Fakultät für Mathematik und Naturwissenschaften
Fachgruppe Mathematik und Informatik



BERGISCHE
UNIVERSITÄT
WUPPERTAL

Hessian-Free Force-Gradient Integrators, Hierarchical Splitting Methods, and Their Application to Lattice QCD

Dissertation

zur Erlangung des akademischen Grades
Doktor der Naturwissenschaften (Dr. rer. nat.)

vorgelegt von
Kevin Schäfers

Juli 2026

Betreuer: Prof. Dr. Michael Günther

Prüfungstermin:	17. November 2026
Gutachter:	Prof. Dr. Michael Günther Prof. Dr. Nicole Marheineke Prof. Dr. Michael Peardon
Prüfungskommission:	Prof. Dr. Michael Günther Prof. Dr. Nicole Marheineke Prof. Dr. Michael Peardon Prof. Dr. Francesco Knechtli Prof. Dr. Andreas Bartel Dr. Jacob Finkenrath

Acknowledgments

First of all, I would like to thank Prof. Dr. Michael Günther for supervising this PhD thesis. Thank you, Michael, for introducing me to GARK methods after I submitted my Bachelor's thesis on multirate integration. Your ODE course, taken in parallel, awakened my interest in geometric numerical integration. With you already supervising my Master's thesis on numerical integration schemes for lattice QCD simulations, you ideally prepared me for my doctoral studies. I am very grateful that you gave me the opportunity to pursue a PhD in an application that was entirely new to me, while enabling me to combine my two primary research interests: geometric numerical integration and multirate integration. I enjoyed the many discussions we had on the wide range of open problems we set out to solve, and I hope that the results presented in this thesis demonstrate that our discussions have been fruitful.

Next, I would like to thank Prof. Dr. Matthias Ehrhardt. Your courses "Einführung in die Numerik" and "Einführung in die Numerik gewöhnlicher Differentialgleichungen" sparked my interest in numerical analysis. Moreover, you invited me to the ECMI modelling week 2019 in Darmstadt, which was one of the reasons I chose to write my Bachelor's thesis in numerical mathematics under the supervision of Dr. Christoph Hachtel and Prof. Dr. Andreas Bartel. You were the starting point of my motivation to pursue a PhD. Thank you very much, Matthias!

Moreover, I want to thank Prof. Dr. Francesco Knechtli and Dr. Jacob Finkenrath for their guidance through the fundamentals of lattice QCD simulations. You made it possible to ideally bring my expertise in numerical analysis to the research project, resulting in our papers dealing with Hessian-free force-gradient integrators. I really enjoyed working with you on these papers and I hope that these will not be the last problems we tackle together.

I also thank all colleagues from the research unit FOR5269 for the nice atmosphere and very fruitful discussions during the past four years. I am very happy that the extension of the research unit has been granted and that a new generation of PhD students can pursue a PhD in such a great working atmosphere that I was able to experience. At this point, I also want to express my gratitude to the German Research Foundation (DFG) for funding the research unit and, with it, my PhD position. The financial support allowed me to fully focus on my research and to visit many exciting conferences and workshops over the past years.

Many thanks to Prof. Dr. Andreas Bartel for always having an open door for discussions. Starting from my Bachelor's thesis through to my last article during my PhD journey on hierarchical splitting methods, you were always available for all kinds of questions.

Thanks to all my former and present colleagues of the working group Applied and Computational Mathematics. Working with you as part of the ACM group was and is a pure joy! Our coffee breaks, spent discussing everything apart from research, always provided the ideal reset to return to my research problems with fresh eyes. And of course I specifically enjoyed our intense football discussions (although I still cannot understand how one can support another club from the Ruhrgebiet apart from VfL Bochum). For all of you who were not too involved in football and had to listen to the almost daily discussions: I'm sorry!

Last but not least, I would like to thank my family and in particular my wife Miri for the love and support I received throughout the years. Our joint walks with our wonderful dog Sansa were the ideal breaks during the writing of this thesis that helped me to clear my mind and return to my desk with renewed energy.

Contents

1. Introduction **1**
1.1. Literature Review and Contribution 1
1.2. Outline 4

I. Hessian-Free Force-Gradient Integrators and Hierarchical Splitting Methods **5**

2. Geometric Numerical Integration **7**
2.1. Motivation: The Mathematical Pendulum 7
2.2. Classifications of Dynamical Systems 8
2.3. Hamiltonian Systems 9
2.3.1. Properties of Hamiltonian Systems 9
2.3.2. Numerical Integration of the Mathematical Pendulum 11

3. Splitting Methods **15**
3.1. Splitting Idea and Elementary Examples 15
3.2. Raising the Order of Splitting Methods via Composition 17
3.3. Order Conditions via the Baker–Campbell–Hausdorff Formula 19
3.3.1. Calculus of Lie Derivatives 19
3.3.2. Gröbner's Lemma and the Baker–Campbell–Hausdorff Formula . 20
3.3.3. Order Conditions for Splitting Methods 21
3.4. Splitting Methods of RKN Type 22
3.4.1. Splitting for Separable Hamiltonian Systems 22
3.4.2. Geometric Properties 23
3.4.3. Order Conditions 25
3.4.4. Numerical Stability 26

4. Hessian-Free Force-Gradient Integrators **27**
4.1. Motivation: The Force-Gradient Approach 27
4.2. Hessian-Free Force-Gradient Integrators 30
4.2.1. Derivation of Hessian-Free Counterparts 30
4.2.2. Order Conditions 32
4.2.3. Geometric Properties 34
4.3. Linear Stability Analysis 35
4.4. Detection of Promising Integrator Variants 38
4.4.1. An Accuracy-Related Efficiency Measure 38
4.4.2. The Relative Stability Threshold 40
4.4.3. A Set of Non-Dominated Integrator Variants 42
4.5. Numerical Results 48
4.6. Summary 49

5. A Hierarchical Splitting Approach for N-Split Systems **51**
5.1. Splitting Methods for N-split Systems 51
5.2. Hierarchical Splitting Methods 52
5.2.1. Iterative Splitting of the Vector Field 52
5.2.2. Convergence Order and Accuracy 54
5.2.3. Composition Methods 57

5.3. Numerical Results 59
5.4. Summary 60

6. Multirate Hierarchical Splitting Methods **61**
6.1. Motivation: Dynamical Systems with Multirate Potential 61
6.2. Multirate Hierarchical Splitting Methods 63
6.3. The Concept of Computational Order 66
6.4. Numerical Results 68
6.5. Summary 69

7. Hierarchical Force-Gradient Integrators **71**
7.1. Splitting Trees for Separable Hamiltonian Systems 71
7.2. Hierarchical (Hessian-Free) Force-Gradient Integrators 72
7.3. Numerical Results 73
7.4. Summary 75

II. HMC Simulations in Lattice QCD **77**

8. The Hybrid Monte Carlo Algorithm **79**
8.1. Monte Carlo Methods 79
8.2. Markov Chain Monte Carlo Methods 80
8.3. The Hybrid Monte Carlo Method 81
8.4. Tuning the HMC Algorithm 86

9. Gauge Field Generation in Lattice QCD **89**
9.1. Quantum Fields on a Lattice 89
9.1.1. The Wilson Gauge Action 90
9.1.2. Handling Fermions on the Lattice 90
9.2. The HMC Algorithm in Lattice QCD 93
9.2.1. Hamiltonian Equations of Motion 94
9.2.2. The Gauge Force and the Fermion Force 94
9.2.3. The Multirate Potential in Lattice QCD 95
9.2.4. Even-Odd Reduction 96
9.2.5. Hasenbusch Mass Preconditioning 97
9.3. Splitting Methods for Lattice QCD 98
9.3.1. Splitting Methods 98
9.3.2. Force-Gradient Integrators 99
9.3.3. Hessian-Free Force-Gradient Integrators 99
9.4. Tuning of HMC Simulations in Lattice QCD 100

10. A Modified Cayley Transform for SU(3) **101**
10.1. The Cayley Transform 102
10.2. Derivation of the Modified Cayley Transform 103
10.3. Approximation of the Link Updates 107
10.4. Numerical Results 112
10.5. Summary 114

11. Numerical Results **115**
11.1. The Two-Dimensional Schwinger Model 115
11.1.1. Comparison of the Minimum-Error Variants 116
11.1.2. Comparison of the Numerical Stability 117
11.2. Simulations with Two Heavy Wilson Fermions 117

11.3. Simulations with Two Twisted-Mass Fermions 121
11.4. Multirate Simulations with Two Wilson Fermions 125
11.5. Summary 127

12. Conclusion and Outlook **129**
12.1. Conclusion 129
12.1.1. Hessian-Free Force-Gradient Integrators 129
12.1.2. Hierarchical Splitting Methods 129
12.1.3. Hierarchical Hessian-Free Force-Gradient Integrators 130
12.1.4. Modified Cayley Transform for SU(3) 130
12.2. Outlook 131
12.2.1. Classification of Sixth-Order Hessian-Free Force-Gradient Integrators 131
12.2.2. Modified Cayley Transform in Large-Scale Simulations with Fermions 131
12.2.3. Extensions of the Hierarchical Splitting Approach 131
12.2.4. Hierarchical Splitting Approaches for the RMHMC Algorithm . . 132
12.2.5. Further Application Domains of the Hierarchical Splitting Approach 132

A. Minimum-Error Variants **133**

List of Figures

2.1. Mathematical pendulum. 7
2.2. Energy drift of the Euler methods. 8
2.3. Time-reversibility of the Hamiltonian flow. 11
2.4. Energy conservation of discrete gradient methods. 12
2.5. The symplectic two-form under transformations. 13
2.6. Time-reversibility of numerical integration schemes. 13

3.1. The splitting idea. 16
3.2. The Lie–Trotter and the Strang splitting. 16
3.3. Composition techniques. 18
3.4. Modified Hamiltonian. 25

4.1. Application of a force-gradient integrator to the mathematical pendulum. 29
4.2. Comparison of a force-gradient integrator and its Hessian-free counterpart. 32
4.3. Efficiency measures for `ABACABA` and `ABADABA`. 42
4.4. Efficiency measures for `BACABACAB` and `BADABADAB`. 45
4.5. Efficiency measures for `BABAB` and `ABABA`. 46
4.6. Illustration of the outer solar system. 48
4.7. Convergence order of Hessian-free force-gradient integrators applied to the outer solar system. 48
4.8. Work-precision diagram comparing different Hessian-free force-gradient integrators applied to the outer solar system. 49
4.9. Relative energy error of a Hessian-free force-gradient integrator applied to the outer solar system. 50

5.1. A splitting tree for $N = 5$ partitions. 53
5.2. Splitting tree of the Lie–Trotter and the Strang splitting for N-split systems. 54
5.3. Splitting tree for a damped and forced harmonic oscillator. 60
5.4. Numerical results for hierarchical splitting methods applied to a damped and forced harmonic oscillator. 60

6.1. Force magnitudes in a four body-problem with multirate potential. . . . 62
6.2. Multirate splitting tree of a generalized impulse method for N-split systems. 63
6.3. Multirate splitting tree for a four body-problem with multirate potential. 68
6.4. Numerical results for multirate hierarchical splitting methods applied to a four body-problem with multirate potential. 70

7.1. Splitting tree for separable Hamiltonian systems. 71
7.2. Work-precision diagram for hierarchical (Hessian-free) force-gradient integrators applied to a four body-problem with multirate potential. 74
7.3. Energy behavior of a hierarchical Hessian-free force-gradient integrator applied to a four body-problem with multirate potential. 74

8.1. HMC simulations for a bivariate Gaussian distribution. 86

9.1. Link matrices. 89
9.2. Plaquette. 90

10.1. The modified Cayley phase. 106

10.2. Convergence order of Cayley-based integrator variants. 112
10.3. Variance of $\Delta\mathcal{H}$ vs. number of force evaluations per unit trajectory of Cayley-based integrators and their exponential-based counterparts. . . . 113
10.4. Acceptance rate vs. number of force evaluations per unit trajectory of Cayley-based integrators and their exponential-based counterparts. . . . 113

11.1. Comparison of the accuracy of force-gradient integrators and their Hessian-free counterparts in the two-dimensional Schwinger model. 118
11.2. Comparison of the numerical stability of `BACAB` and `BADAB` in the two-dimensional Schwinger model. 119
11.3. Lattice QCD simulations with two heavy Wilson fermions: variance of $\Delta\mathcal{H}$ vs. force evaluations per trajectory for all non-dominated integrator variants. 120
11.4. Twisted-mass simulations: numerical stability comparison of the minimum-error and maximum-stability variants of `BABAB`. 123
11.5. Twisted-mass simulations: numerical stability comparison of the minimum-error and maximum-stability variants of `ABADABA`. 124

Chapter 1.

Introduction

In many branches of physics, the behavior of complex systems is modeled by systems of ordinary differential equations (ODEs). Often, these systems can be modeled as Hamiltonian systems [140]. The associated vector fields possess intrinsic geometric properties, such as symplecticity, volume preservation, and energy conservation, which are essential for the correct long-time behavior of the underlying physical system. Preserving these properties numerically requires dedicated structure-preserving numerical integration schemes. This motivates the field of *geometric numerical integration* [61].

In addition to their geometric structure, many systems of ODEs exhibit a natural decomposition into a sum of $N \geq 2$ partitions. Such partitions may arise from differences in nonlinearity, stiffness, dynamical behavior, evaluation cost, or geometric properties of the individual subsystems. Efficient numerical integration schemes exploit this structure by applying dedicated numerical methods to the individual partitions.

A prominent example of such systems arises when performing gauge field generation in lattice quantum chromodynamics (QCD) [33, 77] using the hybrid Monte Carlo (HMC) algorithm [38]. Gauge field generation is a highly computationally demanding task: the production of a single ensemble can require several weeks of runtime on large-scale high-performance computing (HPC) systems. Consequently, the development of numerical integration schemes tailored specifically to this application not only enables the generation of more ensembles or larger statistics within a fixed computational budget, but also helps to reduce the associated energy consumption.

In the molecular dynamics step of the HMC algorithm, we must solve Hamiltonian equations of motion while preserving crucial geometric properties of the Hamiltonian flow, particularly the *volume preservation* and the *time-reversibility*. Furthermore, the potential energy of the Hamiltonian can often be decomposed into several components, where the computationally expensive parts of the right-hand side vary slowly enough (or are weakly coupled enough) that evaluating them at a coarser rate does not compromise the accuracy of the overall scheme. Numerical methods that evaluate the expensive components less frequently can therefore lead to substantial performance gains. This idea is realized by employing different step sizes for the integration of the individual components of the right-hand side in so-called *multirate methods* [59] or *multiple time-stepping techniques* [137, 9]. Therefore, lattice QCD simulations provide a particularly challenging and practically relevant application in the intersection of geometric numerical integration and multirate integration.

1.1. Literature Review and Contribution

Geometric numerical integration focuses on numerical methods that respect the underlying geometric and structural properties of differential equations. It has been shown that numerical approximations preserving qualitative features of the exact flow exhibit significantly improved long-time behavior, even when their local truncation error is comparable to or larger than those of conventional numerical integration schemes [61].

In the context of Hamiltonian systems, the dynamics of particle accelerators motivated the construction of the first symplectic integrators [107]. Since then, a wide variety of geometric numerical integration schemes for Hamiltonian systems has been developed.

These approaches encompass Runge–Kutta-based methods, including additive Runge–Kutta (ARK) methods [3] and generalized additive Runge–Kutta (GARK) methods [60], variational integrators [89], and splitting methods [94, 18]. Splitting methods have become particularly popular for separable Hamiltonian systems of the form $\mathcal{H}(\boldsymbol{p},\boldsymbol{q}) = \mathcal{T}(\boldsymbol{p})+\mathcal{V}(\boldsymbol{q})$ with kinetic energy $\mathcal{T}(\boldsymbol{p})$ and potential energy $\mathcal{V}(\boldsymbol{q})$. The well-known Verlet method [141] can be interpreted as the application of the Strang splitting [128] to such systems. Most existing work on splitting methods focuses on systems where the right-hand side is split into two subsystems (two-split systems). For general N-split systems, splitting methods are commonly constructed using the Lie–Trotter [136] and the Strang splitting [128], as well as their compositions, see for example [93]. More sophisticated splitting methods for N-split systems can be derived using a hierarchical splitting approach that we proposed in

[118] K. S. and M. Günther: *A hierarchical splitting approach for N-split differential equations*, arXiv preprint arXiv:2601.12878 (2026).

This approach applies splitting methods for two-split systems iteratively to obtain an overall splitting method for N-split systems.

If the Hamiltonian exhibits the finer structure

$$\mathcal{H}(\boldsymbol{p},\boldsymbol{q}) = \mathcal{T}(\boldsymbol{p}) + \mathcal{V}(\boldsymbol{q}), \quad \mathcal{T}(\boldsymbol{p}) = \tfrac{1}{2}\boldsymbol{p}^\top \boldsymbol{M}^{-1}\boldsymbol{p}, \tag{1.1}$$

with a constant, symmetric, and positive definite mass matrix $\boldsymbol{M}$, then conventional splitting methods can be further enhanced by incorporating commutators into the integration process [132]. This commutator-based splitting approach, known as splitting methods with modified potentials [11] or force-gradient integrators [29, 101, 76], has shown to provide more accurate integrator variants, particularly for higher convergence orders. However, the evaluation of commutators typically requires access to Hessian information of the potential, which is often computationally expensive or unavailable in large-scale simulation codes. To address this limitation, approximate update steps that avoid explicit commutator evaluations have been introduced [144, 145]. Our novel class of *Hessian-free force-gradient integrators* introduced in

[115] K. S., J. Finkenrath, M. Günther, F. Knechtli: *Hessian-free force-gradient integrators*, Computer Physics Communications 309 (2025), p. 10947,

generalizes this idea to the entire class of force-gradient integrators. A refined error analysis revealed additional order conditions arising from the approximation. Based on this order theory, we performed a complete classification of Hessian-free force-gradient integrators, yielding more than 40 distinct integrator variants. All variants were implemented in the open-source software package `openQCD`, with the extended version publicly available at

[114] K. S. and J. Finkenrath: *openQCD_force-gradient*, Version v1.0.2, Nov. 2024, GitHub repository.

The results presented in the proceeding

[116] K. S., J. Finkenrath, M. Günther, F. Knechtli: *Hessian-free force-gradient integrators and their application to lattice QCD simulations*, PoS LATTICE2024 (2025), 025,

indicated that numerical stability plays a crucial role in determining the efficiency of such integration schemes when applied to lattice QCD simulations. This observation motivated a detailed linear stability analysis of force-gradient integrators and their Hessian-free variants in

[117] K. S., J. Finkenrath, M. Günther, F. Knechtli: *Numerical stability of force-gradient integrators and their Hessian-free variants in lattice QCD simulations*, Computer Physics Communications 321 (2026), p. 110034.

The stability analysis enabled the identification of integrator variants that achieve a good trade-off between accuracy and numerical stability, thereby permitting larger step sizes and improved overall performance.

Numerical methods that employ different step sizes for different components are the subject of *multirate integration* [59]. The origins of multirate integration date back to [106] and were later developed for multistep methods [51]. These foundations led to inter/extrapolation-based multirate methods [6, 113]. Instead of using information derived by inter/extrapolation, internal stages can be used for coupling the integration schemes for the subsystems. This alternative idea resulted in multirate extensions of partitioned Runge–Kutta (PRK) [57] and GARK methods [58].

However, most existing multirate approaches do not explicitly account for geometric structure preservation. Within the GARK framework, we derived a subclass of symplectic and symmetric multirate GARK methods [119]. In parallel, multirate variational integrators have been proposed in [99]. Another approach combining geometric numerical integration and multirate integration is within the framework of splitting methods, yielding symmetric and symplectic multiple time-stepping schemes. Such methods, often referred to as *nested integration schemes*, are widely used in molecular dynamics simulations [137, 9] and also constitute common choices for gauge field generation in lattice QCD simulations [123, 138]. A classical example is the *impulse method* [61, Chapter VIII.4.1], which can be regarded as a multirate extension of the Strang splitting. Building on our hierarchical splitting framework [118], we introduced a class of multirate hierarchical splitting methods for N-split systems that include nested integration schemes as special cases. Therefore, the framework provides a rigorous error analysis for nested integrators. Moreover, the framework proposes a novel reweighting technique for the multirate factors that reduces computational overhead.

The Hamiltonian systems arising when performing gauge field generation in lattice QCD via the HMC algorithm are characterized by a phase space that is the cotangent bundle $T^*\mathcal{G}$ over the direct product $\mathcal{G} = \mathrm{SU}(3)^n$ of $n \in \mathbb{N}$ copies of the special unitary group SU(3). This introduces an additional requirement for the numerical integration scheme: the numerical approximations must be situated on the manifold to avoid non-physical values. Lie group methods [98, 32, 27] achieve this by using the matrix exponential to map elements of the corresponding Lie algebra $\mathfrak{su}(3)$ into the Lie group SU(3). There exist symmetric [148], as well as symplectic [19] Lie group methods. Moreover, there are known examples which are both symmetric and symplectic, see for instance the generalized midpoint rule presented in [109]. Although the exponential map is the natural choice, the use of alternative local parameterizations is possible. For quadratic Lie groups, the Cayley transform [54] is a prominent choice to replace the exponential map, resulting in Cayley-based Lie group methods [34, 35, 81, 69]. However, the Cayley transform does not define a suitable mapping from $\mathfrak{su}(3)$ to SU(3). In

[120] K. S., M. Peardon, M. Günther: *A modified Cayley transform for SU(3) molecular dynamics simulations*, Computer Physics Communications 315 (2025), p. 109740,

we derived a modified Cayley transform for SU(3) that is an alternative local parameterization. Splitting methods based on this modified Cayley transform show beneficial numerical stability properties compared to their exponential-based counterparts.

1.2. Outline

This thesis primarily builds on the publications [115, 116, 117, 118, 120] on Hessian-free force-gradient integrators, (multirate) hierarchical splitting methods, and a modified Cayley transform for SU(3). The combination of Hessian-free force-gradient integrators and hierarchical splitting methods yields a novel class of *(multirate) hierarchical Hessian-free force-gradient integrators*, which are then used to perform lattice QCD simulations.

The thesis is structured in two main parts. The first part is devoted to the derivation, analysis, and classification of the numerical integration schemes. Motivated by applications in lattice QCD, particular emphasis is given to the numerical integration of separable Hamiltonian systems with fine structure (1.1).

In Chapter 2, we will start with a concise introduction to geometric numerical integration while particularly focusing on Hamiltonian systems. Subsequently, Chapter 3 discusses splitting methods for two-split ODE systems. Besides a general introduction, special attention to splitting methods of RKN type is given. Chapter 4 introduces the novel class of *Hessian-free force-gradient integrators*, which extends splitting methods of RKN type by a commutator-based splitting approach while not requiring the Hessian of the potential energy. We will perform a thorough error analysis, discuss the geometric properties, and conduct a linear stability analysis. We perform a complete classification of self-adjoint Hessian-free force-gradient integrators with up to six stages and detect particularly efficient integrator variants that excel in both accuracy and numerical stability. In Chapter 5 and Chapter 6, we consider general N-split dynamical ODE systems and propose a novel class of *hierarchical splitting methods*. This subclass of splitting methods for N-split systems employs splitting methods for two-split systems in a hierarchical manner, thereby facilitating the derivation of higher-order splitting methods. Moreover, we extend the hierarchical splitting approach by a multiple time stepping approach. We conduct a thorough error analysis and introduce the concept of computational order for multiple time stepping methods. In Chapter 7, we combine the concepts from the preceding sections to obtain a novel class of *hierarchical (Hessian-free) force-gradient integrators* – a class of multiple time stepping techniques for Hamiltonian systems (1.1) where the potential energy is split into $N-1$ parts ($N \geq 2$).

The second part focuses on the application of the developed numerical methods for gauge field generation in lattice QCD. In Chapter 8, the Hybrid/Hamiltonian Monte Carlo (HMC) algorithm is discussed. In the molecular dynamics (MD) step of this Markov Chain Monte Carlo (MCMC) method, a Hamiltonian system of the form (1.1) must be integrated numerically. We will discuss the demands on the numerical integration scheme and explain why the class of Hessian-free force-gradient integrators is well-suited. Subsequently, in Chapter 9, we will discuss how the HMC algorithm is used for gauge field generation in lattice QCD. We will focus on the potential energy of the Hamiltonian system and demonstrate that it exhibits multirate potential such that numerical integration schemes can benefit from splitting the potential into multiple components. This motivates the use of hierarchical Hessian-free force-gradient integrators for lattice QCD simulations with the HMC algorithm. Chapter 10 presents a modified Cayley transform for SU(3) that can be utilized to numerically approximate the link updates required in splitting methods and (Hessian-free) force-gradient integrators. Different numerical tests are conducted in Chapter 11, demonstrating the superior efficiency of (hierarchical) Hessian-free force-gradient integrators compared to conventional splitting methods and nested integration techniques.

The thesis concludes with a summary of the main contributions and an outlook on future research directions.

Part I.

Hessian-Free Force-Gradient Integrators and Hierarchical Splitting Methods

Chapter 2.

Geometric Numerical Integration

In this chapter, we provide a concise introduction to *geometric numerical integration*. We will concentrate on the fundamental concepts that are essential for the understanding of this thesis, particularly focusing on the geometric numerical integration of Hamiltonian systems. For a comprehensive introduction to geometric numerical integration, we refer to the textbook [61] and the references therein.

2.1. Motivation: The Mathematical Pendulum

As an initial example, we consider the *mathematical pendulum*, characterized by a mass m, a massless rod of length ℓ, and gravitational acceleration g, excluding friction. The mathematical pendulum is illustrated in Fig. 2.1. We choose $m = 1$, $\ell = 1$, $g = 1$, and obtain the system of ODEs

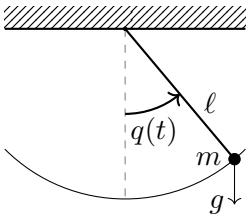


Fig. 2.1. Mathematical pendulum.

$$\begin{pmatrix} \dot{p} \\ \dot{q} \end{pmatrix} = \begin{pmatrix} -\sin(q) \\ p \end{pmatrix}. \tag{2.1}$$

The energy $\mathcal{H}(p, q) = \frac{1}{2}p^2 + 1 - \cos(q)$ of the system (in appropriate units) remains constant over time, as it holds

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{H}(p, q) = p \cdot \dot{p} + \sin(q) \cdot \dot{q} = -p\sin(q) + \sin(q)p = 0. \tag{2.2}$$

Long-term simulations using classical methods such as the Euler methods exhibit secular energy drift, rendering them impractical for capturing the correct qualitative behavior. Specifically, the explicit Euler method

$$\begin{pmatrix} p_1 \\ q_1 \end{pmatrix} = \begin{pmatrix} p_0 \\ q_0 \end{pmatrix} + h \begin{pmatrix} -\sin(q_0) \\ p_0 \end{pmatrix} \tag{2.3}$$

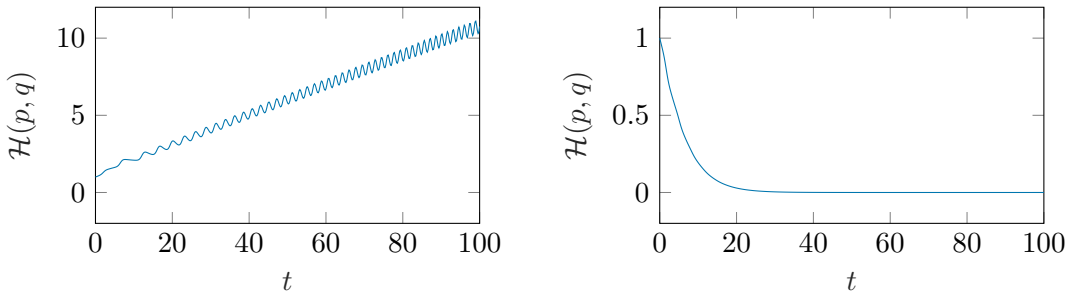
yields numerical approximations with increasing energy, as illustrated in Fig. 2.2a. In contrast, the implicit Euler method

$$\begin{pmatrix} p_1 \\ q_1 \end{pmatrix} = \begin{pmatrix} p_0 \\ q_0 \end{pmatrix} + h \begin{pmatrix} -\sin(q_1) \\ p_1 \end{pmatrix} \tag{2.4}$$

provides numerical approximations with decreasing energy, ultimately converging to the stable equilibrium $(p, q) = (0, 0)$ with energy $\mathcal{H}(0, 0) = 0$, as depicted in Fig. 2.2b.

Consequently, obtaining physically meaningful approximations with these methods requires exceedingly small step sizes h, resulting in substantial computational cost.

This challenge motivates geometric numerical integration, which addresses the development of numerical integration schemes that preserve the structural properties of the



(a) Explicit Euler method (2.3). **(b)** Implicit Euler method (2.4).

Fig. 2.2. Mathematical pendulum. Global error in the total energy on the time interval $[0, 100]$ for the Euler methods (2.3) and (2.4). The numerical simulations were performed with step size $h = 0.2$ and initial value $(p_0, q_0) = (0, \pi/2)$.

underlying systems exactly, i.e., up to machine precision. When an appropriate geometric numerical integration scheme is employed, the admissible step size h is governed solely by accuracy requirements and potential numerical instabilities of the method.

In this chapter, we will begin by presenting a classification of dynamical systems as outlined in [91]. Subsequently, we examine Hamiltonian systems in greater detail and discuss numerical time integration techniques specifically designed for this prominent class of dynamical systems.

2.2. Classifications of Dynamical Systems

We consider a general dynamical ODE system of the form

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}), \tag{2.5}$$

with phase space $\mathcal{M}$, solution $\boldsymbol{x}(t) \in \mathcal{M}$, and right-hand side $\boldsymbol{f}(\boldsymbol{x}) \in \mathfrak{X}(\mathcal{M})$, where $\mathfrak{X}(\mathcal{M})$ denotes the space of smooth vector fields on $\mathcal{M}$. Given an initial value $\boldsymbol{x}_0$ at time $t_0 = 0$, we denote by $\boldsymbol{\varphi}_t(\boldsymbol{x}_0) \coloneqq \boldsymbol{x}(t; \boldsymbol{x}_0)$ the exact flow (i.e., the exact solution) of the ODE (2.5) with initial value $\boldsymbol{x}(0) = \boldsymbol{x}_0$. Following [91], we can perform a primary classification of dynamical systems into three categories:

1. flows which lie in a group of diffeomorphisms[1] (e.g., the set of energy-conserving maps $\{\boldsymbol{\varphi} \colon \mathcal{H}(\boldsymbol{\varphi}(\boldsymbol{x})) = \mathcal{H}(\boldsymbol{x})\}$);
2. flows which lie in a semigroup (e.g., the set of all dissipative maps $\boldsymbol{\varphi} \colon \mathcal{M} \to \mathcal{M}$, where $\mathcal{M}$ is the phase space); and
3. flows which lie in a symmetric space (e.g., the set of all maps $\{\boldsymbol{\varphi} \colon \boldsymbol{\varphi} \circ \boldsymbol{\rho} \circ \boldsymbol{\varphi} = \boldsymbol{\rho}\}$ with involution $\boldsymbol{\rho}$, i.e., $\boldsymbol{\rho}^2 = \mathrm{Id}$).

It is also possible that a flow can be categorized into multiple categories simultaneously, as we will see, for instance, in the subsequent section on Hamiltonian systems.

Geometric numerical integration is relevant to all three categories. Each category imposes different constraints on the numerical integration scheme to guarantee that the numerical approximations exhibit the correct qualitative behavior. For instance, the absence of an inverse element in a semigroup can present challenges when a numerical integration scheme involves fractional time steps going backward in time [64].

As we will focus on the development of numerical integration schemes for Hamiltonian systems, we will investigate their intrinsic geometric properties in detail.

[1] A diffeomorphism on a manifold $\mathcal{M}$ is a mapping $\boldsymbol{\varphi} \colon \mathcal{M} \to \mathcal{M}$ which is differentiable and has a differentiable inverse.

2.3. Hamiltonian Systems

Hamiltonian systems have become fundamental in various branches of physics, including gauge field generation in lattice QCD [77], which will be discussed in Part II of this thesis. A Hamiltonian system with $d \in \mathbb{N}$ degrees of freedom is a dynamical system characterized by a twice continuously differentiable *Hamiltonian*

$$\mathcal{H}\colon \mathcal{M} \subset \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}, \quad (\boldsymbol{p}, \boldsymbol{q}) \mapsto \mathcal{H}(\boldsymbol{p}, \boldsymbol{q}).$$

In the phase space $\mathcal{M}$, the coordinates $(\boldsymbol{p}, \boldsymbol{q})$ represent the *generalized coordinates* $\boldsymbol{q} = (q_1, \dots, q_d)^\top$ and *conjugate momenta* $\boldsymbol{p} = (p_1, \dots, p_d)^\top$. The Hamiltonian equations of motion can be expressed in terms of the Poisson bracket.

Definition 2.1 (Poisson bracket [140])**.** The (canonical) Poisson bracket of two (scalar) smooth functions $f(\boldsymbol{p}, \boldsymbol{q})$ and $g(\boldsymbol{p}, \boldsymbol{q})$ reads

$$\{f, g\} = \sum_{i=1}^{d} \left(\frac{\partial g}{\partial p_i} \frac{\partial f}{\partial q_i} - \frac{\partial g}{\partial q_i} \frac{\partial f}{\partial p_i} \right). \tag{2.6}$$

The *Hamiltonian vector field* is defined as

$$\hat{\mathcal{H}} = \{\cdot, \mathcal{H}\} = \sum_{i=1}^{d} \left(\frac{\partial \mathcal{H}}{\partial p_i} \frac{\partial}{\partial q_i} - \frac{\partial \mathcal{H}}{\partial q_i} \frac{\partial}{\partial p_i} \right),$$

which gives rise to the *equations of motion*

$$\begin{pmatrix} \dot{\boldsymbol{p}} \\ \dot{\boldsymbol{q}} \end{pmatrix} = \begin{pmatrix} \{\boldsymbol{p}, \mathcal{H}\} \\ \{\boldsymbol{q}, \mathcal{H}\} \end{pmatrix} = \begin{pmatrix} -\frac{\partial \mathcal{H}}{\partial \boldsymbol{q}} \\ \frac{\partial \mathcal{H}}{\partial \boldsymbol{p}} \end{pmatrix}. \tag{2.7}$$

The equations of motion can be expressed more compactly as

$$\dot{\boldsymbol{x}} = \boldsymbol{J}^{-1} \nabla \mathcal{H}(\boldsymbol{x}), \tag{2.8}$$

where $\boldsymbol{x} = (\boldsymbol{p}, \boldsymbol{q})$ and

$$\boldsymbol{J}^{-1} = \begin{pmatrix} \boldsymbol{0}_{d\times d} & -\boldsymbol{I}_{d\times d} \\ \boldsymbol{I}_{d\times d} & \boldsymbol{0}_{d\times d} \end{pmatrix} \in \mathbb{R}^{2d\times 2d}.$$

2.3.1. Properties of Hamiltonian Systems

Hamiltonian vector fields possess several intrinsic geometric characteristics that we aim to preserve at a discrete level to obtain numerical approximations that exhibit the correct qualitative behavior.

Energy conservation. The Hamiltonian $\mathcal{H}$ itself is an invariant of the system, i.e., it holds

$$\frac{\mathrm{d}}{\mathrm{d}t} \mathcal{H}(\boldsymbol{x}) = \nabla \mathcal{H}(\boldsymbol{x})^\top \boldsymbol{J}^{-1} \nabla \mathcal{H}(\boldsymbol{x}) = 0, \tag{2.9}$$

due to the skew-symmetry of the structure matrix $\boldsymbol{J}^{-1}$. In particular, the set of all energy-conserving maps $\boldsymbol{\varphi}\colon \mathcal{M} \to \mathcal{M}$ is a group of diffeomorphisms that is closed under composition,

$$\mathfrak{G} = \{\boldsymbol{\varphi}\colon \boldsymbol{\varphi}^* \mathcal{H} = \mathcal{H}\},$$

where $\boldsymbol{\varphi}^* \mathcal{H}$ denotes the *pull-back* of $\mathcal{H}$ by $\boldsymbol{\varphi}$ and is defined by $(\boldsymbol{\varphi}^* \mathcal{H})(\boldsymbol{x}) = \mathcal{H}(\boldsymbol{\varphi}(\boldsymbol{x}))$.

Symplecticity. Another important property of Hamiltonian systems is the *symplecticity* of the flow. We consider the (canonical) *symplectic two-form*

$$\omega\colon \mathbb{R}^{2d} \times \mathbb{R}^{2d} \to \mathbb{R}, \quad \omega(\boldsymbol{\xi}, \boldsymbol{\eta}) = \boldsymbol{\xi}^\top \boldsymbol{J} \boldsymbol{\eta}, \quad \text{with} \quad \boldsymbol{J} = \begin{pmatrix} \boldsymbol{0}_{d\times d} & \boldsymbol{I}_{d\times d} \\ -\boldsymbol{I}_{d\times d} & \boldsymbol{0}_{d\times d} \end{pmatrix}.$$

Definition 2.2 (Symplectic mapping [61])**.** A differentiable mapping $\boldsymbol{g}\colon \mathcal{M} \to \mathbb{R}^{2d}$ is called *symplectic* if the Jacobian matrix $\boldsymbol{g}'(\boldsymbol{p},\boldsymbol{q})$ is symplectic everywhere, i.e., if

$$\boldsymbol{g}'(\boldsymbol{p},\boldsymbol{q})^\top \boldsymbol{J}\boldsymbol{g}'(\boldsymbol{p},\boldsymbol{q}) = \boldsymbol{J} \quad \text{or} \quad \omega(\boldsymbol{g}'(\boldsymbol{p},\boldsymbol{q})\boldsymbol{\xi}, \boldsymbol{g}'(\boldsymbol{p},\boldsymbol{q})\boldsymbol{\eta}) = \omega(\boldsymbol{\xi},\boldsymbol{\eta}).$$

Theorem 2.3 (Poincaré [61])**.** *Let $\mathcal{H}(\boldsymbol{p},\boldsymbol{q})$ be a twice continuously differentiable Hamiltonian function on the state space $\mathcal{M} \subset \mathbb{R}^{2d}$. Then, for each fixed t, the exact flow $\boldsymbol{\varphi}_t$ is a symplectic mapping wherever it is defined.*

Proof. For the sake of completeness, we will provide the proof from [61, Theorem VI.2.4]. The sensitivity matrix $\partial\boldsymbol{\varphi}_t/\partial\boldsymbol{x}_0$ is a solution of the variational equation

$$\dot{\boldsymbol{\Psi}} = \boldsymbol{J}^{-1}\nabla^2\mathcal{H}(\boldsymbol{\varphi}_t(\boldsymbol{x}_0))\boldsymbol{\Psi} \tag{2.10}$$

of the Hamiltonian system (2.8). Here, $\nabla^2\mathcal{H}(\boldsymbol{x})$ denotes the symmetric Hessian matrix of the Hamiltonian $\mathcal{H}$. We therefore obtain

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\left[\left(\frac{\partial\boldsymbol{\varphi}_t}{\partial\boldsymbol{x}_0}\right)^\top \boldsymbol{J}\left(\frac{\partial\boldsymbol{\varphi}_t}{\partial\boldsymbol{x}_0}\right)\right] &= \left(\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial\boldsymbol{\varphi}_t}{\partial\boldsymbol{x}_0}\right)^\top \boldsymbol{J}\left(\frac{\partial\boldsymbol{\varphi}_t}{\partial\boldsymbol{x}_0}\right) + \left(\frac{\partial\boldsymbol{\varphi}_t}{\partial\boldsymbol{x}_0}\right)^\top \boldsymbol{J}\left(\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial\boldsymbol{\varphi}_t}{\partial\boldsymbol{x}_0}\right)\\ &= \left(\frac{\partial\boldsymbol{\varphi}_t}{\partial\boldsymbol{x}_0}\right)^\top \nabla^2\mathcal{H}(\boldsymbol{\varphi}_t(\boldsymbol{x}_0))\underbrace{\boldsymbol{J}^{-\top}\boldsymbol{J}}_{=-\boldsymbol{I}}\left(\frac{\partial\boldsymbol{\varphi}_t}{\partial\boldsymbol{x}_0}\right) + \left(\frac{\partial\boldsymbol{\varphi}_t}{\partial\boldsymbol{x}_0}\right)^\top \nabla^2\mathcal{H}(\boldsymbol{\varphi}_t(\boldsymbol{x}_0))\left(\frac{\partial\boldsymbol{\varphi}_t}{\partial\boldsymbol{x}_0}\right) = 0.\end{aligned}$$

Since $\boldsymbol{\varphi}_0$ is the identity map, it obviously holds

$$\left(\frac{\partial\boldsymbol{\varphi}_t}{\partial\boldsymbol{x}_0}\right)^\top \boldsymbol{J}\left(\frac{\partial\boldsymbol{\varphi}_t}{\partial\boldsymbol{x}_0}\right) = \boldsymbol{J} \tag{2.11}$$

for $t = 0$. Consequently, the relation (2.11) is satisfied for all t and $\boldsymbol{x}_0$, as long as the solution remains in the domain of definition of $\mathcal{H}$. □

The set of all symplectic mappings $\boldsymbol{\varphi}\colon \mathcal{M} \to \mathcal{M}$ defines a group of diffeomorphisms,

$$\mathfrak{G} = \{\boldsymbol{\varphi}\colon \boldsymbol{\varphi}^*\omega = \omega\},$$

where $\boldsymbol{\varphi}^*\omega$ denotes the *pull-back* of the symplectic two-form ω by $\boldsymbol{\varphi}$ and is defined by $(\boldsymbol{\varphi}^*\omega)_{\boldsymbol{x}}(\boldsymbol{\xi},\boldsymbol{\eta}) = \omega\left(\boldsymbol{\varphi}'(\boldsymbol{x})\boldsymbol{\xi}, \boldsymbol{\varphi}'(\boldsymbol{x})\boldsymbol{\eta}\right)$.

Volume preservation. A direct consequence of the symplecticity is that the exact flow $\boldsymbol{\varphi}_t$ of the Hamiltonian system preserves the phase space volume [61, Chapter VI.9]: for every bounded open set $\mathcal{U} \subset \mathbb{R}^{2d}$ and for every t for which $\boldsymbol{\varphi}_t(\boldsymbol{x})$ exists for all $\boldsymbol{x} \in \mathcal{U}$,

$$\int_{\mathcal{U}} \mathrm{d}\boldsymbol{x} = \int_{\boldsymbol{\varphi}_t(\mathcal{U})} \mathrm{d}\boldsymbol{x}.$$

According to the transformation formula, we have

$$\int_{\boldsymbol{\varphi}_t(\mathcal{U})} \mathrm{d}\boldsymbol{x} = \int_{\mathcal{U}} \left|\det\left(\frac{\partial\boldsymbol{\varphi}_t}{\partial\boldsymbol{x}_0}\right)\right| \mathrm{d}\boldsymbol{x}_0.$$

Given the symplecticity (2.11) of the Hamiltonian flow, taking the determinant on both sides and utilizing $\det(\boldsymbol{J}) = 1$ immediately yields

$$\left|\det\left(\frac{\partial\boldsymbol{\varphi}_t}{\partial\boldsymbol{x}_0}\right)\right| = 1, \tag{2.12}$$

resulting in the *volume preservation* of the Hamiltonian flow. The set of volume-preserving mappings $\boldsymbol{\varphi}\colon \mathcal{M} \to \mathcal{M}$ also defines a group of diffeomorphisms,

$$\mathfrak{G} = \{\boldsymbol{\varphi}\colon |\det(\boldsymbol{\varphi}'(\boldsymbol{x}))| = 1 \ \forall \boldsymbol{x} \in \mathcal{M}\}.$$

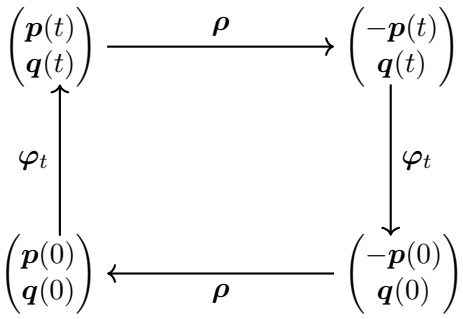


Fig. 2.3. Time-reversibility of the Hamiltonian flow.

Symmetry. Given the group property of the exact flow $\boldsymbol{\varphi}_t$, we have

$$\mathrm{Id} = \boldsymbol{\varphi}_0 = \boldsymbol{\varphi}_{t-t} = \boldsymbol{\varphi}_t \circ \boldsymbol{\varphi}_{-t}.$$

As a result, the Hamiltonian flow is *self-adjoint*,

$$\boldsymbol{\varphi}_t = \boldsymbol{\varphi}_t^* \coloneqq \boldsymbol{\varphi}_{-t}^{-1}. \tag{2.13}$$

Often, the property (2.13) is also referred to as the *symmetry* of the flow.

Time-reversibility. For many Hamiltonian systems, including separable Hamiltonian systems $\mathcal{H}(\boldsymbol{p},\boldsymbol{q}) = \mathcal{T}(\boldsymbol{p}) + \mathcal{V}(\boldsymbol{q})$ with kinetic energy of the form $\mathcal{T}(\boldsymbol{p}) = \frac{1}{2}\boldsymbol{p}^\top \boldsymbol{M}^{-1}\boldsymbol{p}$ (which are of particular interest in this thesis) and potential energy $\mathcal{V}(\boldsymbol{q})$, the Hamiltonian satisfies $\mathcal{H}(\boldsymbol{p},\boldsymbol{q}) = \mathcal{H}(-\boldsymbol{p},\boldsymbol{q})$. This ensures that the equations of motion are invariant under the transformation $(t,\boldsymbol{p},\boldsymbol{q}) \mapsto (-t,-\boldsymbol{p},\boldsymbol{q})$, and provides a natural involution $\boldsymbol{\rho}(\boldsymbol{p},\boldsymbol{q}) = (-\boldsymbol{p},\boldsymbol{q})$ such that the Hamiltonian flow satisfies the *$\boldsymbol{\rho}$-compatibility condition*

$$\boldsymbol{\rho} \circ \boldsymbol{\varphi}_t = \boldsymbol{\varphi}_{-t} \circ \boldsymbol{\rho}. \tag{2.14}$$

Utilizing $\boldsymbol{\rho}^{-1} = \boldsymbol{\rho}$ and the self-adjointness (2.13), we obtain the *time-reversibility*

$$\boldsymbol{\rho} \circ \boldsymbol{\varphi}_t \circ \boldsymbol{\rho} \circ \boldsymbol{\varphi}_t = \mathrm{Id}\,. \tag{2.15}$$

The time-reversibility is illustrated in Fig. 2.3. The set of all time-reversible mappings

$$\mathfrak{S} = \{\boldsymbol{\varphi} \colon \boldsymbol{\varphi} \circ \boldsymbol{\rho} \circ \boldsymbol{\varphi} = \boldsymbol{\rho}\}$$

forms a symmetric space.

Given a numerical integration scheme $\boldsymbol{\Phi}_h$, it preserves the aforementioned geometric properties if, for any step size $h > 0$, the mapping $\boldsymbol{\Phi}_h$ is an element of the corresponding (semi-)group of diffeomorphisms or symmetric space.

2.3.2. Numerical Integration of the Mathematical Pendulum

As an illustration, we highlight the geometric properties of Hamiltonian systems by performing numerical simulations for the mathematical pendulum (2.1).

As already emphasized in Section 2.1, the flow of the mathematical pendulum (2.1) is energy-conserving, see (2.2). The numerical results for the Euler methods (2.3) and (2.4) in Fig. 2.2 demonstrate that the numerical approximations do not exhibit the correct qualitative behavior. For general Hamiltonian systems, discrete gradient methods [55, 23] constitute a class of energy-conserving methods. A discrete gradient method for Hamiltonian systems of the form (2.8) reads

$$\boldsymbol{x}_1 = \boldsymbol{x}_0 + h\boldsymbol{J}^{-1}\overline{\nabla}\mathcal{H}(\boldsymbol{x}_1,\boldsymbol{x}_0), \tag{2.16}$$

where $\overline{\nabla}\mathcal{H}(\hat{\boldsymbol{x}}, \boldsymbol{x})$ is a *discrete gradient*, i.e., a continuous function of $(\hat{\boldsymbol{x}}, \boldsymbol{x})$ satisfying

$$\begin{aligned} \overline{\nabla}\mathcal{H}(\hat{\boldsymbol{x}}, \boldsymbol{x})^\top (\hat{\boldsymbol{x}}, \boldsymbol{x}) &= \mathcal{H}(\hat{\boldsymbol{x}}) - \mathcal{H}(\boldsymbol{x}), \\ \overline{\nabla}\mathcal{H}(\boldsymbol{x}, \boldsymbol{x}) &= \nabla\mathcal{H}(\boldsymbol{x}). \end{aligned} \tag{2.17}$$

In the numerical results presented in Fig. 2.4, the *Gonzalez discrete gradient* [55] is employed as the discrete gradient within (2.16). The numerical results demonstrate that the method preserves the Hamiltonian exactly, i.e., up to round-off errors.

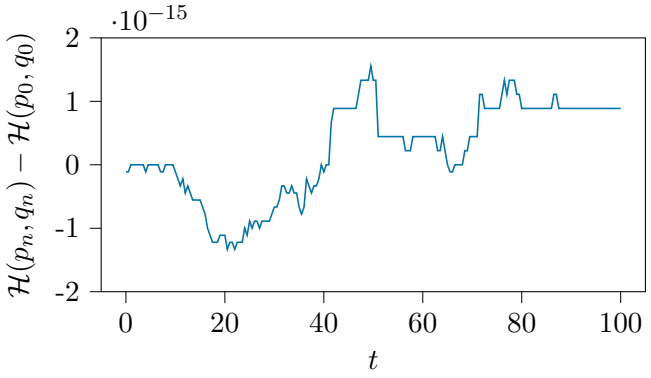


Fig. 2.4. Mathematical pendulum. Error in the total energy on the time interval $[0, 100]$ for the discrete gradient method (2.16) using the Gonzalez discrete gradient [55]. The numerical simulations were performed with step size $h = 0.5$ and initial value $(p_0, q_0) = (0, \pi/2)$.

In contrast to discrete gradient methods, Runge–Kutta (RK) methods can only preserve the energy if the Hamiltonian is quadratic, as symplectic RK methods preserve quadratic invariants [61].

To investigate the symplecticity of the Hamiltonian flow, we examine the application of different numerical integration schemes to the mathematical pendulum (2.1) with large step size $h = 2$ and initial value $(p_0, q_0) = (0, \pi/2)$. For visualization purposes, we consider the vectors $\boldsymbol{\xi} = (1, 0.2)$ and $\boldsymbol{\eta} = (0.3, 0.7)$. The results presented in Fig. 2.5 demonstrate that both the explicit Euler method (2.3) and the implicit Euler method (2.4) are not symplectic. In contrast, the *implicit midpoint rule*

$$\begin{pmatrix} p_1 \\ q_1 \end{pmatrix} = \begin{pmatrix} p_0 \\ q_0 \end{pmatrix} + h \begin{pmatrix} -\sin\left((q_0 + q_1)/2\right) \\ (p_0 + p_1)/2 \end{pmatrix} \tag{2.18}$$

is symplectic and yields

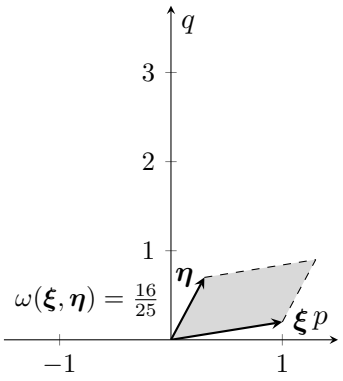
$$\omega\left(\left(\frac{\partial(p_1, q_1)}{\partial(p_0, q_0)}\right)\boldsymbol{\xi}, \left(\frac{\partial(p_1, q_1)}{\partial(p_0, q_0)}\right)\boldsymbol{\eta}\right) = \omega(\boldsymbol{\xi}, \boldsymbol{\eta}) = \frac{16}{25},$$

which visually means that the areas of the parallelograms spanned by $\boldsymbol{\xi}, \boldsymbol{\eta}$ and

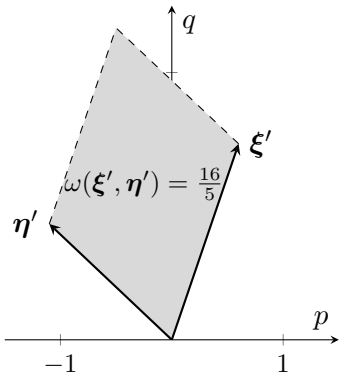
$$\left(\frac{\partial(p_1, q_1)}{\partial(p_0, q_0)}\right)\boldsymbol{\xi}, \ \left(\frac{\partial(p_1, q_1)}{\partial(p_0, q_0)}\right)\boldsymbol{\eta}$$

are equal. To investigate the time-reversibility of the Hamiltonian flow, we compare the explicit Euler method (2.3), the implicit Euler method (2.4), and the implicit midpoint rule (2.18). The numerical results presented in Fig. 2.6 are obtained by computing a single time step of size $h = 1.5$ starting from $(p_0, q_0) = (1, \pi/2)$, demonstrating that the implicit midpoint rule is time-reversible, while both Euler methods fail to preserve this property at a discrete level.

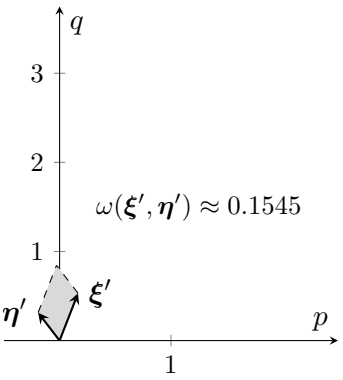
The methods employed so far all belong to the class of Runge–Kutta or discrete gradient methods. Throughout this thesis, we will focus on *splitting methods*, an approach that is fundamentally different from both Runge–Kutta and discrete gradient methods.



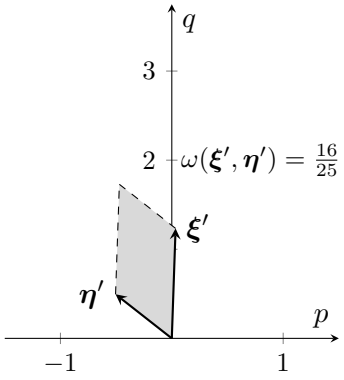
(a) Parallelogram spanned by $\boldsymbol{\xi} = (1, 0.2)$ and $\boldsymbol{\eta} = (0.3, 0.7)$.



(b) Transformation via the explicit Euler method (2.3).

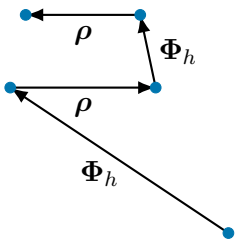


(c) Transformation via the implicit Euler method (2.4).

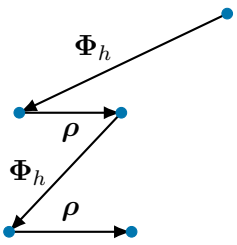


(d) Symplectic transformation via the implicit midpoint rule (2.18).

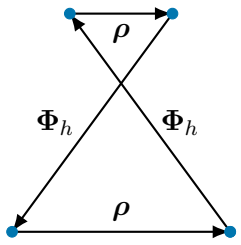
Fig. 2.5. Mathematical pendulum. Transformation of the parallelogram spanned by the vectors $\boldsymbol{\xi}, \boldsymbol{\eta}$ using different numerical integration schemes with step size $h = 2$. The new vectors are defined as $\boldsymbol{\xi}' = (\partial(p_1, q_1)/\partial(p_0, q_0))\boldsymbol{\xi}$ and $\boldsymbol{\eta}' = (\partial(p_1, q_1)/\partial(p_0, q_0))\boldsymbol{\eta}$.



(a) Explicit Euler method (2.3).



(b) Implicit Euler (2.4).



(c) Implicit midpoint rule (2.18).

Fig. 2.6. Mathematical pendulum. Investigation of the time-reversibility (2.15) for the explicit Euler method (2.3), the implicit Euler method (2.4), and the implicit midpoint rule (2.18). All simulations utilize a step size of $h = 1.5$ and start from the initial value $(p_0, q_0) = (1, \pi/2)$.

Chapter 3.

Splitting Methods

In many technical applications, ODE-IVPs of *N-split systems* of the form

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) = \sum_{m=1}^{N} \boldsymbol{f}^{\{m\}}(\boldsymbol{x}), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0 \in \mathbb{R}^n, \tag{3.1}$$

arise. In (3.1), the right-hand side is split additively into $N \geq 2$ partitions based on, for instance, stiffness, nonlinearity, dimension, dynamical behavior, and computational cost. In such cases, it is advantageous to integrate the individual partitions separately using dedicated numerical methods. Ideally, the flows of the individual subsystems can be computed explicitly.

For the sake of simplicity, we will consider the special case of $N = 2$ partitions in this chapter, giving rise to *two-split systems* of the form

$$\dot{\boldsymbol{x}} = \boldsymbol{f}^{\{1\}}(\boldsymbol{x}) + \boldsymbol{f}^{\{2\}}(\boldsymbol{x}), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0 \in \mathbb{R}^n, \tag{3.2}$$

and suppose that the exact flows $\boldsymbol{\varphi}_t^{\{1\}}$ and $\boldsymbol{\varphi}_t^{\{2\}}$ of the subsystems $\dot{\boldsymbol{x}} = \boldsymbol{f}^{\{1\}}(\boldsymbol{x})$ and $\dot{\boldsymbol{x}} = \boldsymbol{f}^{\{2\}}(\boldsymbol{x})$ can be computed explicitly[1].

3.1. Splitting Idea and Elementary Examples

The fundamental concept of splitting methods is to compute a numerical approximation to the exact flow $\boldsymbol{\varphi}_t$, i.e., the exact solution of the two-split system (3.2) by composing evaluations of the flows $\boldsymbol{\varphi}_t^{\{1\}}$ and $\boldsymbol{\varphi}_t^{\{2\}}$ of the individual subsystems. For instance, applying the flow of the first subsystem to $\boldsymbol{x}_0$ yields $\boldsymbol{x}_{1/2} \coloneqq \boldsymbol{\varphi}_h^{\{1\}}(\boldsymbol{x}_0)$, and subsequently applying the flow of the second subsystem yields $\boldsymbol{x}_1 \coloneqq \boldsymbol{\varphi}_h^{\{2\}}(\boldsymbol{x}_{1/2}) = (\boldsymbol{\varphi}_h^{\{2\}} \circ \boldsymbol{\varphi}_h^{\{1\}})(\boldsymbol{x}_0)$. This gives rise to a first consistent splitting method

$$\boldsymbol{\Phi}_h^{[LT]} = \boldsymbol{\varphi}_h^{\{2\}} \circ \boldsymbol{\varphi}_h^{\{1\}}, \tag{3.3}$$

which is commonly referred to as the *Lie–Trotter splitting* [136]. By rearranging the order of the flow evaluations, we obtain the adjoint of the Lie–Trotter splitting (3.3),

$$(\boldsymbol{\Phi}_h^{[LT]})^* = \boldsymbol{\varphi}_h^{\{1\}} \circ \boldsymbol{\varphi}_h^{\{2\}}. \tag{3.4}$$

A common approach to designing self-adjoint numerical integration schemes is to consider the composition of a method $\boldsymbol{\Phi}_h$ and its adjoint $\boldsymbol{\Phi}_h^*$ [111]. By applying this approach to the Lie–Trotter splitting (3.3) and its adjoint (3.4) with a step size of $h/2$, and utilizing the fact that $\boldsymbol{\varphi}_{h/2}^{\{2\}} \circ \boldsymbol{\varphi}_{h/2}^{\{2\}} = \boldsymbol{\varphi}_h^{\{2\}}$, we obtain the numerical method

$$\boldsymbol{\Phi}_h^{[S]} = \boldsymbol{\varphi}_{h/2}^{\{1\}} \circ \boldsymbol{\varphi}_h^{\{2\}} \circ \boldsymbol{\varphi}_{h/2}^{\{1\}}, \tag{3.5}$$

which is known as the *Strang splitting* [128]. Due to its inherent symmetry, the Strang splitting has convergence order $p = 2$.

[1] The general case of $N \geq 2$ partitions and potentially numerical flow approximations of the individual subsystems will be discussed in detail in Chapter 5.

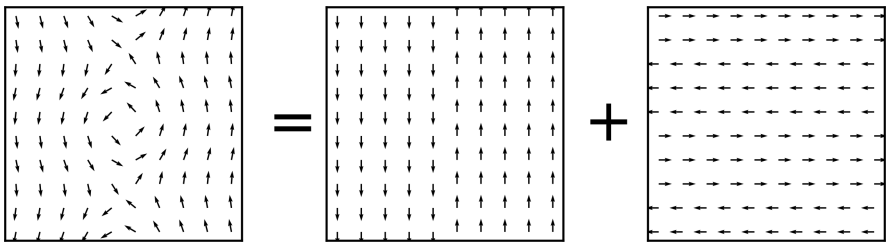


Fig. 3.1. Visualization of the splitting (3.6) for the mathematical pendulum (2.1).

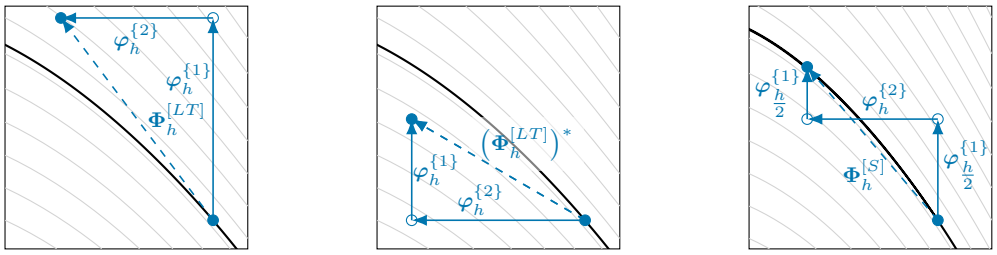
Example 3.1 (Mathematical pendulum)**.** We consider the splitting of the vector field describing the mathematical pendulum (2.1) into its kinetic and potential parts, resulting in the two-split system

$$\dot{\boldsymbol{x}} = \begin{pmatrix} \dot{p} \\ \dot{q} \end{pmatrix} = \underbrace{\begin{pmatrix} 0 \\ p \end{pmatrix}}_{=\boldsymbol{f}^{\{1\}}(\boldsymbol{x})} + \underbrace{\begin{pmatrix} -\sin(q) \\ 0 \end{pmatrix}}_{=\boldsymbol{f}^{\{2\}}(\boldsymbol{x})}, \tag{3.6}$$

where $\boldsymbol{x} = (p, q)^\top$. The splitting (3.6) of the vector field is visualized in Fig. 3.1. The exact flows of the two subsystems $\dot{\boldsymbol{x}} = \boldsymbol{f}^{\{1\}}(\boldsymbol{x})$ and $\dot{\boldsymbol{x}} = \boldsymbol{f}^{\{2\}}(\boldsymbol{x})$ can be computed explicitly and are given by

$$\boldsymbol{\varphi}_t^{\{1\}} \begin{pmatrix} p_0 \\ q_0 \end{pmatrix} = \begin{pmatrix} p_0 \\ q_0 + t p_0 \end{pmatrix} \quad \text{and} \quad \boldsymbol{\varphi}_t^{\{2\}} \begin{pmatrix} p_0 \\ q_0 \end{pmatrix} = \begin{pmatrix} p_0 - t \sin(q_0) \\ q_0 \end{pmatrix}, \tag{3.7}$$

respectively. In Fig. 3.2, a single time step of size $h = 0.5$ has been computed employing the Lie–Trotter splitting (3.3), its adjoint (3.4), and the Strang splitting (3.5). In addition to the superior accuracy of the Strang splitting, the numerical results demonstrate that the ordering of the flow evaluations can have a substantial impact on the numerical approximation.



(a) Lie–Trotter splitting (3.3). **(b)** Adjoint Lie–Trotter (3.4). **(c)** Strang splitting (3.5).

Fig. 3.2. Mathematical pendulum. Application of elementary splitting methods with step size $h = 0.5$ starting from the initial value $(p_0, q_0) = (1, \pi/2)$.

We can further generalize the concept of splitting to define an s-stage splitting method for two-split systems.

Definition 3.2 (s-stage splitting method)**.** A single time step of step size h of an s-stage splitting method $\boldsymbol{\Phi}_h$ advances the given initial value $\boldsymbol{x}(0) = \boldsymbol{x}_0$ to a numerical approximation $\boldsymbol{x}_1 \coloneqq \boldsymbol{\Phi}_h(\boldsymbol{x}_0) \approx \boldsymbol{x}(h)$ and is given as

$$\boldsymbol{\Phi}_h = \boldsymbol{\varphi}_{b_s h}^{\{2\}} \circ \boldsymbol{\varphi}_{a_s h}^{\{1\}} \circ \ldots \circ \boldsymbol{\varphi}_{b_1 h}^{\{2\}} \circ \boldsymbol{\varphi}_{a_1 h}^{\{1\}}, \tag{3.8}$$

where $a_i, b_i \in \mathbb{R}$ denote the integrator coefficients of the splitting method.

Definition 3.3 (Adjoint method)**.** The adjoint of a splitting method (3.8) is defined as

$$\boldsymbol{\Phi}_h^* = \boldsymbol{\varphi}_{a_1 h}^{\{1\}} \circ \boldsymbol{\varphi}_{b_1 h}^{\{2\}} \circ \ldots \circ \boldsymbol{\varphi}_{a_s h}^{\{1\}} \circ \boldsymbol{\varphi}_{b_s h}^{\{2\}}. \tag{3.9}$$

The adjoint method (3.9) exhibits the same order of convergence p as the splitting method (3.8) [61, Theorem II.3.2]. Of particular interest are splitting methods that are identical to their adjoint.

Definition 3.4 (Self-adjoint splitting methods)**.** A splitting method $\boldsymbol{\Phi}_h$ is *self-adjoint*, provided that $\boldsymbol{\Phi}_h = \boldsymbol{\Phi}_h^*$ holds. This is the case if either

$$a_1 = 0, \qquad a_{i+1} = a_{s+1-i}, \qquad \text{and} \qquad b_i = b_{s+1-i}, \tag{3.10a}$$

or

$$b_s = 0, \qquad b_i = b_{s-i}, \qquad \text{and} \qquad a_i = a_{s+1-i}, \tag{3.10b}$$

holds for $i = 1, \ldots, \lfloor \frac{s}{2} \rfloor$. We then also call the splitting method *symmetric*.

By [61, Theorem II.3.2], a self-adjoint method has an even convergence order p. By Taylor expansion, we find that a splitting method (3.8) is consistent, provided that the integrator coefficients satisfy $\sum_i a_i = \sum_i b_i = 1$. Furthermore, if a consistent splitting method is self-adjoint, it inherently possesses a convergence order of $p \geq 2$. For designing higher-order splitting methods, there exist several approaches. One possibility is to consider composition techniques.

3.2. Raising the Order of Splitting Methods via Composition

The core idea of composition techniques for deriving higher-order numerical methods is to compose multiple applications of a base method $\boldsymbol{\Phi}_{\gamma_i h}$ (that in principle can be any one-step method) with different step sizes $\gamma_i h$, $i = 1, \ldots, s$, resulting in *composition schemes* of the form

$$\boldsymbol{\Psi}_h = \boldsymbol{\Phi}_{\gamma_s h} \circ \ldots \circ \boldsymbol{\Phi}_{\gamma_1 h}. \tag{3.11}$$

The composition approach was developed primarily in [131, 146, 92]. The main objective is to increase the order while simultaneously preserving desirable properties of the underlying base method, such as symplecticity or symmetry.

Theorem 3.5. *Let $\boldsymbol{\Phi}_h$ be a one-step method of order p. If the coefficients $\gamma_1, \ldots, \gamma_s \in \mathbb{R}$ satisfy*

$$\sum_{i=1}^{s} \gamma_i = 1, \quad \sum_{i=1}^{s} \gamma_i^{p+1} = 0, \tag{3.12}$$

then the composition method $\boldsymbol{\Psi}_h$ defined by (3.11) *is at least of order $p+1$.*

Proof. See [61, Theorem II.4.1]. □

Remark 3.6. The first condition in (3.12) guarantees the consistency of the composition scheme (3.11), while the second condition eliminates the leading-order error term of the base scheme, resulting in a method of higher order. The system (3.12) has no real-valued solution for odd p. Consequently, the base method $\boldsymbol{\Phi}_h$ must have an even convergence order, such as the second-order Strang splitting (3.5). Furthermore, any real-valued solution of (3.12) necessarily has at least one coefficient $\gamma_i < 0$. Consequently, all composition schemes (3.11) that attain a higher convergence order than their base method have negative fractional time steps.

Example 3.7 (Triple jump composition [146])**.** The smallest value of s that enables a real-valued solution to the system (3.12) is $s = 3$. If we impose symmetry of the weights ($\gamma_1 = \gamma_3$) and let $\boldsymbol{\Phi}_h$ denote a self-adjoint base method of order p, then the composition method (3.11) with weights

$$\gamma_1 = \gamma_3 = \frac{1}{2 - 2^{1/(p+1)}}, \quad \gamma_2 = 1 - 2\gamma_1, \tag{3.13}$$

has convergence order $p + 2$.

The composition weights in (3.13) satisfy $|\gamma_i| > 1$ such that sub-steps leave the integration interval $[t_0, t_0 + h]$, as illustrated in Fig. 3.3a. As the error expansion of the composition method (3.11) involves powers of the weights γ_i, the triple jump composition (3.13) is characterized by large truncation errors. We are therefore interested in solutions of (3.12) that satisfy $|\gamma_i| < 1$, thereby reducing the magnitude of higher powers. This objective can be achieved, for instance, by employing the following approach with $s = 5$ applications.

Example 3.8 (Suzuki's fractals [131])**.** For $s = 5$, a well-known composition technique imposing a symmetric structure with $\gamma_1 = \gamma_2 = \gamma_4 = \gamma_5$ leads to the composition method (3.11) with symmetric base method $\boldsymbol{\Phi}_h$ of even order p and weights

$$\gamma_1 = \gamma_2 = \gamma_4 = \gamma_5 = \frac{1}{4 - 4^{1/(p+1)}}, \quad \gamma_3 = 1 - 4\gamma_1. \tag{3.14}$$

The resulting composition scheme has order $p + 2$.

In contrast to the triple jump composition (3.13), the sub-steps of Suzuki's fractals (3.14) stay inside the integration interval $[t_0, t_0 + h]$, see Fig. 3.3b. Both approaches can be iteratively applied to construct methods of arbitrarily high convergence order. However, both approaches share the drawback that the number of applications of the underlying base scheme increases exponentially with the desired order of the composition scheme.



(a) Triple jump (3.13). **(b)** Suzuki's fractals (3.14).

Fig. 3.3. Illustration of the time stepping procedure of composition techniques for a base method $\boldsymbol{\Phi}_h$ of order $p = 2$.

Many works have focused on the derivation of optimized composition schemes, particularly symmetric compositions of symmetric methods, see e.g. [146, 130, 92, 73, 125] for composition schemes up to order $p = 10$. In a more general perspective, we can extend the composition scheme (3.11) to the form

$$\boldsymbol{\Psi}_h = \boldsymbol{\Phi}_{\beta_s h} \circ \boldsymbol{\Phi}^*_{\alpha_s h} \circ \ldots \circ \boldsymbol{\Phi}_{\beta_1 h} \circ \boldsymbol{\Phi}^*_{\alpha_1 h}, \tag{3.15}$$

such that the composition does not only involve applications of the base scheme $\boldsymbol{\Phi}_h$, but also its adjoint $\boldsymbol{\Phi}^*_h$. In this more general setting, the conditions (3.12) become [61]

$$\sum_{i=1}^{s} \alpha_i + \beta_i = 1, \quad \sum_{i=1}^{s} \beta_i^{p+1} + (-1)^p \alpha_i^{p+1} = 0. \tag{3.16}$$

In contrast to the system (3.12), the system (3.16) also has real-valued solutions for odd p. Consequently, first-order methods, such as the Lie–Trotter splitting (3.3), can be employed as the underlying base method.

Example 3.9 (Ruth's method [107])**.** Ruth's third-order method can be expressed in terms of the composition (3.16) with the Lie–Trotter splitting (3.3) as the base method, $s = 3$ stages, and coefficients

$$\alpha_1 = \frac{3}{8}, \quad \alpha_2 = -\frac{25}{24}, \quad \alpha_3 = 0, \quad \beta_1 = \frac{7}{24}, \quad \beta_2 \quad = \frac{3}{8}, \quad \beta_3 = 1. \tag{3.17}$$

With the Lie–Trotter splitting (3.3) as the underlying base scheme, the composition (3.15) is equivalent to the general splitting method (3.8) for two-split systems. Given an s-stage composition method (3.16), we can rewrite it as a splitting method (3.8) with $s + 1$ stages, $a_1 = \alpha_1$, $b_{s+1} = 0$, and for $i = 1, \dots, s$,

$$a_{j+1} = \alpha_{j+1} + \beta_j, \quad b_j = \alpha_j + \beta_j, \tag{3.18}$$

(with $\alpha_{s+1} = 0$). Conversely, any consistent splitting method (3.8) with $s + 1$ stages can be expressed as a composition method of the form (3.15) using the Lie–Trotter splitting (3.3) as the underlying base scheme [92]. Although there is a formal equivalence, the optimization procedures for identifying efficient splitting methods may differ [18]. Furthermore, this formal equivalence no longer holds when considering splitting methods for N-split systems with $N > 2$, as we will discuss in more detail in Chapter 5.

Instead of relying on composition approaches, we can derive an order theory for splitting methods of the form (3.8) and obtain explicit order conditions on the coefficients a_i and b_i, that, as previously mentioned, may lead to a different optimization process.

3.3. Order Conditions via the Baker–Campbell–Hausdorff Formula

In this thesis, we derive the order conditions for splitting methods via the Baker–Campbell–Hausdorff (BCH) formula, as outlined in [61, Chapter III.5].

3.3.1. Calculus of Lie Derivatives

We consider a general two-split system of the form (3.2). In the spirit of splitting methods, we introduce the *Lie derivatives*

$$\mathcal{L} = \sum_{i=1}^{n} f_i(\boldsymbol{x}) \frac{\partial}{\partial x_i}, \quad \mathcal{L}^{\{m\}} = \sum_{i=1}^{n} f_i^{\{m\}}(\boldsymbol{x}) \frac{\partial}{\partial x_i}. \tag{3.19}$$

We can formally express the exact flow of the overall system (3.2) as [61]

$$\boldsymbol{\varphi}_t(\boldsymbol{x}_0) = \exp(t\mathcal{L}) \operatorname{Id}(\boldsymbol{x}_0) = \exp\left(t(\mathcal{L}^{\{1\}} + \mathcal{L}^{\{2\}})\right) \operatorname{Id}(\boldsymbol{x}_0), \tag{3.20}$$

and the flows of the subsystems $\dot{\boldsymbol{x}} = \boldsymbol{f}^{\{m\}}(\boldsymbol{x})$ as

$$\boldsymbol{\varphi}_t^{\{m\}}(\boldsymbol{x}_0) = \exp(t\mathcal{L}^{\{m\}}) \operatorname{Id}(\boldsymbol{x}_0). \tag{3.21}$$

Example 3.10. For the mathematical pendulum (2.1) with the splitting (3.6), we obtain the Lie derivatives

$$\mathcal{L} = -\sin(q) \frac{\partial}{\partial p} + p \frac{\partial}{\partial q}, \quad \mathcal{L}^{\{1\}} = p \frac{\partial}{\partial q}, \quad \mathcal{L}^{\{2\}} = -\sin(q) \frac{\partial}{\partial p}. \tag{3.22}$$

As $(\mathcal{L}^{\{m\}})^k(\boldsymbol{x}_0) = 0$ for $k \geq 2$, the series expansion of the exponential yields the flows

$$\begin{aligned} \exp(t\mathcal{L}^{\{1\}})\,\mathrm{Id}\begin{pmatrix} p_0 \\ q_0 \end{pmatrix} &= (\,\mathrm{Id} + t\mathcal{L}^{\{1\}})\begin{pmatrix} p_0 \\ q_0 \end{pmatrix} = \begin{pmatrix} p_0 \\ q_0 \end{pmatrix} + t\begin{pmatrix} 0 \\ p_0 \end{pmatrix}, \\ \exp(t\mathcal{L}^{\{2\}})\,\mathrm{Id}\begin{pmatrix} p_0 \\ q_0 \end{pmatrix} &= (\,\mathrm{Id} + t\mathcal{L}^{\{2\}})\begin{pmatrix} p_0 \\ q_0 \end{pmatrix} = \begin{pmatrix} p_0 \\ q_0 \end{pmatrix} + t\begin{pmatrix} -\sin(q_0) \\ 0 \end{pmatrix}, \end{aligned} \tag{3.23}$$

of the subsystems, which are identical to (3.7), as expected.

3.3.2. Gröbner's Lemma and the Baker–Campbell–Hausdorff Formula

Lemma 3.11 (Gröbner [56])**.** *Let $\boldsymbol{\varphi}_s^{\{1\}}$ and $\boldsymbol{\varphi}_t^{\{2\}}$ denote the flows of the subsystems $\dot{\boldsymbol{x}} = \boldsymbol{f}^{\{1\}}(\boldsymbol{x})$ and $\dot{\boldsymbol{x}} = \boldsymbol{f}^{\{2\}}(\boldsymbol{x})$, respectively. For their composition, it holds that*

$$(\boldsymbol{\varphi}_t^{\{2\}} \circ \boldsymbol{\varphi}_s^{\{1\}})(\boldsymbol{x}_0) = \exp(s\mathcal{L}^{\{1\}})\exp(t\mathcal{L}^{\{2\}})\,\mathrm{Id}(\boldsymbol{x}_0). \tag{3.24}$$

Proof. See [61, Lemma III.5.1]. □

The statement of Lemma 3.11 can be generalized to more than two flows. Consequently, we can formally write a splitting method (3.8) as

$$\boldsymbol{\Phi}_h = \exp(a_1 h\mathcal{L}^{\{1\}})\exp(b_1 h\mathcal{L}^{\{2\}})\cdots\exp(a_s h\mathcal{L}^{\{1\}})\exp(b_s h\mathcal{L}^{\{2\}})\,\mathrm{Id}\,. \tag{3.25}$$

In general, the two Lie derivatives $\mathcal{L}^{\{1\}}$ and $\mathcal{L}^{\{2\}}$ do not commute, i.e., the *Lie bracket*

$$[\mathcal{L}^{\{1\}}, \mathcal{L}^{\{2\}}] \coloneqq \mathcal{L}^{\{1\}}\mathcal{L}^{\{2\}} - \mathcal{L}^{\{2\}}\mathcal{L}^{\{1\}} = \sum_{i=1}^{n}\left(\sum_{j=1}^{n}\left(\frac{\partial f_i^{\{2\}}}{\partial x_j} f_j^{\{1\}} - \frac{\partial f_i^{\{1\}}}{\partial x_j} f_j^{\{2\}}\right)\right)\frac{\partial}{\partial x_i} \tag{3.26}$$

does not vanish, $[\mathcal{L}^{\{1\}}, \mathcal{L}^{\{2\}}] \neq 0$. Note that the Lie bracket (3.26) is once again a linear differential operator, resulting in another vector field. Since the Lie bracket generally does not vanish, the composition (3.24) differs from the exact solution (3.20). Gröbner's lemma suggests the use of the BCH formula to analyze the composition of flows. Let $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{n\times n}$ be two (in general non-commuting) matrices, i.e., $[\boldsymbol{A}, \boldsymbol{B}] \neq 0$. The BCH formula yields a series for a matrix $\boldsymbol{C}(t)$ satisfying

$$\exp(t\boldsymbol{A})\exp(t\boldsymbol{B}) = \exp(\boldsymbol{C}(t)).$$

Since we are frequently interested in the construction of self-adjoint splitting methods, we will consider two different variants of the BCH formula.

Lemma 3.12 (BCH formula)**.** *For two matrices $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{n\times n}$, we have*

$$\exp(t\boldsymbol{A})\exp(t\boldsymbol{B}) = \exp\left(\sum_{k\geq 1} t^k \boldsymbol{C}_k\right), \tag{3.27a}$$

where the terms $\boldsymbol{C}_k$ for $k \leq 5$ read

$$\begin{aligned} \boldsymbol{C}_1 &= \boldsymbol{A} + \boldsymbol{B}, \\ \boldsymbol{C}_2 &= \frac{1}{2}[\boldsymbol{A}, \boldsymbol{B}], \\ \boldsymbol{C}_3 &= \frac{1}{12}[\boldsymbol{A}, [\boldsymbol{A}, \boldsymbol{B}]] - \frac{1}{12}[\boldsymbol{B}, [\boldsymbol{A}, \boldsymbol{B}]], \\ \boldsymbol{C}_4 &= \frac{1}{24}\Big[\boldsymbol{A}, [\boldsymbol{B}, [\boldsymbol{B}, \boldsymbol{A}]]\Big], \\ \boldsymbol{C}_5 &= -\frac{1}{720}\Big[\boldsymbol{A}, [\boldsymbol{A}, [\boldsymbol{A}, [\boldsymbol{A}, \boldsymbol{B}]]]\Big] - \frac{1}{720}\Big[\boldsymbol{B}, [\boldsymbol{B}, [\boldsymbol{B}, [\boldsymbol{B}, \boldsymbol{A}]]]\Big] \\ &\quad + \frac{1}{360}\Big[\boldsymbol{A}, [\boldsymbol{B}, [\boldsymbol{B}, [\boldsymbol{B}, \boldsymbol{A}]]]\Big] + \frac{1}{360}\Big[\boldsymbol{B}, [\boldsymbol{A}, [\boldsymbol{A}, [\boldsymbol{A}, \boldsymbol{B}]]]\Big] \\ &\quad + \frac{1}{120}\Big[\boldsymbol{A}, [\boldsymbol{A}, [\boldsymbol{B}, [\boldsymbol{B}, \boldsymbol{A}]]]\Big] + \frac{1}{120}\Big[\boldsymbol{B}, [\boldsymbol{B}, [\boldsymbol{A}, [\boldsymbol{A}, \boldsymbol{B}]]]\Big]. \end{aligned} \tag{3.27b}$$

Proof. See [61, Lemma III.4.3]. □

The higher-order terms $\boldsymbol{C}_4$ and $\boldsymbol{C}_5$ are obtained by simplifying expressions using the *Jacobi identity*,

$$[\boldsymbol{A},[\boldsymbol{B},\boldsymbol{C}]] + [\boldsymbol{C},[\boldsymbol{A},\boldsymbol{B}]] + [\boldsymbol{B},[\boldsymbol{C},\boldsymbol{A}]] = \boldsymbol{0}, \tag{3.28}$$

which holds for arbitrary matrices $\boldsymbol{A},\boldsymbol{B},\boldsymbol{C} \in \mathbb{R}^{n\times n}$.

Lemma 3.13 (Symmetric BCH formula [61])**.** *For two matrices $\boldsymbol{A},\boldsymbol{B} \in \mathbb{R}^{n\times n}$, we have*

$$\exp\left(\tfrac{t}{2}\boldsymbol{A}\right)\exp(t\boldsymbol{B})\exp\left(\tfrac{t}{2}\boldsymbol{A}\right) = \exp\left(\sum_{k\geq 1} t^{2k-1}\boldsymbol{S}_{2k-1}\right), \tag{3.29a}$$

where the terms $\boldsymbol{S}_{2k-1}$ for $k \leq 3$ read

$$\begin{aligned}
\boldsymbol{S}_1 &= \boldsymbol{A}+\boldsymbol{B},\\
\boldsymbol{S}_3 &= -\frac{1}{24}[\boldsymbol{A},[\boldsymbol{A},\boldsymbol{B}]] - \frac{1}{12}[\boldsymbol{B},[\boldsymbol{A},\boldsymbol{B}]],\\
\boldsymbol{S}_5 &= \frac{7}{5760}\Big[\boldsymbol{A},[\boldsymbol{A},[\boldsymbol{A},[\boldsymbol{A},\boldsymbol{B}]]]\Big] - \frac{1}{720}\Big[\boldsymbol{B},[\boldsymbol{B},[\boldsymbol{B},[\boldsymbol{B},\boldsymbol{A}]]]\Big]\\
&\quad + \frac{1}{360}\Big[\boldsymbol{A},[\boldsymbol{B},[\boldsymbol{B},[\boldsymbol{B},\boldsymbol{A}]]]\Big] + \frac{1}{360}\Big[\boldsymbol{B},[\boldsymbol{A},[\boldsymbol{A},[\boldsymbol{A},\boldsymbol{B}]]]\Big]\\
&\quad - \frac{1}{480}\Big[\boldsymbol{A},[\boldsymbol{A},[\boldsymbol{B},[\boldsymbol{B},\boldsymbol{A}]]]\Big] + \frac{1}{120}\Big[\boldsymbol{B},[\boldsymbol{B},[\boldsymbol{A},[\boldsymbol{A},\boldsymbol{B}]]]\Big].
\end{aligned} \tag{3.29b}$$

Proof. The coefficients are a direct consequence of applying the BCH formula (3.27) twice, see [146]. □

The BCH formula is formulated for matrices. Although the series expansion of the exponential of differential operators is calculated exactly as for matrices, the Lie derivatives $\mathcal{L}^{\{1\}}$ and $\mathcal{L}^{\{2\}}$ are unbounded differential operators, so that the series expansions cannot be expected to converge. Nevertheless, we can formally apply the BCH formula and derive a formal identity, i.e., both Taylor series expansions have exactly the same coefficients.

3.3.3. Order Conditions for Splitting Methods

Based on the formulation (3.25), we can expand the product of exponentials to examine the local truncation error of splitting methods. Following the approach from [146], we can formally apply the (symmetric) BCH formula to express the splitting method (3.25) as a single exponential of a series in powers of h. By comparing the resulting series with $h(\mathcal{L}^{\{1\}} + \mathcal{L}^{\{2\}})$ from the exact flow (3.20), we obtain order conditions for splitting methods. A splitting method $\boldsymbol{\Phi}_h$ is convergent of order p, if

$$\boldsymbol{\Phi}_h = \exp\left(h(\mathcal{L}^{\{1\}} + \mathcal{L}^{\{2\}}) + \mathcal{O}(h^{p+1})\right)\mathrm{Id}\,. \tag{3.30}$$

The order conditions up to order $p = 3$ for splitting methods (3.8), applied to general two-split dynamical systems of the form (3.2), read [14]

$$p = 1: \qquad \sum_{i=1}^{s} a_i = 1, \qquad \sum_{i=1}^{s} b_i = 1, \tag{3.31a}$$

$$p = 2: \qquad \sum_{i=1}^{s} b_i \left(\sum_{j=1}^{i} a_j\right) = \frac{1}{2}, \tag{3.31b}$$

$$p = 3: \qquad \sum_{i=1}^{s-1} b_i \left(\sum_{j=i+1}^{s} a_j \right)^2 = \frac{1}{3}, \qquad \sum_{i=1}^{s} a_i \left(\sum_{j=i}^{s} b_j \right)^2 = \frac{1}{3}. \tag{3.31c}$$

One consequence of the order conditions is that any splitting method (3.8) of order $p > 2$ must have at least one of the a_i and also one of the b_i coefficients being strictly negative [14]. The existence of splitting methods of order $p \geq 3$ with non-negative integrator coefficients is of great interest, for example for the numerical integration of nonreversible systems. A possible solution to overcome this order barrier is to consider splitting methods with complex coefficients having positive real part, see e.g. [26, 64, 15]. In this thesis, however, we will focus on real-valued integrator coefficients.

3.4. Splitting Methods of RKN Type

Although the analysis presented in the preceding section is entirely general, there are important subclasses arising in applications whose fine structure does not necessitate considering all order conditions. Many dynamical systems of practical interest can be modeled as separable Hamiltonian systems of the form

$$\mathcal{H}(\boldsymbol{p}, \boldsymbol{q}) = \mathcal{T}(\boldsymbol{p}) + \mathcal{V}(\boldsymbol{q}) = \frac{1}{2} \boldsymbol{p}^\top \boldsymbol{M}^{-1} \boldsymbol{p} + \mathcal{V}(\boldsymbol{q}), \tag{3.32}$$

where the kinetic energy $\mathcal{T}(\boldsymbol{p})$ is quadratic in the momenta $\boldsymbol{p}$, $\boldsymbol{M}$ is a constant, symmetric, and positive definite mass matrix, and $\mathcal{V}(\boldsymbol{q})$ is the potential energy. The equations of motion (2.7) of a Hamiltonian system (3.32) exhibit the fine structure

$$\begin{pmatrix} \dot{\boldsymbol{p}} \\ \dot{\boldsymbol{q}} \end{pmatrix} = \begin{pmatrix} -\nabla \mathcal{V}(\boldsymbol{q}) \\ \boldsymbol{M}^{-1} \boldsymbol{p} \end{pmatrix}. \tag{3.33}$$

By exploiting this structure, optimized splitting methods tailored to these systems can be designed to enhance accuracy while minimizing computational costs compared to conventional splitting methods for general systems (3.2). As second-order ODEs of the form $\ddot{\boldsymbol{x}} = \boldsymbol{g}(\boldsymbol{x})$, when written as first-order systems, exhibit a structure very similar to (3.33), splitting methods designed for Hamiltonian systems (3.32) are sometimes called *splitting methods of RKN type*[2], see e.g. [10].

3.4.1. Splitting for Separable Hamiltonian Systems

As previously illustrated for the mathematical pendulum in (3.6), we will consider the splitting of the vector field

$$\dot{\boldsymbol{x}} = \begin{pmatrix} \dot{\boldsymbol{p}} \\ \dot{\boldsymbol{q}} \end{pmatrix} = \underbrace{\begin{pmatrix} -\nabla \mathcal{V}(\boldsymbol{q}) \\ \boldsymbol{M}^{-1} \boldsymbol{p} \end{pmatrix}}_{= \boldsymbol{f}(\boldsymbol{x})} = \underbrace{\begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{M}^{-1} \boldsymbol{p} \end{pmatrix}}_{= \boldsymbol{f}^{\{1\}}(\boldsymbol{x})} + \underbrace{\begin{pmatrix} -\nabla \mathcal{V}(\boldsymbol{q}) \\ \boldsymbol{0} \end{pmatrix}}_{= \boldsymbol{f}^{\{2\}}(\boldsymbol{x})}, \tag{3.34}$$

where $\boldsymbol{x} = (\boldsymbol{p}, \boldsymbol{q})$. The exact flows of the two subsystems

$$\begin{pmatrix} \dot{\boldsymbol{p}} \\ \dot{\boldsymbol{q}} \end{pmatrix} = \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{M}^{-1} \boldsymbol{p} \end{pmatrix} \tag{3.35a}$$

[2] Runge–Kutta–Nyström (RKN) methods [63] are methods tailored to second-order ODE systems of the form $\ddot{\boldsymbol{x}} = \boldsymbol{g}(\boldsymbol{x})$.

and

$$\begin{pmatrix} \dot{\boldsymbol{p}} \\ \dot{\boldsymbol{q}} \end{pmatrix} = \begin{pmatrix} -\nabla \mathcal{V}(\boldsymbol{q}) \\ \mathbf{0} \end{pmatrix} \tag{3.35b}$$

can be computed explicitly. They are given by

$$\boldsymbol{\varphi}_t^{\{\mathcal{T}\}} \begin{pmatrix} \boldsymbol{p}_0 \\ \boldsymbol{q}_0 \end{pmatrix} = \begin{pmatrix} \boldsymbol{p}_0 \\ \boldsymbol{q}_0 + t\boldsymbol{M}^{-1}\boldsymbol{p}_0 \end{pmatrix} \tag{3.36a}$$

and

$$\boldsymbol{\varphi}_t^{\{\mathcal{V}\}} \begin{pmatrix} \boldsymbol{p}_0 \\ \boldsymbol{q}_0 \end{pmatrix} = \begin{pmatrix} \boldsymbol{p}_0 - t\nabla \mathcal{V}(\boldsymbol{q}_0) \\ \boldsymbol{q}_0 \end{pmatrix}, \tag{3.36b}$$

respectively. In the definition of the flows, we replaced the indices 1 and 2 with the kinetic and potential energy $\mathcal{T}$ and $\mathcal{V}$ to highlight the connection between the flows and the parts of the Hamiltonian. As part of a splitting method (3.8), we will refer to (3.36a) as a *position update* and to (3.36b) as a *momentum update*. The Lie derivatives of the subsystems (3.35) read

$$\hat{\mathcal{T}} \coloneqq \mathcal{L}^{\{1\}} = \boldsymbol{M}^{-1}\boldsymbol{p}\frac{\partial}{\partial \boldsymbol{q}}, \tag{3.37a}$$

$$\hat{\mathcal{V}} \coloneqq \mathcal{L}^{\{2\}} = -\nabla \mathcal{V}(\boldsymbol{q})\frac{\partial}{\partial \boldsymbol{p}}. \tag{3.37b}$$

To highlight the connection between the Lie derivatives and the kinetic and potential parts, we will write $\hat{\mathcal{T}}$ and $\hat{\mathcal{V}}$ instead of $\mathcal{L}^{\{1\}}$ and $\mathcal{L}^{\{2\}}$, respectively.

3.4.2. Geometric Properties

The two subsystems (3.35) are Hamiltonian systems on their own. In particular, the first subsystem (3.35a) constitutes a Hamiltonian system with Hamiltonian $\mathcal{H}^{\{\mathcal{T}\}}(\boldsymbol{p}, \boldsymbol{q}) = \mathcal{T}(\boldsymbol{p})$, whereas the second subsystem (3.35b) corresponds to the equations of motion of the Hamiltonian $\mathcal{H}^{\{\mathcal{V}\}}(\boldsymbol{p}, \boldsymbol{q}) = \mathcal{V}(\boldsymbol{q})$. Consequently, the splitting (3.34) can be regarded as a splitting of the Hamiltonian function into

$$\mathcal{H}(\boldsymbol{p}, \boldsymbol{q}) = \mathcal{H}^{\{\mathcal{T}\}}(\boldsymbol{p}, \boldsymbol{q}) + \mathcal{H}^{\{\mathcal{V}\}}(\boldsymbol{p}, \boldsymbol{q}) = \mathcal{T}(\boldsymbol{p}) + \mathcal{V}(\boldsymbol{q}). \tag{3.38}$$

Symplecticity and volume preservation. As the two subsystems (3.35) are Hamiltonian, the flows (3.36) of the subsystems are both symplectic (and thus volume-preserving), i.e., both flows satisfy (2.11) and (2.12). As the sets of symplectic and volume-preserving maps are both groups of diffeomorphisms that are closed under composition, any composition of symplectic or volume-preserving maps is again symplectic or volume-preserving. Consequently, splitting methods (3.8) based on the splitting (3.34) are symplectic and volume-preserving.

Time-reversibility. The Hamiltonian splitting (3.38) yields two Hamiltonian systems that satisfy $\mathcal{H}^{\{m\}}(\boldsymbol{p}, \boldsymbol{q}) = \mathcal{H}^{\{m\}}(-\boldsymbol{p}, \boldsymbol{q})$, $m \in \{\mathcal{T}, \mathcal{V}\}$. Consequently, the two flows (3.36) are time-reversible and satisfy (2.15). For the time-reversibility of numerical methods, we have the following result.

Theorem 3.14. *If a numerical method, applied to a time-reversible differential equation, satisfies the $\boldsymbol{\rho}$-compatibility condition* (2.14)*, then the numerical flow $\boldsymbol{\Phi}_h$ is a time-reversible map if and only if $\boldsymbol{\Phi}_h$ is self-adjoint.*

Proof. See [61, Theorem V.1.5]. □

To preserve the time-reversibility numerically, we are thus interested in self-adjoint splitting methods satisfying one of the symmetry relations (3.10). As splitting methods satisfying (3.10a) start and end with a momentum update (3.36b), we will refer to these methods as *velocity versions*. Similarly, we will call splitting methods satisfying (3.10b) *position versions* in the context of Hamiltonian systems. The simplest example of a self-adjoint splitting method is the Strang splitting (3.5).

Energy conservation. Both subsystems are Hamiltonian, implying that the two flows (3.36) are energy-conserving. However, each subsystem has a different conserved Hamiltonian. In particular, the flow (3.36a) of the first subsystem (3.35a) conserves the kinetic energy $\mathcal{H}^{\{\mathcal{T}\}}(\boldsymbol{p},\boldsymbol{q}) = \mathcal{T}(\boldsymbol{p})$, while the potential energy $\mathcal{V}(\boldsymbol{q})$ can change arbitrarily. Conversely, the flow (3.36b) of the second subsystem (3.35b) conserves the potential energy $\mathcal{H}^{\{\mathcal{V}\}}(\boldsymbol{p},\boldsymbol{q}) = \mathcal{V}(\boldsymbol{q})$, while the kinetic energy $\mathcal{T}(\boldsymbol{p})$ can change arbitrarily. Consequently, a composition of these maps does not conserve the Hamiltonian $\mathcal{H}(\boldsymbol{p},\boldsymbol{q}) = \mathcal{H}^{\{\mathcal{T}\}}(\boldsymbol{p},\boldsymbol{q}) + \mathcal{H}^{\{\mathcal{V}\}}(\boldsymbol{p},\boldsymbol{q})$ of the overall system, i.e., splitting methods (3.8) based on the splitting (3.34) are not energy-conserving.

Nevertheless, splitting methods based on the splitting (3.34) exhibit good energy conservation. A backward error analysis reveals that splitting methods based on a Hamiltonian splitting such as (3.38) conserve a nearby *modified Hamiltonian*[3]. The error expansion of splitting methods involves commutators of the Lie derivatives $\hat{\mathcal{T}}$ and $\hat{\mathcal{V}}$, such as the Lie bracket (3.26). It can be shown [61, Lemma VII.3.1] that the Lie bracket of the Lie derivatives of two Hamiltonian vector fields is the Lie derivative of another Hamiltonian vector field. This latter vector field is given by the Poisson bracket (2.6) of the two Hamiltonians $\mathcal{H}^{\{\mathcal{T}\}} = \mathcal{T}$ and $\mathcal{H}^{\{\mathcal{V}\}} = \mathcal{V}$.

Example 3.15 (Modified Hamiltonian of the Strang splitting)**.** As an example, we consider the Strang splitting (3.5). Applying the symmetric BCH formula (3.29), we find that

$$\exp(\tfrac{h}{2}\hat{\mathcal{T}})\exp(h\hat{\mathcal{V}})\exp(\tfrac{h}{2}\hat{\mathcal{T}})\ \mathrm{Id} = \exp(h\tilde{\mathcal{L}})\ \mathrm{Id} \tag{3.39a}$$

with

$$\tilde{\mathcal{L}} = \hat{\mathcal{T}} + \hat{\mathcal{V}} + h^2\Big(-\frac{1}{24}[\hat{\mathcal{T}},[\hat{\mathcal{T}},\hat{\mathcal{V}}]] - \frac{1}{12}[\hat{\mathcal{V}},[\hat{\mathcal{T}},\hat{\mathcal{V}}]]\Big) + \mathcal{O}(h^4). \tag{3.39b}$$

Hence, the Strang splitting is the formally exact flow of the modified equation

$$\dot{\tilde{\boldsymbol{x}}} = \tilde{\boldsymbol{f}}(\boldsymbol{x}), \quad \tilde{\boldsymbol{f}} = \tilde{\mathcal{L}}\ \mathrm{Id}\,.$$

The derivation of the modified Hamiltonian is straightforward. Replacing the commutators of the Lie derivatives $\hat{\mathcal{T}}$ and $\hat{\mathcal{V}}$ by Poisson brackets of the Hamiltonians $\mathcal{H}^{\{1\}} = \mathcal{T}$ and $\mathcal{H}^{\{2\}} = \mathcal{V}$ yields the modified Hamiltonian (see (3.29b))

$$\tilde{\mathcal{H}} = \underbrace{\mathcal{T}+\mathcal{V}}_{=\mathcal{H}} + h^2\left(-\frac{1}{24}\{\mathcal{T},\{\mathcal{T},\mathcal{V}\}\} - \frac{1}{12}\{\mathcal{V},\{\mathcal{T},\mathcal{V}\}\}\right) + \mathcal{O}(h^4) \tag{3.39c}$$

of the Strang splitting.

In general, we have the result that a splitting method (3.8) of order p using the splitting (3.34) conserves a modified Hamiltonian $\tilde{\mathcal{H}} = \mathcal{H} + \mathcal{O}(h^p)$ exactly [147], i.e.,

$$\tilde{\mathcal{H}}(\boldsymbol{\Phi}_h(\boldsymbol{x}_0)) = \tilde{\mathcal{H}}(\boldsymbol{x}_0).$$

[3] In the physics literature, this modified Hamiltonian is often called *shadow Hamiltonian*, see e.g. [75].

Example 3.16 (Modified Hamiltonian for the mathematical pendulum)**.** We investigate the modified Hamiltonian (3.39c) of the Strang splitting in more detail by revisiting the mathematical pendulum (2.1). Applying the Strang splitting (3.5) to the mathematical pendulum yields the modified Hamiltonian

$$\tilde{\mathcal{H}} = \frac{1}{2}p^2 + 1 - \cos(q) + h^2 \left(-\frac{1}{24}\cos(q)p^2 + \frac{1}{12}\sin(q)^2 \right) + \mathcal{O}(h^4).$$

In Fig. 3.4, the Hamiltonian along the numerical solution over a long time interval $t \in [0, 100]$ is depicted, demonstrating that the energy of the numerical approximations oscillates around the true value $\mathcal{H}(\boldsymbol{p}_0, \boldsymbol{q}_0)$. This illustrates the excellent long-time energy behavior of splitting methods with the splitting (3.34), although the method is not able to conserve the energy exactly.

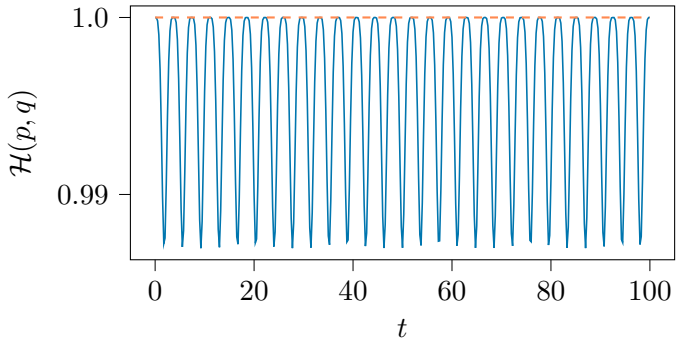


Fig. 3.4. Hamiltonian along the numerical solution obtained via the Strang splitting (3.5) for the mathematical pendulum (2.1) with initial values $(p_0, q_0) = (0, \pi/2)$ on the time interval $t \in [0, 100]$ (solid line). The dashed line indicates the Hamiltonian along the exact solution.

3.4.3. Order Conditions

The order conditions (3.31) for order $p \leq 3$ remain unchanged when considering systems of the form (3.34). Since the kinetic energy $\mathcal{T}(\boldsymbol{p})$ is only quadratic in $\boldsymbol{p}$, it holds

$$\left[\hat{\mathcal{V}}, [\hat{\mathcal{V}}, [\hat{\mathcal{V}}, \hat{\mathcal{T}}]]\right] = 0, \tag{3.40}$$

such that one of the three order-4 conditions vanishes. These simplifications also appear in the order conditions for $p > 4$, so that only a subset of the order conditions for general systems needs to be considered. In Tab. 3.1, the number of independent order conditions for general splitting methods and for splitting methods of RKN type up to order $p = 11$ are summarized.

order p	1	2	3	4	5	6	7	8	9	10	11
general systems	2	1	2	3	6	9	18	30	56	99	186
RKN type	2	1	2	2	4	5	10	14	25	39	69

Tab. 3.1. Number of independent order conditions for splitting methods (3.8) up to order $p = 11$ when applied to general dynamical systems (3.2), and to Hamiltonian systems of the form (3.32) with the splitting (3.34), giving rise to splitting methods of RKN type. The numbers are taken from [18].

3.4.4. Numerical Stability

The preceding discussion establishes that splitting methods for Hamiltonian systems (3.32) can be made symplectic and time-reversible by construction. As the flows of the subsystems (3.36) can be computed explicitly, these splitting methods are explicit. This is a major advantage of splitting the vector field into its kinetic and potential parts. For other numerical integration techniques, such as conventional Runge–Kutta methods that do not rely on any splitting of the right-hand side, the conditions for obtaining a symplectic and/or time-reversible numerical integration scheme imply that the method is necessarily implicit.

A limitation of the splitting approach, however, is that methods based on the splitting (3.34) are only conditionally stable [17]. This is because the flows of the subsystems (3.36) are unbounded, although the overall system (3.33) possesses bounded solutions. To assess the numerical stability, we select a simple test equation with bounded solutions and compute the stability limit of the splitting method (3.8). For separable Hamiltonian systems of the form $\mathcal{H}(\boldsymbol{p}, \boldsymbol{q}) = \mathcal{T}(\boldsymbol{p}) + \mathcal{V}(\boldsymbol{q})$, this is typically the harmonic oscillator. A linear stability analysis of splitting methods for Hamiltonian systems has been conducted in [17].

In the subsequent chapter, we will discuss more sophisticated splitting approaches for separable Hamiltonian systems of the form (3.32), which incorporate specific commutators of the Lie derivatives into the computational process.

Chapter 4.

Hessian-Free Force-Gradient Integrators

4.1. Motivation: The Force-Gradient Approach

The derivation of higher-order splitting methods of RKN type requires only a subset of the general order conditions, since several commutators vanish due to the fine structure of the Hamiltonian, see e.g. (3.40). Beyond this structural simplification, Hamiltonian systems of the form (3.32) possess the feature that a specific nested commutator of the Lie derivatives $\hat{\mathcal{T}}$ and $\hat{\mathcal{V}}$ generates a vector field whose flow can again be computed explicitly. In particular, we find that the nested commutator

$$\hat{\mathcal{C}} \coloneqq [\hat{\mathcal{V}}, [\hat{\mathcal{T}}, \hat{\mathcal{V}}]] = 2\hat{\mathcal{V}}\hat{\mathcal{T}}\hat{\mathcal{V}} - \underbrace{\hat{\mathcal{V}}\hat{\mathcal{V}}\hat{\mathcal{T}}}_{=0} - \underbrace{\hat{\mathcal{T}}\hat{\mathcal{V}}\hat{\mathcal{V}}}_{=0} = 2\hat{\mathcal{V}}\hat{\mathcal{T}}\hat{\mathcal{V}} = 2\nabla\mathcal{V}(\boldsymbol{q})\boldsymbol{M}^{-1}\nabla^2\mathcal{V}(\boldsymbol{q})\frac{\partial}{\partial \boldsymbol{p}} \tag{4.1}$$

depends only on the generalized coordinates $\boldsymbol{q}$. Since (4.1) incorporates the gradient of the force $-\nabla\mathcal{V}$, it is commonly referred to as the *force-gradient* term [28, 100, 76]. The corresponding ODE reads

$$\begin{pmatrix} \dot{\boldsymbol{p}} \\ \dot{\boldsymbol{q}} \end{pmatrix} = \begin{pmatrix} \nabla^2\mathcal{V}(\boldsymbol{q})\boldsymbol{M}^{-1}\nabla\mathcal{V}(\boldsymbol{q}) \\ \boldsymbol{0} \end{pmatrix}, \tag{4.2}$$

which is itself Hamiltonian with Hamiltonian function

$$\mathcal{H}^{\{\mathcal{C}\}}(\boldsymbol{p}, \boldsymbol{q}) \coloneqq \mathcal{C}(\boldsymbol{q}) = -\frac{1}{2}\nabla\mathcal{V}(\boldsymbol{q})^\top \boldsymbol{M}^{-1}\nabla\mathcal{V}(\boldsymbol{q}).$$

Its flow can be computed explicitly and reads

$$\varphi_t^{\{\mathcal{C}\}}\begin{pmatrix} \boldsymbol{p}_0 \\ \boldsymbol{q}_0 \end{pmatrix} = \begin{pmatrix} \boldsymbol{p}_0 + t\nabla^2\mathcal{V}(\boldsymbol{q}_0)\boldsymbol{M}^{-1}\nabla\mathcal{V}(\boldsymbol{q}_0) \\ \boldsymbol{q}_0 \end{pmatrix}. \tag{4.3}$$

As (4.2) is Hamiltonian, the flow (4.3) is symplectic and volume-preserving. Furthermore, since $\mathcal{H}^{\{\mathcal{C}\}}(-\boldsymbol{p}, \boldsymbol{q}) = \mathcal{H}^{\{\mathcal{C}\}}(\boldsymbol{p}, \boldsymbol{q})$, the flow is time-reversible.

The fundamental concept of the force-gradient approach is to incorporate evaluations of the flow (4.3) with step sizes $c_i h^3$ into the splitting method (3.8). This eliminates the force-gradient term from the BCH error expansion, enabling higher-order accuracy with fewer stages s.

Given that $[\hat{\mathcal{V}}, [\hat{\mathcal{T}}, \hat{\mathcal{V}}]] = -[\hat{\mathcal{V}}, [\hat{\mathcal{V}}, \hat{\mathcal{T}}]]$, the vanishing commutator (3.40) implies that the two vector fields $\hat{\mathcal{V}}$ and $\hat{\mathcal{C}}$ commute. Consequently, it holds

$$\exp(b_i h\hat{\mathcal{V}})\exp(c_i h^3\hat{\mathcal{C}}) = \exp(b_i h\hat{\mathcal{V}} + c_i h^3\hat{\mathcal{C}}),$$

i.e., the momentum update (3.36b) and the force-gradient flow (4.3) can be merged into a single *force-gradient update*

$$\begin{aligned} \varphi_{b_i h, c_i h^3}^{\{\mathcal{V},\mathcal{C}\}}\begin{pmatrix} \boldsymbol{p}_0 \\ \boldsymbol{q}_0 \end{pmatrix} &\coloneqq \exp(b_i h\hat{\mathcal{V}} + c_i h^3\hat{\mathcal{C}})\begin{pmatrix} \boldsymbol{p}_0 \\ \boldsymbol{q}_0 \end{pmatrix} \\ &= \begin{pmatrix} \boldsymbol{p}_0 - b_i h\nabla\mathcal{V}(\boldsymbol{q}_0) + 2c_i h^3\nabla^2\mathcal{V}(\boldsymbol{q}_0)\boldsymbol{M}^{-1}\nabla\mathcal{V}(\boldsymbol{q}_0) \\ \boldsymbol{q}_0 \end{pmatrix}. \end{aligned} \tag{4.4}$$

For $b_i \neq 0$, the force-gradient update (4.4) can be interpreted as the flow of a modified potential $\mathcal{H}^{\{\mathcal{V},\mathcal{C}\}}(\boldsymbol{p},\boldsymbol{q}) = \mathcal{V}(\boldsymbol{q}) + \frac{2c_i h^2}{b_i}\mathcal{C}(\boldsymbol{q})$, which explains the alternative terminology *splitting methods with modified potentials*, cf. [16]. It immediately follows that the flow (4.4) is symplectic and time-reversible.

Example 4.1. To illustrate the structural advantage of the force-gradient approach, we consider the symmetric three-stage family

$$\boldsymbol{\Phi}_h = \boldsymbol{\varphi}^{\{\mathcal{V}\}}_{b_1 h} \circ \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{h/2} \circ \boldsymbol{\varphi}^{\{\mathcal{V}\}}_{(1-2b_1)h} \circ \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{h/2} \circ \boldsymbol{\varphi}^{\{\mathcal{V}\}}_{b_1 h} \tag{4.5}$$

of velocity versions that is convergent of order $p = 2$ for arbitrary $b_1 \in \mathbb{R}$. Applying the BCH formula, (4.5) can be formally expressed as

$$\boldsymbol{\Phi}_h = \exp\left(h(\hat{\mathcal{T}} + \hat{\mathcal{V}}) + h^3\Big(\frac{6b_1 - 1}{24}[\hat{\mathcal{T}},[\hat{\mathcal{T}},\hat{\mathcal{V}}]] + \frac{6b_1 - 6b_1^2 - 1}{12}[\hat{\mathcal{V}},[\hat{\mathcal{T}},\hat{\mathcal{V}}]]\Big) + \mathcal{O}(h^5)\right)\mathrm{Id}\,.$$

With a single free parameter, both third-order error coefficients cannot be eliminated simultaneously. However, the force-gradient approach can be used to remove the force-gradient term (4.1). Choosing $b_1 = 1/6$ eliminates the remaining order-3 term $[\hat{\mathcal{T}},[\hat{\mathcal{T}},\hat{\mathcal{V}}]]$. Augmenting the central momentum update with an evaluation of the force-gradient term yields a *force-gradient integrator* [28]

$$\boldsymbol{\Phi}_h = \boldsymbol{\varphi}^{\{\mathcal{V}\}}_{h/6} \circ \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{h/2} \circ \boldsymbol{\varphi}^{\{\mathcal{V},\mathcal{C}\}}_{2h/3,h^3/72} \circ \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{h/2} \circ \boldsymbol{\varphi}^{\{\mathcal{V}\}}_{h/6} \tag{4.6}$$

with $s = 3$ stages and order $p = 4$. By using the notation of [101], we call this force-gradient integrator `BACAB`.

The force-gradient integrator (4.6) has the minimum number of stages $s = 3$ that are necessary to derive a self-adjoint integrator of order $p = 4$. For comparison, conventional self-adjoint splitting methods of order $p = 4$ require at least four stages, e.g., the triple jump composition (3.13) using the Strang splitting (3.5) as the underlying base scheme. After merging subsequent evaluations of the same flow, the composition becomes

$$\boldsymbol{\Phi}_h = \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{\gamma_1 h/2} \circ \boldsymbol{\varphi}^{\{\mathcal{V}\}}_{\gamma_1 h} \circ \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{(\gamma_1+\gamma_2)h/2} \circ \boldsymbol{\varphi}^{\{\mathcal{V}\}}_{\gamma_2 h} \circ \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{(\gamma_1+\gamma_2)h/2} \circ \boldsymbol{\varphi}^{\{\mathcal{V}\}}_{\gamma_1 h} \circ \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{\gamma_1 h/2}, \tag{4.7}$$

with γ_1, γ_2 as defined in (3.13) for $p = 2$. The force-gradient integrator (4.6) exhibits order $p = 4$ with fewer stages and avoids negative coefficients while remaining symplectic and time-reversible.

Hence, commutator-based splitting approaches can overcome the classical order barrier for non-negative integrator coefficients, which can be particularly useful for the development of splitting methods for nonreversible systems, such as linear port-Hamiltonian systems [97].

Example 4.2 (Force-gradient integrator for the mathematical pendulum)**.** We revisit the mathematical pendulum (2.1). For this example, the force-gradient term becomes

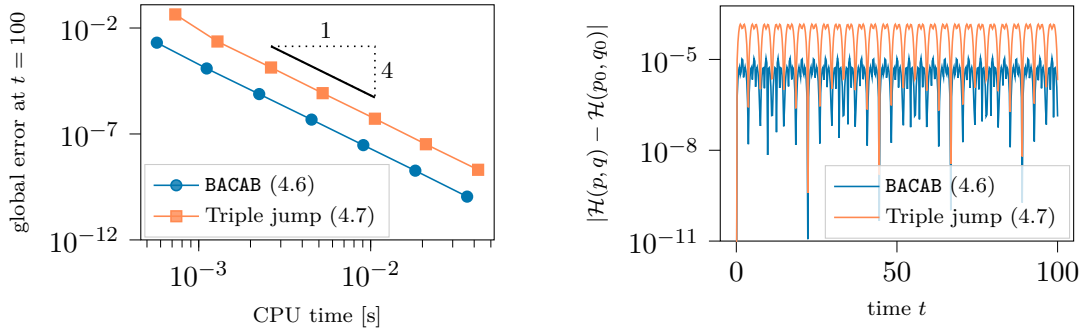
$$\hat{\mathcal{C}} = 2\sin(q)\cos(q)\frac{\partial}{\partial p},$$

such that the force-gradient update reads

$$\boldsymbol{\varphi}^{\{\mathcal{V},\mathcal{C}\}}_{b_i h, c_i h^3}\begin{pmatrix} p_0 \\ q_0 \end{pmatrix} = \begin{pmatrix} p_0 - b_i h \sin(q_0) + 2c_i h^3 \sin(q_0)\cos(q_0) \\ q_0 \end{pmatrix}.$$

We perform numerical simulations over $t \in [0, 100]$ with initial value $(p_0, q_0) = (0, \pi/2)$ for both the triple jump composition (4.7) and the force-gradient integrator (4.6). The

work-precision diagram in Fig. 4.1a shows that both numerical integration schemes scale with convergence order $p = 4$, as expected. Moreover, the force-gradient integrator attains a prescribed error tolerance at lower computational cost. Furthermore, the results depicted in Fig. 4.1b demonstrate that the force-gradient integrator yields smaller errors in the Hamiltonian.



(a) Global error vs. computation time (in seconds) for the force-gradient integrator `BACAB` (4.6) (○) and the triple jump composition (4.7) (□).

(b) Error in the Hamiltonian along numerical solutions of the force-gradient integrator `BACAB` (4.6) and the triple jump composition (4.7) using $h = 0.25$.

Fig. 4.1. Mathematical pendulum. Comparison of the numerical results obtained by the force-gradient integrator (4.6) and the triple jump composition (4.7). For both simulations, the time interval $t \in [0, 100]$ and initial value $(p_0, q_0) = (0, \pi/2)$ are considered.

We can incorporate the force-gradient approach into the entire class of splitting methods for two-split systems, resulting in the following definition of force-gradient integrators.

Definition 4.3 (Force-gradient integrator)**.** A single time step of step size h of an s-stage force-gradient integrator $\boldsymbol{\Phi}_h$ advances the given initial value $\boldsymbol{x}(0) = \boldsymbol{x}_0$ to a numerical approximation $\boldsymbol{x}_1 := \boldsymbol{\Phi}_h(\boldsymbol{x}_0) \approx \boldsymbol{x}(h)$ and is given as

$$\boldsymbol{\Phi}_h = \boldsymbol{\varphi}^{\{\mathcal{V},\mathcal{C}\}}_{b_s h, c_s h^3} \circ \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{a_s h} \circ \ldots \circ \boldsymbol{\varphi}^{\{\mathcal{V},\mathcal{C}\}}_{b_1 h, c_1 h^3} \circ \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{a_1 h}, \tag{4.8}$$

where $a_i, b_i, c_i \in \mathbb{R}$ denote the integrator coefficients of the method. For $c_i = 0$, the corresponding force-gradient update $\boldsymbol{\varphi}^{\{\mathcal{V},\mathcal{C}\}}_{b_i h, c_i h^3}$ reduces to a conventional momentum update $\boldsymbol{\varphi}^{\{\mathcal{V}\}}_{b_i h}$. If $c_i = 0$ for all $i = 1, \ldots, s$, then the force-gradient integrator reduces to a conventional splitting method.

Remark 4.4. We will use the ordering of the flow evaluations to provide unique identifiers for each integrator variant: `A` denotes a position update (3.36a), `B` a momentum update (3.36b) (or a force-gradient update with $c_i = 0$), and `C` a force-gradient update (4.4) with $c_i \neq 0$.

Being compositions of Hamiltonian flows, force-gradient integrators (4.8) are symplectic and volume-preserving. Therefore, they also preserve a nearby modified Hamiltonian exactly. As for conventional splitting methods, a force-gradient integrator (4.8) is time-reversible, provided that the integrator is self-adjoint.

Definition 4.5 (Self-adjoint force-gradient integrators)**.** A force-gradient integrator $\boldsymbol{\Phi}_h$ as defined in (4.8) is *self-adjoint*, provided that $\boldsymbol{\Phi}_h = \boldsymbol{\Phi}_h^*$ holds. This is the case, if either

$$a_1 = 0, \qquad a_{i+1} = a_{s+1-i}, \qquad b_i = b_{s+1-i}, \qquad \text{and} \qquad c_i = c_{s+1-i}, \tag{4.9a}$$

or

$$b_s = c_s = 0, \qquad b_i = b_{s-i}, \qquad c_i = c_{s-i} \qquad \text{and} \qquad a_i = a_{s+1-i}, \tag{4.9b}$$

holds for $i = 1, \dots, \lfloor \frac{s}{2} \rfloor$. We refer to force-gradient integrators satisfying (4.9a) as a *velocity version* and to those satisfying (4.9b) as a *position version*.

In [101], order conditions for self-adjoint force-gradient integrators up to order $p = 8$ have been derived. Furthermore, they conducted a complete classification of integrators with $s \leq 6$ stages, confirming substantial efficiency gains over conventional splitting methods of RKN type.

Despite their advantages, force-gradient integrators face two significant drawbacks. Firstly, there are systems where the force-gradient update (4.4) becomes significantly more expensive than a momentum update (3.36b). In this situation, splitting methods of RKN type with more stages can still result in a more efficient computational process. Secondly, the flow (4.3) requires the Hessian $\nabla^2 \mathcal{V}(\boldsymbol{q})$ of the potential. The Hessian may be difficult to compute analytically, or unavailable in practical software packages. In many large-scale applications, such as in lattice QCD, the force is readily available in software packages, while the Hessian generally is not.

This motivates the development of *Hessian-free force-gradient integrators* that retain the benefits of the force-gradient approach while avoiding evaluations of the Hessian, thereby combining computational efficiency with practical feasibility. The remainder of this chapter develops a theory of Hessian-free force-gradient integrators and summarizes the findings of our papers [115, 116, 117]. Next to a refined error analysis and a discussion of the geometric properties, we will conduct a linear stability analysis that also applies to conventional force-gradient integrators. The main objective of this section is to provide a complete classification of Hessian-free force-gradient integrators with $s \leq 6$ stages.

4.2. Hessian-Free Force-Gradient Integrators

4.2.1. Derivation of Hessian-Free Counterparts

Supposing $b_i \neq 0$, the force-gradient update (4.4) can be approximated via

$$\begin{aligned}
\boldsymbol{\varphi}^{\{\mathcal{V},\mathcal{C}\}}_{b_i h, c_i h^3} \begin{pmatrix} \boldsymbol{p}_0 \\ \boldsymbol{q}_0 \end{pmatrix} &= \exp(b_i h \hat{\mathcal{V}} + c_i h^3 \hat{\mathcal{C}}) \begin{pmatrix} \boldsymbol{p}_0 \\ \boldsymbol{q}_0 \end{pmatrix} = \exp(b_i h \hat{\mathcal{V}} + 2 c_i h^3 \hat{\mathcal{V}} \hat{\mathcal{T}} \hat{\mathcal{V}}) \begin{pmatrix} \boldsymbol{p}_0 \\ \boldsymbol{q}_0 \end{pmatrix} \\
&= \exp\left(-b_i h \nabla \mathcal{V}(\boldsymbol{q}) \frac{\partial}{\partial \boldsymbol{p}} + 2 c_i h^3 \nabla \mathcal{V}(\boldsymbol{q}) \boldsymbol{M}^{-1} \nabla^2 \mathcal{V}(\boldsymbol{q}) \frac{\partial}{\partial \boldsymbol{p}} \right) \begin{pmatrix} \boldsymbol{p}_0 \\ \boldsymbol{q}_0 \end{pmatrix} \\
&= \exp\left(-b_i h \Big(\mathrm{Id} - \frac{2 c_i h^2}{b_i} \nabla \mathcal{V}(\boldsymbol{q}) \boldsymbol{M}^{-1} \frac{\partial}{\partial \boldsymbol{q}} \Big) \nabla \mathcal{V}(\boldsymbol{q}) \frac{\partial}{\partial \boldsymbol{p}} \right) \begin{pmatrix} \boldsymbol{p}_0 \\ \boldsymbol{q}_0 \end{pmatrix} \\
&= \exp\left(-b_i h \exp\Big(- \frac{2 c_i h^2}{b_i} \nabla \mathcal{V}(\boldsymbol{q}_0) \boldsymbol{M}^{-1} \frac{\partial}{\partial \boldsymbol{q}} \Big) \nabla \mathcal{V}(\boldsymbol{q}) \frac{\partial}{\partial \boldsymbol{p}} \right) \begin{pmatrix} \boldsymbol{p}_0 \\ \boldsymbol{q}_0 \end{pmatrix} + \mathcal{O}(h^5).
\end{aligned}$$

Here, $\nabla \mathcal{V}(\boldsymbol{q}_0)$ denotes that the gradient is frozen at $\boldsymbol{q} = \boldsymbol{q}_0$ [32] so that $\partial / \partial \boldsymbol{q}$ acting on $\nabla \mathcal{V}(\boldsymbol{q}_0)$ is defined to be zero. Furthermore, performing a Taylor expansion verifies the formal identity [115]

$$\exp\Big(- \frac{2 c_i h^2}{b_i} \nabla \mathcal{V}(\boldsymbol{q}_0) \boldsymbol{M}^{-1} \frac{\partial}{\partial \boldsymbol{q}} \Big) \nabla \mathcal{V}(\boldsymbol{q}) = \nabla \mathcal{V}\Big(\exp\Big(- \frac{2 c_i h^2}{b_i} \boldsymbol{M}^{-1} \nabla \mathcal{V}(\boldsymbol{q}_0) \frac{\partial}{\partial \boldsymbol{q}} \Big) \boldsymbol{q} \Big).$$

Inserting this identity yields the update

$$\begin{aligned}
\boldsymbol{\varphi}^{\{\mathcal{D}\}}_{b_i h, c_i h^3} \begin{pmatrix} \boldsymbol{p}_0 \\ \boldsymbol{q}_0 \end{pmatrix} &\coloneqq \exp\left(-b_i h \nabla \mathcal{V}\Big(\exp\big(- \frac{2 c_i h^2}{b_i} \boldsymbol{M}^{-1} \nabla \mathcal{V}(\boldsymbol{q}_0) \frac{\partial}{\partial \boldsymbol{q}} \big) \boldsymbol{q} \Big) \frac{\partial}{\partial \boldsymbol{p}} \right) \begin{pmatrix} \boldsymbol{p}_0 \\ \boldsymbol{q}_0 \end{pmatrix} \\
&= \begin{pmatrix} \boldsymbol{p}_0 - b_i h \nabla \mathcal{V}\Big(\boldsymbol{q}_0 - \frac{2 c_i h^2}{b_i} \boldsymbol{M}^{-1} \nabla \mathcal{V}(\boldsymbol{q}_0) \Big) \\ \boldsymbol{q}_0 \end{pmatrix}
\end{aligned} \tag{4.10}$$

that can be decomposed into two steps: starting from the initial value $(\boldsymbol{p}_0, \boldsymbol{q}_0)$, we

1) compute a temporary position update via $\boldsymbol{q}' = \boldsymbol{q}_0 - \frac{2c_i h^2}{b_i} \boldsymbol{M}^{-1} \nabla \mathcal{V}(\boldsymbol{q}_0)$;

2) compute a conventional momentum update with step size $b_i h$ using the temporary position update for evaluating the force, $\boldsymbol{p}' = \boldsymbol{p}_0 - b_i h \nabla \mathcal{V}(\boldsymbol{q}')$.

The update (4.10) no longer involves evaluations of the Hessian $\nabla^2 \mathcal{V}$. Instead, it is effectively replaced by a second force evaluation. We thus refer to (4.10) as a *Hessian-free force-gradient update*.

Remark 4.6. The idea of approximating the force-gradient update (4.4) dates back to [144], where it was referred to as a *modified kick* and has been used to approximate the force-gradient update (4.4) for a specific force-gradient integrator. In the context of lattice field theories, the approximation has been used in [145] to obtain a Hessian-free counterpart of the force-gradient integrator `BACAB` (4.6). The derivation of the flow map (4.10) can thus be regarded as a generalization that makes the approximation accessible to the entire class of force-gradient integrators (4.8).

By substituting the force-gradient updates (4.4) with the Hessian-free counterparts (4.10), we obtain a novel class of *Hessian-free force-gradient integrators*.

Definition 4.7 (Hessian-free force-gradient integrator)**.** A single time step of step size h of an s-stage Hessian-free force-gradient integrator $\boldsymbol{\Phi}_h$ advances the given initial value $\boldsymbol{x}(0) = \boldsymbol{x}_0$ to a numerical approximation $\boldsymbol{x}_1 \coloneqq \boldsymbol{\Phi}_h(\boldsymbol{x}_0) \approx \boldsymbol{x}(h)$ and is given as

$$\boldsymbol{\Phi}_h = \boldsymbol{\varphi}^{\{\mathcal{D}\}}_{b_s h, c_s h^3} \circ \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{a_s h} \circ \ldots \circ \boldsymbol{\varphi}^{\{\mathcal{D}\}}_{b_1 h, c_1 h^3} \circ \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{a_1 h}, \tag{4.11}$$

where $a_i, b_i, c_i \in \mathbb{R}$ denote the integrator coefficients of the method. For $c_i = 0$, the corresponding Hessian-free force-gradient update $\boldsymbol{\varphi}^{\{\mathcal{D}\}}_{b_i h, c_i h^3}$ reduces to a conventional momentum update $\boldsymbol{\varphi}^{\{\mathcal{V}\}}_{b_i h}$. If $c_i = 0$ for all $i = 1, \ldots, s$, then the Hessian-free force-gradient integrator reduces to a conventional splitting method.

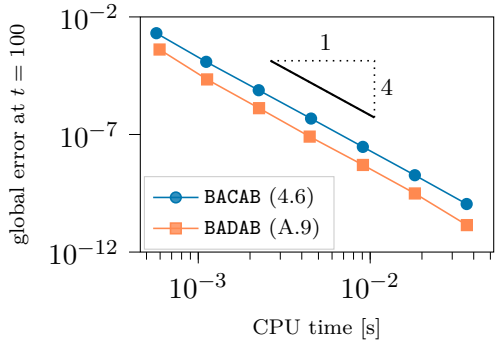
Remark 4.8. We will use the ordering of the flow evaluations to provide unique identifiers for each integrator variant: `A` denotes a position update (3.36a), `B` a momentum update (3.36b) (or a Hessian-free force-gradient update with $c_i = 0$), and `D` a Hessian-free force-gradient update (4.10) with $c_i \neq 0$.

The Hessian-free force-gradient update (4.10) is only defined for $b_i \neq 0$, i.e., the approximation of the force-gradient update (4.4) is only possible if it also contains a momentum update. Despite this theoretical limitation, we did not encounter a situation during our comprehensive investigations where the case $b_i = 0$ and $c_i \neq 0$ arose while developing particularly efficient integrator variants.

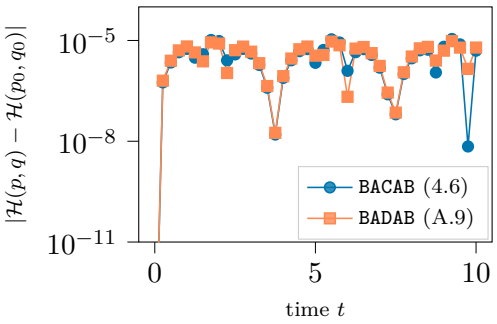
Example 4.9. We revisit the mathematical pendulum (2.1) and the force-gradient integrator (4.6). The Hessian-free force-gradient update (4.10) for the mathematical pendulum reads

$$\boldsymbol{\varphi}^{\{\mathcal{D}\}}_{b_i h, c_i h^3} \begin{pmatrix} p_0 \\ q_0 \end{pmatrix} = \begin{pmatrix} p_0 - b_i h \sin\left(q_0 - \frac{2c_i h^2}{b_i} \sin(q_0)\right) \\ q_0 \end{pmatrix}. \tag{4.12}$$

We compare the force-gradient integrator (4.6) with its Hessian-free counterpart (A.9) by repeating the numerical simulations from Example 4.2. The numerical results depicted in Fig. 4.2a demonstrate that the Hessian-free updates do not affect the convergence order of the integrator and that the Hessian-free variant attains a prescribed error tolerance at lower computational cost. At the same time, the errors in the Hamiltonian depicted in Fig. 4.2b do not show any significant differences.



(a) Global error vs. computation time (in seconds) for the force-gradient integrator `BACAB` (4.6) (○) and its Hessian-free counterpart `BADAB` (A.9) (□).



(b) Error in the Hamiltonian along numerical solutions of the force-gradient integrator `BACAB` (4.6) (○) and its Hessian-free counterpart `BADAB` (□) using $h = 0.25$.

Fig. 4.2. Mathematical pendulum. Comparison of the numerical results obtained by the force-gradient integrator (4.6) its Hessian-free counterpart. For both experiments, the initial value $(p_0, q_0) = (0, \pi/2)$ is considered. The work-precision diagram Fig. 4.2a considers the time interval $t \in [0, 100]$, whereas the results for Fig. 4.2b considers a shorter time interval $t \in [0, 10]$ for visualization purposes.

4.2.2. Order Conditions

Although the two-step procedure described above is used in practice, the following equivalent expression is more convenient for the error analysis.

Theorem 4.10 (Hessian-free force-gradient step)**.** *For the Hessian-free force-gradient update* (4.10)*, we have the formal identity*

$$\varphi^{\{\mathcal{D}\}}_{b_i h, c_i h^3} = \exp(b_i h \hat{\mathcal{D}}_{b_i, c_i, h})\,\mathrm{Id} \tag{4.13a}$$

with

$$\hat{\mathcal{D}}_{b_i,c_i,h} = \sum_{k=0}^{\infty} \frac{1}{k!} \left(\frac{2c_i h^2}{b_i}\right)^k (\widetilde{\mathcal{V}}\hat{\mathcal{T}})^k \hat{\mathcal{V}}, \quad \widetilde{\mathcal{V}} := \nabla\mathcal{V}(\boldsymbol{q}_0)\frac{\partial}{\partial \boldsymbol{p}}. \tag{4.13b}$$

Proof. With the definition of $\widetilde{\mathcal{V}}$, we can write the approximation of the force-gradient update as

$$\exp\left(-b_i h \exp\left(-\frac{2c_i h^2}{b_i}\nabla\mathcal{V}(\boldsymbol{q}_0)\boldsymbol{M}^{-1}\frac{\partial}{\partial \boldsymbol{q}}\right)\nabla\mathcal{V}(\boldsymbol{q})\frac{\partial}{\partial \boldsymbol{p}}\right)\mathrm{Id} = \exp\left(b_i h \exp\left(\frac{2c_i h^2}{b_i}\widetilde{\mathcal{V}}\hat{\mathcal{T}}\right)\hat{\mathcal{V}}\right)\mathrm{Id}\,.$$

Expanding the inner exponential yields

$$\exp\left(b_i h \sum_{k=0}^{\infty} \frac{1}{k!}\left(\frac{2c_i h^2}{b_i}\right)^k (\widetilde{\mathcal{V}}\hat{\mathcal{T}})^k \hat{\mathcal{V}}\right)\mathrm{Id},$$

concluding the proof. □

As the approximation (4.10) of the force-gradient update (4.4) introduces an error of size $\mathcal{O}(h^5)$, the order conditions up to order four remain unchanged compared to conventional force-gradient integrators. We will restrict the order analysis to the subclass of self-adjoint Hessian-free force-gradient integrators. To assess the order conditions up to order $p = 8$, we will extend the symmetric BCH formula (3.29) to the framework of Hessian-free force-gradient integrators. Using (4.13) and the BCH formula, we find that a self-adjoint Hessian-free force-gradient integrator can be formally expressed as

$$\boldsymbol{\Phi}_h = \exp\left(h(\nu\hat{\mathcal{T}} + \sigma\hat{\mathcal{V}}) + h^3\mathcal{O}_3 + h^5\mathcal{O}_5 + h^7\mathcal{O}_7 + \mathcal{O}(h^9)\right)\mathrm{Id}, \tag{4.14a}$$

with

$$\mathcal{O}_3 = \alpha[\hat{\mathcal{T}},[\hat{\mathcal{T}},\hat{\mathcal{V}}]] + \beta[\hat{\mathcal{V}},[\hat{\mathcal{T}},\hat{\mathcal{V}}]], \tag{4.14b}$$

$$\begin{aligned}\mathcal{O}_5 = {}& \gamma_1[\hat{\mathcal{T}},[\hat{\mathcal{T}},[\hat{\mathcal{T}},[\hat{\mathcal{T}},\hat{\mathcal{V}}]]]] + \gamma_2[\hat{\mathcal{T}},[\hat{\mathcal{T}},[\hat{\mathcal{V}},[\hat{\mathcal{T}},\hat{\mathcal{V}}]]]] \\ & + \gamma_3[\hat{\mathcal{V}},[\hat{\mathcal{T}},[\hat{\mathcal{T}},[\hat{\mathcal{T}},\hat{\mathcal{V}}]]]] + \gamma_4[\hat{\mathcal{V}},[\hat{\mathcal{V}},[\hat{\mathcal{T}},[\hat{\mathcal{T}},\hat{\mathcal{V}}]]]] + \gamma_5\tilde{\mathcal{V}}\hat{\mathcal{T}}\tilde{\mathcal{V}}\hat{\mathcal{T}}\hat{\mathcal{V}},\end{aligned} \tag{4.14c}$$

$$\begin{aligned}\mathcal{O}_7 = {}& \zeta_1[\hat{\mathcal{V}},[\hat{\mathcal{V}},[\hat{\mathcal{T}},[\hat{\mathcal{V}},[\hat{\mathcal{T}},[\hat{\mathcal{V}},\hat{\mathcal{T}}]]]]]] + \zeta_2[\hat{\mathcal{V}},[\hat{\mathcal{V}},[\hat{\mathcal{V}},[\hat{\mathcal{T}},[\hat{\mathcal{T}},[\hat{\mathcal{V}},\hat{\mathcal{T}}]]]]]] \\ & + \zeta_3[\hat{\mathcal{V}},[\hat{\mathcal{V}},[\hat{\mathcal{T}},[\hat{\mathcal{T}},[\hat{\mathcal{T}},[\hat{\mathcal{V}},\hat{\mathcal{T}}]]]]]] + \zeta_4[\hat{\mathcal{V}},[\hat{\mathcal{T}},[\hat{\mathcal{V}},[\hat{\mathcal{T}},[\hat{\mathcal{T}},[\hat{\mathcal{V}},\hat{\mathcal{T}}]]]]]] \\ & + \zeta_5[\hat{\mathcal{T}},[\hat{\mathcal{V}},[\hat{\mathcal{V}},[\hat{\mathcal{T}},[\hat{\mathcal{T}},[\hat{\mathcal{V}},\hat{\mathcal{T}}]]]]]] + \zeta_6[\hat{\mathcal{T}},[\hat{\mathcal{V}},[\hat{\mathcal{T}},[\hat{\mathcal{V}},[\hat{\mathcal{T}},[\hat{\mathcal{V}},\hat{\mathcal{T}}]]]]]] \\ & + \zeta_7[\hat{\mathcal{V}},[\hat{\mathcal{T}},[\hat{\mathcal{T}},[\hat{\mathcal{T}},[\hat{\mathcal{T}},[\hat{\mathcal{V}},\hat{\mathcal{T}}]]]]]] + \zeta_8[\hat{\mathcal{T}},[\hat{\mathcal{V}},[\hat{\mathcal{T}},[\hat{\mathcal{T}},[\hat{\mathcal{T}},[\hat{\mathcal{V}},\hat{\mathcal{T}}]]]]]] \\ & + \zeta_9[\hat{\mathcal{T}},[\hat{\mathcal{T}},[\hat{\mathcal{V}},[\hat{\mathcal{T}},[\hat{\mathcal{T}},[\hat{\mathcal{V}},\hat{\mathcal{T}}]]]]]] + \zeta_{10}[\hat{\mathcal{T}},[\hat{\mathcal{T}},[\hat{\mathcal{T}},[\hat{\mathcal{T}},[\hat{\mathcal{T}},[\hat{\mathcal{V}},\hat{\mathcal{T}}]]]]]] \\ & + \zeta_{11}\tilde{\mathcal{V}}\hat{\mathcal{T}}\tilde{\mathcal{V}}\hat{\mathcal{T}}\tilde{\mathcal{V}}\hat{\mathcal{T}}\hat{\mathcal{V}} + \zeta_{12}\hat{\mathcal{T}}\tilde{\mathcal{V}}\hat{\mathcal{T}}\tilde{\mathcal{V}}\hat{\mathcal{T}}\hat{\mathcal{V}}\hat{\mathcal{T}} + \zeta_{13}\hat{\mathcal{T}}\hat{\mathcal{T}}\tilde{\mathcal{V}}\hat{\mathcal{T}}\tilde{\mathcal{V}}\hat{\mathcal{T}}\hat{\mathcal{V}}.\end{aligned} \tag{4.14d}$$

Compared to conventional force-gradient integrators [101], additional error terms involving the coefficients $\gamma_5, \zeta_{11}, \zeta_{12}$, and ζ_{13} arise due to the approximation of the force-gradient update. As a result, we have one additional order-5 condition and three additional order-7 conditions. The coefficients can be computed by iteratively applying the following two modifications of the symmetric BCH formula.

First, we consider symmetric position updates of the form

$$\exp(a_j h\hat{\mathcal{T}}) \exp\left(h(\nu\hat{\mathcal{T}} + \sigma\hat{\mathcal{V}}) + h^3\mathcal{O}_3 + h^5\mathcal{O}_5 + h^7\mathcal{O}_7 + \mathcal{O}(h^9)\right) \exp(a_j h\hat{\mathcal{T}}). \tag{4.15}$$

These updates can also be expressed formally in the form (4.14) with updated coefficients $\nu', \sigma', \gamma'_{1-4}$ and ζ'_{1-10} as for conventional force-gradient integrators [101, eqs. (15)–(18), (A.1)], and

$$\begin{aligned}\gamma'_5 &= \gamma_5, & \zeta'_{11} &= \zeta_{11}, \\ \zeta'_{12} &= \zeta_{12} + a_j^2\gamma_5/3, & \zeta'_{13} &= \zeta_{13} - a_j^2\gamma_5/6.\end{aligned} \tag{4.16}$$

Similarly, we can consider symmetric Hessian-free force-gradient updates

$$\exp(b_j h\hat{\mathcal{D}}_{b_j,c_j,h}) \exp\left(h(\nu\hat{\mathcal{T}}+\sigma\hat{\mathcal{V}})+h^3\mathcal{O}_3+h^5\mathcal{O}_5+h^7\mathcal{O}_7+\mathcal{O}(h^9)\right) \exp(b_j h\hat{\mathcal{D}}_{b_j,c_j,h}). \tag{4.17}$$

Once more, we can expand the product and express the result formally in terms of an exponential (4.14) with updated coefficients $\nu', \sigma', \gamma'_{1-4}$ and ζ'_{1-10} as for conventional force-gradient integrators [101, eqs. (19)–(22), (A.2)], and

$$\begin{aligned}\gamma'_5 &= \gamma_5 + 4c_j^2/b_j, & \zeta'_{11} &= \zeta_{11} + 8c_j^3/(3b_j^2) + 2\sigma\nu c_j^2/(3b_j), \\ \zeta'_{12} &= \zeta_{12} - 2\nu^2 c_j^2/(3b_j), & \zeta'_{13} &= \zeta_{13} + \nu^2 c_j^2/(3b_j).\end{aligned} \tag{4.18}$$

Starting with the centered exponential of the integrator, we can iteratively apply the expansions for the symmetric position updates (4.15) and the symmetric Hessian-free force-gradient updates (4.17) until we finally obtain a formal identity of the Hessian-free force-gradient integrator with an exponential of the form (4.14). A self-adjoint Hessian-free force-gradient integrator satisfying $\nu = \sigma = 1$ is consistent and, due to the inherent symmetry, convergent of order $p \geq 2$. The conditions $\alpha = \beta = 0$, $\gamma_i = 0\,\forall i = 1,\dots,5$, and $\zeta_j = 0\,\forall j = 1,\dots,13$ constitute the order conditions for order three, five, and seven, respectively. Symmetric methods exhibit an even convergence order, implying that satisfying the order conditions up to orders three, five, and seven yields methods of convergence orders four, six, and eight, respectively.

4.2.3. Geometric Properties

Substituting the force-gradient updates (4.4) with the Hessian-free force-gradient updates (4.10) does not affect the time-reversibility of the integrator, as shown by the following theorem.

Theorem 4.11. *The Hessian-free force-gradient updates* (4.10) *are time-reversible.*

Proof. We have to show that the Hessian-free force-gradient update (4.10) satisfies the time-reversibility condition (2.15). Applying $\boldsymbol{\rho}$ to the flow (4.10) yields

$$\boldsymbol{\rho} \circ \boldsymbol{\varphi}^{\{\mathcal{D}\}}_{b_i h, c_i h^3} \begin{pmatrix} \boldsymbol{p}_0 \\ \boldsymbol{q}_0 \end{pmatrix} = \begin{pmatrix} -\boldsymbol{p}_0 + b_i h \nabla \mathcal{V}\Big(\boldsymbol{q}_0 - \frac{2c_i h^2}{b_i} \boldsymbol{M}^{-1} \nabla \mathcal{V}(\boldsymbol{q}_0)\Big) \\ \boldsymbol{q}_0 \end{pmatrix}.$$

Another application of the flow map with the same step size results in

$$\boldsymbol{\varphi}^{\{\mathcal{D}\}}_{b_i h, c_i h^3} \circ \boldsymbol{\rho} \circ \boldsymbol{\varphi}^{\{\mathcal{D}\}}_{b_i h, c_i h^3} \begin{pmatrix} \boldsymbol{p}_0 \\ \boldsymbol{q}_0 \end{pmatrix} = \begin{pmatrix} -\boldsymbol{p}_0 \\ \boldsymbol{q}_0 \end{pmatrix},$$

such that a final application of $\boldsymbol{\rho}$ yields back the initial value $(\boldsymbol{p}_0, \boldsymbol{q}_0)$. In summary, it holds

$$\boldsymbol{\rho} \circ \boldsymbol{\varphi}^{\{\mathcal{D}\}}_{b_i h, c_i h^3} \circ \boldsymbol{\rho} \circ \boldsymbol{\varphi}^{\{\mathcal{D}\}}_{b_i h, c_i h^3} = \mathrm{Id},$$

i.e., the Hessian-free force-gradient update (4.10) is time-reversible. □

As an immediate consequence of Theorem 4.11, a Hessian-free force-gradient integrator is time-reversible, provided that the underlying conventional force-gradient integrator is. Consequently, the symmetry conditions (4.9) also apply to the Hessian-free framework.

While the time-reversibility is preserved, Hessian-free force-gradient integrators are generally no longer symplectic. The sensitivity matrix of the Hessian-free force-gradient update reads

$$\frac{\partial \boldsymbol{\varphi}^{\{\mathcal{D}\}}_{b_i h, c_i h^3}(\boldsymbol{p}_0, \boldsymbol{q}_0)}{\partial (\boldsymbol{p}_0, \boldsymbol{q}_0)} = \begin{pmatrix} \boldsymbol{I} & \boldsymbol{B} \\ \boldsymbol{0} & \boldsymbol{I} \end{pmatrix}, \tag{4.19}$$

where

$$\boldsymbol{B} = -b_i h \nabla^2 \mathcal{V}\Big(\boldsymbol{q}_0 - \frac{2c_i h^2}{b_i} \boldsymbol{M}^{-1} \nabla \mathcal{V}(\boldsymbol{q}_0)\Big)\Big(\boldsymbol{I} - \frac{2c_i h^2}{b_i} \boldsymbol{M}^{-1} \nabla^2 \mathcal{V}(\boldsymbol{q})\Big). \tag{4.20}$$

By inserting the sensitivity matrix into the symplecticity condition (2.11), we derive that the Hessian-free force-gradient update (4.10) is symplectic, if and only if

$$\nabla^2 \mathcal{V}(\boldsymbol{q}) \boldsymbol{M}^{-1} \nabla^2 \mathcal{V}(\boldsymbol{q}') = \nabla^2 \mathcal{V}(\boldsymbol{q}') \boldsymbol{M}^{-1} \nabla^2 \mathcal{V}(\boldsymbol{q}) \tag{4.21}$$

holds for generally different $\boldsymbol{q}, \boldsymbol{q}' \in \mathbb{R}^d$. Although (4.21) is not satisfied in general, there are two subclasses for which the condition is fulfilled. The first subclass constitutes Hamiltonian systems (3.32) with a single degree of freedom ($d = 1$). In this scenario, the matrices in (4.21) become scalar-valued and thus commute. For instance, Hessian-free force-gradient integrators are symplectic when applied to the mathematical pendulum (2.1). The second subclass comprises Hamiltonian systems (3.32) with a quadratic Hamiltonian, i.e., the potential energy is only quadratic in $\boldsymbol{q}$. Then, the Hessian $\nabla^2 \mathcal{V}$ is constant, satisfying the condition (4.21). For this subclass, we have an even stronger result.

Lemma 4.12. *Consider a Hamiltonian system* (3.32) *with a quadratic potential of the form* $\mathcal{V}(\boldsymbol{q}) = \frac{1}{2} \boldsymbol{q}^\top \boldsymbol{N} \boldsymbol{q}$*, where* $\boldsymbol{N} \in \mathbb{R}^{d \times d}$ *is a constant matrix. In this case, the force-gradient update* (4.4) *and the Hessian-free counterpart* (4.10) *are equivalent (for* $b_i \neq 0$*).*

Proof. Let $\tilde{\boldsymbol{N}} = \boldsymbol{N} + \boldsymbol{N}^\top$. Given the structure of the Hamiltonian, the force-gradient update (4.4) reads

$$\begin{pmatrix} \boldsymbol{p}_0 - b_i h \tilde{\boldsymbol{N}} \boldsymbol{q}_0 + 2c_i h^3 \tilde{\boldsymbol{N}} \boldsymbol{M}^{-1} \tilde{\boldsymbol{N}} \boldsymbol{q}_0 \\ \boldsymbol{q}_0 \end{pmatrix}.$$

For the Hessian-free force-gradient update (4.10), we obtain

$$\begin{pmatrix} \boldsymbol{p}_0 - b_i h \tilde{\boldsymbol{N}} (\boldsymbol{q}_0 - \frac{2c_i h^2}{b_i} \boldsymbol{M}^{-1} \tilde{\boldsymbol{N}} \boldsymbol{q}_0) \\ \boldsymbol{q}_0 \end{pmatrix} = \begin{pmatrix} \boldsymbol{p}_0 - b_i h \tilde{\boldsymbol{N}} \boldsymbol{q}_0 + 2c_i h^3 \tilde{\boldsymbol{N}} \boldsymbol{M}^{-1} \tilde{\boldsymbol{N}} \boldsymbol{q}_0 \\ \boldsymbol{q}_0 \end{pmatrix}.$$

Consequently, the two updates are equivalent. □

Although the update is not symplectic in general, the sensitivity matrix (4.19) obviously has determinant one, i.e., (2.12) holds. Consequently, Hessian-free force-gradient integrators remain volume-preserving.

In the absence of symplecticity, Hessian-free force-gradient integrators (4.11) no longer conserve a modified Hamiltonian. This is a notable loss, since the existence of a modified Hamiltonian is a key feature of symplectic methods, implying excellent long-time energy behavior. Instead, the modified differential equation $\dot{\tilde{\boldsymbol{x}}} = \tilde{\boldsymbol{f}}(\boldsymbol{x})$, $\tilde{\boldsymbol{f}} = \tilde{\mathcal{L}}\,\mathrm{Id}$ of a Hessian-free force-gradient integrator reads

$$\tilde{\mathcal{L}} = \{\cdot, \tilde{\mathcal{H}}_{\mathtt{FG}}\} + h^4 \gamma_5 \tilde{\mathcal{V}}\hat{\mathcal{T}}\tilde{\mathcal{V}}\hat{\mathcal{T}}\hat{\mathcal{V}} + h^6 \Big(\zeta_{11} \tilde{\mathcal{V}}\hat{\mathcal{T}}\tilde{\mathcal{V}}\hat{\mathcal{T}}\tilde{\mathcal{V}}\hat{\mathcal{T}}\hat{\mathcal{V}} + \zeta_{12} \hat{\mathcal{T}}\tilde{\mathcal{V}}\hat{\mathcal{T}}\tilde{\mathcal{V}}\hat{\mathcal{T}}\hat{\mathcal{V}}\hat{\mathcal{T}} + \zeta_{13} \hat{\mathcal{T}}\hat{\mathcal{T}}\tilde{\mathcal{V}}\hat{\mathcal{T}}\tilde{\mathcal{V}}\hat{\mathcal{T}}\hat{\mathcal{V}} \Big) + \mathcal{O}(h^8), \quad (4.22)$$

where the first term corresponds to the modified Hamiltonian $\mathcal{H}_{\mathtt{FG}}$ of the underlying conventional force-gradient integrator, utilizing the same set of integrator coefficients. The latter terms introduce an energy drift. In general, without making any further assumption on the potential $\mathcal{V}$, the energy will have a linear drift of size $\mathcal{O}(th^{\max\{4,p\}})$ for a Hessian-free force-gradient integrator of order p and a time interval of length t.

Remark 4.13. It is possible to select the integrator coefficients of a Hessian-free force-gradient integrator to eliminate the drift terms, for instance, by setting $\gamma_5 = 0$. In such cases, we can have a Hessian-free force-gradient integrator of order $p \geq 4$ where the energy will have a linear drift of size $\mathcal{O}(th^q)$ with $q > p$.

Depending on the specific system, the energy drift can be observed when performing numerical simulations for exponentially long time intervals, such as in the numerical experiments conducted in [62] for a particular Hessian-free force-gradient integrator known as the *simplified Takahashi–Imada method*. However, in many practical situations the linear drift of size $\mathcal{O}(th^q)$ does not become observable, since the time interval length t remains comparatively small. This is the case in lattice QCD simulations, discussed in Part II of this thesis, where t corresponds to the length of a single molecular dynamics trajectory, with $t = 2$ being a common choice.

4.3. Linear Stability Analysis

Complementing the accuracy analysis of the preceding section, we now assess the numerical stability of the integrators to prevent numerical instabilities from becoming the limiting factor in practical computations. The position (3.36a), momentum (3.36b), force-gradient (4.4), and Hessian-free force-gradient (4.10) updates are not bounded. Consequently, (Hessian-free) force-gradient integrators are, like conventional splitting methods of RKN type, only conditionally stable. To conduct a numerical stability

analysis, we select a simple test problem with bounded solutions whose flow lies in the same subgroup of the group of diffeomorphisms. In the context of Hamiltonian systems, this is typically the harmonic oscillator (after an appropriate rescaling of variables)

$$\begin{pmatrix}\dot{p}\\ \dot{q}\end{pmatrix} = \begin{pmatrix}0\\ \omega p\end{pmatrix} + \begin{pmatrix}-\omega q\\ 0\end{pmatrix}. \tag{4.23}$$

In [17], a linear stability analysis for conventional splitting methods was conducted by investigating (4.23) as a test equation. This section will discuss the results from the paper [117], where we extended the linear stability analysis to force-gradient integrators and their Hessian-free counterparts.

The first important observation is that (4.23) represents a Hamiltonian system with Hamiltonian $\mathcal{H}(p,q) = \frac{\omega}{2}p^2 + \frac{\omega}{2}q^2$. As a result, Lemma 4.12 applies, i.e., force-gradient integrators and their Hessian-free counterparts are equivalent when applied to the harmonic oscillator (4.23). Consequently, the linear stability analysis for both classes is identical. Without loss of generality, we perform the linear stability analysis for conventional force-gradient integrators.

For the test equation (4.23), the force-gradient term (4.1) is given by

$$\hat{\mathcal{C}} = 2\omega^3 q \frac{\partial}{\partial p}. \tag{4.24}$$

The integrator will typically become unstable for $|z| > z_*$, where $z \coloneqq \omega h$ and z_* represents the *stability threshold* of the integrator. The objective is to determine the value z_* for a given force-gradient integrator by adapting the linear stability analysis for conventional splitting methods from [17]. Starting from the initial value $(p(0), q(0)) = (p_0, q_0)$, the exact solution of the harmonic oscillator (4.23) at time point $h > 0$ is given by

$$\begin{pmatrix}p(h)\\ q(h)\end{pmatrix} = \underbrace{\begin{pmatrix}\cos(z) & -\sin(z)\\ \sin(z) & \cos(z)\end{pmatrix}}_{=\boldsymbol{O}(z)} \begin{pmatrix}p_0\\ q_0\end{pmatrix}, \quad z \coloneqq \omega h.$$

Applying a force-gradient integrator (4.8) to the harmonic oscillator (4.23) yields

$$\begin{pmatrix}p_1\\ q_1\end{pmatrix} = \boldsymbol{K}(z) \begin{pmatrix}p_0\\ q_0\end{pmatrix}$$

with the *stability matrix*

$$\boldsymbol{K}(z) = \prod_{\ell=1}^{s} \begin{pmatrix}1 & -b_{s+1-\ell}z + 2c_{s+1-\ell}z^3\\ 0 & 1\end{pmatrix} \begin{pmatrix}1 & 0\\ a_{s+1-\ell}z & 1\end{pmatrix}, \tag{4.25}$$

which is an approximation to $\boldsymbol{O}(z)$. Additionally, we define the *stability polynomial*

$$\mathfrak{p}(z) \coloneqq \frac{1}{2}\operatorname{tr}(\boldsymbol{K}(z)). \tag{4.26}$$

In the case of self-adjoint force-gradient integrators, the stability matrix (4.25) takes the form

$$\boldsymbol{K}(z) = \begin{pmatrix}\mathfrak{p}(z) & K_{1,2}(z)\\ K_{2,1}(z) & \mathfrak{p}(z)\end{pmatrix}, \tag{4.27a}$$

with even stability polynomial

$$\mathfrak{p}(z) = 1 + \sum_{i=1}^{2(s-1)} \bar{p}_i z^{2i}, \tag{4.27b}$$

and odd polynomials

$$K_{1,2}(z) = \sum_{i=1}^{2s} \bar{k}_i z^{2i-1}, \quad K_{2,1}(z) = \sum_{i=1}^{2s-1} k_i z^{2i-1}, \tag{4.27c}$$

where $\bar{p}_i, k_i, \bar{k}_i$ are homogeneous polynomials in the coefficients a_ℓ, b_ℓ, c_ℓ. In particular, it holds

$$k_{2s-2} = k_{2s-1} = 0$$

for a velocity version (4.9a), and

$$\bar{k}_{2s-1} = \bar{k}_{2s} = 0$$

for a position version (4.9b) of the force-gradient integrator (4.8). Furthermore, for consistent schemes satisfying $\sum_j a_j = \sum_j b_j = 1$, it follows that

$$\bar{p}_1 = -\tfrac{1}{2}, \quad k_1 = 1, \quad \bar{k}_1 = -1,$$

resulting in $\mathfrak{p}(z) = 1 - z^2/2 + \mathcal{O}(z^4)$.

Example 4.14. For the force-gradient integrator (4.6) and its Hessian-free counterpart (A.9), the stability matrix reads

$$\boldsymbol{K}(z) = \begin{pmatrix} 1 - \frac{z^2}{2} + \frac{z^4}{24} - \frac{z^6}{864} & -z + \frac{z^3}{6} - \frac{z^5}{108} + \frac{z^7}{5184} \\ z - \frac{z^3}{6} + \frac{z^5}{144} & 1 - \frac{z^2}{2} + \frac{z^4}{24} - \frac{z^6}{864} \end{pmatrix},$$

with the stability polynomial $\mathfrak{p}(z) = 1 - \frac{z^2}{2} + \frac{z^4}{24} - \frac{z^6}{864}$.

The linear stability analysis presented in [17] can be extended to force-gradient integrators in a straightforward manner. We provide a concise summary of the fundamental results, tailored to the framework of force-gradient integrators. A force-gradient integrator (4.8) is considered stable when applied to the harmonic oscillator (4.23) if $\boldsymbol{K}(z)^i$ is bounded independently of the number of time steps $i \geq 1$.

Definition 4.15 (Stability interval)**.** We denote by z_* the largest non-negative real number such that $\boldsymbol{K}(z)$ is stable for all $z \in (-z_*, z_*)$. We refer to z_* as the stability threshold and call $(-z_*, z_*)$ the stability interval of $\boldsymbol{K}(z)$.

We further introduce the following characterization of stability.

Proposition 4.16. *Let $\boldsymbol{K}(z)$ denote the stability matrix of a force-gradient integrator (4.8) and $\mathfrak{p}(z)$ its stability polynomial. Then, the following statements are equivalent:*

(i) *The matrix $\boldsymbol{K}(z)$ is stable.*

(ii) *The matrix $\boldsymbol{K}(z)$ is diagonalizable with eigenvalues of modulus one.*

(iii) *It holds $|\mathfrak{p}(z)| \leq 1$, and $\boldsymbol{K}(z)$ is similar to the matrix*

$$\boldsymbol{S}(z) = \begin{pmatrix} \cos\Psi(z) & -\sin\Psi(z) \\ \sin\Psi(z) & \cos\Psi(z) \end{pmatrix}, \quad \textit{where } \Psi(z) = \arccos \mathfrak{p}(z).$$

Proof. See [17, Proposition 2.4]. □

Before determining the stability threshold z_* for a specific force-gradient integrator (4.8), we provide an upper bound for z_*.

Definition 4.17. Let $\mathfrak{p}(z)$ denote the stability polynomial of a consistent force-gradient integrator (4.8). We define the value z^* as the largest real non-negative number such that $|\mathfrak{p}(z)| \leq 1$ for all $z \in [0, z^*]$.

Remark 4.18. From the proof of Proposition 4.16, it is evident that $[-z^*, z^*]$ is the largest interval including 0 such that $\boldsymbol{K}(z)$ has eigenvalues of modulus one, i.e., $z_* \leq z^*$.

Given a consistent self-adjoint force-gradient integrator (4.8), the corresponding stability polynomial takes the form $\mathfrak{p}(z) = 1 - \frac{z^2}{2} + \mathcal{O}(z^4)$, implying that $\mathfrak{p}''(0) < 0$. Consequently, $\mathfrak{p}(z)^2 - 1 \leq 0$ for sufficiently small $|z|$. In this case, z^* is the smallest real positive zero with odd multiplicity of the polynomial $\mathfrak{p}(z)^2 - 1$. Therefore, potential instabilities may occur within the interval $[0, z^*]$ at the real zeros with even multiplicity of the polynomial $\mathfrak{p}(z)^2 - 1$. The stability threshold z_* can be determined as follows.

Proposition 4.19. *Let $\boldsymbol{K}(z)$ denote the stability matrix of a self-adjoint force-gradient integrator* (4.8) *and $\mathfrak{p}(z)$ its stability polynomial. Suppose that $0 = z_0 < z_1 < \ldots < z_m$ are the real zeros with even multiplicity of the polynomial $\mathfrak{p}(z)^2 - 1$ in the interval $[0, z^*]$. Then, $z_* = z^*$ if*

$$K_{2,1}(z_\ell) = K_{1,2}(z_\ell) = 0 \tag{4.28}$$

for each $\ell = 1, \ldots, m$. Otherwise, z_* is the smallest z_ℓ that violates the condition (4.28).

Proof. See [17, Proposition 2.7]. □

Example 4.20. We revisit the force-gradient integrator (4.6). The polynomial $\mathfrak{p}(z)^2 - 1$ has no zero with even multiplicity except $z_0 = 0$. Thus, the force-gradient integrator achieves the optimal stability threshold $z_* = z^* = 2\sqrt{3}$.

4.4. Detection of Promising Integrator Variants

Given a fixed number of stages s, solving for the maximum attainable convergence order p often results in families of solutions where certain integrator coefficients remain as degrees of freedom. With the order theory and linear stability analysis, we have two distinct options for selecting the free parameters. One approach uses the degrees of freedom to minimize leading-order error terms, thereby enhancing accuracy. Alternatively, we can select the degrees of freedom to maximize the stability threshold. In addition to these quantities, we must also consider the computational cost of the numerical integration scheme to ensure a fair comparison between different integrator variants. Based on these considerations, we can derive two *efficiency measures*: an accuracy-related efficiency measure $\mathrm{Eff}^{(p)}_{\mathtt{err}}$ and a relative stability threshold $\mathrm{Eff}_{\mathtt{stab}}$.

4.4.1. An Accuracy-Related Efficiency Measure

We consider two integrators $\boldsymbol{\Phi}_h^{\{1\}}$ and $\boldsymbol{\Phi}_h^{\{2\}}$ of the same convergence order p within the classes of splitting methods, conventional force-gradient integrators, and Hessian-free force-gradient integrators. Our goal is to compare such methods and derive a criterion that allows us to identify particularly efficient variants. Both methods exhibit a local truncation error of the form

$$\tau^{\{q\}}(h) \coloneqq \frac{\boldsymbol{\Phi}_h^{\{q\}}(\boldsymbol{x}_0) - \boldsymbol{x}(t_0 + h)}{h} = h^p \boldsymbol{C}^{\{q\}} + \mathcal{O}(h^{p+1}), \quad q = 1, 2.$$

For sufficiently small step sizes, the truncation error is dominated by the leading-order error term $\boldsymbol{C}^{\{q\}}$. In general, $\boldsymbol{C}^{\{q\}}$ is non-scalar and can be expressed as a linear

combination of elementary error contributions,

$$\boldsymbol{C}^{\{q\}} = \sum_{k=1}^{n_p} c_k^{\{q\}} \boldsymbol{C}_k, \tag{4.29}$$

where the coefficients $c_k^{\{q\}}$ depend on the integrator, while the basis elements $\boldsymbol{C}_k$ are independent of the specific method but depend on the underlying system. Since we have in general no a priori information about the relative magnitudes of the error terms $\boldsymbol{C}_k$, we assume they contribute equally to the overall error and introduce an aggregate function

$$\mathrm{Err}_{p+1} \colon \mathbb{R}^{n_p} \to \mathbb{R}_{\geq 0}, \quad (c_1^{\{q\}}, \ldots, c_{n_p}^{\{q\}}) \mapsto \mathrm{Err}_{p+1}^{\{q\}},$$

which maps the vector of leading-order error coefficients to a single non-negative scalar quantity. Additionally, each integrator is characterized by a computational cost per time step, denoted by $\mathrm{cost}^{\{1\}}$ and $\mathrm{cost}^{\{2\}}$, respectively. To compare both methods at equal accuracy, we exploit the asymptotic error scaling and choose step sizes such that their leading-order error terms are expected to be equal. In particular, utilizing $\boldsymbol{\Phi}_h^{\{1\}}$ with step size h and $\boldsymbol{\Phi}_{\tilde{h}}^{\{2\}}$ with step size

$$\tilde{h} = h \cdot \left(\mathrm{Err}_{p+1}^{\{1\}} \Big/ \mathrm{Err}_{p+1}^{\{2\}} \right)^{1/p}$$

yields $h^p \mathrm{Err}_{p+1}^{\{1\}} = \tilde{h}^p \mathrm{Err}_{p+1}^{\{2\}}$. For a simulation over a fixed time interval $t = Nh$, the corresponding computational costs are

$$N \cdot \mathrm{cost}^{\{1\}} \quad \text{and} \quad N \cdot \left(\mathrm{Err}_{p+1}^{\{2\}} \Big/ \mathrm{Err}_{p+1}^{\{1\}} \right)^{1/p} \cdot \mathrm{cost}^{\{2\}}.$$

Hence, it is expected that $\boldsymbol{\Phi}_h^{\{1\}}$ achieves a prescribed error at lower cost than $\boldsymbol{\Phi}_h^{\{2\}}$, provided that

$$\frac{1}{(\mathrm{cost}^{\{1\}})^p \, \mathrm{Err}_{p+1}^{\{1\}}} > \frac{1}{(\mathrm{cost}^{\{2\}})^p \, \mathrm{Err}_{p+1}^{\{2\}}}.$$

This motivates defining the *accuracy-related efficiency measure*

$$\mathrm{Eff}_{\mathtt{err}}^{(p)} := \frac{1}{\mathrm{cost}^p \cdot \mathrm{Err}_{p+1}},$$

which we seek to maximize.

In the present setting, we choose (weighted) Euclidean norms of the leading-order error coefficients as the aggregate function. Furthermore, for Hamiltonian systems of the form (3.32), the computational cost is typically dominated by the number of force and force-gradient evaluations. We therefore model the cost as

$$\mathrm{cost} = n_f + \xi n_g,$$

where $\xi > 0$ denotes the relative cost of a force-gradient evaluation compared to a force evaluation. Following [101], we set $\xi = 2$ for conventional force-gradient integrators. In the Hessian-free framework, where a force-gradient evaluation is replaced by an additional force evaluation, we set $\xi = 1$.

Remark 4.21 (Counting evaluations)**.** The number of force evaluations n_f is increased by one for each (Hessian-free) force-gradient update with $b_i \neq 0$, while the number of gradient evaluations n_g is increased by one for each update with $c_i \neq 0$. For velocity

versions (4.9a), the final update is omitted as it coincides with the first update of the subsequent time step.

For Hessian-free force-gradient integrators, this leads to the efficiency measure

$$\mathrm{Eff}_{\mathtt{err}}^{(p)} = \frac{1}{(n_f + n_g)^p \cdot \mathrm{Err}_{p+1}}, \tag{4.30}$$

with the aggregate functions

$$\mathrm{Err}_{p+1} = \begin{cases} \sqrt{\alpha^2 + \beta^2}, & p = 2, \\ \sqrt{\sum_{i=1}^{4} \gamma_i^2 + \left(\frac{\gamma_5}{4}\right)^2}, & p = 4, \\ \sqrt{\sum_{j=1}^{10} \zeta_j^2 + \left(\frac{\zeta_{11}}{8}\right)^2 + \left(\frac{\zeta_{12}}{8}\right)^2 + \left(\frac{7\zeta_{13}}{24}\right)^2}, & p = 6. \end{cases} \tag{4.31}$$

The chosen weights are derived from the relations[1]

$$\begin{aligned} 4\gamma_4 \hat{\mathcal{V}}\hat{\mathcal{T}}\hat{\mathcal{V}}\hat{\mathcal{T}}\hat{\mathcal{V}} + \gamma_5 \tilde{\mathcal{V}}\hat{\mathcal{T}}\tilde{\mathcal{V}}\hat{\mathcal{T}}\hat{\mathcal{V}}, \\ -8\zeta_1 \hat{\mathcal{V}}\hat{\mathcal{T}}\hat{\mathcal{V}}\hat{\mathcal{T}}\hat{\mathcal{V}}\hat{\mathcal{T}}\hat{\mathcal{V}} + \zeta_{11} \tilde{\mathcal{V}}\hat{\mathcal{T}}\tilde{\mathcal{V}}\hat{\mathcal{T}}\tilde{\mathcal{V}}\hat{\mathcal{T}}\hat{\mathcal{V}}, \\ 8\zeta_6 \hat{\mathcal{T}}\hat{\mathcal{V}}\hat{\mathcal{T}}\hat{\mathcal{V}}\hat{\mathcal{T}}\hat{\mathcal{V}}\hat{\mathcal{T}} + \zeta_{12} \hat{\mathcal{T}}\tilde{\mathcal{V}}\hat{\mathcal{T}}\tilde{\mathcal{V}}\hat{\mathcal{T}}\hat{\mathcal{V}}\hat{\mathcal{T}}, \\ (-3\zeta_4 - 4\zeta_6) \hat{\mathcal{T}}\hat{\mathcal{T}}\hat{\mathcal{V}}\hat{\mathcal{T}}\hat{\mathcal{V}}\hat{\mathcal{T}}\hat{\mathcal{V}} + \zeta_{13} \hat{\mathcal{T}}\hat{\mathcal{T}}\tilde{\mathcal{V}}\hat{\mathcal{T}}\tilde{\mathcal{V}}\hat{\mathcal{T}}\hat{\mathcal{V}}, \end{aligned}$$

obtained by expanding the commutators in (4.14). By construction, this definition is consistent with the efficiency measure in [101] for conventional splitting methods ($c_i = 0$ for all $i = 1, \ldots, s$).

For a fixed number of stages s, determining the maximal attainable order p often yields families of integrators with free parameters in the integrator coefficients. In this situation, the computational cost is fixed by the chosen structure, and maximizing $\mathrm{Eff}_{\mathtt{err}}^{(p)}$ reduces to minimizing the aggregate function Err_{p+1}. The corresponding optimal choices of coefficients are referred to as *minimum-error variants*, cf. [92]. In [115], we performed a complete classification of self-adjoint Hessian-free force-gradient integrators with $s \leq 6$ stages, identifying 43 distinct variants. Whenever free parameters are present, we determine their minimum-error variants. The resulting coefficients are listed in Appendix A, while a comparative overview of all variants, including convergence orders and efficiency values, is provided in Tab. 4.2.

4.4.2. The Relative Stability Threshold

Alternatively, we compare two integrators $\mathbf{\Phi}_h^{\{1\}}$ and $\mathbf{\Phi}_h^{\{2\}}$, which may have different convergence orders. Let $z_*^{\{1\}}$ and $z_*^{\{2\}}$ denote their respective stability thresholds, and let their computational costs per time step be given by $n_f^{\{1\}} + \xi n_g^{\{1\}}$ and $n_f^{\{2\}} + \xi n_g^{\{2\}}$, respectively. For a prescribed frequency $\omega > 0$, the maximal admissible step size ensuring stability of the first integrator is

$$h_{\max}^{\{1\}} = \frac{z_*^{\{1\}}}{\omega}.$$

The second integrator attains the same computational cost over a fixed time interval when using the step size

$$\tilde{h} = \frac{z_*^{\{1\}}}{\omega} \cdot \frac{n_f^{\{2\}} + \xi n_g^{\{2\}}}{n_f^{\{1\}} + \xi n_g^{\{1\}}}.$$

[1] The weight 7/24 for ζ_{13} corresponds to the mean value of 1/3 and 1/4.

If $h_{\max}^{\{2\}} = z_*^{\{2\}}/\omega < \tilde{h}$, then the method $\boldsymbol{\Phi}_{\tilde{h}}^{\{2\}}$ is unstable. Hence, $\boldsymbol{\Phi}_h^{\{1\}}$ allows for a stable integration at lower computational cost, provided that

$$\frac{z_*^{\{2\}}}{\omega} < \frac{z_*^{\{1\}}}{\omega} \cdot \frac{n_f^{\{2\}} + \xi n_g^{\{2\}}}{n_f^{\{1\}} + \xi n_g^{\{1\}}} \quad \Leftrightarrow \quad \frac{z_*^{\{2\}}}{n_f^{\{2\}} + \xi n_g^{\{2\}}} < \frac{z_*^{\{1\}}}{n_f^{\{1\}} + \xi n_g^{\{1\}}}.$$

This naturally leads to the efficiency measure

$$\mathrm{Eff}_{\mathtt{stab}} \coloneqq \frac{z_*}{n_f + \xi n_g}, \tag{4.32}$$

which is called the *relative stability threshold* [17].

In [117], we augmented our comprehensive analysis of self-adjoint (Hessian-free) force-gradient integrators with $s \leq 6$ stages by an analysis involving the relative stability threshold (4.32). For illustrative purposes, we present results for an integrator variant with a single degree of freedom and a variant with two degrees of freedom in the integrator coefficients. Fig. 4.3 depicts results for the force-gradient integrator `ABACABA`

$$\boldsymbol{\Phi}_h = \boldsymbol{\varphi}_{a_1 h}^{\{\mathcal{T}\}} \circ \boldsymbol{\varphi}_{b_1 h}^{\{\mathcal{V}\}} \circ \boldsymbol{\varphi}_{(0.5-a_1)h}^{\{\mathcal{T}\}} \circ \boldsymbol{\varphi}_{(1-2b_1)h, c_2 h^3}^{\{\mathcal{V},\mathcal{C}\}} \circ \boldsymbol{\varphi}_{(0.5-a_1)h}^{\{\mathcal{T}\}} \circ \boldsymbol{\varphi}_{b_1 h}^{\{\mathcal{V}\}} \circ \boldsymbol{\varphi}_{a_1 h}^{\{\mathcal{T}\}}, \tag{4.33a}$$

and its Hessian-free counterpart `ABADABA`

$$\boldsymbol{\Phi}_h = \boldsymbol{\varphi}_{a_1 h}^{\{\mathcal{T}\}} \circ \boldsymbol{\varphi}_{b_1 h}^{\{\mathcal{V}\}} \circ \boldsymbol{\varphi}_{(0.5-a_1)h}^{\{\mathcal{T}\}} \circ \boldsymbol{\varphi}_{(1-2b_1)h, c_2 h^3}^{\{\mathcal{D}\}} \circ \boldsymbol{\varphi}_{(0.5-a_1)h}^{\{\mathcal{T}\}} \circ \boldsymbol{\varphi}_{b_1 h}^{\{\mathcal{V}\}} \circ \boldsymbol{\varphi}_{a_1 h}^{\{\mathcal{T}\}}, \tag{4.33b}$$

with integrator coefficients

$$a_1 = \frac{1}{2} \pm \frac{1}{\sqrt{24 b_1}}, \qquad c_2 = \frac{1}{12}\left(1 \pm \sqrt{6 b_1}(1 - b_1)\right), \tag{4.33c}$$

and one remaining free parameter $b_1 > 0$. On the one hand, the results demonstrate that both efficiency measures exhibit comparable behavior for both conventional force-gradient integrators and their Hessian-free counterparts. On the other hand, it is evident that the values for $\mathrm{Eff}_{\mathtt{err}}^{(4)}$ are already experiencing a substantial decline for minor deviations from the minimum-error variant. Conversely, the relative stability threshold $\mathrm{Eff}_{\mathtt{stab}}$ of the minimum-error variant remains reasonably high in comparison to the maximum attainable value.

We obtain comparable results for multiple degrees of freedom. For instance, the results for the force-gradient integrator `BACABACAB`

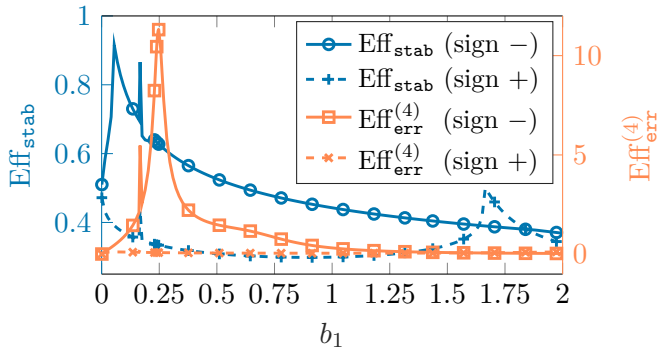
$$\begin{aligned}\boldsymbol{\Phi}_h = \boldsymbol{\varphi}_{b_1 h}^{\{\mathcal{V}\}} \circ \boldsymbol{\varphi}_{a_2 h}^{\{\mathcal{T}\}} \circ \boldsymbol{\varphi}_{b_2 h, c_2 h^3}^{\{\mathcal{V},\mathcal{C}\}} \circ \boldsymbol{\varphi}_{(0.5-a_2)h}^{\{\mathcal{T}\}} \circ \boldsymbol{\varphi}_{(1-2(b_1+b_2))h}^{\{\mathcal{V}\}} \\ \circ \boldsymbol{\varphi}_{(0.5-a_2)h}^{\{\mathcal{T}\}} \circ \boldsymbol{\varphi}_{b_2 h, c_2 h^3}^{\{\mathcal{V},\mathcal{C}\}} \circ \boldsymbol{\varphi}_{a_2 h}^{\{\mathcal{T}\}} \circ \boldsymbol{\varphi}_{b_1 h}^{\{\mathcal{V}\}},\end{aligned} \tag{4.34a}$$

and its Hessian-free counterpart `BADABADAB`

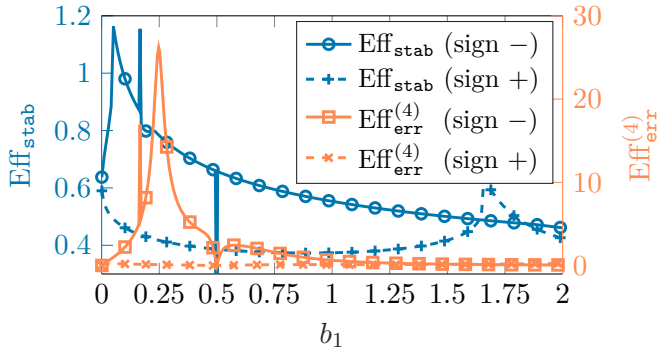
$$\begin{aligned}\boldsymbol{\Phi}_h = \boldsymbol{\varphi}_{b_1 h}^{\{\mathcal{V}\}} \circ \boldsymbol{\varphi}_{a_2 h}^{\{\mathcal{T}\}} \circ \boldsymbol{\varphi}_{b_2 h, c_2 h^3}^{\{\mathcal{D}\}} \circ \boldsymbol{\varphi}_{(0.5-a_2)h}^{\{\mathcal{T}\}} \circ \boldsymbol{\varphi}_{(1-2(b_1+b_2))h}^{\{\mathcal{V}\}} \\ \circ \boldsymbol{\varphi}_{(0.5-a_2)h}^{\{\mathcal{T}\}} \circ \boldsymbol{\varphi}_{b_2 h, c_2 h^3}^{\{\mathcal{D}\}} \circ \boldsymbol{\varphi}_{a_2 h}^{\{\mathcal{T}\}} \circ \boldsymbol{\varphi}_{b_1 h}^{\{\mathcal{V}\}},\end{aligned} \tag{4.34b}$$

with integrator coefficients

$$\begin{aligned} b_1 &= -4 b_2 a_2^2 + 4 b_2 a_2 - b_2 + \frac{1}{6}, \\ c_2 &= 4 a_2^4 b_2^2 - 4 a_2^3 b_2^2 + \frac{2 a_2^2 b_2}{3} + \frac{a_2 b_2^2}{2} - \frac{a_2 b_2}{3} + \frac{1}{144}, \end{aligned} \tag{4.34c}$$



(a) Results for the force-gradient integrator `ABACABA` (4.33a). For $b_1 = 1/6$, the method reduces to `BACAB` (4.6), explaining the respective peak.



(b) Results for the Hessian-free force-gradient integrator `ABADABA` (4.33b). For $b_1 = 1/6$, the method reduces to `BADAB` (A.9), whereas for $b_1 = 1/2$, we have $b_2 = 0$ so that the Hessian-free force-gradient update (4.10) is not defined, explaining the respective peaks.

Fig. 4.3. The efficiency measures $\mathrm{Eff}^{(4)}_{\mathtt{err}}$ and $\mathrm{Eff}_{\mathtt{stab}}$ for the (Hessian-free) force-gradient integrators (4.33) evaluated at different values of $b_1 \in (0, 2]$.

and two free parameters a_2, b_2 are depicted in Fig. 4.4. Once again, both the conventional force-gradient integrator and the Hessian-free variant exhibit similar behavior. Furthermore, the results highlight the sensitivity of the efficiency measure $\mathrm{Eff}^{(4)}_{\mathtt{err}}$ to deviations from the minimum-error coefficients derived in [101] and [115], respectively. Theoretically, it is possible to enhance the relative stability threshold $\mathrm{Eff}_{\mathtt{stab}}$ by approximately a factor of two by selecting integrator coefficients outside the depicted region. In contrast, within these stability-enhanced regions, the efficiency measure $\mathrm{Eff}^{(4)}_{\mathtt{err}}$ drops to remarkably low values. Consequently, a significantly smaller step size is required to attain the same level of accuracy as observed for the minimum-error variant. In comparison to the force-gradient integrator (4.6) and its Hessian-free counterpart (A.9), the stability-enhanced versions of (4.34) exhibit smaller values for both $\mathrm{Eff}^{(4)}_{\mathtt{err}}$ and $\mathrm{Eff}_{\mathtt{stab}}$, rendering them inefficient choices for practical computations.

For other variants of (Hessian-free) force-gradient integrators with $s \leq 6$ stages, we were able to observe a similar behavior.

4.4.3. A Set of Non-Dominated Integrator Variants

In Tab. 4.1 and Tab. 4.2, the properties of all minimum-error variants of (Hessian-free) force-gradient integrators with $s \leq 6$ stages are summarized, including the values for the efficiency measure $\mathrm{Eff}^{(p)}_{\mathtt{err}}$ and the relative stability threshold $\mathrm{Eff}_{\mathtt{stab}}$. These tables provide extended versions of the tables [101, Table 2] and [115, Table 1].

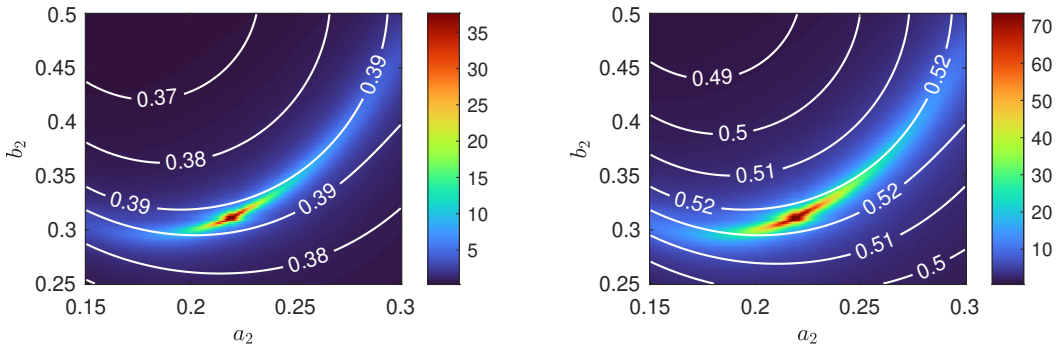
For a complete classification, we identify non-dominated integrator variants of convergence order $p \in \{2, 4, 6\}$, which guarantee that no other variant of the same convergence

	Integrator	n_f	n_g	Err_{p+1}	$\mathrm{Eff}_{\mathtt{err}}^{(p)}$	z_*	$\mathrm{Eff}_{\mathtt{stab}}$	Remarks	Eqs.	ID in [101]
$p=2$	BAB	1	0	0.0932	10.73	2.0000	2.0000	[141, 128]	(A.3)	1
	ABA	1	0	0.0932	10.73	2.0000	2.0000	[141, 128]	(A.6)	2
	CAC	1	1	0.0134	1.33	1.7791	0.5930	[101]		3
	ACA	1	1	0.00648	2.67	2.4495	0.8165	[101]		4
	BABAB	2	0	0.00855	29.24	2.5531	1.2766	[92, 101]	(A.11)	5
	ABABA	2	0	0.00855	29.24	2.5531	1.2766	[92, 101]	(A.14)	6
$p=4$	CABAC	2	1	0.00334	1.17	2.4495	0.6124	[101]		7
	BACAB	2	1	0.000713	5.47	3.4641	0.8660	[129, 28]	(4.6)	8
	CACAC	2	2	0.000595	1.30	3.2714	0.5452	[101]		9
	ACACA	2	2	0.000715	1.08	2.9269	0.4878	[129, 28]		10
	BABABAB	3	0	0.0383	0.32	1.5734	0.5245	[146, 45]	(A.19)	11
	ABABABA	3	0	0.0283	0.44	1.5734	0.5245	[146, 45]	(A.24)	12
	CABABAC	3	1	0.000855	1.87	3.0089	0.6018	[101]		13
	ABACABA	3	1	0.000141	11.31	3.1393	0.6279	[101]	(4.33a)	14
	BACACAB	3	2	0.0000443	9.40	3.1259	0.4466	[101]		15
	ACABACA	3	2	0.0000823	5.06	3.0744	0.4392	[101]		16
	CACACAC	3	3	0.0000167	9.11	3.1079	0.3453	[101]		17
	ACACACA	3	3	0.0000123	12.42	3.1167	0.3463	[101]		18
	BABABABAB	4	0	0.000654	5.97	3.4696	0.8674	[101]	(A.32)	19
	ABABABABA	4	0	0.000610	6.40	2.9894	0.7474	[101]	(A.38)	20
	BABACABAB	4	1	0.0000634	12.17	3.1131	0.5189	[101]		21
	CABABABAC	4	1	0.000294	2.63	3.0233	0.5039	[101]		22
	CABACABAC	4	2	0.00000368	66.34	3.0883	0.3860	[101]		23
	BACABACAB	4	2	0.00000649	37.60	3.1457	0.3932	[101]	(4.34a)	24
	ABACACABA	4	2	0.0000323	7.56	3.1366	0.3921	[101]		25
	ACABABACA	4	2	0.0000464	5.26	3.1036	0.3880	[101]		26
	CACABACAC	4	3	0.00000605	16.53	3.0630	0.3063	[101]		27
	ACACACACA	4	4	0.00000312	15.48	3.1277	0.2606	[101]		29
	ABABABABABA	5	0	0.0000518	30.91	2.9763	0.5953	[101]	(A.54)	31
	BABABABABAB	5	0	0.0000270	59.26	3.1421	0.6284	[101]	(A.46)	30
	CABABABABAC	5	1	0.0000165	25.22	3.1022	0.4432	[101]		32
	ABABACABABA	5	1	0.0000121	34.54	3.1110	0.4444	[101]		33
	BACABABACAB	5	2	0.00000320	47.60	3.1356	0.3484	[101]		34
	BABACACABAB	5	2	0.0000132	11.58	3.1213	0.3468	[101]		35
	ABACABACABA	5	2	0.00000127	120.03	3.1223	0.3469	[101]		36
	ACABABABACA	5	2	0.0000117	13.07	3.1127	0.3459	[101]		37
$p=6$	BACACACAB	4	3	0.00150	0.000668	2.1844	0.2184	[101]		28
	CABACACABAC	5	3	0.0000147	0.0383	3.0957	0.2814	[101]		38
	CACABABACAC	5	3	0.0000264	0.0214	2.9559	0.2687	[101]		39
	ACABACABACA	5	3	0.00000607	0.0931	3.0426	0.2766	[101]		40
	ABACACACABA	5	3	0.000146	0.00388	3.2927	0.2993	[101]		41
	BACACACACAB	5	4	0.00000366	0.0566	3.1389	0.2415	[101]		42
	ACACABACACA	5	4	0.0000139	0.0149	3.0253	0.2327	[101]		43
	CACACACACAC	5	5	0.00000249	0.0353	3.1384	0.2092	[101]		44
	ACACACACACA	5	5	0.00000299	0.0294	3.1725	0.2115	[101]		45
	ABABABABABABABA	7	0	0.0000109	0.78	3.0674	0.4382	[101]		
	BABABABABABABAB	7	0	0.00000609	1.40	3.1603	0.4515	[101]	(4.40)	

Tab. 4.1. Collection of conventional force-gradient integrators in their respective minimum-error variants. For all integrators, the number of force evaluations n_f and force-gradient evaluations n_g per time step, the leading error term Err_{p+1}, the efficiency measure $\mathrm{Eff}_{\mathtt{err}}^{(p)}$, the stability threshold z_*, the relative stability threshold $\mathrm{Eff}_{\mathtt{stab}}$, as well as a remark on the earlier appearance of the integrators, are stated. If available, references to the equations defining the respective variants are provided. The final column contains the integrator IDs utilized in [101]. Non-dominated integrator variants are highlighted via the gray shaded rows.

	Integrator	n_f	n_g	Err_{p+1}	$\mathrm{Eff}^{(p)}_{\mathtt{err}}$	z_*	$\mathrm{Eff}_{\mathtt{stab}}$	Remarks	Eqs.	ID in [115]
$p=2$	BAB	1	0	0.0932	10.73	2.0000	2.0000	[141, 128]	(A.3)	1
	ABA	1	0	0.0932	10.73	2.0000	2.0000	[141, 128]	(A.6)	2
	DAD	1	1	0.0833	3.00	1.7791	0.8895	[101, 62]	(A.2)	3
	ADA	1	1	0.0417	6.00	2.4495	1.2247	[101]	(A.5)	4
	BABAB	2	0	0.00855	29.24	2.5531	1.2766	[92, 101]	(A.11)	5
	ABABA	2	0	0.00855	29.24	2.5531	1.2766	[92, 101]	(A.14)	6
$p=4$	DABAD	2	1	0.00335	3.68	2.4495	0.8165	[101]	(A.10)	7
	BADAB	2	1	0.000728	16.96	3.4641	1.1547	[129, 28, 145]	(A.9)	8
	ADADA	2	2	0.000718	5.44	2.9269	0.7317	[129, 28]	(A.13)	10
	DADAD	2	2	0.000625	6.25	3.2821	0.8205	[115]	(A.8)	9
	BABABAB	3	0	0.0383	0.32	1.5734	0.5245	[146, 45]	(A.19)	11
	ABABABA	3	0	0.0283	0.44	1.5734	0.5245	[146, 45]	(A.24)	12
	DABABAD	3	1	0.000891	4.38	2.9829	0.7457	[115]	(A.18)	13
	ABADABA	3	1	0.000149	26.19	3.1377	0.7844	[115]	(A.22)	14
	ADABADA	3	2	0.0000844	18.95	3.0745	0.6149	[115]	(A.23)	16
	BADADAB	3	2	0.0000498	32.12	3.1261	0.6252	[115]	(A.17)	15
	DADADAD	3	3	0.0000275	28.09	3.1082	0.5180	[115]	(A.16)	17
	ADADADA	3	3	0.0000200	38.57	3.1157	0.5193	[115]	(A.21)	18
	BABABABAB	4	0	0.000654	5.97	3.4696	0.8674	[101]	(A.32)	19
	ABABABABA	4	0	0.000610	6.40	2.9894	0.7474	[101]	(A.38)	20
	DABABABAD	4	1	0.000336	4.76	3.0907	0.6181	[115]	(A.31)	22
	BABADABAB	4	1	0.0000651	24.57	3.1123	0.6225	[115]	(A.29)	21
	ADABABADA	4	2	0.0000471	16.39	3.1039	0.5173	[115]	(A.37)	26
	ABADADABA	4	2	0.0000346	22.32	3.1380	0.5230	[115]	(A.36)	25
	DABADABAD	4	2	0.0000130	59.33	3.0851	0.5142	[115]	(A.27)	23
	BADABADAB	4	2	0.0000105	73.45	3.1457	0.5243	[115]	(A.30)	24
	DADABADAD	4	3	0.0000101	41.06	3.1340	0.4477	[115]	(A.28)	27
	ADADADADA	4	4	0.00000501	48.71	3.1265	0.3908	[115]	(A.35)	29
	ABABABABABA	5	0	0.0000518	30.91	2.9763	0.5953	[101]	(A.54)	31
	BABABABABAB	5	0	0.0000270	59.26	3.1421	0.6284	[101]	(A.46)	30
	DABABABABAD	5	1	0.0000166	46.47	3.0999	0.5166	[115]	(A.45)	32
	ABABADABABA	5	1	0.0000154	50.09	3.1050	0.5175	[115]	(A.51)	33
	BABADADABAB	5	2	0.0000189	21.98	3.1289	0.4470	[115]	(A.43)	35
	ADABABABADA	5	2	0.0000128	32.64	3.1130	0.4447	[115]	(A.53)	37
	BADABABADAB	5	2	0.00000520	80.13	3.1371	0.4482	[115]	(A.44)	34
	ABADABADABA	5	2	0.00000445	93.60	3.1239	0.4463	[115]	(A.52)	36
	DADABABADAD	5	3	0.00000519	47.08	3.1355	0.3919	[115]	(A.42)	39
	DABADADABAD	5	3	0.00000355	68.84	3.1381	0.3923	[115]	(A.41)	38
	ADABADABADA	5	3	0.00000318	76.79	3.1358	0.3920	[115]	(A.49)	40
	ADADABADADA	5	4	0.00000235	64.99	3.1283	0.3476	[115]	(A.50)	43
$p=6$	BADADADAB	4	3	0.00154	0.0055	2.1844	0.3121	[101]	(A.26)	28
	BADADADADAB	5	4	0.00000699	0.27	3.0727	0.3414	[115]	(A.40)	42
	ADADADADADA	5	5	0.00000603	0.17	3.0806	0.3081	[115]	(A.48)	45
	ABABABABABABABA	7	0	0.0000109	0.78	3.0674	0.4382	[101]		
	BABABABABABABAB	7	0	0.00000609	1.40	3.1603	0.4515	[101]	(4.40)	

Tab. 4.2. Collection of Hessian-free force-gradient integrators in their respective minimum-error variants. For all integrators, the number of force evaluations n_f and force-gradient evaluations n_g per time step, the leading error term Err_{p+1}, the efficiency measure $\mathrm{Eff}^{(p)}_{\mathtt{err}}$, the stability threshold z_*, the relative stability threshold $\mathrm{Eff}_{\mathtt{stab}}$, as well as a remark on earlier appearances of the integrators, are stated. The final two columns provide references to the equations defining the respective variants and the integrator IDs utilized in [115]. Non-dominated integrator variants are highlighted via the gray shaded rows.



(a) BACABACAB (4.34a). **(b)** BADABADAB (4.34b).

Fig. 4.4. The efficiency measure $\mathrm{Eff}^{(4)}_{\mathtt{err}}$ for the (Hessian-free) force-gradient integrators (4.34) evaluated at different values of $a_2 \in [0.15, 0.3]$ and $b_2 \in [0.25, 0.5]$. The white contour lines indicate the values for the relative stability threshold $\mathrm{Eff}_{\mathtt{stab}}$.

order achieves simultaneously higher values for both efficiency measures $\mathrm{Eff}^{(p)}_{\mathtt{err}}$ and $\mathrm{Eff}_{\mathtt{stab}}$.

Convergence Order Two

According to the two efficiency measures, the class of conventional splitting methods proves to provide the most efficient choices of second order. In particular, the Strang splitting [128] variants BAB (A.3) and ABA (A.6), as well as the three-stage splitting methods BABAB

$$\boldsymbol{\Phi}_h = \boldsymbol{\varphi}^{\{\mathcal{V}\}}_{\lambda h} \circ \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{h/2} \circ \boldsymbol{\varphi}^{\{\mathcal{V}\}}_{(1-2\lambda)h} \circ \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{h/2} \circ \boldsymbol{\varphi}^{\{\mathcal{V}\}}_{\lambda h} \tag{4.35a}$$

and ABABA

$$\boldsymbol{\Phi}_h = \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{\lambda h} \circ \boldsymbol{\varphi}^{\{\mathcal{V}\}}_{h/2} \circ \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{(1-2\lambda)h} \circ \boldsymbol{\varphi}^{\{\mathcal{V}\}}_{h/2} \circ \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{\lambda h} \tag{4.35b}$$

with one degree of freedom $\lambda \in \mathbb{R}$ constitute promising choices. For the Strang splittings (A.3) and (A.6), we obtain values of $\mathrm{Eff}^{(2)}_{\mathtt{err}} \approx 10.73$ and $\mathrm{Eff}_{\mathtt{stab}} = 2$. Note that a relative stability threshold of $\mathrm{Eff}_{\mathtt{stab}} = 2$ is the maximum possible value for splitting methods [17]. For the three-stage integrators (4.35), we obtain

$$\mathrm{Eff}^{(2)}_{\mathtt{err}}(\lambda) = \frac{6}{(144\lambda^4 - 288\lambda^3 + 228\lambda^2 - 60\lambda + 5)^{1/2}}, \tag{4.36}$$

$$\mathrm{Eff}_{\mathtt{stab}}(\lambda) = \begin{cases} 1/(1-2\lambda)^{1/2}, & \lambda \in \{x \in (-\infty, \frac{1}{4}) \,|\, x \neq 0\}, \\ 1/(2\lambda)^{1/2}, & \lambda \in \{x \in (\frac{1}{4}, \infty) \,|\, x \neq \frac{1}{2}\}, \\ 1/(1-2\lambda)^{1/2}, & \lambda = 0, \\ 2, & \lambda = \frac{1}{4}, \\ 2/(2\lambda)^{1/2}, & \lambda = \frac{1}{2}. \end{cases} \tag{4.37}$$

A visualization of both efficiency measures is presented in Fig. 4.5. The minimum-error variant of the two integrators yields the optimal coefficient [92]

$$\lambda_{\mathtt{err}} = \frac{1}{2} - \frac{(2\sqrt{326} + 36)^{1/3}}{12} + \frac{1}{6(2\sqrt{326} + 36)^{1/3}} \approx 0.1931833275037836. \tag{4.38}$$

Conversely, the relative stability threshold is maximized for $\lambda_{\mathtt{stab}} = 0.25$, where the variants reduce to two applications of the respective Strang splitting. We also obtain the optimal value $\mathrm{Eff}_{\mathtt{stab}} = 2$ for $\lambda \in \{0, 0.5\}$ because, for these values, the splitting methods (4.35) reduce to BAB and ABA, respectively. The interval $\lambda \in [\lambda_{\mathtt{err}}, \lambda_{\mathtt{stab}}]$ encompasses

all non-dominated variants of the integrators (4.35). Depending on the specific system, selecting a value within this interval may lead to the most efficient computational process by providing the optimal trade-off between accuracy and numerical stability. Particularly, for increasingly stiff systems, a value $\lambda > \lambda_{\texttt{err}}$ should be preferred. However, it should be noted that, depending on the specific system, the Lie brackets may be vastly different in magnitude so that the efficiency measure $\mathrm{Eff}_{\texttt{err}}^{(p)}$ becomes inaccurate. Using appropriate system-dependent weights for the Lie brackets and minimizing a weighted norm instead will then result in a different optimal value $\lambda_{\texttt{err}}$.

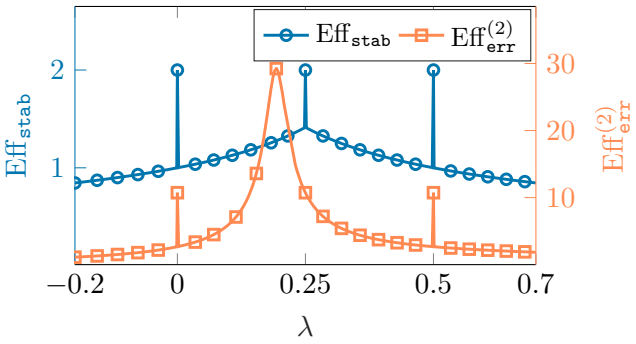


Fig. 4.5. The efficiency measure $\mathrm{Eff}_{\texttt{err}}^{(2)}(\lambda)$ and the relative stability threshold $\mathrm{Eff}_{\texttt{stab}}(\lambda)$ for the splitting methods `BABAB` (4.35a) and `ABABA` (4.35b) evaluated at different values of $\lambda \in [-0.2, 0.7]$.

Convergence Order Four

Most variants of (Hessian-free) force-gradient integrators with $s \leq 6$ stages exhibit convergence order $p = 4$. In [101], there are 24 different variants of conventional force-gradient integrators of order $p = 4$, whereas the Hessian-free framework has 28 different variants with $s \leq 6$ stages and $p = 4$ [115]. Moreover, there are 6 different variants of conventional splitting methods of convergence order $p = 4$ with $s \leq 6$ stages. As these integrators have up to four degrees of freedom in the integrator coefficients, it becomes a computationally demanding task to explore the entire family of solutions. To obtain numerical evidence, we evaluated both measures $\mathrm{Eff}_{\texttt{err}}^{(4)}$ and $\mathrm{Eff}_{\texttt{stab}}$ on a discrete grid with values $a_i, b_i \in [-2, 2]$ and $c_i \in [-0.1, 0.1]$ for all degrees of freedom.

Considering the framework of conventional force-gradient integrators, we find five non-dominated integrator variants of order $p = 4$ with $s \leq 6$ stages. These integrators are highlighted in gray in the overview in Tab. 4.1. It is noteworthy that three of the five integrators are conventional splitting methods, while the force-gradient integrators are particularly interesting due to their high accuracy at the cost of less numerical stability. However, it is important to note that the force-gradient integrator `BACAB` (4.6) exhibits very similar values to `BABABABAB` (A.32) (see Tab. 4.1). When the relative cost of a force-gradient evaluation satisfies $\xi < 2$, the force-gradient integrator `BACAB` (4.6) exhibits larger values for both efficiency measures, making it a promising choice that is particularly advantageous in terms of the relative stability threshold $\mathrm{Eff}_{\texttt{stab}}$. Furthermore, the variants `ABACABA` (4.33a) and `BACABACAB` (4.34a) are dominated by the splitting method `BABABABABAB` due to the choice $\xi = 2$. For $\xi < 2$, the aforementioned variants may turn out as non-dominated variants as well.

In contrast, for the Hessian-free framework, we identified six non-dominated integrators, which are highlighted in gray in Tab. 4.2. Notably, five of these integrators are indeed Hessian-free force-gradient integrators, highlighting their efficiency in terms of both accuracy and numerical stability. The integrator `BADABABADAB` (A.44) is a non-dominated variant, but it exhibits a significantly smaller value for $\mathrm{Eff}_{\texttt{err}}^{(4)}$ and only a slightly greater value for $\mathrm{Eff}_{\texttt{stab}}$ compared to `ABADABADABA` (A.52). Therefore, preference should be

given to ABADABADABA. We did not compare conventional force-gradient integrators with their Hessian-free counterparts as the efficiency of these integrators depends on the value of ξ in (4.30) and whether the force-gradient term is available in practical software implementations.

For both frameworks, it is possible to attain slightly improved values for $\mathrm{Eff}_{\mathtt{stab}}$ by deviating slightly from the minimum-error coefficients. However, these alterations entail a substantial reduction in values for $\mathrm{Eff}^{(4)}_{\mathtt{err}}$, as illustrated, for instance, in Fig. 4.4. Consequently, we propose to continue optimizing the efficiency measure $\mathrm{Eff}^{(4)}_{\mathtt{err}}$ and select one of the non-dominated variants highlighted in Tab. 4.1 and Tab. 4.2, respectively. Selecting a variant with a higher value for $\mathrm{Eff}^{(4)}_{\mathtt{err}}$ corresponds to a smaller value for $\mathrm{Eff}_{\mathtt{stab}}$, and vice versa. Depending on the specific system and the prescribed error tolerances, one variant will provide the optimal trade-off between accuracy and stability, resulting in the most efficient computational process.

Remark 4.22. Next to the integrator variants analyzed in [101, 115], a commonly employed fourth-order splitting method that relies on the composition approach (3.15) is obtained by choosing the coefficients [13]

$$\begin{aligned} \beta_1 &= \alpha_6 = 0.162314550766870, \quad \alpha_1 = \beta_6 = 0.0829844064174052, \\ \beta_2 &= \alpha_5 = 0.370877414979582, \quad \alpha_2 = \beta_5 = 0.233995250731498, \\ \beta_3 &= \alpha_4 = 0.059762097006575, \quad \alpha_3 = \beta_4 = -0.409933719901930, \end{aligned} \tag{4.39}$$

and the Lie–Trotter splitting $\boldsymbol{\Phi}_h = \boldsymbol{\varphi}_h^{\{1\}} \circ \boldsymbol{\varphi}_h^{\{2\}}$ as the underlying base scheme. The resulting method is characterized by the values

$$\mathrm{Err}_5 \approx 0.000183, \quad \mathrm{Eff}^{(4)}_{\mathtt{err}} \approx 4.23, \quad z_* \approx 3.1635, \quad \mathrm{Eff}_{\mathtt{stab}} \approx 0.5272,$$

and hence constitutes a dominated integrator variant.

Convergence Order Six

The minimum-error variant of the 8-stage velocity version BABABABABABABAB [101]

$$\begin{aligned} \boldsymbol{\Phi}_h = \boldsymbol{\varphi}^{\{\mathcal{V}\}}_{b_1 h} \circ \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{a_2 h} \circ \boldsymbol{\varphi}^{\{\mathcal{V}\}}_{b_2 h} \circ \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{a_3 h} \circ \boldsymbol{\varphi}^{\{\mathcal{V}\}}_{b_3 h} \circ \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{a_4 h} \circ \boldsymbol{\varphi}^{\{\mathcal{V}\}}_{(0.5-(b_1+b_2+b_3))h} \\ \circ \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{(1-2(a_2+a_3+a_4))h} \circ \boldsymbol{\varphi}^{\{\mathcal{V}\}}_{(0.5-(b_1+b_2+b_3))h} \circ \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{a_4 h} \circ \boldsymbol{\varphi}^{\{\mathcal{V}\}}_{b_3 h} \circ \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{a_3 h} \\ \circ \boldsymbol{\varphi}^{\{\mathcal{V}\}}_{b_2 h} \circ \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{a_2 h} \circ \boldsymbol{\varphi}^{\{\mathcal{V}\}}_{b_1 h}, \end{aligned} \tag{4.40a}$$

with coefficients

$$\begin{aligned} a_2 &= 0.2465881872786138, \quad b_1 = 0.0833333333333333, \\ a_3 &= 0.6047073875057809, \quad b_2 = 0.3977675859548440, \\ a_4 &= -0.4009869039788007, \quad b_3 = -0.0393336931446257, \end{aligned} \tag{4.40b}$$

and $\mathrm{Eff}^{(6)}_{\mathtt{err}} \approx 1.40$, $\mathrm{Eff}_{\mathtt{stab}} \approx 0.4515$, turns out to dominate all sixth-order self-adjoint (Hessian-free) force-gradient integrators with $s \leq 6$ stages (see Tab. 4.1 and Tab. 4.2 for a comparison to the other variants). The investigations in [101, 115] only consider (Hessian-free) force-gradient integrators with $s \leq 6$ stages. For $s > 6$, there exist more variants of sixth-order integrators that may offer additional non-dominated alternatives. However, these variants have not yet been investigated.

Remarks

Although investigations in [101] and [115] did not explicitly address numerical stability, the results indicate that maximizing the efficiency measure $\mathrm{Eff}^{(p)}_{\mathtt{err}}$, resulting in minimum-error variants, yields promising integrator variants that provide the optimal trade-off between accuracy and numerical stability. By incorporating the relative stability threshold $\mathrm{Eff}_{\mathtt{stab}}$ proposed in [117] into these investigations, we can identify efficient integrator variants that excel in both accuracy and numerical stability.

For convergence order $p = 4$, in particular, the findings demonstrate remarkable efficiency of the force-gradient approach. For convergence order $p = 6$, (Hessian-free) force-gradient integrators with $s \leq 6$ stages do not contain any variant that surpasses the splitting method (4.40). Analyzing (Hessian-free) force-gradient integrators with $s \geq 7$ stages may potentially yield promising variants, which require further investigation.

4.5. Numerical Results

We illustrate the practical performance of the Hessian-free force-gradient integrators presented in this chapter and compare them with conventional splitting methods. To this end, we consider a representative N-body problem arising in celestial mechanics, namely the outer solar system, which can be modeled as a separable Hamiltonian system of the form (3.32). In particular, we consider the Hamiltonian [61]

$$\mathcal{H}(\boldsymbol{p},\boldsymbol{q}) = \frac{1}{2}\sum_{i=0}^{5}\frac{1}{m_i}\boldsymbol{p}_i^\top\boldsymbol{p}_i - G\sum_{i=1}^{5}\sum_{j=0}^{i-1}\frac{m_i m_j}{\|\boldsymbol{q}_i-\boldsymbol{q}_j\|_2},$$

describing the motion of the four outer planets (and Pluto) relative to the sun. Here, the momenta $\boldsymbol{p}$ and positions $\boldsymbol{q}$ are supervectors composed of the vectors $\boldsymbol{p}_i, \boldsymbol{q}_i \in \mathbb{R}^3$. The parameters are taken from [61], and the initial data from [24], corresponding to September 5, 1994 at midnight. Numerical solutions of the outer solar system obtained with the Hessian-free force-gradient integrator (A.52) using a step size $h = 200$ over a time period of 200,000 days are depicted in Fig. 4.6, showing the correct qualitative behavior. Numerical results for other variants of Hessian-free force-gradient integrators exhibit the same behavior, as expected.

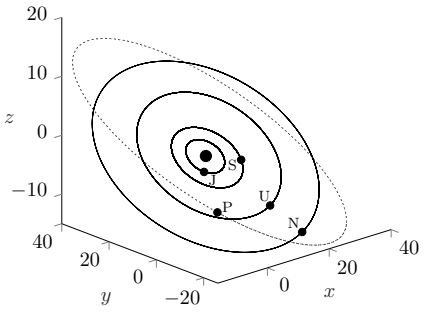


Fig. 4.6. Outer solar system. Numerical solution using the Hessian-free force-gradient integrator `ABADABADABA` (A.52) with $h = 200$ over a time period of 200,000 days.

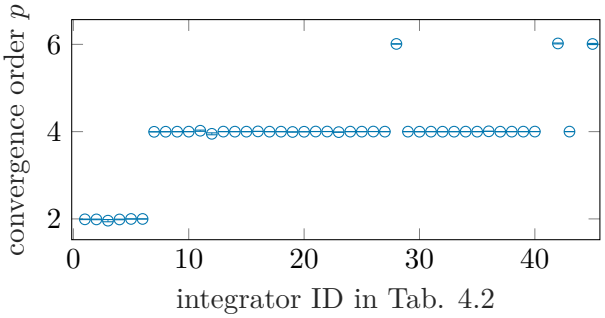


Fig. 4.7. Outer solar system. Numerical verification of the convergence order p of Hessian-free force-gradient integrators. For each variant, the order is measured using simulations with different step sizes. The plot displays the mean and standard deviation of the order p.

The results in Fig. 4.7 confirm that all Hessian-free force-gradient integrators attain the convergence orders reported in Tab. 4.2. These experiments primarily serve as a

consistency check, verifying the correctness of the derived order conditions and their solution. Furthermore, the work-precision diagram in Fig. 4.8 illustrates the efficiency of the integrators. For visualization purposes, only a selection of integrators is depicted: the most efficient splitting method for each number of stages, the commonly used Hessian-free force-gradient integrator `BADAB` (A.9), and the three best-performing Hessian-free force-gradient integrators of order $p = 4$: `BABADABAB` (A.29), `DABADABAD` (A.27), and `ABADABADABA` (A.52). Compared to the most efficient fourth-order splitting method `BABABABABAB` (A.46), the class of Hessian-free force-gradient integrators contains variants reducing the error by more than one order of magnitude for the same number of force evaluations, highlighting the efficiency of the force-gradient approach. For accuracy demands up to 10^{-4}, the Hessian-free force-gradient integrator (A.52) is even competitive with the sixth-order velocity version (4.40).

The numerical results also illustrate that the practical efficiency may deviate from the theoretical efficiency measure $\mathrm{Eff}^{(p)}_{\mathtt{err}}$. For instance, (A.9) ($\mathrm{Eff}^{(4)}_{\mathtt{err}} \approx 16.96$) outperforms (A.46) ($\mathrm{Eff}^{(4)}_{\mathtt{err}} \approx 59.26$), and (A.27) ($\mathrm{Eff}^{(4)}_{\mathtt{err}} \approx 59.33$) performs comparably to (A.52) ($\mathrm{Eff}^{(4)}_{\mathtt{err}} \approx 93.60$). This discrepancy arises because $\mathrm{Eff}^{(p)}_{\mathtt{err}}$ is based on the simplifying assumption that all leading-order error terms contribute equally. Nevertheless, it provides a valuable heuristic for identifying efficient integrator variants without requiring any further assumptions on the specific system.

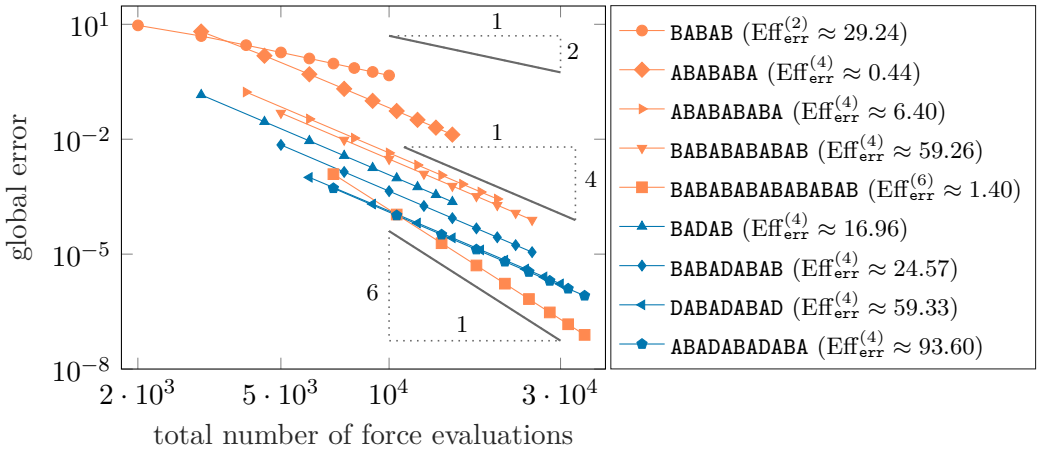


Fig. 4.8. Outer solar system. Global error vs. number of force evaluations for selected Hessian-free force-gradient integrators (blue) and splitting methods (orange). The simulations are performed over a time period of 200,000 days using different numbers of steps, $n_{\mathrm{steps}} \in \{1000, 1500, \ldots, 5000\}$.

Another important aspect in the numerical simulation of Hamiltonian systems is long-time energy behavior. Although Hessian-free force-gradient integrators are not symplectic and therefore do not preserve a modified Hamiltonian, the relative energy error

$$\frac{\mathcal{H}(\boldsymbol{p}_n, \boldsymbol{q}_n) - \mathcal{H}(\boldsymbol{p}_0, \boldsymbol{q}_0)}{\mathcal{H}(\boldsymbol{p}_0, \boldsymbol{q}_0)}$$

shown in Fig. 4.9 for the integrator `ABADABADABA` (A.52) exhibits no visible drift. In this regard, the method behaves similarly to a symplectic integrator and demonstrates excellent long-term energy behavior. This observation is consistent with the observations made in [62] for another N-body problem.

4.6. Summary

In this chapter, we introduced the class of Hessian-free force-gradient integrators and presented the order theory and linear stability analysis developed in the papers [115,

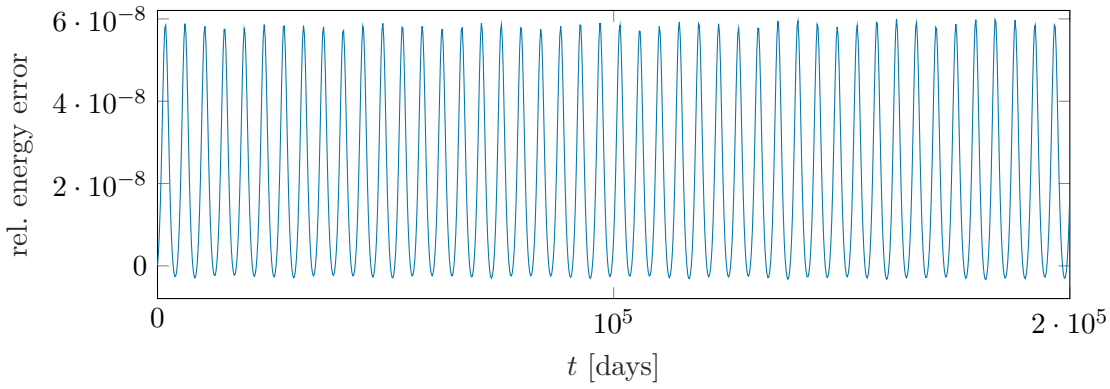


Fig. 4.9. Outer solar system. Relative energy error of the Hessian-free force-gradient integrator (A.52) using $h = 200$ over a time period of 200,000 days.

116, 117]. In contrast to conventional force-gradient integrators, their Hessian-free counterparts do not demand the Hessian $\nabla^2\mathcal{V}$ of the potential. This makes the methods straightforward to implement in existing software packages. We conducted a detailed error analysis and used the order conditions to perform a complete classification of Hessian-free force-gradient integrators with $s \leq 6$ stages. Investigations of the geometric properties showed that self-adjoint Hessian-free force-gradient integrators remain time-reversible and volume-preserving, but they are generally no longer symplectic. The accuracy investigations have been augmented by a linear stability analysis that also applies to conventional force-gradient integrators. By defining an accuracy-related efficiency measure $\mathrm{Eff}^{(p)}_{\mathtt{err}}$ and the relative stability threshold $\mathrm{Eff}_{\mathtt{stab}}$, we were able to detect non-dominated integrator variants, which guarantee that no other variant of the same convergence order achieves simultaneously higher values for both efficiency measures $\mathrm{Eff}^{(p)}_{\mathtt{err}}$ and $\mathrm{Eff}_{\mathtt{stab}}$. Numerical results for an N-body problem demonstrated the superior efficiency of Hessian-free force-gradient integrators compared to conventional splitting methods (of RKN type).

Chapter 5.

A Hierarchical Splitting Approach for N-Split Systems

In the preceding chapters, we focused on the derivation of splitting methods for two-split systems (3.2), and in particular for separable Hamiltonian systems of the form (3.32). In this chapter, we will discuss a *hierarchical splitting approach* to the design of splitting methods for general N-split systems (3.1). The core idea of the approach is to iteratively apply splitting methods for two-split systems such that we can re-use the integrators developed in the preceding chapters. Based on [118], this chapter provides a general introduction to hierarchical splitting methods. Possible extensions of hierarchical splitting methods are discussed in the subsequent chapters. In particular, Chapter 6 discusses a multirate extension of hierarchical splitting methods, whereas Chapter 7 once more considers separable Hamiltonian systems and discusses how to combine the hierarchical splitting approach with (Hessian-free) force-gradient integrators.

5.1. Splitting Methods for N-split Systems

When considering general N-split systems (3.1), the definition of an s-stage splitting method generalizes as follows.

Definition 5.1 (s-stage splitting method for N-split systems)**.** A single time step of step size h of an s-stage splitting method $\boldsymbol{\Phi}_h$ advances the given initial value $\boldsymbol{x}(0) = \boldsymbol{x}_0$ to a numerical approximation $\boldsymbol{x}_1 \coloneqq \boldsymbol{\Phi}_h(\boldsymbol{x}_0) \approx \boldsymbol{x}(h)$ and is given as

$$\boldsymbol{\Phi}_h = \boldsymbol{\varphi}^{\{N\}}_{\alpha_s^{\{N\}}h} \circ \boldsymbol{\varphi}^{\{N-1\}}_{\alpha_s^{\{N-1\}}h} \circ \ldots \circ \boldsymbol{\varphi}^{\{1\}}_{\alpha_s^{\{1\}}h} \circ \ldots \circ \boldsymbol{\varphi}^{\{N\}}_{\alpha_1^{\{N\}}h} \circ \boldsymbol{\varphi}^{\{N-1\}}_{\alpha_1^{\{N-1\}}h} \circ \ldots \circ \boldsymbol{\varphi}^{\{1\}}_{\alpha_1^{\{1\}}h}, \tag{5.1}$$

where $\alpha_i^{\{m\}} \in \mathbb{R}$ denote the integrator coefficients of the splitting method.

The Lie–Trotter splitting (3.3) and the Strang splitting (3.5) can be generalized to N-split systems in a straightforward way. In particular, the generalization of the Lie–Trotter splitting reads

$$\boldsymbol{\Phi}_h = \boldsymbol{\varphi}^{\{N\}}_h \circ \ldots \circ \boldsymbol{\varphi}^{\{2\}}_h \circ \boldsymbol{\varphi}^{\{1\}}_h, \tag{5.2}$$

whereas the Strang splitting generalizes to

$$\boldsymbol{\Phi}_h = \boldsymbol{\varphi}^{\{1\}}_{h/2} \circ \boldsymbol{\varphi}^{\{2\}}_{h/2} \circ \ldots \circ \boldsymbol{\varphi}^{\{N-1\}}_{h/2} \circ \boldsymbol{\varphi}^{\{N\}}_h \circ \boldsymbol{\varphi}^{\{N-1\}}_{h/2} \circ \ldots \circ \boldsymbol{\varphi}^{\{2\}}_{h/2} \circ \boldsymbol{\varphi}^{\{1\}}_{h/2}. \tag{5.3}$$

The state-of-the-art approaches to the design of higher-order splitting methods for N-split systems rely on composition approaches of the form (3.11) and (3.15). Compositions of the form (3.11) and (3.15) designed for $N = 2$ partitions and for general N-split problems have the same order [92], but the optimization of the integrator coefficients may differ. Nevertheless, coefficients optimized for two-split systems can reduce the error significantly [93] compared to generic composition approaches such as the triple jump composition (3.13) or Suzuki's fractals (3.14). One such optimized set of coefficients resulting in a fourth-order composition method is given by (4.39). As discussed in Remark 4.22, this composition technique does not constitute a particularly efficient

integrator variant of RKN type according to the efficiency measures constructed in the preceding section.

Except for the compositions (3.11) and (3.15) based on the Strang splitting (5.3) and the Lie–Trotter splitting (5.2), there is currently no well-established approach for designing splitting methods that can be adapted to a varying number of partitions N.

Furthermore, the consideration of $N > 2$ partitions encompasses an additional drawback. As discussed in Chapter 3, splitting methods (3.8) for two-split systems are equivalent to composition schemes of the form (3.15) by virtue of the relationship (3.18). This equivalence no longer holds for $N > 2$ partitions. Hence, the state-of-the-art approaches only encompass a subclass of splitting methods for N-split systems.

Deriving splitting methods via the BCH formula enables the exploration of the entire class of splitting methods for N-split systems, potentially yielding more efficient integrator variants. Strictly speaking, the BCH formula is formulated for $N = 2$ partitions. In [127], a generalization of the BCH formula to $N \geq 2$ partitions has been presented. This order theory shares the same limitation as the order theory of alternative methods for partitioned systems such as GARK methods [110]. For N-split systems, the number of order conditions increases significantly with N, turning the construction of higher-order methods into a computationally challenging task. Given a number of partitions N, we can solve the nonlinear system of order conditions up to order p, resulting in a set of integrator coefficients for that particular N. However, we cannot in general re-use the obtained set of coefficients if we want to extend the method to $N' > N$ partitions.

Splitting methods for two-split systems have been extensively studied in the literature such that very efficient integrator variants are available, see e.g. [94, 18] and the references therein. Moreover, we performed a complete classification of Hessian-free force-gradient integrators with $s \leq 6$ stages in Chapter 4, and a similar classification has been done for conventional force-gradient integrators in [101]. We aim to re-use these methods to design splitting methods for N-split systems with a *hierarchical splitting approach*. The resulting class of methods constitutes a larger subclass of splitting methods for N-split systems that contains the aforementioned composition approaches as special cases.

5.2. Hierarchical Splitting Methods

Let $\boldsymbol{f}^{\{v\}}(\boldsymbol{x}) \coloneqq \sum_{m\in\boldsymbol{v}} \boldsymbol{f}^{\{m\}}(\boldsymbol{x})$ for any subset $\boldsymbol{v} \subset \{1,\dots,N\}$. The Lie derivative of the subsystem $\dot{\boldsymbol{x}} = \boldsymbol{f}^{\{v\}}(\boldsymbol{x})$ is denoted by $\mathcal{L}^{\{v\}}$, i.e., we can formally express the flow of $\dot{\boldsymbol{x}} = \boldsymbol{f}^{\{v\}}(\boldsymbol{x})$ as $\boldsymbol{\varphi}_t^{\{v\}} = \exp(t\mathcal{L}^{\{v\}})\mathrm{Id}$.

5.2.1. Iterative Splitting of the Vector Field

Let $\boldsymbol{r} \coloneqq \{1,\dots,N\}$. We start the iteration by splitting the vector field $\boldsymbol{f} = \boldsymbol{f}^{\{r\}}$ into

$$\boldsymbol{f}^{\{r\}}(\boldsymbol{x}) = \boldsymbol{f}^{\{i\}}(\boldsymbol{x}) + \boldsymbol{f}^{\{j\}}(\boldsymbol{x}), \quad \boldsymbol{i},\boldsymbol{j} \subset \{1,\dots,N\}, \quad \boldsymbol{i} \dot{\cup} \boldsymbol{j} = \boldsymbol{r}.$$

We compute a numerical approximation of the flow $\boldsymbol{\varphi}_t^{\{r\}} \coloneqq \boldsymbol{\varphi}_t$ of (3.1) with a splitting method for two-split systems,

$$\boldsymbol{\Phi}_h^{\{r\}} = \boldsymbol{\varphi}^{\{j\}}_{b_{s^{\{r\}}}^{\{r\}}h} \circ \boldsymbol{\varphi}^{\{i\}}_{a_{s^{\{r\}}}^{\{r\}}h} \circ \dots \circ \boldsymbol{\varphi}^{\{j\}}_{b_1^{\{r\}}h} \circ \boldsymbol{\varphi}^{\{i\}}_{a_1^{\{r\}}h}.$$

To continue the iterative splitting, both $\boldsymbol{f}^{\{i\}}$ and $\boldsymbol{f}^{\{j\}}$ must be split further. Taking $\boldsymbol{f}^{\{i\}}$ as an example, two cases arise. If $|\boldsymbol{i}| = 1$, we do not want to further split the vector field $\boldsymbol{f}^{\{i\}}$. For this *elementary subsystem*, we either compute $\boldsymbol{\varphi}_h^{\{i\}}$ exactly, or we numerically approximate the flow with a dedicated numerical method $\boldsymbol{\Phi}_h^{\{i\}}$, such as a RK method. If

$|\boldsymbol{i}| > 1$, we further split the vector field $\boldsymbol{f}^{\{\boldsymbol{i}\}}$ into two parts $\boldsymbol{i} = \boldsymbol{v} \dot\cup \boldsymbol{w}$ and apply another splitting method for two-split systems which replaces $\boldsymbol{\varphi}^{\{\boldsymbol{i}\}}_{a_j^{\{\boldsymbol{r}\}}h}$ with

$$\boldsymbol{\Phi}^{\{\boldsymbol{i}\}}_{a_j^{\{\boldsymbol{r}\}}h} = \boldsymbol{\varphi}^{\{\boldsymbol{w}\}}_{a_j^{\{\boldsymbol{r}\}}b^{\{\boldsymbol{i}\}}_{s^{\{\boldsymbol{i}\}}}h} \circ \boldsymbol{\varphi}^{\{\boldsymbol{v}\}}_{a_j^{\{\boldsymbol{r}\}}a^{\{\boldsymbol{i}\}}_{s^{\{\boldsymbol{i}\}}}h} \circ \ldots \circ \boldsymbol{\varphi}^{\{\boldsymbol{w}\}}_{a_j^{\{\boldsymbol{r}\}}b^{\{\boldsymbol{i}\}}_1 h} \circ \boldsymbol{\varphi}^{\{\boldsymbol{v}\}}_{a_j^{\{\boldsymbol{r}\}}a^{\{\boldsymbol{i}\}}_1 h}, \quad j = 1, \ldots, s^{\{\boldsymbol{r}\}}.$$

This process is iteratively repeated until a partition in singletons is achieved, and suitable numerical integration schemes are selected to compute the flows of the elementary subsystems. We represent this *hierarchical splitting* process in terms of a *splitting tree*.

Definition 5.2 (Splitting tree)**.** A splitting tree for $N \in \mathbb{N}$ partitions is a full ordered binary tree[1] $G = (V, E)$ with node set $V \subset \mathbb{P}_{\geq 1}(\{1, \ldots, N\})$ so that

- the root node is $\boldsymbol{r} = \{1, \ldots, N\}$;
- if $\boldsymbol{v}, \boldsymbol{w} \in V$ are child nodes of $\boldsymbol{i} \in V$ (i.e., $\{\boldsymbol{i}, \boldsymbol{v}\}, \{\boldsymbol{i}, \boldsymbol{w}\} \in E$), then $\boldsymbol{v} \dot\cup \boldsymbol{w} = \boldsymbol{i}$; and
- each leaf node $\boldsymbol{\ell} \in V$ has cardinality one, $|\boldsymbol{\ell}| = 1$.

Consequently, a splitting tree has exactly N leaf nodes, and the total number of nodes is $|V| = 2N - 1$. An illustrative example of a splitting tree for $N = 5$ partitions is depicted in Fig. 5.1.

Given a splitting tree $G = (V, E)$, we define the sets $I \subset V$ of all inner nodes and $L \subset V$ of all leaf nodes, $V = I \dot\cup L$. We distinguish between two groups of leaf nodes: for leaf nodes $\boldsymbol{\ell} \in L' \subset L$, we approximate the flow of the subsystem $\dot{\boldsymbol{x}} = \boldsymbol{f}^{\{\boldsymbol{\ell}\}}(\boldsymbol{x})$ with a numerical scheme $\boldsymbol{\Phi}^{\{\boldsymbol{\ell}\}}_h$. For the remaining leaf nodes $\boldsymbol{\ell} \in L \setminus L'$, we compute the exact flow $\boldsymbol{\varphi}^{\{\boldsymbol{\ell}\}}_h$. For any $\boldsymbol{v} \in V$, let $\mathcal{P}^{\{\boldsymbol{v}\}} \subset I$ denote the set of all inner nodes in the unique path from the root node $\boldsymbol{r}$ to the node $\boldsymbol{v}$, excluding $\boldsymbol{v}$ itself. If $\boldsymbol{v}$ is included, we write $\bar{\mathcal{P}}^{\{\boldsymbol{v}\}} \coloneqq \mathcal{P}^{\{\boldsymbol{v}\}} \cup \{\boldsymbol{v}\}$.

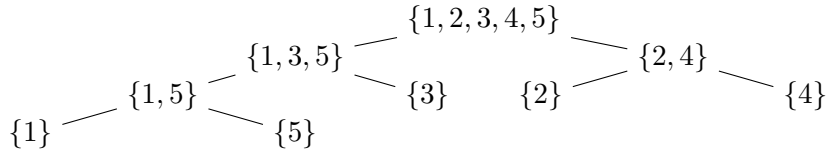


Fig. 5.1. A splitting tree for $N = 5$ partitions.

Definition 5.3 (Hierarchical splitting method)**.** A hierarchical splitting method is a splitting method $\boldsymbol{\Phi}_h$ for N-split systems (3.1) that is defined by a splitting tree $G = (V, E)$ for N partitions such that

- to each inner node $\boldsymbol{i} \in I$, a splitting method for $N = 2$ partitions is assigned,

$$\boldsymbol{\Phi}^{\{\boldsymbol{i}\}}_h = \boldsymbol{\varphi}^{\{\boldsymbol{w}\}}_{b^{\{\boldsymbol{i}\}}_{s^{\{\boldsymbol{i}\}}}h} \circ \boldsymbol{\varphi}^{\{\boldsymbol{v}\}}_{a^{\{\boldsymbol{i}\}}_{s^{\{\boldsymbol{i}\}}}h} \circ \ldots \circ \boldsymbol{\varphi}^{\{\boldsymbol{w}\}}_{b^{\{\boldsymbol{i}\}}_1 h} \circ \boldsymbol{\varphi}^{\{\boldsymbol{v}\}}_{a^{\{\boldsymbol{i}\}}_1 h}, \tag{5.4}$$

  where $\boldsymbol{v} \coloneqq \texttt{left}(\boldsymbol{i}) \in V$ and $\boldsymbol{w} \coloneqq \texttt{right}(\boldsymbol{i}) \in V$;
- to each leaf node $\boldsymbol{\ell} \in L' \subset L$, a numerical integration scheme $\boldsymbol{\Phi}^{\{\boldsymbol{\ell}\}}_h$ consistent with $\dot{\boldsymbol{x}} = \boldsymbol{f}^{\{\boldsymbol{\ell}\}}(\boldsymbol{x})$ is assigned;
- to each leaf node $\boldsymbol{\ell} \in L \setminus L'$, its exact flow $\boldsymbol{\varphi}^{\{\boldsymbol{\ell}\}}_h$ is assigned.

[1] A full binary tree is a binary tree in which each node has either zero or two child nodes. A ordered binary tree is a binary tree in which an ordering is specified for the children of each node. In particular, we distinguish between the left child node $\texttt{left}(\boldsymbol{v})$ and the right child node $\texttt{right}(\boldsymbol{v})$ of a node $\boldsymbol{v} \in V$.

The computational process of a hierarchical splitting method is illustrated in Algorithm 5.1.

Example 5.4. Consider the splitting tree from Fig. 5.1, assign the Strang splitting (3.5) to all inner nodes, and the exact flow $\boldsymbol{\varphi}_h^{\{m\}}$ $(m = 1, \dots, 5)$ to all leaf nodes. The resulting hierarchical splitting method reads

$$\begin{aligned}\boldsymbol{\Phi}_h = \boldsymbol{\varphi}_{h/8}^{\{1\}} \circ \boldsymbol{\varphi}_{h/4}^{\{5\}} \circ \boldsymbol{\varphi}_{h/8}^{\{1\}} \circ \boldsymbol{\varphi}_{h/2}^{\{3\}} \circ \boldsymbol{\varphi}_{h/8}^{\{1\}} \circ \boldsymbol{\varphi}_{h/4}^{\{5\}} \circ \boldsymbol{\varphi}_{h/8}^{\{1\}} \circ \boldsymbol{\varphi}_{h/2}^{\{2\}} \circ \boldsymbol{\varphi}_{h}^{\{4\}} \circ \boldsymbol{\varphi}_{h/2}^{\{2\}} \circ \boldsymbol{\varphi}_{h/8}^{\{1\}} \\ \circ \boldsymbol{\varphi}_{h/4}^{\{5\}} \circ \boldsymbol{\varphi}_{h/8}^{\{1\}} \circ \boldsymbol{\varphi}_{h/2}^{\{3\}} \circ \boldsymbol{\varphi}_{h/8}^{\{1\}} \circ \boldsymbol{\varphi}_{h/4}^{\{5\}} \circ \boldsymbol{\varphi}_{h/8}^{\{1\}}.\end{aligned} \tag{5.5}$$

The Lie–Trotter splitting (5.2) and the Strang splitting (5.3) can also be represented as hierarchical splitting methods with the splitting tree from Fig. 5.2, where we assign the respective splitting method for two-split systems to all inner nodes, and the exact flows $\boldsymbol{\varphi}_h^{\{m\}}$ $(m = 1, \dots, N)$ to the leaf nodes.

In general, the number of sub-steps involving a numerical method $\boldsymbol{\Phi}_h^{\{v\}}$ increases exponentially with the distance of the node $\boldsymbol{v}$ from the root node. Therefore, it is advisable to maintain a small depth of the splitting tree by designing balanced binary trees. Furthermore, the ordering of the flows within the splitting tree can significantly impact the accuracy and numerical stability of the hierarchical splitting method.

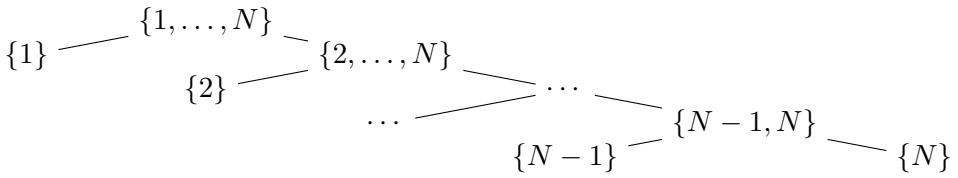


Fig. 5.2. Splitting tree for representing the Lie–Trotter splitting (5.2) and the Strang splitting (5.3) as hierarchical splitting methods.

Algorithm 5.1: Hierarchical splitting method

function STEP$(G, \boldsymbol{x}, h, \boldsymbol{v})$

```
/* Applies the method assigned to node v of the splitting tree G
   to advance the state x by a step of size h.  A time step of
   the hierarchical splitting method is performed for v = r.   */
if v ∈ I then
    for j = 1, . . . , s^{v} do
        x ← STEP(G, x, a_j^{v} h, left(v))
        x ← STEP(G, x, b_j^{v} h, right(v))
else if v ∈ L′ then
    x ← Φ_h^{v}(x)
else
    x ← φ_h^{v}(x)
return x
```

5.2.2. Convergence Order and Accuracy

The order conditions of splitting methods for N-split systems can be derived from the BCH formula, or strictly speaking via its generalization to $N \geq 2$ partitions [127]. For investigating the convergence order and the leading-order error terms of hierarchical splitting methods, we will make use of the following lemma.

Lemma 5.5. *Consider (non-commuting) matrices $\boldsymbol{A}^{\{m\}}, \boldsymbol{B}^{\{m\}} \in \mathbb{R}^{n\times n}$ and arbitrary scalar parameters $\alpha_i^{\{m\}}, \beta_i^{\{m\}} \in \mathbb{R}$ for $i = 1, \dots, s$ with $s \in \mathbb{N}$, and $m = 1, \dots, N$ with $N \in \mathbb{N}$. Additionally, let $p \in \mathbb{N}$. Then, as $h \to 0$, it holds*

$$\prod_{i=1}^{s}\prod_{m=1}^{N} \exp\left(\alpha_i^{\{m\}} h\boldsymbol{A}^{\{m\}} + \beta_i^{\{m\}} h^{p+1}\boldsymbol{B}^{\{m\}} + \mathcal{O}(h^{p+2})\right) = \exp\left(\boldsymbol{C}(h) + h^{p+1}\left(\sum_{m=1}^{N}\sum_{i=1}^{s} \beta_i^{\{m\}}\boldsymbol{B}^{\{m\}}\right) + \mathcal{O}(h^{p+2})\right),$$

where $\exp(\boldsymbol{C}(h)) = \left(\prod_{i=1}^{s}\prod_{m=1}^{N} \exp\left(\alpha_i^{\{m\}} h\boldsymbol{A}^{\{m\}}\right)\right)$.

Proof. A first-order Taylor expansion of the individual factors yields

$$\begin{aligned} &\exp\left(\alpha_i^{\{m\}} h\boldsymbol{A}^{\{m\}} + \beta_i^{\{m\}} h^{p+1}\boldsymbol{B}^{\{m\}} + \mathcal{O}(h^{p+2})\right) \\ =\ &\exp\left(\alpha_i^{\{m\}} h\boldsymbol{A}^{\{m\}}\right) + \beta_i^{\{m\}} h^{p+1}\boldsymbol{B}^{\{m\}} + \mathcal{O}(h^{p+2}). \end{aligned} \tag{5.6}$$

We can substitute (5.6) into the product over i and m and expand. Since every perturbation $\beta_i^{\{m\}} h^{p+1}\boldsymbol{B}^{\{m\}} + \mathcal{O}(h^{p+2})$ is of order h^{p+1}, any term involving a product of perturbations contributes to the remainder term $\mathcal{O}(h^{p+2})$. Therefore, only the terms containing exactly one perturbation enter the leading order error term. Consequently, collecting all terms and inserting $\exp(\boldsymbol{C}(h))$ yields

$$\exp(\boldsymbol{C}(h)) + h^{p+1}\left(\sum_{m=1}^{N}\sum_{i=1}^{s} \beta_i^{\{m\}}\boldsymbol{B}^{\{m\}}\right) + \mathcal{O}(h^{p+2}),$$

which can be interpreted as a first-order Taylor expansion of the asserted expression. □

Theorem 5.6. *Consider a hierarchical splitting method $\boldsymbol{\Phi}_h$ with numerical integration schemes $\boldsymbol{\Phi}_h^{\{\boldsymbol{v}\}}$ of order $p^{\{\boldsymbol{v}\}}$ assigned to nodes $\boldsymbol{v} \in V$,*

$$\boldsymbol{\Phi}_h^{\{\boldsymbol{v}\}} = \exp\left(h\mathcal{L}^{\{\boldsymbol{v}\}} + h^{p^{\{\boldsymbol{v}\}}+1}\mathcal{O}^{\{\boldsymbol{v}\}}_{p^{\{\boldsymbol{v}\}}+1} + \mathcal{O}(h^{p^{\{\boldsymbol{v}\}}+2})\right)\mathrm{Id},$$

with leading-order error term $\mathcal{O}^{\{\boldsymbol{v}\}}_{p^{\{\boldsymbol{v}\}}+1}$. For each node $\boldsymbol{v} \in V$, we define the mapping

$$\delta^{\{\boldsymbol{v}\}} \colon \mathcal{P}^{\{\boldsymbol{v}\}} \to \{0,1\}, \quad \delta^{\{\boldsymbol{v}\}}(\boldsymbol{w}) = \begin{cases} 1, & \texttt{left}(\boldsymbol{w}) \in \bar{\mathcal{P}}^{\{\boldsymbol{v}\}}, \\ 0, & \texttt{right}(\boldsymbol{w}) \in \bar{\mathcal{P}}^{\{\boldsymbol{v}\}}. \end{cases} \tag{5.7}$$

Then, $\boldsymbol{\Phi}_h$ is at least convergent of order $p \coloneqq \min_{\boldsymbol{v}\in I\dot\cup L'} p^{\{\boldsymbol{v}\}}$,

$$\boldsymbol{\Phi}_h = \exp\left(h\mathcal{L} + h^{p+1}\mathcal{O}_{p+1} + \mathcal{O}(h^{p+2})\right)\mathrm{Id},$$

with leading-order error term

$$\mathcal{O}_{p+1} = \sum_{\substack{\boldsymbol{v}\in I\dot\cup L' \\ p^{\{\boldsymbol{v}\}}=p}} \left[\prod_{\boldsymbol{w}\in\mathcal{P}^{\{\boldsymbol{v}\}}} \sum_{j=1}^{s^{\{\boldsymbol{w}\}}} \left(\delta^{\{\boldsymbol{v}\}}(\boldsymbol{w}) a_j^{\{\boldsymbol{w}\}} + (1-\delta^{\{\boldsymbol{v}\}}(\boldsymbol{w})) b_j^{\{\boldsymbol{w}\}}\right)^{p+1}\right] \mathcal{O}^{\{\boldsymbol{v}\}}_{p+1}. \tag{5.8}$$

Proof. We will prove the statement of the theorem by iteratively applying Lemma 5.5 from the leaf nodes up to the root node $\boldsymbol{r}$. For each node $\boldsymbol{v} \in V$, the assigned method $\boldsymbol{\Phi}_h^{\{\boldsymbol{v}\}}$ or flow $\boldsymbol{\varphi}_h^{\{\boldsymbol{v}\}}$ is employed $\prod_{\boldsymbol{w}\in\mathcal{P}^{\{\boldsymbol{v}\}}} s^{\{\boldsymbol{w}\}}$ times with step sizes

$$\tilde{h}^{\{\boldsymbol{v}\}}_{(j_{\boldsymbol{w}})_{\boldsymbol{w}\in\mathcal{P}^{\{\boldsymbol{v}\}}}} \coloneqq \prod_{\boldsymbol{w}\in\mathcal{P}^{\{\boldsymbol{v}\}}} (\delta^{\{\boldsymbol{v}\}}(\boldsymbol{w}) a_{j_{\boldsymbol{w}}}^{\{\boldsymbol{w}\}} + (1-\delta^{\{\boldsymbol{v}\}}(\boldsymbol{w})) b_{j_{\boldsymbol{w}}}^{\{\boldsymbol{w}\}}) h,$$

for $j_{\boldsymbol{w}} = 1, \dots, s^{\{\boldsymbol{w}\}}$, and all $\boldsymbol{w} \in \mathcal{P}^{\{\boldsymbol{v}\}}$. Let $H^{\{\boldsymbol{v}\}}$ denote the set of all step sizes $\tilde{h}^{\{\boldsymbol{v}\}}_{(j_{\boldsymbol{w}})_{\boldsymbol{w}\in\mathcal{P}^{\{\boldsymbol{v}\}}}}$. Note that it holds $\sum_{\tilde{h}\in H^{\{\boldsymbol{v}\}}} \tilde{h} = h$ for all $\boldsymbol{v} \in V$, provided that all splitting methods are consistent, i.e., $\sum_j a_j^{\{\boldsymbol{i}\}} = \sum_j b_j^{\{\boldsymbol{i}\}} = 1$ for all $\boldsymbol{i} \in I$. By Lemma 3.11, we can formally write $\boldsymbol{\Phi}_h$ as a product of $\exp(\tilde{h}\mathcal{L}^{\{\boldsymbol{k}\}})$, $\boldsymbol{k} \in L \setminus L'$, $\tilde{h} \in H^{\{\boldsymbol{k}\}}$, and

$$\exp\left(\tilde{h}\mathcal{L}^{\{\boldsymbol{\ell}\}} + \tilde{h}^{p^{\{\boldsymbol{\ell}\}}+1}\mathcal{O}^{\{\boldsymbol{\ell}\}}_{p^{\{\boldsymbol{\ell}\}}+1} + \mathcal{O}(\tilde{h}^{p^{\{\boldsymbol{\ell}\}}+2})\right),$$

for $\boldsymbol{\ell} \in L'$ and $\tilde{h} \in H^{\{\boldsymbol{\ell}\}}$. Applying Lemma 5.5 for $p \coloneqq \min_{\boldsymbol{\ell}\in L'} p^{\{\boldsymbol{\ell}\}}$ yields

$$\boldsymbol{\Phi}_h = \exp\left(h\tilde{\mathcal{L}} + h^{p+1}\tilde{\mathcal{O}}_{p+1} + \mathcal{O}(h^{p+2})\right)\mathrm{Id}, \tag{5.9}$$

where

$$\tilde{\mathcal{O}}_{p+1} = \sum_{\substack{\boldsymbol{\ell}\in L' \\ p^{\{\boldsymbol{\ell}\}}=p}} \left[\prod_{\boldsymbol{w}\in\mathcal{P}^{\{\boldsymbol{\ell}\}}} \sum_{j=1}^{s^{\{\boldsymbol{w}\}}} \left(\delta^{\{\boldsymbol{\ell}\}}(\boldsymbol{w})a_j^{\{\boldsymbol{w}\}} + (1-\delta^{\{\boldsymbol{\ell}\}}(\boldsymbol{w}))b_j^{\{\boldsymbol{w}\}}\right)^{p+1}\right]\mathcal{O}^{\{\boldsymbol{\ell}\}}_{p+1},$$

and $\tilde{\boldsymbol{\Phi}}_h = \exp(h\tilde{\mathcal{L}})\mathrm{Id}$ denotes the splitting method $\boldsymbol{\Phi}_h$ where all elementary subsystems are solved exactly, i.e., to each leaf node $\boldsymbol{\ell} \in L$ its exact flow $\boldsymbol{\varphi}_h^{\{\boldsymbol{\ell}\}}$ is assigned. We define the set $A \subset I$ of previously investigated inner nodes, initially set to $A = \emptyset$, and the set $B \subset V \setminus A$ of not yet investigated nodes whose child nodes are in $A \,\dot\cup\, L$. We can then interpret $\tilde{\boldsymbol{\Phi}}_h = \exp(h\tilde{\mathcal{L}})\mathrm{Id}$ as the splitting method $\boldsymbol{\Phi}_h$ where all numerical integration schemes assigned to nodes $\boldsymbol{v} \in A \,\dot\cup\, L'$ are replaced with the exact flow $\boldsymbol{\varphi}_h^{\{\boldsymbol{v}\}} = \exp(h\mathcal{L}^{\{\boldsymbol{v}\}})\mathrm{Id}$ and write (5.9) with

$$\tilde{\mathcal{O}}_{p+1} = \sum_{\substack{\boldsymbol{v}\in A\dot\cup L' \\ p^{\{\boldsymbol{v}\}}=p}} \left[\prod_{\boldsymbol{w}\in\mathcal{P}^{\{\boldsymbol{v}\}}} \sum_{j=1}^{s^{\{\boldsymbol{w}\}}} \left(\delta^{\{\boldsymbol{v}\}}(\boldsymbol{w})a_j^{\{\boldsymbol{w}\}} + (1-\delta^{\{\boldsymbol{v}\}}(\boldsymbol{w}))b_j^{\{\boldsymbol{w}\}}\right)^{p+1}\right]\mathcal{O}^{\{\boldsymbol{v}\}}_{p+1}, \tag{5.10}$$

and $p \coloneqq \min_{\boldsymbol{v}\in A\dot\cup L'} p^{\{\boldsymbol{v}\}}$. As long as $A \neq I$, we proceed as follows.

1) Select a node $\boldsymbol{i} \in B$ and consider $\boldsymbol{v} = \texttt{left}(\boldsymbol{i}) \in A \,\dot\cup\, L$, $\boldsymbol{w} = \texttt{right}(\boldsymbol{i}) \in A \,\dot\cup\, L$.

2) The method $\tilde{\boldsymbol{\Phi}}_h$ involves applications of the splitting method

$$\boldsymbol{\Phi}_{\tilde{h}}^{\{\boldsymbol{i}\}} = \exp\left(a_1^{\{\boldsymbol{i}\}}\tilde{h}\mathcal{L}^{\{\boldsymbol{v}\}}\right)\exp\left(b_1^{\{\boldsymbol{i}\}}\tilde{h}\mathcal{L}^{\{\boldsymbol{w}\}}\right)\cdots\exp\left(a_{s^{\{\boldsymbol{i}\}}}^{\{\boldsymbol{i}\}}\tilde{h}\mathcal{L}^{\{\boldsymbol{v}\}}\right)\exp\left(b_{s^{\{\boldsymbol{i}\}}}^{\{\boldsymbol{i}\}}\tilde{h}\mathcal{L}^{\{\boldsymbol{w}\}}\right)\mathrm{Id}$$

with $\tilde{h} \in H^{\{\boldsymbol{i}\}}$. We can replace these occurrences with

$$\boldsymbol{\Phi}_{\tilde{h}}^{\{\boldsymbol{i}\}} = \exp\left(\tilde{h}\mathcal{L}^{\{\boldsymbol{i}\}} + \tilde{h}^{p^{\{\boldsymbol{i}\}}+1}\mathcal{O}^{\{\boldsymbol{i}\}}_{p^{\{\boldsymbol{i}\}}+1} + \mathcal{O}(\tilde{h}^{p^{\{\boldsymbol{i}\}}+2})\right)\mathrm{Id}.$$

3) Add $\boldsymbol{i}$ to the set of investigated nodes A and, if $\boldsymbol{i} \neq \boldsymbol{r}$, add the parent node of $\boldsymbol{i}$ to the set B. Subsequently, formally applying Lemma 5.5 with $p \coloneqq \min_{\boldsymbol{v}\in A\dot\cup L'} p^{\{\boldsymbol{v}\}}$ once more yields (5.10).

Repeating this process until $A = I$, we have $A \,\dot\cup\, L' = I \,\dot\cup\, L'$, $\tilde{\boldsymbol{\Phi}}_h = \boldsymbol{\varphi}_h = \exp(h\mathcal{L})\mathrm{Id}$, and thus obtain the asserted expression. □

Corollary 5.7. *Assigning methods of order at least p to all inner nodes $\boldsymbol{i} \in I$ and to all leaf nodes $\boldsymbol{\ell} \in L'$ is sufficient to obtain a hierarchical splitting method of order p.*

This straightforward extension of the order theory facilitates the derivation of hierarchical splitting methods of order $p > 2$ by utilizing efficient splitting methods for two-split systems that are already available in the literature, such as the methods mentioned in the survey papers [94, 18] on splitting methods and the references therein.

Remark 5.8. In the literature, also methods for ($K > 2$)-split systems are available [5, 4, 126, 127]. By generalizing the concept of a splitting tree, these methods can be integrated into the hierarchical splitting approach: instead of requiring a splitting tree to be a binary tree, the natural generalization requires that if a splitting method for K-split systems is assigned to an inner node $\boldsymbol{i} \in I$, then $\boldsymbol{i}$ must have exactly K child nodes.

Proposition 5.9 (Self-adjoint hierarchical splitting methods)**.** *A hierarchical splitting method $\boldsymbol{\Phi}_h$ is self-adjoint, i.e., it holds $\boldsymbol{\Phi}_h = \boldsymbol{\Phi}_{-h}^{-1} =: \boldsymbol{\Phi}_h^*$, provided that the methods assigned to the leaf nodes are self-adjoint and the splitting methods* (5.4) *assigned to the inner nodes have palindromic sequences of sub-steps.*

Proof. Recall that $(\boldsymbol{\Phi}_h^{\{1\}} \circ \boldsymbol{\Phi}_h^{\{2\}})^* = (\boldsymbol{\Phi}_h^{\{2\}})^* \circ (\boldsymbol{\Phi}_h^{\{1\}})^*$ and that self-adjointness is preserved under scaling of the step size. We proceed by structural induction over the splitting tree. For the leaf nodes $\boldsymbol{\ell} \in L$, the assigned (numerical) flows are self-adjoint as per assumption. Now let $\boldsymbol{i} \in I$ and assume that the maps assigned to $\texttt{left}(\boldsymbol{i})$ and $\texttt{right}(\boldsymbol{i})$ are self-adjoint. Taking the adjoint of the splitting method (5.4) reverses the sequence of sub-steps and replaces each of them by its adjoint, which leaves them unchanged by the induction hypothesis. By assumption, the sequence of sub-steps of the splitting method assigned to node $\boldsymbol{i}$ is palindromic, so that the reversed sequence coincides with the original one and $\boldsymbol{\Phi}_h^{\{\boldsymbol{i}\}}$ is self-adjoint. For $\boldsymbol{i} = \boldsymbol{r}$, this yields the self-adjointness of the hierarchical splitting method. □

Remark 5.10. Next to conventional splitting methods for two-split systems, it is also feasible to assign commutator-based splitting methods (cf. [144, 82, 101, 115, 97]) to nodes of a splitting tree. Assigning a commutator-based splitting method to an inner node $\boldsymbol{i} \in I$ demands that the approach is consistent with the Lie derivatives of the child nodes $\texttt{left}(\boldsymbol{i})$ and $\texttt{right}(\boldsymbol{i})$. We will discuss the incorporation of commutator-based splitting approaches, in particular the (Hessian-free) force-gradient approach for separable Hamiltonian systems (3.32), in detail in Chapter 7.

5.2.3. Composition Methods

Composition techniques constitute a common choice to derive splitting methods of higher order that can also be applied to general N-split systems. Hierarchical splitting methods $\boldsymbol{\Phi}_h$ can also be employed as the underlying base scheme of a composition method to derive higher-order splitting methods for N-split systems. In particular, a composition method of the form (3.11) utilizing a hierarchical splitting method as the underlying base scheme is again a hierarchical splitting method.

Lemma 5.11. *Let $\boldsymbol{\Phi}_h$ be a hierarchical splitting method. Then, any composition method* (3.11) *using $\boldsymbol{\Phi}_h$ as the base method is again a hierarchical splitting method. The hierarchical splitting representation of the composition method differs from the base method $\boldsymbol{\Phi}_h$ only in the root node. In particular, the method $\boldsymbol{\Phi}_h^{\{\boldsymbol{r}\}}$ is replaced with its composition using the same weights as for the overall composition.*

Proof. The composition of the full hierarchical splitting method reads

$$\boldsymbol{\Psi}_h = \boldsymbol{\Phi}_{\gamma_s h} \circ \boldsymbol{\Phi}_{\gamma_{s-1} h} \circ \ldots \circ \boldsymbol{\Phi}_{\gamma_1 h}.$$

By construction, any application of the base method starts with an application of the method $\boldsymbol{\Phi}_{\gamma_i h}^{\{\boldsymbol{r}\}}$ assigned to the root node. Consequently, the initial step of the iterative splitting process yields

$$\boldsymbol{\Phi}_{\gamma_s h}^{\{\boldsymbol{r}\}} \circ \boldsymbol{\Phi}_{\gamma_{s-1} h}^{\{\boldsymbol{r}\}} \circ \ldots \circ \boldsymbol{\Phi}_{\gamma_1 h}^{\{\boldsymbol{r}\}},$$

which is equivalent to directly applying the composition technique to the method $\boldsymbol{\Phi}_h^{\{\boldsymbol{r}\}}$ assigned to the root node. □

When considering compositions of self-adjoint splitting methods, the last flow of the i-th application and the first flow of the $(i+1)$-th application of the base scheme coincide. As it holds $\boldsymbol{\varphi}_{\gamma_{i+1}h}^{\{\boldsymbol{v}\}} \circ \boldsymbol{\varphi}_{\gamma_i h}^{\{\boldsymbol{v}\}} = \boldsymbol{\varphi}_{(\gamma_i+\gamma_{i+1})h}^{\{\boldsymbol{v}\}}$ for the exact flows, these subsequent flow evaluations are frequently combined to reduce computational cost. For example, we can consider the triple jump composition (3.13) with the Strang splitting (3.5) for two-split systems as the underlying base scheme. The composition method is convergent of order $p = 4$ and reads

$$\boldsymbol{\Phi}_h = \boldsymbol{\varphi}_{\gamma_1 h/2}^{\{1\}} \circ \boldsymbol{\varphi}_{\gamma_1 h}^{\{2\}} \circ \boldsymbol{\varphi}_{\gamma_1 h/2}^{\{1\}} \circ \boldsymbol{\varphi}_{\gamma_2 h/2}^{\{1\}} \circ \boldsymbol{\varphi}_{\gamma_2 h}^{\{2\}} \circ \boldsymbol{\varphi}_{\gamma_2 h/2}^{\{1\}} \circ \boldsymbol{\varphi}_{\gamma_1 h/2}^{\{1\}} \circ \boldsymbol{\varphi}_{\gamma_1 h}^{\{2\}} \circ \boldsymbol{\varphi}_{\gamma_1 h/2}^{\{1\}}. \tag{5.11a}$$

If the exact flows of the subsystems are computed, (5.11) is equivalent to

$$\boldsymbol{\Phi}_h = \boldsymbol{\varphi}_{\gamma_1 h/2}^{\{1\}} \circ \boldsymbol{\varphi}_{\gamma_1 h}^{\{2\}} \circ \boldsymbol{\varphi}_{(\gamma_1+\gamma_2) h/2}^{\{1\}} \circ \boldsymbol{\varphi}_{\gamma_2 h}^{\{2\}} \circ \boldsymbol{\varphi}_{(\gamma_1+\gamma_2) h/2}^{\{1\}} \circ \boldsymbol{\varphi}_{\gamma_1 h}^{\{2\}} \circ \boldsymbol{\varphi}_{\gamma_1 h/2}^{\{1\}}. \tag{5.11b}$$

In a hierarchical splitting method, however, we do in general not assign the exact flows to the child nodes of the root node. Consequently, combining subsequent flows would result in a different hierarchical splitting method that potentially exhibits a lower order of convergence, as illustrated in the following example.

Example 5.12. Consider the hierarchical splitting method (5.5) and its splitting tree from Fig. 5.1. When replacing the Strang splitting in the root node with the composition method (5.11a), Theorem 5.6 states that all error terms $\mathcal{O}_3^{\{\boldsymbol{i}\}}$ for $\boldsymbol{i} \in I \setminus \{\boldsymbol{r}\}$ are either scaled by

$$2\left(\frac{\gamma_1}{2}\right)^3 + 2\left(\frac{\gamma_2}{2}\right)^3 + 2\left(\frac{\gamma_3}{2}\right)^3 = \frac{1}{4}\left(\gamma_1^3 + \gamma_2^3 + \gamma_3^3\right) = 0, \quad \text{or} \quad \gamma_1^3 + \gamma_2^3 + \gamma_3^3 = 0,$$

i.e., the third-order error terms are all vanishing. Due to the inherent symmetry of the method, the hierarchical splitting method is convergent of order $p = 4$. Alternatively, we may replace the Strang splitting in the root node with the composition method (5.11b). The leading-order error terms in the right branch of the splitting tree are scaled by

$$\gamma_1^3 + \gamma_2^3 + \gamma_3^3 = 0$$

and thus vanish. However, the leading-order error terms in the left branch are scaled by

$$\left(\frac{\gamma_1}{2}\right)^3 + \left(\frac{\gamma_1+\gamma_2}{2}\right)^3 + \left(\frac{\gamma_2+\gamma_3}{2}\right)^3 + \left(\frac{\gamma_3}{2}\right)^3 = \frac{3}{8}\left(\gamma_1^2\gamma_2 + \gamma_1\gamma_2^2 + \gamma_2^2\gamma_3 + \gamma_2\gamma_3^2\right) \neq 0.$$

As a result, the overall hierarchical splitting method is only convergent of order $p = 2$.

Note that Example 5.12 includes a hierarchical splitting method of order p, wherein splitting methods of a lower convergence order $q < p$ are assigned to nodes of the splitting tree.

Lemma 5.13. *Composition methods of the form* (3.15) *using a first-order hierarchical splitting method* $\boldsymbol{\Phi}_h$ *as the underlying base scheme can be represented as a hierarchical splitting method.*

Proof. Consider the coefficients α_i and β_i from the composition (3.15) and define $c \coloneqq \sum_{i=1}^s \alpha_i$. Provided that the composition is consistent, it holds $\sum_{i=1}^s \alpha_i + \beta_i = 1$ and hence $\sum_{i=1}^s \beta_i = 1 - c$. Starting from the overall system (3.1), we consider the initial splitting

$$\boldsymbol{f}(\boldsymbol{x}) = c\boldsymbol{f}(\boldsymbol{x}) + (1-c)\boldsymbol{f}(\boldsymbol{x}).$$

To the root node, we assign the splitting method

$$\boldsymbol{\Psi}_h^{\{r\}} = \boldsymbol{\varphi}_{\beta_s h}^{\{2\}} \circ \boldsymbol{\varphi}_{\alpha_s h}^{\{1\}} \circ \ldots \circ \boldsymbol{\varphi}_{\beta_1 h}^{\{2\}} \circ \boldsymbol{\varphi}_{\alpha_1 h}^{\{1\}}$$

with the integrator coefficients from the composition (3.15). Assigning the adjoint $\boldsymbol{\Phi}_h^*$ of the hierarchical splitting method to the left child node and the base method $\boldsymbol{\Phi}_h$ to the right child node results in a representation of the composition method (3.15) as a hierarchical splitting method. □

5.3. Numerical Results

To illustrate the advantages of the hierarchical splitting approach, we present numerical results for hierarchical splitting methods applied to linear port-Hamiltonian systems [121] by considering a damped and forced harmonic oscillator

$$\begin{aligned} \dot{\boldsymbol{x}} = \begin{pmatrix} \dot{p} \\ \dot{q} \end{pmatrix} &= \Bigg[\underbrace{\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}}_{=\boldsymbol{J}} - \underbrace{\begin{pmatrix} d & 0 \\ 0 & 0 \end{pmatrix}}_{=\boldsymbol{R}} \Bigg] \underbrace{\begin{pmatrix} 1 & 0 \\ 0 & \omega^2 \end{pmatrix} \begin{pmatrix} p \\ q \end{pmatrix}}_{=\nabla\mathcal{H}=\boldsymbol{Q}\boldsymbol{x}} + \underbrace{\begin{pmatrix} 1/m \\ 0 \end{pmatrix}}_{=\boldsymbol{B}} u(t), \\ y(t) &= \underbrace{\begin{pmatrix} 1/m & 0 \end{pmatrix}}_{=\boldsymbol{B}^\top} \underbrace{\begin{pmatrix} 1 & 0 \\ 0 & \omega^2 \end{pmatrix} \begin{pmatrix} p \\ q \end{pmatrix}}_{=\nabla\mathcal{H}=\boldsymbol{Q}\boldsymbol{x}} = \frac{p}{m}. \end{aligned} \tag{5.12}$$

As a port-Hamiltonian system, (5.12) satisfies the dissipation inequality [121]

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{H}(\boldsymbol{x}) = \underbrace{\nabla\mathcal{H}(\boldsymbol{x})^\top \boldsymbol{J} \nabla\mathcal{H}(\boldsymbol{x})}_{=0} - \underbrace{\nabla\mathcal{H}(\boldsymbol{x})^\top \boldsymbol{R} \nabla\mathcal{H}(\boldsymbol{x})}_{\geq 0} + \underbrace{\nabla\mathcal{H}(\boldsymbol{x})^\top \boldsymbol{B}}_{=y(t)^\top} u(t) \leq y(t)^\top u(t), \tag{5.13}$$

which we aim to preserve at a discrete level. The presence of dissipation renders the use of negative sub-steps inadmissible, so that standard composition approaches of order $p > 2$ are not applicable. Conventional splitting methods for N-split systems can at most achieve order $p = 2$ when demanding the integrator coefficients being non-negative [14]. In [97], it has been shown that the use of commutator-based splitting approaches enables the construction of splitting methods for two-split linear port-Hamiltonian systems of order $p > 2$. We can design hierarchical splitting methods for linear port-Hamiltonian systems by assigning the methods from [97] to nodes of a splitting tree. We augment the system (5.12) by the ODE $\dot{\tau} = 1$ to turn it into an autonomous system and consider the splitting

$$\begin{aligned} \boldsymbol{f}^{\{1\}} &= \begin{pmatrix} \boldsymbol{J}\nabla\mathcal{H}(\boldsymbol{x}) \\ 0 \end{pmatrix}, \quad \boldsymbol{f}^{\{2\}} = \begin{pmatrix} -\boldsymbol{R}\nabla\mathcal{H}(\boldsymbol{x}) \\ 0 \end{pmatrix}, \\ \boldsymbol{f}^{\{3\}} &= \begin{pmatrix} \boldsymbol{B}u(\tau) \\ 0 \end{pmatrix}, \qquad \boldsymbol{f}^{\{4\}} = \begin{pmatrix} \boldsymbol{0} \\ 1 \end{pmatrix}. \end{aligned} \tag{5.14}$$

For the iterative splitting process, we consider the splitting tree depicted in Fig. 5.3, thereby deriving structure-preserving splitting methods of order $p = 4$ for linear port-Hamiltonian systems with $N > 2$ partitions. In Tab. 5.1, the methods assigned to the inner nodes of the splitting tree are summarized. To all leaf nodes, we assign the corresponding exact flow.

We consider the values $\omega = 5$, $d = 0.1$, $m = 0.4$, input $u(t) = 5\sin(t)$, initial value $\boldsymbol{x}(0) = (0, \pi/4)^\top$, and the time interval $t \in [0, 40]$. We compare the hierarchical splitting method (`HSM`) with the fourth-order composition method (4.39) using the Lie–Trotter splitting (5.2) as the underlying base scheme (`BMs6p4`). Note that the method `BMs6p4`

involves, as a conventional splitting method of order $p > 2$, negative integrator coefficients. According to the work-precision diagram in Fig. 5.4a, the hierarchical splitting method is the more efficient choice. Moreover, setting $u(t) = 0$, the results in Fig. 5.4b show that the hierarchical splitting method satisfies the dissipation inequality (5.13), as indicated by the non-increasing values. In contrast, the composition method `BMs6p4` fails to satisfy the dissipation inequality for all step sizes $h > 0$.

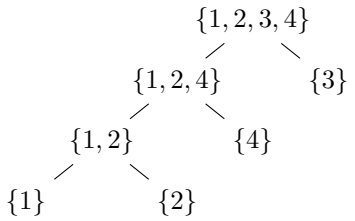
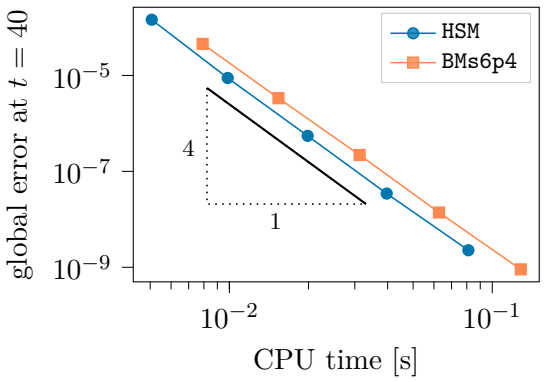


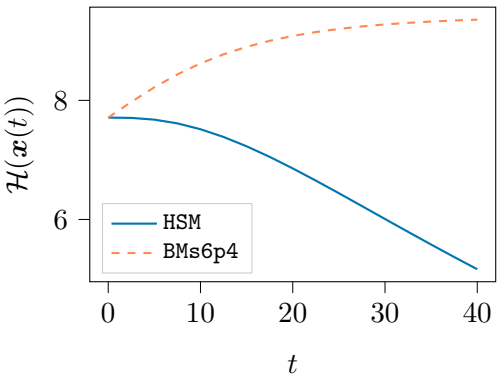
Fig. 5.3. Splitting tree for the damped and forced harmonic oscillator (5.12).

Node	Method
$\{1, 2, 3, 4\}$	[97, Scheme 4 a)]
$\{1, 2, 4\}$	(3.3)
$\{1, 2\}$	[97, Example 5 (ii)]

Tab. 5.1. Hierarchical splitting method for (5.12) based on the splitting tree from Fig. 5.3.



(a) Work-precision diagram for the hierarchical splitting method `HSM` (○) and the composition method `BMs6p4` (□).



(b) Hamiltonian along the numerical trajectories obtained for `HSM` (solid line) and `BMs6p4` (dashed line) with $h = 2.5$ and $u = 0$.

Fig. 5.4. Damped and forced harmonic oscillator. (a) Work-precision diagram and (b) investigation of the Hamiltonian for $u = 0$.

5.4. Summary

In this chapter, we presented a novel class of hierarchical splitting methods for N-split differential equations which apply splitting methods for two-split systems in a hierarchical manner. Since assigning numerical integration schemes of order p to all nodes of a splitting tree suffices to obtain a hierarchical splitting method of order p, the framework facilitates the construction of higher-order splitting methods for N-split systems without relying on composition techniques. We developed a fourth-order energy-consistent hierarchical splitting method for linear port-Hamiltonian systems involving commutator-based approaches that splits the right-hand side into $N = 4$ partitions. The method has been applied to a damped and forced harmonic oscillator, demonstrating the superior efficiency and energy consistency of the hierarchical splitting approach compared to a straightforward construction of splitting methods via composition techniques.

In the next chapter, we will present an extension of hierarchical splitting methods incorporating multiple time stepping techniques, thereby obtaining a framework in the intersection of geometric numerical integration and multirate integration.

Chapter 6.

Multirate Hierarchical Splitting Methods

6.1. Motivation: Dynamical Systems with Multirate Potential

In many technical applications, such as in multi-physics problems, we face situations where

a) many components of $\boldsymbol{x}$ vary slowly, while only a few components of $\boldsymbol{x}$ are characterized by fast dynamics, giving rise to component-wise partitioned systems

$$\begin{pmatrix} \dot{\boldsymbol{x}}^{\{\mathfrak{s}\}} \\ \dot{\boldsymbol{x}}^{\{\mathfrak{f}\}} \end{pmatrix} = \begin{pmatrix} \boldsymbol{g}^{\{\mathfrak{s}\}}(\boldsymbol{x}^{\{\mathfrak{s}\}}, \boldsymbol{x}^{\{\mathfrak{f}\}}) \\ \boldsymbol{g}^{\{\mathfrak{f}\}}(\boldsymbol{x}^{\{\mathfrak{s}\}}, \boldsymbol{x}^{\{\mathfrak{f}\}}) \end{pmatrix}, \quad \begin{pmatrix} \boldsymbol{x}^{\{\mathfrak{s}\}}(0) \\ \boldsymbol{x}^{\{\mathfrak{f}\}}(0) \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}_0^{\{\mathfrak{s}\}} \\ \boldsymbol{x}_0^{\{\mathfrak{f}\}} \end{pmatrix}; \tag{6.1}$$

where the full state $\boldsymbol{x} \in \mathbb{R}^n$ is split into slow components $\boldsymbol{x}^{\{\mathfrak{s}\}} \in \mathbb{R}^{n_\mathfrak{s}}$ and fast components $\boldsymbol{x}^{\{\mathfrak{f}\}} \in \mathbb{R}^{n_\mathfrak{f}}$, $n_\mathfrak{s} + n_\mathfrak{f} = n$, $n_\mathfrak{s} \gg n_\mathfrak{f}$;

b) computationally expensive parts of the right-hand side vary slowly enough (or are weakly coupled enough) that evaluating them at a coarser rate does not compromise the accuracy of the overall scheme, leading to right-hand side partitioned systems

$$\dot{\boldsymbol{x}} = \boldsymbol{f}^{\{\mathfrak{s}\}}(\boldsymbol{x}) + \boldsymbol{f}^{\{\mathfrak{f}\}}(\boldsymbol{x}), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0. \tag{6.2}$$

Any ODE-IVP of the form (6.1) can be transformed into an ODE-IVP of the form (6.2) by setting

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x}^{\{\mathfrak{s}\}} \\ \boldsymbol{x}^{\{\mathfrak{f}\}} \end{pmatrix}, \qquad \boldsymbol{x}_0 = \begin{pmatrix} \boldsymbol{x}_0^{\{\mathfrak{s}\}} \\ \boldsymbol{x}_0^{\{\mathfrak{f}\}} \end{pmatrix},$$
$$\boldsymbol{f}^{\{\mathfrak{s}\}}(\boldsymbol{x}) = \begin{pmatrix} \boldsymbol{g}^{\{\mathfrak{s}\}}(\boldsymbol{x}^{\{\mathfrak{s}\}}, \boldsymbol{x}^{\{\mathfrak{f}\}}) \\ \boldsymbol{0} \end{pmatrix}, \qquad \boldsymbol{f}^{\{\mathfrak{f}\}}(\boldsymbol{x}) = \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{g}^{\{\mathfrak{f}\}}(\boldsymbol{x}^{\{\mathfrak{s}\}}, \boldsymbol{x}^{\{\mathfrak{f}\}}) \end{pmatrix}.$$

As $\boldsymbol{g}^{\{\mathfrak{s}\}} \colon \mathbb{R}^n \to \mathbb{R}^{n_\mathfrak{s}}$ and $\boldsymbol{g}^{\{\mathfrak{f}\}} \colon \mathbb{R}^n \to \mathbb{R}^{n_\mathfrak{f}}$ with $n_\mathfrak{s} \gg n_\mathfrak{f}$, the partition $\boldsymbol{f}^{\{\mathfrak{s}\}}$ becomes computationally expensive.

Example 6.1 (A four body-problem with multirate potential)**.** As an example of a Hamiltonian system exhibiting multirate potential, which we will particularly use to perform numerical tests in Section 6.4, we consider a "solar system" introduced in [61, Example VIII.4.4], which consists of a sun, two planets, and a satellite that orbits rapidly around the first planet. The system is modeled via the separable Hamiltonian

$$\mathcal{H}(\boldsymbol{p}, \boldsymbol{q}) = \sum_{i=0}^{3} \frac{\boldsymbol{p}_i^\top \boldsymbol{p}_i}{2m_i} - \sum_{i=0}^{3} \sum_{j=i+1}^{3} \frac{m_i m_j}{\|\boldsymbol{q}_i - \boldsymbol{q}_j\|}, \tag{6.3}$$

where $\boldsymbol{p}_i, \boldsymbol{q}_i \in \mathbb{R}^2$, $m_0 = 1$ (sun), $m_1 = m_2 = 10^{-2}$ (planets), and $m_3 = 10^{-4}$ (satellite). The Hamiltonian (6.3) is split into three partitions

$$\mathcal{T}(\boldsymbol{p}) = \sum_{i=0}^{3} \frac{\boldsymbol{p}_i^\top \boldsymbol{p}_i}{2m_i}, \quad \mathcal{V}^{\{\mathfrak{f}\}}(\boldsymbol{q}) = -\frac{m_1 m_3}{\|\boldsymbol{q}_1 - \boldsymbol{q}_3\|}, \quad \mathcal{V}^{\{\mathfrak{s}\}}(\boldsymbol{q}) = - \sum_{\substack{i<j \\ (i,j)\neq(1,3)}} \frac{m_i m_j}{\|\boldsymbol{q}_i - \boldsymbol{q}_j\|},$$

where $\mathcal{V}^{\{\mathrm{f}\}}$ and $\mathcal{V}^{\{\mathrm{s}\}}$ denote the fast and slow parts of the potential, respectively. Combining the kinetic energy with the fast part of the potential, the corresponding equations of motion take the form (6.2) with

$$\boldsymbol{f}^{\{\mathrm{s}\}}(\boldsymbol{p},\boldsymbol{q}) = \begin{pmatrix} -\nabla\mathcal{V}^{\{\mathrm{s}\}}(\boldsymbol{q}) \\ \mathbf{0} \end{pmatrix}, \quad \boldsymbol{f}^{\{\mathrm{f}\}}(\boldsymbol{p},\boldsymbol{q}) = \begin{pmatrix} -\nabla\mathcal{V}^{\{\mathrm{f}\}}(\boldsymbol{q}) \\ \nabla\mathcal{T}(\boldsymbol{p}) \end{pmatrix}.$$

Since the slow part of the potential contains five terms, the computational work of evaluating the force $-\nabla\mathcal{V}^{\{\mathrm{s}\}}$ is approximately five times larger compared to the fast force $-\nabla\mathcal{V}^{\{\mathrm{f}\}}$. Simultaneously, the contribution of $\boldsymbol{f}^{\{\mathrm{s}\}}$ to the dynamics of the overall solution is comparatively small, as illustrated in Fig. 6.1. Moreover, the dynamics of the fast component evolve on a much faster time scale.

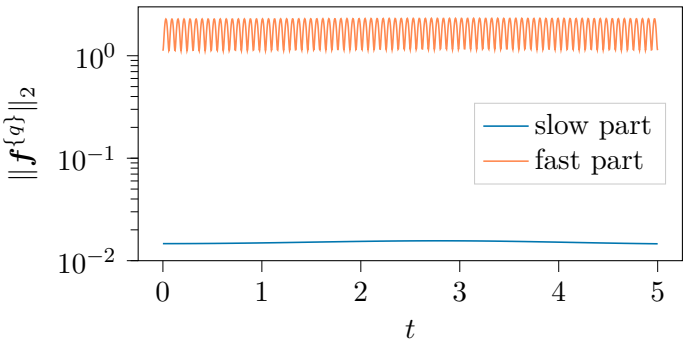


Fig. 6.1. Four body-problem. Magnitudes of the right-hand sides $\boldsymbol{f}^{\{\mathrm{s}\}}$ and $\boldsymbol{f}^{\{\mathrm{f}\}}$ along a numerical solution trajectory on the time interval $t \in [0, 5]$.

Utilizing a numerical integration scheme $\boldsymbol{\Phi}_h$ to integrate the subsystem $\dot{\boldsymbol{x}} = \boldsymbol{f}^{\{\mathrm{f}\}}(\boldsymbol{x})$ requires a much smaller step size h to sufficiently capture the system's dynamics than for the slow subsystem $\dot{\boldsymbol{x}} = \boldsymbol{f}^{\{\mathrm{s}\}}(\boldsymbol{x})$. For the overall system (6.2), the choice of the step size will be governed by the fast dynamics of the system. Hence, the computationally expensive part $\boldsymbol{f}^{\{\mathrm{s}\}}$ will be evaluated more often than it would have been the case when considering the subsystem $\dot{\boldsymbol{x}} = \boldsymbol{f}^{\{\mathrm{s}\}}(\boldsymbol{x})$ in the absence of the fast dynamics. As a consequence, we are interested in designing numerical integration schemes that evaluate the expensive part of the right-hand side less often than the rest. This can be achieved by *multirate methods* or *multiple time stepping techniques*. These methods utilize a large macro step size h that is sufficiently small to capture the dynamics of the slow part, and apply a smaller micro step size h/M, $M \in \mathbb{N}$, to the fast part of the vector field to accurately capture its dynamics.

There exist several approaches to the design of multirate methods, such as inter-/extrapolation based multirate methods [6], and multirate extensions of PRK [57] and GARK [58] methods. However, these approaches do not explicitly account for the classification of dynamical systems from Section 2.2 and hence do not aim for numerical flows that belong to the same category as the exact flow of the system. In the spirit of geometric numerical integration, it is desirable to develop structure-preserving multirate methods. We proposed an approach within the GARK framework in [119], whereas variational multirate integrators have been proposed in parallel in [99]. Another approach to multirate integration, which we will particularly focus on throughout this thesis, is within the framework of splitting and composition methods.

6.2. Multirate Hierarchical Splitting Methods

Suppose a system of the form (6.2) and numerical integrators $\boldsymbol{\Phi}_h^{\{\mathfrak{s}\}}$ and $\boldsymbol{\Phi}_h^{\{\mathfrak{f}\}}$ consistent with the subsystems $\dot{\boldsymbol{x}} = \boldsymbol{f}^{\{\mathfrak{s}\}}(\boldsymbol{x})$ and $\dot{\boldsymbol{x}} = \boldsymbol{f}^{\{\mathfrak{f}\}}(\boldsymbol{x})$. Then, a multirate extension of the Strang splitting (3.5) is given by the *impulse method* [61, Algorithm VIII.4.1] and reads

$$\boldsymbol{\Phi}_h = \boldsymbol{\Phi}_{h/2}^{\{\mathfrak{s}\}} \circ \left(\boldsymbol{\Phi}_{h/M}^{\{\mathfrak{f}\}}\right)^M \circ \boldsymbol{\Phi}_{h/2}^{\{\mathfrak{s}\}}. \tag{6.4}$$

It was already noted in [61] that multiple time stepping techniques can be iteratively extended to problems involving $N > 2$ partitions on different time scales. The idea is to further split the fast partition into $\boldsymbol{f}^{\{\mathfrak{f}\}}(\boldsymbol{x}) = \boldsymbol{f}^{\{\mathfrak{f},\mathfrak{f}\}}(\boldsymbol{x}) + \boldsymbol{f}^{\{\mathfrak{f},\mathfrak{s}\}}(\boldsymbol{x})$, and to replace the numerical integration scheme $\boldsymbol{\Phi}_h^{\{\mathfrak{f}\}}$ with another multiple time stepping method. This approach is widely utilized in molecular dynamics simulations [137, 123, 138] and is consistent with the splitting process of a hierarchical splitting method. We can incorporate this multiple time stepping approach into the framework of hierarchical splitting methods by introducing multirate splitting trees.

Definition 6.2 (multirate splitting tree)**.** A multirate splitting tree is a splitting tree $G = (V, E)$ with weighted edges where for any node $\boldsymbol{v} \in V \setminus \{\boldsymbol{r}\}$, the respective multirate factor $M^{\{\boldsymbol{v}\}} \in \mathbb{N}$ is assigned to the edge connecting the node $\boldsymbol{v}$ with its parent node $\texttt{parent}(\boldsymbol{v})$.

Based on a multirate splitting tree, a multirate extension of a hierarchical splitting method is obtained by substituting (5.4) with

$$\begin{aligned}\boldsymbol{\Phi}_h^{\{\boldsymbol{i}\}} = \left(\boldsymbol{\varphi}^{\{\boldsymbol{w}\}}_{b^{\{\boldsymbol{i}\}}_{s^{\{\boldsymbol{i}\}}} h/M^{\{\boldsymbol{w}\}}}\right)^{M^{\{\boldsymbol{w}\}}} \circ \left(\boldsymbol{\varphi}^{\{\boldsymbol{v}\}}_{a^{\{\boldsymbol{i}\}}_{s^{\{\boldsymbol{i}\}}} h/M^{\{\boldsymbol{v}\}}}\right)^{M^{\{\boldsymbol{v}\}}} \circ \dots \\ \circ \left(\boldsymbol{\varphi}^{\{\boldsymbol{w}\}}_{b^{\{\boldsymbol{i}\}}_1 h/M^{\{\boldsymbol{w}\}}}\right)^{M^{\{\boldsymbol{w}\}}} \circ \left(\boldsymbol{\varphi}^{\{\boldsymbol{v}\}}_{a^{\{\boldsymbol{i}\}}_1 h/M^{\{\boldsymbol{v}\}}}\right)^{M^{\{\boldsymbol{v}\}}},\end{aligned} \tag{6.5}$$

with $\boldsymbol{v} \coloneqq \texttt{left}(\boldsymbol{i}) \in V$ and $\boldsymbol{w} \coloneqq \texttt{right}(\boldsymbol{i}) \in V$. In practice, (6.5) differs from (5.4) only when numerical flow approximations are used. For any analytically computed flow $\boldsymbol{\varphi}_h^{\{\boldsymbol{\ell}\}}$, we can therefore select $M^{\{\boldsymbol{\ell}\}} = 1$.

Example 6.3. The generalization of the impulse method (6.4) to $N > 2$ partitions can be formulated as a multirate hierarchical splitting method where the Strang splitting (3.5) is assigned to all inner nodes $\boldsymbol{i} \in I$ of the multirate splitting tree in Fig. 6.2.

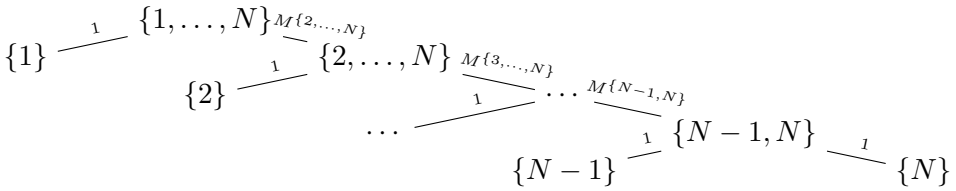


Fig. 6.2. Multirate splitting tree of a generalized impulse method for N-split systems.

Typically, we determine the multirate factor $M \in \mathbb{N}$ such that the step size h/M is sufficiently small to meet the stability and accuracy requirements. For any node $\boldsymbol{v} \in V \setminus \{\boldsymbol{r}\}$, the assigned method is employed with step sizes

$$\frac{\prod_{\boldsymbol{w} \in \mathcal{P}^{\{\boldsymbol{v}\}}} \delta^{\{\boldsymbol{v}\}}(\boldsymbol{w}) a_{j_{\boldsymbol{w}}}^{\{\boldsymbol{w}\}} + (1 - \delta^{\{\boldsymbol{v}\}}(\boldsymbol{w})) b_{j_{\boldsymbol{w}}}^{\{\boldsymbol{w}\}}}{\prod_{\boldsymbol{u} \in \bar{\mathcal{P}}^{\{\boldsymbol{v}\}} \setminus \{\boldsymbol{r}\}} M^{\{\boldsymbol{u}\}}}, \tag{6.6}$$

where $j_{\boldsymbol{w}} = 1, \dots, s^{\{\boldsymbol{w}\}}$ for all $\boldsymbol{w} \in \mathcal{P}^{\{\boldsymbol{v}\}}$. These step sizes can be vastly different in magnitude. For instance, the method `BABABABABAB` (A.46) is characterized by $\max_i |a_i| / \min_j |a_j| \approx 17.23$, and $\max_i |b_i| / \min_j |b_j| \approx 8.12$. The requirements for the multirate factor $M^{\{\boldsymbol{v}\}}$ are driven by the step size (6.6) of the largest modulus to sufficiently resolve the dynamics of the subsystem $\dot{\boldsymbol{x}} = \boldsymbol{f}^{\{\boldsymbol{v}\}}(\boldsymbol{x})$. For the sub-steps with step sizes (6.6) of smaller modulus, a smaller multirate factor $M^{\{\boldsymbol{v}\}}$ is already sufficient to meet the accuracy and stability requirements. As a result, the approach (6.5) may cause substantial computational overhead. This overhead can be mitigated by reweighting the multirate factors for each individual sub-step. Suppose a sub-step requires advancing the solution by $c \cdot h$, $c \in \mathbb{R} \setminus \{0\}$, and the desired step size is h/M, $M \in \mathbb{N}$. We aim to ensure that each sub-step utilizes a step size that is less than or equal to the desired step size h/M. At the same time, we seek a step size as close as possible to h/M to minimize computational overhead.

Lemma 6.4. *The minimum $\tilde{M} \in \mathbb{N}$ subject to the step size constraint $|c \cdot h| / \tilde{M} \leq h/M$ for a given constant $c \in \mathbb{R} \setminus \{0\}$, step size $h > 0$, and multirate factor $M \in \mathbb{N}$ is*

$$\tilde{M} \coloneqq \lceil |c| \cdot M \rceil. \tag{6.7}$$

Proof. The choice (6.7) is feasible since

$$\left| \frac{c \cdot h}{\tilde{M}} \right| = \frac{|c| \cdot h}{\lceil |c| \cdot M \rceil} \leq \frac{|c| \cdot h}{|c| \cdot M} = \frac{h}{M}.$$

For $\tilde{M} = 1$, optimality is evident. For $\tilde{M} > 1$,

$$\frac{h}{M} = \frac{|c| \cdot h}{|c| \cdot M} \leq \frac{|c| \cdot h}{\lfloor |c| \cdot M \rfloor}$$

proves the optimality of the choice (6.7). □

In particular, when computing a sub-step $\boldsymbol{\Phi}_{ch}^{\{\boldsymbol{v}\}}$ in node $\boldsymbol{v}$, the coefficient c reads

$$c^{\{\boldsymbol{v}\}}_{(j_{\boldsymbol{w}})_{\boldsymbol{w} \in \mathcal{P}^{\{\boldsymbol{v}\}}}} = \frac{\prod_{\boldsymbol{w} \in \mathcal{P}^{\{\boldsymbol{v}\}}} \delta^{\{\boldsymbol{v}\}}(\boldsymbol{w}) a^{\{\boldsymbol{w}\}}_{j_{\boldsymbol{w}}} + (1 - \delta^{\{\boldsymbol{v}\}}(\boldsymbol{w})) b^{\{\boldsymbol{w}\}}_{j_{\boldsymbol{w}}}}{\prod_{\boldsymbol{u} \in \mathcal{P}^{\{\boldsymbol{v}\}} \setminus \{\boldsymbol{r}\}} M^{\{\boldsymbol{u}\}}_{(j_{\boldsymbol{k}})_{\boldsymbol{k} \in \mathcal{P}^{\{\boldsymbol{u}\}}}}}, \tag{6.8}$$

where $M^{\{\boldsymbol{u}\}}_{(j_{\boldsymbol{k}})_{\boldsymbol{k} \in \mathcal{P}^{\{\boldsymbol{u}\}}}}$ denotes the reweighted multirate factors of the nodes in $\mathcal{P}^{\{\boldsymbol{v}\}}$. In general, we define

$$M^{\{\boldsymbol{v}\}}_{(j_{\boldsymbol{w}})_{\boldsymbol{w} \in \mathcal{P}^{\{\boldsymbol{v}\}}}} \coloneqq \left\lceil |c^{\{\boldsymbol{v}\}}_{(j_{\boldsymbol{w}})_{\boldsymbol{w} \in \mathcal{P}^{\{\boldsymbol{v}\}}}}| \cdot M^{\{\boldsymbol{v}\}} \right\rceil.$$

For brevity, we write $c^{\{\boldsymbol{v}\}}_{(j_{\boldsymbol{w}})}$ and $M^{\{\boldsymbol{v}\}}_{(j_{\boldsymbol{w}})}$ instead of $c^{\{\boldsymbol{v}\}}_{(j_{\boldsymbol{w}})_{\boldsymbol{w} \in \mathcal{P}^{\{\boldsymbol{v}\}}}}$ and $M^{\{\boldsymbol{v}\}}_{(j_{\boldsymbol{w}})_{\boldsymbol{w} \in \mathcal{P}^{\{\boldsymbol{v}\}}}}$, respectively.

Definition 6.5 (Multirate hierarchical splitting method)**.** A multirate hierarchical splitting method is a splitting method $\boldsymbol{\Phi}_h$ for N-split systems that is defined by a multirate splitting tree $G = (V, E)$ for N partitions such that

- to each inner node $\boldsymbol{i} \in I$, there is an assigned splitting method $\boldsymbol{\Phi}_h^{\{\boldsymbol{i}\}}$ for two-split systems. In particular, if a sub-step advances the solution of the subsystem $\dot{\boldsymbol{x}} = \boldsymbol{f}^{\{\boldsymbol{i}\}}(\boldsymbol{x})$ by a step of size $c \cdot h$ with $c \in \mathbb{R} \setminus \{0\}$, then the sub-step reads

$$\boldsymbol{\Phi}_{ch}^{\{\boldsymbol{i}\}} = \left(\boldsymbol{\varphi}^{\{\boldsymbol{w}\}}_{cb^{\{\boldsymbol{i}\}}_{s^{\{\boldsymbol{i}\}}} h \,/\, M^{\{\boldsymbol{w}\}}_{s^{\{\boldsymbol{i}\}}}}}\right)^{M^{\{\boldsymbol{w}\}}_{s^{\{\boldsymbol{i}\}}}} \circ \left(\boldsymbol{\varphi}^{\{\boldsymbol{v}\}}_{ca^{\{\boldsymbol{i}\}}_{s^{\{\boldsymbol{i}\}}} h \,/\, M^{\{\boldsymbol{v}\}}_{s^{\{\boldsymbol{i}\}}}}}\right)^{M^{\{\boldsymbol{v}\}}_{s^{\{\boldsymbol{i}\}}}} \circ \dots$$

$$\circ \left(\boldsymbol{\varphi}^{\{\boldsymbol{w}\}}_{cb_1^{\{i\}}h \big/ M_1^{\{\boldsymbol{w}\}}}\right)^{M_1^{\{\boldsymbol{w}\}}} \circ \left(\boldsymbol{\varphi}^{\{\boldsymbol{v}\}}_{ca_1^{\{i\}}h \big/ M_1^{\{\boldsymbol{v}\}}}\right)^{M_1^{\{\boldsymbol{v}\}}},$$

with $M_j^{\{\boldsymbol{v}\}} \coloneqq \left\lceil |ca_j^{\{i\}}| \cdot M^{\{\boldsymbol{v}\}} \right\rceil$ and $M_j^{\{\boldsymbol{w}\}} \coloneqq \left\lceil |cb_j^{\{i\}}| \cdot M^{\{\boldsymbol{w}\}} \right\rceil$ for $j = 1, \dots, s^{\{i\}}$;

- to each leaf node $\boldsymbol{\ell} \in L'$, a numerical integration scheme $\boldsymbol{\Phi}_h^{\{\boldsymbol{\ell}\}}$ is assigned;
- to each leaf node $\boldsymbol{\ell} \in L \setminus L'$, its exact flow $\boldsymbol{\varphi}_h^{\{\boldsymbol{\ell}\}}$ is assigned; we set $M^{\{\boldsymbol{\ell}\}} = 1$ and no reweighting is applied.

The computational process of a multirate hierarchical splitting method is illustrated in Algorithm 6.1.

The multiple time stepping approach affects the truncation error of the numerical integration scheme, but the qualitative statement on the convergence order remains unchanged.

Theorem 6.6. *Consider a multirate hierarchical splitting method $\boldsymbol{\Phi}_h$ with numerical integration schemes $\boldsymbol{\Phi}_h^{\{\boldsymbol{v}\}}$ of order $p^{\{\boldsymbol{v}\}}$ assigned to nodes $\boldsymbol{v} \in V$,*

$$\boldsymbol{\Phi}_h^{\{\boldsymbol{v}\}} = \exp\left(h\mathcal{L}^{\{\boldsymbol{v}\}} + h^{p^{\{\boldsymbol{v}\}}+1}\mathcal{O}^{\{\boldsymbol{v}\}}_{p^{\{\boldsymbol{v}\}}+1} + \mathcal{O}(h^{p^{\{\boldsymbol{v}\}}+2})\right) \mathrm{Id},$$

with leading-order error terms $\mathcal{O}^{\{\boldsymbol{v}\}}_{p^{\{\boldsymbol{v}\}}+1}$ and let $\delta^{\{\boldsymbol{v}\}}$ denote the mappings defined in (5.7)*. Then, $\boldsymbol{\Phi}_h$ is at least convergent of order $p \coloneqq \min_{\boldsymbol{v} \in I \dot\cup L'} p^{\{\boldsymbol{v}\}}$,*

$$\boldsymbol{\Phi}_h = \exp\left(h\mathcal{L} + h^{p+1}\mathcal{O}_{p+1} + \mathcal{O}(h^{p+2})\right),$$

with leading-order error term

$$\mathcal{O}_{p+1} = \sum_{\substack{\boldsymbol{v} \in I \dot\cup L' \\ p^{\{\boldsymbol{v}\}} = p}} \left[\sum_{(j_{\boldsymbol{w}}) \in \mathcal{J}^{\{\boldsymbol{v}\}}} \frac{\left(\prod_{\boldsymbol{w} \in \mathcal{P}^{\{\boldsymbol{v}\}}} \delta^{\{\boldsymbol{v}\}}(\boldsymbol{w}) a_{j_{\boldsymbol{w}}}^{\{\boldsymbol{w}\}} + (1 - \delta^{\{\boldsymbol{v}\}}(\boldsymbol{w})) b_{j_{\boldsymbol{w}}}^{\{\boldsymbol{w}\}}\right)^{p+1}}{\left(\prod_{\boldsymbol{w} \in \bar{\mathcal{P}}^{\{\boldsymbol{v}\}} \setminus \{\boldsymbol{r}\}} M_{(j_{\boldsymbol{u}})}^{\{\boldsymbol{w}\}}\right)^{p}} \right] \mathcal{O}^{\{\boldsymbol{v}\}}_{p+1},$$

where $\mathcal{J}^{\{\boldsymbol{v}\}} \coloneqq \left\{ (j_{\boldsymbol{w}})_{\boldsymbol{w} \in \mathcal{P}^{\{\boldsymbol{v}\}}} \,:\, 1 \le j_{\boldsymbol{w}} \le s^{\{\boldsymbol{w}\}} \text{ for all } \boldsymbol{w} \in \mathcal{P}^{\{\boldsymbol{v}\}} \right\}$.

Proof. The root node $\boldsymbol{r}$ is not impacted by the multiple time stepping approach as the assigned splitting method $\boldsymbol{\Phi}_h^{\{\boldsymbol{r}\}}$ still computes a single time step of step size h. Additionally, all nodes $\boldsymbol{\ell} \in L \setminus L'$ remain unaffected, as the exact flow $\boldsymbol{\varphi}_h^{\{\boldsymbol{\ell}\}}$ is assigned to these nodes. Consider any node $\boldsymbol{v} \in I \dot\cup L'$ where a numerical integration scheme $\boldsymbol{\Phi}_h^{\{\boldsymbol{v}\}}$ is assigned. At any occurrence, identified by a tuple $(j_{\boldsymbol{w}}) \in \mathcal{J}^{\{\boldsymbol{v}\}}$, the method computes $M_{(j_{\boldsymbol{w}})}^{\{\boldsymbol{v}\}}$ steps of step size $c_{(j_{\boldsymbol{w}})}^{\{\boldsymbol{v}\}} h / M_{(j_{\boldsymbol{w}})}^{\{\boldsymbol{v}\}}$. These consecutive applications can be formally expressed as

$$\begin{aligned}
\boldsymbol{\Psi}^{\{\boldsymbol{v}\}}_{c_{(j_{\boldsymbol{w}})}^{\{\boldsymbol{v}\}} h} &\coloneqq \left(\boldsymbol{\Phi}^{\{\boldsymbol{v}\}}_{c_{(j_{\boldsymbol{w}})}^{\{\boldsymbol{v}\}} h / M_{(j_{\boldsymbol{w}})}^{\{\boldsymbol{v}\}}}\right)^{M_{(j_{\boldsymbol{w}})}^{\{\boldsymbol{v}\}}} \\
&= \exp\left(c_{(j_{\boldsymbol{w}})}^{\{\boldsymbol{v}\}} h \mathcal{L}^{\{\boldsymbol{v}\}} + M_{(j_{\boldsymbol{w}})}^{\{\boldsymbol{v}\}} \left(\frac{c_{(j_{\boldsymbol{w}})}^{\{\boldsymbol{v}\}} h}{M_{(j_{\boldsymbol{w}})}^{\{\boldsymbol{v}\}}} \right)^{p^{\{\boldsymbol{v}\}}+1} \mathcal{O}^{\{\boldsymbol{v}\}}_{p^{\{\boldsymbol{v}\}}+1} + \mathcal{O}(h^{p^{\{\boldsymbol{v}\}}+2}) \right) \mathrm{Id}.
\end{aligned}$$

We can extend the iterative process outlined in the proof of Theorem 5.6 as follows: between steps 1) and 2), we replace the methods $\boldsymbol{\Phi}_h^{\{\boldsymbol{v}\}}$ with $\boldsymbol{\Psi}_h^{\{\boldsymbol{v}\}}$ for all $\boldsymbol{v} \in V \setminus \{\boldsymbol{r}\}$.

In the proof of Theorem 5.6, the expression (5.8) relied on the interchangeability of the sum and product. As the reweighted multirate factors are generally different, we can no longer interchange the sum and product signs. Consequently, the leading error term of the intermediate steps (5.10) becomes

$$\tilde{\mathcal{O}}_{p+1} = \sum_{\substack{\boldsymbol{v}\in\boldsymbol{A}\dot{\cup}L' \\ p^{\{\boldsymbol{v}\}}=p}} \left[\sum_{(j_{\boldsymbol{w}})\in\mathcal{J}^{\{\boldsymbol{v}\}}} \frac{\left(\prod_{\boldsymbol{w}\in\mathcal{P}^{\{\boldsymbol{v}\}}} \delta^{\{\boldsymbol{v}\}}(\boldsymbol{w}) a_{j_{\boldsymbol{w}}}^{\{\boldsymbol{w}\}} + (1-\delta^{\{\boldsymbol{v}\}}(\boldsymbol{w})) b_{j_{\boldsymbol{w}}}^{\{\boldsymbol{w}\}}\right)^{p+1}}{\left(\prod_{\boldsymbol{w}\in\bar{\mathcal{P}}^{\{\boldsymbol{v}\}}\setminus\{\boldsymbol{r}\}} M_{(j_{\boldsymbol{u}})}^{\{\boldsymbol{w}\}}\right)^{p}} \right] \mathcal{O}_{p+1}^{\{\boldsymbol{v}\}}.$$

The remainder of this proof is identical to the proof of Theorem 5.6. □

Remark 6.7. The use of the reweighting (6.7) presents challenges for composition techniques. In contrast to (singlerate) hierarchical splitting methods, a composition (3.11) based on a multirate hierarchical splitting method is not equivalent to applying the composition (3.11) to the method $\boldsymbol{\Phi}_h^{\{\boldsymbol{r}\}}$ assigned to the root node. In a composition of a multirate hierarchical splitting method, each application of the base scheme utilizes the same reweighted multirate factors, whereas applying the composition to the root node would include the weights γ_i into the reweighting, resulting in generally different multirate factors for each application of the base method. Similar issues arise when considering compositions of the form (3.15).

Remark 6.8. As the number of applications of a method $\boldsymbol{\Phi}_h^{\{\boldsymbol{v}\}}$ grows exponentially with the distance of $\boldsymbol{v}$ to $\boldsymbol{r}$, a general guideline is to assign elementary subsystems $\dot{\boldsymbol{x}} = \boldsymbol{f}^{\{m\}}(\boldsymbol{x})$ that require a smaller step size to leaf nodes with a greater distance to the root node. Consequently, a natural assumption is that the multirate factors along a path from the root node $\boldsymbol{r}$ to a node $\boldsymbol{v}$ are monotonically increasing.

6.3. The Concept of Computational Order

In the limit $M^{\{\boldsymbol{i}\}} \to \infty$, the subsystem $\dot{\boldsymbol{x}} = \boldsymbol{f}^{\{\boldsymbol{i}\}}(\boldsymbol{x})$ will be integrated exactly for any node $\boldsymbol{i} \in I \setminus \{\boldsymbol{r}\}$, provided that all methods assigned to nodes in this branch of the multirate splitting tree are consistent. Informally, in this case, the convergence order of the methods assigned to nodes within this sub-tree does not impact the convergence order of $\boldsymbol{\Phi}_h$. Therefore, we can assign lower-order schemes to nodes of the sub-tree without altering the convergence order of the overall method. In practice, however, we will not perform the limit $M^{\{\boldsymbol{i}\}} \to \infty$. Nonetheless, we would like to impose conditions on the multirate factors ensuring that the scaling behavior for step sizes $h \geq h_{\texttt{min}}$ is superior to the expected scaling based on the order theory. We refer to this as the *computational order* of a multirate integration scheme. The conditions on the multirate factors are derived in the following lemma.

Lemma 6.9. *Let $h > 0$, $c \in \mathbb{R} \setminus \{0\}$, $M \in \mathbb{N}$, and integers $q < p$. It holds*

$$\left| \frac{(ch)^{q+1}}{\lceil |c| \cdot M \rceil^q} \right| = \mathcal{O}(h^{p+1}),$$

provided that the multirate factor is bounded from below by $h^{(q-p)/q} \leq M$.

Proof. By Lemma 6.4 and the lower bound on M, we obtain

$$\begin{aligned} \left| \frac{(ch)^{q+1}}{\lceil |c| \cdot M \rceil^q} \right| &= |c| \cdot \left(\frac{|c| \cdot h}{\lceil |c| \cdot M \rceil} \right)^q h \leq |c| \cdot \left(\frac{h}{M} \right)^q h = |c| \cdot \frac{h^{q+1}}{M^q} \\ &\leq |c| \cdot (h^{(p-q)/q})^q \cdot h^{q+1} = |c| \cdot h^{p+1} = \mathcal{O}(h^{p+1}), \end{aligned}$$

proving the statement of this lemma. □

Algorithm 6.1: Multirate hierarchical splitting method

function STEP$(G, \boldsymbol{x}, c, h, \boldsymbol{v})$

```
/* Applies the method assigned to node v of the multirate
   splitting tree G to advance the state x by a step of size ch.
   A full time step of the multirate hierarchical splitting
   method is performed for c = 1 and v = r.                    */
if v = r then
    M̃ ← 1
else
    M̃ ← ⌈|c| · M^{v}⌉
if v ∈ I then
    for m = 1, ..., M̃ do
        for j = 1, ..., s^{v} do
            x ← STEP(G, x, a_j^{v} c/M̃, h, left(v))
            x ← STEP(G, x, b_j^{v} c/M̃, h, right(v))
else if v ∈ L' then
    for m = 1, ..., M̃ do
        x ← Φ_{ch/M̃}^{v}(x)
else
    x ← φ_{ch}^{v}(x)
return x
```

Roughly speaking, if we replace for all nodes $\boldsymbol{v} \in V \setminus \{\boldsymbol{r}\}$ the factor $M^{\{\boldsymbol{v}\}}$ with

$$M^{\{\boldsymbol{v}\}}(h) = h^{(p^{\{\boldsymbol{v}\}} - p^{\{\boldsymbol{r}\}})/p^{\{\boldsymbol{v}\}}} \tag{6.9}$$

where $p^{\{\boldsymbol{r}\}}$ is the order of the method in root node $\boldsymbol{r}$, the multirate hierarchical splitting method scales with order $p^{\{\boldsymbol{r}\}}$. To maintain a choice of multirate factors independent of h and p, we arrive at the following definition of computational order.

Definition 6.10. A multirate hierarchical splitting method $\boldsymbol{\Phi}_h$ exhibits a computational order of p for step sizes $h \geq h_{\texttt{min}}$, provided that

- $\boldsymbol{\Phi}_h^{\{\boldsymbol{r}\}}$ has order $p^{\{\boldsymbol{r}\}} = p$;
- for each node $\boldsymbol{v} \in V \setminus \{\boldsymbol{r}\}$ with $p^{\{\boldsymbol{v}\}} < p^{\{\boldsymbol{r}\}}$, it holds $M^{\{\boldsymbol{v}\}} \geq h_{\texttt{min}}^{(p^{\{\boldsymbol{v}\}} - p^{\{\boldsymbol{r}\}})/p^{\{\boldsymbol{v}\}}}$.

A multirate hierarchical splitting method of computational order p is expected to be more accurate for step sizes $h \geq h_{\texttt{min}}$ compared to its version employing multirate factors (6.9), which scales with order p. In molecular dynamics simulations, this concept has already been applied[1].

[1] In [124], the notion of computational order was referred to as *effective order*. As the term *effective order* [25] is already used in the context of processed splitting methods [12], we use the term computational order to avoid ambiguity.

6.4. Numerical Results

To illustrate the effects of the multiple time stepping approach, we conduct numerical simulations for the four body-problem (6.3). As initial values, we choose

$$\begin{aligned} &\boldsymbol{p}_0(0) = (0,0)^\top, \quad \boldsymbol{p}_1(0) = (0,m_1)^\top, \quad \boldsymbol{p}_2(0) = (0,m_2/2)^\top, \quad \boldsymbol{p}_3(0) = (0,0)^\top, \\ &\boldsymbol{q}_0(0) = (0,0)^\top, \quad \boldsymbol{q}_1(0) = (1,0)^\top, \quad \boldsymbol{q}_2(0) = (4,0)^\top, \quad \boldsymbol{q}_3(0) = (1.01,0)^\top. \end{aligned}$$

We split the Hamiltonian into the three partitions (6.1). We define multirate hierarchical splitting methods by considering the multirate splitting tree depicted in Fig. 6.3. For the time interval $t \in [0,40]$, numerical simulations are conducted for the hierarchical splitting methods defined `HOMF4`, `COMP4`, and `Hs8p6` in Tab. 6.1. For comparison, we also perform numerical tests with state-of-the-art composition schemes. Firstly, we consider the fourth-order composition (4.39) with the Lie–Trotter splitting

$$\boldsymbol{\Phi}_h = \left(\boldsymbol{\varphi}_{h/M}^{\{\mathcal{T}\}} \circ \boldsymbol{\varphi}_{h/M}^{\{\mathcal{V}^{\{\mathrm{f}\}}\}}\right)^M \circ \boldsymbol{\varphi}_h^{\{\mathcal{V}^{\{\mathrm{s}\}}\}} \tag{6.10}$$

as the underlying base scheme (`BMs6p4`). Secondly, we consider a sixth-order composition scheme of the form (3.15) with coefficients

$$\begin{aligned} &\alpha_1 = \beta_{10} = 0.05026276440039221, && \alpha_2 = \beta_9 = 0.31496061692769417, \\ &\alpha_3 = \beta_8 = 0.49242637248987586, && \alpha_4 = \beta_7 = 0.23706391397812188, \\ &\alpha_5 = \beta_6 = 0.34635818985072686, && \alpha_6 = \beta_5 = -0.36276277925434486, \\ &\alpha_7 = \beta_4 = 0.19560248860005314, && \alpha_8 = \beta_3 = -0.42511876779769087, \\ &\alpha_9 = \beta_2 = -0.44734648269547816, && \alpha_{10} = \beta_1 = 0.0985536835006498, \end{aligned} \tag{6.11}$$

and the Lie–Trotter splitting (6.10) as the underlying base scheme (`BMs10p6`). The coefficients (6.11) are obtained by transforming the coefficients a_j, b_j from [13] into coefficients α_j, β_j of the composition (3.15) via [92, eq. (2.6)]. By Lemma 5.13, both `BMs6p4` and `BMs10p6` can be represented as hierarchical splitting methods.

Integrator ID	node $\mathcal{H}$	node $\mathcal{T} + \mathcal{V}^{\{\mathrm{f}\}}$
`HOMF4`	`OMF4`	`OMF4`
`COMP4`	`OMF4`	`Strang` (3.5)
`Hs8p6`	`s8p6`	`s8p6`

Tab. 6.1. Four body-problem. Setup of three hierarchical splitting methods based on the multirate splitting tree from Fig. 6.3, a fourth-order method `HOMF4` based on the `OMF4` method (A.46), a second-order method `COMP4` that, for sufficiently large M, exhibits a computational order of four, and a sixth-order method `Hs8p6` based on the eight-stage method (4.40) of order $p = 6$. To all leaf nodes, the exact flows of the subsystems are assigned.

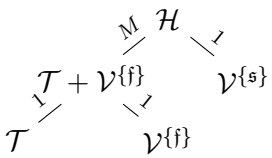


Fig. 6.3. Multirate splitting tree for the four body-problem (6.3).

The numerical results presented in Fig. 6.4a for both composition schemes demonstrate an order reduction to order two[2] when applying the reweighting (6.7) to the multirate factors. These observations are consistent with Remark 6.7.

For the hierarchical splitting method `HOMF4`, we compare the variant with a constant multirate factor $M = 9$ with the variant that applies the reweighting (6.7) to $M = 15$.

[2] Due to the inherent symmetry of the splitting method, the order is two instead of the order of the underlying base scheme ($p = 1$).

This comparison is motivated by the observation that $\max_j \lceil |a_j| \cdot 15 \rceil = 9$ for the integrator coefficients a_j from (A.46), so that both variants yield the same maximum number of sub-steps for the fast subsystem $\mathcal{T} + \mathcal{V}^{\{\mathrm{f}\}}$. The numerical results presented in Fig. 6.4b demonstrate that the reweighting (6.7) substantially enhances the efficiency of the integrator `HOMF4`. Although reweighting the multirate factors results in an order reduction when applied to composition techniques, it contributes to minimizing computational overhead for hierarchical splitting methods that are not based on a composition of a lower-order base method.

The concept of computational order is illustrated in Fig. 6.4c, which shows results for the `COMP4` method with the reweighting (6.7) applied for varying M. The numerical results demonstrate that selecting a step size-dependent multirate factor of $M = 1/h$ yields a method that scales with order $p = 4$, as expected. For fixed multirate factors M, we observe that the method is more accurate for step sizes $h \geq 1/M$.

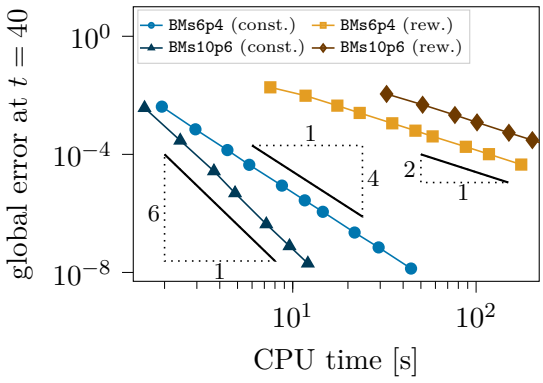
Fig. 6.4d shows a work-precision diagram comparing the three fourth-order methods `HOMF4` (reweighted $M = 15$), `COMP4` (reweighted $M = 400$), and `BMs6p4` (constant $M = 3$), as well as the sixth-order methods `Hs8p6` (reweighted $M = 7$) and `BMs10p6` (constant $M = 3$). The multirate factors are chosen such that each variant achieves the most efficient computational process according to the work-precision diagram in this benchmark. The work-precision diagram demonstrates that the two hierarchical splitting methods `HOMF4` and `Hs8p6` constitute the most efficient choices among all considered methods of order four and six, respectively. In comparison to `BMs6p4`, `HOMF4` achieves a specified error with approximately 27% of the computational cost. The hierarchical splitting method `Hs8p6` only requires approximately 58% of the computational cost to be as accurate as `BMs10p6`. The values correspond to CPU time ratios at a representative error level. For error tolerances of `TOL` $\geq 10^{-8}$, the hierarchical splitting method `HOMF4` is more efficient compared to the sixth-order composition scheme `BMs10p6`.

In summary, the numerical results indicate that the hierarchical splitting approach facilitates the derivation of more efficient splitting methods in comparison to state-of-the-art composition approaches.

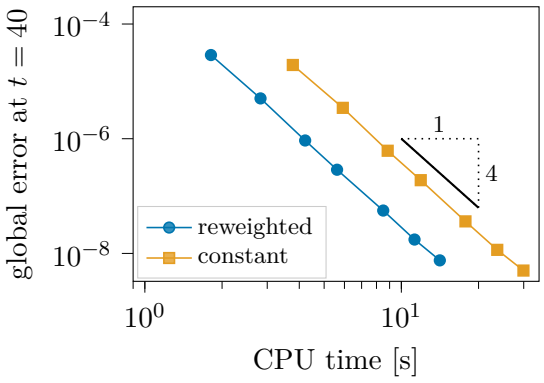
6.5. Summary

In this chapter, we presented an extension of hierarchical splitting methods incorporating multiple time stepping techniques which retain the structure-preserving properties and inherit the order theory of their singlerate counterparts. We proposed a reweighting technique for the multirate factors to reduce computational overhead and introduced the concept of computational order, deriving sufficient conditions under which superior accuracy is observed over a restricted step size range. We developed multirate hierarchical splitting methods for separable Hamiltonian systems with multirate potential and applied them to a four-body problem, demonstrating that the reweighting of the multirate factors contributes to reducing computational overhead. However, we have also shown that the reweighting causes an order reduction when applied to state-of-the-art composition approaches, so that constant multirate factors must be employed to retain the convergence order. This constitutes a substantial advantage of multirate hierarchical splitting methods compared to state-of-the-art composition methods.

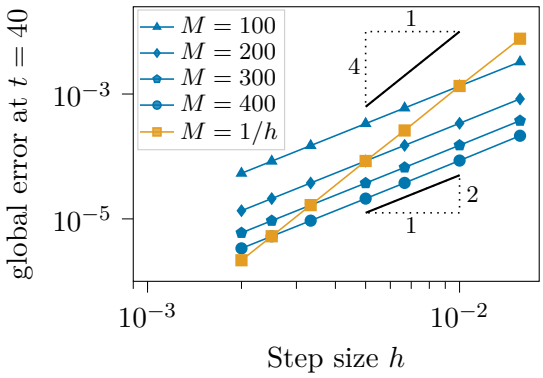
The numerical results presented in this chapter considered a separable Hamiltonian system of the form (3.32), such that (Hessian-free) force-gradient integrators constitute efficient choices for their numerical integration. In the next chapter, we will combine the hierarchical splitting approach with (Hessian-free) force-gradient integrators, thereby obtaining commutator-based approaches that exploit the splitting of the potential $\mathcal{V}$ via multiple time stepping.



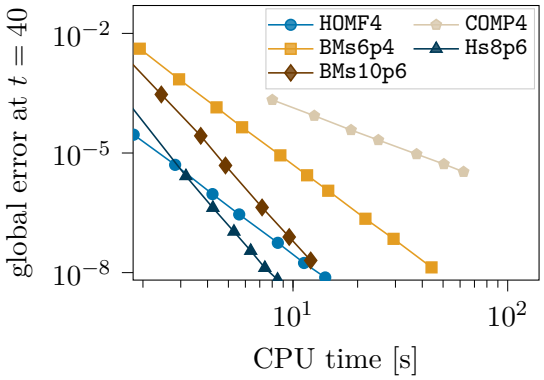
(a) Composition schemes `BMs6p4` and `BMs10p6`. Comparison for $M = 3$ when applying the reweighting (6.7) vs. no reweighting.



(b) `HOMF4` method. Comparison of the variants with $M = 15$ (with reweighting) and $M = 9$ (no reweighting).



(c) `COMP4` method. Comparison of the method for different multirate factors $M \in \{100, 200, 300, 400, 1/h\}$ while applying the reweighting (6.7).



(d) Work-precision diagram for the methods `HOMF4` ($M = 15$, rew.), `COMP4` ($M = 400$, rew.), `Hs8p6` ($M = 7$, rew.), `BMs6p4` ($M = 3$, const.), and `BMs10p6` ($M = 3$, const.).

Fig. 6.4. Four body-problem. Numerical results for (a) the composition methods `BMs6p4` and `BMs10p6`, (b) the fourth-order method `HOMF4`, (c) the `COMP4` scheme, and (d) a work-precision diagram comparing the global error at $t = 40$ vs. CPU time (in seconds).

Chapter 7.

Hierarchical Force-Gradient Integrators

In Chapter 4, we investigated force-gradient integrators and their Hessian-free counterparts for the numerical solution of separable Hamiltonian systems (3.32) as commutator-based splitting approaches that split the Hamiltonian into its kinetic and potential parts. Both force-gradient integrators and their Hessian-free counterparts are splitting methods specifically designed for two-split systems. In contrast, the preceding Chapters 5 and 6 discussed (multirate) hierarchical splitting methods for N-split systems.

In this chapter, we will combine the force-gradient approach with the hierarchical splitting approach, thereby introducing a novel class of *hierarchical (Hessian-free) force-gradient integrators* for separable Hamiltonian systems with fine structure

$$\mathcal{H}(\boldsymbol{p},\boldsymbol{q}) = \mathcal{T}(\boldsymbol{p}) + \mathcal{V}(\boldsymbol{q}) = \frac{1}{2}\boldsymbol{p}^\top \boldsymbol{M}^{-1}\boldsymbol{p} + \sum_{m=1}^{N-1} \mathcal{V}^{\{m\}}(\boldsymbol{q}). \tag{7.1}$$

In (7.1), the potential energy is split into $N-1$ components ($N \geq 2$), where the parts $\mathcal{V}^{\{m\}}$ are ordered such that their respective forces are characterized by increasing contribution to the dynamics of the solution and simultaneously decreasing evaluation cost. This introduces a source of multirate potential, which motivates the use of multiple time stepping techniques, such as multirate hierarchical splitting methods.

7.1. Splitting Trees for Separable Hamiltonian Systems

Our objective is to develop hierarchical splitting methods for separable Hamiltonian systems of the form (7.1), which are explicit by exploiting the separability of the Hamiltonian. Since the Lie derivatives corresponding to parts of the potential energy commute, $[\hat{\mathcal{V}}^{\{m\}}, \hat{\mathcal{V}}^{\{k\}}] = 0$, further splitting a subsystem that contains no kinetic energy yields no change in the numerical approximation. Based on this observation and the guideline established in Chapter 6 – that computationally expensive subsystems should be assigned to leaf nodes closer to the root node – we arrive at the multirate splitting tree depicted in Fig. 7.1.

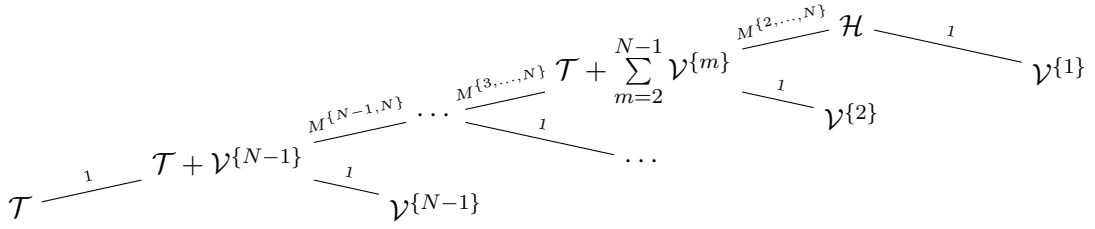


Fig. 7.1. (Multirate) splitting tree for the splitting of separable Hamiltonian systems of the form (7.1).

Remark 7.1 (Nested integration)**.** Neglecting the possibility of applying the reweighting equation (6.7) to the multirate factors M, the splitting process depicted in Fig. 7.1 aligns with the nested integration approach [123, 138] that is widely utilized for gauge field generation in lattice QCD. In a nested integration scheme, splitting methods for two-split

systems are assigned to inner nodes of the splitting tree depicted in Fig. 7.1, while the flows of the elementary subsystems are computed exactly.

To the best of our knowledge, the nested integration framework lacks a comprehensive error analysis. By reformulating nested integration schemes as hierarchical splitting methods, the order theory of Chapter 5 and the error analysis of Chapter 6 become directly applicable. Consequently, the hierarchical splitting framework offers, for the first time, a comprehensive error analysis for nested integration schemes.

7.2. Hierarchical (Hessian-Free) Force-Gradient Integrators

When the splitting tree from Fig. 7.1 is applied to (7.1), the Lie derivatives at each node possess a structure that is consistent with the force-gradient approach. This generalizes the special case $N = 2$, where the potential is not split into different parts. Given an inner node $\boldsymbol{i}$ and its child nodes $\boldsymbol{c_\ell} \coloneqq \texttt{left}(\boldsymbol{i})$ and $\boldsymbol{c_r} \coloneqq \texttt{right}(\boldsymbol{i})$, the force-gradient term (4.1) of the splitting method $\boldsymbol{\Phi}_h^{\{i\}}$ takes the form

$$[\mathcal{L}^{\{c_r\}}, [\mathcal{L}^{\{c_\ell\}}, \mathcal{L}^{\{c_r\}}]] \tag{7.2}$$

with the Lie derivatives

$$\mathcal{L}^{\{c_r\}} = -\nabla \mathcal{V}^{\{m\}}(\boldsymbol{q}) \frac{\partial}{\partial \boldsymbol{p}} \quad \text{and} \quad \mathcal{L}^{\{c_\ell\}} = \boldsymbol{p}^\top \boldsymbol{M}^{-1} \frac{\partial}{\partial \boldsymbol{q}} - \sum_{k=m+1}^{N-1} \nabla \mathcal{V}^{\{k\}}(\boldsymbol{q}) \frac{\partial}{\partial \boldsymbol{p}}$$

for some $m = 1, \dots, N-1$. This generalized force-gradient term simplifies to the conventional force-gradient term with the potential part assigned to the node $\boldsymbol{c_r}$, as the following theorem shows.

Theorem 7.2. *The commutator* (7.2) *simplifies to*

$$[\hat{\mathcal{V}}^{\{m\}}, [\hat{\mathcal{T}}, \hat{\mathcal{V}}^{\{m\}}]], \tag{7.3}$$

where $\hat{\mathcal{T}} \coloneqq \boldsymbol{p}^\top \boldsymbol{M}^{-1} \frac{\partial}{\partial \boldsymbol{q}}$ *and* $\hat{\mathcal{V}}^{\{m\}} \coloneqq -\nabla \mathcal{V}^{\{m\}}(\boldsymbol{q}) \frac{\partial}{\partial \boldsymbol{p}}$.

Proof. For the commutator (7.3), a straightforward computation, analogous to (4.1), yields

$$[\hat{\mathcal{V}}^{\{m\}}, [\hat{\mathcal{T}}, \hat{\mathcal{V}}^{\{m\}}]] = 2 \nabla \mathcal{V}^{\{m\}}(\boldsymbol{q}) \boldsymbol{M}^{-1} \nabla^2 \mathcal{V}^{\{m\}}(\boldsymbol{q}) \frac{\partial}{\partial \boldsymbol{p}}.$$

It remains to show that (7.2) simplifies to the same expression. Expanding the commutator results in

$$[\mathcal{L}^{\{c_r\}}, [\mathcal{L}^{\{c_\ell\}}, \mathcal{L}^{\{c_r\}}]] = 2\mathcal{L}^{\{c_r\}} \mathcal{L}^{\{c_\ell\}} \mathcal{L}^{\{c_r\}} - \mathcal{L}^{\{c_r\}} \mathcal{L}^{\{c_r\}} \mathcal{L}^{\{c_\ell\}} - \mathcal{L}^{\{c_\ell\}} \mathcal{L}^{\{c_r\}} \mathcal{L}^{\{c_r\}}.$$

We investigate the individual terms in more detail and find

$$2\mathcal{L}^{\{c_r\}} \mathcal{L}^{\{c_\ell\}} \mathcal{L}^{\{c_r\}} = 2\hat{\mathcal{V}}^{\{m\}} \hat{\mathcal{T}} \hat{\mathcal{V}}^{\{m\}} + \underbrace{2\, \hat{\mathcal{V}}^{\{m\}} \left(\sum_{k=m+1}^{N-1} \hat{\mathcal{V}}^{\{k\}} \right) \hat{\mathcal{V}}^{\{m\}}}_{=0} = 2\hat{\mathcal{V}}^{\{m\}} \hat{\mathcal{T}} \hat{\mathcal{V}}^{\{m\}},$$

$$\mathcal{L}^{\{c_r\}} \mathcal{L}^{\{c_r\}} \mathcal{L}^{\{c_\ell\}} = \underbrace{\hat{\mathcal{V}}^{\{m\}} \hat{\mathcal{V}}^{\{m\}} \hat{\mathcal{T}}}_{=0} + \underbrace{\hat{\mathcal{V}}^{\{m\}} \hat{\mathcal{V}}^{\{m\}} \left(\sum_{k=m+1}^{N-1} \hat{\mathcal{V}}^{\{k\}} \right)}_{=0} = 0,$$

$$\mathcal{L}^{\{\boldsymbol{c}_\ell\}}\mathcal{L}^{\{\boldsymbol{c}_r\}}\mathcal{L}^{\{\boldsymbol{c}_r\}} = \underbrace{\left(\sum_{k=m+1}^{N-1}\hat{\mathcal{V}}^{\{k\}}\right)\hat{\mathcal{V}}^{\{m\}}\hat{\mathcal{V}}^{\{m\}}}_{=0} + \underbrace{\hat{\mathcal{T}}\hat{\mathcal{V}}^{\{m\}}\hat{\mathcal{V}}^{\{m\}}}_{=0} = 0,$$

such that we obtain

$$[\mathcal{L}^{\{\boldsymbol{c}_r\}},[\mathcal{L}^{\{\boldsymbol{c}_\ell\}},\mathcal{L}^{\{\boldsymbol{c}_r\}}]] = 2\hat{\mathcal{V}}^{\{m\}}\hat{\mathcal{T}}\hat{\mathcal{V}}^{\{m\}} = [\hat{\mathcal{V}}^{\{m\}},[\hat{\mathcal{T}},\hat{\mathcal{V}}^{\{m\}}]],$$

concluding the proof of the theorem. □

As a consequence, evaluating the force-gradient term in $\boldsymbol{\Phi}_h^{\{i\}}$ depends only on the potential $\mathcal{V}^{\{m\}}$ associated with node $\boldsymbol{c_r}$. In other words, the force-gradient update (4.4) evaluates the same force (and its gradient) as conventional hierarchical splitting methods. The same local structure is maintained by the Hessian-free force-gradient updates (4.10). Therefore, we can assign (Hessian-free) force-gradient integrators to inner nodes of the splitting tree from Fig. 7.1 without encountering any structural difficulties.

By assigning (Hessian-free) force-gradient integrators to inner nodes of the splitting tree, we obtain hierarchical splitting methods that we refer to as *hierarchical (Hessian-free) force-gradient integrators*.

In contrast to nested integrators [123, 138], this novel class offers two significant advantages. First, it enables the incorporation of the (Hessian-free) force-gradient approach, which has proven to provide particularly efficient integrator variants. Second, the hierarchical splitting framework introduced the reweighting (6.7) of the multirate factors, thereby contributing to the reduction of computational overhead.

7.3. Numerical Results

We present numerical simulations for the four body-problem of Chapter 6, in which the potential is split into $N-1=2$ parts. In Chapter 11, we will design and apply hierarchical Hessian-free force-gradient integrators for gauge field generation in lattice QCD where the potential is split into $N-1>2$ parts.

For the four body-problem, we adopt the same setup as in Chapter 6, i.e., we split the Hamiltonian (6.3) into the three components (6.1) and consider the splitting tree from Fig. 6.3 that corresponds to the splitting tree Fig. 7.1 with $N=3$. We define a fourth-order hierarchical force-gradient integrator by assigning the force-gradient integrator `BACAB` (4.6) to both inner nodes of the splitting tree. Analogously, we obtain a fourth-order hierarchical Hessian-free force-gradient integrator by assigning the Hessian-free force-gradient integrator `BADAB` (A.9) to both inner nodes. We refer to the resulting methods as `H-BACAB` and `H-BADAB`, respectively.

We repeat the numerical tests presented in Chapter 6 utilizing the hierarchical (Hessian-free) force-gradient integrators `H-BACAB` and `H-BADAB`. The results are presented in Fig. 7.2, demonstrating that the (Hessian-free) force-gradient approach facilitates a more efficient computational process in comparison to the `HOMF4` method discussed in the preceding section. In particular, the Hessian-free counterpart `H-BADAB` yields the most efficient computational process among all fourth-order integrators under consideration. More specifically, investigating CPU time ratios at representative error levels yields that `H-BADAB` requires only approximately 70% of the computational cost to achieve a similar accuracy as `HOMF4`.

The underlying integrators `BACAB` and `BADAB` are state-of-the-art choices among (Hessian-free) force-gradient integrators of order $p=4$. It is anticipated that even more efficient hierarchical (Hessian-free) force-gradient integrators can be obtained by

selecting alternative non-dominated integrator variants from Tab. 4.1 and Tab. 4.2, respectively.

The Hessian-free force-gradient approach is generally not symplectic and therefore does not preserve a nearby modified Hamiltonian. The error in the Hamiltonian, as depicted in Fig. 7.3a, does not exhibit a visible drift. The linear energy drift becomes visible when plotting the difference in the Hamiltonian for the numerical solutions obtained by the methods `H-BACAB` and `H-BADAB`, as depicted in Fig. 7.3b. Compared to the energy error from Fig. 7.3a, the differences in Fig. 7.3b are of smaller magnitude within the time interval $t \in [0, 5000]$, explaining why the drift is not noticeable when investigating the error in the Hamiltonian. Over exponentially long time intervals, the linear energy drift will also become noticeable while investigating the error in the Hamiltonian. In such situations, the slightly more expensive hierarchical force-gradient integrator `H-BACAB` may be the preferred choice.

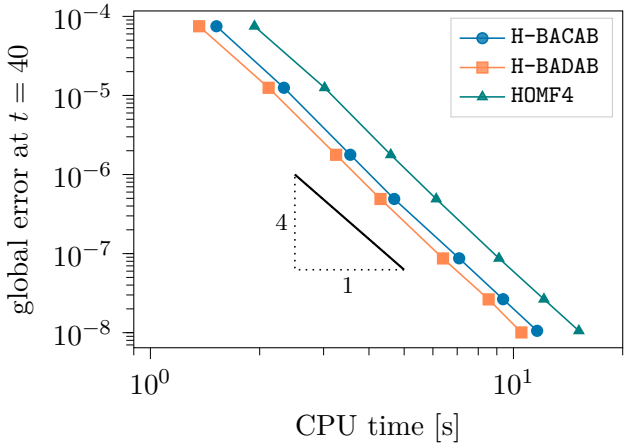
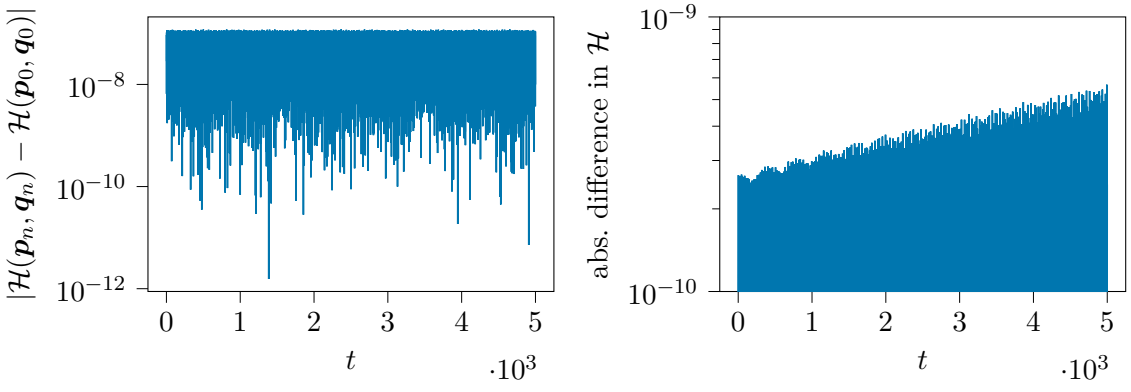


Fig. 7.2. Four body-problem. Work-precision diagram for the methods `H-BACAB` ($M = 12$), `H-BADAB` ($M = 12$), and `HOMF4` ($M = 15$). All methods utilize the reweighting (6.7).



(a) Absolute error in the Hamiltonian for the `H-BADAB` method with $h = 1/10$ and reweighted $M = 12$.

(b) Difference in the Hamiltonian for the `H-BACAB` and the `H-BADAB` method with $h = 1/10$ and reweighted $M = 12$.

Fig. 7.3. Four body-problem. Energy behavior of the hierarchical Hessian-free force-gradient integrator `H-BADAB`.

7.4. Summary

This chapter combined the frameworks of (Hessian-free) force-gradient integrators and (multirate) hierarchical splitting methods, resulting in a class of hierarchical (Hessian-free) force-gradient integrators that incorporates multiple time stepping. We have presented a natural choice of a splitting tree for separable Hamiltonian systems of the form (7.1), and demonstrated that the (Hessian-free) force-gradient approach is applicable to the subsystems arising in the inner nodes. The resulting class of hierarchical (Hessian-free) force-gradient integrators can be regarded as a generalization of nested integration techniques [123, 138] in two ways: first, the comprehensive analysis establishes the validity of (Hessian-free) force-gradient integrators within these multiple time stepping techniques. Second, the reweighting of the multirate factors significantly contributes to reducing computational overhead. As highlighted by the numerical results, these two improvements contribute to the derivation of more efficient multiple time stepping techniques for Hamiltonian systems of the form (7.1), and in particular the Hessian-free variants yield the most efficient computational process with the trade-off that the Hessian-free variants are not symplectic and may exhibit a linear energy drift over exponentially long time intervals.

This closes the first part of this thesis. With the combination of Hessian-free force-gradient integrators and multirate hierarchical splitting methods, we arrive at a class of time-reversible and volume-preserving multirate methods for separable Hamiltonian systems of the form (7.1), thereby unifying the contributions of the preceding chapters into a single coherent framework. The second part of this thesis is devoted to the application of (hierarchical) Hessian-free force-gradient integrators to hybrid Monte Carlo (HMC) simulations in lattice quantum chromodynamics (QCD) – a practically important application in theoretical physics where Hamiltonian systems of the form (7.1) arise naturally in a large-scale high-performance computing setting that demands efficient, time-reversible, and volume-preserving numerical integration schemes.

Part II.

HMC Simulations in Lattice QCD

Chapter 8.

The Hybrid Monte Carlo Algorithm

In this chapter, we provide an introduction to the Hybrid Monte Carlo (HMC) algorithm [38], which has become one of the most widely used Markov Chain Monte Carlo (MCMC) methods in computational physics and Bayesian statistics. The HMC algorithm constitutes a natural application of the geometric numerical integration schemes discussed in the first part of this thesis, and will form the foundation for the lattice QCD simulations discussed in the subsequent chapters.

Consider the measurable space $(\mathbb{R}^n, \mathcal{B}(\mathbb{R}^n))$ where $\mathcal{B}(\mathbb{R}^n)$ denotes the Borel σ-algebra on $\mathbb{R}^n$. Given a probability distribution $\mu\colon \mathcal{B}(\mathbb{R}^n) \to [0,1]$, the *target distribution*, and a function $\mathcal{O}\colon \mathbb{R}^n \to \mathbb{R}$, $\boldsymbol{x} \mapsto \mathcal{O}(\boldsymbol{x})$, we want to estimate the integral

$$\mu(\mathcal{O}) = \int_{\mathbb{R}^n} \mathcal{O}(\boldsymbol{x})\, \mu(\mathrm{d}\boldsymbol{x}). \tag{8.1}$$

In general, we cannot compute (8.1) analytically. When the dimension n is large, numerical computation of (8.1) via any deterministic quadrature is also infeasible [68].

8.1. Monte Carlo Methods

A *Monte Carlo method* estimates (8.1) by generating n_{sample} independent and identically distributed (i.i.d.) samples $\boldsymbol{X}_1, \dots, \boldsymbol{X}_{n_{\text{sample}}}$ from μ, and computing

$$\bar{\mathcal{O}}_{n_{\text{sample}}} = \frac{1}{n_{\text{sample}}} \sum_{k=1}^{n_{\text{sample}}} \mathcal{O}(\boldsymbol{X}_k). \tag{8.2}$$

For $\mu(\mathcal{O}) < \infty$ we have, according to the *law of large numbers*,

$$\bar{\mathcal{O}}_{n_{\text{sample}}} \overset{\text{a.s.}}{\to} \mu(\mathcal{O}) \quad \text{as } n_{\text{sample}} \to \infty. \tag{8.3}$$

If the variance

$$\sigma_0(\mathcal{O})^2 = \int_{\mathbb{R}^n} (\mathcal{O}(\boldsymbol{x}) - \mu(\mathcal{O}))^2\, \mu(\mathrm{d}\boldsymbol{x}) \tag{8.4}$$

of the random variable $\mathcal{O}(\boldsymbol{X})$, $\boldsymbol{X} \sim \mu$, is also finite, the *central limit theorem* states that, as $n_{\text{sample}} \to \infty$, the random variables $\sqrt{n_{\text{sample}}}(\bar{\mathcal{O}}_{n_{\text{sample}}} - \mu(\mathcal{O}))$ converge in distribution to a Gaussian distribution with mean zero and variance (8.4),

$$\sqrt{n_{\text{sample}}}(\bar{\mathcal{O}}_{n_{\text{sample}}} - \mu(\mathcal{O})) \overset{d}{\to} \mathcal{N}(0, \sigma_0(\mathcal{O})^2) \quad \text{as } n_{\text{sample}} \to \infty. \tag{8.5}$$

Consequently, the standard deviation of the error $\bar{\mathcal{O}}_{n_{\text{sample}}} - \mu(\mathcal{O})$, which is often called the *Monte Carlo error*, decreases like $1/\sqrt{n_{\text{sample}}}$.

In most practical situations, we cannot directly generate i.i.d. samples $\boldsymbol{X}_k \sim \mu$. Instead, we will apply Markov chain Monte Carlo methods.

8.2. Markov Chain Monte Carlo Methods

Markov Chain Monte Carlo (MCMC) methods have become an important tool in statistical computing. We will provide a concise introduction to MCMC methods sufficient to introduce the HMC algorithm. For a comprehensive introduction to the MCMC methodology, we refer to [21] and the references therein.

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space with non-empty set Ω, σ-algebra $\mathcal{F}$, and a probability measure $\mathbb{P}$ on which all random variables in this chapter are defined.

Definition 8.1 (Markov chain)**.** A *Markov chain* with state space $\mathbb{R}^n$ is a sequence $\{\boldsymbol{X}_k\}_{k\in\mathbb{N}}$ of random variables satisfying the *Markov property*

$$\mathbb{P}(\boldsymbol{X}_{k+1} \in A \,|\, \boldsymbol{X}_1, \dots, \boldsymbol{X}_k) = \mathbb{P}(\boldsymbol{X}_{k+1} \in A \,|\, \boldsymbol{X}_k)$$

for all $A \in \mathcal{B}(\mathbb{R}^n)$.

Definition 8.2 (Markov transition kernel)**.** A Markov transition kernel is a mapping $\Pi\colon \mathbb{R}^n \times \mathcal{B}(\mathbb{R}^n) \to [0,1]$, such that

1. for every fixed $\boldsymbol{x} \in \mathbb{R}^n$, the function $A \mapsto \Pi_{\boldsymbol{x}}(A)$ is a probability measure on $(\mathbb{R}^n, \mathcal{B}(\mathbb{R}^n))$;
2. for every fixed $A \in \mathcal{B}(\mathbb{R}^n)$, the function $\boldsymbol{x} \mapsto \Pi_{\boldsymbol{x}}(A)$ is measurable.

For our purposes, we will restrict ourselves to Markov chains with *stationary transition probabilities*, i.e., Markov chains where the *transition probabilities*

$$\Pi_{\boldsymbol{x}}(A) \coloneqq \mathbb{P}(\boldsymbol{X}_{k+1} \in A \,|\, \boldsymbol{X}_k = \boldsymbol{x}) \tag{8.6}$$

for transitioning from state $\boldsymbol{x} \in \mathbb{R}^n$ into the set $A \in \mathcal{B}(\mathbb{R}^n)$ do not depend on k.

Definition 8.3 (Stationary distribution)**.** A probability distribution μ is a *stationary distribution* of the Markov chain with transition probabilities $\Pi_{\boldsymbol{x}}$, if

$$\int_{\mathbb{R}^n} \Pi_{\boldsymbol{x}}(A)\, \mu(\mathrm{d}\boldsymbol{x}) = \int_A \mu(\mathrm{d}\boldsymbol{x}) \tag{8.7}$$

for all $A \in \mathcal{B}(\mathbb{R}^n)$.

In practice, the inputs of an MCMC method include a sample of the initial state $\boldsymbol{X}_1$, and a possibly non-normalized density function $\nu\colon \mathbb{R}^n \to [0,\infty)$ of the target distribution μ, i.e., the probability density function is $\frac{1}{Z}\nu(\boldsymbol{x})$, where $Z = \int_{\mathbb{R}^n} \nu(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}$. Note that an MCMC method does not require the value of Z.

MCMC methods generate a Markov chain $\{\boldsymbol{X}_k\}_{k\in\mathbb{N}}$ with target distribution μ as the stationary distribution and estimate $\mu(\mathcal{O})$ via the estimator (8.2). We can formulate MCMC analogues of the law of large numbers (8.3) and the central limit theorem (8.5), which we would like to satisfy.

MCMC analogue of the law of large numbers. For each $\mathcal{O}\colon \mathbb{R}^n \to \mathbb{R}$ such that $\mu(\mathcal{O})$ is finite,

$$\bar{\mathcal{O}}_{n_{\text{sample}}} \overset{\text{a.s.}}{\to} \mu(\mathcal{O}) \quad \text{as } n_{\text{sample}} \to \infty. \tag{8.8}$$

MCMC analogue of the central limit theorem. If, in addition, (8.4) is finite,

$$\sqrt{n_{\text{sample}}}(\bar{\mathcal{O}}_{n_{\text{sample}}} - \mu(\mathcal{O})) \overset{d}{\to} \mathcal{N}(0, \sigma(\mathcal{O})^2) \quad \text{as } n_{\text{sample}} \to \infty, \tag{8.9}$$

for some constant $\sigma(\mathcal{O})$ called the *asymptotic variance* of the estimator.

Supposing that $\{\boldsymbol{X}_k\}_{k\in\mathbb{N}}$ is a stationary Markov chain with $\boldsymbol{X}_1 \sim \mu$, the asymptotic variance satisfies [21]

$$\sigma(\mathcal{O})^2 = \sigma_0(\mathcal{O})^2 + 2\sum_{k=1}^{\infty} \operatorname{Cov}(\mathcal{O}(\boldsymbol{X}_i), \mathcal{O}(\boldsymbol{X}_{i+k})). \tag{8.10}$$

If the samples $\boldsymbol{X}_k$ were independent, the covariances would vanish and we would recover the variance (8.4). In most practical situations, the samples in the Markov chain are not mutually independent, so that $\sigma(\mathcal{O})^2 > \sigma_0(\mathcal{O})^2$. To assess the impact of this requires studying the *autocorrelation times*. Informally, the autocorrelation time quantifies how much the correlation between successive samples reduces the effective sample size. By introducing $\Gamma_{\mathcal{O}}(k) = \operatorname{Cov}(\mathcal{O}(\boldsymbol{X}_i), \mathcal{O}(\boldsymbol{X}_{i+k}))$, we can rewrite the asymptotic variance as

$$\sigma(\mathcal{O})^2 = \Gamma_{\mathcal{O}}(0) + 2\sum_{k=1}^{\infty} \Gamma_{\mathcal{O}}(k).$$

We can then define the *integrated autocorrelation time*

$$\tau_{\mathrm{int}}(\mathcal{O}) = \frac{1}{2} + \sum_{k=1}^{\infty} \frac{\Gamma_{\mathcal{O}}(k)}{\Gamma_{\mathcal{O}}(0)}. \tag{8.11}$$

The integrated autocorrelation time characterizes the number of successive states of the Markov chain required to produce two effectively independent samples. Consequently, out of n_{sample} total samples, the effective number of independent samples is approximately $n_{\mathrm{eff}} = n_{\mathrm{sample}}/2\tau_{\mathrm{int}}(\mathcal{O})$. The goal is to minimize the computational cost required to generate an effectively independent sample, i.e., we seek to minimize

$$2\tau_{\mathrm{int}}(\mathcal{O}) \times \text{cost per sample}. \tag{8.12}$$

8.3. The Hybrid Monte Carlo Method

If $\mathcal{V}\colon \mathbb{R}^d \to \mathbb{R}$, $\boldsymbol{q} \mapsto \mathcal{V}(\boldsymbol{q})$, denotes the negative logarithm of the probability density function $\nu_{\boldsymbol{q}}$ of the target distribution $\mu_{\boldsymbol{q}}$, then

$$\mu_{\boldsymbol{q}}(\mathrm{d}\boldsymbol{q}) = \nu_{\boldsymbol{q}}(\boldsymbol{q})\,\mathrm{d}\boldsymbol{q}, \tag{8.13a}$$

with probability density

$$\nu_{\boldsymbol{q}}(\boldsymbol{q}) = \frac{1}{Z_{\boldsymbol{q}}} \exp(-\mathcal{V}(\boldsymbol{q})), \tag{8.13b}$$

and normalizing constant

$$Z_{\boldsymbol{q}} = \int_{\mathbb{R}^d} \exp(-\mathcal{V}(\boldsymbol{q}))\,\mathrm{d}\boldsymbol{q}. \tag{8.13c}$$

We regard $\mathcal{V}$ as the potential energy of a Hamiltonian system with coordinates $\boldsymbol{q}$. To generate new samples, we follow the idea of the hybrid molecular dynamics/Langevin algorithm [37, 36]. We introduce a fictitious computer time t and Hamiltonian dynamics to specify the development of $\boldsymbol{q}(t)$ in this time. To this end, we introduce a set of fictitious conjugate momenta $\boldsymbol{p}(t) \in \mathbb{R}^d$ and a quadratic kinetic energy $\mathcal{T}(\boldsymbol{p}) = \frac{1}{2}\boldsymbol{p}^\top \boldsymbol{M}^{-1}\boldsymbol{p}$ with a symmetric and positive definite mass matrix $\boldsymbol{M}$. This leads to a separable Hamiltonian system of the form

$$\mathcal{H}(\boldsymbol{p},\boldsymbol{q}) = \mathcal{T}(\boldsymbol{p}) + \mathcal{V}(\boldsymbol{q}) = \frac{1}{2}\boldsymbol{p}^\top \boldsymbol{M}^{-1}\boldsymbol{p} + \mathcal{V}(\boldsymbol{q}),$$

which is precisely the Hamiltonian (3.32) considered in the first part of this thesis.

Considering the probability distribution μ in $\mathbb{R}^{2d}$ ($n = 2d$) and the state $\boldsymbol{x} = (\boldsymbol{p}, \boldsymbol{q})$, we have

$$\mu(\mathrm{d}\boldsymbol{p}\,\mathrm{d}\boldsymbol{q}) = \nu(\boldsymbol{p}, \boldsymbol{q})\,\mathrm{d}\boldsymbol{p}\,\mathrm{d}\boldsymbol{q}, \tag{8.14a}$$

with the joint density

$$\begin{aligned}\nu(\boldsymbol{p}, \boldsymbol{q}) &= \frac{1}{Z}\exp(-\mathcal{T}(\boldsymbol{p}))\exp(-\mathcal{V}(\boldsymbol{q})) \\ &= \frac{1}{\sqrt{(2\pi)^d \cdot \det(\boldsymbol{M})}}\exp(-\mathcal{T}(\boldsymbol{p}))\frac{1}{Z_{\boldsymbol{q}}}\exp(-\mathcal{V}(\boldsymbol{q})),\end{aligned} \tag{8.14b}$$

normalizing constant

$$Z = \int_{\mathbb{R}^{2d}} \exp(-\mathcal{T}(\boldsymbol{p}))\exp(-\mathcal{V}(\boldsymbol{q}))\,\mathrm{d}\boldsymbol{p}\,\mathrm{d}\boldsymbol{q}, \tag{8.14c}$$

and $\mathrm{d}\boldsymbol{p}\,\mathrm{d}\boldsymbol{q}$ denoting the ordinary Lebesgue measure on $\mathbb{R}^{2d}$.

The $\boldsymbol{q}$-marginal (8.13a) of (8.14b) is the target distribution. The $\boldsymbol{p}$-marginal

$$\mu_{\boldsymbol{p}}(\mathrm{d}\boldsymbol{p}) = \nu_{\boldsymbol{p}}(\boldsymbol{p})\,\mathrm{d}\boldsymbol{p} \tag{8.15a}$$

with

$$\nu_{\boldsymbol{p}}(\boldsymbol{p}) = \frac{1}{Z_{\boldsymbol{p}}}\exp(-\mathcal{T}(\boldsymbol{p})) \tag{8.15b}$$

and

$$Z_{\boldsymbol{p}} = \int_{\mathbb{R}^d} \exp(-\mathcal{T}(\boldsymbol{p}))\,\mathrm{d}\boldsymbol{p} = \sqrt{(2\pi)^d \cdot \det(\boldsymbol{M})} \tag{8.15c}$$

is Gaussian with zero mean and covariance matrix $\boldsymbol{M}$.

Example 8.4. We consider an example from [21] with a bivariate Gaussian distribution $\boldsymbol{q} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{N})$ with

$$\boldsymbol{N} = \begin{pmatrix} 1 & 0.95 \\ 0.95 & 1 \end{pmatrix}.$$

By introducing corresponding momenta $\boldsymbol{p} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, we obtain the Hamiltonian

$$\mathcal{H}(\boldsymbol{p}, \boldsymbol{q}) = \frac{\boldsymbol{p}^\top \boldsymbol{p}}{2} + \frac{1}{2}\boldsymbol{q}^\top \boldsymbol{N}^{-1}\boldsymbol{q}.$$

For this Hamiltonian, the joint density (8.14b) reads

$$\nu(\boldsymbol{p}, \boldsymbol{q}) = \frac{1}{2\pi}\exp\left(-\frac{\boldsymbol{p}^\top \boldsymbol{p}}{2}\right)\frac{10}{\sqrt{39}\pi}\exp\left(-\frac{1}{2}\boldsymbol{q}^\top \boldsymbol{N}^{-1}\boldsymbol{q}\right).$$

The Hamiltonian system has two core features we will exploit. First, samples from the $\boldsymbol{p}$-marginal are easily available. Second, the Hamiltonian equations of motion (2.7) preserve the distribution μ, as the following theorem states.

Theorem 8.5. *The probability measure μ in $\mathbb{R}^{2d}$ with density* (8.14b) *is preserved by the exact flow $\boldsymbol{\varphi}_t$ of the Hamiltonian system* (3.32)*, i.e.,*

$$\mu(\boldsymbol{\varphi}_t(D)) = \mu(D)$$

for each domain $D \subset \mathbb{R}^{2d}$ and $t \in \mathbb{R}$.

Proof. The theorem is a direct consequence of the volume preservation (2.12) and the energy conservation (2.9) of the Hamiltonian flow, cf. [20, Theorem 2.7]. □

In practice, however, we will not be able to compute the Hamiltonian flow exactly. Instead, we will replace $\boldsymbol{\varphi}_t$ with a numerical flow $\boldsymbol{\Psi}_t$ such that the probability measure μ is still preserved. Unfortunately, numerical methods such as the splitting methods from Chapter 3 cannot preserve both the phase space volume and the Hamiltonian simultaneously, i.e., Theorem 8.5 does not hold for numerical flows $\boldsymbol{\Psi}_t$ and the distribution (8.14a) is not preserved exactly. Dropping the requirement of energy conservation, we need to correct the bias introduced by the time discretization error.

Suppose the Hamiltonian system (3.32) is solved numerically over the time interval $t \in [0, \tau]$ of *trajectory length* $\tau > 0$ with initial values $(\boldsymbol{\pi}_0, \boldsymbol{q}_0)$. Here, the momenta $\boldsymbol{\pi}_0$ are obtained by sampling from the Gaussian distribution. The coordinates $\boldsymbol{q}_0$ correspond to the current state $\boldsymbol{X}_k = (\boldsymbol{p}_0, \boldsymbol{q}_0)$ of the Markov chain. The numerical solution is obtained by computing $n_{\text{steps}} \in \mathbb{N}$ steps of step size $h = \tau / n_{\text{steps}}$ with a numerical method $\boldsymbol{\Phi}_h$, i.e., $\boldsymbol{\Psi}_\tau = (\boldsymbol{\Phi}_h)^{n_{\text{steps}}}$. We refer to this numerical time integration part of the algorithm as the *molecular dynamics step*. The result $\boldsymbol{\Psi}_\tau(\boldsymbol{\pi}_0, \boldsymbol{q}_0)$ is called the *candidate sample*.

To correct for the bias introduced by the time discretization, we incorporate a *Metropolis–Hastings acceptance step* [95, 66]. In its generalized form [42], a candidate sample is accepted with probability $\min\{1, r\}$, where the ratio r depends on the proposal. Here, the acceptance probability takes the form

$$\alpha(\boldsymbol{\pi}_0, \boldsymbol{q}_0) = \min\left\{1, \frac{\nu(\boldsymbol{\Psi}_\tau(\boldsymbol{\pi}_0, \boldsymbol{q}_0))}{\nu(\boldsymbol{\pi}_0, \boldsymbol{q}_0)}\right\} = \min\left\{1, \exp(-\Delta\mathcal{H}) \left|\det \frac{\partial \boldsymbol{\Psi}_\tau(\boldsymbol{\pi}_0, \boldsymbol{q}_0)}{\partial(\boldsymbol{\pi}_0, \boldsymbol{q}_0)}\right|\right\}, \tag{8.16}$$

where

$$\Delta\mathcal{H} = \mathcal{H}(\boldsymbol{\Psi}_\tau(\boldsymbol{\pi}_0, \boldsymbol{q}_0)) - \mathcal{H}(\boldsymbol{\pi}_0, \boldsymbol{q}_0). \tag{8.17}$$

In case of acceptance, we add the candidate sample to the Markov chain, $\boldsymbol{X}_{k+1} = \boldsymbol{\Psi}_\tau(\boldsymbol{\pi}_0, \boldsymbol{q}_0)$. Otherwise, the sample is rejected and we set $\boldsymbol{X}_{k+1} = (-\boldsymbol{\pi}_0, \boldsymbol{q}_0)$. Combining the molecular dynamics step with the acceptance step, we obtain the *Hybrid Monte Carlo (HMC) algorithm* [38] that is summarized in Algorithm 8.1.

Algorithm 8.1: HMC algorithm.

Given the current state of the Markov chain $\boldsymbol{X}_k = (\boldsymbol{p}_0, \boldsymbol{q}_0) \in \mathbb{R}^{2d}$, the algorithm outputs the next state $\boldsymbol{X}_{k+1} = (\boldsymbol{p}_1, \boldsymbol{q}_1) \in \mathbb{R}^{2d}$ as follows:
Generate $\boldsymbol{\pi}_0 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{M})$
Perform the molecular dynamics step and compute $\boldsymbol{\Psi}_\tau(\boldsymbol{\pi}_0, \boldsymbol{q}_0)$
$\Delta\mathcal{H} = \mathcal{H}(\boldsymbol{\Psi}_\tau(\boldsymbol{\pi}_0, \boldsymbol{q}_0)) - \mathcal{H}(\boldsymbol{\pi}_0, \boldsymbol{q}_0)$
Draw $u \sim \mathcal{U}(0, 1)$
if $\min\left\{1, \exp(-\Delta\mathcal{H}) \left|\det \frac{\partial \boldsymbol{\Psi}_\tau(\boldsymbol{\pi}_0, \boldsymbol{q}_0)}{\partial(\boldsymbol{\pi}_0, \boldsymbol{q}_0)}\right|\right\} \geq u$ **then**
  Output $(\boldsymbol{p}_1, \boldsymbol{q}_1) = \boldsymbol{\Psi}_\tau(\boldsymbol{\pi}_0, \boldsymbol{q}_0)$
else
  Output $(\boldsymbol{p}_1, \boldsymbol{q}_1) = (-\boldsymbol{\pi}_0, \boldsymbol{q}_0)$

Due to its connection to the Hamiltonian equations of motion, the HMC algorithm is often called *Hamiltonian Monte Carlo*, cf. [20].

Theorem 8.6. *Consider the Markov chain $\{\boldsymbol{X}_k = (\boldsymbol{p}_k, \boldsymbol{q}_k)\}_{k \in \mathbb{N}}$ generated by iterating Algorithm 8.1. The probability distribution μ with density* (8.14b) *is a stationary distribution of this Markov chain, provided that the numerical integration scheme $\boldsymbol{\Psi}_\tau$ is time-reversible.*

Proof. We follow the proof of [20, Theorem 5.7] and decompose an iteration of the HMC algorithm into two parts. The first part generates momenta $\boldsymbol{\pi}_0 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{M})$, resulting in

a mapping $(\boldsymbol{p}_0, \boldsymbol{q}_0) \mapsto (\boldsymbol{\pi}_0, \boldsymbol{q}_0)$. Since the new momentum $\boldsymbol{\pi}_0$ is drawn exactly from the correct marginal distribution, the mapping preserves the distribution. It remains to show that also the second part of the HMC algorithm, involving the molecular dynamics step $(\boldsymbol{\pi}_0, \boldsymbol{q}_0) \mapsto (\boldsymbol{p}', \boldsymbol{q}') \coloneqq \boldsymbol{\Psi}_\tau(\boldsymbol{\pi}_0, \boldsymbol{q}_0)$ and the acceptance step $(\boldsymbol{p}', \boldsymbol{q}') \mapsto (\boldsymbol{p}_1, \boldsymbol{q}_1)$, preserves the distribution. For this update, the transition probabilities of the corresponding Markov transition kernel are given by

$$\Pi_{\boldsymbol{x}}(A) = \alpha(\boldsymbol{x})\mathbf{1}_A(\boldsymbol{\Psi}_\tau(\boldsymbol{x})) + (1 - \alpha(\boldsymbol{x}))\mathbf{1}_A(\boldsymbol{\rho}(\boldsymbol{x})),$$

where $\boldsymbol{x} = (\boldsymbol{\pi}_0, \boldsymbol{q}_0)$,

$$\mathbf{1}_A(\boldsymbol{x}) = \begin{cases} 1, & \boldsymbol{x} \in A, \\ 0, & \boldsymbol{x} \notin A, \end{cases}$$

denotes the indicator function, and $\boldsymbol{\rho}(\boldsymbol{\pi}_0, \boldsymbol{q}_0) = (-\boldsymbol{\pi}_0, \boldsymbol{q}_0)$ is the involution used to define the time-reversibility (2.15). We obtain

$$\int_{\mathbb{R}^n} \Pi_{\boldsymbol{x}}(A)\nu(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = \int_{\mathbb{R}^n} \nu(\boldsymbol{x})\mathbf{1}_A(\boldsymbol{\rho}(\boldsymbol{x}))\,\mathrm{d}\boldsymbol{x} + \int_{\mathbb{R}^n} \alpha(\boldsymbol{x})\nu(\boldsymbol{x})\mathbf{1}_A(\boldsymbol{\Psi}_\tau(\boldsymbol{x}))\,\mathrm{d}\boldsymbol{x} - \int_{\mathbb{R}^n} \alpha(\boldsymbol{x})\nu(\boldsymbol{x})\mathbf{1}_A(\boldsymbol{\rho}(\boldsymbol{x}))\,\mathrm{d}\boldsymbol{x}.$$

Applying the substitution $\boldsymbol{x} \mapsto \boldsymbol{\rho}(\boldsymbol{x})$ to the first integral, we obtain

$$\int_{\mathbb{R}^n} \Pi_{\boldsymbol{x}}(A)\nu(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = \int_A \nu(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} + \int_{\mathbb{R}^n} \alpha(\boldsymbol{x})\nu(\boldsymbol{x})\mathbf{1}_A(\boldsymbol{\Psi}_\tau(\boldsymbol{x}))\,\mathrm{d}\boldsymbol{x} - \int_{\mathbb{R}^n} \alpha(\boldsymbol{x})\nu(\boldsymbol{x})\mathbf{1}_A(\boldsymbol{\rho}(\boldsymbol{x}))\,\mathrm{d}\boldsymbol{x},$$

which follows from $\nu \circ \boldsymbol{\rho} = \nu$ and $\boldsymbol{\rho} \circ \boldsymbol{\rho} = \mathrm{Id}$. Applying the substitution $\boldsymbol{x} \mapsto \boldsymbol{\rho}(\boldsymbol{\Psi}_\tau(\boldsymbol{x}))$ to the third integral, and using that $\boldsymbol{\rho} \circ \boldsymbol{\rho} = \mathrm{Id}$, $\nu \circ \boldsymbol{\rho} = \nu$, and $|\det \boldsymbol{\rho}'(\boldsymbol{x})| = 1$, yields

$$\int_{\mathbb{R}^n} \Pi_{\boldsymbol{x}}(A)\nu(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \int_A \nu(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} + \int_{\mathbb{R}^n} \alpha(\boldsymbol{x})\nu(\boldsymbol{x})\mathbf{1}_A(\boldsymbol{\Psi}_\tau(\boldsymbol{x}))\,\mathrm{d}\boldsymbol{x} - \int_{\mathbb{R}^n} \alpha(\boldsymbol{\rho}(\boldsymbol{\Psi}_\tau(\boldsymbol{x})))\nu(\boldsymbol{\Psi}_\tau(\boldsymbol{x}))\mathbf{1}_A(\boldsymbol{\Psi}_\tau(\boldsymbol{x}))\left|\det \frac{\partial \boldsymbol{\Psi}_\tau(\boldsymbol{x})}{\partial \boldsymbol{x}}\right| \mathrm{d}\boldsymbol{x}.$$

For the acceptance probability (8.16), we obtain

$$\alpha(\boldsymbol{\rho}(\boldsymbol{\Psi}_\tau(\boldsymbol{x}))) = \min\left\{1, \frac{\nu(\boldsymbol{\Psi}_\tau(\boldsymbol{\rho}(\boldsymbol{\Psi}_\tau(\boldsymbol{x}))))}{\nu(\boldsymbol{\rho}(\boldsymbol{\Psi}_\tau(\boldsymbol{x})))}\left|\det \frac{\partial \boldsymbol{\Psi}_\tau}{\partial \boldsymbol{x}}(\boldsymbol{\rho}(\boldsymbol{\Psi}_\tau(\boldsymbol{x})))\right|\right\}.$$

According to the time-reversibility (2.15) of $\boldsymbol{\Psi}_\tau$, it holds $\boldsymbol{\Psi}_\tau(\boldsymbol{\rho}(\boldsymbol{\Psi}_\tau(\boldsymbol{x}))) = \boldsymbol{\rho}(\boldsymbol{x})$. Thus, the numerator simplifies to $\nu(\boldsymbol{\rho}(\boldsymbol{x})) = \nu(\boldsymbol{x})$, where we utilized that $\nu \circ \boldsymbol{\rho} = \nu$. Similarly, using $\nu \circ \boldsymbol{\rho} = \nu$, the denominator simplifies to $\nu(\boldsymbol{\rho}(\boldsymbol{\Psi}_\tau(\boldsymbol{x}))) = \nu(\boldsymbol{\Psi}_\tau(\boldsymbol{x}))$. Moreover, the Jacobian can be simplified. Differentiating the time-reversibility condition (2.15) yields

$$\frac{\partial \boldsymbol{\rho}}{\partial \boldsymbol{x}}(\boldsymbol{\Psi}_\tau(\boldsymbol{\rho}(\boldsymbol{\Psi}_\tau(\boldsymbol{x})))) \cdot \frac{\partial \boldsymbol{\Psi}_\tau}{\partial \boldsymbol{x}}(\boldsymbol{\rho}(\boldsymbol{\Psi}_\tau(\boldsymbol{x})))) \cdot \frac{\partial \boldsymbol{\rho}}{\partial \boldsymbol{x}}(\boldsymbol{\Psi}_\tau(\boldsymbol{x})) \cdot \frac{\partial \boldsymbol{\Psi}_\tau}{\partial \boldsymbol{x}}(\boldsymbol{x}) = \boldsymbol{I}.$$

Taking the absolute determinant of both sides and using $|\det \boldsymbol{\rho}'(\boldsymbol{x})| = 1$ yields

$$\left|\det \frac{\partial \boldsymbol{\Psi}_\tau}{\partial \boldsymbol{x}}(\boldsymbol{\rho}(\boldsymbol{\Psi}_\tau(\boldsymbol{x})))\right| = \left|\det \frac{\partial \boldsymbol{\Psi}_\tau}{\partial \boldsymbol{x}}(\boldsymbol{x})\right|^{-1}.$$

Consequently, the acceptance probability simplifies to

$$\alpha(\boldsymbol{\rho}(\boldsymbol{\Psi}_\tau(\boldsymbol{x}))) = \min\left\{1, \frac{\nu(\boldsymbol{x})}{\nu(\boldsymbol{\Psi}_\tau(\boldsymbol{x}))}\left|\det \frac{\partial \boldsymbol{\Psi}_\tau}{\partial \boldsymbol{x}}(\boldsymbol{x})\right|^{-1}\right\}$$

$$= \min\left\{\frac{\nu(\boldsymbol{\Psi}_\tau(\boldsymbol{x}))}{\nu(\boldsymbol{x})}\left|\det\frac{\partial\boldsymbol{\Psi}_\tau}{\partial\boldsymbol{x}}(\boldsymbol{x})\right|, 1\right\}\frac{\nu(\boldsymbol{x})}{\nu(\boldsymbol{\Psi}_\tau(\boldsymbol{x}))}\left|\det\frac{\partial\boldsymbol{\Psi}_\tau}{\partial\boldsymbol{x}}(\boldsymbol{x})\right|^{-1}$$
$$= \alpha(\boldsymbol{x})\frac{\nu(\boldsymbol{x})}{\nu(\boldsymbol{\Psi}_\tau(\boldsymbol{x}))}\left|\det\frac{\partial\boldsymbol{\Psi}_\tau}{\partial\boldsymbol{x}}(\boldsymbol{x})\right|^{-1}.$$

Therefore, the third integral simplifies considerably and we obtain

$$\begin{aligned}\int_{\mathbb{R}^n}\Pi_{\boldsymbol{x}}(A)\nu(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} &= \int_A \nu(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} + \int_{\mathbb{R}^n}\alpha(\boldsymbol{x})\nu(\boldsymbol{x})\mathbf{1}_A(\boldsymbol{\Psi}_\tau(\boldsymbol{x}))\,\mathrm{d}\boldsymbol{x}\\ &\quad - \int_{\mathbb{R}^n}\alpha(\boldsymbol{x})\nu(\boldsymbol{x})\mathbf{1}_A(\boldsymbol{\Psi}_\tau(\boldsymbol{x}))\,\mathrm{d}\boldsymbol{x}\\ &= \int_A \nu(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}.\end{aligned}$$

Consequently, the time-reversibility of the integrator $\boldsymbol{\Psi}_\tau$ suffices to show that the target distribution μ is a stationary distribution of the Markov chain generated by the HMC algorithm. □

A sufficient condition for the time-reversibility of $\boldsymbol{\Psi}_\tau = (\boldsymbol{\Phi}_h)^{n_{\text{steps}}}$ is that the underlying numerical integration scheme $\boldsymbol{\Phi}_h$ is time-reversible. In practice, the computation of the acceptance probability (8.16) may be infeasible because of the need to compute

$$\left|\det\frac{\partial\boldsymbol{\Psi}_\tau(\boldsymbol{\pi}_0,\boldsymbol{q}_0)}{\partial(\boldsymbol{\pi}_0,\boldsymbol{q}_0)}\right|.$$

If the numerical integration scheme $\boldsymbol{\Phi}_h$ (and thus $\boldsymbol{\Psi}_\tau$) is also volume-preserving and satisfies (2.12), then the acceptance probability simplifies to

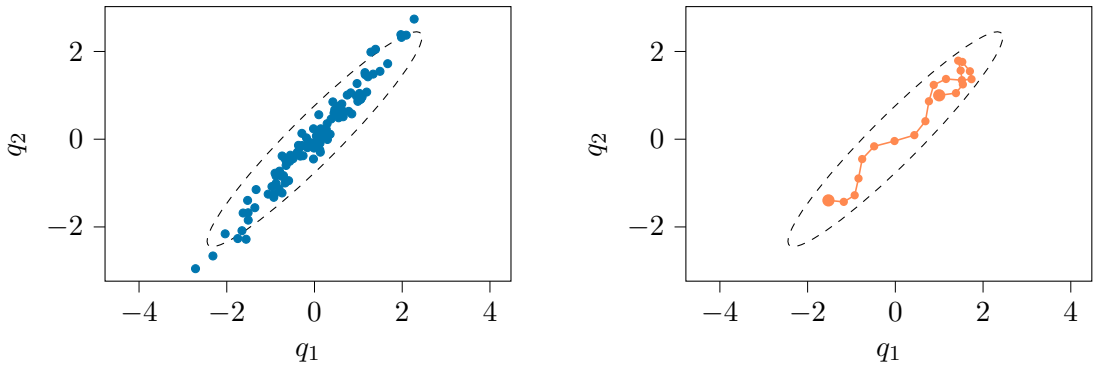
$$\alpha(\boldsymbol{x}) = \min\left\{1, \exp(-\Delta\mathcal{H})\right\}, \tag{8.18}$$

such that it becomes easily computable. Within this thesis, we will assume that $\boldsymbol{\Phi}_h$ (and thus $\boldsymbol{\Psi}_\tau$) is both time-reversible and volume-preserving. For a discussion of variants neglecting the volume preservation, see for example [42].

Example 8.7. We consider the bivariate Gaussian distribution from Example 8.4. Starting from an initial state $\boldsymbol{q} = (1,1)^\top$, we compute 100 trajectories of length $\tau = 5$ utilizing the time-reversible and volume-preserving Hessian-free force-gradient integrator `BADAB` (A.9) with $h = 0.25$. For this step size, the method achieves an acceptance rate of 100%. The samples of the position coordinates are depicted in Fig. 8.1a. Fig. 8.1b shows the position coordinates after each time step of size h for the first trajectory.

Theorem 8.6 shows that the target distribution μ is a stationary distribution of the Markov chain generated by Algorithm 8.1. However, Theorem 8.6 does not guarantee that (8.8) and (8.9) hold. Typically, the HMC algorithm satisfies (8.8) and is *ergodic*[1], such that the Markov chain will asymptotically converge to its (unique) stationary distribution. Establishing ergodicity of the HMC algorithm rigorously is non-trivial and beyond the scope of this thesis; we refer to [80] for a detailed analysis. The following example shows that a poor choice of the trajectory length τ may result in a periodic chain.

[1] Ergodicity means that the chain almost surely visits every set of positive probability. Roughly speaking, the chain is *irreducible* (it can reach any measurable set from any initial state) and *aperiodic* (the chain does not get trapped in cycles).



(a) Position coordinates of the samples of the Markov chain.

(b) The first trajectory including the position coordinates after each time step.

Fig. 8.1. HMC simulations for the bivariate Gaussian distribution from Example 8.4 utilizing the Hessian-free force-gradient integrator `BADAB` (A.9) with trajectory length $\tau = 5$ and step size $h = 0.25$. The simulation started from an initial state $\boldsymbol{q} = (1, 1)^\top$. The ellipses plotted show the 95% confidence ellipses of the bivariate Gaussian distribution.

Example 8.8. Consider the harmonic oscillator (4.23) with $\omega = 1$, corresponding to the Hamiltonian $\mathcal{H}(p, q) = (p^2 + q^2)/2$. The exact flow is periodic with period 2π. For $\tau = \pi$ and supposing that the Hamiltonian equations of motion are solved exactly, the sequence of values generated by the HMC algorithm is $q_0, -q_0, q_0, \ldots$ such that the chain is not aperiodic.

Remark 8.9. The HMC simulation starts from the initial state $\boldsymbol{X}_1$, which may be far from stationarity. Therefore, we will perform a *thermalization* or *burn-in* phase and wait until the Markov chain has converged to its stationary distribution. The samples created during this thermalization phase are discarded and will not be used for the estimator (8.2). In practice, the number of thermalization steps is determined by monitoring observables such as $\mathcal{O}(\boldsymbol{x})$ and checking that their running averages have stabilized to a satisfactory degree.

8.4. Tuning the HMC Algorithm

The HMC algorithm offers several degrees of freedom that can be tuned to optimize the simulation: the trajectory length τ, the mass matrix $\boldsymbol{M}$, the numerical integration scheme $\boldsymbol{\Phi}_h$, and its step size h. As illustrated in Example 8.8, the performance of the HMC algorithm may be very sensitive to changes in τ [87]. To increase the robustness of the algorithm, one approach is to randomly vary τ from trajectory to trajectory [87]. Similarly, the mass matrix $\boldsymbol{M}$ can be adapted to the geometry of the target distribution, for instance by preconditioning with an estimate $\boldsymbol{\Sigma}$ of the covariance matrix for $\boldsymbol{q}$ via $\boldsymbol{M} = \boldsymbol{\Sigma}^{-1}$ [21, Section 5.4.1]. A comprehensive study of these tuning strategies is beyond the scope of this thesis. Since the focus of this thesis is on the numerical integration schemes used inside the molecular dynamics step, we regard τ and $\boldsymbol{M}$ as fixed and focus on the selection of the numerical integration scheme $\boldsymbol{\Phi}_h$ and its step size h.

Self-adjoint splitting methods and (Hessian-free) force-gradient integrators are volume-preserving and time-reversible. Therefore, they are feasible choices for the HMC algorithm. The symplecticity of splitting methods and conventional force-gradient integrators implies that the numerical flow preserves a nearby modified Hamiltonian exactly. Consequently, they show excellent long-time energy behavior. This results in a small difference $\Delta\mathcal{H}$ in the Hamiltonian, which contributes to a larger acceptance probability in (8.18). In contrast, Hessian-free force-gradient integrators are in general not symplectic such that

they do not preserve a nearby modified Hamiltonian exactly. According to the discussion from Section 4.2.3, the energy will have a linear drift of size $\mathcal{O}(\tau h^{\max\{4,p\}})$ with τ the trajectory length. For very long trajectory lengths τ, this energy drift may reduce the acceptance probability. Fortunately, we typically consider rather small trajectory lengths τ. When applying the HMC algorithm for gauge field generation in lattice QCD, trajectory lengths of $\tau \approx 2$ are a common choice. For these comparatively small interval lengths, the energy drift becomes negligible so that no significant differences in the acceptance probability were observed between a conventional force-gradient integrator and its Hessian-free counterpart.

In HMC simulations, we are interested in selecting the integrator $\boldsymbol{\Phi}_h$ and its step size h such that the computational cost (8.12) is minimized. For fixed τ, this reduces to a minimization of the cost per trajectory since the choice of the integrator, for sufficiently small h, does not influence the autocorrelation times. As $h \to 0$, all numerical integration schemes converge to the same Hamiltonian flow such that the next sample does not depend on the choice of the integrator. In particular, we seek to minimize the computational cost required to achieve the optimal acceptance rate [133]

$$\langle P_{\text{acc}} \rangle_{\text{opt}} = \exp(-1/p) \approx \begin{cases} 0.61, & p = 2, \\ 0.78, & p = 4, \\ 0.85, & p = 6, \end{cases} \tag{8.19}$$

with p the convergence order of the numerical integration scheme. Note that there also exist alternative approaches to determining an optimal acceptance probability based on different objective functions, see for example [8]. For normally distributed $\Delta\mathcal{H}$, the acceptance rate corresponds to the variance $\sigma^2(\Delta\mathcal{H})$ via [79]

$$\langle P_{\text{acc}} \rangle = \operatorname{erfc}\left(\sqrt{\sigma^2(\Delta\mathcal{H})/8}\right), \tag{8.20}$$

where erfc denotes the complementary error function. Consequently, for fixed integrator $\boldsymbol{\Phi}_h$, we aim at maximizing the step size h while satisfying

$$\sigma^2(\Delta\mathcal{H}) \leq 8 \cdot \operatorname{erfc}^{-1}(\langle P_{\text{acc}} \rangle_{\text{opt}})^2, \tag{8.21}$$

i.e., we are primarily concerned with the region

$$10^{-2} \leq \sigma^2(\Delta\mathcal{H}) \leq 1, \tag{8.22}$$

which corresponds to acceptance probabilities ranging from 61.7% to 96%.

Splitting methods and (Hessian-free) force-gradient integrators applied to separable Hamiltonian systems (3.32) are only conditionally stable, cf. Section 4.3. If h is too large, the integrator may leave its stability domain. Even within the stability domain, $\Delta\mathcal{H}$ may become too large, leading to acceptance rates far below the optimal value (8.19). In contrast, if the step size h is chosen too small, we obtain values of $\Delta\mathcal{H}$ that result in an acceptance rate $\langle P_{\text{acc}} \rangle$ that is significantly larger than (8.19).

Depending on the potential $\mathcal{V}$ and the mass matrix $\boldsymbol{M}$, a problem-dependent trade-off between accuracy and numerical stability determines the most efficient computational process. The investigations from Section 4.4 detected non-dominated integrator variants that constitute promising choices for achieving the optimal acceptance rate (8.19) at the lowest computational cost, cf. the overview in Tab. 4.1 and Tab. 4.2.

The subsequent chapter discusses the application of the HMC algorithm to gauge field generation in lattice QCD, where the specific structure of the potential gives rise to a natural multirate hierarchy of forces that can be exploited by the hierarchical Hessian-free force-gradient integrators developed in the first part of this thesis.

Chapter 9.

Gauge Field Generation in Lattice QCD

Monte Carlo simulations of lattice quantum chromodynamics (QCD) describe the physics of hadrons from their constituents, i.e., they model the strong interaction between *quarks* (the subatomic constituents of matter), and *gluons* (their interacting counterparts). We do not provide a comprehensive treatment of lattice QCD, but rather focus on the mathematical challenges arising inside the molecular dynamics step of the HMC algorithm when performing gauge field generation in lattice QCD. For a more comprehensive overview on lattice QCD, we refer to [33, 77].

9.1. Quantum Fields on a Lattice

We briefly discuss the construction of lattice quantum fields to obtain the *path integral* of lattice QCD, that is, an integral of the form (8.1). For defining the fields and symmetries of classical continuum QCD on the lattice, we focus on the action formulations proposed by Wilson [143].

A lattice gauge field is a field of gluons defined on a four-dimensional hypercubic grid with three spatial dimensions, one time dimension, and *lattice spacing* a – the *lattice*. In general, the lattice has size $V = T \times L^3$, where T denotes the number of lattice sites in the time direction and L the number in each spatial direction. Moreover, we assume periodic boundary conditions.

Each *lattice site* (a grid point) is labeled by a Cartesian four-vector $\boldsymbol{n} = (\boldsymbol{n}_s, n_t)$ and has four links connecting nearest neighbors in the four positive coordinate directions. These four directions are denoted by unit vectors $\hat{\boldsymbol{\mu}}$, $\mu = 1, 2, 3, 4$. We put a link matrix $\boldsymbol{U}_\mu(\boldsymbol{n})$ on the *link* (the edge) interconnecting the lattice site $\boldsymbol{n}$ with its neighbor in direction μ, $\boldsymbol{n} + a\hat{\boldsymbol{\mu}}$. In QCD, the link matrices are elements of the *special unitary group*

$$\mathrm{SU}(3) \coloneqq \{\boldsymbol{U} \in \mathbb{C}^{3\times 3} : \boldsymbol{U}^\dagger \boldsymbol{U} = \boldsymbol{I},\ \det(\boldsymbol{U}) = 1\}. \tag{9.1}$$

Our convention is that $\boldsymbol{U}_\mu(\boldsymbol{n})$ corresponds to the link from the lattice site $\boldsymbol{n}$ to $\boldsymbol{n} + a\hat{\boldsymbol{\mu}}$, whereas $\boldsymbol{U}_\mu(\boldsymbol{n})^{-1} = \boldsymbol{U}_\mu(\boldsymbol{n})^\dagger$ corresponds to the opposite direction, see Fig. 9.1.



Fig. 9.1. Link matrix $\boldsymbol{U}_\mu(\boldsymbol{n})$ connecting lattice site $\boldsymbol{n}$ to $\boldsymbol{n} + a\hat{\boldsymbol{\mu}}$ and $\boldsymbol{U}_\mu(\boldsymbol{n})^\dagger$ corresponding to the opposite direction.

Given a lattice of volume $V = T \times L^3$, we have a lattice gauge field consisting of $4V$ link matrices. We call the set of all links arranged on the lattice a *configuration* and denote it by $[\boldsymbol{U}]$. In present-day research, typical configurations consist of $\mathcal{O}(10^8)$ links.

In lattice QCD, we are interested in computing expectation values of a functional $\mathcal{O}([\boldsymbol{U}])$, which is called an *observable* and is defined via the path integral. The path integral involves a high-dimensional integration over the links $\boldsymbol{U}_\mu(\boldsymbol{n})$ at each lattice site $\boldsymbol{n}$. In particular, since the Lie group SU(3) is an eight-dimensional manifold, the dimension

is $8 \cdot 4V = 32V$. The *curse of dimensionality* [68] makes any deterministic quadrature completely infeasible, motivating the use of the HMC algorithm from Chapter 8.

As a first step, we will consider the *Wilson gauge action* while neglecting fermions, i.e., we start with a pure gauge theory that describes only gluons and excludes quarks.

9.1.1. The Wilson Gauge Action

The central building block of the Wilson gauge action is the *plaquette*. The plaquette is the smallest closed loop of links in the (μ, ν) plane starting at the lattice site $\boldsymbol{n}$ in direction μ, see Fig. 9.2. The plaquette in the (μ, ν) plane at lattice site $\boldsymbol{n}$ is given by

$$\mathcal{P}_{\mu,\nu}(\boldsymbol{n}) = \boldsymbol{U}_\nu(\boldsymbol{n})^\dagger \boldsymbol{U}_\mu(\boldsymbol{n} + a\hat{\boldsymbol{\nu}})^\dagger \boldsymbol{U}_\nu(\boldsymbol{n} + a\hat{\boldsymbol{\mu}}) \boldsymbol{U}_\mu(\boldsymbol{n}). \tag{9.2}$$

Using the plaquette (9.2), the Wilson gauge action is defined as [143]

$$\mathcal{S}^{\{\mathfrak{G}\}}([\boldsymbol{U}]) = \frac{\beta}{3} \sum_{\boldsymbol{n}} \sum_{\mu<\nu} \mathfrak{Re}\big(\mathrm{tr}(\boldsymbol{I} - \mathcal{P}_{\mu,\nu}(\boldsymbol{n}))\big), \tag{9.3}$$

where the *inverse coupling constant* $\beta \geq 0$ is related to the *bare gauge coupling* g_0 in the continuum via the relation $\beta = 6/g_0^2$ for SU(3).

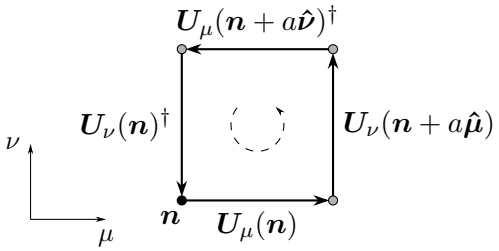


Fig. 9.2. Plaquette in the (μ, ν) plane at lattice site $\boldsymbol{n}$.

Given the Wilson gauge action (9.3), we can now formulate the path integral for the pure gauge theory. The expectation value of a functional $\mathcal{O}([\boldsymbol{U}])$, which is called an *observable*, is defined by

$$\langle \mathcal{O}([\boldsymbol{U}]) \rangle = \frac{1}{Z} \int \mathcal{O}([\boldsymbol{U}]) \exp(-\mathcal{S}^{\{\mathfrak{G}\}}([\boldsymbol{U}]))[\mathrm{d}\boldsymbol{U}], \tag{9.4}$$

with *partition function*

$$Z = \int \exp(-\mathcal{S}^{\{\mathfrak{G}\}}([\boldsymbol{U}]))[\mathrm{d}\boldsymbol{U}], \tag{9.5}$$

and $[\mathrm{d}\boldsymbol{U}] = \prod_{\boldsymbol{n},\mu} \mathrm{d}\boldsymbol{U}_\mu(\boldsymbol{n})$ the product Haar measure on $\mathrm{SU}(3)^{4V}$, where $\mathrm{d}\boldsymbol{U}_\mu(\boldsymbol{n})$ denotes the Haar measure on SU(3). The integration in (9.4) is weighted by the *Boltzmann factor* $\exp(-\mathcal{S}^{\{\mathfrak{G}\}}([\boldsymbol{U}]))$.

9.1.2. Handling Fermions on the Lattice

The gauge links $\boldsymbol{U}_\mu(\boldsymbol{n})$ are situated on the links of the lattice. In contrast, quark and antiquark fields are defined on the lattice sites $\boldsymbol{n}$. $\boldsymbol{\psi}(\boldsymbol{n})$ represents the quark field and $\bar{\boldsymbol{\psi}}(\boldsymbol{n})$ the antiquark field at site $\boldsymbol{n}$. They are independent *Grassmann variables*[1]. The quark field $\boldsymbol{\psi}(\boldsymbol{n})$ at site $\boldsymbol{n}$ consists of 12 components corresponding to the combination of three *color* indices and four *spinor* indices. We denote by $\boldsymbol{\psi}_i^\alpha(\boldsymbol{n})$ the quark field

[1]Two Grassmann variables ψ_i, ψ_j satisfy $\psi_i \psi_j = -\psi_j \psi_i$ so that $\psi_i^2 = 0$.

component at site $\boldsymbol{n}$ with color index $i \in \{1,2,3\}$ and spinor index $\alpha \in \{1,2,3,4\}$. The same convention applies to the antiquark field $\bar{\boldsymbol{\psi}}(\boldsymbol{n})$. We define the *gamma matrices*

$$\begin{aligned} \boldsymbol{\gamma}_1 &= \begin{pmatrix} & & & \mathrm{i} \\ & & \mathrm{i} & \\ & -\mathrm{i} & & \\ -\mathrm{i} & & & \end{pmatrix}, \quad \boldsymbol{\gamma}_2 = \begin{pmatrix} & & & -1 \\ & & 1 & \\ & 1 & & \\ -1 & & & \end{pmatrix}, \\ \boldsymbol{\gamma}_3 &= \begin{pmatrix} & & \mathrm{i} & \\ & & & -\mathrm{i} \\ -\mathrm{i} & & & \\ & \mathrm{i} & & \end{pmatrix}, \quad \boldsymbol{\gamma}_4 = \begin{pmatrix} & & 1 & \\ & & & 1 \\ 1 & & & \\ & 1 & & \end{pmatrix}, \end{aligned} \tag{9.6}$$

which are generators of the Clifford- or Dirac-algebra, i.e., they satisfy

$$\boldsymbol{\gamma}_\mu \boldsymbol{\gamma}_\nu = -\boldsymbol{\gamma}_\nu \boldsymbol{\gamma}_\mu \ (\mu \neq \nu), \quad \text{and} \quad \boldsymbol{\gamma}_\mu^2 = \boldsymbol{I}.$$

Together with the Kronecker delta δ and the *hopping parameter* $\kappa \coloneqq 1/(2am_0 + 8)$ involving the *bare mass parameter* m_0, we have all ingredients to define the *Wilson fermion action*. It is defined as [143]

$$\mathcal{S}^{\{\mathfrak{F}\}}([\boldsymbol{U}, \boldsymbol{\psi}, \bar{\boldsymbol{\psi}}]) = \sum_{\boldsymbol{n},\boldsymbol{m}} \sum_{\alpha,\beta} \sum_{i,j} \bar{\boldsymbol{\psi}}_i^\alpha(\boldsymbol{n}) [\boldsymbol{D}]_{i,j}^{\alpha,\beta} \boldsymbol{\psi}_j^\beta(\boldsymbol{m}), \tag{9.7}$$

where, for each $(\boldsymbol{n}, \boldsymbol{m})$, the *Wilson–Dirac operator* $\boldsymbol{D}$ has components

$$\begin{aligned} [\boldsymbol{D}]_{i,j}^{\alpha,\beta} = \delta(\boldsymbol{n},\boldsymbol{m})\delta_{\alpha,\beta}\delta_{i,j} - \kappa \sum_\mu \Big\{ & \delta(\boldsymbol{n} + a\hat{\boldsymbol{\mu}}, \boldsymbol{m}) [\boldsymbol{I} - \boldsymbol{\gamma}_\mu]^{\alpha,\beta} [\boldsymbol{U}_\mu(\boldsymbol{n})^\dagger]_{i,j} \\ & + \delta(\boldsymbol{n} - a\hat{\boldsymbol{\mu}}, \boldsymbol{m}) [\boldsymbol{I} + \boldsymbol{\gamma}_\mu]^{\alpha,\beta} [\boldsymbol{U}_\mu(\boldsymbol{n} - a\hat{\boldsymbol{\mu}})]_{i,j} \Big\}. \end{aligned} \tag{9.8}$$

The Wilson–Dirac operator satisfies

$$\frac{\boldsymbol{D}(m_0)}{2\kappa(m_0)a} = \frac{\boldsymbol{D}(0)}{2\kappa(0)a} + m_0 \boldsymbol{I},$$

and we write $\boldsymbol{D}(m_0)$ to emphasize the dependence on m_0.

We can now define the partition function of Wilson's lattice QCD in terms of the gauge field $[\boldsymbol{U}]$ and a number N_f of quark fields $[\boldsymbol{\psi}_f]$, $f = 1, \ldots, N_\mathrm{f}$, whose bare mass parameters are $m_{0,f}$.

Remark 9.1. It is common to separate groups of quarks with different masses when defining N_f. For example, $N_\mathrm{f} = 2 + 1$ means that the simulation includes two light quark flavors (with equal mass), and one heavier flavor with a larger mass.

The partition function reads

$$Z = \int \exp\left(-\mathcal{S}^{\{\mathfrak{G}\}}([\boldsymbol{U}]) - \sum_{f=1}^{N_\mathrm{f}} \mathcal{S}^{\{\mathfrak{F},f\}}([\boldsymbol{U}, \boldsymbol{\psi}_f, \bar{\boldsymbol{\psi}}_f]) \right) [\mathrm{d}\boldsymbol{U}] \prod_{f=1}^{N_\mathrm{f}} [\mathrm{d}\bar{\boldsymbol{\psi}}_f][\mathrm{d}\boldsymbol{\psi}_f], \tag{9.9}$$

where the action $\mathcal{S}^{\{\mathfrak{F},f\}}$ corresponds to the fermion action with bare mass parameter $m_0 = m_{0,f}$, and $[\mathrm{d}\boldsymbol{\psi}_f] = \prod_{\boldsymbol{n}} \mathrm{d}\boldsymbol{\psi}_f(\boldsymbol{n})$ with $\mathrm{d}\boldsymbol{\psi}_f(\boldsymbol{n})$ denotes the *Berezin measure* over all 12 Grassmann components at lattice site $\boldsymbol{n}$ (similarly for $[\mathrm{d}\bar{\boldsymbol{\psi}}_f]$). The Berezin measure is defined algebraically by the two rules

$$\int 1 \, \mathrm{d}\psi = 0, \quad \int \psi \, \mathrm{d}\psi = 1,$$

for a Grassmann variable ψ.

The quark and antiquark fields in (9.9) take values in the Grassmann algebra. In principle, the Grassmann-valued quark fields could be represented explicitly, but this would require a representation space of dimension 2^{12V}, growing exponentially with the lattice volume V. In contrast, storing the gauge field requires $9 \cdot 4V$ complex numbers, which only increases linearly with the lattice volume. Consequently, a direct simulation is completely infeasible for any realistic lattice size. Instead, we make use of the *Matthews–Salam* formula [90]

$$\int \exp\Big(- \sum\nolimits_{i,j} \bar{\boldsymbol{\psi}}_i \boldsymbol{A}_{i,j} \boldsymbol{\psi}_j \Big) [\mathrm{d}\bar{\boldsymbol{\psi}}][\mathrm{d}\boldsymbol{\psi}] = \det(\boldsymbol{A})$$

and solve the Berezin integral over the quark and antiquark fields analytically, resulting in the partition function

$$Z = \int \exp\Big(-\mathcal{S}^{\{\mathfrak{G}\}}([\boldsymbol{U}])\Big) \prod_{f=1}^{N_\mathrm{f}} \det(\boldsymbol{D}(m_{0,f}))[\mathrm{d}\boldsymbol{U}]. \tag{9.10}$$

The Wilson–Dirac operator has an important property

$$\boldsymbol{\Gamma}_5 \boldsymbol{D}(m_{0,f}) \boldsymbol{\Gamma}_5 = \boldsymbol{D}(m_{0,f})^\dagger, \tag{9.11}$$

where $\boldsymbol{\Gamma}_5 \in \mathbb{C}^{12V \times 12V}$ has components

$$[\boldsymbol{\Gamma}_5]_{i,j}^{\alpha,\beta} = [\boldsymbol{\gamma}_5]^{\alpha,\beta} \delta_{i,j} \delta(\boldsymbol{n}, \boldsymbol{m}), \qquad \boldsymbol{\gamma}_5 \coloneqq \boldsymbol{\gamma}_1 \boldsymbol{\gamma}_2 \boldsymbol{\gamma}_3 \boldsymbol{\gamma}_4,$$

for each $(\boldsymbol{n}, \boldsymbol{m})$. This property is called the $\boldsymbol{\gamma}_5$-*hermiticity*. Since $\det(\boldsymbol{\Gamma}_5) = 1$, the $\boldsymbol{\gamma}_5$-hermiticity (9.11) yields

$$\det(\boldsymbol{D}(m_{0,f})) = \det(\boldsymbol{\Gamma}_5 \boldsymbol{D}(m_{0,f}) \boldsymbol{\Gamma}_5) = \det(\boldsymbol{D}(m_{0,f})^\dagger) = \big(\det(\boldsymbol{D}(m_{0,f}))\big)^*, \tag{9.12}$$

showing that the fermion determinant $\det(\boldsymbol{D}(m_{0,f}))$ is real.

The following discussions are rather general and apply to quark flavor configurations requiring multiple pseudofermion fields. For the sake of simplicity, however, we will from now on consider the case of $N_\mathrm{f} = 2$ mass-degenerate Wilson quarks, i.e., both flavors have the same mass m_0 and thus the same Dirac operator $\boldsymbol{D}(m_0)$. This is a close approximation to the real world, where both the up- and down-quarks have similar masses [77]. For a pair of mass-degenerate fermion flavors, the partition function (9.10) can be simplified to

$$Z = \int \exp\Big(-\mathcal{S}^{\{\mathfrak{G}\}}([\boldsymbol{U}])\Big) \det\left(\boldsymbol{D}(m_0) \boldsymbol{D}(m_0)^\dagger\right) [\mathrm{d}\boldsymbol{U}], \tag{9.13}$$

as (9.12) implies $\det(\boldsymbol{D}(m_0))^2 = \det\left(\boldsymbol{D}(m_0) \boldsymbol{D}(m_0)^\dagger\right)$.

The most widely used method to include the fermion determinant in (9.13) is to represent it by an integral over a complex-valued auxiliary *pseudofermion* field $\boldsymbol{\phi}$

$$\det(\boldsymbol{D}(m_0) \boldsymbol{D}(m_0)^\dagger) = \int \exp\Big(-(\boldsymbol{D}(m_0)^{-1} \boldsymbol{\phi})^\dagger \boldsymbol{D}(m_0)^{-1} \boldsymbol{\phi}\Big) \, [\mathrm{d}\boldsymbol{\phi}], \tag{9.14}$$

where the representation is based on the integral representation

$$\frac{1}{\det(\boldsymbol{A})} = \int \exp(-\boldsymbol{\eta}^\dagger \boldsymbol{A} \boldsymbol{\eta}) [\mathrm{d}\boldsymbol{\eta}] \tag{9.15}$$

for $\boldsymbol{A} \in \mathbb{C}^{n \times n}$, $\boldsymbol{\eta} \in \mathbb{C}^n$, and $[\mathrm{d}\boldsymbol{\eta}] = \prod_{i=1}^n \frac{\mathrm{d}\mathfrak{Re}(\eta_i) \mathrm{d}\mathfrak{Im}(\eta_i)}{\pi}$ which ensures the normalization

$$\int \exp(-\boldsymbol{\eta}^\dagger \boldsymbol{I} \boldsymbol{\eta}) [\mathrm{d}\boldsymbol{\eta}] = 1.$$

The integral (9.15) is absolutely convergent, iff $\boldsymbol{A} + \boldsymbol{A}^\dagger$ is positive definite [44].

We replace the fermion action $\mathcal{S}^{\{\mathfrak{F}\}}$ with the *pseudofermion action*

$$\mathcal{S}^{\{\mathfrak{PF}\}}([\boldsymbol{U}, \boldsymbol{\phi}]) = \langle \boldsymbol{D}(m_0)^{-1}\boldsymbol{\phi}, \boldsymbol{D}(m_0)^{-1}\boldsymbol{\phi}\rangle = \langle \boldsymbol{\phi}, (\boldsymbol{D}(m_0)\boldsymbol{D}(m_0)^\dagger)^{-1}\boldsymbol{\phi}\rangle, \tag{9.16}$$

where the inner product of two pseudofermion fields $\boldsymbol{\phi}$ and $\tilde{\boldsymbol{\phi}}$ is defined as

$$\langle \boldsymbol{\phi}, \tilde{\boldsymbol{\phi}}\rangle = \sum_{\boldsymbol{n},\alpha,j} (\phi_j^\alpha(\boldsymbol{n}))^* \tilde{\phi}_j^\alpha(\boldsymbol{n}).$$

The partition function becomes

$$Z = \int \exp\left(-\mathcal{S}^{\{\mathfrak{G}\}}([\boldsymbol{U}]) - \mathcal{S}^{\{\mathfrak{PF}\}}([\boldsymbol{U}, \boldsymbol{\phi}])\right) [\mathrm{d}\boldsymbol{\phi}][\mathrm{d}\boldsymbol{U}]. \tag{9.17}$$

Note that $\boldsymbol{\phi}(\boldsymbol{n}) \in \mathbb{C}^{12}$ such that a pseudofermion field consists of $12V$ complex numbers.

In summary, introducing pseudofermion fields yields a partition function (9.17) that has a similar structure to the partition function (9.5) of the pure gauge theory. Analogously, the path integral (9.4) generalizes to

$$\langle \mathcal{O}([\boldsymbol{U}])\rangle = \frac{1}{Z}\int \mathcal{O}([\boldsymbol{U}]) \exp\left(-\mathcal{S}([\boldsymbol{U}, \boldsymbol{\phi}])\right) [\mathrm{d}\boldsymbol{\phi}][\mathrm{d}\boldsymbol{U}], \tag{9.18}$$

with action

$$\mathcal{S}([\boldsymbol{U}, \boldsymbol{\phi}]) = \mathcal{S}^{\{\mathfrak{G}\}}([\boldsymbol{U}]) + \mathcal{S}^{\{\mathfrak{PF}\}}([\boldsymbol{U}, \boldsymbol{\phi}]). \tag{9.19}$$

9.2. The HMC Algorithm in Lattice QCD

The most widely used method to perform lattice QCD simulations is the HMC algorithm from Chapter 8. By introducing a field of fictitious momenta $[\boldsymbol{P}]$, we obtain a Hamiltonian system whose phase space is the cotangent bundle $([\boldsymbol{P}], [\boldsymbol{U}]) \in T^*\mathcal{G}$ over the matrix Lie group $\mathcal{G} = \mathrm{SU}(3)^{4V}$. The fibers are isomorphic to its Lie algebra $\mathfrak{g} = \mathfrak{su}(3)^{4V}$, where

$$\mathfrak{su}(3) \coloneqq \left\{\boldsymbol{A} \in \mathbb{C}^{3\times 3} :\ \boldsymbol{A}^\dagger = -\boldsymbol{A},\ \mathrm{tr}(\boldsymbol{A}) = 0\right\} \tag{9.20}$$

is the Lie algebra of SU(3). The Lie algebra $\mathfrak{su}(3)$ is an eight-dimensional linear space and has a basis consisting of generators $\boldsymbol{T}_j = \frac{\mathrm{i}}{\sqrt{2}}\boldsymbol{\lambda}_j$, $j = 1, \ldots, 8$, where $\boldsymbol{\lambda}_j$ are the *Gell-Mann matrices* [52]

$$\begin{aligned}
\boldsymbol{\lambda}_1 &= \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \boldsymbol{\lambda}_2 = \begin{pmatrix} 0 & -\mathrm{i} & 0 \\ \mathrm{i} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \boldsymbol{\lambda}_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \\
\boldsymbol{\lambda}_4 &= \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \quad \boldsymbol{\lambda}_5 = \begin{pmatrix} 0 & 0 & -\mathrm{i} \\ 0 & 0 & 0 \\ \mathrm{i} & 0 & 0 \end{pmatrix}, \quad \boldsymbol{\lambda}_6 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \\
\boldsymbol{\lambda}_7 &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -\mathrm{i} \\ 0 & \mathrm{i} & 0 \end{pmatrix}, \quad \boldsymbol{\lambda}_8 = \frac{1}{\sqrt{3}}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -2 \end{pmatrix}.
\end{aligned} \tag{9.21}$$

Each Lie algebra element $\boldsymbol{P}_\mu(\boldsymbol{n})$ can be expressed as $\boldsymbol{P}_\mu(\boldsymbol{n}) = \sum_{j=1}^8 p_\mu^j(\boldsymbol{n})\boldsymbol{T}_j$ with real-valued coefficients $p_\mu^j(\boldsymbol{n}) \in \mathbb{R}$. The generators $\boldsymbol{T}_j$ are linked to the Lie group elements $\boldsymbol{U} \in \mathrm{SU}(3)$ via right-invariant differential operators $\boldsymbol{e}_j$ whose action on $\boldsymbol{U}$ is defined as

$$\boldsymbol{e}_j(\boldsymbol{U}) = -\boldsymbol{T}_j\boldsymbol{U}. \tag{9.22}$$

The differential operators (9.22) can be regarded as a generalization of the vector fields $\partial/\partial q_j$ to the Lie group setting. As an extension to the gauge field $[\boldsymbol{U}]$, we define $8 \cdot 4V$ right-invariant differential operators $\boldsymbol{e}^j_{\boldsymbol{n},\mu}$ whose action on a function of the lattice gauge field $\mathcal{F}([\boldsymbol{U}])$ is defined as

$$\boldsymbol{e}^j_{\boldsymbol{n},\mu}(\mathcal{F}([\boldsymbol{U}])) = \left.\frac{\mathrm{d}\mathcal{F}([\boldsymbol{U}_\epsilon])}{\mathrm{d}\epsilon}\right|_{\epsilon=0}, \quad (\boldsymbol{U}_\epsilon)_\nu(\boldsymbol{m}) = \begin{cases} \exp(-\epsilon \boldsymbol{T}_j)\boldsymbol{U}_\mu(\boldsymbol{n}), & (\nu,\boldsymbol{m}) = (\mu,\boldsymbol{n}), \\ \boldsymbol{U}_\nu(\boldsymbol{m}), & (\nu,\boldsymbol{m}) \neq (\mu,\boldsymbol{n}). \end{cases}$$

At this point we have all ingredients to define the Hamiltonian system. Given the action $\mathcal{S}([\boldsymbol{U}])$, the Hamiltonian reads

$$\mathcal{H}([\boldsymbol{P}],[\boldsymbol{U}]) = \mathcal{T}([\boldsymbol{P}]) + \mathcal{S}([\boldsymbol{U}]), \qquad \mathcal{T}([\boldsymbol{P}]) = \frac{1}{2}\sum_{\boldsymbol{n},\mu} \mathrm{tr}(\boldsymbol{P}_\mu(\boldsymbol{n})^\dagger \boldsymbol{P}_\mu(\boldsymbol{n})). \tag{9.23}$$

With the representation $\boldsymbol{P}_\mu(\boldsymbol{n}) = \sum_{j=1}^8 p^j_\mu(\boldsymbol{n})\boldsymbol{T}_j$, a straightforward computation shows

$$\mathrm{tr}(\boldsymbol{P}_\mu(\boldsymbol{n})^\dagger \boldsymbol{P}_\mu(\boldsymbol{n})) = \sum_{j=1}^8 p^j_\mu(\boldsymbol{n})^2.$$

Consequently, the Hamiltonian (9.23) is a separable Hamiltonian system of the form (3.32) with mass matrix $\boldsymbol{M} = \boldsymbol{I}$ and the action $\mathcal{S}([\boldsymbol{U}])$ taking the role of the potential energy $\mathcal{V}(\boldsymbol{q})$. Hence, splitting methods of RKN type, as well as (Hessian-free) force-gradient integrators, are applicable.

Including pseudofermion fields $\boldsymbol{\phi}$ requires augmenting Algorithm 8.1 with a generation of the pseudofermion field $\boldsymbol{\phi} \propto \exp(-\mathcal{S}^{\{\mathfrak{PF}\}}([\boldsymbol{U},\boldsymbol{\phi}]))$ at the beginning of each trajectory for fixed $[\boldsymbol{U}]$, cf. [84]. The pseudofermion field $\boldsymbol{\phi}$ remains unchanged throughout the molecular dynamics step.

9.2.1. Hamiltonian Equations of Motion

In the molecular dynamics step of Algorithm 8.1, we have to numerically solve the equations of motion corresponding to the Hamiltonian (9.23). Similar to Euclidean space, Hamiltonian equations of motion are most naturally expressed in terms of the Lie derivatives [75]

$$\hat{\mathcal{T}} = \sum_{\boldsymbol{n},\mu,j} p^j_\mu(\boldsymbol{n})\boldsymbol{e}^j_{\boldsymbol{n},\mu}, \quad \hat{\mathcal{S}} = -\sum_{\boldsymbol{n},\mu,j} \boldsymbol{e}^j_{\boldsymbol{n},\mu}(\mathcal{S})([\boldsymbol{U},\boldsymbol{\phi}])\frac{\partial}{\partial p^j_\mu(\boldsymbol{n})}. \tag{9.24}$$

We obtain the equations of motion

$$\begin{aligned} \dot{\boldsymbol{P}}_\mu(\boldsymbol{n}) &= -\textstyle\sum_j \boldsymbol{e}^j_{\boldsymbol{n},\mu}(\mathcal{S})([\boldsymbol{U},\boldsymbol{\phi}])\dfrac{\partial \boldsymbol{P}_\mu(\boldsymbol{n})}{\partial p^j_\mu(\boldsymbol{n})} = -\textstyle\sum_j \boldsymbol{e}^j_{\boldsymbol{n},\mu}(\mathcal{S})([\boldsymbol{U},\boldsymbol{\phi}])\boldsymbol{T}_j, \\ \dot{\boldsymbol{U}}_\mu(\boldsymbol{n}) &= \textstyle\sum_j p^j_\mu(\boldsymbol{n})\boldsymbol{e}^j_{\boldsymbol{n},\mu}(\boldsymbol{U}_\mu(\boldsymbol{n})) = -\sum_j p^j_\mu(\boldsymbol{n})\boldsymbol{T}_j\boldsymbol{U}_\mu(\boldsymbol{n}) = -\boldsymbol{P}_\mu(\boldsymbol{n})\boldsymbol{U}_\mu(\boldsymbol{n}). \end{aligned} \tag{9.25}$$

Depending on the action, the force $\mathcal{F}_{\boldsymbol{n},\mu}([\boldsymbol{U},\boldsymbol{\phi}]) \coloneqq -\sum_j \boldsymbol{e}^j_{\boldsymbol{n},\mu}(\mathcal{S})([\boldsymbol{U},\boldsymbol{\phi}])\boldsymbol{T}_j$ can be computed further. In the subsequent section, we will compute the Wilson gauge force and the Wilson fermion force (for $N_\mathrm{f} = 2$ mass-degenerate Wilson quarks).

9.2.2. The Gauge Force and the Fermion Force

The Wilson gauge force reads [77]

$$\mathcal{F}^{\{\mathfrak{G}\}}_{\boldsymbol{n},\mu}([\boldsymbol{U}]) = -\sum_j \boldsymbol{e}^j_{\boldsymbol{n},\mu}(\mathcal{S}^{\{\mathfrak{G}\}})([\boldsymbol{U}])\boldsymbol{T}_j = \frac{\beta}{3}\left\{\boldsymbol{U}_\mu(\boldsymbol{n})\Sigma_\mu(\boldsymbol{n})\right\}_{\mathrm{TA}}, \tag{9.26}$$

where

$$\Sigma_\mu(\boldsymbol{n}) = \sum_{\nu\neq\mu} [\boldsymbol{U}_\nu(\boldsymbol{n})^\dagger \boldsymbol{U}_\mu(\boldsymbol{n}+a\hat{\boldsymbol{\nu}})^\dagger \boldsymbol{U}_\nu(\boldsymbol{n}+a\hat{\boldsymbol{\mu}}) + \boldsymbol{U}_\nu(\boldsymbol{n}-a\hat{\boldsymbol{\nu}})\boldsymbol{U}_\mu(\boldsymbol{n}-a\hat{\boldsymbol{\nu}})^\dagger \boldsymbol{U}_\nu(\boldsymbol{n}+a(\hat{\boldsymbol{\mu}}-\hat{\boldsymbol{\nu}}))^\dagger] \quad (9.27)$$

denotes the *sum of staples*[2] connected to the link $\boldsymbol{U}_\mu(\boldsymbol{n})$ and $\{\cdot\}_{\mathrm{TA}}$ denotes the traceless and anti-Hermitian part, i.e.,

$$\{\boldsymbol{A}\}_{\mathrm{TA}} \coloneqq \frac{\boldsymbol{A}-\boldsymbol{A}^\dagger}{2} - \frac{\mathrm{tr}(\boldsymbol{A}-\boldsymbol{A}^\dagger)}{6}\boldsymbol{I}.$$

The pseudofermion force reads [77]

$$\mathcal{F}^{\{\mathfrak{PF}\}}_{\boldsymbol{n},\mu}([\boldsymbol{U},\boldsymbol{\phi}]) = -\sum_k \boldsymbol{e}^k_{\boldsymbol{n},\mu}(\mathcal{S}^{\{\mathfrak{PF}\}})([\boldsymbol{U},\boldsymbol{\phi}])\boldsymbol{T}_k = -2\sum_k \mathfrak{Re}\langle \boldsymbol{\chi}, \Delta^k_{\boldsymbol{n},\mu}\boldsymbol{D}(m_0)\boldsymbol{\eta}\rangle \boldsymbol{T}_k, \quad (9.28a)$$

where $\boldsymbol{\eta} \coloneqq \boldsymbol{D}(m_0)^{-1}\boldsymbol{\phi}$, $\boldsymbol{\chi} \coloneqq (\boldsymbol{D}(m_0)^\dagger)^{-1}\boldsymbol{\eta} = (\boldsymbol{D}(m_0)\boldsymbol{D}(m_0)^\dagger)^{-1}\boldsymbol{\phi}$, and

$$\left[(\Delta^k_{\boldsymbol{n},\mu}\boldsymbol{D}(m_0)\boldsymbol{\eta})(\boldsymbol{m})\right]^{\alpha,\beta}_{i,j} = \delta(\boldsymbol{n},\boldsymbol{m})\kappa[\boldsymbol{I}-\boldsymbol{\gamma}_\mu]^{\alpha,\beta}[\boldsymbol{U}_\mu(\boldsymbol{n})^\dagger \boldsymbol{T}_k]_{i,j}\boldsymbol{\eta}(\boldsymbol{n}+a\hat{\boldsymbol{\mu}})^\beta_j - \delta(\boldsymbol{m},\boldsymbol{n}+a\hat{\boldsymbol{\mu}})\kappa[\boldsymbol{I}+\boldsymbol{\gamma}_\mu]^{\alpha,\beta}[\boldsymbol{T}_k\boldsymbol{U}_\mu(\boldsymbol{n})]_{i,j}\boldsymbol{\eta}(\boldsymbol{n})^\beta_j. \quad (9.28b)$$

9.2.3. The Multirate Potential in Lattice QCD

In the case of $N_{\mathrm{f}} = 2$ mass-degenerate Wilson quarks without further preconditioning, the action in (9.23) consists of the gauge action $\mathcal{S}^{\{\mathfrak{G}\}}$ and the pseudofermion action $\mathcal{S}^{\{\mathfrak{PF}\}}$, i.e., $\mathcal{S} = \mathcal{S}^{\{\mathfrak{G}\}} + \mathcal{S}^{\{\mathfrak{PF}\}}$. The local structure of the sum of staples (9.27) makes the gauge force (9.26) relatively cheap to evaluate. In contrast, evaluating the pseudofermion force (9.28) demands *inversions* of the Wilson–Dirac operator, i.e., we have to solve the linear system

$$\boldsymbol{D}(m_0)\boldsymbol{D}(m_0)^\dagger\boldsymbol{\chi} = \boldsymbol{\phi}. \quad (9.29)$$

The linear system (9.29) has dimension $12V$ and is sparse. Therefore, sparse linear algebra solvers such as the generalized minimal residual (GMRES) method [108] or the conjugate gradient (CG) method [67] are common choices to solve the linear system (9.29). Although the system is sparse, the pseudofermion action introduces a global coupling of the links. Therefore, evaluating the pseudofermion force (9.28) constitutes by far the major cost in standard HMC simulations.

Simultaneously, the contribution of the gauge force to the dynamics of the solution is much larger than the contribution of the pseudofermion force, cf. [85, Fig. 3].

Consequently, the Hamiltonian system (9.23) exhibits multirate potential that can be exploited by means of multiple time stepping techniques, such as the hierarchical (Hessian-free) force-gradient integrators from Chapter 7.

In the present form, we would split the Hamiltonian into its kinetic part, the gauge part, and the fermion part, i.e., into a three-split system. In practice, factorizations of the fermion determinant $\det(\boldsymbol{D}(m_0)\boldsymbol{D}(m_0)^\dagger)$ are utilized. Prominent examples are *even-odd preconditioning* [33] and *Hasenbusch mass preconditioning* [65]. Although these techniques are originally preconditioners intended to lower the linear algebra cost when computing the inversions of the Wilson–Dirac operator, they simultaneously split the pseudofermion action into $k \in \mathbb{N}$ components

$$\mathcal{S}^{\{\mathfrak{PF}\}} = \sum_{m=1}^k \mathcal{S}^{\{\mathfrak{PF},m\}},$$

[2] A staple is the product of the three links in each plaquette containing $\boldsymbol{U}_\mu(\boldsymbol{n})$ other than $\boldsymbol{U}_\mu(\boldsymbol{n})$ itself.

which often also introduces multirate behavior. Let $\mathcal{F}^{\{m\}}$, $m = 1, \ldots, k$, denote the force corresponding to the action $\mathcal{S}^{\{\mathfrak{P}\mathfrak{F},m\}}$ and $\mathcal{F}^{\{k+1\}}$ the gauge force. We can assign the gauge force and the pseudofermion forces to groups of forces characterized by similar evaluation cost and similar contributions to the overall dynamics of the solution. Therefore, we define $K \leq k+1$ disjoint index sets $I_1, \ldots, I_K \subset \{1, \ldots, k+1\}$ such that

$$\bigcup_{j=1}^{K} I_j = \{1, \ldots, k+1\}, \quad I_j \cap I_\ell = \emptyset \text{ for } j \neq \ell.$$

Each index set gives rise to a group of forces

$$\mathcal{F}^{\{I_j\}} \coloneqq \sum_{m \in I_j} \mathcal{F}^{\{m\}}$$

with comparable evaluation costs and contribution to the overall dynamics. We use the convention that the groups are sorted such that the evaluation costs are decreasing while the contribution to the dynamics simultaneously increases. This provides us with a multirate hierarchy

$$\begin{array}{ccccccccc} \operatorname{cost}(\mathcal{F}^{\{I_1\}}) & > & \operatorname{cost}(\mathcal{F}^{\{I_2\}}) & > \ldots > & \operatorname{cost}(\mathcal{F}^{\{I_{K-1}\}}) & > & \operatorname{cost}(\mathcal{F}^{\{I_K\}}), \\ \|\mathcal{F}^{\{I_1\}}\| & < & \|\mathcal{F}^{\{I_2\}}\| & < \ldots < & \|\mathcal{F}^{\{I_{K-1}\}}\| & < & \|\mathcal{F}^{\{I_K\}}\|, \end{array}$$

which we can exploit by utilizing multiple time stepping techniques. For a hierarchical splitting method and in particular the splitting tree from Fig. 7.1, we will consider $K+1$ leaf nodes, and the actions corresponding to forces of group j are assigned to the leaf node with label $\mathcal{V}^{\{j\}}$.

As examples, we will discuss even-odd reduction and Hasenbusch mass preconditioning in more detail.

9.2.4. Even-Odd Reduction

Even-odd reduction is a standard preconditioning for Wilson fermions that relies on an indexing of the lattice points such that the even points ($\sum_\mu n_\mu \bmod 2 = 0$) are ordered first, followed by the odd points ($\sum_\mu n_\mu \bmod 2 = 1$). The Wilson–Dirac operator then exhibits the block structure

$$\boldsymbol{D}(m_0) = \begin{pmatrix} \boldsymbol{D}_{\mathrm{ee}} & \boldsymbol{D}_{\mathrm{eo}} \\ \boldsymbol{D}_{\mathrm{oe}} & \boldsymbol{D}_{\mathrm{oo}} \end{pmatrix}. \tag{9.30}$$

For (9.8), the matrices $\boldsymbol{D}_{\mathrm{ee}}$ and $\boldsymbol{D}_{\mathrm{oo}}$ are proportional to the identity matrix. The Schur decomposition of (9.30) is given by [77]

$$\boldsymbol{D}(m_0) = \begin{pmatrix} \boldsymbol{I} & \boldsymbol{D}_{\mathrm{eo}}\boldsymbol{D}_{\mathrm{oo}}^{-1} \\ \boldsymbol{0} & \boldsymbol{I} \end{pmatrix} \begin{pmatrix} \hat{\boldsymbol{D}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{D}_{\mathrm{oo}} \end{pmatrix} \begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{D}_{\mathrm{oo}}^{-1}\boldsymbol{D}_{\mathrm{oe}} & \boldsymbol{I} \end{pmatrix}, \tag{9.31}$$

where

$$\hat{\boldsymbol{D}} \coloneqq \boldsymbol{D}_{\mathrm{ee}} - \boldsymbol{D}_{\mathrm{eo}}\boldsymbol{D}_{\mathrm{oo}}^{-1}\boldsymbol{D}_{\mathrm{oe}} \tag{9.32}$$

is the Schur complement that is only defined on the even lattice points. It is called the *even-odd preconditioned Dirac operator*. From the Schur decomposition (9.31), it follows

$$\det(\boldsymbol{D}(m_0)) = \det(\boldsymbol{D}_{\mathrm{oo}}) \det(\hat{\boldsymbol{D}}).$$

Both factors are real [77]. For $N_\mathrm{f} = 2$ mass-degenerate Wilson quarks, we obtain the factorization

$$\det(\boldsymbol{D}(m_0)\boldsymbol{D}(m_0)^\dagger) = \det(\boldsymbol{D}_\mathrm{oo})^2 \det(\hat{\boldsymbol{D}}\hat{\boldsymbol{D}}^\dagger). \tag{9.33}$$

We can insert the factorization (9.33) into (9.13). Due to the structure of the block-diagonal term, we can compute the corresponding term exactly. With the identity $-\log(\det(\boldsymbol{A})) = -\operatorname{tr}(\log(\boldsymbol{A}))$, and introducing a pseudofermion field $\boldsymbol{\phi}_\mathrm{e}$ that is only defined on the even lattice points for $\det(\hat{\boldsymbol{D}})$, we obtain the pseudofermion action

$$\mathcal{S}^{\{\mathfrak{PF}\}}([\boldsymbol{U}, \boldsymbol{\phi}_\mathrm{e}]) = -2\operatorname{tr}\big(\log(\boldsymbol{D}_\mathrm{oo})\big) + \langle \boldsymbol{\phi}_\mathrm{e}, (\hat{\boldsymbol{D}}\hat{\boldsymbol{D}}^\dagger)^{-1}\boldsymbol{\phi}_\mathrm{e}\rangle. \tag{9.34}$$

The contribution of the first term in (9.34) to the force can be computed exactly. Similar to (9.28), the force originating from the operator $\hat{\boldsymbol{D}}$ is

$$-2\sum_k \mathfrak{Re}\langle \boldsymbol{\chi}_\mathrm{e}, \Delta^k_{\boldsymbol{n},\mu}\hat{\boldsymbol{D}}\boldsymbol{\eta}_\mathrm{e}\rangle_\mathrm{e}\boldsymbol{T}_k,$$

where $\boldsymbol{\eta}_\mathrm{e} \coloneqq \hat{\boldsymbol{D}}^{-1}\boldsymbol{\phi}_\mathrm{e}$ and $\boldsymbol{\chi}_\mathrm{e} \coloneqq (\hat{\boldsymbol{D}}^\dagger)^{-1}\boldsymbol{\eta}_\mathrm{e} = (\hat{\boldsymbol{D}}\hat{\boldsymbol{D}}^\dagger)^{-1}\boldsymbol{\phi}_\mathrm{e}$. Due to the preconditioning, the Dirac equation (9.29) involving $\hat{\boldsymbol{D}}\hat{\boldsymbol{D}}^\dagger$ is cheaper to solve.

9.2.5. Hasenbusch Mass Preconditioning

Another preconditioning technique relies on the identity [77]

$$\det(\boldsymbol{D}(m_0)\boldsymbol{D}(m_0)^\dagger) = \det(\check{\boldsymbol{D}}\check{\boldsymbol{D}}^\dagger)\det((\boldsymbol{D}(m_0)\check{\boldsymbol{D}}^{-1})(\boldsymbol{D}(m_0)\check{\boldsymbol{D}}^{-1})^\dagger), \tag{9.35}$$

which holds for any non-singular matrix $\check{\boldsymbol{D}}$. Note that when even-odd reduction is utilized, the factorization (9.35) can be applied to the even-odd preconditioned Dirac operator (9.32) instead of the full Dirac operator $\boldsymbol{D}(m_0)$. Originally, the *Hasenbusch mass preconditioning* [65] considers the factorization (9.35) with $\check{\boldsymbol{D}} = \boldsymbol{D}(m_1)$, where $m_1 > m_0$. Another special case of the factorization (9.35) is obtained by choosing $\check{\boldsymbol{D}}$ such that, for each $(\boldsymbol{n}, \boldsymbol{m})$, the matrix has components

$$[\check{\boldsymbol{D}}]^{\alpha,\beta}_{i,j} = [\boldsymbol{D}(m_0)]^{\alpha,\beta}_{i,j} + \mathrm{i}\mu[\boldsymbol{\gamma}_5]^{\alpha,\beta}\delta_{i,j}\delta(\boldsymbol{n}, \boldsymbol{m}),$$

with *shift parameter* μ. Then, the factorization (9.35) becomes

$$\det(\boldsymbol{D}(m_0)\boldsymbol{D}(m_0)^\dagger) = \det(\boldsymbol{D}(m_0)\boldsymbol{D}(m_0)^\dagger + \mu^2\boldsymbol{I}) \cdot \frac{\det(\boldsymbol{D}(m_0)\boldsymbol{D}(m_0)^\dagger)}{\det(\boldsymbol{D}(m_0)\boldsymbol{D}(m_0)^\dagger + \mu^2\boldsymbol{I})}, \tag{9.36}$$

where we use the identity $\det(\boldsymbol{D}(m_0)\boldsymbol{D}(m_0)^\dagger + \mu^2\boldsymbol{I}) = \det(\check{\boldsymbol{D}}\check{\boldsymbol{D}}^\dagger)$ that can be shown via the $\boldsymbol{\gamma}_5$-hermiticity (9.11). Since the first determinant of (9.36) is identical to the determinant of the Dirac operator for $N_\mathrm{f} = 2$ twisted-mass Wilson fermions [77], the shift parameter μ is also called *twisted-mass parameter*. The shift parameter μ separates the high modes larger than μ^2 of the Dirac operator $\boldsymbol{D}(m_0)$ from the lower modes. The high modes dominate the first determinant of (9.36), whereas the lower modes dominate the ratio in (9.36).

For the twisted-mass choice, inserting the factorization (9.35) into (9.13) and introducing two pseudofermion fields $\boldsymbol{\phi}_1$, $\boldsymbol{\phi}_2$ results in the pseudofermion action

$$\mathcal{S}^{\{\mathfrak{PF}\}}([\boldsymbol{U}, \boldsymbol{\phi}_1, \boldsymbol{\phi}_2]) \;=\; \langle \boldsymbol{\phi}_1, (\check{\boldsymbol{D}}\check{\boldsymbol{D}}^\dagger)^{-1}\boldsymbol{\phi}_1\rangle \;+\; \langle \boldsymbol{\phi}_2, (\boldsymbol{D}(m_0)\boldsymbol{D}(m_0)^\dagger)^{-1}\check{\boldsymbol{D}}\check{\boldsymbol{D}}^\dagger\boldsymbol{\phi}_2\rangle. \tag{9.37}$$

Originally, the idea was to choose $\check{\boldsymbol{D}}$ such that both $\check{\boldsymbol{D}}$ and $\boldsymbol{D}(m_0)\check{\boldsymbol{D}}^{-1}$ are better-conditioned matrices, cf. [65]. In the spirit of multiple time stepping techniques, we may choose $\check{\boldsymbol{D}}$ such that the forces originating from the two parts in (9.37) introduce

a multirate potential. The parameters are chosen such that the force corresponding to the first part of (9.37) is cheap to compute and the dominating contribution to the total pseudofermion force, whereas the force corresponding to the second part of (9.37) is more expensive to compute, but contributes less to the total force, cf. [138]. This approach is closely related to the separation of infrared (low-frequency) and ultraviolet (high-frequency) modes [104].

Remark 9.2. The determinant factorization (9.35) can be generalized by introducing $r \in \mathbb{N}$ matrices $\check{\boldsymbol{D}}_i$, $i = 1, \dots, r$, and considering the factorization

$$\det(\boldsymbol{D}(m_0)\boldsymbol{D}(m_0)^\dagger) = \det(\check{\boldsymbol{D}}_r\check{\boldsymbol{D}}_r^\dagger) \cdot \prod_{i=1}^{r} \frac{\det(\check{\boldsymbol{D}}_{i-1}\check{\boldsymbol{D}}_{i-1}^\dagger)}{\det(\check{\boldsymbol{D}}_i\check{\boldsymbol{D}}_i^\dagger)}, \tag{9.38}$$

where $\check{\boldsymbol{D}}_0 \coloneqq \boldsymbol{D}(m_0)$. In this form, the pseudofermion action splits into $r+1$ partitions.

9.3. Splitting Methods for Lattice QCD

Since the Hamiltonian (9.23) is of the form (3.32), we can employ (hierarchical) splitting and force-gradient approaches from the preceding chapters. Compared to Euclidean space, the flows of the subsystems, as well as the (Hessian-free) force-gradient updates, change. In particular, the application to lattice QCD introduces an additional demand on the numerical integration scheme. The ODEs for the link matrices in (9.25) are of Lie-type, i.e., the exact flow evolves on a Lie group manifold. In particular, the link matrices are elements of the special unitary group SU(3). To avoid non-physical numerical approximations, we demand that the numerical integration scheme satisfies the *closure property*, i.e., it yields numerical approximations situated on the Lie group manifold. We will adapt the definitions of splitting methods and (Hessian-free) force-gradient integrators when applied to the Hamiltonian (9.23) encountered in lattice QCD simulations. Since the convergence order and geometric properties of splitting methods depend only on the algebraic properties of the Lie derivatives and the group structure of the flows, the order conditions and the geometric properties established in the first part of this thesis remain valid in the Lie group setting. Moreover, splitting methods and (Hessian-free) force-gradient integrators satisfy the additional closure property. The same holds for the hierarchical splitting approach, i.e., we can apply the techniques from Chapter 7 without modification.

9.3.1. Splitting Methods

Splitting the Hamiltonian (9.23) into its kinetic energy $\mathcal{T}$ and its action $\mathcal{S}$ results in the two subsystems

$$\begin{pmatrix} \dot{\boldsymbol{P}}_\mu(\boldsymbol{n}) \\ \dot{\boldsymbol{U}}_\mu(\boldsymbol{n}) \end{pmatrix} = \begin{pmatrix} \boldsymbol{0} \\ -\boldsymbol{P}_\mu(\boldsymbol{n})\boldsymbol{U}_\mu(\boldsymbol{n}) \end{pmatrix}, \tag{9.39a}$$

and

$$\begin{pmatrix} \dot{\boldsymbol{P}}_\mu(\boldsymbol{n}) \\ \dot{\boldsymbol{U}}_\mu(\boldsymbol{n}) \end{pmatrix} = \begin{pmatrix} \mathcal{F}_{\boldsymbol{n},\mu}([\boldsymbol{U}, \boldsymbol{\phi}]) \\ \boldsymbol{0} \end{pmatrix}. \tag{9.39b}$$

The two subsystems can be solved analytically. The flows are given by

$$\varphi_t^{\{\mathcal{T}\}}([\boldsymbol{P}], [\boldsymbol{U}]) : \qquad \begin{pmatrix} \boldsymbol{P}_\mu(\boldsymbol{n}) \\ \boldsymbol{U}_\mu(\boldsymbol{n}) \end{pmatrix} \mapsto \begin{pmatrix} \boldsymbol{P}_\mu(\boldsymbol{n}) \\ \exp(-t\boldsymbol{P}_\mu(\boldsymbol{n}))\boldsymbol{U}_\mu(\boldsymbol{n}) \end{pmatrix}, \tag{9.40a}$$

where $\exp\colon \mathfrak{su}(3) \to \mathrm{SU}(3)$ denotes the matrix exponential, and

$$\varphi_t^{\{\mathcal{S}\}}([\boldsymbol{P}],[\boldsymbol{U}]): \qquad \begin{pmatrix}\boldsymbol{P}_\mu(\boldsymbol{n})\\ \boldsymbol{U}_\mu(\boldsymbol{n})\end{pmatrix} \mapsto \begin{pmatrix}\boldsymbol{P}_\mu(\boldsymbol{n}) + t\mathcal{F}_{\boldsymbol{n},\mu}([\boldsymbol{U},\boldsymbol{\phi}])\\ \boldsymbol{U}_\mu(\boldsymbol{n})\end{pmatrix}, \tag{9.40b}$$

respectively. Consequently, a s-stage splitting method (3.8) is a composition of link updates (9.40a) and momentum updates (9.40b). The link update (9.40a) is the exact flow of the ODE (9.39a) of Lie-type, i.e., the updated link matrices remain elements of SU(3). Since the momentum updates (9.40b) leave the link matrices unchanged, self-adjoint splitting methods are symplectic (thus volume-preserving), time-reversible, and satisfy the closure property.

9.3.2. Force-Gradient Integrators

For defining force-gradient integrators, we need to compute the force-gradient term (4.1) with the Lie derivatives from (9.24). We obtain the force-gradient term

$$[\hat{\mathcal{S}},[\hat{\mathcal{T}},\hat{\mathcal{S}}]] = 2\sum_{\boldsymbol{m},\nu,k}\sum_{\boldsymbol{n},\mu,j} \boldsymbol{e}^k_{\boldsymbol{m},\nu}(\mathcal{S})([\boldsymbol{U},\boldsymbol{\phi}])\boldsymbol{e}^k_{\boldsymbol{m},\nu}(\boldsymbol{e}^j_{\boldsymbol{n},\mu}(\mathcal{S}))([\boldsymbol{U},\boldsymbol{\phi}])\frac{\partial}{\partial p^j_\mu(\boldsymbol{n})}. \tag{9.41}$$

The flow (4.3) corresponding to the force-gradient term (9.41) becomes

$$\varphi_t^{\{\mathcal{C}\}}([\boldsymbol{P}],[\boldsymbol{U}]): \quad \begin{pmatrix}\boldsymbol{P}_\mu(\boldsymbol{n})\\ \boldsymbol{U}_\mu(\boldsymbol{n})\end{pmatrix} \mapsto \begin{pmatrix}\boldsymbol{P}_\mu(\boldsymbol{n}) + 2t\sum\limits_j \sum\limits_{\boldsymbol{m},\nu,k} \boldsymbol{e}^k_{\boldsymbol{m},\nu}(\mathcal{S})\boldsymbol{e}^k_{\boldsymbol{m},\nu}(\boldsymbol{e}^j_{\boldsymbol{n},\mu}(\mathcal{S}))\boldsymbol{T}_j\\ \boldsymbol{U}_\mu(\boldsymbol{n})\end{pmatrix}, \tag{9.42}$$

where both the gradient and the Hessian of the action $\mathcal{S}$ are evaluated in $([\boldsymbol{U},\boldsymbol{\phi}])$. Merging the flow (9.42) with the momentum update (9.40b) yields force-gradient updates

$$\begin{aligned}\varphi_{b_ih,c_ih^3}^{\{\mathcal{S},\mathcal{C}\}}([\boldsymbol{P}],[\boldsymbol{U}]): \quad \begin{pmatrix}\boldsymbol{P}_\mu(\boldsymbol{n})\\ \boldsymbol{U}_\mu(\boldsymbol{n})\end{pmatrix} \mapsto \begin{pmatrix}\boldsymbol{P}_\mu(\boldsymbol{n})\\ \boldsymbol{U}_\mu(\boldsymbol{n})\end{pmatrix} + \begin{pmatrix} b_ih\mathcal{F}_{\boldsymbol{n},\mu}([\boldsymbol{U},\boldsymbol{\phi}])\\ \boldsymbol{0}\end{pmatrix}\\ + \begin{pmatrix} 2c_ih^3\sum\limits_j \sum\limits_{\boldsymbol{m},\nu,k} \boldsymbol{e}^k_{\boldsymbol{m},\nu}(\mathcal{S})\boldsymbol{e}^k_{\boldsymbol{m},\nu}(\boldsymbol{e}^j_{\boldsymbol{n},\mu}(\mathcal{S}))\boldsymbol{T}_j\\ \boldsymbol{0}\end{pmatrix}.\end{aligned} \tag{9.43}$$

Consequently, a s-stage force-gradient integrator (4.8) is a composition of link updates (9.40a) and force-gradient updates (9.43). Since force-gradient integrators use the same link updates (9.40a) as splitting methods, self-adjoint force-gradient integrators have the same geometric properties as splitting methods.

9.3.3. Hessian-Free Force-Gradient Integrators

As for Euclidean space, we can approximate the force-gradient updates (9.43) such that the Hessian of the action is effectively replaced by a second force evaluation. By performing steps analogously to the derivation from Section 4.2.1, we obtain Hessian-free force-gradient updates defined as

$$\varphi_{b_ih,c_ih^3}^{\{\mathcal{D}\}}([\boldsymbol{P}],[\boldsymbol{U}]): \quad \begin{pmatrix}\boldsymbol{P}_\mu(\boldsymbol{n})\\ \boldsymbol{U}_\mu(\boldsymbol{n})\end{pmatrix} \mapsto \begin{pmatrix}\boldsymbol{P}_\mu(\boldsymbol{n}) + b_ih\mathcal{F}_{\boldsymbol{n},\mu}([\boldsymbol{U}',\boldsymbol{\phi}])\\ \boldsymbol{U}_\mu(\boldsymbol{n})\end{pmatrix}, \tag{9.44}$$

where $[\boldsymbol{U}']$ is the result of the temporary link update

$$\boldsymbol{U}'_\mu(\boldsymbol{n}) = \exp\left(-\frac{2c_ih^2}{b_i}\mathcal{F}_{\boldsymbol{n},\mu}([\boldsymbol{U},\boldsymbol{\phi}])\right)\boldsymbol{U}_\mu(\boldsymbol{n}),$$

and exp once again denotes the matrix exponential. Consequently, a Hessian-free force-gradient integrator (4.11) is a composition of link updates (9.40a) and Hessian-free force-gradient updates (9.44). Since Hessian-free force-gradient integrators do not change the link updates (9.40a) compared to conventional force-gradient integrators, they also satisfy the closure property. This makes Hessian-free force-gradient integrators feasible choices for HMC simulations in lattice QCD, provided that they are self-adjoint.

9.4. Tuning of HMC Simulations in Lattice QCD

The primary objective in setting up the HMC algorithm is to obtain sufficiently large statistics at the lowest possible computational cost. As an example, we may be interested in creating an ensemble consisting of $10,000$ configurations where the computation of a single trajectory requires two minutes on a large-scale HPC system. With these values, the ensemble generation takes approximately two weeks of computing time. If we can tune the algorithm such that we only require 90% of the computational cost per trajectory (at a fixed autocorrelation time), the total computing time is reduced by more than one day. This illustrates that even modest relative improvements in the cost per trajectory can have a significant practical impact, motivating a careful selection of the integrator.

When setting up the HMC algorithm for gauge field generation in lattice QCD, there are many degrees of freedom we can choose to optimize the computational efficiency of the method. Note that a discussion on choosing the trajectory length τ, the mass matrix $\boldsymbol{M}$, the numerical integration scheme $\boldsymbol{\Phi}_h$, and its step size h has already been carried out in Section 8.4. However, we would like to mention that, in the context of lattice QCD, there are recent works dealing with the tuning of the trajectory length, cf. [103].

In lattice QCD, there are many system parameters that remain as degrees of freedom, such as the number and values of Hasenbusch masses used in the factorization (9.38). Recent references proposing tuning techniques for Hasenbusch masses are [102, 135]. A thorough discussion of these tuning strategies is beyond the scope of this thesis.

As the focus of this thesis is on the numerical integration schemes utilized in the molecular dynamics step, we assume that the trajectory length τ, as well as the system parameters (such as the Hasenbusch masses), are already fixed. For fixed system parameters, we have a fixed multirate hierarchy of the forces such that we can assign them to K groups of forces characterized by similar evaluation cost and similar contributions to the overall dynamics. This leads to a splitting of the Hamiltonian (9.23) into $K+1$ partitions, thereby fixing the shape of the splitting tree from Fig. 7.1.

We therefore focus on the remaining degrees of freedom: the choice of the numerical integration scheme $\boldsymbol{\Phi}_h$, which is taken to be a (multirate) hierarchical Hessian-free force-gradient integrator, its step size h, and the multirate factors $M^{\{\boldsymbol{v}\}}$ assigned to the nodes $\boldsymbol{v}$ of the splitting tree. This thesis contributes to this tuning process by providing the efficiency measures $\mathrm{Eff}_{\mathtt{err}}^{(p)}$ and $\mathrm{Eff}_{\mathtt{stab}}$, and the leading-order error term (5.8) for hierarchical methods, to select particularly efficient integrator variants for solving the Hamiltonian equations of motion. Besides the choice of the integrator variant, a maximization of the efficiency measures tunes potential degrees of freedom in the integrator coefficients. The step size h, as well as the multirate factors $M^{\{\boldsymbol{v}\}}$, are tuned manually when setting up the integrator for creating a specific ensemble. For this manual tuning, we aim to maximize h and simultaneously minimize the multirate factors $M^{\{\boldsymbol{v}\}}$ such that the acceptance probability is at least the optimal acceptance probability (8.19).

In the spirit of the hierarchical splitting approach, we may replace the exact link updates (9.40a) by numerical approximations situated on the Lie group manifold. In the following chapter, we will present a numerical approximation to the link updates (9.40a) by utilizing a modified Cayley transform for SU(3).

Chapter 10.

A Modified Cayley Transform for SU(3)

This chapter is based on the paper [120] and studies the use of the Cayley transform and a modification thereof as an alternative local parameterization for numerical integration on the special unitary group SU(3). Consider a matrix Lie group $\mathcal{G}$ and its corresponding Lie algebra $\mathfrak{g}$. The differential equations for the link variables in (9.25) are of the form

$$\dot{\boldsymbol{U}} = -\boldsymbol{P}(t)\boldsymbol{U}, \quad \boldsymbol{U}(0) = \boldsymbol{U}_0, \tag{10.1}$$

where $\boldsymbol{U} \in \mathcal{G}$ and $\boldsymbol{P}(t) \in \mathfrak{g}$ is defined implicitly via the differential equations for the momentum variables. An ODE of the form (10.1) is called *ODE of Lie-type*. A formula for the solution of (10.1) is obtained by the Magnus approach [88].

Theorem 10.1 (Magnus approach)**.** *The solution of the ODE* (10.1) *can be written as* $\boldsymbol{U}(t) = \exp(\boldsymbol{\Omega}(t))\boldsymbol{U}_0$ *with* $\boldsymbol{\Omega}(t)$ *defined by the auxiliary ODE*

$$\dot{\boldsymbol{\Omega}} = \mathrm{d}\exp_{\boldsymbol{\Omega}}^{-1}(-\boldsymbol{P}(t)), \quad \boldsymbol{\Omega}(0) = \boldsymbol{0}. \tag{10.2}$$

As long as $\|\boldsymbol{\Omega}(t)\| < \pi$*, the convergence of the expansion*

$$\mathrm{d}\exp_{\boldsymbol{\Omega}}^{-1}(\boldsymbol{A}) = \sum_{k=0}^{\infty} \frac{B_k}{k!} \mathrm{ad}_{\boldsymbol{\Omega}}^k(\boldsymbol{A}), \tag{10.3}$$

with Bernoulli numbers B_k*,* $\boldsymbol{A} \in \mathfrak{g}$*, and adjoint operator* $\mathrm{ad}_{\boldsymbol{\Omega}}(\boldsymbol{A}) = [\boldsymbol{\Omega}, \boldsymbol{A}] = \boldsymbol{\Omega}\boldsymbol{A} - \boldsymbol{A}\boldsymbol{\Omega}$*, is guaranteed.*

Proof. See [61, Theorem IV.7.1]. □

We may use the Magnus approach to derive numerical integration schemes that yield numerical approximations $\boldsymbol{U}_n$ situated on the Lie group manifold. Given a numerical approximation $\boldsymbol{U}_n \in \mathcal{G}$ at time point t_n, we can compute one time step of a conventional numerical integration scheme (such as a Runge–Kutta method) with step size h of the auxiliary ODE-IVP (10.2), resulting in a numerical approximation $\boldsymbol{\Omega}_1 \approx \boldsymbol{\Omega}(h) \in \mathfrak{g}$. As the Lie algebra $\mathfrak{g}$ is a linear space, the use of conventional numerical integration schemes yields numerical approximations in $\mathfrak{g}$. Then, the numerical approximation $\boldsymbol{U}_{n+1} = \exp(\boldsymbol{\Omega}_1)\boldsymbol{U}_n \in \mathcal{G}$ constitutes a numerical approximation at time point $t_{n+1} = t_n + h$. The matrix exponential $\exp\colon \mathfrak{g} \to \mathcal{G}$ defines a mapping of the Lie algebra to the Lie group. Since a matrix Lie group $\mathcal{G}$ is closed with respect to matrix multiplication, the numerical approximation $\boldsymbol{U}_{n+1}$ is situated on the Lie group manifold.

However, it is impractical to work with the infinite series (10.3). Instead, we work with the truncated series

$$\texttt{dexpinv}(\boldsymbol{\Omega}, \boldsymbol{A}, q) = \sum_{k=0}^{q} \frac{B_k}{k!} \mathrm{ad}_{\boldsymbol{\Omega}}^k(\boldsymbol{A}) \tag{10.4}$$

with truncation index q. If the numerical integration scheme applied to the auxiliary ODE-IVP

$$\dot{\boldsymbol{\Omega}} = \texttt{dexpinv}(\boldsymbol{\Omega}, -\boldsymbol{P}(t), q), \quad \boldsymbol{\Omega}(0) = \boldsymbol{0},$$

has order p, then a truncation index of $q \geq \min\{0, p-2\}$ is sufficient to obtain a method for the ODE of Lie-type (10.1) of order p [98].

In the spirit of the local coordinates approach [61, Algorithm IV.5.3], we may replace the matrix exponential with any *local parameterization* $\Lambda\colon \mathfrak{g} \to \mathcal{G}$, $\boldsymbol{A} \mapsto \Lambda(\boldsymbol{A})$, that is a local diffeomorphism between the Lie group $\mathcal{G}$ and the corresponding Lie algebra $\mathfrak{g}$ near $\boldsymbol{A} = \boldsymbol{0}$.

Suppose $\mathcal{G}$ is a quadratic matrix Lie group of the form

$$\mathcal{G} = \{\boldsymbol{Y} : \boldsymbol{Y}^\dagger \boldsymbol{E} \boldsymbol{Y} = \boldsymbol{E}\}, \tag{10.5}$$

where $\boldsymbol{E}$ is a given constant matrix. The corresponding Lie algebra is given by

$$\mathfrak{g} = \{\boldsymbol{\Omega} : \boldsymbol{E}\boldsymbol{\Omega} + \boldsymbol{\Omega}^\dagger \boldsymbol{E} = \boldsymbol{0}\}.$$

Prominent examples of quadratic Lie groups are the orthogonal group $\mathrm{O}(n)$ and the unitary group $\mathrm{U}(n)$. For quadratic Lie groups of the prescribed form, the *Cayley transform* [54]

$$\operatorname{cay}(\boldsymbol{\Omega}) = (\boldsymbol{I} - \boldsymbol{\Omega})^{-1}(\boldsymbol{I} + \boldsymbol{\Omega}), \quad \boldsymbol{\Omega} \in \mathfrak{g}, \tag{10.6}$$

provides an alternative local parameterization $\operatorname{cay}\colon \mathfrak{g} \to \mathcal{G}$. In contrast to the matrix exponential, the inverse of the derivative of the Cayley transform is not an infinite series. Specifically, we obtain [61, Lemma IV.8.8]

$$\mathrm{d}\operatorname{cay}_{\boldsymbol{\Omega}}^{-1}(\boldsymbol{A}) = \frac{1}{2}(\boldsymbol{I} - \boldsymbol{\Omega})\boldsymbol{A}(\boldsymbol{I} + \boldsymbol{\Omega}). \tag{10.7}$$

Consequently, no truncation of a series is needed. Moreover, an evaluation of the Cayley transform may be computationally less expensive than an evaluation of the matrix exponential. There exist several approaches to define Lie group methods that are based on the Cayley transform (10.6), see for example [34, 35, 81, 69]. The computational efficiency of Cayley-based methods motivates the derivation of such integrator variants for SU(3) and in particular gauge field generation in lattice QCD.

10.1. The Cayley Transform

The special orthogonal group $\mathrm{SO}(n)$ and the special unitary group $\mathrm{SU}(n)$ are not quadratic Lie groups of the form (10.5). However, it can be shown that the Cayley transform (10.6) defines a suitable local parameterization for $\mathrm{SO}(n)$. Since the eigenvalues of any skew-symmetric matrix $\boldsymbol{\Omega} \in \mathfrak{so}(n)$ are purely imaginary and complex eigenvalues of real-valued matrices occur in complex conjugate pairs, it can be shown that

$$\det(\operatorname{cay}(\boldsymbol{\Omega})) = 1, \quad \forall\, \boldsymbol{\Omega} \in \mathfrak{so}(n),$$

i.e., $\operatorname{cay}(\boldsymbol{\Omega}) \in \mathrm{SO}(n)$ for all $\boldsymbol{\Omega} \in \mathfrak{so}(n)$. In general, the eigenvalues of complex-valued matrices do not necessarily occur in complex conjugate pairs. Elements of the Lie algebra $\mathfrak{su}(2)$ corresponding to the Lie group $\mathrm{SU}(2)$ are a special case in which the eigenvalues occur in complex conjugate pairs. The Lie algebra $\mathfrak{su}(2)$ is three-dimensional and has a basis consisting of generators $\mathrm{i}\boldsymbol{\sigma}_j$, where $\boldsymbol{\sigma}_j$ are the Pauli matrices

$$\boldsymbol{\sigma}_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \boldsymbol{\sigma}_2 = \begin{pmatrix} 0 & -\mathrm{i} \\ \mathrm{i} & 0 \end{pmatrix}, \quad \boldsymbol{\sigma}_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{10.8}$$

For every basis element $\mathrm{i}\boldsymbol{\sigma}_j$, a direct computation shows that

$$\operatorname{cay}(\mathrm{i}\boldsymbol{\sigma}_j)^\dagger \operatorname{cay}(\mathrm{i}\boldsymbol{\sigma}_j) = \boldsymbol{I} \quad \text{and} \quad \det(\operatorname{cay}(\mathrm{i}\boldsymbol{\sigma}_j)) = 1.$$

Consequently, the Cayley transform (10.6) defines a suitable local parameterization of SU(2). In [142], a Cayley-based Strang splitting has been utilized in the HMC algorithm to perform simulations in an SU(2) Yang–Mills theory, demonstrating that the use of the Cayley transform instead of the matrix exponential can increase the computational efficiency of the HMC algorithm. However, the Cayley transform does not define a mapping from $\mathfrak{su}(3)$ to SU(3).

Lemma 10.2. *The Cayley transform* (10.6) *maps elements from the Lie algebra* $\mathfrak{su}(3)$ *into the unitary group* U(3)*, but not necessarily into the special unitary group* SU(3)*.*

Proof. Since the Lie algebra $\mathfrak{su}(3)$ is a sub-algebra of $\mathfrak{u}(3)$ and U(3) is a quadratic Lie group of the form (10.5), the Cayley transform maps elements from $\mathfrak{su}(3)$ into U(3). It remains to show that the Cayley transform does not necessarily map into the sub-group SU(3). Any Lie algebra element $\boldsymbol{\Omega} \in \mathfrak{su}(3)$ can be expressed as $\boldsymbol{\Omega} = \sum_{j=1}^{8} c_j \boldsymbol{T}_j$ with coefficients $c_j \in \mathbb{R}$ and the generators $\boldsymbol{T}_j = \frac{\mathrm{i}}{\sqrt{2}}\boldsymbol{\lambda}_j$ defined by the Gell-Mann matrices (9.21). For $\mathrm{i}\boldsymbol{\lambda}_8 \in \mathfrak{su}(3)$, we obtain

$$\operatorname{cay}(\mathrm{i}\boldsymbol{\lambda}_8) = \operatorname{diag}\left(\tfrac{1}{2} + \mathrm{i}\tfrac{\sqrt{3}}{2} \quad \tfrac{1}{2} + \mathrm{i}\tfrac{\sqrt{3}}{2} \quad -\tfrac{1}{7} - \mathrm{i}\tfrac{4\sqrt{3}}{7}\right),$$

which is a unitary matrix with determinant $\det(\operatorname{cay}(\mathrm{i}\boldsymbol{\lambda}_8)) = \frac{13}{14} + \mathrm{i}\frac{3\sqrt{3}}{14} \neq 1$ and hence an element of the unitary group U(3), but not of the special unitary group SU(3). □

Consequently, applying the Cayley transform (10.6) in lattice QCD simulations would yield non-physical link matrices. In the subsequent section, we will derive a modification to the Cayley transform (10.6) that defines a local parameterization of SU(3). We will show in Section 10.3 that the modified Cayley transform not only resolves the determinant issue but also introduces a beneficial damping effect compared to the matrix exponential.

10.2. Derivation of the Modified Cayley Transform

Consider a matrix $\boldsymbol{B} \in \mathbb{C}^{3\times 3}$. Then, the Cayley–Hamilton theorem implies

$$\boldsymbol{B}^3 - \operatorname{tr}(\boldsymbol{B})\boldsymbol{B}^2 + \tfrac{1}{2}\big(\operatorname{tr}(\boldsymbol{B})^2 - \operatorname{tr}(\boldsymbol{B}^2)\big)\boldsymbol{B} - \det(\boldsymbol{B})\boldsymbol{I} = 0. \tag{10.9}$$

Taking the trace of this matrix expression yields

$$\begin{aligned}\det(\boldsymbol{B}) &= \tfrac{1}{3}\operatorname{tr}(\boldsymbol{B}^3) - \tfrac{1}{3}\operatorname{tr}(\boldsymbol{B})\operatorname{tr}(\boldsymbol{B}^2) + \tfrac{1}{6}\operatorname{tr}(\boldsymbol{B})^3 - \tfrac{1}{6}\operatorname{tr}(\boldsymbol{B})\operatorname{tr}(\boldsymbol{B}^2)\\ &= \tfrac{1}{3}\operatorname{tr}(\boldsymbol{B}^3) - \tfrac{1}{2}\operatorname{tr}(\boldsymbol{B})\operatorname{tr}(\boldsymbol{B}^2) + \tfrac{1}{6}\operatorname{tr}(\boldsymbol{B})^3.\end{aligned} \tag{10.10}$$

For $\boldsymbol{B} = \boldsymbol{I} + \boldsymbol{\Omega}$ with $\boldsymbol{\Omega}$ being traceless, (10.10) simplifies to

$$\begin{aligned}\det(\boldsymbol{I}+\boldsymbol{\Omega}) &= \tfrac{1}{3}\operatorname{tr}\left((\boldsymbol{I}+\boldsymbol{\Omega})^3\right) - \tfrac{1}{2}\operatorname{tr}(\boldsymbol{I}+\boldsymbol{\Omega})\operatorname{tr}\left((\boldsymbol{I}+\boldsymbol{\Omega})^2\right) + \tfrac{1}{6}\operatorname{tr}(\boldsymbol{I}+\boldsymbol{\Omega})^3\\ &= 1 - \tfrac{1}{2}\operatorname{tr}(\boldsymbol{\Omega}^2) + \tfrac{1}{3}\operatorname{tr}(\boldsymbol{\Omega}^3)\\ &= 1 - \tfrac{1}{2}\operatorname{tr}(\boldsymbol{\Omega}^2) + \det(\boldsymbol{\Omega}),\end{aligned} \tag{10.11}$$

where we applied (10.10) to the traceless matrix $\boldsymbol{\Omega}$ in the last step.

Lemma 10.3. *If* $\boldsymbol{\Omega} \in \mathbb{C}^{3\times 3}$ *is an anti-Hermitian matrix, then for any angle* $\theta \in (-\pi, \pi]$*, the matrix*

$$\boldsymbol{U}(\boldsymbol{\Omega}) = (\boldsymbol{I} - \mathrm{e}^{-\mathrm{i}\theta}\boldsymbol{\Omega})^{-1}(\boldsymbol{I} + \mathrm{e}^{\mathrm{i}\theta}\boldsymbol{\Omega}) \tag{10.12}$$

obeys $\boldsymbol{U}(\boldsymbol{\Omega})^\dagger \boldsymbol{U}(\boldsymbol{\Omega}) = \boldsymbol{I}$ *and thus is an element of the Lie group* U(3)*.*

Proof. By using the commutativity of the matrices and that $\boldsymbol{\Omega}^\dagger = -\boldsymbol{\Omega}$ holds, we obtain

$$\begin{aligned}\boldsymbol{U}(\boldsymbol{\Omega})^\dagger \boldsymbol{U}(\boldsymbol{\Omega}) &= (\boldsymbol{I} + \mathrm{e}^{\mathrm{i}\theta}\boldsymbol{\Omega})^\dagger ((\boldsymbol{I} - \mathrm{e}^{-\mathrm{i}\theta}\boldsymbol{\Omega})^{-1})^\dagger (\boldsymbol{I} - \mathrm{e}^{-\mathrm{i}\theta}\boldsymbol{\Omega})^{-1}(\boldsymbol{I} + \mathrm{e}^{\mathrm{i}\theta}\boldsymbol{\Omega}) \\ &= (\boldsymbol{I} - \mathrm{e}^{-\mathrm{i}\theta}\boldsymbol{\Omega})(\boldsymbol{I} + \mathrm{e}^{\mathrm{i}\theta}\boldsymbol{\Omega})^{-1}(\boldsymbol{I} - \mathrm{e}^{-\mathrm{i}\theta}\boldsymbol{\Omega})^{-1}(\boldsymbol{I} + \mathrm{e}^{\mathrm{i}\theta}\boldsymbol{\Omega}) \\ &= \boldsymbol{I}.\end{aligned}$$

□

Remark 10.4. For $\theta = 0$, (10.12) reduces to the conventional Cayley transform (10.6) since $\mathrm{e}^{-\mathrm{i}\theta}|_{\theta=0} = \mathrm{e}^{\mathrm{i}\theta}|_{\theta=0} = 1$.

We next determine the angle θ such that the transformation (10.12) maps elements $\boldsymbol{\Omega} \in \mathfrak{su}(3)$ into SU(3), i.e., such that $\det(\boldsymbol{U}(\boldsymbol{\Omega})) = 1$.

Lemma 10.5. Consider the matrix $\boldsymbol{U}$ given by (10.12) and $\boldsymbol{\Omega} \in \mathfrak{su}(3)$. It holds

$$\det(\boldsymbol{U}(\boldsymbol{\Omega})) = 1 \quad \Leftrightarrow \quad \det(\boldsymbol{I} + \mathrm{e}^{\mathrm{i}\theta}\boldsymbol{\Omega}) \in \mathbb{R}.$$

Proof. With $\det(\boldsymbol{U}(\boldsymbol{\Omega})) = \det(\boldsymbol{I} + \mathrm{e}^{\mathrm{i}\theta}\boldsymbol{\Omega})/\det(\boldsymbol{I} - \mathrm{e}^{-\mathrm{i}\theta}\boldsymbol{\Omega})$, it holds

$$\det(\boldsymbol{U}(\boldsymbol{\Omega})) = 1 \quad \Leftrightarrow \quad \det(\boldsymbol{I} + \mathrm{e}^{\mathrm{i}\theta}\boldsymbol{\Omega}) = \det(\boldsymbol{I} - \mathrm{e}^{-\mathrm{i}\theta}\boldsymbol{\Omega}).$$

By applying (10.11), we obtain

$$\begin{aligned}& 1 + \det(\mathrm{e}^{\mathrm{i}\theta}\boldsymbol{\Omega}) - \tfrac{1}{2}\operatorname{tr}(\mathrm{e}^{2\mathrm{i}\theta}\boldsymbol{\Omega}^2) = 1 + \det(-\mathrm{e}^{-\mathrm{i}\theta}\boldsymbol{\Omega}) - \tfrac{1}{2}\operatorname{tr}(\mathrm{e}^{-2\mathrm{i}\theta}\boldsymbol{\Omega}^2) \\ \Leftrightarrow \quad & \mathrm{e}^{3\mathrm{i}\theta}\det(\boldsymbol{\Omega}) - \frac{\mathrm{e}^{2\mathrm{i}\theta}}{2}\operatorname{tr}(\boldsymbol{\Omega}^2) = -\mathrm{e}^{-3\mathrm{i}\theta}\det(\boldsymbol{\Omega}) - \frac{\mathrm{e}^{-2\mathrm{i}\theta}}{2}\operatorname{tr}(\boldsymbol{\Omega}^2).\end{aligned}$$

Applying Euler's formula then yields

$$\cos(3\theta)\det(\boldsymbol{\Omega}) - \tfrac{1}{2}\mathrm{i}\sin(2\theta)\operatorname{tr}(\boldsymbol{\Omega}^2) = -\cos(3\theta)\det(\boldsymbol{\Omega}) + \tfrac{1}{2}\mathrm{i}\sin(2\theta)\operatorname{tr}(\boldsymbol{\Omega}^2).$$

Since the determinant of $\boldsymbol{\Omega} \in \mathfrak{su}(3)$ is purely imaginary and $\operatorname{tr}(\boldsymbol{\Omega}^2) \in \mathbb{R}$, we can rewrite the expression and obtain

$$\begin{aligned}& \mathrm{i}\cos(3\theta)\mathfrak{Im}(\det(\boldsymbol{\Omega})) - \tfrac{1}{2}\mathrm{i}\sin(2\theta)\operatorname{tr}(\boldsymbol{\Omega}^2) = -\mathrm{i}\cos(3\theta)\mathfrak{Im}(\det(\boldsymbol{\Omega})) + \tfrac{1}{2}\mathrm{i}\sin(2\theta)\operatorname{tr}(\boldsymbol{\Omega}^2) \\ \Leftrightarrow \quad & \cos(3\theta)\mathfrak{Im}(\det(\boldsymbol{\Omega})) - \tfrac{1}{2}\sin(2\theta)\operatorname{tr}(\boldsymbol{\Omega}^2) = -\cos(3\theta)\mathfrak{Im}(\det(\boldsymbol{\Omega})) + \tfrac{1}{2}\sin(2\theta)\operatorname{tr}(\boldsymbol{\Omega}^2).\end{aligned}$$

The left-hand side and the right-hand side coincide with the imaginary part of $\det(\boldsymbol{I}+\mathrm{e}^{\mathrm{i}\theta}\boldsymbol{\Omega})$ and $\det(\boldsymbol{I} - \mathrm{e}^{-\mathrm{i}\theta}\boldsymbol{\Omega})$, respectively. Since $\mathfrak{Im}(\det(\boldsymbol{I} + \mathrm{e}^{\mathrm{i}\theta}\boldsymbol{\Omega})) = -\mathfrak{Im}(\det(\boldsymbol{I} - \mathrm{e}^{-\mathrm{i}\theta}\boldsymbol{\Omega}))$, the only solution is given by

$$\mathfrak{Im}(\det(\boldsymbol{I} + \mathrm{e}^{\mathrm{i}\theta}\boldsymbol{\Omega})) = 0 = \mathfrak{Im}(\det(\boldsymbol{I} - \mathrm{e}^{-\mathrm{i}\theta}\boldsymbol{\Omega})),$$

which concludes the proof. □

Theorem 10.6. *The angle*

$$\theta(\boldsymbol{\Omega}) := \sin^{-1}\left(-\frac{1}{2}\left(\gamma(\boldsymbol{\Omega})^{-1} \mp \sqrt{\gamma(\boldsymbol{\Omega})^{-2} + 1}\right)\right),$$

with $\gamma(\boldsymbol{\Omega}) := 4\mathfrak{Im}(\det(\boldsymbol{\Omega}))/\operatorname{tr}(\boldsymbol{\Omega}^2)$, *ensures* $\det(\boldsymbol{U}(\boldsymbol{\Omega})) = 1$.

Proof. Applying (10.11) and Euler's formula yields

$$\begin{aligned}\det(\boldsymbol{I} + \mathrm{e}^{\mathrm{i}\theta}\boldsymbol{\Omega}) &= 1 + \mathrm{e}^{3\mathrm{i}\theta}\det(\boldsymbol{\Omega}) - \tfrac{1}{2}\mathrm{e}^{2\mathrm{i}\theta}\operatorname{tr}(\boldsymbol{\Omega}^2) \\ &= 1 + (\cos(3\theta) + \mathrm{i}\sin(3\theta))\det(\boldsymbol{\Omega}) - \tfrac{1}{2}(\cos(2\theta) + \mathrm{i}\sin(2\theta))\operatorname{tr}(\boldsymbol{\Omega}^2).\end{aligned}$$

According to Lemma 10.5, we demand $\mathfrak{Im}(\det(\boldsymbol{I} + \mathrm{e}^{\mathrm{i}\theta}\boldsymbol{\Omega})) = 0$, i.e.,

$$0 = \cos(3\theta)\mathfrak{Im}(\det(\boldsymbol{\Omega})) - \tfrac{1}{2}\sin(2\theta)\operatorname{tr}(\boldsymbol{\Omega}^2) \;\Leftrightarrow\; \tfrac{1}{2}\sin(2\theta)\operatorname{tr}(\boldsymbol{\Omega}^2) = \cos(3\theta)\mathfrak{Im}(\det(\boldsymbol{\Omega})).$$

Using the double- and triple-angle formula results in an equivalent expression given by

$$\sin(\theta)\cos(\theta)\operatorname{tr}(\boldsymbol{\Omega}^2) = \cos(\theta)\left(1 - 4\sin^2(\theta)\right)\mathfrak{Im}(\det(\boldsymbol{\Omega})).$$

This gives the trivial result $\cos(\theta) = 0$, for which $\boldsymbol{U} = \boldsymbol{I} \in \mathrm{SU}(3)$, as well as a non-trivial constraint

$$\sin(\theta)\operatorname{tr}(\boldsymbol{\Omega}^2) = \left(1 - 4\sin^2(\theta)\right)\mathfrak{Im}(\det(\boldsymbol{\Omega})).$$

For $\mathfrak{Im}(\det(\boldsymbol{\Omega})) = 0$, $\sin(\theta) = 0$ solves this equation. For $\mathfrak{Im}(\det(\boldsymbol{\Omega})) \neq 0$, we obtain

$$\sin(\theta)\operatorname{tr}(\boldsymbol{\Omega}^2) = (1 - 4\sin^2(\theta))\mathfrak{Im}(\det(\boldsymbol{\Omega})) \;\Leftrightarrow\; \sin^2(\theta) + \frac{\operatorname{tr}(\boldsymbol{\Omega}^2)}{4\mathfrak{Im}(\det(\boldsymbol{\Omega}))}\sin(\theta) - \frac{1}{4} = 0.$$

Solving this equation for $\sin(\theta)$ yields

$$\sin(\theta) = -\frac{1}{2}\left(\frac{\operatorname{tr}(\boldsymbol{\Omega}^2)}{4\mathfrak{Im}(\det(\boldsymbol{\Omega}))} \mp \sqrt{\left(\frac{\operatorname{tr}(\boldsymbol{\Omega}^2)}{4\mathfrak{Im}(\det(\boldsymbol{\Omega}))}\right)^2 + 1}\right).$$

By introducing the parameter $\gamma(\boldsymbol{\Omega}) \coloneqq 4\mathfrak{Im}(\det(\boldsymbol{\Omega}))/\operatorname{tr}(\boldsymbol{\Omega}^2)$, the angle θ is given by

$$\theta(\boldsymbol{\Omega}) = \sin^{-1}\left(-\frac{1}{2}\left(\gamma(\boldsymbol{\Omega})^{-1} \mp \sqrt{\gamma(\boldsymbol{\Omega})^{-2} + 1}\right)\right). \qquad \square$$

Due to the quadratic equation for $\sin(\theta)$, there are two solutions in general. They are depicted in Fig. 10.1. One branch (shown in orange) diverges at $\gamma = 0$ and for small values does not give a real phase (denoted by the black dashed lines). If we exclude this branch, the phase is single-valued, differentiable for all γ, and bounded such that $\sin(\theta) \in (-1/2, 1/2)$ (shown in blue). The functional form of the remaining branch is given by

$$\sin(\theta) = \begin{cases} 0, & \gamma(\boldsymbol{\Omega}) = 0, \\ -\frac{1}{2}\left(\gamma(\boldsymbol{\Omega})^{-1} - \sqrt{\gamma(\boldsymbol{\Omega})^{-2} + 1}\right), & \gamma(\boldsymbol{\Omega}) > 0, \\ -\frac{1}{2}\left(\gamma(\boldsymbol{\Omega})^{-1} + \sqrt{\gamma(\boldsymbol{\Omega})^{-2} + 1}\right), & \gamma(\boldsymbol{\Omega}) < 0. \end{cases}$$

By applying $\sin^{-1}$ on both sides and choosing the branch $\theta \in (-\pi/6, \pi/6)$, we have $\boldsymbol{U}(-\boldsymbol{\Omega}) = \boldsymbol{U}(\boldsymbol{\Omega})^\dagger$ as the following lemma shows.

Lemma 10.7. *Taking the angle $\theta \in (-\pi/6, \pi/6)$ ensures $\theta(-\boldsymbol{\Omega}) = -\theta(\boldsymbol{\Omega})$, which in turn ensures that the crucial property $\boldsymbol{U}(-\boldsymbol{\Omega}) = \boldsymbol{U}(\boldsymbol{\Omega})^\dagger$ holds.*

Proof. Without loss of generality, assume $\gamma(\boldsymbol{\Omega}) > 0$. Then, $\gamma(-\boldsymbol{\Omega}) = -\gamma(\boldsymbol{\Omega}) < 0$ and thus

$$\begin{aligned} \theta(-\boldsymbol{\Omega}) &= \sin^{-1}\left(-\frac{1}{2}\left(-\gamma(\boldsymbol{\Omega})^{-1} + \sqrt{\gamma(\boldsymbol{\Omega})^{-2} + 1}\right)\right) \\ &= \sin^{-1}\left(\frac{1}{2}\left(\gamma(\boldsymbol{\Omega})^{-1} - \sqrt{\gamma(\boldsymbol{\Omega})^{-2} + 1}\right)\right) \\ &= -\sin^{-1}\left(-\frac{1}{2}\left(\gamma(\boldsymbol{\Omega})^{-1} - \sqrt{\gamma(\boldsymbol{\Omega})^{-2} + 1}\right)\right) = -\theta(\boldsymbol{\Omega}). \end{aligned}$$

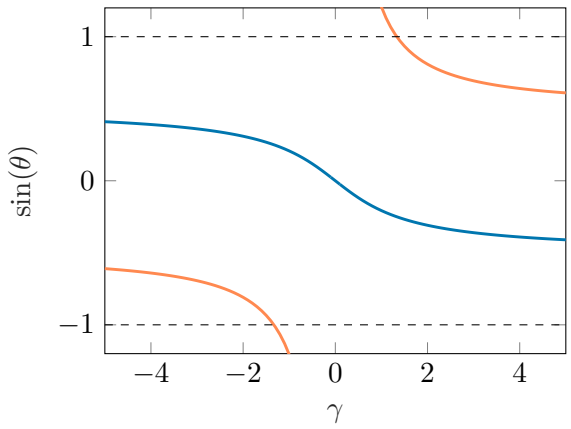


Fig. 10.1. The modified Cayley phase. The two branches of $\sin(\theta)$ as a function of γ. The diverging branch (orange) is excluded; the selected branch (blue) is bounded with $\sin(\theta) \in (-1/2, 1/2)$.

Consequently, it holds

$$\boldsymbol{U}(-\boldsymbol{\Omega}) = (\boldsymbol{I} + \mathrm{e}^{\mathrm{i}\theta(\boldsymbol{\Omega})}\boldsymbol{\Omega})^{-1}(\boldsymbol{I} - \mathrm{e}^{-\mathrm{i}\theta(\boldsymbol{\Omega})}\boldsymbol{\Omega}),$$

which is identical to

$$\boldsymbol{U}(\boldsymbol{\Omega})^{\dagger} = (\boldsymbol{I} - \mathrm{e}^{-\mathrm{i}\theta(\boldsymbol{\Omega})}\boldsymbol{\Omega})(\boldsymbol{I} + \mathrm{e}^{\mathrm{i}\theta(\boldsymbol{\Omega})}\boldsymbol{\Omega})^{-1},$$

due to the commutativity of the matrices. □

We are now able to define the modified Cayley transform for SU(3).

Definition 10.8 (Modified Cayley transform for SU(3))**.** Let $\boldsymbol{\Omega} \in \mathfrak{su}(3)$. With

$$\theta(\boldsymbol{\Omega}) \coloneqq \begin{cases} 0, & \gamma(\boldsymbol{\Omega}) = 0, \\ \sin^{-1}\left(-\frac{1}{2}\left(\gamma(\boldsymbol{\Omega})^{-1} - \sqrt{\gamma(\boldsymbol{\Omega})^{-2}+1}\right)\right), & \gamma(\boldsymbol{\Omega}) > 0, \\ \sin^{-1}\left(-\frac{1}{2}\left(\gamma(\boldsymbol{\Omega})^{-1} + \sqrt{\gamma(\boldsymbol{\Omega})^{-2}+1}\right)\right), & \gamma(\boldsymbol{\Omega}) < 0, \end{cases} \tag{10.13}$$

and $\gamma(\boldsymbol{\Omega}) \coloneqq 4\mathfrak{Im}(\det(\boldsymbol{\Omega}))/\operatorname{tr}(\boldsymbol{\Omega}^2)$ so that $\theta(\boldsymbol{\Omega}) \in (-\pi/6, \pi/6)$, the modified Cayley transform

$$\widetilde{\mathrm{cay}}(\boldsymbol{\Omega}) \coloneqq \left(\boldsymbol{I} - \mathrm{e}^{-\mathrm{i}\theta(\boldsymbol{\Omega})}\boldsymbol{\Omega}\right)^{-1}\left(\boldsymbol{I} + \mathrm{e}^{\mathrm{i}\theta(\boldsymbol{\Omega})}\boldsymbol{\Omega}\right), \tag{10.14}$$

maps elements from $\mathfrak{su}(3)$ into SU(3). Moreover, it is a local diffeomorphism near $\boldsymbol{\Omega} = 0$.

The modified Cayley transform (10.14) provides an alternative local parameterization for numerical integration on the special unitary group SU(3).

Remark 10.9 (Implementation of the modified Cayley transform)**.** In practical computations, one can considerably simplify the computation of $\mathrm{e}^{\mathrm{i}\theta} = \cos(\theta) + \mathrm{i}\sin(\theta)$ and $-\mathrm{e}^{-\mathrm{i}\theta} = -\cos(\theta) + \mathrm{i}\sin(\theta)$ via

$$\sin(\theta) = -\frac{1}{2}\left(\gamma(\boldsymbol{\Omega})^{-1} \mp \sqrt{\gamma(\boldsymbol{\Omega})^{-2}+1}\right), \qquad \cos(\theta) = \sqrt{1-\sin(\theta)^2}.$$

The inverse of $\boldsymbol{C} \coloneqq (\boldsymbol{I} - \mathrm{e}^{-\mathrm{i}\theta(\boldsymbol{\Omega})}\boldsymbol{\Omega})$ can be computed via $\boldsymbol{C}^{-1} = \operatorname{adj}(\boldsymbol{C})/\det(\boldsymbol{C})$.

Remark 10.10. Practical implementations for lattice QCD simulations frequently work with traceless and Hermitian momenta $\tilde{\boldsymbol{P}}_\mu(\boldsymbol{n}) \coloneqq -\mathrm{i}\boldsymbol{P}_\mu(\boldsymbol{n})$ rather than the Lie algebra elements $\boldsymbol{P}_\mu(\boldsymbol{n})$. Let $\tilde{\boldsymbol{P}}_\mu(\boldsymbol{n}) \in \mathbb{C}^{3\times 3}$ be traceless and Hermitian. With

$$\tilde{\theta}(\tilde{\boldsymbol{P}}_\mu(\boldsymbol{n})) = \begin{cases} \pi/2, & \tilde{\gamma}(\tilde{\boldsymbol{P}}_\mu(\boldsymbol{n})) = 0, \\ \cos^{-1}\left(\frac{1}{2}\left(\tilde{\gamma}(\tilde{\boldsymbol{P}}_\mu(\boldsymbol{n}))^{-1} - \sqrt{\tilde{\gamma}(\tilde{\boldsymbol{P}}_\mu(\boldsymbol{n}))^{-2}+1}\right)\right), & \tilde{\gamma}(\tilde{\boldsymbol{P}}_\mu(\boldsymbol{n})) > 0, \\ \cos^{-1}\left(\frac{1}{2}\left(\tilde{\gamma}(\tilde{\boldsymbol{P}}_\mu(\boldsymbol{n}))^{-1} + \sqrt{\tilde{\gamma}(\tilde{\boldsymbol{P}}_\mu(\boldsymbol{n}))^{-2}+1}\right)\right), & \tilde{\gamma}(\tilde{\boldsymbol{P}}_\mu(\boldsymbol{n})) < 0, \end{cases}$$

and $\tilde{\gamma}(\tilde{\boldsymbol{P}}_\mu(\boldsymbol{n})) = 4\det(\tilde{\boldsymbol{P}}_\mu(\boldsymbol{n}))/\operatorname{tr}(\tilde{\boldsymbol{P}}_\mu(\boldsymbol{n})^2)$ so that $\tilde{\theta}(\tilde{\boldsymbol{P}}_\mu(\boldsymbol{n})) \in (\pi/3, 2\pi/3)$, the mapping

$$\left(\boldsymbol{I} + \mathrm{e}^{-\mathrm{i}\tilde{\theta}(\tilde{\boldsymbol{P}}_\mu(\boldsymbol{n}))}\tilde{\boldsymbol{P}}_\mu(\boldsymbol{n})\right)^{-1}\left(\boldsymbol{I} + \mathrm{e}^{\mathrm{i}\tilde{\theta}(\tilde{\boldsymbol{P}}_\mu(\boldsymbol{n}))}\tilde{\boldsymbol{P}}_\mu(\boldsymbol{n})\right)$$

maps $\tilde{\boldsymbol{P}}_\mu(\boldsymbol{n})$ into the special unitary group SU(3).

Next, we will investigate the use of the modified Cayley transform inside splitting methods and (Hessian-free) force-gradient integrators to numerically solve the Hamiltonian equations of motion (9.25) in the molecular dynamics step of the HMC algorithm.

10.3. Approximation of the Link Updates

We consider the splitting (9.39) of the Hamiltonian equations of motion and the exact flows (9.40) of the individual subsystems. The momentum update (9.40b) has the same structure as applying one step of the explicit Euler method with step size t. Similarly, the link update (9.40a) is identical to computing one time step of the *Lie–Euler method*, the Runge–Kutta Munthe-Kaas method [98] that is based on the explicit Euler method.

Example 10.11 (Lie–Euler method for SU(3))**.** Consider the ODE of Lie-type (10.1) with $\boldsymbol{P}(t) \in \mathfrak{su}(3)$ for all t, and $\boldsymbol{U}_0 \in \mathrm{SU}(3)$. Let $\boldsymbol{A}, \boldsymbol{\Omega} \in \mathfrak{su}(3)$ and $\Lambda\colon \mathfrak{su}(3) \to \mathrm{SU}(3)$ denote a local parameterization. The differential of the local parameterization Λ reads

$$\left(\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\Omega}}\Lambda(\boldsymbol{\Omega})\right)\boldsymbol{A} = \mathrm{d}\,\Lambda_{\boldsymbol{\Omega}}(\boldsymbol{A})\Lambda(\boldsymbol{\Omega}),$$

where $\mathrm{d}\,\Lambda_{\boldsymbol{\Omega}}\colon \mathfrak{su}(3) \to \mathfrak{su}(3)$. Then, a numerical approximation $\boldsymbol{U}_1 \approx \boldsymbol{U}(h)$ is obtained by

$$\boldsymbol{U}_1 = \Lambda\left(h\mathrm{d}\,\Lambda_{\boldsymbol{\Omega}}^{-1}(-\boldsymbol{P}_0)\big|_{\boldsymbol{\Omega}=\boldsymbol{0}}\right)\boldsymbol{U}_0, \tag{10.16}$$

where $\boldsymbol{P}_0 = \boldsymbol{P}(0)$ and $\mathrm{d}\,\Lambda_{\boldsymbol{\Omega}}^{-1}$ denotes the inverse of the operator $\mathrm{d}\,\Lambda_{\boldsymbol{\Omega}}$. For $\Lambda = \exp$, the update becomes

$$\boldsymbol{U}_1 = \exp\left(-h\boldsymbol{P}_0\right)\boldsymbol{U}_0.$$

Using another local parameterization in (10.16) yields a consistent approximation to the exact flow (9.40a) and, depending on the particular parameterization, may result in a more efficient computational process. In [142], this idea has been applied successfully for SU(2) using the Cayley transform (10.6), resulting in a more efficient computational process for the Strang splitting (3.5). We can adapt this idea by utilizing the modified Cayley transform (10.14) and obtain Cayley-based splitting methods and (Hessian-free) force-gradient integrators for SU(3). Replacing the matrix exponential in the exact flow (9.40a) by the modified Cayley transform (10.14) results in a consistent numerical approximation to the exact flow, as it can be regarded as the application of a Lie–Euler step (10.16) with $\Lambda = \widetilde{\mathrm{cay}}$. For this purpose, we need to compute $\mathrm{d}\,\widetilde{\mathrm{cay}}_{\boldsymbol{\Omega}}^{-1}(\boldsymbol{A})$ at $\boldsymbol{\Omega} = \boldsymbol{0}$.

Theorem 10.12. *The differential of* $\widetilde{\operatorname{cay}}(\boldsymbol{\Omega})$ *is given as*

$$\left(\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\Omega}}\,\widetilde{\operatorname{cay}}(\boldsymbol{\Omega})\right)\boldsymbol{A}\Big|_{\boldsymbol{\Omega}=\mathbf{0}} = \left(\mathrm{d}\,\widetilde{\operatorname{cay}}_{\boldsymbol{\Omega}}(\boldsymbol{A})\right)\widetilde{\operatorname{cay}}(\boldsymbol{\Omega}).$$

The operator $\mathrm{d}\,\widetilde{\operatorname{cay}}_{\boldsymbol{\Omega}}$ *at* $\boldsymbol{\Omega}=\mathbf{0}$ *is given as*

$$\mathrm{d}\,\widetilde{\operatorname{cay}}_{\boldsymbol{\Omega}=\mathbf{0}}(\boldsymbol{A}) = 2\boldsymbol{A},$$

and consequently its inverse reads

$$\mathrm{d}\,\widetilde{\operatorname{cay}}^{-1}_{\boldsymbol{\Omega}=\mathbf{0}}(\boldsymbol{A}) = \tfrac{1}{2}\boldsymbol{A}.$$

Proof. Applying the Leibniz rule results in

$$\begin{aligned}\left(\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\Omega}}\,\widetilde{\operatorname{cay}}(\boldsymbol{\Omega})\right)\boldsymbol{A} = \left(\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\Omega}}(\boldsymbol{I}-\mathrm{e}^{-\mathrm{i}\theta(\boldsymbol{\Omega})}\boldsymbol{\Omega})^{-1}\right)\boldsymbol{A}(\boldsymbol{I}+\mathrm{e}^{\mathrm{i}\theta(\boldsymbol{\Omega})}\boldsymbol{\Omega})\\ +(\boldsymbol{I}-\mathrm{e}^{-\mathrm{i}\theta(\boldsymbol{\Omega})}\boldsymbol{\Omega})^{-1}\left(\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\Omega}}(\boldsymbol{I}+\mathrm{e}^{\mathrm{i}\theta(\boldsymbol{\Omega})}\boldsymbol{\Omega})\right)\boldsymbol{A}. \quad (10.17)\end{aligned}$$

We perform an auxiliary computation by considering the equation

$$\left(\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\Omega}}(\boldsymbol{I}-\mathrm{e}^{-\mathrm{i}\theta(\boldsymbol{\Omega})}\boldsymbol{\Omega})^{-1}(\boldsymbol{I}-\mathrm{e}^{-\mathrm{i}\theta(\boldsymbol{\Omega})}\boldsymbol{\Omega})\right)\boldsymbol{A} = 0.$$

Applying the Leibniz rule to this equation, followed by a simple algebraic manipulation, results in

$$\begin{aligned}\left(\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\Omega}}(\boldsymbol{I}-\mathrm{e}^{-\mathrm{i}\theta(\boldsymbol{\Omega})}\boldsymbol{\Omega})^{-1}\right)\boldsymbol{A} = -(\boldsymbol{I}-\mathrm{e}^{-\mathrm{i}\theta(\boldsymbol{\Omega})}\boldsymbol{\Omega})^{-1}\left(\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\Omega}}(\boldsymbol{I}-\mathrm{e}^{-\mathrm{i}\theta(\boldsymbol{\Omega})}\boldsymbol{\Omega})\right)\boldsymbol{A}\\ \cdot(\boldsymbol{I}-\mathrm{e}^{-\mathrm{i}\theta(\boldsymbol{\Omega})}\boldsymbol{\Omega})^{-1}. \quad (10.18)\end{aligned}$$

Inserting (10.18) into (10.17) yields

$$\begin{aligned}\left(\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\Omega}}\,\widetilde{\operatorname{cay}}(\boldsymbol{\Omega})\right)\boldsymbol{A} = (\boldsymbol{I}-\mathrm{e}^{-\mathrm{i}\theta(\boldsymbol{\Omega})}\boldsymbol{\Omega})^{-1}\Bigg[-\left(\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\Omega}}(\boldsymbol{I}-\mathrm{e}^{-\mathrm{i}\theta(\boldsymbol{\Omega})}\boldsymbol{\Omega})\right)\boldsymbol{A}\\ +\left(\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\Omega}}(\boldsymbol{I}+\mathrm{e}^{\mathrm{i}\theta(\boldsymbol{\Omega})}\boldsymbol{\Omega})\right)\boldsymbol{A}\,\widetilde{\operatorname{cay}}(\boldsymbol{\Omega})^{-1}\Bigg]\widetilde{\operatorname{cay}}(\boldsymbol{\Omega}),\end{aligned}$$

so that

$$\begin{aligned}\mathrm{d}\,\widetilde{\operatorname{cay}}_{\boldsymbol{\Omega}}(\boldsymbol{A}) = (\boldsymbol{I}-\mathrm{e}^{-\mathrm{i}\theta(\boldsymbol{\Omega})}\boldsymbol{\Omega})^{-1}\Bigg[\left(\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\Omega}}(\boldsymbol{I}-\mathrm{e}^{-\mathrm{i}\theta(\boldsymbol{\Omega})}\boldsymbol{\Omega})\right)\boldsymbol{A}\\ +\left(\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\Omega}}(\boldsymbol{I}+\mathrm{e}^{\mathrm{i}\theta(\boldsymbol{\Omega})}\boldsymbol{\Omega})\right)\boldsymbol{A}\,\widetilde{\operatorname{cay}}(\boldsymbol{\Omega})^{-1}\Bigg].\end{aligned}$$

By applying the chain rule and inserting $\boldsymbol{\Omega}=\mathbf{0}$, we immediately get

$$-\left(\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\Omega}}(\boldsymbol{I}-\mathrm{e}^{-\mathrm{i}\theta(\boldsymbol{\Omega})}\boldsymbol{\Omega})\right)\boldsymbol{A}\Big|_{\boldsymbol{\Omega}=\mathbf{0}} = \left(\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\Omega}}(\boldsymbol{I}+\mathrm{e}^{\mathrm{i}\theta(\boldsymbol{\Omega})}\boldsymbol{\Omega})\right)\boldsymbol{A}\Big|_{\boldsymbol{\Omega}=\mathbf{0}} = \boldsymbol{A},$$

and with $\widetilde{\operatorname{cay}}(\mathbf{0}) = \boldsymbol{I}$ finally

$$\mathrm{d}\,\widetilde{\operatorname{cay}}_{\boldsymbol{\Omega}=\mathbf{0}}(\boldsymbol{A}) = 2\boldsymbol{A}.$$

Hence, its inverse is given by

$$\mathrm{d}\,\widetilde{\operatorname{cay}}^{-1}_{\boldsymbol{\Omega}=\mathbf{0}}(\boldsymbol{A}) = \tfrac{1}{2}\boldsymbol{A},$$

proving the statements of the theorem. □

Remark 10.13. It holds $\mathrm{d}\,\mathrm{cay}^{-1}_{\boldsymbol{\Omega}=\mathbf{0}}(\boldsymbol{A}) = \mathrm{d}\,\widetilde{\mathrm{cay}}^{-1}_{\boldsymbol{\Omega}=\mathbf{0}}(\boldsymbol{A}) = \boldsymbol{A}/2$.

Based on Theorem 10.12, the Lie–Euler step (10.16) with $\Lambda = \widetilde{\mathrm{cay}}$ becomes

$$\boldsymbol{U}_1 = \widetilde{\mathrm{cay}}\left(-\tfrac{h}{2}\boldsymbol{P}_0\right)\boldsymbol{U}_0.$$

The update of the entire link field $[\boldsymbol{U}]$ reads

$$\boldsymbol{\Phi}_t^{\{\mathcal{T}\}}([\boldsymbol{P}],[\boldsymbol{U}]): \quad \begin{pmatrix}\boldsymbol{P}_\mu(\boldsymbol{n})\\ \boldsymbol{U}_\mu(\boldsymbol{n})\end{pmatrix} \mapsto \begin{pmatrix}\boldsymbol{P}_\mu(\boldsymbol{n})\\ \widetilde{\mathrm{cay}}(-\tfrac{t}{2}\boldsymbol{P}_\mu(\boldsymbol{n}))\boldsymbol{U}_\mu(\boldsymbol{n})\end{pmatrix}. \tag{10.19}$$

To emphasize the use of the numerical approximation (10.19) instead of the exact flow (9.40a), we will replace the link updates $\boldsymbol{\varphi}_{a_i h}^{\{\mathcal{T}\}}$ with the approximation $\boldsymbol{\Phi}_{a_i h}^{\{\mathcal{T}\}}$ in the definition of the splitting methods and (Hessian-free) force-gradient integrators.

In general, the Lie–Euler method is a Lie group method of first order. As we will show, the numerical integration scheme $\boldsymbol{\Phi}_h^{\{\mathcal{T}\}}$ is convergent of order $p = 2$ when applied to (9.39a).

Lemma 10.14. *Consider the angle $\theta(\boldsymbol{\Omega})$ from* (10.13) *and let $\boldsymbol{\Omega} = -\frac{h}{2}\boldsymbol{P}_0$ according to the Lie–Euler step* (10.19)*. As $h \to 0$, it holds $\theta(-\frac{h}{2}\boldsymbol{P}_0) = \mathcal{O}(h)$.*

Proof. It holds

$$\gamma(-\tfrac{h}{2}\boldsymbol{P}_0) = \frac{4\mathfrak{Im}(\det(-\frac{h}{2}\boldsymbol{P}_0))}{\mathrm{tr}((-\frac{h}{2}\boldsymbol{P}_0)^2)} = \frac{-\frac{h^3}{8}4\mathfrak{Im}(\det(\boldsymbol{P}_0))}{\frac{h^2}{4}\,\mathrm{tr}(\boldsymbol{P}_0^2)} = -\tfrac{h}{2}\gamma(\boldsymbol{P}_0) = \mathcal{O}(h).$$

For the sake of brevity, we write $\gamma = \gamma(-\frac{h}{2}\boldsymbol{P}_0)$. For $\gamma = 0$, we have $\theta(-\frac{h}{2}\boldsymbol{P}_0) = 0$, which satisfies the condition $\theta(-\frac{h}{2}\boldsymbol{P}_0) = \mathcal{O}(h)$ trivially. For $\gamma \neq 0$, we may write

$$\begin{aligned}\theta(-\tfrac{h}{2}\boldsymbol{P}_0) &= \sin^{-1}\left(-\frac{1}{2}\left(\frac{1}{\gamma} - \mathrm{sgn}(\gamma)\sqrt{\frac{1+\gamma^2}{\gamma^2}}\right)\right) = \sin^{-1}\left(-\frac{1}{2}\left(\frac{1}{\gamma} - \frac{\sqrt{1+\gamma^2}}{\gamma}\right)\right)\\ &= \sin^{-1}\left(-\frac{1}{2\gamma}\left(1-\sqrt{1+\gamma^2}\right)\right).\end{aligned}$$

With the expansion $\sqrt{1+x^2} = 1 + \frac{x^2}{2} + \mathcal{O}(x^4)$, we obtain

$$\theta(-\tfrac{h}{2}\boldsymbol{P}_0) = \sin^{-1}\left(-\frac{1}{2\gamma}\left(1-\left(1+\frac{\gamma^2}{2}+\mathcal{O}(\gamma^4)\right)\right)\right) = \sin^{-1}\left(-\frac{1}{2}\left(-\frac{\gamma}{2}+\mathcal{O}(\gamma^3)\right)\right).$$

Since $\gamma = \mathcal{O}(h)$ and $\sin^{-1}(x) = x + \mathcal{O}(x^3)$, we obtain $\theta(-\frac{h}{2}\boldsymbol{P}_0) = \mathcal{O}(h)$. □

Theorem 10.15. *The Lie–Euler step* (10.19) *numerically approximates the exact flow* (9.40a) *with second order of accuracy.*

Proof. It is sufficient to show that

$$\widetilde{\mathrm{cay}}(-\tfrac{h}{2}\boldsymbol{P}_0) - \exp(-h\boldsymbol{P}_0) = \mathcal{O}(h^3).$$

The series expansion of the matrix exponential yields

$$\exp(-h\boldsymbol{P}_0) = \boldsymbol{I} - h\boldsymbol{P}_0 + \frac{h^2}{2}\boldsymbol{P}_0^2 + \mathcal{O}(h^3).$$

For sufficiently small h, we can expand the modified Cayley transform (10.14) via Neumann series and obtain the expansion (with $\theta = \theta(-\frac{h}{2}\boldsymbol{P}_0)$)

$$\widetilde{\mathrm{cay}}(-\tfrac{h}{2}\boldsymbol{P}_0) = \boldsymbol{I} - h\cos(\theta)\boldsymbol{P}_0 + \frac{h^2}{4}(1+\mathrm{e}^{-2\mathrm{i}\theta})\boldsymbol{P}_0^2 + \mathcal{O}(h^3). \tag{10.20}$$

According to Lemma 10.14, we have $\theta = \mathcal{O}(h)$ and thus

$$\cos(\theta) = 1 + \mathcal{O}(h^2), \qquad \text{and} \qquad \mathrm{e}^{-2\mathrm{i}\theta} = 1 + \mathcal{O}(h).$$

Inserting these expansions into the expansion of the modified Cayley transform yields

$$\widetilde{\mathrm{cay}}(-\tfrac{h}{2}\boldsymbol{P}_0) = \boldsymbol{I} - h\boldsymbol{P}_0 + \frac{h^2}{2}\boldsymbol{P}_0^2 + \mathcal{O}(h^3),$$

such that $\widetilde{\mathrm{cay}}(-\frac{h}{2}\boldsymbol{P}_0) - \exp(-h\boldsymbol{P}_0) = \mathcal{O}(h^3)$. □

Theorem 10.16. *Consider a splitting method or (Hessian-free) force-gradient integrator $\boldsymbol{\Phi}_h$ that is time-reversible and volume-preserving. Replacing the link update $\boldsymbol{\varphi}_h^{\{\mathcal{T}\}}$ (9.40a) with the Lie–Euler step $\boldsymbol{\Phi}_h^{\{\mathcal{T}\}}$ (10.19) affects neither the time-reversibility nor the volume preservation of the underlying method.*

Proof. Since the momentum updates (9.40b), force-gradient updates (9.43), and Hessian-free force-gradient updates (9.44) remain unchanged, we only need to investigate the approximation (10.19) of the link update (9.40a).

Time-reversibility. To prove the time-reversibility (2.15) of the Lie-Euler step (10.19), we have to show that

$$\boldsymbol{\rho} \circ \boldsymbol{\Phi}_h^{\{\mathcal{T}\}} \circ \boldsymbol{\rho} \circ \boldsymbol{\Phi}_h^{\{\mathcal{T}\}}(\boldsymbol{P}_0, \boldsymbol{U}_0) = (\boldsymbol{P}_0, \boldsymbol{U}_0),$$

with $\boldsymbol{\rho}(\boldsymbol{P}_0, \boldsymbol{U}_0) = (-\boldsymbol{P}_0, \boldsymbol{U}_0)$ changing the sign of momenta. A straightforward application of $\boldsymbol{\Phi}_h^{\{\mathcal{T}\}}$, followed by applying $\boldsymbol{\rho}$ yields $\big(-\boldsymbol{P}_0, \widetilde{\mathrm{cay}}(-\frac{h}{2}\boldsymbol{P}_0)\boldsymbol{U}_0\big)$. Once again applying $\boldsymbol{\Phi}_h^{\{\mathcal{T}\}}$ and using the identity $\widetilde{\mathrm{cay}}(\boldsymbol{\Omega})\,\widetilde{\mathrm{cay}}(-\boldsymbol{\Omega}) = \boldsymbol{I}$ results in $(-\boldsymbol{P}_0, \boldsymbol{U}_0)$. A final application of $\boldsymbol{\rho}$ yields the initial values $(\boldsymbol{P}_0, \boldsymbol{U}_0)$ again, showing the time-reversibility of the approximated link update (10.19).

Volume preservation. For the volume preservation (2.12), we have to show that the approximated link update $\boldsymbol{\Phi}_h^{\{\mathcal{T}\}}$ satisfies (2.12). The Jacobian reads

$$\frac{\partial \boldsymbol{\Phi}_h^{\{\mathcal{T}\}}(\boldsymbol{P}_0, \boldsymbol{U}_0)}{\partial(\boldsymbol{P}_0, \boldsymbol{U}_0)} = \begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} \\ -\frac{h}{2}\mathrm{d}\,\widetilde{\mathrm{cay}}_{\boldsymbol{P}}(-\frac{h}{2}\boldsymbol{P}_0)\boldsymbol{U}_0 & \widetilde{\mathrm{cay}}(-\frac{h}{2}\boldsymbol{P}_0) \end{pmatrix}.$$

Since $\widetilde{\mathrm{cay}}\colon \mathfrak{su}(3) \to \mathrm{SU}(3)$, it holds $\det(\widetilde{\mathrm{cay}}(-\frac{h}{2}\boldsymbol{P}_0)) = 1$ and thus also the overall Jacobian has determinant 1, proving the volume preservation. □

Summarizing, we conclude that replacing the link update (9.40a) with the approximation (10.19) in any splitting method or (Hessian-free) force-gradient integrator still results in a consistent numerical integration scheme that is time-reversible, volume-preserving, and yields numerical approximations of the links in SU(3).

In the spirit of the hierarchical splitting approach, replacing the link update (9.40a) with the approximation (10.19) corresponds to a replacement of the flow in the leaf node to which the kinetic part of the Hamiltonian is assigned. According to Theorem 10.15, the Lie–Euler method (10.19) has order $p = 2$. Consequently, Theorem 5.6 implies that the convergence order of splitting methods and (Hessian-free) force-gradient integrators up to order $p = 2$ remains unchanged. Based on the discussion in Section 5.2.3, methods of order $p > 2$ keep their order of convergence if the underlying splitting method is either a composition (3.11) designed for second-order base schemes, or a composition (3.15) designed for first-order base schemes. Therefore, composition techniques enable the derivation of Cayley-based methods of arbitrarily high convergence order.

We conclude this section by discussing the convergence order of different methods utilizing the approximation (10.19).

Example 10.17. The Cayley-based version of the Strang splitting (3.5) in its velocity version BAB (A.3) reads

$$\boldsymbol{\Phi}_h = \boldsymbol{\varphi}_{h/2}^{\{\mathcal{S}\}} \circ \boldsymbol{\Phi}_h^{\{\mathcal{T}\}} \circ \boldsymbol{\varphi}_{h/2}^{\{\mathcal{S}\}}. \tag{10.21}$$

Similarly, the position version ABA (A.6) reads

$$\boldsymbol{\Phi}_h = \boldsymbol{\Phi}_{h/2}^{\{\mathcal{T}\}} \circ \boldsymbol{\varphi}_h^{\{\mathcal{S}\}} \circ \boldsymbol{\Phi}_{h/2}^{\{\mathcal{T}\}}.$$

Both methods remain convergent of second order.

Example 10.18. Considering the second-order minimum-error variants BABAB (A.11) and ABABA (A.14), we obtain the Cayley-based variants

$$\boldsymbol{\Phi}_h = \boldsymbol{\varphi}_{b_1 h}^{\{\mathcal{S}\}} \circ \boldsymbol{\Phi}_{h/2}^{\{\mathcal{T}\}} \circ \boldsymbol{\varphi}_{(1-2b_1)h}^{\{\mathcal{S}\}} \circ \boldsymbol{\Phi}_{h/2}^{\{\mathcal{T}\}} \circ \boldsymbol{\varphi}_{b_1 h}^{\{\mathcal{S}\}} \tag{10.22}$$

and

$$\boldsymbol{\Phi}_h = \boldsymbol{\Phi}_{a_1 h}^{\{\mathcal{T}\}} \circ \boldsymbol{\varphi}_{h/2}^{\{\mathcal{S}\}} \circ \boldsymbol{\Phi}_{(1-2a_1)h}^{\{\mathcal{T}\}} \circ \boldsymbol{\varphi}_{h/2}^{\{\mathcal{S}\}} \circ \boldsymbol{\Phi}_{a_1 h}^{\{\mathcal{T}\}}, \tag{10.23}$$

respectively. Both methods remain convergent of second order. The optimal value of a_1 and b_1 have been computed under the assumption that the flows of the subsystems are computed exactly. With the numerical approximation of the link updates, the optimality of ABABA is no longer guaranteed: By Theorem 5.6, the coefficient a_1 affects the discretization error of the link updates. In contrast, BABAB remains optimal as b_1 has no impact on the computation of the link updates.

Example 10.19. We consider the composition (3.11) with the Strang splitting (10.21) of order $p = 2$ as the underlying base scheme. Utilizing the weights (3.13), we obtain

$$\boldsymbol{\Phi}_h = \boldsymbol{\varphi}_{\gamma_1 h/2}^{\{\mathcal{S}\}} \circ \boldsymbol{\Phi}_{\gamma_1 h}^{\{\mathcal{T}\}} \circ \boldsymbol{\varphi}_{(\gamma_1+\gamma_2)h/2}^{\{\mathcal{S}\}} \circ \boldsymbol{\Phi}_{\gamma_2 h}^{\{\mathcal{T}\}} \circ \boldsymbol{\varphi}_{(\gamma_1+\gamma_2)h/2}^{\{\mathcal{S}\}} \circ \boldsymbol{\Phi}_{\gamma_1 h}^{\{\mathcal{T}\}} \circ \boldsymbol{\varphi}_{\gamma_1 h/2}^{\{\mathcal{S}\}}, \tag{10.24}$$

which corresponds to the minimum-error variant BABABAB (A.19). By construction of the composition (cf. Section 5.2.3), the Cayley-based version is convergent of order four.

Similarly, a composition of (10.21) via Suzuki's fractals (3.14) also results in a Cayley-based method of order $p = 4$.

Example 10.20. Considering the composition (3.11) with seven applications of the Strang splitting (10.21) as the underlying base scheme with weights [100]

$$\begin{aligned} &\gamma_1 = \gamma_7 = 0.78451361047755726382, \quad \gamma_2 = \gamma_6 = 0.23557321335935813368, \\ &\gamma_3 = \gamma_5 = -1.17767998417887100 69, \quad \gamma_4 = 1.31518632068391121 89, \end{aligned} \tag{10.25}$$

yields a Cayley-based composition scheme of order $p = 6$. As a sixth-order splitting method with $s = 8$ stages, we refer to this composition method as s8p6.

Example 10.21. The Cayley-based version of the Hessian-free force-gradient integrator BADAB (A.9) reads

$$\boldsymbol{\Phi}_h = \boldsymbol{\varphi}_{h/6}^{\{\mathcal{S}\}} \circ \boldsymbol{\Phi}_{h/2}^{\{\mathcal{T}\}} \circ \boldsymbol{\varphi}_{2h/3,h^3/72}^{\{\mathcal{D}\}} \circ \boldsymbol{\Phi}_{h/2}^{\{\mathcal{T}\}} \circ \boldsymbol{\varphi}_{h/6}^{\{\mathcal{S}\}}. \tag{10.26}$$

Since the Hessian-free force-gradient integrator does not rely on a composition approach, the leading-order error terms of the approximated link updates do not cancel. Consequently, the Cayley-based version is only convergent of order $p = 2$.

Example 10.22. Based on the minimum-error variant of BABABABABAB (A.46), the Cayley-based version reads

$$\begin{aligned} \boldsymbol{\Phi}_h = \boldsymbol{\varphi}_{b_1 h}^{\{\mathcal{S}\}} \circ \boldsymbol{\Phi}_{a_2 h}^{\{\mathcal{T}\}} \circ \boldsymbol{\varphi}_{b_2 h}^{\{\mathcal{S}\}} \circ \boldsymbol{\Phi}_{a_3 h}^{\{\mathcal{T}\}} \circ \boldsymbol{\varphi}_{(0.5-b_1-b_2)h}^{\{\mathcal{S}\}} \circ \boldsymbol{\Phi}_{(1-2a_2-2a_3)h}^{\{\mathcal{T}\}} \circ \boldsymbol{\varphi}_{(0.5-b_1-b_2)h}^{\{\mathcal{S}\}} \\ \circ \boldsymbol{\Phi}_{a_3 h}^{\{\mathcal{T}\}} \circ \boldsymbol{\varphi}_{b_2 h}^{\{\mathcal{S}\}} \circ \boldsymbol{\Phi}_{a_2 h}^{\{\mathcal{T}\}} \circ \boldsymbol{\varphi}_{b_1 h}^{\{\mathcal{S}\}}. \end{aligned} \tag{10.27}$$

This splitting method of RKN type is not a composition of a second-order base scheme and therefore its Cayley-based version is only convergent of order $p = 2$.

10.4. Numerical Results

To assess the practical benefits of the modified Cayley transform, we perform HMC simulations and compare the proposed Cayley-based integrators with their matrix exponential-based counterparts. We consider a setting similar to that of [142] where the Cayley transform has been investigated for simulations in an SU(2) Yang–Mills theory using the HMC algorithm. Specifically, we perform numerical simulations in an SU(3) Yang–Mills theory using the HMC algorithm on a two-dimensional lattice of size $V = 32 \times 32$ with lattice spacing $a = 1$. The Hamiltonian takes the form (9.23) with the action given by the Wilson gauge action (9.3) and inverse coupling constant $\beta = 2$. For performing the numerical tests, we use the MATLAB implementation available in [112].

In a first simulation, we computed 1000 trajectories of length $\tau = 2$ for varying step sizes h. We consider all integrators mentioned in the preceding section with the link updates (9.40a) approximated via (10.19). The results are depicted in Fig. 10.2, confirming the theoretical findings on the convergence order of the Cayley-based integrators. On

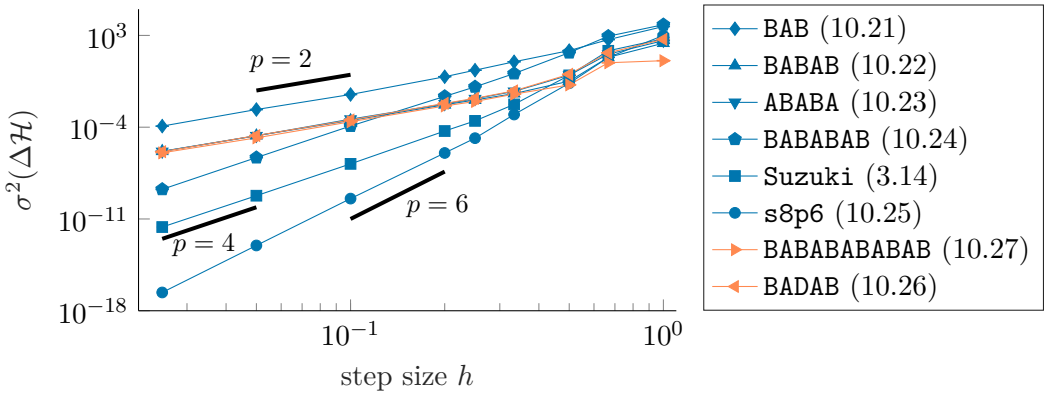


Fig. 10.2. Variance of $\Delta\mathcal{H}$ vs. step size h for different decomposition algorithms using the modified Cayley transform $\Lambda = \widetilde{\mathrm{cay}}$. For an integrator of order p, the variance $\sigma^2(\Delta\mathcal{H})$ scales with order $2p$. Whereas the order of second-order integrators and composition schemes (blue lines) remains unchanged, higher-order splitting methods and force-gradient integrators relying on the BCH formula are affected by an order reduction (orange lines). For all composition techniques, the method `BAB` (10.21) has been used as the underlying base scheme.

this small lattice, second-order algorithms turn out to be the most efficient ones [134]. Therefore, we focus on a comparison of Cayley-based integrators of second order and commonly used integrators that are based on the matrix exponential. According to (8.19), the optimal acceptance rate of second-order integrators is $\langle P_{\mathrm{acc}} \rangle_{\mathrm{opt}} = \exp(-1/2) \approx 61\%$. The expected acceptance rate can be determined from the (empirical) variance $\sigma^2(\Delta\mathcal{H})$ via (8.20).

In our MATLAB implementation, the modified Cayley transform is slightly less expensive to evaluate than the matrix exponential (using the analytical formula derived in [74]). Depending on the total number of links, evaluating the modified Cayley transform is approximately 10–20% faster[1]. In a preliminary, non-optimized implementation of the modified Cayley transform in `openQCD` [86], there were no significant differences in the computational cost compared to the implementation of the matrix exponential. Since the overall computational cost of the HMC algorithm is governed by the number of force evaluations n_f, these potentially minor variations in the evaluation costs of the local parameterizations become negligible.

[1] In [142], the Cayley-based Strang splitting was approximately 4.5 times faster than the version based on the matrix exponential. In [142], however, the matrix exponential has been computed via the significantly slower matrix decomposition method described in [96].

The central question is whether the application of the modified Cayley transform leads to an increased acceptance probability, enabling the use of larger step sizes and consequently reducing the number of force evaluations per trajectory of length τ. We aim at minimizing the total number of force evaluations per unit trajectory n_f/τ while achieving the optimal acceptance rate of 61%. For this purpose, we start from a thermalized configuration and compute 1000 trajectories of varying trajectory length τ and step size $h = \tau/2$. In Fig. 10.3, the variance $\sigma^2(\Delta\mathcal{H})$ is depicted for different values of n_f/τ. The horizontal dashed line indicates the optimal acceptance rate of 61%. Similarly, Fig. 10.4 shows the acceptance rate vs. n_f/τ. Both figures indicate that the method `ABABA` (10.23) achieves the most efficient computational process among all integrators under investigation. This is consistent with the investigations made in [134]. Furthermore, the results confirm that second-order integrators with $\Lambda = \widetilde{\mathrm{cay}}$ yield a more efficient computational process. Significant differences in the acceptance rate are observed particularly by comparing the Cayley-based variant (10.23) with the matrix exponential-based counterpart (A.14) of `ABABA`.

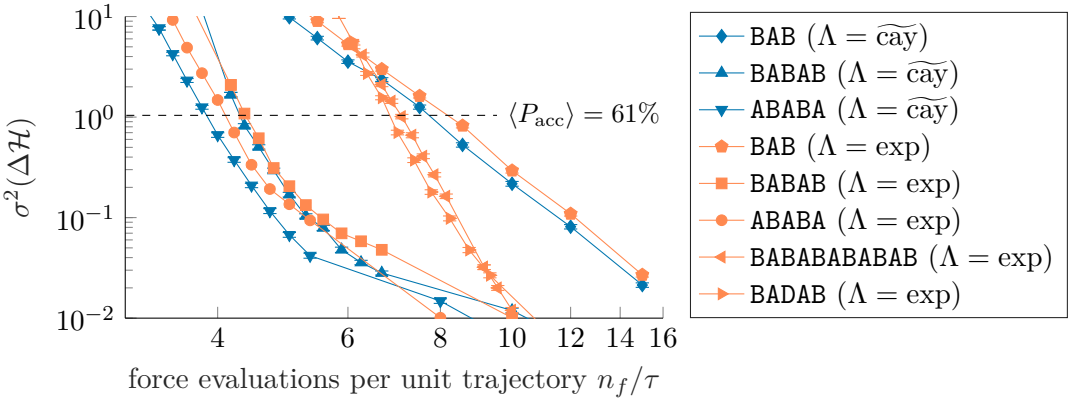


Fig. 10.3. Variance of $\Delta\mathcal{H}$ vs. number of force evaluations per unit trajectory n_f/τ for second-order Cayley-based integrators (blue lines) and a selection of commonly used integrators using the matrix exponential (orange lines).

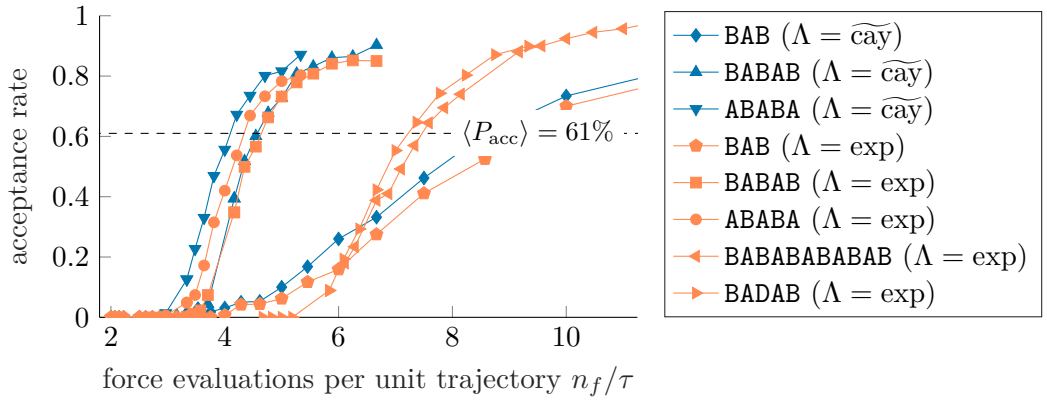


Fig. 10.4. Acceptance rate vs. number of force evaluations per unit trajectory n_f/τ for second-order Cayley-based integrators (blue lines) and a selection of commonly used integrators using the matrix exponential (orange lines).

For smaller step sizes, the exponential-based integrators become better than their Cayley-based counterparts. This is not surprising since the use of the modified Cayley transform introduces an additional error source by approximating the exact flow (9.40a). Despite the additional error source, the Cayley-based variants perform better

for acceptance rates close to $\langle P_{\text{acc}} \rangle_{\text{opt}} = 61\%$, i.e., in the region of interest. In this region, the employed step sizes are close to the stability threshold. Since $|\cos(\theta)| \leq 1$ and $|\mathrm{e}^{-2\mathrm{i}\theta}| = 1$, it follows from the expansion (10.20) that the modified Cayley transform shows a damping behavior in the leading terms compared to the matrix exponential. The numerical results provide evidence that the damping of the modified Cayley transform has a stabilizing effect that results in higher acceptance rates in the HMC algorithm. Specifically, the use of the modified Cayley transform as the local parameterization for splitting methods of second order turns out to be promising since particularly efficient variants that do not rely on composition techniques can be selected without sacrificing the convergence order.

Remark 10.23 (Projection-based modification of the Cayley transform)**.** An alternative approach to make the Cayley transform suitable for SU(3) is given by the projection

$$\widehat{\mathrm{cay}}(\boldsymbol{\Omega}) \coloneqq \det(\mathrm{cay}(\boldsymbol{\Omega}))^{-1/3} \, \mathrm{cay}(\boldsymbol{\Omega}). \tag{10.28}$$

The mappings (10.28) and (10.14) have similar evaluation costs. However, an expansion of the projection (10.28) reveals

$$\widehat{\mathrm{cay}}(-\tfrac{h}{2}\boldsymbol{P}_0) = \det(\mathrm{cay}(-\tfrac{h}{2}\boldsymbol{P}_0))^{-1/3} \cdot \left[\boldsymbol{I} - h\boldsymbol{P}_0 + \frac{h^2}{2}\boldsymbol{P}_0^2 - \frac{h^3}{4}\boldsymbol{P}_0^3 + \mathcal{O}(h^4) \right].$$

With $|\det(\mathrm{cay}(-\frac{h}{2}\boldsymbol{P}_0))| = 1$, the projection does not have the beneficial damping behavior like the modified Cayley transform (10.14). Instead, the expansion even has a larger constant in the $\mathcal{O}(h^3)$ term compared to the matrix exponential. Consequently, the use of the projection (10.28) will not result in beneficial stability properties.

10.5. Summary

The Cayley transform (10.6) constitutes a computationally efficient local parameterization for SU(2), though it is not directly applicable to SU(3). To address this, we developed a modified Cayley transform (10.14) that maps elements from the Lie algebra $\mathfrak{su}(3)$ into SU(3). This modification enables the derivation of Cayley-based versions of splitting methods and (Hessian-free) force-gradient integrators by substituting the link update (9.40a) with the approximation (10.19). The numerical flow (10.19) is time-reversible, volume-preserving, and achieves accuracy of second order. Utilizing this approximation in second-order integrators or methods based on compositions of a second-order base scheme maintains the overall convergence order. However, for other methods, such as most higher-order minimum-error variants from Appendix A, the approximation of the link updates results in an order reduction to $p = 2$, consistent with the order theory of hierarchical splitting methods (cf. Theorem 5.6). The expansion (10.20) of the modified Cayley transform reveals a damping behavior in the leading terms relative to the matrix exponential. Pure gauge simulations on a two-dimensional lattice of size $V = 32 \times 32$ provide numerical evidence that this damping behavior enhances the numerical stability of the integrators, resulting in higher acceptance rates in the HMC algorithm. These findings indicate that the modified Cayley transform, when employed as the local parameterization, may prove especially beneficial for lattice QCD simulations on relatively small lattices, where second-order integrators are commonly employed.

The subsequent chapter conducts HMC simulations in lattice QCD for several test scenarios, demonstrating the practical efficiency gains achievable through the numerical integration schemes developed in the first part of this thesis.

Chapter 11.

Numerical Results

In this chapter, we will conduct HMC simulations in lattice QCD for the generation of different ensembles. We will compare the proposed (hierarchical) Hessian-free force-gradient integrators with commonly employed (nested) integrators based on conventional splitting methods.

Hessian-free force-gradient integrators have been implemented into the software package `openQCD` [86]. The modified version is available via [114]. This version is used to perform the numerical simulations in Section 11.2. A branch of the repository also extends the nested integration framework by the reweighting (6.7) of the multirate factors. This branch is used to perform the numerical simulations in Section 11.4. Hessian-free force-gradient integrators have also been implemented successfully into the software package `tmLQCD` [139]. This modified version has been utilized to perform the numerical tests from Section 11.3. The extension of these software packages enables the use of the force-gradient approach via the Hessian-free framework in lattice QCD simulations. In contrast, conventional force-gradient integrators are not available in these software packages since the Hessian of the different actions is not readily available.

It would be desirable to compare the performance of conventional force-gradient integrators with their Hessian-free counterparts when applied in the molecular dynamics step of the HMC algorithm to perform gauge field generation in lattice field theories. To compare conventional force-gradient integrators with their Hessian-free counterparts, we begin by considering the *two-dimensional Schwinger model* [122] – quantum electrodynamics (QED) in one spatial and one temporal dimension. As a lattice quantum field theory, it shares many structural features with lattice QCD, including a fermion determinant and a gauge field, while being significantly cheaper to simulate. As the force-gradient term (4.1) has been implemented for the Schwinger model [124], it becomes an ideal benchmark for comparing conventional force-gradient integrators with their Hessian-free counterparts. Subsequently, in Section 11.2, we will move to lattice QCD and compare the computational efficiency of Hessian-free force-gradient integrators with conventional splitting methods for a setup with two heavy Wilson fermions that does not require multiple time stepping. In Section 11.3, we perform simulations for twisted-mass fermions to evaluate the accuracy of the stability threshold, cf. Proposition 4.19. To evaluate the efficiency of Hessian-free force-gradient integrators inside a hierarchical splitting method and the efficiency gain by utilizing the reweighting (6.7) of the multirate factors, Section 11.4 considers a setup with two mass-degenerate Wilson fermions that benefits from the multiple time stepping approach.

11.1. The Two-Dimensional Schwinger Model

The two-dimensional Schwinger model is defined by the Hamiltonian

$$\mathcal{H}([\boldsymbol{P}],[\boldsymbol{U}]) = \frac{1}{2}\sum_{\boldsymbol{n},\mu} p_\mu(\boldsymbol{n})^2 + \mathcal{S}^{\{\mathfrak{G}\}}([\boldsymbol{U}]) + \mathcal{S}^{\{\mathfrak{PF}\}}([\boldsymbol{U},\boldsymbol{\phi}])$$

with lattice volume $V = L \times T$, $\mathcal{S}^{\{\mathfrak{G}\}}([\boldsymbol{U}])$ the gauge action, and $\mathcal{S}^{\{\mathfrak{PF}\}}([\boldsymbol{U},\boldsymbol{\phi}])$ the pseudofermion action. The links $U_\mu(\boldsymbol{n})$ on the lattice, connecting the sites $\boldsymbol{n}$ and

$\boldsymbol{n} + \hat{\boldsymbol{\mu}}$, are elements of the Abelian Lie group U(1) and can be expressed as $U_\mu(\boldsymbol{n}) = \exp(\mathrm{i} q_\mu(\boldsymbol{n})) \in \mathrm{U}(1)$, $q_\mu(\boldsymbol{n}) \in [-\pi, \pi]$, where $\mu \in \{x, t\}$ labels the spatial and temporal directions. The momenta $p_\mu(\boldsymbol{n}) \in \mathbb{R}$ are elements of the corresponding Lie algebra. The action $\mathcal{S}$ is given by the sum of the gauge part

$$\mathcal{S}^{\{\mathfrak{G}\}}([\boldsymbol{U}]) = \beta \sum_{\boldsymbol{n}} (1 - \mathfrak{Re}(\mathcal{P}(\boldsymbol{n})))$$

summed over all plaquettes

$$\mathcal{P}(\boldsymbol{n}) = \boldsymbol{U}_t(\boldsymbol{n})^\dagger \boldsymbol{U}_x(\boldsymbol{n} + \hat{\boldsymbol{t}})^\dagger \boldsymbol{U}_t(\boldsymbol{n} + \hat{\boldsymbol{x}}) \boldsymbol{U}_x(\boldsymbol{n}),$$

scaled by the inverse gauge coupling constant $\beta \geq 0$. The fermion part (for $N_\mathrm{f} = 2$ mass-degenerate fermion flavors) is given by

$$\mathcal{S}^{\{\mathfrak{P}\mathfrak{F}\}}([\boldsymbol{U}, \boldsymbol{\phi}]) = \boldsymbol{\phi}^\dagger \left(\boldsymbol{D}^\dagger \boldsymbol{D} \right)^{-1} \boldsymbol{\phi},$$

with pseudofermion field $\boldsymbol{\phi}$ and Wilson–Dirac operator with components

$$\boldsymbol{D}_{\boldsymbol{n},\boldsymbol{m}} = (2 + m_0) \boldsymbol{I} \delta_{\boldsymbol{n},\boldsymbol{m}} - \frac{1}{2} \sum_\mu \left((\boldsymbol{I} - \boldsymbol{\sigma}_\mu) \boldsymbol{U}_\mu(\boldsymbol{n}) \delta_{\boldsymbol{n},\boldsymbol{m}-\hat{\boldsymbol{\mu}}} + (\boldsymbol{I} + \boldsymbol{\sigma}_\mu) \boldsymbol{U}_\mu^\dagger(\boldsymbol{n} - \hat{\boldsymbol{\mu}}) \delta_{\boldsymbol{n},\boldsymbol{m}+\hat{\boldsymbol{\mu}}} \right),$$

where m_0 denotes the bare mass parameter, $\delta_{\boldsymbol{n},\boldsymbol{m}}$ the Kronecker delta acting on the pseudofermion field via

$$\sum_{\boldsymbol{m}} \delta_{\boldsymbol{n},\boldsymbol{m}} \boldsymbol{\phi}(\boldsymbol{m}) = \boldsymbol{\phi}(\boldsymbol{n}),$$

and $\boldsymbol{\sigma}_x \equiv \boldsymbol{\sigma}_1$ and $\boldsymbol{\sigma}_t \equiv \boldsymbol{\sigma}_2$ the Pauli matrices from (10.8).

11.1.1. Comparison of the Minimum-Error Variants

On a lattice of size 16×16 with inverse coupling constant $\beta = 1$ and bare mass parameter $m_0 = 0.352443$ [30], we compute 2000 trajectories of length $\tau = 1$ with $n_\mathrm{steps} = 20, 25, 30, 35, 40$ number of time steps for all integrator coefficient sets in [101] and [115]. We choose a comparatively large bare quark mass corresponding to the scaling variable $z = 0.8$ in [30, Tab. 4] to ensure a numerically stable integration for the setup without using multiple time stepping techniques (cf. Chapter 6) and Hasenbusch mass-preconditioning (cf. Section 9.2.5). For all variants, we consider both the conventional force-gradient updates (4.4) and Hessian-free force-gradient updates (4.10). Using the measured dependence of the variance $\sigma^2(\Delta\mathcal{H})$ on the number of time steps n_steps together with the relation (8.20), we estimate for each integrator variant the number of force evaluations required to achieve an acceptance rate of 90%.

In Fig. 11.1, we perform a pair-wise comparison of force-gradient integrators and their Hessian-free counterparts with the coefficients being selected from either [101] or [115]. As shown in Fig. 11.1a, the Hessian-free variants generally require less force evaluations to achieve the prescribed acceptance rate of 90%. The largest differences occur for variants affected by order reduction, such as `ADABADABADA` (A.49). For these integrator variants, there are larger differences in the optimal integrator coefficients in [101] and [115]. Moreover, methods of order $p \leq 4$ are more efficient for the given system. This behavior is expected on moderate lattice volumes, where the asymptotic regime in which higher-order integrators dominate has not yet been reached. Instead, the reduced number of force evaluations for lower-order schemes outweighs the improved step size scaling of higher-order integrators.

For simplicity, force-gradient integrators are often implemented by approximating the force-gradient updates via the Hessian-free force-gradient updates (4.10), see for

example [102]. In these implementations, the integrator coefficients from [101] are utilized. The derivation of these integrator coefficients did not account for additional error terms introduced by the approximation of the force-gradient updates. The results in Fig. 11.1b show that the Hessian-free variants are more efficient when combined with the optimized coefficients from Appendix A rather than the coefficients from [101]. In contrast, Fig. 11.1c highlights that the minimum-error coefficients from Appendix A are only recommended for Hessian-free force-gradient integrators. Using the coefficients for conventional force-gradient integrators yields less efficient schemes. This is consistent with their design, which targets the Hessian-free formulation. For conventional force-gradient integrators, the coefficients from [101] should be employed since they are optimized for these methods.

11.1.2. Comparison of the Numerical Stability

Although the linear stability analysis for force-gradient integrators and their Hessian-free variants coincides [117], the two frameworks generally differ when applied to lattice field theories. Consequently, the natural question arises regarding the impact of the Hessian-free force-gradient updates (4.10) on the numerical stability of the integrator. As an example, we compare the numerical stability of the conventional force-gradient integrator `BACAB` (4.6) and its Hessian-free variant `BADAB` (A.9) by computing 1000 trajectories of length $\tau = 4$ on a 16×16 lattice with parameters $\beta = 1$ and $m_0 = 0.352443$ (see [30]) for different number of time steps n_{steps} per trajectory. The results in Fig. 11.2 demonstrate that both integrators become unstable simultaneously. These results provide numerical evidence that both frameworks possess similar stability properties when applied to lattice field theories, consistent with the linear stability analysis where both formulations are equivalent. Moreover, both frameworks yield comparable values for $\sigma^2(\Delta\mathcal{H})$ when employing the same number of time steps n_{steps}. Given the lower computational cost observed in the regime of interest, the Hessian-free variants are the more efficient choice.

Overall, the numerical experiments in the two-dimensional Schwinger model demonstrate that Hessian-free force-gradient integrators retain the favorable stability properties of conventional force-gradient integrators while reducing the computational cost required to achieve a fixed acceptance rate. Since Hessian-free force-gradient integrators are readily available in software packages such as `openQCD` [86] and `tmLQCD` [139], we will perform lattice QCD simulations utilizing (hierarchical) Hessian-free force-gradient integrators in the following sections.

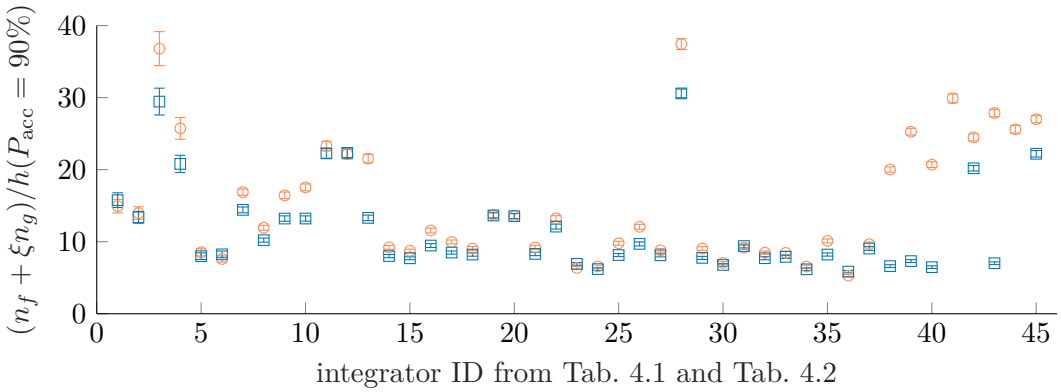
11.2. Simulations with Two Heavy Wilson Fermions

As an initial benchmark of Hessian-free force-gradient integrators in lattice QCD simulations, we consider an ensemble used in [78] generated with $N_{\text{f}} = 2$ mass-degenerate, non-perturbatively $\mathcal{O}(a)$ improved[1] Wilson quarks at a mass equal to half of the physical charm quark, $m_0 = m_c/2$. The mass m_0 is heavier than the physical up/down-quark mass, which leads to a better-conditioned Dirac operator and lower simulation cost.

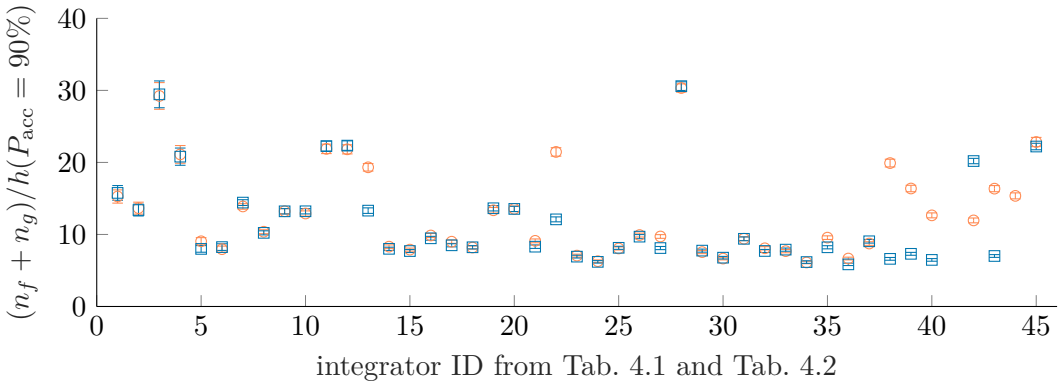
The fermion part is decomposed using even-odd reduction (cf. Section 9.2.4) in combination with one Hasenbusch mass preconditioning term (cf. Section 9.2.5) with shift parameter μ. Using the identity

$$\det(\boldsymbol{A}) = \frac{\det(\boldsymbol{A} + \mu^2 \boldsymbol{I})}{\det(\boldsymbol{I} + \mu^2 \boldsymbol{A}^{-1})}, \tag{11.1}$$

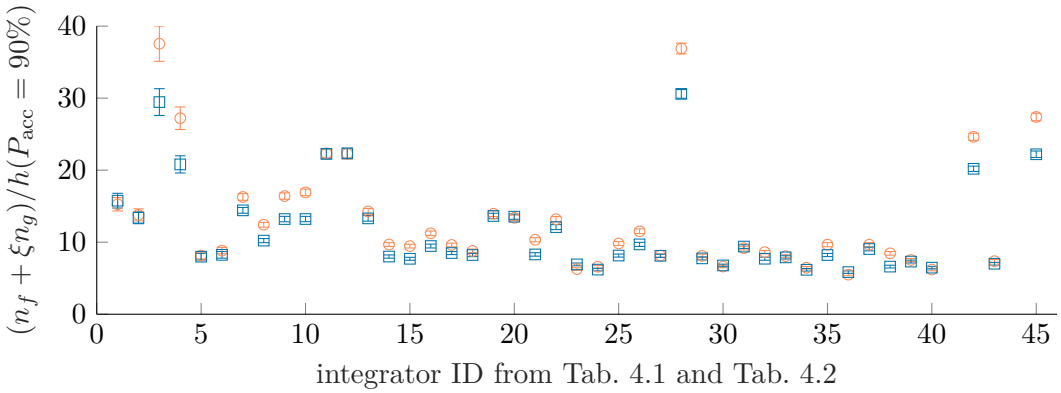
[1] The Wilson–Dirac operator (9.8) has $\mathcal{O}(a)$ discretization errors. The *clover-improved* Wilson–Dirac operator adds an additional term involving the *clover coefficient* c_{sw} that is tuned to cancel the $\mathcal{O}(a)$ errors, resulting in $\mathcal{O}(a^2)$ discretization errors. For details, see [33].



(a) Minimum-error variants of conventional force-gradient integrators (with coefficients derived in [101], orange circles) vs. minimum error variants of Hessian-free force-gradient integrators (with coefficients derived in [115] and given in Appendix A, blue squares).



(b) Hessian-free force-gradient integrators with coefficients from [101] (orange circles) vs. Hessian-free force-gradient integrators with coefficients from [115] (blue squares). The coefficients from [101] with ID = 41 yield $(n_f + n_g)/h(P_{\text{acc}} = 90\%) \approx 63$ and are not displayed for visualization purposes.



(c) Conventional force-gradient integrators with coefficients from [115] (red circles) vs. Hessian-free force-gradient integrators with coefficients from [115] (blue squares).

Fig. 11.1. Two-dimensional Schwinger model. Estimation of the required number of force evaluations per unit trajectory $(n_f + \xi n_g)/h$ to reach an acceptance probability of $P_{\text{acc}} = 90\%$. Evaluations of the force-gradient term (9.41) are considered twice as expensive as a force evaluation ($\xi = 2$), whereas the Hessian-free force-gradient update (9.44) effectively replaces the force-gradient term with a second force evaluation ($\xi = 1$). The sub-figures show a pair-wise comparison of force-gradient integrators and their Hessian-free counterparts with the coefficients being selected from either [101] or [115].

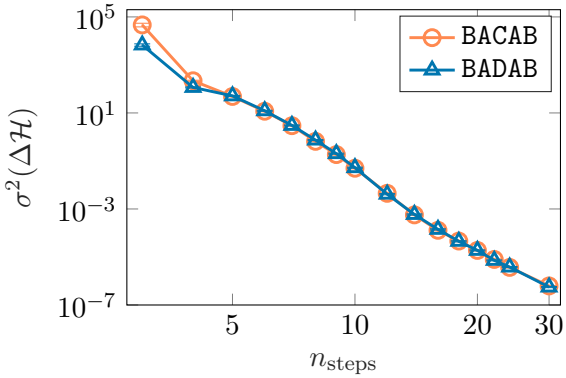


Fig. 11.2. Two-dimensional Schwinger model. Variance of $\Delta\mathcal{H}$ vs. the number of time steps per trajectory n_{steps} for the conventional force-gradient integrator `BACAB` (4.6) (○) and its Hessian-free counterpart `BADAB` (A.9) (△). For any value of n_{steps}, 1000 trajectories of length $\tau = 4$ have been computed.

which holds for any non-singular matrix $\boldsymbol{A}$ and constant μ, we can write the determinant factorization (9.33) as

$$\det(\boldsymbol{D}(m_0)\boldsymbol{D}(m_0)^\dagger) = \frac{\det\left(\hat{\boldsymbol{D}}\hat{\boldsymbol{D}}^\dagger + \mu^2\boldsymbol{I}\right)}{\det\left(\boldsymbol{I} + \mu^2(\hat{\boldsymbol{D}}\hat{\boldsymbol{D}}^\dagger)^{-1}\right)} \cdot \det(\boldsymbol{D}_{\text{oo}})^2,$$

with $\hat{\boldsymbol{D}}$ from (9.32) and the block-diagonal term $\boldsymbol{D}_{\text{oo}}$. The clover-improved Wilson–Dirac operator can be represented as a complex matrix with next-neighbor interactions. For a more detailed discussion, we refer to [49]. We obtain the pseudofermion action

$$\mathcal{S}^{\{\mathfrak{P}\mathfrak{F}\}}([\boldsymbol{U}, \boldsymbol{\phi}_1, \boldsymbol{\phi}_2]) = \langle\boldsymbol{\phi}_1, (\hat{\boldsymbol{D}}\hat{\boldsymbol{D}}^\dagger + \mu^2\boldsymbol{I})^{-1}\boldsymbol{\phi}_1\rangle + \langle\boldsymbol{\phi}_2, (\boldsymbol{I} + \mu^2(\hat{\boldsymbol{D}}\hat{\boldsymbol{D}}^\dagger)^{-1})\boldsymbol{\phi}_2\rangle - 2\,\mathrm{tr}\left(\log(\boldsymbol{D}_{\text{oo}})\right).$$

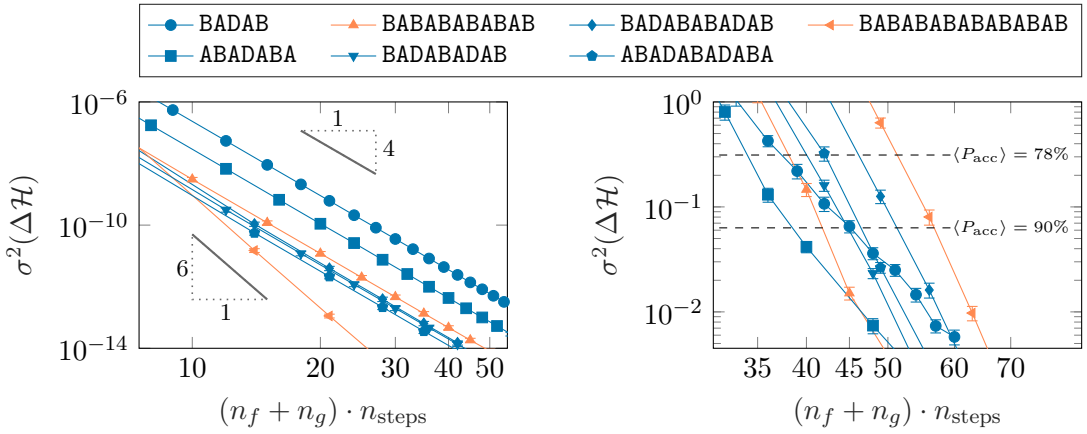
For the chosen bare mass parameter m_0, the resulting force hierarchy does not necessitate the use of multiple time stepping techniques, making it an ideal benchmark to evaluate the efficiency of the proposed Hessian-free force-gradient integrators without making use of the hierarchical splitting approach and multiple time stepping techniques[2]. The numerical tests are performed on a 48×24^3 lattice with inverse coupling constant $\beta = 5.3$ and hopping parameter $\kappa = 0.1327$ using an extended version of `openQCD v2.4` [114] including all minimum-error variants of Hessian-free force-gradient integrators reported in Appendix A.

Starting from a thermalized configuration, we computed 100 trajectories of length $\tau = 0.1$ with varying step sizes $h = \tau/n_{\text{steps}}$ ($n_{\text{steps}} \in \mathbb{N}$) for all non-dominated Hessian-free force-gradient integrators and splitting methods highlighted in Tab. 4.2. For these comparatively small step sizes, the integrators are not affected by numerical instabilities, and the variance scales as expected with order $2p$, as depicted in Fig. 11.3a. In this "scaling phase", the performance of the integrators is consistent with the efficiency measure $\mathrm{Eff}^{(p)}_{\mathtt{err}}$. However, the depicted region of $\sigma^2(\Delta\mathcal{H}) < 10^{-6}$ is not of practical interest as we are aiming to maximize the step size h while satisfying (8.21), i.e., we are interested in the region where the variance satisfies (8.22).

Repeating the simulations with $\tau = 2$ yields the results depicted in Fig. 11.3b, emphasizing that fourth-order integrators turn out to be the most efficient choice for

[2]We also performed numerical tests with nested integrators [123, 138] that employ a smaller step size to the gauge force. For the chosen lattice parameters, however, we were not able to observe any significant improvement in the acceptance probability by applying nested integration techniques.

this setup. Particularly, the Hessian-free force-gradient integrator ABADABA (A.22) allows for the most efficient computational process, i.e., it provides the best trade-off between accuracy and numerical stability for this specific setup. Moreover, BADAB (A.9) turns out to be an efficient choice as a more stable but less accurate variant. In contrast, the more accurate but less stable variants turn out to be less efficient.



(a) Trajectory length $\tau = 0.1$. The integrators BAB (A.3) and BABAB (A.11) achieve values of $\sigma^2(\Delta\mathcal{H}) > 10^{-7}$ for $(n_f + n_g) \cdot n_{\text{steps}} = 50$ and are not included for visualization purposes.

(b) Trajectory length $\tau = 2$. The integrators BAB (A.3) and BABAB (A.11) achieve values of $\sigma^2(\Delta\mathcal{H}) > 5 \cdot 10^{-1}$ for $(n_f + n_g) \cdot n_{\text{steps}} = 80$ and are not included for visualization purposes.

Fig. 11.3. Variance of $\Delta\mathcal{H}$ vs. the number of force evaluations per trajectory $(n_f + n_g) \cdot n_{\text{steps}}$ for all non-dominated variants of self-adjoint Hessian-free force-gradient integrators (blue lines) and conventional splitting methods (orange lines). For any integrator and step size, 100 trajectories have been computed.

For $p = 4$, the optimal acceptance rate (8.19) is $\langle P_{\text{acc}} \rangle_{\text{opt}} \approx 78\%$. Based on the results presented in Fig. 11.3b, we can estimate the required number of force evaluations per trajectory to achieve $\sigma^2(\Delta\mathcal{H}) \approx 0.3121$, resulting in an acceptance probability of approximately 78%. For the six fourth-order integrators depicted in Fig. 11.3b, the computational costs are approximated by taking the two data points closest to the optimal acceptance rate, and then performing linear interpolation in logarithmic space. The results are summarized in Tab. 11.1. Comparing the most efficient Hessian-free force-gradient integrator ABADABA (A.22) with the most efficient conventional splitting method BABABABABAB (A.46), we can conclude that, for this ensemble, the Hessian-free force-gradient integrator requires approximately 89% of the computational cost to achieve the optimal acceptance rate.

ID	$(n_f + n_g) \cdot n_{\text{steps}}$ for $\langle P_{\text{acc}} \rangle_{\text{opt}}$
BADAB	37.3751
ABADABA	34.0219
BABABABABAB	38.1049
BADABADAB	40.0282
BADABABADAB	46.1456
ABADABADABA	42.0871

Tab. 11.1. Comparison of the required computational cost $(n_f + n_g) \cdot n_{\text{steps}}$ to achieve the optimal acceptance rate $\langle P_{\text{acc}} \rangle_{\text{opt}} \approx 78\%$ for fourth-order integrators. The results are obtained by performing a linear interpolation in logarithmic space of the two data points in Fig. 11.3b that are closest to the optimal acceptance rate.

Unless there are substantial differences in the acceptance probabilities, the choice of

the numerical integration scheme will not affect the integrated autocorrelation times [31], since all integrators approximate the same Hamiltonian flow numerically. To demonstrate this, we tuned the step size h for the two integrators `ABADABA` and `BABABABABAB` to achieve an acceptance rate of $\langle P_{\text{acc}} \rangle \geq 90\%$, resulting in a step size of $h = 0.2$ for both integrators. We then computed 2000 trajectories of length $\tau = 2$, evaluated the cost measure [31]

$$\texttt{cost} = \frac{(n_f + n_g) \cdot n_{\text{steps}}}{P_{\text{acc}} \cdot \tau}, \tag{11.2}$$

and estimated the integrated autocorrelation times of two different observables: the topological charge Q_0 measured at Wilson flow time t_0, and the Wilson flow reference scale t_0/a^2. The results are summarized in Tab. 11.2, demonstrating that the integrated autocorrelation times are similar for both integrators, despite the lower acceptance rate of the Hessian-free force-gradient integrator. According to the cost measure (11.2), the integrator `ABADABA` only requires approximately 84% of the computational cost compared to `BABABABABAB`.

ID	$(n_f + n_g) \cdot n_{\text{steps}}$	P_{acc}	$\tau_{\text{int}}(t_0)$ [MDU]	$\tau_{\text{int}}(Q_0)$ [MDU]	`cost`
`BABABABABAB`	50	97.5%	37.40 ± 12.66	22.91 ± 6.52	25.64
`ABADABA`	40	92.3%	28.15 ± 9.24	22.05 ± 6.31	21.67

Tab. 11.2. Comparison of the performance of two tuned setups based on 2000 trajectories of length $\tau = 2$. Both setups use a step size of $h = 0.2$. The results involve the number of force evaluations per trajectory, the acceptance rate P_{acc}, integrated autocorrelation times $\tau_{\text{int}}(t_0)$ and $\tau_{\text{int}}(Q_0)$ in molecular dynamics units (MDUs), and the measure (11.2) for the computational cost.

Despite the simplicity of the ensemble, the results demonstrate the superior efficiency of Hessian-free force-gradient integrators compared to conventional splitting methods. They moreover highlight the importance of numerical stability, even for the rather large fermion masses considered here. Consequently, we conclude that Hessian-free force-gradient integrators allow for more efficient gauge field generation in lattice QCD compared to conventional splitting methods.

11.3. Simulations with Two Twisted-Mass Fermions

The linear stability analysis from Section 4.3 considers the harmonic oscillator as a test equation to determine the stability threshold z_*, cf. Proposition 4.19. For interacting field theories, it is commonly argued [40, 71] that the instability in lattice QCD can be likened to that of a collection of oscillator modes, with the onset of the instability caused by the mode of highest frequency, ω_{max}. We aim to provide numerical evidence on the accuracy of the stability criterion $|h\omega_{\text{max}}| < z_*$ when performing lattice QCD simulations.

To investigate the bounds of the stability condition in the context of lattice QCD, we employ the twisted-mass formulation [48, 47], where the twisted-mass parameter μ modifies the Dirac operator such that the spectrum of $\boldsymbol{D}(m_0)\boldsymbol{D}(m_0)^\dagger$ is shifted away from zero:

$$\lambda_{\min}(\boldsymbol{D}(m_0)\boldsymbol{D}(m_0)^\dagger + \mu^2 \boldsymbol{I}) \geq \mu^2.$$

Hence, the twisted-mass term introduces a UV cut-off and suppresses near-zero modes of the Dirac operator, which are known to cause numerical instabilities in lattice simulations. At so-called maximal twist, i.e., when the Partially Conserved Axial Current (PCAC)

mass is tuned to zero[3], the probability density of eigenvalues peaks at the bounds [2]. We investigate the stability bound by considering $N_\mathrm{f} = 2$ twisted-mass fermions with clover coefficient $c_\mathrm{sw} = 1.57551$ and hopping parameter $\kappa = 0.137322$, cf. [1]. We consider a setup with one Hasenbusch mass [65] and multirate hierarchical splitting methods (without utilizing the reweighting (6.7)) based on the splitting tree from Fig. 7.1 for $N = 4$ partitions. We split the action into three components. For the Hasenbusch mass, we select $\kappa\mu_1 = 0.1$ and assign the ratio operator with lower bound $\lambda_\mathrm{min} \geq (\mu_1^2 - \mu_0^2)/\mu_0^2 \approx \mu_1^2/\mu_0^2$ to the node labeled $\mathcal{V}^{\{1\}}$ employing the largest step size (macro level). The additional Hasenbusch term $\boldsymbol{D}(m_0)\boldsymbol{D}(m_0)^\dagger + \mu_1^2\boldsymbol{I}$ is assigned to the node labeled $\mathcal{V}^{\{2\}}$ (intermediate level), and the Iwasaki pure gauge action[4] to the node labeled $\mathcal{V}^{\{3\}}$ employing the smallest step size (micro level). We choose the multirate factors $M^{\{2,3,4\}} = 2$ and $M^{\{3,4\}} = 1$ which ensure that the numerical instabilities are not caused by the sub-macro levels.

In a first setup, we assign the same conventional splitting method `BABAB` (4.35a) to all inner nodes of the splitting tree. On the one hand, we consider the minimum-error variant with $\lambda = \lambda_{\mathtt{err}}$ chosen according to (4.38) with $\mathrm{Eff}^{(2)}_{\mathtt{err}}(\lambda_{\mathtt{err}}) \approx 29.24$ and $z_*(\lambda_{\mathtt{err}}) \approx 2.5531$. On the other hand, we consider $\lambda = \lambda_{\mathtt{stab}} = 0.25$, resulting in two steps of `BAB` (A.3) with step size $h/2$, which maximizes the relative stability threshold and yields $\mathrm{Eff}^{(2)}_{\mathtt{err}}(\lambda_{\mathtt{stab}}) \approx 10.73$ and $z_*(\lambda_{\mathtt{stab}}) = 4$. When applied to the harmonic oscillator (4.23), selecting $\lambda_{\mathtt{stab}}$ instead of $\lambda_{\mathtt{err}}$ allows for the utilization of a $z_*(\lambda_{\mathtt{stab}})/z_*(\lambda_{\mathtt{err}}) \approx 1.5667$ times larger step size.

For varying number of time steps n_steps on the macro level, we computed 30 trajectories of length $\tau = 1$. The results are depicted in Fig. 11.4. Firstly, the numerical results confirm that the minimum-error variant is more accurate than the maximum-stability variant. Secondly, we estimated the number of time steps n_steps per trajectory for which the integrators become unstable. As a criterion for instability, we choose

$$\sigma^2(\Delta\mathcal{H}(n_\mathrm{steps})) + 1.5 \cdot \mathrm{std}\left(\sigma^2(\Delta\mathcal{H}(n_\mathrm{steps}))\right) > f(n_\mathrm{steps}), \tag{11.3}$$

where $\sigma^2(\Delta\mathcal{H}(n_\mathrm{steps}))$ denotes the variance of $\Delta\mathcal{H}$ when computing n_steps time steps per trajectory, $\mathrm{std}(\sigma^2(\Delta\mathcal{H}(n_\mathrm{steps})))$ its standard deviation, and $f(n_\mathrm{steps}) = c_0 + c_1/(n_\mathrm{steps})^{2p}$ is a least squares fitting function of the data $(n_\mathrm{steps}, \sigma^2(\Delta\mathcal{H}(n_\mathrm{steps})))$ for

$$n_\mathrm{steps} > \begin{cases} 40, & \lambda = \lambda_{\mathtt{err}}, \\ 26, & \lambda = \lambda_{\mathtt{stab}}. \end{cases} \tag{11.4}$$

We denote the onset of instability, i.e., the largest integer satisfying (11.3), as n_min. The estimates for n_min are highlighted in Fig. 11.4 by the dotted vertical lines and summarized in Tab. 11.3. They indicate that the maximum-stability version allows for a larger step size before becoming unstable, which is consistent with the linear stability analysis. Furthermore, Tab. 11.3 contains the ratios between the two integrator variants, emphasizing that the theoretical ratio of $z_*(\lambda_{\mathtt{stab}})/z_*(\lambda_{\mathtt{err}}) \approx 1.5667$ provides a good estimate for evaluating the numerical stability properties of decomposition algorithms when applied to lattice QCD simulations. Additionally, the results demonstrate that, as the ratio μ_1^2/μ_0^2 increases, the maximum-stability version becomes increasingly competitive in achieving the optimal acceptance rate of $\langle P_\mathrm{acc}\rangle_\mathrm{opt} \approx 61\%$ at minimal computational cost.

[3] The PCAC mass is an effective quark mass defined through the axial Ward identity [33]. Maximal twist is defined where the PCAC mass is tuned to zero, leading to $\mathcal{O}(a)$ improved lattice discretization effects.

[4] The Iwasaki pure gauge action [70] is an improved extension of the Wilson gauge action (9.3) in which, in addition to plaquette terms, rectangular Wilson loops are included. These modifications reduce lattice discretization errors while leaving the continuum limit unchanged.

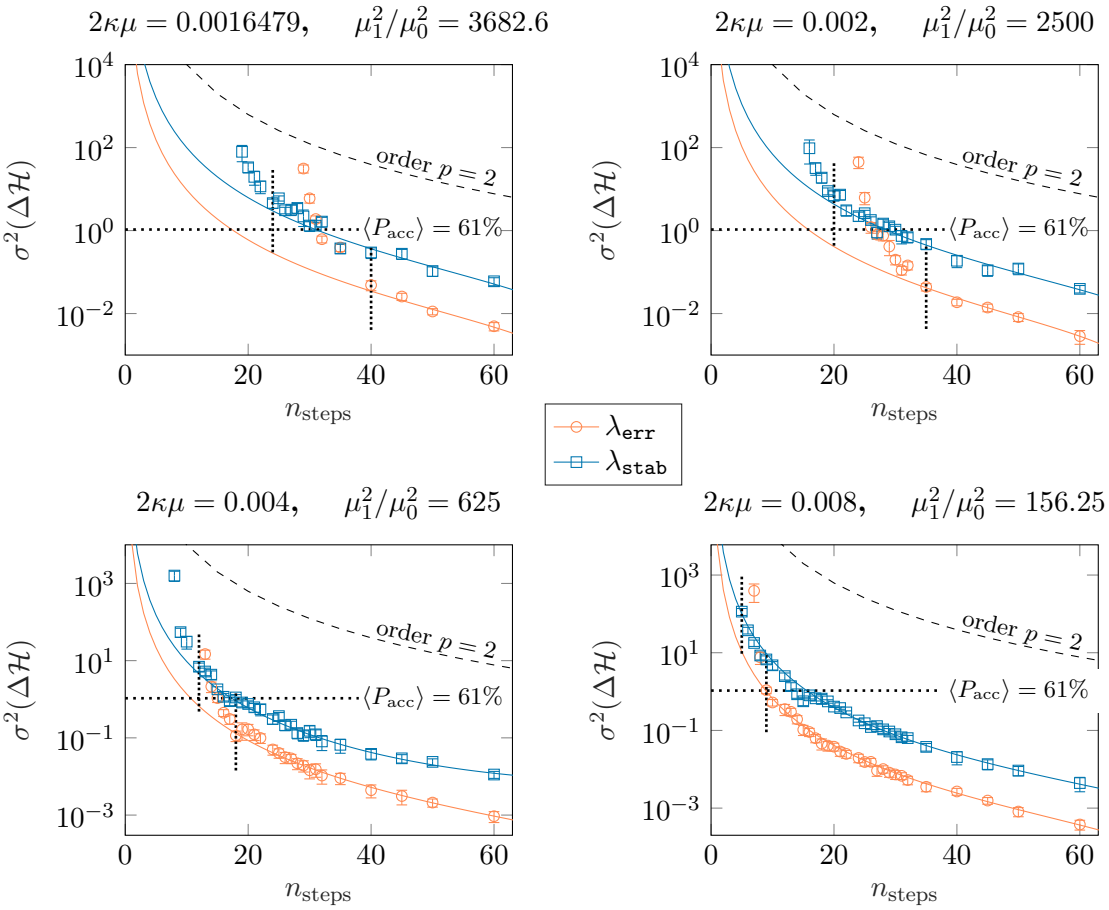


Fig. 11.4. Twisted-mass simulations. Variance of $\Delta\mathcal{H}$ vs. the number of time steps n_{steps} per trajectory for the splitting method BABAB (4.35a) with $\lambda = \lambda_{\texttt{err}}$ (○) and $\lambda = \lambda_{\texttt{stab}}$ (□). The solid lines denote the least squares fitting function $f(n_{\text{steps}})$. The vertical dotted lines indicate the number of time steps for which the integrator becomes unstable, see (11.3). The horizontal dotted lines indicate the optimal acceptance probability of $\langle P_{\text{acc}} \rangle_{\text{opt}} = \exp(-1/2) \approx 61\%$. The dashed lines are reference lines for convergence order $p = 2$. For any n_{steps}, 30 trajectories of length $\tau = 1$ have been computed.

In particular, for $\mu_1^2/\mu_0^2 = 156.25$, the minimum-error variant achieves the optimal acceptance rate for larger step sizes. In contrast, for $\mu_1^2/\mu_0^2 = 3682.6$, the maximum-stability variant allows for fewer time steps per trajectory.

In a second setup, we assign variants of the Hessian-free force-gradient integrator ABADABA (4.33b) to all inner nodes. In particular, we compare the minimum-error variant (A.22) with the variant maximizing the stability threshold that is obtained by choosing a_1 and c_2 according to (4.33c) with

$$b_1 = 0.050446855826563. \tag{11.5}$$

Maximizing the stability threshold yields $z_* \approx 4.6703$ and thus a larger relative stability threshold $\text{Eff}_{\texttt{stab}} \approx 1.1676$ (vs. $z_* \approx 3.1377$ and $\text{Eff}_{\texttt{stab}} \approx 0.7844$ for the minimum-error variant). However, this variant is far less accurate with $\text{Eff}^{(4)}_{\texttt{err}} \approx 0.24$ (vs. $\text{Eff}^{(4)}_{\texttt{err}} \approx 26.19$ for the minimum-error variant). According to the linear stability analysis, the maximum-stability variant allows for the utilization of a

$$z_*(\texttt{stab})/z_*(\texttt{err}) = 4.6703/3.1377 \approx 1.4884$$

times larger step size compared to the minimum-error variant. For varying number of time steps n_{steps} on the macro level, we computed 30 trajectories of length $\tau = 0.35$.

The results are depicted in Fig. 11.5. Similar to the investigations of the splitting

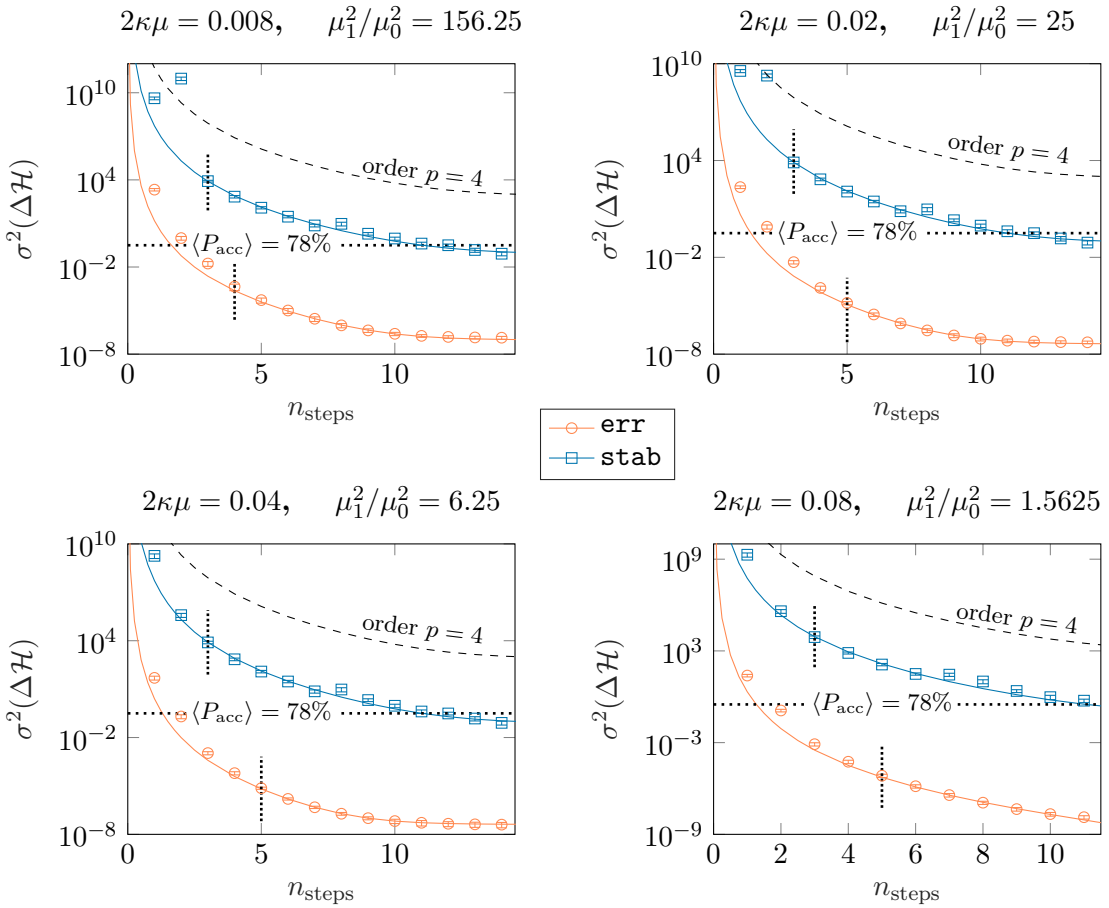


Fig. 11.5. Twisted-mass simulations. Variance of $\Delta\mathcal{H}$ vs. the number of time steps n_{steps} per trajectory for the Hessian-free force-gradient integrator `ABADABA` (4.33b) in its minimum-error variant (A.22) (○) and its maximum-stability variant (11.5) (□). The solid lines denote the least squares fitting function $f(n_{\text{steps}})$. The vertical dotted lines indicate the number of time steps for which the integrator becomes unstable, see (11.3). The horizontal dotted lines indicate the optimal acceptance probability of $\langle P_{\text{acc}} \rangle_{\text{opt}} = \exp(-1/4) \approx 78\%$. The dashed lines are reference lines for convergence order $p = 4$. For any n_{steps}, 30 trajectories of length $\tau = 0.35$ have been computed.

method `BABAB` in Fig. 11.4, we determined the onset of instability via (11.3), where for the determination of the least squares fitting functions, we use for both integrator variants the data $(n_{\text{steps}}, \sigma^2(\Delta\mathcal{H}(n_{\text{steps}})))$ for $n_{\text{steps}} > 6$. These estimates are indicated in Fig. 11.5 by the dotted vertical lines. The estimates, along with the ratios between the two integrator variants, are summarized in Tab. 11.4. We once again observe that the theoretical ratio of $z_*(\texttt{stab})/z_*(\texttt{err}) \approx 1.4884$ provides a good estimate for evaluating the numerical stability. However, for the ratios $\mu_1^2/\mu_0^2 \in \{156.25, 25\}$, we observe that the maximum-stability variant exhibits a significant increase of $\sigma^2(\Delta\mathcal{H})$ after passing the onset of instability. This can be attributed to the significant inaccuracy of the maximum-stability version. The instabilities begin at extremely high errors, where none of the samples have been accepted, and we have already exited the region of interest. In summary, across all four scenarios, the minimum-error variant (A.22) consistently achieves the optimal acceptance probability of $\langle P_{\text{acc}} \rangle_{\text{opt}} \approx 78\%$ while requiring less time steps per trajectory and thus is the more efficient choice. This once more emphasizes that maximizing the stability threshold without considering the accuracy is not advisable, as the accuracy becomes the bottleneck. In contrast, the minimum-error variant has a

slightly smaller stability threshold while simultaneously achieving significantly enhanced accuracy.

μ_1^2/μ_0^2	$n_{\min}(\lambda_{\texttt{err}})$	$n_{\min}(\lambda_{\texttt{stab}})$	$n_{\min}(\lambda_{\texttt{err}})/n_{\min}(\lambda_{\texttt{stab}})$
3682.60	40	24	1.6667
2500.00	35	20	1.7500
625.00	18	12	1.5000
156.25	9	5	1.8000

Tab. 11.3. Twisted-mass simulations. Estimates of the onset of instability $n_{\min}$ according to (11.3) for the splitting method BABAB (4.35a) with $\lambda \in \{\lambda_{\texttt{err}}, \lambda_{\texttt{stab}}\}$ and different values of μ_1^2/μ_0^2. The last column contains the ratios between the two integrator variants, indicating how much larger step sizes can be employed for the maximum-stability variant compared to the minimum-error variant.

μ_1^2/μ_0^2	$n_{\min}(\texttt{err})$	$n_{\min}(\texttt{stab})$	$n_{\min}(\texttt{err})/n_{\min}(\texttt{stab})$
156.25	4	3	1.3333
25.00	5	3	1.6667
6.25	5	3	1.6667
1.5625	5	3	1.6667

Tab. 11.4. Twisted-mass simulations. Estimates of the onset of instability $n_{\min}$ according to (11.3) for the Hessian-free force-gradient integrator ABADABA (4.33b) in its minimum-error variant (A.22) and its maximum-stability variant (11.5) for different values of μ_1^2/μ_0^2. The last column contains the ratios between the two integrator variants, indicating how much larger step sizes can be employed for the maximum-stability variant compared to the minimum-error variant.

11.4. Multirate Simulations with Two Wilson Fermions

To evaluate the hierarchical splitting approach in combination with Hessian-free force-gradient integrators in lattice QCD simulations, we perform numerical tests for a modification of the setup used in Section 11.2. We once again consider a setup with $N_{\mathrm{f}} = 2$ mass-degenerate, non-perturbatively $\mathcal{O}(a)$ improved Wilson fermions on a 64×32^3 lattice. The gauge coupling is fixed at $\beta = 5.3$, and the hopping parameter is set to $\kappa = 0.13550$, with clover coefficient $c_{\mathrm{sw}} = 1.909519$. The fermion part is decomposed using even-odd reduction [33] in combination with three Hasenbusch mass preconditioning terms with shift parameters

$$\mu_0 = 0, \quad \mu_1 = 0.01, \quad \mu_2 = 0.1, \quad \mu_3 = 1.0,$$

such that the determinant is factorized as

$$\det(\boldsymbol{D}(m_0)\boldsymbol{D}(m_0)^\dagger) = \prod_{i=0}^{2} \frac{\det(\hat{\boldsymbol{D}}\hat{\boldsymbol{D}}^\dagger + \mu_i^2 \boldsymbol{I})}{\det(\hat{\boldsymbol{D}}\hat{\boldsymbol{D}}^\dagger + \mu_{i+1}^2 \boldsymbol{I})} \cdot \det(\hat{\boldsymbol{D}}\hat{\boldsymbol{D}}^\dagger + \mu_3^2 \boldsymbol{I}) \cdot \det(\boldsymbol{D}_{\mathrm{oo}})^2,$$

with even-odd preconditioned Dirac operator (9.32). The pseudofermion action becomes

$$\begin{aligned}\mathcal{S}^{\{\mathfrak{P}\mathfrak{F}\}}([\boldsymbol{U}, \boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \boldsymbol{\phi}_3, \boldsymbol{\phi}_4]) &= \langle \boldsymbol{\phi}_1, (\hat{\boldsymbol{D}}\hat{\boldsymbol{D}}^\dagger + \mu_3^2 \boldsymbol{I})^{-1} \boldsymbol{\phi}_1 \rangle \\ &+ \langle \boldsymbol{\phi}_2, (\hat{\boldsymbol{D}}\hat{\boldsymbol{D}}^\dagger + \mu_2^2 \boldsymbol{I})^{-1} (\hat{\boldsymbol{D}}\hat{\boldsymbol{D}}^\dagger + \mu_3^2) \boldsymbol{\phi}_2 \rangle + \langle \boldsymbol{\phi}_3, (\hat{\boldsymbol{D}}\hat{\boldsymbol{D}}^\dagger + \mu_1^2 \boldsymbol{I})^{-1} (\hat{\boldsymbol{D}}\hat{\boldsymbol{D}}^\dagger + \mu_2^2) \boldsymbol{\phi}_3 \rangle \\ &+ \langle \boldsymbol{\phi}_4, (\hat{\boldsymbol{D}}\hat{\boldsymbol{D}}^\dagger)^{-1} (\hat{\boldsymbol{D}}\hat{\boldsymbol{D}}^\dagger + \mu_1^2) \boldsymbol{\phi}_4 \rangle - 2 \operatorname{tr}\big(\log(\boldsymbol{D}_{\mathrm{oo}})\big),\end{aligned}$$

involving four pseudofermion fields. We set up multirate hierarchical Hessian-free force-gradient integrators based on the splitting tree from Fig. 7.1 with $N = 4$ partitions. The action term involving the pseudofermion field ϕ_4 is assigned to the node labeled $\mathcal{V}^{\{1\}}$ (macro level), the remaining terms of the pseudofermion action to the node labeled $\mathcal{V}^{\{2\}}$ (intermediate level), and the gauge action to the node labeled $\mathcal{V}^{\{3\}}$ (micro level). For evaluating the efficiency gain by utilizing the reweighting (6.7) of the multirate factors in lattice QCD simulations, we consider a state-of-the-art nested integration setup that is a common choice in lattice QCD simulations (see for example [22]), and two different hierarchical Hessian-free force-gradient integrators that are summarized in Tab. 11.5. The state-of-the-art choice s[3, 6, 6]p2 uses the second-order BABAB method (A.11) in the root node labeled $\mathcal{H}$, which can be attributed to its high value in the relative stability threshold ($\text{Eff}_{\texttt{stab}} \approx 1.2766$), and the balance in the integrator coefficients a_i which enables the use of the nested integration approach without introducing excessive computational overhead. Despite its good performance in practical computations, the choice in the root node leads to an overall hierarchical splitting method that is only convergent of order $p = 2$. The fourth-order hierarchical Hessian-free force-gradient integrators are thus expected to show a better scaling with the step size h, which is particularly important for increasing lattice volumes [43]. Each hierarchical Hessian-free force-gradient integrator is tested in two configurations: one with constant multirate factors (tuned manually) and one with reweighted multirate factors. The reweighted multirate factors $M^{\{v\}}$ are chosen such that $\max_j \lceil |c_j| \cdot M^{\{v\}} \rceil$ equals to the multirate factor used in the integrator variant with constant multirate factors.

		nodes			**multirate factors**		
ID	$\mathcal{H}$	$\mathcal{H} - \mathcal{V}^{\{1\}}$	$\mathcal{T} + \mathcal{V}^{\{3\}}$	n_{steps}	$M^{\{2,3,4\}}$	$M^{\{3,4\}}$	**rew.**
s[3, 6, 6]p2	BABAB	BABABABABAB	BABABABABAB	5	1	1	✗
s[3, 4, 6]p4	BADAB	ABADABA	BABABABABAB	3	2	2	✗
$\texttt{s[3, 4, 6]p4}_{\texttt{rew}}$					4	10	✓
s[4, 4, 6]p4	ABADABA	ABADABA	BABABABABAB	3	2	2	✗
$\texttt{s[4, 4, 6]p4}_{\texttt{rew}}$					3	12	✓

Tab. 11.5. Setup of hierarchical splitting methods and Hessian-free force-gradient integrators for lattice QCD simulations with two Wilson fermions. The table summarizes the integrator ID, the methods assigned to the inner nodes of the splitting tree, the number of time steps n_{steps} per trajectory, the multirate factors assigned to the edges of the splitting tree, and whether the method uses the reweighting (6.7) of the multirate factors.

Starting from a thermalized configuration, we computed 100 trajectories of length $\tau = 2$ for all four setups. The numerical results are given in Tab. 11.6. The results emphasize that both hierarchical Hessian-free force-gradient integrators benefit from the reweighting (6.7) of the multirate factors, as they both achieve similar values for $\sigma^2(\Delta\mathcal{H})$ as their respective counterparts with constant multirate factors while requiring less force evaluations. The reweighting for s[3, 4, 6]p4 only reduces the number of evaluations of the gauge force, whose computational cost is comparatively negligible. In contrast, the reweighting for s[4, 4, 6]p4 also reduces the number of force evaluations of the force terms assigned to the node labeled $\mathcal{V}^{\{2\}}$, which can have a significant impact on the overall computational time. In our simulations, we were able to observe that $\texttt{s[4, 4, 6]p4}_{\texttt{rew}}$ only requires approximately 83% of the computational cost compared to s[4, 4, 6]p4 while achieving comparable values for $\sigma^2(\Delta\mathcal{H})$. Surprisingly, $\texttt{s[4, 4, 6]p4}_{\texttt{rew}}$ achieves even slightly smaller values for $\sigma^2(\Delta\mathcal{H})$ than s[4, 4, 6]p4. The generation of an ensemble consisting of 10,000 configurations would require approximately four days less computing time when using the variant utilizing the reweighting of the multirate factors. This

once more highlights that the reweighting facilitates the reduction of computational overhead, thereby offering significantly more efficient integrator variants. Furthermore, we can conclude that the hierarchical Hessian-free force-gradient integrators achieve similar performances compared to the state-of-the-art choice $\mathtt{s}[3,6,6]\mathtt{p2}$. On the one hand, a fine-tuning of the integrators is difficult for the given setup since the low value of n_{steps} implies that increasing or decreasing its value by one already leads to major differences in the step size. Therefore, it is difficult to evaluate which integrator achieves the best performance for this setup. On the other hand, the hierarchical Hessian-free force-gradient integrators have a better scaling (due to their convergence order of $p = 4$), such that it is anticipated that their performance becomes superior as soon as the lattice volume is increased further. A thorough analysis of hierarchical Hessian-free force-gradient integrators and the impact of the reweighting (6.7) on their overall performance in lattice QCD simulations requires additional simulations for other setups, which is part of future research.

ID	$\sigma^2(\Delta\mathcal{H})$	$\langle P_{\text{acc}} \rangle$	no. force evaluations $\mathcal{V}^{\{1\}}$	$\mathcal{V}^{\{2\}}$	$\mathcal{V}^{\{3\}}$
$\mathtt{s}[3,6,6]\mathtt{p2}$	0.032884 ± 0.00517273	92.78%	11	51	251
$\mathtt{s}[3,4,6]\mathtt{p4}$	0.060233 ± 0.00769773	90.23%	10	48	421
$\mathtt{s}[3,4,6]\mathtt{p4}_{\mathtt{rew}}$	0.064307 ± 0.00919525	89.91%	10	48	331
$\mathtt{s}[4,4,6]\mathtt{p4}$	0.020031 ± 0.00332963	94.36%	12	80	701
$\mathtt{s}[4,4,6]\mathtt{p4}_{\mathtt{rew}}$	0.011128 ± 0.00204863	95.79%	12	64	410

Tab. 11.6. Numerical results for the hierarchical splitting methods and Hessian-free force-gradient integrators from Tab. 11.5. The results contain the empirical variance $\sigma^2(\Delta\mathcal{H})$ and its standard deviation, the expected acceptance probability according to (8.20), and the number of force evaluations per trajectory.

11.5. Summary

The numerical results presented in this chapter emphasize that Hessian-free force-gradient integrators constitute promising choices for performing HMC simulations in lattice QCD. Numerical tests for the two-dimensional Schwinger model highlight that the accuracy and numerical stability of force-gradient integrators and their Hessian-free counterparts are similar. Since Hessian-free force-gradient integrators effectively replace the force-gradient term by another force evaluation, they are straightforward to implement into existing software packages and are characterized by reduced computational cost. The numerical tests from Section 11.2 demonstrate that the framework of Hessian-free force-gradient integrators provides more efficient integrator variants compared to conventional splitting methods, which, according to our accuracy and numerical stability investigations, can be attributed to their superior numerical stability. However, the results from Section 11.3 show that maximizing the numerical stability while not paying attention to the accuracy of the method does not lead to efficient integrator variants. This is consistent with our investigations from Section 4.4.3, where we proposed maximizing the efficiency measure $\text{Eff}^{(p)}_{\mathtt{err}}$ while detecting integrator variants that are also characterized by a comparatively large relative stability threshold $\text{Eff}_{\mathtt{stab}}$, cf. the non-dominated integrator variants highlighted in Tab. 4.2. The simulations conducted for the hierarchical Hessian-free force-gradient integrators in Section 11.4 emphasize that the extension of nested integrators [123, 138] by a) including the framework of Hessian-free force-gradient integrators and b) proposing the reweighting (6.7) of the multirate factors yields promising multiple time stepping schemes for lattice QCD simulations.

Throughout this chapter, we considered rather large fermion masses, which highlights that the relative stability threshold $\mathrm{Eff}_{\texttt{stab}}$ is already important for larger fermion masses. On the one hand, it is anticipated that the relative stability threshold $\mathrm{Eff}_{\texttt{stab}}$ will become increasingly important for smaller quark masses. Thus, it is expected that the methods `BADAB` (A.9) and `ABADABA` (A.22) will become more efficient choices compared to `BABABABABAB` (A.46) due to their superior numerical stability. On the other hand, increasing the lattice size does not affect the stability threshold but the accuracy demands, i.e., smaller step sizes h must be employed to achieve the same acceptance probability. Consequently, it is anticipated that the efficiency measure $\mathrm{Eff}^{(p)}_{\texttt{err}}$, which assesses the accuracy at the same computational cost, becomes increasingly important for larger lattice sizes. Moreover, the total costs for ensemble generation by the HMC algorithm scale with the volume V as [43]

$$\mathrm{cost}_{\texttt{HMC}} \propto V^{1+1/(2p)}, \tag{11.6}$$

such that higher-order integrators become beneficial for increasing lattice size. This interplay has to be studied further to examine in which situations a certain integrator provides the best possible trade-off.

Chapter 12.

Conclusion and Outlook

12.1. Conclusion

This thesis developed geometric numerical integration methods at the intersection of splitting methods and multirate integration, with lattice QCD as the primary application domain. The main contributions can be summarized as follows.

12.1.1. Hessian-Free Force-Gradient Integrators

Based on the publications [115, 116, 117], this thesis presented a framework of Hessian-free force-gradient integrators, that is, a class of commutator-based splitting methods of RKN type that makes use of the force-gradient approach without necessitating the Hessian of the potential $\mathcal{V}(\boldsymbol{q})$. We derived order conditions up to order $p = 8$ and conducted a linear stability analysis (that also covers conventional force-gradient integrators). Based on the error and stability analysis, we introduced two distinct efficiency measures: an accuracy-related efficiency measure $\mathrm{Eff}^{(p)}_{\mathtt{err}}$ defined via a (possibly weighted) norm of the leading-order error coefficients relative to the computational cost, and a stability-related measure $\mathrm{Eff}_{\mathtt{stab}}$ that tracks the stability threshold relative to the computational cost. With these two measures, we identified non-dominated integrator variants among all self-adjoint Hessian-free force-gradient integrators with $s \leq 6$ stages, which we analyzed via a complete classification in Appendix A. Hessian-free force-gradient integrators are in general no longer symplectic, but they remain volume-preserving and time-reversible (considering a self-adjoint composition), rendering them efficient choices for the molecular dynamics step inside the HMC algorithm and thus for performing gauge field generation in lattice QCD. All minimum-error variants of Hessian-free force-gradient integrators presented in Appendix A have been implemented into `openQCD` [114]. Detailed numerical tests demonstrate the superior efficiency of Hessian-free force-gradient integrators compared to conventional splitting methods for lattice QCD simulations. In Section 11.2, the use of Hessian-free force-gradient integrators enabled gauge field generation with only approximately 89% of the computational cost compared to conventional splitting methods. The twisted-mass simulations from Section 11.3 demonstrated that the linear stability threshold z_* accurately predicts the onset of instability in lattice QCD simulations.

12.1.2. Hierarchical Splitting Methods

Hierarchical splitting methods have been introduced in [118] and constitute a novel framework which facilitates the derivation of splitting methods for N-split systems by iteratively applying splitting methods for two-split systems. For this class, the order theory of splitting methods for two-split systems naturally extends. The framework naturally extends to multiple time stepping, retaining the structure-preserving properties and inheriting the order theory of the singlerate counterparts. We proposed a reweighting technique for the multirate factors to reduce computational overhead and introduced the concept of computational order, deriving sufficient conditions under which superior accuracy is observed over a restricted step size range. We derived hierarchical splitting

methods for linear port-Hamiltonian systems, separable Hamiltonian systems with multirate potential, and gauge field generation in lattice QCD. Numerical tests for these examples demonstrate the efficiency of the hierarchical splitting approach compared to a straightforward construction of splitting methods via composition techniques.

12.1.3. Hierarchical Hessian-Free Force-Gradient Integrators

Commutator-based splitting approaches can be incorporated into the hierarchical splitting framework, provided that the approach is consistent with the subsystem that is considered in the inner node where the commutator-based splitting approach is assigned. We have shown that for the splitting tree from Fig. 7.1, which is inspired from nested integration [123, 138], force-gradient integrators, as well as their Hessian-free counterparts, can be assigned to inner nodes of the splitting tree without any structural difficulties. Therefore, we were able to combine the two frameworks of Hessian-free force-gradient integrators and hierarchical splitting methods, resulting in hierarchical Hessian-free force-gradient integrators. These integrators constitute efficient choices for separable Hamiltonian systems of the form (7.1) and in particular exploit the fine structure of the potential energy via multiple time stepping. By reformulating nested integration schemes as hierarchical splitting methods, the framework provides, for the first time, a rigorous error analysis for this widely used class of methods. Moreover, the hierarchical splitting framework offers two key advantages: first, it incorporates the Hessian-free force-gradient approach which provides more efficient integrator variants compared to conventional splitting methods (of RKN type). Second, the hierarchical splitting approach offers a reweighting (6.7) of the multirate factors. This reweighting contributes to reducing computational overhead, which has also been demonstrated by different numerical tests in Section 7.3 and Section 11.4. For the numerical tests in Section 11.4, we have implemented the reweighting technique into `openQCD` [114]. For the integrator using `ABADABA` on the macro level, replacing constant multirate factors with reweighted ones yields an integrator variant that only requires 83% of the computational cost while achieving similar values of $\sigma^2(\Delta\mathcal{H})$.

12.1.4. Modified Cayley Transform for SU(3)

The hierarchical splitting approach allows for numerically approximating the flows of the elementary subsystems assigned to the leaf nodes of a splitting tree. In the context of lattice QCD simulations, we derived a modified Cayley transform (10.14) for SU(3) in [120] and discussed the possibility of approximating the link updates (9.40a) via (10.19) utilizing the modified Cayley transform. The approximation affects neither the time-reversibility nor the volume preservation, and is accurate of second order. Therefore, utilizing the approximated link updates does not lead to an order reduction for second-order methods or methods based on compositions of a second-order base scheme. The expansion (10.20) of the modified Cayley transform reveals a damping behavior in the leading terms relative to the matrix exponential. Numerical tests emphasize that this damping behavior enhances the numerical stability of the integrators, thus enhancing the acceptance probability in the HMC algorithm.

Taken together, the frameworks developed in this thesis provide a comprehensive toolkit for the design of efficient, structure-preserving numerical integration schemes for HMC simulations in lattice QCD, with computational efficiency gains demonstrated across multiple test scenarios ranging from the two-dimensional Schwinger model to full lattice QCD with dynamical fermions.

12.2. Outlook

The framework of Hessian-free force-gradient integrators and the hierarchical splitting approach provide many interesting open challenges for future research. We will point to some possible future research questions while focusing on aspects relevant for gauge field generation in lattice QCD.

12.2.1. Classification of Sixth-Order Hessian-Free Force-Gradient Integrators

In Chapter 4, we performed a complete classification of self-adjoint Hessian-free force-gradient integrators with $s \leq 6$ stages while incorporating the efficiency measures $\mathrm{Eff}_{\mathtt{err}}^{(p)}$ and $\mathrm{Eff}_{\mathtt{stab}}$ into our investigations. Consequently, we have a complete classification of self-adjoint Hessian-free force-gradient integrators of order $p = 2$ and $p = 4$. For $p = 6$, the conventional splitting method `BABABABABABABAB` (4.40) currently constitutes the best-known choice in terms of both efficiency measures. By performing a complete classification of Hessian-free force-gradient integrators with $s > 6$ stages, the framework is expected to yield more efficient integrator variants, since it contains conventional splitting methods such as (4.40) when setting all integrator coefficients c_i equal to zero. Detecting promising sixth-order Hessian-free force-gradient integrators can be particularly useful for lattice QCD simulations with larger lattice sizes due to the volume scaling (11.6) of the computational cost for ensemble generation via the HMC algorithm.

12.2.2. Modified Cayley Transform in Large-Scale Simulations with Fermions

The damping behavior of the modified Cayley transform led to improved numerical stability in the pure gauge simulations conducted in Section 10.4. It remains an open question whether these observations generalize to practical lattice QCD simulations, where larger lattices with dynamical fermions requiring hierarchical splitting methods are considered. In particular, it would be valuable to investigate whether the use of the modified Cayley transform yields further efficiency gains in practical simulations with smaller quark masses where the numerical stability becomes increasingly important.

12.2.3. Extensions of the Hierarchical Splitting Approach

Despite its large flexibility in the design of splitting methods for N-split systems, hierarchical splitting methods currently entail certain limitations. In the present form, we can only assign a single numerical integration scheme to a node of the splitting tree, i.e., we always choose the same numerical integration scheme to numerically approximate the flow of a certain subsystem. The multiple time stepping approach constitutes a first step to weaken this restriction since the reweighting (6.7) of the multirate factors allows the flow to be approximated by a varying number of time steps of the same method. We have shown that the reweighting technique selects the optimal multirate factors. For sufficiently large multirate factors, the reweighting allows for a fine resolution such that most of the computational overhead can be removed. For multirate factors closer to one (which is commonly the case in lattice QCD simulations), the impact of the reweighting is limited. In the limit case $M = 1$, the reweighting has no impact at all. Here, it would be desirable to allow for different numerical integration schemes $\boldsymbol{\Phi}_h^{\{\boldsymbol{v}\}}$ depending on the coefficient c defining the flow $\boldsymbol{\varphi}_{ch}^{\{\boldsymbol{v}\}}$ we wish to numerically approximate. This can be achieved by extending the hierarchical splitting approach to allow multiple numerical integration schemes at a single node $\boldsymbol{v}$.

12.2.4. Hierarchical Splitting Approaches for the RMHMC Algorithm

Critical slowing down[1] currently contributes greatly to the systematic errors in lattice calculations. The Riemannian manifold HMC (RMHMC) algorithm [39, 53] aims at overcoming critical slowing down via Fourier acceleration of slow moving modes. The key idea is to use a link-dependent mass matrix $\boldsymbol{M}([\boldsymbol{U}])^{-1}$ to identify low and high momentum modes. Therefore, according to [72], we consider a non-separable Hamiltonian

$$\mathcal{H} = \mathcal{S}^{\{\mathfrak{G}\}}([\boldsymbol{U}]) + \mathcal{S}^{\{\mathfrak{PF}\}}([\boldsymbol{U}, \boldsymbol{\phi}]) + \tfrac{1}{2}\left[\boldsymbol{p}^\dagger \boldsymbol{M}([\boldsymbol{U}])^{-1}\boldsymbol{p} + \boldsymbol{\pi}^\dagger \boldsymbol{M}([\boldsymbol{U}])\boldsymbol{\pi} + \boldsymbol{\psi}^\dagger \boldsymbol{\psi}\right] \tag{12.1}$$

with an auxiliary field $\boldsymbol{\psi}$ and its corresponding momenta $\boldsymbol{\pi}$. These auxiliary quantities are introduced to cancel the additional force term caused by the link-dependence of the mass matrix, which is required to recover the original probability density. A major challenge lies in the non-separability of the Hamiltonian due to the link-dependent mass matrix $\boldsymbol{M}([\boldsymbol{U}])^{-1}$. The kinetic part can no longer be solved analytically such that a structure-preserving numerical integration generally requires implicit schemes. A combination of the hierarchical splitting framework with an extended phase space approach [105] may lead to explicit structure-preserving numerical time integration schemes for Hamiltonian systems of the form (12.1). In the extended phase space approach [105], an extended Hamiltonian is constructed on the product of two copies of the original phase space, whose fine structure allows for explicit symplectic integration in the extended phase space. However, besides the higher computational demand of solving a system of double dimension, we only know that the exact flow preserves the symplectic structure of the extended system, and not necessarily of the original system. Here, it remains to be investigated whether the extended system can be used to compute a numerical flow of the original system that is volume-preserving and time-reversible.

12.2.5. Further Application Domains of the Hierarchical Splitting Approach

The hierarchical splitting approach has been used to define (multirate) hierarchical splitting methods for ODEs with special emphasis on separable Hamiltonian systems. Building on the example from Section 5.3, port-Hamiltonian systems [121] constitute a class of dynamical ODE systems whose right-hand side naturally splits into $N > 2$ partitions. This structure can be exploited by means of splitting methods, cf. for example [83, 97, 50]. When considering coupled port-Hamiltonian systems [41], it is natural to split the right-hand side into $N > 3$ partitions to exploit the structure of the individual port-Hamiltonian subsystems. Therefore, a next natural step constitutes a comprehensive study of the hierarchical splitting approach for (coupled) port-Hamiltonian systems.

Splitting methods constitute a promising class of numerical time integration schemes beyond the ODE case, involving splitting methods for DAEs [7], PDEs [18], and SDEs [46]. All these classes introduce additional challenges for the design of splitting methods. Although hierarchical splitting methods have been formulated for ODEs, the hierarchical splitting approach is very general and can in principle be adapted to obtain hierarchical versions of splitting methods for DAEs, PDEs, and SDEs. However, further analysis is needed to determine whether the hierarchical splitting approach is compatible with the additional challenges arising from the type of equations.

[1] As the continuum limit ($a \to 0$) is approached, correlations extend over increasingly many lattice sites, so that Monte Carlo updates decorrelate configurations more slowly, resulting in increasing integrated autocorrelation times.

Appendix A.

Minimum-Error Variants

We derive integrator coefficients for all integrator variants of self-adjoint Hessian-free force-gradient integrators with up to $s \leq 6$ stages that yield the respective minimum-error variant, i.e., potential degrees of freedom in the integrator coefficients are chosen such that the efficiency measure (4.30) is maximized. In order to reduce the number of unknowns, we will take into account in advance the self-adjointness of the integrator by applying the symmetry conditions (4.9), as well as the consistency conditions $\sum_i a_i = \sum_i b_i = 1$ throughout this section. For any number of stages s, we obtain different variants based on the coefficients c_i set to zero in order to reduce the number n_f of force evaluations per time step. All variants are summarized by the total number of force evaluations n_f per time step, the attained convergence order p, the number of degrees of freedom ν in the integrator coefficients, and the set of coefficients resulting in the minimum-error variant. The resulting value in the efficiency measure (4.30) is provided as well. Note that for counting the number of force evaluations n_f for the velocity versions, the force evaluation of the first flow is neglected since the last flow evaluation of time step k and the first flow evaluation of time step $k+1$ require the same force evaluation such that it can be re-used.

$s = 2$ Stages

For $s = 2$ stages, the maximum attainable order is $p = 2$.

Velocity Versions

A self-adjoint and consistent velocity version with two stages takes the form

$$\boldsymbol{\Phi}_h = \boldsymbol{\varphi}^{\{\mathcal{D}\}}_{h/2,c_1h^3} \circ \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{h} \circ \boldsymbol{\varphi}^{\{\mathcal{D}\}}_{h/2,c_1h^3} \tag{A.1}$$

with one degree of freedom $c_1 \in \mathbb{R}$. The variant has convergence order $p = 2$ with leading-order error coefficients $\alpha = 1/12$ and $\beta = 1/24 + 2c_1$.

`DAD`. The free parameter can be used to set $\beta = 0$, resulting in the set of coefficients

$$\boldsymbol{a} = \begin{pmatrix} 0 & 1 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 1/2 & 1/2 \end{pmatrix}, \quad \boldsymbol{c} = \begin{pmatrix} -1/48 & -1/48 \end{pmatrix}, \tag{A.2a}$$

and the values

$$p = 2, \quad n_f = 2, \quad \nu = 1, \quad \mathrm{Err}_3 = 1/12, \quad \mathrm{Eff}^{(2)}_{\mathtt{err}} = 3. \tag{A.2b}$$

`BAB`. For $c_1 = 0$, we obtain the velocity version of the Strang splitting (3.5) where the role of the first and second subsystem are interchanged. The integrator coefficients are

$$\boldsymbol{a} = \begin{pmatrix} 0 & 1 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 1/2 & 1/2 \end{pmatrix}, \quad \boldsymbol{c} = \begin{pmatrix} 0 & 0 \end{pmatrix}, \tag{A.3a}$$

and result in the values

$$p = 2, \quad n_f = 1, \quad \nu = 0, \quad \mathrm{Err}_3 = \sqrt{5}/24 \approx 0.0932, \quad \mathrm{Eff}^{(2)}_{\mathtt{err}} = 24/\sqrt{5} \approx 10.73. \tag{A.3b}$$

Position Versions

For the position versions, we obtain the general form

$$\boldsymbol{\Phi}_h = \boldsymbol{\varphi}_{h/2}^{\{\mathcal{T}\}} \circ \boldsymbol{\varphi}_{h,c_1h^3}^{\{\mathcal{D}\}} \circ \boldsymbol{\varphi}_{h/2}^{\{\mathcal{T}\}} \tag{A.4}$$

with a free parameter $c_1 \in \mathbb{R}$. The method has convergence order $p = 2$ and leading-order error coefficients $\alpha = -1/24$ and $\beta = -1/12 + c_1$.

`ADA.` We can set $\beta = 0$ with the integrator coefficients

$$\boldsymbol{a} = \begin{pmatrix} 1/2 & 1/2 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 1 & 0 \end{pmatrix}, \quad \boldsymbol{c} = \begin{pmatrix} 1/12 & 0 \end{pmatrix}, \tag{A.5a}$$

resulting in

$$p = 2, \quad n_f = 1, \quad \nu = 0, \quad \mathrm{Err}_3 = \sqrt{5}/24 \approx 0.0932, \quad \mathrm{Eff}_{\mathtt{err}}^{(2)} = 24/\sqrt{5} \approx 10.73. \tag{A.5b}$$

`ABA.` For $c_1 = 0$, the integrator reduces to the Strang splitting (3.5) with coefficients

$$\boldsymbol{a} = \begin{pmatrix} 1/2 & 1/2 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 1 & 0 \end{pmatrix}, \quad \boldsymbol{c} = \begin{pmatrix} 0 & 0 \end{pmatrix}, \tag{A.6a}$$

and

$$p = 2, \quad n_f = 1, \quad \nu = 0, \quad \mathrm{Err}_3 = \sqrt{5}/24 \approx 0.0932, \quad \mathrm{Eff}_{\mathtt{err}}^{(2)} = 24/\sqrt{5} \approx 10.73. \tag{A.6b}$$

Remark A.1. The coefficients, as well as the values of Err_3, are identical to the classification of conventional force-gradient integrators [101]. Since evaluations of the force-gradient term (4.1) are effectively replaced with a second force evaluation, the efficiency values $\mathrm{Eff}_{\mathtt{err}}^{(2)}$ increase compared to the efficiency values reported in [101].

$s = 3$ Stages

With $s = 3$ stages, the derivation of fourth-order Hessian-free force-gradient integrators is possible. In the absence of Hessian-free force-gradient updates ($c_i = 0$, $i = 1, 2, 3$), only convergence order $p = 2$ can be achieved.

Velocity Versions

Self-adjoint three-stage velocity versions exhibit the form

$$\boldsymbol{\Phi}_h = \boldsymbol{\varphi}_{b_1h,c_1h^3}^{\{\mathcal{D}\}} \circ \boldsymbol{\varphi}_{h/2}^{\{\mathcal{T}\}} \circ \boldsymbol{\varphi}_{(1-2b_1)h,c_2h^3}^{\{\mathcal{D}\}} \circ \boldsymbol{\varphi}_{h/2}^{\{\mathcal{T}\}} \circ \boldsymbol{\varphi}_{b_1h,c_1h^3}^{\{\mathcal{D}\}} \tag{A.7}$$

with free parameters $b_1, c_1, c_2 \in \mathbb{R}$. Solving for order $p = 4$ while keeping c_1 as a free parameter yields

$$b_1 = 1/6, \quad c_2 = 1/72 - 2c_1.$$

`DADAD.` Selecting c_1 to maximize the efficiency measure results in

$$\boldsymbol{a} = \begin{pmatrix} 0 \\ 1/2 \\ 1/2 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 1/6 \\ 2/3 \\ 1/6 \end{pmatrix}, \quad \boldsymbol{c} = \begin{pmatrix} -0.000881991367333 \\ 0.015652871623554 \\ -0.000881991367333 \end{pmatrix}, \tag{A.8a}$$

and

$$p = 4, \quad n_f = 4, \quad \nu = 1, \quad \mathrm{Err}_5 \approx 0.000625, \quad \mathrm{Eff}_{\mathtt{err}}^{(4)} \approx 6.25. \tag{A.8b}$$

`BADAB`. Setting $c_1 = 0$ results in a unique solution with

$$\boldsymbol{a} = \begin{pmatrix} 0 & 1/2 & 1/2 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 1/6 & 2/3 & 1/6 \end{pmatrix}, \quad \boldsymbol{c} = \begin{pmatrix} 0 & 1/72 & 0 \end{pmatrix}, \tag{A.9a}$$

and

$$p = 4, \quad n_f = 3, \quad \nu = 0, \quad \mathrm{Err}_5 \approx 0.000728, \quad \mathrm{Eff}^{(4)}_{\mathtt{err}} \approx 16.96. \tag{A.9b}$$

This variant is the Hessian-free counterpart of the force-gradient integrator (4.6).

`DABAD`. Conversely, setting $c_2 = 0$ yields

$$\boldsymbol{a} = \begin{pmatrix} 0 & 1/2 & 1/2 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 1/6 & 2/3 & 1/6 \end{pmatrix}, \quad \boldsymbol{c} = \begin{pmatrix} 1/144 & 0 & 1/144 \end{pmatrix}, \tag{A.10a}$$

$$p = 4, \quad n_f = 3, \quad \nu = 0, \quad \mathrm{Err}_5 \approx 0.00335, \quad \mathrm{Eff}^{(4)}_{\mathtt{err}} \approx 3.68. \tag{A.10b}$$

`BABAB`. For $c_1 = c_2 = 0$, it is not possible to achieve convergence order four. We can choose b_1 to minimize Err_3, resulting in [92]

$$\boldsymbol{a} = \begin{pmatrix} 0 \\ 1/2 \\ 1/2 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 0.1931833275037836 \\ 0.6136333449924328 \\ 0.1931833275037836 \end{pmatrix}, \quad \boldsymbol{c} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \tag{A.11a}$$

$$p = 2, \quad n_f = 2, \quad \nu = 1, \quad \mathrm{Err}_3 \approx 0.00855, \quad \mathrm{Eff}^{(2)}_{\mathtt{err}} \approx 29.24. \tag{A.11b}$$

Position Versions

The position versions with $s = 3$ stages read

$$\boldsymbol{\Phi}_h = \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{a_1 h} \circ \boldsymbol{\varphi}^{\{\mathcal{D}\}}_{h/2, c_1 h^3} \circ \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{(1-2a_1)h} \circ \boldsymbol{\varphi}^{\{\mathcal{D}\}}_{h/2, c_1 h^3} \circ \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{a_1 h}, \tag{A.12}$$

where $a_1, c_1 \in \mathbb{R}$ are the degrees of freedom.

`ADADA`. Making use of all free parameters yields two real-valued solution of which the optimal one reads

$$\boldsymbol{a} = \begin{pmatrix} \frac{1}{2} - \frac{1}{2\sqrt{3}} \\ \frac{1}{\sqrt{3}} \\ \frac{1}{2} - \frac{1}{2\sqrt{3}} \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 1/2 \\ 1/2 \\ 0 \end{pmatrix}, \quad \boldsymbol{c} = \begin{pmatrix} \frac{2-\sqrt{3}}{48} \\ \frac{2-\sqrt{3}}{48} \\ 0 \end{pmatrix}, \tag{A.13a}$$

and results in the values

$$p = 4, \quad n_f = 4, \quad \nu = 0, \quad \mathrm{Err}_5 \approx 0.000718, \quad \mathrm{Eff}^{(4)}_{\mathtt{err}} \approx 5.44. \tag{A.13b}$$

`ABABA`. Setting $c_1 = 0$, we can no longer achieve order four. Choosing a_1 to minimize Err_3 results in the position counterpart of (A.11),

$$\boldsymbol{a} = \begin{pmatrix} 0.1931833275037836 \\ 0.6136333449924328 \\ 0.1931833275037836 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 1/2 \\ 1/2 \\ 0 \end{pmatrix}, \quad \boldsymbol{c} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \tag{A.14a}$$

$$p = 2, \quad n_f = 2, \quad \nu = 1, \quad \mathrm{Err}_3 \approx 0.00855, \quad \mathrm{Eff}^{(2)}_{\mathtt{err}} \approx 29.24. \tag{A.14b}$$

Remark A.2. All variants except for (A.8) have the same optimal coefficients as reported in [101] for conventional force-gradient integrators. However, the principal error coefficients and values of $\mathrm{Eff}^{(p)}_{\mathtt{err}}$ change due to the additional error term γ_5 (except for the splitting methods `BABAB` (A.11) and `ABABA` (A.14)). The Hessian-free force-gradient integrator (A.9) has already been derived in [145]. In [124], the Hessian-free force-gradient integrator `BADAB` (A.9) has been applied to the two-dimensional Schwinger model inside a *nested integrator* [123, 138]. In terms of the efficiency measure $\mathrm{Eff}^{(4)}_{\mathtt{err}}$, `BADAB` (A.9) is the most efficient fourth-order Hessian-free force-gradient integrator with $s \leq 3$ stages. The splitting methods `BABAB` (A.11) and `ABABA` (A.14) are the most efficient second-order integrators within this framework.

$s = 4$ Stages

For $s = 4$ stages, all variants allow for the derivation of integrators with order $p = 4$.

Velocity Versions

For the velocity versions, we have the structure

$$\boldsymbol{\Phi}_h = \boldsymbol{\varphi}^{\{\mathcal{D}\}}_{b_1 h, c_1 h^3} \circ \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{a_2 h} \circ \boldsymbol{\varphi}^{\{\mathcal{D}\}}_{(0.5-b_1)h, c_2 h^3} \circ \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{(1-2a_2)h} \circ \boldsymbol{\varphi}^{\{\mathcal{D}\}}_{(0.5-b_1)h, c_2 h^3} \circ \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{a_2 h} \circ \boldsymbol{\varphi}^{\{\mathcal{D}\}}_{b_1 h, c_1 h^3} \quad \text{(A.15)}$$

with $a_2, b_1, c_1, c_2 \in \mathbb{R}$. The order-three conditions yield the solutions

$$b_1 = \frac{1}{12}\left(6 + \frac{1}{a_2(a_2-1)}\right), \quad c_1 = -\frac{1}{288}\left(6 + 288c_2 - \frac{1}{a_2(a_2-1)^2}\right),$$

with free parameters $a_2 \in \mathbb{R} \setminus \{0, 1\}$ and $c_2 \in \mathbb{R}$.

`DADADAD`. Minimizing the leading-order error coefficients yields

$$\boldsymbol{a} = \begin{pmatrix} 0 \\ 0.273005515864808 \\ 0.453988968270384 \\ 0.273005515864808 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 0.080128674198082 \\ 0.419871325801918 \\ 0.419871325801918 \\ 0.080128674198082 \end{pmatrix}, \quad \boldsymbol{c} = \begin{pmatrix} 0.000271601364672 \\ 0.002959399979707 \\ 0.002959399979707 \\ 0.000271601364672 \end{pmatrix}, \quad \text{(A.16a)}$$

$$p = 4, \quad n_f = 6, \quad \nu = 2, \quad \mathrm{Err}_5 \approx 0.0000275, \quad \mathrm{Eff}^{(4)}_{\mathtt{err}} \approx 28.09. \quad \text{(A.16b)}$$

`BADADAB`. At $c_1 = 0$, the global minimum yields

$$\boldsymbol{a} = \begin{pmatrix} 0 \\ 0.281473422092232 \\ 0.437053155815536 \\ 0.281473422092232 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 0.087960811032557 \\ 0.412039188967443 \\ 0.412039188967443 \\ 0.087960811032557 \end{pmatrix}, \quad \boldsymbol{c} = \begin{pmatrix} 0 \\ 0.003060423791562 \\ 0.003060423791562 \\ 0 \end{pmatrix}, \quad \text{(A.17a)}$$

$$p = 4, \quad n_f = 5, \quad \nu = 1, \quad \mathrm{Err}_5 \approx 0.0000498, \quad \mathrm{Eff}^{(4)}_{\mathtt{err}} \approx 32.12. \quad \text{(A.17b)}$$

`DABABAD`. Conversely, setting $c_2 = 0$ yields

$$\boldsymbol{a} = \begin{pmatrix} 0 \\ 0.258529167713908 \\ 0.482941664572184 \\ 0.258529167713908 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 0.06527448132325 \\ 0.43472551867675 \\ 0.43472551867675 \\ 0.06527448132325 \end{pmatrix}, \quad \boldsymbol{c} = \begin{pmatrix} 0.003595899064589 \\ 0 \\ 0 \\ 0.003595899064589 \end{pmatrix}, \quad \text{(A.18a)}$$

$$p = 4, \quad n_f = 4, \quad \nu = 1, \quad \mathrm{Err}_5 \approx 0.000891, \quad \mathrm{Eff}^{(4)}_{\mathtt{err}} \approx 4.38. \quad \text{(A.18b)}$$

`BABABAB`. Setting $c_1 = c_2 = 0$, the unique solution is given by the triple jump composition (3.13) with `BAB` (A.3) as the underlying base scheme,

$$\boldsymbol{a} = \begin{pmatrix} 0 \\ \frac{1}{2-\sqrt[3]{2}} \\ -\frac{\sqrt[3]{2}}{2-\sqrt[3]{2}} \\ \frac{1}{2-\sqrt[3]{2}} \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} \frac{1}{2\cdot(2-\sqrt[3]{2})} \\ \frac{1-2\sqrt[3]{2}}{2\cdot(2-\sqrt[3]{2})} \\ \frac{1-2\sqrt[3]{2}}{2\cdot(2-\sqrt[3]{2})} \\ \frac{1}{2\cdot(2-\sqrt[3]{2})} \end{pmatrix}, \quad \boldsymbol{c} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \tag{A.19a}$$

$$p = 4, \quad n_f = 3, \quad \nu = 0, \quad \mathrm{Err}_5 \approx 0.0383, \quad \mathrm{Eff}^{(4)}_{\mathtt{err}} \approx 0.32. \tag{A.19b}$$

Position Versions

The position counterparts read

$$\boldsymbol{\Phi}_h = \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{a_1 h} \circ \boldsymbol{\varphi}^{\{\mathcal{D}\}}_{b_1 h, c_1 h^3} \circ \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{(0.5-a_1)h} \circ \boldsymbol{\varphi}^{\{\mathcal{D}\}}_{(1-2b_1)h, c_2 h^3} \circ \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{(0.5-a_1)h} \circ \boldsymbol{\varphi}^{\{\mathcal{D}\}}_{b_1 h, c_1 h^3} \circ \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{a_1 h} \tag{A.20}$$

with the free parameters $a_1, b_1, c_1, c_2 \in \mathbb{R}$. Similarly to the velocity versions, we can solve the third-order conditions for a_1 and c_1, resulting in

$$a_1 = \frac{1}{2} \pm \frac{1}{\sqrt{24 b_1}}, \quad c_1 = \frac{1}{24}\left(1 - 12c_2 \pm \sqrt{6 b_1}(1 - b_1)\right),$$

where b_1 and c_2 are remaining degrees of freedom.

`ADADADA`. We reach the global minimum at

$$\begin{aligned} \boldsymbol{a} &= \begin{pmatrix} 0.116438749543126 \\ 0.383561250456874 \\ 0.383561250456874 \\ 0.116438749543126 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 0.283216992495952 \\ 0.433566015008096 \\ 0.283216992495952 \\ 0 \end{pmatrix}, \\ \boldsymbol{c} &= \begin{pmatrix} 0.001247201195115 \\ 0.002974030329635 \\ 0.001247201195115 \\ 0 \end{pmatrix}. \end{aligned} \tag{A.21a}$$

The minimum-error variant is characterized by

$$p = 4, \quad n_f = 6, \quad \nu = 2, \quad \mathrm{Err}_5 \approx 0.0000200, \quad \mathrm{Eff}^{(4)}_{\mathtt{err}} \approx 38.57. \tag{A.21b}$$

`ABADABA`. Setting $c_1 = 0$, we obtain the minimum-error variant with the coefficients

$$\begin{aligned} \boldsymbol{a} &= \begin{pmatrix} 0.089775972994422 \\ 0.410224027005578 \\ 0.410224027005578 \\ 0.089775972994422 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 0.247597680043986 \\ 0.504804639912028 \\ 0.247597680043986 \\ 0 \end{pmatrix}, \\ \boldsymbol{c} &= \begin{pmatrix} 0 \\ 0.006911440413815 \\ 0 \\ 0 \end{pmatrix}, \end{aligned} \tag{A.22a}$$

resulting in

$$p = 4, \quad n_f = 4, \quad \nu = 1, \quad \mathrm{Err}_5 \approx 0.000149, \quad \mathrm{Eff}^{(4)}_{\mathtt{err}} \approx 26.19. \tag{A.22b}$$

`ADABADA`. Conversely, $c_2 = 0$ yields

$$\boldsymbol{a} = \begin{pmatrix} 0.136458051118946 \\ 0.363541948881054 \\ 0.363541948881054 \\ 0.136458051118946 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 0.315267858070664 \\ 0.369464283858672 \\ 0.315267858070664 \\ 0 \end{pmatrix}, \quad \boldsymbol{c} = \begin{pmatrix} 0.002427032834125 \\ 0 \\ 0.002427032834125 \\ 0 \end{pmatrix}, \tag{A.23a}$$

$$p = 4, \quad n_f = 5, \quad \nu = 1, \quad \mathrm{Err}_5 \approx 0.0000844, \quad \mathrm{Eff}^{(4)}_{\mathtt{err}} \approx 18.95. \tag{A.23b}$$

`ABABABA`. The conventional splitting method, obtained by setting $c_1 = c_2 = 0$, with $s = 4$ stages has a unique solution and corresponds to a triple jump composition (3.13) with `ABA` (A.6) as the underlying base scheme, which yields

$$\boldsymbol{a} = \begin{pmatrix} \frac{1}{2\cdot(2-\sqrt[3]{2})} \\ \frac{1-2\sqrt[3]{2}}{2\cdot(2-\sqrt[3]{2})} \\ \frac{1-2\sqrt[3]{2}}{2\cdot(2-\sqrt[3]{2})} \\ \frac{1}{2\cdot(2-\sqrt[3]{2})} \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} \frac{1}{2-\sqrt[3]{2}} \\ -\frac{\sqrt[3]{2}}{2-\sqrt[3]{2}} \\ \frac{1}{2-\sqrt[3]{2}} \\ 0 \end{pmatrix}, \quad \boldsymbol{c} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \tag{A.24a}$$

$$p = 4, \quad n_f = 3, \quad \nu = 0, \quad \mathrm{Err}_5 \approx 0.0283, \quad \mathrm{Eff}^{(4)}_{\mathtt{err}} \approx 0.44. \tag{A.24b}$$

Remark A.3. The most efficient Hessian-free force-gradient integrator with $s = 4$ stages is given by the position version `ADADADA` (A.21) with an efficiency value of $\mathrm{Eff}^{(4)}_{\mathtt{err}} \approx 38.57$. Compared to the most efficient three-stage integrator `BADAB` (A.9) with $\mathrm{Eff}^{(4)}_{\mathtt{err}} \approx 16.96$, the four-stage integrator is expected to reduce the global error approximately by a factor of $38.57/16.96 \approx 2.27$ using the same number of force evaluations. The conventional splitting methods `BABABAB` (A.19) and `ABABABA` (A.24) are convergent of order $p = 4$, but they exhibit low efficiency values. This highlights once again that the triple jump composition (3.13) is characterized by large truncation errors. In practice, we should instead choose a Hessian-free force-gradient integrator or one of the more efficient conventional splitting method with $s > 4$ stages that we will discuss below.

$s = 5$ Stages

Increasing the number of stages to $s = 5$ allows for the first sixth-order Hessian-free force-gradient integrator. Although there is one additional order condition compared to conventional force-gradient integrators, both frameworks demand $s = 5$ stages for attaining convergence order $p = 6$.

Velocity Versions

The velocity versions with $s = 5$ stages read

$$\begin{aligned} \boldsymbol{\Phi}_h = \boldsymbol{\varphi}^{\{\mathcal{D}\}}_{b_1 h, c_1 h^3} \circ \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{a_2 h} \circ \boldsymbol{\varphi}^{\{\mathcal{D}\}}_{b_2 h, c_2 h^3} \circ \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{(0.5-a_2)h} \circ \boldsymbol{\varphi}^{\{\mathcal{D}\}}_{(1-2b_1-2b_2)h, c_3 h^3} \circ \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{(0.5-a_2)h} \\ \circ \boldsymbol{\varphi}^{\{\mathcal{D}\}}_{b_2 h, c_2 h^3} \circ \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{a_2 h} \circ \boldsymbol{\varphi}^{\{\mathcal{D}\}}_{b_1 h, c_1 h^3}, \end{aligned} \tag{A.25}$$

where $a_2, b_1, b_2, c_1, c_2, c_3 \in \mathbb{R}$ are the degrees of freedom.

`BADADADAB`. Although the Hessian-free framework involves an additional order-5 condition, it has the same unique solution achieving order $p = 6$ as conventional force-gradient integrators [101], namely

$$\begin{aligned}
\boldsymbol{a} &= \begin{pmatrix} 0 \\ a_2 \\ 0.5 - a_2 \\ 0.5 - a_2 \\ a_2 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} b_1 \\ b_2 \\ 1 - 2(b_1 + b_2) \\ b_2 \\ b_1 \end{pmatrix}, \quad \boldsymbol{c} = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_2 \\ c_1 \end{pmatrix}, \\
a_2 &= \frac{1}{2} + \frac{\sqrt[3]{675 + 75\sqrt{6}}}{30} + \frac{5}{2\sqrt[3]{675 + 75\sqrt{6}}}, \quad b_1 = \frac{a_2}{3}, \quad b_2 = -\frac{5a_2}{3}(a_2 - 1), \\
c_1 &= 0, \quad c_2 = -\frac{5a_2^2}{144} + \frac{a_2}{36} - \frac{1}{288}, \quad c_3 = \frac{1}{144} - \frac{a_2}{36}\left(\frac{a_2}{2} + 1\right),
\end{aligned} \tag{A.26a}$$

resulting in the values

$$p = 6, \quad n_f = 7, \quad \nu = 0, \quad \mathrm{Err}_7 \approx 0.00154, \quad \mathrm{Eff}^{(6)}_{\mathtt{err}} \approx 0.0055. \tag{A.26b}$$

Since the solution is unique and satisfies $c_1 = 0$, we have `DADADADAD` = `BADADADAB`. To emphasize that the integrator starts and ends with a conventional momentum update (3.36b), we refer to the variant as `BADADADAB` and the variant `DADADADAD` is omitted.

`DABADABAD`. Demanding $c_2 = 0$, only convergence order $p = 4$ can be achieved while three integrator coefficients remain as free parameters. Maximizing $\mathrm{Eff}^{(4)}_{\mathtt{err}}$ yields

$$\begin{aligned}
\boldsymbol{a} &= \begin{pmatrix} 0 \\ 0.197279141794602 \\ 0.302720858205398 \\ 0.302720858205398 \\ 0.197279141794602 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 0.060885008530668 \\ 0.288579639891554 \\ 0.301070703155556 \\ 0.288579639891554 \\ 0.060885008530668 \end{pmatrix}, \\
\boldsymbol{c} &= \begin{pmatrix} 0.000429756946246 \\ 0 \\ 0.002373498029145 \\ 0 \\ 0.000429756946246 \end{pmatrix},
\end{aligned} \tag{A.27a}$$

$$p = 4, \quad n_f = 6, \quad \nu = 3, \quad \mathrm{Err}_5 \approx 0.0000130, \quad \mathrm{Eff}^{(4)}_{\mathtt{err}} \approx 59.33. \tag{A.27b}$$

`DADABADAD`. Similarly, setting $c_3 = 0$ results in the minimum-error variant

$$\begin{aligned}
\boldsymbol{a} &= \begin{pmatrix} 0 \\ 0.227758000273404 \\ 0.272241999726596 \\ 0.272241999726596 \\ 0.227758000273404 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 0.070935378258660 \\ 0.322911610232109 \\ 0.212306023018462 \\ 0.322911610232109 \\ 0.070935378258660 \end{pmatrix}, \\
\boldsymbol{c} &= \begin{pmatrix} 0.000067752132787 \\ 0.001597508440746 \\ 0 \\ 0.001597508440746 \\ 0.000067752132787 \end{pmatrix},
\end{aligned} \tag{A.28a}$$

and is characterized by

$$p = 4, \quad n_f = 7, \quad \nu = 3, \quad \mathrm{Err}_5 \approx 0.0000101, \quad \mathrm{Eff}^{(4)}_{\mathtt{err}} \approx 41.06. \tag{A.28b}$$

`BABADABAB`. Reducing the number of force evaluations by setting $c_1 = c_2 = 0$ results in

$$\boldsymbol{a} = \begin{pmatrix} 0 \\ 0.200395293638238 \\ 0.299604706361762 \\ 0.299604706361762 \\ 0.200395293638238 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 0.073943321445602 \\ 0.258244950046509 \\ 0.335623457015778 \\ 0.258244950046509 \\ 0.073943321445602 \end{pmatrix}, \quad \boldsymbol{c} = \begin{pmatrix} 0 \\ 0 \\ 0.003147048491590 \\ 0 \\ 0 \end{pmatrix}, \tag{A.29a}$$

$$p = 4, \quad n_f = 5, \quad \nu = 2, \quad \mathrm{Err}_5 \approx 0.0000651, \quad \mathrm{Eff}^{(4)}_{\mathtt{err}} \approx 24.57. \tag{A.29b}$$

`BADABADAB`. A more efficient minimum-error variant is obtained for $c_1 = c_3 = 0$ with

$$\boldsymbol{a} = \begin{pmatrix} 0 \\ 0.219039425103133 \\ 0.280960574896867 \\ 0.280960574896867 \\ 0.219039425103133 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 0.068466565514186 \\ 0.311000565033563 \\ 0.241065738904502 \\ 0.311000565033563 \\ 0.068466565514186 \end{pmatrix}, \quad \boldsymbol{c} = \begin{pmatrix} 0 \\ 0.001602470431500 \\ 0 \\ 0.001602470431500 \\ 0 \end{pmatrix}, \tag{A.30a}$$

$$p = 4, \quad n_f = 6, \quad \nu = 2, \quad \mathrm{Err}_5 \approx 0.0000105, \quad \mathrm{Eff}^{(4)}_{\mathtt{err}} \approx 73.45. \tag{A.30b}$$

`DABABABAD`. In contrast, setting $c_2 = c_3 = 0$ yields a less efficient minimum-error variant

$$\boldsymbol{a} = \begin{pmatrix} 0 \\ 0.190585159174513 \\ 0.309414840825487 \\ 0.309414840825487 \\ 0.190585159174513 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 0.036356798097337 \\ 0.340278911234329 \\ 0.246728581336668 \\ 0.340278911234329 \\ 0.036356798097337 \end{pmatrix}, \quad \boldsymbol{c} = \begin{pmatrix} 0.002005691094612 \\ 0 \\ 0 \\ 0 \\ 0.002005691094612 \end{pmatrix}, \tag{A.31a}$$

$$p = 4, \quad n_f = 5, \quad \nu = 2, \quad \mathrm{Err}_5 \approx 0.000336, \quad \mathrm{Eff}^{(4)}_{\mathtt{err}} \approx 4.76. \tag{A.31b}$$

`BABABABAB`. Finally, setting $c_1 = c_2 = c_3 = 0$, we obtain a family of fourth-order conventional splitting methods with a single free parameter. The respective minimum-error variant is given by

$$\boldsymbol{a} = \begin{pmatrix} 0 \\ 0.520943339103990 \\ -0.020943339103990 \\ -0.020943339103990 \\ 0.520943339103990 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 0.164498651557576 \\ 1.235692651138917 \\ -1.800382605392986 \\ 1.235692651138917 \\ 0.164498651557576 \end{pmatrix}, \quad \boldsymbol{c} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \tag{A.32a}$$

and is characterized by the values

$$p = 4, \quad n_f = 4, \quad \nu = 1, \quad \mathrm{Err}_5 \approx 0.000654, \quad \mathrm{Eff}^{(4)}_{\mathtt{err}} \approx 5.97. \tag{A.32b}$$

Position Versions

The position versions of five-stage Hessian-free force-gradient integrators are given by

$$\begin{aligned}\boldsymbol{\Phi}_h = \boldsymbol{\varphi}_{a_1 h}^{\{\mathcal{T}\}} \circ \boldsymbol{\varphi}_{b_1 h, c_1 h^3}^{\{\mathcal{D}\}} \circ \boldsymbol{\varphi}_{a_2 h}^{\{\mathcal{T}\}} \circ \boldsymbol{\varphi}_{(0.5-b_1)h, c_2 h^3}^{\{\mathcal{D}\}} \circ \boldsymbol{\varphi}_{(1-2a_1-2a_2)h}^{\{\mathcal{T}\}} \circ \boldsymbol{\varphi}_{(0.5-b_1)h, c_2 h^3}^{\{\mathcal{D}\}} \\ \circ \boldsymbol{\varphi}_{a_2 h}^{\{\mathcal{T}\}} \circ \boldsymbol{\varphi}_{b_1 h, c_1 h^3}^{\{\mathcal{D}\}} \circ \boldsymbol{\varphi}_{a_1 h}^{\{\mathcal{T}\}},\end{aligned} \tag{A.33}$$

where $a_1, a_2, b_1, c_1, c_2 \in \mathbb{R}$ are the degrees of freedom. Solving for order $p = 4$, we obtain the equations

$$\begin{aligned} a_1 &= \left(3 - 6a_2 + 12a_2 b_1 \mp \sqrt{3}\sqrt{1 - 24a_2^2 b_1 + 48 a_2^2 b_1^2}\right)/6, \\ c_2 &= \left(2 - 12 a_2 b_1 + 24 a_2 b_1^2 \mp \sqrt{3}\sqrt{1 - 24 a_2^2 b_1 + 48 a_2^2 b_1^2} - 48 c_1\right)/48, \end{aligned} \tag{A.34}$$

with three remaining free parameters a_2, b_1, c_1.

`ADADADADA`. The minimum-error variant is obtained by using the minus sign in (A.34), resulting in the integrator coefficients

$$\begin{aligned} \boldsymbol{a} &= \begin{pmatrix} 0.094471605659163 \\ 0.281057227947299 \\ 0.248942332787076 \\ 0.281057227947299 \\ 0.094471605659163 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 0.227712700174579 \\ 0.272287299825421 \\ 0.272287299825421 \\ 0.227712700174579 \\ 0 \end{pmatrix}, \\ \boldsymbol{c} &= \begin{pmatrix} 0.000577062053569 \\ 0.000817399268485 \\ 0.000817399268485 \\ 0.000577062053569 \\ 0 \end{pmatrix}, \end{aligned} \tag{A.35a}$$

and the values

$$p = 4, \quad n_f = 8, \quad \nu = 3, \quad \mathrm{Err}_5 \approx 0.00000501, \quad \mathrm{Eff}_{\mathtt{err}}^{(4)} \approx 48.71. \tag{A.35b}$$

`ABADADABA`. Restricting ourselves to solutions with $c_1 = 0$ yields the minimum-error variant (with minus sign in (A.34)):

$$\begin{aligned} \boldsymbol{a} &= \begin{pmatrix} 0.047802682977081 \\ 0.265994592108478 \\ 0.372405449828882 \\ 0.265994592108478 \\ 0.047802682977081 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 0.143282503449494 \\ 0.356717496550506 \\ 0.356717496550506 \\ 0.143282503449494 \\ 0 \end{pmatrix}, \\ \boldsymbol{c} &= \begin{pmatrix} 0 \\ 0.002065558490728 \\ 0.002065558490728 \\ 0 \\ 0 \end{pmatrix}, \end{aligned} \tag{A.36a}$$

$$p = 4, \quad n_f = 6, \quad \nu = 2, \quad \mathrm{Err}_5 \approx 0.0000346, \quad \mathrm{Eff}_{\mathtt{err}}^{(4)} \approx 22.32. \tag{A.36b}$$

`ADABABADA`. Alternatively, we may restrict our investigations to solutions satisfying $c_2 = 0$. The minimum-error variant is obtained using the minus sign in (A.34) and reads

$$\boldsymbol{a} = \begin{pmatrix} 0.118030603246046 \\ 0.295446189611111 \\ 0.173046414285686 \\ 0.295446189611111 \\ 0.118030603246046 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 0.273985556386628 \\ 0.226014443613372 \\ 0.226014443613372 \\ 0.273985556386628 \\ 0 \end{pmatrix}, \quad \boldsymbol{c} = \begin{pmatrix} 0.001466561305710 \\ 0 \\ 0 \\ 0.001466561305710 \\ 0 \end{pmatrix}, \tag{A.37a}$$

$$p = 4, \quad n_f = 6, \quad \nu = 2, \quad \mathrm{Err}_5 \approx 0.0000471, \quad \mathrm{Eff}^{(4)}_{\mathtt{err}} \approx 16.39. \tag{A.37b}$$

`ABABABABA`. Setting $c_1 = c_2 = 0$, we obtain a family of conventional splitting methods of RKN type. The respective minimum-error variant is obtained for

$$\boldsymbol{a} = \begin{pmatrix} 0.178617895844809 \\ -0.066264582669818 \\ 0.775293373650018 \\ -0.066264582669818 \\ 0.178617895844809 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 0.712341831062606 \\ -0.212341831062606 \\ -0.212341831062606 \\ 0.712341831062606 \\ 0 \end{pmatrix}, \quad \boldsymbol{c} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \tag{A.38a}$$

and is characterized by the values

$$p = 4, \quad n_f = 4, \quad \nu = 1, \quad \mathrm{Err}_5 \approx 0.000610, \quad \mathrm{Eff}^{(4)}_{\mathtt{err}} \approx 6.40. \tag{A.38b}$$

Remark A.4. With $s = 5$ stages, we obtain the first Hessian-free force-gradient integrator of order $p = 6$ with `BADADADAB` (A.26). It is the same unique solution as for conventional force-gradient integrators reported in [101], although we have to satisfy an additional order condition due to the approximation of the force-gradient update. Since the additional order condition in general leads to an order reduction for some integrator variants (as we will see in the classification of integrators with $s = 6$ stages), this can be regarded as a lucky coincidence. All other integrator variants in their minimum-error variant are convergent of order $p = 4$. `BADABADAB` (A.30) constitutes a remarkably efficient integrator variant with $\mathrm{Eff}^{(4)}_{\mathtt{err}} \approx 73.45$.

$s = 6$ Stages

With $s = 6$ stages, we obtain additional sixth-order variants. Due to the additional order condition, some variants that achieve order $p = 6$ as conventional force-gradient integrators are limited to order $p = 4$ in their Hessian-free counterparts.

Velocity Versions

With $s = 6$ stages, the velocity versions take the form

$$\begin{aligned} \boldsymbol{\Phi}_h = {} & \boldsymbol{\varphi}^{\{\mathcal{D}\}}_{b_1 h, c_1 h^3} \circ \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{a_2 h} \circ \boldsymbol{\varphi}^{\{\mathcal{D}\}}_{b_2 h, c_2 h^3} \circ \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{a_3 h} \circ \boldsymbol{\varphi}^{\{\mathcal{D}\}}_{(0.5-b_1-b_2)h, c_3 h^3} \circ \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{(1-2a_2-2a_3)h} \\ & \circ \boldsymbol{\varphi}^{\{\mathcal{D}\}}_{(0.5-b_1-b_2)h, c_3 h^3} \circ \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{a_3 h} \circ \boldsymbol{\varphi}^{\{\mathcal{D}\}}_{b_2 h, c_2 h^3} \circ \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{a_2 h} \circ \boldsymbol{\varphi}^{\{\mathcal{D}\}}_{b_1 h, c_1 h^3}, \end{aligned} \tag{A.39}$$

with $a_2, a_3, b_1, b_2, c_1, c_2, c_3 \in \mathbb{R}$ being free parameters.

`BADADADADAB`. We obtain a single real-valued variant of order $p = 6$ via the coefficients

$$\boldsymbol{a} = \begin{pmatrix} 0 \\ 0.270990466773838 \\ 0.635374358266882 \\ -0.812729650081440 \\ 0.635374358266882 \\ 0.270990466773838 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 0.090330155591279 \\ 0.430978044876253 \\ -0.021308200467532 \\ -0.021308200467532 \\ 0.430978044876253 \\ 0.090330155591279 \end{pmatrix},$$
$$\boldsymbol{c} = \begin{pmatrix} 0 \\ 0.002637435980472 \\ -0.000586445610932 \\ -0.000586445610932 \\ 0.002637435980472 \\ 0 \end{pmatrix}, \tag{A.40a}$$

where $c_1 = 0$. Hence, we refer to this variant as `BADADADADAB` and the variant `DADADADADAD` is omitted. This integrator variant is characterized by the values

$$p = 6, \quad n_f = 9, \quad \nu = 0, \quad \mathrm{Err}_7 \approx 0.00000699, \quad \mathrm{Eff}^{(6)}_{\mathtt{err}} \approx 0.27. \tag{A.40b}$$

`DABADADABAD`. Setting $c_2 = 0$ yields

$$\boldsymbol{a} = \begin{pmatrix} 0 \\ 0.068597474282941 \\ 0.284851197274498 \\ 0.293102656885122 \\ 0.284851197274498 \\ 0.068597474282941 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} -0.029456704762871 \\ 0.228751459942521 \\ 0.300705244820350 \\ 0.300705244820350 \\ 0.228751459942521 \\ -0.029456704762871 \end{pmatrix},$$
$$\boldsymbol{c} = \begin{pmatrix} 0.000410146066173 \\ 0 \\ 0.001249935251564 \\ 0.001249935251564 \\ 0 \\ 0.000410146066173 \end{pmatrix}, \tag{A.41a}$$

$$p = 4, \quad n_f = 8, \quad \nu = 4, \quad \mathrm{Err}_5 \approx 0.00000355, \quad \mathrm{Eff}^{(4)}_{\mathtt{err}} \approx 68.84. \tag{A.41b}$$

`DADABABADAD`. Conversely, setting $c_3 = 0$ results in

$$\boldsymbol{a} = \begin{pmatrix} 0 \\ 0.203263079324187 \\ 0.200698071607808 \\ 0.192077698136010 \\ 0.200698071607808 \\ 0.203263079324187 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 0.066202529912271 \\ 0.267856111220228 \\ 0.165941358867501 \\ 0.165941358867501 \\ 0.267856111220228 \\ 0.066202529912271 \end{pmatrix},$$
$$\boldsymbol{c} = \begin{pmatrix} 0.000012570620797 \\ 0.001042408779514 \\ 0 \\ 0 \\ 0.001042408779514 \\ 0.000012570620797 \end{pmatrix}, \tag{A.42a}$$

$$p = 4, \quad n_f = 8, \quad \nu = 4, \quad \mathrm{Err}_5 \approx 0.00000519, \quad \mathrm{Eff}^{(4)}_{\mathtt{err}} \approx 47.08. \tag{A.42b}$$

`BABADADABAB`. For $c_1 = c_2 = 0$, the minimum-error variant is defined by the coefficients

$$\boldsymbol{a} = \begin{pmatrix} 0 \\ 0.122268182901557 \\ 0.203023211433263 \\ 0.349417211330360 \\ 0.203023211433263 \\ 0.122268182901557 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 0.055200549768959 \\ 0.127408150658963 \\ 0.317391299572078 \\ 0.317391299572078 \\ 0.127408150658963 \\ 0.055200549768959 \end{pmatrix},$$
$$\boldsymbol{c} = \begin{pmatrix} 0 \\ 0 \\ 0.001487834491987 \\ 0.001487834491987 \\ 0 \\ 0 \end{pmatrix}, \tag{A.43a}$$

and is characterized by the values

$$p = 4, \quad n_f = 7, \quad \nu = 3, \quad \mathrm{Err}_5 \approx 0.0000189, \quad \mathrm{Eff}^{(4)}_{\mathtt{err}} \approx 21.98. \tag{A.43b}$$

`BADABABADAB`. Setting $c_1 = c_3 = 0$ yields an efficient minimum-error variant with

$$\boldsymbol{a} = \begin{pmatrix} 0 \\ 0.201110227930330 \\ 0.200577842713366 \\ 0.196623858712608 \\ 0.200577842713366 \\ 0.201110227930330 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 0.065692416344302 \\ 0.264163604920340 \\ 0.170143978735358 \\ 0.170143978735358 \\ 0.264163604920340 \\ 0.065692416344302 \end{pmatrix},$$
$$\boldsymbol{c} = \begin{pmatrix} 0 \\ 0.001036943019757 \\ 0 \\ 0 \\ 0.001036943019757 \\ 0 \end{pmatrix}, \tag{A.44a}$$

$$p = 4, \quad n_f = 7, \quad \nu = 3, \quad \mathrm{Err}_5 \approx 0.00000520, \quad \mathrm{Eff}^{(4)}_{\mathtt{err}} \approx 80.13. \tag{A.44b}$$

`DABABABABAD`. For $c_2 = c_3 = 0$, the optimal set of coefficients reads

$$\boldsymbol{a} = \begin{pmatrix} 0 \\ 0.282918304065611 \\ -0.002348009438292 \\ 0.438859410745362 \\ -0.002348009438292 \\ 0.282918304065611 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 0.080181913812571 \\ -1.372969015964262 \\ 1.792787102151691 \\ 1.792787102151691 \\ -1.372969015964262 \\ 0.080181913812571 \end{pmatrix},$$
$$\boldsymbol{c} = \begin{pmatrix} 0.000325098077953 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0.000325098077953 \end{pmatrix}, \tag{A.45a}$$

and results in

$$p = 4, \quad n_f = 6, \quad \nu = 3, \quad \mathrm{Err}_5 \approx 0.0000166, \quad \mathrm{Eff}^{(4)}_{\mathtt{err}} \approx 46.47. \tag{A.45b}$$

`BABABABABAB`. With $c_1 = c_2 = c_3 = 0$, we obtain a fourth-order family of conventional splitting methods with two degrees of freedom. The efficiency measure is maximized for

$$\boldsymbol{a} = \begin{pmatrix} 0 \\ 0.253978510841060 \\ -0.032302867652700 \\ 0.556648713623280 \\ -0.032302867652700 \\ 0.253978510841060 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 0.083983152628767 \\ 0.682236533571909 \\ -0.266219686200676 \\ -0.266219686200676 \\ 0.682236533571909 \\ 0.083983152628767 \end{pmatrix}, \quad \boldsymbol{c} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \tag{A.46a}$$

and yields

$$p = 4, \quad n_f = 5, \quad \nu = 2, \quad \mathrm{Err}_5 \approx 0.0000270, \quad \mathrm{Eff}^{(4)}_{\mathtt{err}} \approx 59.26. \tag{A.46b}$$

Position Versions

The position counterparts with $s = 6$ stages read

$$\begin{aligned} \boldsymbol{\Phi}_h = \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{a_1 h} \circ \boldsymbol{\varphi}^{\{\mathcal{D}\}}_{b_1 h, c_1 h^3} \circ \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{a_2 h} \circ \boldsymbol{\varphi}^{\{\mathcal{D}\}}_{b_2 h, c_2 h^3} \circ \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{(0.5 - a_1 - a_2) h} \circ \boldsymbol{\varphi}^{\{\mathcal{D}\}}_{(1 - 2b_1 - 2b_2) h, c_3 h^3} \\ \circ \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{(0.5 - a_1 - a_2) h} \circ \boldsymbol{\varphi}^{\{\mathcal{D}\}}_{b_2 h, c_2 h^3} \circ \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{a_2 h} \circ \boldsymbol{\varphi}^{\{\mathcal{D}\}}_{b_1 h, c_1 h^3} \circ \boldsymbol{\varphi}^{\{\mathcal{T}\}}_{a_1 h}, \end{aligned} \tag{A.47}$$

where $a_1, a_2, b_1, b_2, c_1, c_2, c_3 \in \mathbb{R}$ denote the degrees of freedom.

`ADADADADADA`. There exist four different real-valued solutions with convergence order $p = 6$. The variant maximizing $\mathrm{Eff}^{(6)}_{\mathtt{err}}$ is

$$\begin{aligned} \boldsymbol{a} &= \begin{pmatrix} 0.109534125980058 \\ 0.426279051773841 \\ -0.035813177753899 \\ -0.035813177753899 \\ 0.426279051773841 \\ 0.109534125980058 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 0.268835839917653 \\ 0.529390037396794 \\ -0.596451754628894 \\ 0.529390037396794 \\ 0.268835839917653 \\ 0 \end{pmatrix}, \\ \boldsymbol{c} &= \begin{pmatrix} 0.000806354602850 \\ 0.007662601517364 \\ -0.011627206142396 \\ 0.007662601517364 \\ 0.000806354602850 \\ 0 \end{pmatrix}, \end{aligned} \tag{A.48a}$$

$$p = 6, \quad n_f = 10, \quad \nu = 0, \quad \mathrm{Err}_7 \approx 0.00000603, \quad \mathrm{Eff}^{(6)}_{\mathtt{err}} \approx 0.17. \tag{A.48b}$$

All four solutions have coefficients $c_i \neq 0$. We will obtain families of fourth-order integrators as soon as we demand at least one c_i to be equal to zero (except for c_1). The case $c_1 = 0$ is omitted since it has a unique solution for $p = 6$ by setting $a_1 = 0$, resulting in the five-stage method `BADADADAB` (A.26).

`ADABADABADA`. Setting $c_2 = 0$, we obtain a family of fourth-order methods with four degrees of freedom. Maximizing the efficiency measure yields

$$\boldsymbol{a} = \begin{pmatrix} 0.083684971641549 \\ 0.225966488946428 \\ 0.190348539412023 \\ 0.190348539412023 \\ 0.225966488946428 \\ 0.083684971641549 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 0.199022868372193 \\ 0.197953981691206 \\ 0.206046299873202 \\ 0.197953981691206 \\ 0.199022868372193 \\ 0 \end{pmatrix},$$
$$\boldsymbol{c} = \begin{pmatrix} 0.000437056543403 \\ 0 \\ 0.000870457820984 \\ 0 \\ 0.000437056543403 \\ 0 \end{pmatrix}, \tag{A.49a}$$

$$p = 4, \quad n_f = 8, \quad \nu = 4, \quad \mathrm{Err}_5 \approx 0.00000318, \quad \mathrm{Eff}^{(4)}_{\mathtt{err}} \approx 76.79. \tag{A.49b}$$

`ADADABADADA`. Similarly, setting $c_3 = 0$ yields fourth-order integrators with four free parameters. Minimizing Err_5 results in

$$\boldsymbol{a} = \begin{pmatrix} 0.082541033171754 \\ 0.228637847036999 \\ 0.188821119791247 \\ 0.188821119791247 \\ 0.228637847036999 \\ 0.082541033171754 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 0.196785139280847 \\ 0.206783248777282 \\ 0.192863223883742 \\ 0.206783248777282 \\ 0.196785139280847 \\ 0 \end{pmatrix},$$
$$\boldsymbol{c} = \begin{pmatrix} 0.000317260402502 \\ 0.000555360763892 \\ 0 \\ 0.000555360763892 \\ 0.000317260402502 \\ 0 \end{pmatrix}, \tag{A.50a}$$

$$p = 4, \quad n_f = 9, \quad \nu = 4, \quad \mathrm{Err}_5 \approx 0.00000235, \quad \mathrm{Eff}^{(4)}_{\mathtt{err}} \approx 64.99. \tag{A.50b}$$

`ABABADABABA`. Decreasing the number of force evaluations by demanding $c_1 = c_2 = 0$ yields

$$\boldsymbol{a} = \begin{pmatrix} 0.134257092137626 \\ -0.007010267216916 \\ 0.372753175079290 \\ 0.372753175079290 \\ -0.007010267216916 \\ 0.134257092137626 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} -0.485681409840328 \\ 0.767464037573892 \\ 0.436434744532872 \\ 0.767464037573892 \\ -0.485681409840328 \\ 0 \end{pmatrix},$$
$$\boldsymbol{c} = \begin{pmatrix} 0 \\ 0 \\ 0.002836723107629 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \tag{A.51a}$$

$$p = 4, \quad n_f = 6, \quad \nu = 3, \quad \mathrm{Err}_5 \approx 0.0000154, \quad \mathrm{Eff}^{(4)}_{\mathtt{err}} \approx 50.09. \tag{A.51b}$$

`ABADABADABA`. Setting $c_1 = c_3 = 0$, we obtain the most efficient fourth-order integrator with $s \leq 6$ stages (according to the efficiency measure $\mathrm{Eff}^{(4)}_{\mathtt{err}}$). The integrator variant is defined by the coefficients

$$\begin{aligned} \boldsymbol{a} &= \begin{pmatrix} 0.062702644098210 \\ 0.193174566017780 \\ 0.244122789884010 \\ 0.244122789884010 \\ 0.193174566017780 \\ 0.062702644098210 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 0.149293739165427 \\ 0.220105234408407 \\ 0.261202052852332 \\ 0.220105234408407 \\ 0.149293739165427 \\ 0 \end{pmatrix}, \\ \boldsymbol{c} &= \begin{pmatrix} 0 \\ 0.000966194415594 \\ 0 \\ 0.000966194415594 \\ 0 \\ 0 \end{pmatrix}, \end{aligned} \tag{A.52a}$$

and is characterized by

$$p = 4, \quad n_f = 7, \quad \nu = 3, \quad \mathrm{Err}_5 \approx 0.00000445, \quad \mathrm{Eff}^{(4)}_{\mathtt{err}} \approx 93.60. \tag{A.52b}$$

`ADABABABADA`. By demanding $c_2 = c_3 = 0$, the optimal solution is obtained for

$$\begin{aligned} \boldsymbol{a} &= \begin{pmatrix} 0.115889910143319 \\ 0.388722377182381 \\ -0.004612287325700 \\ -0.004612287325700 \\ 0.388722377182381 \\ 0.115889910143319 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 0.282498420841510 \\ -0.625616553474143 \\ 1.686236265265266 \\ -0.625616553474143 \\ 0.282498420841510 \\ 0 \end{pmatrix}, \\ \boldsymbol{c} &= \begin{pmatrix} 0.001208219887746 \\ 0 \\ 0 \\ 0 \\ 0.001208219887746 \\ 0 \end{pmatrix}, \end{aligned} \tag{A.53a}$$

and is characterized by

$$p = 4, \quad n_f = 7, \quad \nu = 3, \quad \mathrm{Err}_5 \approx 0.0000128, \quad \mathrm{Eff}^{(4)}_{\mathtt{err}} \approx 32.64. \tag{A.53b}$$

`ABABABABABA`. Setting $c_1 = c_2 = c_3 = 0$ considers position versions of conventional six-stage splitting methods. Its minimum-error variant is obtained for

$$\boldsymbol{a} = \begin{pmatrix} 0.275008121233242 \\ -0.134795009910679 \\ 0.359786888677437 \\ 0.359786888677437 \\ -0.134795009910679 \\ 0.275008121233242 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} -0.084429619507071 \\ 0.354900057157426 \\ 0.459059124699290 \\ 0.354900057157426 \\ -0.084429619507071 \\ 0 \end{pmatrix}, \quad \boldsymbol{c} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \tag{A.54a}$$

and achieves

$$p = 4, \quad n_f = 5, \quad \nu = 2, \quad \mathrm{Err}_5 \approx 0.0000518, \quad \mathrm{Eff}^{(4)}_{\mathtt{err}} \approx 30.89. \tag{A.54b}$$

Remark A.5. For $s = 6$ stages, we obtain two additional integrator variants of order $p = 6$, the velocity version `BADADADADAB` (A.40) with efficiency $\text{Eff}^{(6)}_{\texttt{err}} \approx 0.27$ and the position version `ADADADADADA` (A.48) with $\text{Eff}^{(6)}_{\texttt{err}} \approx 0.17$. Both versions are significantly more efficient than the five-stage method `BADADADAB` (A.26) with $\text{Eff}^{(6)}_{\texttt{err}} \approx 0.0055$: using the same number of total force evaluations, they reduce the global errors approximately by a factor of $0.27/0.0055 \approx 49.09$ and $0.17/0.0055 \approx 30.91$, respectively.

In terms of the efficiency measure $\text{Eff}^{(4)}_{\texttt{err}}$, the position version `ABADABADABA` (A.52) constitutes the most efficient integrator variant among all fourth-order variants with $s \leq 6$ stages ($\text{Eff}^{(4)}_{\texttt{err}} \approx 93.60$). The conventional splitting method `BABABABABAB` (A.46) became a state-of-the-art scheme in lattice QCD simulations [77] due to its high efficiency ($\text{Eff}^{(4)}_{\texttt{err}} \approx 59.26$) compared to methods with fewer stages.

Bibliography

[1] A. Abdel-Rehim et al. "First physics results at the physical pion mass from $N_f = 2$ Wilson twisted mass fermions at maximal twist". In: *Phys. Rev. D* 95.9 (2017), p. 094515. DOI: 10.1103/PhysRevD.95.094515.

[2] C. Alexandrou et al. "Adaptive Aggregation-based Domain Decomposition Multigrid for Twisted Mass Fermions". In: *Phys. Rev. D* 94.11 (2016), p. 114509. DOI: 10.1103/PhysRevD.94.114509.

[3] A. L. Araújo, A. Murua, and J. M. Sanz-Serna. "Symplectic Methods Based on Decompositions". In: *SIAM J. Numer. Anal.* 34.5 (1997), pp. 1926–1947. DOI: 10.1137/S0036142995292128.

[4] W. Auzinger, H. Hofstätter, D. Ketcheson, and O. Koch. "Practical splitting methods for the adaptive integration of nonlinear evolution equations. Part I: Construction of optimized schemes and pairs of schemes". In: *BIT* 57.1 (2017), pp. 55–74. DOI: 10.1007/s10543-016-0626-9.

[5] W. Auzinger, O. Koch, and M. Thalhammer. "Defect-based local error estimators for high-order splitting methods involving three linear operators". In: *Numer. Algorithms* 70.1 (2015), pp. 61–91. DOI: 10.1007/s11075-014-9935-8.

[6] A. Bartel and M. Günther. "Inter/Extrapolation-Based Multirate Schemes: A Dynamic-Iteration Perspective". In: *Progress in Differential-Algebraic Equations II*. Ed. by T. Reis, S. Grundel, and S. Schöps. Cham: Springer International Publishing, 2020, pp. 73–90. DOI: 10.1007/978-3-030-53905-4_3.

[7] A. Bartel et al. "Splitting techniques for DAEs with port-Hamiltonian applications". In: *Appl. Numer. Math.* 214 (2025), pp. 28–53. DOI: 10.1016/j.apnum.2025.03.004.

[8] A. Beskos et al. "Optimal tuning of the hybrid Monte Carlo algorithm". In: *Bernoulli* 19.5A (2013), pp. 1501–1534. DOI: 10.3150/12-BEJ414.

[9] J. J. Biesiadecki and R. D. Skeel. "Dangers of Multiple Time Step Methods". In: *J. Comput. Phys.* 109.2 (1993), pp. 318–328. DOI: 10.1006/jcph.1993.1220.

[10] S. Blanes, F. Casas, and A. Escorihuela-Tomàs. "Runge–Kutta–Nyström symplectic splitting methods of order 8". In: *Appl. Numer. Math.* 182 (2022), pp. 14–27. DOI: 10.1016/j.apnum.2022.07.010.

[11] S. Blanes, F. Casas, C. González, and M. Thalhammer. "Generalisation of splitting methods based on modified potentials to nonlinear evolution equations of parabolic and Schrödinger type". In: *Comput. Phys. Commun.* 295 (2024), p. 109007. DOI: 10.1016/j.cpc.2023.109007.

[12] S. Blanes, F. Casas, and A. Murua. "Composition Methods for Differential Equations with Processing". In: *SIAM J. Sci. Comput.* 27.6 (2006), pp. 1817–1843. DOI: 10.1137/030601223.

[13] S. Blanes and P. Moan. "Practical symplectic partitioned Runge–Kutta and Runge–Kutta–Nyström methods". In: *J. Comput. Appl. Math.* 142.2 (2002), pp. 313–330. DOI: 10.1016/S0377-0427(01)00492-7.

[14] S. Blanes and F. Casas. "On the necessity of negative coefficients for operator splitting schemes of order higher than two". In: *Appl. Numer. Math.* 54.1 (2005), pp. 23–37. DOI: 10.1016/j.apnum.2004.10.005.

[15] S. Blanes, F. Casas, C. Gonzalez, and M. Thalhammer. *Splitting methods with complex coefficients for linear and nonlinear evolution equations*. 2024. arXiv: 2410.13011 [math.NA].

[16] S. Blanes, F. Casas, C. González, and M. Thalhammer. "Efficient Splitting Methods Based on Modified Potentials: Numerical Integration of Linear Parabolic Problems and Imaginary Time Propagation of the Schrödinger Equation". In: *Commun. Comput. Phys.* 33.4 (2023), pp. 937–961. DOI: 10.4208/cicp.OA-2022-0247.

[17] S. Blanes, F. Casas, and A. Murua. "On the Linear Stability of Splitting Methods". In: *Found. Comput. Math.* 8.3 (2008), pp. 357–393. DOI: 10.1007/s10208-007-9007-8.

[18] S. Blanes, F. Casas, and A. Murua. "Splitting methods for differential equations". In: *Acta Numer.* 33 (2024), pp. 1–161. DOI: 10.1017/S0962492923000077.

[19] G. Bogfjellmo and H. Marthinsen. "High-Order Symplectic Partitioned Lie Group Methods". In: *Foundations of Computational Mathematics* 16.2 (2016), pp. 493–530. DOI: 10.1007/s10208-015-9257-9.

[20] N. Bou-Rabee and J. M. Sanz-Serna. "Geometric integrators and the Hamiltonian Monte Carlo method". In: *Acta Numer.* 27 (2018), pp. 113–206. DOI: 10.1017/S0962492917000101.

[21] S. Brooks, A. Gelman, G. L. Jones, and X.-L. Meng, eds. *Handbook of Markov chain Monte Carlo*. Chapman & Hall/CRC Handbooks of Modern Statistical Methods. CRC Press, Boca Raton, FL, 2011, pp. xxvi+592. DOI: 10.1201/b10905.

[22] M. Bruno et al. "Simulation of QCD with Nf = 2 + 1 flavors of non-perturbatively improved Wilson fermions". In: *Journal of High Energy Physics* 2015.2 (2015), p. 43. DOI: 10.1007/JHEP02(2015)043.

[23] C. Budd et al. "Geometric integration using discrete gradients". In: *Philos. Trans. R. Soc. A* 357.1754 (Apr. 1999), pp. 1021–1045. DOI: 10.1098/rsta.1999.0363.

[24] L. Burckhardt et al. "Ahnerts Kalender für Sternfreunde 1994". In: *Astronomische Nachrichten* 314 (1993), p. 398.

[25] J. Butcher. "Order and effective order". In: *Appl. Numer. Math.* 28.2 (1998), pp. 179–191. DOI: 10.1016/S0168-9274(98)00043-9.

[26] F. Castella, P. Chartier, S. Descombes, and G. Vilmart. "Splitting methods with complex times for parabolic equations". In: *BIT* 49.3 (Sept. 2009), pp. 487–508. DOI: 10.1007/s10543-009-0235-y.

[27] E. Celledoni, A. Marthinsen, and B. Owren. "Commutator-free Lie group methods". In: *Future Generation Computer Systems* 19.3 (2003), pp. 341–352.

[28] S. A. Chin. "Symplectic integrators from composite operator factorizations". In: *Phys. Lett. A* 226.6 (1997), pp. 344–348. DOI: 10.1016/S0375-9601(97)00003-0.

[29] S. A. Chin and D. W. Kidwell. "Higher-order force gradient symplectic algorithms". In: *Phys. Rev. E* 62 (6 2000), pp. 8746–8752. DOI: 10.1103/PhysRevE.62.8746.

[30] N. Christian, K. Jansen, K. Nagai, and B. Pollakowski. "Scaling test of fermion actions in the Schwinger model". In: *Nucl. Phys. B* 739.1 (2006), pp. 60–84. DOI: 10.1016/j.nuclphysb.2006.01.029.

[31] M. A. Clark, B. Joó, A. D. Kennedy, and P. J. Silva. "Improving dynamical lattice QCD simulations through integrator tuning using Poisson brackets and a force-gradient integrator". In: *Phys. Rev. D* 84 (7 2011), 071502(R). DOI: 10.1103/PhysRevD.84.071502.

[32] P. E. Crouch and R. Grossman. "Numerical integration of ordinary differential equations on manifolds". In: *J. Nonlinear Sci.* 3.1 (1993), pp. 1–33. DOI: 10.1007/BF02429858.

[33] T. DeGrand and C. DeTar. *Lattice Methods for Quantum Chromodynamics*. World Scientific, 2006.

[34] F. Diele, L. Lopez, and R. Peluso. "The Cayley transform in the numerical solution of unitary differential systems". In: *Adv. Comput. Math.* 8.4 (1998), pp. 317–334. DOI: 10.1023/A:1018908700358.

[35] F. Diele, L. Lopez, and T. Politi. "One step semi-explicit methods based on the Cayley transform for solving isospectral flows". In: *J. Comput. Appl. Math.* 89.2 (1998), pp. 219–223. DOI: 10.1016/S0377-0427(97)00236-7.

[36] S. Duane and J. Kogut. "The theory of hybrid stochastic algorithms". In: *Nucl. Phys. B* 275.3 (1986), pp. 398–420. DOI: 10.1016/0550-3213(86)90606-1.

[37] S. Duane. "Stochastic quantization versus the microcanonical ensemble: Getting the best of both worlds". In: *Nucl. Phys. B* 257 (1985), pp. 652–662. DOI: 10.1016/0550-3213(85)90369-4.

[38] S. Duane, A. Kennedy, B. J. Pendleton, and D. Roweth. "Hybrid Monte Carlo". In: *Phys. Lett. B* 195.2 (1987), pp. 216–222. DOI: 10.1016/0370-2693(87)91197-X.

[39] S. Duane and B. J. Pendleton. "Gauge invariant Fourier acceleration". In: *Phys. Lett. B* 206.1 (1988), pp. 101–106. DOI: 10.1016/0370-2693(88)91270-1.

[40] R. Edwards, I. Horváth, and A. Kennedy. "Instabilities and non-reversibility of molecular dynamics trajectories". In: *Nucl. Phys. B* 484.1 (1997), pp. 375–399. DOI: 10.1016/S0550-3213(96)00618-9.

[41] M. Ehrhardt, M. Günther, and D. Ševčovič. *Structure-Preserving Coupling and Decoupling of Port-Hamiltonian Systems*. 2025. arXiv: 2511.20150 [math.DS].

[42] Y. Fang, J. M. Sanz-Serna, and R. D. Skeel. "Compressible generalized hybrid Monte Carlo". In: *J. Chem. Phys.* 140.17 (May 2014), p. 174108. DOI: 10.1063/1.4874000.

[43] J. Finkenrath. "Review on Algorithms for dynamical fermions". In: *PoS* LATTICE2022 (2023), p. 227. DOI: 10.22323/1.430.0227.

[44] J. Finkenrath, F. Knechtli, and B. Leder. "Corrigendum to "One flavor mass reweighting in lattice QCD" [Nucl. Phys. B 877 (2) (2013) 441–456]". In: *Nucl. Phys. B* 880 (2014), pp. 574–575. DOI: 10.1016/j.nuclphysb.2014.01.019.

[45] E. Forest and R. D. Ruth. "Fourth-order symplectic integration". In: *Phys. D* 43.1 (1990), pp. 105–117. DOI: 10.1016/0167-2789(90)90019-L.

[46] J. M. Foster, G. dos Reis, and C. Strange. "High Order Splitting Methods for SDEs Satisfying a Commutativity Condition". In: *SIAM J. Numer. Anal.* 62.1 (2024), pp. 500–532. DOI: 10.1137/23M155147X.

[47] R. Frezzotti and G. C. Rossi. "Chirally improving Wilson fermions 1. O(a) improvement". In: *JHEP* 08 (2004), p. 007. DOI: 10.1088/1126-6708/2004/08/007.

[48] R. Frezzotti, P. A. Grassi, S. Sint, and P. Weisz. "Lattice QCD with a chirally twisted mass term". In: *JHEP* 08 (2001), p. 058. DOI: 10.1088/1126-6708/2001/08/058.

[49] A. Frommer et al. "Adaptive Aggregation-Based Domain Decomposition Multigrid for the Lattice Wilson–Dirac Operator". In: *SIAM J. Sci. Comput.* 36.4 (2014), A1581–A1608. DOI: 10.1137/130919507.

[50] A. Frommer, M. Günther, B. Liljegren-Sailer, and N. Marheineke. *Operator splitting for port-Hamiltonian systems*. 2023. arXiv: 2304.01766 [math.NA].

[51] C. W. Gear and D. R. Wells. "Multirate linear multistep methods". In: *BIT* 24.4 (1984), pp. 484–502. DOI: 10.1007/BF01934907.

[52] M. Gell-Mann. "The Eightfold Way: A Theory of strong interaction symmetry". In: *The Eightfold Way*. CRC Press, 2018, pp. 11–57. DOI: 10.1201/9780429496615.

[53] M. Girolami and B. Calderhead. "Riemann Manifold Langevin and Hamiltonian Monte Carlo Methods". In: *J. R. Stat. Soc. Ser. B Stat. Methodol.* 73.2 (2011), pp. 123–214. DOI: 10.1111/j.1467-9868.2010.00765.x.

[54] G. H. Golub and C. F. Van Loan. *Matrix Computations - 4th Edition*. Philadelphia, PA: Johns Hopkins University Press, 2013. DOI: 10.1137/1.9781421407944.

[55] O. Gonzalez. "Time integration and discrete Hamiltonian systems". In: *J. Nonlinear Sci.* 6.5 (1996), pp. 449–467. DOI: 10.1007/BF02440162.

[56] W. Gröbner. *Die Lie-Reihen und ihre Anwendungen*. Vol. 3. Mathematische Monographien [Mathematical Monographs]. VEB Deutscher Verlag der Wissenschaften, Berlin, 1960, pp. vii+112.

[57] M. Günther, A. Kværnø, and P. Rentrop. "Multirate Partitioned Runge-Kutta Methods". In: *BIT* 41.3 (2001), pp. 504–514. DOI: 10.1023/A:1021967112503.

[58] M. Günther and A. Sandu. "Multirate generalized additive Runge Kutta methods". In: *Numer. Math.* 133.3 (2016), pp. 497–524. DOI: 10.1007/s00211-015-0756-z.

[59] M. Günther and A. Sandu. *Multirate methods for ordinary differential equations*. 2025. arXiv: 2505.20062 [math.NA].

[60] M. Günther, A. Sandu, K. Schäfers, and A. Zanna. *Symplectic GARK methods for partitioned Hamiltonian systems*. 2023. arXiv: 2103.04110 [math.NA].

[61] E. Hairer, C. Lubich, and G. Wanner. *Geometric numerical integration: structure-preserving algorithms for ordinary differential equations*. Second. Vol. 31. Springer Ser. Comput. Math. Springer-Verlag, Berlin, 2006. DOI: 10.1007/3-540-30666-8.

[62] E. Hairer, R. I. McLachlan, and R. D. Skeel. "On energy conservation of the simplified Takahashi-Imada method". In: *ESAIM Math. Model. Numer. Anal.* 43.4 (2009), pp. 631–644. DOI: 10.1051/m2an/2009019.

[63] Hairer, Ernst and Nørsett, Syvert P. and Wanner, Gerhard. *Solving ordinary differential equations, 1. Nonstiff problems / E. Hairer ; S. P. Nørsett ; G. Wanner*. 2. rev. ed., corr. 3. print. Solving ordinary differential equations; Springer series in computational mathematics. Berlin [u.a.]: Springer, 2008.

[64] E. Hansen and A. Ostermann. "High order splitting methods for analytic semigroups exist". In: *BIT* 49.3 (Sept. 2009), pp. 527–542. DOI: 10.1007/s10543-009-0236-x.

[65] M. Hasenbusch. "Speeding up the hybrid Monte Carlo algorithm for dynamical fermions". In: *Phys. Lett. B* 519.1 (2001), pp. 177–182. DOI: 10.1016/S0370-2693(01)01102-9.

[66] W. K. Hastings. "Monte Carlo sampling methods using Markov chains and their applications". In: *Biometrika* 57.1 (1970), pp. 97–109. DOI: 10.1093/biomet/57.1.97.

[67] M. R. Hestenes, E. Stiefel, et al. "Methods of conjugate gradients for solving linear systems". In: *J. Res. Natl. Bur. Stand.* 49.6 (1952), pp. 409–436. DOI: 10.6028/jres.049.044.

[68] A. Hinrichs, E. Novak, M. Ullrich, and H. Woźniakowski. "The curse of dimensionality for numerical integration of smooth functions". In: *Math. Comp.* 83.290 (2014), pp. 2853–2863. DOI: 10.1090/S0025-5718-2014-02855-X.

[69] A. Iserles. "On Cayley-Transform Methods for the Discretization of Lie-Group Equations". In: *Found. Comput. Math.* 1.2 (2001), pp. 129–160. DOI: 10.1007/s102080010003.

[70] Y. Iwasaki. *Renormalization Group Analysis of Lattice Theories and Improved Lattice Action. II – four-dimensional non-abelian SU(N) gauge model*. 2011. arXiv: 1111.7054 [hep-lat].

[71] B. Joó et al. "Instability in the molecular dynamics step of a hybrid Monte Carlo algorithm in dynamical fermion lattice QCD simulations". In: *Phys. Rev. D* 62 (11 2000), p. 114501. DOI: 10.1103/PhysRevD.62.114501.

[72] C. Jung and N. H. Christ. "Riemannian manifold HMC with fermions". In: *PoS LATTICE2023*. Vol. 453. 2024, p. 009. DOI: 10.22323/1.453.0009.

[73] W. Kahan and R.-C. Li. "Composition constants for raising the orders of unconventional schemes for ordinary differential equations". In: *Math. Comp.* 66.219 (1997), pp. 1089–1099. DOI: 10.1090/S0025-5718-97-00873-9.

[74] N. Kaiser. "Solving the matrix exponential function for the Lie groups SU(3), SU(4) and Sp(2)". In: *Eur. Phys. J. A* 58.9 (2022), p. 170. DOI: 10.1140/epja/s10050-022-00816-5.

[75] A. D. Kennedy, P. J. Silva, and M. A. Clark. "Shadow Hamiltonians, Poisson brackets, and gauge theories". In: *Phys. Rev. D* 87 (3 2013), p. 034511. DOI: 10.1103/PhysRevD.87.034511.

[76] A. D. Kennedy, M. A. Clark, and P. J. Silva. "Force gradient integrators". In: *PoS* LATTICE2009 (2010). DOI: 10.22323/1.091.0021.

[77] F. Knechtli, M. Günther, and M. Peardon. *Lattice quantum chromodynamics: practical essentials*. SpringerBriefs in Physics. Springer Dordrecht, 2017. DOI: 10.1007/978-94-024-0999-4.

[78] F. Knechtli, T. Korzec, M. Peardon, and J. A. Urrea-Niño. "Optimizing creation operators for charmonium spectroscopy on the lattice". In: *Phys. Rev. D* 106 (3 2022), p. 034501. DOI: 10.1103/PhysRevD.106.034501.

[79] F. Knechtli and U. Wolff. "Dynamical fermions as a global correction". In: *Nucl. Phys. B* 663.1 (2003), pp. 3–32. DOI: 10.1016/S0550-3213(03)00369-9.

[80] S. Livingstone, M. Betancourt, S. Byrne, and M. Girolami. "On the geometric ergodicity of Hamiltonian Monte Carlo". In: *Bernoulli* 25.4A (2019), pp. 3109–3138.

[81] L. Lopez and T. Politi. "Applications of the Cayley approach in the numerical solution of matrix differential systems on quadratic groups". In: *Appl. Numer. Math.* 36.1 (2001), pp. 35–55. DOI: 10.1016/S0168-9274(99)00049-5.

[82] M. A. López-Marcos, J. M. Sanz-Serna, and R. D. Skeel. "Explicit Symplectic Integrators Using Hessian–Vector Products". In: *SIAM J. Sci. Comput.* 18.1 (1997), pp. 223–238. DOI: 10.1137/S1064827595288085.

[83] J. Lorenz, T. Zwerschke, M. Günther, and K. Schäfers. "Operator splitting for coupled linear port-Hamiltonian systems". In: *Appl. Math. Lett.* 160 (2025), p. 109309. DOI: 10.1016/j.aml.2024.109309.

[84] M. Luscher. "Computational Strategies in Lattice QCD". In: *Les Houches Summer School: Session 93: Modern perspectives in lattice QCD: Quantum field theory and high performance computing*. Feb. 2010, pp. 331–399. arXiv: 1002.4232 [hep-lat].

[85] M. Lüscher. "Schwarz-preconditioned HMC algorithm for two-flavor lattice QCD". In: *Comput. Phys. Commun.* 165.3 (2005), pp. 199–220. DOI: 10.1016/j.cpc.2004.10.004.

[86] M. Lüscher et al. *openQCD v2.4*. 2022. URL: https://luscher.web.cern.ch/luscher/openQCD/.

[87] P. B. Mackenzie. "An improved hybrid Monte Carlo method". In: *Phys. Lett. B* 226.3 (1989), pp. 369–371. DOI: 10.1016/0370-2693(89)91212-4.

[88] W. Magnus. "On the exponential solution of differential equations for a linear operator". In: *Communications on Pure and Applied Mathematics* 7.4 (1954), pp. 649–673. DOI: 10.1002/cpa.3160070404.

[89] J. E. Marsden and M. West. "Discrete mechanics and variational integrators". In: *Acta Numer.* 10 (2001), pp. 357–514. DOI: 10.1017/S096249290100006X.

[90] P. T. Matthews and A. Salam. "The Green's functions of quantised fields". In: *Il Nuovo Cimento (1943-1954)* 12.4 (1954), pp. 563–565. DOI: 10.1007/BF02781302.

[91] R. I. McLachlan and G. R. W. Quispel. "What kinds of dynamics are there? Lie pseudogroups, dynamical systems and geometric integration". In: *Nonlinearity* 14.6 (2001), p. 1689. DOI: 10.1088/0951-7715/14/6/315.

[92] R. I. McLachlan. "On the Numerical Integration of Ordinary Differential Equations by Symmetric Composition Methods". In: *SIAM J. Sci. Comput.* 16.1 (1995), pp. 151–168. DOI: 10.1137/0916010.

[93] R. I. McLachlan. "Tuning Symplectic Integrators is Easy and Worthwhile". In: *Commun. Comput. Phys.* 31.3 (2022), pp. 987–996. DOI: 10.4208/cicp.OA-2021-0154.

[94] R. I. McLachlan and G. R. W. Quispel. "Splitting methods". In: *Acta Numer.* 11 (2002), pp. 341–434. DOI: 10.1017/S0962492902000053.

[95] N. Metropolis et al. "Equation of State Calculations by Fast Computing Machines". In: *Journal of Chemical Physics* 21.6 (June 1953), pp. 1087–1092. DOI: 10.1063/1.1699114.

[96] C. Moler and C. Van Loan. "Nineteen Dubious Ways to Compute the Exponential of a Matrix, Twenty-Five Years Later". In: *SIAM Review* 45.1 (2003), pp. 3–49. DOI: 10.1137/S00361445024180.

[97] M. Mönch and N. Marheineke. "Commutator-based operator splitting for linear port-Hamiltonian systems". In: *Appl. Numer. Math.* 210 (2025), pp. 25–38. DOI: 10.1016/j.apnum.2024.12.007.

[98] H. Munthe-Kaas. "Runge-Kutta methods on Lie groups". In: *BIT* 38 (1998), pp. 92–111.

[99] S. Ober-Blöbaum, T. Wenger, T. Gail, and S. Leyendecker. *Variational multirate integrators*. 2024. arXiv: 2406.12991 [math.NA].

[100] I. P. Omelyan, I. M. Mryglod, and R. Folk. "Construction of high-order force-gradient algorithms for integration of motion in classical and quantum systems". In: *Phys. Rev. E* 66 (2 2002), p. 026701. DOI: 10.1103/PhysRevE.66.026701.

[101] I. Omelyan, I. Mryglod, and R. Folk. "Symplectic analytically integrable decomposition algorithms: classification, derivation, and application to molecular dynamics, quantum and celestial mechanics simulations". In: *Comput. Phys. Commun.* 151.3 (2003), pp. 272–314. DOI: 10.1016/S0010-4655(02)00754-3.

[102] J. C. Osborn. "Tuning HMC parameters with gradients". In: *PoS LATTICE2023*. Vol. 453. 2024, p. 023. DOI: 10.22323/1.453.0023.

[103] J. Ostmeyer and P. Buividovich. "Minimal autocorrelation in hybrid Monte Carlo simulations using exact Fourier acceleration". In: *Comput. Phys. Commun.* 313 (2025), p. 109624. DOI: 10.1016/j.cpc.2025.109624.

[104] M. Peardon and J. Sexton. "Multiple molecular dynamics time-scales in Hybrid Monte Carlo fermion simulations". In: *Nuclear Physics B - Proceedings Supplements* 119 (2003), pp. 985–987. DOI: 10.1016/S0920-5632(03)01738-9.

[105] P. Pihajoki. "Explicit methods in extended phase space for inseparable Hamiltonian problems". In: *Celest. Mech. Dyn. Astron.* 121 (2015), pp. 211–231. DOI: 10.1007/s10569-014-9597-9.

[106] J. R. Rice. "Split Runge-Kutta method for simultaneous equations". In: *J. Res. Natl. Bur. Stand. B* 64 (1960), p. 151.

[107] R. D. Ruth. "A Canonical Integration Technique". In: *IEEE Trans. Nucl. Sci.* 30.4 (1983), pp. 2669–2671. DOI: 10.1109/TNS.1983.4332919.

[108] Y. Saad and M. H. Schultz. "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems". In: *SIAM J. Sci. Stat. Comput.* 7.3 (1986), pp. 856–869. DOI: 10.1137/0907058.

[109] A. Saccon. "Midpoint rule for variational integrators on Lie groups". In: *International Journal for Numerical Methods in Engineering* 78.11 (2009), pp. 1345–1364. DOI: 10.1002/nme.2541.

[110] A. Sandu and M. Günther. "A Generalized-Structure Approach to Additive Runge–Kutta Methods". In: *SIAM J. Numer. Anal.* 53.1 (2015), pp. 17–42. DOI: 10.1137/130943224.

[111] Sanz-Serna, J. M. and Calvo, M. P. *Numerical Hamiltonian problems*. 1. ed. Applied mathematics and mathematical computation. London [u.a.]: Chapman & Hall, 1994.

[112] K. Schäfers. *pure-gauge_SU3*. Version v1.0. May 2026. DOI: 10.5281/zenodo.20162677.

[113] K. Schäfers, A. Bartel, M. Günther, and C. Hachtel. "Spline-oriented inter/extrapolation-based multirate schemes of higher order". In: *Appl. Math. Lett.* 136 (2023), p. 108464. DOI: 10.1016/j.aml.2022.108464.

[114] K. Schäfers and J. Finkenrath. *openQCD_force-gradient*. Version v1.0.2. Nov. 2024. DOI: 10.5281/zenodo.14187110.

[115] K. Schäfers, J. Finkenrath, M. Günther, and F. Knechtli. "Hessian-free force-gradient integrators". In: *Comput. Phys. Commun.* 309 (2025), p. 109478. DOI: 10.1016/j.cpc.2024.109478.

[116] K. Schäfers, J. Finkenrath, M. Günther, and F. Knechtli. "Hessian-free force-gradient integrators and their application to lattice QCD simulations". In: *PoS* LATTICE2024 (2025), p. 025. DOI: 10.22323/1.466.0025.

[117] K. Schäfers, J. Finkenrath, M. Günther, and F. Knechtli. "Numerical stability of force-gradient integrators and their Hessian-free variants in lattice QCD simulations". In: *Comput. Phys. Commun.* 321 (2026), p. 110034. DOI: 10.1016/j.cpc.2026.110034.

[118] K. Schäfers and M. Günther. *A hierarchical splitting approach for N-split differential equations*. 2026. arXiv: 2601.12878 [math.NA].

[119] K. Schäfers, M. Günther, and A. Sandu. *Symplectic multirate generalized additive Runge–Kutta methods for Hamiltonian systems*. 2023. arXiv: 2306.04389 [math.NA].

[120] K. Schäfers, M. Peardon, and M. Günther. "A modified Cayley transform for SU(3) molecular dynamics simulations". In: *Comput. Phys. Commun.* 315 (2025), p. 109740. DOI: 10.1016/j.cpc.2025.109740.

[121] A. van der Schaft and D. Jeltsema. "Port-Hamiltonian Systems Theory: An Introductory Overview". In: *Found. Trends Syst. Control* 1.2-3 (2014), pp. 173–378. DOI: 10.1561/2600000002.

[122] J. Schwinger. "Gauge Invariance and Mass. II". In: *Phys. Rev.* 128 (5 1962), pp. 2425–2429. DOI: 10.1103/PhysRev.128.2425.

[123] J. Sexton and D. Weingarten. "Hamiltonian evolution for the hybrid Monte Carlo algorithm". In: *Nucl. Phys. B* 380.3 (1992), pp. 665–677. DOI: 10.1016/0550-3213(92)90263-B.

[124] D. Shcherbakov et al. "Adapted Nested Force-Gradient Integrators: The Schwinger Model Case". In: *Commun. Comput. Phys.* 21.4 (2017), pp. 1141–1153. DOI: 10.4208/cicp.OA-2016-0048.

[125] M. Sofroniou and G. Spaletta. "Derivation of symmetric composition constants for symmetric integrators". In: *Optim. Methods Softw.* 20.4-5 (2005), pp. 597–613. DOI: 10.1080/10556780500140664.

[126] R. J. Spiteri, A. Tavassoli, S. Wei, and A. Smolyakov. "Beyond Strang: a Practical Assessment of Some Second-Order 3-Splitting Methods". In: *Commun. Appl. Math. Comput.* 7.1 (Feb. 2025), pp. 95–114. DOI: 10.1007/s42967-023-00314-5.

[127] R. J. Spiteri and S. Wei. *A pair of Second-order complex-valued, N-split operator-splitting methods.* 2024. arXiv: 2407.02677 [math.NA].

[128] G. Strang. "On the construction and comparison of difference schemes". In: *SIAM J. Numer. Anal.* 5.3 (1968), pp. 506–517.

[129] M. Suzuki. "New Scheme of Hybrid Exponential Product Formulas with Applications to Quantum Monte-Carlo Simulations". In: *Computer Simulation Studies in Condensed-Matter Physics VIII.* Ed. by D. P. Landau, K. K. Mon, and H.-B. Schüttler. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995, pp. 169–174. DOI: 10.1007/978-3-642-79991-4_21.

[130] M. Suzuki and K. Umeno. "Higher-Order Decomposition Theory of Exponential Operators and Its Applications to QMC and Nonlinear Dynamics". In: *Computer Simulation Studies in Condensed-Matter Physics VI.* Ed. by D. P. Landau, K. K. Mon, and H.-B. Schüttler. Berlin, Heidelberg: Springer Berlin Heidelberg, 1993, pp. 74–86.

[131] M. Suzuki. "Fractal decomposition of exponential operators with applications to many-body theories and Monte Carlo simulations". In: *Phys. Lett. A* 146.6 (1990), pp. 319–323. DOI: 10.1016/0375-9601(90)90962-N.

[132] M. Suzuki. "Hybrid exponential product formulas for unbounded operators with possible applications to Monte Carlo simulations". In: *Phys. Lett. A* 201.5 (1995), pp. 425–428. DOI: 10.1016/0375-9601(95)00266-6.

[133] T. Takaishi. "Choice of integrator in the hybrid Monte Carlo algorithm". In: *Comput. Phys. Commun.* 133.1 (2000), pp. 6–17. DOI: 10.1016/S0010-4655(00)00161-2.

[134] T. Takaishi and P. de Forcrand. "Testing and tuning symplectic integrators for the hybrid Monte Carlo algorithm in lattice QCD". In: *Phys. Rev. E* 73 (3 2006), p. 036706. DOI: 10.1103/PhysRevE.73.036706.

[135] J. Torsiello et al. "Automated tuning for HMC mass ratios". In: *PoS LATTICE2024.* Vol. 466. 2025, p. 052. DOI: 10.22323/1.466.0052.

[136] H. F. Trotter. "On the Product of Semi-Groups of Operators". In: *Proc. Amer. Math. Soc.* 10.4 (1959), pp. 545–551. DOI: 10.2307/2033649. (Visited on 11/25/2025).

[137] M. Tuckerman, B. J. Berne, and G. J. Martyna. "Reversible multiple time scale molecular dynamics". In: *J. Chem. Phys.* 97.3 (Aug. 1992), pp. 1990–2001. DOI: 10.1063/1.463137.

[138] C. Urbach, K. Jansen, A. Shindler, and U. Wenger. "HMC algorithm with multiple time scale integration and mass preconditioning". In: *Comput. Phys. Commun.* 174.2 (2006), pp. 87–98. DOI: 10.1016/j.cpc.2005.08.006.

[139] C. Urbach et al. *tmLQCD.* Version 5.9.3. URL: https://github.com/etmc/tmLQCD.

[140] V.I. Arnold. *Mathematical Methods of Classical Mechanics.* 2nd ed. 1989. Graduate Texts in Mathematics. New York, NY: Springer New York, Imprint: Springer, 1989. DOI: 10.1007/978-1-4757-2063-1.

[141] L. Verlet. "Computer "Experiments" on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules". In: *Phys. Rev.* 159 (1967), pp. 98–103. DOI: 10.1103/PhysRev.159.98.

[142] M. Wandelt, M. Günther, and M. Muniz. "Geometric integration on Lie groups using the Cayley transform with focus on Lattice QCD". In: *J. Comput. Appl. Math.* 387 (2021), p. 112495. DOI: 10.1016/j.cam.2019.112495.

[143] K. G. Wilson. "Confinement of quarks". In: *Phys. Rev. D* 10 (8 1974), pp. 2445–2459. DOI: 10.1103/PhysRevD.10.2445.

[144] J. Wisdom, M. Holman, and J. Touma. "Symplectic correctors". In: *Integration algorithms and classical mechanics.* Amer. Math. Soc., 1996, pp. 217–244.

[145] H. Yin and R. D. Mawhinney. "Improving DWF Simulations: the Force Gradient Integrator and the Möbius Accelerated DWF Solver". In: *PoS* LATTICE2011 (2012). DOI: 10.22323/1.139.0051.

[146] H. Yoshida. "Construction of higher order symplectic integrators". In: *Phys. Lett. A* 150.5-7 (1990), pp. 262–268. DOI: 10.1016/0375-9601(90)90092-3.

[147] H. Yoshida. "Recent progress in the theory and application of symplectic integrators". In: *Celest. Mech. Dyn. Astron.* 56.1 (1993), pp. 27–43. DOI: 10.1007/BF00699717.

[148] A. Zanna, K. Engø, and H. Z. Munthe-Kaas. "Adjoint and Selfadjoint Lie-group Methods". In: *BIT* 41.2 (2001), pp. 395–421. DOI: 10.1023/A:1021950708869.