# Quantum methods for machine learning and classical dynamics

Barthe, A.M.

# Part II.

# Simulating classical and quantum systems on bosonic systems

# CHAPTER 6

# Continuous Variables Quantum Algorithm for solving Ordinary Differential Equations

## 6.1. Introduction

Differential equations are fundamental to understanding the dynamic behavior of various natural phenomena and technological processes, making them indispensable in a wide range of scientific, engineering, and mathematical disciplines. In the community, the typical problem one studies is the Initial Value Problem (IVP).

**Problem 6.1**
*Given an initial condition* $u_0 \in \mathbb{R}^N$*, an analytic function* $\boldsymbol{F} : \mathbb{R}^N \to \mathbb{R}^N$*, and an integration time* $t \in \mathbb{R}_{\geq 0}$ *find the solution* $u \colon \mathbb{R}_{\geq 0} \to \mathbb{R}^N$ *to the differential equation* $\frac{du}{dt} = \boldsymbol{F}(u)$*.*

Many works have tackled this problem, with linearization and with space and time discretization techniques, finding an approximation to the IVP of differential equations can be turned into solving a high-dimensional linear system of equations of the form $Ax = b$. This is the ideal setting

The contents of this chapter have been published in [38].

for the application of the HHL algorithm [171] which in some situations promises to solve with an exponential advantage in the size of the system, returning the solution as an amplitude-encoded state $|\mathbf{u}\rangle = \frac{1}{\|u\|}\sum_{j=1}^{N} u_j |j\rangle$. Several works have proposed approaches that scale polylogarithmically in the size of the problem $N$ for linear ODEs [42] and non-linear ODEs [172]. However, it is known that the HHL algorithm is subject to certain limitations [171, 173]: the state preparation of $|b\rangle$, the readout of the results from the amplitude-encoded state $|\mathbf{u}\rangle$, the sparsity of $A$, and the condition number of $A$. In particular, related to that last caveat, a lower bound of the complexity has been proven in [174] that applies to any quantum algorithm returning the solution of an ODE as an amplitude-encoded state, which is the case for the vast majority of exact quantum ODE solvers. Their complexity is bounded by $\Omega(e^{\delta t})$ where $t$ is the integration time of the IVP and $\delta$ is a parameter derived from the spectrum of the matrix $M$ characterizing the linear ODE $\dot{x} = Mx$.

In this chapter we propose a different approach to solve ODEs, and explore to which extent the above limitations could be circumvented. We study the translation of arbitrary dynamics into a Schrodinger-like equation [175]. Mapping the native space of the ODE to a space where the time evolution is unitary effectively reduces an arbitrary ODE problem to a Hamiltonian time evolution. While previous works approximate the infinite-dimensional Hilbert space with a finite-dimensional Hilbert space, in this chapter, we will explore an alternative way of dealing with the infinite dimensionality of the Hilbert space, working directly in a continuous variable framework.



The rest of the chapter is structured as follows: in Section 6.2 we cover background information such as the Koopman–von Neumann (KvN) framework and how previous works have used this formalism in quantum algorithms for ODEs. We also introduce continuous variable computing. In Section 6.3, we will present our main contribution, a continuous variable algorithm making use of the KvN framework, and explain how it is better suited to solve the initial distribution problem rather than the initial value problem. In Section 6.4 we discuss the limitations of the proposed algorithm and the next steps. Finally, we present our conclusions in Section 6.5.

## 6.2. Background

### 6.2.1. Koopman–von Neumann classical mechanics

The Koopman–von Neumann classical mechanics describe the evolution of arbitrary classical dynamical systems as the Hamiltonian evolution of wave functions in an infinite-dimensional Hilbert space, where each mode corresponds to one of the $N$ coordinates of the ODE. While the full theory behind this framework is more comprehensive, we present the basic elements of the Koopman–von Neumann (KvN) framework [176]. The central element of this framework is the Hamiltonian describing the time evolution. For an ODE defined as $\frac{du}{dt} = \mathbf{F}(u)$, we define the KvN Hamiltonian as

$$H_{\mathbf{F}} \coloneqq \frac{1}{2}\left(\hat{p}\mathbf{F}(\hat{q}) + \mathbf{F}(\hat{q})\hat{p}\right) \tag{6.1}$$

where $\hat{q} = [\hat{q}_1, \hat{q}_2, ..., \hat{q}_N]$ is the vector of the position operators for each of the $N$ modes, and $\hat{p}$ is the vector of their respective momentum operators. We also use the notation $|x\rangle_{\hat{q}}$ for a position operator eigenstate with eigenvalue $x$. Looking at the position operator in the Heisenberg picture, we can write that the operator follows the same equation as the ODE $\dot{x} = \mathbf{F}(x)$ (full derivation in [177]):

$$\frac{d\hat{q}}{dt} = \frac{[\hat{q}, H]}{\mathbf{i}h} = \mathbf{F}(\hat{q}). \tag{6.2}$$

Therefore, the expectation value of the position operator follows the trajectory of the ODE. Applying a position eigenstate $|u_0\rangle_{\hat{q}}$ to Equation (6.2), we can show that KvN evolution of the position eigenstate with an eigenvalue equal to the initial condition will result in a position eigenstate with the corresponding eigenvalue following exactly the solution of the ODE

$$|u(t)\rangle_{\hat{q}} = e^{iH_{\mathbf{F}}t}\,|u_0\rangle_{\hat{q}}\,, \tag{6.3}$$

where $u(t)$ is the solution to the IVP (Problem 6.1). As explained in [178] the state of a system in an infinite-dimensional Hilbert space is described by a wave function that can be expressed in the $\hat{q}$ representation $|\psi\rangle = \int \psi(u)\,|u\rangle_{\hat{q}}\,du$. Thanks to the linearity of the time evolution, evolving a wave function described by $\psi : u \to \sqrt{p_0(u)}$ using the KvN Hamiltonian will correspond to evolving the initial distribution $p_0 \colon \mathbb{R}^n \to [0, 1]$, solving a slightly different problem to the IVP. We introduce the initial distribution problem (IDP) which is, as we will see, a better fit for the types of computations we can expect CV quantum computers to perform. It is

also a natural extension of the IVP when there is uncertainty in the initial state.

**Problem 6.2**
*Given an initial probability distribution* $p_0 \colon \mathbb{R}^N \to [0,1]$*, an analytic function* $\boldsymbol{F} : \mathbb{R}^N \to \mathbb{R}^N$*, and an integration time* $t \in \mathbb{R}_{\geq 0}$ *find the probability distribution* $p_t \colon \mathbb{R}^N \to [0,1]$ *evolved for a time* $t$ *according to the differential equation* $\frac{du}{dt} = \boldsymbol{F}(u)$*.*

### 6.2.2. Previous work: Finite Hilbert space approximations

The use of the KvN framework in the context of solving the IVP has rapidly been identified as an opportunity for quantum computing [176, 179, 180]. To the best of our knowledge, all previous approaches reduce the infinite-dimensional Hilbert space to a finite dimension to perform an approximate simulation on a qubit-based quantum computer. This is done either with a truncation of the infinite-dimensional Hilbert space or via discretization of the phase space. However, the reduction to a finite Hilbert space is far from trivial [181], as finding rigorous bounds for the truncation error is still a work in progress. We explore an alternative way by working with infinite dimensional systems, which then avoids truncation problems, but does so at a cost. This introduces another class of issues, such as the appearance of the norm of unbounded operators in the Trotter error. Specifically, we propose an algorithm directly compiled as a sequence of continuous variables gates, meant to be executed on a bosonic quantum computer (e.g. photonics).



## 6.3. Proposed Algorithm

### 6.3.1. Approximation of the time evolution of the KvN Hamiltonian

Using the generators corresponding to the set of gates defined in Section 2.4.1, in this section we propose an algorithm to derive a sequence of gates approximating the time evolution of the Koopman–von Neumann Hamiltonian for a one-dimensional polynomial ODE. Any polynomial function describing the ODE can be written as $\mathbf{F}(u) = \sum_{k=0}^{d} a_k u^k$, and the corresponding KvN Hamiltonian is $H = \sum_k^d a_k \frac{1}{2}\{\hat{p}, \hat{q}^k\}$, where $\{A, B\} = (AB + BA)$ is the anti-commutator. Because of the way this

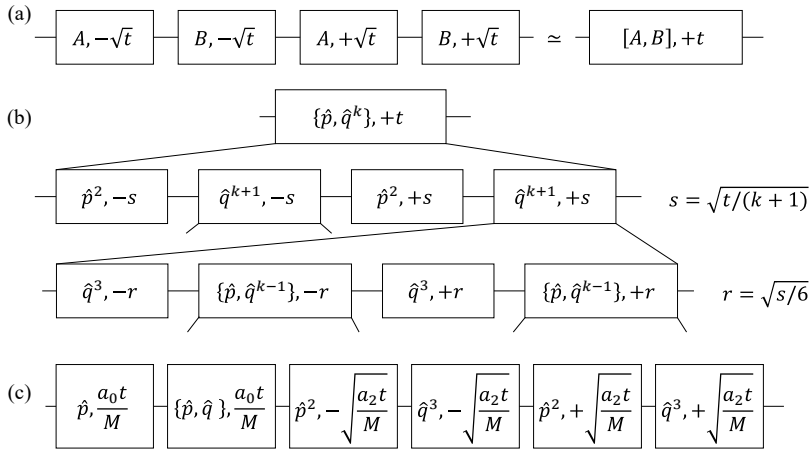


Figure 6.1.: (a) Group commutation relationship used to approximate a gate, (b) two-step nested structure to expand any monomial gate as per Algorithm 2. This leads to a recursive decomposition that is stopped when $\hat{q}^{k+1} = \hat{q}^3$ in which case the native gate is used. (c) this sequence repeated $M$ times is an example of a simple Trotter scheme for a quadratic 1D ODE

Hamiltonian naturally decomposes as a sum of Hamiltonian that are close to the native set of gates, we choose a Trotterisation [182] approach to evolve this Hamiltonian. We break down the algorithm into two parts, each of which is detailed in the Appendix in Section 6.6.1. Firstly, we introduce the monomial subroutine (Algorithm 2) to approximate any gate with a generator of the form $\{\hat{p}, \hat{q}^k\} \forall k \in \mathbb{N}$. We use the group commutation relationship depicted in Figure 6.1-(a) in a nested recursive expansion as illustrated in Figure 6.1-(b). Secondly, we use the monomial subroutine to be able to approximate any polynomial as detailed in Algorithm 3, and in particular the polynomial of the Hamiltonian of interest $H$.

Computing the error for such an approximation is a challenge, as it would require a complete Trotter error theory [182] for continuous variables which, to the best of our knowledge, has not yet been resolved. Naive extensions from finite systems run into trouble as the operators become unbounded.

Indeed, the general Trotter error is expressed as $O(\|[A,B]\|t^2)$, and as in infinite-dimensional Hilbert spaces operator norms are not bounded, this would, in theory, yield an infinite error. We present in the Appendix a potential direction to deal with this.

As an example, we present a simple scheme for an arbitrary 1D quadratic ODE in Figure 6.1-(c). This sequence of gates is derived by applying Algorithm 3 to a generic quadratic polynomial $\mathbf{F}(u) = a_0 + a_1 u + a_2 u^2$. The associated error comes in two different scalings. On the one hand, the first two blocks correspond to the linear part of the ODE ($a_0 + a_1 u$), their time parameter evolving linearly with time. Therefore general knowledge of Trotter theory yields an error $O((t/M)^2)$. On the other hand, the four last blocks correspond to the quadratic part of the ODE ($a_2 u^2$), and applying the Baker–Campbell–Hausdorff formula to Figure 6.1-(a) the error scales as $O((t/M)^{3/2+1})$. Comparing these two error scalings we realize that one is smaller than the other when ideally they should be of the same order. This illustrates that the above scheme could benefit from a better sequence of numbers of trotter steps $M$ (having fewer linear blocks than quadratic blocks for example). Finally, for the overall repetition $M$ times, the error $\varepsilon$ will be $O((t/M)^{5/2})$ and the number of cubic gates required for the quadratic scheme is $2M = O(\varepsilon^{2/5} t)$ with a potentially large prefactor due to the commutator norms, to be determined in future work.

### 6.3.2. Chaos and position eigenstates

The theoretical framework presented in Section 6.2.1 uses position eigenstates. However, these states are not physical states, as the amplitude of the wave function expressed in the Fock space is not square integrable. They can be interpreted as infinitely squeezed states, therefore we decided to approximate position eigenstates with finitely squeezed states. This approximation is equivalent to approximating the Dirac delta with a Gaussian where the standard deviation is controllable. A state squeezed by an amount $r$, (meaning that the squeezing generator is time evolved for a time $r$ starting from a coherent state), yields a standard deviation of $\sigma_0 = e^{-r}$. Then displacing this state to the position $x_0$ can be written as

$$|x_0, \sigma_0\rangle = \frac{1}{\sigma_0\sqrt{\pi}} \int_{-\infty}^{+\infty} \exp\left(-\frac{(x-x_0)^2}{\sigma_0^2}\right) |x\rangle_{\hat{q}}\, dx. \tag{6.4}$$

Using the position eigenstates basis for decomposition, we realize that this is a continuous weighted sum over position eigenstates and because the time evolution is linear, the time evolution of the sum of position eigenstates

is equal to the sum of the time evolution of position eigenstates. This means that effectively the resulting state will correspond to the evolution of a Gaussian distribution as the initial distribution under the differential equation.

Therefore in order to solve the IVP, we need to find a bound for the variance of the solution. Using knowledge of an upper bound of the Lyapunov exponent $\lambda$, with a required accuracy $\varepsilon$ and an integration time $t$, we can use it to choose the initial squeezing factor of the state. Indeed per the definition of the Lyapunov exponent, the standard deviation of the final state would be upper bounded: $\sigma_f < \sigma_0 e^{\lambda t}$ therefore in order to have the requested accuracy $\varepsilon$ we need to have $\varepsilon < e^{-r} e^{\lambda t}$, and we can conclude we need the squeezing factor to be

$$r > \lambda t + \log(\varepsilon). \tag{6.5}$$

It is important to note that this is not an unbiased estimator. For non-linear ODEs it is not guaranteed that the mean of the evolution of a Gaussian distribution will evolve accordingly to the solution of the ODE with a Dirac delta as an initial condition. Our suggestion to remedy that is to sufficiently squeeze the state initially.

However, we propose to use the algorithm to solve the IDP instead of the IVP, focusing on the evolution of the distribution itself rather than a single initial condition. Executing the algorithm on a continuous variables quantum computer becomes a machine to sample from that evolved distribution. This could for example help identify attractors or other properties of the dynamical system. Such properties include for example the Lyapunov exponent, which could be derived as the variance of the position operator $\hat{q}$.



### 6.3.3. Overall algorithm

We present a description of the algorithm for a 1D polynomial ODE in Algorithm 1, and explain it here. Starting with the preparation of the state, the vacuum state is first squeezed according to the required precision. Each mode is then displaced by an amount corresponding to the initial value. Secondly, we implement the approximated version of the time evolution of the KvN Hamiltonian, by executing the sequence of gates returned as in Section 6.3.1. Finally, we perform a homodyne measurement of the position and return the sampled value.

**Algorithm 1** Solve 1D-Poly-ODE

**Inputs** :

- $\mathbf{F}(u) = \sum_{k=0}^{d} a_k u^k$ the polynomial function characterising the ODE
- $u_0 = u(0) \in \mathbb{R}$ the initial condition
- $t$ the requested time interval
- An $M$ sequence (designed to reach an accuracy $\varepsilon$)
- $\lambda$ an upper bound on the Lyapunov exponent

**Outputs** : $\tilde{u}(t)$ an approximation of the ODE solution $u(t)$ such that $\|\tilde{u}(t) - u(t)\| < \varepsilon$

**Execute**

start from the vacuum state $|0\rangle$

$\{\hat{p}, \hat{q}\}$ , $\lambda t + \log(\varepsilon)$

$\hat{p}$, $u_0$ ▷ Initial state preparation

apply the sequence G from Algorithm 3 $\mathbf{Poly}(d, \{a_k\}_{k\in[0,d]}, t)$ ▷ Time evolution

sample and return $\langle \psi \mid \hat{q} \mid \psi \rangle$ ▷ Homodyne measurement

## 6.4. Discussion and next steps

So far we have looked into a single-mode algorithm, but adding beam-splitters guarantees the full universality of CV for several modes. The explicit procedure to extend the algorithm proposed in Section 6.3 from one-dimensional ODEs to multi-dimensional ODEs is out of scope of this chapter but will be considered in future work. It is however worth noting that beam-splitters can be considered a cheap gate in photonics.

As discussed in Section 6.3.1 and in the Appendix, more work is required to fully characterize the Trotter error in infinite-dimensional Hilbert spaces. Such analysis will enable choosing the appropriate number of Trotter steps $M$ at each commutator breakdown (Figure 6.1-(a)). We highlight the difference with the truncated approaches that limit support of the state which shall be a finite number of Fock states, while the Trotter approach limits the expected energy of the state but enables infinite-dimensional support. For this reason, we expect the final error scaling to be different between our approach and the previously introduced methods relying on Hilbert space truncation.

In addition, the difficult choice of the number of Trotter steps at each commutator breakdown hints towards a potential avenue performing a vari-

ational version of this algorithm, with sequences of displacement, squeezing, quadratic and cubic gates with tuneable execution time parameters. These parameters would be optimized to maximize the match between the output of the algorithm and given time-series data.

Finally, with fully characterized tighter Trotter error bounds, the main goal is to try and compare the complexity of such an algorithm to that of qubit-based algorithms. For example the Stellar rank [183] corresponds to the number of zeros of the Husimi function (a phase space representation of a state) that characterizes the "non-Gaussianity" of a state. The stellar rank is equivalent to the minimal number of photon additions necessary to engineer the state. This enables the introduction of an infinite hierarchy of states, that can be used in the context of the study of complexity of continuous variables quantum computing.

## 6.5. Conclusion

We propose a new perspective on the Koopman–von Neumann formalism, that allows mapping arbitrary classical dynamics to an infinite-dimensional Hilbert space where the dynamics are unitary. This effectively transforms an ODE problem into a Hamiltonian evolution problem. While previous works consider these dynamics in an approximated finite Hilbert space, we chose to keep working in an infinite-dimensional Hilbert space and propose a Continuous Variable Algorithm to solve one-dimensional polynomial ODEs. We propose that for such a CV system, the natural problem to be tackled is not the basic IVP, but rather its natural probabilistic extension the IDP. The current work does not allow for a full complexity analysis due to the problems with the estimation of the Trotter error, however, we do provide the algorithm with a sub-optimal sequence solving an arbitrary quadratic ODE, including the scaling of the number of gates.



## 6.6. Appendix

### 6.6.1. Algorithm subroutines

For the first subroutine, we will use the following commutation formulas extensively:

$$\left[\hat{p}^2, \hat{q}^{k+1}\right] = (k+1)i\{\hat{p}, \hat{q}^k\} \tag{6.6}$$

$$\left[\{\hat{p}, \hat{q}^k - 1\}, \hat{q}^3\right] = 6iq^{k+1} \tag{6.7}$$

The first subroutine's goal is to approximate a monomial of quadrature commutators of arbitrary degree $k$ : $\{\hat{p}, \hat{q}^k\}$. The degree $k = 0$ simply corresponds to the generator of the position displacement gate $H = \{\hat{p}, \hat{q}^0\} = 2\hat{p}$. The degree $k = 1$ corresponds to the generator of the squeezing gate $H = \{\hat{p}, \hat{q}^1\} = 2(\hat{p}\hat{q} + \hat{q}\hat{p})$. For higher degrees, generators of the form $\{\hat{p}, \hat{q}^k\}$ are not part of our native set of gates as described in Section 2.4.1. From Equation (6.6) we can obtain it as the commutator between $\hat{p}^2$ and $\hat{q}^{k+1}$ using the following approximation derived from second-order Trotter $e^{-ihA}e^{-ihB}e^{+ihA}e^{+ihB} = e^{-h^2[A,B]+O(h^3)}$. However, for $k \leq 3$ $\hat{q}^{k+1}$ is out of our set of gates. Looking at Equation (6.7) we realise that $\hat{q}^{k+1}$ is proportional to the commutator between $\{\hat{p}, \hat{q}^k - 1\}$ and $\hat{q}^3$. These principles are illustrated in Figure 6.1-(b), and we present the recursive Algorithm 2 to generate a sequence of gates that approximate a monomial of arbitrary degree. Sequences of gates are presented as a sequence of generators evolved for a certain time, using the following operators: the $+$ operation concatenates lists, the $*$ operation with an integer $M$ repeats $M$ times the sequence, $-$ is a sequence realized backward with opposite times (e.g. $-[\ \{A, t\}, \{B, -s\}] = [\ \{B, s\}, \{A, -t\}]$. Using this Algorithm 2 we can generate the gate sequence for the time evolution of the full KvN Hamiltonian $H = \sum_k^d a_k\{\hat{p}, \hat{q}^k\}$ Trotterising each of the monomials, as detailed in the Algorithm 3.

### 6.6.2. Trotter error and unbounded operators

The general Trotter error is expressed as $O(\|[A, B]\|t^2)$, which includes the spectral norm of an unbounded operator. However, on physical computers the energy of states is limited, therefore we present a potential way forward introducing the energy-constrained circle norm (ECCN), inspired by the energy-constrained diamond norm that is extensively used in the study of continuous variable channel capacities [184]. We define the energy-constrained circle norm for an operator $M$, and using $\hat{N}$ for the number operator as:

$$\|M\|_N := \max_{|\psi\rangle, \langle\psi|\hat{N}|\psi\rangle \leq N} \frac{\langle\psi \mid M \mid \psi\rangle}{\langle\psi \mid \psi\rangle} \tag{6.8}$$

For future work, we would on the one hand characterize the ECCN for all gates in our model of computation as seen in Section 2.4.1 as a function of the evolution time and the energy constraint, and on the other hand estimate the evolution of expectation value of the number of photons using the Heisenberg picture $[H, \hat{N}]$. With this combined information we expect it should be possible to find some bounds. This represents an extensive

**Algorithm 2 Mono**: Recursive Gate sequence to approximate the evolution of $\{\hat{p}, \hat{q}^k\}$

**Hyper-parameters** :

- $M$ a sequence of integers for the number of Trotter steps for each order.

**Inputs** :

- $k \geq 2$ the target degree of the monomial
- $t$ the requested time interval

**Outputs** :

- $G = [\{H_i, t_i\}]_{i \in [0,L]}$ a sequence of gates (defined as a generator for a certain time)

**Execute** :
**if** $k = 2$ **then** ▷ end of recursive
$s \coloneqq \frac{1}{M(k,1)} \sqrt{\frac{t}{3}}$
$G \coloneqq [\{\hat{p}^2, -s\}, \{\hat{q}^3, -s\}], \{\hat{p}^2, +s\}], \{\hat{q}^3, +s\}]] * M(k, 1)$
**else** ▷ intermediate levels
$r \coloneqq \frac{1}{M(k,1)} \sqrt{\frac{s}{6}}$, to get $\hat{q}^{k+1}$ for $s$ using Equation (6.7):
$F \coloneqq \big(\mathbf{Mono}(k-1, -r) + \big[\hat{q}^3, -r\big] + \mathbf{Mono}(k-1, +r) + \big[\hat{q}^3, +r\big]\big) *$
$M(k, 1)$ ▷ recursive
$s \coloneqq \frac{1}{M(k,2)} \sqrt{\frac{t}{k+1}}$, to get $\{\hat{p}, \hat{q}^k\}$ for $t$ using Equation (6.6):
$G \coloneqq \big(F + \big[p^2, -s\big] - F + \big[p^2, +s\big]\big) * M(k, 2)$
**end if**
**Return** $G$

**Algorithm 3 Poly**: approximate time evolution of KvN Hamiltonian

**Hyper-parameters** :

- $M$ a sequence of integers for the number of Trotter steps for each order.

**Inputs** :

- $d$ the degree of the polynomial
- $\{a_k\}_{k\in[0,d]}$ the coefficient of the polynomial
- $t$ the requested time interval

**Outputs** :

- $G = [\{H_i, t_i\}]_{i\in[0,L]}$ a sequence of gates

**Execute** :
$G \coloneqq \left[\{\hat{p}, \frac{a_0 t}{M_r M(0)}\}\right] * M(0,0)$
$G \coloneqq G + \left[\{\hat{p}\hat{q} + \hat{q}\hat{p}, \frac{a_1 t}{M_r M(1)}\}\right] * M(1,0)$
**for** $k = 2$ to $d$ **do**
    $G \coloneqq G + \textbf{Mono}\left(k, \frac{a_k t}{M_r M(k,0)}\right) * M(k,0)$ ▷ using Algorithm 2
**end for**
$G \coloneqq G * M_r$
**Return** $G$

piece of work and is left for later analysis, this chapter's scope being limited to outlining an idea. Such an analysis will enable the optimization of the number of Trotter steps for each breakdown into commutator in the nested structure, tagged $M$ in the Algorithm 2.