Universität
Augsburg
University

Classical networks for the Hubbard model with a tilted potential

Master Thesis

submitted by

Jonas Kell

on 13th February 2025

University of Augsburg
Faculty of Mathematics, Natural Sciences, and Materials Engineering
Institute of Physics
Chair for theoretical Physics III

1st Corrector: Prof. Dr. Markus Heyl
2nd Corrector: Prof. Dr. Thilo Kopp

Table of Contents

List of Abbreviations

Abbreviation	Meaning
TP III	Chair for theoretical Physics III
AI	Artificial Intelligence
RBM	Restricted Boltzmann Machine
GPT	Generative Pre-Trained Transformer
MC	Monte Carlo
VCN	Variational Classical Network
TDVP	Time-dependent Variational Principle
CPU	Central Processing Unit
HPC	High Performance Computing
LiCCA	Linux Compute Cluster Augsburg

1 Introduction

When I picked a subject for my master thesis and started working on submissions related to it, I could not have forseen the developments that would happen in the field of *Artificial Intelligence* (AI) during the time until I would have to submit it.

While working on my *bachelor thesis* [1] in late 2022, AI was in the middle of having major breakthroughs in image-processing and similar specialized tasks. A sudden spike in performance of such models could, among other factors, be attributed to the *transformer* architecture, I also happened to research in said bachelor thesis.

Just weeks after this, AI hit the mainstream with the introduction of *Chat-GPT* [2] around the new Year of 2023. The access to this rapidly empowering and yet established technology had so far mainly been gated by the required technological experience and powerful hardware. That changed, when the chatbot Chat-GPT was made available for the masses. Various image-generation-services followed in the same time frame, causing an immediate availability of generative AI for basically all types of media.

From one moment to the next, everybody on the planet who had access to an internet browser, could harness this technology in a simple web interface, without any prior technological knowledge. This quickly lead to a global AI-arms-race [3].

So into this situation one has to integrate this research. While doing my bachelor, AI was used in purpose-built research and large-scale industrial-applications. By now, every outlet is attaching buzzwords like *AI* to every product and research.

My bachelor thesis was likely run through a plagiarism software, as it has been standard for probably 10-15 years – my master thesis could very well be first run through an *AI-detection software* (that is itself "AI"), even though such means of cheating are only *possible* since 24 months, while classical plagiarism is as old as written word. The fact that I develop my research in open-source repositories might even cause or at least increase the likelihood of such a tool flagging my work – chances are real, that this paragraph will be submitted into training data the moment I commit it. That in itself causing the next pupil generating their report with AI to draw upon these words unknowingly.

Then the news hit just some weeks ago: the *Nobel Prize in Physics, 2024* was awarded to *J. J. Hopfield* and *G. E. Hinton* "for foundational discoveries and inventions that enable machine learning with artificial neural networks" [4]. Ironically, this was met with a huge controversy [5] in most social-media outlets – why? – "AI is not physics".

It took the people – even the physicists themselves working in fields that do not directly touch AI – two years of exposure to main stream Chat-GPT, to forget that before the *generative-AI-phase*, machine-learning was a highly scientific tool to advance major research breakthroughs. This gets even more ironic, once one realizes that the original (most cited) publication of Hopfield *"Neural networks and physical systems with emergent collective computational abilities"* [6], attempted to unify neurobiological research with solid state physics.

One decade ago in 2014, the Nobel laureate released *"Whatever Happened to Solid State Physics?"* [7]. Like a caricature, this text mirrors this very question "Does X belong to field Y". The letter makes the early developments of condensed-matter-physics a subject of discussion and shows the voices of 1958, that ask whether solid state physics should even be a subfield of physics (as it was more material-science like, as opposed to e.g. laser-physics).

The early subfield of condensed matter was in the 1960s and 1970s closely tied to (neuro-) biology. This very connection also leading to coining of the term *neuron* in 1982 for the architectural design of the first *Hopfield Networks* and *Restricted Boltzmann Machines* (RBM), the precursors to modern machine-learning-architectures. And the very *computational dynamics* research that was criticized in 1979 to be "no neurobiology", but won the Swartz-Prize for the new field of computational neurobiology in 2012.

Looking back onto this history, I can't help but being proud having the opportunity to try working on my own piece of condensed-matter-research at a university with as much solid-state-expertise as the *University of Augsburg*.

2 Theory

Some parts of *chapter 2 Theory* were first written for a *practical training report* [8]: `/practical-training-latex` that was part of initial works on the thesis-subject. The calculations and text was supplemented and adapted for this final thesis. How the parts have been adapted, as well as the original author being the thesis author, can be verified in the version history of the repository [8].

In the first part of this thesis, the theoretical background of the researched *Hubbard model* is outlined and the computational problem of gaining information about the *time-evolution* is presented. Then existing perturbative approximation algorithms are adapted to the relevant geometry, to attempt to provide a method of calculating time-dependent behavior of the present quantum-mechanical system. Details to the exact scientific question are introduced, after the basics have been defined.

2.1 Physical Basics

Achieving the goal of calculating observables for a specific time-evolved Hamiltonian and system geometry is mathematically straight forward. However, the computational expense for solving the time-evolution for arbitrary quantum-mechanical problems – in relation to the system size – is too large for practical purposes.

Here, the goal is to bring down the necessary number of sampled states from $\mathcal{O}(2^n)$ to around the $\mathcal{O}(n)$-range (depending on the required precision) with the use of perturbative simplifications and Monte-Carlo sampling.

The time-evolution will be executed by employing an expansion in the Interaction Picture to a fixed degree, which results in a computational complexity of $\mathcal{O}(n^{2 \cdot \text{degree}})$ per evaluated state in the worst case. Depending on the geometry (number of nearest neighbors c), this can be reduced to $\mathcal{O}(\text{f}(c) \cdot n)$, with $\text{f}(c)$ being the number of neighbors reachable from any fixed starting site in at most as many nearest-neighbor-hops as the highest degree of approximation.

By placing a specific restriction on the initial state, all instances where an effective Hamiltonian needs to be evaluated in fact only require the evaluation of the difference of the effective Hamiltonian on two states that differ only in a fixed number of occupations. Through this, the act of choosing the next state in the Monte-Carlo-Markov chain can be brought down from $\mathcal{O}(\text{f}(c) \cdot n)$ to $\mathcal{O}(\text{f}(c))$ and similarly also the cost of evaluating the most expensive observable drops to $\mathcal{O}(\text{f}(c))$ (except the model-quality gauging energy and its fluctuation, that will have a cost $\mathcal{O}(\text{f}(c) \cdot n)$ for a single evaluation).

2.1.1 Different Particle-Types, their Operators and Mappings

To describe different particle-classes in the setting of many-body-physics on discrete lattices, the *ladder operator formalism* is a useful tool. The inherit properties of the respective particle are guaranteed to be upheld by the *(anti-) commutation relations*.
In some cases, transformations may be used to map problems formulated in one particle type onto equivalent formulations in a different particle type. For the examples a discrete lattice of arbitrary enumeration – which also takes care of potential additional spin-degrees – is assumed.

Bosons

$$\left[\hat{b}_l, \hat{b}_m\right] = 0$$
$$\left[\hat{b}_l^\dagger, \hat{b}_m^\dagger\right] = 0 \tag{1}$$
$$\left[\hat{b}_l, \hat{b}_m^\dagger\right] = \delta_{l,m}$$

The bosonic operators follow the commutation-relations described in Equation 1, with δ the *Kronecker-Delta* [9], which allows for occupation numbers of all natural numbers or 0 per site.

Fermions

For fermionic systems it is important, that their wave-function anti-symmetrizes. This results in the anti-commutation relations of Equation 2.

$$\left\{\hat{c}_l, \hat{c}_m\right\} = 0$$
$$\left\{\hat{c}_l^\dagger, \hat{c}_m^\dagger\right\} = 0 \tag{2}$$
$$\left\{\hat{c}_l, \hat{c}_m^\dagger\right\} = \delta_{l,m}$$

From Equation 2, one can derive that the operator combination $\hat{c}_l^\dagger\hat{c}_l$, the *number operator*, is *idempotent*:

$$\left(\hat{c}_l^\dagger\hat{c}_l\right)^n = \hat{c}_l^\dagger\hat{c}_l \qquad \forall n \in \mathbb{N} \tag{3}$$

This is a property that is necessary for performing a specific calculation later and makes sense, once one realizes, this restricts the fermionic occupations to the values 0 and 1.

Hard-Core Bosons

While we require the number operator for the investigated particle type to be idempotent, it would be convenient to work with an operator that commutes and doesn't anti-commute like the fermions. In this regard, the hard-core bosons behave like a typical boson, but with extra restrictions that limit the occupation to either 0 or 1, meaning $\hat{h}_l^\dagger\hat{h}_l^\dagger = \hat{h}_l\hat{h}_l = 0$ [10]. The required

properties are fulfilled, if the particles obey the commutation-relations from Equation 4 [11], which in turn makes it possible to derive the idempotence-relation in Equation 5.

$$\left[\hat{h}_l, \hat{h}_m\right] = 0$$
$$\left[\hat{h}_l^\dagger, \hat{h}_m^\dagger\right] = 0 \tag{4}$$
$$\left[\hat{h}_l, \hat{h}_m^\dagger\right] = \left(1 - 2 \cdot \hat{h}_m^\dagger \hat{h}_m\right) \cdot \delta_{l,m}$$

$$\left(\hat{h}_l^\dagger \hat{h}_l\right)^n = \hat{h}_l^\dagger \hat{h}_l \qquad \forall n \in \mathbb{N} \tag{5}$$

This additionally grants the helpful property $\hat{h}_l^\dagger \hat{h}_l + \hat{h}_l \hat{h}_l^\dagger = 1$.

Spins / Pauli-Matrices

Basic Hamiltonians, like the *Ising-Model* are often expressed in the *spin basis*. The operators that act on this basis are the *Pauli matrices*, which have the useful property of being a basis for complex matrices. Most *quantum computing* formalisms are expressed in the spin basis and heavily rely on the Pauli matrices [12]:

$$\hat{\sigma}^0 = \mathbb{1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad \hat{\sigma}^1 = \hat{\sigma}^x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$
$$\hat{\sigma}^2 = \hat{\sigma}^y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \qquad \hat{\sigma}^3 = \hat{\sigma}^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{6}$$

For spin-particles on a lattice, the operators need to obey the relations in Equation 7.

$$\left\{\hat{\sigma}_l^\alpha, \hat{\sigma}_l^\beta\right\} = 2 \cdot \delta_{\alpha,\beta}$$
$$\left[\hat{\sigma}_l^\alpha, \hat{\sigma}_m^\beta\right] = 2 \cdot i \cdot \varepsilon_{\alpha,\beta,\gamma} \cdot \delta_{l,m} \cdot \hat{\sigma}_l^\gamma \tag{7}$$

With the complex unit i and the *Levi-Civita-Symbol* $\varepsilon_{\alpha,\beta,\gamma}$.

Transformation: Express Spins as Fermions

For mapping between fermions and spins, there exists a well known transformation, the *Jordan-Wigner-Transformation* [13] (Equation 9).
The following transformations use Equation 8, where the equality follows from the fact that these operators are all different representations of the number operator.

$$(-1)^{\sum_{k=0}^{l-1}(\hat{\sigma}_k^z+1)/2} = (-1)^{\sum_{k=0}^{l-1}\hat{c}_k^\dagger\hat{c}_k} = (-1)^{\sum_{k=0}^{l-1}\hat{h}_k^\dagger\hat{h}_k} = P(l) \tag{8}$$

With the easily verifiable property $P(l) \cdot P(l) = 1$. Also $P(l)$ commutes with all $\hat{c}_m$, $\hat{c}_m^\dagger$, $\hat{h}_m$ and $\hat{h}_m^\dagger$ if $l \leq m$ and furthermore $[P(l), P(m)] = 0$, always.

$$\hat{c}_l^\dagger = P(l) \cdot \frac{\hat{\sigma}_l^x + i \cdot \hat{\sigma}_l^y}{2} = P(l) \cdot \hat{\sigma}_l^+$$

$$\hat{c}_l = P(l) \cdot \frac{\hat{\sigma}_l^x - i \cdot \hat{\sigma}_l^y}{2} = P(l) \cdot \hat{\sigma}_l^-$$

$$\hat{c}_l^\dagger \hat{c}_l = \left(\hat{\sigma}_l^z + 1\right)/2$$

$$\Longleftrightarrow \qquad\qquad\qquad\qquad (9)$$

$$\hat{\sigma}_l^x = \left(\hat{c}_l + \hat{c}_l^\dagger\right) \cdot P(l)$$

$$\hat{\sigma}_l^y = i \cdot \left(\hat{c}_l - \hat{c}_l^\dagger\right) \cdot P(l)$$

$$\hat{\sigma}_l^z = 2 \cdot \hat{c}_l^\dagger \hat{c}_l - 1$$

For performing commutation-relation calculations, a helper program was written in conjunction with this thesis and related submissions. The program is called *Math-Manipulator* [14] and the transformations in this chapter have been validated with it in a supplementary repository: [15]: `/jordan-wigner-transformation` .

Transformation: Express Fermions as Hard-Core Bosons

The tensor-network library *ITensor* provides in their documentation an example for a mapping between fermions and hard-core bosons, used to simplify their calculations [16]. The transformations may be called *fermionization/bosonization* or respectively also Jordan-Wigner-Transformation.
Equation 10 presents the transformation:

$$\hat{c}_l^\dagger = P(l) \cdot \hat{h}_l^\dagger$$

$$\hat{c}_l = P(l) \cdot \hat{h}_l$$

$$\Longleftrightarrow \qquad\qquad\qquad\qquad (10)$$

$$\hat{h}_l^\dagger = P(l) \cdot \hat{c}_l^\dagger$$

$$\hat{h}_l = P(l) \cdot \hat{c}_l$$

Transformation: Express Spins as Hard-Core Bosons

Finally this confirms the transformation that will be required later: expressing a measurement in the spin-basis in terms of hard-core bosonic operators.
Equation 11 can be directly obtained by plugging Equation 10 into Equation 9.

$$\hat{\sigma}_l^x = \hat{h}_l + \hat{h}_l^\dagger$$
$$\hat{\sigma}_l^y = i \cdot \left(\hat{h}_l - \hat{h}_l^\dagger \right) \tag{11}$$
$$\hat{\sigma}_l^z = 2 \cdot \hat{h}_l^\dagger \hat{h}_l - 1$$

This connection shows, that spin-$\frac{1}{2}$ systems and hard-core bosonic systems can be used to investigate the same problems in different notations. Equation 12 shows, which states are equivalent for these two notations.

$$|\downarrow\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |0\rangle$$
$$\hat{\sigma}^+ |\downarrow\rangle = |\uparrow\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |1\rangle = \hat{h}^\dagger |0\rangle \tag{12}$$

This exact mapping is important to conserve sign-consistency across all calculations. E.g. swapping the $|0\rangle = |\downarrow\rangle$ correspondence to $|0\rangle = |\uparrow\rangle$, would introduce a hidden minus sign in observables depending on $\hat{\sigma}^y$, because of the implicit order of basis states in the definition of $\hat{\sigma}^y$.

2.1.2 Hubbard Model and discussed Hamiltonian

The system in question is described by a *Hubbard model Hamiltonian*, that includes an electrical field. It is shown in Equation 13.

$$\mathcal{H} = \mathcal{H}_0 + \hat{V} \tag{13}$$

$$\mathcal{H}_0 = U \cdot \sum_l \hat{n}_{l,\uparrow} \hat{n}_{l,\downarrow} + \sum_{l,\sigma} \underbrace{\left(\vec{E} \cdot \vec{r}_l \right)}_{\varepsilon_l} \hat{n}_{l,\sigma} \tag{14}$$

$$\hat{V} = -J \cdot \sum_{\langle l,m \rangle, \sigma} \left(\hat{h}_{l,\sigma}^\dagger \hat{h}_{m,\sigma} + \hat{h}_{m,\sigma}^\dagger \hat{h}_{l,\sigma} \right) \tag{15}$$

$\vec{E}$ describes the vector of said electrical field and $\vec{r}_l$ the position of the site with index l. The number operators $\hat{n}_{l,\sigma} = \hat{h}_{l,\sigma}^\dagger \hat{h}_{l,\sigma}$ measure the occupation on site l with the respective spin σ. U, J and ε are constants that describe the interaction strength. They all have the unit of energy. The scalar product that is defined as ε can be evaluated, based on the system geometry. In this case, the system is described by a regular square pattern that can be seen in Figure 2.1. For such a system, the energy difference that is acquired from hopping from site l to m ($\Delta E_{l \to m}$) (provided, l and m are nearest neighbors) is described by Equation 16. By assuming a default

value for ε_0, one gets the relation from [Equation 17](#) for ε_l (using the telescoping sum over nearest-neighbors-hopping along the path from $\vec{r_0}$ to $\vec{r_l}$).
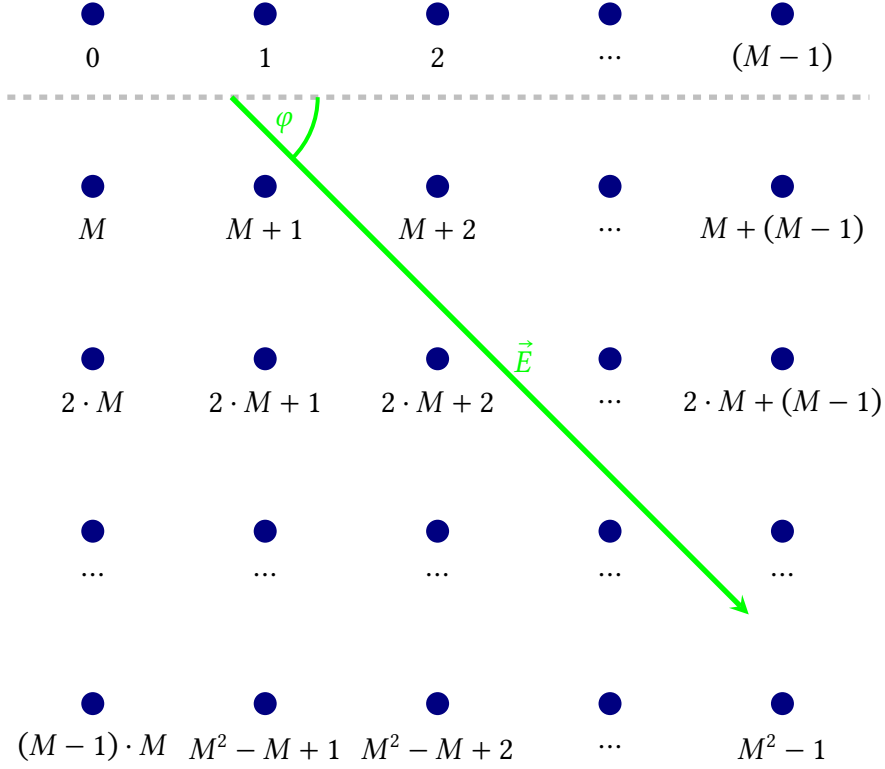

Figure 2.1: Graphical representation of how the examined square system is laid out. The sites are labeled with the index l, they later can be identified by. Each of the sites has 4 nearest neighbors, except the ones on the borders, that have 3 or only 2 nearest neighbors (as the system is not periodic). A system of side length M with size N is depicted, which means that it has $M^2 = N$ sites. In green, the Electrical field vector $\vec{E}$ is depicted, it is parametrized by the field strength E and the angle φ.

$$\Delta E_{l \to m} = \varepsilon_m - \varepsilon_l = \vec{E} \cdot \left(\vec{r_m} - \vec{r_l} \right)$$
$$= E \cdot \left[\cos(\varphi) \cdot (m\%M - l\%M) + \sin(\varphi) \cdot \left(\left\lfloor \frac{m}{M} \right\rfloor - \left\lfloor \frac{l}{M} \right\rfloor \right) \right] \tag{16}$$

$$\varepsilon_0 = 0 \Rightarrow \varepsilon_l = \varepsilon_l - \varepsilon_0 = \Delta E_{0 \to x} + \cdots + \Delta E_{y \to l}$$
$$= E \cdot \left[\cos(\varphi) \cdot l\%M + \sin(\varphi) \cdot \left\lfloor \frac{l}{M} \right\rfloor \right] \tag{17}$$

During measurements in the [Numerical Experiments](#) section, a one-dimensional version of this is used, that will be labeled *chain*. It will be obtained by cutting the first row of the respective square geometry and is consistent with basically all mathematical derivations performed for the *square* case, because of the nearest-neighbor notation that is independent of the underlying geometry. Only in the derivation of *[subsection 2.2.3 Spin-Current](#)* differences based on the

dimensionality appear, however they do not affect the end result. The reason for the truncation is to reduce thy system size for the verification of convergence towards exact comparisons. While such a modification necessarily affects the physical properties, it is fine for verification of convergence. Also interesting physical – system-size independent – effects only manifest in experiments with lattice site numbers far beyond any possible exact diagonalization anyway.

The Hubbard model [17] is a spin-dependent quantum-mechanical model. Because of this, the Hamiltonian includes terms for both the spin directions ↑ and ↓ (most of the time denoted by a summation over σ).

In principle, the operators $\hat{h}_{l,\sigma}^{(\dagger)}$ can be any type of quantum-mechanical ladder operator, like the aforementioned bosonic or fermionic ones. In this application the decision was made to use *hard-core bosons* [10] for they happen to obey multiple useful properties.

Typically for bosons, the Hubbard model is transformed into the *Bose-Hubbard model* [18]. However, hard-core bosons have the added property that they can only have the occupation numbers of 0 and 1 per-spin-and-site in the investigated energy range. This means they behave like fermions, only that their wave-function symmetrizes instead of anti-symmetrizes [19]. Because of the restriction, one term of the Bose-Hubbard model drops out and the studied model in equations 13, 14 and 15 is produced. Also, the typical Bose-Hubbard model is not spin-dependent, while this property is kept here. Hard-core bosons on their own have relevant properties that make them worth studying. Experimental research on hard-core bosons can give insight into e.g. *ultra-cold gasses* and *Bose-Einstein condensation* [10]. In this case it is important for the following derivations, that the operators on different sites commute, the number operator is idempotent (5) and can be mapped to spin-basis notation (11).

When adjusting the commutation-relations in Equation 4 for spin-dependency Equation 18 is obtained.

$$
\begin{aligned}
\left[\hat{h}_{l,\sigma}, \hat{h}_{l',\sigma'}\right] &= \left[\hat{h}_{l,\sigma}^{\dagger}, \hat{h}_{l',\sigma'}^{\dagger}\right] = 0 \\
\left[\hat{h}_{l,\sigma}, \hat{h}_{l',\sigma'}^{\dagger}\right] &= \left(1 - 2 \cdot \hat{h}_{l,\sigma}^{\dagger}\hat{h}_{l,\sigma}\right) \cdot \delta_{l,l'} \cdot \delta_{\sigma,\sigma'}
\end{aligned}
\tag{18}
$$

For ease of notation, the two spin degrees of freedom may be described by the alternate naming scheme provided in Equation 19. The related code implementation, comments and some calculations in linked material [20] use the letters c and d, as that used to be the first designation wile the thesis was developed.

$$
\hat{h}_{l,\uparrow}^{(\dagger)} \leftrightarrow \hat{h}_{l}^{(\dagger)} \qquad\qquad \hat{h}_{l,\downarrow}^{(\dagger)} \leftrightarrow \hat{d}_{l}^{(\dagger)}
\tag{19}
$$

For simplification purposes, the two spin directions are assumed to have the same coupling constants (U, J and ε_l). This means, all later results *must* be symmetrical in terms of the spin directions ↑ and ↓.

Translated into the alternative notation, Equation 14 reads like Equation 20 and Equation 15 reads like Equation 21.

$$\mathcal{H}_0 = U \cdot \sum_l \hat{h}_l^\dagger \hat{h}_l \hat{d}_l^\dagger \hat{d}_l + \sum_l \varepsilon_l \hat{h}_l^\dagger \hat{h}_l + \sum_l \varepsilon_l \hat{d}_l^\dagger \hat{d}_l \tag{20}$$

$$\hat{V} = -J \cdot \sum_{\langle l,m \rangle} \left(\hat{h}_l^\dagger \hat{h}_m + \hat{h}_m^\dagger \hat{h}_l + \hat{d}_l^\dagger \hat{d}_m + \hat{d}_m^\dagger \hat{d}_l \right) \tag{21}$$

Several of the previous equations feature the *nearest neighbor* summing notation with pointy brackets $\langle \rangle$. This is supposed to express, that in the case of $\sum_{\langle l,m \rangle}$ for each possible l, m will take on the indices of all neighbor sites of l. However, if a pair $l = x, m = y$ is part of the sum, then the inverse $l = y, m = x$ will not be. To be more in-line with the code implementation, Equation 22 gives an alternative notation using []. In such sums, each pair $l = x, m = y$ and $l = y, m = x$ will be included in the sum for each nearest-neighbor-relation x, y.

$$\sum_{\langle l,m \rangle} \left(F(l,m) \, \hat{h}_l^\dagger \hat{h}_m + F(m,l) \, \hat{h}_m^\dagger \hat{h}_l \right) \overset{\overset{\text{rename-swap}}{l \text{ and } m}}{=}$$
$$\sum_{\langle l,m \rangle} F(l,m) \, \hat{h}_l^\dagger \hat{h}_m + \sum_{\langle m,l \rangle} F(l,m) \, \hat{h}_l^\dagger \hat{h}_m = \tag{22}$$
$$\sum_{[l,m]} F(l,m) \, \hat{h}_l^\dagger \hat{h}_m \overset{\overset{! \text{ caution }!}{}}{\neq} 2 \cdot \sum_{\langle l,m \rangle} F(l,m) \, \hat{h}_l^\dagger \hat{h}_m$$

With that convention, it is possible to rewrite Equation 21 to Equation 23.

$$\hat{V} = -J \cdot \sum_{[l,m]} \left(\hat{h}_l^\dagger \hat{h}_m + \hat{d}_l^\dagger \hat{d}_m \right) \tag{23}$$

2.1.3 Interaction Picture and Time Evolution

Solving the quantum-mechanical many-body time-evolution problem in general requires at least a time-complexity of $\mathcal{O}(2^n)$. For some special cases it is possible to solve it analytically. E.g. for one-dimensional systems, the Hubbard model can be solved analytically even with the addition of an electrical field [21].

For arbitrary geometry in two or even more dimensions, this is no longer possible. In that case, approximations are required to lower the computational complexity to a reasonable level to obtain the required measurements. Here, the calculation is performed in the *Interaction Picture* (relevant equations from [19]), along the lines of the calculation performed in [22].

The equations provided so far show operators from the *Schrödinger Picture*. For this section, the operators and states will be labeled whether they are Schrödinger- or Interaction Picture. All un-labeled elements are assumed to be Schrödinger Picture.

A general state in the Schrödinger Picture can be expanded in a basis as shown in Equation 24. It is time-evolved using the full Hamiltonian in the Schrödinger Picture (Equation 25, all units so that $\hbar = 1$).

$$\left|\Psi^{\text{S}}\right\rangle = \left|\Psi^{\text{S}}(t=0)\right\rangle = \sum_{N} |N\rangle \underbrace{\langle N|\Psi^{\text{S}}\rangle}_{\Psi_N} = \sum_{N} \Psi_N |N\rangle \tag{24}$$

$$\left|\Psi^{\text{S}}(t)\right\rangle = e^{-i\mathcal{H}^{\text{S}}t}\left|\Psi^{\text{S}}(t=0)\right\rangle = e^{-i\mathcal{H}^{\text{S}}t}\left|\Psi^{\text{S}}\right\rangle \tag{25}$$

In the Interaction Picture, operators have a time evolution, that depends only on the $\mathcal{H}_0^{\text{S}}$-part, shown in Equation 26 (assuming the operators are time-independent in the Schrödinger Picture).

$$\hat{\text{A}}^{\text{I}}(t) = e^{i\mathcal{H}_0^{\text{S}}t}\hat{\text{A}}^{\text{S}}e^{-i\mathcal{H}_0^{\text{S}}t} \quad \Rightarrow \quad \mathcal{H}_0^{\text{S}} = \mathcal{H}_0^{\text{I}} = \mathcal{H}_0$$
$$\text{especially:} \quad \hat{\text{V}}^{\text{I}}(t) = e^{i\mathcal{H}_0^{\text{S}}t}\hat{\text{V}}^{\text{S}}e^{-i\mathcal{H}_0^{\text{S}}t} \tag{26}$$

The time-evolution of a general state in the Interaction Picture on the other hand is obtained as shown in Equation 27.

$$\left|\Psi^{\text{I}}(t)\right\rangle = e^{i\mathcal{H}_0 t}\left|\Psi^{\text{S}}(t)\right\rangle \overset{25}{=} e^{i\mathcal{H}_0 t}e^{-i\mathcal{H}^{\text{S}}t}\left|\Psi^{\text{S}}\right\rangle \quad \Rightarrow \quad \left|\Psi^{\text{S}}(t)\right\rangle = e^{-i\mathcal{H}_0 t}\left|\Psi^{\text{I}}(t)\right\rangle \tag{27}$$

Differentiating Equation 27 and substituting the special case in Equation 26 along the way, one gets Equation 28.

$$\begin{aligned}
\frac{\mathrm{d}\left|\Psi^{\text{I}}(t)\right\rangle}{\mathrm{d}t} &\overset{27}{=} \frac{\mathrm{d}}{\mathrm{d}t}\left(e^{i\mathcal{H}_0 t}e^{-i\mathcal{H}^{\text{S}}t}\right)\left|\Psi^{\text{S}}\right\rangle \\
&= -i\cdot e^{i\mathcal{H}_0 t}\left(\mathcal{H}^{\text{S}} - \mathcal{H}_0\right)e^{-i\mathcal{H}^{\text{S}}t}\left|\Psi^{\text{S}}\right\rangle \\
&\overset{13}{=} -i\cdot e^{i\mathcal{H}_0 t}\hat{\text{V}}^{\text{S}}\cdot\mathbb{1}\cdot e^{-i\mathcal{H}^{\text{S}}t}\left|\Psi^{\text{S}}\right\rangle \\
&= -i\cdot e^{i\mathcal{H}_0 t}\hat{\text{V}}^{\text{S}}e^{-i\mathcal{H}_0 t}e^{i\mathcal{H}_0 t}e^{-i\mathcal{H}^{\text{S}}t}\left|\Psi^{\text{S}}\right\rangle \\
&\overset{27}{=} -i\cdot e^{i\mathcal{H}_0 t}\hat{\text{V}}^{\text{S}}e^{-i\mathcal{H}_0 t}\left|\Psi^{\text{I}}(t)\right\rangle \\
&\overset{26}{=} -i\cdot \hat{\text{V}}^{\text{I}}(t)\left|\Psi^{\text{I}}(t)\right\rangle
\end{aligned} \tag{28}$$

The equation of motion in Equation 28 is solved by the ansatz in Equation 29 with the *time-ordering-operator* $\mathbb{T}$ and the *time-evolution-operator* $\hat{\text{P}}^{\text{I}}(t)$.

$$\begin{aligned}
\left|\Psi^{\text{I}}(t)\right\rangle &= \mathbb{T}\left\{e^{-i\int_0^t \mathrm{d}t'\hat{\text{V}}^{\text{I}}(t')}\right\}\left|\Psi^{\text{I}}(0)\right\rangle \\
&= \hat{\text{P}}^{\text{I}}(t)\left|\Psi^{\text{I}}(0)\right\rangle \overset{27}{=} \hat{\text{P}}^{\text{I}}(t)\left|\Psi^{\text{S}}(0)\right\rangle
\end{aligned} \tag{29}$$

Plugging this in, one finally obtains the general time-evolution for the state in the Schrödinger Picture in Equation 30. This requires the *base-energies* $E_0(N)$, where $e^{-i\mathcal{H}_0 t}|N\rangle = e^{-iE_0(N)t}|N\rangle$.

$$
\begin{aligned}
\left|\Psi^S(t)\right\rangle &\overset{27}{=} e^{-i\mathcal{H}_0 t}\left|\Psi^I(t)\right\rangle \overset{29}{=} e^{-i\mathcal{H}_0 t}\hat{P}^I(t)\left|\Psi^S\right\rangle \\
&= e^{-i\mathcal{H}_0 t}\cdot \mathbb{1}\cdot\hat{P}^I(t)\left|\Psi^S\right\rangle\cdot\mathbb{1} \\
&= e^{-i\mathcal{H}_0 t}\sum_N |N\rangle\left\langle N\middle|\hat{P}^I(t)\middle|\Psi^S\right\rangle\cdot\frac{\left\langle N\middle|\Psi^S\right\rangle}{\left\langle N\middle|\Psi^S\right\rangle} \\
&\overset{24}{=} \sum_N \underbrace{e^{-i\mathcal{H}_0 t}}_{e^{-iE_0 t}}\underbrace{\frac{\left\langle N\middle|\hat{P}^I(t)\middle|\Psi^S\right\rangle}{\left\langle N\middle|\Psi^S\right\rangle}}_{e^{\mathcal{H}_N(t)}}\Psi_N |N\rangle \\
&= \sum_N e^{-iE_0(N)t}e^{\mathcal{H}_N(t)}\Psi_N |N\rangle = \sum_N e^{\mathcal{H}_{\mathrm{eff}}(N,t)}\Psi_N |N\rangle
\end{aligned}
\tag{30}
$$

Up to this point, the calculation is still exact. Now the effective Hamiltonian $\mathcal{H}_N(t)$ could be approximated in different ways. Typically $e^{\mathcal{H}_N(t)}$ is *Taylor-expanded*, but in this context this approximation would converge quite poorly. To combat this and require a smaller order of expansion, one could employ the *cumulant-expansion* [23] to expand the object in terms of its exponent $\mathcal{H}_N(t)$ and not the whole exponential $e^{\mathcal{H}_N(t)}$. The first and second order cumulants for the cumulant expansion are provided in Equation 31 [22] with the *cumulant-average* in respect to the state $|N\rangle$, $\langle\cdot\rangle_{c(N)}$.

$$
\begin{aligned}
\left\langle\hat{A}\right\rangle_{c(N)} &= \frac{\left\langle N\middle|\hat{A}\middle|\Psi^S\right\rangle}{\left\langle N\middle|\Psi^S\right\rangle} \\
\left\langle\hat{A}\hat{B}\right\rangle_{c(N)} &= \frac{\left\langle N\middle|\hat{A}\hat{B}\middle|\Psi^S\right\rangle}{\left\langle N\middle|\Psi^S\right\rangle} - \frac{\left\langle N\middle|\hat{A}\middle|\Psi^S\right\rangle\cdot\left\langle N\middle|\hat{B}\middle|\Psi^S\right\rangle}{\left\langle N\middle|\Psi^S\right\rangle^2}
\end{aligned}
\tag{31}
$$

Plugging this into the rearranged result from Equation 30 following the derivation in [22] and [24], an expansion for $\mathcal{H}_N(t)$ in Equation 32 is obtained.

$$\langle N|\Psi^{\mathrm{S}}\rangle \cdot e^{\mathcal{H}_N(t)} \stackrel{30}{=} \langle N|\hat{\mathrm{P}}^{\mathrm{I}}(t)|\Psi^{\mathrm{S}}\rangle \stackrel{29}{=} \langle N|\mathbb{T}e^{-i\int_0^t \mathrm{d}t'\hat{V}^{\mathrm{I}}(t')}|\Psi^{\mathrm{S}}\rangle$$

$$\stackrel{[22]}{\Longleftrightarrow}$$

$$
\begin{aligned}
\mathcal{H}_N(t) &= \sum_{v=1}^{\infty} \frac{(-i)^v}{v!} \int_0^t \mathrm{d}t_1 \int_0^t \mathrm{d}t_2 \cdots \int_0^t \mathrm{d}t_v \left\langle \mathbb{T}\, \hat{V}^{\mathrm{I}}(t_1)\hat{V}^{\mathrm{I}}(t_2)\cdots \hat{V}^{\mathrm{I}}(t_v)\right\rangle_{\mathrm{c}(N)} \\
&= -i \int_0^t \mathrm{d}t_1 \frac{\langle N|\hat{V}^{\mathrm{I}}(t_1)|\Psi^{\mathrm{S}}\rangle}{\langle N|\Psi^{\mathrm{S}}\rangle} \\
&\quad - \frac{1}{2}\int_0^t \mathrm{d}t_1 \int_0^t \mathrm{d}t_2 \left(\frac{\langle N|\mathbb{T}\hat{V}^{\mathrm{I}}(t_1)\hat{V}^{\mathrm{I}}(t_2)|\Psi^{\mathrm{S}}\rangle}{\langle N|\Psi^{\mathrm{S}}\rangle} - \frac{\langle N|\hat{V}^{\mathrm{I}}(t_1)|\Psi^{\mathrm{S}}\rangle \cdot \langle N|\hat{V}^{\mathrm{I}}(t_2)|\Psi^{\mathrm{S}}\rangle}{\langle N|\Psi^{\mathrm{S}}\rangle^2}\right) \\
&\quad + \cdots
\end{aligned}
\tag{32}
$$

As $\hat{V}^{\mathrm{I}}(t') \propto J$, if J is a small perturbation (in comparison to U and ε) this quickly converges, even when taking only few terms into account.

2.1.4 Operators in the Interaction Picture

As Equation 32 reveals, by this process the problem was converted into finding $\hat{V}^{\mathrm{I}}(t')$ to be able to calculate the first two orders of the expansion.

Operators in the Interaction Picture can be derived from their Schrödinger Picture variants like previously shown in Equation 26. With this, one can derive a useful expression for their *equation of motion* (Equation 33).

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\hat{\mathrm{A}}^{\mathrm{I}}(t) &= i\left[\mathcal{H}_0^{\mathrm{S}}, \hat{\mathrm{A}}^{\mathrm{I}}(t)\right] \stackrel{26}{=} i\left[\mathcal{H}_0^{\mathrm{S}}, e^{i\mathcal{H}_0^{\mathrm{S}}t}\hat{\mathrm{A}}^{\mathrm{S}}e^{-i\mathcal{H}_0^{\mathrm{S}}t}\right] \\
&= ie^{i\mathcal{H}_0^{\mathrm{S}}t}\left[\mathcal{H}_0^{\mathrm{S}}, \hat{\mathrm{A}}^{\mathrm{S}}\right]e^{-i\mathcal{H}_0^{\mathrm{S}}t} = i\left\{\left[\mathcal{H}_0^{\mathrm{S}}, \hat{\mathrm{A}}^{\mathrm{S}}\right]\right\}^{\mathrm{I}}(t)
\end{aligned}
\tag{33}
$$

Equation 33 also defines the operator $\{*\}^{\mathrm{I}}(t)$, which simply is $e^{i\mathcal{H}_0^{\mathrm{S}}t}\{*\}e^{-i\mathcal{H}_0^{\mathrm{S}}t}$. This section uses (t) for indicating that operators in the Interaction Picture have a time-dependence. This is superficial and only for better readability, as in fact $\hat{\mathrm{h}}_m^{\dagger\mathrm{I}}(t) = \hat{\mathrm{h}}_m^{\dagger\mathrm{I}} = \left\{\hat{\mathrm{h}}_m^{\dagger\mathrm{S}}\right\}^{\mathrm{I}}(t)$.

If $\left[\mathcal{H}_0^{\mathrm{S}}, \hat{\mathrm{A}}^{\mathrm{S}}\right]$ now is a function of $\hat{\mathrm{A}}^{\mathrm{S}}$, one gets a *differential equation* that can be solved to obtain $\hat{\mathrm{A}}^{\mathrm{I}}(t)$.

For the repetitive calculation of the objects of type $\left[\mathcal{H}_0^{\mathrm{S}}, \hat{\mathrm{A}}^{\mathrm{S}}\right]$ for various $\hat{\mathrm{A}}^{\mathrm{S}}$, the tool *Math-Manipulator* was specifically developed. Like the transformations in 2.1.1, calculations in the following section have been validated or calculated from scratch in this tool. The relevant files can be found in [15]: `/calculate-V`, and [15]: `/time-evolution`. The equations 34 and 35 list exemplary results that were computed with the tool.

$$\frac{d}{dt}\left(\hat{d}_m^{\dagger I}\hat{d}_m^{I}\right)(t) = i\left\{\left[\mathcal{H}_0, \hat{d}_m^{\dagger S}\hat{d}_m^{S}\right]\right\}^{I}(t) \stackrel{MM}{=} 0 \quad \Rightarrow \quad \left(\hat{d}_m^{\dagger I}\hat{d}_m^{I}\right)(t) = \hat{d}_m^{\dagger S}\hat{d}_m^{S} \tag{34}$$

$$\begin{aligned}
\frac{d}{dt}\hat{h}_m^{\dagger I}(t) &= i\left\{\left[\mathcal{H}_0, \hat{h}_m^{\dagger S}\right]\right\}^{I}(t) \stackrel{MM}{=} i\left\{\left(\varepsilon_m + U\hat{d}_m^{\dagger S}\hat{d}_m^{S}\right)\hat{h}_m^{S}\right\}^{I}(t) \\
&= ie^{i\mathcal{H}_0^S t}\left(\varepsilon_m + U\hat{d}_m^{\dagger S}\hat{d}_m^{S}\right)\hat{h}_m^{S}e^{-i\mathcal{H}_0^S t} \\
&= i\left(\varepsilon_m + Ue^{i\mathcal{H}_0^S t}\hat{d}_m^{\dagger S}\hat{d}_m^{S}e^{-i\mathcal{H}_0^S t}\right)e^{i\mathcal{H}_0^S t}\hat{h}_m^{S}e^{-i\mathcal{H}_0^S t} \\
&\stackrel{26}{=} i\left(\varepsilon_m + U\left(\hat{d}_m^{\dagger I}\hat{d}_m^{I}\right)(t)\right)\hat{h}_m^{\dagger I}(t)
\end{aligned} \tag{35}$$

With this and the idempotence-relation Equation 5 one can first derive Equation 36 and finally use an exponential function as a natural ansatz to solve Equation 35 with Equation 37.

$$\begin{aligned}
e^{a\cdot\hat{h}_{l,\sigma}^{\dagger S}\hat{h}_{l,\sigma}^{S}} &= \sum_{m=0}^{\infty}\frac{a^m\left(\hat{h}_{l,\sigma}^{\dagger S}\hat{h}_{l,\sigma}^{S}\right)^m}{m!} \stackrel{5}{=} 1 + \left[\sum_{m=1}^{\infty}\frac{a^m}{m!}\right]\cdot\left(\hat{h}_{l,\sigma}^{\dagger S}\hat{h}_{l,\sigma}^{S}\right) \\
&= 1 + \left[\sum_{m=0}^{\infty}\frac{a^m}{m!} - 1\right]\cdot\left(\hat{h}_{l,\sigma}^{\dagger S}\hat{h}_{l,\sigma}^{S}\right) = 1 + (e^a - 1)\cdot\hat{h}_{l,\sigma}^{\dagger S}\hat{h}_{l,\sigma}^{S}
\end{aligned} \tag{36}$$

$$\begin{aligned}
\stackrel{35,34}{\Longrightarrow} \hat{h}_m^{\dagger I}(t) &= e^{i\cdot\varepsilon_m\cdot t + i\cdot U\left(\hat{d}_m^{\dagger I}\hat{d}_m^{I}\right)(t)\cdot t}\hat{h}_m^{\dagger S} \stackrel{34}{=} e^{i\cdot\varepsilon_m\cdot t + i\cdot U\cdot\hat{d}_m^{\dagger S}\hat{d}_m^{S}\cdot t}\hat{h}_m^{\dagger S} \\
&\stackrel{36}{=} e^{i\cdot\varepsilon_m\cdot t}\left(1 + \left(e^{i\cdot U\cdot t} - 1\right)\hat{d}_m^{\dagger S}\hat{d}_m^{S}\right)\hat{h}_m^{\dagger S}
\end{aligned} \tag{37}$$

From analogous calculations follows Equation 38:

$$\begin{aligned}
\hat{h}_m^{\dagger I}(t) &= e^{i\cdot\varepsilon_m\cdot t}\left(1 + \left(e^{i\cdot U\cdot t} - 1\right)\hat{d}_m^{\dagger S}\hat{d}_m^{S}\right)\hat{h}_m^{\dagger S} \\
\hat{h}_m^{I}(t) &= e^{-i\cdot\varepsilon_m\cdot t}\left(1 + \left(e^{-i\cdot U\cdot t} - 1\right)\hat{d}_m^{\dagger S}\hat{d}_m^{S}\right)\hat{h}_m^{S} \\
\hat{d}_m^{\dagger I}(t) &= e^{i\cdot\varepsilon_m\cdot t}\left(1 + \left(e^{i\cdot U\cdot t} - 1\right)\hat{h}_m^{\dagger S}\hat{h}_m^{S}\right)\hat{d}_m^{\dagger S} \\
\hat{d}_m^{I}(t) &= e^{-i\cdot\varepsilon_m\cdot t}\left(1 + \left(e^{-i\cdot U\cdot t} - 1\right)\hat{h}_m^{\dagger S}\hat{h}_m^{S}\right)\hat{d}_m^{S}
\end{aligned} \tag{38}$$

Injecting these derivations into Equation 23, finally $\hat{V}^{I}(t)$ can be derived in Equation 39.

$$\hat{V}^I(t) = \left\{ \hat{V}^S \right\}^I (t) \overset{23}{=} -J \cdot \sum_{[l,m]} \left\{ \left(\hat{h}_l^{\dagger S} \hat{h}_m^S + \hat{d}_l^{\dagger S} \hat{d}_m^S \right) \right\}^I (t)$$

$$= -J \cdot \sum_{[l,m]} \left(\hat{h}_l^{\dagger I}(t) \hat{h}_m^I(t) + \hat{d}_l^{\dagger I}(t) \hat{d}_m^I(t) \right) \tag{39}$$

$$\overset{MM}{=} -J \cdot \sum_{[l,m]} \left[\Lambda_A(l,m,t) \cdot \hat{F}_A(l,m) + \Lambda_B(l,m,t) \cdot \hat{F}_B(l,m) + \Lambda_C(l,m,t) \cdot \hat{F}_C(l,m) \right]$$

It is possible to express these operator-strings as single hoppings, decorated with number operators. One receives Equation 40, which is symmetrical in $\uparrow$ and $\downarrow$ (with $\overline{\uparrow} = \downarrow$ and $\overline{\downarrow} = \uparrow$).

$$\Lambda_A(l,m,t) \overset{MM}{=} e^{i \cdot (\varepsilon_l - \varepsilon_m) \cdot t} \qquad \hat{F}_A(l,m) \overset{MM}{=} \sum_{\sigma \in \{\uparrow, \downarrow\}} \hat{h}_{l,\sigma}^{\dagger S} \hat{h}_{m,\sigma}^S \left(1 + 2 \cdot \hat{n}_{l,\overline{\sigma}}^S \hat{n}_{m,\overline{\sigma}}^S - \hat{n}_{l,\overline{\sigma}}^S - \hat{n}_{m,\overline{\sigma}}^S \right)$$

$$\Lambda_B(l,m,t) \overset{MM}{=} e^{i \cdot (\varepsilon_l - \varepsilon_m + U) \cdot t} \qquad \hat{F}_B(l,m) \overset{MM}{=} \sum_{\sigma \in \{\uparrow, \downarrow\}} \hat{h}_{l,\sigma}^{\dagger S} \hat{h}_{m,\sigma}^S \left(\hat{n}_{l,\overline{\sigma}}^S - \hat{n}_{l,\overline{\sigma}}^S \hat{n}_{m,\overline{\sigma}}^S \right) \tag{40}$$

$$\Lambda_C(l,m,t) \overset{MM}{=} e^{i \cdot (\varepsilon_l - \varepsilon_m - U) \cdot t} \qquad \hat{F}_C(l,m) \overset{MM}{=} \sum_{\sigma \in \{\uparrow, \downarrow\}} \hat{h}_{l,\sigma}^{\dagger S} \hat{h}_{m,\sigma}^S \left(\hat{n}_{m,\overline{\sigma}}^S - \hat{n}_{m,\overline{\sigma}}^S \hat{n}_{l,\overline{\sigma}}^S \right)$$

The expressions for the dressed operators in Equation 40 can be rearranged to better reflect their physical effects. Equation 41 can be obtained and verified by comparing the measurements in a truth table (the $\hat{n}_{*,*}^S$ measure either a 0 or 1 on a state, without modifying it).

$$\hat{F}_A(l,m) \overset{MM}{=} \sum_{\sigma \in \{\uparrow, \downarrow\}} \hat{h}_{l,\sigma}^{\dagger S} \hat{h}_{m,\sigma}^S \left(\hat{n}_{l,\overline{\sigma}}^S \overset{\wedge}{=} \hat{n}_{m,\overline{\sigma}}^S \right)$$

$$\hat{F}_B(l,m) \overset{MM}{=} \sum_{\sigma \in \{\uparrow, \downarrow\}} \hat{h}_{l,\sigma}^{\dagger S} \hat{h}_{m,\sigma}^S \hat{n}_{l,\overline{\sigma}}^S \left(1 - \hat{n}_{m,\overline{\sigma}}^S \right) \tag{41}$$

$$\hat{F}_C(l,m) \overset{MM}{=} \sum_{\sigma \in \{\uparrow, \downarrow\}} \hat{h}_{l,\sigma}^{\dagger S} \hat{h}_{m,\sigma}^S \hat{n}_{m,\overline{\sigma}}^S \left(1 - \hat{n}_{l,\overline{\sigma}}^S \right)$$

The $\overset{\wedge}{=}$ here means "1 if the measurements of the operators are equivalent, 0 otherwise".

Base Energy

In Equation 30 the effective Hamiltonian $\mathcal{H}_{\text{eff}}(N,t) = -iE_0(N)t + \mathcal{H}_N(t)$ was defined. This construct will be necessary in the following sections. $E_0(N)$ is calculated quite swiftly, as can be see in Equation 42 (with $n_{l,\sigma}$ - the occupation-number in $|N\rangle$, not the operator).

$$E_0(N) = \frac{\langle N|\mathcal{H}_0|N\rangle}{\langle N|N\rangle}$$

$$\overset{14}{=} \langle N|U \cdot \sum_l \hat{n}_{l,\uparrow}\hat{n}_{l,\downarrow}|N\rangle + \langle N|\sum_{l,\sigma} \varepsilon_l \hat{n}_{l,\sigma}|N\rangle \tag{42}$$

$$= U \cdot \sum_l n_{l,\uparrow}n_{l,\downarrow} + \sum_{l,\sigma} \varepsilon_l n_{l,\sigma}$$

Calculate the first order of $\mathcal{H}_N(t)$

The last missing element for $\mathcal{H}_{\text{eff}}(N,t) = -iE_0(N)t + \mathcal{H}_N(t)$ now is $\mathcal{H}_N(t)$. At least for the first order the integration in Equation 32 is doable, with now knowing the form of $\hat{V}^I(t)$.
The result of the integration can be seen in Equation 43.

$$\mathcal{H}_N^1(t) \overset{32}{=} -i \int_0^t dt' \frac{\langle N|\hat{V}^I(t')|\Psi^S\rangle}{\langle N|\Psi^S\rangle}$$

$$\overset{24}{=} -i\frac{1}{\Psi_N} \int_0^t dt' \sum_K \langle N|\hat{V}^I(t')|K\rangle \Psi_K$$

$$\overset{39}{=} i \cdot J \sum_K \sum_{[l,m]} \frac{\Psi_K}{\Psi_N} \int_0^t dt' \left[e^{i \cdot (\varepsilon_l - \varepsilon_m) \cdot t'} \cdot \langle N|\hat{F}_A(l,m)|K\rangle + \right.$$

$$\left. e^{i \cdot (\varepsilon_l - \varepsilon_m + U) \cdot t'} \cdot \langle N|\hat{F}_B(l,m)|K\rangle + e^{i \cdot (\varepsilon_l - \varepsilon_m - U) \cdot t'} \cdot \langle N|\hat{F}_C(l,m)|K\rangle \right] \tag{43}$$

$$= J \sum_K \sum_{[l,m]} \frac{\Psi_K}{\Psi_N} \left[\frac{e^{i \cdot (\varepsilon_l - \varepsilon_m) \cdot t} - 1}{\varepsilon_l - \varepsilon_m} \cdot \langle N|\hat{F}_A(l,m)|K\rangle + \right.$$

$$\left. \frac{e^{i \cdot (\varepsilon_l - \varepsilon_m + U) \cdot t} - 1}{\varepsilon_l - \varepsilon_m + U} \cdot \langle N|\hat{F}_B(l,m)|K\rangle + \frac{e^{i \cdot (\varepsilon_l - \varepsilon_m - U) \cdot t} - 1}{\varepsilon_l - \varepsilon_m - U} \cdot \langle N|\hat{F}_C(l,m)|K\rangle \right]$$

$$= J \sum_K \sum_{[l,m]} \frac{\Psi_K}{\Psi_N} \left[\Pi_A(l,m,t) \cdot \langle N|\hat{F}_A(l,m)|K\rangle + \right.$$

$$\left. \Pi_B(l,m,t) \cdot \langle N|\hat{F}_B(l,m)|K\rangle + \Pi_C(l,m,t) \cdot \langle N|\hat{F}_C(l,m)|K\rangle \right]$$

In the second to last step, the i is moved into the integral.
While it seems like $\sum_K \sum_{[l,m]}$ is even a larger summation than the previously problematic ones, because the operators $\hat{F}_A(l,m)$, $\hat{F}_B(l,m)$ and $\hat{F}_C(l,m)$ have interaction range limited to nearest neighbors, actually the summation is really sparse and most entries are 0. This summation can be efficiently evaluated in $\mathcal{O}(\#(\text{sites}) \cdot \#(\text{nearest neighbors}))$. How this could be done, can be looked up in the implementation [20]: `/computation-scripts/hamiltonian.py` in the specialization class `FirstOrderCanonicalHamiltonian`.

Calculate the second order of $\mathcal{H}_N(t)$

For the calculation of the second order, the time-ordering-operator becomes relevant again. However, it is not necessary for the right side of the integral as the two integrals can be fully localized to one factor each. For the left side it is necessary to integrate different cases separately.

$$\mathcal{H}_N^2(t) \overset{32}{=} -\frac{1}{2} \int_0^t dt_1 \int_0^t dt_2 \left(\frac{\langle N|\mathbb{T}\hat{V}^I(t_1)\hat{V}^I(t_2)|\Psi^S\rangle}{\langle N|\Psi^S\rangle} - \frac{\langle N|\hat{V}^I(t_1)|\Psi^S\rangle \cdot \langle N|\hat{V}^I(t_2)|\Psi^S\rangle}{\langle N|\Psi^S\rangle^2} \right)$$

$$\overset{24}{=} -\frac{1}{2} \left[\sum_K \frac{\Psi_K}{\Psi_N} \int_0^t \int_0^t \langle N|\mathbb{T}\hat{V}^I(t_1)\hat{V}^I(t_2)|K\rangle \, dt_1 dt_2 \right.$$

$$\left. - \sum_M \sum_L \frac{\Psi_M \Psi_L}{\Psi_N{}^2} \int_0^t \langle N|\hat{V}^I(t_1)|M\rangle \, dt_1 \cdot \int_0^t \langle N|\hat{V}^I(t_2)|L\rangle \, dt_2 \right]$$

$$\overset{39}{=} -\frac{J^2}{2} \left[\sum_K \sum_{X,Y \in \{A,B,C\}} \sum_{[l,m]} \sum_{[j,k]} \frac{\Psi_K}{\Psi_N} \right.$$

$$\int_0^t \int_0^t \langle N|\mathbb{T}\Lambda_X(l,m,t_1)\hat{F}_X^{t_1}(l,m) \cdot \Lambda_Y(j,k,t_2)\hat{F}_Y^{t_2}(j,k)|K\rangle \, dt_1 dt_2$$

$$- \sum_M \sum_L \sum_{V,W \in \{A,B,C\}} \sum_{[o,p]} \sum_{[q,r]} \frac{\Psi_M \Psi_L}{\Psi_N{}^2}$$

$$\left. \int_0^t \langle N|\Lambda_V(o,p,t_1) \cdot \hat{F}_V(o,p)|M\rangle \, dt_1 \cdot \int_0^t \langle N|\Lambda_W(q,r,t_2) \cdot \hat{F}_W(q,r)|L\rangle \, dt_2 \right]$$

(44)

$$= -\frac{J^2}{2} \left[\sum_K \sum_{X,Y \in \{A,B,C\}} \sum_{[l,m]} \sum_{[j,k]} \frac{\Psi_K}{\Psi_N} \right.$$

$$\int_0^t \int_0^t \Lambda_X(l,m,t_1) \cdot \Lambda_Y(j,k,t_2) \langle N|\mathbb{T}\hat{F}_X^{t_1}(l,m) \cdot \hat{F}_Y^{t_2}(j,k)|K\rangle \, dt_1 dt_2$$

$$- \sum_M \sum_L \sum_{V,W \in \{A,B,C\}} \sum_{[o,p]} \sum_{[q,r]} \frac{\Psi_M \Psi_L}{\Psi_N{}^2}$$

$$\left. (-1) \cdot \Pi_V(o,p,t) \cdot \Pi_W(q,r,t) \langle N|\hat{F}_V(o,p)|M\rangle \langle N|\hat{F}_W(q,r)|L\rangle \right]$$

One might notice in the second to last step the $\hat{F}_*^{t_*}(*,*)$ is indicated with a t_* to keep its time-ordering association, even after writing out the $\hat{V}^I(t_*)$. In the last step an additional -1 appears, because of a $-1 = i \cdot i$ that was introduced to allow for integration of $\Lambda_*(*,*,t_*)$ like in Equation 43.

$$\varepsilon_l - \varepsilon_m = \widetilde{U}_A(l, m)$$
$$\varepsilon_l - \varepsilon_m + U = \widetilde{U}_B(l, m) \tag{45}$$
$$\varepsilon_l - \varepsilon_m - U = \widetilde{U}_C(l, m)$$

Inverting the steps performed in [25], the integrals can be split with the time-ordering-operator for evaluation in Equation 46.

$$\int_0^t dt_1 \int_0^{t_1} dt_2 A(t_1)B(t_2) = \int_0^t dt_1 \int_0^{t_1} dt_2 \mathbb{T} A(t_1)B(t_2)$$

$$\int_0^t dt_1 \int_{t_1}^t dt_2 B(t_2)A(t_1) = \int_0^t dt_1 \int_{t_1}^t dt_2 \mathbb{T} A(t_1)B(t_2) \tag{46}$$

$$\int_0^t dt_1 \left(\int_0^{t_1} dt_2 \mathbb{T} A(t_1)B(t_2) + \int_{t_1}^t dt_2 \mathbb{T} A(t_1)B(t_2) \right) = \int_0^t dt_1 \int_0^t dt_2 \mathbb{T} A(t_1)B(t_2)$$

What remains is the evaluation of the integral that needs to respect the time-ordering.

$$\int_0^t \int_0^t \Lambda_X(l, m, t_1) \cdot \Lambda_Y(j, k, t_2) \langle N | \mathbb{T} \hat{F}_X^{t_1}(l, m) \cdot \hat{F}_Y^{t_2}(j, k) | K \rangle \, dt_1 dt_2$$

$$\overset{45,\,40}{=} \int_0^t \int_0^t e^{i \cdot \widetilde{U}_X(l, m) \cdot t_1} \cdot e^{i \cdot \widetilde{U}_Y(j, k) \cdot t_2} \langle N | \mathbb{T} \hat{F}_X^{t_1}(l, m) \cdot \hat{F}_Y^{t_2}(j, k) | K \rangle \, dt_1 dt_2$$

$$\overset{46}{=} \int_0^t \int_0^{t_1} e^{i \cdot \widetilde{U}_X(l, m) \cdot t_1} e^{i \cdot \widetilde{U}_Y(j, k) \cdot t_2} \, dt_2 dt_1 \cdot \langle N | \hat{F}_X(l, m) \hat{F}_Y(j, k) | K \rangle \tag{47}$$

$$+ \int_0^t \int_{t_1}^t e^{i \cdot \widetilde{U}_X(l, m) \cdot t_1} e^{i \cdot \widetilde{U}_Y(j, k) \cdot t_2} \, dt_2 dt_1 \cdot \langle N | \hat{F}_Y(j, k) \hat{F}_X(l, m) | K \rangle$$

$$\Gamma_1(X, l, m, Y, j, k) =$$

$$= \int_0^t \int_0^{t_1} e^{i \cdot \widetilde{U}_X(l,m) \cdot t_1} e^{i \cdot \widetilde{U}_Y(j,k) \cdot t_2} \, \mathrm{d}t_2 \mathrm{d}t_1 = \int_0^t e^{i \cdot \widetilde{U}_X(l,m) \cdot t_1} \int_0^{t_1} e^{i \cdot \widetilde{U}_Y(j,k) \cdot t_2} \, \mathrm{d}t_2 \mathrm{d}t_1$$

$$= \int_0^t e^{i \cdot \widetilde{U}_X(l,m) \cdot t_1} \frac{e^{i \cdot \widetilde{U}_Y(j,k) \cdot t} - 1}{i \cdot \widetilde{U}_Y(j,k)} \, \mathrm{d}t_1 = \frac{e^{i \cdot \widetilde{U}_X(l,m) \cdot t} - 1}{\widetilde{U}_X(l,m) \cdot \widetilde{U}_Y(j,k)} + \int_0^t \frac{e^{i \cdot \left(\widetilde{U}_X(l,m) + \widetilde{U}_Y(j,k) \right) \cdot t_1}}{i \cdot \widetilde{U}_Y(j,k)} \, \mathrm{d}t_1 \qquad (48)$$

$$= \begin{cases} \widetilde{U}_X(l,m) + \widetilde{U}_Y(j,k) = 0 : & \frac{e^{i \cdot \widetilde{U}_X(l,m) \cdot t} - 1}{\widetilde{U}_X(l,m) \cdot \widetilde{U}_Y(j,k)} - \frac{i \cdot t}{\widetilde{U}_Y(j,k)} \\[2ex] \widetilde{U}_X(l,m) + \widetilde{U}_Y(j,k) \neq 0 : & \frac{e^{i \cdot \widetilde{U}_X(l,m) \cdot t} - 1}{\widetilde{U}_X(l,m) \cdot \widetilde{U}_Y(j,k)} - \frac{e^{i \cdot \left(\widetilde{U}_X(l,m) + \widetilde{U}_Y(j,k) \right) \cdot t} - 1}{\widetilde{U}_Y(j,k) \cdot \left(\widetilde{U}_X(l,m) + \widetilde{U}_Y(j,k) \right)} \end{cases}$$

$$\Gamma_2(X, l, m, Y, j, k) =$$

$$= \int_0^t \int_{t_1}^t e^{i \cdot \widetilde{U}_X(l,m) \cdot t_1} e^{i \cdot \widetilde{U}_Y(j,k) \cdot t_2} \, \mathrm{d}t_2 \mathrm{d}t_1 = \int_0^t e^{i \cdot \widetilde{U}_X(l,m) \cdot t_1} \int_{t_1}^t e^{i \cdot \widetilde{U}_Y(j,k) \cdot t_2} \, \mathrm{d}t_2 \mathrm{d}t_1$$

$$= \int_0^t e^{i \cdot \widetilde{U}_X(l,m) \cdot t_1} \frac{e^{i \cdot \widetilde{U}_Y(j,k) \cdot t} - e^{i \cdot \widetilde{U}_Y(j,k) \cdot t_1}}{i \cdot \widetilde{U}_Y(j,k)} \, \mathrm{d}t_1$$

$$= \frac{e^{i \cdot \widetilde{U}_Y(j,k) \cdot t} - e^{i \cdot \left(\widetilde{U}_X(l,m) + \widetilde{U}_Y(j,k) \right) \cdot t}}{\widetilde{U}_X(l,m) \cdot \widetilde{U}_Y(j,k)} - \int_0^t \frac{e^{i \cdot \left(\widetilde{U}_X(l,m) + \widetilde{U}_Y(j,k) \right) \cdot t_1}}{i \cdot \widetilde{U}_Y(j,k)} \, \mathrm{d}t_1 \qquad (49)$$

$$= \begin{cases} \widetilde{U}_X(l,m) + \widetilde{U}_Y(j,k) = 0 : & \frac{e^{i \cdot \widetilde{U}_Y(j,k) \cdot t} - e^{i \cdot \left(\widetilde{U}_X(l,m) + \widetilde{U}_Y(j,k) \right) \cdot t}}{\widetilde{U}_X(l,m) \cdot \widetilde{U}_Y(j,k)} + \frac{i \cdot t}{\widetilde{U}_Y(j,k)} \\[2ex] \widetilde{U}_X(l,m) + \widetilde{U}_Y(j,k) \neq 0 : & \frac{e^{i \cdot \widetilde{U}_Y(j,k) \cdot t} - e^{i \cdot \left(\widetilde{U}_X(l,m) + \widetilde{U}_Y(j,k) \right) \cdot t}}{\widetilde{U}_X(l,m) \cdot \widetilde{U}_Y(j,k)} + \frac{e^{i \cdot \left(\widetilde{U}_X(l,m) + \widetilde{U}_Y(j,k) \right) \cdot t} - 1}{\widetilde{U}_Y(j,k) \cdot \left(\widetilde{U}_X(l,m) + \widetilde{U}_Y(j,k) \right)} \end{cases}$$

Adding Equation 48 and Equation 49, Equation 50 can be verified, which is necessary when comparing with the integration result of the second half of Equation 44.

$$\Gamma_1(X, l, m, Y, j, k) + \Gamma_2(X, l, m, Y, j, k) =$$

$$= \frac{e^{i \cdot \widetilde{U}_X(l,m) \cdot t} - 1}{\widetilde{U}_X(l,m) \cdot \widetilde{U}_Y(j,k)} + \frac{e^{i \cdot \widetilde{U}_Y(j,k) \cdot t} - e^{i \cdot \left(\widetilde{U}_X(l,m) + \widetilde{U}_Y(j,k) \right) \cdot t}}{\widetilde{U}_X(l,m) \cdot \widetilde{U}_Y(j,k)} \qquad (50)$$

$$= -\frac{e^{i \cdot \widetilde{U}_X(l,m) \cdot t} - 1}{\widetilde{U}_X(l,m)} \cdot \frac{e^{i \cdot \widetilde{U}_Y(j,k) \cdot t} - 1}{\widetilde{U}_Y(j,k)} = -\Pi_X(l, m, t) \cdot \Pi_Y(j, k, t)$$

With the integrals rewritten, Equation 44 can be re-arranged to obtain the final result:

$$\mathcal{H}_N^2(t) \overset{47,\,48,\,49}{=} -\frac{J^2}{2} \sum_{X,Y\in\{A,B,C\}} \sum_{[l,m]} \sum_{[j,k]}$$

$$\left[\sum_K \frac{\Psi_K}{\Psi_N} \Gamma_1(X,l,m,Y,j,k) \cdot \langle N|\hat{F}_X(l,m)\cdot\hat{F}_Y(j,k)|K\rangle \right.$$

$$+ \sum_K \frac{\Psi_K}{\Psi_N} \Gamma_2(X,l,m,Y,j,k) \cdot \langle N|\hat{F}_Y(j,k)\cdot\hat{F}_X(l,m)|K\rangle$$

$$\left. - \sum_M \sum_L \frac{\Psi_M\Psi_L}{\Psi_N{}^2} \left[\Gamma_1(X,l,m,Y,j,k)+\Gamma_2(X,l,m,Y,j,k)\right] \cdot \langle N|\hat{F}_X(l,m)|M\rangle \cdot \langle N|\hat{F}_Y(j,k)|L\rangle \right] \tag{51}$$

As for the first order perturbation, this seems to have made this computationally more expensive than the full diagonalization. But again, because of the limited interaction range (for second order the range is a sphere of at most two nearest-neighbor-hops), this can be efficiently evaluated in $\mathcal{O}(\#(\text{sites})\cdot\#(\text{nearest neighbors})^2)$ by using the sparse nature of the operators.

The Operator $\hat{F}_X(l,m)\hat{F}_Y(j,k)$ is actually already far too complicated to write down any of its mixed terms in an ordered, closed form. To get around this, in the implementation the first half is applied to the left, the state modified accordingly and then the second half is applied – as this is much more efficient than calculating the closed form.

See [20]: `/calculation-helpers/generatesecondorder.py` on how this could be done. Close inspection discloses Equation 51 being 0 for all cases, where $l \neq j \wedge l \neq k \wedge m \neq j \wedge n \neq k$, if specific properties hold for Ψ_*. In a subsequent section this optimization will be used.

2.2 Observable Measurements

After the derivation in the previous section, it is now possible to obtain the time-evolution of a particular wave-function. However, we can not make use of this information directly. To gain insight into what is happening, it is necessary to compute *observable quantities* from the wave-function, that can then be compared to substantiate theoretical claims or validate the model with experimental data.

2.2.1 Observable Sampling

The evaluation of observables is a primary requirement for this application. First, it is necessary to acquire a more general formula, one later can plug specific observables into.

From Equation 30 and applying *dagger* † to said equation, Equation 52 follows.

$$\langle N|\Psi^{\mathrm{S}}(t)\rangle \overset{30}{=} \langle N| \sum_K e^{\mathcal{H}_{\mathrm{eff}}(K,t)} \Psi_K \, |K\rangle$$

$$= \sum_K e^{\mathcal{H}_{\mathrm{eff}}(K,t)} \Psi_K \underbrace{\langle N|K\rangle}_{\delta_{N,K}} = e^{\mathcal{H}_{\mathrm{eff}}(N,t)} \Psi_N \tag{52}$$

$$\overset{\dagger}{\Rightarrow} \langle \Psi^{\mathrm{S}}(t)|N\rangle = e^{\mathcal{H}_{\mathrm{eff}}^*(N,t)} \Psi_N^*$$

In Equation 53 by starting with the default way of calculating a normalized *expectation-value* for an observable $\hat{\mathcal{O}}$ [26] and applying multiple modifications, a representation can be obtained to be effectively sampled later.

$$\frac{\langle\Psi^{\mathrm{S}}(t)|\hat{\mathcal{O}}|\Psi^{\mathrm{S}}(t)\rangle}{\langle\Psi^{\mathrm{S}}(t)|\Psi^{\mathrm{S}}(t)\rangle} = \frac{\sum\limits_N \langle\Psi^{\mathrm{S}}(t)|N\rangle\,\langle N|\hat{\mathcal{O}}|\Psi^{\mathrm{S}}(t)\rangle}{\sum\limits_K \langle\Psi^{\mathrm{S}}(t)|K\rangle\,\langle K|\Psi^{\mathrm{S}}(t)\rangle}$$

$$\overset{52}{=} \frac{\sum\limits_N e^{\mathcal{H}_{\mathrm{eff}}^*(N,t)} \Psi_N^* \,\langle N|\hat{\mathcal{O}}|\Psi^{\mathrm{S}}(t)\rangle}{\sum\limits_K e^{\mathcal{H}_{\mathrm{eff}}^*(K,t)} \Psi_K^* e^{\mathcal{H}_{\mathrm{eff}}(K,t)} \Psi_K}$$

$$= \frac{\sum\limits_N e^{\mathcal{H}_{\mathrm{eff}}^*(N,t)} \Psi_N^* \,\langle N|\hat{\mathcal{O}}|\Psi^{\mathrm{S}}(t)\rangle}{\sum\limits_K |e^{\mathcal{H}_{\mathrm{eff}}(K,t)}|^2 |\Psi_K|^2} \cdot \frac{\langle N|\Psi^{\mathrm{S}}(t)\rangle}{\langle N|\Psi^{\mathrm{S}}(t)\rangle}$$

$$= \sum\limits_N \frac{e^{\mathcal{H}_{\mathrm{eff}}^*(N,t)} \Psi_N^* \,\langle N|\Psi^{\mathrm{S}}(t)\rangle}{\sum\limits_K |e^{\mathcal{H}_{\mathrm{eff}}(K,t)}|^2 |\Psi_K|^2} \cdot \frac{\langle N|\hat{\mathcal{O}}|\Psi^{\mathrm{S}}(t)\rangle}{\langle N|\Psi^{\mathrm{S}}(t)\rangle}$$

$$\overset{52}{=} \sum\limits_N \underbrace{\frac{|e^{\mathcal{H}_{\mathrm{eff}}(N,t)}|^2 |\Psi_N|^2}{\sum\limits_K |e^{\mathcal{H}_{\mathrm{eff}}(K,t)}|^2 |\Psi_K|^2}}_{\mathrm{P}(N,t)} \cdot \frac{\langle N|\hat{\mathcal{O}}|\Psi^{\mathrm{S}}(t)\rangle}{\langle N|\Psi^{\mathrm{S}}(t)\rangle} \tag{53}$$

$$= \sum\limits_N \mathrm{P}(N,t) \frac{\langle N|\hat{\mathcal{O}}|\Psi^{\mathrm{S}}(t)\rangle}{\langle N|\Psi^{\mathrm{S}}(t)\rangle} = \sum\limits_N \mathrm{P}(N,t) \frac{\sum\limits_K \langle N|\hat{\mathcal{O}}|K\rangle\,\langle K|\Psi^{\mathrm{S}}(t)\rangle}{\langle N|\Psi^{\mathrm{S}}(t)\rangle}$$

$$\overset{52}{=} \sum\limits_N \mathrm{P}(N,t) \frac{\sum\limits_K \langle N|\hat{\mathcal{O}}|K\rangle\, e^{\mathcal{H}_{\mathrm{eff}}(K,t)} \Psi_K}{e^{\mathcal{H}_{\mathrm{eff}}(N,t)} \Psi_N}$$

$$= \sum\limits_N \mathrm{P}(N,t) \underbrace{\sum\limits_K \langle N|\hat{\mathcal{O}}|K\rangle\, e^{\mathcal{H}_{\mathrm{eff}}(K,t)-\mathcal{H}_{\mathrm{eff}}(N,t)} \frac{\Psi_K}{\Psi_N}}_{\hat{\mathcal{O}}_{\mathrm{loc}}(N,t)} = \sum\limits_N \mathrm{P}(N,t) \hat{\mathcal{O}}_{\mathrm{loc}}(N,t)$$

By inserting Equation 52 into Equation 53, a probability-dependent formula is obtained, that can be used to plug in specific observables. The probability of a base-state at a specified time $\mathrm{P}(N,t)$

is important for this calculation. The *subsection 2.4.1 Monte Carlo Sampling* will discuss the viability of obtaining these probabilities and strategies of doing so. The matrix-element $\langle N|\hat{\mathcal{O}}|K\rangle$ has the important property of being 0 for almost all combinations of $\langle N|$ and $|K\rangle$, depending on the chosen observable $\hat{\mathcal{O}}$, which is what makes the calculation of the *local-observable* $\hat{\mathcal{O}}_{\text{loc}}(N, t)$ possible in the first place.

2.2.2 Occupation Types

One possibly interesting observable would be the measurement of *double-occupation* (one particle of spin-up and one of spin-down) per site. The operator $\hat{\mathcal{O}}_{\text{do}}(l)$, measuring the double-occupation on site l, can be written as $\hat{\mathcal{O}}_{\text{do}}(l) = \hat{n}_{l,\uparrow}\hat{n}_{l,\downarrow}$.

As the used basis-states are eigenstates of the number-operators, the evaluation of this operator is quite straight forward. $\langle N|\hat{\mathcal{O}}_{\text{do}}(l)|K\rangle$ becomes $\delta_{N,K} \cdot n_{l,\uparrow} \cdot n_{l,\downarrow}$ with the occupation-number (not operator) of the specific site and spin in $\langle N|$, $n_{l,\sigma}$. Hereby, $\hat{\mathcal{O}}_{\text{loc}}(N, t)$ can be simply evaluated to the result shown in Equation 54.

$$\hat{\mathcal{O}}_{\text{loc}}(N, t) = \sum_K \langle N|\hat{\mathcal{O}}_{\text{do}}(l)|K\rangle \, e^{\mathcal{H}_{\text{eff}}(K,t) - \mathcal{H}_{\text{eff}}(N,t)} \frac{\Psi_K}{\Psi_N} = n_{l,\uparrow} \cdot n_{l,\downarrow} \tag{54}$$

The operator measuring *single-occupation* $\hat{\mathcal{O}}_{\text{so}}(l, \sigma)$ is derived in the exact same way.

$$\hat{\mathcal{O}}_{\text{loc}}(N, t) = \sum_K \langle N|\hat{\mathcal{O}}_{\text{so}}(l, \sigma)|K\rangle \, e^{\mathcal{H}_{\text{eff}}(K,t) - \mathcal{H}_{\text{eff}}(N,t)} \frac{\Psi_K}{\Psi_N} = n_{l,\sigma} \tag{55}$$

2.2.3 Spin-Current

For measuring the spin-polarized flow of particles, a slightly more complex observable must be employed. The current operator is dependent on the Hamiltonian and must be derived from the *continuity equation* (Equation 57) in the *Heisenberg Picture* (Equation 56) [27].

$$\hat{A}^{\text{H}}(t) = e^{i\mathcal{H}^{\text{S}}t}\hat{A}^{\text{S}}e^{-i\mathcal{H}^{\text{S}}t} = \left\{ \hat{A}^{\text{S}} \right\}^{\text{H}}(t) \quad \Rightarrow \quad \mathcal{H}^{\text{S}} = \mathcal{H}^{\text{H}} \tag{56}$$

It is important to notice, that the time-evolution in the Heisenberg Picture $\{*\}^{\text{H}}(t)$ is a different one than the previous $\{*\}^{\text{I}}(t)$. For the discrete gradient $\vec{\nabla}_{\text{disc}}$ it is important to remember that the system lives on a discrete, two-dimensional lattice. There, $\partial_x f(k) = f(k+1) - f(k)$ with any site k and $k+1$ being the "next" site in positive x-direction.

$$\frac{d\hat{n}^{H}_{k,\sigma}(t)}{dt} = i\left[\mathcal{H}^{H}, \hat{n}^{H}_{k,\sigma}(t)\right] = i\left\{\left[\mathcal{H}^{S}, \hat{n}^{S}_{k,\sigma}\right]\right\}^{H}(t)$$

$$= -\vec{\nabla}_{\text{disc}} \cdot \vec{j}^{H}_{k,\sigma}(t) = -\left\{\vec{\nabla}_{\text{disc}} \cdot \vec{j}^{S}_{k,\sigma}\right\}^{H}(t) \tag{57}$$

$$\overset{\text{2-dim.}}{\Rightarrow} \begin{pmatrix} \partial_{x} \\ \partial_{y} \end{pmatrix} \cdot \begin{pmatrix} j^{S}_{x\,k,\sigma} \\ j^{S}_{y\,k,\sigma} \end{pmatrix} = \partial_{x} j^{S}_{x\,k,\sigma} + \partial_{y} j^{S}_{y\,k,\sigma} = -i \cdot \left[\mathcal{H}^{S}, \hat{n}^{S}_{k,\sigma}\right]$$

The commutator can be swiftly evaluated, a reference calculation can also be found in
[15]: `/current` . The calculation is somewhat specific to the dimension and geometry, in terms of what neighbor indices are summed over. For arguing about indices and directions, without worrying about geometric details, $\langle l \overset{+}{<} m \rangle$ will describe all possible values of m that fulfill both the properties that the site m is a nearest neighbor of the site l and the site of m lies either in positive x-direction, or y-direction of the other site, but not in negative direction. This bisects the sites along the diagonal of the coordinate system and makes the calculation work for all lattices with symmetry around the x- and y-axis.

$$\left[\mathcal{H}^{S}, \hat{n}^{S}_{k,\sigma}\right] \overset{13}{=} \left[\mathcal{H}^{S}_{0}, \hat{n}^{S}_{k,\sigma}\right] + \left[\hat{V}^{S}, \hat{n}^{S}_{k,\sigma}\right] \overset{34}{=} \left[\hat{V}^{S}, \hat{n}^{S}_{k,\sigma}\right]$$

$$\overset{23}{=} -J \cdot \sum_{[l,m]} \left(\left[\hat{h}^{\dagger S}_{l}\hat{h}^{S}_{m}, \hat{n}^{S}_{k,\sigma}\right] + \left[\hat{d}^{\dagger S}_{l}\hat{d}^{S}_{m}, \hat{n}^{S}_{k,\sigma}\right]\right)$$

$$\overset{\text{MM}}{=} J \cdot \sum_{[l,m]} \left(\hat{h}^{\dagger S}_{l,\sigma}\hat{h}^{S}_{m,\sigma} \cdot \delta_{k,l} - \hat{h}^{\dagger S}_{l,\sigma}\hat{h}^{S}_{m,\sigma} \cdot \delta_{k,m}\right)$$

$$= J \cdot \sum_{\langle l,m \rangle} \left(\hat{h}^{\dagger S}_{l,\sigma}\hat{h}^{S}_{m,\sigma} \cdot \delta_{k,l} - \hat{h}^{\dagger S}_{l,\sigma}\hat{h}^{S}_{m,\sigma} \cdot \delta_{k,m} + \hat{h}^{\dagger S}_{m,\sigma}\hat{h}^{S}_{l,\sigma} \cdot \delta_{k,m} - \hat{h}^{\dagger S}_{m,\sigma}\hat{h}^{S}_{l,\sigma} \cdot \delta_{k,l}\right)$$

$$= J \cdot \sum_{l} \sum_{\langle l \overset{+}{<} m \rangle} \left(\hat{h}^{\dagger S}_{l,\sigma}\hat{h}^{S}_{m,\sigma} \cdot \delta_{k,l} - \hat{h}^{\dagger S}_{m,\sigma}\hat{h}^{S}_{l,\sigma} \cdot \delta_{k,l}\right) \tag{58}$$

$$+ J \cdot \sum_{m} \sum_{\langle m \overset{+}{>} l \rangle} \left(\hat{h}^{\dagger S}_{m,\sigma}\hat{h}^{S}_{l,\sigma} \cdot \delta_{k,m} - \hat{h}^{\dagger S}_{l,\sigma}\hat{h}^{S}_{m,\sigma} \cdot \delta_{k,m}\right)$$

$$= J \cdot \sum_{\langle k \overset{+}{<} m \rangle} \left(\hat{h}^{\dagger S}_{k,\sigma}\hat{h}^{S}_{m,\sigma} - \hat{h}^{\dagger S}_{m,\sigma}\hat{h}^{S}_{k,\sigma}\right) + J \cdot \sum_{\langle k \overset{+}{>} l \rangle} \left(\hat{h}^{\dagger S}_{k,\sigma}\hat{h}^{S}_{l,\sigma} - \hat{h}^{\dagger S}_{l,\sigma}\hat{h}^{S}_{k,\sigma}\right)$$

$$= J \cdot \sum_{l:\,\langle k,l \rangle} \left(\hat{h}^{\dagger S}_{k,\sigma}\hat{h}^{S}_{l,\sigma} - \hat{h}^{\dagger S}_{l,\sigma}\hat{h}^{S}_{k,\sigma}\right)$$

$$\partial_{x} j^{S}_{x\,k,\sigma} + \partial_{y} j^{S}_{y\,k,\sigma} = \hat{\mathcal{O}}_{\text{sp-kin, dir}}(k \to 2, \sigma) - \hat{\mathcal{O}}_{\text{sp-kin, dir}}(4 \to k, \sigma)$$

$$+ \hat{\mathcal{O}}_{\text{sp-kin, dir}}(k \to 3, \sigma) - \hat{\mathcal{O}}_{\text{sp-kin, dir}}(1 \to k, \sigma)$$

$$\overset{57}{=} -i \cdot \left[\mathcal{H}^{S}, \hat{n}^{S}_{k,\sigma}\right] \overset{58}{=} \tag{59}$$

$$i \cdot J \left(\hat{h}^{\dagger S}_{2,\sigma}\hat{h}^{S}_{k,\sigma} - \hat{h}^{\dagger S}_{k,\sigma}\hat{h}^{S}_{2,\sigma} - \hat{h}^{\dagger S}_{k,\sigma}\hat{h}^{S}_{4,\sigma} + \hat{h}^{\dagger S}_{4,\sigma}\hat{h}^{S}_{k,\sigma} + \hat{h}^{\dagger S}_{3,\sigma}\hat{h}^{S}_{k,\sigma} - \hat{h}^{\dagger S}_{k,\sigma}\hat{h}^{S}_{3,\sigma} - \hat{h}^{\dagger S}_{k,\sigma}\hat{h}^{S}_{1,\sigma} + \hat{h}^{\dagger S}_{1,\sigma}\hat{h}^{S}_{k,\sigma}\right)$$
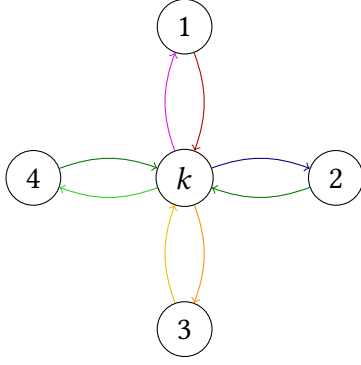
Figure 2.2: Exemplary cutout of a square lattice. The nearest neighbors to a central node with index k are depicted. The numeration is arbitrary, but matches the indices that are used in Equation 59. The colors match the terms that correspond to their transition.

Equation 60 describes the resulting observable measuring such kinetics depending on the direction $\hat{\mathcal{O}}_{\text{sp-kin, dir}}(l \to m, \sigma)$.

$$\hat{\mathcal{O}}_{\text{sp-kin, dir}}(l \to m, \sigma) = iJ \left(\hat{h}^{\dagger}_{m,\sigma} \hat{h}_{l,\sigma} - \hat{h}^{\dagger}_{l,\sigma} \hat{h}_{m,\sigma} \right) \tag{60}$$

In this case, the used basis-states are not eigenstates of the operators. $\langle N|\hat{h}^{\dagger}_{m,\sigma} \hat{h}_{l,\sigma}|K\rangle$ becomes $\delta_{N,\widetilde{N}} \cdot n_{m,\sigma} \cdot (1 - n_{l,\sigma})$, where $|\widetilde{N}\rangle$ is the state obtained when the particle number of site m,σ and the one of site l,σ are swapped (this *hopping* only has an effect, when there is a particle on the original site and no particle yet on the target site, or vice versa). Evaluating the whole operator with the signs correctly and in an efficient manner can best be looked up in the implementation [20]: `/computation-scripts/observables.py`.

Overall, the evaluation of this observable requires knowing the value of an object of the form presented in Equation 61.

$$\hat{\mathcal{O}}_{\text{loc}}(N,t) = i \cdot \left[n_{m,\sigma} \cdot (1 - n_{l,\sigma}) - n_{l,\sigma} \cdot (1 - n_{m,\sigma}) \right] \cdot \frac{\Psi_{\widetilde{N}}}{\Psi_N} \cdot e^{\mathcal{H}_{\text{eff}}(\widetilde{N},t) - \mathcal{H}_{\text{eff}}(N,t)} \tag{61}$$

The *subsection 2.4.2 Analytical Simplifications for Special Cases* will examine how to compute this efficiently for states $|N\rangle$ and $|\widetilde{N}\rangle$ that are connected via the hopping between (nearest-neighbor) lattice sites.

Finally, it is important to notice that the objects from Equation 61 are not real-valued, but complex. Only the complete observable with the full sum over all basis-states $|N\rangle$ over $\hat{\mathcal{O}}_{\text{loc}}(N,t)$ has a fully vanishing imaginary part. Especially when the observable is approximated with an incomplete set of basis-states, computationally an imaginary component remains. It is important to monitor the magnitude of this value and monitor it going to 0. If it does not fully vanish, this indicates an error in the sampling strategy or the implementation or the number of sampled states is not great enough.

One could entertain the question whether is is guaranteed for the imaginary part to vanish with a large enough sample size. On first thought, if the Hamiltonian is only approximated, might there be a possibility for asymmetrical terms to introduce a non-vanishing imaginary part? Conveniently, rewriting Equation 53 in the manner of Equation 62, yields two equal-valued expressions for the value of the observation that are complex conjugations $*$ of each other.

However, this guarantees that no matter how "bad" the $\Psi^S(t)$ approximates the real wave-function, for full sampling the result must have an imaginary part of 0.

$$
\frac{\left\langle \Psi^S(t)\middle|\hat{\mathcal{O}}\middle|\Psi^S(t)\right\rangle}{\left\langle \Psi^S(t)\middle|\Psi^S(t)\right\rangle} \cdot \left(\sum_K \left|e^{\mathcal{H}_{\text{eff}}(K,t)}\right|^2 |\Psi_K|^2 \right) =
$$

$$
\sum_N \left|e^{\mathcal{H}_{\text{eff}}(N,t)}\right|^2 |\Psi_N|^2 \frac{\left\langle N\middle|\hat{\mathcal{O}}\middle|\Psi^S(t)\right\rangle}{\left\langle N\middle|\Psi^S(t)\right\rangle} \overset{*}{=} \sum_N \left|e^{\mathcal{H}_{\text{eff}}(N,t)}\right|^2 |\Psi_N|^2 \frac{\left\langle \Psi^S(t)\middle|\hat{\mathcal{O}}\middle|N\right\rangle}{\left\langle \Psi^S(t)\middle|N\right\rangle}
$$

(62)

2.2.4 Reduced Density Matrix

Typically, the *density matrix* $\rho = |\Psi(t)\rangle \langle \Psi(t)|$ encodes the full quantum information of the state. Because the number of entries in the matrix is the number of base-states squared – which is exponential in the number of sites, exactly what the approximation is supposed to avoid – it is not feasible to access this quantity.

While it is not possible to access the complete matrix, there exists one closely related entity, with multiple uses that are going to be utilized in the subsequent section. The *reduced density matrix* ρ_A is obtained upon *tracing out* (Equation 63, [28]) the part B from a system that is completely partitioned into two parts A and B. The basis of A is $\{|\Phi_k\rangle\}_{k=0}^{2^2-1=3=N} = \{|11\rangle, |10\rangle, |01\rangle, |00\rangle\}$ and the basis of B $\{|X_k\rangle\}_{k=0}^{2^{2\cdot\#(\text{sites})-2}-1=M}$. Notice the order of basis states starting with $|1\dots1\rangle$ then $|1\dots10\rangle$ and ending with $|0\dots0\rangle$. As stated in Equation 12, this is the necessary convention to keep consistent signs, as $|0\dots0\rangle \to |1\dots1\rangle$ would introduce exactly one -1 into all measurements with $\hat{\sigma}^y$.

$$
\rho_A = \text{Tr}_B\left(\rho_{AB}\right) = \sum_{k=0}^{M} \left(\mathbb{1}_A \otimes \langle X_k|\right) \rho_{AB} \left(\mathbb{1}_A \otimes |X_k\rangle\right)
$$

(63)

If – for this application – the part A is reduced to contain only two particles l, σ and m, μ, then ρ_A will be a complex 4×4 matrix. As it is again not feasible to obtain via tracing out, the matrix can be calculated by rewriting it in the basis of Pauli matrices (2.1.1) [24]:

$$
\rho_A(t) = \rho_{l,\sigma,m,\mu}(t) = \frac{1}{4} \sum_{\alpha,\beta\in\{0,x,y,z\}} \left\langle \hat{\sigma}_{l,\sigma}^\alpha \hat{\sigma}_{m,\mu}^\beta \right\rangle_{(t)} \left(\hat{\sigma}^\alpha \otimes \hat{\sigma}^\beta\right)
$$

(64)

Equation 64 uses the property that the 16 matrices $\hat{\sigma}^\alpha \otimes \hat{\sigma}^\beta$ are a complete basis of the complex 4×4 matrices. The observable $\hat{\sigma}_{l,\sigma}^\alpha \hat{\sigma}_{m,\mu}^\beta$ can be translated into hard-core bosonic operators, thanks to the previously derived Equation 11 and with that can easily be measured like all other observables. The factor $\frac{1}{4}$ follows from a normalization argument in Equation 65, that uses properties of density matrices [28] and algebraic properties of the Pauli matrices [29].

$$\left\langle \hat{\sigma}_{l,\sigma}^{\alpha} \hat{\sigma}_{m,\mu}^{\beta} \right\rangle_{(t)} = \text{Tr}\left(\rho_{l,\sigma,m,\mu}(t) \left(\hat{\sigma}_{l,\sigma}^{\alpha} \otimes \hat{\sigma}_{m,\mu}^{\beta} \right) \right)$$

$$\overset{64}{=} \sum_{\alpha',\beta' \in \{0,x,y,z\}} \frac{\left\langle \hat{\sigma}_{l,\sigma}^{\alpha'} \hat{\sigma}_{m,\mu}^{\beta'} \right\rangle_{(t)}}{4} \text{Tr}\left(\left(\hat{\sigma}^{\alpha'} \otimes \hat{\sigma}^{\beta'} \right) \left(\hat{\sigma}^{\alpha} \otimes \hat{\sigma}^{\beta} \right) \right)$$

$$= \sum_{\alpha',\beta' \in \{0,x,y,z\}} \frac{\left\langle \hat{\sigma}_{l,\sigma}^{\alpha'} \hat{\sigma}_{m,\mu}^{\beta'} \right\rangle_{(t)}}{4} \delta_{\alpha',\alpha} \delta_{\beta',\beta} \text{Tr}\left(\mathbb{1}_{4\times4} \right)$$

$$= \frac{4}{4} \cdot \left\langle \hat{\sigma}_{l,\sigma}^{\alpha} \hat{\sigma}_{m,\mu}^{\beta} \right\rangle_{(t)}$$

(65)

Equation 64 now has the advantage that the $\left\langle \hat{\sigma}_{l,\sigma}^{\alpha} \hat{\sigma}_{m,\mu}^{\beta} \right\rangle_{(t)}$ can be evaluated by sampling from a state $\Psi(t)$, that has been acquired by any desired means or sampling methods.

The operators $\left(\hat{\sigma}^{\alpha} \otimes \hat{\sigma}^{\beta} \right)$ were manually translated into the required complex base-matrices in [20]: `/calculation-helpers/concurrence/spin-to-z-basis-transformation.py`, which is the same as the *Kronecker product* [30] of the known matrix-representations (Equation 6), precisely because of the consistent basis-convention.

As hinted, one can now employ Equation 11 to translate the operators $\hat{\sigma}_{l,\sigma}^{\alpha} \hat{\sigma}_{m,\mu}^{\beta}$ into a form that may be inserted into the calculation of $\hat{\mathcal{O}}_{\text{loc}}(N,t)$ in Equation 53. The non-trivial translations are listed in Equation 66 (given $l \neq m$, the rest of the 16 factors can be trivially mirrored or multiplied from the give ones, because the $\hat{\sigma}^{0} = \mathbb{1}$ and $\hat{\sigma}^{z}$ operators are diagonal in the used basis and the eigenvalue of all $\mathbb{1}$ is 1).

$$\hat{\mathcal{O}} = \hat{\sigma}_{l,\sigma}^{x} \hat{\sigma}_{m,\mu}^{0} : \quad \hat{\mathcal{O}}_{\text{loc}}(N,t) = \frac{\Psi_{\widetilde{N}}}{\Psi_N} \cdot e^{\mathcal{H}_{\text{eff}}(\widetilde{N},t) - \mathcal{H}_{\text{eff}}(N,t)}$$

$$\hat{\mathcal{O}} = \hat{\sigma}_{l,\sigma}^{y} \hat{\sigma}_{m,\mu}^{0} : \quad \hat{\mathcal{O}}_{\text{loc}}(N,t) = \frac{\Psi_{\widetilde{N}}}{\Psi_N} \cdot e^{\mathcal{H}_{\text{eff}}(\widetilde{N},t) - \mathcal{H}_{\text{eff}}(N,t)} \cdot i \cdot (1 - 2 \cdot n_{l,\sigma})$$

$$\hat{\mathcal{O}} = \hat{\sigma}_{l,\sigma}^{z} \hat{\sigma}_{m,\mu}^{0} : \quad \hat{\mathcal{O}}_{\text{loc}}(N,t) = \frac{\Psi_{\widetilde{N}}}{\Psi_N} \cdot e^{\mathcal{H}_{\text{eff}}(\widetilde{N},t) - \mathcal{H}_{\text{eff}}(N,t)} \cdot (2 \cdot n_{l,\sigma} - 1)$$

$$\hat{\mathcal{O}} = \hat{\sigma}_{l,\sigma}^{x} \hat{\sigma}_{m,\mu}^{x} : \quad \hat{\mathcal{O}}_{\text{loc}}(N,t) = \frac{\Psi_{\widetilde{\widetilde{N}}}}{\Psi_N} \cdot e^{\mathcal{H}_{\text{eff}}(\widetilde{\widetilde{N}},t) - \mathcal{H}_{\text{eff}}(N,t)} \tag{66}$$

$$\hat{\mathcal{O}} = \hat{\sigma}_{l,\sigma}^{y} \hat{\sigma}_{m,\mu}^{y} : \quad \hat{\mathcal{O}}_{\text{loc}}(N,t) = \frac{\Psi_{\widetilde{\widetilde{N}}}}{\Psi_N} \cdot e^{\mathcal{H}_{\text{eff}}(\widetilde{\widetilde{N}},t) - \mathcal{H}_{\text{eff}}(N,t)} \cdot (2 \cdot n_{l,\sigma} - 1) \cdot (1 - 2 \cdot n_{m,\mu})$$

$$\hat{\mathcal{O}} = \hat{\sigma}_{l,\sigma}^{x} \hat{\sigma}_{m,\mu}^{y} : \quad \hat{\mathcal{O}}_{\text{loc}}(N,t) = \frac{\Psi_{\widetilde{\widetilde{N}}}}{\Psi_N} \cdot e^{\mathcal{H}_{\text{eff}}(\widetilde{\widetilde{N}},t) - \mathcal{H}_{\text{eff}}(N,t)} \cdot i \cdot (1 - 2 \cdot n_{m,\mu})$$

$$\hat{\mathcal{O}} = \hat{\sigma}_{l,\sigma}^{y} \hat{\sigma}_{m,\mu}^{x} : \quad \hat{\mathcal{O}}_{\text{loc}}(N,t) = \frac{\Psi_{\widetilde{\widetilde{N}}}}{\Psi_N} \cdot e^{\mathcal{H}_{\text{eff}}(\widetilde{\widetilde{N}},t) - \mathcal{H}_{\text{eff}}(N,t)} \cdot i \cdot (1 - 2 \cdot n_{l,\sigma})$$

Based on the sampled state $|N\rangle$, in Equation 66 the state $|\widetilde{N}\rangle$ is the same state as $|N\rangle$, but with the occupation of site l and spin direction σ *flipped*, meaning an occupation of 0 in $|N\rangle$ is 1 in $|\widetilde{N}\rangle$ and vice versa. Furthermore, $|\widetilde{\widetilde{N}}\rangle$ indicates a flipping of both the occupations of l,σ and m,μ.

The difference of effective Hamiltonian with either one or two flipped occupations can be evaluated efficiently and is the last necessary optimization that will be presented in *section 2.4 Time-Complexity Optimizations*.

It should be noted, that $\hat{\sigma}^z_{l,\sigma}$ measurements are equivalent to re-scaled single-site-occupation measurements like in Equation 55.

2.2.5 Entanglement Measurements

The $\rho_{l,\sigma,m,\mu}(t)$ from Equation 64 has been specifically chosen to give the full quantum information of a subsystem of two arbitrary site-indices and spin-directions. Most importantly, as the research is performed on quantum systems, it is a goal to extract quantum properties and the prime candidate for this is extracting the *entanglement*.

Research in [24] has already used the same method of obtaining the reduced density matrix as presented in Equation 64, with the goal of measuring the entanglement. There, the number of sampled states, required to obtain the measurements of the observables, was called out as a limiting factor. Monte-Carlo sampling was suggested as a fix for this limitation, which will actually be introduced in *subsection 2.4.1 Monte Carlo Sampling*.

In this work, the *concurrence* was chosen as the *entanglement monotone* to provide insight into the entanglement between different sites. The concurrence has been introduced in [31] as a measure to obtain the entanglement of two spins from the reduced density matrix that contains these two spins. A concurrence measurement does not give the value for the *entanglement of formation* directly, but it needs to be rescaled as shown in Figure 2.3. As can be see in the diagram, the rescaling does not introduce major additional features and for that reason the concurrence will be studied without rescaling to the entanglement. The point of minimal entanglement (maximum separability) and maximal entanglement map to the same values and the intermediate values communicate the same qualitative information.
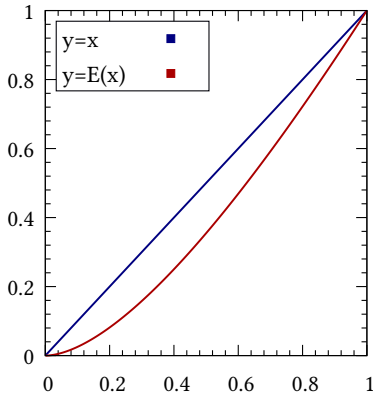

Figure 2.3: Plot of the function that connects the entanglement $E_{\Psi(t)}$ of a subsystem with the concurrence of a subsystem $C(\Psi(t))$ in the relevant region $0 \leq x \leq 1$. According to [31]: $E_{\Psi(t)} = E(C(\Psi(t)))$, with the function $E(x) = H(\frac{1}{2} + \frac{1}{2}\sqrt{1-x^2})$ and the *binary entropy function* $H(x) = -\left[x \cdot \log_2(x) + (1-x) \cdot \log_2(1-x)\right]$. Compared with the identity $y = x$, it shows that the values at the borders of the region $x = 0$ and $x = 1$ are the same for both functions. The examined function $E(x)$ increases monotonically for the relevant input range and lies reasonably close to the identity.

[20]: `/calculation-helpers/concurrence/` holds multiple scripts used to validate that the calculations in the following section give the same result, no matter which of the alternative implementation-/calculation-strategies are used.

[31] introduces the *magic basis* shown in Equation 67 as opposed to the counting basis in the beginning of *subsection 2.2.4 Reduced Density Matrix*. The magic basis consists of *Bell states* with particular phases.

$$|m_1\rangle = \frac{1}{\sqrt{2}}(|11\rangle + |00\rangle)$$

$$|m_2\rangle = \frac{1}{\sqrt{2}}i(|11\rangle - |00\rangle)$$

$$|m_3\rangle = \frac{1}{\sqrt{2}}i(|10\rangle + |01\rangle) \tag{67}$$

$$|m_4\rangle = \frac{1}{\sqrt{2}}(|10\rangle - |01\rangle)$$

The Bell state is a construct tightly linked to entanglement, consequently it is by design straight forward to write down the concurrence of a pure state expressed in the magic basis. For $|\Psi(t)\rangle = \sum_{i=1}^{4} \alpha_i |m_i\rangle$ the concurrence $C(\Psi(t))$ is $C(\Psi(t)) = \left|\sum_{i=1}^{4} \alpha_i^2\right|$. This is equivalent to saying for a pure state expressed in the counting basis $|\Psi(t)\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle$, the concurrence is calculated $C(\Psi(t)) = 2|\alpha\delta - \beta\gamma|$. This gives the same value as for all calculations that extract the concurrence from the density matrix that is calculated from these pure state coefficients. But while all states obtained via time evolution $|\Psi^S(t)\rangle$ – as described in previous chapters – are pure states, through the reduction to a "two-spin-system" of particles l, σ and m, μ to $\rho_{l,\sigma,m,\mu}(t)$ the remaining system is likely to be a mixed state [32].

The calculation necessarily needed to be generalized to take the density matrix as an input. [33] gives two alternate ways of calculating the concurrence from the density matrix. Both require the *spin-flip* operation $\widetilde{\rho}$. The original paper [31] describes this operation for density matrix ρ expressed in the counting basis to be: transforming it into the magic basis of Equation 67, taking the complex conjugate, then transforming it back. As per [33] this is equivalent to the operation described in Equation 68, where ρ and $\hat{\sigma}^y$ (Equation 6) are expressed in the standard counting basis.

$$\widetilde{\rho} = \left(\hat{\sigma}^y \otimes \hat{\sigma}^y\right) \rho^* \left(\hat{\sigma}^y \otimes \hat{\sigma}^y\right) \tag{68}$$

[20]: `/calculation-helpers/concurrence/basis-transformation.py` verifies this equivalence and shows the $\widetilde{*}$-operation modifies a hermitian matrix like shown in Equation 69.

$$\rho = \begin{pmatrix} a & b & c & d \\ b^* & e & f & g \\ c^* & f^* & h & k \\ d^* & g^* & k^* & l \end{pmatrix} \quad \Rightarrow \quad \widetilde{\rho} = \begin{pmatrix} l & -k & -g & d \\ -k^* & h & f & -c \\ -g^* & f^* & e & -b \\ d^* & -c^* & -b^* & a \end{pmatrix} \tag{69}$$

With that knowledge Equation 70 states how to calculate the concurrence $C(\rho)$ from the reduced density matrix ρ.

$$R = \sqrt{\sqrt{\rho} \cdot \widetilde{\rho} \cdot \sqrt{\rho}}$$
$$\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \lambda_4, \quad \{\lambda_i\} = \mathrm{Eigenvalues}(R) \tag{70}$$
$$C(\rho) = \max(0, \lambda_1 - \lambda_2 - \lambda_3 - \lambda_4)$$

In this formula, the square root $\sqrt{M}$ of a square matrix M is a square matrix of the same dimensionality, fulfilling the property $\sqrt{M} \cdot \sqrt{M} = M$. A candidate can be obtained numerically by calculating the eigenvalues of the matrix (which for a hermitian matrix are all positive real numbers), taking their square root and re-assembling the matrix with the eigenvectors like in Equation 71 (λ_i different values than in Equation 70).

$$M = Q \Lambda Q^{\mathrm{T}}$$
$$\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$$
$$\sqrt{\Lambda} = \mathrm{diag}(\sqrt{\lambda_1}, \sqrt{\lambda_2}, \sqrt{\lambda_3}, \sqrt{\lambda_4}) \tag{71}$$
$$\sqrt{M} = Q \sqrt{\Lambda} Q^{\mathrm{T}}$$

While this operation is quite complicated, it is required only once per calculation and only for a matrix of fixed size 4×4. To avoid it, [33] alternatively provides that the λ_i from Equation 70 are the square roots of the non-hermitian matrix $\rho\widetilde{\rho}$. For exact calculations both ways of calculating yield the same results, which is implemented and checked. If there are any inconsistencies in the order of basis vectors however, the results diverge. Which in fact is a good check to verify the bases are consistent across the whole implementation.

2.2.6 Energy

The energy $E(t)$ (Equation 72) is a key measure of many quantum-mechanical experiments, with the operator corresponding to this observable being the Hamiltonian that governs the time-evolution. The variance $\mathrm{Var}_E(t)$ of this quantity (Equation 73) describes the energy-fluctuations in the system.

$$E(t) = \frac{\left\langle \Psi^{\mathrm{S}}(t) \middle| \mathcal{H}^{\mathrm{S}} \middle| \Psi^{\mathrm{S}}(t) \right\rangle}{\left\langle \Psi^{\mathrm{S}}(t) \middle| \Psi^{\mathrm{S}}(t) \right\rangle} = \left\langle \mathcal{H}^{\mathrm{S}} \right\rangle_{(t)} \tag{72}$$

$$\mathrm{Var}_E(t) = \frac{\left\langle \Psi^{\mathrm{S}}(t) \middle| \mathcal{H}^{\mathrm{S}^2} \middle| \Psi^{\mathrm{S}}(t) \right\rangle}{\left\langle \Psi^{\mathrm{S}}(t) \middle| \Psi^{\mathrm{S}}(t) \right\rangle} - \left(\frac{\left\langle \Psi^{\mathrm{S}}(t) \middle| \mathcal{H}^{\mathrm{S}} \middle| \Psi^{\mathrm{S}}(t) \right\rangle}{\left\langle \Psi^{\mathrm{S}}(t) \middle| \Psi^{\mathrm{S}}(t) \right\rangle} \right)^2 = \left\langle \mathcal{H}^{\mathrm{S}^2} \right\rangle_{(t)} - \left\langle \mathcal{H}^{\mathrm{S}} \right\rangle_{(t)}^2 \tag{73}$$

Without any additional constrictions, the energy – as well as all other cumulants of the Hamiltonian operator – must be conserved under normal time evolution, as Equation 74 shows. This comes from the fact that an exact Hamiltonian commutes with itself in all instances and allows to show that the energy $E(t) = E(t = 0)$. With equivalent reasoning, also the higher order

cumulants must necessarily be constant for all times. The energy as first cumulant being 0 or constant comes down to the choice of energy reference.

$$
\begin{aligned}
E(t) &\overset{72}{=} \frac{\left\langle \Psi^{S}(t) \middle| \mathcal{H}^{S} \middle| \Psi^{S}(t) \right\rangle}{\left\langle \Psi^{S}(t) \middle| \Psi^{S}(t) \right\rangle} \overset{25}{=} \frac{\left\langle \Psi^{S} \middle| e^{i\mathcal{H}^{S}t} \mathcal{H}^{S} e^{-i\mathcal{H}^{S}t} \middle| \Psi^{S} \right\rangle}{\left\langle \Psi^{S} \middle| e^{i\mathcal{H}^{S}t} e^{-i\mathcal{H}^{S}t} \middle| \Psi^{S} \right\rangle} \\
&= \frac{\left\langle \Psi^{S} \middle| e^{i\mathcal{H}^{S}t} e^{-i\mathcal{H}^{S}t} \mathcal{H}^{S} \middle| \Psi^{S} \right\rangle}{\left\langle \Psi^{S} \middle| e^{i\mathcal{H}^{S}t} e^{-i\mathcal{H}^{S}t} \middle| \Psi^{S} \right\rangle} = \frac{\left\langle \Psi^{S} \middle| \mathcal{H}^{S} \middle| \Psi^{S} \right\rangle}{\left\langle \Psi^{S} \middle| \Psi^{S} \right\rangle} = E(t=0)
\end{aligned}
\tag{74}
$$

The higher order cumulants vanish depending on the initial state. Equation 75 shows the variance calculation expanded in the eigenbasis of the Hamiltonian. For initial states that are comprised of only one element of that basis $\left| \Psi^{S} \right\rangle \propto \left| E_i \right\rangle$, $\mathcal{H}^{S} \left| E_i \right\rangle = E_i \cdot \left| E_i \right\rangle$, the higher order cumulants vanish. Looking at the last statement in Equation 75, in the case that all $c(E_i) = 0$ except for one specific i, the difference must vanish. Otherwise the variance is non-zero, but still constant in time because of the argument in Equation 74.

$$
\begin{aligned}
\mathrm{Var}_{E}(t=0) &\overset{73}{=} \frac{\left\langle \Psi^{S} \middle| \mathcal{H}^{S^2} \middle| \Psi^{S} \right\rangle}{\left\langle \Psi^{S} \middle| \Psi^{S} \right\rangle} - \left(\frac{\left\langle \Psi^{S} \middle| \mathcal{H}^{S} \middle| \Psi^{S} \right\rangle}{\left\langle \Psi^{S} \middle| \Psi^{S} \right\rangle} \right)^{2} \\
&= \sum_{i} \sum_{j} \frac{\left\langle E_i \middle| c^*(E_i) c(E_j) \mathcal{H}^{S^2} \middle| E_j \right\rangle}{\left\langle \Psi^{S} \middle| \Psi^{S} \right\rangle} - \left(\sum_{i} \sum_{j} \frac{\left\langle E_i \middle| c^*(E_i) c(E_j) \mathcal{H}^{S} \middle| E_j \right\rangle}{\left\langle \Psi^{S} \middle| \Psi^{S} \right\rangle} \right)^{2} \\
&= \frac{\sum_{i} |c(E_i)|^2 E_i^{2}}{\left\langle \Psi^{S} \middle| \Psi^{S} \right\rangle} - \left(\sum_{i} \frac{|c(E_i)|^2 E_i}{\left\langle \Psi^{S} \middle| \Psi^{S} \right\rangle} \right)^{2} = \frac{\sum_{i} |c(E_i)|^2 E_i^{2} - \left(\sum_{i} |c(E_i)|^2 E_i \right)^{2}}{\left\langle \Psi^{S} \middle| \Psi^{S} \right\rangle}
\end{aligned}
\tag{75}
$$

While it might not seem to give much insight to calculate the time evolved measurements of these values if they are constant, in reality this serves the purpose of *checking* the accuracy of potential approximations. If the time evolution is performed with an approximated, effective Hamiltonian, then the exact one will no longer commute with it and the energy and variance will vary over time.

Measuring $\hat{\mathcal{O}}_{\mathrm{loc}}(N, t)$ for $\hat{\mathcal{O}} = \mathcal{H}^{S}$ is straight forward, as Equation 76 shows.

$$
\begin{aligned}
\hat{\mathcal{O}}_{\mathrm{loc}}(N,t) &\overset{53}{=} \sum_K \langle N|\mathcal{H}^{\mathrm{S}}|K\rangle\, e^{\mathcal{H}_{\mathrm{eff}}(K,t)-\mathcal{H}_{\mathrm{eff}}(N,t)}\frac{\Psi_K}{\Psi_N} \\
&\overset{14,15}{=} \sum_K \langle N|U\cdot\sum_l \hat{\mathrm{n}}_{l,\uparrow}\hat{\mathrm{n}}_{l,\downarrow} + \sum_{l,\sigma}\varepsilon_l\hat{\mathrm{n}}_{l,\sigma}|K\rangle\, e^{\mathcal{H}_{\mathrm{eff}}(K,t)-\mathcal{H}_{\mathrm{eff}}(N,t)}\frac{\Psi_K}{\Psi_N} \\
&\quad - J\cdot\sum_K \langle N|\sum_{\langle l,m\rangle,\sigma}\left(\hat{\mathrm{h}}^\dagger_{l,\sigma}\hat{\mathrm{h}}_{m,\sigma} + \hat{\mathrm{h}}^\dagger_{m,\sigma}\hat{\mathrm{h}}_{l,\sigma}\right)|K\rangle\, e^{\mathcal{H}_{\mathrm{eff}}(K,t)-\mathcal{H}_{\mathrm{eff}}(N,t)}\frac{\Psi_K}{\Psi_N} \\
&= \sum_l\sum_K \langle N|U\cdot\hat{\mathrm{n}}_{l,\uparrow}\hat{\mathrm{n}}_{l,\downarrow} + \varepsilon_l\left(\hat{\mathrm{n}}_{l,\uparrow}+\hat{\mathrm{n}}_{l,\downarrow}\right)|K\rangle\, e^{\mathcal{H}_{\mathrm{eff}}(K,t)-\mathcal{H}_{\mathrm{eff}}(N,t)}\frac{\Psi_K}{\Psi_N} \\
&\quad - J\cdot\sum_{\langle l,m\rangle,\sigma}\sum_K \langle N|\hat{\mathrm{h}}^\dagger_{l,\sigma}\hat{\mathrm{h}}_{m,\sigma} + \hat{\mathrm{h}}^\dagger_{m,\sigma}\hat{\mathrm{h}}_{l,\sigma}|K\rangle\, e^{\mathcal{H}_{\mathrm{eff}}(K,t)-\mathcal{H}_{\mathrm{eff}}(N,t)}\frac{\Psi_K}{\Psi_N} \\
&= \sum_l\left(U\cdot n_{l,\uparrow}n_{l,\downarrow} + \varepsilon_l\left(n_{l,\uparrow}+n_{l,\downarrow}\right)\right) \\
&\quad - J\cdot\sum_{\langle l,m\rangle,\sigma}\sum_K \langle N|\hat{\mathrm{h}}^\dagger_{l,\sigma}\hat{\mathrm{h}}_{m,\sigma} + \hat{\mathrm{h}}^\dagger_{m,\sigma}\hat{\mathrm{h}}_{l,\sigma}|K\rangle\, e^{\mathcal{H}_{\mathrm{eff}}(K,t)-\mathcal{H}_{\mathrm{eff}}(N,t)}\frac{\Psi_K}{\Psi_N} \\
&\overset{42}{=} E_0(N,t) - J\cdot\sum_{\langle l,m\rangle,\sigma}\left(n_{l,\sigma}(1-n_{m,\sigma}) + (1-n_{l,\sigma})n_{m,\sigma}\right)e^{\mathcal{H}_{\mathrm{eff}}(\widetilde{N},t)-\mathcal{H}_{\mathrm{eff}}(N,t)}\frac{\Psi_{\widetilde{N}}}{\Psi_N} \\
&= E_0(N,t) - J\cdot\sum_{\langle l,m\rangle,\sigma}\left(n_{l,\sigma}\neq n_{m,\sigma}\right)e^{\mathcal{H}_{\mathrm{eff}}(\widetilde{N},t)-\mathcal{H}_{\mathrm{eff}}(N,t)}\frac{\Psi_{\widetilde{N}}}{\Psi_N} \\
&= E_0(N,t) + E_{\mathrm{V}}(N,t)
\end{aligned}
\tag{76}
$$

$|\widetilde{N}\rangle$ is here the obtained state when starting on state $|N\rangle$ and performing a swapping of the occupations of sites l,σ and m,σ. On comparison with Equation 42 it can be seen that the definition for $E_0(N,t)$ is consistent for the expansion and the observable calculation. Newly defined is the energy $E_{\mathrm{V}}(N,t)$, originating from $\hat{\mathrm{V}}$. As hinted for the other observables earlier, the difference of Hamiltonians with local modifications $\mathcal{H}_{\mathrm{eff}}(\widetilde{N},t) - \mathcal{H}_{\mathrm{eff}}(N,t)$ can be evaluated in constant time. However, the energy is the first observable that requires $\mathcal{O}(\#(\mathrm{sites}))$ on its own to evaluate. This can not be reduced. As will become apparent in the calculation of the variance, this should be an upper bound for observable-calculations.

The variance can be evaluated by splitting the expectation values for the parts $\mathcal{H}^{\mathrm{S}}_0$ and $\hat{\mathrm{V}}^{\mathrm{S}}$ using the *covariance* as presented in Equation 77.

$$
\begin{aligned}
\mathrm{Var}_E(t) &\overset{73}{=} \left\langle \mathcal{H}^{\mathrm{S}^2}\right\rangle_{(t)} - \left\langle \mathcal{H}^{\mathrm{S}}\right\rangle^2_{(t)} \overset{13}{=} \left\langle \left(\mathcal{H}^{\mathrm{S}}_0+\hat{\mathrm{V}}^{\mathrm{S}}\right)^2\right\rangle_{(t)} - \left\langle \mathcal{H}^{\mathrm{S}}_0+\hat{\mathrm{V}}^{\mathrm{S}}\right\rangle^2_{(t)} \\
&= \left\langle \mathcal{H}^{\mathrm{S}^2}_0\right\rangle_{(t)} - \left\langle \mathcal{H}^{\mathrm{S}}_0\right\rangle^2_{(t)} + \left\langle \hat{\mathrm{V}}^{\mathrm{S}^2}\right\rangle_{(t)} - \left\langle \hat{\mathrm{V}}^{\mathrm{S}}\right\rangle^2_{(t)} \\
&\quad + \left\langle \mathcal{H}^{\mathrm{S}}_0\hat{\mathrm{V}}^{\mathrm{S}}+\hat{\mathrm{V}}^{\mathrm{S}}\mathcal{H}^{\mathrm{S}}_0\right\rangle_{(t)} - 2\cdot\left\langle \mathcal{H}^{\mathrm{S}}_0\right\rangle_{(t)}\cdot\left\langle \hat{\mathrm{V}}^{\mathrm{S}}\right\rangle_{(t)}
\end{aligned}
\tag{77}
$$

It is important to notice, that only $\mathcal{H}_0^{S}\hat{V}^{S} + \hat{V}^{S}\mathcal{H}_0^{S}$ is a hermitian operator, $\mathcal{H}_0^{S}\hat{V}^{S}$ alone is not. The derivation for half of the terms can be taken from Equation 76, only $\left\langle \mathcal{H}_0^{S}\hat{V}^{S} + \hat{V}^{S}\mathcal{H}_0^{S} \right\rangle_{(t)}$, $\left\langle \mathcal{H}_0^{S^2} \right\rangle_{(t)}$ and $\left\langle \hat{V}^{S^2} \right\rangle_{(t)}$ require closer inspection.

For the case of $\left\langle \mathcal{H}_0^{S}\hat{V}^{S} + \hat{V}^{S}\mathcal{H}_0^{S} \right\rangle_{(t)}$ and $\left\langle \mathcal{H}_0^{S^2} \right\rangle_{(t)}$, Equation 78 shows the expressions for $\hat{\mathcal{O}}_{\text{loc}}(N,t)$. Comparing with Equation 76, the sums can be fully separated, as the $|N\rangle$ are eigenstates of $\mathcal{H}_0^{S}$ and the $\mathcal{H}_0^{S}$-sum can always be moved outside without interfering with the states that are next to $\hat{V}^{S}$.

$$
\begin{aligned}
\hat{\mathcal{O}} = \mathcal{H}_0^{S^2} : \quad & \hat{\mathcal{O}}_{\text{loc}}(N,t) = (E_0(N,t))^2 \\
\hat{\mathcal{O}} = \mathcal{H}_0^{S}\hat{V}^{S} : \quad & \hat{\mathcal{O}}_{\text{loc}}(N,t) = E_0(N,t) \cdot E_V(N,t) \\
\hat{\mathcal{O}} = \hat{V}^{S}\mathcal{H}_0^{S} : \quad & \hat{\mathcal{O}}_{\text{loc}}(N,t) = [E_0(N,t) \cdot E_V(N,t)]^{*}
\end{aligned}
\tag{78}
$$

Unfortunately, calculating $\hat{\mathcal{O}}_{\text{loc}}(N,t)$ for $\hat{\mathcal{O}} = \hat{V}^{S^2}$ in the exact same way has a way higher runtime complexity than all previous observables. Plugging it into Equation 53 like previous operators, one can not use the strategy of pulling out one sum early, as $|N\rangle$ is not an eigenvector of $\hat{V}^{S}$. And the full expression requires $\mathcal{O}([\#(\text{sites})]^2)$ operations to evaluate.

There are multiple ways to get around this, though. First, there is the possibility of introducing advantageous terms as presented in Equation 79, yielding the desired expression.

$$
\begin{aligned}
\hat{\mathcal{O}}_{\text{loc}}(N,t) \overset{53}{=} & \sum_{K} \langle N|\hat{V}^{S^2}|K\rangle\, e^{\mathcal{H}_{\text{eff}}(K,t) - \mathcal{H}_{\text{eff}}(N,t)} \frac{\Psi_K}{\Psi_N} - (E_V(N,t))^2 + (E_V(N,t))^2 \\[6pt]
\overset{15,76}{=} & J^2 \sum_{[l,m],\sigma} \sum_{[a,b],\mu} \Bigg(\sum_{K} e^{\mathcal{H}_{\text{eff}}(K,t) - \mathcal{H}_{\text{eff}}(N,t)} \frac{\Psi_K}{\Psi_N} \langle N|\hat{h}_{l,\sigma}^{\dagger S}\hat{h}_{m,\sigma}^{S}\hat{h}_{a,\mu}^{\dagger S}\hat{h}_{b,\mu}^{S}|K\rangle \\[4pt]
& - \sum_{M} \sum_{L} \frac{e^{\mathcal{H}_{\text{eff}}(M,t) - \mathcal{H}_{\text{eff}}(N,t)}}{e^{\mathcal{H}_{\text{eff}}(N,t) - \mathcal{H}_{\text{eff}}(L,t)}} \frac{\Psi_M \Psi_L}{\Psi_N^2} \langle N|\hat{h}_{l,\sigma}^{\dagger S}\hat{h}_{m,\sigma}^{S}|L\rangle \langle N|\hat{h}_{a,\mu}^{\dagger S}\hat{h}_{b,\mu}^{S}|M\rangle \Bigg) \\[4pt]
& + (E_V(N,t))^2 \\[6pt]
= & J^2 \sum_{[l,m],\sigma} \sum_{[a,b],\mu} \Bigg(e^{\mathcal{H}_{\text{eff}}(\tilde{\tilde{N}}(l,m,\sigma,a,b,\mu),t) - \mathcal{H}_{\text{eff}}(N,t)} \frac{\Psi_{\tilde{\tilde{N}}(l,m,\sigma,a,b,\mu)}}{\Psi_N} I_{\text{occ}}(N,l,m,\sigma,a,b,\mu) \\[4pt]
& - \frac{e^{\mathcal{H}_{\text{eff}}(\tilde{N}(a,b,\mu),t) - \mathcal{H}_{\text{eff}}(N,t)}}{e^{\mathcal{H}_{\text{eff}}(N,t) - \mathcal{H}_{\text{eff}}(\tilde{N}(l,m,\sigma),t)}} \frac{\Psi_{\tilde{N}(a,b,\mu)} \Psi_{\tilde{N}(l,m,\sigma)}}{\Psi_N^2} \left(n_{l,\sigma} \neq n_{m,\sigma} \right) \left(n_{a,\mu} \neq n_{b,\mu} \right) \Bigg) \\[4pt]
& + (E_V(N,t))^2
\end{aligned}
\tag{79}
$$

$I_{\text{occ}}(N,l,m,\sigma,a,b,\mu)$ is in this relation a four-way interaction term depending on the occupations of the state $|N\rangle$ and the indices l, m, a and b, as well as the spin directions σ and μ. It must take the cases into account where $\sigma = \mu$ and $l = a$ or $l = b$ or $m = a$ or $m = b$ and treat them accordingly. Through the introduction of the extra terms, Equation 79 first seems overly complicated and yet still $\mathcal{O}([\#(\text{sites})]^2)$, because of the two sums over l and a. However, because of the limited inter-

action range of $\mathcal{H}_{\text{eff}}(*, t)$ – based on the order of expansion that is taken into account – under specific conditions (which will be touched in *subsection 2.4.3 Geometry-Dependent Interaction-Range*), the bracket will in fact cancel for terms with sites l and a far enough apart. For larger systems basically all sites are "far enough" apart, meaning the expression can be reduced to a single sum $\mathcal{O}(\#(\text{sites}))$ that for each site only needs to cover the *sphere of possible interactions*. All terms between the states $|\widetilde{N}(*)\rangle \leftrightarrow |N\rangle$ have been previously stated to be possible to be evaluated efficiently if connected by a two-occupation-swap. And also the four-occupation-interactions between $|\widetilde{\widetilde{N}}(*)\rangle \leftrightarrow |N\rangle$ can each be reduced to two two-occupation-swaps. All this together shows that it is possible to reduce second order operators to a computational effort of $\mathcal{O}(\#(\text{sites}))$ times the size of the interaction sphere. This reduction is implemented in the codebase and verified in [20]: `/computation-scripts/compareenergyvariance.py` .

For a second way to acquire the desired values, it is necessary to derive a different expression for $\hat{\mathcal{O}}_{\text{loc}}(N, t)$ than the one previously presented in Equation 53. Starting with an operator that is a second power of any other operator and can be written $\hat{\mathcal{O}} = \hat{\mathcal{O}}'^2$, one can derive the alternate formulation in Equation 80.

$$
\begin{aligned}
\frac{\langle \Psi^S(t)|\hat{\mathcal{O}}'\hat{\mathcal{O}}'|\Psi^S(t)\rangle}{\langle \Psi^S(t)|\Psi^S(t)\rangle} &= \frac{\sum\limits_{N} \langle \Psi^S(t)|\hat{\mathcal{O}}'|N\rangle \langle N|\hat{\mathcal{O}}'|\Psi^S(t)\rangle}{\sum\limits_{K} \langle \Psi^S(t)|K\rangle \langle K|\Psi^S(t)\rangle} \\[2mm]
&\stackrel{52}{=} \sum_{N} \frac{\langle \Psi^S(t)|\hat{\mathcal{O}}'|N\rangle \langle N|\hat{\mathcal{O}}'|\Psi^S(t)\rangle}{\sum\limits_{K} e^{\mathcal{H}_{\text{eff}}^{*}(K,t)}\Psi_K^* e^{\mathcal{H}_{\text{eff}}(K,t)}\Psi_K} \cdot \frac{\langle \Psi^S(t)|N\rangle}{\langle \Psi^S(t)|N\rangle} \cdot \frac{\langle N|\Psi^S(t)\rangle}{\langle N|\Psi^S(t)\rangle} \\[2mm]
&= \sum_{N} \frac{\langle \Psi^S(t)|N\rangle \langle N|\Psi^S(t)\rangle}{\sum\limits_{K} |e^{\mathcal{H}_{\text{eff}}(K,t)}|^2|\Psi_K|^2} \cdot \frac{\langle \Psi^S(t)|\hat{\mathcal{O}}'|N\rangle}{\langle \Psi^S(t)|N\rangle} \cdot \frac{\langle N|\hat{\mathcal{O}}'|\Psi^S(t)\rangle}{\langle N|\Psi^S(t)\rangle} \\[2mm]
&\stackrel{52}{=} \sum_{N} \frac{e^{\mathcal{H}_{\text{eff}}^{*}(N,t)}\Psi_N^* e^{\mathcal{H}_{\text{eff}}(N,t)}\Psi_N}{\sum\limits_{K} |e^{\mathcal{H}_{\text{eff}}(K,t)}|^2|\Psi_K|^2} \cdot \left| \frac{\langle N|\hat{\mathcal{O}}'|\Psi^S(t)\rangle}{\langle N|\Psi^S(t)\rangle} \right|^2 \\[2mm]
&= \sum_{N} \underbrace{\frac{|e^{\mathcal{H}_{\text{eff}}(N,t)}|^2|\Psi_N|^2}{\sum\limits_{K} |e^{\mathcal{H}_{\text{eff}}(K,t)}|^2|\Psi_K|^2}}_{\text{P}(N,t)} \cdot \left| \frac{\langle N|\hat{\mathcal{O}}'|\Psi^S(t)\rangle}{\langle N|\Psi^S(t)\rangle} \right|^2 \\[2mm]
&\stackrel{53}{=} \sum_{N} \text{P}(N,t) \underbrace{\left| \sum_{K} \langle N|\hat{\mathcal{O}}'|K\rangle \, e^{\mathcal{H}_{\text{eff}}(K,t)-\mathcal{H}_{\text{eff}}(N,t)} \frac{\Psi_K}{\Psi_N} \right|^2}_{\hat{\mathcal{O}}_{\text{loc}}(N,t)}
\end{aligned}
\tag{80}
$$

Compared to the previous expression, the result is now much cheaper to compute – as Equation 76 pins the value of the sum in the absolute square to $E_V(N, t)$. Combining the two methods of employing locality and strategic expansion of the time evolved wave-function makes it pos-

sible to efficiently evaluate most observables. Also, the result from Equation 78 for $\hat{\mathcal{O}} = \mathcal{H}_0^{S^2}$ holds, as $E_0(N, t)$ is real-valued.

$$\hat{\mathcal{O}} = \hat{V}^{S^2} : \quad \hat{\mathcal{O}}_{\text{loc}}(N, t) \overset{80,76}{=} |E_V(N, t)|^2 \tag{81}$$

So now, to get the full expectation value for the energy fluctuations, it is only necessary to measure $E_0(N, t)$ and $E_V(N, t)$ for each sampled state – making it exactly as expensive to compute as the energy itself. Then however, it requires aggregating $E_0(N, t)$, $E_V(N, t)$, $(E_0(N, t))^2$, $|E_V(N, t)|^2$ and $E_0(N, t) \cdot E_V(N, t) + [E_0(N, t) \cdot E_V(N, t)]^*$ separately (scaled with the probability) for all sampled states to be able to combine all terms according to Equation 77 in the end.

2.3 Variational Classical Networks

When performing time-evolution with a perturbative approach, one ought to watch out for resonances. In the case of degenerate energy solutions, so-called *secular terms* [34] might be introduced, if non-degenerate perturbation theory is used to find the solution to a degenerate problem. Often inaccuracies manifest as linear terms in a function where only oscillating terms would be expected. The linear term constantly grows over time, guaranteeing the approximation is only valid for short times and then diverges further and further from the exact solution.

For the examined Hamiltonian the first occurrence of such terms can be spotted in the second order cumulant expansion of the Hamiltonian. Equation 48 and Equation 49 specifically list extra solutions for the cases where the two energies are the same. This resonance causes a linear term to appear in the expansion, while all other pre-factors so far were complex exponentials – meaning they are oscillating.

The idea to combat this is to replace the perturbative parameters that were calculated analytically with variational parameters instead. This has the motive to generate a physically motivated structure, but the fine details of the parameter weights are adjusted by optimization. E.g. for the *neural-network quantum state* [35] approach, a neural network structure is used to represent a wave-function. Reason for this being the hope that the exponentially complicated wave-function can be expressed as a "higher truth" only requiring few parameters to describe a seemingly complicated wave-function. While it would be basically impossible to always find the few-parameter-solution, *approximating* it close enough with optimization akin to machine-learning should be a suitable replacement.

The following section aims to achieve better approximations than the ones found by cumulant expansion by letting the parameters be optimized during the step to reflect the situation more closely.

2.3.1 General Theory of VCN

For this application, the way of approximating the wave-function is using a *classical network*. The network parameters now should be obtained via the *time-dependent variational principle*

(TDVP) [36]. For this reason, the method as a whole is dubbed a *variational classical network* (VCN).

In [22] this strategy is used on comparable applications on similar quantum problems. The research in [37] even applies the method to two-dimensional geometries. Both reach the same expressions for the advancement of the parameters to later times. This section reiterates the equations, unifying the parts where different conventions might be used.

$\mathcal{H}_N(t)$ from Equation 32 should be approximated with a parametrized expression $\mathcal{H}_{\text{VCN}}(N, \vec{\eta}(t))$. This means – compared with Equation 30 – the effective Hamiltonian is now expressed as $\mathcal{H}_{\text{eff}}(N, \vec{\eta}, t) = -iE_0(N)t + \mathcal{H}_{\text{VCN}}(N, \vec{\eta}(t))$. The Hamiltonian depends on the choice of the complex parameter vector $\vec{\eta}(t)$.

Historically, the TDVP was derived for real-valued parameters [36]. To extend the principle to complex parameters, the networks structure could be adapted through the introduction of two real-valued parameters per complex parameter [38]. For *holomorphic* functions the following also follows from an action-principle [37] – or in short: for fully complex differentiable [39] parametrizations no special treatment is required.

Generally, then the following method is applied: To begin with, $\vec{\eta}(t)$ is initialized to a known value for $t = 0$ to try to achieve $\mathcal{H}_{\text{VCN}}(N, \vec{\eta}(t))|_{t=0} \approx 0$. The initialized parameters should additionally have some small noise added to them to prevent unwanted dead-points like an unstable equilibrium. The variational nudge $\dot{\vec{\eta}}$ to the parameters $\vec{\eta}$ then is calculated so that the overlap of the states in Equation 82 is maximized.

$$
\begin{aligned}
\left| \Psi_{\vec{\eta}+\delta\cdot\dot{\vec{\eta}}}(t+\delta) \right\rangle \quad &\leftrightarrow \quad e^{-i\mathcal{H}\delta} \left| \Psi_{\vec{\eta}}(t) \right\rangle \\
\left| \Psi_{\vec{\eta}}(t) \right\rangle &= \sum_N e^{-iE_0(N)t + \mathcal{H}_{\text{VCN}}(N, \vec{\eta}(t))} \Psi_N \left| N \right\rangle \\
&\stackrel{24}{\Rightarrow} \Psi_{N, \vec{\eta}}(t) = e^{-iE_0(N)t + \mathcal{H}_{\text{VCN}}(N, \vec{\eta}(t))} \Psi_N
\end{aligned}
\tag{82}
$$

This method tries to find the update to the parameters that most accurately reflects what would happen if the system was time-evolved for a small time-step δ with the full Hamiltonian. With a set of starting parameters for $t = 0$ and an update-strategy for small time-steps, generating the parameters for later times is possible with repeated stepping. The mentioned overlap is calculated with a measurement that uses the *Fubini-Study metric* ([22] and [37], where [37] contains a detailed derivation in the supplementary material).

Solving for maximum overlap yields the *TDVP-equation* (Equation 83), using the *variational derivative* (Equation 84) and the *local energy* (Equation 85), which both are required for calculating the *covariance matrix* (Equation 86) and the *TDVP-force* (Equation 87). The $\langle * \rangle_{(\vec{\eta}(t))}$ denotes the expectation values with respect to the normalized probability distribution over all the states $\left| \Psi_{N, \vec{\eta}}(t) \right\rangle$ [37]. Meaning the observable-sampling-expression from Equation 53 may be used as-is, if $\mathcal{H}_{\text{eff}}(N, t)$ is set to the parametrized version $\mathcal{H}_{\text{eff}}(N, \vec{\eta}, t)$.

$$
\begin{aligned}
\sum_{k'} \overleftrightarrow{S}_{k, k'} \dot{\vec{\eta}}_{k'} &= -i\vec{F}_k \\
\dot{\vec{\eta}} &= -i\overleftrightarrow{S}^{-1}\vec{F}
\end{aligned}
\tag{83}
$$

All of the previous and following steps are performed for one fixed time t – if not stated otherwise. For that reason, one can write $\vec{\eta}$ for $\vec{\eta}(t)$, as the specific time-point t is implied until explicitly referenced. Here, $*^{-1}$ is the pseudo inverse of a square matrix (that might not be invertible normally, as it might not have full rank or very small eigenvalues making the inversion numerically instable).

$$
O_k(N) = \frac{\partial \ln \Psi_{N,\vec{\eta}}(t)}{\partial \eta_k}
$$

$$
\overset{82}{=} \frac{\partial \ln}{\partial \eta_k} e^{-iE_0(N)t + \mathcal{H}_{\mathrm{VCN}}(N,\vec{\eta}) + \ln \Psi_N} = \frac{\partial \mathcal{H}_{\mathrm{VCN}}(N,\vec{\eta})}{\partial \vec{\eta}_k}
\tag{84}
$$

The Ψ_N can be expressed as an exponential (which cancels with the ln) and then all terms not depending on variational parameters can easily be differentiated to 0.

$$
E_{\mathrm{loc}}(N) = \frac{\langle N|\mathcal{H}|\Psi_{\vec{\eta}}(t)\rangle}{\langle N|\Psi_{\vec{\eta}}(t)\rangle}
$$

$$
\overset{76}{=} E_0(N,t) - J \cdot \sum_{\langle l,m\rangle, \sigma} \left(n_{l,\sigma} \neq n_{m,\sigma}\right) e^{\mathcal{H}_{\mathrm{eff}}(\tilde{N},\vec{\eta},t) - \mathcal{H}_{\mathrm{eff}}(N,\vec{\eta},t)} \frac{\Psi_{\tilde{N}}}{\Psi_N}
\tag{85}
$$

The local energy $E_{\mathrm{loc}}(N)$ here is the same as the local observable $\hat{\mathcal{O}}_{\mathrm{loc}}(N,t)$ for $\hat{\mathcal{O}} = \mathcal{H}^{\mathrm{S}}$, already calculated in subsection 2.2.6, but with a parametrized effective Hamiltonian.

$$
\overset{\leftrightarrow}{S}_{k,k'} = \left\langle O_k^* O_{k'}\right\rangle_{(\vec{\eta})} - \left\langle O_k^*\right\rangle_{(\vec{\eta})} \cdot \left\langle O_{k'}\right\rangle_{(\vec{\eta})}
\tag{86}
$$

$$
\vec{F}_k = \left\langle E_{\mathrm{loc}} O_k^*\right\rangle_{(\vec{\eta})} - \left\langle E_{\mathrm{loc}}\right\rangle_{(\vec{\eta})} \cdot \left\langle O_k^*\right\rangle_{(\vec{\eta})}
\tag{87}
$$

For the update-strategy the first-order *Euler's method* is chosen, meaning $\vec{\eta}(t+\delta) = \vec{\eta}(t) + \delta \cdot \dot{\vec{\eta}}(t)$. This could be improved by using the second-order *Heun's method*. Higher order approximators are most likely never worth it, as the expectation values for the local energy and variational derivative are sampled with Monte Carlo sampling for larger systems (see subsection 2.4.1). The observables will be noisy because of this, making it unnecessary to use a more exact stepping-procedure, as there is no use in stepping more precisely when the stepping direction is not exact. The added computational expense of higher order approximators would be put to better use by using smaller step sizes or a higher number of Monte Carlo samples.

2.3.2 Application to this Task

Depending on the chosen parametrization, different $\mathcal{H}_{\mathrm{VCN}}(N,\vec{\eta})$ might be obtained. This changes the value of the variational derivatives in Equation 84. The reference [40] states how to apply the method to *restricted Boltzmann machines* (RBM) [35]. In this thesis, a simple linear dependency on the parameters is chosen:

$$\mathcal{H}_N(t) = \sum_i C_i(t) \cdot \Phi_i(N) \quad \leftrightarrow \quad \mathcal{H}_{\text{VCN}}(N, \vec{\eta}) = \sum_i \eta_i \cdot \Phi_i(N) \tag{88}$$

Here, $C_*(t)$ are analytically calculated functions, while the η_* are complex parameters to replace them at time t. The choice of the $\Phi_i(N)$ dictates the range of interaction that could technically be covered. For the structure in Equation 88, the comparison with the cumulant expansion yields expressions for $\Phi_l(N)$ and $C_l(t)$ in a controlled manner. By identifying the terms in Equation 43, a possible choice for the first terms could be chosen like in Equation 89 (absolute lattice-site enumeration like in Figure 2.1).

$$
\begin{aligned}
C_1(t) &= \Pi_{\text{A}}(0,1,t) & \Phi_1(N) &= J \sum_l \sum_{\langle l \overset{+(x)}{<} m \rangle} \sum_K \frac{\Psi_K}{\Psi_N} \langle N|\hat{\text{F}}_{\text{A}}(l,m)|K\rangle \\[1em]
C_2(t) &= \Pi_{\text{B}}(0,1,t) & \Phi_2(N) &= J \sum_l \sum_{\langle l \overset{+(x)}{<} m \rangle} \sum_K \frac{\Psi_K}{\Psi_N} \langle N|\hat{\text{F}}_{\text{B}}(l,m)|K\rangle \\[1em]
C_3(t) &= \Pi_{\text{C}}(0,1,t) & \Phi_3(N) &= J \sum_l \sum_{\langle l \overset{+(x)}{<} m \rangle} \sum_K \frac{\Psi_K}{\Psi_N} \langle N|\hat{\text{F}}_{\text{C}}(l,m)|K\rangle \\[1em]
C_4(t) &= \Pi_{\text{A}}(1,0,t) & \Phi_4(N) &= J \sum_l \sum_{\langle l \overset{+(x)}{>} m \rangle} \sum_K \frac{\Psi_K}{\Psi_N} \langle N|\hat{\text{F}}_{\text{A}}(l,m)|K\rangle \\[1em]
C_5(t) &= \Pi_{\text{B}}(1,0,t) & \Phi_5(N) &= J \sum_l \sum_{\langle l \overset{+(x)}{>} m \rangle} \sum_K \frac{\Psi_K}{\Psi_N} \langle N|\hat{\text{F}}_{\text{B}}(l,m)|K\rangle \\[1em]
C_6(t) &= \Pi_{\text{C}}(1,0,t) & \Phi_6(N) &= J \sum_l \sum_{\langle l \overset{+(x)}{>} m \rangle} \sum_K \frac{\Psi_K}{\Psi_N} \langle N|\hat{\text{F}}_{\text{C}}(l,m)|K\rangle \\[1em]
C_7(t) &= \Pi_{\text{A}}(0,M,t) & \Phi_7(N) &= J \sum_l \sum_{\langle l \overset{+(y)}{<} m \rangle} \sum_K \frac{\Psi_K}{\Psi_N} \langle N|\hat{\text{F}}_{\text{A}}(l,m)|K\rangle \\[1em]
&\quad\cdots & &\qquad\cdots
\end{aligned}
\tag{89}
$$

The dissection separates the structure into 12 different factors. The terms $C_1(t)$ to $C_6(t)$ describe the neighbor interaction in x-direction, the terms $C_7(t)$ to $C_{12}(t)$ are identical with the interactions in y-direction (not printed, as they are identical with the x swapped for y and the 1 swapped for an M). The neighbor convention was already introduced earlier. To reiterate, $\langle l \overset{+(x)}{<} m \rangle$ means "all sites m that have an x-coordinate that is strictly larger than the one of site l and are nearest-neighbors of l". For the chain-geometry only the terms 1 through 6 are relevant, as there are no neighbors in y-direction.

Combining all this, one arrives at Equation 90 for the first terms of the variational derivatives.

$$O_k(N) \overset{84,88}{=} \Phi_i(N) \tag{90}$$

For that reason and the fact that the calculation of $\mathcal{H}_{\text{eff}}(\widetilde{N}, \vec{\eta}, t) - \mathcal{H}_{\text{eff}}(N, \vec{\eta}, t)$ mostly boils down to calculating differences of $\Phi_*(\widetilde{N}) - \Phi_*(N)$, the $\Phi_*(N)$ and differences thereof need to be evaluated most efficiently.

2.3.3 Explicit Time-Dependency

The method described so far seems to be the minimal viable parametrization at first. Of course it should be possible to introduce more terms that are inspired from higher order perturbation theory. Currently, for the two-dimensional grid the last section suggested 12 parameters. Considering that half of them can be expressed as complex conjugates of some other, this leaves only 6 distinct parameters. While this does not provide many degrees of freedom to encode information, it should be sufficient for some basic calculations and a verification of the method. At least for validating the hypothesis it should be better than the first-order model without variational parameters.

In previous sections, the TDVP-equation was derived from maximizing the overlap of a variational state that has its parameters numerically integrated to a future time and the same state being time-evolved with the Hamiltonian for a small time-step. However, as [38] shows, the TDVP-formalism can also be derived from an *action principle*. And as *Noether's theorem* states: if a method keeps the *Lagrangian invariant* (like the mentioned principle that derives parameter-variations, which are symmetries of the action), the system's energy must be *exactly* conserved over time [41] (independent of how "bad" the parametrization is).

In order to adhere to the structure of the thesis, at this point some numerical results are revealed in advance. As experiments in *section 4.4 Insufficient Variational Ansatz* show, this simple parametrization does not result in the energy being conserved at all. Upon closer inspection, the resulting changes to the variational parameters seem to be more or less independent from the initial values. After implementation reasons for this behavior were ruled out to the best possible degree, it seemed unclear what was the cause for this.

Yet the algorithm *must* conserve the system's energy – for small enough time-steps – most importantly already at the very first variational time-step. This leads to the solution, because the only remaining thing to influence the effective Hamiltonian (at $t = 0$, where all introduced variational parameters are approximately zero) is the base-energy with its explicit time-dependency. Presenting this realization in this case out-of-order (presenting and doing experiments on a clearly still unviable model before presenting the correct one) is supposed to reflect the work-process that was necessary to arrive at this final version. In first theoretical discussions about the application of TDVP, the algorithm's explicit dependency on time in the base-energy case had been touched and even been presented in the correct form to solve this shortcoming. Yet, only after going through two alternate base-energy parametrizations unsuccessfully and arriving at the final working solution independently of this hint, the relevance of the statement became clear. In hindsight, it is straight forward to see why the explicit time-dependency contribution of the base-energy to the effective Hamiltonian is detrimental, when looking at equations 83 and 84.

Starting at a point t in time, it comes natural to numerically integrate a quantity $\vec{\eta}$ if one has access to the time-derivative $\dot{\vec{\eta}}$. In the method up to this point through the partial derivative with respect to the variational parameters: $\frac{\partial \mathcal{H}_{\text{eff}}(N, \vec{\eta}, t)}{\partial \vec{\eta}_k} = \frac{\partial\left(-iE_0(N)t + \mathcal{H}_{\text{VCN}}(N, \vec{\eta})\right)}{\partial \vec{\eta}_k} = \frac{\partial \mathcal{H}_{\text{VCN}}(N, \vec{\eta})}{\partial \vec{\eta}_k}$, the explicit time dependency is dropped. However classically the time-derivative is obtained as per the chain rule: $\frac{\mathrm{d}f(t, x(t))}{\mathrm{d}t} = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial x} \cdot \frac{\mathrm{d}x}{\mathrm{d}t}$, which rightfully does not drop the explicit dependency. So to avoid having to derive the algorithm again, but with explicit time-dependency in mind, the most direct strategy would be to eliminate all explicit time-dependencies. This would result in a completely variational effective Hamiltonian $\mathcal{H}_{\text{eff}}^{\text{VCN}}(N, \vec{\eta})$, where all that is time-dependent is encoded in the current state of $\vec{\eta}$.

Following the linear strategy of Equation 88, the resulting structure is presented in Equation 91.

$$\mathcal{H}_{\text{eff}}^{\text{VCN}}(N, \vec{\eta}) = \sum_{n=1}^{\#(\text{var.})} \eta_n^{\text{var.}} \cdot \Phi_n^{\text{var.}}(N) + \sum_{l=0}^{L} \eta_l^{\text{b.e.}} \cdot \Phi_l^{\text{b.e.}}(N) = \sum_l \eta_l \cdot \Phi_l(N) \tag{91}$$

$$\vec{\eta} = \left(\eta_1^{\text{var.}}, \eta_2^{\text{var.}}, \dots, \eta_{\#(\text{var.})}^{\text{var.}}, \eta_0^{\text{b.e.}}, \eta_1^{\text{b.e.}}, \dots, \eta_L^{\text{b.e.}}\right)^{\text{T}}$$

The $\eta_n^{\text{var.}}$ and $\Phi_n^{\text{var.}}(N)$ are the same as in subsection 2.3.2 and still depend on the chosen parametrization of $\mathcal{H}_N(t)$. When compared with Equation 42, Equation 92 presents itself as an ideal choice for $\eta_l^{\text{b.e.}}$ and $\Phi_l^{\text{b.e.}}(N)$, as it upholds $\sum_{l=0}^{L} \eta_l^{\text{b.e.}} \cdot \Phi_l^{\text{b.e.}}(N) = -i \cdot E_0(N) \cdot t$ for a lattice with L sites.

$$E_0(N) \overset{42}{=} U \cdot \sum_m n_{m,\uparrow} n_{m,\downarrow} + \sum_{l,\sigma} \varepsilon_l n_{l,\sigma}$$

$$\eta_l^{\text{b.e.}} = \begin{cases} l = 0 : & -i \cdot U \cdot t \\ l > 0 : & -i \cdot \varepsilon_l \cdot t \end{cases} \tag{92}$$

$$\Phi_l^{\text{b.e.}}(N) = \begin{cases} l = 0 : & \sum_m n_{m,\uparrow} n_{m,\downarrow} \\ l > 0 : & \sum_\sigma n_{l,\sigma} \end{cases}$$

Other partitions of the parameters are also possible, e.g. having multiple parameters for the encoding of the double-occupation term. With this modification, the equations in subsection 2.3.1 can be used as derived and the explicit time-dependency is eliminated.

2.4 Time-Complexity Optimizations

While at this point all computations are analytically correct, some of the expressions are still unnecessarily expensive to evaluate. By further simplifying terms that can be known beforehand to cancel, the computation can be sped up drastically. So far, when final runtime-complexities have been stated, it was assumed that the optimizations in this section would be implemented.

2.4.1 Monte Carlo Sampling

Mathematically, the problem of finding the solution to such a quantum-mechanical many-body model is solved by the calculations provided up to this point. However, Equation 53 reveals this not viable with larger systems in practice. The calculation requires summing over all base-states. Their number is given in Equation 93 (factor 2 in the exponent for the spin-degree) to be of exponential size in regards to the number of lattice-sites. Considering a physical material has at least in the order of 10^{23} (from *Avogadros-Constant*) degrees of freedom, an $\mathcal{O}(2^{\#(\text{sites})})$ computational-complexity is not good enough to simulate interestingly-sized systems.

$$\#(\text{states}) = 2^{\#(\text{sites})\cdot 2} \tag{93}$$

Different strategies exist to circumvent this limitation. Here, the focus will be put onto randomized algorithms that use *Monte-Carlo sampling* in order to generate a limited number of states, representative of the real probability-distribution. This method avoids having to sum over the complete *Hilbert-Space*. The strategy of solving problems of this kind with the strategy of *variational Monte-Carlo* is e.g. suggested in [42]. This means starting with a random state and applying modifications to it (= variation). When the accepting/rejecting of these modifications is coupled to the probability the state has, based on its energy in the system's energy-distribution, the resulting states will have an energy/probability distribution that resembles the one they had if they were sampled from the full Hilbert-Space. The derivation for the whole process is shown in [26].

However, computing the normalized probability for the states is still required to compare against a (e.g. thermal) reference-distribution. As can be seen in the probability in Equation 94, that was defined in Equation 53, still a full sum over all states is required to get the normalization factor.

$$\mathrm{P}(N,t) = \frac{\left|e^{\mathcal{H}_{\text{eff}}(N,t)}\right|^2 |\Psi_N|^2}{\sum\limits_K \left|e^{\mathcal{H}_{\text{eff}}(K,t)}\right|^2 |\Psi_K|^2} \tag{94}$$

However, by employing the *Metropolis-Hastings-Algorithm* instead of only the *Metropolis-Algorithm* to decide on the acceptance/rejection of proposed states, the requirement of normalizing the probability can be dropped. In Equation 96 the *acceptance-ratio* α – that is necessary for the Metropolis-Hastings-Algorithm to work – is derived (as per [43]) to decide on the transition from state $|N\rangle$ to $|\widetilde{N}\rangle$. The algorithm proposes to generate a new state $|\widetilde{N}\rangle$ and calculate the corresponding α. Then it requires generating a random number u between 0 and 1, pulled from a uniform distribution. If $u \leq \alpha$, $|\widetilde{N}\rangle$ is accepted and will be used as the new current state in the next step of the sampling. If $u > \alpha$, the proposed state is rejected and $|N\rangle$ will be used again as the next state. The function $f(N,t)$, provided in Equation 95, fulfills the necessary property of being proportional to the probability $\mathrm{P}(N,t)$.

$$f(N,t) = \left|e^{\mathcal{H}_{\text{eff}}(N,t)}\right|^2 |\Psi_N|^2 \propto \mathrm{P}(N,t) \tag{95}$$

$$
\begin{aligned}
\alpha = \frac{\mathrm{P}(\widetilde{N},t)}{\mathrm{P}(N,t)} &= \frac{f(\widetilde{N},t)}{f(N,t)} \overset{95}{\doteq} \frac{\left|e^{\mathcal{H}_{\mathrm{eff}}(\widetilde{N},t)}\right|^2 |\Psi_{\widetilde{N}}|^2}{\left|e^{\mathcal{H}_{\mathrm{eff}}(N,t)}\right|^2 |\Psi_N|^2} \\
&= \frac{|\Psi_{\widetilde{N}}|^2}{|\Psi_N|^2} \frac{e^{\Re(\mathcal{H}_{\mathrm{eff}}(\widetilde{N},t))+i\Im(\mathcal{H}_{\mathrm{eff}}(\widetilde{N},t))+\Re(\mathcal{H}_{\mathrm{eff}}(\widetilde{N},t))-i\Im(\mathcal{H}_{\mathrm{eff}}(\widetilde{N},t))}}{e^{\Re(\mathcal{H}_{\mathrm{eff}}(N,t))+i\Im(\mathcal{H}_{\mathrm{eff}}(N,t))+\Re(\mathcal{H}_{\mathrm{eff}}(N,t))-i\Im(\mathcal{H}_{\mathrm{eff}}(N,t))}} \\
&= \frac{|\Psi_{\widetilde{N}}|^2}{|\Psi_N|^2} e^{2\cdot\Re(\mathcal{H}_{\mathrm{eff}}(\widetilde{N},t))-2\cdot\Re(\mathcal{H}_{\mathrm{eff}}(N,t))} \\
&= \frac{|\Psi_{\widetilde{N}}|^2}{|\Psi_N|^2} e^{2\cdot\Re(\mathcal{H}_{\mathrm{eff}}(\widetilde{N},t)-\mathcal{H}_{\mathrm{eff}}(N,t))}
\end{aligned}
\tag{96}
$$

Multiple factors must be considered which kind of variation is applied to the states in this schema. However, probably the most important criteria is how quickly Equation 96 can be evaluated, as this will be done tens to hundreds of times between each sampled state.

Finally, the method of sampling a selection of distributed states results in Equation 97 as a rewrite to Equation 53 [26].

$$
\frac{\langle \Psi^{\mathrm{S}}(t)|\hat{\mathcal{O}}|\Psi^{\mathrm{S}}(t)\rangle}{\langle \Psi^{\mathrm{S}}(t)|\Psi^{\mathrm{S}}(t)\rangle} \overset{53}{\doteq} \sum_N \mathrm{P}(N,t)\cdot\hat{\mathcal{O}}_{\mathrm{loc}}(N,t) \overset{[26]}{\approx} \frac{1}{|\{N\}_{\mathrm{MC}}|} \sum_{\{N\}_{\mathrm{MC}}} \hat{\mathcal{O}}_{\mathrm{loc}}(N,t)
\tag{97}
$$

2.4.2 Analytical Simplifications for Special Cases

Generally, handling the computation of the difference of the effective Hamiltonian for two states $\mathcal{H}_{\mathrm{eff}}(\widetilde{N},t)-\mathcal{H}_{\mathrm{eff}}(N,t)$ is the most expensive computation that is required up to this point. For one, it is e.g. required in Equation 61 for the calculation of the spin-polarized kinetics. Furthermore – as stated in *subsection 2.4.1 Monte Carlo Sampling* – it is also needed to compute the transition probability between two sampled states in Equation 96. The calculation of $\mathcal{H}_{\mathrm{eff}}(N,t)$ on its own requires an effort of $\mathcal{O}(\#(\text{sites})\cdot\#(\text{nearest neighbors}))$. However, for states $|N\rangle$ and $|\widetilde{N}\rangle$ that are connected with only small changes, most of the elements in the sum cancel and the complexity may here even be reduced to $\mathcal{O}(\#(\text{nearest neighbors}))$. This is quite attractive, because it decouples the per-step computational costs from the number of lattice sites – a vital step for simulating large systems efficiency – however this requires extra analytical calculations.

In this section analytical simplifications are presented to get the objects $E_0(N) - E_0(\widetilde{N})$ and $\mathcal{H}_N(t) - \mathcal{H}_{\widetilde{N}}(t)$. Caution, do not miss the extra *minus* required, to calculate the required $\mathcal{H}_{\mathrm{eff}}(\widetilde{N},t) - \mathcal{H}_{\mathrm{eff}}(N,t)$ from this!

Evidently, this also needs to hold for the variational versions of the effective Hamiltonian that were introduced in subsection 2.3.3. As the $\Phi_*(N)$ are inspired by the form of the base-energy-terms in $E_0(N)$ or parts of $\mathcal{H}_N(t)$, all the same arguments apply: $\mathcal{H}_{\mathrm{eff}}^{\mathrm{VCN}}(\widetilde{N},\vec{\eta}) - \mathcal{H}_{\mathrm{eff}}^{\mathrm{VCN}}(N,\vec{\eta}) = \sum_l \eta_l \cdot [\Phi_l(\widetilde{N}) - \Phi_l(N)]$. Yet, all of the re-writes must be done precisely to produce the correct results.

Initial State

The initial state of the system before the start of the time-evolution is encoded in Ψ_N, defined by Equation 24. Different configurations can be considered here. Logically, the best choice for this would be the one that saves the most computational effort and provides the biggest symmetry for the terms, to promote the possibility of terms canceling.

A choice that fulfills these requirements trivially would be the perfectly uniform distribution of probability for all the base-states. For most of the sources that describe similar methods (e.g. like [24] or [22], where the latter describes how the approach also works for other initial states) this initial state has been chosen for its advantages.

The calculation of the uniform probability is straight-forward and listed in Equation 98, while the state would be written down like Equation 99.

$$\Psi_N = \frac{1}{\sqrt{\#(\text{states})}} \overset{93}{=} \frac{1}{\sqrt{2^{\#(\text{sites})\cdot 2}}} = \frac{1}{2^{\#(\text{sites})}} \tag{98}$$

$$\left|\Psi^S(t=0)\right\rangle = \bigotimes_{l=1}^{\#(\text{states})} \frac{1}{2}\left(1 + \hat{h}_{l,\uparrow}^{\dagger S} + \hat{h}_{l,\downarrow}^{\dagger S} + \hat{h}_{l,\uparrow}^{\dagger S}\hat{h}_{l,\downarrow}^{\dagger S}\right)|0\rangle \tag{99}$$

Central advantage of the uniform distribution of base-state-probabilities is that all Ψ_N are equal, which makes it possible to cancel most terms from the sums, as $\Psi_N/\Psi_{\tilde{N}} = 1$. The following two optimizations require this assumption.

As the states under consideration are $|N\rangle$ and $|\widetilde{N}\rangle$, the occupation-numbers $n_{l,\sigma}$ will describe the occupation of $|N\rangle$ and $\widetilde{n}_{l,\sigma}$ the occupation of $|\widetilde{N}\rangle$. Both of the following modifications restrict the values $\widetilde{n}_{l,\sigma}$ to make them dependent on $n_{l,\sigma}$.

Single Flip Modification

In this simplification, the maximum difference that can happen is that a *flipping* event occurs on site i and spin σ_i. In the language of hard-core bosons, this means one occupation turning form a 0 into a 1 or the inverse of that. This operation does not conserve the number of particles, so depending on the application this might not be a desired modification for a Monte-Carlo-step. Still, the calculation of the reduced density matrix as described in subsection 2.2.4 requires calculation of the effective Hamiltonian difference between two states connected by such a modification.

The flipping results in the values for $\widetilde{n}_{l,\sigma}$ are given in Equation 100.

$$\widetilde{n}_{l,\sigma} = \begin{cases} l = i \wedge \sigma = \sigma_i : & (1 - n_{i,\sigma_i}) \\ \text{else:} & n_{l,\sigma} \end{cases} \tag{100}$$

This simplifies the energy difference $E_0(N) - E_0(\widetilde{N})$ like equation Equation 101 presents.

$$E_0(N) - E_0(\widetilde{N}) = U \sum_l n_{l,\downarrow} n_{l,\uparrow} - U \sum_l \widetilde{n}_{l,\downarrow} \widetilde{n}_{l,\uparrow} + \sum_{l,\sigma} \varepsilon_l n_{l,\sigma} - \sum_{l,\sigma} \varepsilon_l \widetilde{n}_{l,\sigma}$$

$$\overset{100}{=} \varepsilon_i \left(2n_{i,\sigma_i} - 1 \right) + U \cdot \begin{cases} \sigma_i = \uparrow: & n_{i,\downarrow}(2n_{i,\uparrow} - 1) \\ \sigma_i = \downarrow: & n_{i,\uparrow}(2n_{i,\downarrow} - 1) \end{cases} \tag{101}$$

The optimized calculation of $\mathcal{H}_N(t) - \mathcal{H}_{\widetilde{N}}(t)$ or the derived $\Phi_l(N) - \Phi_l(\widetilde{N})$ is more involved to get correct. Still, the process is mathematically straight forward. For hints and further resources on how this was implemented, see *section 3.2 Python-Script Generation*. How the correctness of the optimizations is assured is briefly touched in *section 3.4 Simplification Verifications*. Finally, a generalized (if maybe not optimally efficient) method is outlined in *subsection 2.4.3 Geometry-Dependent Interaction-Range*.

Hopping, Swapping or Double-Flip Modification

A possible Monte-Carlo-modification that conserves particle number is the change that occurs when a particle from a specific site and spin (occupation 1, then 0) is transferred to a different site and spin combination that currently is un-occupied (occupation 0, then 1). This so-called *hopping* event occurs in this example from site i and spin σ_i to the site j and spin σ_j.

The hopping is a subset of the *swapping* modification that occurs in the case of two occupation-numbers being exchanged. The difference is that, while swapping two identical occupations is possible, having exactly one particle on the relevant two sites is a pre-requirement for hopping. Depending on the use-case, this is a relevant difference, yet in case of calculation of the differences of effective Hamiltonians it is irrelevant: On comparison with e.g. Equation 61, one can see that the term containing $\Delta = \mathcal{H}_{\text{eff}}(\widetilde{N}, t) - \mathcal{H}_{\text{eff}}(N, t)$ is multiplied with 0 in the cases where hopping is not possible.

Furthermore, the only way one could possibly get a contribution by the difference of effective Hamiltonians is in the case where $n_{i,\sigma_i} \neq n_{j,\sigma_j}$. Because if they are the same, swapping does not change the state and a check can be performed beforehand, to save on computational resources (if $|\widetilde{N}\rangle = |N\rangle$, trivially $\Delta = \mathcal{H}_{\text{eff}}(\widetilde{N}, t) - \mathcal{H}_{\text{eff}}(N, t) = 0$).

Thirdly, one could also think about an event where two occupation-numbers are flipped simultaneously on two different site- / spin-indices, or *double-flipping*. In the case where hopping is possible, this is yet again the exact same operation (the 0 changes to a 1 and the 1 on the second site to a 0). Differences are shown in Table 2.1.

For the case of swapping, Equation 102 shows the base-energy difference $E_0(N) - E_0(\widetilde{N})$ – the double flipping implementation would be constructed analogously to Equation 101.

Type	n_{i,σ_i}	n_{j,σ_j}	$\widetilde{n}_{i,\sigma_i}$	$\widetilde{n}_{j,\sigma_j}$	Δ
hopping	0	0	×	×	×
	0	1	1	0	Δ_1
	1	0	0	1	Δ_2
	1	1	×	×	×
swapping	0	0	0	0	0
	0	1	1	0	Δ_1
	1	0	0	1	Δ_2
	1	1	1	1	0
double-flipping	0	0	1	1	Δ_3
	0	1	1	0	Δ_1
	1	0	0	1	Δ_2
	1	1	0	0	Δ_4

Table 2.1: Basic tabular depiction of what the three different modification-operations do to the occupation-numbers of a state. For a difference of effective Hamiltonians Δ a placeholder value is listed. Where the exact value is known to be zero, this is noted, otherwise a variable is listed. The same variable indicates the same value, or simply put, in these cases the methods are exchangeable.

$$
\begin{aligned}
E_0(N) - E_0(\widetilde{N}) &= U \sum_l n_{l,\downarrow} n_{l,\uparrow} - U \sum_l \widetilde{n}_{l,\downarrow} \widetilde{n}_{l,\uparrow} + \sum_{l,\sigma} \varepsilon_l n_{l,\sigma} - \sum_{l,\sigma} \varepsilon_l \widetilde{n}_{l,\sigma} \\
&= \left(\varepsilon_i - \varepsilon_j\right)\left(n_{i,\sigma_i} - n_{j,\sigma_j}\right) + U \cdot \begin{cases} \sigma_i = \sigma_j : & \left(n_{i,\uparrow} - n_{j,\uparrow}\right)\left(n_{i,\downarrow} - n_{j,\downarrow}\right) \\ \sigma_i \neq \sigma_j : & \left(n_{i,\uparrow} - n_{j,\downarrow}\right)\left(n_{i,\downarrow} - n_{j,\uparrow}\right) \end{cases} \\
&= \left(\varepsilon_i - \varepsilon_j\right)\left(n_{i,\sigma_i} - n_{j,\sigma_j}\right) + U \left(n_{i,\sigma_i} - n_{j,\sigma_j}\right)\left(n_{i,\overline{\sigma_i}} - n_{j,\overline{\sigma_j}}\right)
\end{aligned} \tag{102}
$$

To get the simplest possible implementation for more complicated terms, it is advised to implement double-flipping first. It is the simplest of the three and (with the correct pre-factors that restrict the starting occupation-numbers) can be used to implement the other two. This is verified in [20]: `/computation-scripts/compareobservables.py`.

2.4.3 Geometry-Dependent Interaction-Range

While the optimizations described in the previous section greatly increase computational efficiency, they need to be applied for each $\mathcal{H}_N(t)$-order and each $\Phi_*(N)$ separately. The problem is that for two dimensions the number of interactions to consider grows quickly, while visualizing the symmetries of what terms are equivalent gets progressively harder. When looking at a chain, each higher order expands the interaction range by a one-site-step. So taking one more order into account, two sites are added to the possibilities for each interaction – this linearity is still somewhat manageable. In two dimensions, the number of affected sites in relation to the interaction range grows quadratically, as visualized in Figure 2.4.
As show in *subsection 2.4.2 Analytical Simplifications for Special Cases*, the described methods require calculating the differences between the effective Hamiltonians on two states that only
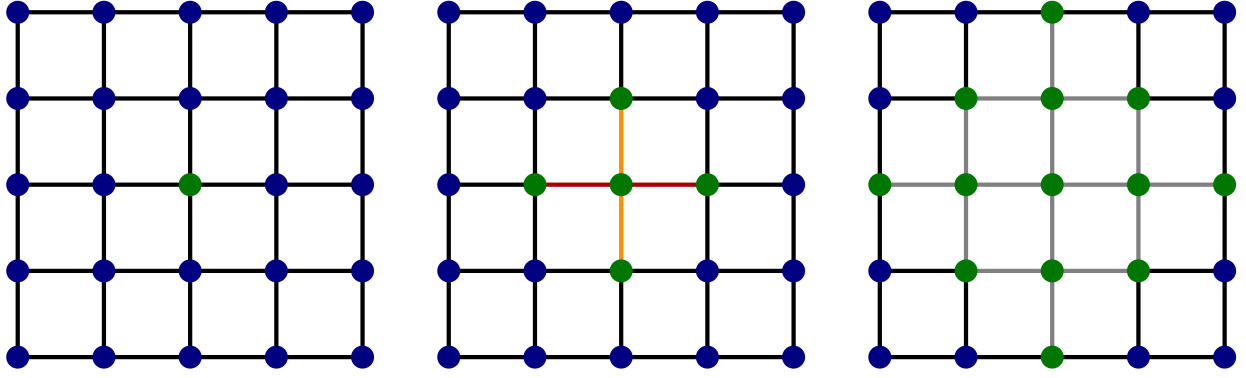
Figure 2.4: A depiction of the relevant sites and interactions that might be affected when a modification-event (e.g. a single flip on the center site) occurs. From left to right, the graphic shows the case for *base-energy only*, *first order perturbation theory* and *second order perturbation theory*. In case of a modification to the center-site the occupation of all green sites possibly influences the outcome of the calculation (in the corresponding order). For the first and second order the terms can be identified with their corresponding colored edges.

differ in localized modifications. Each modification flips the occupation of at least one site. To guarantee the differences of the effective Hamiltonians are evaluated correctly, only the terms that contain a modified site-occupation need to be evaluated. The terms that only contain non-modified occupation-numbers are guaranteed to cancel.

For the zeroth-order (base-energy) this is swiftly calculated and can be written down in simple terms like in e.g. equations 100 or 102. In the case of the first order, it is simple enough to identify the terms. Looking back at Equation 43, it becomes clear that all terms are described by *edges* of two adjacent sites. These relevant edges are colored red and orange in Figure 2.4. The symmetry is clear enough to spot that all horizontal and vertical edges respectively have the same analytical value. This is because the $\Pi_*(l, m, t)$ are translationally invariant and are complex conjugates of each other when inverting the indices – which is the reason for arriving at $2 \cdot 3 \cdot 2 = 12$ terms in Equation 89, of which 6 are identified in Table 2.2 (times two for horizontal/vertical versions).

l	$\uparrow$	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
l	$\downarrow$	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
m	$\uparrow$	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
m	$\downarrow$	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
$\Pi_*(l, m, t)$					A		C		A	C			B	A	A		
$\Pi_*(m, l, t)$			A	A	B		C	A		C		A					

Table 2.2: A list of all occupation-configurations for the first order terms. The indices of the two involved sites are l and m. Each hold a site for spin up and down particles $\uparrow$ and $\downarrow$. Not all 16 configurations have a representative term that results from the perturbation theory. The ones that do reference the letters A, B and C from the $\Pi_*(l, m, t)$ in Equation 43. A second line takes the other combination $l \leftrightarrow m$ into account.

Looking at all 16 occupation-number-combinations, the complexity is already extreme. Some combinations contribute to no terms, some contribute multiple times and some terms have more states contributing to them, some less. Identifying and analytically optimizing the first order

therefore is *just* possible, albeit complicated. For the variational parametrization, it would be possible and maybe even more efficient to just have 16 parameters, mapping one to one onto the occupation-number-combination of the respective bond. In second order, however, this all breaks down. Taking a second look at Figure 2.4, each term in Equation 51 now corresponds to *two pairs of connections*, of which are 34 different ones with the edges adjacent (assuming no edge-cases, because at the border all is different again) and even more disconnected ones that might or might not be cases from the first order. For identifying these, one must take rotational and mirror symmetries into account and the check for the border is vastly more complicated. Because each of these double-bounds is formed by 3 sites with each two spin-degrees, there are in total $2^{3 \cdot 2} = 64$ base-cases that need to be mapped. The numbers of base-cases 16 and 64 here are only comparably low, as they take symmetries into account. For brute-forcing all combinations without the manual identification of symmetries, one would get $2^{5 \cdot 2} = 1.024$ cases for the first order and $2^{13 \cdot 2} = 67.108.864$ – it would most definitely not be a viable strategy to convert each of these into their own parameter. Even if it was possible, this would diminish the physical bias we hope to get from defining the $\Phi_*(N)$ according to the cumulant expansion.

Even more problematic is the case of double-flips, because the interaction-spheres of these two modifications might be independent, overlapping slightly or even so much that both modifications are inside one bond. While not impossible for the second order, it is highly complicated to analytically compute the most efficient formula for the difference of effective Hamiltonians. And the process is not easily generalizable to higher orders because of order-specific symmetries.

Pre-Computed Interaction Spheres

In this case, this was solved by defining pre-computed sets of indices that must be taken into account.

Given a formula that needs to sum over all indices once (like Equation 43), or once over all indices for each index (like Equation 51 or Equation 79), it is clear that they require a runtime-complexity of $\mathcal{O}(\#(\text{sites}))$ or even $\mathcal{O}((\#(\text{sites}))^2)$ to evaluate. Yet all of them have so far also appeared as differences of two sums – one for the un-modified state and one for a state that had undergone a localized modification – of which most elements cancel.

One can assume it is possible to compute the neighbors of a site in constant time – still it takes computational time. This and the problem from before can be solved by taking time at the start of the program to generate a cache of all tuples of indices, which would appear in the sum that is being optimized, but only restricted to a pre-set range around the index of the modification – a *sphere of influence*. Because even if the generation takes $\mathcal{O}((\#(\text{sites}))^2)$ to generate this cache for a two-site-modification – after generation the lookup becomes practically $\mathcal{O}(1)$ (and is faster than computing the neighbor indices, even if all of that has asymptotically constant complexity). This means effectively always a constant number of indices needs to be taken into account for all calculations (which then can be evaluated in constant time). While it may be possible to analytically derive a more optimal solution with more canceling terms, this method easily scales. Only the radius of the pre-computed cache needs to be chosen to be large enough to be compatible with the used order of the expansion. These geometry-dependent caches are

generated in [20]: `/computation-scripts/systemgeometry.py` and the caches are used at various places.

The difference that this makes depends on the system size. As the number of influenced indices is constant, the strategy gets relatively more effective the more sites the system has (so that a comparably large portion of them is left out). For a chain, optimizations will be noticed comparably earlier than for the square lattice – this is because in two dimensions the sphere of influence grows as an area and not linearly at the two edges. But in the limit this strategy allows to keep the targeted complexity class, while scaling to the order of the expansion.

3 Implementation and Calculation Details

This part of the thesis is supposed to give a general overview about used methods for the implementation of the so-far theoretical methods. The entire codebase employs several strategies to arrive at a working and configurable end result and not all parts of it can be touched upon in detail. It also does not make sense to include most parts of the code in this thesis, it is best viewed in the provided reference [20]. Key parts and methods will be briefly presented, as while taking the same amount of work to create the code does not make for nearly as much content as the analytical theory.

3.1 Math-Manipulator

As stated in *chapter 2 Theory*, some of the analytical calculations involving ladder-operators have been performed with the tool *Math-Manipulator* [14]. This tool was purpose-built to play with operators that obey specific commutation-relations in a way which lies in between writing on paper and using a *computer-algebra-system*. Originally, this started as a passion-project with the purpose of gaining knowledge about *lexing*, *parsing* and the handling of *operator trees* to gain deeper experience with the working of compilers [44]. After adding a LaTeX-renderer and full browser support to minimize the friction of adopting such a new tool into established workflows, it became clear that it could be useful to perform some of the necessary analytical calculations. Having encountered many exercise-problems with the main focus on ladder-operator commutation-relations over the course of the master curriculum, oftentimes the tediousness of such calculations started to obscure the real lessons of the exercises. The project tries to establish itself as a low-friction option that fits into manual "on-paper" workflows and is yet powerful, extensible and provides reproducible calculations.

Some sections in the practical training report [8]: `/practical-training-latex` are dedicated to outlining the implementation, giving a basic overview of the theoretical backbone and redirecting to further resources. [8]: `/project-work-presentation` is discussing advantages and disadvantages of classical "on-paper" calculation vs. fully strict computer-algebra-systems. As both of these reports devote their focus to painting a more detailed picture, there will be no in-depth presentation of the side-project to this thesis. In the case of further interest it is best to follow the linked resources.

As a summarizing statement to gauge the necessity and usefulness of this detour, only the next paragraph should be sufficient.

While there already exist better tools for similar use-cases, the introduction of this new one does not aim to replace, imitate or upstage any of them. The focus of Math-Manipulator is to position itself as a playground to experiment with the behavior of non-commutative operators and related mathematical objects, while reducing the expense on repetitive manual calculations and providing results that are easier to reproduce. During the development process and the

first-hand usage on an actual scientific problem, it helped making the operator behavior tangible. Even though the time spent on developing the tool might outweigh the time that would have been necessary for writing down the final version of the calculations, during re-calculations and iterations it already could save many hours. Furthermore, only having such a consistent iteration cycle made it possible to arrive at the cleanliness of the final presented calculations. And finally, the stored save-files of the calculations make it possible to share the equations that are not put into the final thesis in a standardized way – without resorting to inclusion of large and complex parts in the appendix and supplementary material.

3.2 Python-Script Generation

For the main implementation of the theoretically derived algorithm the scripting language Python was chosen. It is a standard pick for work in the scientific field and has a vast library of powerful packages to meet the needs of research applications. While the language is known to have worse performance for direct implementations than modern statically compiled languages, it provides automatic runtime-checks for errors like division by zero and automatic type-casting. That, together with good built-in support for plotting and other visualization packages makes the language a good choice for designing and debugging a first-draft implementation.

Not all scripts were written by hand though. Previously stated in *subsection 2.4.2 Analytical Simplifications for Special Cases*, the difference $\mathcal{H}_N(t) - \mathcal{H}_{\tilde{N}}(t)$ can be optimized from a simple canonical implementation to a case specific re-write, that however has constant time-complexity instead of it being linearly dependent on the number of lattice sites.

As described in the accompanying report [8]: `/practical-training-latex`, some calculations need to manage a quickly growing number of terms. The comparatively high base-complexity is caused by the two-dimensional geometry and two spin-degrees on each site – both of which make for a relatively high number of interactions that need to be taken into account. Exemplary, each of the three $\hat{F}_A(l, m)$, $\hat{F}_B(l, m)$ and $\hat{F}_C(l, m)$ is composed of a sum of four summands, as one can see in Equation 40. Each of those must be treated for four cases: $(\sigma_i = \uparrow \wedge \sigma_j = \uparrow)$, $(\sigma_i = \uparrow \wedge \sigma_j = \downarrow)$, $(\sigma_i = \downarrow \wedge \sigma_j = \uparrow)$ and $(\sigma_i = \downarrow \wedge \sigma_j = \downarrow)$ (treats the interacting spin-combinations that occur in *section 2.4.2 Hopping, Swapping or Double-Flip Modification*). This makes 16 differently-shaped terms, or combined with the pre-factors 48 terms to fully write down with possible differences for x- and y-direction not yet taken into account.

Instead of manually identifying more symmetries to arrive at a short and still general formulation the code for the functions that need to take all the cases into account was generated programmatically:

The generating script is [20]: `/calculation-helpers/simplificationtermhelper.py` and the generated script [20]: `/computation-scripts/analyticalcalcfunctions.py`. It is efficient to (during this step) reduce the number of operations as much as possible and avoid running unnecessary calculations on every iteration (e.g. $n_{l,\uparrow} + 1 - (1 + n_{l,\uparrow})$ which is definitely 0, no matter the value of $n_{l,\uparrow}$). For this reason, the terms were folded by trying to pre-evaluate to definite zeros or constant terms. This was performed automatically with the

Python package *SymPy* [45]. It is only possible to pre-calculate this information in the case of generated code – for general code the work would need to be done during runtime.

3.3 Programming Style Optimizations

As mentioned in the previous section, some code has been directly generated with scripts. Because of this, these files are not really implemented the way a human would implement them and a lot of repetitive and direct code might be produced, instead of optimizing for readability. Python as a scripting language is not compiled in a classical sense and does not even have *just in time compilation* (yet, as of early 2025). Because of that, code is executed in the way it is written and no micro-optimizations are applied beforehand.

The process of optimizing code for speed, before having a working version, is called *premature optimization* and is most of the time ill-advised. However when code is generated anyway, it lends itself to pick up a 10 % speedup through choosing the most advantageous coding paradigms for fast execution.

As the occupation numbers of hard-core bosons are limited to 0 and 1, all analytical operator-strings (like the ones in Equation 40) can be re-written as expressions of boolean logic. One test was performed, whether these logical expressions could be rephrased to be evaluated more quickly in a closed form. For the purpose of optimizing the boolean clause the logic minimizer *Espresso* [46] was employed. An excerpt of the resulting code is shown in the appendix at 7.3.

Boolean Logic optimized	23.12 s
Branch free optimized	22.02 s
Default implementation	18.8 s

Table 3.1: Results of a qualitative measurement of the speed of different programming styles. The units of time-measurement are arbitrary and not applicable to a context other than their direct comparison. For each of the provided options, an excerpt of the implementation is listed in *chapter 7 Appendix*. A runtime-difference above 10 % can be observed between the runs.

Another experimental change was to remove switches and code in a branchless style. Such an optimization is oftentimes applied by modern compilers to take advantage of the *pipelining* powers of modern CPUs and relieve the branch-predicting algorithms. The resulting version of the test-function can be found in the appendix at 7.2.

Both of theses versions are benchmarked against the straight-forward implementation which can be found at the end of the thesis: 7.1. Timing results in a direct comparison are listed in Table 3.1. Interestingly, both of the "optimizations" do in fact increase the runtime of the trial-function, while having the same output for identical inputs. For one, this rules out that Python makes changes to the code on its own (at least in the used configuration), as branch-rewrites and operator replacement are one of the primary ways (even just in time) compilers increase code throughput.

The strengths of Python – dynamical typing and automatic type-casting on runtime – might be the reason for this unexpected behavior. To allow for these features, the language performs

assertions and type-checks for any statement in the background. This leads to substantially more and further branching code being executed than is expected. Therefore, while replacing a seemingly processing-delay introducing if-operation with several logical operations should increase the performance for a low-level language, in the case of Python the in practice fastest strategy seems to be executing as few *lines* of code as possible.

For this reason, the generated scripts contain very deep branching – even though it seems inefficient in terms of best-practices – it has been tested to be faster in this application. As the dynamical typing and runtime-error-checking of Python are in the end no longer required (as the prototype code runs stable), the next logical step would be harnessing the runtime-speedup that could be gained by rewriting in a low-level compiled language.

3.4 Simplification Verifications

It is a central goal of this thesis to proof that the described methods are viable to solve the time-evolution on the specified Hamiltonian and geometry. While it is necessary to apply enough optimizations to allow for reasonable complicated experiment-sizes to be run in the scope of a master thesis, it is also necessary to guarantee that no errors are introduced through such modifications. As the runtime-optimized versions nearly always have a higher code-complexity than the canonical implementations, it might be easy to introduce small errors, which may not be obvious in the beginning.

To try to mitigate similar issues and give a reasonable guarantee of correctness for all code, the optimized and the un-optimized versions of the same functions are always implemented independently. When on comparison both paths give the same value, the likelihood of implementation of the proposed simplifications being correct is extremely high. Not only that, but if independent implementations produce the same result this can be taken as an overall validation of the whole application.

Most major building pieces therefore have been treated in this way and checked for bugs. E.g. the correctness of the effective Hamiltonian difference is checked to be correct in [20]: `/computation-scripts/comparehamiltonians.py`. At the same location, with analogous naming convention, more self-validation scripts are included in the final repository. Some of these include measurements of the runtime, to validate the statements made about computational complexity (results shown in the following section).

4 Numerical Experiments

The last section of this thesis will present the numerical results that have been generated per the previously described methods. All experiments and their exact configurations are stored in [20]: `/calculation-helpers/aggregator.py`. Because of that, it is possible to fully reproduce the measurements, identically to how everything is presented here. This includes the elements that depend on randomness, as all random generators are consistently seeded and the seeds are included in the logs.

The code is implemented to be executed only on CPU, so running it requires no setup like configuring graphics cards – just having an up to date version of Python and some basic packages. However, while small-scale tests will run fine on nearly every machine, more involved calculations require a lot of processing power. Most of the computational work can be sped up by utilizing multiple CPU cores in parallel and the scripts support multithreading for all expensive operations. In the case of this work, all calculations have been run using high performance computing (HPC) resources on the Linux Compute Cluster Augsburg (LiCCA). Access to LiCCA for the scope of this thesis work has been graciously granted by the University of Augsburg.

To run experiments of this codebase on HPC, documentation and control-scripts are included in [20]: `/hpc-augsburg/`. In order to share the results produced by HPC-measurements, data files in a uniform zip format are produced.

In the spirit of making research as transparent and reproducible as possible, the repository [8]: `/thesis-latex/plotgeneration/` includes all scripts to generate every plot directly from the source data. On compilation of the LaTeX source the graphs get rendered, too.

4.1 Time-Complexity Verification

At multiple times in the theoretical derivation idealized time-complexities have been stated. In practice, different hardware-related factors might come into play. The code to verify the correctness of the simplifications is timed to allow for basic verification of the announced runtime complexities.

As shown in Figure 4.1, the solid lines (that represent the runtime of the optimized versions) are approximately constant. This verifies the functions having a runtime of $\mathcal{O}(1)$. For the unoptimized versions (dashed lines) as expected a polynomial dependency can be observed. In the beginning the curves run steeper, while the influence of the border-cases is important. For larger systems the behavior tends to a linear course, which on a logarithmic plot verifies the polynomial time-dependency.

The plot further shows, that for this implementation the double flipping is the most expensive modification, closely followed by swapping. This is expected, as they are equivalent except for some cases being trivially zero in the case of hopping. Single flips logically are the cheapest modification.
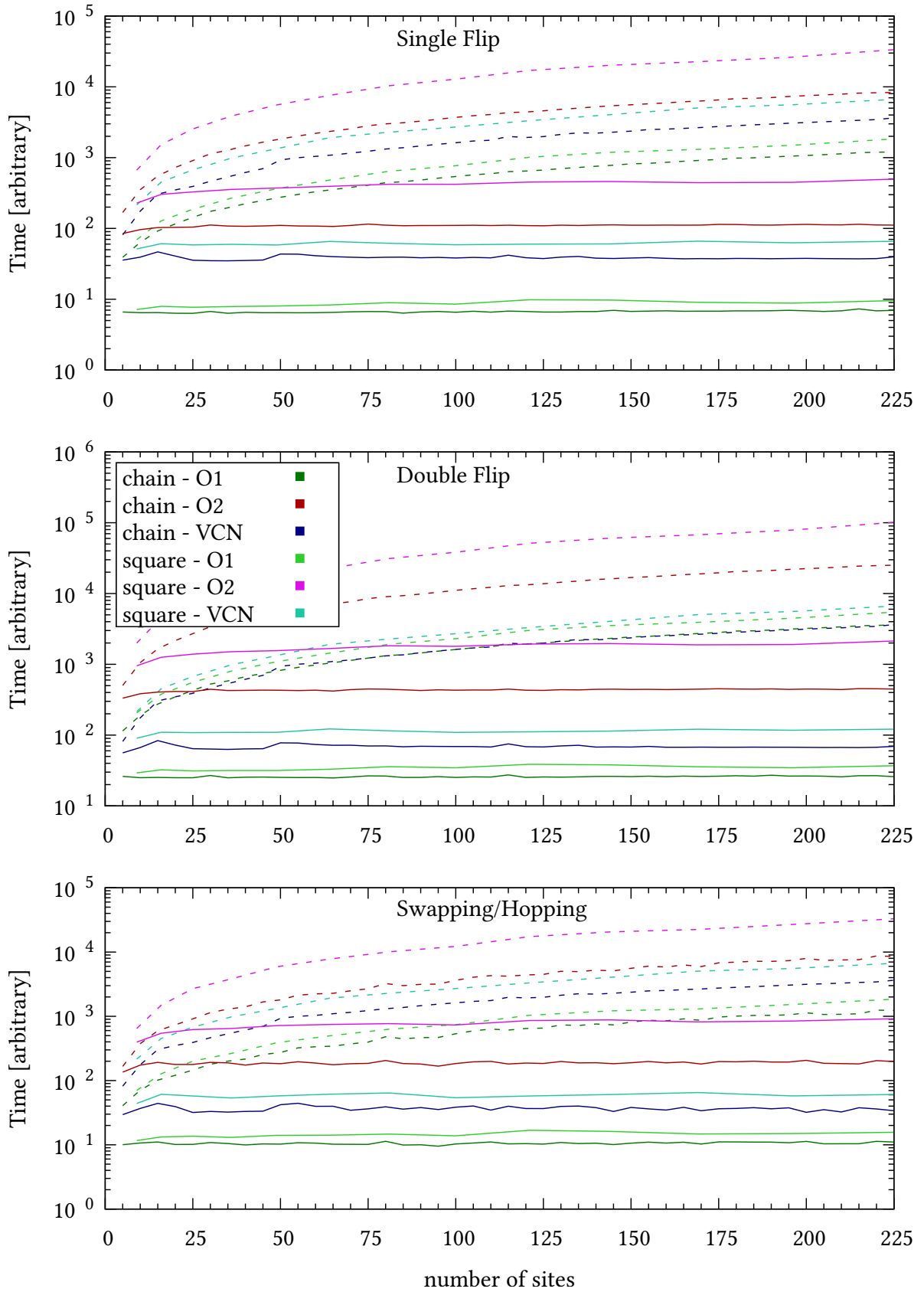
Figure 4.1: Plot of a comparison of runtimes for calculating the difference of effective Hamiltonians for an arbitrary amount of randomly generated states for the three different described local modifications. Solid lines represent the approximate time-dependency of the optimized version of the calculation function on the number of sites in the system. The dashed lines show the relation for the un-optimized version. The graph includes measurements for first (O1) and second order (O2) perturbation theory and for a first order variational classical network (VNC).

As the second order implementation is the most complicated and least optimized it is distinctly the slowest. And because of the variational classical network implementation being more general than the first order perturbation theory, it too has slightly higher runtime. Lastly, the square geometry always requires more calculation time than the chain one, as for the same number of sites a two-dimensional system has more neighbor interactions than a one-dimensional one.

4.2 Perturbation-Approximation Convergence

Giving absolute statements about the quality of a numerical method is naturally very difficult. All approximations by definition omit some aspects in comparison to the exact result. This however is the exact purpose of a model – simplifying by leaving out parts, but still capturing the relevant characteristics. In conclusion, there are two ways of gauging the quality of a model: comparing to other models and checking if the approximation holds in the limit to a more general case. Both of these checks will be employed to some degree in this or following sections.

To begin with, it should be repeated that a perturbative expansion is designed to capture the influence of a "small" perturbation to an underlying system. The cumulant expansion, that was presented in *section 2.1 Physical Basics*, is based on the interaction energy J being small compared to the energy scale of the base energy (U and ε_*). In the limit of J going to zero, the errors resulting from the expansion are expected to also become smaller. It should be the goal, to show that the expansion is *controlled* – meaning for a fixed J, a large enough expansion-order can be chosen, or equivalently, each order of the expansion is good enough to make statements for J up to a specific magnitude in relation to U.

For the first verification, therefore the Figure 4.2 shows a family of measurements that were performed for different magnitudes of the parameter J.

To balance the amount of space that is taken up by the numerous plots in the subsequent sections, supplementary plots without a direct reference ore less importance are added in the end of the thesis. In this case, three more plots (showing different observables but otherwise from the same experimental dataset) are included in the appendix at 7.4.

The measurements show the expected tendencies: The relative error is the smallest for times close to $t = 0$. For larger times the relative error continuously grows. In the start the perturbative measurement and the exact measurement are identical. Then, for later times, the relative error constantly grows. This is expected, as the perfect alignment from $t = 0$ is constantly worsening due to errors, resulting from the truncated expansion, that accumulate in the perturbative calculation until the relative error is maximal at the point of total de-phasing. The reference [22] suggests that the approximation on the basis of the cumulant expansion provides a correct description for timescales on the order of $\mathcal{O}(J^{-1})$. No quantitative analysis of that statement has been carried out, yet the suggested dependency seems to be in-line with the trend seen in the data.
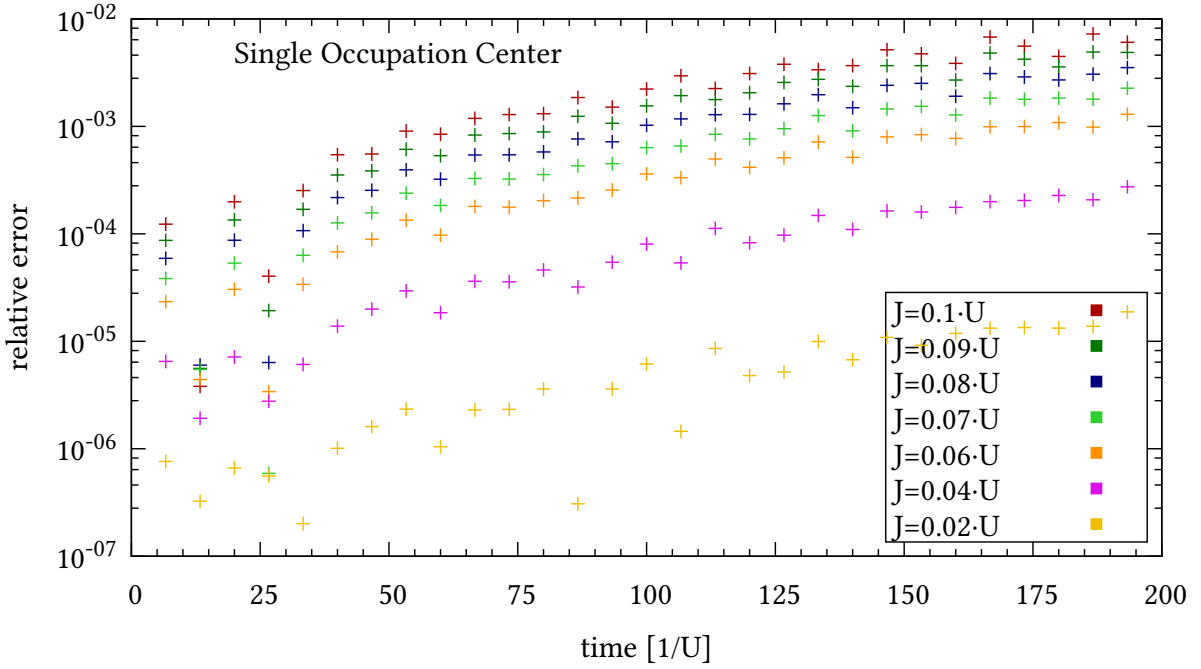
Figure 4.2: Family of measurements of the observable that describes the occupation of a single site and spin direction near the center of a system with *chain* geometry and $n = 6$ sites. The magnitude of the parameter J is varied in relation to the parameter U, while the remaining simulation-settings are kept constant. On the y-axis, the relative error (in relation to the measurement taken from exact diagonalization) is shown logarithmically scaled. The measurement was taken with exact sampling.

Furthermore, the correct behavior for a limit of a small perturbative parameter J can be also attested. The data shows a clear downwards-shift to smaller relative errors for the whole measurement series, the closer J/U comes to zero. At the limit $J = 0$, the error should also vanish, as there is no perturbation left in the system.

Viewed from the opposite side, it must be stressed that the expansion only is viable to predict the behavior of systems with weak quantum fluctuations. Were the interactions too large, the expansion could not be safely truncated after few terms [22].

As stated by Equation 74, the energy and variance stay constant for exact measurements of the system. It is therefore a measurement of the quality of a perturbative method, how good these two observables are being conserved over time. Figure 4.3 shows the course of the energy measurement, while Figure 4.4 shows the corresponding values for the variance.

The conservation of energy and variance is perfectly validated by the measurements from exact diagonalization (here in blue). From this point on, measurements will be labeled as having "exact sampling", if for the calculations all possible base-states were examined and their normalized probabilities were calculated. The chosen sampling strategy is independent of the used effective Hamiltonian.

In this case, for the blue curves the effective Hamiltonian has been calculated by exact diagonalization and the sampling has been exact – making the full measurement having no approximations. Because of the two spin-degrees per site and the exponential nature of the number of base states, a system of size $n = 4$ is already close to the limit that can be reasonably diagonalized exactly – at least with the presented setup. By applying more optimizations to the
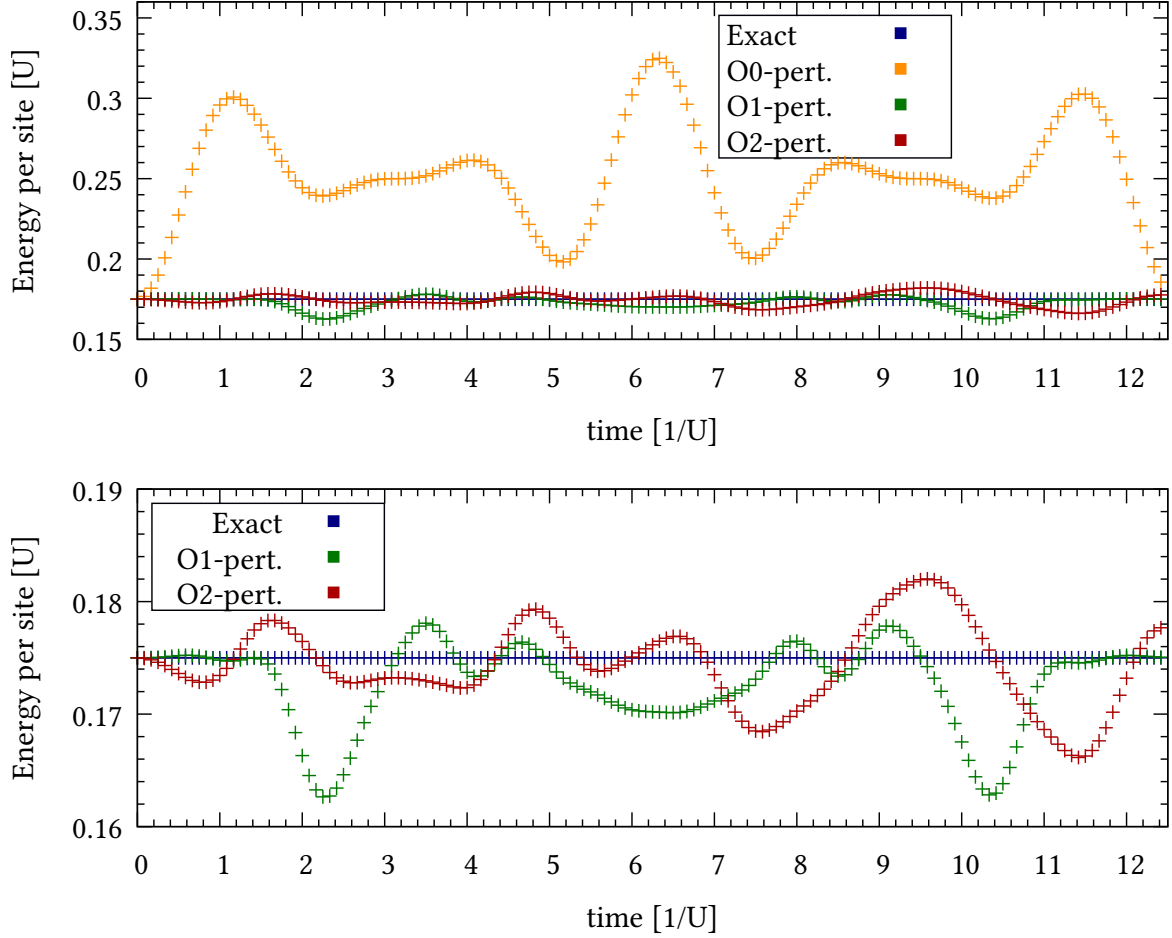
Figure 4.3: Plot of the course of the energy over time, for the case of a system with chain geometry and $n = 4$ sites. The exact energy has been calculated with exact diagonalization for all times (real measurements). Both plots show the same data, but the zeroth order approximation has been omitted from the lower graph in order to better visualize the differences between first and second order. The sampling strategy for the perturbative approximations has been exact sampling.

exact diagonalization, a few more sites can be considered, yet in this case the focus will be put on optimizing the perturbative approach.

The three degrees of perturbative measurements, while having exact sampling, rely on an effective Hamiltonian that is obtained from the truncated cumulant expansion – making the methods approximations and not exact. Effects of additionally replacing the sampling strategy with non-exact ones will be investigated in following sections.

First of all, it is easily seen that the zeroth order perturbative effective Hamiltonian does not conserve the energy or variance at all. While for the variance the center of the oscillation stays around the exact value and for the energy it at least comes back to it in the end of the measurement, the observables clearly instantly deviate from the energy at $t = 0$ starting at the first time-step. This was to be expected, as the energy-measurement takes the interactions (mediated by the interaction strength J) into account, but the effective Hamiltonian is truncated to not have any knowledge of J. Yet, the zeroth order curves can be taken as a reference of
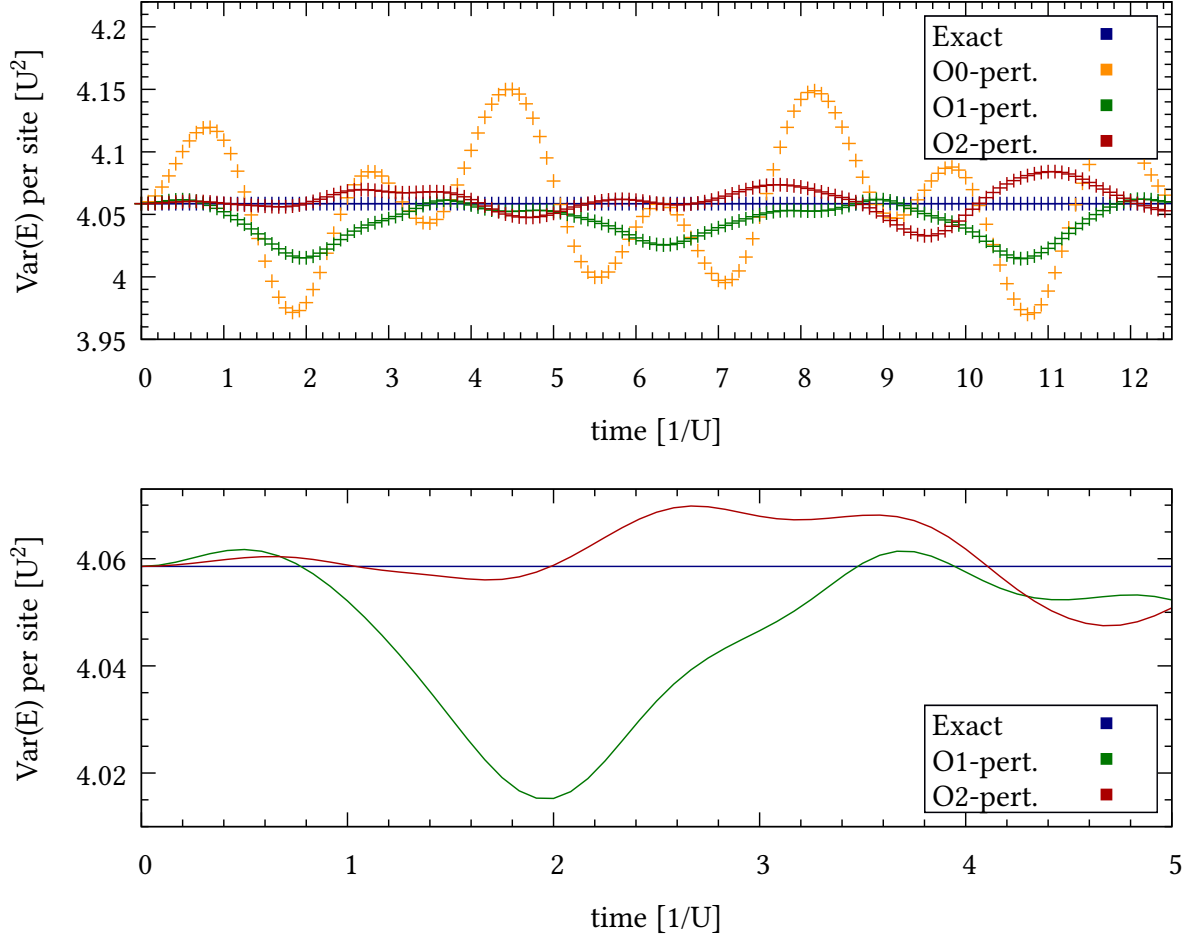
Figure 4.4: Plot of the variance measurements corresponding to the energy from Figure 4.3. Again, both plots show the same data, only the second graph omits the zeroth order and presents the differences between first and second order in a zoomed-in fashion. The measurements in the lower graph consist still only of individual data points, but the plot here is done with lines to allow to differentiate the curves in the beginning section.

scale: while all approximations can have fluctuations, no approximation should have larger fluctuations than this trivial case.

Inspecting the effects of the proposed approximation, represented by the green and red curves, the consistency is significantly better than for zeroth order. Both methods keep the observables far more constant than the zeroth order experiment and all oscillations are now around the values they are supposed to have in the exact case. For the variance, the second order calculations seem to be clearly closer to the expected solution. In the case of the energy, for the times smaller 1 (in the presented units) a bigger gap can be seen between the second order calculation and the exact result than between the first order and the exact value.

It is always necessary to look at the overall energy scale and check if this might be expected or the result of a mistake. A general rule of thumb for this approximation would be to state that an approximation from cumulant expansion to order x should have an error in the order of $\mathcal{O}(\frac{J^2}{U^2})$. For the presented measurement this is generally fulfilled. So both approximations seem to be correctly derived and implemented and should be capable of calculating helpful results. If it was

worth deriving the second order effective Hamiltonian will be examined in a third experiment for this section.

Lastly, a more involved observable will be calculated. As *subsection 2.2.4 Reduced Density Matrix* describes, the reduced density matrix of a small subsystem can be extracted only from the measurement of local observables. This process was performed for a system with chain geometry and $n = 4$ (to allow still having access to the exact diagonalization for the purpose of comparison). In Figure 4.5 and Figure 4.6, the concurrence and purity of such a density matrix of two adjacent spin-up particles are displayed. The values of the individual observables for the Pauli operator basis for the same subsystem can be found in the appendix at 7.5.
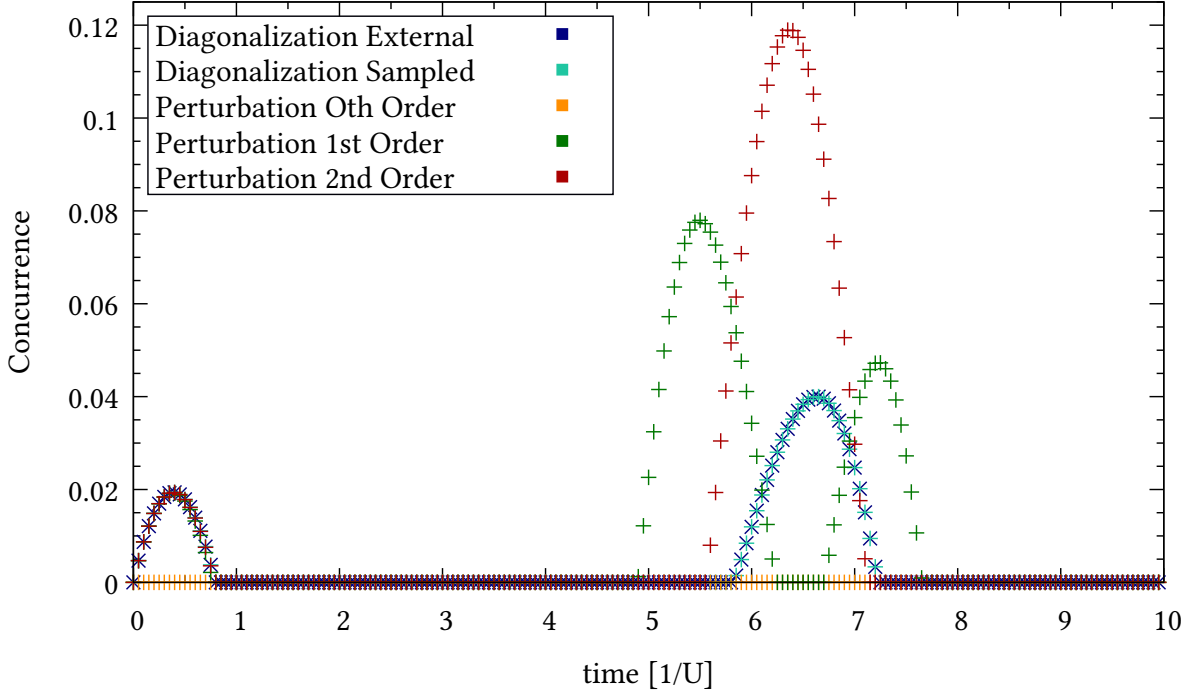
Figure 4.5: Concurrence measurements extracted from the reduced density matrix of a subsystem of two adjacent spin-up particles in a system with chain geometry and $n = 4$ sites. The results from exact diagonalization have been calculated by two independent means. Once by replacing the effective Hamiltonian with one gained from exact diagonalization, once by calculating the full time-evolution of the complete density matrix and tracing out the required subsystem (here labeled as "External"). This is used for verification. As the perturbative algorithm does not use the full density matrix anywhere, this needed to be implemented independently. The measurement uses exact sampling.

To start off the analysis, it should be noted how the external measurements and the ones that reconstruct the density matrix from observables (albeit the values being obtained by exact diagonalization and exact sampling) match exactly. This verifies the described method of obtaining a reduced density matrix without access to the full density matrix.

The concurrence measurements now clearly show the superiority of the second order cumulant expansion. Position and shape of the second peak are clearly replicated much more precise by the second order calculations, than by the first order ones. It can also be noted, that while the purity measurement for the zeroth order is not totally off, the zeroth order calculation completely fails to replicate the features which are necessary to calculate the concurrence. Not
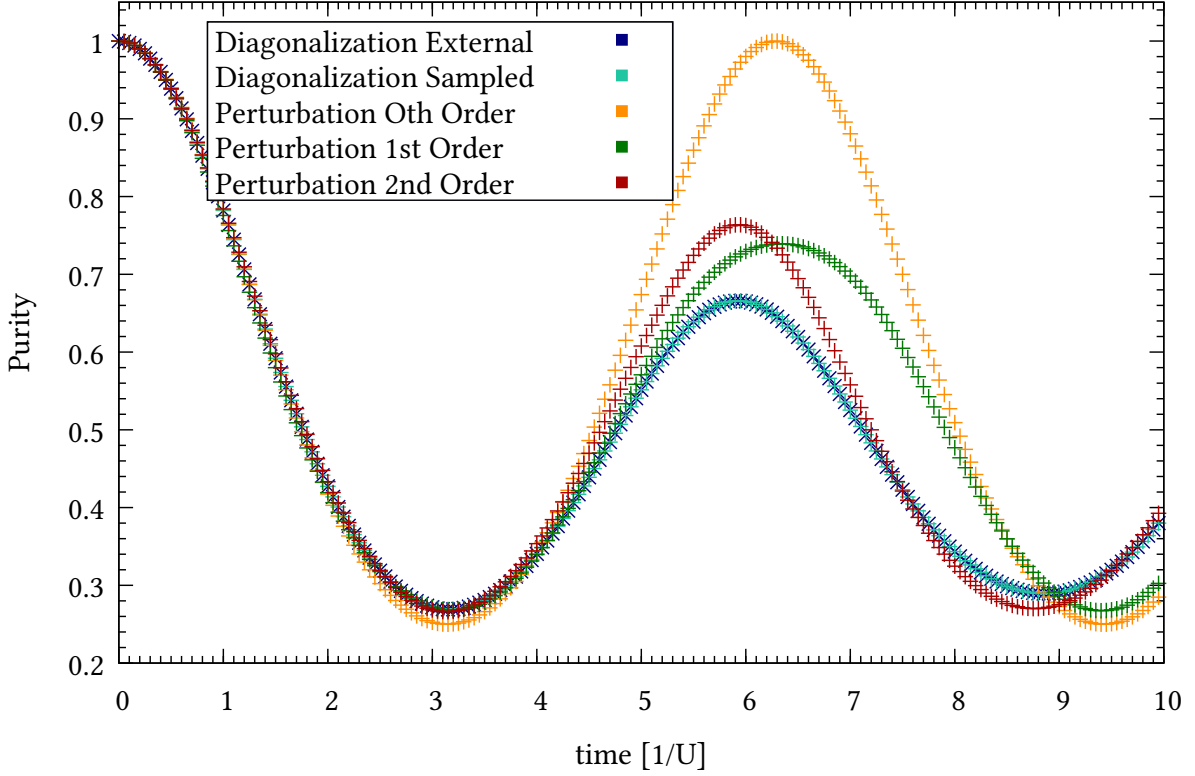
Figure 4.6: Purity measurement, belonging to the experiment in Figure 4.5.

even the first peak (which also the first order replicates perfectly) can be reproduced while the interaction J is ignored.

Looking at the observables from which this density matrix was obtained (see appendix, 7.5), it can be clearly seen that the observables corresponding to Pauli-z seem to be the most difficult to replicate. A Pauli-z measurement is equivalent to a re-scaled measurement of a single occupation.

4.3 Monte-Carlo-Sampling Convergence

So far, only exact sampling has been used. However, as illustrated in *subsection 2.4.1 Monte Carlo Sampling*, for systems with too many base states it is necessary to switch to a sampling method that only looks at a subset of these states. An attempt to verify the convergence of the Monte-Carlo sampling strategy was made in Figure 4.7. One additional measurement for a different observable is included in the appendix at 7.6.

Upon inspection of the trend of the standard deviation in relation to the number of states sampled by Monte-Carlo, a clear correspondence can be spotted. For both measured observables, a higher number of Monte-Carlo samples directly corresponds to a lower standard deviation. This behavior was expected, as theoretically for an infinite number of samples it should be possible to perfectly replicate the original probability distribution. Taking this experiment as a reference, future Monte-Carlo sampling will be performed with 30 000 samples as a default
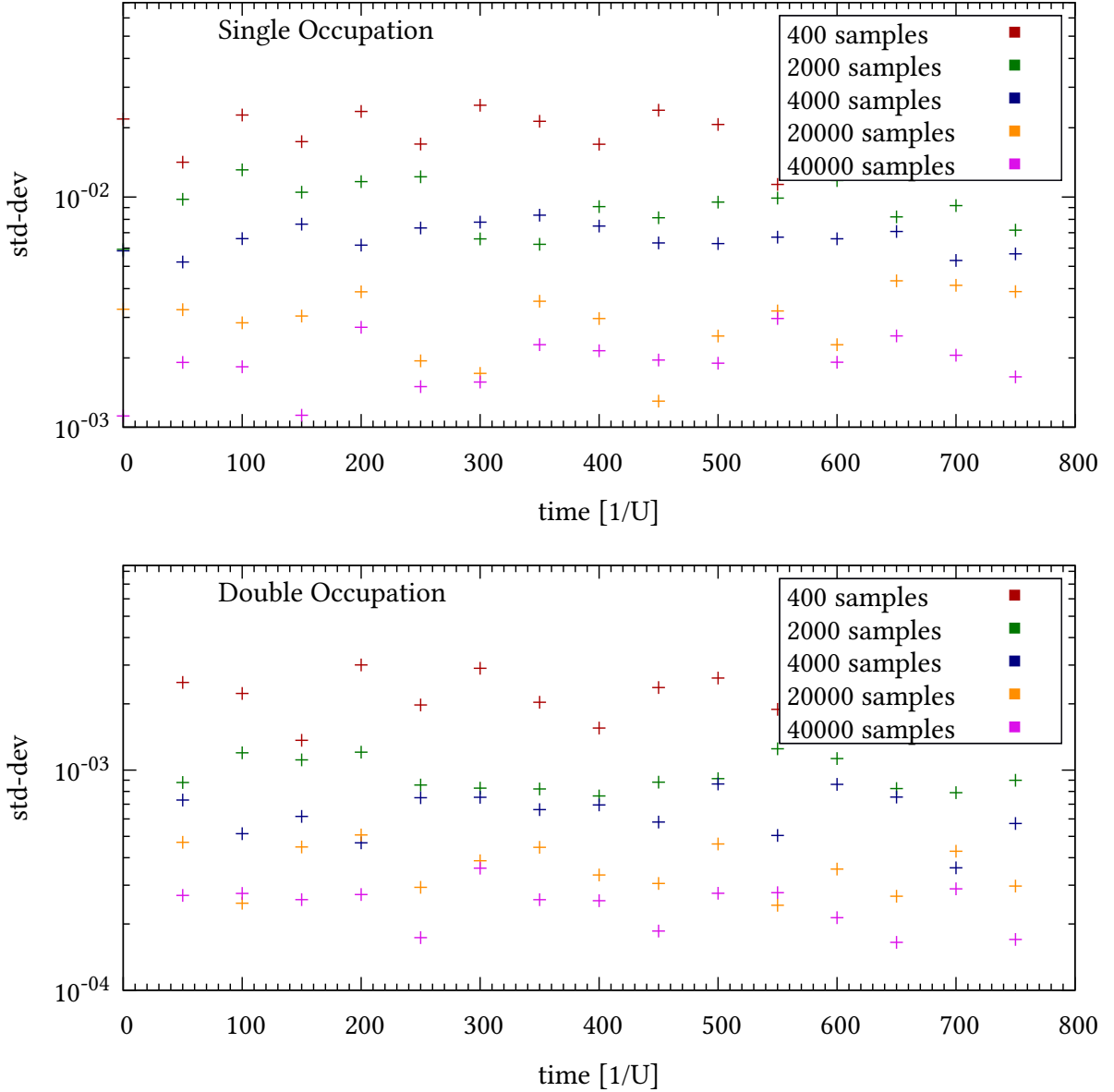
Figure 4.7: Standard deviation of occupation measurements, obtained from calculations on the same system and parameterization as in the previous section, but with Monte-Carlo sampling instead of exact sampling. Ten runs of the same experiment, with different seeds, have been calculated. This resulted in ten slightly different datasets between which the standard deviation could be calculated. The five series differ in the number of samples which have been taken into account for each single observable calculation.

setting. While this may be much more than would have been needed to sample this small test-system exactly, it is an incredible reduction for systems with $n \approx 100$ sites.

Finally, upon closely inspecting Figure 4.7 again, one can spot that the overall spread of the single occupation is considerably higher than for the double occupation. This was already noted at the end of the previous section, where the statement was presented how the measurement of single occupation (corresponding to Pauli-z measurements) seem to be harder to reproduce than the other Pauli observables.

4.4 Insufficient Variational Ansatz

As previously hinted in the theoretical derivation, this section does not present successful measurements. For a comparably long time, an unknown error was preventing the suggested variational classical networks algorithm from working. The problem lay, as previously revealed, in the explicit time-dependency of the base-energy that had not been properly dealt with. Before coming back to successful verifications of the described methods, this segment is meant to briefly outline the discovery process that led to finding the reason of the misbehavior.

In Figure 4.8 an energy and variance measurement is presented. The plot and experiment do thereby not differ from the ones from previous sections, except for the inclusion of VCN measurements in comparison to the established perturbative effective Hamiltonians. In the appendix at 7.7, the VCN-parameters and their analytical comparisons (as described in *subsection 2.3.2 Application to this Task*) are plotted.

At the start of the debugging-process a relatively large remaining imaginary part kept appearing for the measurements of all observables, as long as an effective Hamiltonian parametrized by VCN was used. However, this problem could be isolated to a sub-class in the implementation, that was incorrectly missing an overwrite for one of the methods of its parent class. The imaginary part that was remaining after this had been fixed is sufficiently small to be attributed to numerical noise, as studying Figure 4.8 reveals.

Yet, the resulting architecture is still not able to keep the energy nor variance constant – which is an inherent property of the theoretical model. One could hypothesize, that the problem lies in a numerical integration-step that is too large. For this reason, the experiment varies the effective step size that is used for integration. While this results in changes to the resulting values of the observables, it does not result in differences of the ability of the parametrization to conserve the energy. A convergence can be spotted, yet definitely no convergence to a constant energy.

As a second hypothesis, the problem was suspected to be related to the initialization. Therefore the experiment in Figure 4.9 also tries to compute the energy and variance, but with purposefully large and different initial values for the variational parameters. In this case, the corresponding variational parameters are more interesting and can be found in the appendix at 7.8.

The misbehavior of the model was already appearing on the very first time-step. Therefore it could be assumed, that maybe an unlucky choice of initial values was causing the divergence of the VCN-parameters (compare appendix, 7.8). However, it can be quickly spotted that neither the energy, nor the variance, nor the variational parameters seem to have a different course. All the trends seem to have just been offset by the initial value.

Consequently, this could only be explained in one way: the development of the parameters in this parametrization is independent of the parameters themself. This however eliminates the $\mathcal{H}_N(t)$-part from being responsible and only the explicitly time-dependent base-energy part remains as a possible candidate. A repeated inspection of Figure 4.8 supports this hypothesis, as the basic shape of the energy seems to strongly correlate with the shape of the series of zeroth order energy measurements.
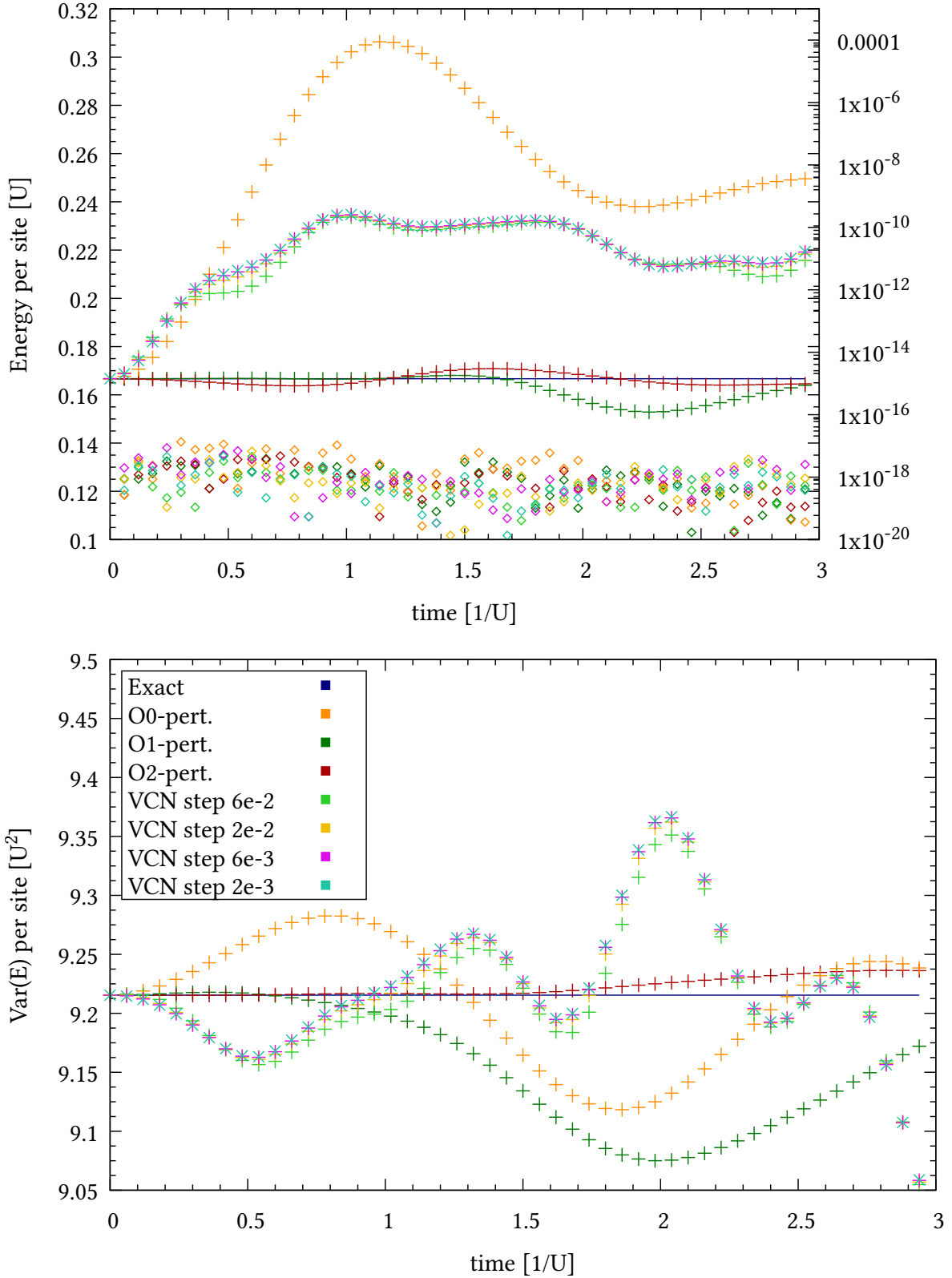
Figure 4.8: Plot of energy and variance of a system with chain geometry and $n = 6$ lattice sites. The energy plot additionally contains the imaginary part of the observed quantities, which is depicted by the circle-symbol. For the imaginary parts, the right y-axis is used. Four different VCN-series are plotted. These differ by the effective step size, describing the magnitude of the time-step for numerical integration. Exact sampling was used, to restrict possible problems to the effective Hamiltonian without having to deal with Monte-Carlo noise. The exact energies and variances from this point on are just a continuation of the values at time $t = 0$. This is indicated by the solid line and no measurement points.
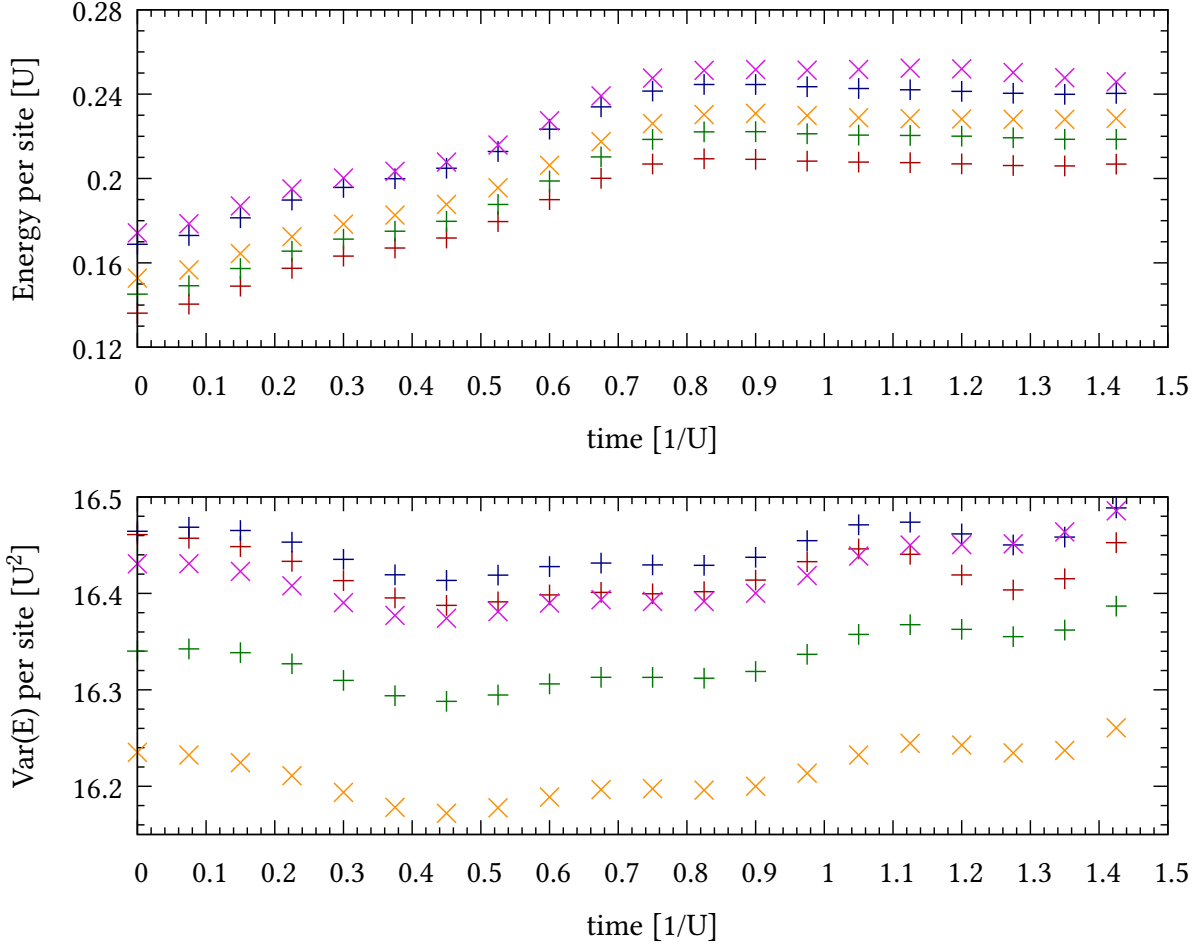
Figure 4.9: Plot of energy and variance of a system with chain geometry and $n = 8$ lattice sites. Each run was performed with a different seed and the spread of the initial parameter values was purposefully set very high. Exact sampling has been used.

As already revealed in the theoretical derivation section, eliminating the explicit time-dependency from the base-energy had been the necessary step to allow the model to perform its duty properly.

4.5 Improvements through VCN

By using the correct parametrization, introduced in *subsection 2.3.3 Explicit Time-Dependency*, the VCN approximation starts to fulfill its theorized properties. Firstly, the validation measurements in Figure 4.10 show that the energy can be conserved by the VCN method. Comparing the scale of the y-axis with the plots from previous sections, it becomes evident that these results are significantly closer to conserving energy and variance than the results from first order cumulative methods (for reasons of readability not included in these plots). Furthermore, the experiment shows that a convergence towards the elusive constant function seems to be present and can be systematically reached by reducing the step size far enough.

It is however unlikely that perfect convergence can be achieved by just raising this one parameter. For one, it should be stressed that each of the consecutive series in Figure 4.10 is twice as expensive to evaluate as the one before it. Halving the size of the intermediate integration steps requires doubling the number of sampled states and consequently also at least doubles the calculation time. Secondly, the trend to which the presented curves converge does not seem to be perfectly constant. This can probably be attributed to the accumulation of numerical errors for a large enough amount of steps.

For these reasons it is definitely still expedient to try integrating higher order terms into the variational ansatz or using higher order numerical integrators. However, considering the scope of a master thesis, the first order VCN parametrization that was suggested should be sufficient. A full comparison between the analytically derived approximations and a VCN parametrization with sensible hyperparameters is finally shown in Figure 4.11. The dependency of the results for this measurement on the effective step size is included in the appendix at 7.9.

With Figure 4.11, the first full application of the VCN method on a square system is compared to the analytically derived expansion. Satisfactorily, it can be stated that the VCN ansatz clearly beats even the second order effective Hamiltonian with analytically derived coefficients. On inspection of the convergence-dependency on the effective step size (see appendix, 7.9) the run that was labeled with 20 intermediate steps is chosen as a good compromise of computational cost and benefits to precision. The measurement series clearly shows that making the effective step size as small as possible does not come with endlessly increasing precision. To reiterate, this must probably be attributed to numerical errors accumulating.

Similar to the experiment in Figure 4.11, the appendix additionally contains the exact same experiment just for a square system of side length $n = 3$, resulting in 9 lattice sites (see appendix, 7.10). For a system of nine sites, the exact sampling strategy is already too expensive to finish the calculations for the smallest effective step sizes in a reasonable duration. The calculation is aborted after too much time has elapsed. Yet again, the results obtained by using 20 intermediate steps seem to be a good compromise in terms of cost and use.

Although the hyperparameters between the experiments are practically identical, the corse of the measurements in 7.10 has a qualitatively different trend. This could be attributed to the overwhelming influence of edge effects for such small systems.
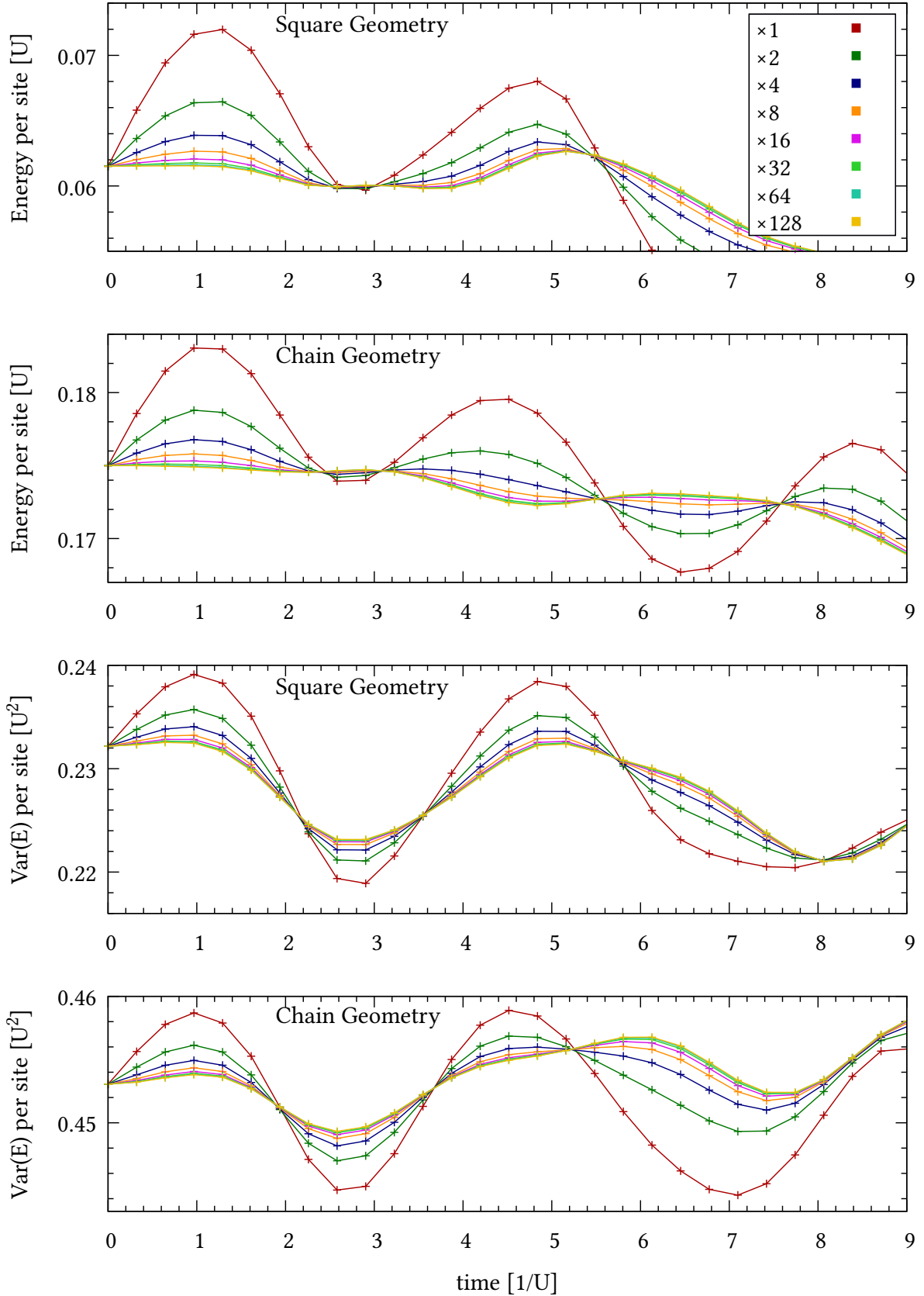
Figure 4.10: Plot of energy and variance of a system with either chain geometry and $n = 4$ lattice sites or with square geometry of side length $n = 2$, also resulting in 4 lattice sites. The eight different runs differ by the effective step size that is used for numerical integration. For e.g. the red curve no intermediate integration step is used, the step size is equivalent to the distance of plotted points. Consequently, for the orange curve the size of one integration step is only one eighth of the step in the red curve. Exact sampling was used.
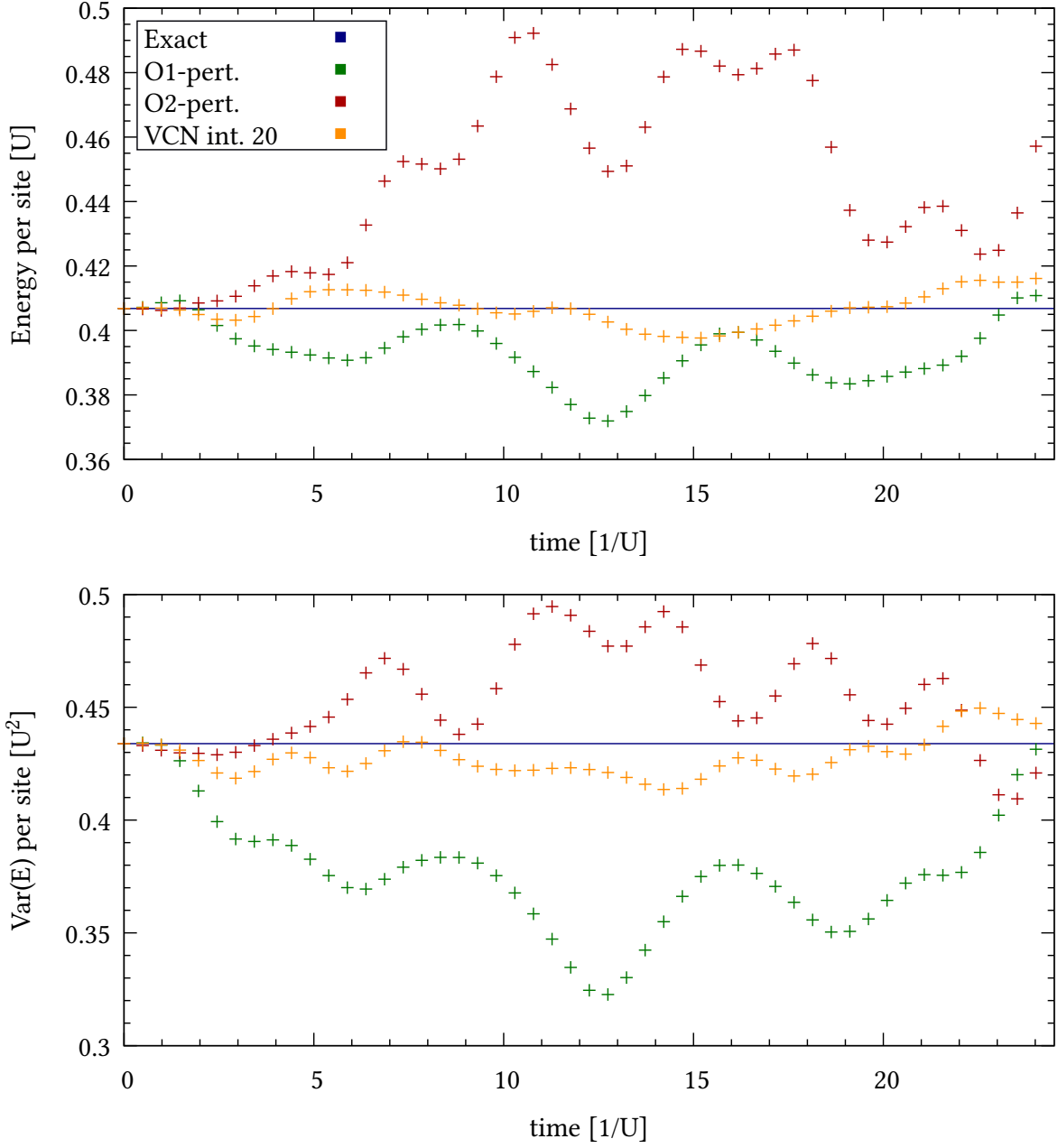
Figure 4.11: Plot of energy and variance of a system with square geometry of side length $n = 2$, resulting in 4 lattice sites. The first and second order effective Hamiltonians from cumulant expansion are compared to a VCN effective Hamiltonian with an effective step size of one 20th the size of the depicted steps. Exact sampling was used.

It could, however, also be the result of an *even-odd-effect*, between the experiment with side length $n = 2$ and $n = 3$. To rule out this last possibility, for the last remaining experiment a series of systems with only even side lengths has been examined.

The final section will present a final experiment that combines all previously accumulated experience.

4.6 System-Size Dependency of Measurements

As a final experiment one attempt will be made to leave the realm of system sizes that can be exactly diagonalized. The past sections have verified the implemented code can likely produce useful experimental results for large systems.

For the sampling strategy Monte Carlo sampling with $30\,000$ samples per operation is used. Considering a 10×10 square system of the presented Hubbard model has around 1.6×10^{29} base states, this seems exceptionally reasonable.
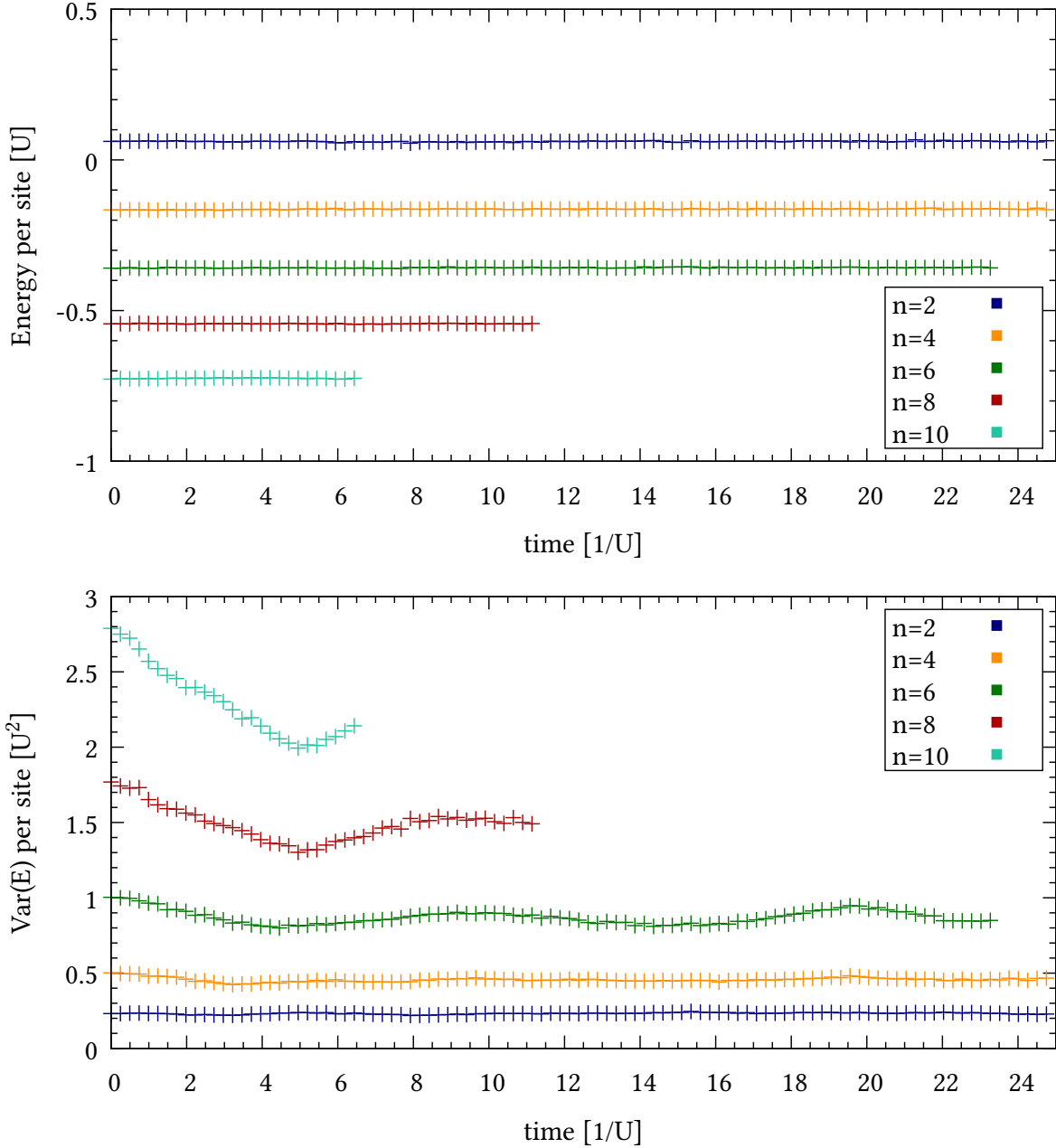
Figure 4.12: Trend of the energy and variance for five systems with square geometry and increasing side length n. Because of the square geometry, the number of sites is $n \cdot n$. The measurements have been taken with an effective Hamiltonian parametrized by VCN and Monte Carlo sampling.

To show the behavior of the method over a large range of system sizes, a series of square systems with side lengths 2-4-6-8-10 was chosen. The electrical field was purposefully orientated away from 45°, to coerce different behavior for vertical and horizontal interactions.

As for the other experiments, the energy is used to gauge the quality of the approximation. The plots for the course of energy and variance are depicted in Figure 4.12. Again, for the large systems the computation might be aborted early as the runtime grows at least linearly with the number of sites and the number of sites grows quadratically with the parameter n. It can be seen, that the model is perfectly capable of keeping the energies constant – the variance is kept stable for the smaller systems, but it starts to dip for the larger ones. This could be explained by the relatively tiny amount of variational parameters that are present in the employed parametrization. The larger systems might very well require more degrees of freedom in $\mathcal{H}_{\mathrm{VCN}}(N, \vec{\eta})$ to be properly encodable.

To show an observable different than the energy, Figure 4.13 depicts the course of the spin current. The plot shows the differences between a vertically and a horizontally oriented current flow.

Both currents show relatively little dependency on the size of the system – the differences of the smallest system can most likely be explained with the comparably large influence of edge effects.

Even though the system is governed by a steep potential gradient as a result of the electrical field, the current shows oscillating behavior centered around zero and no transport. While this might seem unintuitive when imagining a continuous system, for a discrete lattice (as is the case here) this is exactly what is expected. This combination of properties causes particles to experience *Wannier-Stark localization* [47], resulting in locally oscillatory behavior of the particles and the absence of transport. Because of a lack of time, this has not been examined more extensively or quantitatively.

Supplementary to this set of experiments, a measurement of the purity and the concurrence is included in the appendix at 7.11. The subsystems that define the reduced density matrices are the same as for the current in Figure 4.13. While the purity behaves in similarly consistent fashion to the current, only the concurrence measurement of the smallest system seems to produce measurable spikes beyond the first one. While this might have multiple reasons, one likely factor is the previously stated challenge to get good readings for single site occupations (Pauli-z measurements). Further research needs to go into this direction, to be able to make a statement about the viability of extracting the concurrence for use in quantitative measurements via this method.
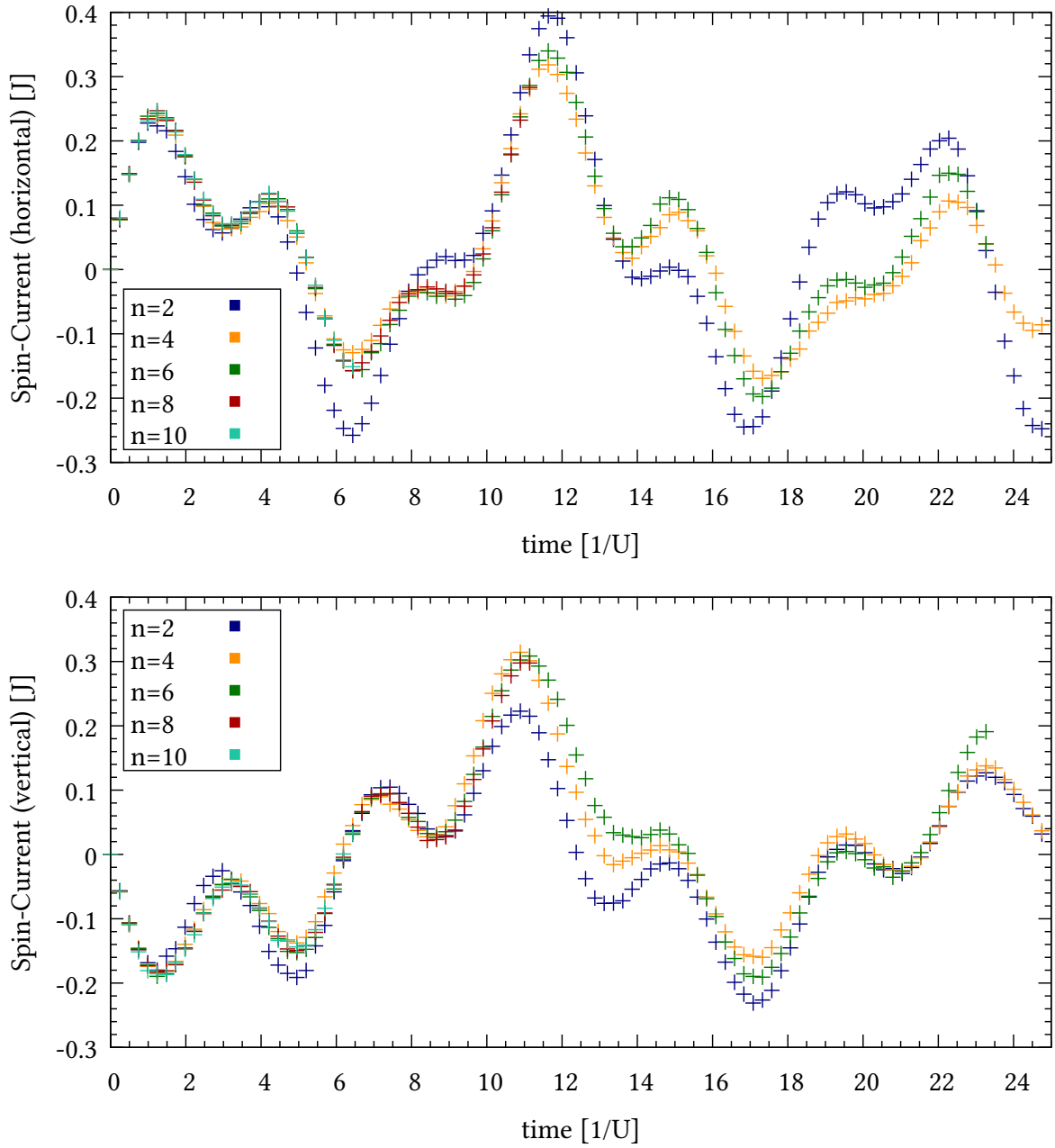
Figure 4.13: Trend of the spin current between two adjacent sites with spin-up in the center of the respective system. The experiment and hyperparameters are the same as in Figure 4.12. The different course for horizontal and vertical exchange results from the asymmetric direction of the electrical field.

5 Conclusion

While working on this research subject over the course of my practical training, my project work and finally the master thesis, the material turned out to be more challenging than expected. Initially, work advanced with steady progress and clear goals. Because of the seemingly very complicated commutation relations, I spent a lot of time on the development of the Math-Manipulator to streamline similar work. I had the feeling that several people at the chair would benefit from a tool to validate and improve the calculations that are traditionally done on paper, without resorting to the full adoption of a computer-algebra-system. Looking back at the calculations eventually needed for the thesis, this tool seems to have been under-utilized.

However, during the development I was able to significantly improve my understanding for the different types of operators, their properties and differences. This not only directly lead to the calculations now being as straight forward as they are, but also – as my understanding increased – numerous recalculations were required. In that context, having this consistent tool saved many hours. So I can confidently say that the time for the side-project was not wasted but well invested.

Consequently, when it came down to practical realization of the mathematical derivations, I spent a lot of time writing extra alternative calculations and programmed checks to verify the correctness of the analytical calculations and numerical implementations. This lead to many bugs and quirks being found – many of which with only small effects on the result. Still, this strategy (as opposed to taking the first "working" code) made me highly confident to state that the final implementation is bug-free and correctly reflects the described methods accurately to a high degree. This would be especially important, as a logical next step would be to produce an even more efficient implementation in a lower-level, compiled language like C(++) or Rust to improve base level performance.

While this thesis and the corresponding code aspires to provide adequate comparisons between the methods and validate their viability, because of the amount of inefficient small operations in Python it lacks the efficiency required for taking on experiments on very large systems. This shows that the described computational complexity class can be reached and the method can be used to handle the underlying problem – yet there still are a lot of micro-optimizations left to tackle. Also, several generalizations needed in previous revisions/comparisons could be taken out, if but the final iteration of the method is needed.

Getting all processes to work as intended turned out to be more challenging than first anticipated. For one, a problem in two dimensions and non-periodic boundary conditions has a lot more edge-cases than the same method on a one-dimensional, linear chain with periodic boundary conditions. When tracking which interactions cancel, the overlapping "spheres of influence" produce a significantly higher number of interaction terms in this problem compared to descriptions of previous applications to simpler geometric lattices.

This, combined with two more unexpectedly complex road-blocks in the implementation and the explicit time-dependency caused my available time to erode. Nearing the end of the allocated

time-frame, the work was still containing less "Physics" (Concrete new insights, obtained by experimentally observing the effects of parameter variations) than we were hoping. In the words of my supervisor Markus Heyl, in the cases where we can not imagine what is stopping the experiment from behaving as expected: "That's science!". Now at the end I can proudly state, that this motivated me to finally solve all remaining road-blocks before the deadline.

Which is also the reason I am happy to be able to provide a fully tested and detailed implementation and verification of the hypothesized solution. To provide the biggest contribution to science, I believe in providing clean final code and transparent plotting process for making numerical methods reproducible and expandable. And I hope the code can be build upon in future research, that may focus on honing the experiment's parameters and not being held up with debugging while replicating theoretical methods.

6 Bibliography

[1] J. Kell, *Investigation of transformer architectures for geometrical graph structures and their application to two-dimensional spin systems*, (Oct. 7, 2022) `https://github.com/jonas-kell/bachelor-thesis-documents`.

[2] *ChatGPT — Release Notes | OpenAI Help Center*, `https://help.openai.com/en/articles/6825453-chatgpt-release-notes` (visited on 11/16/2024).

[3] *What's going on with Nvidia stock and the booming AI market?*, `https://www.techtarget.com/whatis/feature/Whats-going-on-with-Nvidia-stock-and-the-booming-AI-market` (visited on 11/16/2024).

[4] *The Nobel Prize in Physics 2024*, NobelPrize.org, `https://www.nobelprize.org/prizes/physics/2024/summary/` (visited on 11/16/2024).

[5] *Is AI physics or chemistry? Nobel Prize wins spark debate*, euronews, (Oct. 12, 2024) `https://www.euronews.com/next/2024/10/12/is-ai-physics-or-chemistry-nobel-prize-wins-spark-debate-about-techs-role-in-science` (visited on 11/16/2024).

[6] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities", *Proceedings of the National Academy of Sciences of the United States of America* **79**, 2554 (1982).

[7] J. J. Hopfield, "Whatever Happened to Solid State Physics?", *Annual Review of Condensed Matter Physics* **5**, 1–13 (2014).

[8] J. Kell, *LaTeX sources for this report, the thesis and presentations*, (2024) `https://github.com/jonas-kell/master-thesis-documents`.

[9] F. Schwabl, *Quantenmechanik für Fortgeschrittene (QM II)*, Springer-Lehrbuch, 10.1007/978-3-540-85076-2 (Springer, Berlin, Heidelberg, 2008).

[10] F. Tennie, V. Vedral, and C. Schilling, "Universal upper bounds on the Bose-Einstein condensate and the Hubbard star", *Physical Review B* **96**, 064502 (2017).

[11] N. H. Lindner, A. Auerbach, and D. P. Arovas, "Vortex Dynamics and Hall Conductivity of Hard Core Bosons", *Physical Review B* **82**, 134510 (2010).

[12] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*, 10.1017/CBO9780511976667 (Dec. 9, 2010).

[13] O. Derzhko, "Jordan-Wigner fermionization for spin-1/2 systems in two dimensions: A brief review", 10.48550/arXiv.cond-mat/0101188 (2001).

[14] J. Kell, *Math-Manipulator Tool - Info and Repository*, (2024) `https://github.com/jonas-kell/math-manipulator`.

[15] J. Kell, *Supplementary Calculations Perfomed with Math-Manipulator*, (2024) `https://github.com/jonas-kell/master-thesis-mm-calculations`.

[16] *Fermions and Jordan-Wigner String*, `https://itensor.org/docs.cgi?page=tutorials/fermions` (visited on 07/29/2024).

[17] J. Hubbard, "Electron correlations in narrow energy bands", *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, `10.1098/rspa.1963.0204` (1963).

[18] H. A. Gersch and G. C. Knollman, "Quantum Cell Model for Bosons", *Phys. Rev.* **129**, 959–967 (1963).

[19] F. Schwabl, *Quantenmechanik (QM I)*, Springer-Lehrbuch, `10.1007/978-3-540-73675-2` (Springer, Berlin, Heidelberg, 2007).

[20] J. Kell, *Thesis Code implementation and Reference*, (2024) `https://github.com/jonas-kell/master-thesis-code`.

[21] Y. Zheng, "Exact solution of $U \to \infty$ 1D Hubbard model in electric field", *Physics Letters A* **394**, 127112 (2021).

[22] R. Verdel, M. Schmitt, Y.-P. Huang, P. Karpov, and M. Heyl, "Variational classical networks for dynamics in interacting quantum matter", *Physical Review B* **103**, 165103 (2021).

[23] R. Kubo, "Generalized Cumulant Expansion Method", *Journal of the Physical Society of Japan* **17**, 1100–1120 (1962).

[24] M. Schmitt and M. Heyl, "Quantum dynamics in transverse-field Ising models from classical networks", *SciPost Phys.* **4**, 013 (2018).

[25] H. Haber, "The time evolution operator as a time-ordered exponential", *Physics 215 - Winter 2018* (2018).

[26] M. Schmitt and M. Reh, "jVMC: Versatile and performant variational Monte Carlo leveraging automated differentiation and GPU acceleration", *SciPost Physics Codebases*, `10.21468/scipostphyscodeb.2` (2022).

[27] X. Zotos, F. Naef, and P. Prelovsek, "Transport and conservation laws", *Physical Review B* **55**, 11029–11032 (1997).

[28] J. Preskill, *Lecture Notes for Physics 229: Quantum Information and Computation*, `9781506189918` (CreateSpace Independent Publishing Platform, Jan. 12, 2015), 322 pp.

[29] J. J. Sakurai and J. Napolitano, *Modern Quantum Mechanics*, 3rd ed., `10.1017/9781108587280` (Cambridge University Press, Sept. 17, 2020).

[30] *Tensor product - Encyclopedia of Mathematics*, `http://encyclopediaofmath.org/index.php?title=Tensor_product&oldid=43382` (visited on 11/23/2024).

[31] S. Hill and W. K. Wootters, "Entanglement of a Pair of Quantum Bits", *Physical Review Letters* **78**, 5022–5025 (1997).

[32] R. LaRose, *QuIC Seminar - Semester I - Seminar 6*, Quantum States and Partial Trace, https://www.ryanlarose.com/uploads/1/1/5/8/115879647/quic06-states-trace.pdf, (Nov. 19, 2018) https://www.ryanlarose.com/semester-i.html (visited on 12/07/2024).

[33] W. K. Wootters, "Entanglement of Formation of an Arbitrary State of Two Qubits", *Physical Review Letters* **80**, 2245–2248 (1998).

[34] A. S. Richardson and J. M. Finn, "Symplectic integrators with adaptive time steps", *Plasma Physics and Controlled Fusion* **54**, 014004 (2012).

[35] G. Carleo and M. Troyer, "Solving the quantum many-body problem with artificial neural networks", *Science* **355**, 602–606 (2017).

[36] "The time-dependent variational principle (TDVP)", in *Geometry of the Time-Dependent Variational Principle in Quantum Mechanics*, edited by P. Kramer and M. Saraceno (Springer, Berlin, Heidelberg, 1981), pp. 3–14.

[37] M. Schmitt and M. Heyl, "Quantum Many-Body Dynamics in Two Dimensions with Artificial Neural Networks", *Physical Review Letters* **125**, 100503 (2020).

[38] J. Broeckhove, L. Lathouwers, E. Kesteloot, and P. Van Leuven, "On the equivalence of time-dependent variational principles", *Chemical Physics Letters* **149**, 547–550 (1988).

[39] J. D. Gray and S. A. Morris, "When Is a Function That Satisfies the Cauchy-Riemann Equations Analytic?", *The American Mathematical Monthly* **85**, 246–256 (1978).

[40] C.-Y. Park and M. J. Kastoryano, "Geometry of learning neural quantum states", *Physical Review Research* **2**, 023232 (2020).

[41] R. Marinho Jr, "Noether's theorem in classical mechanics revisited", *European Journal of Physics* **28**, 10.1088/0143-0807/28/1/004 (2006).

[42] W. M. C. Foulkes, L. Mitas, R. J. Needs, and G. Rajagopal, "Quantum Monte Carlo simulations of solids", *Rev. Mod. Phys.* **73**, 33–83 (2001).

[43] S. Chib and E. Greenberg, "Understanding the Metropolis-Hastings Algorithm", *The American Statistician* **49**, 327–335 (1995).

[44] A. V. Aho, M. S. Lam, R. Sethi, and J. D. Ullman, *Compilers: Principles, Techniques, and Tools*, 2nd ed., 0321486811 (2006).

[45] *SymPy 1.12.1rc1 documentation*, https://docs.sympy.org/latest/index.html (visited on 04/12/2024).

[46] classabbyamp, *Espresso Logic Minimizer*, (Oct. 4, 2017) https://github.com/classabbyamp/espresso-logic (visited on 02/10/2025).

[47] M. Schulz, C. A. Hooley, R. Moessner, and F. Pollmann, "Stark Many-Body Localization", *Phys. Rev. Lett.* **122**, 040606 (2019).

7 Appendix

7.1 Code-Optimization: Default

[20]: `/calculation-helpers/timing-efficiency/analyticallogicalfunctions.py`

```python
def part_A_flipping(
    Lu: int, Ld: int, Mu: int, Md: int, flip_up: bool, flip_l: bool
) -> int:

    if flip_up:
        if flip_l:
            Lup = not Lu
            Ldp = Ld
            Mup = Mu
            Mdp = Md
        else:
            Lup = Lu
            Ldp = Ld
            Mup = not Mu
            Mdp = Md
    else:
        if flip_l:
            Lup = Lu
            Ldp = not Ld
            Mup = Mu
            Mdp = Md
        else:
            Lup = Lu
            Ldp = Ld
            Mup = Mu
            Mdp = not Md

    res = 0
    res += Lu and not Mu and Ld == Md
    res -= Lup and not Mup and Ldp == Mdp
    res += Ld and not Md and Lu == Mu
    res -= Ldp and not Mdp and Lup == Mup
```

7.2 Code-Optimization: If-free

[20]: `/calculation-helpers/timing-efficiency/analyticallogicalfunctions.py`

```python
def part_A_flipping_if_free(
    Lu: int, Ld: int, Mu: int, Md: int, flip_up: bool, flip_l: bool
) -> int:

    Lup = Lu == (not (flip_up and flip_l))
    Mup = Mu == (not flip_up or flip_l)  # (not (flip_up and not
      flip_l))
    Ldp = Ld == (flip_up or not flip_l)  # (not (not flip_up and
      flip_l))
    Mdp = Md == (flip_up or flip_l)  # (not (not flip_up and not
      flip_l))

    res = 0
    res += Lu and not Mu and Ld == Md
    res -= Lup and not Mup and Ldp == Mdp
    res += Ld and not Md and Lu == Mu
    res -= Ldp and not Mdp and Lup == Mup

    return res
```

7.3 Code-Optimization: Espresso

[20]: `/calculation-helpers/timing-efficiency/logicalcalcfunctions.py`

```python
def part_A_flipping(
    Lu: int, Ld: int, Mu: int, Md: int, flip_up: bool, flip_l: bool
) -> int:
    res = 0
    res += (
        (Lu and not Ld and not Mu and not Md)
        or (not Lu and Ld and not Mu and not Md)
        or (Lu and Ld and Mu and not Md)
        or (Lu and Ld and not Mu and Md)
    )
    res -= (
        (Lu and not Ld and not Mu and Md and not flip_up)
        or (not Lu and Ld and not Mu and Md and not flip_up and
        ↪  not flip_l)
        or (Lu and Ld and Mu and Md and not flip_l)
        or (Lu and not Ld and Mu and not Md and flip_up and not
        ↪  flip_l)
        or (not Lu and Ld and Mu and not Md and flip_up)
        or (not Lu and not Ld and not Mu and not Md and flip_l)
        or (Lu and not Ld and Mu and not Md and not flip_up and
        ↪  flip_l)
        or (not Lu and Ld and not Mu and Md and flip_up and flip_l)
        or (Lu and Ld and not Mu and not Md)
    )
    return res
```

7.4 Small Interaction-Term Limit: Extra Plots

Referenced by Figure 4.2.
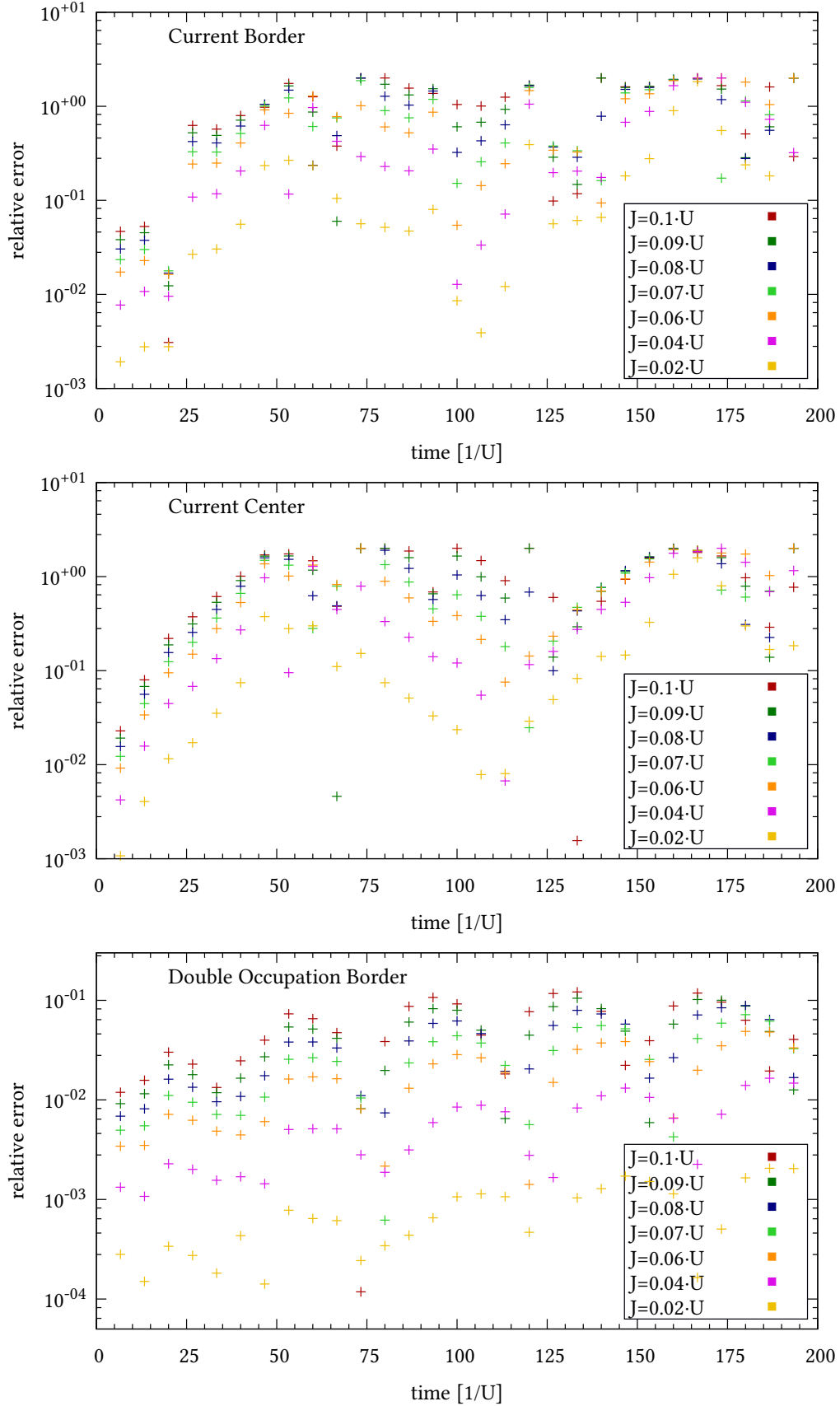
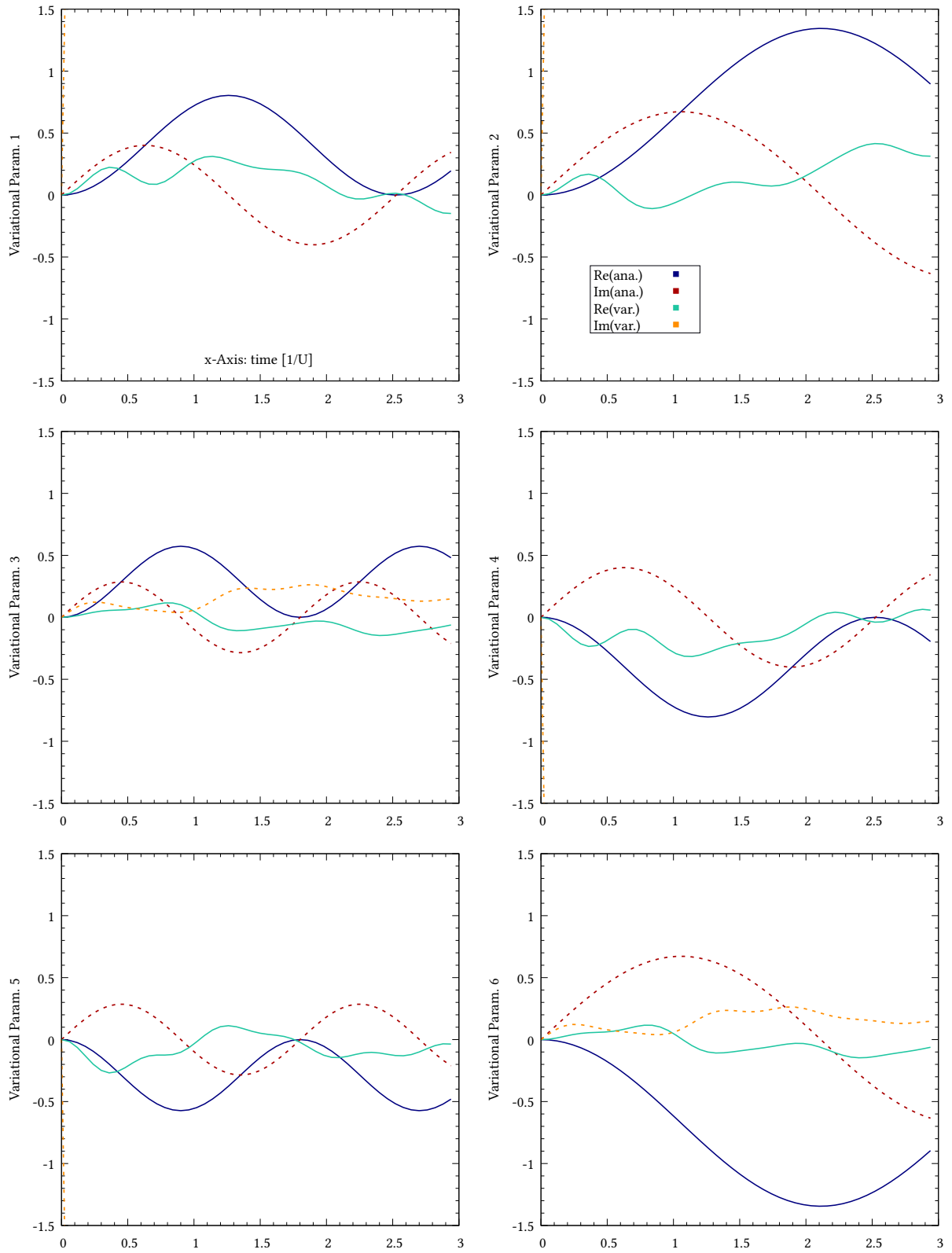## 7.5 Density Matrix Measurement: Pauli Observables

Referenced by Figure 4.5.
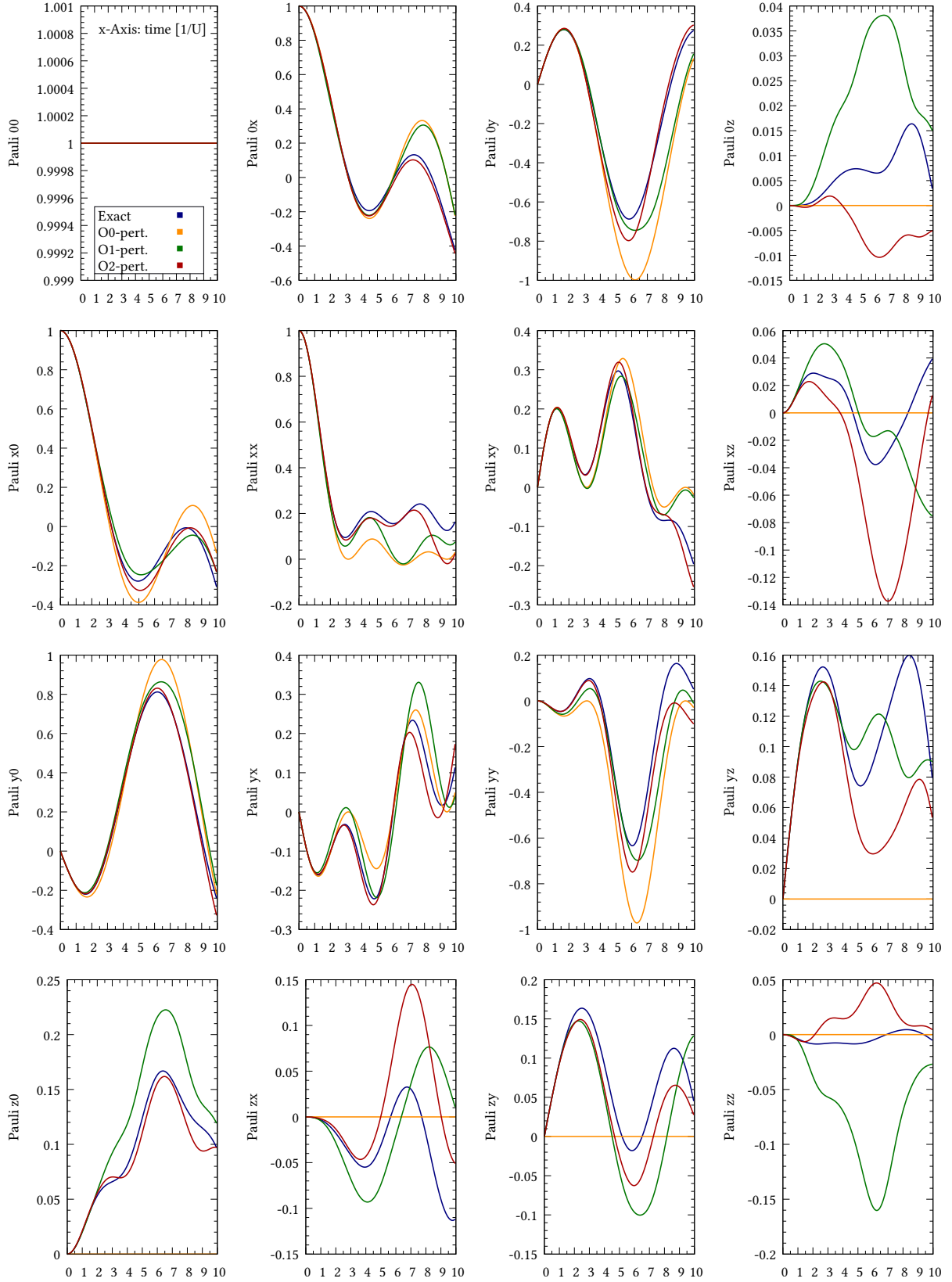
7.6 Monte-Carlo-Sampling Convergence: Current
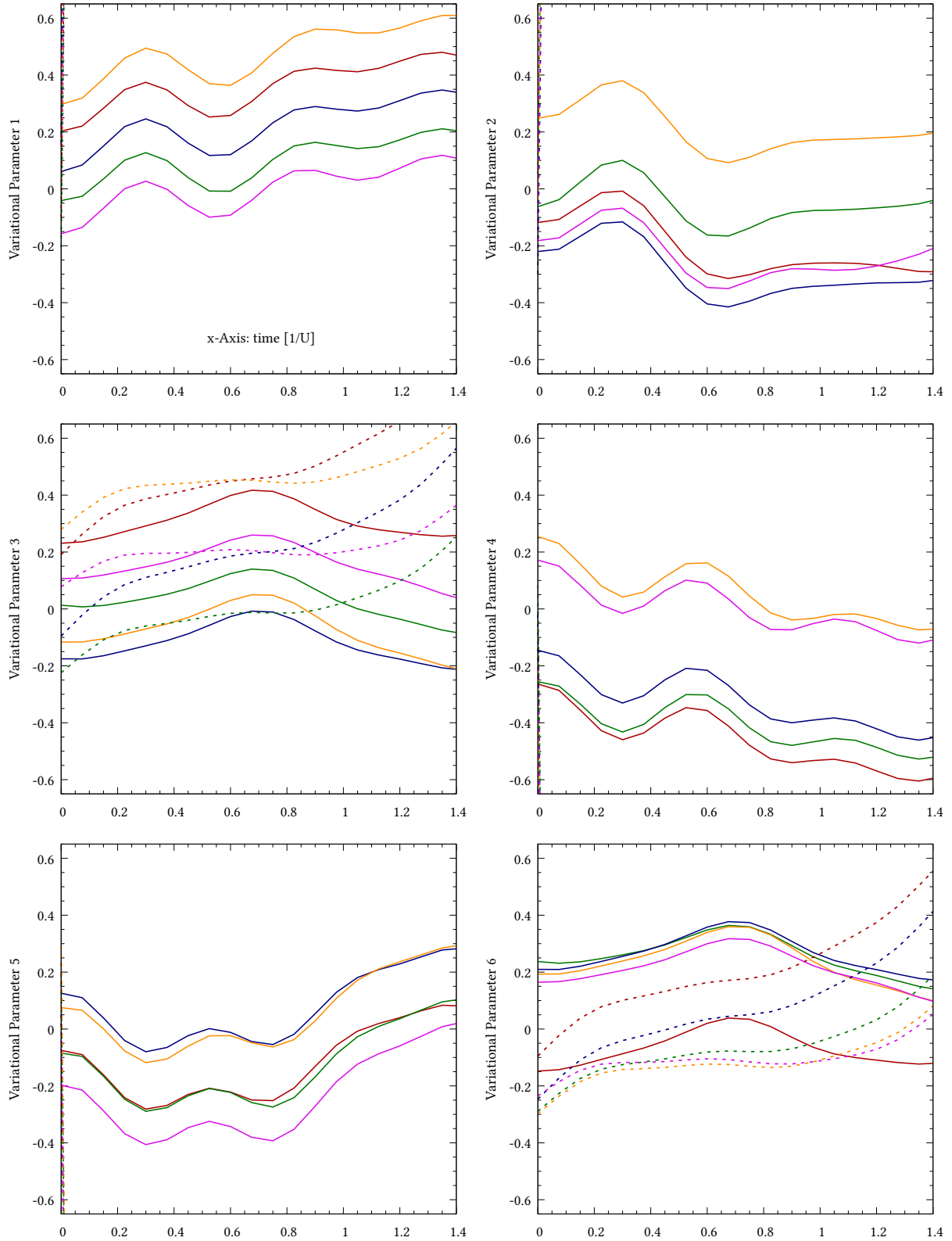
Referenced by Figure 4.7.
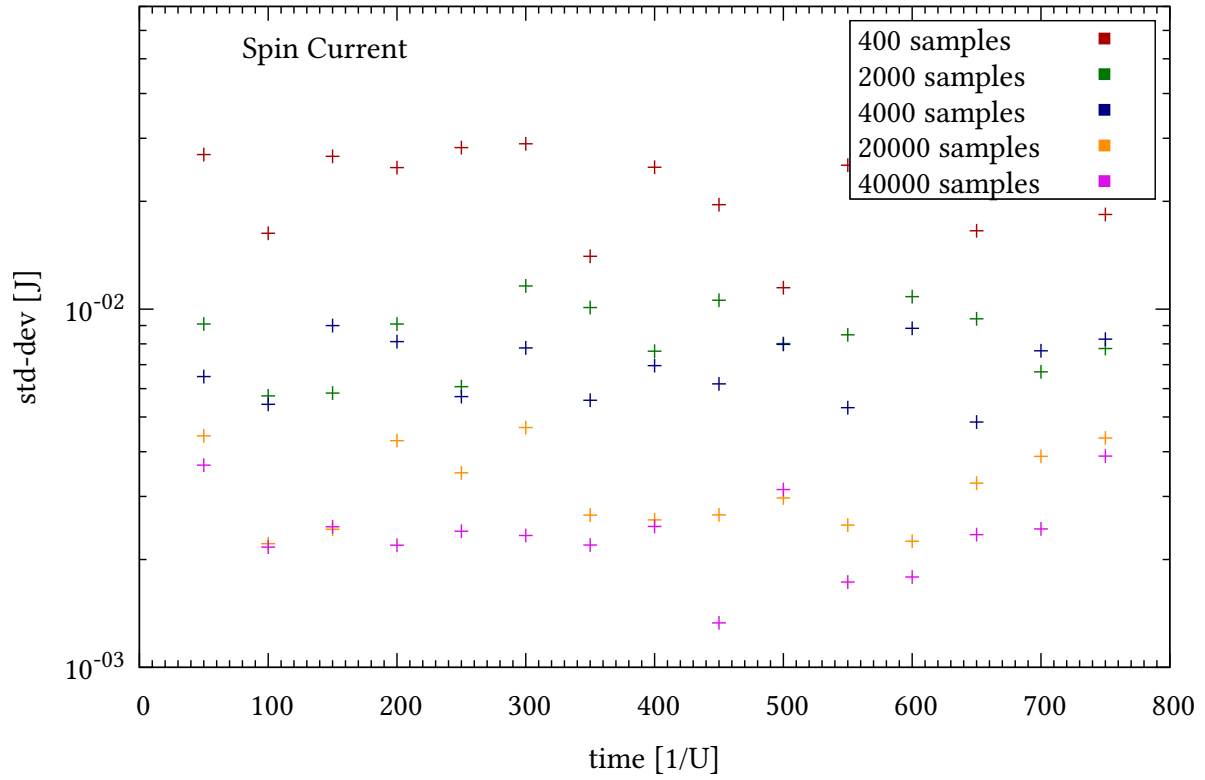
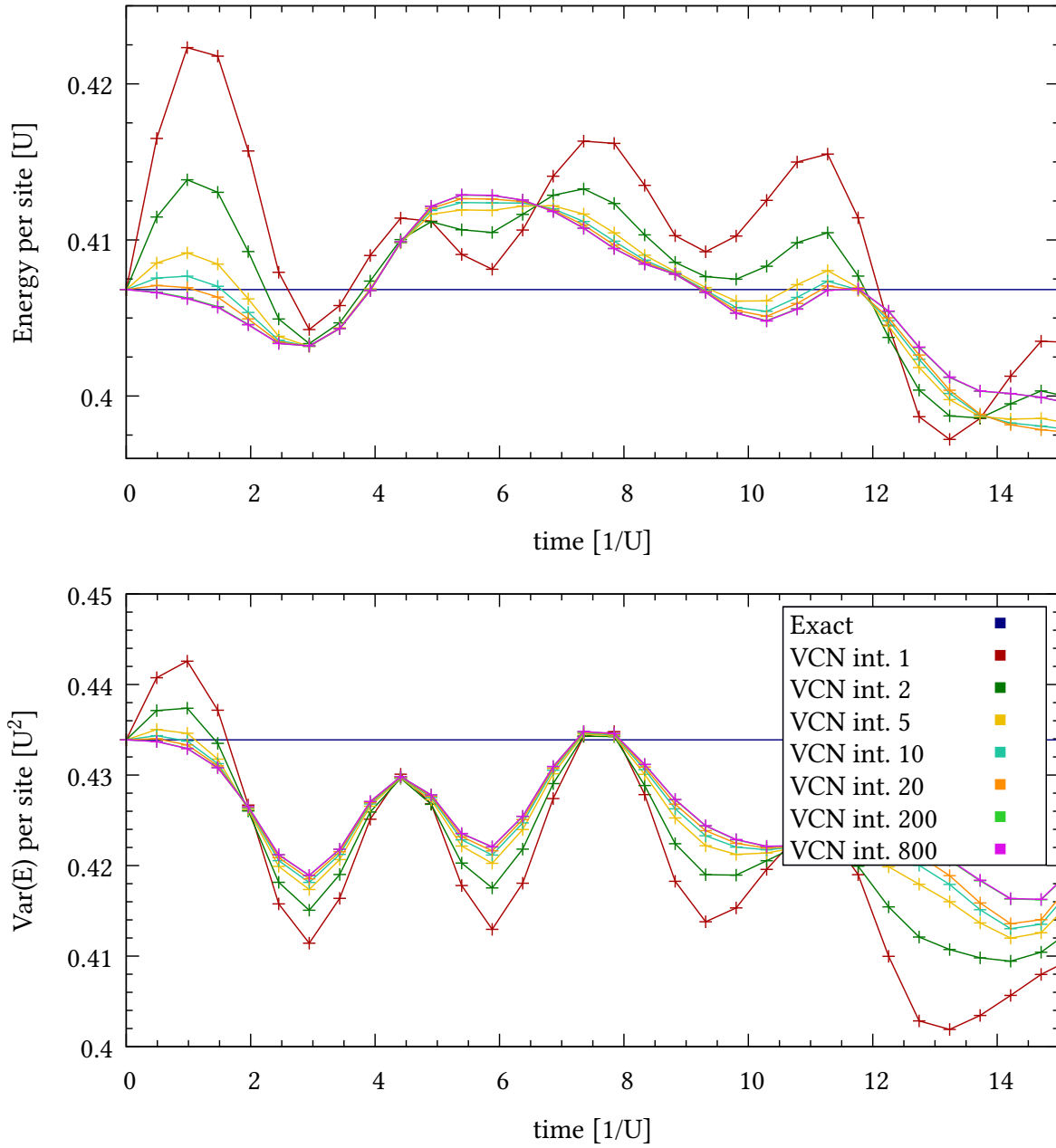## 7.7 VCN Problem: Parameters for Step Size

Referenced by Figure 4.8.

7.8 VCN Problem: Parameters for Starting Values

Referenced by Figure 4.9. Note the imaginary part (represented by the dotted line) of parameters 1, 2, 4 and 5 diverging nearly instantly.

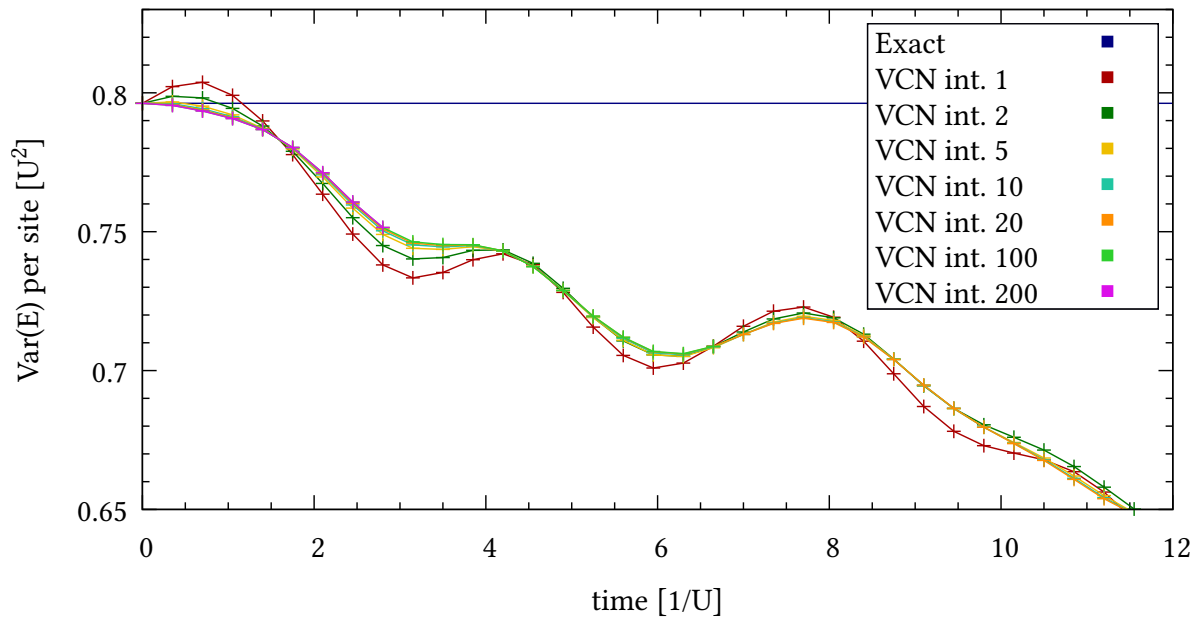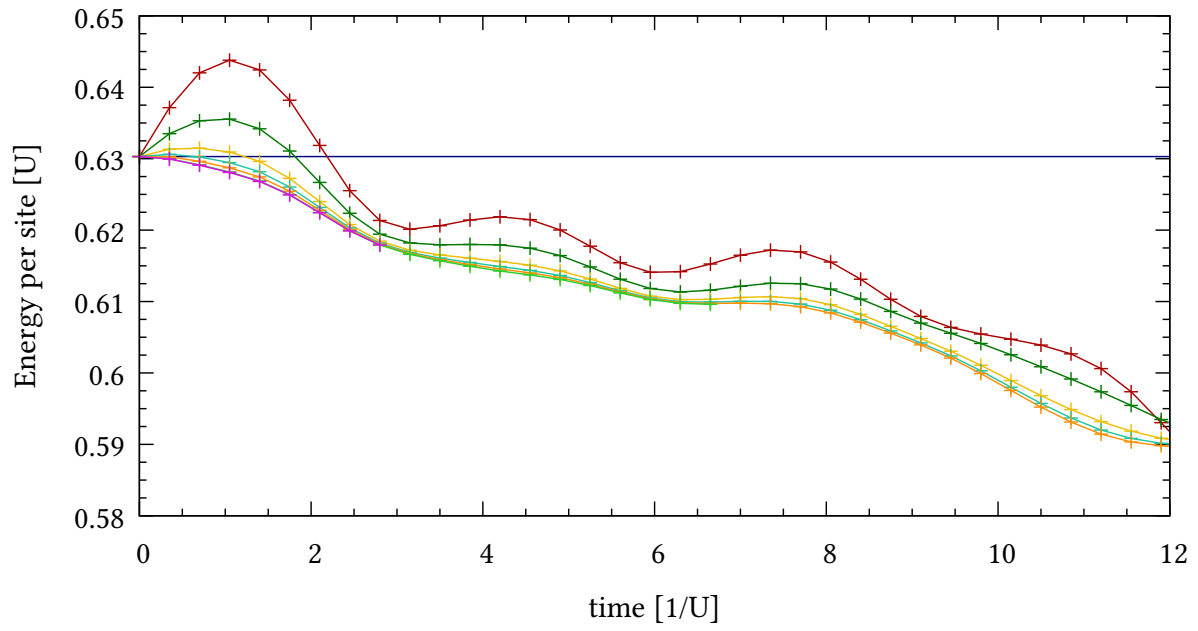## 7.9 VCN Application: Eff. Step Size for Square System

Referenced by Figure 4.11.

7.10 VCN Application: Larger Square System

Referenced at the end of section 4.5. The parameters are identical to the experiment in Figure 4.11, just with a larger system.
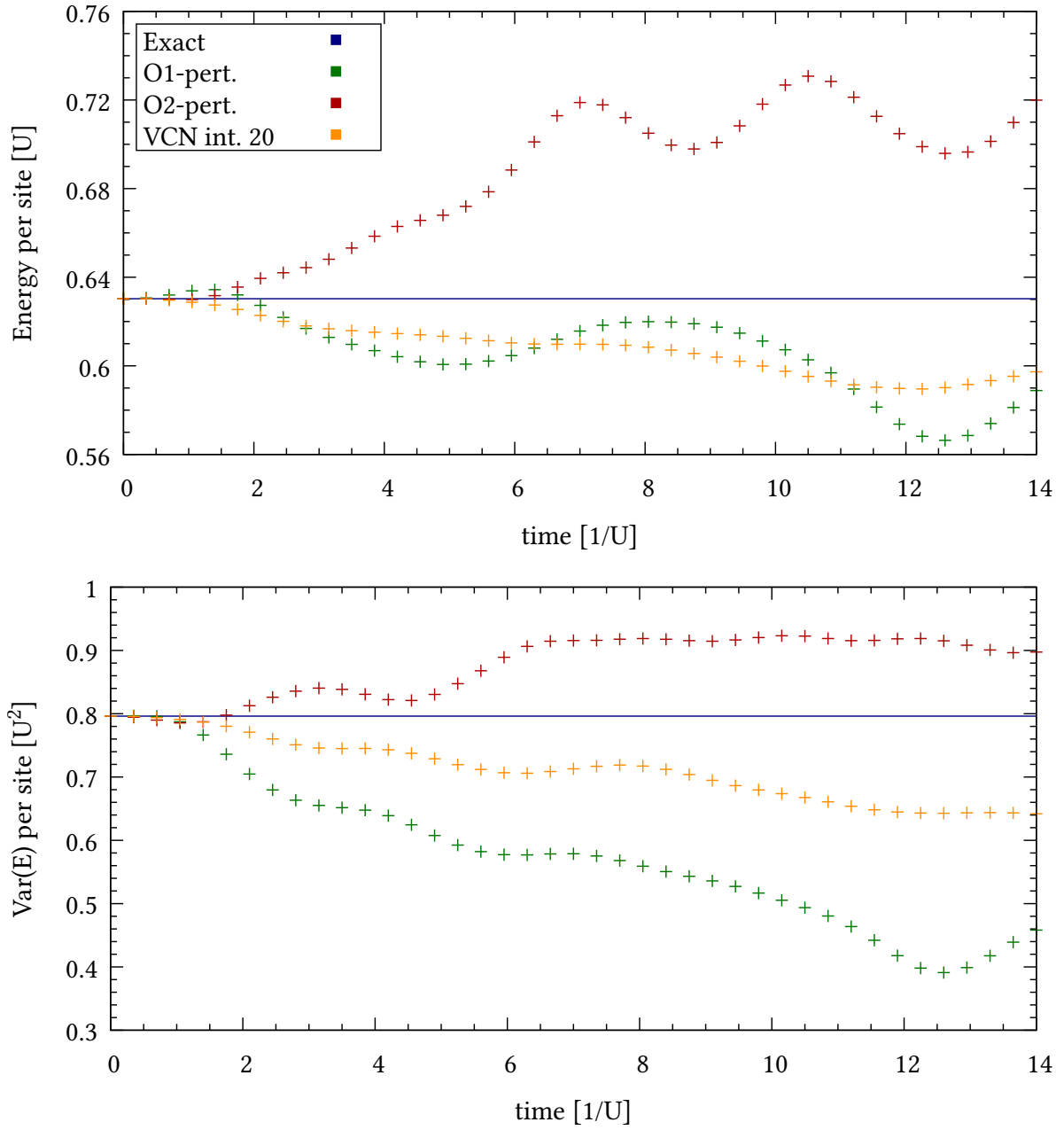
7.11 VCN Application: System Size Dependency

Referenced at the end of section 4.6 and further details in Figure 4.12