Implicit partitioned Runge-Kutta integrators for simulations of gauge theories

Master Thesis

Master's Program
Computer Simulation in Science

Supervisors:

Prof. Dr. Michael Günther
Prof. Dr. Francesco Knechtli

Michèle Wandelt
Faculty of Mathematics and Natural Science
University of Wuppertal

April 30, 2010

Contents

List of Figures

List of Tables

Acknowledgement

First of all, I would like to thank my supervisors Prof. Dr. Michael Günther and Prof. Dr. Francesco Knechtli for the idea of this interesting interdisciplinary Master Thesis. Your doors were opened at any time and I am grateful for your continual support.

Additionally, I would like to send a warm thanks to Dr. Michael Striebel for the proofreading of this work.

Furthermore, a special thank goes to Matthias Lichter for the introduction in the secrets of Matlab and especially Mathematica. Without Mathematica, I would have been lost in the calculation of the method coefficients.

Wuppertal, April 2010 Michèle Wandelt

Introduction

In the simulations of gauge theories, expectation values of certain operators have to be calculated. This is usually performed using a Hybrid Monte Carlo method that combines a Metropolis step with a Molecular Dynamics step. During the Molecular Dynamics step, Hamiltonian equations of motion have to be solved through an integration scheme. The state-of-the-art integration methods are the Leapfrog scheme as well as spliting methods.

At the beginning of this thesis, there was the question:

- Are there any higher order numerical integration schemes besides the Leapfrog or splitting methods for simulations of gauge theories?

For this to be possible, it has to be taken into account that the numerical integration method has to fulfill some desired properties. First of all, it has to be symmetric, i. e. time reversible and of a preferably high convergence order. Due to the fact that the intgration method has to be applied in gauge theories with elements regarded to be situated in a Lie group, the preservation of the Lie group structure has to be fulfilled as well.
As mentioned in the title of this thesis, implicit partitioned Runge-Kutta methods are chosen to be examined in this work.

In chapter 1, a lattice gauge theory is introduced. It is described from a mathematical point of view with respect to the simulations.The necessary concepts as link, plaquette and staple are depicted here. Additionally, the Hamilton operator and the derivation of its equations of motion are outlined.

Chapter 2 describes the Markov process used in a Metropolis method. Moreover, the Hybrid Monte Carlo algorithm is characterized in detail. The necessity of the properties symmetry and symplecticity (=volume-preservation) of the numerical intergration scheme are discussed. These features are used in the detailed balance condition of the Markov process to ensure the reachability of the equilibrium distribution of the field.

The numerical integration is the main part of this thesis and situated in chapter 3. It starts with an examination of the aforementioned desired properties. The characteristics symmetry and convergence order are described and connected to well-known integration schemes including the Leapfrog method. (The mentioned splitting methods are not investigated here, they can be found in [1].)

Furthermore, some facts on differential equations on Lie groups are carried together.
Considering the aforementioned properties, partitioned Runge-Kutta methods for Lie groups are developed. During this process, there arise some difficulties: First of all, the solution of the differential equations of course has to be an element of the Lie group. This is achieved using a Munthe-Kaas method which replaces the differential equation in the Lie group through a differential equation in the appropriate Lie algebra and maps the result back in the Lie group via an exponential function. In this process, the differential equation is replaced by a truncated series which depends on the desired convergence order. Moreover, the integration method has to be symmetric. Due to the shape of the previously described mapping, this is none too easy. There occur problems concerning the exponential function in the mapping, such that an additional term is needed. Finally, the convergence order of the partitioned Runge-Kutta method is derived via Taylor expansions. Since the differential equation is a suitable truncation of a series, this has to be done for each desired convergence order separately. At the end of this chapter, there are two symmetric implicit partitioned Runge-Kutta methods with appropriate conditions for the symmetry and its convergence orders 2 and 3 derived. For the executed simulations, there are values for the coefficients needed. The coefficients for convergence order 2 are taken from [2]. For convergence order 3, they are chosen according to calculations performed with the computer algebra system Mathematica.

In chapter 4, the details of the model of an $SU(2, \mathbb{C})$ lattice gauge field are described. Then, the used fixed point iterations are discussed. In the last paragraph, the results of the executed simulations are presented which indicate, that a convergence order 4 can be achieved. These simulations are performed using the software package Matlab. For the investigation of the derived Runge-Kutta method, a lattice of size 4×4 with gauge coupling $\beta = 2.0$ is chosen. Furthermore, the dependence on the lattice volume and the gauge coupling are examined.

Chapter 1

Gauge Fields on a Lattice

The first chapter serves as a rough introduction to gauge fields on a lattice. The necessary information for understanding the numerical simulations will be provided here. Thereby, the sections concerning the gauge field and the Wilson action are described in [3] and [4]. Parts of section 1.3 can be found in [5].

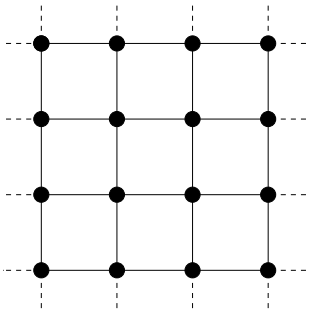

Figure 1.1: Lattice sites $[x]$. The field $[\phi]$ on a discrete lattice $[x]$ is given. It can be imagined as a color space at the sites x.

Let a field $[\phi]$ on a discrete lattice $[x]$ be given.
One of the fundamental objects of quantum field theory is the calculation of the expectation value $\langle A \rangle$ of some operator $A\Big([\phi]\Big)$. This is done via a path integral of the lattice action $S\Big([\phi]\Big)$

$$\langle A \rangle = \frac{1}{Z} \int [d\phi] \ \exp\Big(-S\Big([\phi]\Big)\Big) \, A\Big([\phi]\Big) \tag{1.1}$$

with partition function

$$Z = \int [d\phi] \ \exp\Big(-S\Big([\phi]\Big)\Big),$$

integration measure $[d\phi]$, and lattice action $S\Big([\phi]\Big)$. The lattice action $S\Big([\phi]\Big)$ has to fulfill two conditions. It must reach the correct continuum limit and has to be gauge invariant. This will be described more detailed in the following pragraphs.

1.1 Gauge Field

Let an equidistant lattice with lattice spacing a and periodic boundary conditions be given. We introduce a gauge field $[U]$ that is a set of matrices being elements of the special unitary Lie group $SU(N, \mathbb{C})$.

$$SU(N, \mathbb{C}) = \{X \in Gl(N, \mathbb{C}) : X^\dagger = X^{-1} \text{ and } det(X) = 1\}.$$

The matrices $U_{x,\mu}$ of the set $[U]$ represent the links between two adjacent lattice sites from site x in direction μ and are shown as arrows in figure 1.2.

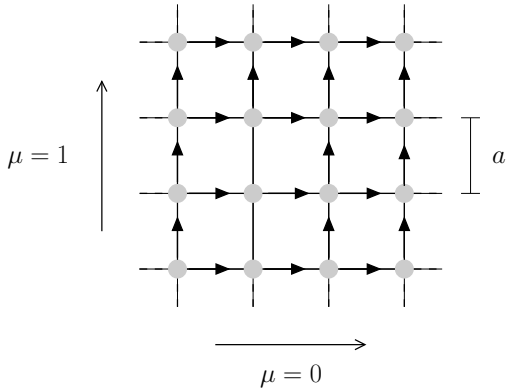

Figure 1.2: 2-dimensional field $[U]$ of lattice links on an equidistant lattice with lattice spacing a and periodic boundary conditions. The orientation is given as $\mu = 0, 1$.

Definition 1.1 (Link matrices)**.** *A matrix*

$$U_{x,\mu} \in SU(N, \mathbb{C})$$

is called link matrix. It is situated on the link from the lattice site x to its next neighbour $x + a\hat{\mu}$ in direction $\mu = 0, 1$ and will be shortly referred as link. $\hat{\mu}$ denotes the unit vector in direction μ, such that $\hat{0} = (1, 0)^T$ and $\hat{1} = (0, 1)^T$ holds. The vector $\hat{\mu}$ is scaled by the lattice constant a.

A link is orientated such that $U_{x,\mu}$ can be seen as a forward connection. Thus the link $U_{x+a\hat{\mu},-\mu}$ from site $x + a\hat{\mu}$ in the reverse direction $-\mu$ is a backward connection. They are related through $U_{x+a\hat{\mu},-\mu} = U^\dagger_{x,\mu}$ since the backward connection is simply the reversed forward one.

Figure 1.3: Left: Link $U_{x,\mu}$ from site x in direction μ. Right: Link $U^\dagger_{x,\mu}$ in the reverse direction (from site $x + a\mu$ in direction $-\mu$).

Definition 1.2 (Plaquette variable)**.** *The plaquette variable*

$$U_{01}(x) = U_{x,01} := U_{x,0}U_{x+a\hat{0},1}U^{\dagger}_{x+a\hat{1},0}U^{\dagger}_{x,1} \tag{1.2}$$

related to site x in the $(\hat{0},\hat{1})$-plane is the smallest closed loop of link matrices in counterclockwise direction. It is defined as the product over contiguous link matrices starting and ending at site x and also called plaquette.

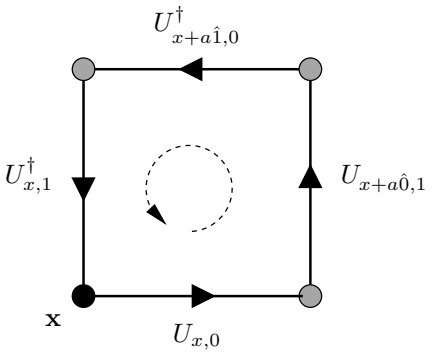

Figure 1.4: The plaquette is the shortest closed loop on the lattice.

Besides the plaquette, the staple is another important notation in quantum field theories. For every link $U_{x,\mu}$ from site x to site $x+a\hat{\mu}$ there exist shortest paths from site $x+a\hat{\mu}$ to site x not containing the link itself. These paths are called staples of the link $U_{x,\mu}$.

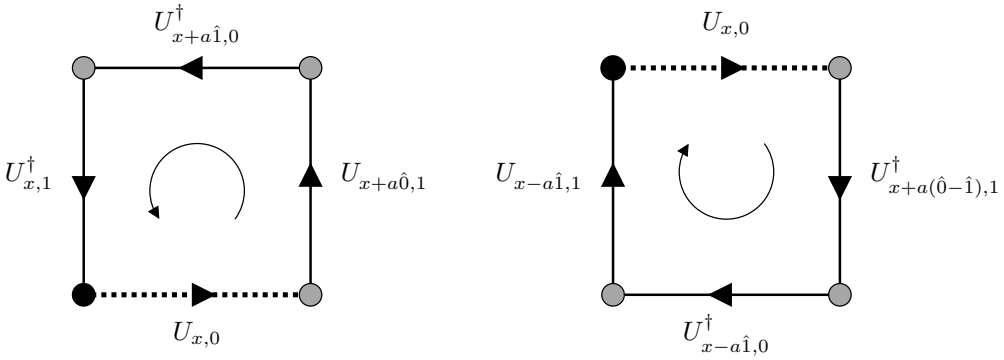

Figure 1.5: Staples of the link $U_{x,\mu}$.

Definition 1.3 (Staples)**.** *The sum of all shortest paths is called sum of staples and will be denoted with $V_{x,\mu} := V(U_{x,\mu})$. The sum of staples belonging to the link $U_{x,0}$, respective link $U_{x,1}$ are*

$$V_{x,0} := V(U_{x,0}) = U_{x+a\hat{0},1}U^{\dagger}_{x+a\hat{1},0}U^{\dagger}_{x,1} + U^{\dagger}_{x+a(\hat{0}-\hat{1}),1}U^{\dagger}_{x-a\hat{1},0}U_{x-a\hat{1},1}$$

resp.
$$V_{x,1} := V(U_{x,1}) = U_{x+a\hat{1},0}U^{\dagger}_{x+a\hat{0},1}U^{\dagger}_{x,0} + U^{\dagger}_{x+a(\hat{1}-\hat{0}),0}U^{\dagger}_{x-a\hat{0},1}U_{x-a\hat{0},0}$$

1.2 Wilson Action

The lattice action has to be gauge invariant. In this paragraph, we introduce the Wilson gauge action and show its gauge invariance. For this purpose, the expression gauge transformation has to be explained.

Definition 1.4 (Gauge transformation)**.** *Let* $[W]$ *be a set of matrices* W_x*,*

$$W_x \in SU(N, \mathbb{C}),$$

situated on the lattice sites. The gauge transformation

$$\phi_x \to \phi'_x = W_x \phi_x.$$

rotates the elements of the color space $[\phi]$*. The link matrices are transformed as*

$$U_{x,\mu} \to U'_{x,\mu} = W_x U_{x,\mu} W^\dagger_{x+a\hat{\mu}}.$$

Thus the gauge inner product $\phi^\dagger(x) U_{x,\mu} \phi(x + a\hat{\mu})$ is invariant under the gauge transformation because $\phi^\dagger_x U_{x,\mu} \phi_{x+a\hat{\mu}}$ transforms to

$$\begin{aligned}(\phi'_x)^\dagger U'_{x,\mu} \phi'_{x+a\hat{\mu}} &= \Big(W_x \phi_x\Big)^\dagger \cdot W_x U_{x,\mu} W^\dagger_{x+a\hat{\mu}} \cdot W_{x+a\hat{\mu}} \phi_{x+a\hat{\mu}} \\ &= \phi^\dagger_x \cdot W^\dagger_x W_x \cdot U_{x,\mu} \cdot W^\dagger_{x+a\hat{\mu}} W_{x+a\hat{\mu}} \cdot \phi_{x+a\hat{\mu}} \\ &= \phi^\dagger_x U_{x,\mu} \phi_{x+a\hat{\mu}}.\end{aligned}$$

The Wilson action is the lattice action corresponding to the plaquettes. It will be denoted as $S_G\Big([U]\Big)$ in the following.

Definition 1.5 (Wilson action)**.** *The Wilson action*

$$S_G\Big([U]\Big) = \sum_x \beta\left(1 - \frac{1}{N} \mathrm{Re}\Big(tr\Big(U_{01}(x)\Big)\Big)\right) \tag{1.3}$$

depends on the sum of the real part of the trace of all plaquettes. Note that this sum contains every plaquette with just one orientation. The factor β *is a hopping parameter and can be seen as inverse temperature.*

Remark 1.6 (Gauge invariance of the Wilson action)**.** *The Wilson action is gauge invariant if* $S_G\Big([U]\Big) = S_G\Big([U']\Big)$ *holds.*

Due to the fact that the Wilson action depend on the trace of the plaquette, the gauge invariance can be easily shown:

$$\begin{aligned}U'_{01}(x) &= U_{x,0} U_{x+a\hat{0},1} U^\dagger_{x+a\hat{1},0} U^\dagger_{x,1} \\ &= W_x U_{x,0} W^\dagger_{x+a\hat{0}} W_{x+a\hat{0}} U_{x+a\hat{0},1} W^\dagger_{x+a(\hat{0}+\hat{1})} (W_{x+a\hat{1}} U_{x+a\hat{1},0} W^\dagger_{x+a(\hat{1}+\hat{0})})^\dagger (W_x U_{x,1} W^\dagger_{x+a\hat{1}})^\dagger \\ &= W_x U_{x,0} W^\dagger_{x+a\hat{0}} W_{x+a\hat{0}} U_{x+a\hat{0},1} W^\dagger_{x+a(\hat{0}+\hat{1})} W_{x+a(\hat{1}+\hat{0})} U^\dagger_{x+a\hat{1},0} W^\dagger_{x+a\hat{1}} W_{x+a\hat{1}} U^\dagger_{x,1} W^\dagger_x \\ &= W_x U_{x,0} U_{x+a\hat{0},1} U^\dagger_{x+a\hat{1},0} U^\dagger_{x,1} W^\dagger_x \\ &= W_x U_{01}(x) W^\dagger_x.\end{aligned}$$

All except for the first and the last transformation matrices W_x vanish. Because of the properties of the trace, the trace of a closed path of link variabes is gauge invariant.

$$\begin{aligned} tr\Big(U'_{01}(x)\Big) &= tr\Big(W_x U_{01}(x) W_x^\dagger\Big) \\ &= tr\Big(W_x^\dagger W_x U_{01}(x)\Big) \\ &= tr\Big(U_{01}(x)\Big). \end{aligned}$$

Thus the Wilson action is gauge invariant. It is possible to create other gauge invariant expressions, for example by another closed path of links. For a small lattice spacing $a \to 0$ the continuum limit will be reached.

Remark 1.7 (Expectation value of a gauge field)**.** *We can transfer the expectation value $\langle A \rangle$ of equation* (1.1) *to the one of a bosonic gauge field $[U]$ with Wilson action $S_G\Big([U]\Big)$. It reads*

$$\langle A \rangle = \frac{1}{Z} \int [dU] \ \exp\Big(S_G\Big([U]\Big)\Big) \, A\Big([U]\Big) \tag{1.4}$$

with Haar measure $[dU]$ (see definition A.1) and partition function

$$Z = \int [dU] \ \exp\Big(-S_G\Big([U]\Big)\Big).$$

The expectation value $\langle A \rangle$ will be computed numerically. As we shall see later, we use a Hybrid Monte Carlo method to evaluate the path integral of equation (1.4). For this purpose, we need the Hamiltonian of the gauge field.

1.3 Hamilton Operator

The Hamiltonian $\mathcal{H}$ represents the total energy of the bosonic field. It is a conserved quantity, such that its time derivative $\dot{\mathcal{H}}$ vanishes. Furthermore, the Hamiltonian induces the Hamiltonian equations of motion.

Momenta

We introduce a field $[P]$ of fictious momenta on the lattice needed for the definition of the Hamiltonian. The momenta $[P]$ are associated with the links $[U]$. For every link matrix $U_{x,\mu}$ there exists a fictitious conjugated momentum $P_{x,\mu}$. These elements $P_{x,\mu}$ are traceless, and hermitian $(N \times N)$-matrices, i. e.

$$P_{x,\mu} = P_{x,\mu}^\dagger \qquad \text{and} \qquad tr\Big(P_{x,\mu}\Big) = 0.$$

The traceless and anti-hermitian matrices $iP_{x,\mu}$ are elements of the Lie algebra

$$\mathfrak{su}(N, \mathbb{C}) = \{X \in Gl(N) \mid X + X^\dagger = 0 \text{ and } tr\Big(X\Big) = 0\}.$$

associated to the Lie group $SU(N, \mathbb{C})$.

Definition 1.8. *The Hamiltonian*

$$\mathcal{H}\Big([U,P]\Big) = E_{kin}\Big([P]\Big) + S_G\Big([U]\Big) \tag{1.5}$$

depends on the set of link matrices $[U]$ *and its conjugated momenta* $[P]$*. It consists of the kinetic energy* E_{kin} *and the Wilson action* S_G*.*

The kinetic energy is associated with the set of momenta $[P]$ and reads

$$E_{kin}\Big([P]\Big) = \frac{1}{2}\sum_x \sum_{\mu=0,1} tr\Big(P_{x,\mu}^2\Big). \tag{1.6}$$

The Hamilton operator induces the Hamiltonian equations of motion:

$$\frac{\partial \mathcal{H}\Big([U,P]\Big)}{\partial U_{x,\mu}} = -\frac{\partial P_{x,\mu}}{\partial t} = -\dot{P}_{x,\mu}$$
$$\text{and} \qquad \frac{\partial \mathcal{H}\Big([U,P]\Big)}{\partial P_{x,\mu}} = \quad \frac{\partial U_{x,\mu}}{\partial t} = \quad \dot{U}_{x,\mu}.$$

Note that the time derivatives $\dot{U}_{x,\mu}$ and $\dot{P}_{x,\mu}$ do not depend on real time but on a fictious computer time. This means the Hamilton operator and its equations of motion are quite artificial because the momenta and the time are factitious. Nevertheless, we can develop formulas to compute the Hamiltonian equations of motion.

1.4 Hamiltonian Equations of Motion

The Hamiltonian equations of motion read

$$\frac{\partial \mathcal{H}\Big([U,P]\Big)}{\partial P_{x,\mu}} = \quad \dot{U}_{x,\mu} = iP_{x,\mu}U_{x,\mu} \tag{1.7}$$
$$\text{and} \qquad \frac{\partial \mathcal{H}\Big([U,P]\Big)}{\partial U_{x,\mu}} = -\dot{P}_{x,\mu} = -i\frac{\beta}{N}\Big\{U_{x,\mu}V_{x,\mu}\Big\}_{TA} \tag{1.8}$$

with traceless and anti-hermitian operator

$$\Big\{U_{x,\mu}V_{x,\mu}\Big\}_{TA} = \frac{1}{2}W_{x,\mu} - \frac{1}{2N}tr\Big(W_{x,\mu}\Big)\cdot I_N. \tag{1.9}$$

using $W_{x,\mu} = U_{x,\mu}V_{x,\mu} - V_{x,\mu}^\dagger U_{x,\mu}^\dagger$. The variable $V_{x,\mu}$ is called sum of staples as mentioned in definition 1.3.

1.4.1 Time Derivatives of the Links

The differential equation

$$\dot{U}_{x,\mu} = iP_{x,\mu}\cdot U_{x,\mu}. \tag{1.10}$$

can be motivated from the structure of the links and the momenta as follows: The elements $U_{x,\mu}$ of the Lie group $SU(N,\mathbb{C})$ ca be identified with the $(N+1)$-dimensional unit sphere

$$S_{N+1} = \{x \in \mathbb{R}^{N+1} \text{ with } \|x\|_2 = 1\}.$$

It is possible to define a small rotation

$$R : SU(N,\mathbb{C}) \to SU(N,\mathbb{C}), \qquad U_{x,\mu} \mapsto \tilde{U}_{x,\mu} = R \cdot U_{x,\mu} \tag{1.11}$$

on the unit sphere S_{N+1}. Every small rotation can be seen as infinitesimal transformation

$$R = I_N + \delta R \tag{1.12}$$

which consists of the identity and an infinitesimal rotation δR with $\delta \to 0$.
It follows

$$\tilde{U}_{x,\mu} \overset{(1.11)}{=} R \cdot U_{x,\mu} \overset{(1.12)}{=} (I_N + \delta R) \cdot U_{x,\mu} = U_{x,\mu} + \delta R \cdot U_{x,\mu} = U_{x,\mu} + \dot{U}_{x,\mu}$$

and thus

$$\dot{U}_{x,\mu} = \delta R \cdot U_{x,\mu}. \tag{1.13}$$

Lemma 1.9. *The infinitesimal rotation matrices δR are elements of the Lie algebra $\mathfrak{su}(N,\mathbb{C})$. They can be identified with the momenta $P_{x,\mu}$ via*

$$\delta R = iP_{x,\mu}. \tag{1.14}$$

Proof. We show that the infinitesimal rotation matrices δR have to be traceless and anti-hermitian.
Because we use the Lie group properties, the fact that the rotation matrix R is situated in the special unitary Lie group is the crucial point in this proof.

- Due to the fact that R is unitary, δR has to be anti-hermitian: The inverse of R reads

  $$R^{-1} = R^\dagger = (I_N + \delta R)^\dagger = I_N + \delta R^\dagger.$$

  It holds

  $$I_N = R \cdot R^{-1} = (I_N + \delta R) \cdot (I_N + \delta R^\dagger) = I_N + \delta R^\dagger + \delta R + \delta^2 R R^\dagger.$$

  After a linearization we get

  $$I_N \doteq I_N + \delta R^\dagger + \delta R$$

  and thus δR has to be anti-hermitian, i. e. $\delta R = -\delta R^\dagger$.

- Since R is a special matrix, i. e. $\det(R) = 1$, it follows that δR has to be traceless: Using the identity

  $$\det\Big(\exp(\delta R)\Big) = \exp\Big(tr\Big(\delta R\Big)\Big)$$

and the Taylor expansion

$$R = I_N + \delta R + \frac{1}{2}\Big(\delta R\Big)^2 + \ldots = \exp(\delta R)$$

of R, we get

$$1 = \det(R) = \det\Big(\exp(\delta R)\Big) = \exp\Big(tr\Big(\delta R\Big)\Big).$$

Hence, δR has to be traceless.

□

The equation of motion

$$\dot{U}_{x,\mu} = iP_{x,\mu} \cdot U_{x,\mu}. \tag{1.10}$$

immediately follows from (1.13) and (1.14).

1.4.2 Time Derivatives of the Momenta

The derivation of the equation of motion for the momenta needs a few considerations: Due to the energy conservation, it holds

$$\dot{\mathcal{H}}\Big([U,P]\Big) = \dot{E}_{kin}\Big([P]\Big) + \dot{S}_G\Big([U]\Big) = 0.$$

Thus we have to deduce the time derivatives of the kinetic energy and the Wilson action to get a concrete formula for $\dot{\mathcal{H}}$: While it is evident that

$$\begin{aligned}\dot{E}_{kin}\Big([P]\Big) &= \frac{1}{2}\sum_x \sum_{\mu=0,1} tr\Big(\dot{P}_{x,\mu}P_{x,\mu} + P_{x,\mu}\dot{P}_{x,\mu}\Big) \\ &= \sum_x \sum_{\mu=0,1} tr\Big(\dot{P}_{x,\mu}P_{x,\mu}\Big)\end{aligned} \tag{1.15}$$

holds, the calculation of the derivative of the Wilson action

$$\dot{S}_G\Big([U]\Big) = -\frac{\beta}{2N}\sum_x \sum_{\mu=0,1} tr\Big(\dot{U}_{x,\mu}V_{x,\mu} + V^\dagger_{x,\mu}\dot{U}^\dagger_{x,\mu}\Big) \tag{1.16}$$

requires some additional work.

The derivative of the Wilson action

We start with the Wilson action and use the property (A.4) of the trace:

$$\begin{aligned}S_G\Big([U]\Big) &= \sum_x \beta\Bigg(1 - \frac{1}{N}\,\mathrm{Re}\Big(tr\Big(U_{01}(x)\Big)\Big)\Bigg) \\ &= \sum_x \beta\Bigg(1 - \frac{1}{2N} tr\Big(U_{01}(x) + U^\dagger_{01}(x)\Big)\Bigg).\end{aligned}$$

Then, we replace the plaquette variable $U_{01}(x)$ with the product of links (see equation (1.2)) and rewrite the Wilson action as

$$\begin{aligned} S_G\Big([U]\Big) &= -\frac{\beta}{2N}\sum_x \Big(tr\Big(U_{x,0}U_{x+a\hat{0},1}U^\dagger_{x+a\hat{1},0}U^\dagger_{x,1} + \Big(U_{x,0}U_{x+a\hat{0},1}U^\dagger_{x+a\hat{1},0}U^\dagger_{x,1}\Big)^\dagger\Big)\Big) \\ &= -\frac{\beta}{2N}\sum_x \Big(tr\Big(U_{x,0}U_{x+a\hat{0},1}U^\dagger_{x+a\hat{1},0}U^\dagger_{x,1} + U_{x,1}U_{x+a\hat{1},0}U^\dagger_{x+a\hat{0},1}U^\dagger_{x,0}\Big)\Big). \end{aligned}$$

The time derivative $\dot{S}_G\Big([U]\Big)$ of the Wilson action can be computed in a straightforward way. Because a reordering of the matrix product has no effect on the trace (see equation (A.3)), the time derivations of the links are placed at the beginning and the end of a product of links.

$$\begin{aligned} \dot{S}_G\Big([U]\Big) = -\frac{\beta}{2N}\sum_x tr\Big(& \dot{U}_{x,0}U_{x+a\hat{0},1}U^\dagger_{x+a\hat{1},0}U^\dagger_{x,1} + \dot{U}_{x,1}U_{x+a\hat{1},0}U^\dagger_{x+a\hat{0},1}U^\dagger_{x,0} \\ & + \dot{U}_{x+a\hat{0},1}U^\dagger_{x+a\hat{1},0}U^\dagger_{x,1}U_{x,0} + \dot{U}_{x+a\hat{1},0}U^\dagger_{x+a\hat{0},1}U^\dagger_{x,0}U_{x,1} \\ & + U^\dagger_{x,1}U_{x,0}U_{x+a\hat{0},1}\dot{U}^\dagger_{x+a\hat{1},0} + U^\dagger_{x,0}U_{x,1}U_{x+a\hat{1},0}\dot{U}^\dagger_{x+a\hat{0},1} \\ & + U_{x,0}U_{x+a\hat{0},1}U^\dagger_{x+a\hat{1},0}\dot{U}^\dagger_{x,1} + U_{x,1}U_{x+a\hat{1},0}U^\dagger_{x+a\hat{0},1}\dot{U}^\dagger_{x,0}\Big) \end{aligned} \tag{1.17}$$

Taking account of some additional information, we simplify this expression. Since we sum over all lattice sites x, we can shift some indices. This will be done in such a way that we can factorise the time derivatives of the links, e.g.

$$x + a\hat{1}, 0 \to x, 0 \quad \text{and} \quad x + a\hat{0}, 1 \to x, 1.$$

The shifts in direction -0 and -1 imply the changes

$$\begin{aligned} x + a\hat{0}, 1 &\xrightarrow{-0} x, 1 & \qquad x + a\hat{1}, 0 &\xrightarrow{-1} x, 0, \\ x + a\hat{1}, 0 &\xrightarrow{-0} x + a(\hat{1} - \hat{0}), 0, & \qquad x + a\hat{0}, 1 &\xrightarrow{-1} x + a(\hat{0} - \hat{1}), 1, \\ x, 1 &\xrightarrow{-0} x - a\hat{0}, 1, & \qquad x, 0 &\xrightarrow{-1} x - a\hat{1}, 0, \\ x, 0 &\xrightarrow{-0} x - a\hat{0}, 0, & \qquad x, 1 &\xrightarrow{-1} x - a\hat{1}, 1. \end{aligned}$$

Using these shifts and equation (1.17), we get

$$\begin{aligned} &\sum_x tr\Big(\dot{U}_{x,0}U_{x+a\hat{0},1}U^\dagger_{x+a\hat{1},0}U^\dagger_{x,1} + \dot{U}_{x+a\hat{1},0}U^\dagger_{x+a\hat{0},1}U^\dagger_{x,0}U_{x,1}\Big) \\ &\qquad = \sum_x tr\Big(\dot{U}_{x,0}\Big(U_{x+a\hat{0},1}U^\dagger_{x+a\hat{1},0}U^\dagger_{x,1} + U^\dagger_{x+a(\hat{0}-\hat{1}),1}U^\dagger_{x-a\hat{1},0}U_{x-a\hat{1},1}\Big)\Big) \end{aligned}$$

$$\begin{aligned} \text{and} \quad &\sum_x tr\Big(\dot{U}_{x,1}U_{x+a\hat{1},0}U^\dagger_{x+a\hat{0},1}U^\dagger_{x,0} + \dot{U}_{x+a\hat{0},1}U^\dagger_{x+a\hat{1},0}U^\dagger_{x,1}U_{x,0}\Big) \\ &\qquad = \sum_x tr\Big(\dot{U}_{x,1}\Big(U_{x+a\hat{1},0}U^\dagger_{x+a\hat{0},1}U^\dagger_{x,0} + U^\dagger_{x+a(\hat{1}-\hat{0}),0}U^\dagger_{x-a\hat{0},1}U_{x-a\hat{0},0}\Big)\Big). \end{aligned}$$

The inner sums of the expressions above consist of the sum of staples of the links $U_{x,0}$ and $U_{x,1}$. This implies

$$\sum_x tr\Big(\dot{U}_{x,0}U_{x+a\hat{0},1}U^\dagger_{x+a\hat{1},0}U^\dagger_{x,1} + \dot{U}_{x+a\hat{1},0}U^\dagger_{x+a\hat{0},1}U^\dagger_{x,0}U_{x,1}\Big) = \sum_x tr\Big(\dot{U}_{x,0}V_{x,0}\Big)$$

and
$$\sum_x tr\Big(\dot{U}_{x,1}U_{x+a\hat{1},0}U^\dagger_{x+a\hat{0},1}U^\dagger_{x,0} + \dot{U}_{x+a\hat{0},1}U^\dagger_{x+a\hat{1},0}U^\dagger_{x,1}U_{x,0}\Big) = \sum_x tr\Big(\dot{U}_{x,1}V_{x,1}\Big).$$

The other 4 addends of equation (1.17) lead to the inverse expressions

$$\sum_x tr\Big(U^\dagger_{x,1}U_{x,0}U_{x+a\hat{0},1}\dot{U}^\dagger_{x+a\hat{1},0} + U_{x,1}U_{x+a\hat{1},0}U^\dagger_{x+a\hat{0},1}\dot{U}^\dagger_{x,0}\Big) = \sum_x tr\Big(V^\dagger_{x,0}\dot{U}^\dagger_{x,0}\Big)$$

and
$$\sum_x tr\Big(U^\dagger_{x,0}U_{x,1}U_{x+a\hat{1},0}\dot{U}^\dagger_{x+a\hat{0},1} + U_{x,0}U_{x+a\hat{0},1}U^\dagger_{x+a\hat{1},0}\dot{U}^\dagger_{x,1}\Big) = \sum_x tr\Big(V^\dagger_{x,1}\dot{U}^\dagger_{x,1}.\Big)$$

The insertion of these equations in the derivative of the Wilson action of equation (1.17) yields

$$\begin{aligned}\dot{S}_G\Big([U]\Big) &= -\frac{\beta}{2N}\sum_x tr\Big(\dot{U}_{x,0}V_{x,0} + \dot{U}_{x,1}V_{x,1} + V^\dagger_{x,0}\dot{U}^\dagger_{x,0} + V^\dagger_{x,1}\dot{U}^\dagger_{x,1}\Big)\\ &= -\frac{\beta}{2N}\sum_x\sum_{\mu=0,1} tr\Big(\dot{U}_{x,\mu}V_{x,\mu} + V^\dagger_{x,\mu}\dot{U}^\dagger_{x,\mu}\Big)\end{aligned}$$

and coincides with equation (1.16).

The derivative of the Hamiltonian

We can outline the derivative of the Hamiltonian

$$0 = \dot{\mathcal{H}}\Big([U,P]\Big) = \dot{E}_{kin}\Big([P]\Big) + \dot{S}_G\Big([U]\Big)$$

using the kinetic energy (1.15) and the Wilson action (1.16):

$$\dot{\mathcal{H}}\Big([U,P]\Big) = \sum_x\sum_{\mu=0,1} tr\Big(\dot{P}_{x,\mu}P_{x,\mu}\Big) - \frac{\beta}{2N}\sum_x\sum_{\mu=0,1} tr\Big(\dot{U}_{x,\mu}V_{x,\mu} + V^\dagger_{x,\mu}\dot{U}^\dagger_{x,\mu}\Big).$$

Having in mind that we are seeking for the derivative of the momenta $\dot{P}_{x,\mu}$, we replace $\dot{U}_{x,\mu}$ in $\dot{S}_G\Big([U]\Big)$ with $iP_{x,\mu}U_{x,\mu}$. Again, we use the property (A.3) of the trace. Furthermore, $P_{x,\mu}$ is hermitian, i. e. $P_{x,\mu} = P^\dagger_{x,\mu}$. Thus the derivative of the Wilson action is rearranged as

$$\begin{aligned}\dot{S}_G\Big([U]\Big) &= -\frac{\beta}{2N}\sum_x\sum_{\mu=0,1} tr\Big(iP_{x,\mu}U_{x,\mu}V_{x,\mu} + V^\dagger_{x,\mu}\Big(iP_{x,\mu}U_{x,\mu}\Big)^\dagger\Big)\\ &= -\frac{\beta}{2N}\sum_x\sum_{\mu=0,1} tr\Big(iP_{x,\mu}U_{x,\mu}V_{x,\mu} - iV^\dagger_{x,\mu}U^\dagger_{x,\mu}P^\dagger_{x,\mu}\Big)\\ &= -i\frac{\beta}{2N}\sum_x\sum_{\mu=0,1} tr\Big(\Big(U_{x,\mu}V_{x,\mu} - V^\dagger_{x,\mu}U^\dagger_{x,\mu}\Big)P_{x,\mu}\Big)\end{aligned}$$

and inserted in the formula of the derivative of the Hamiltonian such that

$$\begin{aligned}\dot{\mathcal{H}}\Big([U,P]\Big) &= \sum_x \sum_{\mu=0,1} tr\Big(\dot{P}_{x,\mu}P_{x,\mu} - i\frac{\beta}{2N}\Big(U_{x,\mu}V_{x,\mu} - V_{x,\mu}^\dagger U_{x,\mu}^\dagger\Big)P_{x,\mu}\Big) \\ &= \sum_x \sum_{\mu=0,1} tr\Big(\Big(\dot{P}_{x,\mu} - i\frac{\beta}{2N}\Big(U_{x,\mu}V_{x,\mu} - V_{x,\mu}^\dagger U_{x,\mu}^\dagger\Big)\Big)P_{x,\mu}\Big)\end{aligned}$$

holds. Then we use the shortcut

$$F_{x,\mu} := -i\frac{\beta}{2N}\Big(U_{x,\mu}V_{x,\mu} - V_{x,\mu}^\dagger U_{x,\mu}^\dagger\Big) \tag{1.18}$$

and insert it in the formula above:

$$0 = \dot{\mathcal{H}}\Big([U,P]\Big) = \sum_x \sum_{\mu=0,1} tr\Big(\Big(\dot{P}_{x,\mu} + F_{x,\mu}\Big)P_{x,\mu}\Big).$$

The derivative of the momentum

It follows the sufficient condition

$$0 = tr\Big(\Big(\dot{P}_{x,\mu} + F_{x,\mu}\Big)P_{x,\mu}\Big). \tag{1.19}$$

for all sites x and all directions μ. Then we can determine the form of $\dot{P}_{x,\mu}$ with a consideration of the properties of the trace.
The most important feature is that the momentum $P_{x,\mu}$ is traceless. Since equation (1.19) holds for all traceless and hermitian matrices $P_{x,\mu}$, the matrix $P_{x,\mu}$ has to be multiplied with a constant times the identity,

$$\dot{P}_{x,\mu} + F_{x,\mu} = c \cdot I_N,$$

with so far unknown constant c. This means

$$\dot{P}_{x,\mu} = c \cdot I_N - F_{x,\mu}. \tag{1.20}$$

Due to the fact that the derivative $\dot{P}_{x,\mu}$ of the momentum $P_{x,\mu}$ also has to be traceless, we can determine the constant via equation (1.20) considering the trace:

$$tr\Big(\dot{P}_{x,\mu}\Big) = tr\Big(c \cdot I_N - F_{x,\mu}\Big) = tr\Big(c \cdot I_N\Big) - tr\Big(F_{x,\mu}\Big) = c \cdot N - tr\Big(F_{x,\mu}\Big).$$

With $tr\Big(\dot{P}_{x,\mu}\Big) = 0$ we yield

$$c = \frac{1}{N} tr\Big(F_{x,\mu}\Big)$$

and get the result

$$\dot{P}_{x,\mu} = \frac{1}{N} tr\Big(F_{x,\mu}\Big) \cdot I_N - F_{x,\mu}$$

which can be rewritten by means of (1.18) as

$$\dot{P}_{x,\mu} = i\frac{\beta}{2N}\Big(U_{x,\mu}V_{x,\mu} - V^{\dagger}_{x,\mu}U^{\dagger}_{x,\mu}\Big) - \frac{1}{N}tr\Big(i\frac{\beta}{2N}\Big(U_{x,\mu}V_{x,\mu} - V^{\dagger}_{x,\mu}U^{\dagger}_{x,\mu}\Big)\Big)\cdot I_N$$

With help of the notation of the traceless anti-hermitian operator of equation (1.9) the equation of motion concerning the momenta amounts to

$$\dot{P}_{x,\mu} = i\frac{\beta}{N}\Big\{U_{x,\mu}V_{x,\mu}\Big\}_{TA}. \tag{1.8}$$

Hence, the Hamiltonian equations of motion are derived.

Chapter 2

Hybrid Monte Carlo Method

Our aim is to calculate the expectation value $\langle A \rangle$ (see equation (1.4)) of an observable A over an ensemble of gauge field configurations, i. e.

$$\langle A \rangle = \sum_{\{U\}} A(U) \frac{\exp\Big(-S\big([U]\big)\Big)}{Z}, \tag{2.1}$$

where the partition function reads

$$Z = \sum_{\{U\}} \exp\Big(-S\big(U\big)\Big).$$

Note that $S\big([U]\big)$ denotes the Wilson action and $\{U\}$ means all configurations. We assume that the expectation value can not be calculated directly, so it has to be computed by a numerical simulation. Thereby, the occurence of almost all configurations is very small such that we need many configurations to reach the correct expectation value $\langle A \rangle$. To scale down the computational effort, we need a more sophisticated method called importance sampling. This is implemented by choosing configurations according to the Boltzmann-distributed probability

$$p_i := p\big([U^i]\big) = \frac{1}{Z} \exp\Big(-S\big([U^i]\big)\Big) \tag{2.2}$$

such that configurations $[U^i]$ occurring with a high probability will be preferred. This approach is realized in the Metropolis Monte Carlo method and can be improved to a Hybrid Monte Carlo method. The original idea of the Hybrid Monte Carlo method has been described in [6]. It can also be found in [4] and [7]. Section 2.1 is taken in large parts from [7]. For section 2.2, the references [6] and [4] are used.

2.1 Metropolis Monte Carlo Method

The Metropolis Monte Carlo method is based on the principle of a Markov process which is described below.

2.1.1 Markov Process

A Markov process is a stochastic method which generates a new configuration from one or more old configurations. Thereby, the probability $p_i^{(k)}$ to find configuration $[U^i]$ at time point k depends only on the previous configurations themselves and not on the point in time. If the new configuration depends only on its predecessor the Markov process is called Markov process of first order.

Definition 2.1 (Stochastic matrix)**.** *A matrix T is called stochastic if and only if its elements are greater than or equal to zero and if the sum over each row is equal to one, i. e.*

$$T_{ij} \geq 0 \quad \forall i,j \qquad \text{and} \qquad \sum_j T_{ij} = 1 \quad \forall i.$$

Theorem 2.2 (Perron-Frobenius)**.** *Let T be a stochastic matrix. Then there exists a non-negative vector v with $vT = v$. This means v is a (left-) eigenvector of T with eigenvalue 1. If all elements T_{ij} are larger than* 0*, the eigenvalue* 1 *is a simple one.*

Let $\{[U]\}$ be a set of configurations. Then the Markov process will be created with help of the transition probabilites $T\Big([U^i] \to [U^j]\Big) =: T_{ij}$ such that the Markov chain reads

$$[U^0] \xrightarrow{T_{01}} [U^1] \xrightarrow{T_{12}} [U^2] \xrightarrow{T_{23}} \dots \tag{2.3}$$

with indices $0, 1, 2, \dots$ representing the elements of an index set.
We are interested in the equilibrium distribution of the fields $[U]$, which is the fixed point of the Markov process (2.3) under the conditions described as follows. First of all, the transition propabilities have to be ergodic, which means that every possible configuration can be reached with a certain probability from any other one. This property will be denoted with $T_{ij} \geq 0$. Furthermore, the stability criterion

$$\sum_i p_i^{(k+1)} T_{ij} = p_j^{(k)} \quad \forall j \tag{2.4}$$

has to be fulfilled. The stability criterion can be written in a more compact form as

$$p^{(k+1)} = p^{(k)} T \tag{2.5}$$

with row vector $p^{(k)} = \Big(p_1^{(k)}, p_2^{(k)}, \dots\Big)$ and transition matrix T. The row vector $p^{(k)}$ represents the probability distribution of the configurations after k steps in the Markov chain. Due to the theorem of Perron-Frobenius we get for $\lim_{k\to\infty} p^{(k)} = p$ the fixed point equation $pT = p$. With strong ergodicity $T_{ij} > 0$, the fixed point p will be unique.
The aim of the Metropolis algorithm is to construct the transition matrix T to reach a prescribed equilibrium distribution p, in our case the Boltzmann distribution (see equation (2.2)). For this purpose, we use the detailed balance

condition, which is a stronger requirement than the stability criterion, but easier to handle:

$$p_i T_{ij} = p_j T_{ji}. \tag{2.6}$$

Note that the detailed balance condition is not necessary but sufficient. We obtain the stability criterion (2.4) if we sum over i and use definition of a stochastic matrix (see definition 2.1):

$$\sum_i p_i T_{ij} = \sum_i p_j T_{ji} = p_j \sum_i T_{ji} = p_j.$$

2.1.2 Metropolis Algorithm

With the aforementioned concepts we can introduce the Metropolis algorithm and analyze the acceptance and update step.

Algorithm 2.1 (Metropolis Algorithm)**.** *Let p_i, resp. p_j denote the probability for the occurence of the configuration $[U^i]$, resp. $[U^j]$. The transition matrix will be created from an initial field configuration $[U^i]$ by alternating an update and an acceptance step:*

1. *Update step:*
   Create a test configuration $[U^j]$ randomly.
2. *Acceptance step:*
   Accept the configuration $[U^j]$ with transition probability

$$T_{ij} = min\Big(1, \frac{p_j}{p_i}\Big). \tag{2.7}$$

Acceptance step

The acceptance step has to fulfill the stability criterion (2.4) to reach the fixed point of the Markov process. Indeed, it satisfies the detailed balance condition (2.6) because it holds

$$p_i T_{ij} = p_i \cdot \min\Big(1, \frac{p_j}{p_i}\Big) = \min\Big(p_i, p_j\Big) = p_j \cdot \min\Big(\frac{p_i}{p_j}, 1\Big) = p_j T_{ji}.$$

Hence, with strong ergodicity ($T_{ij} > 0$ for all i, j) the Markov process will tend to a unique fixed point. Concerning the action of the model, the fixed point should be Boltzmann distributed, i. e. the probability for each configuration $[U^i]$ reads

$$p_i = \frac{1}{Z} \cdot \exp\Big(-S\Big([U^i]\Big)\Big).$$

With $\Delta S := S\Big([U^j]\Big) - S\Big([U^i]\Big)$ the transition probability can be rewritten as

$$T_{ij} = \min\Big(1, \exp\Big(-\Delta S\Big)\Big). \tag{2.8}$$

This means the new configuration will always be accepted if the action has become smaller. Otherwise, a uniformly distributed random number $r \in [0, 1]$ has to be generated and the new configuration will be accepted if $r < T_{ij}$ holds.

Update step

The update can be carried out by changing one or more elements of the old configuration. If we replace just one element we call it local update. In doing so, we will get a high acceptance rate since the difference of two actions will be small.
Refreshing all elements of a given configuration is called global update and produces a lower acceptance rate because the average difference of two successive actions $S([U^i])$ and $S([U^j])$ will be relatively large. Thus the local update will naturally be favoured in a Metropolis Monte Carlo algorithm unless the acceptance step of a local update will cause as much computational effort as the acceptance step of a global update.

2.2 Hybrid Monte Carlo Method

In quantum field theories, we can distinguish between two different sorts of fields, the bosonic and the fermionic field. If we do not just consider the bosonic but also the fermionic field, the acceptance step comprises the inversion the of a large matrix called Dirac operator. Unfortunately, this inversion needs much computational time such that one global update will be preferred instead of many local ones.
Nevertheless to reach a high acceptance rate the idea is to replace the action $S[U]$ with the Hamiltonian $\mathcal{H}([U,P])$ (see definition 1.8) and combine the Metropolis method with a Molecular Dynamics method. For this purpose, we use

$$p_i = \frac{1}{Z} \cdot \exp\Big(-\mathcal{H}\Big([U^i, P^i]\Big)\Big) \tag{2.9}$$

as new probability distribution. Since the Hamiltonian is a constant in time the advantages of getting a high acceptance rate and saving many matrix inversions by performing a global update are combined.

Molecular Dynamics Method

In the Molecular Dynamics method, the Hamiltonian equations of motion

$$\frac{\partial \mathcal{H}\Big([U(t),P(t)]\Big)}{\partial U} = -\frac{\partial P(t)}{\partial t} = -\dot{P}(t)$$

$$\text{and} \qquad \frac{\partial \mathcal{H}\Big([U(t),P(t)]\Big)}{\partial P} = \frac{\partial U(t)}{\partial t} = \dot{U}(t)$$

will be calculated by numerical integration. These equations define a trajectory $[U(t),P(t)]$ through phase space where the variable t denotes the fictious time. Starting from an initial configuration $[U^i,P^i]$ at time $t_0 = 0$, the new configuration $[U^j,P^j]$ at time t will be obtained via a numerical integration through phase space.
Because the Hamiltonian is conserved in time, the Hamiltonians of two successive configurations will be the same up to the numerical errors of the integration method.

Algorithm 2.2 (Hybrid Monte Carlo Algorithm)**.** *During the Hybrid Monte Carlo algorithm the following steps will be carried out:*

1. *Select an initial configuration* $[U^i]$ *randomly.*
2. *Create conjugated momenta* $[P^i]$ *randomly according to equation* (2.11)*.*
3. *Reach the new configuration* $[U^j, P^j]$ *by performing a Molecular Dynamics step.*
4. *Compute the difference of the Hamiltonians as* $\Delta\mathcal{H} := \mathcal{H}^j - \mathcal{H}^i$*.*
5. *Accept the new configuration with acceptance probability*
$$P_A\Big([U^i, P^i] \to [U^j, P^j]\Big) = \min\Big(1, \frac{p_j}{p_i}\Big) \overset{(2.9)}{=} \min\Big(1, \exp(-\Delta\mathcal{H})\Big). \quad (2.10)$$
6. *Start at step 2.*

Choice of the initial field and momenta refreshment

The initial field configuration $[U]$ can be chosen arbitrarily, for example with uniformly distributed random numbers, because the system will converge to the unique fixed point.
The conjugated momenta $[P]$ are chosen at random from a Gaussian distribution
$$\mathcal{P}_G([P]) \sim \exp\Big(-\frac{1}{2}\sum_{x,\mu} Tr(P_{x,\mu}^2)\Big) = \exp\Big(-E_{kin}\Big([P]\Big)\Big) \quad (2.11)$$
with mean 0 and variance 1. It is important to generate a new field of momenta in each step. This has to be done regardless of acceptance or rejection of the new configuration to ensure ergodicity.

Acceptance step

During the acceptance step, the configuration $[U^i, P^i]$ and its successor $[U^j, P^j]$ have to be considered. The total energy in terms of the Hamiltonian
$$\mathcal{H}\Big([U, P]\Big) = E_{kin}\Big([P]\Big) + S\Big([U]\Big)$$
from equation (1.5) is used to decide whether the new configuration will be accepted or rejected.
If the energy becomes smaller, the new configuration $[U^j, P^j]$ is always accepted. Otherwise a uniformly distributed random number $r \in [0, 1)$ has to be generated. In the case of r being smaller or equal than $P_A\Big([U^i, P^i] \to [U^j, P^j]\Big)$ given in equation (2.10) the new configuration is also accepted. The configuration $[U^i]$ is replaced by $[U^j]$ to proceed in the next step. To save computing time the value of the action $S([U^i])$ can be changed to $S([U^j])$.
Whenever the acceptance probability $P_A\Big([U^i, P^i] \to [U^j, P^j]\Big)$ is smaller than the random number r, the new configuration is dismissed and the old one $[U^i]$ is used again in the next step.

Convergence of the Markov chain

It is essential for the Hybrid Monte Carlo method that the Markov process converges to the fixed point of the equlibrium distribution of the field configurations $[U]$.To ensure this, the numerical integration scheme has to fulfill the detailed balance condition concerning the action $S([U])$, i. e.

$$p_i T_{ij} = p_j T_{ji} \qquad \text{with} \quad p_i \sim \exp\Big(-S\Big([U^i]\Big)\Big).$$

Again, the total transition probability to reach configuration $[U^j]$ from $[U^i]$ is denoted with T_{ij}. Compared with the transition probability of the Metropolis Monte Carlo method, the transition probability T_{ij} for the Hybrid Monte Carlo method is a complicated expression. It is obtained via an integration of the different probabilities $\mathcal{P}_A, \mathcal{P}_G$ and $\mathcal{P}_M$ over the conjugated momenta:

$$T_{ij} = \int [dP^i dP^j]\, \mathcal{P}_A\Big([U^i, P^i] \to [U^j, P^j]\Big) \mathcal{P}_G([P^i])\, \mathcal{P}_M\Big([U^i, P^i] \to [U^j, P^j]\Big). \tag{2.12}$$

$\mathcal{P}_M\Big([U^i, P^i] \to [U^j, P^j]\Big)$ denotes the probability to reach the new configuration $[U^j, P^j]$ from the old one $[U^j, P^j]$ via the molecular dynamics step.

Lemma 2.3. *(Time reversibility of the integrator) The numerical method used in the Hybrid Monte Carlo method has to be time-reversible, (i. e. in mathematical notation symmetric) to fulfill the detailed balance condition*

$$p_i T_{ij} = p_j T_{ji}$$

with $p_i \sim \exp\Big(-S\Big([U^i]\Big)\Big)$ *and transition probability.*

$$T_{ij} = \int [dP^i dP^j]\, \mathcal{P}_A\Big([U^i, P^i] \to [U^j, P^j]\Big) \mathcal{P}_G([P^i])\, \mathcal{P}_M\Big([U^i, P^i] \to [U^j, P^j]\Big).$$

Proof. The time-reversibility means that the probability to attain configuration $[U^j, P^j]$ from configuration $[U^i, P^i]$ is the same as the probability to get $[U^i, -P^i]$ from the start configuration $[U^j, -P^j]$ with reversed momenta $[-P^j]$. It is essential for the proof of the detailed balance condition (2.6) that the numerical integration scheme for solving the equations of motion is symmetric or time-reversible. This means that the mapping $[U^i, P^i] \to [U^j, P^j]$ is reversible in the sense that

$$\mathcal{P}_M\Big([U^i, P^i] \to [U^j, P^j]\Big) = \mathcal{P}_M\Big([U^j, -P^j] \to [U^i, -P^i]\Big). \tag{2.13}$$

Furthermore, it holds

$$p_i \mathcal{P}_G\Big([P^i]\Big) = \exp\Big(-\mathcal{H}([U^i, P^i])\Big)$$

since $S\Big([U^i]\Big) + E_{kin\Big([P^i]\Big)} = \mathcal{H}\Big([U^i, P^i]\Big)$. Using the formula (2.10) for the

acceptance probability, we attain

$$p_i \cdot \mathcal{P}_G([P^i]) \cdot P_A\Big([U^i, P^i] \to [U^j, P^j]\Big) = \exp\Big(-\mathcal{H}\Big([U^i, P^i]\Big)\Big) \cdot \min\Big(1, \exp(-\Delta\mathcal{H})\Big) = \exp\Big(-\mathcal{H}\Big([U^j, P^j]\Big)\Big) \cdot \min\Big(\exp\Big(\Delta\mathcal{H}, 1\Big) = p_j \cdot \mathcal{P}_G([P^j]) \cdot P_A\Big([U^j, P^j] \to [U^i, P^i]\Big).$$

Thus, we yield with equation (2.13)

$$p_i T_{ij} = p_i \int [dP^i dP^j]\, \mathcal{P}_G([P^i])\, \mathcal{P}_A\Big([U^i, P^i] \to [U^j, P^j]\Big) \mathcal{P}_M\Big([U^i, P^i] \to [U^j, P^j]\Big)$$
$$= p_j \int [dP^i dP^j]\, \mathcal{P}_G([P^j]) \cdot P_A\Big([U^j, P^j] \to [U^i, P^i]\Big)\, \mathcal{P}_M\Big([U^j, -P^j] \to [U^i, -P^i]\Big).$$

After changing the sign of the momenta (i. e. $[P^i] \leftrightarrow [-P^i]$ and $[P^j] \leftrightarrow [-P^j]$) we get

$$p_i T_{ij} = p_j \int [-dP^i - dP^j]\, \mathcal{P}_G([-P^j]) \cdot P_A\Big([U^j, -P^j] \to [U^i, -P^i]\Big) \mathcal{P}_M\Big([U^j, P^j] \to [U^i, P^i]\Big)$$

The Hamiltonian $\mathcal{H}$ does not depend on the sign of the momenta such that the gaussian distribution and the acceptance probability both remain unchanged after a change of a sign. Since $[dP^i dP^j]$ equals $[-dP^i - dP^j]$, we can rewrite the last equation as

$$p_i T_{ij} = p_j \int [dP^i dP^j]\, \mathcal{P}_G([P^j]) \cdot P_A\Big([U^j, P^j] \to [U^i, P^i]\Big) \cdot \mathcal{P}_M\Big([U^j, P^j] \to [U^i, P^i]\Big) = p_j T_{ji}.$$

□

Lemma 2.4 (Area-preservation)**.** *Let the numerical integration method used in the Hybrid Monte Carlo method be time-reversible. If the numerical integration scheme is also area-preservating (i. e. symplectic), the Markov process converges to the fixed point of the equlibrium distribution of the field configurations $[U]$.*

Proof. We have the Boltzmann- and Gaussian-distributed probabilities

$$p([U]) \sim \exp\Big(-S\Big([U]\Big)\Big) \qquad \text{and} \qquad \mathcal{P}_G([P]) \sim \exp\Big(-E_{kin}\Big([P]\Big)\Big).$$

Both probabilities are normalized such that

$$\int [dU] p([U]) = 1 \qquad \text{and} \qquad \int [dP] \mathcal{P}_G([P]) = 1$$

holds. It follows

$$\begin{aligned} p([U])\mathcal{P}_G([P]) &= \frac{1}{n}\exp\Big(-S\Big([U]\Big)\Big)\exp\Big(-E_{kin}\Big([P]\Big)\Big) \\ &= \frac{1}{n}\exp\Big(-\mathcal{H}\Big([U,P]\Big)\Big) \end{aligned}$$

with a normalization constant

$$n = \int [dU\,dP]\exp\Big(-\mathcal{H}\Big([U,P]\Big)\Big).$$

The normalization factors should be the same for all configurations. That means, the factor

$$n_i = \int [dU^i dP^i]\exp\Big(-\mathcal{H}\Big([U^i,P^i]\Big)\Big)$$

of configuration $[U^j, P^j]$ should be the same as the one of the succeeding configuration $[U^j, P^j]$, i. e.

$$\begin{aligned} n_j &= \int [dU^j dP^j]\exp\Big(-\mathcal{H}\Big([U^j,P^j]\Big)\Big) \\ &= \int [dU^i dP^i]\exp\Big(-\mathcal{H}\Big([U^i,P^i]\Big)\exp\Big(-\Delta\mathcal{H}\Big) \end{aligned}$$

with $\Delta\mathcal{H} = \mathcal{H}[U^j, P^j] - \mathcal{H}[U^i, P^i]$. Using

$$\langle\exp\Big(-\Delta\mathcal{H}\Big)\rangle = \frac{1}{n_i}\int [dU^i dP^i]\exp\Big(-\mathcal{H}\Big([U^i,P^i]\Big)\exp\Big(-\Delta\mathcal{H}\Big)$$

it follows

$$n_j = n_i\langle\exp\Big(-\Delta\mathcal{H}\Big)\rangle.$$

If the numerical integration method is area-preservating, it holds $n_j = n_i$. Hence, the expectation value of $\langle\exp\Big(-\Delta\mathcal{H}\Big)\rangle$ has to be equal to one. □

In case of no area-preservation, there has to be a correction in the acceptance step to ensure this.

Chapter 3

Numerical Integration

In the previous chapters, the effective calculation of the expectation value $\langle A \rangle$ of a gauge field (see equation (2.1)) by a Hybrid Monte Carlo algorithm is described. In doing so, the Hamiltonian

$$\mathcal{H}\Big([U,P]\Big) = E_{kin}\Big([P]\Big) + S_G\Big([U]\Big)$$

(see equation (1.5)) has to be evaluated.
For the numerical integration it is important to note that the kinetic energy E_{kin} is composed of the traceless and hermitian momenta P whereas the Wilson action S_G depends on the link matrices U which are elements of the Lie group $SU(N,\mathbb{C})$.
The sets of links U and momenta P will be calculated in a Molecular Dynamic step by means of solving the equations of motion

$$\dot{U}(t) = iP(t)U(t) \qquad \text{and} \qquad \dot{P}(t) = i\frac{\beta}{N}\Big\{U(t)V(t)\Big\}_{TA}$$

given in the equations (1.7) and (1.8) numerically. For convenience, these equations will be denoted as

$$\dot{U}(t) = f\Big(U(t), iP(t)\Big) \qquad \text{and} \qquad \dot{P}(t) = g\Big([U(t)]\Big). \tag{3.1}$$

Since the sum of staples $V(t)$ (see definition 1.3) consists of the surrounding links of $U(t)$, we consider it as fixed. Thus $\dot{P}(t)$ can be expressed as function g of the whole field $[U(t)]$. The variables β and N are constants.
The numerical integration during the Molecular Dynamic step is the crucial point in the Hybrid Monte Carlo algorithm and will be investigated in detail in this chapter.We start with discussing the properties of the integration method and present known methods for computing differential equations in Lie groups. Afterwards, partitioned Runge-Kutta methods are introduced and brought forward on matrix Lie groups. These methods can be found in [2]. Finally, we place our emphasis on solving the equations of motion (3.1) with partitioned Runge-Kutta methods and check the time-reversibility and convergence order of this scheme.

3.1 Desired Properties of the Integration Scheme

A suitable integration scheme has to fulfill several properties:

- First of all, the numerical solution $\Big(U, P\Big)$ has to consist of a link U situated in the Lie group and an associated momentum P, which is traceless and hermitian.
- Additionally, the scheme has to be symmetric or time-reversible to fulfill the detailed balance condition required in the Hybrid Monte Carlo algorithm.
  Note that the concept of symmetry is widely used in mathematical language whereas the item time-reversibility occurs frequently in physical literature.
- The integration should also be volume-preservating. If this is not the case, a correction in the acceptance step has to be performed. The volume-preservation will not be investigated in this work.
- Furthermore, preferably a high convergence order should be obtained to allow larger step sizes and reduce computing time.

3.1.1 Differential Equations on Lie Groups

Let a matrix Lie group $G \in GL(n)$ be given. ($GL(n)$ is the set of all quadratic and invertible matrices of dimension $n \times n$.) A Lie group is also a differentiable manifold and has a tangent space $T_U G$ in every point $U \in G$. The tangent space $\mathfrak{g} = T_I G$ at the identity I is the appropriate Lie algebra of the Lie group G. Note that the dependence of the Lie algebra element A on the Lie group element U is expressed in the whole section by A_U.

Lemma 3.1 (Differential equations on manifolds)**.** *Let U be an element of the Lie group G and A_U an element of its associated Lie Algebra $\mathfrak{g}$. Then it holds:*

- *$A_U U$ is an element of the tangent space $T_U G := \{A_U U | A_U \in \mathfrak{g}\}$.*
- *$\dot{U} = A_U U$ defines a differential equation on the manifold G.*

Proof. With $A_U \in \mathfrak{g}$ and the definition of the tangent space $T_I G$ there exists a differentiable path $\alpha(t)$ in G with $\alpha(0) = I$ and $\dot{\alpha}(0) = A_U$. For a fixed $U \in G$ the path $\gamma(t) = \alpha(t)U$ satisfies $\gamma(0) = U$ and $\dot{\gamma}(0) = A_U U$.
Thus $A_U U$ is an element of the tangent space $T_U G$ and $\dot{U} = A_U U$ defines a differential equation on the manifold G. □

Theorem 3.2. *Let G be a matrix Lie goup and $\mathfrak{g}$ its Lie algebra. The solution of the differential equation*

$$\dot{U} = A_U U \tag{3.2}$$

satisfies $U(t) \in G$ for all t, if $A_U \in \mathfrak{g}$ for all $U \in G$ and the initial value U_0 is an element of the Lie group G.

It is a consequence of the last theorem that the differential equation $\dot{U}(t) = A_U(t)U(t)$ can be solved by finding a suitable expression $\Omega(t)$ in the Lie algebra $\mathfrak{g}$ and map it into the Lie group G. One way to get the solution of the differential equation (3.2) is constituted by the theorem of Magnus.

Theorem 3.3 (Magnus, 1954)**.** *The solution of* $\dot{U}(t) = A_U(t)U(t)$ *with* $A_U(t) \in \mathfrak{g}$ *and* $U(t)$ *in the appropriate Lie group* G *can be written as* $U(t) = \exp\Big(\Omega(t)\Big)U_0$ *with* $U_0 \in G$ *and* $\Omega(t)$ *definded by the derivative of the inverse exponential map*

$$\dot{\Omega}(t) = d\exp_\Omega^{-1}\Big(A_U(t)\Big) = \sum_{k\geq 0} \frac{B_k}{k!} ad_\Omega^k\Big(A_U(t)\Big) \tag{3.3}$$

with initial value $\Omega(t_0) = 0$*. As long as* $\|\Omega(t)\| < \pi$*, the convergence of the* $d\exp_\Omega^{-1}$ *expansion is assured.*

There are still some unknowns in this theorem that have to be explained.

Definition 3.4 (Bernoulli numbers)**.** *The elements* B_k *in theorem 3.3 are the Bernoulli numbers, defined by*

$$\sum_{k\geq 0} \frac{B_k}{k!} x^k = \frac{x}{\exp(x) - 1}.$$

The first few Bernoulli numbers are $B_0 = 1, B_1 = -\frac{1}{2}, B_2 = \frac{1}{6}, B_3 = 0$*.*

Definition 3.5 (Adjoint operator)**.** *The adjoint operator* $ad_\Omega(A)$ *is a linear operator*

$$ad : \mathfrak{g} \to \mathfrak{g}\,, \quad A \mapsto ad_\Omega(A) = [\Omega, A]$$

for a fixed Ω *and uses matrix commutators.*
The adjoint operator can be used iteratively, such that ad_Ω^k *denotes the* k*-th iterated application of the linear operator* ad_Ω*. By convention,* $ad_\Omega^0(A)$ *is set to* A*.*

Differential equations on the special unitary Lie group

Let us return to the equation $\dot{U}(t) = f\Big(U(t), iP(t)\Big) = iP(t)U(t)$. Due to the fact that $U(t)$ is an element of the Lie group $SU(N, \mathbb{C})$ and $iP(t)$ its associated Lie algebra element, this is a special case of equation (3.2).
Applying lemma 3.1, we know that

$$\dot{U}(t) = iP(t)U(t)$$

is a differential equation on the Lie group $SU(N, \mathbb{C})$. It follows from theorem 3.2 that this equation has a solution $U(t)$ in $SU(N, \mathbb{C})$ if we solve an initial value problem with initial value $U_0 \in SU(N, \mathbb{C})$. This can be performed using the theorem of Magnus above.

3.1.2 Symmetry (= Time-Reversibility)

To ensure the convergence of the Markov process to the equilibrium distribution, the symmetry is a necessary property of the integration method. We investigate the symmetry of a numerical one-step scheme ϕ_h by the term of its adjoint method ϕ_h^*.

Definition 3.6 (Adjoint method)**.** *The adjoint method ϕ_h^* of a method ϕ_h is the inverse map of the original method with reversed time step $-h$, i. e. ϕ_h^* is identified with ϕ_{-h}^{-1}. The index h expresses the dependence of the scheme on the step size h.*

As an example, we formulate the adjoint of the implicit midpoint rule:

1. We start with $y_1 = \phi_h(y_0) = y_0 + h \cdot f\Big(0.5 \cdot (y_0 + y_1)\Big)$.

2. The adjoint method will be obtained by reversing the time step, i. e. exchange $y_1 \leftrightarrow y_0$ and $h \leftrightarrow -h$. Thus we get

$$y_0 = \phi_{-h}(y_1) = y_1 - h \cdot f\Big(0.5 \cdot (y_1 + y_0)\Big)$$

   as intermediate step.

3. Finally, we invert the map $\phi_{-h}(y_1)$. This means, the equation above is solved for y_1. We get the result

$$y_1 = \phi_{-h}^{-1}(y_0) = y_0 + h \cdot f\Big(0.5 \cdot (y_1 + y_0)\Big)$$

   which is the adjoint method $\phi_h^*(y_0)$.

Definition 3.7 (Symmetry)**.** *A numerical one-step method ϕ_h is called symmetric or time-reversible, if it satisfies*

$$\phi_h \circ \phi_{-h} = id \quad \textit{or equivalently} \quad \phi_h = \phi_{-h}^{-1} =: \phi_h^*.$$

For the implicit midpoint rule, it holds $\phi_h = \phi_h^*$, thus this method is symmetric, i. e. time-reversible.

3.1.3 Convergence Order

In general, the convergence order of a method is composed of the consistency order and an additional stability criterion. For one step schemes the stability condition is automatically fulfilled, such that the consistency and the convergence order coincide.

Definition 3.8 (Consistency order)**.** *A method is called consistent, if the local discretization error tends to zero for a step size $h \to 0$:*

$$\|\tau(h)\| \leq \gamma(h) \qquad \textit{with} \quad \lim_{h \to 0} \gamma(h) \to 0.$$

Thereby, the local discretization error $\tau(h)$ is defined as

$$\tau(h) = \frac{\phi(t_0 + h) - \phi_1(h)}{h}.$$

with numerical solution ϕ_1 (after one step). $\phi(t)$ is the exact solution of the initial value problem $\phi' = f(t)$ with initial value $\phi(t_0) = \phi_0$. The method has consistency order p, if $\|\tau(h)\| = \mathcal{O}(h^p)$ holds.

The consistency order will be obtained by expanding the exact and numerical solution of the differential equation in a Taylor series and afterwards calculating the local discretization error $\tau(h)$.
Since we solve the two equations for the links U and the momenta P simultaneously, we have to expand the exact solutions as well as the numerical solutions U_1 and P_1 and consider both local discretization errors

$$\tau_U(h) = \frac{U(t_0 + h) - U_1(h)}{h} \quad \text{and} \quad \tau_P(h) = \frac{P(t_0 + h) - P_1(h)}{h}$$

to get the consistency order of the system (U, P).

Remark 3.9 (Numerical error after one trajectory)**.** *Let the convergence order of a one-step scheme using the step size h be p. This implies a numerical error of order h^{p+1} after one step.*
If a whole trajectory of length $\tau = n \cdot h$ is computed, the integration is performed n times with step size h. After one trajectory, this implies a total error of order

$$n \cdot h^{p+1} = \frac{\tau}{h} \cdot h^{p+1} = h^p.$$

3.2 The Störmer-Verlet (= Leapfrog) Method

We start with the investigation of the properties symmetry and convergence order on the basis of already known one-step methods applied to the problem of solving the equations of motion of (3.1). First of all, these methods are formulated for the general system of differential equations

$$\dot{y} = f(y, z), \quad \dot{z} = g(y, z)$$

with initial values $y(t_0) = y_0$ and $z(t_0) = z_0$. They can be found in [2] as well.

3.2.1 Lie-Euler Method

Definition 3.10 (Explicit Euler Method)**.** *The explicit Euler method reads*

$$\begin{pmatrix} y_1 \\ z_1 \end{pmatrix} = \phi_h \begin{pmatrix} y_0 \\ z_0 \end{pmatrix}, \qquad \begin{aligned} y_1 &= y_0 + h \cdot f(y_0, z_0) \\ z_1 &= z_0 + h \cdot g(y_0, z_0). \end{aligned} \tag{3.4}$$

Lemma 3.11 (Properties of the explicit Euler scheme)**.** *The explicit Euler scheme is not symmetric and has convergence order one because $\|\tau_y\| = \mathcal{O}(h) = \|\tau_z\|$.*

Proof. For symmetry, it has to hold $\phi_h^* = \phi_h$. The explicit Euler scheme is not symmetric, because exchanging $y_0 \leftrightarrow y_1$, $z_0 \leftrightarrow z_1$ and $h \leftrightarrow -h$ yields the adjoint method

$$\phi_h^* \begin{pmatrix} y_0 \\ z_0 \end{pmatrix} = \begin{pmatrix} y_0 + h \cdot f(y_1, z_1) \\ z_0 + h \cdot g(y_1, z_1) \end{pmatrix}$$

and thus $\phi_h^* \neq \phi_h$.
The consistency order of the scheme will be obtained by comparing the Taylor series of the exact and numerical solution of the system (3.4). The local discretization errors read

$$\tau_y(h) = \frac{y_0 + h\dot{y}(t_0) + \frac{h^2}{2}\ddot{y}(t_0) - \Big(y_0 + hf(y_0, z_0)\Big)}{h} = \frac{\frac{h^2}{2}\ddot{y}(t_0)}{h} = \frac{h}{2}\ddot{y}(t_0)$$

$$\text{and} \quad \tau_z(h) = \frac{z_0 + h\dot{z}(t_0) + \frac{h^2}{2}\ddot{z}(t_0) - \Big(y_0 + hg(y_0, z_0)\Big)}{h} = \frac{\frac{h^2}{2}\ddot{z}(t_0)}{h} = \frac{h}{2}\ddot{z}(t_0).$$

such that the convergence order is one. □

This method is not symmetric, and has just convergence order one such that it is not suitable for simulation of gauge theories.

Lie Euler Method

Nevertheless, the explicit Euler method serves as an example to restate a general method as a method for solving differential equations on a Lie group. In this case, it is called Lie-Euler method. For the equations of motion

$$\frac{\partial \mathcal{H}\Big([U,P]\Big)}{\partial U} = -\dot{P} = -g\Big(\Big[U(t)\Big]\Big) \qquad \text{and} \qquad \frac{\partial \mathcal{H}\Big([U,P]\Big)}{\partial P} = \dot{U} = iP(t)U(t)$$

in phase space (U, P) with initial values $(U(t_0), P(t_0)) = (U_0, P_0)$ it holds $U_1 = U_0 + h\dot{U}(t_0) = U_0 + hiP(t_0)U(t_0) = (I + hiP_0)U_0 = \exp(hiP_0)U_0 + \mathcal{O}(h^2)$
Thus, the method can be changed to

$$\begin{pmatrix} U_1 \\ P_1 \end{pmatrix} = \phi_h \begin{pmatrix} U_0 \\ P_0 \end{pmatrix}, \qquad \begin{matrix} U_1 = \exp(hiP_0)U_0 & \in SU(N, \mathbb{C}) \\ iP_1 = iP_0 + h \cdot ig(U_0) & \in \mathfrak{su}(N, \mathbb{C}) \end{matrix} \tag{3.5}$$

Symplectic Lie Euler Method

We can reformulate the explicit Lie-Euler method to the so-called symplectic Lie-Euler method by evaluating the Hamilonian equations of motion at (U_0, P_1) and get

$$\begin{pmatrix} U_1 \\ P_1 \end{pmatrix} = \phi_h \begin{pmatrix} U_0 \\ P_0 \end{pmatrix}, \left\{ \begin{array}{ll} U_1 & = \exp(hiP_1)U_0 \\ P_1 & = P_0 + hg(U_0) \end{array} \right\} \text{or} \left\{ \begin{array}{ll} U_1 & = \exp(hiP_0)U_0 \\ P_1 & = P_0 + hg(U_1) \end{array} \right\} \tag{3.6}$$

which is a symplectic (i. e. a volume-preservating) method of order one. It can be easily seen that this method is not symmetric.

3.2.2 Störmer-Verlet Method

The combination of the two symplectic Euler methods from equation (3.6) yields the Störmer-Verlet scheme which is also mentioned as Leapfrog method.

Definition 3.12 (Störmer-Verlet Method)**.** *The Störmer-Verlet scheme reads*

$$P_{\frac{1}{2}} = P_0 + \frac{h}{2} g(U_0), \qquad U_1 = \exp\Big(hiP_{\frac{1}{2}}\Big)U_0, \qquad P_1 = P_{\frac{1}{2}} + \frac{h}{2} g(U_1). \tag{3.7}$$

It is known as symmetric(i.e time-reversible), symplectic (i. e. volume-preserving) and has convergence order two. Due to its properties, this method is used in many applications, for example in simulations of lattice gauge fields. In these simulations, a whole trajectory is computed at once. During the calculation of one trajectory with length $\tau = n \cdot h$, the integration (3.7) is carried out n times with step size h:

$$\begin{aligned}
P_{\frac{1}{2}} &= P_0 + \frac{h}{2} g(U_0), & U_1 &= \exp\Big(hiP_{\frac{1}{2}}\Big)U_0, & P_1 &= P_{\frac{1}{2}} + \frac{h}{2} g(U_1),\\
P_{\frac{3}{2}} &= P_1 + \frac{h}{2} g(U_1), & U_2 &= \exp\Big(hiP_{\frac{3}{2}}\Big)U_1, & P_2 &= P_{\frac{3}{2}} + \frac{h}{2} g(U_2),\\
&\vdots & &\vdots & &\vdots\\
P_{n-\frac{1}{2}} &= P_{n-1} + \frac{h}{2} g(U_{n-1}), & U_n &= \exp\Big(hiP_{n-\frac{1}{2}}\Big)U_{n-1}, & P_n &= P_{n-\frac{1}{2}} + \frac{h}{2} g(U_n).
\end{aligned}$$

In doing so, we start with an initial configuration $(U(t_0), P(t_0))$ at time t_0. Then, n trajectories of step size h are computed such that the final configuration $(U(t_0 + \tau), P(t_0 + \tau))$ is reached.
The Störmer-Verlet method can be summarized in the following algorithm:

Algorithm 3.1 (Leapfrog method)**.**
Let the initial values $(U(t_0), P(t_0))$ at time t_0 be given. For simplicity, t_0 is set to zero. Thereby, the integration can be divided in three parts:

1. *Start with carrying out one explicit Euler half-step for the momentum*
   - $P_{\frac{h}{2}} = P_0 + \frac{h}{2} \cdot g\Big(U_0\Big)$.
2. *For $k = 1, \ldots, n$ and $l = 1, \ldots, n-1$ calculate alternately*
   - $U_{k \cdot h} = \exp\Big(P_{(k-\frac{1}{2}) \cdot h}\Big) \cdot U_{(k-1) \cdot h}$
   - $P_{(l+\frac{1}{2}) \cdot h} = P_{(l-\frac{1}{2}) \cdot h} + h \cdot g\Big(U_{l \cdot h}\Big)$

   such that the iteration ends with U_τ) and $P_{\tau - \frac{h}{2}}$. In this part, the integration step size will be h.
3. *The last half-step reads*
   - $P_\tau = P_{\tau - \frac{h}{2}} + \frac{h}{2} \cdot g\Big(U_\tau\Big)$.

Note that the integration uses the step size h in the main part but only step size $\frac{h}{2}$ in the first and last halfstep to evaluate each link U and each momentum P at two different points in time.
The convergence order of the Störmer-Verlet method is 2. This implies a numerical error of order h^3 after one step. Since the integration is performed $n = \frac{\tau}{h}$ times with an error of order h^3, the numerical error after one trajectory is of order h^2 (see remark 3.9).

3.3 Partitioned Runge-Kutta Methods for Lie Groups

3.3.1 Partitioned Runge-Kutta Methods in General

Our aim is to reformulate the Störmer-Verlet method as implicit partitioned Runge-Kutta method. As the investigated Störmer-Verlet method it also has to be symmetric and should be volume-preserving. Additionally, we wish to get a method of order greater than two to enlarge the step sizes, thus we have to investigate the attainable order of the partitioned Runge-Kutta method.

Definition 3.13 (Partitioned Runge-Kutta Method)**.** *A coupled system*

$$\dot{y} = f(y,z), \qquad \dot{z} = g(y,z)$$

of ordinary differential equations may be solved by a partitioned Runge-Kutta method of the form

$$\begin{aligned} y_1 &= y_0 + h\sum_{j=1}^{s} b_j K_j, & z_1 &= z_0 + h\sum_{j=1}^{s} \hat{b}_j L_j, \\ K_j &= f\Big(y_0 + h\sum_{k=1}^{s} a_{j,k}K_k,\ z_0 + h\sum_{k=1}^{s} \hat{a}_{j,k}L_k\Big), \\ L_j &= g\Big(y_0 + h\sum_{k=1}^{s} a_{j,k}K_k,\ z_0 + h\sum_{k=1}^{s} \hat{a}_{j,k}L_k\Big) \end{aligned}$$

with initial values y_0 and z_0. The coefficients $b_j, a_{jk}, \hat{b}_j, \hat{a}_{jk}$ and increments K_j and L_j belong to the s stages of y_1 and z_1.

Definition 3.14 (Butcher tableau)**.** *The coefficients $b_j, a_{j,k}, \hat{b}_j, \hat{a}_{j,k}$ of the Runge-Kutta scheme can be denoted in a Butcher tableau. For example, the Butcher tableau for the coefficients $a_{j,k}$ and b_j $(j,k = 1,\dots,s)$ looks like*

$$\begin{array}{c|cccc} a_1 & a_{1,1} & a_{1,2} & \dots & a_{1,s} \\ a_2 & a_{2,1} & a_{2,2} & \dots & a_{2,s} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_s & a_{s,1} & a_{s,2} & \dots & a_{s,s} \\ \hline & b_1 & b_2 & \dots & b_s. \end{array}$$

The entries of the left column a_j express the row sums of the coefficients $a_{j,k}$, i. e. $a_j = \sum_{k=1}^{s} a_{j,k}$.

Using these coefficients, we can distinguish between explicit and implicit Runge-Kutta methods: We have an

- explicit Runge-Kutta method if $a_{j,k} = 0 \; \forall k \geq j$
- and an implicit Runge-Kutta method if $a_{j,k} \neq 0$ for one or more $k \geq j$.

Theorem 3.15 (Symmetric Runge-Kutta scheme)**.** *The adjoint method of an s-stage Runge-Kutta method defined by definition 3.13 is again an s-stage Runge-Kutta method. Its coefficients are given by*

$$\begin{aligned} a^*_{j,k} &= b_{s+1-k} - a_{s+1-j,s+1-k}\,, & b^*_j &= b_{s+1-k}, \\ \hat{a}^*_{j,k} &= \hat{b}_{s+1-j} - \hat{a}_{s+1-j,s+1-k}\,, & \hat{b}^*_j &= \hat{b}_{s+1-k}. \end{aligned}$$

The Runge-Kutta method of definition 3.13 is symmetric if

$$a_{s+1-j,s+1-k} + a_{j,k} = b_k \quad \textit{and} \quad \hat{a}_{s+1-j,s+1-k} + \hat{a}_{j,k} = \hat{b}_k \quad \textit{for all } j,k. \tag{3.8}$$

Explicit Runge-Kutta schemes can not fulfill equations (3.8) with $j = k$ and thus can not be symmetric. So they are not suitable to be adopted in the simulations of gauge fields and hence we have to investigate implicit partitioned Runge-Kutta methods.

Runge-Kutta Method for Lie Groups

Transferred to our problem of solving the differential equations (3.1)

$$\dot{U}(t) = f\Big(U(t), iP(t)\Big) \qquad \text{and} \qquad i\dot{P}(t) = ig\Big([U(t)]\Big).$$

with initial values $U(t_0) \in SU(N,\mathbb{C})$ and $iP(t_0) \in \mathfrak{su}(N,\mathbb{C})$, the partitioned Runge-Kutta method reads as follows:

$$\text{Calculate} \quad U_1 = U_0 + h\sum_{j=1}^{s} b_j K_j \quad \text{and} \quad iP_1 = iP_0 + h\sum_{j=1}^{s} \hat{b}_j L_j \quad \text{with increments}$$

$$K_j = f\Big(U_0 + h\sum_{k=1}^{s} a_{j,k}K_k,\; P_0 + h\sum_{k=1}^{s} \hat{a}_{j,k}L_k\Big) \quad \text{and} \quad L_j = ig\Big(U_0 + h\sum_{k=1}^{s} a_{j,k}K_k\Big)$$

for the stages $j = 1, \ldots, s$ with the same coefficients $b_j, a_{j,k}$ and $\hat{b}_j, \hat{a}_{j,k}$ used above.

The solution iP_1 will be part of the Lie algebra $\mathfrak{su}(N,\mathbb{C})$. Nevertheless, there is a problem concerning the solution U_1. The initial value U_0 is an element of $SU(N,\mathbb{C})$, but the increments K_j are elements of the assosiated Lie algebra $\mathfrak{su}(N,\mathbb{C})$. Thus, U_1 is a sum of one Lie group and Lie algebra elements and therefore not in the Lie group.
However, we know that the differential equation $\dot{U}(t) = iP(t)U(t)$ can be solved by finding a suitable expression $\Omega(t)$ in the Lie algebra $\mathfrak{su}(N,\mathbb{C})$ and map it into the Lie group. This needs some general considerations.

3.3.2 Munthe-Kaas Method

Let G be a Lie group and $\mathfrak{g}$ the appropriate Lie algebra. Consider the differential equation

$$\dot{U} = A(t)U(t) \quad \text{with} \quad A(t) \in \mathfrak{g} \text{ and } U(t) \in G,$$

that should be solved by a Runge-Kutta method.
Following the idea of Munthe-Kaas that uses the exponential map as transformation from the Lie algebra to the Lie group, the solution will be identified with

$$U(t) = \exp\Big(\Omega(t)\Big)U_0. \tag{3.9}$$

The initial value $\Omega(t_0)$ has to be zero in order that the initial values of $U(t)$ and $\exp\Big(\Omega(t)\Big)U_0$ coincide.
It is evident that the unknown function has changed from $U(t) \in G$ to $\Omega(t) \in \mathfrak{g}$. So we face the problem of getting the function $\Omega(t) \in \mathfrak{g}$ that can fortunately be obtained by the following approach using the derivative of the inverse exponential map $d\exp_\Omega^{-1}$ (see equation (3.3)).

Algorithm 3.2 (Munthe-Kaas, 1999)**.** *We consider the differential equation*

$$\dot{U} = A(t)U(t)$$

with $A(t) \in \mathfrak{g}$ and $U(t)$ in the appropriate Lie group G. This problem can be solved by the following steps:

- *Take a suitable truncation*

$$\dot{\Omega} = \sum_{k=0}^{q} \frac{B_k}{k!} ad_\Omega^k(A) = A - \frac{1}{2}[\Omega, A] + \frac{1}{6}\Big[\Omega, [\Omega, A]\Big] + \dots$$

  of the differential equation $\dot{\Omega} = d\exp_\Omega^{-1}\Big(A(t)\Big)$ given in equation (3.3).

- *Use a Runge-Kutta scheme for the computation of the numerical solution $\Omega_1 \approx \Omega(t_0 + h)$.*

- *Calculate the numerical solution of $\dot{U} = A(t)U(t)$ as $U_1 = \exp\Big(\Omega_1\Big)U_0$.*

The suitable truncation can be found as follows:

Theorem 3.16 (Suitable truncation of $\dot{\Omega}$)**.** *If the Runge-Kutta method is of order p and the truncation index q of*

$$\dot{\Omega} = \sum_{k=0}^{q} \frac{B_k}{k!} ad_\Omega^k(A)$$

satisfies $q \geq p - 2$, then the method of the Munthe-Kaas algorithm is of order p.

Runge-Kutta Methods Applied to the Equations of Motion

Our problem

$$\dot{U}(t) = f\Big(U(t), iP(t)\Big) \qquad \text{and} \qquad \dot{P}(t) = g\Big([U(t)]\Big) \tag{3.1}$$

with initial values $U(t_0) \in SU(N,\mathbb{C})$ and $iP(t_0) \in \mathfrak{su}(N,\mathbb{C})$ can be solved with a partitioned Runge-Kutta method by means of the Munthe-Kaas method.

Instead of finding the solution of $U(t)$ in the Lie group and the solution of $iP(t)$ in the Lie algebra, we use $U(t) = \exp\Big(\Omega(t)\Big)U_0$ with initial value $\Omega(t_0) = 0$. Then we replace the differential equations of (3.1) by

$$\begin{aligned} \dot{\Omega}(t) &= f\Big(\Omega(t), iP(t)\Big) = d\exp_\Omega^{-1}(iP) \\ \text{and} \quad i\dot{P}(t) &= ig\Big(\Big[\exp\Big(\Omega(t)\Big)\cdot U(t_0)\Big]\Big). \end{aligned} \tag{3.10}$$

This is advantageous because we solve both differential equations in the Lie algebra $\mathfrak{su}(N,\mathbb{C})$ and get the solution $\Big(U(t), P(t)\Big) = \Big(\exp\Big(\Omega(t)\Big)U(t_0), P(t)\Big)$ in the phase space via the mapping (3.9).

Remark 3.17 (Example of suitable truncations)**.** *It is sufficient to expand the series of the inverse exponential map $\dot{\Omega}$ up to order $p-2$ to derive method order p (see theorem 3.16).The truncation of $\dot{\Omega}$ reads*

$$\dot{\Omega} = d\exp^{-1}(iP) = \sum_{k=0}^{p-2} \frac{B_k}{k!} ad_\Omega^k(iP).$$

We yield Runge-Kutta methods of order $p = 2$ and $p = 3$ with

$$\begin{aligned} p = 2: &\qquad \dot{\Omega} = f(\Omega, iP) = B_0 iP, \\ p = 3: &\qquad \dot{\Omega} = f(\Omega, iP) = B_0 iP + B_1[\Omega, iP]. \end{aligned}$$

using the Bernoulli numbers $B_0 = 1$ and $B_1 = -\frac{1}{2}$ (see definition 3.4) .

Definition 3.18 (Runge-Kutta method applied to the equations of motion)**.** *The system of equations* (3.10) *with initial values $(\Omega(t_0), P(t_0)) = (\Omega_0, P_0)$ may be solved with a partitioned Runge-Kutta method. The scheme can be denoted as*

$$\Omega_1 = \Omega_0 + h\sum_{j=1}^{s} b_j K_j \quad \text{and} \quad iP_1 = iP_0 + h\sum_{j=1}^{s} \hat{b}_j L_j.$$

The increments K_j and L_j read

$$\begin{aligned} K_j &= f\Big(\Omega_0 + h\sum_{k=1}^{s} a_{j,k}K_k,\ iP_0 + h\sum_{k=1}^{s} \hat{a}_{j,k}L_k\Big) \\ \text{and} \quad L_j &= ig\Big(\exp\Big(\Omega_0 + h\sum_{k=1}^{s} a_{j,k}K_k\Big)U_0\Big) \end{aligned}$$

for $j = 1, \ldots, s$ with coefficients $b_j, a_{j,k}$ and $\hat{b}_j, \hat{a}_{j,k}$.

Due to the fact that the initial value of $\Omega(t)$ vanishes, the notation of this Runge-Kutta method can be simplyfied as follows:

Remark 3.19 (Simplified partitioned Runge-Kutta scheme)**.**

$$\Omega_1 = h\sum_{j=1}^{s} b_j K_j \quad \textit{and} \quad iP_1 = iP_0 + h\sum_{j=1}^{s} \hat{b}_j L_j.$$

with increments

$$K_j = f\Big(\underbrace{h\sum_{k=1}^{s} a_{j,k}K_k}_{=:Y_j},\ \underbrace{iP_0 + h\sum_{k=1}^{s} \hat{a}_{j,k}L_k}_{=:Z_j}\Big) = f\Big(Y_j, Z_j\Big)$$

$$\textit{and} \quad L_j = ig\Big(\exp\Big(\underbrace{h\sum_{k=1}^{s} a_{j,k}K_k}_{=:Y_j}\Big)U_0\Big) = ig\Big(\exp\Big(Y_j\Big)U_0\Big)$$

and coefficients $b_j, a_{j,k}$ *and* $\hat{b}_j, \hat{a}_{j,k}$ *for* $j,k = 1,\dots,s$.

In doing so we reach the numerical solution $\Big(U_1, P_1\Big) = \Big(\exp(\Omega_1)\cdot U_0, P_1\Big)$ that will be investigated in detail in the following section.

Remark 3.20 (Kind of the variables)**.** *Note that* P_0*, and* P_1 *are traceless and hermitian matrices. The other variables* Ω_1*,* K_j*,* L_j*,* Y_j*, and* Z_j *are elements of the Lie algebra* $\mathfrak{su}(N,\mathbb{C})$*. We solve both equations in the Lie algebra.*

3.3.3 Conditions of Symmetry (= Time-Reversibility)

The Runge-Kutta method is symmetric, if the method coincides with its adjoint method. Unfortunately, theorem 3.15 can not be transferred to the Runge-Kutta method on Lie groups, except for the special case of convergence order $p = 2$. Even in this case, it has to be combined with a further condition concerning the exponential function. Thus, we have to derive conditions for the symmetry manually.
Given is the system

$$\begin{aligned} U_1 = \exp\Big(\Omega_1\Big)U_0\,, \qquad \Omega_1 = h\sum_{j=1}^{s} b_j K_j\,, \qquad iP_1 = iP_0 + h\sum_{j=1}^{s} \hat{b}_j L_j \\ \text{with} \qquad K_j = f(Y_j, Z_j)\,, \qquad L_j = ig\Big(\Big[\exp(Y_j)U_0\Big]\Big)\,, \\ Y_j = h\sum_{k=1}^{s} a_{j,k}K_k\,, \qquad Z_j = iP_0 + h\sum_{k=1}^{s} \hat{a}_{j,k}L_k. \end{aligned} \tag{3.11}$$

Special Case: Convergence Order $p = 2$

For the special case of convergence order 2, the function $f(Y_j, Z_j)$ just depends on its second element. As explained in the following paragraph in more detail, the adjoint of the system will be attained as follows:

- Exchange $U_0 \leftrightarrow U_1,\ P_0 \leftrightarrow P_1,\ h \leftrightarrow -h$ and $\Omega_1 \leftrightarrow -\Omega_1$. Thereby, the indices of the increments K_j, L_j, Y_j and Z_j are changed to $s+1-j$.

- Rearrange the formulas due to the shape of the original method.

- Substitute the indices $s+1-j$, $s+1-k$ through j, k.

Hence, the adjoint of the method is denoted with

$$U_1 = \exp\Big(\Omega_1\Big)U_0\,, \qquad \Omega_1 = h\sum_{j=1}^{s} b_{s+1-j}K_j\,, \qquad iP_1 = iP_0 + h\sum_{j=1}^{s}\hat{b}_{s+1-j}L_j$$

$$\text{with} \quad K_j = f(Z_j) \quad \text{depending on} \quad Z_j = iP_0 + h\sum_{k=1}^{s}(\hat{b}_{s+1-k} - \hat{a}_{j,k})L_k$$

$$\text{and} \quad L_j = ig\Big(\Big[\exp(Y_j)\exp(\Omega_1)U_0\Big]\Big) \quad \text{depending on} \quad Y_j = -h\sum_{k=1}^{s} a_{s+1-j,s+1-k}K_k.$$

Thus, we get the symmetry conditions

$$\begin{aligned}
\Omega_1 :&\quad b_j^* = b_{s+1-j}\,,\\
P_1 :&\quad \hat{b}_j^* = \hat{b}_{s+1-j}\,,\\
K_j :&\quad \hat{a}_{j,k}^* = \hat{b}_{s+1-k} - \hat{a}_{s+1-j,s+1-k}\,,\\
L_j :&\quad a_{j,k}^* = b_{s+1-k} - a_{s+1-j,s+1-k}\,, \quad \text{if } Y_j \text{ and } \Omega_1 \text{ commutate.}
\end{aligned} \tag{3.12}$$

for $j = 1, \ldots, s$. The symmetry conditions coincide with the conditions given in theorem 3.15. Since exponential functions of matrices can just be subsumed if the matrices commmutate, this would imply an additional condition for the symmetry concerning the increments L_j. For example, the colums of the coefficients $a_{j,k}$ denoted in a Butcher tableau can be multiples of the coefficients b_k. For this special case, there are coefficients, given in paragraph 3.4.1.

General Convergence Order

We run into problems concerning the symmetry of the increments L_j for $j = 1, \ldots, s$. These problems are

- $K_j = f(Y_j, Z_j)$ and $L_j = ig\Big(\exp(Y_j)U_0\Big)$ both depend on Y_j. Thus the symmetry conditions for the coefficients $a_{j,k}$ have to coincide for both increments K_j and L_j for $j, k = 1, \ldots, s$.

- As we have seen, a second exponential function $\exp(\Omega_1)$ occurs inside the adjoint of L_j. Since exponential functions of matrices can just be subsumed if the matrices commmutate, this would imply an additional condition for the symmetry, which should preferably be avoided.

Due to these problems, the function g will be modified:

- First of all, we exchange the coefficients $a_{j,k}$ with new coefficients $c_{j,k}$, $j,k = 1,\dots,s$. We get

$$L_j = ig\Big(\Big[\exp(W_j)U_0\Big]\Big) \quad \text{with} \quad W_j = h\sum_{k=1}^{s} c_{j,k}K_k.$$

  Thus K_j and L_j do not depend on the same coefficients and we get 2 seperate symmetry conditions.

- The problem concerning the exponential function will be resolved as follows: We introduce a function $\exp(\frac{1}{2}\Omega_1)$ inside g, such that

$$L_j = ig\Big(\Big[\exp(W_j)\exp(\frac{1}{2}\Omega_1)U_0\Big]\Big)$$

  holds. Due to the shape of $U_1 = \exp(\Omega_1)U_0$, both exponential functions occur as well in the adjoint of L_j and therefore do not have to be resumed.

Because of the aforementioned changes, the whole system of the Runge-Kutta method changes to a symmetric one:

Lemma 3.21 (Symmetric Runge-Kutta method with a general convergence order)**.** *The symmetric Runge-Kutta method for a general convergence order reads*

$$\begin{aligned} U_1 &= \exp\Big(\Omega_1\Big)U_0\,, \qquad \Omega_1 = h\sum_{j=1}^{s} b_jK_j\,, \qquad iP_1 = iP_0 + h\sum_{j=1}^{s}\hat{b}_jL_j\,,\\ K_j &= f(Y_j,Z_j)\,, \qquad L_j = ig\Big(\Big[\exp(W_j)\exp(\frac{1}{2}\Omega_1)U_0\Big]\Big)\\ Y_j &= h\sum_{k=1}^{s} a_{j,k}K_k\,, \quad W_j = h\sum_{k=1}^{s} c_{j,k}K_k\,, \quad Z_j = iP_0 + h\sum_{k=1}^{s}\hat{a}_{j,k}L_k. \end{aligned} \tag{3.13}$$

Remark 3.22. *The adjoined method has the same shape as the system given in* (3.13) *with new coefficients denoted with a star. It looks like*

$$\begin{aligned} U_1 &= \exp(\Omega_1)U_0\,, \qquad \Omega_1 = h\sum_{j=1}^{s} b_j^*K_j\,, \qquad iP_1 = iP_0 + h\sum_{j=1}^{s}\hat{b}_j^*L_j\,,\\ K_j &= f(Y_j,Z_j)\,, \qquad L_j = ig\big(\big[\exp(W_j)\exp(\frac{1}{2}\Omega_1)U_0\big]\big)\,,\\ Y_j &= h\sum_{k=1}^{s} a_{j,k}^*K_k\,, \quad W_j = h\sum_{k=1}^{s} c_{j,k}^*K_k\,, \quad Z_j = iP_0 + h\sum_{k=1}^{s}\hat{a}_{j,k}^*L_k. \end{aligned} \tag{3.14}$$

Lemma 3.23 (Conditions for a symmetric Runge-Kutta method with a general convergence order)**.** *The method described in lemma 3.21 is symmetric if the conditions*

$$\Omega_1: \qquad b_j^* = b_{s+1-j} \tag{3.15}$$

$$P_1: \qquad \hat{b}_j^* = \hat{b}_{s+1-j} \tag{3.16}$$

$$Y_j: \qquad a_{j,k}^* = -a_{s+1-j,s+1-k} \tag{3.17}$$

$$Z_j: \qquad \hat{a}_{j,k}^* = \hat{b}_{s+1-k} - \hat{a}_{s+1-j,s+1-k} \tag{3.18}$$

$$W_j: \qquad c_{j,k}^* = -c_{s+1-j,s+1-k} \tag{3.19}$$

hold for $j,k = 1,\dots,s$.

They are obtained by exchanging

$$U_0 \leftrightarrow U_1,\ P_0 \leftrightarrow P_1,\ h \leftrightarrow -h \ \text{ and } \ \Omega_1 \leftrightarrow -\Omega_1.$$

Since we reversed the time-step, we also have to replace the increments K_j and L_j by K_{s+1-j} and L_{s+1-j} for $j = 1, \dots, s$. So we yield the new system

$$U_0 = \exp\Big(-\Omega_1\Big)U_1\,, \qquad -\Omega_1 = -h\sum_{j=1}^{s} b_j K_{s+1-j}\,, \qquad iP_0 = iP_1 - h\sum_{j=1}^{s} \hat{b}_j L_{s+1-j}\,,$$

$$K_{s+1-j} = f(Y_{s+1-j}, Z_{s+1-j})\,, \qquad L_{s+1-j} = ig\Big(\Big[\exp(W_{s+1-j})\exp(-\frac{1}{2}\Omega_1)U_1\Big]\Big)\,,$$

$$Y_{s+1-j} = -h\sum_{k=1}^{s} a_{j,k}\,K_{s+1-k}\,, \quad W_{s+1-j} = -h\sum_{k=1}^{s} c_{j,k}\,K_{s+1-k}\,,$$

$$Z_{s+1-j} = iP_1 - h\sum_{k=1}^{s} \hat{a}_{j,k} L_{s+1-k}$$

as intermediate step. We can transform the equations of the first line and insert them into L_j and Z_j. If we substitute the indices j and k by $s+1-j$ and $s+1-k$, we will obtain the adjoint method of the system (3.13):

$$U_1 = \exp\Big(\Omega_1\Big)U_0\,, \qquad \Omega_1 = h\sum_{j=1}^{s} b_{s+1-j}K_j\,, \qquad iP_1 = iP_0 + h\sum_{j=1}^{s} \hat{b}_{s+1-j} L_j\,,$$

$$K_j = f(Y_j, Z_j)\,, \qquad L_j = ig\Big(\Big[\exp(W_j)\exp(\frac{1}{2}\Omega_1)U_0\Big]\Big)$$

$$Y_j = -h\sum_{k=1}^{s} a_{s+1-j,s+1-k}\,K_k\,, \quad W_j = -h\sum_{k=1}^{s} c_{s+1-j,s+1-k}\,K_k\,,$$

$$Z_j = iP_0 + h\sum_{k=1}^{s} (\hat{b}_{s+1-k} - \hat{a}_{s+1-j,s+1-k})\,L_k.$$

A comparison of these coefficients and the ones of the adjoined method (see (3.14)) leads to the symmetry conditions described in lemma (3.23) Note that the conditions (3.17) and (3.19) differ from the symmetry condition for general partitioned Runge-Kutta methods given in theorem 3.15.

The question of the existence of a partitioned Runge-Kutta method for Lie groups that is simultaneously symmetric and has a convergence order larger than 2 arises. We will see that it is possible to create a partitioned Runge-Kutta method of convergence order three and combine it with the symmetry conditions above. Using this scheme, we can solve the equations of motion (3.1)

$$\dot{U}(t) = iP(t)U(t) \quad \text{and} \quad i\dot{P}(t) = ig\Big(\Big[U(t)\Big]\Big).$$

Remark 3.24 (Even convergence orders for symmetric methods)**.** *The property symmetry implies an even convergence order, provided that the model error is small enough.*

3.3.4 Derivation of the Order Conditions

The order conditions for the partitioned Runge-Kutta method of order p are derived via Taylor expansions of the exact solution $(U(t), P(t))$ and the numerical solution $(U_1(t), P_1(t))$ up to order p in the neighbourhood of $t_0 = 0$. The comparison of the coefficients leads to the needed order conditions. To get order conditions of order p, the coefficients of both solutions have to coincide up to order p. Since the original function $U(t)$ has been replaced by $\exp(\Omega(t)U_0)$, it is sufficient to compare the coefficients of the Taylor expansions of the new unknown functions $\Omega(t)$ and $\Omega_1(t)$ instead of $U(t)$ and $U_1(t)$.

Preparation of the Taylor Expansions

Remark 3.25 (Taylor expansions of exact and numerical solution)**.** *The Taylor expansions of* $\Omega(t)$*,* $P(t)$*,* $\Omega_1(t)$ *and* $P_1(t)$ *read*

$$\Omega(t_0+h) = \sum_{k=0}^{\infty} \frac{h^k}{k!}\Omega^{(k)}(t_0), \qquad P(t_0+h) = \sum_{k=0}^{\infty} \frac{h^k}{k!}P^{(k)}(t_0), \tag{3.20}$$

$$\Omega_1(t_0+h) = \sum_{k=0}^{\infty} \frac{h^k}{k!}\Omega_1^{(k)}(t_0), \qquad P_1(t_0+h) = \sum_{k=0}^{\infty} \frac{h^k}{k!}P_1^{(k)}(t_0). \tag{3.21}$$

We develop the Taylor expansions around $t_0 = 0$. Thus using the Leibniz rule, the m-th derivatives of Ω_1 and P_1 evaluated at the point 0 read

$$\begin{aligned} \Omega_1^{(m)}(0) &= m\sum_{j=1}^{s} b_j \Omega^{(m)}(0) \\ \text{and} \qquad P_1^{(m)}(0) &= m\sum_{j=1}^{s} \hat{b}_j P^{(m)}(0). \end{aligned} \tag{3.22}$$

These derivatives depend on the unknown function $\Omega(t)$ that is the solution of the differential equation $\dot{\Omega}(t) = dexp^{-1}_{\Omega(t)}$. Due to the fact that the truncation of $\dot{\Omega}(t)$ has to be adopted to the desired convergence order (see remark 3.17), the computation of the Taylor series is not performed for a general case here. The calculation takes place in the subsequent paragraphs concerning the examples of convergence orders 2 and 3.

3.4 Munthe-Kaas Method for Convergence Order 2 and 3

We show that it is possible to create a symmetric partitioned Runge-Kutta method with convergence order $p = 2$ and $p = 3$ to solve the equations of motion (3.1)

$$\dot{U}(t) = iP(t)U(t) \quad \text{and} \quad i\dot{P}(t) = ig\Big(\Big[U(t)\Big]\Big)$$

with initial values $\Big(U_0, iP_0\Big)$.

The Munthe-Kaas method

The solution of the equation of motion concerning the links $U(t)$ is situated in the Lie group $SU(N,\mathbb{C})$. As mentioned in the last paragraph, it can not be directly obtained via a Runge-Kutta method. This means, we have to rewrite the function $U(t)$ as

$$U(t) = \exp\Big(\Omega(t)\Big)U_0$$

such that the unknown function changes to $\Omega(t)$. $\Omega(t)$ is the solution of the differential equation

$$\dot{\Omega}(t) = f\Big(\Omega(t), iP(t)\Big) = d\exp_{\Omega}^{-1}(iP(t)) \qquad \text{(see (3.3))}$$

situated in the Lie algebra $\mathfrak{su}(N,\mathbb{C})$. Hence, the solution $\Omega(t)$ of $\dot{\Omega}(t)$ can be attained via a Runge-Kutta method in the Lie algebra $\mathfrak{su}(N,\mathbb{C})$.
In this manner, the differential equations (3.1) being situated both in a Lie group and in a Lie algebra are replaced with the two differential equations in a Lie algebra. So we can solve

$$\begin{aligned} \dot{\Omega}(t) &= f\Big(\Omega(t), iP(t)\Big) = d\exp_{\Omega}^{-1}(iP) \\ \text{and} \quad i\dot{P}(t) &= ig\Big(\Big[\exp\Big(\Omega(t)\Big)\cdot U(t_0)\Big]\Big). \end{aligned} \qquad \text{(see (3.10))}$$

with the algorithm of Munthe-Kaas (see algorithm 3.2). Thereby, the initial values of (3.10) read $\Big(\Omega_0, iP_0\Big) = (0, iP_0)$. The function $\dot{\Omega}(t)$ has to be suitably truncated, according to the desired convergence order p of the Runge-Kutta method (as described in remark 3.17):

$$\dot{\Omega}(t) = f\Big(\Omega(t), iP(t)\Big) = d\exp_{\Omega(t)}^{-1}(iP) = \sum_{k=0}^{p-2} \frac{B_k}{k!} ad_{\Omega}^k(iP).$$

Our aim is to attain methods of convergence order 2 and 3. Thus, we can use the suitable truncation of $d\exp_{\Omega}^{-1}(iP)$, i. e.

$$\dot{\Omega}(t) = f(\Omega(t), iP(t)) = iP(t) \tag{3.23}$$

$$\dot{\Omega}(t) = f(\Omega(t), iP(t)) = iP(t) - \frac{1}{2}[\Omega(t), iP(t)], \tag{3.24}$$

as mentioned in remark 3.17. Note that hese truncations imply model errors of order $p+1$, i. e. the model error of (3.23) is of order 3 and and that of (3.24) of order 4.

3.4.1 Symmetric Partitioned Runge-Kutta Method of Convergence Order 2

The symmetric partitioned Runge-Kutta method of convergence order $p = 2$ is a special case because it can be seen as reformulated Leapfrog method. We start with the differential equations

$$\begin{aligned} \dot{\Omega}(t) &= f\Big(\Omega(t), iP(t)\Big) = d\exp_{\Omega}^{-1}(iP) \\ \text{and} \quad i\dot{P}(t) &= ig\Big(\Big[\exp\Big(\Omega(t)\Big)\cdot U(t_0)\Big]\Big). \end{aligned} \qquad \text{(see (3.10))}$$

Using the truncation

$$\dot{\Omega}(t) = f(\Omega(t), iP(t)) = iP(t)$$

given in equation (3.23), we get the differential equations

$$\begin{aligned} \dot{\Omega}(t) &= f\Big(\Omega(t), iP(t)\Big) = iP(t) \\ \text{and} \quad i\dot{P}(t) &= ig\Big(\Big[\exp\Big(\Omega(t)\Big) \cdot U(t_0)\Big]\Big). \end{aligned} \tag{3.25}$$

The solution $\Big(\Omega_1, P_1\Big)$ of these differential equations $\Big(\Omega_1, P_1\Big)$ can be calculated with the Runge-Kutta method of $s = 2$ stages as follows (see remark 3.19): Since the function $f\Big(\Omega(t), iP(t)\Big) = iP(t)$ only depends on its second variable, the symmetric Runge-Kutta method reads

Lemma 3.26 (Partitioned Runge-Kutta Method of Convergence Order 2)**.**

Compute

$$\Omega_1 = h\sum_{j=1}^{s=2} b_j K_j \quad \textit{and} \quad iP_1 = iP_0 + h\sum_{j=1}^{s=2} \hat{b}_j L_j \tag{3.26}$$

with increments

$$K_j = f\Big(Y_j, Z_j\Big) = Z_j \quad \textit{and} \quad L_j = ig\Big(\exp(Y_j)U_0\Big)$$

depending on the functions

$$Y_j = h\sum_{k=1}^{s=2} a_{j,k} K_k \quad \textit{and} \quad Z_j = iP_0 + h\sum_{k=1}^{s=2} \hat{a}_{j,k} L_k$$

for the stages $j = 1, 2$. The variables $a_{j,k}, b_j, \hat{a}_{j,k}$ and $\hat{b}_j$ with $j, k = 1, 2$ denote the coefficients of the method.

We reach the numerical solution (U_1, P_1) of the original problem (3.1) via $\Big(U_1, P_1\Big) = \Big(\exp(\Omega_1) \cdot U_0, P_1\Big)$.

Order conditions

As described in the last section, we have to compare the Taylor series of the exact and numerical solution of (3.25) to determine the convergence order. We know from remark 3.25 that these Taylor expansions read

$$\begin{aligned} \Omega(t_0 + h) &= \sum_{k=0}^{s=2} \frac{h^k}{k!}\Omega^{(k)}(t_0), & P(t_0 + h) &= \sum_{k=0}^{s=2} \frac{h^k}{k!}P^{(k)}(t_0), \\ \Omega_1(t_0 + h) &= \sum_{k=0}^{s=2} \frac{h^k}{k!}\Omega_1^{(k)}(t_0), & P_1(t_0 + h) &= \sum_{k=0}^{s=2} \frac{h^k}{k!}P_1^{(k)}(t_0). \end{aligned}$$

Thus we have to compute the derivatives of $\Omega(t)$ and $P(t)$ around $t_0 = 0$ up to second order. While $\dot{\Omega}(0)$ is chosen as $iP(0) = iP_0$, it is evident that

$$\dot{P}(0) = g(U_0) \quad \text{and} \quad \ddot{\Omega}(0) = i\dot{P}(0) = ig([U_0])$$

holds. The second derivative of the function P reads

$$\ddot{P}(0) = \dot{g}\Big(\Big[\exp\Big(\Omega(0)\Big)\, U_0\Big]\Big) = \frac{\partial g}{\partial U}([U_0]) \cdot \exp\Big(\Omega(0)\Big)\, \dot{\Omega}(0) U_0 = \frac{\partial g}{\partial U}(U_0) \cdot iP_0 U_0.$$

Furthermore, the derivatives of the numerical solution Ω_1, P_1 have to be computed up to second order using the equations (3.22). Hence, the derivatives at $t_0 = 0$ read

$$\begin{aligned}
\dot{\Omega}_1(0) &= \sum_{j=1}^{s=2} b_j Z_j(0)\,, \\
\ddot{\Omega}_1(0) &= \sum_{j=1}^{s=2} b_j \Big(2\dot{Z}_j(0)\Big)\,, \\
\dot{P}_1(0) &= \sum_{j=1}^{s=2} \hat{b}_j g\Big(\Big[\exp\Big(Y_j(0)\Big)\, U_0\Big]\Big)\,, \\
\text{and} \quad \ddot{P}_1(0) &= 2\sum_{j=1}^{s=2} \hat{b}_j \frac{\partial g}{\partial U}\Big(\Big[\exp\Big(Y_j(0)\Big)\, U_0\Big]\Big) \exp\Big(Y_j(0)\Big)\, \dot{Y}_j(0) U_0.
\end{aligned}$$

Hence, we have to compute

$$\begin{aligned}
Y_j(0) &= 0\,, \\
Z_j(0) &= iP_0\,, \\
\dot{Y}_j(0) &= \sum_{k=1}^{s=2} a_{j,k} Z_k(0) = \Big(\sum_{k=1}^{s=2} a_{j,k}\Big)\, iP_0\,, \\
\text{and} \quad \dot{Z}_j(0) &= \sum_{k=1}^{s=2} \hat{a}_{j,k} g\Big(\Big[\exp\Big(Y_k(0)\Big)\, U_0\Big]\Big) = \Big(\sum_{k=1}^{s=2} \hat{a}_{j,k}\Big)\, g([U_0]).
\end{aligned}$$

Afterwards, we insert these values in the derivatives of the numerical solution and get

$$\begin{aligned}
\dot{\Omega}_1(0) &= \Big(\sum_{j=1}^{s=2} b_j\Big)\, iP_0\,, \\
\ddot{\Omega}_1(0) &= \Big(\sum_{j=1}^{s=2} b_j \Big(2\sum_{k=1}^{s=2} \hat{a}_{j,k}\Big)\Big)\, g([U_0])\,, \\
\dot{P}_1(0) &= \Big(\sum_{j=1}^{s=2} \hat{b}_j\Big)\, g([U_0])\,, \\
\text{and} \quad \ddot{P}_1(0) &= 2\Big(\sum_{j=1}^{s=2} \hat{b}_j \Big(\sum_{k=1}^{s=2} a_{j,k}\Big)\Big)\, \frac{\partial g}{\partial U}([U_0]) iP_0 U_0.
\end{aligned}$$

At this point we have collected the necessary information to set up the order conditions for the Runge-Kutta scheme (3.26) with truncated function $\dot{\Omega}(t) =$

$iP(t)$. We get the conditions

$$\dot{\Omega}_1(0) = \dot{\Omega}(0): \qquad \sum_{j=1}^{s=2} b_j = 1 \quad \text{for the links } U \tag{3.27}$$

$$\dot{P}_1(0) = \dot{P}(0): \qquad \text{and} \quad \sum_{j=1}^{s=2} \hat{b}_j = 1 \quad \text{for the momenta } P \tag{3.28}$$

to attain convergence order one. Convergence order two will be obtained by

$$\ddot{\Omega}_1(0) = \ddot{\Omega}(0): \qquad \sum_{j=1}^{s=2}\Big(b_j \sum_{k=1}^{s=2} \hat{a}_{j,k}\Big) = \frac{1}{2} \quad \text{for the links } U \tag{3.29}$$

$$\dot{P}_1(0) = \dot{P}(0): \quad \text{and} \quad \sum_{j=1}^{s=2}\Big(\hat{b}_j \sum_{k=1}^{s=2} a_{j,k}\Big) = \frac{1}{2} \quad \text{for the momenta } P. \tag{3.30}$$

Remark 3.27. *The model error due to the truncation $\dot{\Omega} = iP$ is of order* 3. *Since the used partitioned Runge-Kutta method is symmetric, the order conditions* (3.29) *and* (3.30) *are already fulfilled through the conditions for the symmetry and convergence order 1 (see remark 3.24)*

Coefficients

Due to the fact that the symmetric partitioned Runge-Kutta method with convergence order two can be seen as reformulated Leapfrog method, we take the coefficients of the Leapfrog method. They are denoted as Butcher tableaus and given in table 3.1. It is shown in the appendix B.1, that they fulfill the order conditions (3.27)-(3.30) for convergence order two as well as the symmetry conditions given in equation (3.12).

This means we have indeed a symmetric Runge-Kutta method of convergence order two. As we will see in the next chapter, the simulation of a gauge field with this method and these coefficients works well.

links U:

0	0	0
1	$\frac{1}{2}$	$\frac{1}{2}$
	$\frac{1}{2}$	$\frac{1}{2}$

momenta P:

$\frac{1}{2}$	$\frac{1}{2}$	0
$\frac{1}{2}$	$\frac{1}{2}$	0
	$\frac{1}{2}$	$\frac{1}{2}$

Table 3.1: Butcher tableaus of the coefficients of the Störmer-Verlet (=Leapfrog) method as partitioned Runge-Kutta method

3.4.2 Symmetric Partitioned Runge-Kutta Method of Convergence Order 3

Finally, our aim is to investigate symmetric Runge-Kutta methods of convergence order larger than two for the problem of solving

$$\dot{U}(t) = iP(t)U(t) \qquad \text{and} \qquad i\dot{P}(t) = ig([U(t)]) \tag{3.1}$$

with initial values $\big(U_0, P_0\big)$. As in section 3.4.1, we rewrite our differential equations (3.1) as

$$\begin{aligned} \dot{\Omega}(t) &= f\big(\Omega(t), iP(t)\big) = d\exp_\Omega^{-1}(iP) \\ \text{and} \quad i\dot{P}(t) &= ig\Big(\Big[\exp\big(\Omega(t)\big)\cdot U(t_0)\Big]\Big). \end{aligned} \qquad \text{(see (3.10))}$$

Using $U(t) = \exp\big(\Omega(t)\big)\cdot U(t_0)$ and the truncation

$$\dot{\Omega}(t) = f(\Omega(t), iP(t)) = iP(t) - \frac{1}{2}[\Omega(t), iP(t)]$$

given in equation (3.24), we get the differential equations

$$\begin{aligned} \dot{\Omega}(t) &= f\big(\Omega(t), iP(t)\big) = iP(t) - \frac{1}{2}[\Omega(t), iP(t)] \\ \text{and} \quad i\dot{P}(t) &= ig\Big(\Big[\exp\big(\Omega(t)\big)\cdot U(t_0)\Big]\Big). \end{aligned}$$

Due to the symmetry of the Runge-Kutta scheme, we have to use the more complicated Runge-Kutta method prescribed in lemma 3.21:

Lemma 3.28 (Partitioned Runge-Kutta Method for Convergence Order 3)**.**

Compute

$$\Omega_1 = h\sum_{j=1}^{s=3} b_j K_j \quad \textit{and} \quad iP_1 = iP_0 + h\sum_{j=1}^{s=3} \hat{b}_j L_j \tag{3.31}$$

with increments

$$\begin{aligned} K_j &= f\big(Y_j, Z_j\big) = Z_j - \frac{1}{2}[Y_j, Z_j] \\ \textit{and} \quad L_j &= ig\Big(\Big[\exp\big(W_j\big)\,\exp\big(\frac{1}{2}\Omega_1\big)\,U_0\Big]\Big) \end{aligned}$$

depending on the functions

$$Y_j = h\sum_{k=1}^{s=3} a_{j,k}K_k\,, \quad Z_j = iP_0 + h\sum_{k=1}^{s=3} \hat{a}_{j,k}L_k \quad \textit{and} \quad W_j = h\sum_{k=1}^{s=3} c_{j,k}K_k$$

for the stages $j = 1, \ldots, s$ and coefficients $b_j, a_{j,k}, c_{j,k}$, $\hat{b}_j, \hat{a}_{j,k}$ used above. At the end we attain the numerical solution of our problem (3.1) *as*

$$\big(U_1, P_1\big) = \big(\exp(\Omega_1)U_0, P_1\big).$$

Order conditions

Again, we have to compare the Taylor series of the exact and numerical solution of the differential equations (3.31) to determine the convergence order. Thus, we compute the Taylor expansions of these solutions:

$$\begin{aligned} \Omega(t_0+h) &= \sum_{k=0}^{s=3} \frac{h^k}{k!}\Omega^{(k)}(t_0), & P(t_0+h) &= \sum_{k=0}^{s=3} \frac{h^k}{k!}P^{(k)}(t_0), \\ \Omega_1(t_0+h) &= \sum_{k=0}^{s=3} \frac{h^k}{k!}\Omega_1^{(k)}(t_0), & P_1(t_0+h) &= \sum_{k=0}^{s=3} \frac{h^k}{k!}P_1^{(k)}(t_0). \end{aligned}$$

These Taylor expansions are performed around $t_0 = 0$. Since the detailed computation is very long and similar to the calculation in subsection 3.4.2, the derivation of the Taylor expansions are left out. Just the results are given here. The Taylor expansions of the exact solution read

$$\begin{aligned}
\dot{\Omega}(0) &= iP_0\,,\\
\ddot{\Omega}(0) &= ig([U_0])\,,\\
\dddot{\Omega}(0) &= i\frac{\partial g}{\partial U}\Big([U_0]\Big)\cdot iP_0U_0 - \frac{1}{2}\Big[ig\Big([U_0]\Big), iP_0\Big] - \Big[iP_0, ig\Big([U_0]\Big)\Big]\,,\\
\dot{P}(0) &= g([U_0])\,,\\
\ddot{P}(0) &= \frac{\partial g}{\partial U}\Big([U_0]\Big)\cdot iP_0U_0\, ig([U_0])\,,\\
\dddot{P}(0) &= \frac{\partial^2 g}{\partial U^2}\Big([U_0]\Big)\cdot\Big(iP_0U_0\Big)^2 + \frac{\partial g}{\partial U}\Big([U_0]\Big)\cdot\Big((iP_0)^2 + ig\Big([U_0]\Big)\Big)\,U_0\,.
\end{aligned} \tag{3.32}$$

In comparison, the Taylor expansions of the numerical solution are computed with $s = 3$ stages as

$$\begin{aligned}
\dot{\Omega}_1(0) &= \Big(\sum_{j=1}^{s} b_j\Big)\cdot iP_0\,,\\
\ddot{\Omega}_1(0) &= 2\Big(\sum_{j=1}^{s} b_j \sum_{k=1}^{s}\hat{a}_{j,k}\Big)\cdot ig\Big([U_0]\Big)\,,\\
\dddot{\Omega}_1(0) &= 3\sum_{j=1}^{s} b_j\Bigg(2\sum_{k=1}^{s}\hat{a}_{j,k}\Big(\sum_{l=1}^{s} c_{k,l} + \frac{1}{2}\sum_{m=1}^{s} b_m\Big)\cdot i\frac{\partial g}{\partial U}\Big([U_0]\Big)\cdot iP_0\cdot U_0\\
&\qquad - \Big(\sum_{k=1}^{s} a_{j,k}\sum_{l=1}^{s}\hat{a}_{k,l}\Big)\Big[ig\Big([U_0]\Big), iP_0\Big]\\
&\qquad - \Big(\sum_{k=1}^{s} a_{j,k}\Big)\cdot\Big(\sum_{l=1}^{s}\hat{a}_{j,l}\Big)\cdot\Big[iP_0, ig\Big([U_0]\Big)\Big]\Bigg)\,,\\
\dot{P}_1(0) &= \Big(\sum_{j=1}^{s}\hat{b}_j\Big)\cdot g\Big([U_0]\Big)\,,\\
\ddot{P}_1(0) &= 2\sum_{j=1}^{s}\hat{b}_j\Big(\sum_{k=1}^{s} c_{j,k} + \frac{1}{2}\sum_{l=1}^{s} b_l\Big)\cdot\frac{\partial g}{\partial U}\Big([U_0]\Big)\cdot iP_0\cdot U_0\,,\\
\dddot{P}_1(0) &= 3\sum_{j=1}^{s}\hat{b}_j\Big(\sum_{k=1}^{s} c_{j,k} + \frac{1}{2}\sum_{l=1}^{s} b_l\Big)^2\cdot\frac{\partial^2 g}{\partial U^2}\Big([U_0]\Big)\Big(iP_0U_0\Big)^2\\
&\quad + 3\sum_{j=1}^{s}\hat{b}_j\Big(\sum_{k=1}^{s} c_{j,k} + \frac{1}{2}\sum_{l=1}^{s} b_l\Big)^2\cdot\frac{\partial g}{\partial U}\Big([U_0]\Big)\Big(iP_0\Big)^2 U_0\\
&\quad + 3\sum_{j=1}^{s}\hat{b}_j\Big(2\sum_{k=1}^{s} c_{j,k}\sum_{l=1}^{s}\hat{a}_{k,l} + \sum_{k=1}^{s} b_k\sum_{l=1}^{s}\hat{a}_{k,l}\Big)\cdot\frac{\partial g}{\partial U}\Big([U_0]\Big)\, ig\Big([U_0]\Big)\, U_0\,.
\end{aligned}$$

At this point the necessary information is given to compute the order conditions. We get the conditions for convergence order m by a comparison of the m-th derivative of the exact solution $\Omega(t), iP(t)$ with the numerical one $\Omega_1(t), iP_1(t)$:

Conditions for convergence order 1

$$\dot{\Omega}_1(0) = \dot{\Omega}(0): \qquad \sum_{j=1}^{s=3} b_j = 1$$

$$\dot{P}_1(0) = \dot{P}(0) \qquad \sum_{j=1}^{s=3} \hat{b}_j = 1.$$

Conditions for convergence order 2

$$\ddot{\Omega}_1(0) = \ddot{\Omega}(0): \qquad \sum_{j=1}^{s} b_j \sum_{k=1}^{s} \hat{a}_{j,k} = \frac{1}{2}$$

$$\ddot{P}_1(0) = \ddot{P}(0): \qquad \sum_{j=1}^{s} \hat{b}_j \Big(\sum_{k=1}^{s} c_{j,k} + \frac{1}{2} \sum_{l=1}^{s} b_l \Big) = \frac{1}{2}$$

Conditions for convergence order 3

$$\dddot{\Omega}_1(0) = \dddot{\Omega}(0): \qquad \sum_{j=1}^{s} b_j \sum_{k=1}^{s} \hat{a}_{j,k} \Big(\sum_{l=1}^{s} c_{k,l} + \frac{1}{2} \sum_{m=1}^{s} b_m \Big) = \frac{1}{6}$$

$$\sum_{j=1}^{s} b_j \sum_{k=1}^{s} a_{j,k} \sum_{l=1}^{s} \hat{a}_{k,l} = \frac{1}{6}$$

$$\sum_{j=1}^{s} b_j \Big(\sum_{k=1}^{s} a_{j,k} \Big) \Big(\sum_{l=1}^{s} \hat{a}_{j,l} \Big) = \frac{1}{3}$$

$$\dddot{P}_1(0) = \dddot{P}(0): \qquad \sum_{j=1}^{s} \hat{b}_j \Big(\sum_{k=1}^{s} c_{j,k} + \frac{1}{2} \sum_{l=1}^{s} b_l \Big)^2 = \frac{1}{3}$$

$$\sum_{j=1}^{s} \hat{b}_j \Big(2 \sum_{k=1}^{s} c_{j,k} \sum_{l=1}^{s} \hat{a}_{k,l} + \sum_{m=1}^{s} b_m \sum_{n=1}^{s} \hat{a}_{m,n} \Big) = \frac{1}{3}\,.$$

Remark 3.29. *The method is symmetric and has convergence order 3. This implies a convergence order 4, provided that the model error is small enough (see remark 3.24).*

The truncation $\dot{\Omega}(t) = iP(t) - \frac{1}{2}[\Omega(t), iP(t)]$ implies a model error of order 4. For a convergence order of 4, the model error has to be 5 such that the additional term

$$\frac{1}{6}\Big[\Omega(t), [\Omega(t), iP(t)]\Big] =: \dot{X}(t) \tag{3.33}$$

has to be used.

Remark 3.30 (Vanishing additional term of the model evaluated at $t_0 = 0$)**.** *The term $\dot{X}(t) := \frac{1}{6}\Big[\Omega(t), [\Omega(t), iP(t)]\Big]$ and its derivations $\ddot{X}(t)$ and $\dddot{X}(t)$ evaluated at $t_0 = 0$ lead to*

$$\dot{X}(0) = 0\,, \quad \ddot{X}(0) = 0\,, \quad \textit{and} \quad \dddot{X}(0) = 0\,.$$

.

Proof. Compute the derivatives of the term (3.33) up to third order and evaluate them at $t_0 = 0$. We get

$$\begin{aligned}
\ddot{X}(t) &= \frac{1}{6}\Big[\dot{\Omega}(t), [\Omega(t), iP(t)]\Big] + \frac{1}{6}\Big[\Omega(t), [\dot{\Omega}(t), iP(t)] + [\Omega(t), i\dot{P}(t)]\Big] \\
\dddot{X}(t) &= \frac{1}{6}\Big[\ddot{\Omega}(t), [\Omega(t), iP(t)]\Big] + \frac{1}{6}\Big[\dot{\Omega}(t), [\dot{\Omega}(t), iP(t)] + [\Omega(t), i\dot{P}(t)]\Big] \\
&\quad + \frac{1}{6}\Big[\dot{\Omega}(t), [\dot{\Omega}(t), iP(t)] + [\Omega(t), i\dot{P}(t)]\Big] \\
&\quad + \frac{1}{6}\Big[\Omega(t), [\ddot{\Omega}(t), iP(t)] + [\dot{\Omega}(t), i\dot{P}(t)] + [\dot{\Omega}(t), i\dot{P}(t)] + [\Omega(t), i\ddot{P}(t)]\Big].
\end{aligned}$$

Since the initial value $\Omega(0)$ equals zero, an evaluation at $t_0 = 0$ leads to

$$\begin{aligned}
\dot{X}(0) &= \frac{1}{6}\Big[0, [\Omega(t), iP(t)]\Big] = 0 \\
\ddot{X}(0) &= \frac{1}{6}\Big[\dot{\Omega}(t), [0, iP(t)]\Big] + \frac{1}{6}\Big[0, [\dot{\Omega}(t), iP(t)] + [\Omega(t), i\dot{P}(t)]\Big] \\
&= \frac{1}{6}\Big[\dot{\Omega}(t), 0\Big] + 0\Big] = 0 \\
\dddot{X}(0) &= \frac{1}{6}\Big[\ddot{\Omega}(t), [0, iP(t)]\Big] + \frac{1}{6}\Big[\dot{\Omega}(t), [\dot{\Omega}(t), iP(t)] + [0, i\dot{P}(t)]\Big] \\
&\quad + \frac{1}{6}\Big[\dot{\Omega}(t), [\dot{\Omega}(t), iP(t)] + [0, i\dot{P}(t)]\Big] \\
&\quad + \frac{1}{6}\Big[0, [\ddot{\Omega}(t), iP(t)] + [\dot{\Omega}(t), i\dot{P}(t)] + [\dot{\Omega}(t), i\dot{P}(t)] + [\Omega(t), i\ddot{P}(t)]\Big] \\
&= \frac{1}{6}\Big[\ddot{\Omega}(t), 0\Big] + \frac{1}{6}\Big[\dot{\Omega}(t), [\dot{\Omega}(t), iP(t)] + 0\Big] + \frac{1}{6}\Big[\dot{\Omega}(t), [\dot{\Omega}(t), iP(t)] + 0\Big] + \frac{1}{6}\cdot 0.
\end{aligned}$$

With $\dot{\Omega}(0) = iP(0) = iP_0$ it follows

$$\begin{aligned}
\dddot{X}(t) &= \frac{1}{6}\Big[\dot{\Omega}(t), [iP_0, iP_0] + 0\Big] + \frac{1}{6}\Big[\dot{\Omega}(t), [iP_0, iP_0] + 0\Big] \\
&= \frac{1}{6}\Big[\dot{\Omega}(t), 0 + 0\Big] + \frac{1}{6}\Big[\dot{\Omega}(t), 0 + 0\Big] = 0.
\end{aligned}$$

□

Due to the fact that the term (3.33) has no influence on the Taylor expansions around $t_0 = 0$, the total convergence order of the symmetric implicit partitioned Runge-Kutta method given in lemma (3.28) is 4. Hence, this method is mentioned as symmetric implicit partitioned Runge-Kutta method of convergence order 4 in the following paragraphs.

Coefficients for a partitioned Runge-Kutta method of order 4

The coefficients used for the symmetric partitioned Runge-Kutta method of convergence order 4, are chosen according to calculations performed with the computer algebra system Mathematica. For these computations, the aforementioned conditions on the convergence order and the symmetry conditions (3.15)-(3.19) are used.

links U:

$a_{j,k}$:				
0	0	0	0
0	$-\frac{\sqrt{3}}{6}$	0	$\frac{\sqrt{3}}{6}$
0	0	0	0
b_j:	0	1	0

$c_{j,k}$:			
$-\frac{\sqrt{3}}{6}$	$-\frac{\sqrt{3}}{6}$	0	0
0	0	0	0
$\frac{\sqrt{3}}{6}$	0	0	$\frac{\sqrt{3}}{6}$
b_j:	0	1	0

momenta P:

$\hat{a}_{j,k}$:			
$\frac{3+\sqrt{3}}{6}$	$\frac{3+\sqrt{3}}{6}$	0	0
$\frac{1}{2}$	$\frac{3+\sqrt{3}}{12}$	0	$\frac{3-\sqrt{3}}{12}$
$\frac{3-\sqrt{3}}{6}$	$\frac{1}{2}$	0	$-\frac{\sqrt{3}}{6}$
$\hat{b}_j$:	$\frac{1}{2}$	0	$\frac{1}{2}$

Table 3.2: Butcher tableaus of the coefficients of the partitioned Runge-Kutta method of convergence order 3

Chapter 4

Simulation

4.1 Model: An $SU(2,\mathbb{C})$ Lattice Gauge Field

In this section we describe the simulation of an $SU(2,\mathbb{C})$ gauge field situated on a lattice executed with a Hybrid Monte Carlo method. Concerning the replicability of the simulation, the required facts of the previous chapters are gleaned. We start with a reminder of the Hybrid Monte Carlo algorithm described in paragraph 2.2. It works as follows:

1. Select an initial configuration $[U^i]$ of links randomly.
2. Create conjugated momenta $[P^i]$ randomly according to the equation
$$\mathcal{P}_G([P]) \sim \exp\Big(-\frac{1}{2}\sum_{x,\mu} tr(P_{x,\mu}^2)\Big) = \exp\Big(-E_{kin}\big([P]\big)\Big). \tag{2.11}$$
3. Molecular Dynamics step: Reach the new configuration $[U^j, P^j]$ using the symmetric partitioned Runge-Kutta method.
4. Monte Carlo step:
Compute the difference of the Hamiltonians as $\Delta\mathcal{H} := \mathcal{H}^j - \mathcal{H}^i$. Then accept the new configuration with acceptance probability
$$P_A\big([U^i, P^i] \to [U^j, P^j]\big) = \min\Big(1, \frac{p_j}{p_i}\Big) \overset{(2.9)}{=} \min\Big(1, \exp(-\Delta\mathcal{H})\Big).$$
5. Start at step 2.

Molecular Dynamics and Monte Carlo step

For the simulation of the lattice gauge field, we need the Hamiltonian and its equations of motion. These formulas are given in paragraph 1.2 and 1.3. The Hamiltonian is composed of the kinetic energy and the Wilson action (see equations (1.6) and (1.3)) as

$$\mathcal{H}\big([U,P]\big) = E_{kin}\big([P]\big) + S_G\big([U]\big) \tag{1.5}$$

$$\text{with} \quad E_{kin}\Big([P]\Big) = \frac{1}{2}\sum_x \sum_{\mu=0,1} tr\Big(P_{x,\mu}^2\Big)$$
$$\text{and} \quad S_G\Big([U]\Big) = \sum_x \beta\Big(1 - \frac{1}{N}\,\mathrm{Re}\Big(tr(U_{01}(x))\Big)\Big)$$

The Hamiltonian equations of motion for an an $SU(2,\mathbb{C})$ gauge field

$$\frac{\partial \mathcal{H}\Big([U,P]\Big)}{\partial P_{x,\mu}} = \quad \dot{U}_{x,\mu} = iP_{x,\mu}U_{x,\mu} \tag{1.7}$$

$$\text{and} \qquad \frac{\partial \mathcal{H}\Big([U,P]\Big)}{\partial U_{x,\mu}} = -\dot{P}_{x,\mu} = -i\frac{\beta}{N}\Big\{U_{x,\mu}V_{x,\mu}\Big\}_{TA} \tag{1.8}$$

are used in the Molecular Dynamics step. They will be solved with the partitioned Runge-Kutta scheme described in section 3.3.

Representation of the links and its conjugated momenta

We have a gauge field composed of link variables $U_{x,\mu}$ situated in the special unitary Lie group $SU(N,\mathbb{C})$ mentioned in section 1.1. Furthermore, we have an additional field with fictous conjugated momenta $P_{x,\mu}$ used for the computation of the Hamiltonian $\mathcal{H}\Big([U,P]\Big)$. These momenta are traceless and hermitian matrices, such that $iP_{x,\mu}$ is an element of the Lie algebra $\mathfrak{su}(N,\mathbb{C})$.

Notation 4.1. *For convenience of notation, we leave away the indices x and μ of the links $U_{x,\mu}$ and momenta $P_{x,\mu}$ as of now. Hence, the notations*

$$U := U_{x,\mu} \quad \textit{and} \quad P := P_{x,\mu}$$

mean one special but arbitrary pair of matrices.

Due to the fact that we simulate a gauge field of $SU(2,\mathbb{C})$ matrices, the variable N used in the equations (1.3) and (1.8) is set to $N = 2$. Thus, we deal with 2×2-matrices in the whole simulation.

Definition 4.2 (Pauli matrices)**.** *The Pauli matrices are a set of 2×2 complex Hermitian and unitary matrices*

$$\sigma_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad \sigma_2 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \qquad \sigma_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Remark 4.3 (Pauli matrices)**.** *Note that the complex conjugate and transposed matrices of the Pauli matrices are the Pauli matrices itself such that it holds*

$$\sigma_1^\dagger = \sigma_1, \quad \sigma_2^\dagger = \sigma_2, \quad \sigma_3^\dagger = \sigma_3. \tag{4.1}$$

The product of 2 Pauli matrices can be described through the Levi-Cita-symbol ε_{jkl} and the Kronecker delta δ_{jk} as

$$\sigma_j\sigma_k = \delta_{jk} + i\sum_k \varepsilon_{jkl}\sigma_l.$$

Hence, we get

$$\sigma_j\sigma_k = \begin{cases} i\sigma_l & \text{if } (j,k) \text{ is equal to } (1,2) \text{ or } (2,3) \text{ or } (3,1) \\ I_2 & \text{if } j \text{ is equal to } k \\ -i\sigma_l & \text{if } (j,k) \text{ is equal to } (2,1) \text{ or } (3,2) \text{ or } (1,3). \end{cases}$$

Furhermore, it holds $det(\sigma_j) = -1$ *and* $tr(\sigma_j) = 0$ *for* $j = 1,2,3$.

This means, the link variables U and the momenta P can be created as a linear combination of these Pauli matrices and the identity I_2.

Lemma 4.4. *Every matrix* $U \in SU(2,\mathbb{C})$ *can be represented by*

$$U = \sum_{j=1}^{3} x_j i\sigma_j + x_4 \cdot I_2 = \begin{pmatrix} x_4 + ix_3, & x_2 + ix_1 \\ -x_2 + ix_1, & x_4 - ix_3 \end{pmatrix}$$

with complex i *and a vector* $x \in \mathbb{R}^4$ *and* $\|x\|_2 = 1$, *such that* x *is an element of the unit sphere* S_3.

Proof. We have to show that the complex conjugate and transposed matrix $U^\dagger$ will be the inverse of U. Moreover, the determinant of U has to be equal to 1. So with

$$U^\dagger = \sum_{j=1}^{3} x_j \cdot (-i\sigma_j^\dagger) + x_4 \cdot I_2 \overset{(4.1)}{=} -\sum_{j=1}^{3} x_j i\sigma_j + x_4 \cdot I_2$$

we get

$$\begin{aligned} U \cdot U^\dagger &= \left(\sum_{j=1}^{3} x_j i\sigma_j + x_4 \cdot I_2\right) \cdot \left(-\sum_{j=1}^{3} x_j i\sigma_j + x_4 \cdot I_2\right) \\ &= \left(\sum_{j=1}^{3} x_j \sigma_j\right)^2 + (x_4 - x_4) \cdot \left(\sum_{j=1}^{3} x_j i\sigma_j\right) + x_4^2 \cdot I_2 \\ &= \sum_{j=1}^{3} x_j^2 \sigma_j^2 + x_1x_2(\sigma_1\sigma_2 + \sigma_2\sigma_1) + x_1x_3(\sigma_1\sigma_3 + \sigma_3\sigma_1) + x_2x_3(\sigma_2\sigma_3 + \sigma_3\sigma_2) + x_4^2 \cdot I_2 \\ &= \sum_{j=1}^{4} x_j^2 \cdot I_2 + x_1x_2(\sigma_3 - \sigma_3) + x_1x_3(-\sigma_2 + \sigma_2) + x_2x_3(\sigma_1 - \sigma_1) \\ &= \|x\| \cdot I_2 = 1 \cdot I_2 = I_2 \end{aligned}$$

and also $U^\dagger \cdot U = I_2$.

It holds $\det(U) = 1$ because

$$\begin{aligned} \det(U) &= (x_4 + ix_3) \cdot (x_4 - ix_3) - (-x_2 + ix_1) \cdot (x_2 + ix_1) \\ &= x_4^2 - i^2x_3^2 + (x_2^2 - i^2x_1^2) = x_4^2 + x_3^2 + x_1^2 + x_2^2 = \|x\|_2 = 1. \end{aligned}$$

□

Lemma 4.5. *The conjugated momenta P of the links have to be hermitian matrices with trace zero. They are constructed as linear combinations of the traceless and hermitian Pauli matrices σ_1, σ_2 and σ_3. Additionally, the momenta have to be gaussian distributed according to the probability distribution $P_G([P])$ given in equation (2.11). The choice of*

$$P = \frac{1}{\sqrt{2}} \sum_{j=1}^{3} y_j \sigma_j = \frac{1}{\sqrt{2}} \begin{pmatrix} y_3 & y_1 - iy_2 \\ y_1 + iy_2 & -y_3 \end{pmatrix}$$

with gaussian distributed random numbers y_j ($j = 1, 2, 3$) (with mean 0 and variance 1) fulfills these requirements.

Proof. It holds

$$tr(\sigma_i \sigma_j) = \begin{cases} 2 & \text{if } i = j \\ 0 & \text{if } i \neq j. \end{cases}$$

and thus,

$$tr(P^2) = \frac{1}{2} tr\Big(\big(\sum_{j=1}^{3} y_j \sigma_j \big)^2 \Big) = \frac{1}{2} \sum_{i,j=1}^{3} y_i y_j tr(\sigma_i \sigma_j) = \sum_{j=1}^{3} y_j^2.$$

With gaussian distributed random numbers y_j we get

$$\exp\Big(-\frac{1}{2} tr(P^2) \Big) = \exp\Big(-\frac{1}{2} \sum_j y_j^2 \Big)$$

such that the the configuration $[P]$ will be gaussian distributed as well. □

4.2 Details of the Simulation

There are different integration schemes implemented using the software package Matlab. First of all, the symmetric partitioned Runge-Kutta schemes of convergence order 2 and 4 are used. Since these new schemes are compared with the well known Leapfrog (=Störmer-Verlet) method described in algorithm 3.1, the Leapfrog method is computed as well.
The symmetric partitioned Runge-Kutta schemes are implemented using the coefficients given in table 3.1 and 3.2. Due to the fact that these Runge-Kutta methods are implicit, a fixed point iteration is needed to compute the increments K_j, L_j, Y_j, Z_j and W_j for $j = 1, \ldots, s$. This has to be done for the whole field at once.

Notation 4.6 (Increments for the whole field and all stages)**.** *We have a field of size $L \times T$. Since we simulate a 2-dimensional lattice, we concern about $2 \times L \times T$ links U with its conjugated momenta P. These elements will be computed through a Runge-Kutta method using the increments K_j and L_j with $j = 1, \ldots, s$ stages. Due to the discriminability of these increments for the different positions in the field, we denote the increments K_j and L_j used to compute the link U at position x with $K_{x,j}$ and $L_{x,j}$.*

The fixed point iteration works as follows:

- Start with the initial values $K^0_{x,j}$ and $L^0_{x,j}$ set to 0. Set the index $i = 0$.
- Compute
  - Y^{i+1}_j, Z^{i+1}_j and W^{i+1}_j for $j = 1, \ldots, s$ (from K^i_j and L^i_j)
  - and K^{i+1}_j and L^{i+1}_j for $j = 1, \ldots, s$ depending on the above values

  for all field positions x until

$$\max\left(\frac{\|K^{i+1}_{x,j} - K^i_{x,j}\|_2}{\|K^{i+1}_{x,j}\|_2}, \frac{\|L^{i+1}_{x,j} - L^i_{x,j}\|_2}{\|L^{i+1}_{x,j}\|_2}, j = 1 \ldots, s\right) \tag{4.2}$$

  is smaller than a predifined value. Thereby, the index i is incremented in each step.

In the described simulation, this predefined value is set to 10^{-8} without further considerations. All increments are computed using its matrix notation. For a temporization, they can be rewritten in terms of a base of Pauli matrices (see definition 4.2), such that only its factors have to be used in the fixed point iteration.

4.3 Results

We simulate a 2-dimensional lattice gauge field with periodic boundary conditions using a Hybrid Monte Carlo method. During the Molecular Dynamics step, a whole trajectory is computed. The length τ of the trajectory is fixed and set to 1 in the whole simulation. This means that $n = \frac{1}{h}$ integration steps with step size h are performed to compute one trajectory. Afterwards, the energy change along a trajectory $\Delta\mathcal{H}$ is calculated and used in the Monte Carlo step. Since the expectation values of different variables have to be compared, there are 5000 trajectories computed in each simulation.

4.3.1 Model of a 4x4-Lattice with $\beta = 2.0$

As a first model, a 4×4 lattice with gauge coupling $\beta = 2.0$ is computed applying the aforementioned different integration schemes. It can be seen in the following figures that the Runge-Kutta method of convergence order 2 coincides with the Leapfrog method. The Runge-Kutta method of convergence order 4 differs from these schemes.

Convergence Order and Acceptance Rate

The main result of this thesis is the possibility to use a Runge-Kutta scheme such that convergence orders higher than 2 can be achieved. This result can be measured through the mean absolute value of the energy change along a trajectory and is represented in figure 4.1. Thereby, a convergence order p of

the numerical integration methods yields an energy change of order p after one trajectory (see remark 3.9).
As expected, the convergence orders of the Leapfrog and the symmetric partitioned Runge-Kutta method of order 2 coincide. The convergence order of the symmetric partitioned Runge-Kutta method derived with order 3 is equal to 4. At a first glance, this result is surprising. It can be explained due to the symmetry (see remark 3.24) and the vanishing additional term of the model with model error of order h^5 (see remark 3.30).

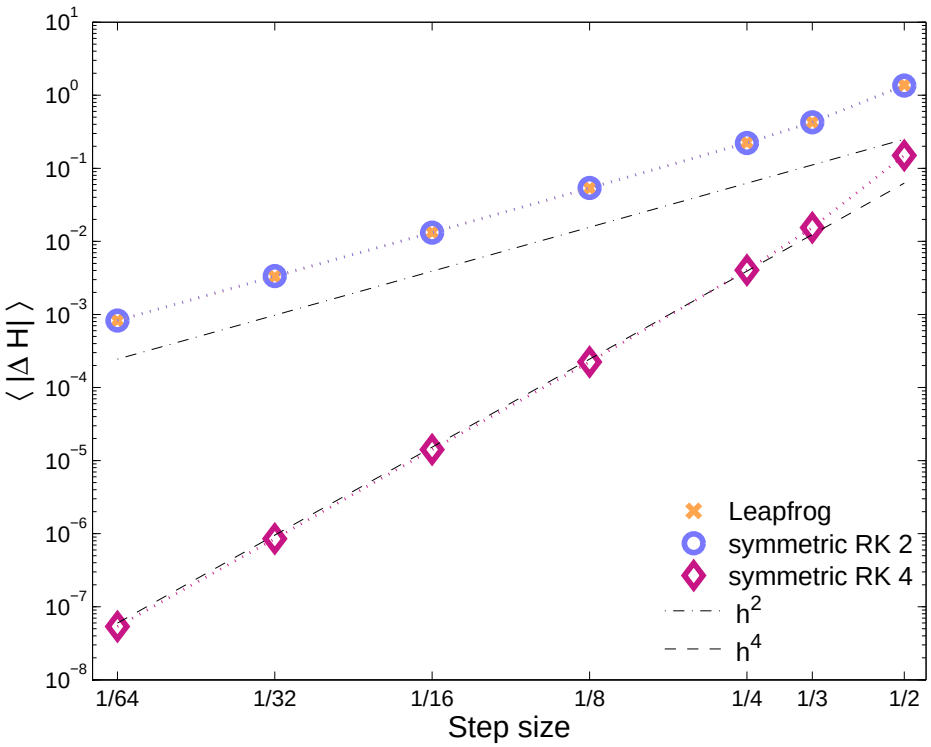

Figure 4.1: The convergence order measured through the mean absolute value of the energy change along a trajectory. It can be read off as the slopes of the lines. The convergence errors of one trajectory are of order 2 resp. 4, and thus of order 3 resp. 5 per integration step.

Thereby, the acceptance rate of the partitioned Runge-Kutta method of order 4 drops slower with increasing step size than the other investigated methods (see figure 4.2).

Correctness of the Simulation

The correctness of the symmetric partitioned Runge-Kutta method is checked through a comparison with the Leapfrog method concerning the expectation values of $\exp(-\Delta\mathcal{H})$ and the mean plaquette value. In doing so, the errors are computed regarding the integrated autocorrelation time.

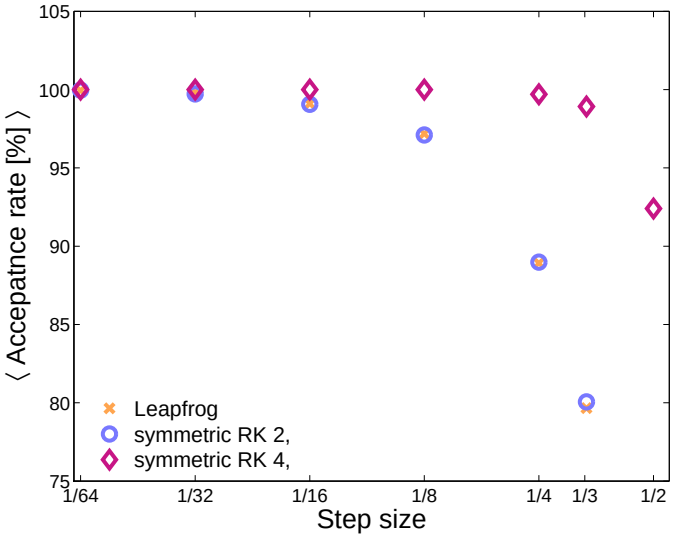

Figure 4.2: Acceptance rate of the simulations after 5000 trajectories.

The reference value of the mean plaquette value of the model is known as $\langle tr(U_\square)\rangle = 0.8669(2)$ (personal communication with Prof. Dr. Knechtli). It is met in the realized simulations as visualized in figure 4.3.

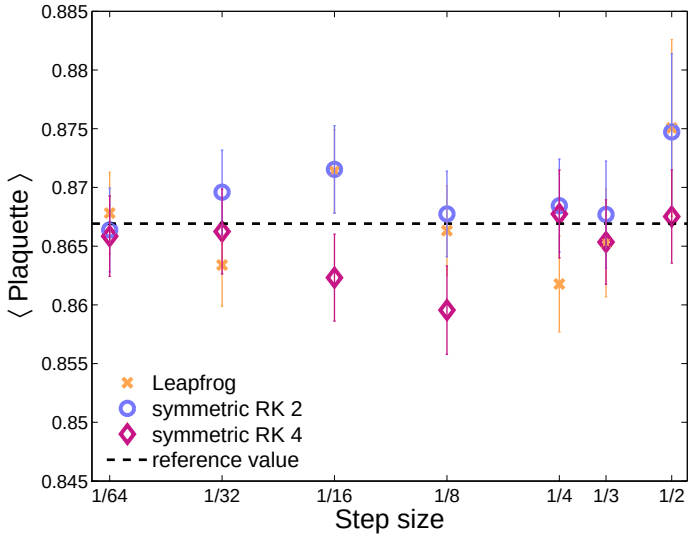

Figure 4.3: Expectation value of the mean plaquette value

The expectation value of $\exp(-\Delta\mathcal{H})$ serves as a measure for the volume-preservation. It has to be equal to one in the region of the errors. This is illustrated in figure 4.4. It can be seen that there may occur instabilities for larger step sizes if methods of convergence order 2 are used. These instabilities vanish for the Runge-Kutta method of order 4.

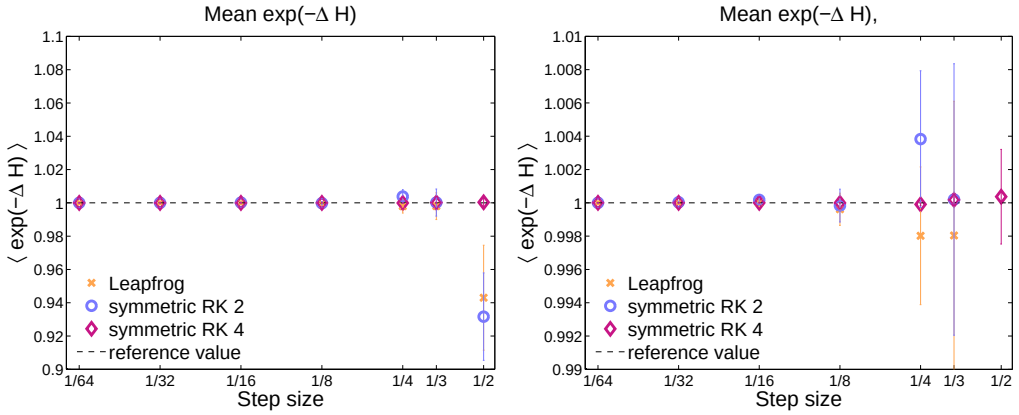

Figure 4.4: Expectation value of $\exp(-\Delta\mathcal{H})$

Computational Time

Concerning the use of the Runge-Kutta schemes described in the subsections 3.4.1 and 3.4.2, the computational time is of deep interest. Because the Runge-Kutta method is implicit, a fixed point iteration is needed (see paragraph 4.2). The time consumption strongly depends on

- the shape of the increments K_j and L_j for $j = 1, \ldots, s$,
- the used method coefficients
- and the stopping criterion of the fixed point iteration.

In figure 4.5, the mean number of fixed point iterations per integration step is shown. It increases strongly with the step size. On the other hand, the number $n = \tau/h$ of integration steps decreases with larger step sizes. Thus, the computational time decreases with larger step sizes depending on the number of fixed point iterations, except for the step size $\frac{1}{2}$ as shown in figure 4.6.
A comparison of the computational time at a similar acceptance rate leads to the conclusion that the implemented partitioned Runge-Kutta method of order 4 is approximately 14 times slower than the Leapfrog method. The details can be found in table 4.1.

step size	acceptance rate [%]	time per trajectory [s]	factor
1/6 (Leapfrog)	99.06	0.14	1
1/16 (sRK 2)	99.06	0.88	6.4
1/3 (sRK 4)	98.92	1.94	14.2

Table 4.1: Comparison of the computational time at a similar accepatance rate of approximately 99 %.

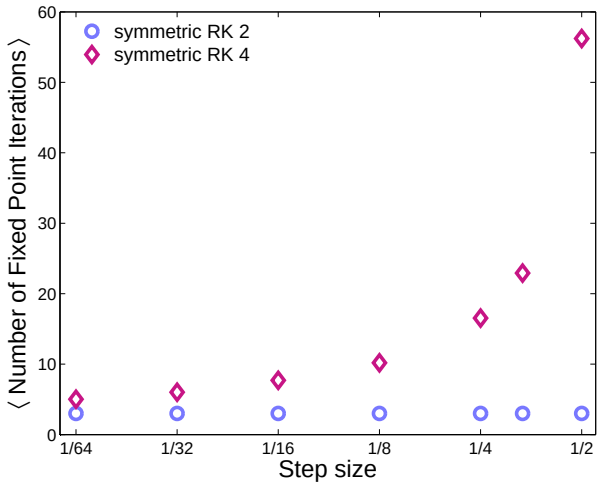

Figure 4.5: Mean number of fixed point iterations per integration step

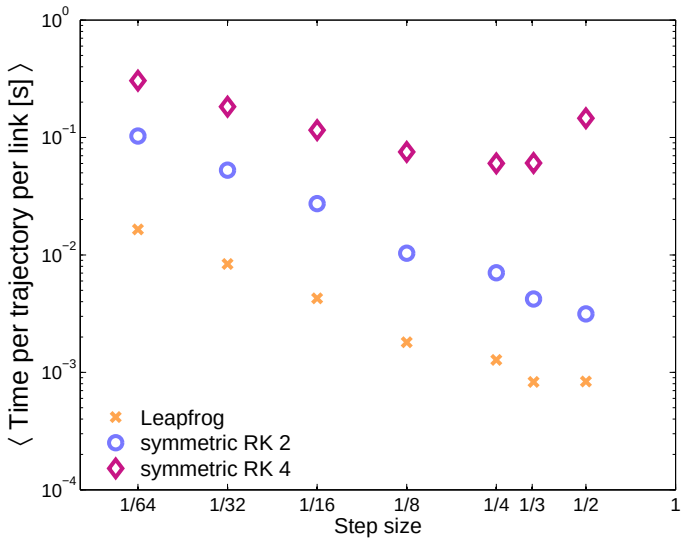

Figure 4.6: Mean time consumption per trajectory per link.

4.3.2 Volume Dependence

The volume dependence of the simulations is interesting. Thus, the simulations of the Leapfrog and the symmetric implicit partitioned Runge-Kutta method are executed for the different lattices of size $4 \times 4, 6 \times 6$ and 8×8 and $\beta = 2.0$ An examination of the convergence order shows that its properties are the same for all lattice size. Just the factor of the order increases with the lattice size for both investigated methods as visualized in figure 4.7.
Next, the expectation value of $\exp(-\Delta\mathcal{H})$ is represented in figure 4.8. It can be seen that there may occur instabilities for larger lattices concerning the larger step sizes in the Leapfrog method. In contrast, the higher-order method is able to use larger step sizes.
The most interesting point concerning the volume dependence is the behaviour

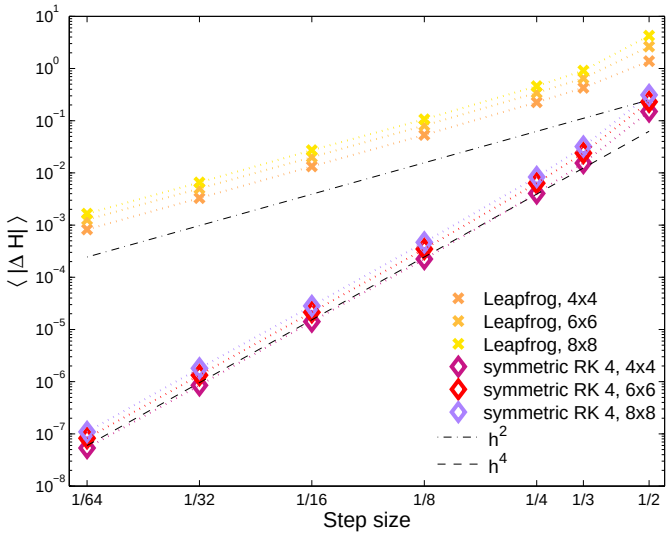

Figure 4.7: Convergence orders given through the expectation value of the absolute value of the energy change along a trajectory.

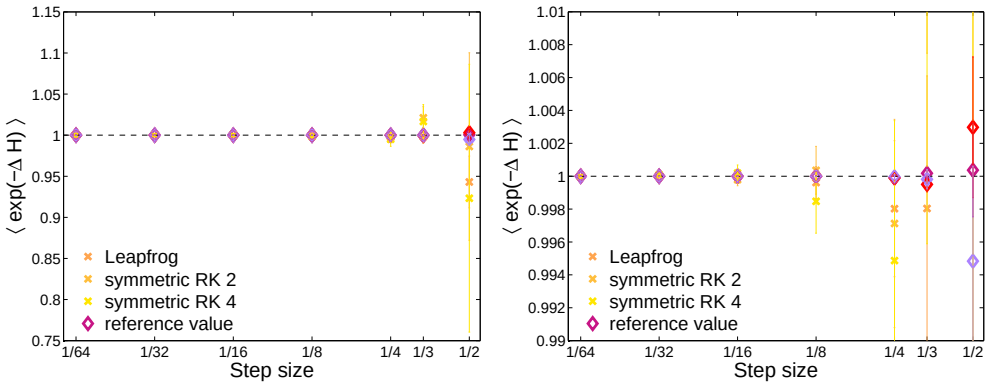

Figure 4.8: Expectation value of $\exp(-\Delta\mathcal{H})$.

of the computational cost per link. A comparison of the number of fixed point iterations (see figure 4.9) shows that this number remains for larger lattices. Thus, the time consumption does not depend on the volume of the lattice as visualized in figure 4.10.

As a last point, the time consumption per trajectory per link of both investigated methods is compared concerning a similar acceptance rate. Choosig an acceptance rate of approximately 98.5%, we get the result, that the computational time of the Leapfrog method and the symmetric partitioned Runge-Kutta method of order 4 do not grow with the volume. This result is visualized in figure 4.11.

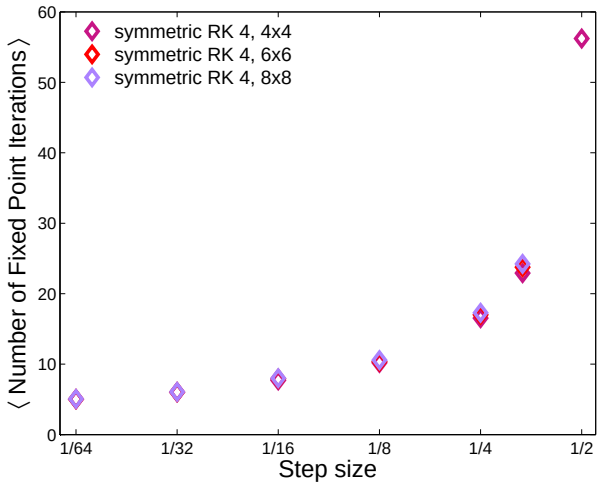

Figure 4.9: Number of fixed point iterations per integration step.

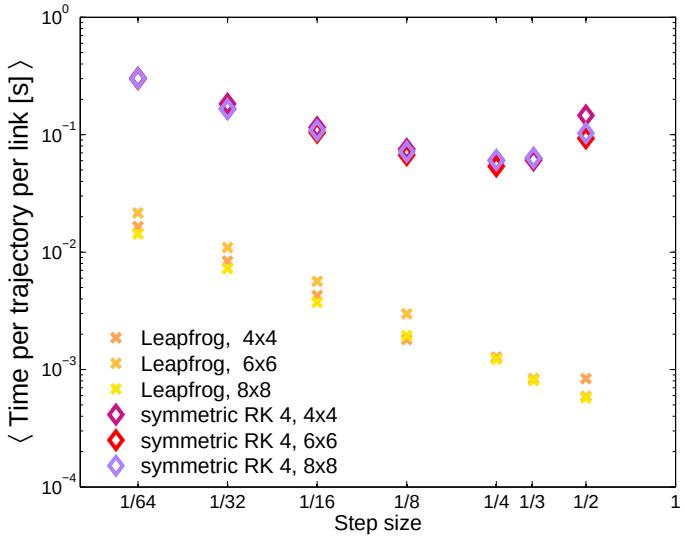

Figure 4.10: Mean time consumption per trajectory per link.

4.3.3 Dependence on the Gauge Coupling β

For completeness, the dependence on the gauge coupling term β is investigated. We fix the lattice volume to a size of 4×4 links. Then we use the different values $1.0, 2.0, 3.0$ and 4.0 for β and examine the convergence orders and the used computational time.
As a result, the convergence orders remain as illustrated in figure 4.12. Thereby, the convergence errors increase due to a factor depending on an increasing β. The number of fixed point iterations increases with β (see figure 4.13). Due to the strong dependence of the computational time on the fixed point iterations, the consumed time increases with β as well.

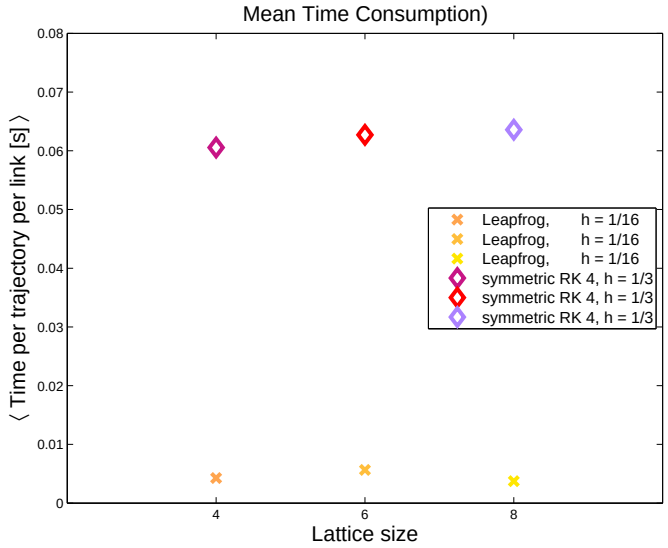

Figure 4.11: Comparison of the computational time for a similar acceptance rate.

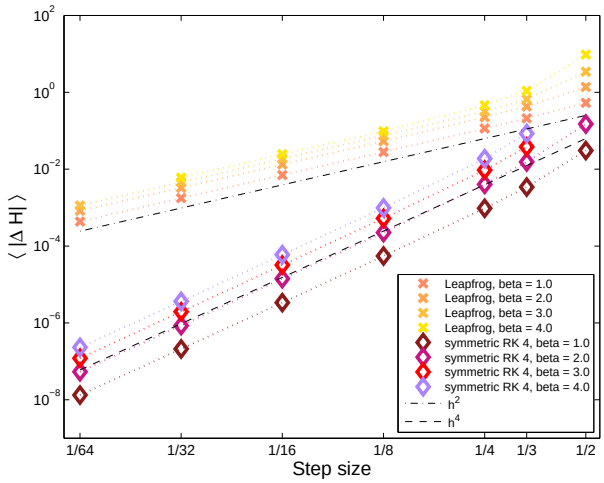

Figure 4.12: Convergence orders of different values for β. The convergence orders are 2 and 4. The factor of the error increases with β.

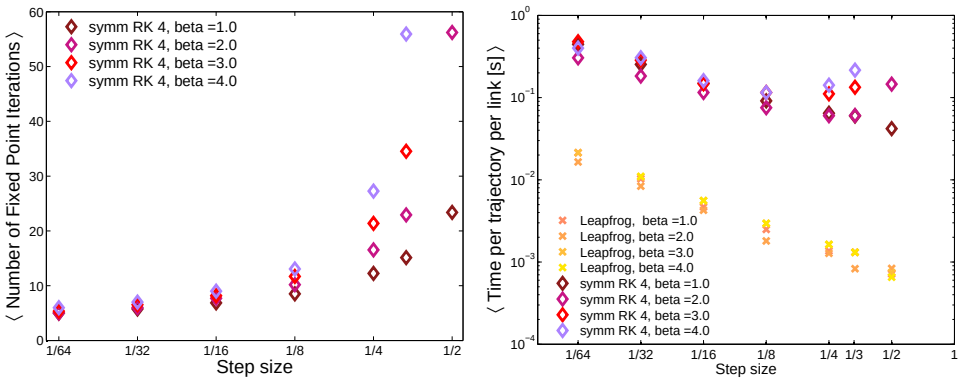

Figure 4.13: Left: The number of fixed point iterations for different values of β. Right: The computational time increases with β.

Conclusion and Outlook

The main result of this work is the insight that implicit partitioned Runge-Kutta methods can be used during the simulations of gauge theories. It can be mentioned, that symmetric implicit partitioned Runge-Kutta methods have an even convergence order provided that the model error is small enough. Due to the fact that a convergence order of 4 is reached in the simulations, it has been shown that a convergence order greater than 2 can in fact be attained.
Furthermore, the investigated implicit partitioned Runge-Kutta method of convergence order 4 is stable, which allows to use larger step sizes. So far, the complexity of the integration scheme described in lemma 3.21 prevents the method from being more efficient than the Leapfrog integration scheme. Though, it needs to be said, that the main objective of this thesis has been the investigation of the feasibility of higher-order integration schemes besides the Leapfrog and splitting methods. Thus, there has been no attempt in optimizing this method until now.

The symplecticity, i.e volume-preservation is not considered in this thesis. Although the expectation value of the energy chage along a trajectory $\exp(-\Delta\mathcal{H})$ serves as a measure for volume-preservation, it should be shown numerically by means of the Jacobian of the field configurations.
As a next step, the time consumption of the implicit partitioned Runge-Kutta method should be examined. The fixed point iteration depends on matrix computations so far. It can be improved, e.g using a base of Pauli matrices. With this ansatz, only the coefficients of the Pauli base would have to be computed. Furthermore, the stopping criterion used in the fixed point iteration can be considered as well as the method coefficients. Probably, the coefficients can be chosen in a way that the expensive parts have to be evaluated less often. Another possibility to save computational time is the change of the increment

$$L_j = ig\Big(\Big[\exp\Big(W_j\Big)\,\exp\Big(\frac{1}{2}\Omega_1\Big)\,U_0\Big]\Big).$$

Because of the two matrix exponential functions, its evaluation is very costly.

Another idea is the exchange of the used mapping $U(t) = \exp(\Omega(t)U_0)$ given in equation (3.9) and its appropriate differential equation $\dot{\Omega}(t) = \exp_\Omega^{-1}$ (see equation (3.3)) with another one, e.g. the Cayley map. In doing so, the exponential function and the truncation of the derivative of the inverse exponential map would be replaced. Since the computation of the matrix exponential function

is time-consuming, this could be advantageous. Another point is that the Taylor series used to determine the convergence orders depend on the truncation of the model and have to be computed for each convergence order separately. The derivative of the inverse Cayley map is not truncated, such that higher convergence orders can be derived from lower ones.
As a general remark, it can be said that the implicit partitioned Runge-Kutta method does not depend on the size of its used matrices. Thus, the scheme can be applied for any differential equations with result in a Lie group.

Appendix A

Theory

Definition A.1 (Haar measure)**.** *For any compact group G the Haar measure is the unique measure dU on G which is*

- *normalized:*
$$\int_G f(U)dU = 1,$$
- *invariant:*
$$\int_G f(U)dU = \int_G f(VU)dU = \int_G f(UV)dU \quad \forall V \in G.$$

Definition A.2 (Lie Algebra)**.** *A Lie algebra $\mathfrak{g}$ is a vector space over a field $\mathbb{F}$ with composition*

$$[\quad,\quad] : \mathfrak{g} \times \mathfrak{g} \longrightarrow \mathfrak{g}, \qquad (x, y) \longmapsto [x, y]$$

which is called Lie bracket or commutator and satisfies the following conditions:

L1 *Antisymmetry*
$$[x, y] = -[y, x] \qquad \forall x, y \in \mathfrak{g}$$

L2 *Bilinearity*
$$[\alpha x + \beta y, z] = \alpha[x, z] + \beta[y, z]$$
$$[x, \alpha y + \beta z] = \alpha[x, y] + \beta[x, z]$$
$\forall x, y, z \in \mathfrak{g} \quad$ *and* $\quad \alpha, \beta \in \mathbb{F}$

L3 *Jacobi identity*
$$\Big[x, [y, z]\Big] + \Big[y, [z, x]\Big] + \Big[z, [x, y]\Big] = 0 \qquad \forall x, y, z \in \mathfrak{g}$$

Remark A.3. *We will consider the general linear Lie algebra $\mathfrak{g}$ with commutator*
$$[A, B] = AB - BA \qquad \forall A, B \in \mathfrak{g}.$$

Definition A.4 (Lie Group)**.** *A Lie group G is a group which is also a differentiable manifold. The group operation is a differentiable mapping $G \times G \to G$.*

Definition A.5 (Tangent space of a matrix Lie group)**.** *Let G be a matrix Lie group. The tangent space at the point U of this Lie group is defined as*

$$T_U G := \{A_U U | A_U \in \mathfrak{g}\}.$$

Remark A.6 (Connection between Lie group and Lie algebra)**.** *Since the Lie group is a differentiable manifold, it has a tangent space in every point. The tangent space at the identity*

$$T_I G := \{A_I I | A_I \in \mathfrak{g}\}.$$

has the structure of a Lie algebra.

Definition A.7 (Exponetial map)**.** *The exponential map*

$$\exp : \mathfrak{g} \to G, \qquad A \mapsto \exp(A) = \sum_{k=0}^{\infty} \frac{A^k}{k!} \tag{A.1}$$

with $A \in \mathfrak{g}$ and $\exp(A) \in G$ maps an element of the Lie algebra $\mathfrak{g}$ onto an element of the Lie group G.

Theorem A.8. *The determinant of a Lie group and the trace of a Lie algebra are connected via*

$$\det\Big(\exp(A)\Big) = \exp\Big(tr\Big(A\Big)\Big).$$

Remark A.9. *For square matrices U, V, W it holds*

$$tr\Big(U+V\Big) = tr\Big(U\Big) + tr\Big(V\Big) \tag{A.2}$$

$$tr\Big(UV\Big) = tr\Big(VU\Big) \tag{A.3}$$

and for unitary matrices U

$$\mathrm{Re}\Big(tr\Big(U\Big)\Big) = \frac{1}{2} tr\Big(U + U^\dagger\Big) \tag{A.4}$$

Proof. The first two equations are obvious, and equation (A.4) follows from

$$\begin{aligned} tr\Big(U + U^\dagger\Big) &= tr\Big(\Big(\mathrm{Re}(U) + \mathrm{Im}(U) + \mathrm{Re}(U) - \mathrm{Im}(U)\Big)\Big) \\ &= tr\Big(2 \cdot \mathrm{Re}(U)\Big) = 2 \cdot tr\Big(\mathrm{Re}(U)\Big) \end{aligned}$$

□

Appendix B

Calculations

B.1 Check of the symmetry and the convergence order 2

Remark B.1. *The coefficiens given in table 3.1 fulfill the symmetry conditions* (3.12) *and the order conditions* (3.27)-(3.30) *for convergence order* 2.

Proof. The order conditions will be checked with the coefficients from the Butcher tableaus denoted in table 3.1:

$$(3.27): \sum_{j=1}^{s=2} b_j = b_1 + b_2 = \frac{1}{2} + \frac{1}{2} = 1$$

$$(3.28): \sum_{j=1}^{s=2} \hat{b}_j = \hat{b}_1 + \hat{b}_2 = \frac{1}{2} + \frac{1}{2} = 1$$

$$(3.29): \sum_{j=1}^{s=2} \Big(b_j \sum_{k=1}^{s=2} \hat{a}_{jk} \Big) = b_1 \cdot (\hat{a}_{11}\hat{a}_{12}) + b_2 \cdot (\hat{a}_{21}\hat{a}_{22}) = \frac{1}{2} \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{2}$$

$$(3.30): \sum_{j=1}^{s=2} \Big(\hat{b}_j \sum_{k=1}^{s=2} a_{jk} \Big) = \hat{b}_1 \cdot (a_{11} + a_{12}) + \hat{b}_2 \cdot (a_{21} + a_{22}) = \frac{1}{2} \cdot 0 + \frac{1}{2} \cdot 1 = 0 + \frac{1}{2} = \frac{1}{2}$$

This means the partitioned Runge-Kutta method (3.26) with coefficients given in table 3.1 fulfills convergence order two. Considering the symmetry conditions

$$\begin{aligned} b_1 &= \frac{1}{2} = b_2 & \qquad \hat{b}_1 &= \frac{1}{2} = \hat{b}_2 \\ \hat{a}_{1,1} &= \hat{b}_2 - \hat{a}_{22} = \frac{1}{2} - 0 = \frac{1}{2} & \qquad a_{1,1} &= b_2 - a_{22} = \frac{1}{2} - \frac{1}{2} = 0 \\ \hat{a}_{1,2} &= \hat{b}_2 - \hat{a}_{21} = \frac{1}{2} - \frac{1}{2} = 0 & \qquad a_{1,2} &= b_2 - a_{21} = \frac{1}{2} - \frac{1}{2} = 0 \\ \hat{a}_{2,1} &= \hat{b}_1 - \hat{a}_{12} = \frac{1}{2} - 0 = \frac{1}{2} & \qquad a_{2,1} &= b_1 - a_{12} = \frac{1}{2} - 0 = \frac{1}{2} \\ \hat{a}_{2,2} &= \hat{b}_1 - \hat{a}_{11} = \frac{1}{2} - \frac{1}{2} = 0 & \qquad a_{2,2} &= b_1 - a_{11} = \frac{1}{2} - 0 = \frac{1}{2} \end{aligned}$$

mentioned in the system of equations (3.12), the method is also symmetric. □

Bibliography

[1] J. C. Sexton and D. H. Weingarten. Hamiltonian evolution for the hybrid Monte Carlo algorithm. *Nucl. Phys.*, B380:665–678, 1992.

[2] C. Lubich E. Hairer and G. Wanner. *Geometric Numerical Integration.* Springer-Verlag Berlin Heidelberg, 2006.

[3] Christof Gattringer and Christian B. Lang. *Quantum chromodynamics on the lattice, an introductory presentation.* Lecture notes in physics ; 788. Springer, 2010.

[4] Istvan Montvay and Gernot Munster. *Quantum Fields on a Lattice.* Cambridge University Press, 1994.

[5] Thomas DeGrand and Carleton DeTar. *Lattice Methods for Quantum Chromodynamics.* World Scientific, 2006.

[6] B. J. Pendleton S. Duane, A. D. Kennedy and D. Roweth. Hybrid monte carlo. *Physics Letters B*, 195:216–222, 1987.

[7] Carsten Urbach. Untersuchung der Reversibilitätsverletzung beim Hybrid-Monthe-Carlo-Algorithmus. Master's thesis, Freie Universitaet Berlin, 2002.