of the estimates of the elements of Eq. (3.27) are scaling as

$$\sigma^2\left(\mathrm{Re}\left(e^{i\zeta}\langle 0|^{\otimes n} U |0\rangle^{\otimes n}\right)\right) = 4\sigma^2(\hat{P}_+) = \mathcal{O}(\frac{1}{N_r}). \tag{3.67}$$

After estimating $\mathrm{Re}\big(e^{i\zeta}\langle 0|^{\otimes n} U |0\rangle^{\otimes n}\big)$, the amplitudes of $f^*_{k,i}f_{l,j}$ or $f^*_{k,i}\lambda_m$ have to be calculated classically and multiplied. Since they are not depending on each other, all of the circuits with the same input parameters $\boldsymbol{\theta}(\tau)$ can be run in parallel.

We denote the realization of the matrices A and C in Eq. (3.23) as $\hat{A}$ and $\hat{C}$, respectively.

The variances of the estimates $\hat{A}_{k,l}(\boldsymbol{\theta}(\tau))$ and $\hat{C}_k(\boldsymbol{\theta}(\tau))$ are scaling as

$$\sigma^2\left(\hat{A}_{k,l}(\tau)\right)\mathcal{O}\left(\frac{N_d^2}{N_r}\right), \quad \sigma^2\left(\hat{C}_k(\tau)\right)\mathcal{O}\left(\frac{N_d N}{N_r}\right). \tag{3.68}$$

Then by Chebyshev's inequality, we can conclude that the probability of estimation error is bounded by

$$P\left(|A_{k,l} - \hat{A}_{k,l}| \geq b\right) \leq \frac{N_d^2}{N_r b^2}, \quad P\left(|C_k - \hat{C}_k| \geq b\right) \leq \frac{N_d N}{N_r b^2}. \tag{3.69}$$

We can conclude that we face up with estimation errors arise due to shot noise and the representation errors from calculating the matrix elements on the quantum circuits. The following lemma holds:

Lemma 3. *With the probability of at least* $1-\eta$, $\eta \in [0,1]$, *the following bounds*

hold:

$$\|A - \hat{A}\| \leq \frac{\|\{\sigma_{k,l}\}_{k,l=1}^{N_V}\|}{\sqrt{N_r \eta}}, \tag{3.70}$$
$$\|C - \hat{C}\| \leq \frac{\|\{\sigma_{k}\}_{k=1}^{N_V}\|}{\sqrt{N_r \eta}},$$

where we used the notations

$$\sigma_{k,l} = \sqrt{\sum_{i,j=1}^{N_d} |f_{k,i}^* f_{l,j}|^2}, \quad \sigma_k = \sqrt{\sum_{i=1}^{N_d} \sum_{m=1}^{N} |f_{i,k}^* \lambda_m|^2}. \tag{3.71}$$

For a detailed proof, see Appendix 3.E.

This lemma provides us with upper bounds on the errors on A and C that arise from shot noise. These errors translate into an error in f as defined in Eq. (3.29), which can be estimated with the following lemma:

Lemma 4. *Consider the linear equation*

$$Af = C \ , \tag{3.72}$$

where A is a a non-singular $N_V \times N_V$-dimensional matrix and C an N_V-dimensional vector. Let us introduce disturbances in A and C by $A \mapsto A + \xi R$ and $C \mapsto C + \xi r$, leading to the disturbed linear equation

$$(A + \xi R)\hat{f} = C + \xi r \ , \tag{3.73}$$

where $\hat{f}$ is the solution vector under the disturbance, $\xi \geq 0$ is a real scalar, R is an $N_V \times N_V$ dimensional matrix and r is an N_V dimensional vector. Then, we can write the following estimate:

$$\frac{\|\hat{f} - f\|}{\|f\|} \leq \xi \kappa(A) \left(\frac{\|r\|}{\|C\|} + \frac{\|R\|}{\|A\|} \right) + \mathcal{O}(\xi^2) \ , \tag{3.74}$$

where $\kappa(A) := \|A\|\|A^{-1}\|$ is the condition number of A.

We give a proof of this lemma in Appendix 3.F.

If A is invertible, we can identify $f = A^{-1}C$, and can apply this theorem to get:

$$\|f - \hat{f}\| \leq \|A^{-1}C\| \xi \kappa(A) \left(\frac{\|r\|}{\|C\|} + \frac{\|R\|}{\|A\|} \right) + \mathcal{O}(\xi^2) \ . \tag{3.75}$$

In order to apply the theorem to our analysis above, we are defining $R = \{\sigma_{k,l}\}_{k,l=1}^{N_V}$, $r = \{\sigma_k\}_{k=1}^{N_V}$ and

$$\xi = \frac{1}{\sqrt{N_r \eta}} \,. \tag{3.76}$$

If in a specific instance, A is singular, it is not possible to invert it and thus apply Lemma 4. However, one can apply a regularization procedure by deviating the matrix elements of A slightly in such a way that it becomes non-singular. This will be leading to an additional error which, as we described in Sec. 3.2.3, is one of the error sources we are not taking into consideration in our analysis. For a way of dealing with this issue, see techniques used in Ref. [26, 116, 128, 152].

Taken together, we arrive at the following number of total circuit evaluations, combining Eqs. (3.39), (3.46), (3.62), (3.75), and the results of Thm. 3.5:

$$N_{circ} = N_\tau^{(\delta)} s N_r^{(\delta)} N_V N_d (N_V N_d + N) \tag{3.77}$$

$$N_r^{(\delta)} = \frac{9\Sigma^2}{L_{fy}^2} \left(\frac{\epsilon_{target}^{(\delta)}}{\left(1 + F(N_\tau^{(\delta)}, s)\right)^{N_\tau^{(\delta)}} - 1} - \left(\frac{T}{N_\tau^{(\delta)}}\right)^{p+1} \frac{K L_{ft}^p M}{F(N_\tau^{(\delta)}, s)} \right)^{-2} \tag{3.78}$$

$$\Sigma = \|A^{-1}C\| \frac{\kappa(A)}{\sqrt{\eta}} \left(\frac{\|r\|}{\|C\|} + \frac{\|R\|}{\|A\|} \right) + \mathcal{O}(\xi^2) \tag{3.79}$$

$$F(N_\tau, s) = \frac{b_{max}}{a_{max}} \left(\left(1 + L_{fy} a_{max} \frac{T}{N_\tau}\right)^s - 1 \right). \tag{3.80}$$

To provide further details and to make the analysis concrete, in Sec. 3.5.3, we select a specific toy model as a representative example. We estimate the quantities $\|A\|$, $\|C\|$, $\|A^{-1}C\|$, $\kappa(A)$ and L_{fy} based on this toy model and in Sec. 3.6, we will estimate $\|R\|$, $\|r\|$, a_{max}, b_{max}, K, M and $L_{f\tau}$ for general Runge-Kutta methods and a chosen Ansatz.

3.5.3 Estimations based on a toy model

In order to apply the error and resource estimates from Sec. 3.4 in practice (such as we do in Sec.3.6), we need to make estimates on several quantities. Especially challenging is the estimation of the condition number $\kappa(A)$, $\|A\|$, $\|C\|$ and $\|A^{-1}C\|$ that are needed for the estimates on shot noise as derived in Sec. 3.5.2 and the estimation of the Lipschitz constant L_{fy}, as defined in Eq. (3.37).

As those quantities depend on A and C, we are taking the following considerations: Both A and C depend on expectation values from quantum circuits

as written in Eq. (3.65). The parameters $\boldsymbol{\theta}(\tau_k)$ at specific time steps τ_k vary a lot depending on the instance being solved. Since the expectation values that constitute the coefficients of A and C are depending on these parameters as well as the chosen Ansatz, it is non-viable to estimate them in general.

However, motivated by the fact that expectation values of parameterized quantum circuits defined as in our chosen Ansatz (see Eq. (3.18)) resemble truncated Fourier series (see Refs. [67, 154]), we simulate plausible coefficients of A and C according to the following toy model:

Let us assume that each of the elements $A_{k,l}$ and C_k take the form of functions $w_{k,l}(\tilde{\theta})$ and $w_k(\tilde{\theta})$ in one input parameter $\tilde{\theta}$, respectively:

$$w_{k,l}(\tilde{\theta}) := \alpha_{1,k,l} \cos\big(\alpha_{2,k,l}\tilde{\theta} + \alpha_{3,k,l}\big) + \alpha_{4,k,l} \sin\big(\alpha_{5,k,l}\tilde{\theta}\big), \tag{3.81}$$

$$w_k(\tilde{\theta}) := \alpha_{1,k} \cos\big(\alpha_{2,k}\tilde{\theta} + \alpha_{3,k}\big) + \alpha_{4,k} \sin\big(\alpha_{5,k}\tilde{\theta}\big) \ . \tag{3.82}$$

We are choosing the amplitudes $\alpha_{1,k,l}$, $\alpha_{1,k}$ and $\alpha_{4,k,l}$, $\alpha_{4,k}$ to be randomly drawn from the Gaussian distribution $\mathcal{N}(1, 0.1)$, and the phases $\alpha_{2,k,l}$, $\alpha_{2,k}$ and $\alpha_{5,k,l}$, $\alpha_{5,k}$ and phase differences $\alpha_{3,k,l}$, $\alpha_{3,k}$ to be randomly drawn from the Gaussian distribution $\mathcal{N}(0, 0.1)$.

Condition number and norms on A and C

In the literature, it is often assumed that the condition number of a matrix scales polynomially in its dimension [155]. If this assumption holds for the $N_V \times N_V$-dimensional matrix A, there exists a constant Γ, such that with high probability, the following bound holds:

$$\kappa(A) \leq N_V^{\Gamma} \ . \tag{3.83}$$

In order to find out if this bound also holds if the coefficients of A are sampled according to the toy model as described above, we plot such $\kappa(A)$ and the functions N_V^2, N_V^3 and N_V^4 for varying N_V in Fig. 3.2.

Most of the condition numbers lie above the threshold N_V^2, while for $N_V \geq 10$, most condition numbers lie below the threshold N_V^3. Typically, more than 10 parameters are chosen in applications of the variational algorithm (We are using $N_V = 25$ in our numerical analysis in Sec. 3.6). In rare cases, A constructed by the toy model is ill-conditioned, and $\kappa(A)$ significantly exceeds those bounds. As described earlier, those cases can be mitigated by a regularization of A with the drawback of introducing an additional error.

We therefore assume that the bound in Eq. (3.83) holds with high probability for our toy model, with the constant $\Gamma = 3$.

Analogously, we estimate the norms $\|A\|$, $\|C\|$ and $\|A^{-1}C\|$ based on A and C that follow the introduced toy model. In Fig. 3.3, we plot those norms and the functions $f_1(N_V) = N_V$ and $f_2(N_V) = \sqrt{N_V}$ for varying N_V.

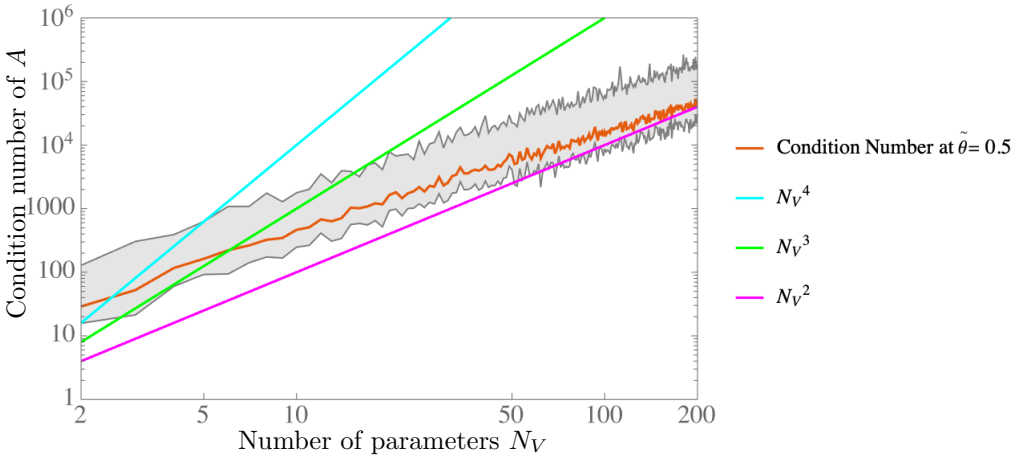

Figure 3.2: A log plot comparing the condition number obtained from the toy model with $\tilde{\theta} = 1/2$ and different upper bounds due to N_V^2, N_V^3 and N_V^4. We sampled 100 different matrices A according to the toy model and calculated the resulting condition number for each sample. The orange line shows the median and the gray shaded area shows the range between the 0.16 and 0.84 quantiles of the condition number estimates. We used the Frobenius norm.

We see that it is reasonable to assume the scaling $\|A\| = \Theta(N_V)$ and $\|C\| = \Theta(\sqrt{N_V})$. For $\|A^{-1}C\|$, we cannot make similarly reasonable estimates due to the high fluctuation. However, we can assume an upper bound $\|A^{-1}C\| \leq 60$, which holds with high probability in the number of parameter range $N_V \in [0, 100]$. Similar to managing the condition number, it is also possible to enforce this bound by a regularization of the matrix A.

The above analysis on estimates and bounds for $\kappa(A)$, $\|A\|$, $\|C\|$ and $\|A^{-1}C\|$ included into the estimate in Eq. (3.46) leads to the following inequality to hold with high probability:

$$\|f - \hat{f}\| < \delta \leq \frac{60}{\sqrt{N_r \eta}} N_V^3 \left(\frac{\|\{\sigma_k\}_{k=1}^{N_V}\|}{\sqrt{N_V}} + \frac{\|\{\sigma_{k,l}\}_{k,l=1}^{N_V}\|}{N_V} \right) . \tag{3.84}$$

Hence, we conclude the following upper bound for Σ (see Eq. (3.46)) with high probability:

$$\Sigma \leq \frac{60}{\sqrt{\eta}} N_V^3 \left(\frac{\|\{\sigma_k\}_{k=1}^{N_V}\|}{\sqrt{N_V}} + \frac{\|\{\sigma_{k,l}\}_{k,l=1}^{N_V}\|}{N_V} \right) . \tag{3.85}$$

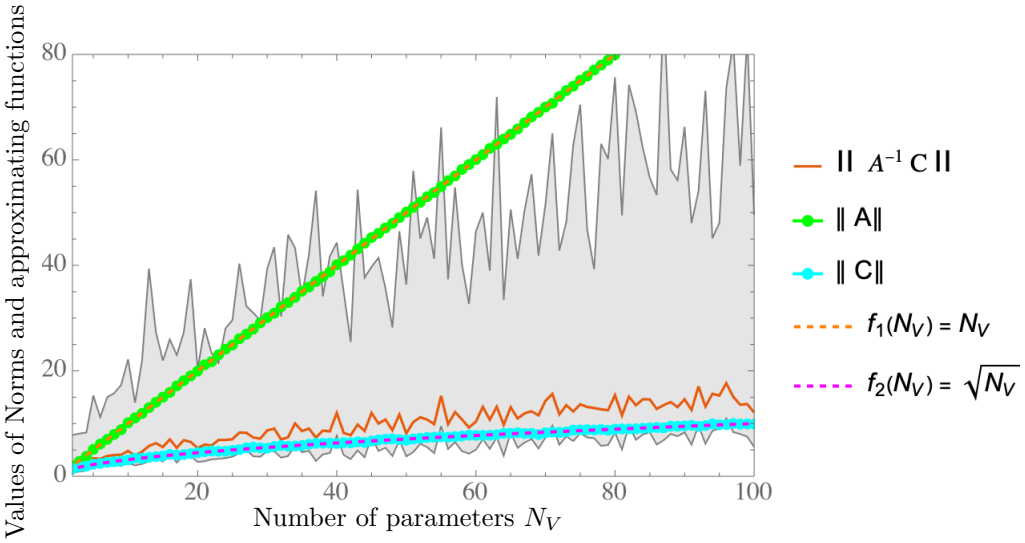

Figure 3.3: A plot comparing the norms $\|A\|$, $\|C\|$ and $\|A^{-1}C\|$ obtained from the toy model with the functions $f_1(N_V) = N_V$ and $f_2(N_V) = \sqrt{N_V}$. In order to estimate $\|A^{-1}C\|$, we did 100 samples for A and C and calculated the resulting norm for each sample. The orange line shows the median and the gray shaded area shows the range between the 0.16 and 0.84 quantiles of the estimates for $\|A^{-1}C\|$. For $\|A\|$, we used the Frobenius norm and for $\|C\|$ and $\|A^{-1}C\|$ the 2 norm.

Estimation of the Lipschitz constant

We have seen in Sec. 3.4, that the Lipschitz constant L_{fy} of $f(\boldsymbol{\theta}(\tau))$ with respect to the $\boldsymbol{\theta}(\tau)$ has a direct influence on the error and resource estimates. In fact, as we will see in Sec. 3.6, the sensitivity of the total cost is higher with L_{fy} than with most of the other parameters. Let us therefore analyze the Lipschitz constant for the variational algorithm described in Sec. 3.2.2. In this case, $f(\boldsymbol{\theta}(\tau))$ is defined as

$$f(\boldsymbol{\theta}(\tau)) = f\left(\boldsymbol{\theta}(\tau)\right) := A^{-1}\left(\boldsymbol{\theta}(\tau)\right) C\left(\boldsymbol{\theta}(\tau)\right) \ , \tag{3.86}$$

and the Lipschitz constant L_{fy} turns into the upper bound

$$L_{fy} \geq \frac{\| f\left(\boldsymbol{\theta}_1(\tau)\right) - f\left(\boldsymbol{\theta}_2(\tau)\right) \|}{\| \boldsymbol{\theta}_1(\tau) - \boldsymbol{\theta}_2(\tau) \|} \ , \quad \forall \boldsymbol{\theta}_1(\tau), \boldsymbol{\theta}_2(\tau) \ . \tag{3.87}$$

We are again modeling the matrix A and the vector C with the toy model described in Eqs. (3.81) and (3.82). This toy model has a single input variable $\tilde{\theta}$ instead of an N_V-dimensional vector $\boldsymbol{\theta}(\tau)$. Let us vary this parameter, while we keep the other (randomly drawn) parameters of the toy model fixed, and

define the following function:

$$\mathrm{Lip}\left(\tilde{\theta}_1, \tilde{\theta}_2\right) := \frac{\left\| f\left(\tilde{\theta}_1\right) - f\left(\tilde{\theta}_2\right) \right\|}{\left\| \tilde{\theta}_1 - \tilde{\theta}_2 \right\|} \,. \tag{3.88}$$

We show two 3D plots of this function, with varying $\tilde{\theta}_1$ and $\tilde{\theta}_2$ and fixed dimensions of A and C with $N_V = 25$ in Fig. 3.4. For each of the two plots, the random toy model parameters are drawn independently.

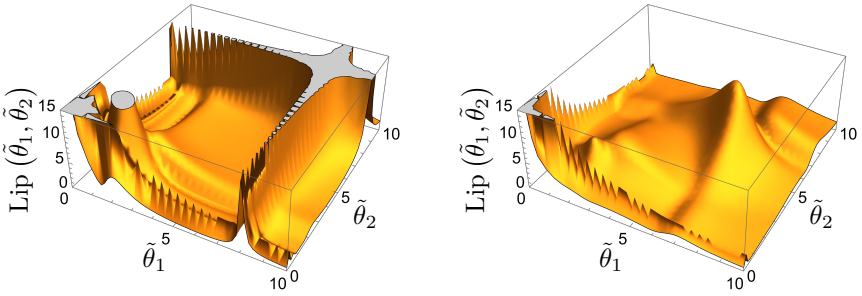

Figure 3.4: The function Lip $\left(\tilde{\theta}_1, \tilde{\theta}_2\right)$ defined in Eq. (3.88) plotted for two different drawings of random matrix and vector parameters according to the toy model from Eqs. (3.81) and (3.82). We fixed $N_V = 25$ and varied the parameters $\tilde{\theta}_1$ and $\tilde{\theta}_2$ between 0 and 10.

At the gray areas, the function Lip $\left(\tilde{\theta}_1, \tilde{\theta}_2\right)$ exceeds the value of 15. We see that in order to be able to bound Lip $\left(\tilde{\theta}_1, \tilde{\theta}_2\right)$ from above by a Lipschitz constant L_{fy}, one has to guarantee that at each update step from $\tilde{\theta}_1$ to $\tilde{\theta}_2$, the function Lip $\left(\tilde{\theta}_1, \tilde{\theta}_2\right)$ stays inside the region below the threshold $L_{fy} = 15$. It is possible to take higher estimates for L_{fy} and create possibly larger areas for valid $\tilde{\theta}_1$ and $\tilde{\theta}_2$. However, this comes to the account of less tight error and resource bounds in Thms. 3.4 and 3.5.

Interestingly enough, the function Lip $\left(\tilde{\theta}_1, \tilde{\theta}_2\right)$ keeps similar patterns for changing the mean and the variance of the distributions from which the amplitudes of the toy model are drawn. However, it shows an increasingly more rugged landscapes for both increasing the mean and the variance of the random distributions of the phases of the toy model. As expected, the function Lip $\left(\tilde{\theta}_1, \tilde{\theta}_2\right)$ behaves periodically with respect to the phase difference parameter of the toy model, with the period depending on the phases.

Under the assumption that the toy model we chose models the behavior of A and C sufficiently well, we use the estimates for $\kappa(A)$, $\|A\|$, $\|C\|$, $\|A^{-1}C\|$ and L_{fy} derived in this section in order to concretize the error and resource estimates from Sec. 3.4, and to numerically analyze them in the following section.

3.6 Numerical analysis of the error and resource estimates

In this section, we are numerically analyzing the error and resource estimates from Sec. 3.3 and Sec. 3.4 for an application of the RKMs to solving a simple ODE and for an application of the variational algorithm presented in Sec. 3.2.2 to solving a partial DE coming from finance.

3.6.1 Solving a classical ODE without shot noise

Let us analyze the cost function provided in Corollary 1 with a simple ODE:

$$\frac{\partial \boldsymbol{\theta}(\tau)}{\partial \tau} = \frac{\boldsymbol{\theta}(\tau)}{2}, \quad \boldsymbol{\theta}(0) = 1 \ . \tag{3.89}$$

This ODE has the exponential function $\boldsymbol{\theta}(\tau) = \exp\left(\frac{\tau}{2}\right)$ as a solution. We choose this simple ODE, because it is easy to analyze and to estimate the corresponding parameters that are used for the cost function.

Parameter estimation and sensitivity analysis for the classical ODE solver

The Lipschitz constant L_{fy} is equal to 0.5. If we pick a final time $T = 5$, then the function $f(\boldsymbol{\theta}(\tau)) = \frac{1}{2}\boldsymbol{\theta}(\tau)$ is upper bounded by $M = 13$. The derivative of $f(\boldsymbol{\theta}(\tau))$ with respect to τ is equal to $0.25\exp\left(\frac{1}{2}\tau\right)$ (since we know $\boldsymbol{\theta}(\tau) = \exp\left(\frac{1}{2}\tau\right)$). Thus we can choose a Lipschitz constant $L_{f\tau} = 3.1$, which upper bounds this derivative for times $\tau \in [0, T = 5]$.

It can be easily checked that the other bounds required in Thm. 3.1 hold as well.

The approximation of $b_{max} := \max_i |b_i|$ is depending on the chosen Runge-Kutta method. While there exist Runge-Kutta methods that have coefficients $|b_i| > 1$, those are the exception as one can see by looking at several methods (for example, in [140]).

To estimate the constant K, let us look at its definition [140, Chapter 318]:

$$K := \sum_{|t|=p+1} \frac{1}{\sigma(t)} \left| \Phi(t) - \frac{1}{t!} \right|, \tag{3.90}$$

where t is a rooted tree of order $|t|$, the quantities $\sigma(t)$ and $t!$ are defined as products of factorials, and lie between 1 and $|t|!$. The quantity $\Phi(t)$ depends on the coefficients of the chosen method. For a rigorous definition of these terms, see [140]. As it is discussed in [140, Chapter 318], it is not possible to create a general rule for estimating the terms in K. However, what can be said is that

the number of terms in the sum, the number of unlabeled rooted trees with $p+1$ nodes, is asymptotically equal to [156] $0.439 \cdot 2.956^{p+1} \cdot (p+1)^{-3/2}$. This series scales exponentially in p. Thus, if the summands $\frac{1}{\sigma(t)}\left|\Phi(t) - \frac{1}{t!}\right|$ were constant in p, K would scale exponentially in p as well. However, in [140] it is said that it is reasonable to assume that K is constant in $\Delta\tau$. In Ref. [140, Chapter 244], several Runge-Kutta (and related) methods were developed for which the corresponding error constants were estimated, all of which are upper-bounded in magnitude by 1. In the following analysis, we therefore assume that $K \leq 5$ holds with high probability. But, as we will see later, the total cost does not change as much in the parameter K as it does in other parameters, and we found that the implications of our work still hold qualitatively with a pessimistic upper bound of $K \leq 0.439 \cdot 2.956^{p+1} \cdot (p+1)^{-3/2}$. Let us further choose a target error of $\epsilon_{target} = 0.001$.

We collect the estimates for the analysis in this section in Table 3.2. Based

Parameter	Estimate
b_{max}	1
L_{fy}	0.5
T	5
K	5
$L_{f\tau}$	3.1
M	13
ϵ_{target}	0.001

Table 3.2: Estimates of various parameters that optimize the savings by using a higher-order Runge-Kutta method instead of the Euler method. Used for the analysis in Fig. 3.6.

on these default values, we are analyzing the sensitivity of the total cost on tuning the parameters in Fig. 3.5. We show how the total cost changes when a single default value is multiplied with a scaling factor, whereas the remaining parameters are kept at default. We see that the cost does go down for higher-order Runge-Kutta methods and that the parameter with the most effect is the total time T, the Lipschitz constant L_{fy}, the parameter b_{max} stemming from the Runge-Kutta method and the total target error ϵ_{target}. The spike in the cost at order $p = 6$ is due to the non-linear relationship between the order of the method and the number of stages (see Table 3.1).

Numerical analysis of the classical ODE solver

Using the parameters estimated above in Table 3.2, we estimate the total cost for Runge-Kutta methods of different orders. For the relation between the minimum number of stages s needed for a particular order p, we use Table 3.1.

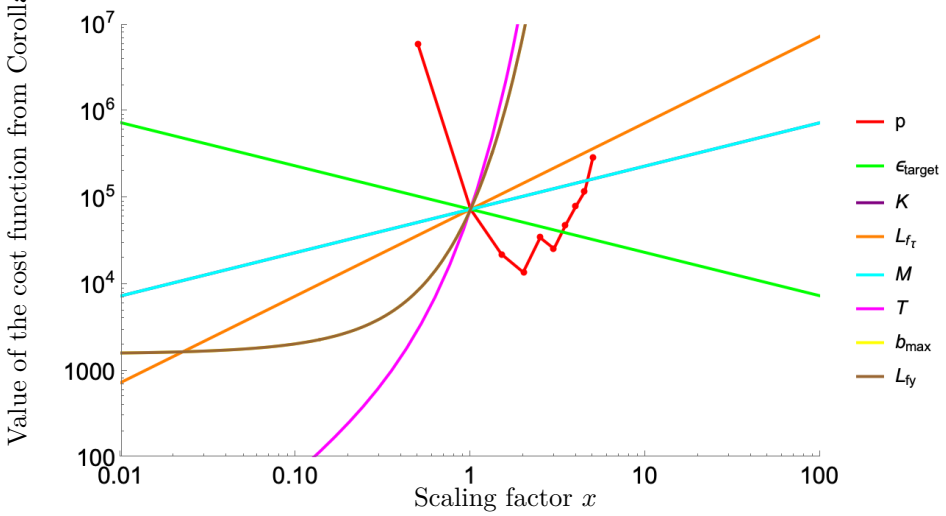

Figure 3.5: A plot comparing the sensitivity of the cost function with respect to different parameters. The intersecting point in the middle is the value of the cost function where all parameters are chosen default as given in Table 3.2 as well as $p = 2$ as a default. In each graph, we are changing one parameter, while keeping all the other parameters at default. We are changing the parameter by multiplying with the scaling factor x given as the abscissa. The graphs for M and K overlap as well as the graphs for L_{fy} and $b_{\max}$. The continuous red line for p is just for visualization purposes, as p is integer.

We show the results in Fig. 3.6. On the left hand side is a table with the cost depending on the order and the ratio of the Euler method compared to a Runge-Kutta method of order p. On the right hand side, we plotted this ratio depending on the Runge-Kutta order. We see that for our particular example, all methods of order $2 \leq p \leq 10$ are more cost efficient than the Euler method, where a Runge-Kutta method of order 4 is the most cost efficient with savings of a factor of ca. 10^3 compared to the cost of the Euler method. There is a second spike for the order $p = 6$, because of the non-linear relationship between the order of the method and the number of stages as shown in Table 3.1.

3.6.2 Solving a linear PDE with the variational quantum algorithm

We are numerically analyzing the estimates from Sec. 3.4 and Sec. 3.5 for an application of the variational algorithm presented in Sec. 3.2.2 to solving the Black Scholes model, which is a linear partial DE coming from finance. The motivation for choosing this model is twofold: On the one hand, we are using this example in order to specify the estimates done in previous sections and

p	cost(p)	cost(1)/cost(p)	$N_\tau^{(0)}$
1	2.25×10^7	1.00	2.25×10^7
2	9.60×10^4	2.35×10^2	4.80×10^4
3	1.99×10^4	1.13×10^3	6.63×10^3
4	1.01×10^4	2.22×10^3	2.54×10^3
5	1.38×10^4	1.64×10^3	2.29×10^3
6	1.03×10^4	2.18×10^3	1.47×10^3
7	1.36×10^4	1.65×10^3	1.52×10^3
8	1.71×10^4	1.32×10^3	1.56×10^3
9	2.07×10^4	1.09×10^3	1.60×10^3
10	3.33×10^4	6.76×10^2	2.08×10^3

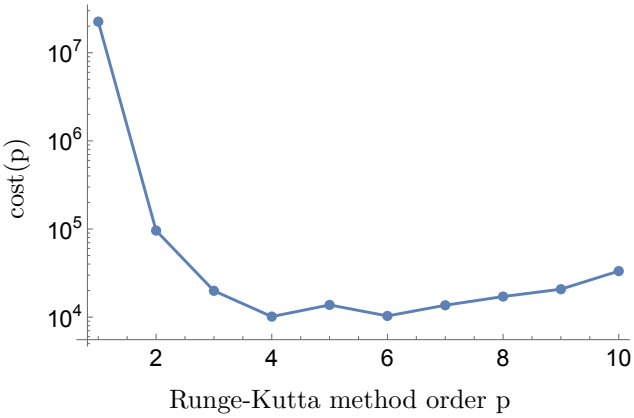

Figure 3.6: Comparison of the cost for different RKM orders, where the estimated parameters are fine tuned to maximize the savings by using a higher order as given in Table 3.2. In the first column, we have the Runge-Kutta method order p. In the second column is the total cost of an algorithm that uses a Runge-Kutta method of order p, in the third column the ratio of an algorithm that uses the Euler method ($p = 1$) with an algorithm that uses a Runge-Kutta method of order p, and in the fourth column is the minimal number of time steps $N_\tau^{(0)}$. In the plot, we show the cost, which is calculated according to Corollary 1, plotted against the order of the Runge-Kutta method.

in showing the consequences of the choice of RKMs on the total cost. On the other hand, we are using this example to stress the wide range of DEs on which the variational algorithm presented in Sec. 3.2.2 can be applied to.

Black Scholes model and error analysis

Let us begin with describing the problem of option pricing and the application of the variational algorithm to it. This application has been studied before in several ways [26, 127, 128, 132].

A European call option is used in practice in the following way: Initially, at time $t = 0$, an option is acquired, specifying a particular underlying asset, an expiration time t_{final}, and a strike price K. When time reaches $t = t_{final}$, the option buyer faces a decision: whether to exercise the option by buying the asset at the strike price K or to refrain from exercising it. The buyer's rational choice at time t_{final} is determined by the price $S(t_{final})$ of the underlying asset at that moment. If $S(t_{final}) > K$, indicating that the final asset price exceeds the strike price, the buyer can use the option to purchase the asset for the price K and immediately sell it at the higher market price of $S(t_{final})$ in an ideal market. This transaction results in a profit of $S(t_{final}) - K$ for the buyer. Conversely, if $S(t_{final}) \leq K$, the buyer would not choose to exercise the option by purchasing the asset since they would not be able to generate a profit from selling it in the market.

One can conclude that the payoff of the buyer is equal to

$$V(t_{final}, S) = \max\{S(t_{final}) - K, 0\} \ . \tag{3.91}$$

Since the stock price at time t_{final} is unknown, one models the stock price stochastically. A simple model is the so-called Black-Scholes model which characterizes the stock price $S(t)$ as a stochastic variable that follows a geometric Brownian motion:

$$dS(t) = \mu S(t)dt + \sigma S(t)dW_t \ , \tag{3.92}$$

where μ is the drift of the stock price, σ is its standard deviation (called 'volatility') and the random variable dW_t is a Wiener process. The arbitrage assumption states that it is impossible to build a portfolio which gives positive return without risk. By incorporating this assumption and using Itô calculus, the stochastic DE in $S(t)$ is transformed to a parabolic partial DE (PDE) that models the price of a call option $V(t, S)$:

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} = rV \ . \tag{3.93}$$

This PDE is called the Black Scholes equation. The parameters of this model are the volatility σ of the stock price and the risk-free interest rate r. Both are assumed to be independent of time. The Black Scholes can be mapped to the imaginary time Schrödinger equation with the following transformations:

Applying the transformations $\tau = (t_{final} - t)\sigma^2$, $x = \log(S)$, and subsequently

$u(\tau, x) = e^{-ax-b\tau}V(\tau, x)$ with the parameters $a = \frac{1}{2} - \frac{r}{\sigma^2}$ and $b = -\frac{1}{2}a^2 - \frac{r}{\sigma^2}$, where $u(\tau, x)$ is a modified price, one obtains [26]:

$$\frac{\partial}{\partial\tau}u(\tau, x) = \frac{1}{2}\frac{\partial^2}{\partial x^2}u(\tau, x) \ . \tag{3.94}$$

Note that $\tau \in [0, T]$, where we write $T = t_{final}\sigma^2$. This equation is equivalent to the imaginary time Schrödinger equation in Eq. (3.15), where $|\psi(\tau)\rangle$ is a quantum state representing the option price and $\mathcal{H} = -\frac{1}{2}\frac{\partial^2}{\partial x^2}$ is the Hamiltonian operator. Solving Eq. (3.15) is hence equivalent to solving the Black-Scholes equation in Eq. (3.93) after reversing the transformations.

We can interpret the state with the option price in the following way:

Consider a register of n qubits. On this chain, we define a set of 2^n pairwise orthogonal states $\{|x\rangle\}_x$. We select an interval of possible stock prices and discretize the interval into 2^n points. Consequently, we associate the basis states with stock prices, where the state $|0\rangle^{\otimes n}$ corresponds to the minimum stock price, and the state $|1\rangle^n$ corresponds to the maximum stock price of the chosen interval.

The quantum state from Eq. (3.16) encodes the option value corresponding to a particular stock price in the amplitudes of $|\psi(\tau)\rangle$ in the following way:

$$|\psi(\tau)\rangle = \sum_x \sqrt{p_x(\tau)}\,|x\rangle \ , \quad \sum_x p_x(\tau) = 1, \quad \forall\tau. \tag{3.95}$$

The boundary condition is $p_x(0) = \gamma(0)V(0, x)e^{-ax}$, where $\gamma(0)$ is the normalization constant and $V(0, x)$ is the option price at time $\tau = 0$. Thus, by obtaining the probability $p_x(\tau)$ at time $T = t_{final}\sigma^2$ for a particular stock price $x = \log(S)$, the corresponding option price can be calculated as:

$$V(t_{final}, x) = \gamma^{-1}(T)p_x(T)e^{ax+bt_{final}\sigma} \ . \tag{3.96}$$

In the following subsections, we are estimating the parameters based on the application of the variational algorithm to this option pricing use case and do a numerical analysis.

Parameter estimations and sensitivity analysis for the variational quantum algorithm

We are now making educated guesses for the parameters that are related to the use case of option pricing as described above and analyze the sensitivity of our error and resource analysis from Sec. 3.4 with respect to these parameters. In this and the following section, we use the symbol $\sim$ for approximations of a term with a scalar, including rounding errors.

In several papers [26, 127, 128, 132], the application of the variational al-

gorithm as described has been applied to option pricing. We are basing our parameter estimates on Ref. [26, Sec. 5.1.], where the authors take an Ansatz with the parameters $N = 16$, $N_V = 25$ and $N_d = 1$. That means that the matrix A is a 25×25 matrix, the vectors C and $\boldsymbol{\theta}$ have 25 elements, as well as the matrix $\sigma_{k,l}$ and the vector σ_k that we defined in Lemma 3. We take the probability η from the same lemma equal to $\eta \sim 0.05$.

A typical total time t_{final} in option pricing is one year and a typical volatility is $\sigma = 0.2 \cdot 1/\text{year}^2$. Thus, $t_{final} \cdot \sigma^2 = T \sim 0.04$.

In Ref. [26, Chapter 3], the parameters $f_{k,j}$ are defined as $f = i/2$. With this, let us approximate the perturbations $\|\sigma_{k,l}\|$ and $\|\sigma_k\|$:

It makes sense to see them in terms of the maximal deviation of individual entries. Given $f = i/2$, they are for the matrix equal to $N_d^2 \cdot 1/2 \cdot 2$ and for the vector $N_d N \cdot 1/2 \cdot 2$. Thus, we assume the total norms to scale as $\|\sigma_{k,l}\| \leq N_V N_d^2 \cdot 1$ and $\|\sigma_k\| \leq N_V N_d N \cdot 1$.

For approximations of the parameters a_{max}, b_{max} and K, see our discussion in Sec. 3.6.1. We pick now the parameters $a_{max} = b_{max} = 1$ and the $K = 5$. But as above, we see later that N_{circ} does not depend strongly on K and our results are still qualitatively valid for a large variety of K.

In line with the discussion in Sec. 3.5.1, we estimate the Lipschitz constant L_{fy} to be equal to $L_{fy} = 15$, the condition number to be upper bounded as $\kappa(A) \leq N_V^3 = 15625$ and $M = 60$ with the caveat of having to assure the toy model is chosen adequate enough and the bounds can be guaranteed by regularization to satisfy these bound.

We are also estimating a probability coming from the shot noise of $\eta = 0.05$. Combining the estimates, we are therefore getting with high probability the upper bound

$$\Sigma \leq \frac{60}{\sqrt{\eta}} N_V^3 \left(\frac{N_V N_d N}{\sqrt{N_V}} + \frac{N_V N_d^2}{N_V} \right) \sim 3.4 \times 10^8 \ . \tag{3.97}$$

The parameter $L_{f\tau}$ is defined as an upper bound to the time derivative of $f(\boldsymbol{\theta}(\tau))$. Since the time dependence of f is fully carried via the function $\boldsymbol{\theta}(\tau)$, we can apply a reasoning similar to that for estimating L_{fy}. We assume that the same behavior and the same bounds hold and, therefore, estimate $L_{f\tau} = L_{fy} = 15$.

We cannot determine if the other bounds hold that are required in Thm. 3.1, since we do not have full knowledge of the function $f(\boldsymbol{\theta}(\tau))$.

For estimating the quantity S as defined in Eq. (3.61), we need to take the following into account: We estimate that for all $1 \leq k \leq N_V$, $|\theta_k^*(T)| \sim \frac{\|\boldsymbol{\theta}\|_2}{N_V}$. Together with taking $f_{k,j} = i/2$ as above, we get $S \sim 1$.

Lastly, we are choosing a target error of $\epsilon_{target} \sim 0.001$. Let us thus make the following estimates, collected in Table 3.3:

Let us now show a plot where we examine the scaling of N_{circ} with respect

Parameter	Estimate
a_{max}	1
b_{max}	1
L_{fy}	15
T	0.04
K	5
$L_{f\tau}$	15
M	60
S	1
Σ	3.4×10^8
ϵ_{target}	0.001

Table 3.3: Estimates of various parameters based on the option pricing application. Used for the analysis in Figs. 3.7 and 3.8.

to the input parameters. It is difficult to make general statements about the scaling of N_{circ} from Eq. (3.62), since it does not have a trivial form, thus a look at a plot helps with understanding its scaling. We take as default the parameters estimated above and analyze in Fig. 3.7 the deviations caused by changing a single parameter away from the default.

We conclude from this figure, that N_{circ} does not change much for the parameters a_{max}, M, $L_{f\tau}$, K and Σ. As it was proved impossible to estimate K in a general way, it is promising to see that our estimate of N_{circ} does not scale with it as much as with other parameters.

In addition, N_{circ} changes faster for the parameters p, T, b_{max}, ϵ_{target}, and L_{fy}. For p in particular, we see that a minimum of N_{circ} is reached for a value higher than for $p = 1$. We will examine the behavior with p with more detail in the next section.

Numerical analysis of the variational quantum algorithm

For the numerical analysis, we use the parameters given in Table 3.3 that are fit to the option pricing use case in order to calculate N_{circ}, which is the total number of circuit evaluations needed for the algorithm. As we mentioned in Sec. 3.6.2, the parameter K is the most challenging to estimate. For this analysis, we choose $K = 5$. A comparison of the costs of algorithms based on Runge-Kutta and Euler methods is given in Fig. 3.8. We see that the highest saving in N_{circ} compared to the Euler method can be done with a Runge-Kutta method of order $p = 2$.

To reach a target error of $\epsilon_{target} = 0.001$, we need an $N_r^{(\delta)} \sim 10^{22}$, which is equivalent to requiring classical machine precision for the entries of matrices A and vectors C ($\epsilon \sim 1/\sqrt{N_r^{(\delta)}} \sim 5 \times 10^{-12}$). The total number of circuit

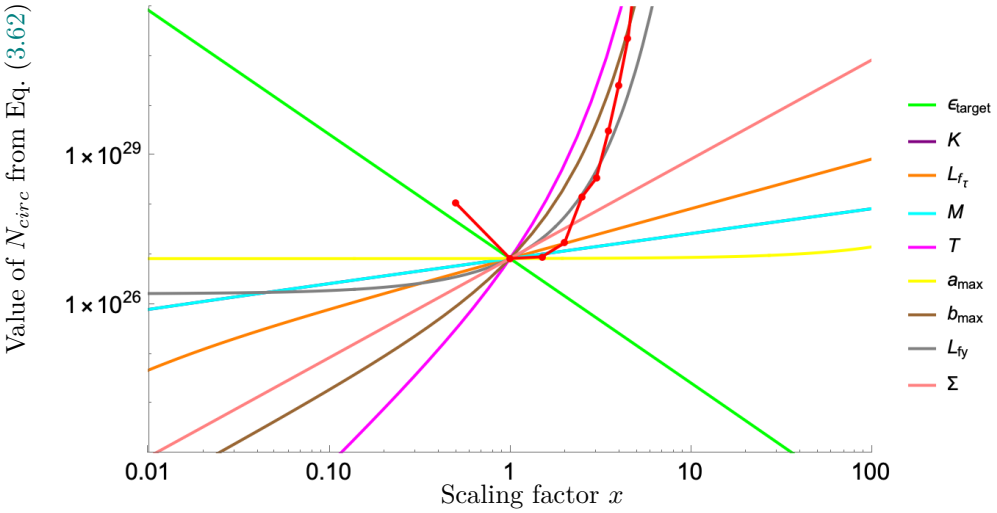

Figure 3.7: A plot comparing the sensitivity of N_{circ} with respect to different parameters. The intersecting point in the middle is the value of N_{circ} where all parameters are chosen default as given in Table 3.3 as well as $p = 2$ as a default. In each graph, we are changing one parameter, while keeping all the other parameters at default. We are changing the parameter by multiplying with the scaling factor x given as the abscissa. The following colors are corresponding to N_{circ} with one parameter changed: p, ϵ_{target}, $L_{f\tau}$, M and K, T, $a_{\max}$, $b_{\max}$, L_{fy}, Σ, where x is the scaling factor at the abscissa. The graphs for M and K overlap. The continuous red line for p is just for visualization purposes, as p is integer.

evaluations N_{circ} is equal to 1.62×10^{28}.

In Ref. [18, Supplementary information VIA], the time that the superconducting Sycamore chip can be used before having to be recalibrated is around 1 day. That implies that all calculations have to be done in at most 24 hours. The time for one readout of the Sycamore chip takes around $4\mu s$ [157]. That means that it is possible to evaluate around 2×10^{10} quantum circuits.

It is obvious that the high accuracy is needed because of the inversion of the matrix A and the resulting error propagation. The number $N_r^{(\delta)}$ given in Thm. 3.5 depends quadratically on the factor Σ estimated in (3.97) to be upper bounded with high probability by 3.4×10^8. If it was possible to reach a bound $\Sigma \leq 1$, the number $N_r^{(\delta)}$ would decrease to the order 10^7 which would be feasible for quantum hardware, .

In order to illustrate the potential resource savings that can be gained by choosing a higher-order RKM instead of the Euler method, we provide a second analysis based on a different choice of parameters provided in Table 3.4. This choice is inspired by the estimates in Table 3.3, changing some of the parameters within reasonable ranges in order to increase the resource savings. The resulting

resource requirements are shown in Fig. 3.9. We conclude that by choosing a Runge-Kutta method of order $p = 4$, there have to be done a factor of $\sim 2.56 \times 10^3$ less circuit evaluations than when choosing the Euler method. However, the number of shots for each circuit ($N_r^{(\delta)} \sim 1.98 \times 10^{26}$) is still too high in order to be realized on any quantum hardware.

p	$N_{circ}(p)$	$N_{circ}(1)/N_{circ}(p)$	$N_r^{(\delta)}$	$N_\tau^{(\delta)}$	Circuits
1	2.13×10^{29}	1	7.03×10^{21}	2.96×10^4	3.03×10^7
2	1.62×10^{28}	13.18	3.87×10^{22}	2.04×10^2	4.19×10^5
3	1.75×10^{28}	12.21	1.53×10^{23}	37.06	1.14×10^5
4	3.31×10^{28}	6.45	5.19×10^{23}	15.55	6.38×10^4
5	3.38×10^{29}	6.31×10^{-1}	5.48×10^{24}	10.03	6.17×10^4
6	7.79×10^{29}	2.74×10^{-1}	1.56×10^{25}	6.96	4.99×10^4
7	7.49×10^{30}	2.85×10^{-2}	1.41×10^{26}	5.74	5.30×10^4
8	7.00×10^{31}	3.05×10^{-3}	1.25×10^{27}	4.98	5.62×10^4
9	6.45×10^{32}	3.31×10^{-4}	1.08×10^{28}	4.47	5.96×10^4
10	2.16×10^{34}	9.9×10^{-6}	3.03×10^{29}	4.33	7.11×10^4

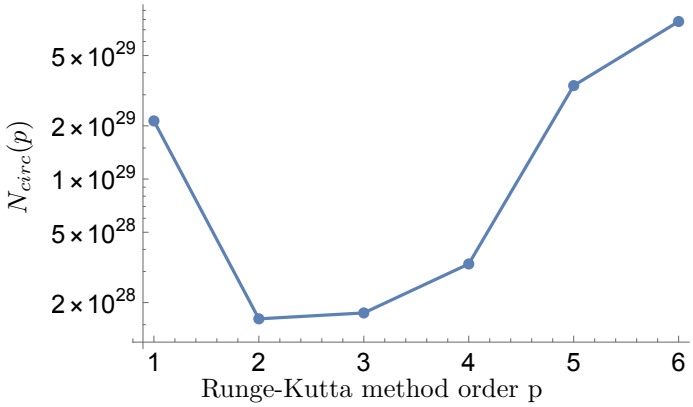

Figure 3.8: Comparisons of the resource requirements for different RKM orders, for the parameters in Table 3.3, applicable to the option pricing application. In the first column, we have the RKM order p. In the second column is N_{circ} of an algorithm that uses a RKM of order p and in the third column the ratio of N_{circ} of an algorithm that uses the Euler method ($p = 1$) with N_{circ} of an algorithm that uses a RKM of order p. In the third, fourth, and fifth columns we show $N_r^{(\delta)}$, the number of time steps $N_\tau^{(\delta)}$ and the number of different circuits. In the plot, we show $N_{circ}(p)$ plotted against p. The total number of circuit evaluations N_{circ} is calculated according to Eq. (3.62).

p	$N_{circ}(p)$	$N_{circ}(1)/N_{circ}(p)$	$N_r^{(\delta)}$	$N_\tau^{(\delta)}$	Circuits
1	1.12×10^{37}	1	1.15×10^{25}	9.56×10^{8}	9.80×10^{11}
2	2.63×10^{34}	4.28×10^{2}	3.93×10^{25}	3.26×10^{5}	6.68×10^{8}
3	6.33×10^{33}	1.78×10^{3}	9.57×10^{25}	2.15×10^{4}	6.61×10^{7}
4	4.39×10^{33}	2.56×10^{3}	1.98×10^{26}	5.41×10^{3}	2.22×10^{7}
5	1.00×10^{34}	1.12×10^{3}	6.78×10^{26}	2.40×10^{3}	1.48×10^{7}
6	1.11×10^{34}	1.01×10^{3}	1.14×10^{27}	1.36×10^{3}	9.76×10^{6}
7	2.60×10^{34}	4.33×10^{2}	3.06×10^{27}	9.22×10^{2}	8.50×10^{6}
8	5.90×10^{34}	1.91×10^{2}	7.61×10^{27}	6.88×10^{2}	7.75×10^{6}
9	1.33×10^{35}	84.69	1.82×10^{28}	5.47×10^{2}	7.29×10^{6}
10	4.92×10^{35}	22.87	6.48×10^{28}	4.63×10^{2}	7.59×10^{6}

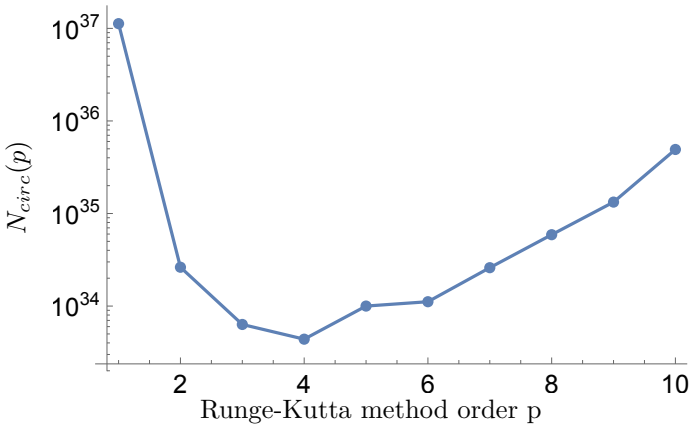

Figure 3.9: Comparison of N_{circ} for different RKM orders, where the estimated parameters are fine-tuned to maximize the savings by using a higher order p as given in Table 3.4. In the first column, we have the Runge-Kutta method order p. In the second column is N_{circ} of an algorithm that uses a Runge-Kutta method of order p and in the third column the ratio of N_{circ} of an algorithm that uses the Euler method ($p = 1$) with N_{circ} of an algorithm that uses a Runge-Kutta method of order p. In the third, fourth, and fifth columns, we show the number of shots per circuit $N_r^{(\delta)}$, the number of time steps $N_\tau^{(\delta)}$, and the number of circuits. In the plot, we show $N_{circ}(p)$ plotted against p. The total number of circuit evaluations N_{circ} is calculated according to Eq. (3.62).

3.7 Conclusions

In this chapter, we developed error and resource estimates of variational algorithms for solving differential equations based on Runge-Kutta methods. In particular, the estimates depend on different parameters that come from both the chosen method and the differential equation at hand, and depend on both

Parameter	Estimate
a_{max}	1
b_{max}	0.5
L_{fy}	0.1
T	4
K	20
$L_{f\tau}$	15
M	60
S	1
Σ	3.4×10^8
ϵ_{target}	0.001

Table 3.4: Estimates of various parameters that optimize the savings by using a higher-order Runge-Kutta method instead of the Euler method. Used for the analysis in Fig. 3.9.

the error from the Runge-Kutta method and the error that comes from shot noise in the noisy evaluations of the differential function. Additionally, we performed numerical simulations of the minimal resources required for both solving a simple ODE by using Runge-Kutta methods without shot noise and for solving a variational algorithm applied to a linear PDE from option pricing. This shows that our method is not restricted to ODEs but can also be applied to solving partial or stochastic differential equations. Furthermore, the algorithms we analyzed are based on the variational quantum algorithms for solving real and imaginary time evolution [116, 117]; our analysis can therefore be directly applied to them.

Our results suggest that depending on certain parameters, such as the difference between initial and final time, Lipschitz constants of the differential function, and target error, the resource requirements change drastically.

Moreover, we show that for the particular ODE that we solve in the case without shot noise and a chosen target error of at most $\epsilon_{target} = 0.001$, a fourth-order Runge-Kutta method is the most resource-efficient, while for the option pricing use case solved by a variational quantum algorithm and the same target error of $\epsilon_{target} = 0.001$, the most resource-efficient method is a second-order Runge-Kutta method. By choosing these methods, we can minimize the total cost by a factor of 2.22×10^3 for the first case and the total number of quantum circuit evaluations by a factor of 13.18 for the option pricing use case. For solving the option pricing PDE, even when using a Runge-Kutta method with the order of $p = 2$, we estimated that the algorithm needs at least 1.62×10^{28} evaluations of quantum circuits in order to compute the option price at final time with a maximum error of $\epsilon_{target} = 0.001$ in trace distance. In practical scenarios, the state has to be read out, and therefore the total resource

requirements will be even higher.

However, our results are worst-case estimates that guarantee staying within the target error, while in practice, one might expect to achieve them with lower resource requirements for sufficiently well-behaved functions. In particular, Lemma 4 gives an upper bound for errors induced by solving linear systems. But even for ill-conditioned problems, a smaller error is possible in practice (see for example Ref. [158]). Also, the estimates in Secs. 3.5.3 and 3.6 are upper bounds that can in general be tightened for specific problems at hand.

The PDE chosen in our use case, the Black-Scholes equation, can be solved analytically, so only more complicated dynamics are of practical interest. Considering this and the fact that a requirement of at least 3.87×10^{22} evaluations for each of the 4.19×10^5 quantum circuits (of which only 10^3 can be run in parallel, see Sec. 3.5.1) is unrealistic on current quantum computing platforms, our results are rather pessimistic for practical implementations of this algorithm in option pricing. As discussed in the previous section, this high requirement of quantum circuit evaluations mainly stems from the error-propagation of inverting a matrix of estimated observables. This leads us to question the general feasibility of the underlying quantum algorithm. However, related approaches as proposed in Refs. [119] and [120] which do not require matrix inversion, might therefore require a much lower number of quantum circuit evaluations.

We show that by tuning the parameters in the variational algorithm and therefore reaching realms outside of this use case, we can get a saving factor of 2.56×10^3 by choosing a Runge-Kutta method of order $p = 4$ instead of the Euler method. This suggests that after careful analysis of the parameters, one can choose a Runge-Kutta method that minimizes the resource requirements.

Our analysis had a number of limitations, which however likely do not make matters better. We do not analyze possible stability issues of the differential equations, because it proves very challenging to include them in the estimation of the variational algorithm and the option pricing use case. However, for a full picture of the error and resource analysis, the stability of methods and DEs has to be considered. We neglected possible representation errors that stem from the quantum circuits not being able to approximate a state that encodes the option price and which can capture the dynamics of the variational parameters satisfyingly. In practical scenarios, these have to be taken into account and can possibly be estimated depending on the chosen quantum computing platform (see for example Ref. [131]). Further, we assumed circuit error such as gate infidelity, bias and SPAM errors to be negligible, as well as errors introduced to potentially necessary matrix regularization of the matrix defined in Eq. (3.27). However, as long as when adding those sources of errors, our error bounds (for example in Eq. (3.84)) still hold, our results can be applied to these scenarios as well.

It might be possible to tighten our estimates in a few different ways. Recent works [135, 152] showed how it is possible to decrease the number of state

preparations for estimations of the matrices A from scaling as $\Theta(N_V^2)$ to $\Theta(N_V)$ at the cost of accuracy of the matrix entries. Combining these results with our analysis might further decrease the total resource requirements. Several bounds in our analysis are not tight and could show to be overly pessimistic in real scenarios, like the local truncation error of the Runge-Kutta method (Thm. 3.1) and the bounds and estimates on the shot noise (Lemma 3 and Sec. 3.6.2). Further, the resource requirements can possibly be decreased by using linear multistep methods that only use one new stage per time step and reuse evaluations from previous time steps. Adapting our analysis to these methods can give new insights into a comparison of Runge-Kutta methods and linear multistep methods.

Since the shot noise error introduced for the analysis in Sec. 3.4 is Gaussian noise, it might give further insight to formulate the differential equations as stochastic differential equations and to analyze the error and resource requirements within this framework.

It might be promising to apply this framework to use cases that use Runge-Kutta methods and have similar error sources, such as quantum algorithms for solving other differential equations or to classical algorithms that are using noisy data, which so far has been barely examined. Also, it might be possible to apply this analysis to the training of neural networks [159].

3.A Proof of Theorems 3.2 and 3.4

Proof. Let us denote the RKM calculated $\boldsymbol{\theta}_n$ with an error-carrying $\hat{f}$ at time step n as $\hat{\boldsymbol{\theta}}_n$. Calculating the LTE in $\hat{\boldsymbol{\theta}}_n$, we assume that it is calculated from a noiseless $\boldsymbol{\theta}(\tau_{n-1})$. For an s-stage RKM $\hat{\boldsymbol{\theta}}_n$ is calculated by:

$$\hat{\boldsymbol{\theta}}_n = \boldsymbol{\theta}(\tau_{n-1}) + \Delta\tau \sum_{i=1}^{s} b_i \hat{k}_i \left(\tau_{n-1}; \boldsymbol{\theta}(\tau_{n-1}); \{\hat{k}_m\}_{m=1}^{i-1}\right) , \tag{3.98}$$

where we write

$$\hat{k}_1(\tau_{n-1}; \boldsymbol{\theta}(\tau_{n-1})) := \hat{f}\left(\tau_{n-1}; \boldsymbol{\theta}(\tau_{n-1})\right), \quad \text{and}$$

$$\hat{k}_i \left(\tau_{n-1}; \boldsymbol{\theta}(\tau_{n-1}); \{\hat{k}_m\}_{m=1}^{i-1}\right) := \hat{f}\left(\tau_{n-1} + c_i\Delta\tau; \boldsymbol{\theta}(\tau_{n-1}) + \Delta\tau \sum_{m=1}^{i-1} a_{i,m}\hat{k}_m\right),$$

for all $i \geq 2$,

and where we used the abbreviation $\hat{k}_m$ by dropping the arguments.

Thus, the LTE in the presence of noise is:

$$\hat{\ell_n} := \boldsymbol{\theta}(\tau_n) - \hat{\boldsymbol{\theta}}_n = \boldsymbol{\theta}(\tau_n) - \boldsymbol{\theta}(\tau_{n-1}) - \Delta\tau \sum_{i=1}^{s} b_i \hat{k}_i \left(\tau_{n-1}; \boldsymbol{\theta}(\tau_{n-1}); \{\hat{k}_m\}_{m=1}^{i-1}\right) \tag{3.99}$$

We can immediately write

$$\hat{\ell}_n \leq \boldsymbol{\theta}(\tau_n) - \boldsymbol{\theta}_n + \boldsymbol{\theta}_n - \hat{\boldsymbol{\theta}}_n \leq \ell_n + \boldsymbol{\theta}_n - \hat{\boldsymbol{\theta}}_n \ , \tag{3.100}$$

where the noiseless LTE ℓ_n can be upper bounded by the bound in Thm. 3.1.

Let us define the global truncation error of an RKM with s stages and n time steps in the presence of noise as:

$$\hat{e}_n := \boldsymbol{\theta}(\tau_n) - \hat{\boldsymbol{\theta}}_0 - \Delta\tau \sum_{r=1}^{n} \sum_{i=1}^{s} b_i \breve{k}_i \left(\tau_{r-1}; \hat{\boldsymbol{\theta}}_{r-1}; \{\breve{k}_m\}_{m=1}^{i-1}\right) \ , \tag{3.101}$$

where we write

$$\breve{k}_1 \left(\tau_{r-1}; \hat{\boldsymbol{\theta}}_{r-1}\right) := \breve{f} \left(\tau_{r-1}; \hat{\boldsymbol{\theta}}_{r-1}\right), \quad \text{and}$$

$$\breve{k_i} \left(\tau_{r-1}; \hat{\boldsymbol{\theta}}_{r-1}; \{\breve{k}_m\}_{m=1}^{i-1}\right) := \breve{f} \left(\tau_{r-1} + c_i \Delta\tau; \hat{\boldsymbol{\theta}}_{r-1} + \Delta\tau \sum_{m=1}^{i-1} a_{i,m} \breve{k}_m\right),$$

for all $i \geq 2$,

and again use the abbreviation $\breve{k}_m$ by dropping the arguments. We write the superscript $\breve{f}$ instead of $\hat{f}$ in order to distinguish the noisy evaluation with noisy inputs with the noisy evaluation with noiseless inputs. They both however carry the error δ compared to the noise-free case, as written in Eq. (3.46). Recursively, we get:

$$\hat{e}_{n+1} - \hat{e}_n = \hat{\ell}_{n+1} + \Delta\tau \sum_{i=1}^{s} b_i \left(\hat{k}_i \left(\tau_n; \boldsymbol{\theta}(\tau_n); \{\hat{k}_m\}_{m=1}^{i-1}\right) - \breve{k}_i \left(\tau_n; \hat{\boldsymbol{\theta}}_n; \{\breve{k}_m\}_{m=1}^{i-1}\right)\right) \ . \tag{3.102}$$

Taking the absolute values from the latter equation, we get

$$\|\hat{e}_{n+1}\| \leq \|\hat{e}_n\| + \|\hat{\ell}_{n+1}\| \tag{3.103}$$

$$+ \Delta\tau \sum_{i=1}^{s} |b_i| \left\| \hat{k}_i \left(\tau_n; \boldsymbol{\theta}(\tau_n); \{\hat{k}_m\}_{m=1}^{i-1}\right) - \breve{k}_i \left(\tau_n; \hat{\boldsymbol{\theta}}_n; \{\breve{k}_m\}_{m=1}^{i-1}\right) \right\| \ . \tag{3.104}$$

We assume that the function $f(\tau, \boldsymbol{\theta}(\tau))$ satisfies the Lipschitz condition:

$$\left\| f(\tau_n, \boldsymbol{\theta}(\tau_n)) - f(\tau_n, \hat{\boldsymbol{\theta}}_n) \right\| \leq L_{fy} \|\boldsymbol{\theta}(\tau_n) - \hat{\boldsymbol{\theta}}_n\| \ , \tag{3.105}$$

and that the noise evaluations are upper bounded by

$$\|\hat{k}_i - k_i\| \leq \delta \tag{3.106}$$

$$\|\breve{k}_i - k_i\| \leq \delta \ . \tag{3.107}$$

We introduce the notation

$$S_i := \left\| \hat{k}_i \left(\tau_n; \boldsymbol{\theta}(\tau_n); \{\hat{k}_m\}_{m=1}^{i-1} \right) - \breve{k}_i \left(\tau_n; \hat{\boldsymbol{\theta}}_n; \{\breve{k}_m\}_{m=1}^{i-1} \right) \right\| . \tag{3.108}$$

Using the triangle inequality, we get

$$S_1 = \left\| \hat{k}_1 \left(\tau_n; \boldsymbol{\theta}(\tau_n) \right) \right) - \breve{k}_1 \left(\tau_n; \hat{\boldsymbol{\theta}}_n \right) \Big\| \tag{3.109}$$

$$\leq \left\| \hat{k}_1 \left(\tau_n; \boldsymbol{\theta}(\tau_n) \right) \right) - k_1 \left(\tau_n; \boldsymbol{\theta}(\tau_n) \right) \right) \Big\| + \left\| k_1 \left(\tau_n; \hat{\boldsymbol{\theta}}_n \right) - \breve{k}_1 \left(\tau_n; \hat{\boldsymbol{\theta}}_n \right) \right\|$$
$$+ \left\| k_1 \left(\tau_n; \boldsymbol{\theta}(\tau_n) \right) \right) - k_1 \left(\tau_n; \hat{\boldsymbol{\theta}}_n \right) \right\| \tag{3.110}$$

$$\leq 2\delta + L_{fy} \|\hat{e}_n\| . \tag{3.111}$$

Again using the triangle inequality, we get for S_i, $i \geq 2$:

$$S_i = \left\| \hat{k}_i \left(\tau_n; \boldsymbol{\theta}(\tau_n); \{\hat{k}_m\}_{m=1}^{i-1} \right) - \breve{k}_i \left(\tau_n; \hat{\boldsymbol{\theta}}_n; \{\breve{k}_m\}_{m=1}^{i-1} \right) \right\| \tag{3.112}$$

$$\leq \left\| \hat{k}_i \left(\tau_n; \boldsymbol{\theta}(\tau_n); \{\hat{k}_m\}_{m=1}^{i-1} \right) - k_i \left(\tau_n; \boldsymbol{\theta}(\tau_n); \{\hat{k}_m\}_{m=1}^{i-1} \right) \right\| \tag{3.113}$$

$$+ \left\| k_i \left(\tau_n; \hat{\boldsymbol{\theta}}_n; \{\breve{k}_m\}_{m=1}^{i-1} \right) - \breve{k}_i \left(\tau_n; \hat{\boldsymbol{\theta}}_n; \{\breve{k}_m\}_{m=1}^{i-1} \right) \right\| \tag{3.114}$$

$$+ \left\| k_i \left(\tau_n; \boldsymbol{\theta}(\tau_n); \{\hat{k}_m\}_{m=1}^{i-1} \right) - k_i \left(\tau_n; \hat{\boldsymbol{\theta}}_n; \{\breve{k}_m\}_{m=1}^{i-1} \right) \right\| \tag{3.115}$$

$$\leq 2\delta + L_{fy} \left\| \boldsymbol{\theta}(\tau_n) - \hat{\boldsymbol{\theta}}_n \right\| + \Delta\tau \sum_{m=1}^{i-1} L_{fy} |a_{im}| \left\| \hat{k}_m - \breve{k}_m \right\| \tag{3.116}$$

$$\leq 2\delta + L_{fy} \|\hat{e}_n\| + \Delta\tau \sum_{m=1}^{i-1} L_{fy} |a_{im}| S_m \tag{3.117}$$

$$\leq S_1 + \Delta\tau \sum_{m=1}^{i-1} L_{fy} |a_{im}| S_m \ . \tag{3.118}$$

Then, we can write the upper bound of the error of the s order RKM as

$$\|\hat{e}_{n+1}\| \leq \|\hat{e}_n\| + \|\hat{\ell}_{n+1}\| + \Delta\tau \sum_{i=1}^{s} |b_i| S_i \ , \tag{3.119}$$

where from the above analysis, we obtained the following recursion:

$$S_1 \leq 2\delta + L_{fy}\|\hat{e}_n\| \tag{3.120}$$

$$S_i \leq S_1 + \Delta\tau \sum_{m=1}^{i-1} L_{fy}|a_{im}|S_m, \quad \forall i \geq 2. \tag{3.121}$$

Let us use that $|a_{ij}|$ is upper bounded with $a_{max} := \max_{i,j} |a_{i,j}|$. Then we can write:

$$S_1 \leq 2\delta + L_{fy}\|\hat{e}_n\| \tag{3.122}$$

$$S_i \leq S_1 + \Delta\tau L_{fy} a_{max} \sum_{m=1}^{i-1} S_m, \quad \forall i \geq 2. \tag{3.123}$$

A proof by induction shows that for all $i \geq 1$,

$$S_i \leq S_1 \left(1 + \Delta\tau L_{fy} a_{max}\right)^{i-1}. \tag{3.124}$$

This result together with Eq. (3.119) gives us the error estimate:

$$\|\hat{e}_{n+1}\| \leq \|\hat{e}_n\| + \|\hat{\ell}_{n+1}\| + \Delta\tau S_1 \sum_{i=1}^{s} |b_i| \left(1 + \Delta\tau L_{fy} a_{max}\right)^{i-1} \tag{3.125}$$

$$\leq \|\hat{e}_n\| + \|\hat{\ell}_{n+1}\| + \Delta\tau S_1 b_{max} \frac{\left(1 + \Delta\tau L_{fy} a_{max}\right)^s - 1}{L_{fy}\Delta\tau a_{max}} \tag{3.126}$$

$$\leq \|\hat{e}_n\| + \left(\max_n \|\hat{\ell}_n\| + (2\delta + L_{fy}\|\hat{e}_n\|)\, b_{max} \frac{\left(1 + \Delta\tau L_{fy} a_{max}\right)^s - 1}{L_{fy} a_{max}} \right)$$

$$\leq \alpha\|\hat{e}_n\| + \beta, \quad \text{for all} \quad s \geq 1, \tag{3.127}$$

where we used the notation

$$\alpha = 1 + F(n+1, s), \quad \beta = \frac{2\delta}{L_{fy}} F(n+1, s) + \max_n \|\hat{\ell}_n\|, \tag{3.128}$$

$$F(n+1, s) := \frac{b_{max}}{a_{max}} \left(\left(1 + \frac{\Theta}{n+1}\right)^s - 1 \right), \quad \Theta = L_{fy} a_{max} T, \tag{3.129}$$

an the fact that $\Delta\tau = T/(n+1)$ holds. . Because $\|\hat{e}_0\| = 0$, we get

$$\|\hat{e}_{n+1}\| \leq \frac{\alpha^{n+1}-1}{\alpha-1}\beta. \tag{3.130}$$

The error estimate of the LTE goes analogous to the error estimate of the global truncation error. We can write

$$\|\hat{\ell}_n\| \leq \|\ell_n\| + \Delta\tau \sum_{i=1}^{s} |b_i| T_i \tag{3.131}$$

with the recursive relation

$$T_1 := \delta, \quad T_i := T_1 + \Delta\tau \sum_{m=1}^{i-1} L_{fy}|a_{im}|T_m. \tag{3.132}$$

Thus, we get the bounds

$$\|\hat{\ell}_n\| \leq \|\ell_n\| + \frac{\delta}{L_{fy}} F(n+1, s), \quad \text{for } s \geq 1. \tag{3.133}$$

Using these upper bounds, we can rewrite Eq. (3.130) as follows

$$\|\hat{e}_{n+1}\| \leq \frac{\alpha^{n+1}-1}{\alpha-1}\beta^* \ , \tag{3.134}$$

where

$$\beta^* = \frac{3\delta}{L_{fy}} F(n+1, s) + \max_n \|\ell_n\| \tag{3.135}$$

According to Thm. 3.1, we can bound the LTE ℓ_n and get after $N_t := n+1$ steps the following error:

$$\|\hat{e}_{N_\tau}\| \leq \frac{(1+F(N_\tau, s))^{N_\tau} - 1}{F(N_\tau, s)} \left(\frac{3\delta}{Lfy} F(N_\tau, s) + \left(\frac{T}{N_\tau}\right)^{p+1} K L_{f\tau}^p M \right). \tag{3.136}$$

That ends the proof for Thm. 3.4, for which $f(\tau, \boldsymbol{\theta}(\tau)) = f(\boldsymbol{\theta}(\tau))$. One can see that noise free case corresponds to $\delta = 0$, for which $y(\tau_n)$ corresponds to $\boldsymbol{\theta}(\tau_n)$ and y_n to $\boldsymbol{\theta}_n$, which proofs Thm. 3.2. □

3.B Proof of Theorem 3.3

Proof. Let us denote the upper bound of the noise free $\epsilon_{ODE}^{(\delta)}$ as

$$\epsilon_{target}^{(0)} := \frac{(1+F(N_\tau,s))^{N_\tau}-1}{F(N_\tau,s)}\left(\frac{T}{N_\tau}\right)^{p+1} KL_{f\tau}^{p}M \ , \tag{3.137}$$

To find N_τ we can solve the latter equation numerically. However, under the reasonable assumption that $\Theta << N_\tau$, we can use the following approximations

$$\left(\frac{\Theta}{N_\tau}+1\right)^{s} = 1+\frac{s\Theta}{N_\tau}+O\left(\frac{\Theta^2}{N_\tau^2}\right), \tag{3.138}$$

$$\left(\frac{b_{max}(s\Theta)}{a_{max}N_\tau}+1\right)^{N_\tau} \approx \exp\left(\frac{b_{max}s\Theta}{a_{max}}\right), \tag{3.139}$$

to rewrite

$$F(N_\tau,s) \approx \frac{b_{max}}{a_{max}}\frac{s\Theta}{N_\tau}, \quad (1+F(N_\tau,s))^{N_\tau} \approx \exp\left(\frac{b_{max}s\Theta}{a_{max}}\right) \tag{3.140}$$

Using this, we rewrite Eq. (3.137) in a way to obtain an approximate solution $N_\tau^{(0)}$ given by Eq. (3.41). □

3.C Proof of Theorem 3.5

Proof. We model the shot noise with single shot variance Σ and the number of shots N_r, so $\delta = \Sigma/\sqrt{N_r}$ holds. Solving the resulting Eq. (3.48) for N_r we get:

$$N_r^{(\delta)} := \frac{9\Sigma^2}{L_{fy}^2}\left(\frac{\epsilon_{target}^{(\delta)}}{(1+F(N_\tau,s))^{N_\tau}-1}-\left(\frac{T}{N_\tau}\right)^{p+1}\frac{KL_{f\tau}^{p}M}{F(N_\tau,s)}\right)^{-2}, \tag{3.141}$$

Substituting it in the cost function (3.35) we get:

$$\mathcal{C}(N_\tau,N_r^{(\delta)},s,p) = \frac{9\Sigma^2 p}{L_{fy}^2}N_\tau^{2p+3} \tag{3.142}$$

$$\times\left(\frac{F(N_\tau,s)((1+F(N_\tau,s))^{N_\tau}-1)}{\epsilon_{target}^{(\delta)}F(N_\tau,s)N_\tau^{p+1}-T^{p+1}KL_{f\tau}^{p}M((1+F(N_\tau,s))^{N_\tau}-1)}\right)^{2},$$

where

$$\epsilon^{(\delta)}_{target} \neq \frac{T^{p+1} K L^p_{f\tau} M((1+F(N_\tau,s))^{N_\tau}-1)}{F(N_\tau,s) N_\tau^{p+1}} = \epsilon^{(0)}_{target}, \tag{3.143}$$

that is always true. We want to find the optimal value for N_τ that minimizes the latter cost. To this end, we set the derivative of the cost function with respect to N_τ to zero. The equation we get in nonlinear and can be solved numerically.

However, we can use the approximation in Eq. (3.139) and the expression

$$N_\tau \log\left(\frac{b_{max} s\Theta}{a_{max} N_\tau}+1\right) = \log\left(\frac{b_{max} s\Theta}{a_{max} N_\tau}+1\right)^{N_\tau} \approx \left(\frac{b_{max} s\Theta}{a_{max}}\right). \tag{3.144}$$

Those two approximations give the following simplified equation

$$\frac{\epsilon^{(\delta)}_{target}\left(a_{max} N_\tau(N_\tau+\Theta) - b_{max} s\Theta\left(2N_\tau^2 e^{\frac{b_{max}s\Theta}{a_{max}}} + N_\tau\left(2s\Theta e^{\frac{b_{max}s\Theta}{a_{max}}}-1\right)-\Theta\right)\right)}{(N_\tau+\Theta)(a_{max}N_\tau+b_{max}s\Theta)\left(e^{\frac{b_{max}s\Theta}{a_{max}}}-1\right)}$$
$$+\frac{2\epsilon^{(\delta)}_{target}(b_{max}s\Theta)e^{\frac{b_{max}s\Theta}{a_{max}}}}{a_{max}\left(e^{\frac{b_{max}s\Theta}{a_{max}}}-1\right)} - \frac{a_{max} K M L^p_{f\tau}\left(\frac{T}{N_\tau}\right)^{p+1}(2N_\tau p + N_\tau - 2(s-1)\Theta)}{b_{max}s\Theta}$$
$$= 0\ .$$

Neglecting some of the terms due to the fact that $\Theta << N_\tau$ we simplify the equation to the form that it can be solved analytically:

$$N^{(\delta)}_\tau = T L_{f\tau}\left(\frac{\left(e^{b_{max}sL_{fy}T}-1\right)KM(2p+1)}{s b_{max} L_{fy}\epsilon^{(\delta)}_{target}}\right)^{1/p}. \tag{3.145}$$

That ends the proof. □

3.D McLachlan's variational principle

Theorem 3.6. *McLachlan's variational principle [149] applied to the imaginary time evolution of a state $|\psi(\boldsymbol{\theta}(\tau))\rangle$, is given by:*

$$\delta\|(d/d\tau+\mathcal{H})\,|\psi(\theta(\tau))\rangle\,\| = 0\ . \tag{3.146}$$

Assuming θ to be real, it is solved by the following ordinary differential equations [150]:

$$\sum_j A_{ij}\dot{\theta}_j = C_i, \tag{3.147}$$

where the matrix elements are

$$A_{ij} = \mathfrak{Re}\left(\frac{\partial \langle\psi(\theta(\tau))|}{\partial\theta_i}\frac{\partial |\psi(\theta(\tau))\rangle}{\partial\theta_j}\right), \tag{3.148}$$

$$\text{and} \quad C_i = \mathfrak{Re}\left(-\frac{\partial \langle\psi(\theta(\tau))|}{\partial\theta_i}\mathcal{H}|\psi(\theta(\tau))\rangle\right). \tag{3.149}$$

Proof. Note that the variation in Eq. (3.146) is equivalent to the variation of its square $\delta\|(d/d\tau+\mathcal{H})|\psi(\theta(\tau))\rangle\|^2 = 0$. The derivative can be written as follows

$$\frac{\partial |\psi(\theta(\tau))\rangle}{\partial\tau} = \frac{\partial |\psi(\theta(\tau))\rangle}{\partial\theta_i}\frac{\partial\theta_i}{\partial\tau} = \frac{\partial |\psi(\theta(\tau))\rangle}{\partial\theta_i}\dot{\theta}_i. \tag{3.150}$$

Next, we expand the following:

$$\begin{aligned}\|(d/d\tau+\mathcal{H})|\psi(\theta(\tau))\rangle\|^2 &= \big((d/d\tau+\mathcal{H})|\psi(\theta(\tau))\rangle\big)^\dagger\big((d/d\tau+\mathcal{H})|\psi(\theta(\tau))\rangle\big) \\ &= \sum_{ij}\frac{\partial \langle\psi(\theta(\tau))|}{\partial\theta_i}\frac{\partial |\psi(\theta(\tau))\rangle}{\partial\theta_j}\dot{\theta}_i^*\dot{\theta}_j + \sum_i\frac{\partial \langle\psi(\theta(\tau))|}{\partial\theta_i}(\mathcal{H})|\psi(\theta(\tau))\rangle\dot{\theta}_i^* \\ &+ \sum_i\langle\psi(\theta(\tau))|\mathcal{H}\frac{\partial |\psi(\theta(\tau))\rangle}{\partial\theta_i}\dot{\theta}_i + \langle\psi(\theta(\tau))|\mathcal{H}^2|\psi(\theta(\tau))\rangle\ .\end{aligned} \tag{3.151}$$

Thus, the variation of this term with respect to $\dot{\theta}_i$ yields:

$$\begin{aligned}&\delta\|(d/d\tau+\mathcal{H})|\psi(\theta(\tau))\rangle\|^2 \\ &= \left(\sum_{ij}\frac{\partial \langle\psi(\theta(\tau))|}{\partial\theta_i}\frac{\partial |\psi(\theta(\tau))\rangle}{\partial\theta_j}\dot{\theta}_j + \frac{\partial \langle\psi(\theta(\tau))|}{\partial\theta_i}\mathcal{H}|\psi(\theta(\tau))\rangle\right)\delta\dot{\theta}_i^* \\ &+ \left(\sum_{ij}\frac{\partial \langle\psi(\theta(\tau))|}{\partial\theta_j}\frac{\partial |\psi(\theta(\tau))\rangle}{\partial\theta_i}\dot{\theta}_j^* + \langle\psi(\theta(\tau))|\mathcal{H}\frac{\partial |\psi(\theta(\tau))\rangle}{\partial\theta_i}\right)\delta\dot{\theta}_i\ .\end{aligned} \tag{3.152}$$

The variational principle is satisfied if latter equation is equal to zero, hence

when following expression holds:

$$\sum_j \frac{\partial \langle\psi(\theta(\tau))|}{\partial\theta_i}\frac{\partial |\psi(\theta(\tau))\rangle}{\partial\theta_j}\dot{\theta}_j = -\frac{\partial \langle\psi(\theta(\tau))|}{\partial\theta_i}\mathcal{H}|\psi(\theta(\tau))\rangle \ . \tag{3.153}$$

If $\dot{\theta}_i$ is real, the variation changes to

$$\begin{aligned}&\delta\|(d/d\tau+\mathcal{H})|\psi(\theta(\tau))\rangle\|^2 \\ =&\sum_j\left(\frac{\partial \langle\psi(\theta(\tau))|}{\partial\theta_i}\frac{\partial |\psi(\theta(\tau))\rangle}{\partial\theta_j}+\frac{\partial \langle\psi(\theta(\tau))|}{\partial\theta_j}\frac{\partial |\psi(\theta(\tau))\rangle}{\partial\theta_i}\right)\dot{\theta}_j\delta\dot{\theta}_i \\ &+\left(\frac{\partial \langle\psi(\theta(\tau))|}{\partial\theta_i}\mathcal{H}|\psi(\theta(\tau))\rangle+\langle\psi(\theta(\tau))|\mathcal{H}\frac{\partial |\psi(\theta(\tau))\rangle}{\partial\theta_i}\right)\delta\dot{\theta}_i \ .\end{aligned} \tag{3.154}$$

This is equal to zero when the following holds:

$$\begin{aligned}&\sum_j\left(\frac{\partial \langle\psi(\theta(\tau))|}{\partial\theta_i}\frac{\partial |\psi(\theta(\tau))\rangle}{\partial\theta_j}+\frac{\partial \langle\psi(\theta(\tau))|}{\partial\theta_j}\frac{\partial |\psi(\theta(\tau))\rangle}{\partial\theta_i}\right)\dot{\theta}_j \\ =&-\left(\frac{\partial \langle\psi(\theta(\tau))|}{\partial\theta_i}\mathcal{H}|\psi(\theta(\tau))\rangle+\langle\psi(\theta(\tau))|\mathcal{H}\frac{\partial |\psi(\theta(\tau))\rangle}{\partial\theta_i}\right) \ ,\end{aligned} \tag{3.155}$$

which is equivalent to Eq. (3.147). □

3.E Shot noise estimates

Theorem 3.7. *(Shot noise error for evaluating A defined in Eq. (3.27))*
For N_r evaluations of each of the circuits that calculate the matrix elements of A, we get the following bound for the probability:

$$P\left(\|A-\hat{A}_S\|<\frac{\|\{\sigma_{k,l}\}_{k,l=1}^{N_V}\|}{\sqrt{N_r\eta}}\right)>1-\eta \ , \tag{3.156}$$

where $0<\eta\leq 1$ and the elements of the standard deviation matrix are

$$\sigma_{k,l}=\sqrt{\sum_{i,j=1}^{N_d}|f^*_{k,i}f_{l,j}|^2} \ . \tag{3.157}$$

Proof. For a random matrix A with finite non-zero variance matrix σ^2 and

expectation values matrix $\hat{A}_S$ the multi-dimensional Chebyshev's inequality

$$P(\|A - \hat{A}_S\| \geq k\|\sigma\|) \leq \frac{1}{k^2} \ , \tag{3.158}$$

holds, for any real number $k > 0$. If we measure each circuit N_r times, the mean value is calculated as

$$\hat{A}_S = \frac{1}{N_r} \sum_{m=1}^{N_r} \hat{A}_m \ , \tag{3.159}$$

where each $\hat{A}_m$ is the matrix calculated by evaluating each circuit one time. The $\sigma_{k,l}(\hat{A}_m)$ is the standard deviation for each matrix element of $\hat{A}_m$, and the total standard deviation for each element in $\hat{A}_S$ is given by

$$\sigma_{k,l}(\hat{A}_S) = \frac{\sigma_{k,l}(\hat{A}_m)}{\sqrt{N_r}} \ . \tag{3.160}$$

These are the elements that form the matrix $\sigma = \{\sigma_{k,l}(\hat{A}_m)\}_{k,l=1}^{N_V}$. Defining $\eta := 1/k^2$, we get therefore

$$P\left(\|A - \hat{A}_S\| < \frac{\|\sigma\|}{\sqrt{N_r \eta}}\right) > 1 - \eta \ . \tag{3.161}$$

We can further bound the elements of σ in the following way. Each matrix element $\sigma_{k,l}(\hat{A}_S)$ is in general evaluated by several circuits. Let us bound the standard deviation of each single circuit by the maximum standard deviation 1, since the eigenvalues of the Pauli-X matrix are $\{1, -1\}$. We obtain:

$$\sigma_{k,l}(\hat{A}_m) \leq \sigma_{k,l} = \sqrt{\sum_{i,j=1}^{N_d} |f^*_{k,i} f_{l,j}|^2}. \tag{3.162}$$

That ends the proof. □

Theorem 3.8. *(Shot noise error for evaluating C defined in Eq. (3.27))*
For N_r evaluations of each of the circuits that calculate the elements of C, we get the following bound:

$$P\left(\|C - \hat{C}_S\| < \frac{\|\{\sigma_k\}_{k=1}^{N_V}\|}{\sqrt{N_r \eta}}\right) > 1 - \eta \ , \tag{3.163}$$

where $0 < \eta \leq 1$ *and*

$$\sigma_k = \sqrt{\sum_{i=1}^{N_d} \sum_{m=1}^{N} |f_{i,k}^* \lambda_m|^2} \ , \tag{3.164}$$

holds.

Proof. The proof is similar to the proof of Thm. 3.7, but the standard deviation of the matrix elements is bounded by Eq. (3.164). □

Corollary 3. *The result of the Thms. 3.7 and 3.8 are valid for all matrices* A *and vectors* C *that are calculated with the circuits in Fig. 3.1, in particular for all possible input parameter* $\boldsymbol{\theta}$ *of the Ansatz in Eq. (3.18). Therefore, we can state that with probability of at least* $1 - \eta$ *the following bounds hold for all* $\boldsymbol{\theta}$*:*

$$\|A(\boldsymbol{\theta}) - \hat{A}(\boldsymbol{\theta})\| \leq \frac{\|\{\sigma_{k,l}\}_{k,l=1}^{N_V}\|}{\sqrt{N_r \eta}} \ , \tag{3.165}$$

and

$$\|C(\boldsymbol{\theta}) - \hat{C}(\boldsymbol{\theta})\| \leq \frac{\|\{\sigma_k\}_{k=1}^{N_V}\|}{\sqrt{N_r \eta}} \ . \tag{3.166}$$

3.F Error bound theorems

Proof of Lemma4: Let us identify $\hat{f}$ with $f(\xi)$, the vector under disturbance ξ. Accordingly, $f(0) = f$. Assume that the derivative $\frac{\partial f(\xi)}{\partial \xi}$ exists and calculate the derivative of Eq. (3.73) with respect to ξ:

$$(A + \xi R)\frac{\partial f(\xi)}{\partial \xi} + Rf(\xi) = r \tag{3.167}$$

At $\xi = 0$, we have:

$$\frac{\partial f(\xi)}{\partial \xi}\Big|_{\xi=0} = A^{-1}(r - Rf(0)) \ . \tag{3.168}$$

The Taylor expansion of $f(\xi)$ around $\xi = 0$ reads

$$f(\xi) = f(0) + \xi \frac{\partial f(\xi)}{\partial \xi}\Big|_{\xi=0} + \mathcal{O}(\xi^2) \tag{3.169}$$

$$= f(0) + \xi A^{-1}(r - Rf(0)) + \mathcal{O}(\xi^2) \tag{3.170}$$

Therefore, we can estimate

$$\frac{\|f(\xi)-f(0)\|}{\|f(0)\|} \leq \xi \frac{\|A^{-1}(r-Rf(0))\|}{\|f(0)\|} + \mathcal{O}(\xi^2) \tag{3.171}$$

$$\leq \xi \frac{\|A^{-1}\|\|r-Rf(0)\|}{\|f(0)\|} + \mathcal{O}(\xi^2) \tag{3.172}$$

$$\leq \xi \|A^{-1}\| \frac{\|r\|+\|R\|\|f(0)\|}{\|f(0)\|} + \mathcal{O}(\xi^2) \tag{3.173}$$

$$\leq \xi \|A^{-1}\| \left(\frac{\|r\|}{\|f(0)\|} + \|R\| \right) + \mathcal{O}(\xi^2) \tag{3.174}$$

$$\leq \xi \|A^{-1}\|\|A\| \left(\frac{\|r\|}{\|A\|\|f(0)\|} + \frac{\|R\|}{\|A\|} \right) + \mathcal{O}(\xi^2) \tag{3.175}$$

$$\leq \xi \|A^{-1}\|\|A\| \left(\frac{\|r\|}{\|Af(0)\|} + \frac{\|R\|}{\|A\|} \right) + \mathcal{O}(\xi^2) \tag{3.176}$$

$$\leq \xi \kappa(A) \left(\frac{\|r\|}{\|C\|} + \frac{\|R\|}{\|A\|} \right) + \mathcal{O}(\xi^2)\ . \tag{3.177}$$

□

Proof of Lemma 2: The gradient $\nabla_{\boldsymbol{\theta}}\phi(\boldsymbol{\theta}(\tau))$ evaluates in the chosen circuit as

$$\frac{\partial\phi(\theta(\tau))}{\partial\theta_k} = \sum_{j=1}^{N_d} \left(f_{k,j} R_{k,j} |0\rangle^{\otimes n} \langle\phi(\theta(\tau))| + |\phi(\theta(\tau))\rangle \langle 0|^{\otimes n} R_{k,j}^\dagger f_{k,j}^\dagger \right)\ ,$$

where we used Eq. (3.26). Then the trace norm of the dot product with the parameter vector $\boldsymbol{\theta}^*(T)$ evaluates as

$$\|\nabla_{\boldsymbol{\theta}}\phi(\boldsymbol{\theta_0}(T)) \cdot \boldsymbol{\theta}^*(T)\|_1 \tag{3.178}$$

$$= \left\| \sum_{k=1}^{N_V} \left[\sum_{j=1}^{N_d} \left(f_{k,j} R_{k,j} |0\rangle^{\otimes n} \langle\phi(\boldsymbol{\theta_0}(T))| + |\phi(\boldsymbol{\theta_0}(T))\rangle \langle 0|^{\otimes n} R_{k,j}^\dagger f_{k,j}^\dagger \right) \right] \theta_k^*(T) \right\|_1$$

$$\leq \sum_{k=1}^{N_V} \left\| \left[\sum_{j=1}^{N_d} \left(f_{k,j} R_{k,j} |0\rangle^{\otimes n} \langle\phi(\boldsymbol{\theta_0}(T))| + |\phi(\boldsymbol{\theta_0}(T))\rangle \langle 0|^{\otimes n} R_{k,j}^\dagger f_{k,j}^\dagger \right) \right] \theta_k^*(T) \right\|_1$$

$$\leq \sum_{k=1}^{N_V} \left[\sum_{j=1}^{N_d} \left(|f_{k,j}| \left\| R_{k,j} |0\rangle^{\otimes n} \langle\phi(\boldsymbol{\theta_0}(T))| \right\|_1 \right.\right.$$

$$\left.\left. + \left\| |\phi(\boldsymbol{\theta_0}(T))\rangle \langle 0|^{\otimes n} R_{k,j}^\dagger \right\|_1 |f_{k,j}^\dagger| \right) \right] |\theta_k^*(T)|$$

$$\leq \sum_{k=1}^{N_V} \left[\sum_{j=1}^{N_d} \left(|f_{k,j}| + |f_{k,j}^{\dagger}| \right) \right] |\theta_k^*(T)| = \sum_{k=1}^{N_V} \left[\sum_{j=1}^{N_d} 2|f_{k,j}| \right] |\theta_k^*(T)| \ ,$$

where in the third line, we used the fact that any quantum state has trace norm at most one. □