Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Mathematical Sciences

Sjur Svorkmo Bergmann

Structure-Preserving Schemes for Stochastic Differential Equations

Adaptation of Hamiltonian Boundary Value Methods to Stochastic Hamiltonian Systems

Master's thesis in Applied Physics and Mathematics
Supervisor: Anne Kværnø
July 2022

NTNU
Kunnskap for en bedre verden

Sjur Svorkmo Bergmann

Structure-Preserving Schemes for Stochastic Differential Equations

Adaptation of Hamiltonian Boundary Value Methods to Stochastic Hamiltonian Systems

Master's thesis in Applied Physics and Mathematics
Supervisor: Anne Kværnø
July 2022

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Mathematical Sciences

NTNU
Norwegian University of
Science and Technology

Abstract

In this thesis, a class of implicit Runge–Kutta (IRK) methods known as Hamiltonian Boundary Value Methods (HBVMs) are considered. These methods are especially suited to solve canonical Hamiltonian systems; with the right choice of parameters, they preserve polynomial Hamiltonians exactly, while for general Hamiltonian systems, they can be made energy-preserving down to machine precision for arbitrary stepsizes. They are based on an extended collocation condition, and can be constructed with many different quadrature bases; here, HBVMs based on Gauß–Legendre or Lobatto are considered. After presenting the background theory on HBVMs, it is demonstrated that any HBVM with s fundamental stages and k total stages have deterministic order $2s$ and Hamiltonian error convergence of order $2k$.

The main goal of the thesis is to adapt HBVMs to Stochastic Differential Equations (SDEs). Limiting the scope to Stochastic Hamiltonian Systems (SHSes) wich can be expressed in terms of a single integrand, it is shown that the HBVM(k,s) has strong or mean square (ms) order of accuracy s, while the Hamiltonian error convergence is of order k in the weak sense. For polynomial stochastic Hamiltonians of degree ν, if the parameters of the Gauß or Lobatto based HBVM satisfy the conservation condition $2k \geq \nu s$, the method is completely energy-preserving.

Several numerical experiments for both deterministic and single integrand Hamiltonian systems are offered to support the theory, namely the quadratic Harmonic Oscillator (called Kubo Oscillator for SDEs), the cubic Hénon–Heiles problem, a polynomial Hamiltonian of order six and the non-polynomial Kepler problem in two dimensions. While the strong error convergence neatly fits in with the theory, for non-conservative choices of the parameter $k > s$, the weak Hamiltonian convergence order is somewhat lower than expected. The methods with conservative choices of k does nevertheless preserve the Hamiltonian exactly.

A chapter is devoted to discussing implementation of an IRK solver for SDEs in `Python`, comparing solvers based on traditional scientific computing ecosystem offered through the `numpy` and `scipy` packages with the relatively new `jax`; the latter turns out to be a lot faster while still offering similar degree of precision.

Sammendrag

I denne oppgaven betraktes en klasse med implisitte Runge–Kutta-metoder (RK-metoder) som kalles Hamiltonian Boundary Value Methods (HBVMs). Disse metodene er spesielt godt egnet til å løse kanoniske Hamiltonske systemer; med riktig valg av parametre bevarer de polynomiske Hamiltoner nøyaktig, mens de kan gjøres energibevarende for generelle Hamiltoner ned til maskinpresisjon for en vilkårlig skrittlengde. De er basert på en utvidet kollokasjonsbetingelse og kan lages med mange forskjellige kvadraturbasiser; her betraktes først og fremst Gauß–Legendre og Lobatto. Etter å ha presentert relevant bakgrunnsteori, demonstreres det at enhver Gauß–HBVM med s grunntrinn og k trinn totalt har deterministisk orden $2s$ og Hamiltonsk feilkonvergens av orden $2k$. For Lobatto-HBVMs gjelder det samme resultatet med et ekstra trinn.

Hovedmålet har vært å tilpasse HBVMs til å kunne brukes på stokastiske differensiallikninger (SDEer). Etter å ha snevret oppgaven oppgaven inn til stokastiske Hamiltonske systemer (SHSer) som lar seg skrives med kun én integrand, blir det vist at HBVM(k, s) har sterk konvergensorden s og Hamiltonsk feilkonvergens i svak forstand av orden k. For polynomiske stokastiske Hamiltoner av grad ν vil metoden være fullstendig energibevarende hvis metodens parametre oppfyller bevaringsbetingelsen $2k \geq \nu s$.

Flere numeriske eksperimenter for både deterministiske og stokastiske Hamiltonske systemer ble gjennomført for å underbygge de teoretiske resultatene: en endimensjonal kvadratisk harmonisk oscillator (kalt Kubo-oscillator for stokastisk variant), det kubiske Hénon–Heiles-problemet, en sjettegrads polynomisk Hamilton, samt det ikke-polynomiske Kepler-problemet i to dimensjoner. Mens den observerte sterke feilkonvergensen passer svært godt overens med den teoretiske ordenen, ga ikke-bevarende valg av $k > s$ noe lavere enn teoretisk svak feilkonvergens for Hamiltonen. Likevel er Hamiltonen nøyaktig bevart for k valgt konservativ.

Et kapittel er også viet til implementasjonen av en IRK-løser for SDEer i `Python`, der løsere basert på det tradisjonelle økosystemet for vitenskapelige beregninger i form av pakkene `numpy` og `scipy` sammenlignes med det relativt nye biblioteket `jax`; det siste viser seg å være mye raskere samtidig som det har en tilsvarende grad av nøyaktighet.

Preface

> I've observed that convenient approximations bring you closest to comprehending the true nature of things.
>
> *Hard-Boiled Wonderland and The End of the World*
> Haruki Murakami

> We have normality. I repeat, we have normality. Anything you still can't cope with is therefore your own problem.
>
> *The Hichhiker's Guide to the Galaxy*
> Douglas Adams

This Master's thesis is a continuation of the my specialization project [1], wich was written in the autumn of 2021 and finished in January 2022. That project dealt mainly with the conservation of quadratic invariants of Stratonovich SDEs as presented by Hong *et al.* [2], but some attention was also paid to Stochastic Hamiltonian Systems with additive noise, for which one can observe a drift in the expectation of the Hamiltonian [3]. As a part of the review process, the energy-preserving methods of Brugnano *et al.* [4] were descried, but were only noted in the final draft.

During the numerical experiments related to the work, some of the methods achieved a higher order of mean-square convergence than was expected for multidimensional noise. It turned out that the test problem (the Kubo Oscillator) allowed a reformulation with a single integrand, for which type of problem numerical methods automatically attain half that of their deterministic order [5]. In this work, the results of Debrabant and Kværnø [5] and the methods of Brugnano *et al.* [4] are combined to present energy-preserving methods for Stochastic Differential Equations of arbitrary high order.

Acknowledgements

This work is dedicated to Håkon Fredheim, who (whether he knows it or not) handed me a way back into the study community, without which I never would have finished this Master's Degree. His high standards and idiosynchratic views are much fun and greatly stimulating.

This thesis would have been impossible without Anne Kværnø with her eminent guidance and invaluable patience, good humour and always supportive attitude; she deserves the greatest of thanks for mentoring me. I might have sought her as a supervisor more for her personal qualities than her field of expertise, but feel greatly honored to have worked with such a brilliant numerical mathematician.

I will also thank my parents, who have both been of great support in many a trying moment, even if they couldn't for the life of theirs understand what I've been doing most of the time. Lastly I would express my gratitude to the community I have found in different musical ensembles of the student societies, which have kept me going when everything else seemed pointless. Though, they must take some of the blame for the long time I have taken getting here.

Contents

Figures

Tables

Nomenclature

$0_{d_1 \times d_2 \times \ldots}, 1_{d_1 \times d_2 \times \ldots}$	Vectors of specified dimension where all entries have the given value; dimensions might be omitted when obvious from context, e.g. zero entries in matrices.
I_d	Identity matrix with rank d.
J	Skew-symmetric structure matrix; see 2.2 for definition.
e_a	Basis unit vector with value 1 at entry a and 0 at all other entries.
Gauss-2s, Lobatto-2s	Gauß(–Legendre) and (Gauß–)Lobatto IIIA collocation methods of deterministic order 2s.
Gauß-HBVM(k,s), Lobatto-HBVM(k,s)	Hamiltonian Boundary Value Method with parameters k,s based on Gauß and Lobatto quadrature, respectively.
$\operatorname{sgn}(\cdot)$	Function $\operatorname{sgn} : \mathbb{F} \to \{-1, 1\}$ returns the sign of the input.
$\frac{\mathrm{d}}{\mathrm{d}t}$	Derivative w.r.t. variable t.
∂_t	Partial derivative w.r.t. some scalar variable t.
$\otimes$	Kronecker product.
$\nabla_x, \nabla_q, \nabla_p$	Gradient w.r.t. x, q or p.
δ_{ij}	Kronecker delta, which is 1 if $i = j$ and 0 otherwise.
$\mathbb{N}$	Set of natural numbers.
$\mathbb{R}, \mathbb{R}^n, \mathbb{R}^{n \times m}$	Set of numbers, n-dimensional vectors and $n \times m$-dimensional matrices with real-valued entries, respectively.
$N(\mu, \sigma^2)$	Normal distribution with mean μ and variance σ^2; note that $N(\mu, \sigma^2) = \mu + \sigma \cdot N(0, 1)$.
$C^n\left(\mathbb{R}^{d_1}, \mathbb{R}^{d_2}\right)$	Set of all functions mapping from $\mathbb{R}^{d_1}$ to $\mathbb{R}^{d_2}$ with continuous derivatives of degree n.
$\mathbb{P}_n(D)$	The set of polynomials of degree less than or equal to n on the domain D.
$W(t)$	Wiener Process (Brownian Motion) at time t; $W(t) \in \mathbb{R}^d$, $d \in \mathbb{N}$.
$X(t), Q(t), P(t)$	Stochastic time-dependent variable.
$\mathrm{d}X(t)$	Symbolic form of stochastic variable X.
$t_0, x_0, q_0, p_0, X_0, Q_0, P_0$	Initial values, capital letter indicating that it is stochastic.
$x(t), q(t), p(t)$	Deterministic time-dependent variables.
$\mathrm{d}\mu(t)$	Single integrand measure.
$\mathrm{d}W(t)$	Symbolic form of Wiener Process.
$\circ \mathrm{d}W(t)$, $\circ \mathrm{d}X(t)$ $\circ\, \mathrm{d}\mu(t)$	Stratonovich sense of stochastic process.
$\Delta t, \Delta W, \Delta \mu, \Delta Z$	Discrete increments of variables t, W, μ and Z.

Chapter 1

Introduction

Change isn't always for the best; this could be the slogan for geometric integration. Conserved quantities often arise in differential equations modelling physical systems, reflecting underlying governing principles such as conservation laws. In many use cases, it might be more important that numerical scheme used for integration conserves these quantities as it is that they are cheap to compute. This holds both for systems of Ordinary Differential Equations (ODEs) and systems of Stochastic Differential Equations (SDEs), although (as is often the case) it might be slightly more difficult for the latter. A special class of problems for which ODE and SDE amounts to about the same, as well as a class of method which works almost equally well on both, will be explored in this thesis.

1.1 Conserving the Hamiltonian

Hamiltonian systems, originally developed as a reformulation of Newtonian mechanics by way of Lagrange, are now found in a wide range of fields. From applications in classical and particularly celestial mechanics, for instance models of planetary and stellar motion [6, Sec. I.2–I.3], Hamiltonian dynamics are now used for models in everything from chemistry and molecular dynamics [6, 7] to economics (see for instance [8]) and other social sciences. A recent example of the latter, in response to the climate crisis, is an attempt to model a sustainable society through a steady state model combining both economic variables (Gross Domestic Product and Complexity, or the level of sophistication of the products in an economy) with ecological variables (Total energy consumption and CO_2 emmisions) [9]. In mechanics, the Hamiltonian function is usually an expression for the total energy of the system, being the sum of kinetic and potential energy [10]. Whence, the error in the Hamiltonian of a system is interchangeably called the energy error.

Hamiltonian functions are a particular instance of *first integrals, invariants* or *constants of motion*. For Poisson systems, which are a generalization of the canonical Hamiltonian system, the class of Casimir funcions (if such exists for the system) are invariant in time. Moreover, as was discovered by Poisson in 1809, the Poisson bracket of two first integrals is again a first integral [6, pp. 256]. As an example consider the fully or completely integrable Kepler problem in $\mathbb{R}^3$, for which the energy, the angular momentum vector, as well as their composition the Runge–Lenz vector, are invariants [7, pp. 46–47].

It has been long known that Hamiltonians systems are closely associated with symplecticity; it was shown by Jacobi in 1837 that symplectic transformations preserve the Hamiltonian character of ODEs [6, pp. 186], and by Poincaré in 1899 that their flow is symplectic [6, pp. 184]. For this reason, numerical integration of Hamiltonian systems have until the last two decades mainly been investigated for symplectic methods, whose timestep approximation constitutes a symplectic mapping (see e.g. [6, 11]). Symplectic methods nearly conserve Hamiltonian for exponentially long times [6, pp. 367], though without care this property might be lost by using adaptive stepsizes [6, Ch. VIII]. In addition, symplectic integrators automatically preserve all quadratic invariants [12]. However, a numerical method cannot be both symplectic and energy-preserving for general Hamiltonian problems [13, 14]. In fact, no Runge–Kutta (RK) method can preserve all polynomial invariants of order higher than two [6, pp. 106].

Despite there not being one method to preserve them all, many numerical methods have been developed or shown to preserve *specific* invariants. A prominent class of such methods are the discrete gradient methods, for which the numerical integrator is expressed through the discretization of the gradient of an invariant or Lyapunov function of the system. With good discretization, such methods will by constrution preserve this specific first integral exactly [15]; examples of these discrete gradient methods include the Averaged Vector Field (AVF) method [16]. Other prominent strategies to preserve first integrals involve projecting numerical solutions onto the manifold defined by the invariant, thoroughly discussed by [6] [6], and methods designed to preserve the path or *line integral* of the invariants, known as Line Integral Methods (LIMs) [17]; this thesis concerns methods of the latter kind.

1.1.1 Hamiltonian Boundary Value Methods

For any polynomial Hamiltonian function of a canonical Hamiltonian system, there exists RK methods which can preserve it exactly [18]; more specificly, the AVF constitutes the energy-preserving RK method of the least number of stages with this property [19]. Iavernaro and Trigiante [20] presented a framework for designing conservative methods based on discretization of the line integral using an *extended collocation condition*. An alternative energy-preserving method framework was given by Hairer [21] based on Lagrange polynomials, which is closely connected with collocation methods. All these different findings can be tied to Hamiltonian Boundary Value Methods (HBVMs),[1] which are methods designed to preserve polynomial Hamiltonians of arbitrary degree exactly [4, 24].

In essence, the Hamiltonian Boundary Value Methods (HBVMs), which with parameters k, s can be formulated as RK methods with k stages and order $2s$, are be derived from the *discrete line integral* of a Hamiltonian system through imposing the extended collocation condition [24]. Different quadrature might be used as a basis, but here, the symmetric Gauß(–Legendre) and Lobatto are considered, as presented by Brugnano *et al.* [4, 25]. These have the advantage of being symmetric methods, and the Gauß collocation-based methods achieve the highest possible order for methods conserving the Hamiltonian with the lowest number of stages [25]. Consequently, they coincide with the AVF method [24].

HBVMs have been written about extensively in an ODE setting (see e.g. [4, 24–30]), as have Line Integral Methods (LIMs) in general.[2] A natural extension of the HBVMs, which only focus on preserving the energy of a canonical Hamiltonian system, is to adapt it to preserve the Hamiltonian of more general Poisson systems, as well as one or several invariants of the ODE. The first item was the focus of [33] [33], whereas Brugnano and Iavernaro [17] provided a framework for preserving invariants. These last methods, called *Generalized HBVMs*, can for methods of order $2s$ and k "nodes" be adapted to achieve a global error of $\mathcal{O}\left(\Delta t^{2k}\right)$ in an *arbitrary* number of invariants, still retaining a block size s.

One of the drawbacks of the HBVMs is that they are not symplectic. Burrage and Burrage [34] created a class of methods in an attempt to redress this, but they lost the favourable energy behaviour of HBVMs (as expected from previously mentioned results on energy-preservation and symplecticity). However, it is possible to perturb the coefficient matrix of Gauß collocation meth-

[1] The Boundary Value Method part of the name is legacy of the eponymous generalization of linear multistep methods, where the multistep formula depends on $k = k_1 + k_2$ boundary conditions partitioned into k_1 intitial and k_2 terminal conditions [22, 23].

[2] The authors Brugnano and Iavernaro together with different collaborators have had a prodigious output on the subject of LIMs over the last fifteen years. Unfortunately, of the papers come off as slightly repetative and might give the impression of language and theory being slightly rushed; moreover, the large number of articles to keep track of can be confusing. Despite the dubious reputation of the publisher (Axioms were rejected by the Norwegian National Board of Scholarly Publishing (NPU) as a scientific publishing channel [32]), their review article [31] from 2018 seems to offer a nice and sussinct introduction of the framework; it has also been cited more than 40 times by researchers other than Brugnano and Iavernaro.

ods in each timestep to ensure that the methods are symplectic and conserve the Hamiltonian function. The size of the optimal perturbation parameter α is such that the method still maintains an error $\mathcal{O}\left(\Delta t^{2s}\right)$ and an energy error $\mathcal{O}\left(\Delta t^{2k}\right)$ in the general case. This has been done by Brugnano *et al.* [35] in constructing the Energy and QUadratic Invariant Preserving Method (EQUIP) methods. Note that to find the optimal perturbation α^* for each timestep, they need perform a few newton iterations, which are somewhat costly.

In the year of writing, Amodio *et al.* [36] unified these frameworks for even more general classes of methods, originally referred to as Poisson Hamiltonian Boundary Vaule Methods (PHBVMs) for the generalized HBVM and Enhanced PHBVMs (EPHBVMs) for a version preserving Casimirs (cf. EQUIP methods).

1.2 Numerical Methods for Stochastic Differential Equations

Numerical schemes for SDEs are often adaptations of schemes for solving ODEs, as the standard deterministic methods rarely work in general [37, pp. XXII]. An effective approach to develop schemes is to truncate Itô–/Stratonovich–Taylor series, also known as Wagner–Platen series expansions, after the necessary number of terms to achieve the desired order of convergence, analogous to Taylor series expansions for ODE schemes [37, pp. XXVI–XXVII].

Error convergence order or order of accuracy of SDE schemes are in general measured in either *weak* or *strong* sense. Strong error measurements compares the method error to an exact solution along each trajectory of processes, usually measured in the mean square (ms) sense (see Definition 2.33). Weak error is concerned with the distribution of the methods, or the "error of the means" [38], possibly transformed by some function from a suitably large class [39, pp. 98] (see Definition 2.34). Numerical schemes are usually developed for either strong or weak approximations, for which they are referred to as weak or strong schemes. Although a certain strong order of accuracy guarantees the method at least the same weak order accuracy [37, pp. 23–24], they can otherwise be quite unrelated [39, pp. 98]; an example will be explored in this thesis.

Due to the advantageous properties of its calculus as well as its colored noise better representing the effects studied, the Stratonovich formulation of SDEs is especially popular in physics [37, Sec. 6.1]. For solving Stratonovich SDEs, Burrage and Burrage constructed [40] and gave order conditions using B-series [41] for a general class of Stochastic Runge–Kutta (SRK) methods for strong approximation, while Komori [42] and Rößler [43] developed a similar Stochastic Runge–Kutta (SRK) framework for scalar and multidimensional Wiener Processes, respectively. Although one can construct such SRK methods of arbitrary high order in theory, they generally involve exponentially more terms than their deterministic counterpart [40].

1.2.1 Geometric Integration and Stochastic Hamiltonian Systems

Stochastic Hamiltonian Systems (SHSes), or Hamiltonian systems in some way perturbed by random fluctuations, arise within many different domains of the natural sciences. It is possible to formulate Stochastic Hamiltonian Systems (SHSes) in the sense of Itô, (see [39, pp. 319]), but the current thesis concerns the Stratonovich SHSes. As for deterministic Hamiltonian systems, the phase flow of SHSes a.s. preserves the symplectic structure; Milstein and Tretyakov [39, Ch. 5] have constructed several schemes especially suited to SHSes, in particular SHSes with separable Hamiltonian and/or additive noise, many of which are symplectic.

As for numerical methods in general, geometric integration of SDEs have also gained traction. For general SHSes, Hong *et al.* [2] give conditions for which SRK methods are symplectic and conserve quadratic invariants, while Anmarkrud and Kværnø [44] demonstrate that these conditions can be used as simplifying conditions during method construction. Discrete gradient approaches to integrate SDEs based on the framework presented by McLachlan *et al.* [15] was taken by Hong *et al.* [45] and Li *et al.* [46]; the latter also constructed linear projection methods, which they proved to be a subclass of the stochastic discrete gradients.

There have also been attempts to apply HBVMs to SDEs. Burrage and Burrage show that for SHSes with general additive noise for scalar [34] and multidimensional [3] Wiener Processes, the (at most quartic) Hamiltonian drifts in expectation; this drift can be preserved by midpoint-based methods inspired by HBVMs. In both papers, the tested scheme amounts to composition methods treating the deterministic and noise term separately, much alike to Strang splitting [1]. However, D'Ambrosio *et al.* [47] shows through perturbation analysis that the drift error of these methods strongly depend on the size of the noise term. Chen *et al.* [48] presents an adaptation inspired by Hairer [21] of order one which is exactly drift-preserving, amending the issues of the previous schemes.

Some investigation of SDEs have been of a class of problems where the drift and diffusion functions are equal up to a constant; Debrabant and Kværnø [5] call such problems single integrand SDEs, as they allow the problem to be rewritten using a single integrand with an adapted measure. These single integrands problems can be solved with standard deterministic RK methods using a randomly perturbed timestep, maintaining half the deterministic error convergence order for both weak and strong order of convergence [5]; as such, methods of arbitrary high order can be cheaply attained, compared to the far more complicated SRK methods of Burrage and Burrage [40]. Cohen *et al.* [49] extends these results to hold for any B-series methods, noting that the geometric properties of the method in a deterministic setting also holds for single integrand problems. In particular, this has been shown to hold for EQUIP methods [50].

1.3 In This Thesis

The specialization project [1], the precursor of this work, limited its scope to energy-preserving methods for quadratic (or, in the case of SHSes with additive noise, quartic) invariants; here, the subject matter is preserving energy for SHSes with higher order polynomial and general Hamiltonians. This thesis is devoted to exploring the construction and energy-preserving properties of the HBVMs, as well as extending their results for ODE applications to Single Integrand Hamiltonian Systems (SIHSes).

1.3.1 Thesis Structure

After the frontmatter, the thesis is structured thus:

- This chapter reviews relevant literature and presents the thesis problem and structure
- Chapter 2 offers a brief introduction to geometric integration and SDE theory
- Chapter 3 is devoted to the HBVMs in a deterministic setting, offering a derivation (Section 3.1), a summary of properties (Section 3.2) and some numerical tests (Section 3.3)
- Chapter 4 concerns Single Integrand Hamiltonian Systems (SIHSes) and the application of HBVMs, presenting features of single integrand problems (Section 4.1), SIHSes (Section 4.2), results for HBVMs applied to SIHSes (Section 4.2.1), and some complementary numerical experiments (Section 4.3)
- Chapter 5, where features of the implementation are investigated
- Chapter 6, where the thesis matter is discussed and further work suggested
- Appendix A and B, consisting of selected HBVM Butcher tables and additional convergence results, respectively
- Backmatter, in the form of Abbreviations and Bibliography

It should be noted that, as this thesis continues the work done in the specialization project [1], many sections of the preliminary Chapter 2 repeat and expand upon material already presented in the previous work. In particular:

- Section 2.1 is based on [1, Section 2.4.1 & 2.4.2], albeit restructured and with more background on Poisson Systems and Poisson brackets;
- Section 2.2.1 is essentially [1, Section 2.4.3] rewritten for non-autonomous equations, while Section 2.2.2 is a slightly expanded version of [1, Section 2.4.4];
- Section 2.4 is almost word for word collected from [1, Section 2.2 & 2.3].

Section 2.3 is new, however, as are the subsequent chapters herein.

Chapter 2

Preliminaries

This thesis demands some knowledge about SDEs and numerical integration to be easily understood. At Norwegian University of Science and Technology (NTNU), the subjects MA8109 - Stochastic Processes and Differential Equations, MA8404 - Numerical Integration of Time Dependent Differenetial Equations, as well as their prerequisites, should give sufficient background knowledge. For the uninitiated reader (or one just looking for a quick recap), background theory is also presented here.

2.1 Hamiltonian, Poisson and Invariant Systems

Within this section, mainly *autonomous* ODEs, i.e. ODEs not explicitly time-dependent, of the format

$$\dot{x} = f(x(t)), \qquad x(t_0) = x_0, \qquad f: U \to \mathbb{R}^d, \quad U \subset \mathbb{R}^d \tag{2.1}$$

although it is quite possible to generalize to non-autonomous ODEs as well. This theory is mainly adapted from Hairer *et al.* [6, Ch.IV & VI], unless stated otherwise.

2.1.1 Hamiltonian Systems

The derivation of the Hamiltonian system is adapted from Hairer *et al.* [6].

The origin of Hamiltonian function comes from a reformulation of the *Lagrangian* of a mechanical system, given in the form

$$L(q,\dot{q}) = T(q,\dot{q}) - U(q), \qquad q(t_0), \dot{q}(t_0) = q_0, \dot{q}_0 \in \mathbb{R}^d, \tag{2.2}$$

where $T(q,\dot{q})$ is an expression for the kinetic energy and $U(q)$ is an expression for the potential energy in said system; q refers to position and $\dot{q}$ its time derivative. To arrive at a Hamiltonian expression for the system, the Lagrangian needs to be regular, i.e. have that the *Legendre transform*

$$p = \phi(q,\dot{q}) = \nabla_q L(q,\dot{q}), \tag{2.3}$$

is bijective (injective and surjective). The definition of the *Hamiltonian* for the system is given by

$$H(q,p) := p^T \dot{q} - L(q,\dot{q}). \tag{2.4}$$

Taking the derivative of (2.4), using (2.3) and the *Euler-Lagrange equations*

$$\nabla_q L(q,\dot{q}) = \frac{\mathrm{d}}{\mathrm{d}t}\left(\nabla_{\dot{q}} L(q,\dot{q})\right), \tag{2.5}$$

this can be restated as a system of equations on the form

$$\dot{q} = \nabla_p H(q,p), \tag{2.6a}$$
$$\dot{p} = -\nabla_q H(q,p). \tag{2.6b}$$

Remark. It is assumed here and after that $H \in C^1(U, \mathbb{R})$, $U \subset \mathbb{R}^{2d}$, i.e. that H is a continuously differentiable function on a subset U of $\mathbb{R}^{2d}$.

Definition 2.1. A *separable* Hamiltonian system is a system of ODEs with a Hamailtonian of the form

$$H(q,p) = p^T M^{-1} p + V(t,q), \quad M \in \mathbb{R}^{d\times d}, \quad V \in C^1\left([t_0,T]\times\mathbb{R}^d, \mathbb{R}\right), \tag{2.7}$$

so called as the p and q components are separable.

Definition 2.2. A Hamiltonian system is said to be *canonical* or of a *canonical form* if the system can be expressed as

$$\dot{x} = J\nabla_x H(x), \tag{2.8}$$

where J is the skew-symmetric *canonical structure matrix* [7] and H is an analytic scalar function i.e.

$$J = \begin{bmatrix} 0 & I_d \\ -I_d & 0 \end{bmatrix}, \qquad H \in C^1\left(\mathbb{R}^{2d}, \mathbb{R}\right). \tag{2.9}$$

Remark. The system of equations arrived at (2.6) can be stated equivalently as a canonical Hamiltonian system by setting $H(q,p) = H(x)$.

Remark. The Hamiltonian function associated with an ODE is often referred to as the *energy* of the system.

Definition 2.3. The *flow of an ODE* from an initial value x_0 at a (fixed) time t, denoted $\varphi_t(x_0)$, is

$$\varphi_t(x_0) = x(t),$$

where x satisfies (2.1) for the initial value x_0.

Remark. Variational equations, arising from $\partial_{x_0}\varphi_t$, are closely tied to symplecticity, yet outside the scope of this thesis.

Definition 2.4. [6, pp. 185] An ODE of the form (2.8) is *locally Hamiltonian* if there exists a neighbourhood for every $x_0 \in U$ where $f(x) = J\nabla_x H(x)$ for some function H.

Symplecticity and the Connection to Hamiltonians

Definition 2.5. [6, pp.183] Let ω be a bilinear mapping $\omega : \mathbb{R}^{2d} \times \mathbb{R}^{2d} \to \mathbb{R}$ such that

$$\omega(\xi,\eta) = \xi^T J \eta, \quad J = \begin{bmatrix} 0 & I_d \\ -I_d & 0 \end{bmatrix}.$$

A differentiable mapping $g : U \to \mathbb{R}^{2d}$ for an open set $U \subset \mathbb{R}^{2d}$ is called *symplectic* if

$$\omega(g'(x)\xi, g'(x)\eta) = \xi^T g'(x)^T J g'(x)\eta = \xi^T J \eta = \omega(\xi,\eta) \quad \forall \xi,\eta \in \mathbb{R}^{2d}. \tag{2.10}$$

Remark. For symplectic linear mappings $g(x) \mapsto Ax$, it follows that

$$\omega(Ax,Ay) = (Ax)^T JAy = x^T A^T JAy = x^T J y^T = \omega(x,y) \qquad \forall x,y \in \mathbb{R}^{2d}.$$

More succinctly, $A^T JA = J$.

Theorem 2.1. *[15, pp.184] Let φ_t be the flow of a Hamiltonian system* (2.8). *Then φ_t is a symplectic mapping.*

Theorem 2.2. *[15, pp.185] An ODE $\dot{x} = f(x)$, $x(t_0) = x_0 \in U$ for $f \in C^1\left(U, \mathbb{R}^{2d}\right)$ is said to be* locally Hamiltonian *if and only if its flow φ_t (see Definition 2.3) is symplectic.*

Remark. In fact, the symplecticity condition is a quadratic invariant of the variational equation of Hamiltonian systems.

2.1.2 Invariants

Definition 2.6. [6] An *invariant*, *first integral*, *conserved quantity* or *constant of motion*[7, Ch. 3.3] is a function $I \in C^1\left(\mathbb{R}^d, \mathbb{R}\right)$ such that for any x satisfying an ODE of the form (2.1), it holds that

$$\frac{\mathrm{d}}{\mathrm{d}t} I(x) = \nabla_x I(x)\dot{x} = \nabla_x I^T f(t,x) = 0 \qquad \forall t \in [t_0, T]. \tag{2.11}$$

From the definition above, it follows that

$$\begin{aligned} \int_{t_0}^{t} \frac{\mathrm{d}}{\mathrm{d}\tau} I(x(\tau))\,\mathrm{d}\tau = I(x(t)) - I(x_0) = 0 \quad \forall t \in [t_0, T] \\ \Leftrightarrow I(x(t)) = I(x_0) \quad \forall t \in [t_0, T]. \end{aligned} \tag{2.12}$$

Linear invariants are of the form

$$I(x) = d^T x, \qquad d \in \mathbb{R}^d, \tag{2.13}$$

whereas quadratic invariants are of the form

$$I(x) = x^T C x, \qquad C \in \mathbb{R}^{d \times d} \text{ s.t. } C^T = C. \tag{2.14}$$

Note that invariants can also be of cubic or higher polynomial order, or not polynomial at all. It is also quite possible that certain problems have several invariants - see e.g. [6, Ch.VI.6.4].

Remark. Taking the derivative of $H(x(t))$ with respect to t leads to the expression

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} H(x(t)) = \nabla_x H(x(t))^T \dot{x} = \nabla_x H(x(t))^T J \,\nabla_x H(x(t)) = 0 \\ \Rightarrow H(x(t)) - H(x_0) = \int_0^t \frac{\mathrm{d}}{\mathrm{d}t} H(x(\tau))\mathrm{d}\tau = 0 \quad \forall t \end{aligned} \tag{2.15}$$

if x satisfies (2.8). From comparing (2.11) and (2.12) with (2.8) and (2.15), it becomes clear that Hamiltonians are invariants of their associated differential equation.

2.1.3 Poisson Systems

A useful tool for the discussion of Hamiltonians and invariants are the so-called Poisson Brackets.

Definition 2.7. A *Poisson bracket* is a bilinear and antisymmetric operation on continuous functions $F, G \in C^1\left([t_0, T] \times \mathbb{R}^d, \mathbb{R}\right)$ given by the equation

$$\{F, G\}(t,x) = \nabla_x G(t,x)^T S(t,x) \nabla_x F(t,x), \tag{2.16}$$

where $S : [t_0, T] \times \mathbb{R}^d \to \mathbb{R}^{2d \times 2d}$ is a skew-symmetric matrix function, i.e.

$$S(t,x)^T = -S(t,x) \quad \forall t \in [t_0, T]. \tag{2.17}$$

Note that the argument are often written implicitly, that is: $\{F, G\}(t,x) = \{F, G\}$.

Remark. Poisson brackets satisfy the Jacobi identity[6, citing Jacobi, 1862]

$$\{\{F, G\}, H\} + \{\{H, F\}, G\} + \{\{G, H\}, F\} = 0, \tag{2.18}$$

as well as Leibniz' rule

$$\{F \cdot G, H\} = F \cdot \{G, H\} + G \cdot \{F, H\}, \tag{2.19}$$

the last one corresponding to standard chain rule of deterministic calculus.

Remark. The symplectic map of Definition 2.5 is a Poisson bracket, as are all other constant structure matrices[7, Ch. 3.3].

Remark. For any ODE with an invariant I it would hold that

$$\{x, I\} = \nabla_x I(x)^T I_d \dot{x} = \nabla_x I(x)^T \dot{x} = 0,$$

where the structure matrix is just the identity matrix I_d. For canonical Hamiltonian systems in particular,

$$\{x, I\} = \{H, I\} = \nabla_x I(x)^T J \nabla_x H(x) = 0,$$

with the structure matrices I_d and J, respectively.

Theorem 2.3. *[6, citing Poisson, 1809] If $I_1(x)$ and $I_2(x)$ are first integrals, so are their Poisson bracket $\{I_1, I_2\}(x)$.*

McLachlan *et al.* [15] offers a reformulation of the system (2.1) with an invariant $I(x)$ to the following:

$$\dot{x} = S(t, x)\nabla_x I, \qquad S : [t_0, T] \times \mathbb{R}^d \to \mathbb{R}^{d\times d}, \quad S^T = -S; \tag{2.20}$$

in other words, the ODE can be expressed by a combination of a skew-symmetric matrix function S and its invariant I. It follows readily that

$$\frac{\mathrm{d}}{\mathrm{d}t} I(x(t)) = \nabla_x I^T \, S \, \nabla_x I = 0 \qquad \forall t \in [t_0, T], \tag{2.21}$$

owing to the skew-symmetry of S.

Given that the first integral doesn't vanish, they give the following form of S defined by the invariant:

$$S = \frac{f \cdot \nabla_x I^T - \nabla_x I \cdot f^T}{\|\nabla_x I\|^2}, \qquad \|\nabla_x I\|^2 = \nabla_x I^T \nabla_x I, \tag{2.22}$$

where f is as in (2.1).

Definition 2.8. Systems of equations which can be written in the form

$$\dot{x} = S(t, x)\, \nabla_x H(t, x), \quad t \in [t_0, T], \quad x(t_0) = x_0 \in \mathbb{R}^d, \tag{2.23}$$

where S is given as in (2.17), are called *Poisson systems*. Note that the function H is still denoted a Hamiltonian [6, Ch. VII.2].

Remark. It is easy to see that Hamiltonian systems are a special case of $2d$-dimensional Poisson systems with $S(t, x) = J$.

Definition 2.9. [6, Ch. VII.2] A function $C(t, x)$ is called a *Casimir function* of a Poisson system if it holds that

$$\nabla_x C(t, x)^T S(t, x) = 0 \quad \forall x. \tag{2.24}$$

Remark. It follows by definition that a Casimir function is a first integral of any Poisson system with structure matrix $S(t, x)$.

2.2 Geometric Integration of Ordinary Differential Equations

In this section, the most important definitions and results regarding structure-preserving properties of RK methods are presented.

2.2.1 Runge–Kutta Methods for ODEs

Definition 2.10. An s-stage Runge–Kutta method applied the ODE

$$\dot{x} = f(t, x(t)), \quad x(t_0) = x_0, \quad t \in [t_0, T],$$

take the form

$$y_i = x_n + \Delta t \sum_{j=1}^{s} a_{ij} f(t_n + c_j \Delta t, y_j), \qquad i = 1, \dots, s, \tag{2.25}$$

$$x_{n+1} = x_n + \Delta t \sum_{i=1}^{s} b_i f(t_n + c_i \Delta t, y_i), \qquad n = 0, \dots, N \tag{2.26}$$

where Δt is an equidistant temporal step-length, with x_0 a given initial value.

Remark. A non-autonomous ODE is used in the definition above to show the explicit presence of the abscissa $\{c_i\}_{i=1}^{s}$.

These methods can be written in the form of a Butcher table (Table 2.1), where $b^T = (b_1, \cdots, b_s)$, $c = A \cdot 1_s$ are s-dimensional vectors and A an $s \times s$-matrix, with each element i, j corresponding to coefficients in the RK-methods. Note that

- Methods with only zero entries on and above the diagonal of A, i.e. $a_{ij} = 0$ for $j \geq i$ are called *explicit*.
- Methods with zero entries above and some non-zero entries on the diagonal of A, i.e. $a_{ij} = 0$ for $j > i$, are called *diagonally implicit*.
- Methods with non-zero entries on and above the diagonal of A are called *implicit*.
- k_i is called *stage i*.
- Methods are called *irreducible* if they do not have equivalent stages.

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array}$$

Table 2.1: Butcher table.

2.2.2 Structure-Preservation and Symplecticity

Here, the main concern are autonomous ODEs, i.e. equations of the form

$$\dot{x} = f(x(t)), \quad x(t_0) = x_0, \quad t \in [t_0, T]. \tag{2.27}$$

Definition 2.11. [51, Ch. IV.3] Consider the test equation

$$\dot{x} = \lambda x(t), \quad t \in [t_0, T], \quad x(t_0) = x_0 \in \mathbb{C}, \quad \lambda \in \mathbb{C}. \tag{2.28}$$

Applying a RK method to (2.28) yields the relation

$$x_1 = R(\lambda \Delta t) x_0, \quad \text{where} \quad R(z) = 1 + z b^T (I_s - zA)^T 1_s = \frac{\det\left(I_s - zA + z 1_s b^T\right)}{\det(I_s - zA)}. \tag{2.29}$$

The function $R(z)$ is called the *stability function* of the RK method.

Definition 2.12. [51, Ch. IV.3] A numerical (one-step) method is *A-stable* if its stability domain contains the whole negative plane, i.e.

$$\mathbb{C}^- = \{z \in \mathbb{C} : \quad \Re(z) < 0\} \quad \subset \quad S = \{z \in \mathbb{C} : \quad |R(z)| \leq 1\}. \tag{2.30}$$

For RK methods with (2.29) as stability function, A-stability is achived if and only if

$$|R(iz)| \leq 1 \quad \forall z \in \mathbb{R}, \quad R(z) \in C^1 \quad \text{for } \Re(z) < 0. \tag{2.31}$$

Definition 2.13. [51, Ch. IV.12] Let

$$M := BA + A^T B - bb^T, \quad B = \operatorname{diag}\{b\}. \tag{2.32}$$

If the coefficients of an s-stage RK method satisfies the conditions

1. $b_i \geq 0 \quad \text{for } i = 1, \ldots, s$ and
2. M is positive semi-definite (psd),

it is said to be *algebraically stable*.

Definition 2.14. [6, pp. 42] The *adjoint method* of a one-step method Φ_h, denoted Φ_h^*, is the inverse map of the original methods with reversed time step, i.e.

$$\Phi_h^* := \Phi_{-h}^{-1}. \tag{2.33}$$

Φ_h is *symmetric* or *time-reversible* if

$$\Phi_h \circ \Phi_{-h} = I \qquad \text{or equivalently} \qquad \Phi_h = \Phi_{-h}^{-1} = \Phi_h^*; \tag{2.34}$$

in other words, the method is its own adjoint.

Theorem 2.4. *[6, pp. 147] The adjoint method of an s-stage RK method is itself an s-stage RK method with coefficients*

$$a_{ij}^* = b_{s+1-j} - a_{s+1-i,s+1-j}, \quad b_i^* = b_{s+1-i} \quad \forall i, j.$$

The RK method is symmetric if

$$a_{s+1-i,s+1-j} + a_{ij} = b_j \quad \forall i, j.$$

Definition 2.15. [6, pp. 187] A numerical one-step method Φ_h is a *symplectic method* if the mapping of the method with

$$x_{n+1} = \Phi_h(x_n)$$

is symplectic when applied to a smooth Hamiltonian system, i.e.

$$\left(\partial_{x_n} x_{n+1}\right)^T J \left(\partial_{x_n} x_{n+1}\right) = \Phi_h'(x_n)^T J \Phi_h(x_n) = J. \tag{2.35}$$

Theorem 2.5. *[6, Thm. VI.4.9] For any RK method, the following statements are equivalent:*

- *The method is symmetric for linear problems $\dot{x} = Lx$.*
- *The methods is symplectic for quadratic Hamiltonian problems $H(x) = \frac{1}{2}x^T Cx$, where C is symmetric.*
- *The stability function satisfies $R(-z)R(z) = 1 \, \forall z \in \mathbb{C}$.*

Theorem 2.6. *[6, Thm. IV.2.2 & Thm. VI.4.3] An s-stage RK method is symplectic and conserves all quadratic invariants of an ODE if*

$$M = 0, \tag{2.36}$$

where M is defined as in (2.32).

Remark. From Theorem 2.6 it should be readily apparent that symplecitc RK methods are also algebraically stable.

Theorem 2.7. *[6, pp. 106] No RK method can conserve all cubic or higher order polynomial invariants.*

Another interesting and related result from Zhong and Marsden [13]:

Theorem 2.8. *Numerical methods exactly preserving energy or other conserved quantities*[1] *beyond quadratic degree of a Hamiltonian ODE cannot be symplectic.*

Remark. Although symplectic methods cannot preserve energy exactly in general, they still have strong energy conservation properties [3, 6, 13]. Nevertheless, this result has motivated research into non-symplectic methods.

Definition 2.16. [51, Proposition IV.3.8] If the coefficients of an s-stage RK method satisfies

$$A^T e_s = b, \tag{2.37}$$

i.e. the last row of A equals the b vector, it is said to be *stiffly accurate*.

2.3 Collocation Methods

A class of structure-preserving Implicit Runge–Kutta (IRK) methods with particular relevance to this thesis, *collocation methods*, are discussed in [51, 52, Ch. II.7 & IV.5]; central features are repeated here.

2.3.1 Legendre Polynomials

An class of funtions with particular importance to collocation methods considered here are the Legendre polynomials. The form and relevant properties of Legendre polynomials, as well as their shifted and scaled verision, are presented in this section.

Definition 2.17. The *Legendre polynomials* $\{\widehat{P}_k\}_{k=0}^{\infty}$ are functions $\widehat{P}_k \in \mathbb{P}_k[-1,1]$ such that

$$\widehat{P}_k(x) = \frac{1}{2^k} \sum_{j=0}^{\lfloor k/2 \rfloor} (-1)^j \binom{k}{j} \binom{2(k-j)}{k} x^{2(k-j)}, \quad k = 0, 1, \ldots, , \tag{2.38}$$

satisfying a recursive formula

$$\widehat{P}_0(x) = 1, \quad \widehat{P}_1(x) = x, \quad \widehat{P}_{k+1}(x) = \frac{2k+1}{k+1}\, x\, \widehat{P}_k(x) - \frac{k}{k+1}\widehat{P}_{k-1}(x), \quad k = 1, 2, \ldots \tag{2.39}$$

Definition 2.18. The *shifted* Legendre polynomials typically refers to a shift of (2.38) to $[0,1]$, i.e. $\widetilde{P}_k(x) = \widetilde{P}_k(2x-1)$, leading to the reformulation

$$\widetilde{P}_k(x) = \frac{1}{k!} \frac{d^k}{dx^k} \left(x^k (x-1)^k \right) = \sum_{j}^{k} (-1)^{j+k} \binom{k}{j} \binom{j+k}{j} x^j, \tag{2.40}$$

with the recursive form

$$\widetilde{P}_0(x) = 1, \quad \widetilde{P}_1(x) = 2x-1, \quad \widetilde{P}_{k+1}(x) = \frac{2k+1}{k+1}(2x-1)\widetilde{P}_k(x) - \widetilde{P}_{k-1}(x), \quad k = 1, 2, \ldots \tag{2.41}$$

Brugnano *et al.* lists the following properties of the shifted Legendre polynomials in [4].

1. Symmetry:
$$\widetilde{P}_i(1-x) = (-1)^i \widetilde{P}_i(x), \quad i = 0, 1, \ldots \tag{2.42a}$$
2. Endpoint symmetry:
$$\widetilde{P}_i(0) = (-1)^i, \quad \widetilde{P}_i(1) = 1, \quad i = 0, 1, \ldots \tag{2.42b}$$

[1] Here, only invariants are considered

3. Properties of the integral:

$$\begin{aligned} 2\int_0^x \widetilde{P}_0(\theta)\mathrm{d}\theta &= 2x = \widetilde{P}_1(x) + \widetilde{P}_0(x), \\ 2(2k+1)\int_0^x \widetilde{P}_k(\theta)\mathrm{d}\theta &= \widetilde{P}_{k+1}(x) - \widetilde{P}_{k-1}(x), \quad k = 1,2,\dots \end{aligned} \tag{2.42c}$$

4. Property of the differential equation:

$$\frac{\mathrm{d}}{\mathrm{d}x}\left((x^2-x)\widetilde{P}_k'(x)\right) + k(k+1)\widetilde{P}_k(x) = 0, \quad k = 0,1,\dots \tag{2.42d}$$

Legendre polynomials are members of the Hilbert space $L^2[a,b]$ for $[a,b] \in \{[-1,1],[0,1]\}$, with the inner product

$$< f, g >_{L^2[a,b]} := \int_a^b f(x)g(x)\mathrm{d}x \tag{2.43}$$

and norm

$$\|f\|_{L^2[a,b]} := \left(\int_a^b f^2(x)\mathrm{d}x\right)^{\frac{1}{2}}. \tag{2.44}$$

In the ODE reference literature (e.g. [51]), the shifted Legendre polynomials are often scaled or normalized.

Definition 2.19. The k-th *shifted and scaled Legendre Polynomial* $P_k \in \mathbb{P}_k[0,1]$ is given as

$$P_k(x) = \frac{\sqrt{2k+1}}{k!}\frac{d^k}{dx^k}\left(x^k(x-1)^k\right) = \sqrt{2k+1}\sum_{j=0}^{n}(-1)^{j+k}\binom{k}{j}\binom{j+k}{j}x^j \tag{2.45}$$

and satisfies

$$< P_k, P_j >_{L^2[0,1]} = \delta_{kj}, \quad j,k = 0,1,\dots \tag{2.46}$$

Analogously to the above and as noted in [51, Ch. IV.5], the shifted and scaled Legendre polynomials satisfy

$$\begin{aligned} \int_0^x P_0(\theta)\mathrm{d}\theta &= \xi_1 P_1(x) + \frac{1}{2}P_0(x), \\ \int_0^x P_k(\theta)\mathrm{d}\theta &= \xi_{k+1}P_{k+1}(x) - \xi_k P_{k-1}(x), \quad \text{for } k = 1,2,\dots \end{aligned} \tag{2.47}$$

with

$$\xi_k = \frac{1}{2\sqrt{4k^2-1}}. \tag{2.48}$$

The rest of the properties (2.42) can be adapted for the normalized version with suitable scaling.

All the aforementioned Legendre polynomials are orthogonal faimilies, in that

$$< P_k, P_j >_{L^2[a,b]} = \delta_{kj} \cdot C_k, \quad j,k = 0,1\cdots,$$

where

$$C_k = \begin{cases} \frac{1}{k+1/2} & \text{for Legendre polynomials on } [-1,1], \\ \frac{1}{2k+1} & \text{for shifted Legendre polynomials on } [0,1], \\ 1 & \text{for shifted and normalized Legendre polynomials.} \end{cases} \tag{2.49}$$

Therefore, they form an orthogonal basis on their respective spaces $L^2[-1,1]$ and $L^2[0,1]$, with the shifted and scaled version also orthonormal.

Definition 2.20. The *Legendre series* of $f \in L^2[-1,1]$ is given as [53, Ch. 10.1]

$$Lf = \sum_{k=0}^{\infty} \widehat{f}_k \widehat{P}_k, \quad \widehat{f}_k = \left(k + \frac{1}{2}\right) \int_a^b f(x)\widehat{P}_k(x)\mathrm{d}x. \tag{2.50}$$

Remark. One arrives at the shifted Legendre series if one replaces $[-1,1]$, $\widehat{P}_k(x)$ and $\left(k+\frac{1}{2}\right)$ with $[0,1]$, $\widetilde{P}_k(x)$ and $(2k+1)$.

Remark. Using the shifted and scaled Legendre polynomial in the Legendre series only leads to a redefinition of f using an orthonormal basis [54, Appendix D]; that is,

$$f(x) = \sum_{k=1}^{\infty} < f, P_k >_{L^2[0,1]} P_k(x) = \sum_{k=1}^{\infty} P_k(x) \int_0^1 P_k(x) f(x)\mathrm{d}x. \tag{2.51}$$

2.3.2 Derivation of Methods

Collocation methods gets their name from the fact that they satisfy the *collocation condition*:

Definition 2.21. Consider a general ODE of the form

$$\dot{x} = f(t,x), \quad t \in [t_0, T], \quad x(t_0) = x_0 \in \mathbb{R}^d. \tag{2.52}$$

A polynomial $u \in \mathbb{P}_s$ such that $u(0) = x(t_0)$ satisfies the *collocation condition* if

$$u'(c_i) = f(\tau_i, u(c_i)), \quad \tau_i = t_0 + c_i \Delta t, \tag{2.53}$$

for s points $0 \leq c_1 < \cdots < c_s \leq 1$.

There exist exactly one polynomial of order $s-1$ which interpolates s arbitrary disctinct abscissae, and when these are known, said polynomial can be written in terms of Lagrange polynomials. Thus, letting $\tau(\theta) = t_0 + \theta \Delta t$ and $\tau_j = t_0 + c_j \Delta t$, one can write

$$u'(\theta) = \sum_{j=1}^{s} f(\tau_j, u(c_j))\ell_j(\theta), \quad \ell_j(\theta) = \prod_{\substack{k=1 \\ k \neq j}}^{s} \frac{\theta - c_k}{c_j - c_k}. \tag{2.54}$$

It follows that

$$u(c) - u(0) = \int_{t_0}^{t_0 + c\Delta t} u'(\theta(\tau))\mathrm{d}\tau = \Delta t \int_0^c \sum_{j=1}^{s} f(\tau_j, u(c_j))\ell_j(\theta)\mathrm{d}\theta = \Delta t \sum_{j=1}^{s} f(\tau_j, u(c_j)) \int_0^c \ell_j(\theta)\mathrm{d}\theta$$

$$\Rightarrow \quad u(c) = x_0 + \Delta t \sum_{j=1}^{s} f(\tau_j, u(c_j)) \int_0^c \ell_j(\theta)\mathrm{d}\theta.$$

To arrive at a standard RK formulation of a collocation method, one simply sets

$$a_{ij} = \int_0^{c_i} \ell_j(\theta)\mathrm{d}\theta, \quad b_i = \int_0^1 \ell_i(\theta)\mathrm{d}\theta, \quad \text{for } i,j = 1,\dots,s. \tag{2.55}$$

Then, by setting $y_i = u(c_i)$ for $i = 1,\dots,s$ and letting $x(t_0 + \Delta t) \approx u(1)$, it follows that

$$\begin{aligned} y_i = u(c_i) = u(0) + \Delta t \sum_{j=1}^{s} \left(\int_0^{c_i} \ell_j(\theta)\mathrm{d}\theta \right) f(\tau_j, u(c_j)) \\ = x_0 + \Delta t \sum_{j=1}^{s} a_{ij} f(\tau_j, y_j) \qquad & \text{for } i = 1,\dots,s; \\ x_1 = u(1) = u(0) + \Delta t \sum_{i=1}^{s} \left(\int_0^{1} \ell_i(\theta)\mathrm{d}\theta \right) f(\tau_j, u(c_i)) \\ = x_0 + \Delta t \sum_{i=1}^{s} b_i f(\tau_i, y_i). \end{aligned}$$

The same procedure is then applied for subsequent steps of the method with only the initial value adjusted, i.e. $x_0 \rightsquigarrow x_n$ and $x_1 \rightsquigarrow x_{n+1}$ for $n = 1,\dots,N-1$, where $N = (T - t_0)/\Delta t$.

The most defining feature of the collocation method is the way in which one arrives at the abscissae $c = \{c_i\}_{i=1}^{n}$. The main classes of methods use the roots of

$$\frac{\mathrm{d}^s}{\mathrm{d}x^s}\left(x^s(x-1)^s\right) \tag{2.56a}$$

$$\frac{\mathrm{d}^{s-1}}{\mathrm{d}x^{s-1}}\left(x^s(x-1)^{s-1}\right) \tag{2.56b}$$

$$\frac{\mathrm{d}^{s-1}}{\mathrm{d}x^{s-1}}\left(x^{s-1}(x-1)^s\right) \tag{2.56c}$$

$$\frac{\mathrm{d}^{s-2}}{\mathrm{d}x^{s-2}}\left(x^{s-1}(x-1)^{s-1}\right) \tag{2.56d}$$

leading to Gauß-Legendre (2.56a), Radau (2.56b and 2.56c) and Lobatto (2.56d) quadrature, respectively [51, Ch. IV.5].

Remark. The roots of (2.56a) are also the roots of shifted Legendre polynomial $\widetilde{P}_s(x)$, whereas (2.56d) can alternatively stated as $(x^2 - x)\widetilde{P}_s{}'(x)$ [4]. For Lobatto abscissae $\{c_i\}_{i=0}^{s}$, Brugnano *et al.* notes in [4] that

$$\int_0^{c_i} \widetilde{P}_s(x)\mathrm{d}x = 0, \quad i = 0,1,\dots,s. \tag{2.57}$$

Remark. It is common to denote methods as type I (2.56b), type II (2.56c) and type III (2.56d) in addition to the name of the quadrature applied. This refers to the included endpoints: Radau type I (Radau left) has a root at zero, Radau type II (Radau right) at 1 and Lobatto type III at both 0 and 1.

In this thesis, the interest is mainly in the Gauß and Lobatto IIIA methods, as they have roots symmetrically placed on the interval [0,1]. Gauß-Legendre methods are consequently referred to as Gauß methods hereafter, while in later chapters Lobatto methods denotes the Lobatto IIIA methods, unless otherwise specified.

2.3.3 Important properties

In this section, important properties applying to the methods studied in this thesis are gathered. Here, it is assumed that coefficients are gathered as in Definition 2.22, where b and c are column vectors. Three simplifying conditions are usually applied[51, Ch. IV.5]:

Definition 2.22. For s-stage collocation methods with coefficients $A = \{a_{ij}\}_{i,j=1}^{s}$, $b = \{b_i\}_{i=1}^{s}$ and $c = \{c_i\}_{i=1}^{s}$, the *simplifying conditions* $B(p)$, $C(\eta)$ and $D(\zeta)$ are given as follows.

$$\begin{aligned}
B(p): &\quad \sum_{i=1}^{s} b_i c_i^{q-1} = \frac{1}{q}, && q = 1,\dots,p;\\
C(\eta): &\quad \sum_{j=1}^{s} a_{ij} c_j^{q-1} = \frac{c_i^q}{q}, && i = 1,\dots,s, \quad q = 1,\dots,\eta;\\
D(\zeta): &\quad \sum_{i=1}^{s} b_i c_i^{q-1} a_{ij} = \frac{b_j}{q}\left(1 - c_j^q\right), && j = 1,\dots,s, \quad q = 1,\dots,\zeta.
\end{aligned}$$

Remark. $B(p)$ essentially means the quadrature formula for (b_i, c_i) is of order p [51, Ch. IV.5]. Consequently, $s$stage collocations methods derived from (2.56a) and (2.56d) satisfy $B(2s)$ and $B(2s-2)$, respectively.

Lemma 2.9. *[51, Lemma IV.5.4] For an RK method with distinct abscissae and non-zero weights, it holds that*

1. $$C(s), B(s+\nu) \implies D(\nu),$$
2. $$D(s), B(s+\nu) \implies C(\nu).$$

Theorem 2.10. *[51, Thm. IV.5.1] If the coefficients of an RK method satisfies $B(p)$, $C(\eta)$ and $D(\zeta)$ with $p \le \eta + \zeta + 1$ and $p \le 2\eta + 2$, then the method is of order p.*

Remark. s-stage Gauß methods satisfy conditions $B(2s)$, $C(s)$ and $D(s)$, and is consequently of order $2s$. s-stage Lobatto methods' coefficients a_{ij} are derived by imposing $C(s)$ while the quadrature satisfes $B(2s-2)$, from which $D(s-2)$ follows; they are consequently of order $2s-2$.[2]

Lemma 2.11. *[6, Thm. V.2.1] A collocation method of is symmetric if its quadrature rule has symmetric roots, i.e.*

$$c_{s+1-i} = 1 - c_i \quad \textit{for } i = 1,\dots,s. \tag{2.58}$$

Selected Properties

- Gauß methods have the highest achievable order for an s-stage RK method [6, Ch. II.1.3].
- By [51, Corollary IV.12.15] and (2.56), the B-stability and algebraic stability property coincides for collocation methods.
- Gauß methods are symplectic, whereas Lobatto IIIA are not.
- From Lemma 2.11, Gauß and Lobatto IIIA methods are symmetric.
- Lobatto IIIA methods are stiffly accurate, whereas Gauß methods are not.

2.3.4 Construction of Methods

As mentioned before, for s-stage collocation methods, the entries of c are set to the roots of (2.56). For collocation methods, the quadrature weights b satisfy $B(s)$, which in matrix form becomes

$$V^T b = a, \tag{2.59}$$

where V, called the Vandermonde matrix [51, pp. 78], and a has the form

$$V := \begin{bmatrix} 1 & c_1 & \cdots & c_1^{s-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & c_s & \cdots & c_s^{s-1} \end{bmatrix} \in \mathbb{R}^{s\times s}, \quad a = \left[\frac{1}{q}\right]_{q=1}^{s}. \tag{2.60}$$

[2] For a summary of properties of all the main collocation methods, the reader is referred to [51, chIV.5, Table 5.13].

(2.59) is a full rank linear system; consequently, b can be found simply by solving it. However, explicit expressions for the weights of the Gauß and Lobatto quadrature take the form [see e.g. 53, Ch. 10.4]

$$b_i = \begin{cases} \dfrac{1}{c_i(1-c_i)\left(\widetilde{P}_s'(c_i)\right)^2} & \text{using Gauß nodes,} \\ \dfrac{1}{s(s+1)(\widetilde{P}_s(c_i))^2} & \text{using Lobatto nodes,} \end{cases} \qquad \text{for } i = 1,\dots,s, \tag{2.61}$$

where $\widetilde{P}_j$ are the shifted (but not scaled) Legendre polynomials (see Appendix 2.3.1). To find the matrix A, some other intermediary quantities should be introduced.

Let W be an $s \times s$-matrix with entries such that

$$w_{ij} = P_{j-1}(c_i), \quad i = 1,\dots,s, \quad j = 1,\dots,s, \tag{2.62}$$

where $\{P_{j-1}\}_{j=1}^{s}$ are the orthonormal (i.e. shifted and scaled) Legendre polynomials up to degree s. For methods based on the quadrature (2.56), it holds that

$$\mathcal{X} := W^{-1}AW = \begin{bmatrix} \xi_0 & -\xi_1 & & \\ \xi_1 & 0 & \ddots & \\ & \ddots & \ddots & \beta_{s-1} \\ 0 & & \xi_{s-1} & \beta_s \end{bmatrix}, \quad \xi_k = \frac{1}{2\sqrt{|4k^2-1|}}, \quad k = 0,\dots,s-1; \tag{2.63}$$

where β_{s-1}, β_s are method class specific parameters[51, Example IV.5.16]. For Gauß methods, these are set to $-\xi_{s-1}$ and zero, respectively, leading to the matrix

$$\mathcal{X}_G = \begin{bmatrix} \xi_0 & -\xi_1 & & \\ \xi_1 & 0 & \ddots & \\ & \ddots & \ddots & -\xi_{s-1} \\ 0 & & \xi_{s-1} & 0 \end{bmatrix}. \tag{2.64}$$

Definition 2.23. [51, Definition IV.5.10] An $s \times s$-matrix W satisfies the $T(\eta,\zeta)$-*condition* for the quadrature formula (b,c) if

1. W is nonsingular
2. $w_{ij} = P_{j-1}(c_i)$ for $i = 1,\dots,s$ and $j = 1,\dots,(\max\{\eta,\zeta\}+1)$
3. $$W^T BW = \begin{bmatrix} I_{\zeta+1} & 0 \\ 0 & R \end{bmatrix} \quad \text{with } R \in \mathbb{R}^{s-\zeta-1 \times s-\zeta-1} \text{ arbitrary,}$$

where η, ζ are integers such that $0 \le \eta, \zeta \le s-1$.

Theorem 2.12. *[51, Thm. IV.5.11] Let W satisfy $T(\eta,\zeta)$ for some quadrature formula (b,c). Then, for the RK method based on the quadrature and with the matrix $\mathcal{X} := W^{-1}AW$, it holds that*

1. *$C(\eta)$ $\Leftrightarrow$ the first η columns of $\mathcal{X}$ equals those of $\mathcal{X}_G$*
2. *$D(\zeta)$ $\Leftrightarrow$ the first ζ rows of $\mathcal{X}$ equals those of $\mathcal{X}_G$*

Remark. This theorem holds for much more general $\mathcal{X}$-matrices than (2.63), as long as the RK method is quadrature based.

An important (and quite restrictive) result:

Lemma 2.13. *[51, Lemma IV.5.9] For any quadrature formula of order greater or equal to $2s-1$, the matrix W as given in* (2.62) *satisfies*

$$W^T B = W^{-1} \qquad \Leftrightarrow \qquad W^T BW = I_s. \tag{2.65}$$

Lastly, let

$$M(x) := \prod_{i=1}^{s}(x - c_i) = C\left(P_s(x) + \alpha_1 P_{s-1}(x) + \alpha_2 P_{s-2}(x)\right). \tag{2.66}$$

Note that by setting $(\alpha_1, \alpha_2, \beta_s, \beta_{s-1}) = (0, 0, 0, 1)$, one arrives at the Gauß methods, whereas by setting $(\alpha_1, \alpha_2, \beta_s, \beta_{s-1}) = (0, -\sqrt{\frac{2s+1}{2s-3}}, 0, 0)$ one arrives at the Lobatto methods.

2.4 Stochastic Differential Equations

This section is almost in its entirety retrieved from the preliminaries of the specialization project [1] with only minor corrections and additions.

Stochastic calculus is an attempt to analytically study systems, physical or otherwise, which cannot be exactly expressed through ODEs due to being perturbed by random fluctuations or noise. Although attempts to treat such noisy ODEs was attempted before this, usually by circumventing the theoretical foundation, the first "proper" formalism was introduced by Itô [55] in 1944. Stratonovich offered an alternative formulation in 1966 [56], which in contrast to Itô's framework closely resembles the deterministic calculus, yet interprets noise in a slightly different way. Both are accurate interpretations of noise, but for slightly different systems. Stratonovich calculus is often more accurate when working with systems with "colored" noise, which often arise in physics [39], while Itô calculus aptly represent "white" or unstructured noise, better describing phenomena of biology, population statistics and financial markets [57].

Usually, SDEs are written in one of two equivalent ways:

$$\mathrm{d}X(t) = b(t,X(t))\mathrm{d}t + \sigma(t,X(t)) * \mathrm{d}W(t), \quad t \in [t_0, T], \quad X(t_0) = X_0 \in \mathbb{R}^d, \tag{2.67a}$$

and

$$X(t) = X_0 + \int_{t_0}^{t} b(\tau,X(\tau))\mathrm{d}\tau + \int_{t_0}^{t} \sigma(\tau,X(\tau)) * \mathrm{d}W(\tau), \tag{2.67b}$$

with $W(t)$ being an m-dimensional Wiener Process and $*$ replaced with " " when referring to Itô SDEs and "$\circ$" when referring to Stratonovich SDEs. All the involved quantities are explained in this section. To shortly summarize the most commonly used associated phrases:

- (2.67a) is the symbolic form of writing an SDE, whereas (2.67b) is the integral form and "proper" way to write SDEs
- $b(t,X(t))$ is called the drift term and is the deterministic part of the equation, whereas $\sigma(t,X(t))$ is called the diffusion term and is the stochastic or "noisy" part of the equation
- when σ is independent of $X(t)$, i.e. $\sigma(t,X) = \sigma(t)$, one refers to the system as an SDE with additive noise, while for $\sigma(t,X(t))$'s explicitly depending on $X(t)$, it is referred to as an SDE with multiplicative noise

The following definitions and results are necessary to work with SDEs, and follows the notation of Øksendal [57] and Kuo [58]. In this thesis, when concerned with SDEs, operating in complete probability spaces $\{\Omega, \mathcal{F}, \mathcal{F}(t), P\}$ is considered a given. As is convention, when considering stochastic processes $X(t,\omega) : [t_0, T] \times \Omega \to \mathbb{R}^d$, the notation $X(t)$ with *sample path* ω implicit is often applied. For an introduction to the foundations of probability theory, the reader is referred tothe aforementioned sources.

2.4.1 Wiener Processes and Stochastic Processes

Essential to the mathematical foundation of SDEs is the concept of the Wiener Processes.[3]

Definition 2.24. [58] A function $W(t)$ for $t \in [t_0, T]$ is a *Wiener Process* if it has

1. $W(0) = 0 \quad$ a.s. (Zero initial value)
2. $W(t) - W(s) \sim N(0, t-s)$ (Normally distributed increments)
3. $W(t_2) - W(t_1) \perp W(s_2) - W(s_1)$ for $0 \leq s_1 < s_2 < t_1 < t_2 \leq T$ (Independent increments)
4. $\lim_{t \to s} W(t, \omega) - W(s, \omega) = 0 \quad \forall \omega \in \Omega$ a.s. (Continuous sample paths)

This can be generalized to an m dimensions with

$$W(t) = \{W_1(t), W_2(t), \dots, W_m(t)\}^T,$$

where W_i and W_j are independent Wiener Processes $\forall i, j = 1, 2, \dots, m : i \neq j$

Property 2.25. Properties of the Wiener Processes $W(t)$

1. $\mathrm{E}[W(s)W(t)] = \min\{s, t\}$
2. $\mathrm{E}[W(t)|W(s)] = \mathrm{E}[W(t)|\mathcal{F}(s)] = W(s)$ for $s < t$
3. $\widetilde{W}(t) := W(t_0 + t) - W(t_0)$ is a Wiener Process for any fixed $t_0 \geq 0$ (Translation invariance)
4. $\widetilde{W}(t) := W(\lambda t)/\sqrt{\lambda}$ is a Wiener Process for any $\lambda \in \mathbb{R}^+$ (Scale invariance)
5. $W(t)$ is nowhere differentiable a.s.

Remark. $\mathcal{F}(t) \subset \mathcal{F}$ is the σalgebra generated by $W(t)$.

The following definitions and results are necessary to work with SDEs, and follows notation from [57] and [58].

Definition 2.26. Let $\{\Omega, \mathcal{F}, \mathcal{F}(t), P\}$ be a complete probability space with probability measure P. Then, the *space of stochastic processes* $\mathcal{L}_{ad}(L^2[t_0, T], \Omega) = \mathcal{L}^2_{ad}$ is the space of all functions $f, g : [t_0, T] \times \Omega \to \mathbb{R}$ such that

- f is $\mathcal{B}[t_0, T] \times \mathcal{F}$-measurable
- f is $\mathcal{F}(t)$-adapted
- $\int_{t_0}^{T} \mathrm{E}\left[|f|^2\right] \mathrm{d}t < \infty \quad$ for almost all (a.a.) $\omega \in \Omega$

equipped with an inner product

$$< f, g >_{\mathcal{L}^2_{ad}} := \int_{t_0}^{T} \mathrm{E}[|fg|] \mathrm{d}t$$

and a norm

$$\|f\|_{\mathcal{L}^2_{ad}} :=< f, f >^{\frac{1}{2}}_{\mathcal{L}^2_{ad}} .$$

2.4.2 Itô Calculus

Definition 2.27. [57, pp.29] Let $f \in \mathcal{L}^2_{ad}$ and let $\{\varphi_n\}_{n \in \mathbb{N}} \in M^2_{step} \subset \mathcal{L}^2_{ad}$ be a sequence of stochastic step functions such that

$$\mathrm{E}\left[\int_{t_0}^{T} (f(t, \omega) - \varphi_n(t, \omega))^2\right] \xrightarrow{n \to \infty} 0.$$

Then, the *Itô integral* $I(f) \in L^2(\Omega)$ is given by

$$I(f) := \int_{t_0}^{T} f(t) \mathrm{d}W(t) = \lim_{n \to \infty} \int_{t_0}^{T} \varphi_n \mathrm{d}W(t). \tag{2.68}$$

[3] The terms Brownian motion $B(t)$ and Wiener Process $W(t)$ are used interchangeably in the literature. Following the reasoning of Kloeden and Platen [37, pp.40] rather than the notation used by Øksendal [57], the author prefers the denotation Wiener Processes in this thesis.

Remark. For a better insight into the steps needed in this construction, see Kuo [58, Ch. 4.3] and Øksendal [57, pp. 26–29].

Lemma 2.14. *Properties of the Itô integral.*
Let $f, g \in \mathcal{L}^2_{ad}$ and $t_0 < t_1 < T$. Then the following properties holds for the Itô integral.

- *Linearity:* $\int_{t_0}^{T}(af + bg)dW(t) = a\int_{t_0}^{T} f dW(t) + b\int_{t_0}^{T} g dW(t) \qquad \forall a, b \in \mathbb{R}$
- *Partition:* $\int_{t_0}^{T} f dW(t) = \int_{t_0}^{t_1} f dW(t) + \int_{t_1}^{T} f dW(t)$ *for a.a.* $\omega \in \Omega$
- *Expectation:* $\mathrm{E}\left[\int_{t_0}^{T} f dW(t)\right] = 0$ *for a.a.* $\omega \in \Omega$
- *Itô isometry:* $\mathrm{E}\left[\left(\int_{t_0}^{T} f dW(t)\right)^2\right] = \mathrm{E}\left[\int_{t_0}^{T} |f|^2 dt\right]$
- $\int_{t_0}^{T} f dW(t)$ *is* $\mathcal{F}(T)$*-measurable*

Definition 2.28. Let $A \in \mathbb{R}^{n\times n}$ be any real-valued matrix. Then the *trace of A* is an operator $\mathrm{tr}\{\cdot\} : \mathbb{R}^{n\times n} \to \mathbb{R}$ such that

$$\mathrm{tr}\{A\} = \sum_{i=1}^{n} A_{ii}, \tag{2.69}$$

i.e. the sum of the diagonal elements of A.

For multidimensional arrays of order greater than two, the *generalized trace* contracts the first and last dimension of the array by summation, implicitly assuming the dimensions have the same length.

Remark. Higher order arrays occuring here are usually a composition of the Gradient or Hessian of a function with some matrix functions. The gradient and Hessian of a vector function (e.g. $b(t,X) \in \mathbb{R}^d$) would be multidimensional arrays of order two and three, respectively, i.e. $\nabla_x b(t,X) \in \mathbb{R}^{d\times d}$ and $\nabla_x^2 b(t,X) \in \mathbb{R}^{d\times d\times d}$. The gradient and Hessian of a matrix function (e.g. $\sigma(t,X) \in \mathbb{R}^{d\times m}$) would be multidimensional arrays of order three and four, respectively, i.e. $\nabla_x \sigma(t,X) \in \mathbb{R}^{d\times d\times m}$ and $\nabla_x^2 \sigma(t,X) \in \mathbb{R}^{d\times d\times d\times m}$.

As an example, let $\Sigma \in \mathbb{R}^{d\times m}$, $g \in C^2\left(\mathbb{R}^d, \mathbb{R}^d\right) \Rightarrow \Sigma^T \nabla_x^2 g(X)\Sigma \in \mathbb{R}^{m\times d\times m}$. Then, $\mathrm{tr}\left\{\Sigma^T \nabla_x^2 g(X)\Sigma\right\} : \mathbb{R}^{m\times d\times m} \to \mathbb{R}^m$ such that

$$\mathrm{tr}\left\{\Sigma^T \nabla_x^2 g(X)\Sigma\right\} = \sum_{k=1}^{m}\sum_{i,j=1}^{d} \Sigma_{ik}\partial_{x_i}\partial_{x_j} g(X)\Sigma_{jk}.$$

Definition 2.29. [57, pp. 44, 48] Let $W(t)$ be an m-dimensional Wiener Process on $\{\Omega, \mathcal{F}, P\}$. A *d-dimensional Itô process* $X(t)$ on $\{\Omega, \mathcal{F}, P\}$ is a stochastic process or integral such that

$$X(t) = X_0 + \int_{t_0}^{t} b(\tau,\omega)\mathrm{d}\tau + \int_{t_0}^{t} \sigma(\tau,\omega)\mathrm{d}W(\tau), \tag{2.70}$$

where $\sigma \in \mathcal{L}^2_{ad}$, such that

$$P\left(\int_{t_0}^{t} \sigma(\tau,\omega)^2 \mathrm{d}\tau < \infty \quad \forall t \geq t_0\right) = 1, \quad \text{(almost surely finite)}$$

b is $\mathcal{F}(t)$-adapted and

$$P\left(\int_{t_0}^{t} |b(\tau,\omega)|\mathrm{d}\tau < \infty \quad \forall t \geq t_0\right) = 1.$$

Theorem 2.15. *Itô's Formula [55].*
Let $X(t)$ be a d_1-dimensional stochastic process satisfying

$$dX(t) = b(t,X)dt + \sigma(t,X)dW(t), \qquad X(t_0) = X_0, \quad t \in [t_0, T],$$

where W is an m-dimensional Wiener Process.
Furthermore, let $g \in C^{1,2}\left([t_0,T]\times\mathbb{R}^{d_1},\mathbb{R}^{d_2}\right)$ and $Y(t)=g(t,X(t))$. Then,

$$\begin{aligned} dY(t) &= \partial_t g dt + (\nabla_x g)^T dX + \frac{1}{2} dX^T \left(\nabla_x^2 g\right) dX \\ &= \left(\partial_t g + (\nabla_x g)^T b + \frac{1}{2}\mathrm{tr}\left\{\sigma^T\left(\nabla_x^2 g\right)\sigma\right\}\right)dt + (\nabla_x g)^T \sigma dW(t), \end{aligned} \tag{2.71}$$

where, using usual convention, $dW_j dW_k = 0$ if $j \neq k$, $(dW_j)^2 = dt$ and $dW_j dt = dt dW_j = (dt)^2 = 0\,\forall j$.
Element-wise, for $i = 1,\ldots,d_2$, this becomes

$$\begin{aligned} dY_i(t) &= \partial_t g_i dt + \sum_{j=1}^{d_1} \partial_{x_j} g_i dX_j + \frac{1}{2}\sum_{j,k=1}^{d_1} \partial_{x_j}\partial_{x_k} g_i dX_j^2 \\ &= \left(\partial_t g_i + \sum_{j=1}^{d_1}\left(\partial_{x_j} g_i b_j + \frac{1}{2}\sum_{k=1}^{d_1}\partial_{x_j}\partial_{x_k} g_i \sum_{l=1}^{m}\sigma_{jl}\sigma_{kl}\right)\right)dt + \sum_{j=1}^{d_1}\partial_{x_j} g_i \sum_{l=1}^{m}\sigma_{jl} dW_l(t) \end{aligned} \tag{2.72}$$

Theorem 2.16. *Existence and Uniqueness.[57, 58, pp.70 & 190–192]*
Let $b : [t_0,T]\times\mathbb{R}^d \to \mathbb{R}^d$ and $\sigma : [t_0,T]\times\mathbb{R}^d \to \mathbb{R}^{d\times m}$ be measurable functions satisfying

1. *Lipschitz condition:*

$$\|b(t,x)-b(t,y)\|_2 + \|\sigma(t,x)-\sigma(t,y)\|_F \le C\|x-y\|_2 \quad \forall x,y \in \mathbb{R}^d \tag{2.73}$$

   for some constant C.

2. *Linear growth condition:*

$$\|b(t,x)\|_2 + \|\sigma(t,x)\|_F \le K(1+\|x\|_2) \quad \forall x \in \mathbb{R}^d \tag{2.74}$$

   for some constant K.

Let $X_0 \in \mathcal{F}_0 \subset \mathcal{F}$ be a random variable independent of the σ-algebra $\mathcal{F}(\infty)$ generated by the m-dimensional Wiener Process $W(t,\omega)$. Assume that X_0 is finite in the mean square, i.e. $\mathrm{E}\left[|X_0|^2\right] < \infty$.
Then, if $X(t)$ satisfies the equation

$$dX(t) = b(t,X)dt + \sigma(t,X)dW(t), \qquad X(t_0) = X_0, \quad t \in [t_0,T], \tag{2.75}$$

it is a unique and t-continuous solution adapted to the filtration $\mathcal{F}^{X_0}(t) = \mathcal{F}_0 \otimes \mathcal{F}(t)$ with a finite ms bound, i.e.

$$\mathrm{E}\left[\int_{t_0}^{T} |X(t)^2| dt\right] < \infty. \tag{2.76}$$

Remark. [58, pp. 190] For $X(t)$ to solve (2.75), it holds that

- $\sigma_{ij} \in \mathcal{L}^2_{ad}$ for $i=1,\ldots,d$ and $j=1,\ldots,m$
- $f_i \in \mathcal{L}^1_{ad}$ for $i=1,\ldots,d$
- (2.75) is satisfied for every $t \in [t_0,T]$ a.s.

2.4.3 Stratonovich Calculus

Definition 2.30. [58, pp. 120] Let $X(t)$ and $Y(t)$ be (possibly multidimensional) Itô processes. The *Stratonovich integral of $X(t)$ w.r.t. $Y(t)$* is given by

$$\int_{t_0}^{t} X(\tau)\circ \mathrm{d}Y(\tau) = \int_{t_0}^{t} X(t)\mathrm{d}Y(t) + \frac{1}{2}\int_{t_0}^{t} \mathrm{d}X(\tau)\cdot\mathrm{d}Y(\tau). \tag{2.77}$$

Equivalently, in symbolic form, we get that

$$X(t)\circ\mathrm{d}Y(t) = X(t)\mathrm{d}Y(t) + \frac{1}{2}\mathrm{d}X(t)\cdot\mathrm{d}Y(t). \tag{2.78}$$

Theorem 2.17. *[56] Let $X(t)$ be an Itô process such that*

$$dX(t)=\overline{b}(t,X)dt+\sigma(t,X)dW(t), \qquad b(t,X)\in\mathbb{R}^d,\ \sigma(t,X)\in\mathbb{R}^{d\times m}, \tag{2.79}$$

where $W(t)$ is an m-dimensional Wiener Process. This corresponds to the Stratonovich process

$$\begin{aligned} dX(t) &= \underline{b}dt+\sigma\circ dW(t) \\ &= \left(\overline{b}-\frac{1}{2}\mathrm{tr}\left\{\sigma^T\nabla_x\sigma\right\}\right)dt+\sigma\circ dW(t). \end{aligned} \tag{2.80}$$

Equivalently, $\overline{b}=\underline{b}+\frac{1}{2}\mathrm{tr}\left\{\sigma^T\nabla_x\sigma\right\}$.

Remark. Consider $\sigma(t,X(t))\in\mathbb{R}^{d\times 1}$ and $\nabla_x\sigma(t,X(t))\in\mathbb{R}^{d\times d\times 1}$ for $m=1$, i.e.

$$\mathrm{tr}\left\{\sigma^T\nabla_x\sigma\right\}_i=\sum_{k=1}^{1}\sum_{j=1}^{d}\sigma_{jk}\partial_{x_j}\sigma_{ik}=\sum_{j=1}^{d}\sigma_{j1}\partial_{x_j}\sigma_{i1}.$$

Proof. From the definition of the Stratonovich integral, we get that

$$\sigma\circ\mathrm{d}W(t)=\sigma\mathrm{d}W(t)+\mathrm{d}\sigma\cdot\mathrm{d}W(t). \tag{2.81}$$

Using Itô's formula (2.71), it follows that

$$\begin{aligned} \mathrm{d}\sigma(t,X)\cdot\mathrm{d}W &= \left(\left(\partial_t\sigma+(\nabla_x\sigma)^T b+\frac{1}{2}\mathrm{tr}\left\{\sigma^T\left(\nabla_x^2\sigma\right)\sigma\right\}\right)\mathrm{d}t+(\nabla_x\sigma)^T\sigma\mathrm{d}W\right)\cdot\mathrm{d}W \\ &= \left((\nabla_x\sigma)^T\sigma\mathrm{d}W\right)^T\mathrm{d}W \\ &= \mathrm{d}W^T\sigma^T\left(\nabla_x\sigma\right)\mathrm{d}W \\ &= \mathrm{tr}\left\{\sigma^T\nabla_x\sigma\right\}\mathrm{d}t. \end{aligned} \tag{2.82}$$

as $\mathrm{d}t\mathrm{d}W_i(t)=0=\mathrm{d}W_i\mathrm{d}W_j$ for $i\neq j$ and $\mathrm{d}W_i^2=\mathrm{d}t\ \forall i,j=1,\dots,m$.

Then, inserting into the symbolic expression for the SDE,

$$\begin{aligned} \mathrm{d}X(t)=\overline{b}\mathrm{d}t+\sigma\mathrm{d}W(t) &= \overline{b}\Delta t+\sigma\circ\mathrm{d}W(t)-\mathrm{d}\sigma\cdot\mathrm{d}W(t) \\ &= \left(\overline{b}-\mathrm{tr}\left\{\sigma^T\nabla_x\sigma\right\}\right)\mathrm{d}t+\sigma\circ\mathrm{d}W(t) \\ &= \underline{b}\mathrm{d}t+\sigma\circ\mathrm{d}W(t). \end{aligned}$$

□

Corollary 2.17.1. *For SDEs with additive noise, i.e. $\sigma(t,X(t))=\sigma(t)\in\mathbb{R}^{d\times m}$, the Itô and Stratonovich formulations coincide; that is, $\overline{b}=\underline{b}$.*

Proof.

$$\nabla_x\sigma=\nabla_x\sigma(t)=0_{d\times d\times m} \qquad \Rightarrow \qquad \underline{b}=\overline{b}+\frac{1}{2}\mathrm{tr}\left\{\sigma^T\nabla_x\sigma\right\}=\overline{b}+0_d=\overline{b}.$$

□

Theorem 2.18. *Stratonovich Chain Rule.*
Let $X(t)$ be a stochastic Stratonovich process such that

$$dX(t)=\underline{b}(t,X)dt+\sigma(t,X)\circ dW(t) \tag{2.83}$$

and let $g\in C^2([t_0,T]\times\mathbb{R}^{d_1},\mathbb{R}^{d_2})$. Then,

$$\begin{aligned} dY(t)=dg(t,X(t)) &= \partial_t g dt+\nabla_x g\cdot dX \\ &= \left(\partial_t g+(\nabla_x g)^T\underline{b}\right)dt+(\nabla_x g)^T\sigma\circ dW(t). \end{aligned} \tag{2.84}$$

Proof. The Stratonovich process in Theorem 2.18 corresponds with the Itô process

$$\begin{aligned} \mathrm{d}X(t) &= \left(\underline{b} + \frac{1}{2}\mathrm{tr}\left\{\sigma^T \nabla_x \sigma\right\}\right)(t,X)\mathrm{d}t + \sigma(t,X)\mathrm{d}W(t) \\ &= \overline{b}(t,X)\mathrm{d}t + \sigma(t,X)\mathrm{d}W(t). \end{aligned} \tag{2.85}$$

Using the Chain Rule for Itô processes (Theorem 2.15), it follows that

$$\mathrm{d}g(t,X(t)) = \left(\partial_t g + (\nabla_x g)^T\, \overline{b} + \frac{1}{2}\mathrm{tr}\left\{\sigma^T \left(\nabla_x^2 g\right)\sigma\right\}\right)\mathrm{d}t + (\nabla_x g)^T\, \sigma \mathrm{d}W(t) \tag{2.86}$$

Looking at the last term and expanding using Definition 2.30:

$$\begin{aligned} (\nabla_x g)^T\, \sigma \mathrm{d}W(t) &= (\nabla_x g)^T\, \sigma \circ \mathrm{d}W(t) - \frac{1}{2}\mathrm{d}\left((\nabla_x g)^T\, \sigma\right) \cdot \mathrm{d}W(t) \\ &= (\nabla_x g)^T\, \sigma \circ \mathrm{d}W(t) - \frac{1}{2}\left(\mathrm{tr}\left\{\sigma^T \left(\nabla_x^2 g\right)\sigma\right\} + (\nabla_x g)^T \mathrm{tr}\left\{\sigma^T \nabla_x \sigma\right\}\right)\mathrm{d}t, \end{aligned} \tag{2.87}$$

using the same argumentation as for (2.82). Hence, inserting this into (2.86),

$$\begin{aligned} \mathrm{d}g(t,X(t)) = &\left(\partial_t g + (\nabla_x g)^T\, \overline{b} + \frac{1}{2}\mathrm{tr}\left\{\sigma^T \left(\nabla_x^2 g\right)\sigma\right\}\right)\mathrm{d}t + (\nabla_x g)^T\, \sigma \circ \mathrm{d}W(t) \\ &- \frac{1}{2}\left(\mathrm{tr}\left\{\sigma^T \left(\nabla_x^2 g\right)\sigma\right\} + (\nabla_x g)^T \mathrm{tr}\left\{\sigma^T \nabla_x \sigma\right\}\right)\mathrm{d}t \\ = &\left(\partial_t g + (\nabla_x g)^T\, \underline{b}\right)\mathrm{d}t + (\nabla_x g)^T\, \sigma \circ \mathrm{d}W(t) \\ = &\,\partial_t g \mathrm{d}t + \nabla_x g \cdot \mathrm{d}X. \end{aligned} \tag{2.88}$$

□

2.4.4 Numerical Simulation

Definition 2.31. A *one-step approximation* $Y_{t,x}(t+\Delta t)$ to a stochastic process $X(t)$ as given by (2.75) is given as

$$Y_{t,x}(t+\Delta t) = x + \Phi(t,x,\Delta t; W(\theta) - W(t), \theta \in [t, t+\Delta t]), \quad t \in [t_0, T], \tag{2.89}$$

with $Y_0 = X_0 = X(t_0)$ and $Y_{k+1} = Y_{t_k,Y_k}(t_{k+1}) = Y_{t_0,X_0}(t_{k+1})$, where $t_k = t_0 + k \cdot \Delta t$ for $k = 0, \ldots, N; t_N = T$.

Definition 2.32. Let $X_{t,x}(t+\Delta t)$ denote a stochastic process which takes the value x at time t, and let $Y_{t,x}(t+\Delta t)$ denote a one-step approximation of $X_{t,x}(t+\Delta t)$. The *local error* $\delta_{t,x}(\Delta t)$ is given by

$$\delta_{t,x}(\Delta t) = X_{t,x}(t+\Delta t) - Y_{t,x}(t+\Delta t), \tag{2.90}$$

while the *global error* $\varepsilon_n(\Delta t)$ is given by

$$\varepsilon_n(\Delta t) = X(t_n) - Y_n. \tag{2.91}$$

Definition 2.33. [39, pp. 1] A one-step method has *strong mean-square order of convergence* γ if

$$\sqrt{\mathrm{E}[|\varepsilon_n(\Delta t)|^2]} \leq K(\Delta t)^\gamma \tag{2.92}$$

for some constant K.

Definition 2.34. [39, pp. 98] A one-step method has *weak order of convergence* γ if, for the exact solution $X(t)$ and its approximation $Y(t)$,

$$|\mathrm{E}[f(X(T))] - \mathrm{E}[f(Y(T))]| \leq K(\Delta t)^\gamma, \tag{2.93}$$

where f is from a sufficiently large class of functions.[4]

[4] Kloeden and Platen are satisfied with polynomials (see [37, pp. XXV]). Here, this is considered too restrictive a condition.

Theorem 2.19. *Fundamental Theorem of Mean-Square Order of Convergence [39, pp. 5]. Consider a system of Itô SDEs*

$$dX(t) = b(t,X(t)) + \sigma(t,X)dW(t), \quad X(t_0) = X_0, \quad t \in [t_0, T]$$

with $b(t,X) \in \mathbb{R}^d$ and $\sigma(t,X) \in \mathbb{R}^{d\times m}$ and $W(t) \in \mathbb{R}^m$ such that the criteria of Theorem 2.16 are fulfilled. Let $\delta_{t,x}$ be the local error of a one-step method from $X_{t,x}(t)$ with arbitrary $t \in [t_0, T - \Delta t]$ and $x \in \mathbb{R}^d$. Suppose that

1. $\left|\mathrm{E}\left[\delta_{t,x}(\Delta t)\right]\right| \leq K_1\sqrt{1+|x|^2}(\Delta t)^{p_1}$
2. $\sqrt{\mathrm{E}\left[|\delta_{t,x}(\Delta t)|^2\right]} \leq K_2\sqrt{1+|x|^2}(\Delta t)^{p_2}$
3. $0 \leq p := p_2 - \frac{1}{2} \leq p_1 - 1$

Then, for any $N \in \mathbb{N}$,

$$\sqrt{\mathrm{E}[|\varepsilon_n(\Delta t)|^2]} \leq K_3\sqrt{1+\mathrm{E}[|X_0|^2]}(\Delta t)^p \quad \textit{for } n = 1,\dots,N. \tag{2.94}$$

In other words, the one-step method has mean-square order of convergence p.

Remark. The constants K_1, K_2, K_3 are independent of X_0 and N; they only depend on the formulation of the problem and the one-step method.

Chapter 3

Hamiltonian Boundary Value Methods

In a series of papers from the last fifteen years (see e.g. [4, 25, 26]), Brugnano, Iavernaro and the late Trigiante introduced a family of RK methods particularly suited to solving deterministic Hamiltonian differential equations. In fact, their family of methods can preserve *any* polynomial Hamiltonian of an arbitrarily high order without significant increase in computational cost [4]. They also perform well for more general Hamiltonians without relying on the expression of the Hamiltonian in the method formula, in contrast with e.g. discrete gradient methods.

For autonomous canonical Hamiltonian systems, meaning ODEs of the form

$$\dot{x} = J\nabla_x H(x), \quad H \in C^1\left(\mathbb{R}^{2d}, \mathbb{R}\right), \quad J = \begin{bmatrix} 0 & I_d \\ -I_d & 0 \end{bmatrix}, \tag{3.1}$$

the Hamiltonian function $H(x(t))$ between t_0 and $t_1 := t_0 + \Delta t$ for some stepsize Δt can be reformulated as a line integral on the form

$$H(x(t_1)) - H(x_0) = \int_{t_0}^{t_1} \frac{\mathrm{d}}{\mathrm{d}t} H(x(t))\mathrm{d}t = \int_0^1 \nabla_x H(u(c))^T u'(c)\mathrm{d}t = 0, \quad c(t) = \frac{t - t_0}{\Delta t}, \tag{3.2}$$

where $u : [0,1] \to \mathbb{R}^{2d}$ is any analytic function such that

$$u(0) = x_0, \quad u(1) = x(t_1).$$

As all Hamiltonian functions are invariants, the integral in (3.2) vanishes when $x(t)$ satisfies (3.1) and $t_0 \leq t_1 \leq T$. The methods considered, called Hamiltonian Boundary Value Method (HBVM), can be considered a particular or even expanded version of collocation methods (see Section 2.3). The distinguishing feature of these methods is that the *line integral* of the Hamiltonian function (the left-hand side of (3.2)) remains zero along the path of the approximation for polynomial H, i.e. $H \in \mathbb{P}_\nu$, $\nu \in \mathbb{N}$.

The derivation of the methods, based on Brugnano *et al.* [4, 26], is presented in Section 3.1. The central properties of HBVMs, as well as a comparison with collocation methods, follows in Section 3.2. Lastly, numerical tests of the methods are found in Section 3.3.

3.1 Derivation of Methods

The derivation of the methods follow along the lines presented by Brugnano *et al.* [26].[1]

[1] The notation used here differs significantly: In the works of Brugnano *et al.* [4, 26], u generally denote the collocation polynomial, whereas σ is used for polynomials in general and their own methods in particular; here, u_H refers to the polynomial used in the derivation of the HBVMs, wit σ is reserved for diffusion terms in the stochastic setting.

Limiting the current scope to the shorter interval $[t_0, t_0+\Delta t]$ allows the following redefinition of (3.1) on $L^2[0,1]$ using orthonormal basis functions (cf. (2.51)):

$$x'(t_0 + c\Delta t) = \Delta t \sum_{j=1}^{\infty} P_{j-1}(c) \underbrace{\int_0^1 P_{j-1}(\theta)\, J\nabla_x H(x(t_0 + \theta\Delta t))\mathrm{d}\theta,}_{\text{const. w.r.t. } c \text{ and/or } t} \tag{3.3}$$

with $c(t)$ as in (3.2). The basis functions considered here are the shifted and scaled Legendre polynomials, which are presented in Section 2.3.1. Note that the term Δt before the sum comes from exchanging the variable of integration, whereas the definite integrals in (3.3) are constant coefficients.

Now, (3.3) is approximated using a collocation polynomial $u_H \in \mathbb{P}_s[0,1]$ with $x(t_0 + \Delta t) \approx u_H(1)$; this polynomial can then be expressed using the differential equation

$$u_H'(c) = \Delta t \sum_{j=1}^{s} P_{j-1}(c) \int_0^1 P_{j-1}(\theta)\, J\nabla_x H(u_H(\theta))\mathrm{d}\theta, \quad u_H(0) = x_0. \tag{3.4}$$

The main difference between (3.3) and (3.4) is that the sum is now finite with s terms.

The next step is to discretize the integrals in (3.4). This leads to the *discrete line integral*

$$\int_0^1 P_{j-1}(\theta)\, J\nabla_x H(u_H(\theta))\mathrm{d}\theta \approx \sum_{l=1}^{k} b_l P_{j-1}(c_l)\, J\nabla_x H(u_H(c_l)) = R_k(\Delta t), \tag{3.5}$$

where the constants $(b_l, c_l)_{l=1}^{k}$ are determined by a quadrature rule on $[0,1]$ and $R_k(\Delta t)$ is the interpolation error of the quadrature rule. If H is a polynomial function, a quadrature rule of sufficiently high order computes the line integral in (3.5) exactly. In other words, the residual $R_k(\Delta t)$ vanishes.

Inserting (3.5) into (3.4) leads to the *extended collocation condition* [4]

$$\begin{aligned} u_H'(c) &= \Delta t \sum_{j=1}^{s} P_{j-1}(c) \sum_{l=1}^{k} P_{j-1}(c_l) b_l\, J\nabla_x H(u_H(c_l)) \\ \implies u_H(c_i) &= \Delta t \sum_{j=1}^{s} \left(\int_0^{c_i} P_{j-1}(c)\mathrm{d}c \right) \sum_{l=1}^{k} P_{j-1}(c_l) b_l\, J\nabla_x H(c_l) \quad \text{for } i = 1, \dots, k. \end{aligned} \tag{3.6}$$

This condition is used to determine the $k \times k$ coefficients in A of the resulting one-step method. Here, the k refers to the number of stages in the method, whereas s refers to the order of the interpolation polynomial.

To arrive at a RK formulation of the method, reorder the terms in (3.6) such that

$$u_H(c_i) = \Delta t \sum_{l=1}^{k} \underbrace{\sum_{j=1}^{s} \left(\int_0^{c_i} P_{j-1}(c)\mathrm{d}c \right) P_{j-1}(c_l) b_l}_{=a_{il}} J\nabla_x H(c_l) \quad \text{for } i = 1, \dots, k, \tag{3.7}$$

whence the formula for the coefficients is given by the equation (with an index name switch)

$$a_{ij} = \sum_{l=1}^{s} \left(\int_0^{c_i} P_{l-1}(c)\mathrm{d}c \right) P_{l-1}(c_j) b_j \quad \text{for } i, j = 1, \dots, k. \tag{3.8}$$

This expression can be rewritten in matrix form as

$$A := \mathcal{I}_s W_s^T B, \tag{3.9a}$$

where $B = \operatorname{diag}\{b\}$ and $\mathcal{I}_s, W_s \in \mathbb{R}^{k\times s}$ such that

$$\mathcal{I}_{s,ij} = \int_0^{c_i} P_{j-1}(c)\mathrm{d}c, \quad W_{s,ij} = P_{j-1}(c_i) \quad \text{for} \quad i = 1,\dots,k, \quad j = 1,\dots,s; \tag{3.9b}$$

$P_j(c)$ is the j–th shifted and scaled Legendre polynomial (cf. Sections 2.3.1 and 2.3.4).

In short, construction of HBVM(k,s) in RK form with coefficients $A = \{a_{ij}\}_{i,j=1}^k$, $b = \{b_i\}_{i=1}^k$ and $c = \{c_i\}_{i=1}^k$ follows this procedure:

1. Set c to be the k distinct roots of an interpolation polynomial (e.g. Gauß or Lobatto).
2. Let b be given by the integral of the Lagrange function, i.e.

   $$b_i = \int_0^1 \ell_i(c)\mathrm{d}c, \quad \text{where} \quad \ell_i(c) = \prod_{\substack{j=1\\ j\neq i}}^k \frac{c-c_j}{c_i-c_j}$$

   or use explicit expressions (see Section 2.3.4 for Gauß and Lobatto quadrature).
3. Set $A := \mathcal{I}_s W_s^T B$, as in (3.9).

3.2 Properties of HBVMs

The following are the main results for HBVM(k,s), as shown and stated by Brugnano *et al.* [4, 26]:

Theorem 3.1. *Given a HBVM(k,s) based on Gaußian quadrature and stepsize Δt applied to the equation* (3.1)*, i.e.*

$$\dot{x} = J\nabla_x H(x(t)), \quad x(t_0) = x_0 \in R^{2d}, t \in [t_0, T].$$

Then it holds that

1. *the method has order of accuracy* $2s$
2. *the method preserves any $H \in \mathbb{P}_\nu$, where $\nu \leq 2k/s$,* exactly
3. *for general Hamiltonians, the method has a local energy error* $\mathcal{O}\left(\Delta t^{2k+1}\right)$
4. *the method is symmetric*

Remark. The first statement follows by Theorem 2.10 if HBVM(k,s) satisfies conditions $B(2s)$, $C(s)$ and $D(s-1)$. By definition, HBVM(k,s) satisfies $B(2k)$, where $k \geq s$, and $C(s)$ follows as each row of A is a multistep method of order greater than or equal to s. Brugnano *et al.* [4] shows that $D(s-1)$ holds true.

The second and third statement follows by definition. It holds for $H \in \mathbb{P}_\nu$ with $\nu \leq 2k/s$ and $u_H \in \mathbb{P}_s$ that

$$H(u_H(t)) - H(x_0) = \int_{t_0}^t \frac{\mathrm{d}}{\mathrm{d}\tau} H(u_H(c(\tau))))\mathrm{d}\tau \in \mathbb{P}_{\nu s} \subseteq \mathbb{P}_{2k/s\cdot s} = \mathbb{P}_{2k}.$$

When the order of the Hamiltonian exceeds the degree of exactness for the underlying quadrature rule, for $t \in [t_0, t_0 + c\Delta t]$, the energy error

$$\begin{aligned}
e_H(\Delta t) &:= H(x_1) - H(x(t_0+\Delta t)) = H(u_H(1)) - H(u(0)) + \underbrace{(H(x(t_0)) - H(x(t_0+\Delta t)))}_{=0} \\
&= \int_0^1 \frac{\mathrm{d}}{\mathrm{d}t} H(u_H(c))\mathrm{d}c = \int_0^1 \nabla_x H(u_H(c))^T u_H'(c)\mathrm{d}c \\
&= \Delta t \sum_{j=1}^k \underbrace{\left(\int_0^1 P_{j-1}(c)\nabla_x H(u_H(c))\mathrm{d}c\right)^T J \left(\int_0^1 P_{j-1}(\theta)\nabla_x H(u_H(\theta))\mathrm{d}\theta\right)}_{=0} + \Delta t \cdot R_k(\Delta t) \\
&= \Delta t \cdot R_k(\Delta t) = \Delta t \cdot \mathcal{O}\left(\Delta t^{2k}\right) = \mathcal{O}\left(\Delta t^{2k+1}\right)
\end{aligned}$$

as $R_k(\Delta t)$ is the interpolation error of a quadrature rule of order $2k$.

Lastly, symmetry follows from Lemma 2.11.

Remark. For $k = s + r$ and $r \geq 0$, the r extra stages are called silent stage by Brugnano *et al.* [4], as they do not affect the method order nor significantly increase computational cost (see Section 5.1.2).

A similar result holds for Lobatto-based HBVMs.

Lemma 3.2. *The results of Theorem 3.1 holds for HBVMs based on Lobatto quadrature with one additional point; that is, the Lobatto-HBVM with $k+1$ stages based on the $s+1$-th Lobatto polynomial satisfies the conditions of Theorem 3.1. As such, Lobatto-HBVMs with $k+1$ stages of degree $2s$ are referred to as HBVM(k,s).*

Remark. The (Gauß–)Lobatto collocation polynomial is one degree lower than Gauß(–Legendre) (see (2.56)); the first stage of Lobatto-based methods is always explicit. The proof can be done in a similar way as to Theorem 3.1 - see [4].

Remark. Lobatto-HBVMs are stiffly accurate. This follows automatically from being based on Lobatto quadrature, for which the last stage of the coefficient matrix A always coincides with the b vector.

Remark. The coefficient matrix A has rank s for both Gauß and Lobatto quadrature. That the rank of the coefficient matrix A depends on s rather than k becomes important in implementation. For more details, see Section 5.1.2.

3.2.1 Limits of Collocation Methods

As should be clear from the previous subsections, there is a strong relation between HBVMs and collocation methods. In fact, the following holds true [4].

Lemma 3.3. *The HBVM(k,s) with $k = s$ coincides with the collocation method based on the same collocation polynomial of degree s.*

Proof. Consider now collocation methods based on Gaußian quadrature. According to [26, Lemma 1], the matrix $\mathcal{I} \in \mathbb{R}^{s \times s}$ can be rewritten as

$$\mathcal{I}_s = W_{s+1}\mathcal{X} = W_{s+1}\begin{bmatrix} \mathcal{X}_s \\ \xi_s \mathrm{e}_s^T \end{bmatrix} \tag{3.10}$$

with $\mathrm{e}_i \in \mathbb{R}^s$ the i-th unit vector and $\mathcal{X}_s$ given as

$$\mathcal{X}_s = \begin{bmatrix} \xi_0 & -\xi_1 & & \\ \xi_1 & 0 & \ddots & \\ & \ddots & \ddots & -\xi_{s-1} \\ 0 & & \xi_{s-1} & 0 \end{bmatrix} \in \mathbb{R}^{s \times s}, \quad \xi_k = \frac{1}{2\sqrt{|4k^2 - 1|}} \tag{3.11}$$

where $\mathcal{X}_s = \mathcal{X}_G$, as seen in Section 2.3.4. It is thus possible to write A as

$$A = \mathcal{I}_s W_s^T B = W_{s+1}\mathcal{X}_s W_s^T B. \tag{3.12}$$

For $k = s$, it holds that

$$W_{s+1}\mathcal{X} = W_s \mathcal{X}_G \tag{3.13}$$

as $w_{s+1,i} = P_s(c_i) = 0$ for all $i = 1, \ldots, s$, with being defined as the roots of $P_s(x)$ on $[0,1]$. □

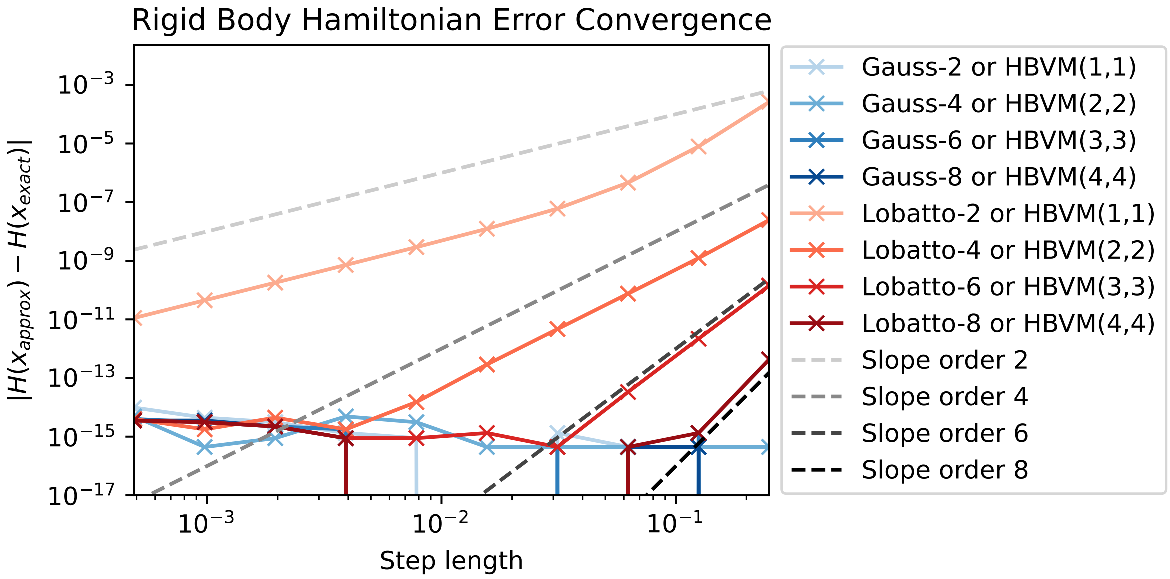


Figure 3.1: Plot of global error convergence in Rigid Body Hamiltonian (3.15) for Gauß and Lobatto collocation methods of order $p \in \{2,4,6,8\}$ with constants $I = [I_1, I_2, I_3]^T = [.8,.6,.2]^T$, $(t_0, T) = (0,1)$ and initial values $x_0 = [\cos(1.1), 0, \sin(1.1)]^T$. The approximations are compared to $H(x_{exact}) = H(x_0)$.

How do the collocation methods fare when trying to exactly preserve Hamiltonians in comparison with HBVMs? From Sections 2.2.2 and 2.3, it should be clear that the symplectic Gauß methods preserve quadratic invariants exactly, whereas the non-symplectic Lobatto (IIIA) methods do not; however, the Lobatto methods are expected to preserve quadratic Hamiltonians. This can most simply be demonstrated by inserting $k = s$ in the presevation condition of HBVMs:

$$\nu \leq \left.\frac{2k}{s}\right|_{k=s} = 2s/s = 2.$$

For a more visual demonstration, consider the Rigid Body Problem in the (2.20) form

$$\dot{x} = S(x) \cdot x = \begin{bmatrix} 0 & x_3/I_3 & -x_2/I_2 \\ -x_3/I_3 & 0 & x_1/I_1 \\ x_2/I_2 & -x_1/I_1 & 0 \end{bmatrix} \cdot x, \qquad x(t_0) = x_0 \in \mathbb{R}^3, \quad t \in [t_0, T] \tag{3.14}$$

with constants $I = [I_1, I_2, I_3]^T = [.8,.6,.2]^T$ and initial values $x_0 = [\cos(1.1), 0, \sin(1.1)]^T$. This Poisson problem has two quadratic invariants

$$H(x) = \frac{1}{2}\left(\frac{1}{I}\right)^T x^2, \qquad C(x) = \frac{1}{2}\|x\|^2 \tag{3.15}$$

the first one will be referred to as the Hamiltonian, whereas the second is a Casimir. In Figure 3.1 one can clearly see that Gauß methods achieves an error in the Hamiltonian close to machine precision even for large stepsizes, whereas Lobatto methods maintain error convergence maching their order; both results are as expected.[2]

For a quadratic canonical Hamiltonian problem, consider the harmonic oscillator

$$\dot{x} = J\nabla_x H(x), \quad H(x) = \frac{1}{2}\|x\|_2^2, \quad x(t_0) = x_0 \in \mathbb{R}^2, \quad t \in [t_0, T] \tag{3.16}$$

[2] Lobatto-2 (the trapezoidal rule) displays a slightly higher order of convergence for the largest stepsizes. This is caused by the Newton iterations not converging within the maximum number of iterations in each timestep. The discussion of the implementation of the Newton iterations are found in Section 5.1.1.

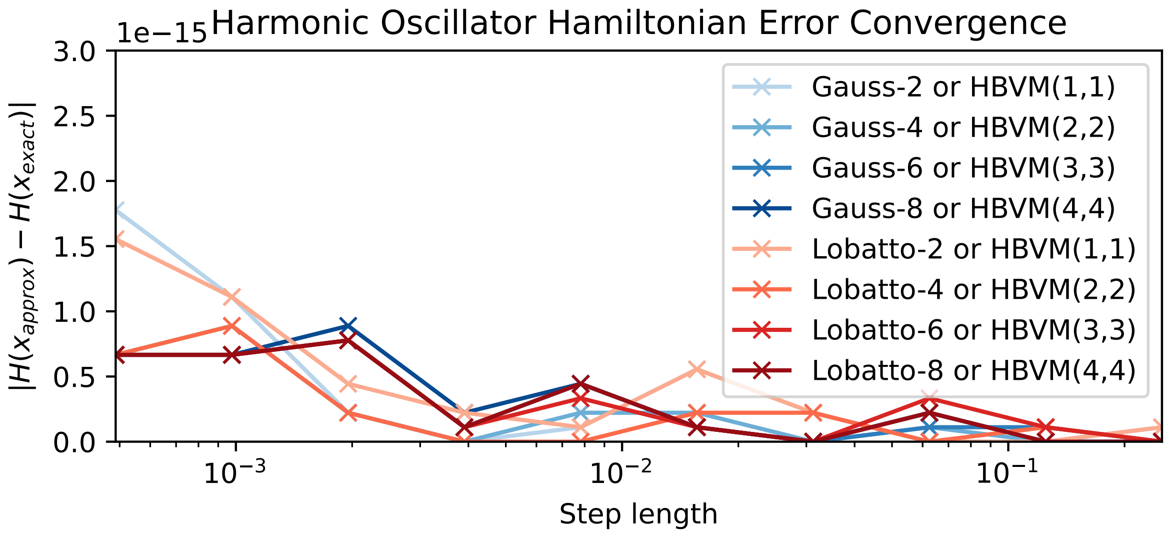


Figure 3.2: One-dimensional Harmonic Oscillator (3.16) Hamiltonian error convergence plot of Gauß and Lobatto collocation methods of order $p \in \{2,4,6,8\}$ with initial value $x_0 = [q_0, p_0]^T = [0,1]$ and $(t_0, T) = (0,1)$. The approximations are compared to $H(x_{exact}) = H(x_0)$.

with initial values $x_0 = [q_0, p_0]^T = [0,1]^T$. As can be seen in the corresponding convergence plot of Figure 3.2, both Gauß and Lobatto methods preserve the quadratic Hamiltonian exactly; the slight increase in global error for smaller stepsizes can be attributed to accumulated roundoff error over an increasing number of timesteps. For their performance on examples of higher order and more general canonical Hamiltonian problems, see Section 3.3.

3.3 Numerical Tests and Results

For demonstrating the convergence properties of the HBVMs presented in Theorem 3.1 and Lemma 3.2, it would be necessary to simulate both polynomial and non-polynomial canonical Hamiltonian systems.

All the test problems are canonical Hamiltonian systems, i.e. have the form

$$\dot{x} = J\nabla_x H(x), \quad J = \begin{bmatrix} 0 & I_d \\ -I_d & 0 \end{bmatrix}, \quad x(t_0) = x_0 \in \mathbb{R}^{2d}, \quad t \in [t_0, T]. \tag{3.17}$$

For convergence plots, the global error is calculated on the interval $t \in [0,1]$ and the reference solution approximated with Gauß-10 using stepsize $\Delta t = 2^{-13}$, unless otherwise specified. In the convergence tables, when no error in Hamiltonian is expected eller observed beyond machine precision, this is reported as '–'.

3.3.1 Harmonic Oscillator

First, revisit the harmonic oscillator of (3.16)

$$\dot{x} = J\nabla_x H(x), \quad H(x) = \frac{1}{2}\|x\|_2^2, x(t_0) = x_0 \in \mathbb{R}^2, \quad t \in [t_0, T] \tag{3.18}$$

as before with initial values $x_0 = [q_0, p_0]^T = [0,1]^T$. This problem has the explicit solution

$$x(t) = \begin{bmatrix} \sin(t) \\ \cos(t) \end{bmatrix}, \tag{3.19}$$

making it useful for computing errors over long time intervals.

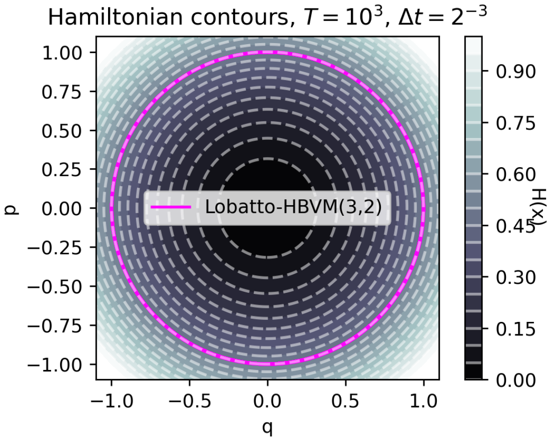


Figure 3.3: The Harmonic Oscillator (3.18) approximated by Lobatto-HBVM(3,2) over time $T = 10^3$ plotted over the contours of the Hamiltonian.

Figure 3.4 displays the long-term error behaviour of HBVMs with parameters $s \in \{2,3\}$, determining the solution order, and $k \in \{3,4,5\}$, determining the "Hamiltonian" order. The error is displayed for the solution variable in the upper plot and the Hamiltonian in the lower plot. Figure 3.3 shows the numerical approximation of one HBVM moving on the contours of the Hamiltonian on the same timespan.

The Hamiltonian error stays close to machine precision over the whole interval; one can only observe random fluctuations most likely due to roundoff errors. However, all the numerical approximations have a weak linear drift away from the exact solution of the ODE. The six methods tested are clustered in two lines, making their individual behaviour hard to discern, but their qualitative behaviour is identical. That the methods with $s = 2$ are clustered in the top line while the methods with $s = 3$ are clustered in the lower line in the top plot will be confirmed with further experiments on error convergence.

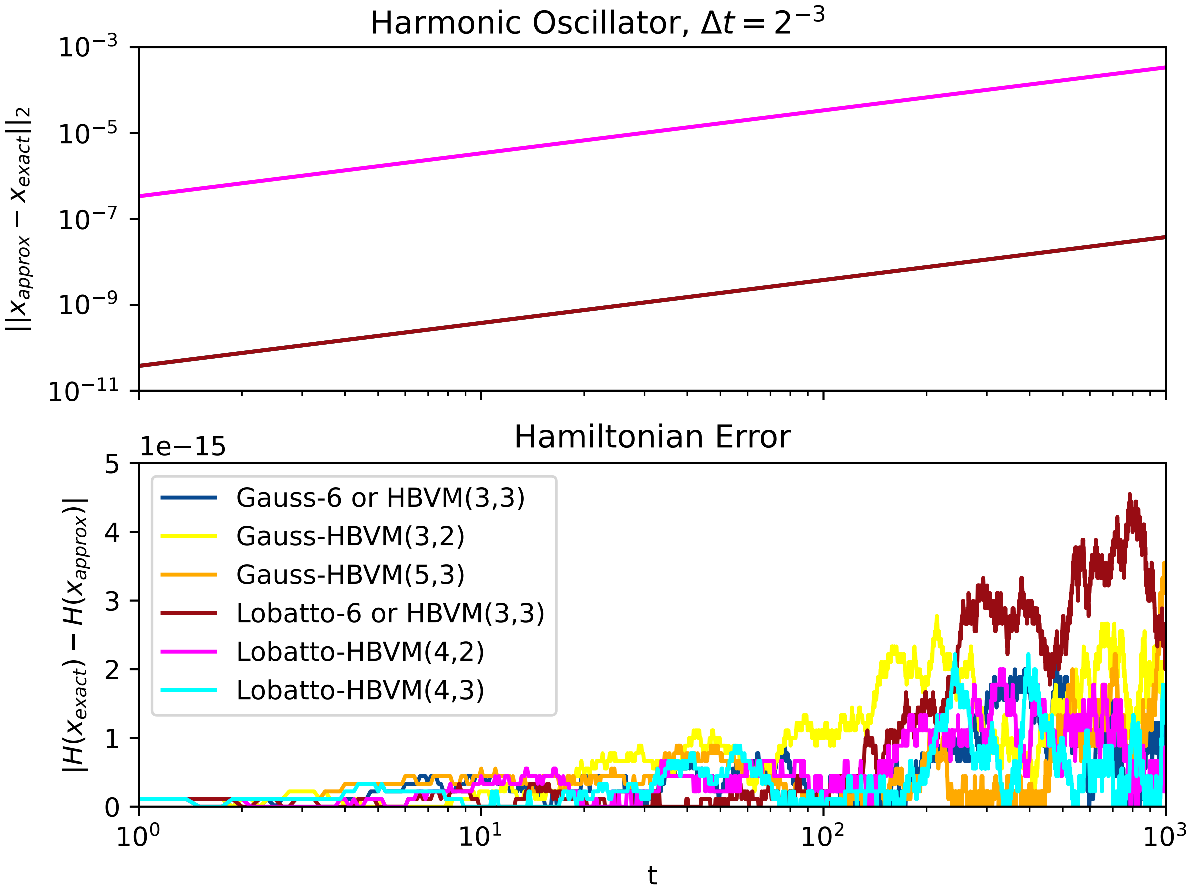


Figure 3.4: Long term error development for Harmonic Oscillator (3.18) with a selection of methods with $T = 10^3$. The solution error is calculated with an exact formula, whereas the Hamiltonian reference solution is the initial value. Note that the horizontal axis is logarithmic and the vertical axis logarithmic and linear for the first and second plot, respectively.

3.3.2 Hénon–Heiles

Now, consider the Hénon–Heiles Problem, which is a cubic Hamiltonian with two degrees of freedom; it has been used as a test in [3, 6, 10] and related later papers. The variant of the Hénon–Heiles Problem studied has the Hamiltonian function

$$H(x) = H(q,p) = \frac{1}{2}\left(\|q\|_2^2 + \|p\|_2^2\right) + \alpha\left(q_1 q_2^2 - \frac{1}{3}q_1^3\right) \tag{3.20}$$

where $\alpha = 16$ as in the paper by Burrage and Burrage [3] and the initial value used is $x_0 = [\sqrt{3}, 1, 1, 1]^T$.

Method Name	Hamiltonian		Solution	
	Expected	Observed	Expected	Observed
Gauss-2 or HBVM(1,1)	2	1.99	2	2.00
Gauss-HBVM(2,1)	–	–	2	2.00
Gauss-4 or HBVM(2,2)	4	3.89	4	3.86
Gauss-HBVM(3,2)	–	–	4	3.85
Gauss-HBVM(4,2)	–	–	4	3.85
Gauss-6 or HBVM(3,3)	6	6.02	6	5.96
Gauss-HBVM(4,3)	8	8.1*	6	5.98
Gauss-HBVM(5,3)	10.0	–	6	5.98
Gauss-HBVM(6,3)	–	–	6	5.98

Table 3.1: Table comparing expected and observed error convergence order in Hamiltonian and ODE solution for deterministic Hénon–Heiles problem, calculated using same data as seen in Figure 3.5b unless otherwise stated. *Approximate value read from figure

In Figure 3.5, global and local error convergence plots for a selection of HBVMs based on Gauß quadrature are shown; Figure B.1 in Appendix B consists of convergence plots for methods based on Lobatto quadrature. Error both in x and $H(x)$ are compared to a reference solution[3], calculated using Gauß-10 and stepsize 2^{-13}. The expected and observed Global convergence orders for the methods both in solution and Hamiltonian are reported in Tables 3.1 and B.1.

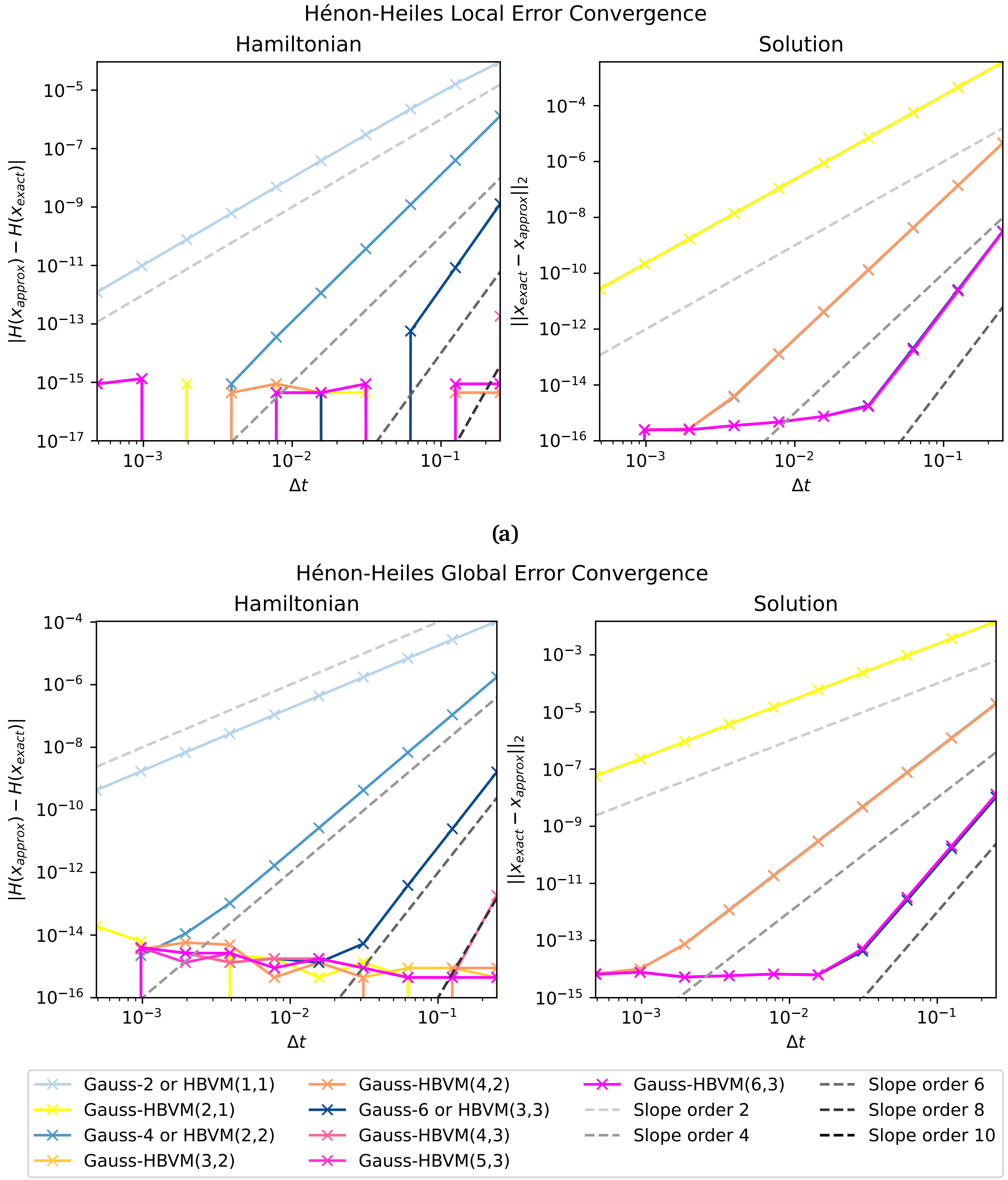
As can be seen, the collocation methods maintain the same order of convergence in $H(x)$ as in x whereas the HBVMs with extra stages (i.e. $k \geq s$) in general achieves machine precision preservation immediately. The exception to this is HBVM$(4,3)$ (for both Gauß and Lobatto quadrature), which converges with order approximately eight, which is the expected order reported in the tables. However, HBVM$(5,3)$ preserves the Hamiltonian down to machine precision, even though an order of ten is expected. It might be that an order of ten would be observed for larger stepsizes than included here. It should be noted that the cubic terms in (3.20) are relatively small, causing all the methods to perform well in terms of Hamiltonian error.

As before is virtually impossible to discern between the different methods for global and local error convergence in Figures 3.5 and B.1, other than being clustered in three lines with slopes $2s+i$ for $s = \{1,2,3\}$ and $i = 1$ and $i = 0$ for local and global convergence, respectively. The reader is therefore referred to the Tables 3.1 and B.1, where the expected and observed convergence order of the methods are summarized; and to Figure 3.6, where global errors of the HBVMs are scaled by the Gauß collocation method of corresponding order, i.e.

$$\text{Relative error} = \frac{\|x_{\text{exact}} - x_{\text{HBVM}(k,s)}\|_2}{\|x_{\text{exact}} - x_{\text{Gauß-2s}}\|_2}. \tag{3.21}$$

The observed convergence order and relative error for each timestep hardly deviates from the collocation method, whence it is clear that the methods maintain the expected theoretical order.

[3] For Hamiltonian plots, the reference solution stays constant, so $H(x_{exact}) := H(x_0)$ is used in error calculations.



(a)

(b)

Figure 3.5: Convergence plots for HBVMs with Gauß-Legendre quadrature with $s \in \{1, 2, 3\}$ on Hénon–Heiles problem for $t \in [0, 1]$. In addition slopes of order a are plotted for $a \in \{2, 4, 6, 8\}$ in Figure 3.5b and for $a \in \{3, 4, 7, 9\}$ in Figure 3.5a. Expected and observed order is reported in Table 3.1.

Moreover, the qualitative behaviour of the HBVMs are identical for Gauß and Lobatto quadrature, whence it would be redundant to report results both for Gauß and Lobatto quadrature for every test problem in the numerical sections of the thesis. Therefore, only the convergence results for HBVMs with one quadrature are included in the numerical sections of the main report, with results for the other quadrature found in Appendix B.

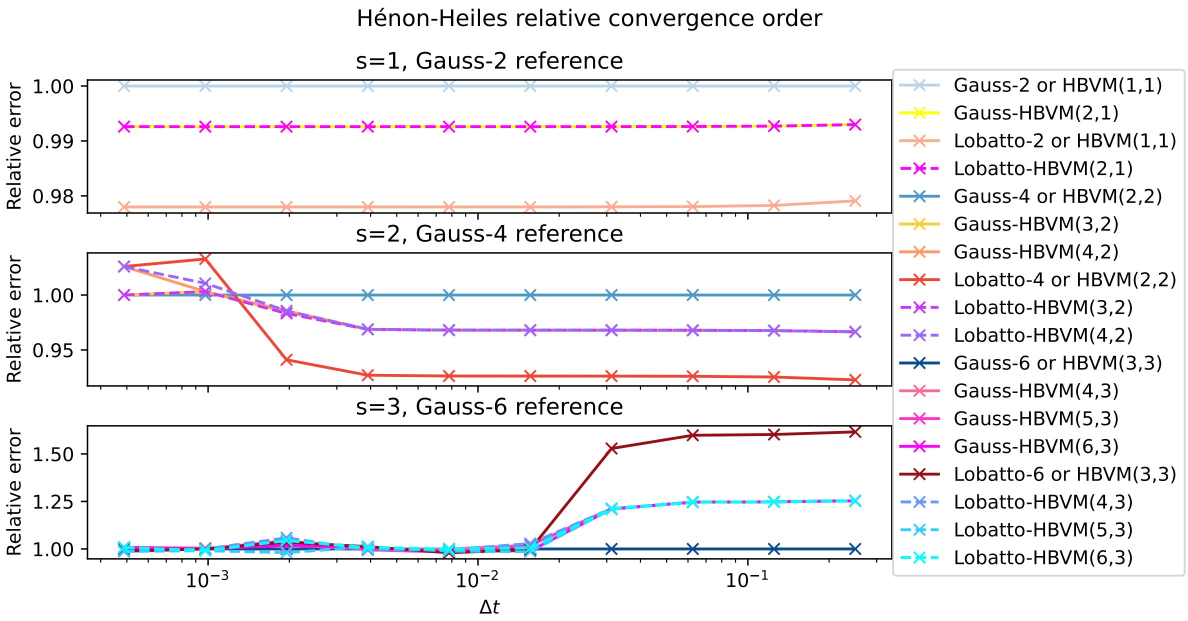


Figure 3.6: Hénon–Heiles global error convergence of HBVM(k,s) with $s \in \{1,2,3\}$; k and the method quadratures are described in the figure legend. The error of each method is scaled by the error of the Gauß collocation method of the same expected order; see (3.21).

3.3.3 Sixth degree Polynomial Hamiltonian

The third test problem is a sixth degree Hamiltonian with one degree of freedom, wich was used as an example in [4, 10]. Its Hamiltonian $H(x) \in \mathbb{P}_\nu(\mathbb{R}^2,\mathbb{R})$ with $\nu = 6$ is given by the formula

$$H(x) = H(q,p) = \frac{1}{3}p^3 - \frac{1}{2}p + \frac{1}{30}q^6 + \frac{1}{4}q^4 - \frac{1}{3}q^3 + \frac{1}{6}, \quad x_0 = [0,1]^T. \tag{3.22}$$

In this case, the problem demands that $k = \nu s/2 = 3s$; in other words, three times as many stages are needed to guarantee exact energy-preservation than of the base collocation method. The need for thrice as many stages without any gain in local error of the solution might seem excessive. For a discussion of a way to side-step the potential increase in computational cost, see Section 5.1.2.

Figure 3.7 shows the contours of the Hamiltonian with an embedded solution generated by Lobatto-HBVM$(6,2)$ from initial value $x_0 = [0,1]^T$. As as can be seen, the solution moves neatly on a single level set.

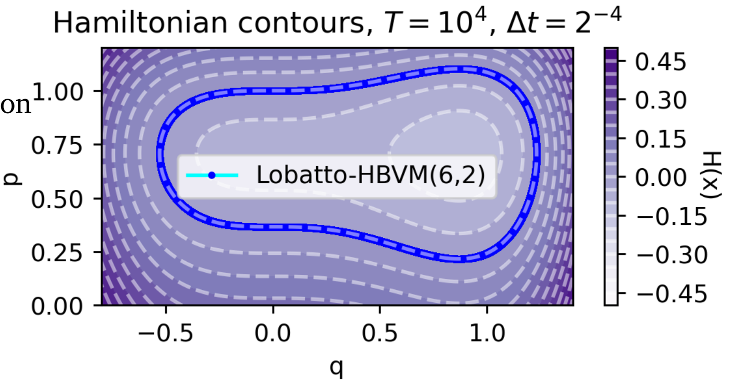


Figure 3.7: Contour plot of Hamiltonian (3.22) with approximation from Lobatto-HBVM$(6,2)$ over $t \in [0,10^4]$ with stepsize 2^{-4}.

In Figure 3.8 and Table 3.2 (and similarly Figure B.2 and B.2 for Lobatto quadrature), convergence results for (3.22) are reported.

As before, the global error in the ODE closely matches the expected order. For any lower choice of k than giving exact preservation, the Hamiltonian error convergence is still significantly better than for the collocation method, albeit more stages are needed to preserve the error exactly than for the Hénon–Heiles problem. More prominently than for the Hénon–Heiles problem, several of the methods preserve energy in practice despite having $k < 3s$, bypassing the expected error convergence. As noted by Brugnano *et al.* [26], one needs only to add enough stages to ensure the error doesn't exceed machine precision for the method to be virtually energy-preserving.

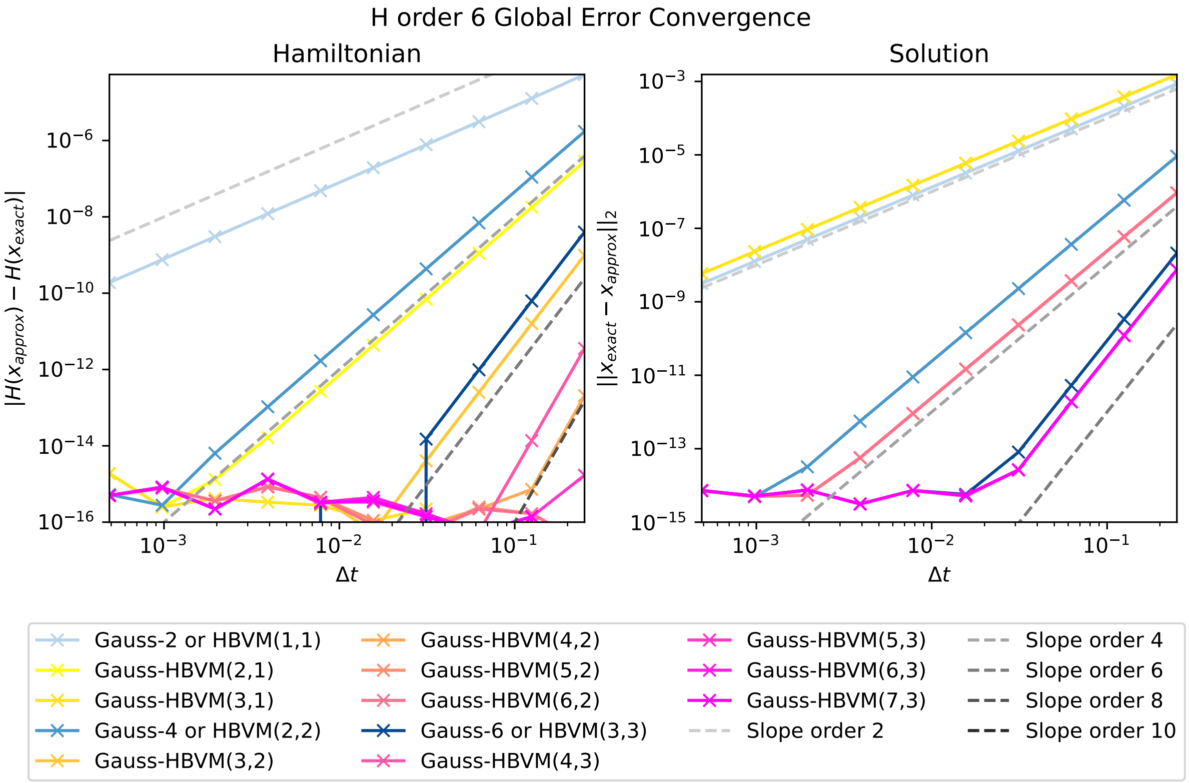


Figure 3.8: Convergence of Gauß-HBVM(k,s) for Hamiltonian of order 6 given by (3.22) for $s \in \{1,2,3\}$ and k up to conservation for $t \in [0,1]$. The reference solution x_{exact} is calculated using Gauß-10 and $\Delta t = 2^{-13}$.

Method Name	Hamiltonian		Solution	
	Expected	Observed	Expected	Observed
Gauss-2 or HBVM(1,1)	2	2.01	2	2.01
Gauss-HBVM(2,1)	4	4.01	2	2.00
Gauss-HBVM(3,1)	–	–	2	2.00
Gauss-4 or HBVM(2,2)	4	3.99	4	4.01
Gauss-HBVM(3,2)	6	5.97	4	3.99
Gauss-HBVM(4,2)	8	–	4	3.99
Gauss-HBVM(5,2)	10	–	4	3.99
Gauss-HBVM(6,2)	–	–	4	3.99
Gauss-6 or HBVM(3,3)	6	6.00	6	6.00
Gauss-HBVM(4,3)	8	8.03	6	6.03
Gauss-HBVM(5,3)	10	–	6	6.03
Gauss-HBVM(6,3)	12	–	6	6.04
Gauss-HBVM(7,3)	14	–	6	6.03

Table 3.2: Expected and observed error convergence orders measured in Hamiltonian and ODE solution for Gauß-HBVMs; the approximated problem is a canonical Hamiltonian system with $H \in \mathbb{P}_6$ given by (3.22), and the order is calculated using the same data as seen in Figure 3.8.

3.3.4 Kepler Problem

As an example of a non-polynomial canonical Hamiltonian system, the Kepler problem is considered. This is a quite simple, yet illustrative non-polynomial problem, which has been studied in [6, 10, 17, 35]. It is an instance of the central force problem in $\mathbb{R}^2$, and with suitable normalization it has the Hamiltonian

$$H(x) = H(q,p) = \frac{1}{2}p^T p - \frac{1}{\|q\|}. \tag{3.23}$$

In addition to being a canonical Hamiltonian problem, the two-dimensional Kepler problem also preserves the angular momentum

$$G(x) = G(q,p) = q \times p = q_1 p_2 - p_1 q_2. \tag{3.24}$$

For the numerical experiments, initial values are determinied by the formula

$$x_0^T = \left[q_0^T, p_0^T\right] = \left[1-e, \quad 0, \quad 0, \quad \sqrt{\frac{1+e}{1-e}}\right], \qquad e = 0.5 \tag{3.25}$$

where $e \in [0,1)$ is the eccentricity of the problem, which ensures orbital movement around a mass centre in $((0,0)$ and a period of 2π [6, ch. I.2].

For illustration, Figure 3.9 shows the solution approximated by Gauß-HBVM$(4,3)$ over 10^4 periods. Figure 3.10 compares the error behaviour of Gauß-HBVM$(4,s)$ for $s \in \{2,3,4\}$. In both plots, stepsize $\Delta t = 2^{-4}$ is used.

As expected, the ODE error decreases significantly when adding a fundamental stage, i.e. increasing s by one and the order of the method by two. All three methods demonstrate an error in Hamiltonian bounded between 10^{-12} and 10^{-11}, although the collocation method is smaller than the other by more than a factor of three and four for HBVM$(4,2)$ and HBVM$(4,3)$, respectively. Still, this is quite decent performance for methods of two and four orders lower error convergence. Otherwise, the Hamiltonian and solution error behaves qualitatively the same, oscillating regularly for all methods.

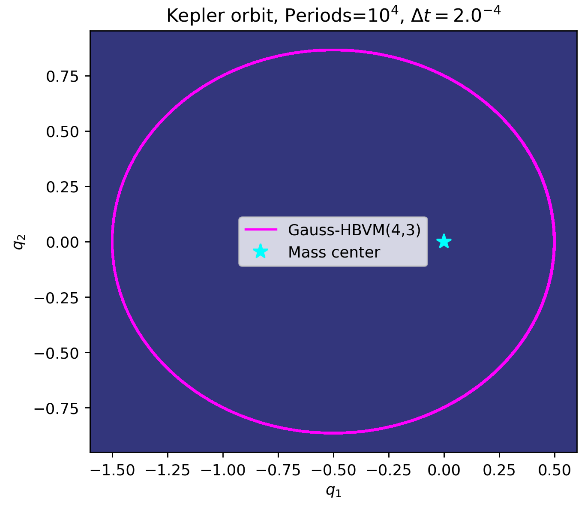


Figure 3.9: Kepler problem (3.23) with initial values (3.25) simulated over 10^4 periods.

The error in the angular momentum, shown in the lowest plot, is bounded for HBVM$(4,2)$ at about 10^{-6} and for HBVM$(4,3)$ at approximately 10^{-9}. Gauß-6 preserves the angular momentum exactly, which is as expected for a symplectic method.

Lastly, convergence results for Lobatto-HBVMs are reported in Figure 3.11 and Table 3.3. The matching Figure B.3 and Table B.3 for Gauß-HBVMs are found in the Appendix. All the methods demonstrate the expected order with some margin of error: HBVM$(5,2)$ has a calculated Hamiltonian errer convergence order of 9.17 compared to an expected order ten, yet its line in the plot overlaps closely with HBVM$(5,3)$, which achieves a measured 10.09. The convergence in the solution very similar to the one seen for the other problems, confirming the order results.

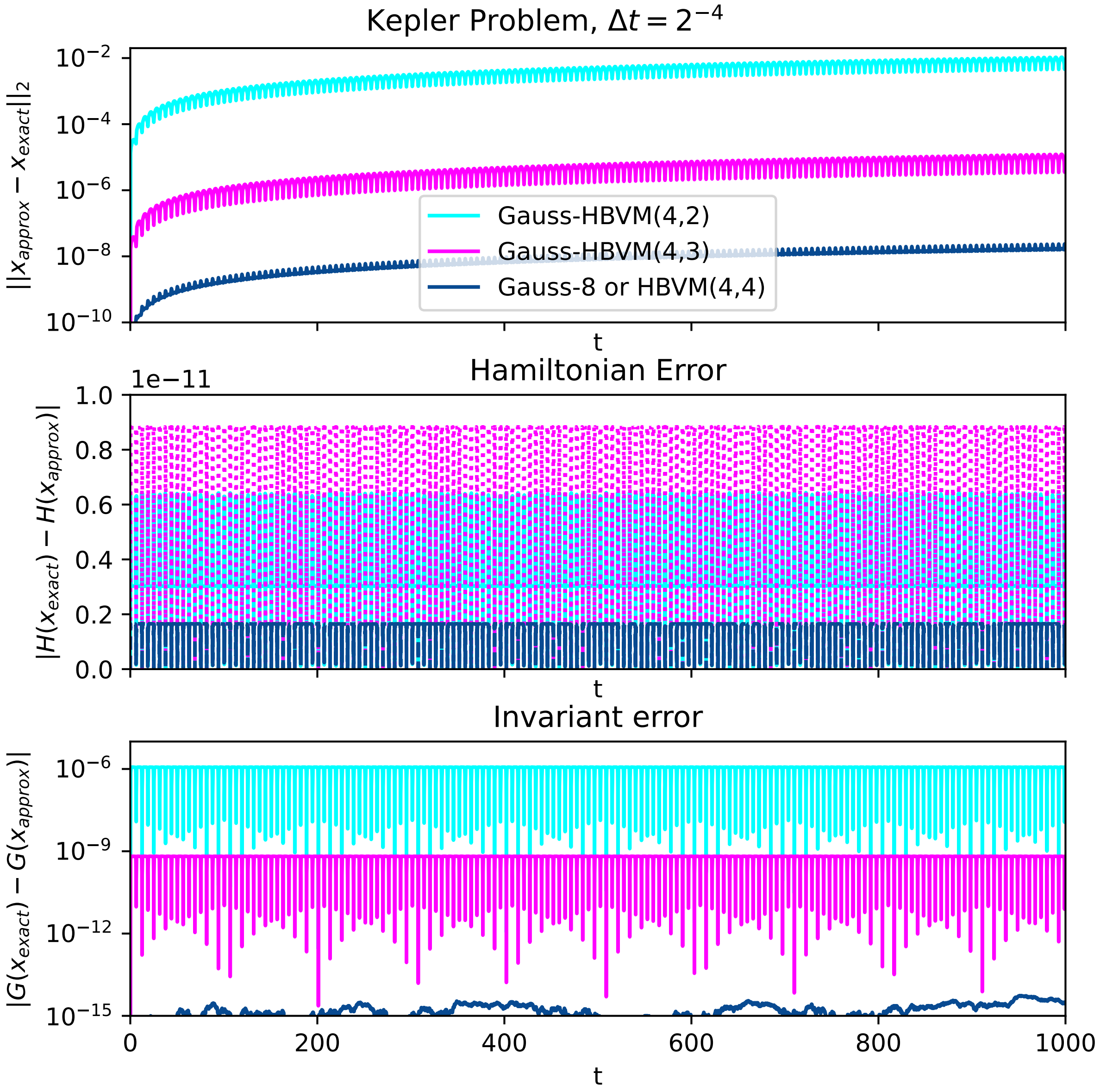


Figure 3.10: Error development for Gauß-HBVMs applied to the Kepler problem with starting values (3.25), with $e = 0.5$, stepsize $\Delta t = 2^{-4}$ and $(t_0, T) = (0, 10^3)$. Solution error is measured in Euclidean norm, comparing the approximation with a reference solution generated by Gauss-10 with stepsize 2^{-7}; Hamiltonian (3.23) is compared with initial value $H(x_0)$; and the angular momentum invariant (3.24) is compared with initial value $G(x_0)$. The first axis is linear, whereas the second axis is logarithmic, in contrast to Figure 3.4.

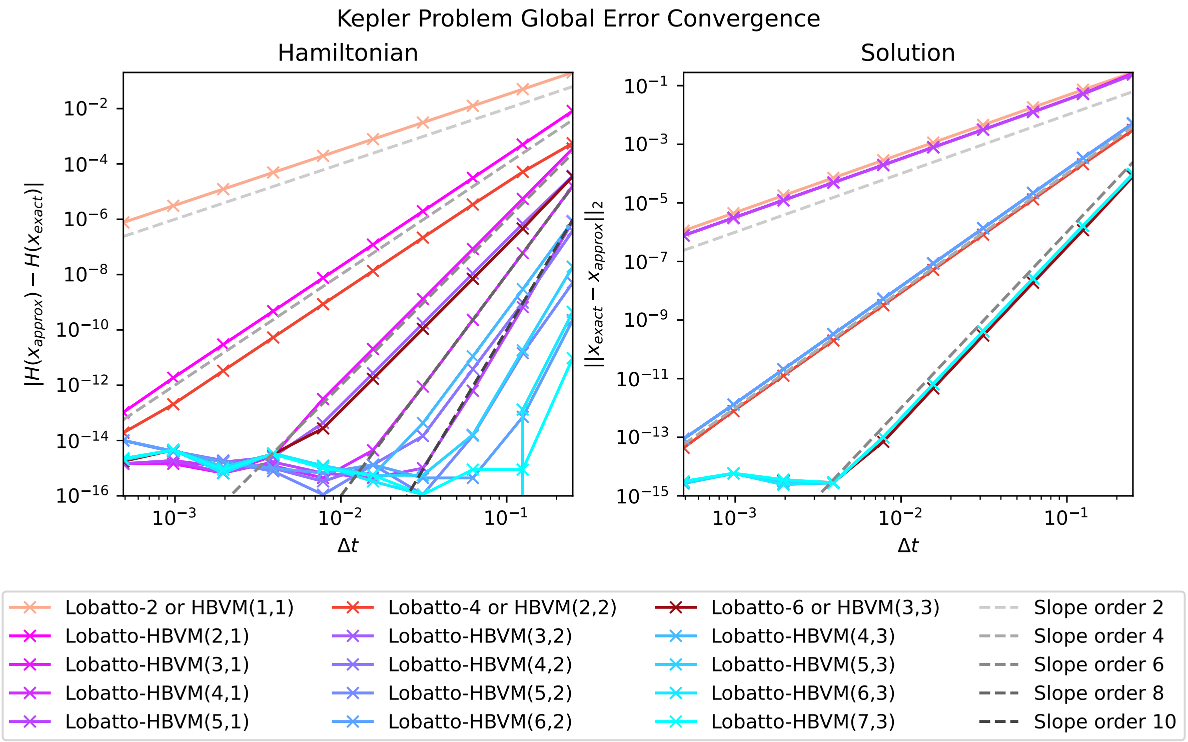


Figure 3.11: Deterministic error convergence plots of Lobatto-HBVMs for Kepler problem (3.23) with initial values (3.25) over time $t \in [0, 1]$ compared with slopes of relevant order.

Method Name	Hamiltonian		Solution	
	Expected	Observed	Expected	Observed
Lobatto-2 or HBVM(1,1)	2	2.00	2	1.99
Lobatto-HBVM(2,1)	4	4.02	2	2.04
Lobatto-HBVM(3,1)	6	6.02	2	2.03
Lobatto-HBVM(4,1)	8	8.03	2	2.03
Lobatto-HBVM(5,1)	10	10.16	2	2.03
Lobatto-4 or HBVM(2,2)	4	3.86	4	4.00
Lobatto-HBVM(3,2)	6	5.93	4	3.98
Lobatto-HBVM(4,2)	8	8.22	4	3.97
Lobatto-HBVM(5,2)	10	9.17	4	3.97
Lobatto-HBVM(6,2)	12	11.78	4	3.97
Lobatto-6 or HBVM(3,3)	6	6.05	6	6.02
Lobatto-HBVM(4,3)	8	8.09	6	5.97
Lobatto-HBVM(5,3)	10	10.09	6	5.97
Lobatto-HBVM(6,3)	12	11.73	6	5.97
Lobatto-HBVM(7,3)	14	–	6	5.97

Table 3.3: Expected and observed error convergence orders for deterministic Kepler problem (3.23) with initial values (3.25). The HBVMs reported are based on Lobatto quadrature and the order is calculated using the same data as seen in Figure 3.11.

Chapter 4

Single Integrand Hamiltonian Systems

Milstein and Tretyakov [39, Ch. 5] states that a general Stochastic Hamiltonian System (SHS) in the Stratonovich sense can be written of the form

$$\begin{aligned} \mathrm{d}Q(t) &= \nabla_p H_0(t,Q,P)\mathrm{d}t + \sum_{i=1}^{m} \nabla_p H_i(t,Q,P) \circ \mathrm{d}W_i(t), \quad Q(t_0) = q_0 \in \mathbb{R}^d, \\ \mathrm{d}P(t) &= -\left(\nabla_q H_0(t,Q,P)\mathrm{d}t + \sum_{i=1}^{m} \nabla_q H_i(t,Q,P) \circ \mathrm{d}W_i(t) \right), \quad P(t_0) = p_0 \in \mathbb{R}^d, \end{aligned} \tag{4.1}$$

for $t \in [t_0, T]$ and $W(t)$ an m-dimensional Wiener Process; or equivalently,

$$\mathrm{d}X(t) = \begin{bmatrix} \mathrm{d}Q \\ \mathrm{d}P \end{bmatrix}(t) = J\nabla_x H_0(t,X)\mathrm{d}t + \sum_{i=1}^{m} J\nabla_x H_i(t,X) \circ \mathrm{d}W_i(t), \quad X(t_0) = X_0 \in \mathbb{R}^{2d}, \tag{4.2}$$

with J the canonical structure matrix presented earlier; given that such functions as $H_i \in C^{1,1}\left([t_0,T] \times \mathbb{R}^{2d}, \mathbb{R}\right)$ for $i = 0, \ldots, m$ exist. Hong *et al.* [2] considers an autonomous canonical variant; that is,

$$\mathrm{d}X(t) = J\nabla_x H_0(X)\mathrm{d}t + \sum_{i=1}^{m} J\nabla_x H_i(X)\mathrm{d}W_i(t), \tag{4.3}$$

with

$$\nabla_x H_i(X)^T J \nabla_x H_j(X) = \{H_j, H_i\}(X) = 0 \quad \forall i,j = 1, \ldots m. \tag{4.4}$$

Said paper extends results for numerical methods presented in [6, ch. IV & VI] to this stochastic setting; a large part of the specialization project [1] was devoted to replicate the findings of Hong *et al.* [2]. However, Milstein and Tretyakov [39] and Hong *et al.* [2] only present methods of relatively low order of both weak and ms convergence, albeit symplectic.

In this chapter, a special variant of (4.3) will be discussed, for which a simple adaptation of HBVMs will achieve arbitrarily high ms order of convergence, as well as perfectly preserving polynomial stochstic Hamiltonians. Section 4.1 will present the structure and main properties of this kind of problem, while in Section 4.2.1 are found the results related to HBVMs. Section 4.3 is devoted to numerical experiments demonstrating the properties of the methodical framework.

4.1 Single Integrand Problems

With basis in (4.3), let now the Hamiltonian functions be such that that $H_i(x) = \sigma_i H(x)$ with $\sigma_0 \in \{0,1\}$ and $\sigma_i \in \mathbb{R}$ for $i = 1, \ldots, m$ where $H \in C^1\left(\mathbb{R}^{2d}, \mathbb{R}\right)$. This allows the following system reformulation

$$\mathrm{d}X(t) \begin{aligned} &= J\nabla_x H(X)\,\sigma_0 \mathrm{d}t + J\nabla_x H(X) \cdot \sigma^T \circ \mathrm{d}W(t) \\ &= J\nabla_x H(X) \cdot \left(\sigma_0 \mathrm{d}t + \sigma^T \circ \mathrm{d}W(t)\right), \end{aligned} \qquad \sigma_0 \in \{0,1\},\ \sigma \in \mathbb{R}^m. \tag{4.5}$$

This is an instance of *single integrand SDEs*:

$$\mathrm{d}X(t) = f(t,X) \circ \mathrm{d}\mu(t), \qquad t \in [t_0, T], \quad X(t_0) = X_0 \in \mathbb{R}^d, \tag{4.6a}$$

where

$$\mathrm{d}\mu(t) := \sigma_0 \mathrm{d}t + \sigma^T \mathrm{d}W(t), \quad \sigma_0 \in \{0,1\}, \sigma \in \mathbb{R}^m, \tag{4.6b}$$

with $W(t)$ an m—dimensional Wiener Process. In integral form, this becomes

$$X(t) = X_0 + \int_{t_0}^{t} f(X(\tau)) \circ \mathrm{d}\mu(\tau), \qquad t \in [t_0, T], \tag{4.7}$$

making the name self-evident.

As noted by Debrabant and Kværnø in the same paper, any single integrand problem with an m-dimensional Wiener Process can be equivalently stated in terms of a modified diffusion constant and scalar Wiener Process

$$\sigma := \sqrt{\sum_{i=1}^{m} \sigma_i^2}, \qquad W := \frac{1}{\sigma} \sum_{i=1}^{m} \sigma_i W_i. \tag{4.8}$$

This kind of problem arises in several applications, of which Debrabant and Kværnø lists several. In Section 4.2, SHSes will be considered.

4.1.1 Numerical Methods on Single-Integrand Problems

To solve single integrand problems, one could apply SRK methods. For general Stratonovich SDEs with m-dimensional noise,

$$\mathrm{d}X(t) = g_0(t,X)\mathrm{d}t + \sum_{i=1}^{m} g_i(t,X) \circ \mathrm{d}W_i(t) \tag{4.9}$$

an SRK method with s stages would be given as [2, 41]

$$\begin{aligned} Y_k &= X_n + \sum_{i=0}^{m} \sum_{j=1}^{s} Z_{jk,i}\; g_i(Y_j) \quad \text{for } k = 1, \dots, s \\ X_{n+1} &= X_n + \sum_{i=0}^{m} \sum_{k=1}^{s} z_{k,i}\; g_i(Y_k) \end{aligned} \tag{4.10}$$

where Z_0, z_0 are the coefficient matrix and vector scaled by the current stepsize $\Delta t$and Z_k, z_k are randomly perturbed coefficient matrices and vectors for $k = 1, \dots m$. However, for single integrand problems, this can be reduced to methods of the form

$$\begin{aligned} Y_k &= X_n + \Delta\mu_n \sum_{j=1}^{s} a_{jk} f(Y_j) \quad \text{for } k = 1, \dots s \\ X_{n+1} &= X_n + \Delta\mu_n \sum_{k=1}^{s} b_k f(Y_k) \end{aligned} \tag{4.11}$$

as $\Delta\mu_n a_{jk} = Z_{jk,0} + \sum_{i=1}^{m} \sigma_i Z_{jk,i}$ and $\Delta\mu_n b_k = z_{k,0} + \sum_{i=1}^{m} \sigma_i z_{k,i}$; here, $\Delta\mu_n$ refers to a discrete approximation of $\mathrm{d}\mu(t)$ for a timestep Δt_n at time t_n, i.e.

$$\Delta\mu_n = \Delta t_n + \sigma(W(t + \Delta t) - W(t)) \approx \Delta t_n + \sigma \Delta W_n \tag{4.12}$$

with ΔW_n the simulated Wiener Process increment over the interval $[t_n, t_n + \Delta t]$. As noted by Debrabant and Kværnø [5], the coefficient matrix and vector A and b can typically be chosen as those of a deterministic RK method.

The following Theorem is a slightly shortened version of the main result of [5].

Theorem 4.1. *Consider a single integrand problem*

$$dX(t) = f(X(t)) \circ d\mu(t), \quad t \in [t_0, T], \quad X(t_0) = X_0 \in \mathbb{R}^d. \tag{4.13}$$

If f satisfies [5, Assumption 4.1], then any SRK method (4.11) *applied to* (4.13) *of deterministic order p_d is of ms as well as weak order of accuracy $p_\mu = \lfloor p_d/2 \rfloor$.*

Remark. This result is very convenient. Methods for general Stratonovich SDEs typically rely on approximations to the Stratonovich–Taylor/Wagner–Platen Series, which for methods of order greater than 1 becomes highly involved. As already seen, the SDE equivalent of RK methods involves a (possibly) unique coefficient matrix and vector for every noise term. For single integrand problems, the deterministic B-series and RK coefficients suffice [5, 49].

4.1.2 Implicit Methods and Truncated Wiener Processes

The RK methods considered in this thesis are all implicit. When applying implicit methods to SDEs with multiplicative noise, it important to ensure that the approximated solution stays bounded in expectation. In [39, ch. 1.3.3], Milstein and Tretyakov demonstrates that with the standard way of simulating Wiener Process, i.e.

$$\mathrm{d}W(t) \approx \Delta W = \sqrt{\Delta t}\xi, \quad \xi \sim N(0,1)$$

implicit methods generates solutions with unbounded expectation for arbitrarily small step-sizes Δt. To counter this problem, they propose to use a *truncated* alternative to ξ, namely

$$\zeta_{\Delta t} = \begin{cases} \xi, & |\xi| \leq A_{\Delta t} \\ \operatorname{sgn}(\xi) A_{\Delta t} & \text{otherwise} \end{cases}$$

where $A_{\Delta t} > 0$. In addition, they demand that

$$\mathrm{E}\big[(\xi - \zeta_{\Delta t})^2\big] \leq \Delta t^k \quad \text{for } k \geq 1. \tag{4.14}$$

Looking at the second moment of the difference between ξ and $\zeta_{\Delta t}$,

$$\begin{aligned} \mathrm{E}\big[(\xi - \zeta_{\Delta t})^2\big] &= \int_{-\infty}^{\infty} \underbrace{(x - x_{\Delta t})^2}_{=0 \text{ for } |x| \leq A_{\Delta t}} \frac{1}{2\pi} \mathrm{e}^{-\frac{x^2}{2}} \mathrm{d}x \\ &= \frac{2}{2\pi} \int_{A_{\Delta t}}^{\infty} (x - A_{\Delta t})^2 \, \mathrm{e}^{-\frac{x^2}{2}} \mathrm{d}x \\ &= \frac{2}{2\pi} \mathrm{e}^{-A_{\Delta t}^2/2} \int_0^{\infty} y \mathrm{e}^{-\frac{y^2}{2}} \mathrm{e}^{-A_{\Delta t} y} \mathrm{d}y \\ &= \frac{1}{\sqrt{2\pi}(A_{\Delta t} + 1)^{\frac{3}{2}}} \mathrm{e}^{-A_{\Delta t}^2/2} \quad < \quad \mathrm{e}^{-A_{\Delta t}^2/2}. \end{aligned}$$

For the integration, it is assumed that $A_{\Delta t} < \pi/2$. Imposing (4.14),

$$\mathrm{E}\big[(\xi - \zeta_{\Delta t})^2\big] < \mathrm{e}^{-A_{\Delta t}^2/2} = \Delta t^k, \quad k \geq 1 \quad \Rightarrow \quad A_{\Delta t} = \sqrt{2k|\ln \Delta t|}.$$

The choice of k would depend on the ms order of accuracy the method should achieve, with the natural choice for single integrand problems being the order p_μ of the method solving (4.13).

4.1.3 Geometric Integration of Single Integrand Problems

As observed by Cohen and Dujardin [59], the noise introduced in single integrand problems does in some sense respect the geometric structure of the phase space. This, as well as the result of Debrabant and Kværnø [5], is further elaborated by Cohen *et al.* [49]. In short, Cohen *et al.* states that the geometric properties of a numerical time integrator of ODEs carries over to the anologous single integrand problem, albeit with the reduction in order as described in Theorem 4.1. Examples of time integrator properties listed by Cohen *et al.* with particular relevance here are symmetry ("self-adjointness"), symplecticity, as well as energy and invariant preservation.

4.2 Hamiltonian Systems

Returning to the context of Hamiltonian systems and using the rationale from Section 4.1, (4.5) can be restated as in the following definition.

Definition 4.1. A Single Integrand Hamiltonian System (SIHS) is an SDE of the form

$$\mathrm{d}X(t) = J\nabla_x H(X) \circ \mathrm{d}\mu(t), \qquad \mathrm{d}\mu(t) = \sigma_0 \mathrm{d}t + \sigma \mathrm{d}W(t), \quad t \in [t_0, T],\, X(t_0) \in \mathbb{R}^{2d} \tag{4.15}$$

where $\sigma_0 \in \{0,1\}$, $\sigma \in \mathbb{R}$ and $W(t)$ is a scalar Wiener Process.

Remark. This is hardly the only case of SHSes of the form (4.3) which satisfies the condition (4.4). For instance, one could consider a coupled N-body problem[1] where each body is perturbed independently, which can be formulated as

$$\mathrm{d}X(t) = J\nabla_x H(X)\mathrm{d}t + \sum_{i=1}^{N} \nabla_x H_i(X)\mathrm{d}W_i(t)$$

where with a coupling term H_c

$$H(X) = H_c(X) + \sum_{i=1}^{N} H_i(X), \qquad \{H_i, H_j\}(x) = 0 \quad \forall x \in \mathbb{R}^{2d},\, i,j = 1,\dots,N.$$

Another hypothetical counterexample would be that noise is introduced in an invariant of the deterministic problem. One particular instance would be a stochastic variant of the Central Force Problem in $\mathbb{R}^3$ with noise acting on the angular momenta, i.e.

$$\begin{aligned}\mathrm{d}Q(t) &= \nabla_p H(Q,P)\mathrm{d}t + \sum_{i=1}^{3} \nabla_p L_i(Q,P) \circ \mathrm{d}W_i(t),\\ \mathrm{d}P(t) &= -\nabla_q H(Q,P)\mathrm{d}t - \sum_{i=1}^{3} \nabla_q L_i(Q,P) \circ \mathrm{d}W_i(t),\end{aligned}$$

with

$$\begin{aligned}H(Q,P) &= \frac{1}{2}P^T P - \frac{1}{\|Q\|},\\ L(Q,P) &= Q \times P \in \mathbb{R}^3.\end{aligned}$$

Such problems are beyond the scope of this thesis, however; they are only provided for demonstration.

[1] cf. [7, ch.4.5.3 & 6.5.2]

4.2.1 Application of Hamiltonian Boundary Value Methods

Combining the results of Section 3.2 and 4.1, this theorem follows.

Theorem 4.2. *Consider a HBVM with* s *active and* k *total stages based on Gaußian quadrature. When applied to canonical SIHSes* (4.15)*, i.e.*

$$dX(t) = J\nabla_x H(X) \circ d\mu(t), \qquad t \in [t_0, T], \quad X(t_0) \in \mathbb{R}^{2d}, \tag{4.17}$$

it holds that

1. *the method has ms order of accuracy s*
2. *for general Hamiltonian functions, the weak order of accuracy is k*
3. *for $H \in \mathbb{P}_\nu$, where $\nu \leq 2k/s$, the energy is preserved exactly*
4. *the method is symmetric*

Remark. These results follows directly by combining the results of [4, 5, 49]. Recall from Definition 2.34 that a method has weak order k if

$$|\mathrm{E}[g(X(T))] - \mathrm{E}[g(Y(T))]| \leq K(\Delta t)^k, \quad K \geq 0,$$

where $g : \mathbb{R}^d \to \mathbb{R}$ is of a sufficiently large class of functions, $X(t)$ refer to the exact solution and $Y(t)$ refers to an approximate solution generated by the method. By setting $g = H$ the result follows immediately.

For HBVMs based on Lobatto quadrature, one gets the following result (remember that it has an extra stage for the same order method).

Lemma 4.3. *Theorem 4.2 holds for any HBVM(k,s) based on Lobatto quadrature as well.*

Related Results

Several generalizations of HBVMs have been mentioned in Section 1.1.1. Continuing the line of reasoning from [49], all their properties should be similarly transferrable to single integrand problems. In fact, some of these generalizations have already been tested on said problems: Li *et al.* [50] presents results for EQUIP methods.

4.3 Numerical Tests

Here, SIHSes of the form (cf. equation 4.15)

$$\mathrm{d}X(t) \begin{aligned} &= J\nabla_x H(X)\,(\mathrm{d}t + \sigma \circ \mathrm{d}W(t)) \\ &= J\nabla_x H(X) \circ \mathrm{d}\mu(t), \end{aligned} \qquad t \in [t_0, T], \quad x_0 \in \mathbb{R}^{2d}, \tag{4.18}$$

with $\sigma \in \mathbb{R}$ and $W(t)$ a scalar Wiener Process are simulated for different canonical Hamiltonian problems H and with J as before. The methods are applied to problems with the same Hamiltonians as in Section 3.3. For the convergence plots, confidence intervals are calculated following the procedure presented by Kloeden and Platen [37, ch.9.3 & 9.4]; the methods are approximately integrated between $(t_0, T) = (0, 1)$ with other experiment features as described in Table 4.1 and respective figure legends. As before, the convergence results found this section include HBVMs of only one quadrature basis, alternating between Gauß and Lobatto. The results for the other sets of methods are found in Appendix B as before.

Problem	σ_0	σ	Batches	Batch simulations	Total	Confidence
Hénon–Heiles (4.21)	1	1	20	5000	100 000	> 99.9%
H order 6 (4.22)	1	1	20	5000	100 000	99.0%
Kepler (4.23)	1	1	20	5000	100 000	99.0%

Table 4.1: Problem variables for SIHS error convergence plots.

4.3.1 Kubo Oscillator

The stochastic variant of the Harmonic Oscillator presented in Section 3.3.1 is also called the Kubo Oscillator, and is often used as a numerical test example [39, ch. 5.4.1]; it was also used as a test problem in the specialization project [1]. Its general formula can be written as

$$\mathrm{d}X(t) = J\,X(t) \circ \mathrm{d}\mu(t), \quad \mathrm{d}\mu(t) = \sigma_0 \mathrm{d}t + \sigma \mathrm{d}W(t), \quad t \in [t_0, T], \quad X(t_0) = x_0 \in \mathbb{R}^2 \qquad (4.19)$$

with J the canonical structure matrix, $\sigma_0, \sigma \in \mathbb{R}$ and Hamiltonian $H(X) = \frac{1}{2}\|X\|_2^2$. As for the deterministic problem, an exact solution is known, which for initial values $x_0 = [0,1]^T$ is

$$X(t) = \begin{bmatrix} \sin(\sigma_0 t + \sigma W(t)) \\ \cos(\sigma_0 t + \sigma W(t)) \end{bmatrix}. \qquad (4.20)$$

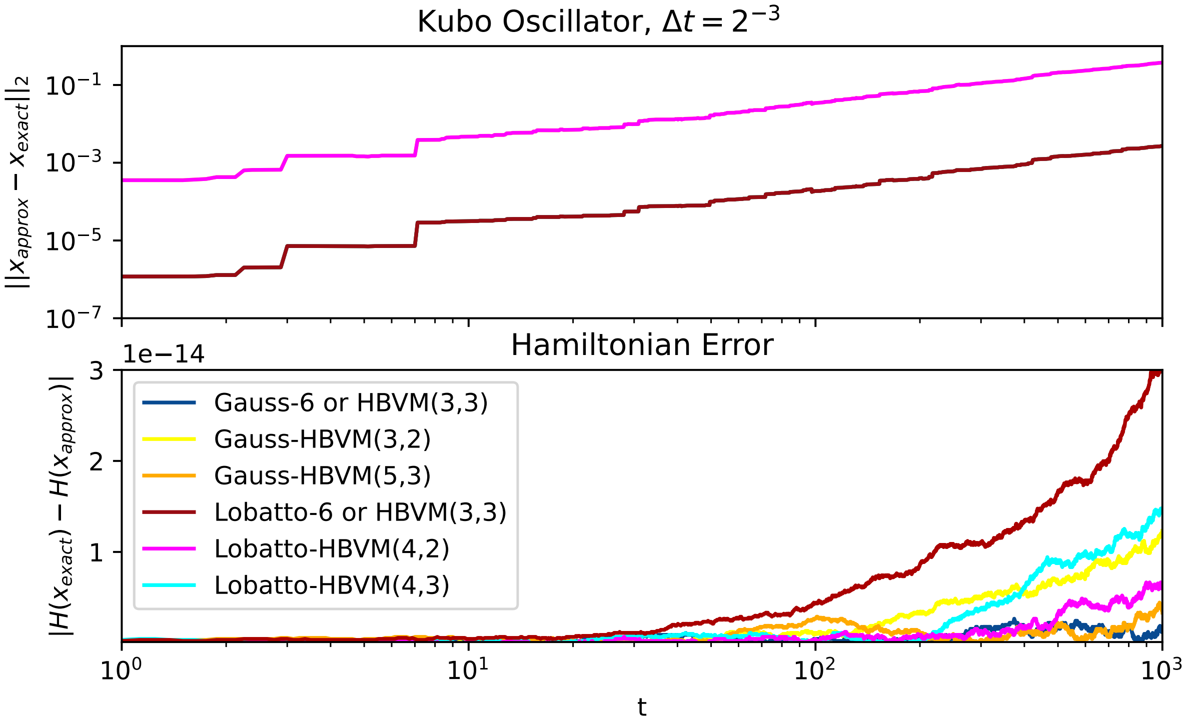


Figure 4.1: Long term error measured in solution and Hamiltonian for a HBVMs applied to single realization of Kubo oscillator (4.19).

In Figure 4.1, the error of different HBVMs applied to the same realization of the problem is plotted. The same stepsize has been used in Figure 4.1 as in Figure 3.4 for the problem's deterministic equivalent. The solution error grows almost linearly but jumps interspersedly, and it is two orders larger than for the Harmonic Oscillator. The Hamiltonian is still close to zero (of magnitude 10^{-14}) and moving randomly as for the deterministic variant, indicating conservation. This larger and slightly more erratic error behaviour is caused by the stochastic nature of the problem. Of course, only looking at one realization is not enough to draw conclusions on the properties of the method; further experiments are needed.

4.3.2 Hénon–Heiles

Here, the Hénon–Heiles problem of Section 3.3.2 with Hamiltonian (3.20), which reads

$$H(X) = H(Q,P) = \frac{1}{2}\left(\|Q\|^2 + \|P\|^2\right) + \alpha\left(Q_1 Q_2^2 - \frac{1}{3}Q_1^3\right) \tag{4.21}$$

is considered with $\alpha = 16$ and the initial value used is $x_0 = [\sqrt{3}, 1, 1, 1]^T$.

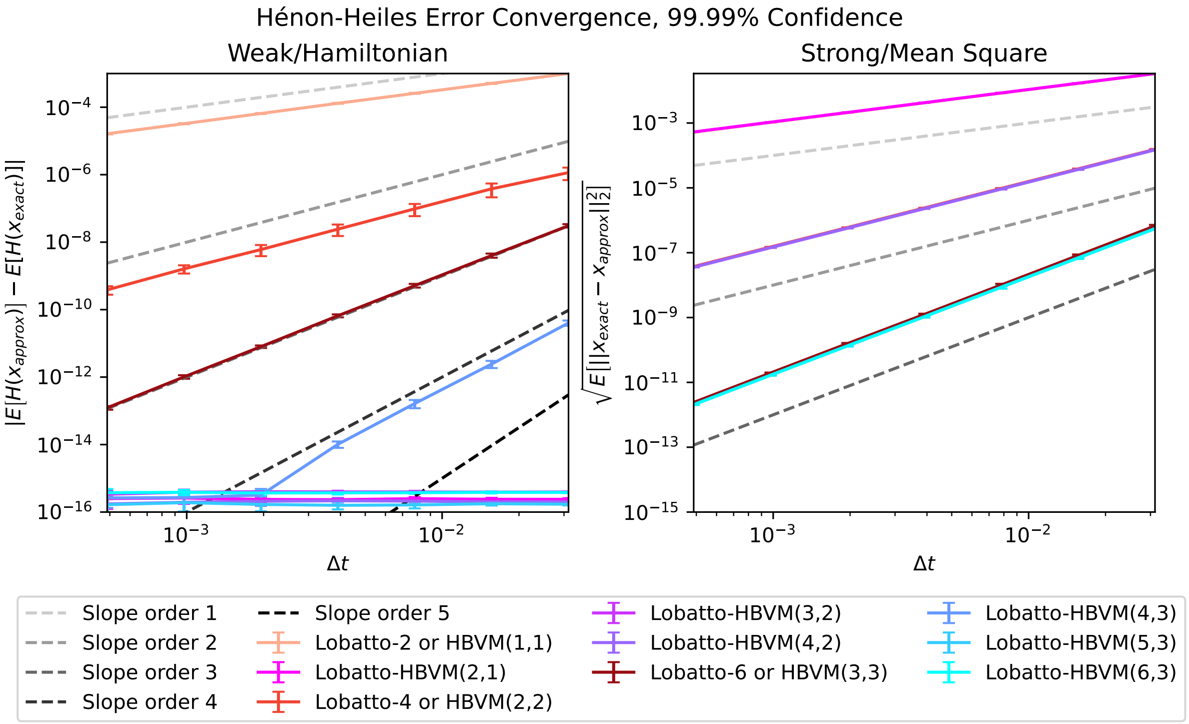


Figure 4.2: Convergence plots for single integrand Hénon–Heiles problem (4.21) for Lobatto-HBVM(k,s) for $s \in \{1,2,3\}$ and k as described in legend. The parameters used are collected in Table 4.1.

Method Name	Hamiltonian		Solution	
	Expected	Observed	Expected	Observed
Lobatto-2 or HBVM(1,1)	1	0.99	1	1.00
Lobatto-HBVM(2,1)	–	–	1	1.00
Lobatto-4 or HBVM(2,2)	2	1.93	2	2.00
Lobatto-HBVM(3,2)	–	–	2	2.00
Lobatto-HBVM(4,2)	–	–	2	2.00
Lobatto-6 or HBVM(3,3)	3	2.99	3	3.01
Lobatto-HBVM(4,3)	4	3.98	3	3.01
Lobatto-HBVM(5,3)	–	–	3	3.01
Lobatto-HBVM(6,3)	–	–	3	3.01

Table 4.2: Expected and observed weak and strong order of convergence for HBVMs based on Lobatto quadrature applied to the single integrand Hénon–Heiles problem. The observed order is calculated using the same data from which Figure 4.2 was made.

In Figure 4.2, both ms and weak (Hamiltonian) error convergence for of Lobatto-HBVMs are shown, with the ms error calculated by comparing the approximation to a reference solution generated using Gauß-10 and smaller stepsize. The expected and observed error convergence orders for the solution, in the strong or ms sense, and the Hamiltonian, in the weak sense, are reported in Table 4.2.

The collocation methods maintain the same strong and weak order of convergence, whereas the HBVMs with extra stages (i.e. $k \geq s$) have the Hamiltonian error down to machine precision for all stepsizes. The exception to this is HBVM$(4,3)$ (for both Gauß and Lobatto quadrature), which converges with order four. This matches the expected results for convergence of $\mathcal{O}\left(\Delta t^k\right)$ rather than $\mathcal{O}(\Delta t^s)$. The phenomenon of closely matching global error size for methods of the same order that was seen for the deterministic case (Section 3.3) is also observed in the ms error, which is its stochastic analogue.

4.3.3 Polynomial Hamiltonian of Degree Six

The experiments of the previous section are now repeated on the problem with Hamiltonian of 6th degree seen in Section 3.3.3, i.e.

$$H(X) = H(Q,P) = \frac{1}{3}P^3 - \frac{1}{2}P + \frac{1}{30}Q^6 + \frac{1}{4}Q^4 - \frac{1}{3}Q^3 + \frac{1}{6}, \quad x_0 = [0,1]^T. \tag{4.22}$$

As noted there, an additional $3s$ silent stages are needed to conserve the Hamiltonian exactly. Error convergence results are shown in Figure 4.3 and Table 4.3. For this problem as well as the SIHS Kepler problem, fewer stepsizes are included in the plots, as the numerical solution of the methods with the lowest order blows up with larger stepsizes.

Some point values for the largest stepsizes with unreasonably large Hamiltonian error variance has been removed from the plots and order calculations. This high variance, as well as somewhat erratic error convergence, can be attributed to the Newton iterations not converging fully in each timestep. Therefore the irregular behaviour does not reflect the methods' "true" properties. It is only the values of HBVM$(3,1)$, HBVM$(5,2)$ and HBVM$(6,2)$ that are affected. When the offending points are removed, they demonstrate the expected error convergence.

From the table and the plots, one can observe that the methods demonstrate ms order very close to one, two and three for methods with parameter $s = 1,2,3$ respectively. For the Hamiltonian error measured in the weak sense, the methods with $k = 3s$ preserves the Hamiltonian down to machine precision (reported as "–" in the tables). The error convergence order for methods with $k < 3s$ is almost the same as is expected, but consistently deviating more from the expectation for larger number of silent stages $r = k - s$. HBVM$(6,3)$ does have slightly higher convergence, but the value which causes this behaviour is more uncertain than the rest, most likely due to non-converging Newton iterations. HBVM$(7,3)$ preserves the Hamiltonian for the stepsizes considered here, even though it is not expected to do so in general.

In Figure 4.4, the solution to the SIHS approximated by the conservative Lobatto-HBVM$(6,2)$ (Figure 4.4a) and non-conservative Lobatto-HBVM$(2,2)$ (Figure 4.4b) is plotted against the contours of the Hamiltonian function. The order of the methods, the stepsize and the time interval is identical to the method seen in Figure 3.7, where the deterministic equivalent of the problem was solved.

The individual points of the approximated solutions, connected with light blue and red lines in the plot, are now of a greater distance between each other and more randomly dispersed; for the ODE, these lines were completely covered by the points. Even so, the points generated by the conservative HBVM$(6,2)$ stays neatly on the same level set. The Lobatto method of the same order does not manage to preserve the Hamiltonian, however, veering significantly away from the expected level in both directions. This supports the convergence findings.

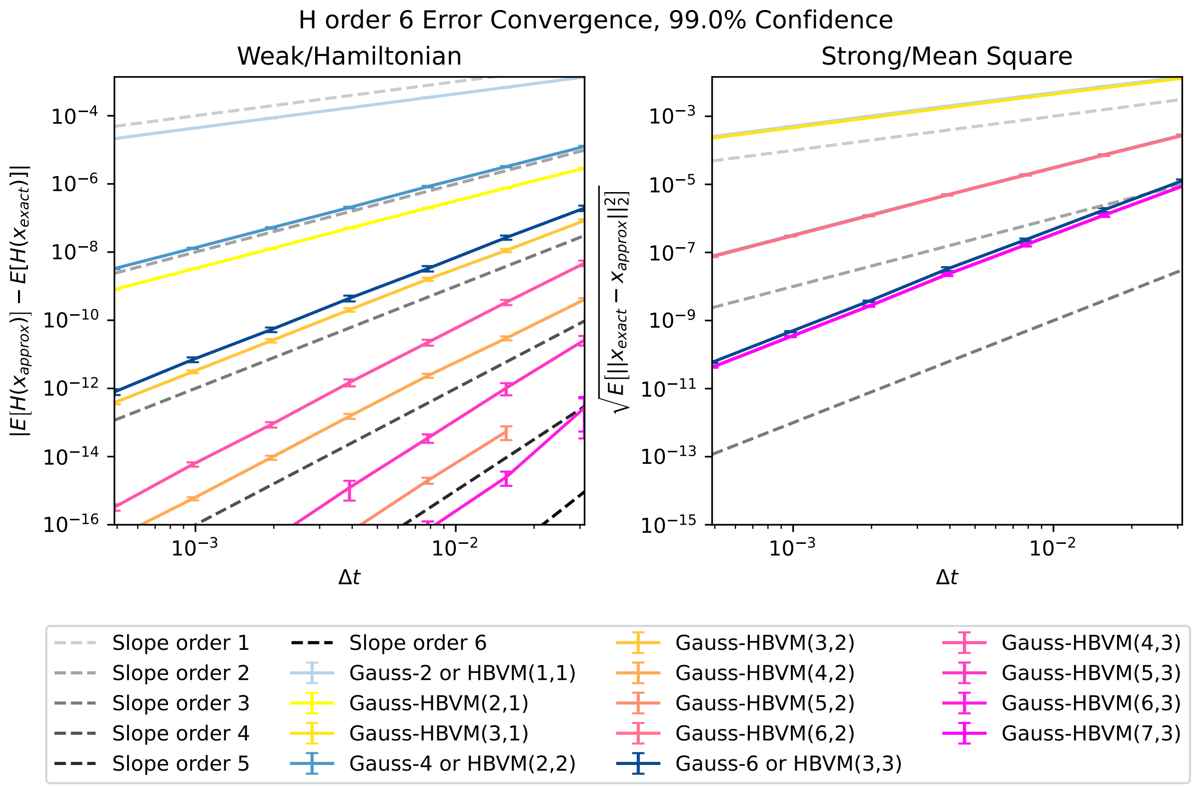
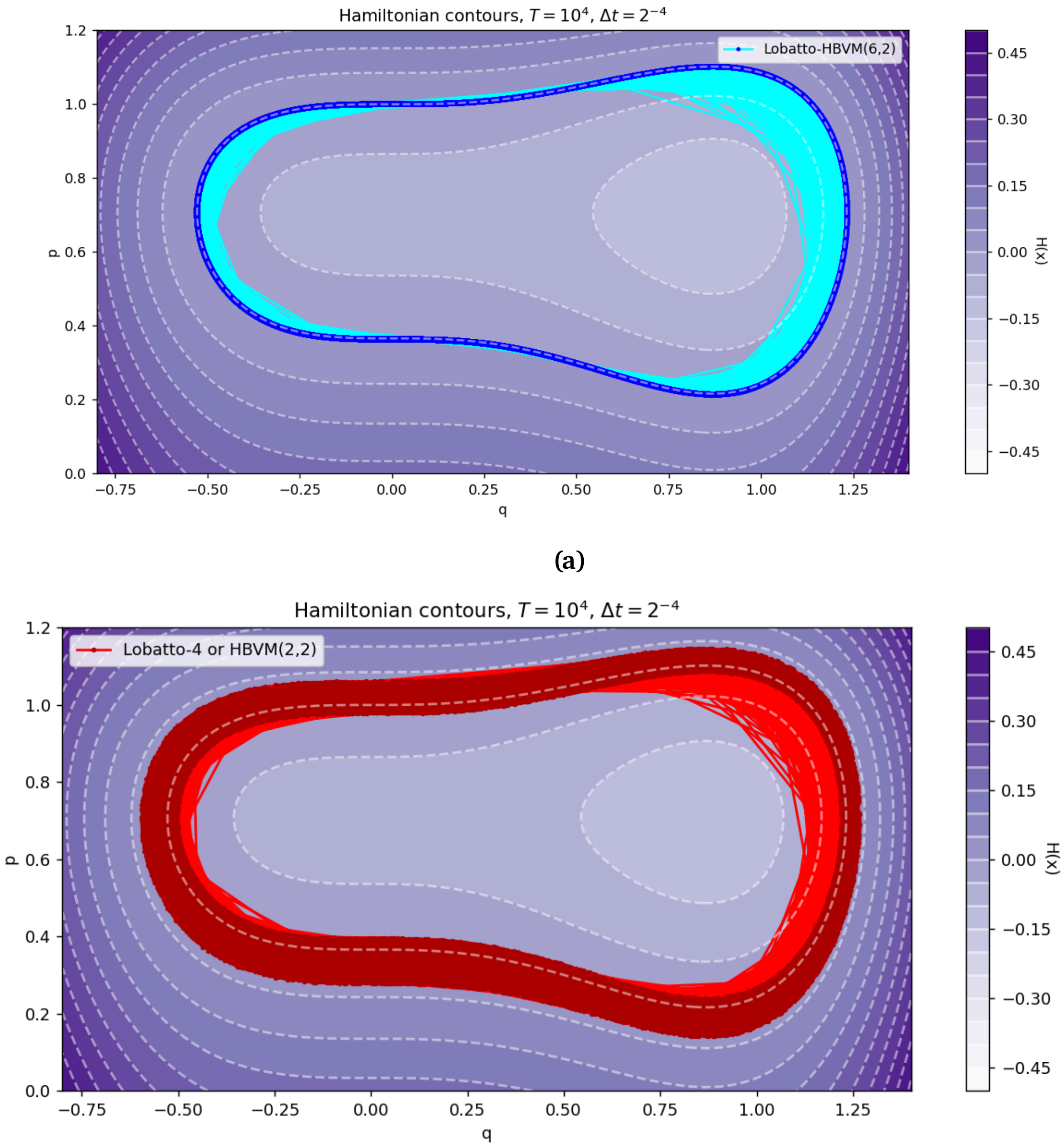


Figure 4.3: Weak and strong error convergence of HBVMs based on Lobatto quadrature in weak (Hamiltonian) and strong sense for the canonical single integrand problem with H and initial values as in (4.22). The methods are based on parameters $s \in \{1, 2, 3\}$ and k from $k = s$ (standard collocation) and up to theoretical conservation.

Method Name	Hamiltonian		Solution	
	Expected	Observed	Expected	Observed
Gauss-2 or HBVM(1,1)	1	1.00	1	0.97
Gauss-HBVM(2,1)	2	1.97	1	0.97
Gauss-HBVM(3,1)	–	–	1	0.97
Gauss-4 or HBVM(2,2)	2	1.98	2	1.97
Gauss-HBVM(3,2)	3	2.96	2	1.97
Gauss-HBVM(4,2)	4	3.88	2	1.97
Gauss-HBVM(5,2)	5	4.75	2	1.97
Gauss-HBVM(6,2)	–	–	2	1.97
Gauss-6 or HBVM(3,3)	3	2.99	3	2.95
Gauss-HBVM(4,3)	4	3.96	3	2.94
Gauss-HBVM(5,3)	5	4.80	3	2.94
Gauss-HBVM(6,3)	6	6.77	3	2.94
Gauss-HBVM(7,3)	7	–	3	2.94

Table 4.3: Expected and observed weak (Hamiltonian) and strong (ms) order of HBVMs based on Gauß quadrature applied to sixth degree H from (4.22). The observed orders are calculated from the same data as the plots in Figure 4.3.



(a)

(b)

Figure 4.4: Contour plots of H of SIHS (4.22) and approximation of one realization generated by Lobatto-HBVMs with $(\sigma_0, \sigma) = (1,1)$. Further features are found in figure title and legend.

4.3.4 Kepler Problem

For the single integrand stochastic variant of the Kepler problem of 3.3.4, which has the Hamiltonian and angular momentum invariants

$$H(X) = H(Q,P) = \frac{1}{2}P^T P - \frac{1}{||Q||}, \qquad G(X) = G(Q,P) = Q_1 P_2 - P_1 Q_2 \tag{4.23}$$

the intial values of (3.25) are again used:

$$x_0^T = \left[x_0^T, x_0^T\right] = \left[1-e, \quad 0, \quad 0, \quad \sqrt{\frac{1+e}{1-e}}\right], \qquad e = 0.5. \tag{4.24}$$

This problem was also approximated by Li *et al.* [46]. A long term error plot for a single realization approximated by the same methods as applied in Figure 3.10 for the SIHS is presented in Figure 4.5. As in Figure 4.1, the error is much larger than for the corresponding ODE, and while the Gauß-8 has slightly better performance in terms of conserving the Hamiltonin, their long term behaviour is still comparative. Only the Gauß collocation method preserves the angular momentum, which is expected.

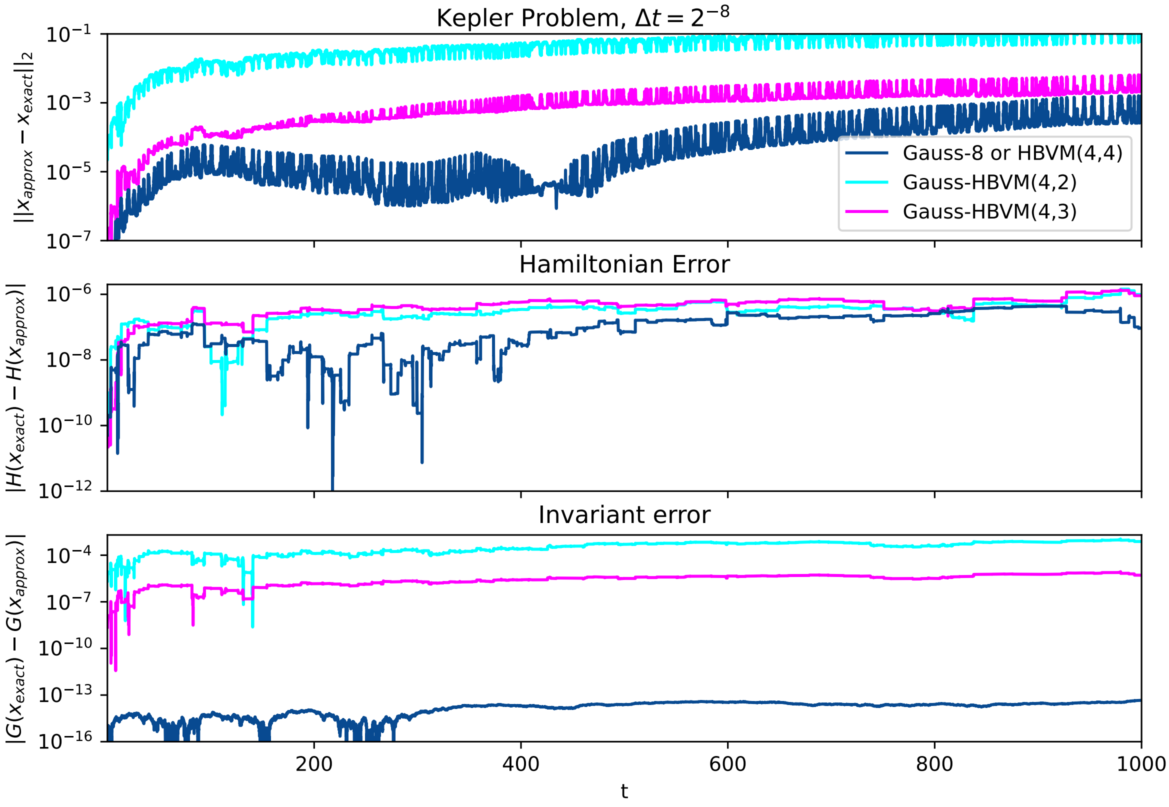


Figure 4.5: Long term error development for selected methods on the SIHS Kepler problem (4.23) with initial values (4.24).

The error convergence results are shown in Figure 4.6 and Table 4.4. The reference solution is generated with Gauß-10 (of order five for single integrand problems) and a smaller stepsize than used in the plots. As for Figure 3.8 and Table 3.2, some values with high error variance are removed from the plots. The reasoning is the same as for the above: Their greater variance is most likely caused by non-converging Newton iterations, which would cause incorrect qualitative behaviour. Besides, the plots are already crowded enough as is.

As before, the HBVMs achieve ms order very close to the theoretical order. Again, the Hamiltonian error in the weak sense error is slightly worse than the theoretical order for an increasing number of extra stages. HBVM$(7,3)$ virtually preserves energy for all but the largest stepsize.

One point to consider is that the reference solution is only of weak order five, whereas HBVM$(6,2)$, HBVM$(6,3)$ and HBVM$(7,3)$ are expected to be of weak order six and seven. The step size used is still four times smaller than the smallest used in calculating the convergence — it should not influence the data much.

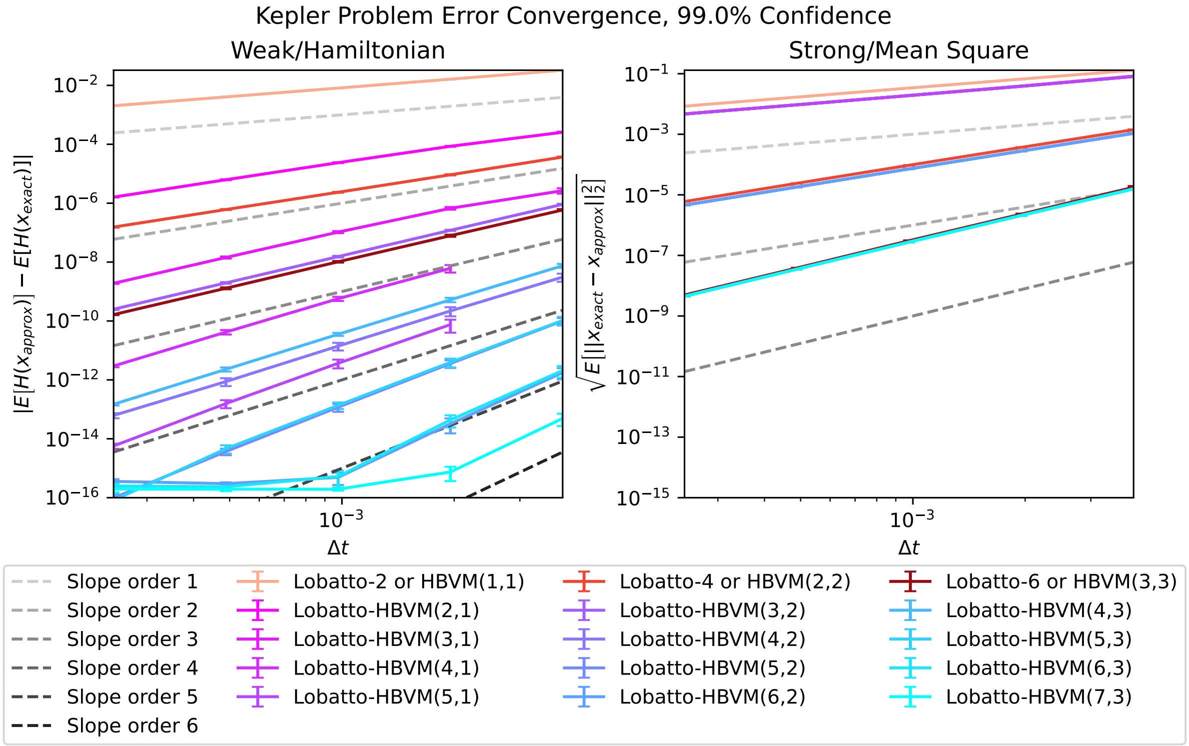


Figure 4.6: Convergence plots for SIHS Kepler problem (4.23) with initial values (4.24) for Lobatto-HBVM(k,s) with $s \in \{1,2,3\}$ and k up to conservation.

Method Name	Hamiltonian		Solution	
	Expected	Observed	Expected	Observed
Lobatto-2 or HBVM(1,1)	1	1.00	1	0.99
Lobatto-HBVM(2,1)	2	1.83	1	1.04
Lobatto-HBVM(3,1)	3	2.61	1	1.03
Lobatto-HBVM(4,1)	4	3.67	1	1.03
Lobatto-HBVM(5,1)	5	4.56	1	1.03
Lobatto-4 or HBVM(2,2)	2	1.98	2	1.97
Lobatto-HBVM(3,2)	3	2.96	2	1.97
Lobatto-HBVM(4,2)	4	3.90	2	1.97
Lobatto-HBVM(5,2)	5	4.91	2	1.97
Lobatto-HBVM(6,2)	6	5.78	2	1.97
Lobatto-6 or HBVM(3,3)	3	2.95	3	2.95
Lobatto-HBVM(4,3)	4	3.91	3	2.94
Lobatto-HBVM(5,3)	5	4.82	3	2.94
Lobatto-HBVM(6,3)	6	5.57	3	2.94
Lobatto-HBVM(7,3)	7	–	3	2.94

Table 4.4: Expected and observed weak (Hamiltonian) and strong (ms) error convergence order of Lobatto-HBVMs applied to the SIHS Kepler problem. The same data is used to calculate the observed order as to generate the plots in Figure 4.6.

Chapter 5

Implementation

When simulating SDEs, one are generally concerned with problems of very large scale. In addition to carring the original dimensionality of the deterministic problem, a finer temproral discretization is necessary in general to ensure both the stability of the numerical method as well as to keep the error at a comparable size. However, one is generally not concerned with a single solution given a set of initial conditions, or alternatively comparing the performance of the method across a specter of intial conditions. Rather, the interest is in the properties of the distribution of the stochastic process which arises from such deterministic initial conditions.

To get sufficiently accurate measurements of the distribution features, one needs to simulate randomly sampled paths numbering in the thousands or even millions. Besides the need for good (pseudo-)Random Number Generators (RNGs) which generate practically independent random samples from a Gaußian distribution for the Wiener increments, this places large constraints on both memory usage and and computational efficiency. However, while each timestep of a one-step method is completely depenent upon the last timestep, each realization is completely independent of any other sample path. Consequently, the problem of simulating many paths becomes embarrassingly parallel.

`python`, consistently ranking as the most popular general-purpose programming language in the world [60–62], isn't very competitive in terms of computation speed out of the box. Even with the commonly used `numpy` [63] and `scipy` [64], with backend consisting of `Fortan` and different variants of `C` (`C`, `C++` and `Cython`), the performance pales in comparison with pure `C`, `C++` or `Fortran`. To make `python` competitive, one should turn to a variant of Just In Time (JIT) or Ahead Of Time (AOT) compilation. The `numba` library [65] offers ways to JIT-compile most `numpy` and `scipy` functions using simple function decorators, and also has easy to understand parallel functionality. However, it does not support `scipy`'s efficient LU-factorization, which is used in the implementation here.

Another option is the Domain Specific Language (DSL) `jax` [66], developed by Google, which offers both JIT-compilation with similar syntax to `numba` and intuitive paralell mapping. In fact, it offers its own very efficient implementation of many `scipy` and `numpy` functions, as well as forward and backward automatic differentiation functionality working across many types of programming structures and functions. What is more, it composes or "merges" operations during compilation to reduce the total number of computations. It is designed to accelerate code on Graphical Processing Units (GPUs) and Tensor Processing Units (TPUs) specifically, however offers a significant improvement on standard Central Processing Units (CPUs) as well, as will be demonstrated here. After encountering the performance and implementation issues mentioned in the paragraph above, the code base for this thesis was converted to `jax`. For those not interested in using `python`, but want a similarly user-friendly language, the proprietary `MATLAB`, `julia` [67] might be a feasible solution.

In this Chapter, the main features of the implementation related to this thesis is described. For the sake of transparency, source code, the specialization project report [1] as well as a version of this text can be found at [68].

5.1 Implicit HBVM Solver

5.1.1 Implicit Runge–Kutta Solver

Compared to constructing solvers for explicit RK methods, in which each stage of the method can be calculated iteratively as a single explicit expression, the procedure for general IRK methods, such as the HBVMs, is somewhat involved. In fact, for a d-dimensional differential equation and an s-stage RK method, one potentially need to solve a nonlinear system of $d \cdot s$ equations iteratively in every timestep of the approximation. Hairer and Wanner [51, Ch. IV.8] presents a strategy for solving this problem through the use of simplified Newton iterations. Note that for Hamiltonian systems, such iterations demand the availability of the Hessian of H.

For an IRK method (cf. Section 2.2.1)

$$y_i = x_n + \Delta t \sum_{j=1}^{s} a_{ij} f(t_n + c_j \Delta t, y_j), \qquad i = 1, \dots, s \tag{5.1a}$$

$$x_{n+1} = x_n + \Delta t \sum_{i=1}^{s} b_i f(t_n + c_i \Delta t, y_i), \qquad n = 0, \dots N \tag{5.1b}$$

applied to the general ODE

$$\dot{x} = f(t, x(t)), \quad x(t_0) = x_0 \in \mathbb{R}^d, \quad t \in [t_0, T]$$

the strategy per timestep is this:

1. Substitute the s stage-approximations y_i with a smaller variable $z_i = y_i - x_n$ to minimize roundoff errors
2. Use the approximation of the Jacobian $\nabla_x f(t_n + c_i \Delta t, y_n + z_i) \approx \nabla_x f(t_n, y_n) =: df_n$ for $i = 1, \dots, s$, which is constant for each timestep
3. Pick a good starting value (here, $z^0 = 0$ has been used)
4. Solve the following linear system with respect to approximation increment $\Delta Z_i \in \mathbb{R}^{d \cdot s}$:

   $$\Gamma_n \Delta Z_i := (I_{d \cdot s} - \Delta t A \otimes df_n) \Delta Z_i = -Z_i + \Delta t (A \otimes I_d) F(Z_i) =: -G(Z_i)$$

   where $Z_i = [z_1, \dots, z_s]_i^T$ and $F(Z_i) = [f(z_1), \dots, f(z_s)]^T$
5. Terminate iteration when the approximation is adequate, which is when

   $$\eta_i \|\Delta Z_i\| \leq \kappa \cdot \mathtt{Tol}, \quad \eta_i = \frac{\theta_i}{1 - \theta_i}, \quad \theta_i = \frac{\|\Delta Z_i\|}{\|\Delta Z_{i-1}\|}$$

   with $\eta_0 = \max\{\eta_{old}, U_{round}\}^{0.8}$, where U_{round} is the rounding unit, and $\kappa \in \{10^{-1}, 10^{-2}\}$.

The matrix Γ_i is constant for every timestep, so the system needs factorizing only once per timestep. Hairer and Wanner [51, Ch. IV.8] recommend allowing for up to between seven and ten iterations per timestep; if the calculation fails to terminate before that or the iteration diverges, they recommend restarting the iterations with a smaller stepsize. Here, an equidistant stepsize has been used, partially to allow for simulating Wiener Processes before the approximations and making the process replicable. If the stepsize is adjusted during the numerical integration, the simulated Wiener increment would have to be generated at each timestep. This is not impossible, but make it far more difficult to compare different methods on the same simulated Wiener Processes.

Due to some quirks in `jax`[1] and personal inexperience with the software, the approximation procedure does not terminate if the Newton iterations fail to converge. This might be a source of unexpectedly large errors, as has been pointed out in Sections 3.2.1 and 4.3, but can be avoided by initially picking a sufficiently small stepsize or adjusting the stepsize during integration, the latter option demanding that the Wiener increments are simulated consecutively.

[1] It demands highly functional and vectorized code and has very limited support for conditional statements

5.1.2 Silent Stages and Efficient Implementation

The s-stage Gauß and $s+1$-stage Lobatto collocation methods both have coefficient matrices A of rank s and are irreducible (not counting the explicit initial step of Lobatto methods). In contrast, for a HBVM(k,s), the coefficient matrix always has rank s regardless of the number of total stages k. Naively applying the procedure above to the full coefficient matrix will be highly unfeasible for high order HBVMs conserving $H \in \mathbb{P}_\nu$ exactly: as noted in Section 3.3.3, $\nu = 6$ demands three times as many stages as for the same order collocation method on which it is based.

However, as noted in virtually every paper related to HBVMs in particular and LIMs in general, it is quite possible to sidestep this issue. In fact, thinking of the method in terms of the Butcher table formulation might be counterproductive: Essentialy, $r := k-s$ stages of a HBVM(k,s) can be expressed as a linear combination of the remaining s stages. Brugnano *et al.* [26] denote the s stages *fundamental*, whereas the r subsequent stages are denoted *silent stage*. The procedure for a block-blended implementation of Gauß-HBVMs is outlined by Brugnano *et al.* [26], but is not repeated here as there was no time to implement it, ironically. For the problems and methods considered here, the extra computational cost was not debilitating, which will be demonstrated later.

5.1.3 Simulation of SDEs

To get good estimates on the properties of the distributions of stochastic processess, relatively large sample sizes of normally distributed variables has to be generated. Following the procedure of Kloeden and Platen [37, Ch. 9.3 & 9.4], it is possible to calculate $(1-\alpha)$ confidence intervals on the sample error mean by partioning the sample into a number of batches. This leads to an obvious entry point to parallelism: run each batch in parallel on different computational devices (CPUs, GPUs, TPUs). As noted before, every sample path should be independent. Consequently, no information should pass between the batches as long as they are generated independently, i.e. with different seeds and using RNGs with sufficiently large periods [39][Ch. 2.6]. Milstein and Tretyakov [39] also points out that only a small part of the total period of pseudo-RNGs should be used in a single simulation.

For replicability and accuracy of the generated samples, some caution in choice of random number generator should be applied. The `numpy` library offers random number generation, but has been modifying its routines over the last five years [69]. Its legacy methods use a bit generator based on the Mersenne Twister algorithm, which is a Generalized Feedback Shift Register generator (GSFR) with a period of $2^{19937}-1$ and has relatively good statistical properties [39, Ch. 2.6]. However, long periods are not in itself necessarily a good criterion; O'Neill [70] discusses this, as well as other flaws of the Mersenne Twister. It is also noted to often need an unecessarily large overhead in terms of time and space requirement for parallel systems [71].

As an alternative to the Mersenne Twister, O'Neill [70] proposes a family of RNGs dubbed Permuted Congruential Generators (PCGs), which sidesteps many of the issues of Linear Congruential Generators (LCGs). A 64-bit PCG is now the default RNG routine of `numpy`, also offering a version especially suited to Parallel Random Number Generation (PRNG). In contrast, `jax` generates random numbers using Threefish algorithm, which are counter based and especially designed for massively PRNG [71]. Both Threefish and PCG methods show better better statistical properties than the Mersenne Twister, while simultaneously being faster, more computationally efficient and less space-demanding, reproducible and suited to parallel computations [70, 71].

Although it should be possible to adapt stepsizes during the time integration by generating the Wiener increments in parallel and in each timestep, the process would make it harder to compare approximations from different methods along the same paths, as some methods might have converging Newton iterations, whereas others might not. It would of course be completely unfeasible when doing error convergence measurements, as the dependent variable *is* the stepsize. For orbital systems such as the Kepler problem, adaptive stepsize schemes based on known features of the system has been developed for deterministic problems, even maintaining the modified Hamiltonian of the sysetem [72]. It is not unthinkable that adaptive stepsize schemes for SIHSes might be developed in a similar fashion.

For implicit methods such as are deployed here, the Gaußian variables of the Wiener increments has to be truncated to ensure that the one-step approximation stays bounded in expectation; the procedure is outlined in Section 4.1.2.

5.2 Performance

5.2.1 Calculation of Method Coefficients

An explicit expression for an arbitrary HBVM of k stages and s fundamental stages was written in Section 3.1 as a function of matrices consisting of shifted and scaled Legendre polynomials evaluated at k abscissae, as well as some integrated terms. For lower order methods, it does not pose too hard a task to calculate these manually, but it can quickly become quite involved. There exist several open-source Computer Algebra Systems (CASs) which makes it possible to calculate the coefficients quite quickly and with little manual effort. One could for instance use `sympy` [73], which is written in pure `python` and has good support for Legendre polynomials.

With the help of `sympy`, functionality to construct Butcher tables for arbitrary HBVMs was written for this thesis. A selection of these are included in Appendix A. Although `sympy` offers symbolic algebra solvers, it doesn't allow algebraic expressions for methods based on polynomial roots higher than quartic degree found with its CAS. However, for purely numerical expressions of the matrices, the construction is relatively quick and equally precise for the use in the solver.

In Figure 5.3, timings of the algorithm constructing the Butcher tables as a function of total number of stages are averaged over ten runs, with construction of methods of the same strong order connected by lines. As can be seen, the construction time is a linear function of the number of silent stages and closer to quadratic for the number of fundamental stages. More importantly, it is relatively cost efficient, clocking in at about one minute even for HBVM$(19,10)$, which has order nineteen weak and order ten strong convergence, or deterministic global error convergence order twenty and Hamiltonian error convergence of order thirty-eight. As such, calcualting the method could demand but a small part of the total time spent generating solutions.

5.2.2 Automatic Differentiation

`jax` supports both forward and backwards differentiation with quite high precision and speed. Moreover, it can differentiate through a series of operators within its own framework.

To assess the performance of the automatic differentiation on SIHSes, several test problems are implemented both in terms of "explicit" expressions for $f = J\nabla_x H$ and $\nabla_x f = J\nabla_x^2 H$ and functions given by autmatically differentiating H once and twice for f and f. In addition to already discussed problems, the Double Well SIHS with Hamiltonian

$$H(X) = H(Q,P) = \frac{1}{2}P^2 + \frac{1}{4}Q^4 - \frac{1}{2}Q^2, \quad x_0 = [\sqrt{2}, \sqrt{2}]^T$$

is also implemented. The approximations for the two problem formulations are then compared pathwise in norm against time, resulting in the plot seen in Figure 5.1. As can be seen, the difference between the two sets of solutions is quite small. In fact, for the chosen stepsize, the difference between the solutions is much smaller than the method error (see Section 4.3), indicating that the automatic differentiation can be used relatively safely.

5.2.3 Different Solver Implementations

Combining the features described above with the automatic differentiation routines of `jax`, one can build a solver for any canonical Hamiltonian single integrand system using any HBVM has been developed. As inputs it only needs an expression for the Hamiltonian $H(x)$, the quadrature

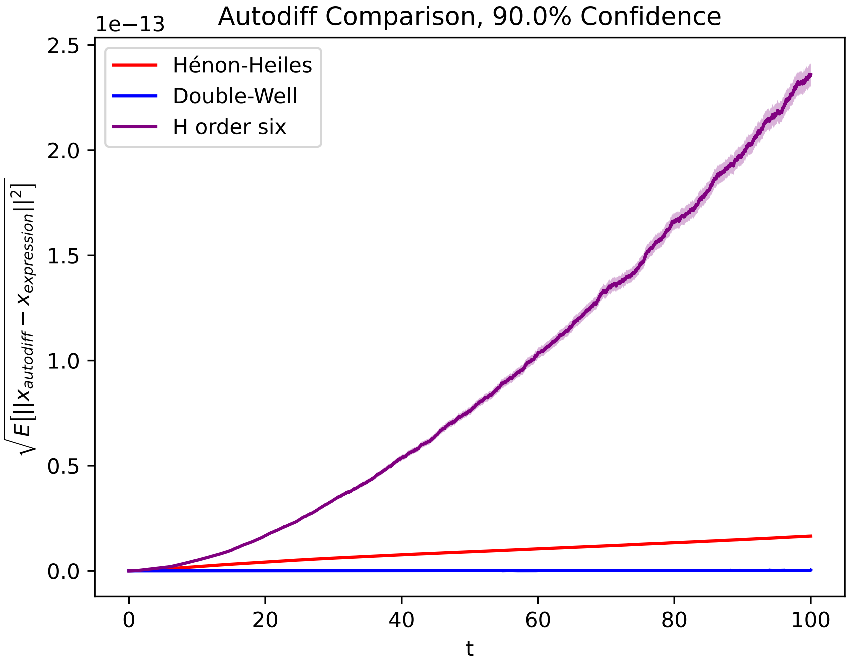


Figure 5.1: A comparison of solutions generated by Gauß-HBVM(4, 2) with and without automatic differentiation for three test problems with parameters $\sigma = 1, \Delta t = 10^{-2}, t \in [0, 100]$ over four batches with thousand simulations each. A 90% confidence interval is plotted around the solutions.

basis and the parameters k, s of the method, allowing it to solve the system for any $x_0, [t_0, T], \Delta t$ and number of simulated paths, given that the method is stable for these parameters.

To assess the performance of the "new" `jax` based implentation with and without automatic differentiation ("autodiff"), it is compared with a solver written using only `numpy` and `scipy` functions. Little work has been put into optimizing this "old" variant, yet it might be illustrative of much scientific computing programs written with the help of said packages. Importantly, it is not written for parallel computation, so it is greatly disfavoured by increasing the number of simulation batches.

In addition to comparing the speed for the `jax` implementation with both automatic differentiation and eplicitly stated functions, a "mixed" solver, where the vector field function $f = J\nabla_x H$ and its Jacobian $\nabla_x f = J\nabla_x^2 H$ is written in `jax` and JIT-compiled, while the rest of the solver backend relies on `numpy` and `scipy` routines.

All these solver configurations are then used to simulate the Hénon–Heiles problem (4.21) with variying simulation parameters which are collected in Table 5.2b. The simulation timings are averaged over 10 runs. The tests are performed on a Lenovo Yoga Slim 7 14ARE05 using a 2.00 GHz "AMD Ryzen 7 4700U with Radeon Graphics" processor with eight cores and 16 GB RAM. Figure 5.2 displays the results.

The computation time plotted against the number of silent stages, as seen in Figure 5.2a yields some weird behaviour. The new solvers are slower for fewer than three extra stages than for ten extra, oddly enough. Except for an anomalous value at three silent stages (not unexpected for only ten runs), both the old and the mixed implementation (`jax` function evaluations but `numpy` and `scipy` solver routines) demonstrate a linear increase in computational cost.

When considering a variable number of timesteps (Figure 5.2b), the old method initially performs best, reflecting less dispatch overhead in `numpy` and `scipy` per operation. It is very quickly outpaced by the mixed variant, which performs about three times better, yet demonstrates a similar linear growth. Still, for larger numbers of timesteps, they are both completely outclassed by the pure `jax` implementations, which beyond the initial one second of base time expenditure

Silent Stages	new	autodiff	old	mixed
0	2.148	2.354	11.140	6.258
1	3.057	3.274	13.279	6.140
2	4.102	4.305	16.935	6.649
3	6.071	5.598	29.572	7.580
4	1.305	1.275	21.519	8.290
5	1.612	1.527	24.073	8.532
6	1.732	1.646	24.509	9.505
7	1.886	1.793	26.528	10.007
8	2.123	2.045	28.540	10.667
9	2.559	2.451	31.030	11.608
10	2.834	2.754	32.934	12.324

(a) Varied number of silent stages $r = k - s$.

Timesteps	new	autodiff	old	mixed
10^1	1.010	1.485	0.046	0.108
10^2	0.924	1.450	0.357	0.080
10^3	0.953	1.440	4.531	0.802
10^4	1.165	1.708	38.781	8.073
10^5	3.170	3.979	375.418	81.006
10^6	23.307	26.936	3139.148	810.461

(b) Varied number of timesteps.

Batches	new	autodiff	old	mixed
1	5.794	5.991	37.457	12.549
2	8.479	8.176	68.171	17.597
3	4.008	4.136	93.964	28.521
4	4.542	4.171	127.601	38.071
5	4.614	4.349	161.363	47.456
6	4.553	4.192	188.720	55.432
7	5.015	4.798	223.989	65.222
8	4.937	5.094	249.793	73.496

(c) Varied number of batches.

Simulations	new	autodiff	old	mixed
10^0	0.886	1.077	0.037	0.063
10^1	4.909	5.199	0.362	0.116
10^2	3.115	3.423	3.607	1.225
10^3	3.382	3.741	35.547	12.599
10^4	6.257	6.407	352.433	128.610
10^5	41.837	40.828	3971.330	1241.154

(d) Varied number of simulations.

Table 5.1: Timings of different solver configurations applied to the SIHS Hénon–Heiles problem (4.21) with parameters as seen in Table 5.2. The leftmost column represents the value of dependent variable, while the remaining values are timings in seconds averaged over ten runs. The data is visualized in Figure 5.2.

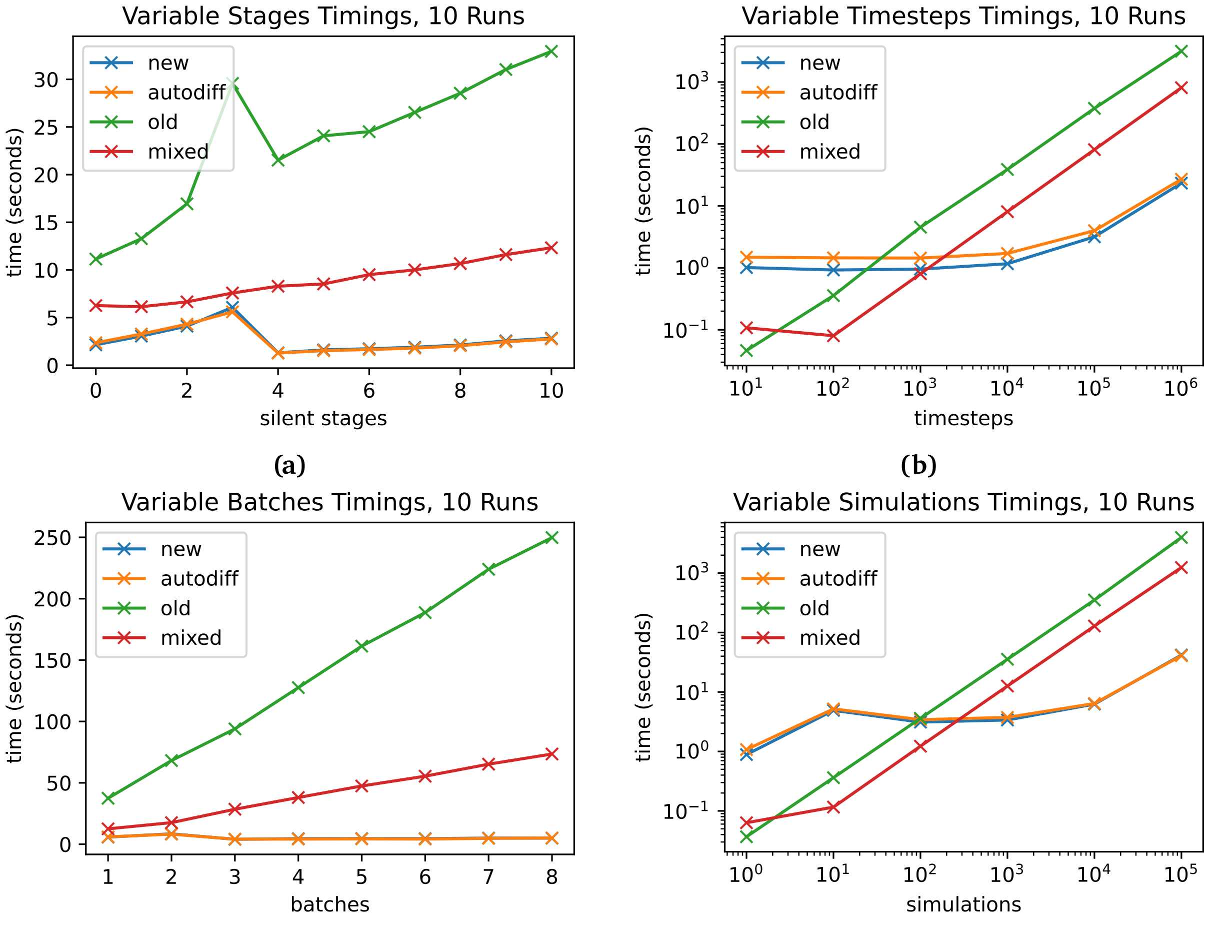
Label	Solver backend	Function evaluations
new	`jax`	`jax`
autodiff	`jax`	`jax` + automatic differentiation
old	`numpy + scipy`	`numpy`
mixed	`numpy + scipy`	`jax`

(a) Features of the different solver configurations.

Figure & Table Number	Method	Timesteps	Batches	Simualtions
5.2a & 5.1a	Gauß-HBVM($-$, 4)	100	1	100
5.2b & 5.1b	Gauß-HBVM(6, 3)	–	1	1
5.2c & 5.1c	Gauß-HBVM(6, 3)	100	–	100
5.2d & 5.1d	Gauß-HBVM(6, 3)	10	1	–

(b) Features of each performance timing series.

Table 5.2: Variables used in the performance timings seen in Figures 5.2 and Tables 5.1, with "–" indicating the independent variable in each subfigure and -table. The solver configurations behind each label is described in Figure 5.2a



(a)

(b)

(c)

(d)

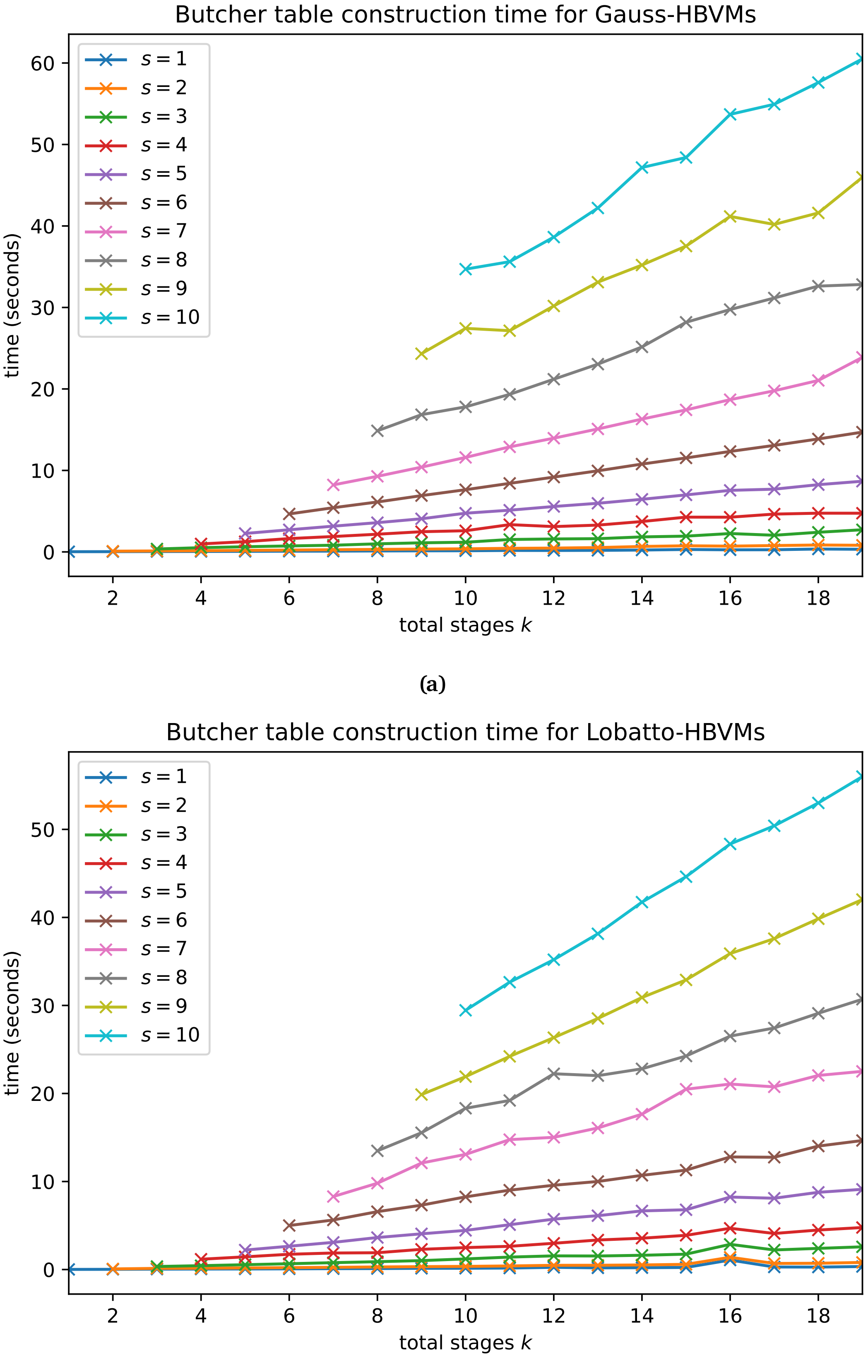
Figure 5.2: Computation time of four solver variants as function of varying problem parameters when solving the SIHS Hénon–Heiles problem (4.21). In Figure 5.2a, the independent variable is the number of of the HBVM; in Figure 5.2b, the number of timesteps are varied; in Figure 5.2d, the number of simulations (for one batch) is varied; and in Figure 5.2c, the number of batches is varied. The stepsize used is $\Delta t = 10^{-1}$. The solver configuration behind each label is summarised remaining parameters for each series are collected in Table 5.2b.

does not need more than two seconds extra for hundred thousand timesteps compared to ten timesteps. In fact, both the new solver (`jax` using formulas) and the autodiff solver (`jax` using automatic differentiation) are more than hundred times faster for a million timesteps than the old solver.

Figure 5.2c shows the performance of the methods using an increasing number of batches. As is expected, the timings of the old and mixed solvers increase close to linearly as a function of batch numbers. For the parallel new and autodiff implementation, however, eight batches (the maximum number of computation cores available) takes no longer time to compute than one. Still, this increases augments the relative speed by a factor of eight.

For a variable number of simulations per batch (Figure 5.2d), a slightly different pattern emerges. The `jax` implementations initially demonstrate a jump in time expenditure (going from one to ten simulation), at which point they stabilize at slightly below ten seconds per run, which performance the keep up until reaching a hundred thousand batch simulations. Again, the speed gain for the largest number of simulations for the new solver compared to the old is close to a factor hundred.

It is unlikely that the performance gain of the new solver over the old observed for simulations and timesteps variations can be combined for much additional speed gain. The computations happen on the same cores, so that when reaching the maximum computational capacity per time unit for the cores, which happens before hundred thousand timesteps and ten thousand simulations, no additional speed gain per core can be had. Even so, combining with the gain from parallelizing, the new solvers are more than eight hundred times faster than the original implementation. Moreover, the `jax` implementation could be expected to be even more advantageous if applied if run on GPUs or TPUs.



(a)

(b)

Figure 5.3: Construction timings for HBVM(k,s) based on Gaußian (Figure 5.3a) and Lobatto (Figure 5.3b) quadrature as a function of total number of stages k, with each line connecting timings of methods with the same number of fundamental stages s. The plotted values are timings averaged over ten runs.

Chapter 6

Discussion

The HBVMs based on Gauß and Lobatto quadrature, being derived from an extended collocation condition, has been shown to observe the theoretical behaviour outlined by Brugnano *et al.* [4, 26] through numerical experiments on determinsitic Hamiltonian systems. HBVMs with s fundamental and k total stages are shown to converge in solution with order $2s$ and in Hamiltonian with order $2k$. The behaviour observed for Lobatto-based methods is qualitatively the same as that of the methods based on Gaußian quadrature when considering methods of the same order; in other words, the Lobatto based HBVMs need an additional fundamental stage for the same convergence results as the Gauß based methods. HBVMs with parameters such that $k \geq \nu s/2$ for Hamiltonians $H \in \mathbb{P}_\nu$ with the energy shown to be preserved accurately in two test problems. Moreover, for general Hamiltonian systems they far outperform the Gauß(-Legendre) and Lobatto IIIA methods of the same order in terms of preserving the Hamiltonian. In addition, it has been observed that the solution error has a weak linear growth for long time intervals, but the Hamiltonian error stays bounded.

After presenting theory regarding convergence of RK methods on a class of SDEs which allows a reformulation with a single integrand, as well as discussing properties of different kinds of SHSes with particular weight on SIHSes, new order results for the HBVMs applied to SIHSes have been presented: A HBVM(k,s), i.e. with s fundamental and k total stages, is expected to have ms order of accuracy s and Hamiltonian error convergence in the weak sense of order k. Incidentally, the result fits more neatly with the method nomenclature than for ODEs problems. Furthermore, the exact energy-preservation condition is extended polynomial stochastic Hamiltonians for canonical SIHSes: For $H \in \mathbb{P}_\nu$, a HBVM(k,s) preserves the energy exactly as long as $\nu s \leq 2k$. Again, the same convergence results are expected for Gauß based HBVMs as for those based on Lobatto quadrature with one additional stage.

The experimental results for SIHS test problems were poorer than their deterministic counterparts. While the ms error convergence almost perfectly matched the theoretical order, the Hamiltonian error convergence was consistently worse than expected, falling slightly with every additional extra silent stage below conservation. Moreover, the likely convergence failure of the Newton iteration subroutine confounded the results, breaking the desired behaviour of the methods. Nevertheless, the performance was several orders better than for standard collocation methods. In addition, the methods were shown to exactly preserve all the polynomial test Hamiltonians with the required number of silent stages. With enough extra stages, some of the methods did preserve the non-polynomial stochastic Kepler polynomial down to machine precision.

Albeit SHSes might arise in many domains, the range of applications for single integrand problems is smaller. Moreover, the results of Debrabant and Kværnø [5] and Cohen *et al.* [49] are so general that the properties of any deterministic method allowing a B-series formulation can be considered proven for single integrand problems. Furthermore, it has already been shown that EQUIP methods, being an improvement on the HBVMs as they are *both* energy-preserving and symplectic, also have these properties for single integrands [50].[1]

[1] albeit the author only became aware of this with one month remaining on the thesis

Nevertheless, some effort has been put into developing an efficient and reliable solver based on `jax`, which has been shown to be quite flexible in its configurations. It was demonstrated to be far more expedient than comparative solvers using standard scientific `python` packages `numpy` and `scipy`. It is possible that introduction of `jax` in scientific computing might make `python` competitive compared to `C` languages and `julia`, although this must be ascertained by benchmarking solvers implemented in other languages as well.

6.1 Further Work

Continuing the thread from above, there are several improvements possible for the solver. The first and most obvious is to introduce a blended implementation of the HBVMs, which would make the computation cost almost independent of the number of silent stages. Secondly, introducing adaptive stepsize routines would solve the issues of non-convergent Newton iterations, making the solver far more robust. Preferrably, the stepsize adaptation should respect the structure of the phase space. This would involve generating Wiener increments in parallel and on the fly, so to speak, wich is feasible with modern RNGs such as those embedded in `numpy` and `jax`. In addition, other collocation polynomials could be used as basis for the HBVMs.

Another natural continuation of this endeavour is the adapation of other Line Integral Methods discussed in the thesis to single integrand problems; in particular could the Poisson Hamiltonian Boundary Vaule Methods and Enhanced PHBVMs prove useful, as they have the strongest conservative properties. It would not pose much scientific progress for the reasons mentioned above, but the prospect of high order energy and invariant preserving methods for stochastic Poisson systems is intriguing. Finally, finding concrete systems which can be advantageously modelled by SIHS would greatly motivate the application of HBVMs, as well as possibly spur interest and drive development within the field.

In a broader scope, it would be interesting if the HBVMs could be adapted to solve Stratonovich SHSes beyond the single integrand problem. One possible way to do this could be by using the extended collocation condition as a simplifying assumption for stochastic B-series formulation of SRK methods, in a similar manner done for symplectic conditions by Anmarkrud and Kværnø [44]. Other possible approaches could involve splitting and composition, in a manner akin to that of Chen *et al.* [48]. Suffice to say, there is much that can still be done.

Appendix A

Selected HBVM Butcher Tables

$$\begin{array}{c|c} \frac{1}{2} & \frac{1}{2} \\ \hline & 1 \end{array}$$

(a) Gauß-2
/implicit midpoint

$$\begin{array}{c|cc} \frac{1}{2}-\frac{\sqrt{3}}{6} & \frac{1}{4}-\frac{\sqrt{3}}{12} & \frac{1}{4}-\frac{\sqrt{3}}{12} \\ \frac{1}{2}+\frac{\sqrt{3}}{6} & \frac{\sqrt{3}}{12}+\frac{1}{4} & \frac{\sqrt{3}}{12}+\frac{1}{4} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

(b) Gauß-HBVM$(2,1)$

$$\begin{array}{c|ccc} \frac{1}{2}-\frac{\sqrt{15}}{10} & \frac{5}{36}-\frac{\sqrt{15}}{36} & \frac{2}{9}-\frac{2\sqrt{15}}{45} & \frac{5}{36}-\frac{\sqrt{15}}{36} \\ \frac{1}{2} & \frac{5}{36} & \frac{2}{9} & \frac{5}{36} \\ \frac{\sqrt{15}}{10}+\frac{1}{2} & \frac{\sqrt{15}}{36}+\frac{5}{36} & \frac{2\sqrt{15}}{45}+\frac{2}{9} & \frac{\sqrt{15}}{36}+\frac{5}{36} \\ \hline & \frac{5}{18} & \frac{4}{9} & \frac{5}{18} \end{array}$$

(c) Gauß-HBVM$(3,1)$

$$\begin{array}{c|cc} \frac{1}{2}-\frac{\sqrt{3}}{6} & \frac{1}{4} & \frac{1}{4}-\frac{\sqrt{3}}{6} \\ \frac{1}{2}+\frac{\sqrt{3}}{6} & \frac{1}{4}+\frac{\sqrt{3}}{6} & \frac{1}{4} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

(d) Gauß-4 or HBVM$(2,2)$

$$\begin{array}{c|ccc} \frac{1}{2}-\frac{\sqrt{15}}{10} & \frac{5}{36}-\frac{\sqrt{15}}{90} & \frac{2}{9}-\frac{2\sqrt{15}}{45} & \frac{5}{36}-\frac{2\sqrt{15}}{45} \\ \frac{1}{2} & \frac{5}{36}+\frac{\sqrt{15}}{24} & \frac{2}{9} & \frac{5}{36}-\frac{\sqrt{15}}{24} \\ \frac{\sqrt{15}}{10}+\frac{1}{2} & \frac{5}{36}+\frac{2\sqrt{15}}{45} & \frac{2\sqrt{15}}{45}+\frac{2}{9} & \frac{\sqrt{15}}{90}+\frac{5}{36} \\ \hline & \frac{5}{18} & \frac{4}{9} & \frac{5}{18} \end{array}$$

(e) Gauß-HBVM$(3,2)$

$$\begin{array}{c|cccc} 0.0694318442029737 & 0.0411075101645298 & 0.0441278761105956 & 0.00115176467954487 & -0.0169553067516966 \\ 0.330009478207572 & 0.156745048786261 & 0.181140552699928 & 0.0340735295718567 & -0.0419496528504736 \\ 0.669990521792428 & 0.215877075419201 & 0.291999047859416 & 0.144932024731345 & 0.0171823737824664 \\ 0.930568155797026 & 0.190882729320424 & 0.324920812751728 & 0.281944701320677 & 0.132819912404197 \\ \hline & 0.173927422568727 & 0.326072577431273 & 0.326072577431273 & 0.173927422568727 \end{array}$$

(f) Gauß-HBVM$(4,2)$

$$\begin{array}{c|ccc} \frac{1}{2}-\frac{\sqrt{15}}{10} & \frac{5}{36} & \frac{2}{9}-\frac{\sqrt{15}}{15} & \frac{5}{36}-\frac{\sqrt{15}}{30} \\ \frac{1}{2} & \frac{5}{36}+\frac{\sqrt{15}}{24} & \frac{2}{9} & \frac{5}{36}-\frac{\sqrt{15}}{24} \\ \frac{\sqrt{15}}{10}+\frac{1}{2} & \frac{\sqrt{15}}{30}+\frac{5}{36} & \frac{2}{9}+\frac{\sqrt{15}}{15} & \frac{5}{36} \\ \hline & \frac{5}{18} & \frac{4}{9} & \frac{5}{18} \end{array}$$

(g) Gauß-HBVM$(3,3)$

$$\begin{array}{c|cccc} 0.0694318442029737 & 0.0707356892573971 & 0.0144996970177283 & -0.0284764144133224 & 0.0126728723411706 \\ 0.330009478207572 & 0.19677415986557 & 0.141111441620619 & -0.00595558150745303 & -0.00192054177116397 \\ 0.669990521792428 & 0.175847964339891 & 0.332028158938726 & 0.184961135810654 & -0.0228467372968433 \\ 0.930568155797026 & 0.161254550227556 & 0.354548991844596 & 0.311572880413545 & 0.10319173331133 \\ \hline & 0.173927422568727 & 0.326072577431273 & 0.326072577431273 & 0.173927422568727 \end{array}$$

(h) Gauß-HBVM$(4,3)$

$$\begin{array}{c|ccccc} 0.046910077030668 & 0.0375158329969564 & 0.0253556946565759 & -0.0154672616370724 & -0.00921283792456479 & 0.00871864893877282 \\ 0.230765344947158 & 0.125933672355799 & 0.116407082408408 & -0.00233173575421991 & -0.0208422850108475 & 0.0115986109480191 \\ 0.5 & 0.139743609338652 & 0.216304736375064 & 0.142222222222222 & 0.0230095988746188 & -0.0212801668105571 \\ 0.769234655052841 & 0.106864831580075 & 0.260156620260531 & 0.286776180198664 & 0.122907252841275 & -0.00747022982770431 \\ 0.953089922969332 & 0.109744793589322 & 0.248527173174248 & 0.299911706081517 & 0.213958640593107 & 0.0809476095311381 \\ \hline & 0.118463442528095 & 0.239314335249683 & 0.284444444444444 & 0.239314335249683 & 0.118463442528095 \end{array}$$

(i) Gauß-HBVM$(5,3)$

Table A.1: Butcher tables for HBVMs based on Gaußian quadrature.

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & \frac{1}{2} & \frac{1}{2} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

(a) Lobatto-2 /trapezoidal rule

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{12} & \frac{1}{3} & \frac{1}{12} \\ 1 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ \hline & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array}$$

(b) Lobatto-HBVM$(2,1)$

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2}-\frac{\sqrt{5}}{10} & \frac{1}{24}-\frac{\sqrt{5}}{120} & \frac{5}{24}-\frac{\sqrt{5}}{24} & \frac{5}{24}-\frac{\sqrt{5}}{24} & \frac{1}{24}-\frac{\sqrt{5}}{120} \\ \frac{\sqrt{5}}{10}+\frac{1}{2} & \frac{\sqrt{5}}{120}+\frac{1}{24} & \frac{\sqrt{5}}{24}+\frac{5}{24} & \frac{\sqrt{5}}{24}+\frac{5}{24} & \frac{\sqrt{5}}{120}+\frac{1}{24} \\ 1 & \frac{1}{12} & \frac{5}{12} & \frac{5}{12} & \frac{1}{12} \\ \hline & \frac{1}{12} & \frac{5}{12} & \frac{5}{12} & \frac{1}{12} \end{array}$$

(c) Lobatto-HBVM$(3,1)$

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{5}{24} & \frac{1}{3} & -\frac{1}{24} \\ 1 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ \hline & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array}$$

(d) Lobatto-HBVM$(2,2)$

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2}-\frac{\sqrt{5}}{10} & \frac{11}{120}-\frac{\sqrt{5}}{120} & \frac{\sqrt{5}}{120}+\frac{5}{24} & \frac{5}{24}-\frac{11\sqrt{5}}{120} & -\frac{\sqrt{5}}{120}-\frac{1}{120} \\ \frac{\sqrt{5}}{10}+\frac{1}{2} & \frac{\sqrt{5}}{120}+\frac{11}{120} & \frac{11\sqrt{5}}{120}+\frac{5}{24} & \frac{5}{24}-\frac{\sqrt{5}}{120} & -\frac{1}{120}+\frac{\sqrt{5}}{120} \\ 1 & \frac{1}{12} & \frac{5}{12} & \frac{5}{12} & \frac{1}{12} \\ \hline & \frac{1}{12} & \frac{5}{12} & \frac{5}{12} & \frac{1}{12} \end{array}$$

(e) Lobatto-HBVM$(3,2)$

$$\begin{array}{c|ccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2}-\frac{\sqrt{21}}{14} & \frac{13}{280}-\frac{\sqrt{21}}{280} & \frac{49}{360}-\frac{\sqrt{21}}{360} & \frac{8}{45}-\frac{8\sqrt{21}}{315} & \frac{49}{360}-\frac{13\sqrt{21}}{360} & \frac{1}{280}-\frac{\sqrt{21}}{280} \\ \frac{1}{2} & \frac{1}{16} & \frac{7\sqrt{21}}{240}+\frac{49}{360} & \frac{8}{45} & \frac{49}{360}-\frac{7\sqrt{21}}{240} & -\frac{1}{80} \\ \frac{\sqrt{21}}{14}+\frac{1}{2} & \frac{\sqrt{21}}{280}+\frac{13}{280} & \frac{49}{360}+\frac{13\sqrt{21}}{360} & \frac{8\sqrt{21}}{315}+\frac{8}{45} & \frac{\sqrt{21}}{360}+\frac{49}{360} & \frac{1}{280}+\frac{\sqrt{21}}{280} \\ 1 & \frac{1}{20} & \frac{49}{180} & \frac{16}{45} & \frac{49}{180} & \frac{1}{20} \\ \hline & \frac{1}{20} & \frac{49}{180} & \frac{16}{45} & \frac{49}{180} & \frac{1}{20} \end{array}$$

(f) Lobatto-HBVM$(4,2)$

$$\begin{array}{c|cccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.117472338035268 & 0.0142830033843551 & 0.0672584313351517 & 0.0572015725440962 & 0.0079789382847165 & -0.022798093331048 & -0.00645151418200396 \\ 0.357384241759677 & 0.0348788826088097 & 0.167379370452486 & 0.153669051207637 & 0.0446285891531954 & -0.0321183851709531 & -0.0110532664914979 \\ 0.642615758240323 & 0.0443865998248312 & 0.221355863319876 & 0.232800599364548 & 0.123760137310106 & 0.0218581076964372 & -0.00154554927547637 \\ 0.882527661964732 & 0.0397848475153373 & 0.212035571479971 & 0.269450250233027 & 0.220227615973647 & 0.121979046813771 & 0.0190503299489782 \\ 1 & 0.0333333333333333 & 0.189237478148923 & 0.277429188517743 & 0.277429188517743 & 0.189237478148923 & 0.0333333333333333 \\ \hline & 0.0333333333333333 & 0.189237478148923 & 0.277429188517743 & 0.277429188517743 & 0.189237478148923 & 0.0333333333333333 \end{array}$$

(g) Lobatto-HBVM$(5,2)$

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2}-\frac{\sqrt{5}}{10} & \frac{\sqrt{5}}{120}+\frac{11}{120} & \frac{5}{24}-\frac{\sqrt{5}}{120} & \frac{5}{24}-\frac{13\sqrt{5}}{120} & -\frac{1}{120}+\frac{\sqrt{5}}{120} \\ \frac{\sqrt{5}}{10}+\frac{1}{2} & \frac{11}{120}-\frac{\sqrt{5}}{120} & \frac{5}{24}+\frac{13\sqrt{5}}{120} & \frac{\sqrt{5}}{120}+\frac{5}{24} & -\frac{\sqrt{5}}{120}-\frac{1}{120} \\ 1 & \frac{1}{12} & \frac{5}{12} & \frac{5}{12} & \frac{1}{12} \\ \hline & \frac{1}{12} & \frac{5}{12} & \frac{5}{12} & \frac{1}{12} \end{array}$$

(h) Lobatto-HBVM$(3,3)$

$$\begin{array}{c|ccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2}-\frac{\sqrt{21}}{14} & \frac{3\sqrt{21}}{1960}+\frac{13}{280} & \frac{\sqrt{21}}{840}+\frac{49}{360} & \frac{8}{45}-\frac{32\sqrt{21}}{735} & \frac{49}{360}-\frac{9\sqrt{21}}{280} & \frac{1}{280}+\frac{3\sqrt{21}}{1960} \\ \frac{1}{2} & \frac{1}{16} & \frac{7\sqrt{21}}{240}+\frac{49}{360} & \frac{8}{45} & \frac{49}{360}-\frac{7\sqrt{21}}{240} & -\frac{1}{80} \\ \frac{\sqrt{21}}{14}+\frac{1}{2} & \frac{13}{280}-\frac{3\sqrt{21}}{1960} & \frac{49}{360}+\frac{9\sqrt{21}}{280} & \frac{8}{45}+\frac{32\sqrt{21}}{735} & \frac{49}{360}-\frac{\sqrt{21}}{840} & \frac{1}{280}-\frac{3\sqrt{21}}{1960} \\ 1 & \frac{1}{20} & \frac{49}{180} & \frac{16}{45} & \frac{49}{180} & \frac{1}{20} \\ \hline & \frac{1}{20} & \frac{49}{180} & \frac{16}{45} & \frac{49}{180} & \frac{1}{20} \end{array}$$

(i) Lobatto-HBVM$(4,3)$

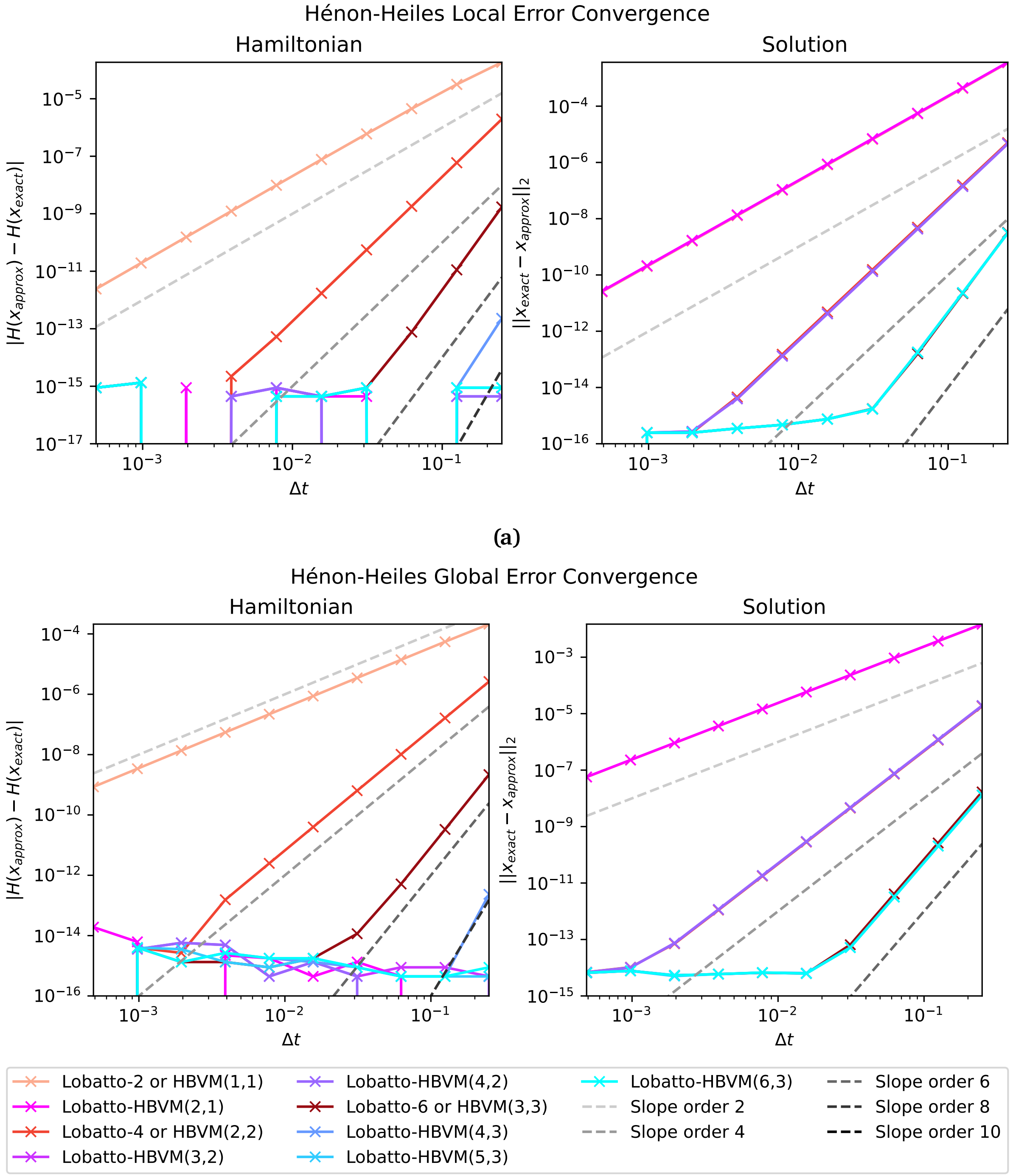
Table A.2: Butcher tables for HBVMs based on Lobatto quadrature.

Appendix B

Additional Convergence Results

Method Name	Hamiltonian		Solution	
	Expected	Observed	Expected	Observed
Lobatto-2 or HBVM(1,1)	2	1.99	2	2.00
Lobatto-HBVM(2,1)	–	–	2	2.00
Lobatto-4 or HBVM(2,2)	4	4.00	4	3.84
Lobatto-HBVM(3,2)	–	–	4	3.85
Lobatto-HBVM(4,2)	–	–	4	3.85
Lobatto-6 or HBVM(3,3)	6	5.84	6	5.99
Lobatto-HBVM(4,3)	8	8.1*	6	5.98
Lobatto-HBVM(5,3)	10.0	–	6	5.98
Lobatto-HBVM(6,3)	–	–	6	5.98

Table B.1: Expected and observed error convergence order in Hamiltonian and solution for deterministic Hénon–Heiles problem with Lobatto-based HBVMs, calculated using same data as seen in Figure B.1b unless otherwise stated. *Approximate value read from figure



(a)

(b)

Figure B.1: Convergence plots for HBVMs with Lobatto quadrature with $s \in \{1,2,3\}$ on Hénon–Heiles problem for $t \in [0,1]$. In addition, slopes of order a are for $a \in \{2,4,6,8\}$ in Figure B.1b and $a \in \{3,4,7,9\}$ in Figure B.1a.

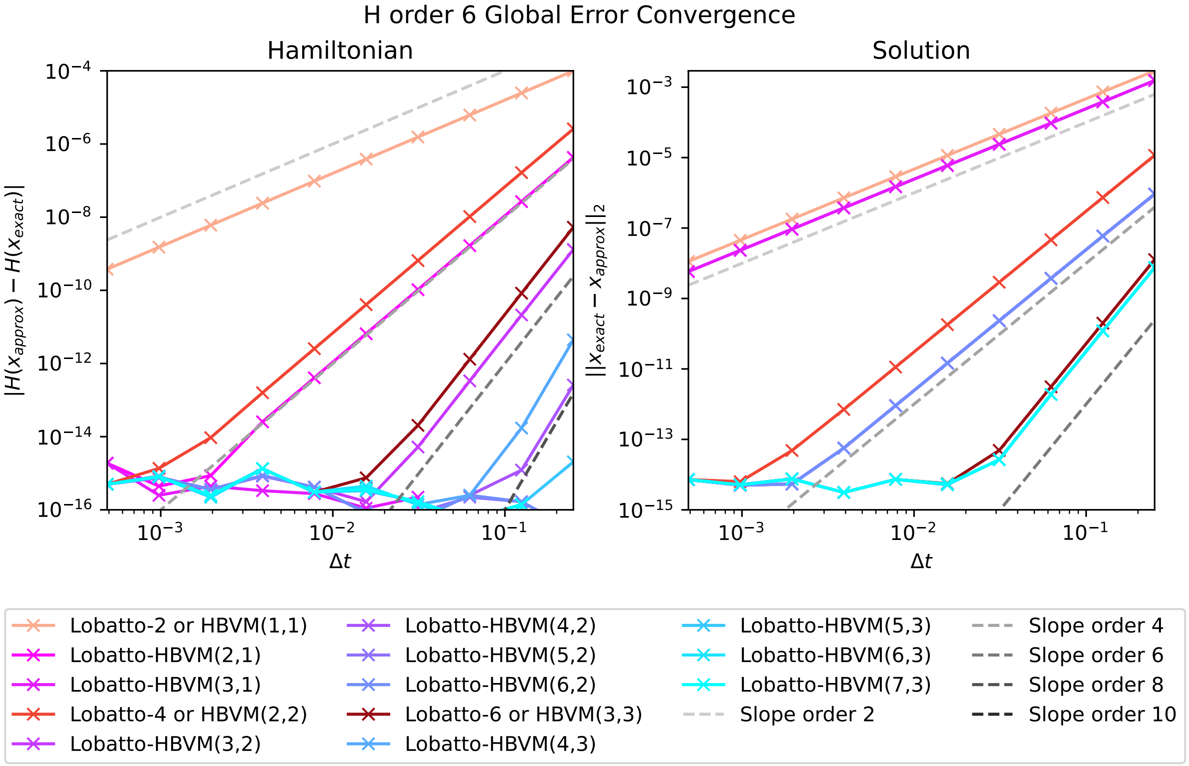


Figure B.2: Convergence of Lobatto-HBVM(k,s) for Hamiltonian of order 6 given by (3.22) for $s \in \{1,2,3\}$ and k up to conservation.

Method Name	Hamiltonian		Solution	
	Expected	Observed	Expected	Observed
Lobatto-2 or HBVM(1,1)	2	2.00	2	2.00
Lobatto-HBVM(2,1)	4	4.00	2	2.00
Lobatto-HBVM(3,1)	–	–	2	2.00
Lobatto-4 or HBVM(2,2)	4	3.99	4	3.98
Lobatto-HBVM(3,2)	6	5.96	4	3.99
Lobatto-HBVM(4,2)	8	–	4	3.99
Lobatto-HBVM(5,2)	10	–	4	3.99
Lobatto-HBVM(6,2)	–	–	4	3.99
Lobatto-6 or HBVM(3,3)	6	6.00	6	6.00
Lobatto-HBVM(4,3)	8	8.01	6	6.03
Lobatto-HBVM(5,3)	10	–	6	6.04
Lobatto-HBVM(6,3)	12	–	6	6.03
Lobatto-HBVM(7,3)	14	–	6	6.04

Table B.2: Table comparing expected and observed error convergence order measured Hamiltonian and Solution for Lobatto-HBVMs; the methods are applied to deterministic canonical Hamiltonian system with $H \in \mathbb{P}_6$ given by (3.22), calculated using same data as seen in Figure B.2.

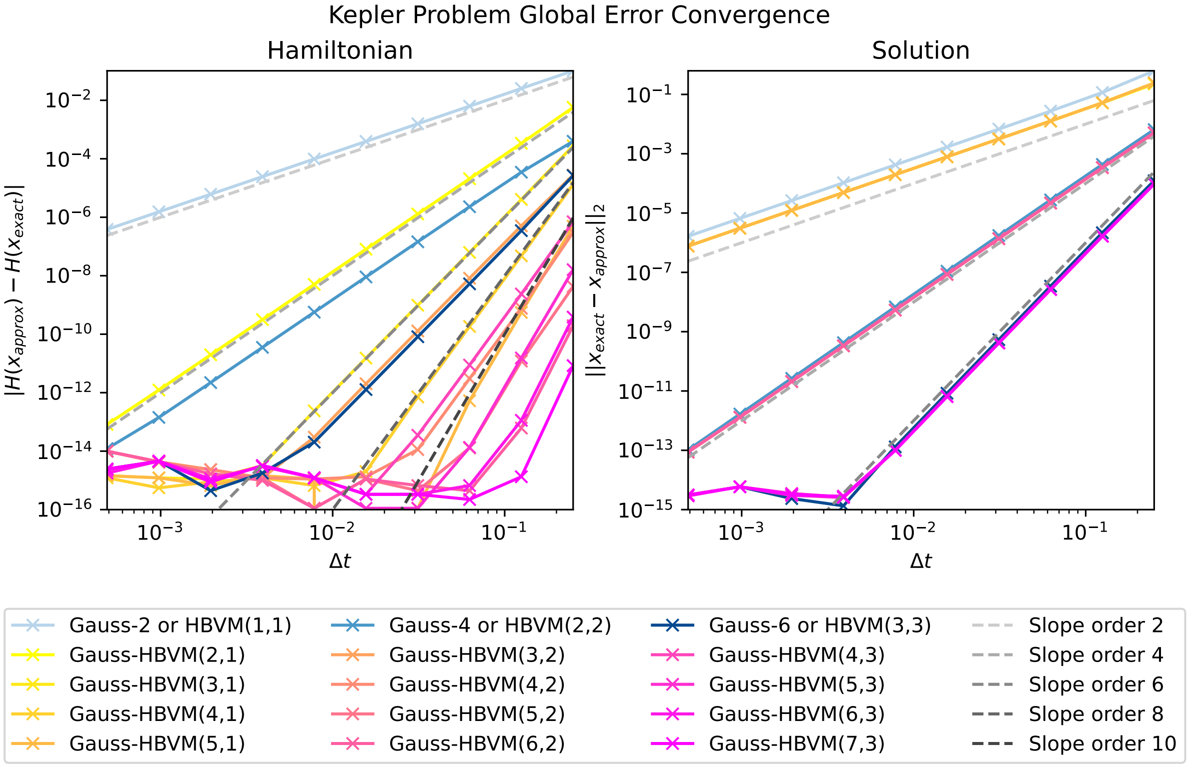


Figure B.3: Deterministic error convergence of Gauß-HBVMs for Kepler problem (3.23) with initial values (3.25) over time $t \in [0, 1]$ compared with slopes of relevant order.

Method Name	Hamiltonian		Solution	
	Expected	Observed	Expected	Observed
Gauss-2 or HBVM(1,1)	2	2.00	2	2.06
Gauss-HBVM(2,1)	4	4.00	2	2.02
Gauss-HBVM(3,1)	6	6.02	2	2.03
Gauss-HBVM(4,1)	8	8.05	2	2.03
Gauss-HBVM(5,1)	10	9.92	2	2.03
Gauss-4 or HBVM(2,2)	4	3.88	4	3.99
Gauss-HBVM(3,2)	6	5.96	4	3.97
Gauss-HBVM(4,2)	8	8.23	4	3.97
Gauss-HBVM(5,2)	10	9.17	4	3.97
Gauss-HBVM(6,2)	12	11.63	4	3.97
Gauss-6 or HBVM(3,3)	6	6.06	6	5.96
Gauss-HBVM(4,3)	8	8.08	6	5.97
Gauss-HBVM(5,3)	10	10.10	6	5.97
Gauss-HBVM(6,3)	12	11.73	6	5.97
Gauss-HBVM(7,3)	14	–	6	5.97

Table B.3: Table of error convergence orders expected and observed for deterministic Kepler problem (3.23) with initial values (3.25). The HBVMs reported are based on Lobatto quadrature and the order is calculated using the same data as seen in figure B.3

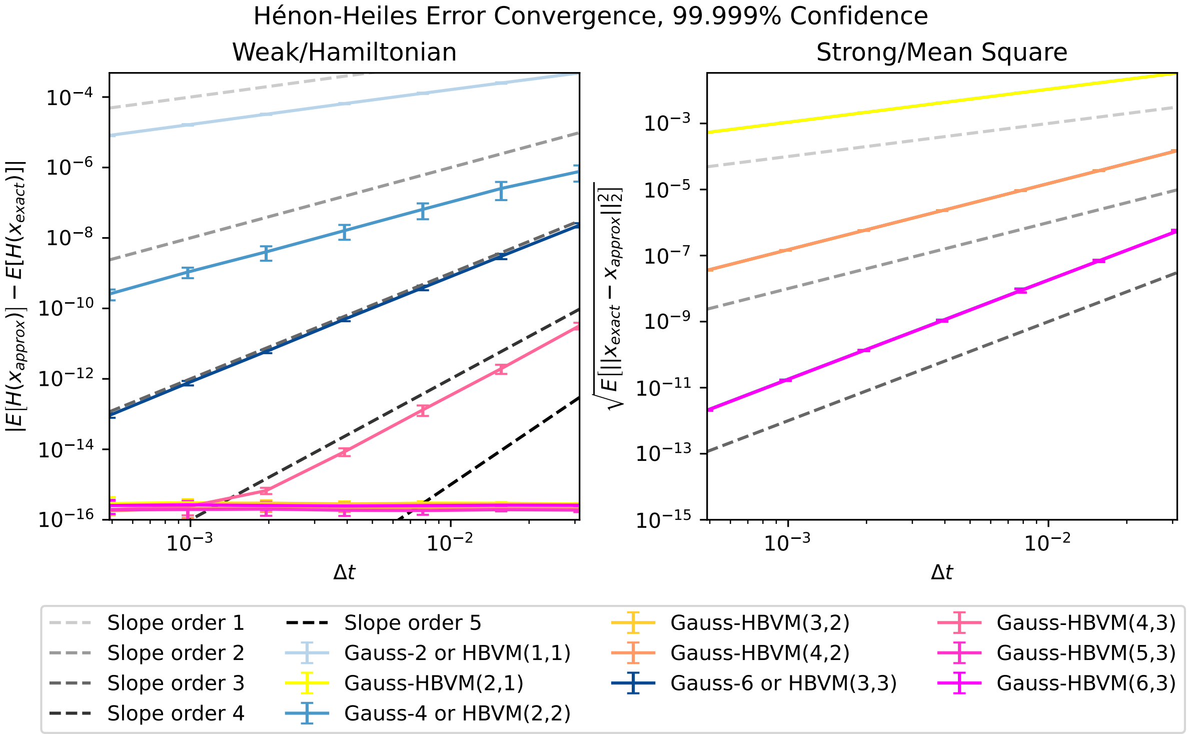


Figure B.4: Convergence plots for single integrand Hénon–Heiles problem (4.21) for Lobatto-HBVM(k,s) for $s \in \{1,2,3\}$ and k, k as described in figure legend and the remaining parameters as described in Table 4.1.

Method Name	Hamiltonian		Solution	
	Expected	Observed	Expected	Observed
Gauss-2 or HBVM(1,1)	1	0.98	1	1.00
Gauss-HBVM(2,1)	–	–	1	1.00
Gauss-4 or HBVM(2,2)	2	1.93	2	2.01
Gauss-HBVM(3,2)	–	–	2	2.00
Gauss-HBVM(4,2)	–	–	2	2.00
Gauss-6 or HBVM(3,3)	3	2.99	3	3.00
Gauss-HBVM(4,3)	4	3.97	3	3.01
Gauss-HBVM(5,3)	–	–	3	3.01
Gauss-HBVM(6,3)	–	–	3	3.01

Table B.4: Expected and observed order of convergence for HBVMs based on Lobatto quadrature applied to the single integrand Hénon–Heiles problem. The observed order is calculated using the same data from which Figure B.4 was made.

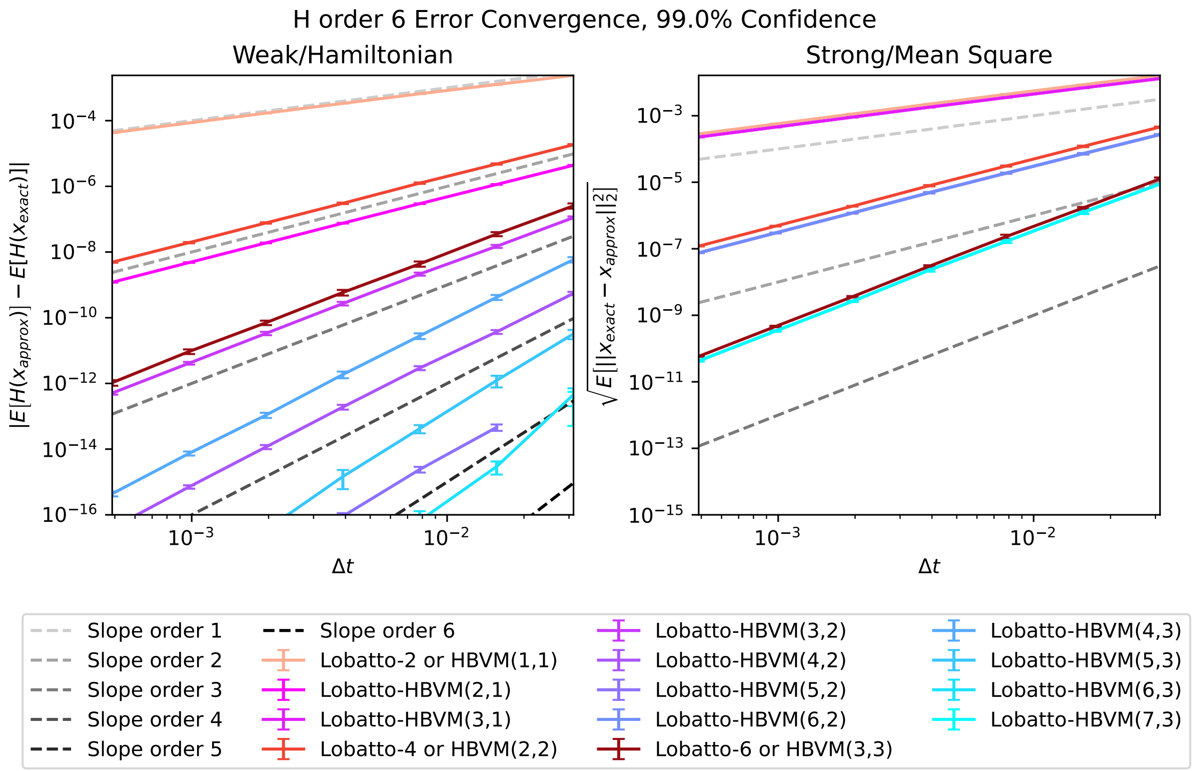


Figure B.5: Weak and strong error convergence plots for SIHS with Hamiltonian and initial values (4.22) simulated with parameters from Table 4.1 using Lobatto-HBVMs with $s \in \{1, 2, 3\}$ and k as described in legend.

Method Name	Hamiltonian		Solution	
	Expected	Observed	Expected	Observed
Lobatto-2 or HBVM(1,1)	1	0.97	1	0.97
Lobatto-HBVM(2,1)	2	1.97	1	0.97
Lobatto-HBVM(3,1)	–	–	1	0.97
Lobatto-4 or HBVM(2,2)	2	1.98	2	1.98
Lobatto-HBVM(3,2)	3	2.96	2	1.97
Lobatto-HBVM(4,2)	4	3.91	2	1.97
Lobatto-HBVM(5,2)	5	4.26	2	1.97
Lobatto-HBVM(6,2)	–	–	2	1.97
Lobatto-6 or HBVM(3,3)	3	2.98	3	2.96
Lobatto-HBVM(4,3)	4	3.95	3	2.95
Lobatto-HBVM(5,3)	5	4.81	3	2.94
Lobatto-HBVM(6,3)	6	7.28	3	2.94
Lobatto-HBVM(7,3)	7	–	3	2.94

Table B.5: Expected and observed weak and strong order of HBVMs based on Lobatto quadrature applied to sixth degree H from (4.22). The observed orders are calculated from the same data as the plots in Figure 4.3.

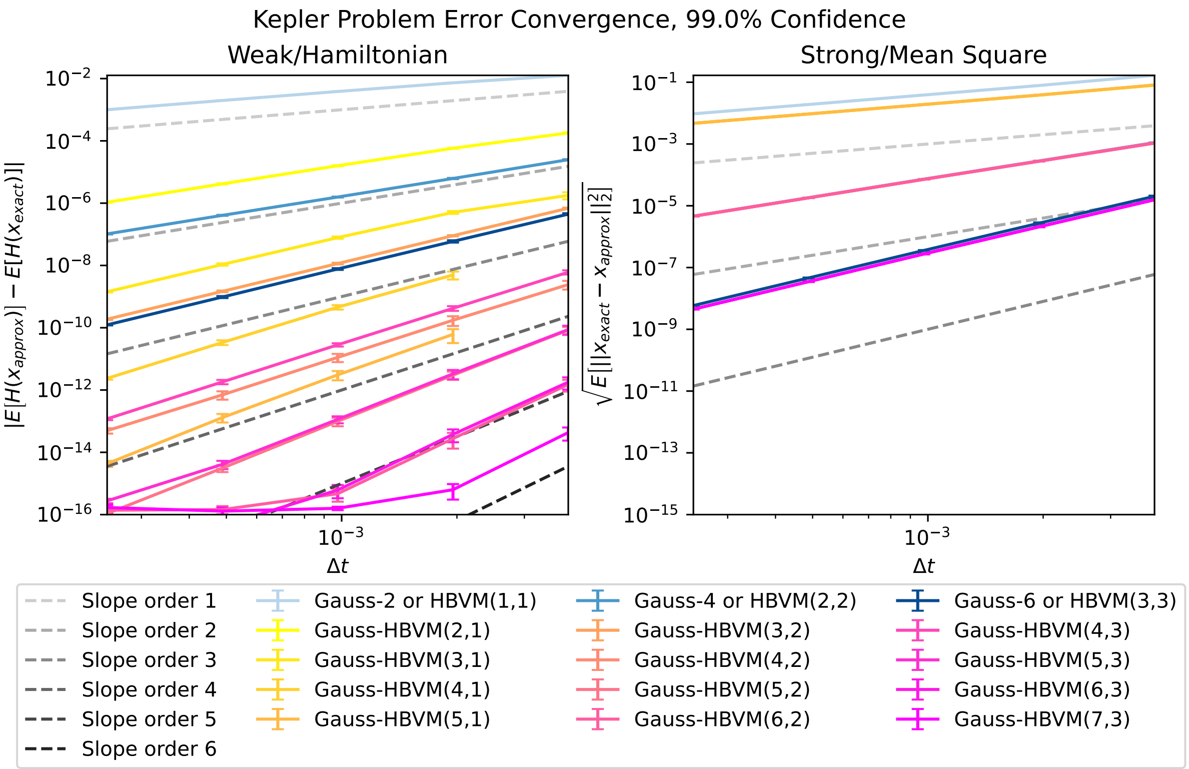


Figure B.6: Error convergence plots for SIHS Kepler (4.23) with initial values (4.24) for Lobatto-HBVMs with $s \in \{1, 2, 3\}$ and k as described in legend; the remaining simulation parameters are described in Table 4.1.

Method Name	Hamiltonian		Solution	
	Expected	Observed	Expected	Observed
Gauss-2 or HBVM(1,1)	1	0.92	1	1.03
Gauss-HBVM(2,1)	2	1.85	1	1.02
Gauss-HBVM(3,1)	3	2.57	1	1.03
Gauss-HBVM(4,1)	4	3.68	1	1.03
Gauss-HBVM(5,1)	5	4.60	1	1.03
Gauss-4 or HBVM(2,2)	2	1.98	2	1.98
Gauss-HBVM(3,2)	3	2.96	2	1.97
Gauss-HBVM(4,2)	4	3.90	2	1.97
Gauss-HBVM(5,2)	5	4.91	2	1.97
Gauss-HBVM(6,2)	6	5.77	2	1.97
Gauss-6 or HBVM(3,3)	3	2.95	3	2.94
Gauss-HBVM(4,3)	4	3.91	3	2.94
Gauss-HBVM(5,3)	5	4.79	3	2.94
Gauss-HBVM(6,3)	6	5.57	3	2.94
Gauss-HBVM(7,3)	7	–	3	2.94

Table B.6: Expected and observed weak (Hamiltonian) and strong (ms) error convergence order of Gauß-HBVMs applied to the SIHS Kepler problem. The same data is used to calculate the observed order as to generate the plots in Figure 4.6.

Abbreviations

A

a.a. almost all. 19, 20
a.e. almost everywhere.
a.s. almost surely. 3, 19, 21
AOT Compiling machine code Ahead OF Time for execution. 51
AVF Averaged Vector Field method. 2

C

CAS Computer Algebra System. 54
cf. short for Latin phrase confer, meaning "compare with". 3, 26, 27, 42, 43, 52
CPU Central Processing Unit. 51, 53

D

DSL Domain Specific Language. 51

E

e.g. exempli gratia (Latin, meaning "for example"). 1, 2, 13, 20, 25, 27
et al. et alums(neutral), alumni(masculine), alumnae(feminine), meaning "and contributors").

G

GPU Graphical Processing Unit. 51, 53, 58

I

i.e. id est, Latin for "that is".. 6–8, 10, 12, 15, 17, 18, 20, 21, 25, 27, 30, 32, 36, 40–43, 46, 53, 60

J

JIT Compiling machine code Just In Time for execution. 51, 55

L

LIM Line Integral Method. 2, 53

- **EQUIP** Energy and QUadratic Invariant Preserving Method. 3, 4, 43, 60
- **HBVM** Hamiltonian Boundary Value Method. viii, ix, 2–4, 25, 28–33, 35–39, 43–50, 52–54, 57, 60–70
- **PHBVM** Poisson Hamiltonain Boundary Value method. 3
    - **EPHBVM** Enhanced PHBVM. 3

LMM Linear Multistep Method.

M

ms mean square. 3, 21, 39, 41, 43, 45–47, 49, 50, 60, 70

N

NPU National Board of Scholarly Publishing. 2
NTNU Norwegian University of Science and Technology. 6

O

ODE Ordinary Differential Equation. 1–4, 7–11, 13, 14, 18, 25, 31, 32, 34–36, 42, 46, 49, 52, 60, 71

P

psd Positive semi-definite; a matrix $M \in \mathbb{R}^d$ is positive semi-definite (non-negative definite) if

$$x^T M x \geq 0 \quad \forall x \in \mathbb{R}^d.$$

. Positive semi-definiteness and non-negative definiteness are equivalent concepts.. 11

R

RK Runge–Kutta, reffering to a family of one-step methods for solving ODEs. 2, 4, 10–12, 14, 16, 17, 25–27, 40, 41, 52, 60

- **ERK** Explicit Runge–Kutta methods; Runge–Kutta methods where alle stages are explicit.

  - **IRK** Implicit Runge–Kutta methods; Runge–Kutta methods where some of the stages are defined implicitly. 12, 52
  - **SRK** Runge–Kutta methods adapted to solve SDEs. 3, 4, 40, 41, 61

- **RNG** Random Number Generator. 51, 53, 61
  - **GSFR** Generalized Feedback Shift Register generator. 53
  - **LCG** Linear Congruential Generator. 53
    - **PCG** Permuted Congruential Generator. 53
  - **PRNG** Parallel Random Number Generation. 53

S

- **SDE** Stochastic Differential Equation. v, 1, 3, 4, 6, 18, 19, 24, 40–42, 51, 60, 72
  - **SHS** Stochastic Hamiltonian System. v, 3, 4, 39, 40, 42, 60, 61
    - **SIHS** Single Integrand Hamiltonian System. viii, ix, 4, 42–44, 46, 48–50, 54, 56, 57, 60, 61, 69, 70

T

TPU Tensor Processing Unit. 51, 53, 58

W

w.r.t. with respect to. x

X

XLA Accelerated Linear Algebra.

Bibliography

[1] S. S. Bergmann, *Conservation of Hamiltonians and Invariants in Stochastic Differential Equations*, Report submitted in TMA4505 Industrial Mathematics, Specialization Project, Jan. 2022.

[2] J. Hong, D. Xu and P. Wang, 'Preservation of quadratic invariants of stochastic differential equations via runge–kutta methods,' English, *Applied Numerical Mathematics*, vol. 87, pp. 38–52, 2015. [Online]. Available: https://doi.org/10.1016/j.apnum.2014.08.003.

[3] P. M. Burrage and K. Burrage, 'Structure-preserving Runge-Kutta methods for stochastic Hamiltonian equations with additive noise,' *Numer. Algorithms*, vol. 65, no. 3, pp. 519–532, 2014, ISSN: 1017-1398. DOI: 10.1007/s11075-013-9796-6. [Online]. Available: https://doi.org/10.1007/s11075-013-9796-6.

[4] L. Brugnano, F. Iavernaro and D. Trigiante, 'Analysis of hamiltonian boundary value methods (hbvms): A class of energy-preserving runge–kutta methods for the numerical solution of polynomial hamiltonian systems,' *Communications in Nonlinear Science and Numerical Simulation*, vol. 20, no. 3, pp. 650–667, Mar. 2015, ISSN: 1007-5704. DOI: 10.1016/j.cnsns.2014.05.030. [Online]. Available: http://dx.doi.org/10.1016/j.cnsns.2014.05.030.

[5] K. Debrabant and A. Kværnø, 'Cheap arbitrary high order methods for single integrand SDEs,' *BIT*, vol. 57, no. 1, pp. 153–168, 2017, ISSN: 0006-3835. DOI: 10.1007/s10543-016-0619-8. [Online]. Available: https://doi.org/10.1007/s10543-016-0619-8.

[6] E. Hairer, C. Lubich and G. Wanner, *Geometric numerical integration* (Springer Series in Computational Mathematics), Second. Springer-Verlag, Berlin, 2006, vol. 31, pp. xviii+644, Structure-preserving algorithms for ordinary differential equations, ISBN: 978-3-540-30663-4.

[7] B. Leimkuhler, *Simulating hamiltonian dynamics*, eng, Cambridge, 2004.

[8] *Hamiltonian dynamics in economics*. Academic Press [Harcourt Brace Jovanovich, Publishers], New York-London, 1976, i and 1–196, J. Econom. Theory **12** (1976), no. 1.

[9] M. Scalia, A. Angelini, F. Farioli, G. F. Mattioli, O. Ragnisco and M. Saviano, 'An ecology and economy coupling model. a global stationary state model for a sustainable economy in the hamiltonian formalism,' English, *Ecological Economics*, vol. 172, 2020. DOI: 10.1016/j.ecolecon.2019.106497.

[10] E. Faou, E. Hairer and T.-L. Pham, 'Energy conservation with non-symplectic methods: Examples and counter-examples,' *BIT*, vol. 44, no. 4, pp. 699–709, 2004, ISSN: 0006-3835. DOI: 10.1007/s10543-004-5240-6. [Online]. Available: https://doi.org/10.1007/s10543-004-5240-6.

[11] J. M. Sanz-Serna and M. P. Calvo, *Numerical Hamiltonian problems* (Applied Mathematics and Mathematical Computation). Chapman & Hall, London, 1994, vol. 7, pp. xii+207, ISBN: 0-412-54290-0.

[12] P. B. Bochev and C. Scovel, 'On quadratic invariants and symplectic structure,' *BIT*, vol. 34, no. 3, pp. 337–345, 1994, ISSN: 0006-3835. DOI: 10.1007/BF01935643. [Online]. Available: https://doi.org/10.1007/BF01935643.

[13] G. Zhong and J. E. Marsden, 'Lie-Poisson Hamilton-Jacobi theory and Lie-Poisson integrators,' *Phys. Lett. A*, vol. 133, no. 3, pp. 134–139, 1988, ISSN: 0375-9601. DOI: 10.1016/0375-9601(88)90773-6. [Online]. Available: https://doi.org/10.1016/0375-9601(88)90773-6.

[14] P. Chartier, E. Faou and A. Murua, 'An algebraic approach to invariant preserving integrators: The case of quadratic and Hamiltonian invariants,' *Numer. Math.*, vol. 103, no. 4, pp. 575–590, 2006, ISSN: 0029-599X. DOI: 10.1007/s00211-006-0003-8. [Online]. Available: https://doi.org/10.1007/s00211-006-0003-8.

[15] R. I. McLachlan, G. R. W. Quispel and N. Robidoux, 'Geometric integration using discrete gradients,' *R. Soc. Lond. Philos. Trans. Ser. A Math. Phys. Eng. Sci.*, vol. 357, no. 1754, pp. 1021–1045, 1999, ISSN: 1364-503X. DOI: 10.1098/rsta.1999.0363. [Online]. Available: https://doi.org/10.1098/rsta.1999.0363.

[16] G. R. W. Quispel and D. I. McLaren, 'A new class of energy-preserving numerical integration methods,' *J. Phys. A*, vol. 41, no. 4, pp. 045206, 7, 2008, ISSN: 1751-8113. DOI: 10.1088/1751-8113/41/4/045206. [Online]. Available: https://doi.org/10.1088/1751-8113/41/4/045206.

[17] L. Brugnano and F. Iavernaro, 'Line integral methods which preserve all invariants of conservative problems,' *J. Comput. Appl. Math.*, vol. 236, no. 16, pp. 3905–3919, 2012, ISSN: 0377-0427. DOI: 10.1016/j.cam.2012.03.026. [Online]. Available: https://doi.org/10.1016/j.cam.2012.03.026.

[18] E. Celledoni, R. I. McLachlan, D. I. McLaren, B. Owren, G. R. W. Quispel and W. M. Wright, 'Energy-preserving Runge-Kutta methods,' *M2AN Math. Model. Numer. Anal.*, vol. 43, no. 4, pp. 645–649, 2009, ISSN: 0764-583X. DOI: 10.1051/m2an/2009020. [Online]. Available: https://doi.org/10.1051/m2an/2009020.

[19] E. Celledoni, B. Owren and Y. Sun, 'The minimal stage, energy preserving Runge-Kutta method for polynomial Hamiltonian systems is the averaged vector field method,' *Math. Comp.*, vol. 83, no. 288, pp. 1689–1700, 2014, ISSN: 0025-5718. DOI: 10.1090/S0025-5718-2014-02805-6. [Online]. Available: https://doi.org/10.1090/S0025-5718-2014-02805-6.

[20] F. Iavernaro and D. Trigiante, 'High-order symmetric schemes for the energy conservation of polynomial Hamiltonian problems,' *JNAIAM J. Numer. Anal. Ind. Appl. Math.*, vol. 4, no. 1-2, pp. 87–101, 2009, ISSN: 1790-8140.

[21] E. Hairer, 'Energy-preserving variant of collocation methods,' *JNAIAM. J. Numer. Anal. Ind. Appl. Math.*, vol. 5, no. 1-2, pp. 73–84, 2010, ISSN: 1790-8140.

[22] L. Brugnano, 'Boundary value methods for the numerical approximation of ordinary differential equations,' in *Numerical analysis and its applications (Rousse, 1996)*, ser. Lecture Notes in Comput. Sci. Vol. 1196, Springer, Berlin, 1997, pp. 78–89. DOI: 10.1007/3-540-62598-4_81. [Online]. Available: https://doi.org/10.1007/3-540-62598-4_81.

[23] L. Brugnano and D. Trigiante, 'Boundary value methods: The third way between linear multistep and Runge-Kutta methods,' in 10-12, vol. 36, Advances in difference equations, II, 1998, pp. 269–284. DOI: 10.1016/S0898-1221(98)80028-X. [Online]. Available: https://doi.org/10.1016/S0898-1221(98)80028-X.

[24] L. Brugnano, F. Iavernaro and D. Trigiante, 'Hamiltonian boundary value methods (energy preserving discrete line integral methods),' *JNAIAM. J. Numer. Anal. Ind. Appl. Math.*, vol. 5, no. 1-2, pp. 17–37, 2010, ISSN: 1790-8140.

[25] L. Brugnano, G. Frasca Caccia and F. Iavernaro, 'Efficient implementation of gauss collocation and hamiltonian boundary value methods,' *Numerical Algorithms*, vol. 65, no. 3, pp. 633–650, Jan. 2014, ISSN: 1572-9265. DOI: 10.1007/s11075-014-9825-0. [Online]. Available: http://dx.doi.org/10.1007/s11075-014-9825-0.

[26] L. Brugnano, F. Iavernaro and D. Trigiante, 'A note on the efficient implementation of hamiltonian bvms,' *Journal of Computational and Applied Mathematics*, vol. 236, no. 3, pp. 375–383, Sep. 2011, ISSN: 0377-0427. DOI: 10.1016/j.cam.2011.07.022. [Online]. Available: http://dx.doi.org/10.1016/j.cam.2011.07.022.

[27] L. Brugnano, F. Iavernaro and D. Trigiante, 'A simple framework for the derivation and analysis of effective one-step methods for ODEs,' *Appl. Math. Comput.*, vol. 218, no. 17, pp. 8475–8485, 2012, ISSN: 0096-3003. DOI: 10.1016/j.amc.2012.01.074. [Online]. Available: https://doi.org/10.1016/j.amc.2012.01.074.

[28] P. Amodio, L. Brugnano and F. Iavernaro, 'Energy-conserving methods for Hamiltonian boundary value problems and applications in astrodynamics,' *Adv. Comput. Math.*, vol. 41, no. 4, pp. 881–905, 2015, ISSN: 1019-7168. DOI: 10.1007/s10444-014-9390-z. [Online]. Available: https://doi.org/10.1007/s10444-014-9390-z.

[29] P. Amodio, L. Brugnano and F. Iavernaro, 'Analysis of spectral Hamiltonian boundary value methods (SHBVMs) for the numerical solution of ODE problems,' *Numer. Algorithms*, vol. 83, no. 4, pp. 1489–1508, 2020, ISSN: 1017-1398. DOI: 10.1007/s11075-019-00733-7. [Online]. Available: https://doi.org/10.1007/s11075-019-00733-7.

[30] P. Amodio, L. Brugnano and F. Iavernaro, 'A note on the continuous-stage Runge-Kutta(-Nyström) formulation of Hamiltonian boundary value methods (HBVMs),' *Appl. Math. Comput.*, vol. 363, pp. 124634, 8, 2019, ISSN: 0096-3003. DOI: 10.1016/j.amc.2019.124634. [Online]. Available: https://doi.org/10.1016/j.amc.2019.124634.

[31] L. Brugnano and F. Iavernaro, 'Line integral solution of differential problems,' *Axioms*, vol. 7, no. 2, 2018, ISSN: 2075-1680. DOI: 10.3390/axioms7020036. [Online]. Available: https://www.mdpi.com/2075-1680/7/2/36.

[32] N. D. for Higher Education and Skills, *Axioms*, Retrieved June 10, 2022, 2022. [Online]. Available: https://kanalregister.hkdir.no/publiseringskanaler/KanalTidsskriftInfo.action?id=494411.

[33] L. Brugnano, M. Calvo, J. I. Montijano and L. Rández, 'Energy-preserving methods for Poisson systems,' *J. Comput. Appl. Math.*, vol. 236, no. 16, pp. 3890–3904, 2012, ISSN: 0377-0427. DOI: 10.1016/j.cam.2012.02.033. [Online]. Available: https://doi.org/10.1016/j.cam.2012.02.033.

[34] K. Burrage and P. M. Burrage, 'Low rank Runge-Kutta methods, symplecticity and stochastic Hamiltonian problems with additive noise,' *J. Comput. Appl. Math.*, vol. 236, no. 16, pp. 3920–3930, 2012, ISSN: 0377-0427. DOI: 10.1016/j.cam.2012.03.007. [Online]. Available: https://doi.org/10.1016/j.cam.2012.03.007.

[35] L. Brugnano, G. Gurioli and F. Iavernaro, 'Analysis of energy and quadratic invariant preserving (EQUIP) methods,' *J. Comput. Appl. Math.*, vol. 335, pp. 51–73, 2018, ISSN: 0377-0427. DOI: 10.1016/j.cam.2017.11.043. [Online]. Available: https://doi.org/10.1016/j.cam.2017.11.043.

[36] P. Amodio, L. Brugnano and F. Iavernaro, 'Arbitrary high-order energy-conserving methods for poisson problems,' *Numerical Algorithms*, Mar. 2022. [Online]. Available: https://doi.org/10.1007/s11075-022-01285-z.

[37] P. E. Kloeden and E. Platen, *Numerical solution of stochastic differential equations* (Applications of Mathematics (New York)). Springer-Verlag, Berlin, 1992, vol. 23, pp. xxxvi+632, ISBN: 3-540-54062-8. DOI: 10.1007/978-3-662-12616-5. [Online]. Available: https://doi.org/10.1007/978-3-662-12616-5.

[38] D. J. Higham, 'An algorithmic introduction to numerical simulation of stochastic differential equations,' *SIAM Rev.*, vol. 43, no. 3, pp. 525–546, 2001, ISSN: 0036-1445. DOI: 10.1137/S0036144500378302. [Online]. Available: https://doi.org/10.1137/S0036144500378302.

[39] G. N. Milstein and M. V. Tretyakov, *Stochastic numerics for mathematical physics* (Scientific Computation), Second. Springer-Verlag, Berlin, 2021, pp. xxv+736, ISBN: 3-540-21110-1. DOI: `10.1007/978-3-030-82040-4`. [Online]. Available: `https://doi.org/10.1007/978-3-030-82040-4`.

[40] K. Burrage and P. M. Burrage, 'High strong order explicit Runge-Kutta methods for stochastic ordinary differential equations,' in 1-3, vol. 22, Special issue celebrating the centenary of Runge–Kutta methods, 1996, pp. 81–101. DOI: `10.1016/S0168-9274(96)00027-X`. [Online]. Available: `https://doi.org/10.1016/S0168-9274(96)00027-X`.

[41] K. Burrage and P. M. Burrage, 'Order conditions of stochastic Runge-Kutta methods by B-series,' *SIAM J. Numer. Anal.*, vol. 38, no. 5, pp. 1626–1646, 2000, ISSN: 0036-1429. DOI: `10.1137/S0036142999363206`. [Online]. Available: `https://doi.org/10.1137/S0036142999363206`.

[42] Y. Komori, 'Multi-colored rooted tree analysis of the weak order conditions of a stochastic Runge-Kutta family,' *Appl. Numer. Math.*, vol. 57, no. 2, pp. 147–165, 2007, ISSN: 0168-9274. DOI: `10.1016/j.apnum.2006.02.002`. [Online]. Available: `https://doi.org/10.1016/j.apnum.2006.02.002`.

[43] A. Rößler, 'Rooted tree analysis for order conditions of stochastic Runge-Kutta methods for the weak approximation of stochastic differential equations,' *Stoch. Anal. Appl.*, vol. 24, no. 1, pp. 97–134, 2006, ISSN: 0736-2994. DOI: `10.1080/07362990500397699`. [Online]. Available: `https://doi.org/10.1080/07362990500397699`.

[44] S. Anmarkrud and A. Kværnø, 'Order conditions for stochastic Runge-Kutta methods preserving quadratic invariants of Stratonovich SDEs,' *J. Comput. Appl. Math.*, vol. 316, pp. 40–46, 2017, ISSN: 0377-0427. DOI: `10.1016/j.cam.2016.08.042`. [Online]. Available: `https://doi.org/10.1016/j.cam.2016.08.042`.

[45] J. Hong, S. Zhai and J. Zhang, 'Discrete gradient approach to stochastic differential equations with a conserved quantity,' *SIAM J. Numer. Anal.*, vol. 49, no. 5, pp. 2017–2038, 2011, ISSN: 0036-1429. DOI: `10.1137/090771880`. [Online]. Available: `https://doi.org/10.1137/090771880`.

[46] X. Li, C. Zhang, Q. Ma and X. Ding, 'Discrete gradient methods and linear projection methods for preserving a conserved quantity of stochastic differential equations,' *Int. J. Comput. Math.*, vol. 95, no. 12, pp. 2511–2524, 2018, ISSN: 0020-7160. DOI: `10.1080/00207160.2017.1408803`. [Online]. Available: `https://doi.org/10.1080/00207160.2017.1408803`.

[47] R. D'Ambrosio, G. Giordano, B. Paternoster and A. Ventola, 'Perturbative analysis of stochastic Hamiltonian problems under time discretizations,' *Appl. Math. Lett.*, vol. 120, Paper No. 107223, 7, 2021, ISSN: 0893-9659. DOI: `10.1016/j.aml.2021.107223`. [Online]. Available: `https://doi.org/10.1016/j.aml.2021.107223`.

[48] C. Chen, D. Cohen, R. D'Ambrosio and A. Lang, 'Drift-preserving numerical integrators for stochastic Hamiltonian systems,' *Adv. Comput. Math.*, vol. 46, no. 2, Paper No. 27, 22, 2020, ISSN: 1019-7168. DOI: `10.1007/s10444-020-09771-5`. [Online]. Available: `https://doi.org/10.1007/s10444-020-09771-5`.

[49] D. Cohen, K. Debrabant and A. Rößler, 'High order numerical integrators for single integrand Stratonovich SDEs,' *Appl. Numer. Math.*, vol. 158, pp. 264–270, 2020, ISSN: 0168-9274. DOI: `10.1016/j.apnum.2020.08.002`. [Online]. Available: `https://doi.org/10.1016/j.apnum.2020.08.002`.

[50] X. Li, C. Zhang, Q. Ma and X. Ding, 'Arbitrary high-order EQUIP methods for stochastic canonical Hamiltonian systems,' *Taiwanese J. Math.*, vol. 23, no. 3, pp. 703–725, 2019, ISSN: 1027-5487. DOI: `10.11650/tjm/180803`. [Online]. Available: `https://doi.org/10.11650/tjm/180803`.

[51] E. Hairer and G. Wanner, *Solving ordinary differential equations. II* (Springer Series in Computational Mathematics), Second. Springer-Verlag, Berlin, 1996, vol. 14, pp. xvi+614, Stiff and differential-algebraic problems, ISBN: 3-540-60452-9. DOI: 10.1007/978-3-642-05221-7. [Online]. Available: https://doi.org/10.1007/978-3-642-05221-7.

[52] E. Hairer, S. P. Nørsett and G. Wanner, *Solving ordinary differential equations. I* (Springer Series in Computational Mathematics), Second. Springer-Verlag, Berlin, 1993, vol. 8, pp. xvi+528, Nonstiff problems, ISBN: 3-540-56670-8.

[53] A. Quarteroni, R. Sacco and F. Saleri, *Numerical mathematics* (Texts in Applied Mathematics), Second. Springer-Verlag, Berlin, 2007, vol. 37, pp. xviii+655, ISBN: 978-3-540-34658-6; 3-540-34658-9. DOI: 10.1007/b98885. [Online]. Available: https://doi.org/10.1007/b98885.

[54] L. C. Evans, *Partial differential equations* (Graduate Studies in Mathematics), Second. American Mathematical Society, Providence, RI, 2010, vol. 19, pp. xxii+749, ISBN: 978-0-8218-4974-3. DOI: 10.1090/gsm/019. [Online]. Available: https://doi.org/10.1090/gsm/019.

[55] K. Itô, 'Stochastic integral,' *Proc. Imp. Acad. Tokyo*, vol. 20, pp. 519–524, 1944, ISSN: 0369-9846. [Online]. Available: http://projecteuclid.org/euclid.pja/1195572786.

[56] R. L. Stratonovich, 'A new representation for stochastic integrals and equations,' *SIAM J. Control*, vol. 4, pp. 362–371, 1966, ISSN: 0363-0129.

[57] B. Øksendal, *Stochastic differential equations* (Universitext), Sixth. Springer-Verlag, Berlin, 2003, pp. xxiv+360, An introduction with applications, ISBN: 3-540-04758-1. DOI: 10.1007/978-3-642-14394-6. [Online]. Available: https://doi.org/10.1007/978-3-642-14394-6.

[58] H.-H. Kuo, *Introduction to stochastic integration* (Universitext). Springer, New York, 2006, pp. xiv+278, ISBN: 978-0387-28720-1; 0-387-28720-5.

[59] D. Cohen and G. Dujardin, 'Energy-preserving integrators for stochastic Poisson systems,' *Commun. Math. Sci.*, vol. 12, no. 8, pp. 1523–1539, 2014, ISSN: 1539-6746. DOI: 10.4310/CMS.2014.v12.n8.a7. [Online]. Available: https://doi.org/10.4310/CMS.2014.v12.n8.a7.

[60] TIOBE Software. 'TIOBE Index for June 2022.' Accessed June 18, 2022, TIOBE Software BV. (2022), [Online]. Available: https://www.tiobe.com/tiobe-index/.

[61] P. Carbonnelle. 'PYPL PopularitY of Programming Language.' Accessed June 18, 2022. (2022), [Online]. Available: https://pypl.github.io/PYPL.html.

[62] S. E. Inc. 'Stack overflow 2021 developer survey.' Accessed June 18, 2022. (2022), [Online]. Available: https://insights.stackoverflow.com/survey/2021.

[63] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke and T. E. Oliphant, 'Array programming with NumPy,' *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. DOI: 10.1038/s41586-020-2649-2. [Online]. Available: https://doi.org/10.1038/s41586-020-2649-2.

[64] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt and SciPy 1.0 Contributors, 'SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python,' *Nature Methods*, vol. 17, pp. 261–272, 2020. DOI: 10.1038/s41592-019-0686-2.

[65] S. K. Lam, A. Pitrou and S. Seibert, 'Numba: A llvm-based python jit compiler,' in *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, ser. LLVM '15, Austin, Texas: Association for Computing Machinery, 2015, ISBN: 9781450340052. DOI: 10.1145/2833157.2833162. [Online]. Available: https://doi.org/10.1145/2833157.2833162.

[66] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne and Q. Zhang, *JAX: Composable transformations of Python+NumPy programs*, version 0.3.13, 2018. [Online]. Available: http://github.com/google/jax.

[67] J. Bezanson, A. Edelman, S. Karpinski and V. B. Shah, 'Julia: A fresh approach to numerical computing,' *SIAM Review*, vol. 59, no. 1, pp. 65–98, Sep. 2017. DOI: 10.1137/141000671.

[68] S. S. Bergmann, *Documents and code related to my master's thesis*, The repository will be made public at the latest when the thesis is submitted., 2022. [Online]. Available: http://github.com/sjursb/masterthesis.

[69] R. Kern. 'Nep 19 - random number generator policy.' Accessed June 18, 2022. (May 2018), [Online]. Available: https://numpy.org/neps/nep-0019-rng-policy.html.

[70] M. E. O'Neill, 'Pcg: A family of simple fast space-efficient statistically good algorithms for random number generation,' Harvey Mudd College, Claremont, CA, Tech. Rep. HMC-CS-2014-0905, Sep. 2014.

[71] J. K. Salmon, M. A. Moraes, R. O. Dror and D. E. Shaw, 'Parallel random numbers: As easy as 1, 2, 3,' in *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, ser. SC '11, Seattle, Washington: Association for Computing Machinery, 2011, ISBN: 9781450307710. DOI: 10.1145/2063384.2063405. [Online]. Available: https://doi.org/10.1145/2063384.2063405.

[72] E. Hairer, 'Variable time step integration with symplectic methods,' in 2-3, vol. 25, Special issue on time integration (Amsterdam, 1996), 1997, pp. 219–227. DOI: 10.1016/S0168-9274(97)00061-5. [Online]. Available: https://doi.org/10.1016/S0168-9274(97)00061-5.

[73] A. Meurer, C. P. Smith, M. Paprocki, O. Čertík, S. B. Kirpichev, M. Rocklin, A. Kumar, S. Ivanov, J. K. Moore, S. Singh, T. Rathnayake, S. Vig, B. E. Granger, R. P. Muller, F. Bonazzi, H. Gupta, S. Vats, F. Johansson, F. Pedregosa, M. J. Curry, A. R. Terrel, Š. Roučka, A. Saboo, I. Fernando, S. Kulal, R. Cimrman and A. Scopatz, 'Sympy: Symbolic computing in python,' *PeerJ Computer Science*, vol. 3, e103, Jan. 2017, ISSN: 2376-5992. DOI: 10.7717/peerj-cs.103. [Online]. Available: https://doi.org/10.7717/peerj-cs.103.

NTNU

Kunnskap for en bedre verden