Structure-preserving General Linear Methods

Norton, Terry

Award date:
2015

Awarding institution:
University of Bath

Structure-preserving General Linear Methods

submitted by

Terence James Taylor Norton

for the degree of Doctor of Philosophy

of the

University of Bath

Department of Mathematical Sciences

July 2015

Summary

Geometric integration concerns the analysis and construction of structure-preserving numerical methods for the long-time integration of differential equations that possess some geometric property, e.g. Hamiltonian or reversible systems. In choosing a structure-preserving method, it is important to consider its efficiency, stability, order, and ability to preserve qualitative properties of the differential system, such as time-reversal symmetry, symplecticity and energy-preservation. Commonly, the symmetric or symplectic Runge–Kutta methods, or the symmetric or G-symplectic linear multistep methods, are chosen as candidates for integration. In this thesis, a class of structure-preserving general linear methods (GLMs) is considered as an alternative choice.

The research performed here includes the construction of a set of theoretical tools for analysing derivatives of B-series (a generalisation of Taylor series). These tools are then applied in the development of an *a priori* theory of parasitism for GLMs, which is used to prove bounds on the parasitic components of the method, and to derive algebraic conditions on the coefficients of the method that guarantee an extension of the time-interval of parasitism-free behaviour. A computational toolkit is also developed to help assist with this analysis, and for other analyses involving the manipulation of B-series and derivative B-series.

High-order methods are constructed using a newly developed theory of composition for GLMs, which is an extension of the classical composition theory for one-step methods. A decomposition result for structure-preserving GLMs is also given which reveals that a memory-efficient implementation of these methods can be performed. This decomposition result is explored further, and it is shown that certain methods can be expressed as the composition of several LMMs.

A variety of numerical experiments are performed on geometric differential systems to validate the theoretical results produced in this thesis, and to assess the competitiveness of these methods for long-time geometric integrations.

Acknowledgements

First of all, I would like to thank my supervisor Adrian Hill for his support, encouragement, and for keeping me entertained with a variety of a random factoids - usually in the form of the latest sporting results. I owe him a great debt in tea and coffee.

I must also thank the members of the department of mathematical sciences, and in particular, those of the numerical analysis group for allowing me the opportunity to share my research during the weekly seminars.

I would also like to thank John Butcher for many invaluable discussions over the years and for hosting my trip to New Zealand in 2012. I am deeply grateful for this truly unforgettable experience.

Personally, I would like to thank my family, and in particular my parents Sarah and James, for their constant support and encouragement throughout my time at University. I particularly appreciate the numerous attempts at memorising the title of this thesis. I also thank Jack, Lawrie, Melroy, Rob, Steve, Toby, Tom, and so many more, for all of your welcomed distractions. If not for you, I may have submitted sooner.

Finally, I must thank my partner, Chelsea. She has proven to be the greatest distraction of all, and is solely responsible for keeping my social life intact throughout the course of the write-up. For this, and so much more, I express my sincerest gratitude.

Contents

Chapter 1

Introduction

General linear methods (GLMs) are a class of iteration-based numerical methods for the integration of ordinary differential equations (ODEs). They were introduced by Butcher [5], Gear [28], Gragg and Stetter [31] as a framework to unify Runge–Kutta methods (RKMs) and linear multistep methods (LMMs); a simple illustration of this relationship is given in Figure 1-1.

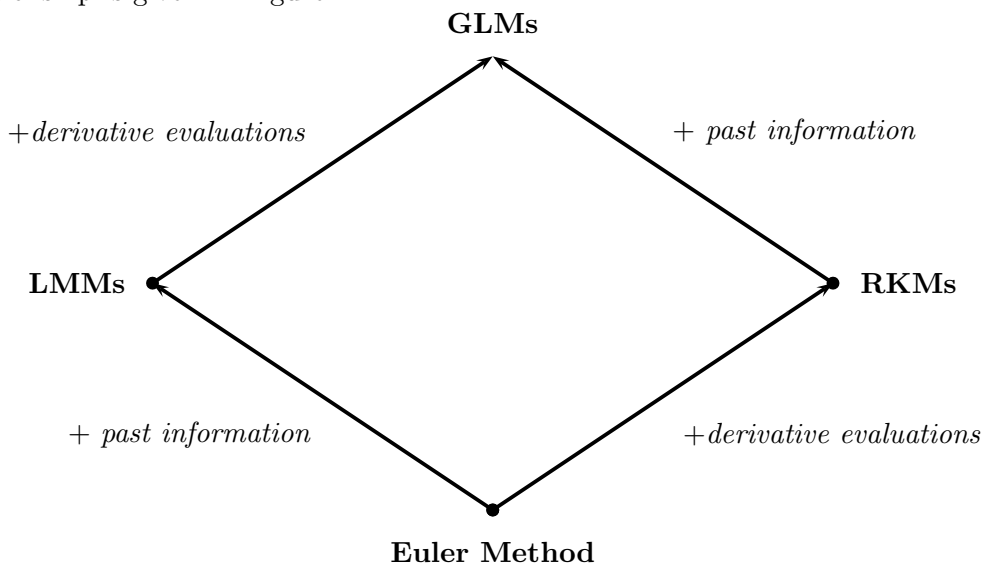


Figure 1-1: Illustration of the relationship between classical numerical methods.

In addition to unifying classical numerical methods, the GLM framework also incorporates non-trivial examples such as the cyclic composition of LMMs [25], the class of two-step RKMs [18], predictor-corrector methods [31], and, from meteorology, the RAW filter [61] to name but a few.

History of GLMs: Early examples of GLMs (though not known by that name) were given in the mid-sixties, where Butcher [5], Gear [28], Gragg and Stetter [31] considered generalised multistep methods; a prominent example being predictor-corrector methods. Shortly after, Butcher [6] introduced a unifying framework for analysing the convergence of these methods, as well as RKMs and LMMs.

Since then, a wide variety of GLMs have been developed. For example, *diagonally implicit multistage integration methods* (a.k.a. DIMSIMs) form a subclass under GLMs where each method is assigned a 'type' that identifies the class of problems for which it is ideally suited for, i.e. non-stiff or stiff problems, and whether it has a parallelizable element to its implementation (see e.g. [7, 12]). *Almost Runge–Kutta methods* form another subclass under GLMs (see e.g. [8, 49]). Here, methods are designed such that they retain the desirable stability properties of explicit RKMs whilst overcoming some of their associated disadvantages, such as low stage order. Similarly, the subclass of *methods with inherent Runge–Kutta stability* [10] are subject to the same stability analysis as RKMs (in the language of GLMs, these methods have a stability matrix with a single non-zero eigenvalue). The final subclass we mention is the subject of this thesis, that is, *structure-preserving* GLMs. These methods possess properties analogous to those of conservative differential systems, e.g. time-reversal symmetry, symplecticity of the flow, energy-preservation.

For further reading on GLMs, see Butcher's 2006 monograph [9], and Chapter 5 of [10], where many examples of above methods can be found.

Structure-preserving methods: A numerical method can be viewed as a map that approximates the evolution operator (also known as the *flow map*) of some differential system. Classical methods such as RK4 achieve a high-order approximation to this operator and are often used in practical applications. While high-order is an attractive property for a method to possess, it alone is not enough to guarantee that other qualitative properties of the differential system are preserved. Methods that satisfy a discrete (or otherwise related) analogue of one of these qualitative properties are called *structure-preserving*, or equivalently, *geometric integrators*.

Symmetry of the evolution operator (with respect to time) is a property of reversible systems. Here, the state of the system after a forward evolution followed immediately by a backward evolution remains unchanged (see e.g. [36, Ch. V]). Numerical methods that are capable of reproducing this behaviour (to machine precision) are called *symmetric*. Examples include the implicit midpoint rule, Gauss methods, Lobatto IIIA/IIIB which are symmetric RKMs (see e.g. [36, Ch. II]). All of these methods are implicit (though the Lobatto methods permit a single explicit stage), which can be

computationally expensive for large-dimensional systems. This cost can be mitigated to some extent by considering the subclass of symmetric, diagonally-implicit RKMs (DIRKs)[1]. Other examples include LMMs based on open/closed Newton–Cotes formulae such as Leapfrog and Simpson's rule (see e.g. [36, Ch. XV]). Here, methods may be either explicit or implicit. Unfortunately, they are prone to parasitic instability which becomes significant in a time $t = O(1)$ [33].

For Hamiltonian systems, the evolution operator is a *symplectic* transformation [47],[36, Ch. VI], which implies that the variational equation conserves quadratic quantities [1, 51]. Only one-step methods (OSMs), such as RKMs, are capable of satisfying the discrete analogue of symplecticity [44, 50]. Examples of symplectic RKMs include the implicit midpoint rule, Gauss methods, Lobatto IIIC (see e.g. [36, Ch. II]). As with symmetric RKMs, these methods are necessarily implicit [50], though the associated cost can again be mitigated through consideration of symplectic DIRKs. Such methods were investigated in [53], and were found to be compositions of the implicit midpoint rule.

An r-step LMM, with inputs approximating the ODE solution at times $t_{n+r}, \ldots, t_n$, cannot be symplectic in the usual sense [60], that is, its underlying one-step method (UOSM) is not a symplectic transformation. However, since LMMs operate in a higher-dimensional phase space, then it is natural to question what the appropriate definition of a symplectic multistep method should be. This leads to the alternative definition of *G-symplecticity*, an idea based on the work of Dahlquist's theory of G-stability [22], which essentially describes the higher-dimensional analogue of a symplectic transformation. Hairer [34] has explored the connection between G-symplecticity and standard symplecticity of LMMs, and has shown that the UOSM is *conjugate-symplectic*. In other words, there exists (as a formal B-series) a similarity transformation for the UOSM such that the corresponding method is symplectic. Other connections have been investigated by Eirola and Sanz-Serna [26], who have shown that a LMM is G-symplectic if and only if it is symmetric. Unfortunately, this also means that G-symplectic LMMs are susceptible to parasitic instability over very short time intervals.

Other classes of structure-preserving methods include those designed for the integration of second-order differential equations (i.e. for problems of the form $y'' = f(y)$.) and separable Hamiltonian systems. For example, there are *partitioned RKMs*, such as the symplectic Euler method which admits an explicit implementation for these problems. Also, there is the class of *partitioned LMMs*. Here, a popular example is the Störmer–Verlet method (see e.g. [36, Ch. I]) which is symmetric, symplectic and

[1] These methods have a lower-triangular stage matrix which means internal stages can be solved sequentially, and in the space of the differential system.

explicit. There are also *symmetric multistep methods for second-order Hamiltonian systems*, which have been shown to possess excellent energy preservation behaviour [35, 33],[36, Ch. XV], as demonstrated by long-time integrations of the outer solar system [43, 48].

In summarising the methods discussed above, we remark that while there exist a number of excellent choices for the integration of second-order differential equations, and separable Hamiltonian systems, there is no clear consensus on the best choice for the integration of general, first-order geometric problems: structure-preserving LMMs suffer from parasitism, and structure-preserving RKMs are necessarily implicit. Since many real-world problems are of this form (cf. Chapter 7), accurate and efficient methods are still highly sought after. Thus, this motivates the search for new methods that overcome the destructive effects of parasitism, whilst keeping the level of implicitness in the method to a minimum.

Structure-preserving GLMs can be symmetric or G-symplectic, and can be designed such that they have the DIRK property, i.e. diagonally-implicit, with some methods permitting a mixture of implicit and explicit stages. Furthermore, it has been shown that methods can be constructed such that they do not suffer from parasitic instability over intervals of length $O(h^{-2})$, where h denotes the time-step [16, 13, 17]. This is an important result for multivalue methods as the presence of parasitism is usually enough to discourage the use of them in practical applications (see below for more on the topic of parasitism). In this thesis, it will be shown that GLMs can be designed with even longer intervals of parasitism-free behaviour (cf. Chapter 3), and that the geometric invariants of a given problem are well-preserved over long times (cf. Chapter 7), therefore providing strong support for the consideration of these methods as alternative candidates for geometric integration.

Parasitism: An important topic in this thesis, and in the analysis of multivalue methods in general, is parasitism. Loosely speaking, parasitism describes the unacceptable growth of perturbations, e.g. rounding error, that can arise in methods with multiple inputs[2]. Dahlquist [21] studied this phenomenon for LMMs by decomposing the truncation error of a method into parasitic and non-parasitic terms, each of which is the solution to some differential system (see also [33]). For weakly-stable methods, such as Leapfrog, it was found that the parasitic solution may grow without bound for problems that are stable for both positive and negative time, e.g. Hamiltonian systems.

Since Dahlquist's work, the analysis of parasitism in multistep methods, as well as

[2]One-step methods do not suffer from parasitism as perturbations can only be made along the trajectory of the solution, and are therefore subject to standard stability analysis.

explaining the long-time preservation of invariants, is usually performed using backward error analysis [32, 35, 33][36, Ch. XV]. This involves the study of the solution to a *modified ODE* that the method satisfies exactly. Using this analysis, it can be shown that the parasitic components of structure-preserving LMMs will only remain bounded on intervals of length $t = O(1)$, which is in agreement with Dahlquist's earlier work. This is an unfortunate result as these methods would otherwise be excellent candidates for the long-time integration of geometric problems.

In contrast, Hairer and Lubich [35],[36, Ch. XV] have studied symmetric multistep methods for second order Hamiltonian systems, and have shown that the parasitic components of an r-step method can remain bounded over intervals of length $t = O(h^{-r-2})$. Furthermore, they have shown that the total energy, and other quadratic invariants, are preserved up to $O(h^r)$ over this interval.

In recent work, D'Ambrosio and Hairer [23] have used backward error analysis to study parasitism in structure-preserving GLMs. In particular, they derive a bound for the parasitic components which explains the good behaviour of these methods over modest intervals. In Chapter 3 of this thesis, we take an *a priori* approach to the analysis of parasitism in a fashion similar to [11]. This yields a framework for proving bounds on the parasitic components (which agrees with that obtained using backward error analysis), as well as enabling the derivation of algebraic conditions for extending the interval of parasitism-free behaviour.

1.1 Outline of the thesis

This thesis is organised as follows. In Chapter 2 we provide the mathematical background for GLMs. We begin by covering the fundamental aspects of GLMs, e.g. notation, tableau representation, starting and finishing methods. We then move on to introduce B-series, the underlying one-step method for GLMs, as well as giving an introduction to parasitism analysis. Finally, we introduce the class of structure-preserving GLMs, i.e. symmetric and G-symplectic GLMs, and explain the derivation behind the corresponding algebraic conditions on the method coefficients.

In Chapter 3, we develop a set of theoretical tools for analysing derivatives of numerical methods. In particular, we construct a formal power series, that we call a *derivative B-series*, which is built using ideas from B-series analysis (see e.g. [36, Ch. III]). These tools are then used to develop an *a priori* theory of parasitism for GLMs. Here, we prove bounds on the parasitic components of the method, and derive algebraic conditions that guarantee an extension of the interval of parasitism-free behaviour. A demonstration that these conditions can be satisfied is given in the form of two fourth-

order, symmetric GLMs.

In Chapter 4, we develop a computational toolkit for assisting the analysis of GLMs. Here, we take an object-oriented approach to programming and describe how to represent rooted trees, B-series and derivative B-series as objects. The implementation details behind some of the advanced operations performed on these objects are also discussed. Applications are given at the end of the chapter where we demonstrate how to use the tools to determine the order of the method, derive its underlying one-step method and perform a parasitism analysis in line with the theory of Chapter 3.

In Chapter 5, we investigate composition of GLMs as a technique for obtaining high-order methods. Two approaches are considered: In the first, we consider a subclass of GLMs where the methods are assumed to take Nordsieck inputs. In the second approach, we consider a generalisation of the composition formulae used for OSMs [63, 58, 45]. The results of this latter approach can be applied to symmetric GLMs as a way of obtaining methods of arbitrarily high order. This is demonstrated computationally in Chapter 7 where methods of order 6 and 8 are constructed.

In Chapter 6, we present a result on the decomposition of structure-preserving GLMs into single-stage GLMs. The connection between these latter methods and LMMs is explored and conditions are found such that a structure-preserving GLM can be written as the composition of several (possibly symmetric) LMMs.

In Chapter 7, we perform a variety of numerical experiments to illustrate some of the key results of this thesis. In particular, we estimate the interval of parasitism-free behaviour for a given set of structure-preserving GLMs, and we make several efficiency comparisons with structure-preserving RKMs on various geometric problems. We also perform long-time integrations using GLMs with the best parasitism-free behaviour, and assess how well they preserve invariants of geometric problems. Finally, we verify computationally that a theoretical order increase can be attained using of the composition formulae given in Chapter 5.

Chapter 2

Background

General linear methods (GLMs) are a class of time-stepping methods for the integration of ordinary differential equations (ODEs). Throughout this thesis, we consider the application of these methods to first order, autonomous, initial value problem (IVPs) of the form

$$\frac{\mathrm{d}y}{\mathrm{d}t}(t) = f(y(t)), \qquad y(0) = y_0, \qquad t \in [-T, T], \tag{2.1}$$

where $y : [-T, T] \to X$, $f : X \to X$, $T > 0$, and it is assumed that $X = \mathbb{R}^d$, $d \in \mathbb{N}$ is an open subset. Furthermore, we assume that the standard Lipschitz condition holds: There exists an $L > 0$ such that

$$||f(y) - f(z)|| \leq L||y - z||, \qquad \text{for all} \qquad y, z \in X.$$

and we express the solution of (2.1) in terms of the initial data y_0 and the *flow map* (also known as the *evolution operator*) $\varphi_t : X \to X$ such that

$$y(t) = \varphi_t(y_0).$$

Of particular interest will be Hamiltonian IVPs. These are described by an even-dimensional system of first order ODEs of the form

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} p(t) \\ q(t) \end{bmatrix} = \begin{bmatrix} -\nabla_q H(p(t), q(t)) \\ \nabla_p H(p(t), q(t)) \end{bmatrix}, \qquad \begin{bmatrix} p(0) \\ q(0) \end{bmatrix} = \begin{bmatrix} p_0 \\ q_0 \end{bmatrix}, \qquad t \in [-T, T], \tag{2.2}$$

where $H : X \to \mathbb{R}$ is the Hamiltonian, $q, p : [-T, T] \to \mathbb{R}^m$, and $p_0, q_0 \in \mathbb{R}^m$, $m \in \mathbb{N}$.

2.1 Introduction to GLMs

The GLM framework incorporates a wide variety of numerical methods, including the classical Runge–Kutta methods (RKMs) and linear multistep methods (LMMs) [6, 9]. Below, we introduce the essential components and theoretical tools that make up this framework.

2.1.1 Notation

A GLM is formed of s-many *stage equations* and r-many *update equations*, with $r, s \in \mathbb{N}$. At time $t = nh$, where $n \in \mathbb{N}_0$ denotes the step number and $h \in \mathbb{R}\backslash\{0\}$ denotes the time-step, the method acts upon a set of inputs $y_1^{[n]}, \dots, y_r^{[n]} \in X$ and generates outputs $y_1^{[n+1]}, \dots, y_r^{[n+1]} \in X$ via the following equations:

$$Y_i = h\sum_{j=1}^{s} a_{ij} f(Y_j) + \sum_{j=1}^{r} u_{ij} y_j^{[n]}, \qquad i = 1, \dots, s, \qquad \text{[Stage Equations]},$$

$$y_i^{[n+1]} = h\sum_{j=1}^{s} b_{ij} f(Y_j) + \sum_{j=1}^{r} v_{ij} y_j^{[n]}, \qquad i = 1, \dots, r, \qquad \text{[Update Equations]},$$

where $a_{ij} \in \mathbb{R}$, $b_{ij}, u_{ij}, v_{ij} \in \mathbb{C}$ denote the method coefficients and $Y_i \in X$ are the stage values. To simplify notation, we define super-vectors

$$y^{[n+1]} = \begin{bmatrix} y_1^{[n+1]} \\ y_2^{[n+1]} \\ \vdots \\ y_r^{[n+1]} \end{bmatrix}, \qquad y^{[n]} = \begin{bmatrix} y_1^{[n]} \\ y_2^{[n]} \\ \vdots \\ y_r^{[n]} \end{bmatrix}, \qquad Y = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_s \end{bmatrix}, \qquad F(Y) = \begin{bmatrix} f(Y_1) \\ f(Y_2) \\ \vdots \\ f(Y_s) \end{bmatrix},$$

where $y^{[n]}, y^{[n+1]} \in X^r$ and $Y, F \in X^s$, and we also define matrices

$$A = [a_{ij}] \in \mathbb{R}^{s\times s}, \qquad B = [b_{ij}] \in \mathbb{C}^{r\times s}, \qquad U = [u_{ij}] \in \mathbb{C}^{s\times r}, \qquad V = [v_{ij}] \in \mathbb{C}^{r\times r}.$$

Then, the stage and update equations may be compactly written as

$$\begin{bmatrix} Y \\ y^{[n+1]} \end{bmatrix} = \begin{bmatrix} A \otimes I_X & U \otimes I_X \\ B \otimes I_X & V \otimes I_X \end{bmatrix} \begin{bmatrix} hF(Y) \\ y^{[n]} \end{bmatrix}. \tag{2.3}$$

Here, $\otimes$ denotes a Kronecker product and I_X is the identity matrix defined on X.

Often, explicit reference to the stage and update equations is not required. Instead, it is more convenient to use the map determined by equations (2.3), i.e. we define

$$\mathcal{M}_h : X^r \to X^r, \qquad \text{such that} \qquad y^{[n+1]} = \mathcal{M}_h\left(y^{[n]}\right).$$

Tableau representation: Every GLM is essentially characterised by its coefficient matrices. Thus, we refer to a method by its *GLM tableau*,

$$\left[\begin{array}{c|c} A & U \\ \hline B & V \end{array}\right].$$

This representation is particularly useful for verifying properties of the method, as well as giving an indication of computational cost, e.g. a strictly lower triangular stage matrix A immediately implies an explicit method.

Example 2.1. Consider the forward Euler (FE) and backward Euler (BE) methods:

$$\underset{\text{[FE]}}{y^{[n+1]} = y^{[n]} + hf\left(y^{[n]}\right),} \qquad \underset{\text{[BE]}}{y^{[n+1]} = y^{[n]} + hf\left(y^{[n+1]}\right).}$$

Expressed in terms of stage and update equations, these are written as

$$\begin{aligned} Y_1 &= y^{[n]}, & Y_1 &= y^{[n]} + hf(Y_1), \\ y^{[n+1]} &= y^{[n]} + hf\left(Y_1\right), & y^{[n+1]} &= y^{[n]} + hf\left(Y_1\right). \\ &\text{[FE]} & &\text{[BE]} \end{aligned}$$

The coefficient matrices are obtained from reading off the coefficients of the $y^{[n]}$ and $hf(Y_1)$. Thus, the GLM tableaux for these methods are respectively given as

$$\left[\begin{array}{c|c} 0 & 1 \\ \hline 1 & 1 \end{array}\right], \qquad \text{and} \qquad \left[\begin{array}{c|c} 1 & 1 \\ \hline 1 & 1 \end{array}\right].$$

◇

Example 2.2. The RAW time filter presented in [61] is a special technique for suppressing the spurious computational mode[1] of the Leapfrog method. It works by applying a standard Leapfrog update followed by two filter operations. The numerical scheme

[1] In meteorology, parasitic components of multivalue methods are usually referred to as the spurious, computational modes.

is presented below

$$\begin{aligned} y_{n+1} &= \overline{\overline{y}}_{n-1} + 2hf(\overline{y}_n), \\ \overline{y}_{n+1} &= y_{n+1} - \frac{\nu(1-\alpha)}{2}(\overline{\overline{y}}_{n-1} - 2\overline{y}_n + y_{n+1}), \\ \overline{\overline{y}}_n &= \overline{y}_n + \frac{\nu\alpha}{2}(\overline{\overline{y}}_{n-1} - 2\overline{y}_n + y_{n+1}), \end{aligned}$$

where $\alpha, \nu \in [0,1]$ are filter parameters, $y_n, \overline{y}_n, \overline{\overline{y}}_n \approx y(nh)$ and $\overline{\overline{y}}_n$ is chosen as the appropriate solution. Typical values for the filter parameters are $\nu = 0.2$, $\alpha = 0.53$.

The above scheme determines the two-step map $(\overline{\overline{y}}_{n-1}, \overline{y}_n) \mapsto (\overline{\overline{y}}_n, \overline{y}_{n+1})$ and may be written in terms of a GLM: Let $y^{[n]} := [\overline{\overline{y}}_{n-1}^T, \overline{y}_n^T]^T$, and recast the scheme into matrix-vector form, i.e.

$$\begin{aligned} Y_1 &= e_2^T y^{[n]}, \\ y^{[n+1]} &= \begin{bmatrix} \nu\alpha & 1-\nu\alpha \\ 1-\nu(1-\alpha) & \nu(1-\alpha) \end{bmatrix} y^{[n]} + h \begin{bmatrix} \nu\alpha \\ 2-\nu(1-\alpha) \end{bmatrix} f(Y_1), \end{aligned}$$

where $e_2 = [0,1]^T$ and Kronecker products have been applied implicitly. Then, we can read off the coefficient matrices to give the corresponding GLM tableau,

$$\left[\begin{array}{c|cc} 0 & 0 & 1 \\ \hline \nu\alpha & \nu\alpha & 1-\nu\alpha \\ 2-\nu(1-\alpha) & 1-\nu(1-\alpha) & \nu(1-\alpha) \end{array}\right].$$

Notice that if we set $\nu = 0$, we obtain the Leapfrog method expressed as a GLM:

$$\left[\begin{array}{c|cc} 0 & 0 & 1 \\ \hline 0 & 0 & 1 \\ 2 & 1 & 0 \end{array}\right]. \tag{2.4}$$

◇

2.1.2 Convergence, consistency and stability

Let us now introduce the key concepts of convergence, consistency and stability that play a fundamental role in numerical analysis. The following definitions closely resemble those introduced in [10]:

Definition 2.3 (**Consistency and stability**). Let (A, U, B, V) denote the coefficient matrices of a GLM. Then, the method is said to be

(a) *Preconsistent*, if $\zeta_1 = 1$ is an eigenvalue of V, i.e. there exist *preconsistency vectors* $u, w \in \mathbb{C}^r \backslash \{0\}$, satisfying $w^{\mathsf{H}} u = 1$, such that $Vu = u$.[2]

(b) *Consistent*, if it is preconsistent, and

$$\begin{aligned} Uu &= \mathbb{1}, && \text{for} && \mathbb{1} = [1, 1, \ldots, 1]^T \in \mathbb{R}^s, \\ B\mathbb{1} + Vv &= u + v, && \text{for some} && v \in \mathbb{C}^r \backslash \{0\}, \end{aligned}$$

where u is the (right) preconsistency vector given in (a).

(c) *Stable*, if it is *zero-stable*[3], i.e. $\sup_{n \geq 0} ||V^n|| < \infty$.

(d) *Strictly-stable*, if it is strictly zero-stable, i.e.

$$\zeta_1 = 1, \qquad |\zeta_i| < 1, \qquad i = 2, \ldots, r,$$

where ζ_i, $i = 1, \ldots, r$ are the eigenvalues of V.

The necessity for the definitions (a)-(c) can be seen by considering the following problems: For consistency, we consider a GLM applied to the IVP

$$\frac{\mathrm{d}y}{\mathrm{d}t} = 1, \qquad y(0) = y_0 \in \mathbb{R}, \qquad t \in [0, 1].$$

Suppose that we fix $y^{[0]} = uy_0 + hv$, where v is as given in Definition 2.3(b). Then, for $n \geq 1$, a consistent GLM will yield

$$y^{[n]} := \mathcal{M}_h^n(y^{[0]}) = u(y_0 + nh) + hv = uy(nh) + hv.$$

This simple example also highlights an important feature of GLMs, namely, inputs do not necessarily have to be approximations of $y(nh)$. In such a case, extra work at the end of the iteration is required to obtain the actual numerical approximation to the solution. Further discussion on this topic is covered in the following sections.

To see the necessity of stability, we instead study the homogeneous IVP

$$\frac{\mathrm{d}y}{\mathrm{d}t} = 0, \qquad y(0) = y_0 \in \mathbb{R}, \qquad t \in [0, 1].$$

[2] The choice of w is usually made such that $u_1 = 1$, i.e. the first component is equal to one.

[3] Dahlquist's work on the stability of multistep methods (see e.g. [33]) revealed that an instability can arise when the method possesses double roots. Thus, for GLMs, we usually assume that the unimodular eigenvalues of V are distinct.

Here, we note that the power-boundedness of V arises from the fact that $y^{[n]} = V^n y^{[0]}$.

Definition 2.4 (Convergence [10])**.** A GLM is *convergent* if for any IVP of the form (2.1), there exists a $u \in \mathbb{C}^r \backslash \{0\}$ and a map $\mathcal{S}_h : X \to X^r$ such that

1. $\mathcal{S}_h(y_0) \longrightarrow u \otimes y_0$ as $h \longrightarrow 0$,
2. $y^{[n]} := \mathcal{M}^n_{\frac{T}{n}} \circ \mathcal{S}_{\frac{T}{n}}(y_0) \longrightarrow u \otimes y(T)$ as $n \longrightarrow \infty$,

for any $T > 0$. (Note: Future usage of the Kronecker product shall be applied implicitly, unless otherwise stated.)

It has been shown, for example in [10], that a consistent and stable GLM is convergent. Conversely, we also have that a convergent GLM is both consistent and stable.

2.1.3 Starting and finishing methods

Every GLM requires a set of starting values $y^{[0]}$ to initialise the method. To obtain these values, a *starting method* is implemented. As inputs to a GLM belong to the product space X^r, a *finishing method* is also required to obtain approximations to the solution $y(nh) \in X$.

Definition 2.5. A *starting method* is defined to be the map $\mathcal{S}_h : X \to X^r$, where

$$\mathcal{S}_h(y_0) = y^{[0]}, \qquad y_0 \in X.$$

A *finishing method* is defined to be the map $\mathcal{F}_h : X^r \to X$ such that

$$\mathcal{F}_h(y^{[n]}) \approx y(nh), \qquad \mathcal{F}_h \circ \mathcal{S}_h(y_0) = y_0, \qquad y_0 \in X.$$

Remark 2.6. It is not a necessary requirement that $\mathcal{F}_h \circ \mathcal{S}_h(y_0) = y_0$ exactly, though many theoretical results assume this is the case. Instead, we could impose that $\mathcal{F}_h \circ \mathcal{S}_h(y_0) = y_0 + O(h^{p+1})$, where $p \in \mathbb{N}$ corresponds to the order (cf. Section 2.1.4) of the method. This relaxation affords greater freedom in the design of the finishing method.

Definition 2.7. A pair of starting and finishing methods are called *consistent* if

$$\begin{aligned} \mathcal{S}_h(y_0) &= uy_0 + hvf(y_0) + O(h^2), && y_0 \in X, \\ \mathcal{F}_h(y) &= w^{\mathsf{H}} y + O(h), && y \in X^r, \end{aligned}$$

where u, w, v are as in the definition of consistency, with w free except for $w^{\mathsf{H}} u = 1$.

Assuming that neither starting nor finishing method is expressed as a formal power series in h (cf. Section 2.3 for such a case), then both may be expressed in terms of stage and update equations (like a GLM). In particular, a consistent starting method is written as

$$\begin{bmatrix} Y_S \\ \mathcal{S}_h(y_0) \end{bmatrix} = \begin{bmatrix} A_S \otimes I_X & \mathbb{1}_S \otimes I_X \\ B_S \otimes I_X & u \otimes I_X \end{bmatrix} \begin{bmatrix} hF(Y_S) \\ y_0 \end{bmatrix},$$

and a consistent finishing method as

$$\begin{bmatrix} Y_F \\ \mathcal{F}_h\left(y^{[n]}\right) \end{bmatrix} = \begin{bmatrix} A_F \otimes I_X & U_F \otimes I_X \\ B_F \otimes I_X & w^{\mathsf{H}} \otimes I_X \end{bmatrix} \begin{bmatrix} hF(Y_F) \\ y^{[n]} \end{bmatrix},$$

where

$$A_S, A_F \in \mathbb{R}^{\widetilde{s}\times\widetilde{s}}, \qquad B_S \in \mathbb{C}^{r\times\widetilde{s}}, \qquad B_F \in \mathbb{C}^{1\times r}, \qquad U_F \in \mathbb{C}^{\widetilde{s}\times r}, \qquad \widetilde{s} \in \mathbb{N},$$

and it is assumed that

$$U_F u = \mathbb{1}_S, \qquad B_S \mathbb{1}_S = v, \qquad \text{where} \qquad \mathbb{1}_S = [1, \ldots, 1]^T \in \mathbb{R}^{\widetilde{s}}. \tag{2.5}$$

As with a GLM, we refer to these methods by their tableaux:

$$\left[\begin{array}{c|c} A_S & \mathbb{1}_S \\ \hline B_S & u \end{array}\right], \qquad \text{and} \qquad \left[\begin{array}{c|c} A_F & U_F \\ \hline B_F & w^{\mathsf{H}} \end{array}\right]. \tag{2.6}$$

Lemma 2.8. *Consider a pair of consistent starting and finishing methods, $\mathcal{S}_h$ and $\mathcal{F}_h$, determined by the tableaux in (2.6). If the coefficient matrices satisfy (2.5),*

$$A_F = A_S - U_F B_S, \qquad \textit{and} \qquad B_F = -w^{\mathsf{H}} B_S,$$

then $\mathcal{F}_h \circ \mathcal{S}_h(y_0) = y_0$ exactly.

Proof. First, let us consider the stage equations of $\mathcal{F}_h \circ \mathcal{S}_h(y_0)$:

$$\begin{aligned} Y_S &= hA_S F(Y_S) + \mathbb{1}_S y_0, \\ Y_F &= hA_F F(Y_F) + U_F \mathcal{S}_h(y_0), \\ &= hA_F F(Y_F) + hU_F B_S F(Y_S) + U_F u y_0. \end{aligned}$$

For sufficiently small h, there exist unique solutions to both sets of stage equations. Now, suppose Y_S is the solution to the first set of equations. Then, for $A_F = A_S - U_F B_S$,

we observe that $Y_F = Y_S$ is a solution for the second set, i.e. observe that

$$Y_S - hA_F F(Y_S) - hU_F B_S F(Y_S) - U_F u y_0 = Y_S - hA_S F(Y_S) - \mathbb{1}_S y_0 = 0,$$

where we have used $U_F u = \mathbb{1}_S$ from (2.5). By uniqueness of solutions, $Y_F = Y_S$ is the only solution to the second set of equations.

Now, let us consider the update equations of $\mathcal{F}_h \circ \mathcal{S}_h(y_0)$:

$$\mathcal{F}_h \circ \mathcal{S}_h(y_0) = w^{\mathsf{H}} \mathcal{S}_h(y_0) + hB_F F(Y_F) = w^{\mathsf{H}} u y_0 + h w^{\mathsf{H}} B_S F(Y_S) + hB_F F(Y_F).$$

If we set $B_F = -w^{\mathsf{H}} B_S$, it then follows from $Y_F = Y_S$ and $w^{\mathsf{H}} u = 1$ that

$$\mathcal{F}_h \circ \mathcal{S}_h(y_0) = y_0 + h w^{\mathsf{H}} B_S F(Y_S) - h w^{\mathsf{H}} B_S F(Y_S) = y_0,$$

as required. □

Hereafter, it shall be assumed that starting and finishing methods are consistent and are determined respectively by the tableaux

$$\left[\begin{array}{c|c} A_S & \mathbb{1}_S \\ \hline B_S & u \end{array}\right], \qquad \text{and} \qquad \left[\begin{array}{c|c} A_S - U_F B_S & U_F \\ \hline -w^{\mathsf{H}} B_S & w^{\mathsf{H}} \end{array}\right]. \tag{2.7}$$

Remark 2.9. Note that if $w^{\mathsf{H}} B_S = 0$ then the finishing method reduces to w^{H}. In this situation we call it *trivial*. This is a desirable property for a finishing method to possess as it implies that no further function evaluations are made. This is particularly useful whenever we are interested computing dense output since this requires the finishing method to be applied frequently.

Example 2.10. Consider the following GLM from [10, p. 416]:

$$\left[\begin{array}{cc|cc} \frac{3+\sqrt{3}}{6} & 0 & 1 & -\frac{3+2\sqrt{3}}{3} \\ -\frac{\sqrt{3}}{3} & \frac{3+\sqrt{3}}{6} & 1 & \frac{3+2\sqrt{3}}{3} \\ \hline \frac{1}{2} & \frac{1}{2} & 1 & 0 \\ \frac{1}{2} & -\frac{1}{2} & 0 & -1 \end{array}\right]. \tag{2.8}$$

The starting values $y^{[0]}$ for the method are required to be of the form

$$\begin{bmatrix} y_0 \\ h^2 \frac{\sqrt{3}}{12} \frac{\mathrm{d}^2 y}{\mathrm{d}t^2}(0) - h^4 \left(\frac{\sqrt{3}}{108} \frac{\mathrm{d}^4 y}{\mathrm{d}t^4}(0) - \frac{9+5\sqrt{3}}{216} f'(y_0) \frac{\mathrm{d}^3 y}{\mathrm{d}t^3}(0) \right) \end{bmatrix},$$

which can be approximated using a starting method with the following tableau

$$\left[\begin{array}{ccc|c} 0 & 0 & 0 & 1 \\ \frac{1}{3} & \frac{1}{3} & 0 & 1 \\ -\frac{\sqrt{3}}{4} & \frac{6+3\sqrt{3}}{4} & -\frac{1+\sqrt{3}}{2} & 1 \\ \hline 0 & 0 & 0 & 1 \\ -\frac{5\sqrt{3}}{24} & \frac{3\sqrt{3}}{8} & -\frac{\sqrt{3}}{6} & 0 \end{array}\right].$$

For this particular method, the finishing method is trivial (observe that the first row of B_S in the starting method is exactly zero).

◇

Approximation to the solution: Having introduced the notation for GLMs, and their starting and finishing methods, we can now demonstrate how to obtain numerical approximations to the solution of the IVP under consideration.

Definition 2.11. The *numerical method as a whole* is written as the composite map

$$\mathcal{F}_h \circ \mathcal{M}_h^n \circ \mathcal{S}_h(y_0) =: y_n, \tag{2.9}$$

where y_n denotes the numerical approximation to $y(nh)$.

Notice here that the finishing method is a passive procedure; that is, it does not feed back into the following update step. Thus, for applications that only require a sample of numerical approximations (as opposed to dense output), the finishing method is only applied occasionally.

Note also that the starting method is only applied once. This is important from a practical point of view as it usually means its computational cost can be neglected. Moreover, we can attempt to minimise the cost of the associated finishing method by choosing a more expensive starting method, i.e. by considering an A_S matrix that is full, we can attempt to make $A_F =: A_S - U_F B_S$ strictly lower-triangular, resulting in an explicit finishing method.

2.1.4 Order

As can be seen in Example 2.10, not all inputs to a GLM are an approximation to the exact solution. Thus, when attempting to the define the *order* of a GLM, a greater degree of flexibility is required to cover the case of general inputs.

Definition 2.12. The pair $(\mathcal{M}_h, \mathcal{S}_h)$ is of *order* $p \in \mathbb{N}$ if

$$\mathcal{M}_h \circ \mathcal{S}_h(y_0) = \mathcal{S}_h \circ \varphi_h(y_0) + C(y_0)h^{p+1} + O(h^{p+2}), \tag{2.10}$$

where $\varphi_h(y_0)$ is the time-h evolution of the IVP, and $C(y_0) \in X^r \backslash \{0\}$ is a vector depending on the method coefficients and various derivatives of f evaluated at y_0.

Definition 2.13. The *maximal order* of a GLM is given by the highest order over all feasible $\mathcal{S}_h$.

The order of the method essentially describes the accuracy of the update step. Closely related to this is the *stage order* of the method, i.e. the accuracy of the stage equations.

Definition 2.14. The pair $(\mathcal{M}_h, \mathcal{S}_h)$ is of *stage order* $q \in \mathbb{N}$ if

$$\overline{\varphi}_{ch}(y_0) - hAF(\overline{\varphi}_{ch}(y_0)) - U\mathcal{S}_h(y_0) = O(h^{q+1}), \quad \text{with} \quad \overline{\varphi}_{ch}(y_0) := \begin{bmatrix} \varphi_{c_1 h}(y_0) \\ \varphi_{c_2 h}(y_0) \\ \vdots \\ \varphi_{c_s h}(y_0) \end{bmatrix},$$

where $c := A\mathbb{1} + v$, and v is as in the definition of consistency.

Stage order is particularly important for stiff problems where it possible to observe a reduction in the order of a method to the value of its stage order [19].

2.1.5 Operations

There exist various operations that can be performed on GLMs. Here, we introduce two important operations that will be used frequently throughout this thesis.

Composition: There are many situations in which we must consider a composition of GLMs (cf. Section 2.1.3 where we computed the composition of $\mathcal{F}_h \circ \mathcal{S}_h(y_0)$). Below, a formula is given in terms of coefficient matrices that shows how to perform this operation: Consider the composition of two GLMs $\mathcal{M}_h^{(2)} \circ \mathcal{M}_h^{(1)}(y)$. The corresponding update equation is written as

$$\begin{aligned} \mathcal{M}_h^{(2)} \circ \mathcal{M}_h^{(1)}(y) &= hB^{(2)}F(Y^{(2)}) + V^{(2)}\mathcal{M}_h^{(1)}(y), \\ &= hB^{(2)}F(Y^{(2)}) + hV^{(2)}B^{(1)}F(Y^{(1)}) + V^{(2)}V^{(1)}y, \end{aligned}$$

with stage equations given by

$$\begin{aligned} Y^{(1)} &= hA^{(1)}F(Y^{(1)}) + U^{(1)}y, \\ Y^{(2)} &= hA^{(2)}F(Y^{(2)}) + U^{(2)}\mathcal{M}_h^{(1)}(y), \\ &= hA^{(2)}F(Y^{(2)}) + hU^{(2)}B^{(1)}F(Y^{(1)}) + U^{(2)}V^{(1)}y, \end{aligned}$$

where the Kronecker products have been applied implicitly. The tableau of the composed method is then given by

$$\left[\begin{array}{cc|c} A^{(1)} & 0 & U^{(1)} \\ U^{(2)}B^{(1)} & A^{(2)} & U^{(2)}V^{(1)} \\ \hline V^{(2)}B^{(1)} & B^{(2)} & V^{(2)}V^{(1)} \end{array}\right]. \tag{2.11}$$

This formula can be applied repeatedly for compositions of many different methods.

Equivalence: There are times where, based on tableaux alone, GLMs appear to be distinct. However, after a practical application they yield identical numerical results (up to rounding error). In such cases, these methods are said to be *equivalent*.

Definition 2.15. Consider a pair of GLMs $\mathcal{M}_h^{(1)}$, $\mathcal{M}_h^{(2)}$. Then, the two are said to be (T,P)-*equivalent* if there exists an invertible matrix $T \in \mathbb{C}^{r\times r}$ and an $s\times s$ permutation matrix P such that their coefficient matrices satisfy

$$\left[\begin{array}{c|c} A^{(2)} & U^{(2)} \\ \hline B^{(2)} & V^{(2)} \end{array}\right] = \left[\begin{array}{c|c} P^{-1}A^{(1)}P & P^{-1}U^{(1)}T \\ \hline T^{-1}B^{(1)}P & T^{-1}V^{(1)}T \end{array}\right].$$

(P)ermutation-equivalence arises from the fact that

$$F(Y) = PP^{-1}F(Y) = PF(P^{-1}Y).$$

(T)ransformation-equivalence arises from studying the numerical method as a whole:

$$\mathcal{F}_h\circ\mathcal{M}_h^n\circ\mathcal{S}_h(y_0) = \mathcal{F}_h\circ(TT^{-1})\mathcal{M}_h^n\circ(TT^{-1})\mathcal{S}_h(y_0) = (\mathcal{F}_h\circ T)(T^{-1}\mathcal{M}_h\circ T)^n(T^{-1}\mathcal{S}_h)(y_0).$$

Remark 2.16. Here, it should be noted that T does not necessarily have to be a linear transformation. Instead, we could replace T with the nonlinear map $T_h : X^r \to X^r$. This type of transformation is considered in Chapter 5.

Notice that under the transformation T, both the starting and finishing methods are also altered. This results in the following tableaux

$$\left[\begin{array}{c|c} A_S & \mathbb{1}_S \\ \hline T^{-1}B_S & T^{-1}u \end{array}\right], \qquad \text{and} \qquad \left[\begin{array}{c|c} A_S - U_F B_S & U_F T \\ \hline -w^{\mathsf{H}} B_S & w^{\mathsf{H}} T \end{array}\right].$$

2.2 B-series

A key theoretical tool in the analysis of GLMs are *B-series*, which are a generalisation of the classical Taylor series. They are used frequently in the analysis of RKMs (see e.g. [36, Ch. III]). For example, with backward error analysis to explain the long-time energy preservation associated with structure-preserving methods [36, Ch. IX].

As was shown in the previous section, we can expect a GLM to take very general inputs, i.e. described in terms of h, y, f and its various derivatives. These types of inputs are known as B-series. Before we formally introduce them, let us first look at a simple example: Consider the second and third derivatives of the solution to IVP (2.1), subject to a scaling by time-step h:

$$\begin{aligned} h^2\frac{\mathrm{d}^2 y}{\mathrm{d}t^2} &= h^2\frac{\mathrm{d}}{\mathrm{d}t}\frac{\mathrm{d}y}{\mathrm{d}t} = h^2\frac{\mathrm{d}}{\mathrm{d}t}f(y) = h^2 f'(y)f(y), \\ h^3\frac{\mathrm{d}^3 y}{\mathrm{d}t^3} &= h^3\frac{\mathrm{d}}{\mathrm{d}t}\frac{\mathrm{d}^2 y}{\mathrm{d}t^2} = h^3\frac{\mathrm{d}}{\mathrm{d}t}f'(y)f(y) = h^3 f''(y)(f(y), f(y)) + h^3 f'(y)f'(y)f(y), \end{aligned}$$

where $'$ denotes differentiation with respect to y, and $f''(y)$ is a bilinear map. The right-hand side gives an alternative representation of these derivatives, in terms of h,f and its derivatives. Such a representation is known as a B-series.

Trees and differentials: In the example above, the f-terms on the right-hand side are called *elementary differentials*. Associated with each one of these is a *rooted tree*.

Definition 2.17. The set of *rooted trees* T is recursively defined as follows:

- the graph $\bullet$ with only one node (root) belongs to T;
- if $\tau_1, \ldots, \tau_m \in T$, then the graph obtained by grafting the roots of $\tau_1, \ldots, \tau_m$ to a new node also belongs to T. It is denoted by $\tau = [\tau_1, \ldots, \tau_m]$, and the new node is the root of τ.

Definition 2.18. The *order* of a tree τ, denoted $|\tau|$, is given its total number of nodes.

Definition 2.19. The *children* of a tree $\tau = [\tau_1, \ldots, \tau_m]$ are given by $\{\emptyset, \tau_1, \ldots, \tau_m\}$.

The recursive definition of a tree introduces some redundancy in the construction of T. For example, let $\tau_1, \tau_2 \in T$ be two distinct trees, then $[\tau_1, \tau_2]$ and $[\tau_2, \tau_1]$ would also be classified as distinct trees. However, in the applications we consider, the ordering of the children is not important. Thus, we introduce the following concept of equivalence among trees.

Definition 2.20. Two trees are said to be *equivalent* if they share the same children.

Another concept closely related to equivalence is the symmetry of a tree, which describes the total number of permutations of all children (including the children's children, and so on) such that the tree is left unchanged. For example, suppose that $\tau_1 = \tau_2 = \bullet$, $\tau_3 = [\bullet]$, then $\tau = [\tau_1, \tau_2, \tau_3] = [\bullet, \bullet, [\bullet]] = [\tau_2, \tau_1, \tau_3]$.

Definition 2.21. The symmetry coefficients $\sigma : T \to \mathbb{R}$ are recursively defined by $\sigma(\bullet) = 1$ and, for $\tau = [\tau_1, \ldots, \tau_m]$,

$$\sigma(\tau) = \sigma(\tau_1) \cdots \sigma(\tau_m) \cdot \mu_1! \mu_2! \cdots \mu_l!, \qquad l \leq m,$$

where the integers $\mu_1, \mu_2, \ldots, \mu_l$ count equivalent trees among $\tau_1, \ldots, \tau_m$.

The primary application for the set of rooted trees is in distinguishing between the various derivatives of f. These are called elementary differentials and are defined as follows:

Definition 2.22. For a given tree $\tau \in T$, the *elementary differential* is a mapping $F(\tau) : X \to X$, defined recursively by

$$\begin{aligned} F(\bullet)(y) &= f(y), \\ F(\tau)(y) &= f^{(m)}(y)(F(\tau_1)(y), \ldots, F(\tau_m)(y)), \qquad \text{for} \qquad \tau = [\tau_1, \ldots, \tau_m]. \end{aligned}$$

Here, notice that the $f^{(m)}(y)$ are m-linear operators. Thus, the ordering of the terms $F(\tau_1)(y), \ldots, F(\tau_m)(y)$ does not affect the computed output. It is for this reason that we say the ordering of the children of a tree is not important.

B-series: Using the definitions given above, we formally define a B-series as follows:

Definition 2.23. For a mapping $a : T \cup \{\emptyset\} \to \mathbb{C}$, a formal series of the form

$$B(a, y) = a(\emptyset)y + \sum_{t \in T} \frac{h^{|\tau|}}{\sigma(\tau)} a(\tau) F(\tau)(y),$$

is called a *B-series*.

$\lvert\tau\rvert$	τ-string	τ	$F(\tau)(y)$	$\sigma(\tau)$	$\gamma(\tau)$
1	$\bullet$		$f(y)$	1	1
2	$[\,\bullet\,]$		$f'(y)f(y)$	1	2
3	$[\,\bullet\,,\bullet\,]$		$f''(y)(f(y),f(y))$	2	3
3	$[[\,\bullet\,]]$		$f'(y)f'(y)f(y)$	1	6
4	$[\,\bullet\,,\bullet\,,\bullet\,]$		$f'''(y)(f(y),f(y),f(y))$	6	4
4	$[[\,\bullet\,],\bullet\,]$		$f''(y)(f'(y)f(y),f(y))$	1	8
4	$[[\,\bullet\,,\bullet\,]]$		$f'(y)f''(y)(f(y),f(y))$	2	12
4	$[[[\,\bullet\,]]]$		$f'(y)f'(y)f'(y)f(y)$	1	24

Table 2.1: B-series trees, graphs, elementary differentials and coefficients.

Example 2.24. Let the mapping $\gamma : T \cup \{\emptyset\} \to \mathbb{R}$ be defined by $\gamma(\emptyset) = 1, \gamma(\bullet) = 1$, and for $\tau = [\tau_1, \ldots, \tau_m]$,

$$\gamma(\tau) = |\tau|\gamma(\tau_1)\cdots\gamma(\tau_m).$$

Then, the B-series given by $B(1/\gamma, y)$ describes the Taylor series expansion of $y(t+h)$.

The functions $\gamma(\tau)$ and $\sigma(\tau)$ are important for identifying multiple occurrences of elementary differentials. For example, the B-series for the fourth order derivative of $y(t)$ contains the term $3f''(f'f, f)$. This elementary differential is associated with the tree $\tau = [[\bullet], \bullet]$, and its coefficient is computed using the formula (see [36, p. 58]):

$$\frac{|\tau|!}{\sigma(\tau)\gamma(\tau)} = \frac{|[[\bullet],\bullet]|!}{\sigma\left([[\bullet],\bullet]\right)\gamma\left([[\bullet],\bullet]\right)} = \frac{4!}{1\cdot 8} = 3.$$

$\diamond$

For trees of order $|\tau| \leq 4$, the graphs, elementary differentials and coefficients are given in Table 2.1. Further details on B-series can be found in Chapters 3 and 4.

2.3 Underlying one-step method

Questions regarding the long-time behaviour of multivalue methods are often tackled through study of a closely connected one-step method (OSM), referred to as the *underlying one-step method* (UOSM).

Definition 2.25 (**Underlying one-step method**)**.** The map $\Phi_h : X \to X$ is called an *underlying one-step method* (UOSM) of a GLM if

$$\mathcal{M}_h \circ \mathcal{S}_h^*(y_0) = \mathcal{S}_h^* \circ \Phi_h(y_0), \qquad \forall\ y_0 \in X. \tag{2.12}$$

for some consistent, *ideal starting method* $\mathcal{S}_h^* : X \to X^r$.

Remark 2.26. In general, it is understood that Φ_h and $\mathcal{S}_h^*$ hold formally as B-series which may be truncated at some large power of h, where terms are smaller than machine precision, in a similar way to (possibly divergent) asymptotic series.

A result formulated by Kirchgraber [42] states that all strictly-stable LMMs possess a UOSM. This result was generalised to strictly-stable GLMs by Stoffer [57], where it was also shown these GLMs possess the property of asymptotic phase, that is,

$$||y_n - y_n^*|| \leq \text{Const} \cdot \rho^n, \qquad \forall\ n \geq 0,$$

where $\rho \in (0, 1)$, y_n are given by the numerical method as a whole and $y_n^* := \Phi_h^n(y_0)$. This property explains why strictly-stable GLMs have the same long-time behaviour as RKMs, and other OSMs.

For stable GLMs in general, we cannot guarantee that the property of asymptotic phase holds. However, the formal existence of a UOSM may be proved.

Theorem 2.27 ([36, pp. 610-611])**.** *Let $\mathcal{M}_h$ be a consistent GLM with V possessing a simple eigenvalue $\zeta_1 = 1$, $\mathcal{F}_h$ a consistent finishing method, and let u, w be as in the definition of preconsistency. Then, there exists a unique formal one-step method*

$$\Phi_h(y_0) = y_0 + h\phi_1(y_0) + h^2\phi_2(y_0) + \ldots,$$

where each $h^i\phi_i : X \to X$, $i \geq 1$ is a B-series; and a unique formal starting method

$$\mathcal{S}_h^*(y_0) = uy_0 + h\mathcal{S}_1(y_0) + h^2\mathcal{S}_2(y_0) + \ldots, \tag{2.13}$$

where each $h^i\mathcal{S}_i : X \to X^r$, $i \geq 1$, is an r-dimensional vector of B-series, such that

$$\mathcal{M}_h \circ \mathcal{S}_h^*(y_0) = \mathcal{S}_h^* \circ \Phi_h(y_0),$$
$$\mathcal{F}_h \circ \mathcal{S}_h^*(y_0) = y_0,$$

hold as formal power series in h.

Proof. Expanding $\mathcal{S}_h^* \circ \Phi_h(y_0) - \mathcal{M}_h \circ \mathcal{S}_h^*(y_0) = 0$ in powers of h, and comparing quantities of $O(h^i)$ in the order $i = 1, 2, \ldots$, we find

$$\begin{aligned}
O(1): &\quad (I_r - V)uy_0 = 0,\\
O(h): &\quad (I_r - V)\mathcal{S}_1(y_0) + u\phi_1(y_0) = B\mathbb{1}f(y_0),\\
O(h^2): &\quad (I_r - V)\mathcal{S}_2(y_0) + u\phi_2(y_0) = BA\mathbb{1}f'(y_0)f(y_0) +\\
&\qquad\qquad (BU \otimes f'(y_0))\mathcal{S}_1(y_0) - \mathcal{S}_1'(y_0)\phi_1(y_0),\\
&\qquad\qquad \vdots
\end{aligned}$$

Similarly, we expand $\mathcal{F}_h \circ \mathcal{S}_h^*(y_0) - y_0 = 0$ in powers of h:

$$\begin{aligned}
O(1): &\quad (w^{\mathsf{H}}u - 1)y_0 = 0,\\
O(h): &\quad w^{\mathsf{H}}\mathcal{S}_1(y_0) = w^{\mathsf{H}}B_S\mathbb{1}_S f(y_0),\\
O(h^2): &\quad w^{\mathsf{H}}\mathcal{S}_2(y_0) = w^{\mathsf{H}}B_S(A_S - U_F B_S)\mathbb{1}_S f'(y_0)f(y_0) +\\
&\qquad\qquad (w^{\mathsf{H}}B_S U_F \otimes f'(y_0))\mathcal{S}_1(y_0),\\
&\qquad\qquad \vdots
\end{aligned}$$

In general, a comparison of terms of $O(h^i)$ will lead to a system of equations of the form

$$\begin{bmatrix} I - V & u \\ w^{\mathsf{H}} & 0 \end{bmatrix} \begin{bmatrix} \mathcal{S}_i(y_0) \\ \phi_i(y_0) \end{bmatrix} = \begin{bmatrix} G_i(y_0) \\ g_i(y_0) \end{bmatrix}, \qquad i \geq 1,$$

where the RHS terms $G_i(y_0)$, $g_i(y_0)$ depend on the known functions $\mathcal{S}_j(y_0), \phi_j(y_0)$ for $j < i$. Now, since ζ_1 is a simple eigenvalue of V and $w^{\mathsf{H}}u = 1$, the LHS matrix is invertible (by the ABCD Lemma [55]). Thus, we may uniquely determine each $\mathcal{S}_i(y_0)$ and $\phi_i(y_0)$, for $i \geq 1$. □

Note, that it now follows from the definition of the UOSM and Theorem 2.27 that

$$\Phi_h(y_0) = \mathcal{F}_h \circ \mathcal{M}_h \circ \mathcal{S}_h^*(y_0).$$

Other connections can also be made between Φ_h and the flow map φ_h, as well as $\mathcal{S}_h^*$ and the practical starting method $\mathcal{S}_h$. This is explained in the following corollary.

Corollary 2.28. *Suppose the pair $(\mathcal{M}_h, \mathcal{S}_h)$ is of order p. Then,*

$$\begin{aligned}
\mathcal{S}_h^*(y_0) &= \mathcal{S}_h(y_0) + O(h^{p+1}),\\
\Phi_h(y_0) &= \varphi_h(y_0) + O(h^{p+1}).
\end{aligned}$$

Proof. Consider an arbitrary truncation of the ideal starting method and UOSM:

$$\widetilde{\mathcal{S}_h}(y_0) = \mathcal{S}_h^*(y_0) + O(h^N), \qquad \widetilde{\Phi_h}(y_0) = \Phi_h(y_0) + O(h^N),$$

for some integer $N \geq 2$. Then, the pair $(\widetilde{\mathcal{S}_h}, \widetilde{\Phi}_h)$ uniquely satisfy (2.12) and $\mathcal{F}_h \circ \widetilde{\mathcal{S}_h}(y_0) = y_0$, up to terms of $O(h^N)$. Now, fixing $N = p + 1$ and recalling the definition of GLM order (2.10), we note that the pair $(\mathcal{S}_h, \varphi_h)$ also satisfy (2.12) and $\mathcal{F}_h \circ \mathcal{S}_h(y_0) = y_0$, up to terms $O(h^{p+1})$. The result now follows from uniqueness of $\mathcal{S}_h^*$ and Φ_h.

□

Example 2.29. Consider solving the linear test equation

$$\frac{\mathrm{d}y}{\mathrm{d}t} = \lambda y, \qquad y(0) = 1, \qquad \lambda \in \mathbb{C}, \tag{2.14}$$

with the Leapfrog method (2.4), initialised using the Euler starting method

$$y^{[0]} = \begin{bmatrix} y_0 \\ y_0 + hf(y_0) \end{bmatrix}.$$

Note here that finishing method is given by the first component.

Defining $z := h\lambda$, we may write the update step as follows

$$y^{[n+1]} = \begin{bmatrix} y_1^{[n+1]} \\ y_2^{[n+1]} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 2z \end{bmatrix} \begin{bmatrix} y_1^{[n]} \\ y_2^{[n]} \end{bmatrix} =: M(z) y^{[n]}.$$

Here, we observe that the problem of determining the UOSM and ideal starting method is equivalent to finding the eigenvalue-eigenvector pairs of $M(z)$ i.e. find $S(z) \in \mathbb{C}^2$ and $\phi(z) \in \mathbb{C}$ such that

$$M(z)S(z) = S(z)\phi(z), \qquad e_1^T S(z) = S_1(z) = 1.$$

The solutions to this problem are given below

$$\phi_1(z) = z + \sqrt{1+z^2}, \qquad S_1(z) = \begin{bmatrix} 1 \\ z + \sqrt{1+z^2} \end{bmatrix},$$

$$\phi_2(z) = z - \sqrt{1+z^2}, \qquad S_2(z) = \begin{bmatrix} 1 \\ z - \sqrt{1+z^2} \end{bmatrix}.$$

Notice for $z = 0$, we have that $S_1(0) = [1, 1]^T$ and $S_2(0) = [1, -1]^T$. Since $S_2(0)$ does not correspond to the preconsistency vector $u = [1, 1]^T$, we conclude that $(S(z), \phi(z)) = (S_1(z), \phi_1(z))$.

◇

2.4 Parasitism

Parasitism is a phenomenon that generally occurs in multivalue methods. It describes the unacceptable growth of perturbations made to the non-principal components of the method (i.e. those not associated with the $\zeta_1 = 1$ eigenvalue of V), which inevitably leads to the corruption of the numerical solution. In contrast, a one-step method will not suffer from parasitism as there is only a single (principal) component approximating the solution.

Example 2.30. Consider the following result due to Hairer: The simple pendulum problem, formulated as a first-order differential system, is given by

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} p(t) \\ q(t) \end{bmatrix} = \begin{bmatrix} -\sin(q(t)) \\ p(t) \end{bmatrix}, \qquad \begin{bmatrix} p(0) \\ q(0) \end{bmatrix} = \begin{bmatrix} p_0 \\ q_0 \end{bmatrix}, \qquad t \in [0, T]. \tag{2.15}$$

This problem is Hamiltonian, with

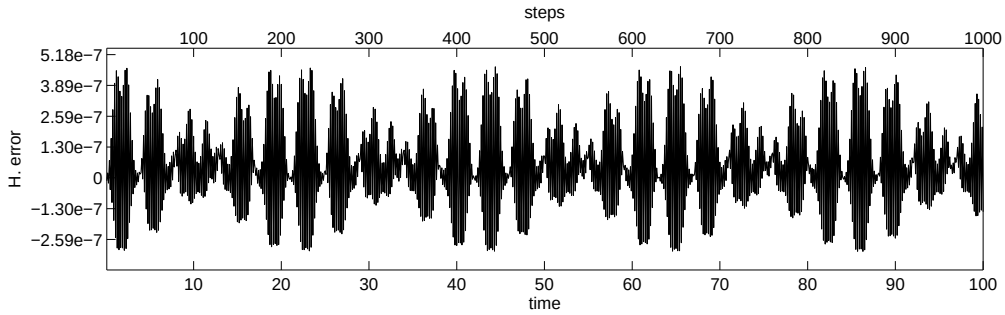
$$H(p, q) = \frac{1}{2}p^2 - \cos(q).$$

For $T = 100$ and $(p_0, q_0) = (0, 1.3)$, we solve this problem using GLM (2.8) with a time-step $h = 0.1$. Figure 2-1a displays the error made in the Hamiltonian at every step of the integration. Here, the error is small and remains bounded over the interval, indicating no parasitic growth. However, if we repeat the experiment with initial data $(p_0, q_0) = (0, 2.3)$, the influence of parasitism is obvious, as is illustrated in Figure 2-1b.

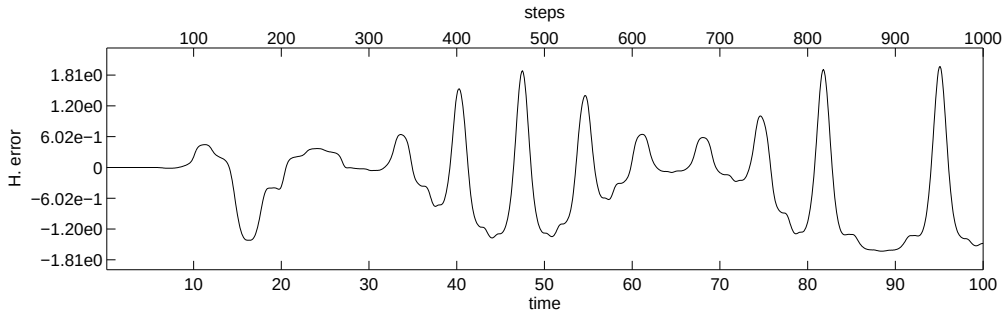
◇

An introduction to parasitism analysis: Consider a stable and consistent GLM applied to the linear test equation (2.14). Furthermore, assume that the eigenvalues of V are unimodular and distinct. For $z := h\lambda$ taken to be small, the update equation can be written as

$$y^{[n+1]} = M(z)y^{[n]}, \qquad \text{where} \qquad M(z) = V + zB(I_s - zA)^{-1}U.$$



(a)



(b)

Figure 2-1: Approximate Hamiltonian preservation in the simple pendulum problem over $[0, 100]$. Numerical solutions are obtained using GLM (2.8) with a time-step $h = 0.1$. Initial data is taken to be (a) $(p_0, q_0) = (0, 1.3)$. (b) $(p_0, q_0) = (0, 2.3)$.

Denote the eigenvalues of $M(z)$ by $\phi_i(z)$, for $i = 1, \ldots, r$. Then, for parasitism-free behaviour, we require the non-principal eigenvalues to satisfy $|\phi_i(z)| \leq 1$ for $i = 2, \ldots, r$.

If we consider expansions of the non-principal eigenvalues, it is possible to derive necessary algebraic conditions for parasitism-free behaviour: Let $F_i(z)$ and $S_i(z)$ respectively denote the left and right eigenvectors of $M(z)$ associated with eigenvalue $\phi_i(z)$, and normalised such that $F_i(z)S_i(z) = 1$, for $i = 2, \ldots, r$. Then, we have that

$$\phi_i(z) = F_i(z)M(z)S_i(z), \qquad \text{where} \qquad F_i(z)S_i(z) = 1,$$

for $i = 2, \ldots, r$. Taking expansions about $z = 0$, we find

$$\begin{aligned}\phi_i(z) &= \phi_i(0) + z\left(F_i'(0)M(0)S_i(0) + F_i(0)M'(0)S_i(0) + F_i(0)M(0)S_i'(0)\right) + O(z^2),\\ &= \zeta_i + z\left(F_i'(0)S_i(0) + F_i(0)BUS_i(0) + F_i(0)S_i'(0)\right) + O(z^2),\end{aligned}$$

where $'$ denotes differentiation with respect to z, and

$$\begin{aligned} & & F_i(z)S_i(z) &= 1, \\ \Rightarrow \quad & F_i(0)S_i(0) + z\left(F_i'(0)S_i(0) + F_i(0)S_i'(0)\right) &= 1 + O(z^2), \\ & \Rightarrow & F_i'(0)S_i(0) &= -F_i(0)S_i'(0). \end{aligned}$$

It now follows that $|\phi_i(z)| \leq |\zeta_i + zF_i(0)BUS_i(0)| + O(|z|^2)$. Thus, to eliminate first-order parasitic effects, we arrive at the following condition:

Definition 2.31 ([13])**.** A GLM is first-order *parasitism-free* if

$$w_i^{\mathsf{H}} BUu_i = 0, \qquad i = 2, \ldots, r, \tag{2.16}$$

where w_i, u_i respectively denote the left and right eigenvectors of V associated with eigenvalues ζ_i.

Example 2.32. Let us again consider GLM (2.8). The left and right eigenvectors of V associated with the non-principal eigenvalue $\zeta_2 = -1$ are given by $u = w = [0, 1]^T$. Checking condition (2.16), we find

$$w_2^{\mathsf{H}} BUu_2 = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} 1 & -\frac{3+2\sqrt{3}}{3} \\ 1 & \frac{3+2\sqrt{3}}{3} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = -\frac{3+2\sqrt{3}}{3} \neq 0.$$

As the GLM fails to eliminate first-order parasitic effects, it will be particularly susceptible to parasitism whenever the eigenvalues of the Jacobian possess non-zero, positive real parts. For Hamiltonian systems, the eigenvalues of the Jacobian come in plus-minus pairs. Thus, they need only have non-zero real parts for parasitism to arise. Such is the case in Example 2.30, for initial data $(p_0, q_0) = (0, 2.3)$, where for part of the periodic orbit, the eigenvalues of the Jacobian are real pairs.

$\diamond$

Later in Chapter 3, we develop an *a priori* parasitism theory that considers general vector fields $f(y)$. This is used to derive higher-order parasitism-free conditions.

2.5 Symmetry

Consider an IVP of the form (2.1), and assume that it is *reversible* [56], i.e. there exists a matrix $R \in \mathbb{R}^{d\times d}$ such that $f(Ry) + Rf(y) = 0$. Then, the flow map φ_t associated with the solution $y(t)$ is known to possess the property of *time-symmetry*.

Definition 2.33. A forward-time evolution $\varphi_t : X \to X$ is *symmetric* if

$$\varphi_t(y_0) = \varphi_{-t}^{-1}(y_0), \qquad y_0 \in X. \tag{2.17}$$

Fundamental properties such as symmetry are often considered in the design of numerical methods. For OSMs, the definition for symmetry follows directly from (2.17).

Definition 2.34. Consider the one-step method $\mathcal{R}_h : X \to X$ and its adjoint method $\mathcal{R}_h^* : X \to X$ defined such that $\mathcal{R}_h^* := \mathcal{R}_{-h}^{-1}$. Then, $\mathcal{R}_h$ is *symmetric* if

$$\mathcal{R}_h(y_0) = \mathcal{R}_h^*(y_0), \qquad y_0 \in X.$$

For GLMs, the action of computing the adjoint often rearranges the inputs of the method (such is the case with the Leapfrog method). This incompatibility leads us to define GLM-symmetry in terms of equivalence to the adjoint.

Definition 2.35. Consider the GLM $\mathcal{M}_h$ and its adjoint method $\mathcal{M}_h^* : X^r \to X^r$ defined such that $\mathcal{M}_h^* := \mathcal{M}_{-h}^{-1}$. Then, $\mathcal{M}_h$ is *symmetric* if there exists an involution matrix $L \in \mathbb{C}^{r\times r}$ such that

$$\mathcal{M}_h(y) = L\mathcal{M}_h^*(Ly), \qquad y \in X^r. \tag{2.18}$$

The requirement for L to be an involution arises from two applications of (2.18), i.e. defining the *symmetric adjoint* to be the map $\mathcal{M}_h^\dagger : X^r \to X^r$, $\mathcal{M}_h^\dagger := L\mathcal{M}_h^* \circ L$, then we need L such that

$$(\mathcal{M}_h^\dagger)^\dagger = (L{\mathcal{M}_h}^* \circ L)^\dagger = L(L{\mathcal{M}_h}^* \circ L)^* L = L^2 \mathcal{M}_h^{**} \circ L^2 = \mathcal{M}_h.$$

2.5.1 Algebraic conditions for symmetry

In order to determine symmetry conditions on the coefficient matrices of a GLM, we first require expressions for the inverse method $\mathcal{M}_h^{-1} : X^r \to X^r$, and the adjoint method $\mathcal{M}_h^* := \mathcal{M}_{-h}^{-1}$:

Let $y^{[n]} \mapsto \mathcal{M}_h^{-1}(y^{[n]})$ in the stage and update equations (2.3), i.e.

$$\begin{bmatrix} Y \\ y^{[n]} \end{bmatrix} = \begin{bmatrix} A & U \\ B & V \end{bmatrix} \begin{bmatrix} hF(Y) \\ \mathcal{M}_h^{-1}(y^{[n]}) \end{bmatrix},$$

where Kronecker products have been applied implicitly. Then, solving for $\mathcal{M}_h^{-1}(y^{[n]})$ yields

$$\begin{bmatrix} Y \\ \mathcal{M}_h^{-1}(y^{[n]}) \end{bmatrix} = \begin{bmatrix} A - UV^{-1}B & UV^{-1} \\ -V^{-1}B & V^{-1} \end{bmatrix} \begin{bmatrix} hF(Y) \\ y^{[n]} \end{bmatrix}.$$

Thus, for methods where V is invertible, the corresponding inverse method is described by the tableau

$$\left[\begin{array}{c|c} A - UV^{-1}B & UV^{-1} \\ \hline -V^{-1}B & V^{-1} \end{array}\right], \tag{2.19}$$

and the corresponding adjoint method is given by reversing the sign of h, i.e.

$$\left[\begin{array}{c|c} UV^{-1}B - A & UV^{-1} \\ \hline V^{-1}B & V^{-1} \end{array}\right]. \tag{2.20}$$

Theorem 2.36 ([16])**.** *A GLM is symmetric if there exist an involution $L \in \mathbb{C}^{r\times r}$ and an $s \times s$ symmetric permutation matrix P such that*

$$\left[\begin{array}{c|c} A & U \\ \hline B & V \end{array}\right] = \left[\begin{array}{c|c} P(UV^{-1}B - A)P & PUV^{-1}L \\ \hline LV^{-1}BP & LV^{-1}L \end{array}\right]. \tag{2.21}$$

Proof. The result follows from a comparison of the tableau for $\mathcal{M}_h$ and the (L, P)-equivalent tableau of the adjoint method (2.20). □

Similar algebraic conditions for symmetric GLMs are given in [36, pp. 612–614].

Example 2.37. Recall the coefficient matrices of the Leapfrog method (2.4):

$$A = 0, \qquad U = \begin{bmatrix} 0 & 1 \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ 2 \end{bmatrix}, \qquad V = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

If we choose

$$L = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \qquad P = 1,$$

then we can see that the method is symmetric:

$$\begin{aligned} A - P(UV^{-1}B - A)P &= 0 - \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \end{bmatrix} + 0 = 0, \\ U - PUV^{-1}L &= \begin{bmatrix} 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \end{bmatrix}, \\ B - LV^{-1}BP &= \begin{bmatrix} 0 \\ 2 \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\ V - LV^{-1}L &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}. \end{aligned}$$

It can also be verified that GLM (2.8) is symmetric for the choice

$$L = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad P = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

◇

Example 2.38. From symmetry conditions (2.21), we can derive the corresponding conditions for RKMs. Since $r = 1$, we have that $V = L = 1$ and $U = \mathbb{1}$. Thus, the symmetry conditions reduce to

$$\mathbb{1}B - PAP = A, \qquad BP = B.$$

These agree with the standard conditions for RKMs (see e.g. [36, p. 147]), where P is typically taken to be the time-reversal permutation.

◇

2.5.2 Symmetric starting and finishing methods

Assuming that the adjoint method exists, it is important to note that it uses slightly different starting and finishing methods, namely, $\mathcal{S}_{-h}$ and $\mathcal{F}_{-h}$. With these, we ensure that the order of the adjoint method matches that of the original.

Lemma 2.39 ([39])**.** *If the pair $(\mathcal{M}_h, \mathcal{S}_h)$ is of order p, then the pair $(\mathcal{M}_h^*, \mathcal{S}_{-h})$ satisfies*

$$\mathcal{M}_h^* \circ \mathcal{S}_{-h}(y_0) = \mathcal{S}_{-h} \circ \varphi_h(y_0) + (-1)^p V^{-1} C(y_0) h^{p+1} + O(h^{p+2}),$$

where $C(y_0)$ is as given in the definition of GLM order.

Proof. Since $\mathcal{M}_h^* \circ \mathcal{M}_{-h}(y^{[n]}) = y^{[n]}$, we write

$$\mathcal{M}_h^* \circ \mathcal{M}_{-h} \circ \mathcal{S}_{-h} \circ \varphi_h(y_0) = \mathcal{S}_{-h} \circ \varphi_h(y_0).$$

From the definition of GLM order (2.10), it follows

$$\mathcal{M}_h^* \circ \left[\mathcal{S}_{-h}(y_0) + (-1)^{p+1} C(\varphi_h(y_0)) h^{p+1} + O(h^{p+2}) \right] = \mathcal{S}_{-h} \circ \varphi_h(y_0).$$

Using $(\mathcal{M}_h^*(y))' z = V^{-1} z + O(h||z||)$ and $C(\varphi_h(y_0)) = C(y_0) + O(h)$ we obtain

$$\mathcal{M}_h^* \circ \mathcal{S}_{-h}(y_0) + (-1)^{p+1} V^{-1} C(y_0) h^{p+1} + O(h^{p+2}) = \mathcal{S}_{-h} \circ \varphi_h(y_0).$$

Re-arranging gives $\mathcal{M}_h^* \circ \mathcal{S}_{-h}(y_0) = \mathcal{S}_{-h} \circ \varphi_h(y_0) + (-1)^p V^{-1} C(y_0) h^{p+1} + O(h^{p+2})$. □

Since symmetry is defined in terms of equivalence to the adjoint method, there must exist another set of starting and finishing methods, that can be applied to $\mathcal{M}_h$, that will also yield the same GLM order. This observation leads us to define symmetric starting and finishing methods.

Definition 2.40. A starting method of an (L, P)-symmetric GLM is *symmetric* if

$$\mathcal{S}_h(y_0) = L\mathcal{S}_{-h}(y_0), \tag{2.22}$$

and the corresponding finishing method is *symmetric* if

$$\mathcal{F}_h(y) = \mathcal{F}_{-h}(Ly). \tag{2.23}$$

Theorem 2.41 ([16])**.** *A starting method* (2.7) *is symmetric if there exists an* $\widetilde{s} \times \widetilde{s}$ *symmetric permutation matrix* P_S *such that*

$$A_S = -P_S A_S P_S, \qquad B_S = -LB_S P_S, \qquad Lu = u. \tag{2.24}$$

A finishing method (2.7) *is symmetric if the corresponding starting method is symmetric and*

$$U_F = P_S U_F L, \qquad w^{\mathsf{H}} L = w^{\mathsf{H}}. \tag{2.25}$$

Proof. The algebraic conditions follow from a comparison of the tableaux for $\mathcal{S}_h, \mathcal{F}_h$ to the permutation-equivalent tableaux of $L\mathcal{S}_{-h}, \mathcal{F}_{-h}\circ L$. In particular,

$$\left[\begin{array}{c|c} A_S & \mathbb{1}_S \\ \hline B_S & u \end{array}\right] = \left[\begin{array}{c|c} -P_S A_S P_S & \mathbb{1}_S \\ \hline -LB_S P_S & Lu \end{array}\right],$$
$$\left[\begin{array}{c|c} A_S - U_F B_S & U_F \\ \hline -w^{\mathsf{H}} B_S & w^{\mathsf{H}} \end{array}\right] = \left[\begin{array}{c|c} P_S(U_F B_S - A_S)P_S & P_S U_F L \\ \hline w^{\mathsf{H}} B_S P_S & w^{\mathsf{H}} L \end{array}\right].$$

□

2.5.3 Necessity of even order

For a symmetric GLM, with symmetric starting and finishing methods, we find that the numerical method as a whole is symmetric:

$$\mathcal{F}_h\circ\mathcal{M}_h^n\circ\mathcal{S}_h(y_0) = \mathcal{F}_h\circ(LL)\mathcal{M}_h^n\circ(LL)\mathcal{S}_h(y_0) = \mathcal{F}_{-h}\circ(\mathcal{M}_h^*)^n\circ\mathcal{S}_{-h}(y_0).$$

As a consequence, we can deduce that the method must be of even order.

Theorem 2.42 ([16])**.** *Consider a consistent, symmetric GLM with symmetric starting and finishing methods. Then, the method is of even order $p \in \mathbb{N}$. Furthermore, the error at time $T = nh$ of the numerical method as a whole has an expansion in even powers of h, i.e.*

$$\mathcal{F}_h \circ \mathcal{M}_h^n \circ \mathcal{S}_h(y_0) - \varphi_{nh}(y_0) = h^p c_{p+1}(y_0, nh) + h^{p+2} c_{p+3}(y_0, nh) + \dots,$$

where each $c_i(y_0, nh)$ is a constant depending on n, h and various derivatives of f evaluated at y_0.

Proof. Let $T \in \mathbb{R}\backslash\{0\}$ be fixed, and define

$$e(n) := \mathcal{F}_{T/n} \circ \mathcal{M}_{T/n}^n \circ \mathcal{S}_{T/n}(y_0) - \varphi_T(y_0).$$

Then, it follows from the symmetry of the numerical method as a whole that $e(n) = e(-n)$. Thus, $e(n)$ is an even function of n and the expansion

$$e(n) = (T/n)^p c_{p+1}(y_0, T) + (T/n)^{p+2} c_{p+3}(y_0, T) + \dots,$$

must contain only even powers of n. The result follows after setting $h = T/n$. □

2.5.4 Connection to the underlying one-step method

Let us now consider the UOSM of symmetric GLM, and in particular, address the question of whether or not symmetry of the GLM implies symmetry of the UOSM.

Theorem 2.43. *Consider an (L, P)-symmetric GLM such that V possesses a simple eigenvalue $\zeta_1 = 1$. If the corresponding finishing method is symmetric, then the UOSM and ideal starting method satisfy*

$$\mathcal{S}_h^*(y_0) = L\mathcal{S}_{-h}^*(y_0),$$
$$\Phi_h(y_0) = \Phi_{-h}^{-1}(y_0).$$

Proof. From Theorem 2.27, there exists a unique pair $(\mathcal{S}_h^*, \Phi_h)$ such that

$$\mathcal{M}_h \circ \mathcal{S}_h^*(y_0) = \mathcal{S}_h^* \circ \Phi_h(y_0), \qquad \text{and} \qquad \mathcal{F}_h \circ \mathcal{S}_h^*(y_0) = y_0,$$

hold formally as power series in h. Now, pre-multiplying through by $\mathcal{M}_h^{-1}$ in the UOSM condition, and letting $y_0 \mapsto \Phi_h^{-1}(y_0)$, $h \mapsto -h$, we find

$$\mathcal{M}_{-h}^{-1} \circ \mathcal{S}_{-h}^*(y_0) = \mathcal{S}_{-h}^* \circ \Phi_{-h}^{-1}(y_0).$$

Symmetry of the GLM then implies

$$\begin{aligned} & L\mathcal{M}_h \circ L\mathcal{S}^*_{-h}(y_0) = \mathcal{S}^*_{-h} \circ \Phi^{-1}_{-h}(y_0), \\ \Rightarrow \qquad & \mathcal{M}_h \circ L\mathcal{S}^*_{-h}(y_0) = L\mathcal{S}^*_{-h} \circ \Phi^{-1}_{-h}(y_0), \end{aligned}$$

and symmetry of the finishing method implies

$$y_0 = \mathcal{F}_{-h} \circ \mathcal{S}^*_{-h}(y_0) = \mathcal{F}_{-h} \circ (LL)\mathcal{S}^*_{-h} = \mathcal{F}_h \circ L\mathcal{S}^*_{-h}(y_0).$$

Thus, we deduce that the pair $(L\mathcal{S}^*_{-h}, \Phi^{-1}_{-h})$ is also a UOSM solution. Uniqueness then implies that

$$\mathcal{S}^*_h(y_0) = L\mathcal{S}^*_{-h}(y_0), \qquad \text{and} \qquad \Phi_h(y_0) = \Phi^{-1}_{-h}(y_0),$$

as required.

□

2.5.5 Non-existence of explicit, parasitism-free methods

Now, we present a negative result concerning the non-existence of explicit, parasitism-free symmetric methods.

Theorem 2.44 ([16])**.** *Consider a consistent, symmetric GLM with V having distinct eigenvalues. Then, the method cannot be both first-order parasitism-free and explicit.*

Proof. Firstly we note that symmetry condition $V = LV^{-1}L$ implies that V is invertible, and distinct eigenvalues imply V is diagonalisable:

$$\operatorname{diag}(1, \zeta_2, \ldots, \zeta_r) = T^{-1}VT, \qquad T = [u_1 | u_2 | \cdots | u_r], \qquad T^{-1} = \left[\begin{array}{c} w_1^{\mathsf{H}} \\ \hline w_2^{\mathsf{H}} \\ \hline \vdots \\ \hline w_r^{\mathsf{H}} \end{array}\right],$$

where $u_1 = u$ is the preconsistency vector, and w_i, u_i, for $i = 1, \ldots, r$ are respectively the left and right eigenvectors of V corresponding to eigenvalues ζ_i. (Note that we do not necessarily have that $w_1 = w$).

Suppose that the method is first-order parasitism-free. It follows from consistency and the parasitism-free condition (2.16) that

$$w_i^{\mathsf{H}} B U u_i = \begin{cases} 1 & \text{for } i = 1, \\ 0 & \text{for } i = 2, \ldots, r. \end{cases}$$

Thus, letting Tr denote the trace of a matrix, we have

$$1 = \sum_{i=1}^{r} w_i^{\mathsf{H}} BU u_i \frac{1}{\zeta_i} = \mathrm{Tr}(T^{-1}BUT(T^{-1}V^{-1}T)) = \mathrm{Tr}(BUV^{-1}) = \mathrm{Tr}(UV^{-1}B).$$

Now, using symmetry condition $A + PAP = UV^{-1}B$, we obtain

$$1 = \mathrm{Tr}(UV^{-1}B) = \mathrm{Tr}(A + PAP) = 2\ \mathrm{Tr}(A),$$

and it follows that $\mathrm{Tr}(A) = \frac{1}{2}$.

Suppose now that the method is also explicit. Then, the stage matrix A must be strictly lower triangular (or strictly upper triangular after a permutation of the stages). Thus, we necessarily have $\mathrm{Tr}(A) = 0$ and we arrive at a contradiction. □

While we cannot have completely explicit, parasitism-free, symmetric GLMs, it is possible to construct methods that only have one implicit stage, e.g. consider the following 4th order GLM:

$$\left[\begin{array}{ccc|cc} 0 & 0 & 0 & 1 & 1 \\ \frac{1}{2} & \frac{1}{2} & 0 & 1 & -2 \\ \frac{3}{2} & \frac{1}{2} & 0 & 1 & -2 \\ \hline \frac{2}{3} & \frac{1}{6} & \frac{1}{6} & 1 & 0 \\ \frac{2}{3} & \frac{1}{6} & \frac{1}{6} & 0 & -1 \end{array}\right]. \tag{2.26}$$

This method is (L, P)-symmetric with

$$L = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \qquad P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

2.6 Symplecticity

Consider a Hamiltonian IVP of the form (2.2). Letting $y = [p^T, q^T]^T \in X$, we may alternatively express (2.2) as

$$\frac{\mathrm{d}y}{\mathrm{d}t} = J^{-1}\nabla H(y) =: f(y), \qquad y(0) = y_0, \qquad t \in [T, -T], \qquad J = \begin{bmatrix} 0 & I_d \\ -I_d & 0 \end{bmatrix},$$

where the Hamiltonian $H : X \to \mathbb{R}$ is assumed to be twice-differentiable. (Here, we note that $X = \mathbb{R}^{2d}$, $d \in \mathbb{N}$.)

In 1899, Poincaré [47] made the important discovery that the flow map φ_t of a Hamiltonian IVP is a *symplectic transformation*. That is, for a bilinear map of the

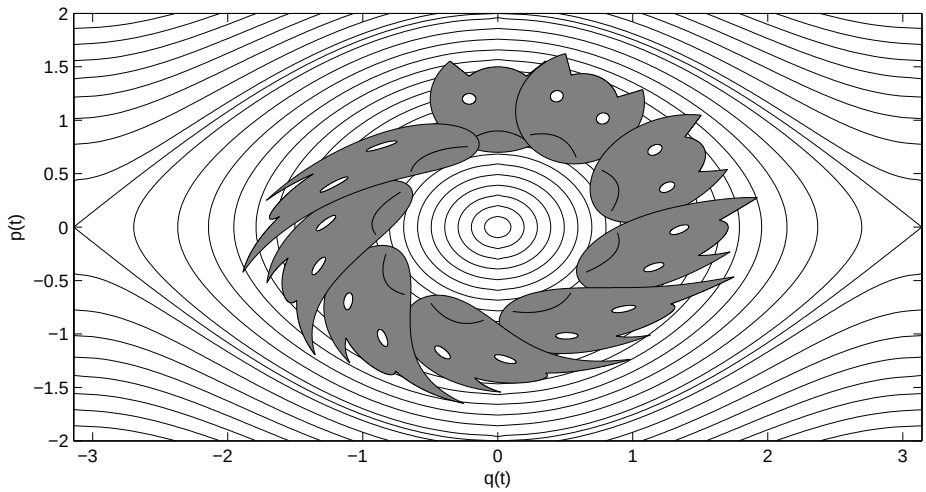


Figure 2-2: Demonstration of symplecticity of the flow map for the simple pendulum problem (2.15). (Here, the exact flow has been approximated using the symplectic Euler method with $h \ll 1$).

form

$$\omega(\xi, \eta) = \xi^T J \eta, \qquad \xi, \eta \in \mathbb{R}^{2d},$$

then

$$\omega(\varphi_t'(y_0)\xi, \varphi_t'(y_0)\eta) = \omega(\xi, \eta),$$

where $'$ here denotes differentiation with respect to y_0.

Definition 2.45 ([36, p. 183])**.** A differentiable map $g : U \to \mathbb{R}^{2d}$, where $U \subset \mathbb{R}^{2d}$ is an open set, is called *symplectic* if the Jacobian matrix $g'(y)$ is everywhere symplectic, i.e.

$$g'(y)^T J g'(y) = J, \qquad \text{or} \qquad \omega(g'(y)\xi, g'(y)\eta) = \omega(\xi, \eta).$$

Example 2.46. Consider the simple pendulum problem (2.15). Here, $d = 1$ and the bilinear map $\omega(\xi, \eta)$ represents the area of a parallelogram. Thus, symplecticity of the flow map implies that it is area-preserving. In Figure 2-2, we demonstrate this property by evolving a set of initial data forward in time. Here, the initial data takes the form of a cat's face and evolution occurs in the clockwise direction. While each face undergoes some distortion, the enclosed area remains constant.

◇

2.6.1 Symplectic numerical methods

As an introduction to symplectic numerical methods, we consider the class of OSMs.

Definition 2.47. A one-step method denoted by the map $\mathcal{R}_h : X \to X$ is called *symplectic* if

$$(\mathcal{R}_h'(y_0))^T J \mathcal{R}_h'(y_0) = J, \qquad y_0 \in X,$$

where $'$ denotes differentiation with respect to y_0.

Example 2.48. Consider the forward Euler method $\mathcal{R}_h(y_0) = y_0 + hf(y_0)$ applied to the simple pendulum problem (2.15). The method is non-symplectic since

$$\begin{aligned}(\mathcal{R}_h'(y_0))^T J \mathcal{R}_h'(y_0) &= J + h\left\{(f'(y_0))^T J + J f'(y_0)\right\} + h^2 (f'(y_0))^T J f'(y_0), \\ &= \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} + h\left\{ \begin{bmatrix} -1 & 0 \\ 0 & -\cos(q) \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & \cos(q) \end{bmatrix} \right\} \\ &\quad + h^2 \begin{bmatrix} 0 & \cos(q) \\ -\cos(q) & 0 \end{bmatrix}, \\ &= \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} + h^2 \begin{bmatrix} 0 & \cos(q) \\ -\cos(q) & 0 \end{bmatrix}.\end{aligned}$$

To demonstrate the non-symplecticity of the method, we again consider the evolution of a set of initial data in phase space, as was performed in Example 2.46. The result of this experiment, when using a time-step $h = \pi/12$, is given in Figure 2-3 and clearly shows each face increasing in size, indicating a lack of area-preservation.

◇

For multivalue methods, there exist various definitions for symplecticity. For example, we might call a method symplectic if its UOSM is also symplectic. This approach was considered by Tang [60] who has shown that the UOSM of a LMM (with inputs approximating the solution) cannot be symplectic. Closely related studies have been performed on GLMs [14],[36, p. 612], where it has been shown that methods can only be symplectic if they are equivalent to OSMs.

Alternatively, we can instead consider a generalisation of Definition 2.45 to X^r. This is known as *G-symplecticity*.

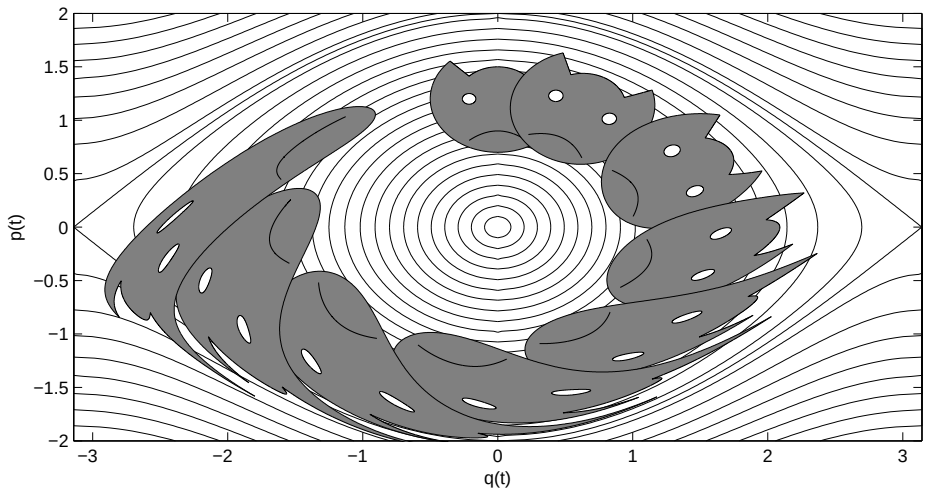


Figure 2-3: Demonstration of non-symplecticity of the Euler method for the simple pendulum problem (2.15).

Definition 2.49. A GLM denoted by the map $\mathcal{M}_h : X^r \to X^r$ is *G-symplectic* if there exists a symmetric, non-singular matrix $G \in \mathbb{C}^{r\times r}$ such that

$$(\mathcal{M}_h'(y^{[0]}))^{\intercal}(G \otimes J)\mathcal{M}_h'(y^{[0]}) = G \otimes J, \qquad y^{[0]} \in X^r,$$

where $'$ denotes differentiation with respect to $y^{[0]}$.

Interestingly, it can be shown that the UOSM of a G-symplectic GLM is *conjugate-symplectic* [24]. In other words, there exists, as a formal B-series, an invertible transformation $\chi_h : X \to X$ such that the composite map $\chi_h^{-1}\circ\Phi_h\circ\chi_h$ is symplectic. This result suggests that with the correct inputs to the method, i.e. $\mathcal{S}_h(y_0) \mapsto \mathcal{S}_h\circ\chi_h(y_0)$, the overall behaviour of the method will be symplectic. However, since χ_h is expressed as a formal series, a practical implementation of this map cannot be achieved.

Other notable results concerning G-symplecticity are that of Eirola and Sanz-Serna [26] where it was shown that every irreducible, symmetric LMM is also G-symplectic, and Hairer [34] where the UOSM of a G-symplectic LMMs was shown to be conjugate-symplectic.

2.6.2 Algebraic conditions for G-symplecticity

The approach we take here in deriving conditions for a G-symplectic GLM is based on the work of Burrage and Butcher 1979 [3], and will require the use of the following result.

Lemma 2.50. *Consider a Hamiltonian IVP where $H(y)$ is assumed to be twice differentiable. For all diagonal matrices $D \in \mathbb{R}^{s\times s}$, the following holds*

$$(D \otimes J)\nabla F(Y) = -(\nabla F(Y))^T (D \otimes J),$$

where $\nabla F(Y) = diag(f'(Y_1), \ldots, f'(Y_s))$ is a block-diagonal matrix with $Y_i \in X$, for $i = 1, \ldots, s$.

Proof. Since the Hessian of a scalar field is symmetric and $J^T = -J$, we find

$$Jf'(y) = JJ^{-1}\nabla^2 H(y) = \nabla^2 H(y) = (\nabla^2 H(y))^T J^{-T} J^T = -(f'(y))^T J.$$

Now, we observe

$$(D \otimes J)\nabla F(Y) = \begin{bmatrix} D_{11} J f'(Y_1) \\ \vdots \\ D_{ss} J f'(Y_s) \end{bmatrix} = \begin{bmatrix} -(f'(Y_1))^T (D_{11} J) \\ \vdots \\ -(f'(Y_s))^T (D_{ss} J) \end{bmatrix} = -(\nabla F)^T (Y)(D \otimes J).$$

□

Theorem 2.51. *Consider a GLM with coefficient matrices (A, U, B, V). The method is G-symplectic if there exists a symmetric, non-singular matrix $G \in \mathbb{C}^{r\times r}$ and a non-singular, diagonal matrix $D \in \mathbb{R}^{s\times s}$, such that*

$$DA + A^T D = B^{\mathsf{H}} G B, \tag{2.27}$$

$$DU = B^{\mathsf{H}} G V, \tag{2.28}$$

$$G = V^{\mathsf{H}} G V. \tag{2.29}$$

Proof. First note that the Fréchet derivative of $\mathcal{M}_h(y^{[0]})$ is found by differentiation with respect to $y^{[0]}$, i.e.

$$\mathcal{M}_h'(y^{[0]}) = V + hB\nabla F(Y)\frac{\partial Y}{\partial y^{[0]}},$$

$$\frac{\partial Y}{\partial y^{[0]}} = hA\nabla F(Y)\frac{\partial Y}{\partial y^{[0]}} + U\frac{\partial y^{[0]}}{\partial y^{[0]}},$$

where each coefficient matrix implicitly multiplies I_X in a Kronecker product.

Now, we observe that the product

$$(\mathcal{M}_h'(y^{[0]}))^{\mathsf{H}} (G \otimes J)\mathcal{M}_h'(y^{[0]}) = T_1 + hT_2 + h^2 T_3,$$

where

$$
\begin{aligned}
T_1 &= V^{\mathsf{H}}(G \otimes J)V, \\
T_2 &= \left(\frac{\partial Y}{\partial y^{[0]}}\right)^T (\nabla F(Y))^T B^{\mathsf{H}}(G \otimes J)V + V^{\mathsf{H}}(G \otimes J)B\nabla F(Y)\frac{\partial Y}{\partial y^{[0]}}, \\
T_3 &= \left(\frac{\partial Y}{\partial y^{[0]}}\right)^T (\nabla F(Y))^T B^{\mathsf{H}}(G \otimes J)B\nabla F(Y)\frac{\partial Y}{\partial y^{[0]}}.
\end{aligned}
$$

Using the properties of the Kronecker product and introducing $\frac{\partial y^{[0]}}{\partial y^{[0]}} = I_{X^r}$ into the expression for T_2, we find

$$
\begin{aligned}
T_1 &= V^{\mathsf{H}}GV \otimes J, \\
T_2 &= \left(\frac{\partial Y}{\partial y^{[0]}}\right)^T (\nabla F(Y))^T (B^{\mathsf{H}}GV \otimes J)\frac{\partial y^{[0]}}{\partial y^{[0]}} + \left(\frac{\partial y^{[0]}}{\partial y^{[0]}}\right)^T (V^{\mathsf{H}}GB \otimes J)\nabla F(Y)\frac{\partial Y}{\partial y^{[0]}}, \\
T_3 &= \left(\frac{\partial Y}{\partial y^{[0]}}\right)^T (\nabla F(Y))^T (B^{\mathsf{H}}GB \otimes J)\nabla F(Y)\frac{\partial Y}{\partial y^{[0]}},
\end{aligned}
$$

and after applying the G-symplectic conditions we obtain

$$
\begin{aligned}
T_1 &= G \otimes J, \\
T_2 &= \left(\frac{\partial Y}{\partial y^{[0]}}\right)^T (\nabla F(Y))^T (DU \otimes J)\frac{\partial y^{[0]}}{\partial y^{[0]}} + \left(\frac{\partial y^{[0]}}{\partial y^{[0]}}\right)^T (U^{\mathsf{H}}D \otimes J)\nabla F(Y)\frac{\partial Y}{\partial y^{[0]}}, \\
T_3 &= \left(\frac{\partial Y}{\partial y^{[0]}}\right)^T (\nabla F(Y))^T ((DA + A^T D) \otimes J)\nabla F(Y)\frac{\partial Y}{\partial y^{[0]}}.
\end{aligned}
$$

Now, in the expression for the Fréchet derivative, we multiply $\frac{\partial Y}{\partial y^{[0]}}$ by $(D \otimes J)$ and re-arrange for $(DU \otimes J)\frac{\partial y^{[0]}}{\partial y^{[0]}}$, to deduce that

$$
T_2 = \left(\frac{\partial Y}{\partial y^{[0]}}\right)^T (\nabla F(Y))^T (D \otimes J)\frac{\partial Y}{\partial y^{[0]}} + \left(\frac{\partial Y}{\partial y^{[0]}}\right)^T (D \otimes J)\nabla F(Y)\frac{\partial Y}{\partial y^{[0]}} - hT_3.
$$

After an application of Lemma 2.50, it follows that $T_2 = -hT_3$. Thus,

$$
(\mathcal{M}_h'(y^{[0]}))^{\mathsf{H}}(G \otimes J)\mathcal{M}_h'(y^{[0]}) = T_1 = G \otimes J,
$$

and the method is G-symplectic by Definition 2.49. □

Example 2.52. Recall the coefficient matrices of the Leapfrog method (2.4):

$$A = 0, \qquad U = \begin{bmatrix} 0 & 1 \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ 2 \end{bmatrix}, \qquad V = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

If we choose

$$D = 2 \qquad G = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$

then we can see that the method is G-symplectic:

$$DA + A^T D - B^{\mathsf{H}} G B = 0 + 0 - \begin{bmatrix} 0 & 2 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \end{bmatrix} = 0,$$

$$DU - B^{\mathsf{H}} G V = \begin{bmatrix} 0 & 2 \end{bmatrix} - \begin{bmatrix} 0 & 2 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \end{bmatrix},$$

$$G - V^{\mathsf{H}} G V = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

It can also be verified that GLM (2.8) is G-symplectic for the choice

$$D = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix}, \qquad G = \begin{bmatrix} 1 & 0 \\ 0 & \frac{3+2\sqrt{3}}{3} \end{bmatrix}.$$

◇

Example 2.53. From the GLM G-symplectic conditions, we can derive the symplectic conditions for an RKM. Since $r = 1$, we have that $G = 1$, $V = 1$ and $U = \mathbb{1}$. It follows that $D = \operatorname{diag}(B)$, and we are then left to satisfy

$$\operatorname{diag}(B)A + A^T \operatorname{diag}(B) = B^T B.$$

This agrees with the standard symplectic condition (see e.g. [36, p. 192]).

◇

2.6.3 Non-existence of explicit, parasitism-free methods

As was the case with symmetric methods, we also have a non-existence result on explicit, parasitism-free, G-symplectic GLMs:

Theorem 2.54. *Consider a consistent, G-symplectic GLM with V having distinct eigenvalues. Then, the method cannot be both first-order parasitism-free and explicit.*

Proof. The proof is similar to that given for symmetric methods: Firstly we note that G-symplectic condition (2.29) implies that V is invertible. Now, we assume the method is first-order parasitism-free. As in the proof of Theorem 2.44, we find

$$1 = \mathrm{Tr}(UV^{-1}B).$$

Combining G-symplectic conditions (2.27) and (2.28), we obtain

$$1 = \mathrm{Tr}(UV^{-1}B) = \mathrm{Tr}(A + D^{-1}A^T D) = 2\mathrm{Tr}(A),$$

and it follows that $\mathrm{Tr}(A) = \frac{1}{2}$. Now, suppose the method is also explicit. Here, we must have $\mathrm{Tr}(A) = 0$, and we arrive at a contradiction. □

Similar again to symmetric methods, it is possible to construct G-symplectic GLMs that are parasitism-free and possess only a single implicit stage. For example, the following method given in [17] is G-symplectic, parasitism-free and 4th order:

$$\left[\begin{array}{ccc|cc} 0 & 0 & 0 & 1 & 1 \\ \frac{2}{3} & 0 & 0 & 1 & -1 \\ \frac{2}{5} & -\frac{3}{10} & \frac{1}{2} & 1 & -\frac{1}{5} \\ \hline \frac{1}{3} & -\frac{3}{8} & \frac{25}{24} & 1 & 0 \\ \frac{1}{3} & \frac{3}{8} & -\frac{5}{24} & 0 & -1 \end{array}\right].$$

Chapter 3

Theoretical toolkit

B-series analysis is a fundamental theoretical tool used to understand various properties of GLMs. For example, it is used to a determine a method's order, long-time stability behaviour, UOSM and ideal starting method (see e.g. [33, Ch. III]). It is also an important component in backward error analysis which has been used to study parasitism in multivalue methods [35, 23][36, Ch. XV].

In the first half of this chapter, we use ideas from B-series analysis to develop a new power series, that we call a *derivative B-series*, for the study of derivatives of GLMs, i.e. allowing for a series representation of $\mathcal{M}_h'(y)v$, $v \in X^r$. These objects naturally arise in perturbation analysis, and in the analysis of parasitism. Thus, in the second half of this chapter, we present a new *a priori* theory of parasitism as a complementary approach to backward error analysis. This yields a framework that is used to bound the parasitic components of a GLM, and to derive higher-order parasitism-free conditions on the coefficients of a method.

3.1 B-series and rooted trees

Earlier in section 2.2, we gave a basic introduction to B-series and the set of rooted trees. Here, we develop the material further by discussing some of the elementary and advanced operations that are performed on these objects.

3.1.1 Tree operations

Recall that the set of rooted trees T is defined recursively as follows:

$$\begin{aligned} &\text{let} && \tau = \bullet \in T, \\ &\text{then also} && \tau = [\tau_1, \ldots, \tau_m] \in T, \qquad \text{where} \qquad \tau_1, \ldots, \tau_m \in T. \end{aligned}$$

Definition 3.1 (**Butcher product**)**.** The *Butcher product* of two trees is defined as

$$\begin{aligned} &v \circ \emptyset = v, \\ &\bullet \circ v = [v], \\ &u \circ v = [u_1, \ldots, u_m, v], \qquad \text{where} \qquad u = [u_1, \ldots, u_m]. \end{aligned}$$

Definition 3.2 (**Pruning**)**.** The *pruning* operation is defined as

$$\begin{aligned} &u \setminus \emptyset = u, \\ &u \setminus u = \emptyset, \\ &u \setminus v = [u_1, \ldots, u_m], \qquad \text{where} \qquad u = [u_1, \ldots, u_m, v]. \end{aligned}$$

The Butcher product can be applied to any pair of rooted trees. Whereas, pruning of a tree u is only valid if there exists a $\tilde{u}$ such that $u = \tilde{u} \circ v$, i.e.

$$u \setminus v = (\tilde{u} \circ v) \setminus v = \tilde{u}.$$

As neither operation is associative nor commutative, expressions involving multiple product or pruning operations should be evaluated from left to right, e.g.

$$\begin{aligned} u \circ v_1 \circ v_2 \circ \cdots \circ v_m &= (((u \circ v_1) \circ v_2) \circ \cdots) \circ v_m, \\ u \setminus v_1 \setminus v_2 \setminus \cdots \setminus v_m &= (((u \setminus v_1) \setminus v_2) \setminus \cdots) \setminus v_m. \end{aligned}$$

Note here that, while the operations are not commutative, a permutation of the $v_1, \ldots, v_m$ in the above expressions does not affect the final tree. For example, let $u = \bullet$, $v_1 =$  and $v_2 = \bullet$. Then,

$u \circ v_1 \circ v_2 = \bullet \circ$  $\circ \bullet =$ $\circ \bullet =$  $=$  $\circ$ $= \bullet \circ \bullet \circ$  $= u \circ v_2 \circ v_1$.

See Table 3.1 for more examples of the Butcher pruning operations.

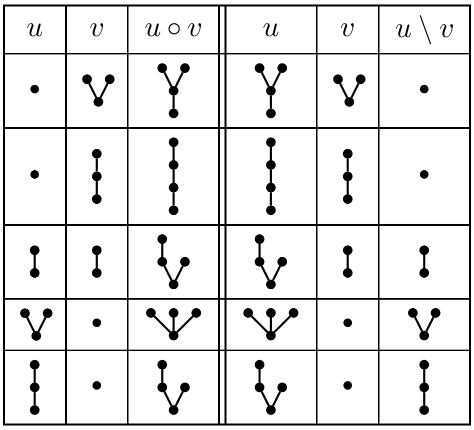


Table 3.1: Applications of the Butcher product and pruning on some trees of up to order 4.

3.1.2 B-series and its properties

Recall that a B-series, with coefficients given by $a : T \cup \{\emptyset\} \to \mathbb{C}$, is written as

$$B(a, y) = a(\emptyset)y + \sum_{\tau \in T} \frac{h^{|\tau|}}{\sigma(\tau)} a(\tau)F(\tau)(y),$$

where $F(\tau)$, $\sigma(\tau)$ respectively denote the elementary differential and symmetry coefficient corresponding to tree τ. As shown below, a B-series is linear in its first argument.

Lemma 3.3 (**Linearity**). *A B-series $B(a, y)$ is linear in its first argument.*

Proof. Let $c_1, c_2 \in \mathbb{R}$ and $a_1, a_2, a_3 : T \cup \{\emptyset\} \to \mathbb{C}$ where $a_3(\tau) := c_1 a_1(\tau) + c_2 a_2(\tau)$. Then,

$$\begin{aligned} B(a_3, y) &= a_3(\emptyset)y + \sum_{\tau \in T} \frac{h^{|\tau|}}{\sigma(\tau)} a_3(\tau)F(\tau)(y) \\ &= c_1 a_1(\emptyset) + c_2 a_2(\emptyset) + \sum_{\tau \in T} \frac{h^{|\tau|}}{\sigma(\tau)} (c_1 a_1(\tau) + c_2 a_2(\tau))F(\tau)(y) \\ &= c_1 a_1(\emptyset) + c_1 \sum_{\tau \in T} \frac{h^{|\tau|}}{\sigma(\tau)} a_1(\tau)F(\tau)(y) + c_2 a_2(\emptyset) + c_2 \sum_{\tau \in T} \frac{h^{|\tau|}}{\sigma(\tau)} a_2(\tau)F(\tau)(y) \\ &= c_1 B(a_1, y) + c_2 B(a_2, y). \end{aligned}$$

□

While the second argument is non-linear, a substitution of the form $B(a, B(b, y))$ will yield another B-series (see e.g. [36, pp. 61–64]). This is known as *composition* and is based on the following result:

Lemma 3.4 ([36, pp. 57–58]). *Let $a : T \cup \{\emptyset\} \to \mathbb{C}$ be a mapping satisfying $a(\emptyset) = 1$. Then,*

$$hf(B(a, y)) = B(a', y),$$

where $a'(\emptyset) = 0$, $a'(\bullet) = 1$ and

$$a'(\tau) = a(\tau_1) \cdots a(\tau_m), \qquad \tau = [\tau_1, \ldots, \tau_m].$$

Proof. It follows from $a(\emptyset) = 1$, that $B(a, y) = y + O(h)$. Thus, one can perform a Taylor series expansion of $hf(B(a, y))$ about y:

$$hf(B(a, y)) = h \sum_{m \geq 0} \frac{1}{m!} f^{(m)}(y)(B(a, y) - y)^m.$$

Now, since $f^{(m)}(y)$ is an m-linear map, we can express the term $f^{(m)}(y)(B(a, y) - y)^m$ as a sum of elementary differentials, i.e.

$$f^{(m)}(B(a, y) - y)^m = \sum_{\tau_1 \in T} \cdots \sum_{\tau_m \in T} \frac{h^{|\tau_1| + \cdots + |\tau_m|}}{\sigma(\tau_1) \cdots \sigma(\tau_m)} a(\tau_1) \cdots a(\tau_m) f^{(m)}(F(\tau_1), \ldots, F(\tau_m)),$$

where we have suppressed the y arguments. Using the definitions of the symmetry coefficients and the $a'(\tau)$, we simplify this expression to

$$f^{(m)}(B(a, y) - y)^m = \sum_{\tau_1 \in T} \cdots \sum_{\tau_m \in T} \frac{h^{|\tau| - 1}}{\sigma(\tau)} \mu_1! \cdots \mu_l! a'(\tau) F(\tau),$$

where $\tau = [\tau_1, \ldots, \tau_m]$ and the μ_j count equal trees among $\tau_1, \ldots, \tau_m$. Returning to the expansion, we find

$$hf(B(a, y)) = \sum_{m \geq 0} \sum_{\tau_1 \in T} \cdots \sum_{\tau_m \in T} \frac{h^{|\tau|}}{\sigma(\tau)} \frac{\mu_1! \cdots \mu_l!}{m!} a'(\tau) F(\tau)(y).$$

Finally, we note that there are $\binom{m}{\mu_1, \mu_2, \ldots, \mu_l}$ possibilities of expressing the tree τ in the form $\tau = [\tau_1, \ldots, \tau_m]$. Thus,

$$hf(B(a, y)) = \sum_{\tau \in T} \frac{h^{|\tau|}}{\sigma(\tau)} a'(\tau) F(\tau)(y) = B(a', y).$$

□

There is a formula available for computing the coefficients of a B-series composition (see [36, p.62]). However, in Chapter 4, we demonstrate that the above lemma is sufficient for the practical implementation. Thus, additional results on composition are regarded as unnecessary.

3.1.3 Extension to vector B-series

Since GLMs act on the product space X^r, we will find it convenient to work with B-series that also belong to this space. Consequently, a slight re-wording of the definition of B-series is required to fit into this situation:

Definition 3.5 (**Vector B-series**)**.** For a mapping $a : T \to \mathbb{C}^r$, a formal series of the form

$$B(a, y) = a(\emptyset)y + \sum_{\tau \in T} \frac{h^{|\tau|}}{\sigma(\tau)} a(\tau) \otimes F(\tau)(y),$$

is called a *vector B-series*.

Example 3.6. Recall from Section 2.1, that GLM (2.8) requires starting values of the form

$$y^{[0]} = \begin{bmatrix} y_0 \\ h^2 \frac{\sqrt{3}}{12} \frac{\mathrm{d}^2 y}{\mathrm{d}t^2}(0) - h^4 \left(\frac{\sqrt{3}}{108} \frac{\mathrm{d}^4 y}{\mathrm{d}t^4}(0) - \frac{9+5\sqrt{3}}{216} f'(y_0) \frac{\mathrm{d}^3 y}{\mathrm{d}t^3}(0) \right) \end{bmatrix}.$$

Before expressing this input as a vector B-series, we must first re-write the second component in terms of elementary differentials:

$$\frac{h^2\sqrt{3}}{12} F\left(\begin{smallmatrix} \bullet \\ | \\ \bullet \end{smallmatrix}\right) - \frac{h^4\sqrt{3}}{108} F\left(\begin{smallmatrix} & \bullet & \\ \bullet & & \bullet \end{smallmatrix}\right) - \frac{h^4\sqrt{3}}{36} F\left(\begin{smallmatrix} \bullet & \\ | & \bullet \\ \bullet & \end{smallmatrix}\right) + \frac{h^4(3+\sqrt{3})}{72} \left(F\left(\begin{smallmatrix} \bullet & \\ | & \bullet \\ \bullet & \end{smallmatrix}\right) + F\left(\begin{smallmatrix} \bullet \\ | \\ \bullet \\ | \\ \bullet \end{smallmatrix}\right) \right).$$

Now, with both components of $y^{[0]}$ taking the form of B-series, we see that the vector a-coefficients are given by

$$a(\emptyset) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad a\left(\begin{smallmatrix} \bullet \\ | \\ \bullet \end{smallmatrix}\right) = \begin{bmatrix} 0 \\ \frac{\sqrt{3}}{12} \end{bmatrix}, \qquad a\left(\begin{smallmatrix} & \bullet & \\ \bullet & & \bullet \end{smallmatrix}\right) = \begin{bmatrix} 0 \\ -\frac{\sqrt{3}}{18} \end{bmatrix},$$

$$a\left(\begin{smallmatrix} \bullet & \\ | & \bullet \\ \bullet & \end{smallmatrix}\right) = \begin{bmatrix} 0 \\ -\frac{\sqrt{3}}{36} \end{bmatrix}, \qquad a\left(\begin{smallmatrix} \bullet & \\ | & \bullet \\ \bullet & \end{smallmatrix}\right) = \begin{bmatrix} 0 \\ \frac{3+\sqrt{3}}{36} \end{bmatrix}, \qquad a\left(\begin{smallmatrix} \bullet \\ | \\ \bullet \\ | \\ \bullet \end{smallmatrix}\right) = \begin{bmatrix} 0 \\ \frac{3+\sqrt{3}}{72} \end{bmatrix}.$$

◇

3.2 Derivative B-series and derivative trees

In this section, we introduce a formal series that will be called a *derivative B-series*, or *DB-series* for short. As the name suggests, this series can describe the differentiated form of a B-series, e.g. the object $\nabla_y B(a, y) \cdot v$, for some $v \in X$, is a DB-series.

3.2.1 Derivative trees

Following the approach used in constructing B-series, we require each term of a DB-series to be associated with a tree. To avoid confusion with the set of rooted trees T, we introduce the set of *derivative trees* to be associated with DB-series.

Definition 3.7 (**Derivative trees**)**.** Consider a tree $u \in T \cup \{\emptyset\}$ written as $u = [u_1, \ldots, u_m]$. Then, the set of derivative trees corresponding to u are given by the set

$$D_\tau(u) := \bigcup_{v_1 \in \mathrm{ch}(u)} \bigcup_{v_2 \in \mathrm{ch}(v_1)} \cdots \bigcup_{v_k \in \mathrm{ch}(v_{k-1})} (u \backslash\, v_1) \circ ((v_1 \backslash\, v_2) \circ (\cdots (v_{k-1} \backslash\, v_k) \circ \times)),$$

where $\mathrm{ch}(u)$ denotes the children of u, i.e. $\mathrm{ch}(u) = \{\emptyset, u_1, \ldots, u_m\}$, and $k \in \mathbb{N}$ denotes the height of the root of u. The set of all derivative trees, DT, is then defined as

$$DT := \{D_\tau(u) | \forall\ u \in T \cup \{\emptyset\}\}.$$

Definition 3.8 (**Order**)**.** The *order* of a derivative tree $d\tau \in DT$, denoted $|d\tau|$, is given by the total number of $\bullet$ nodes.

Remark 3.9. The construction of $D_\tau(u)$ may also be performed recursively:

$$\begin{aligned} D_\tau(\emptyset) &:= \{\times\}, \\ D_\tau(\bullet) &:= \{[\times]\}, \\ D_\tau(u) &:= \{u \circ \times\} \quad \cup \bigcup_{du_1 \in D_\tau(u_1)} \{(u \setminus u_1) \circ du_1\} \quad \cup \cdots \cup \bigcup_{du_m \in D_\tau(u_m)} \{(u \setminus u_m) \circ du_m\}. \end{aligned}$$

Example 3.10. Consider computing the derivative trees associated with $u =$:

$$\begin{aligned} D_\tau(\text{}) &= \left\{(\text{} \setminus \emptyset) \circ \times\right\} \cup \left\{(\text{} \setminus \bullet) \circ ((\bullet \setminus \emptyset) \circ \times)\right\} \cup \left\{(\text{} \setminus \bullet) \circ ((\bullet \setminus \emptyset) \circ \times)\right\}, \\ &= \left\{\text{}\right\} \cup \left\{\text{}\right\}, \\ &= \left\{\text{}\right\}. \end{aligned}$$

◇

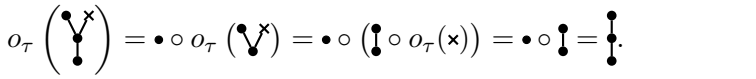
In less formal language, derivative trees are found by grafting a × node to a • node of a given tree τ. Repeating this procedure for all the • nodes of τ, and again for all $\tau \in T$, we obtain the complete set of derivative trees. Each tree will take the form

$$\begin{aligned} & d\tau = [\tau_1, \ldots, \tau_{m-1}, d\tau_0], \quad \text{for some} \quad \tau_1, \ldots, \tau_{m-1} \in T, \qquad d\tau_0 \in DT, \\ \text{equally,} \quad & d\tau = \tau \circ d\tau_0, \qquad\qquad\quad \text{for} \qquad\quad \tau = [\tau_1, \ldots, \tau_{m-1}]. \end{aligned}$$

The construction process described above implies that each derivative tree is associated with a single rooted tree. This tree can be computed by the mapping $o_\tau : DT \to T$, defined recursively as

$$\begin{aligned} o_\tau(\times) &= \emptyset, \\ o_\tau(d\tau) &= \tau \circ o_\tau(d\tau_0), \qquad \text{for} \qquad d\tau = \tau \circ d\tau_0. \end{aligned}$$

Example 3.11. Consider computing the original tree corresponding to  :



◇

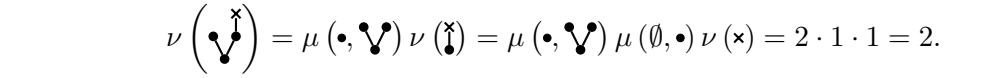
Conversely, a single rooted tree can be associated with various derivative trees, i.e. those belonging to the set $D_\tau(\tau)$. Associated with each $d\tau \in D_\tau(\tau)$ is a *multiplicity* - the number of ways $d\tau$ can be obtained from τ.

Definition 3.12 (**Multiplicity**)**.** The *multiplicity* of a derivative tree $d\tau \in DT$ is given by the mapping $\nu : DT \to \mathbb{R}$, defined recursively by

$$\begin{aligned} \nu(\times) &= 1, \\ \nu(d\tau) &= \mu(o_\tau(d\tau_0), o_\tau(d\tau))\nu(d\tau_0), \qquad \text{for} \qquad d\tau = [\tau_1, \ldots, \tau_{m-1}, d\tau_0], \end{aligned}$$

where $\mu(\tau_j, \tau)$ counts the occurrence of τ_j in $\tau = [\tau_1, \ldots, \tau_m]$ with $\mu(\emptyset, \cdot) = 1$.

Example 3.13. Consider the computation of ν  :



This arises from the equivalence of the trees

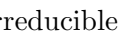
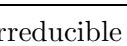
$\lvert d\tau\rvert$	$d\tau$-string	$d\tau$	$o_\tau(d\tau)$	$J(d\tau)(y,v)$	$\nu(d\tau)$	Lem. 3.16
0	$[\times]$			v	1	irreducible
1	$[[\times]]$			$f'(y)v$	1	irreducible
2	$[\bullet,\times]$			$f''(y)(f(y),v)$	1	irreducible
2	$[[[\times]]]$			$f'(y)f'(y)v$	1	
3	$[\bullet,\bullet,\times]$			$f'''(y)(f(y),f(y),v)$	1	irreducible
3	$[\bullet,[\times]]$			$f''(y)(f(y),f'(y)v)$	2	
3	$[[\bullet],\times]$			$f''(y)(f'(y)f(y),v)$	1	irreducible
3	$[[\bullet,\times]]$			$f'(y)f''(y)(f(y),v)$	1	
3	$[[[\times]]]$			$f'(y)f'(y)f'(y)v$	1	

Table 3.2: Examples of derivative trees up to order 3.

◇

Examples of derivative trees and the various quantities associated with each are given in Table 3.2 for trees up to order 3.

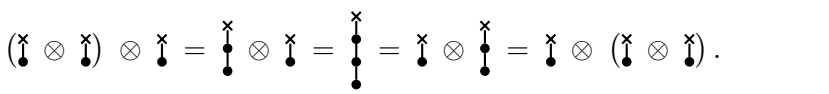
3.2.2 Operations and properties of derivative trees

The purpose of the $\times$ node of a derivative tree is to act as a placeholder, where another derivative tree or rooted tree may be inserted at a later time. This *substitution* property is described by the following tree operation.

Definition 3.14 (Substitution). Let $du \in DT$ and v be either a derivative or rooted tree. Then, the *substitution* operation is defined recursively as

$$\begin{aligned} \times \otimes v &= v, \\ du \otimes v &= (du \setminus du_0) \circ (du_0 \otimes v), \qquad \text{for} \qquad du = [u_1, \ldots, u_{m-1}, du_0]. \end{aligned}$$

The substitution operation is not commutative, but it is associative (for multiple substitutions of derivative trees). A simple example is given below:



This example also demonstrates that higher order derivative trees can be formed after a substitution of lower order trees. This observation leads to the following definition.

Definition 3.15 (**Irreducible derivative trees**)**.** A derivative tree $d\tau \in DT$ is *irreducible* if $d\tau = \tau \circ \times$, for some $\tau \in T \cup \{\emptyset\}$. Otherwise, it is *reducible*.

Lemma 3.16. *Let $du \in DT$ be reducible. Then, there exists a unique decomposition*

$$du = du_1 \otimes du_2 \otimes \cdots \otimes du_k, \qquad \text{for some} \qquad k \leq |du|,$$

where each $du_1, du_2, \ldots, du_k \in DT$ is irreducible.

Proof. Writing $du = u^{(1)} \circ du_0^{(1)}$ for some $u^{(1)} \in T \cup \{\emptyset\}$, $du_0^{(1)} \in DT \backslash \{\times\}$, then it follows from the definition of the substitution operation that

$$du = u^{(1)} \circ du_0^{(1)} = (u^{(1)} \circ x) \otimes du_0^{(1)}, \qquad \text{where} \qquad |du_0^{(1)}| = |du| - |u^{(1)}| < |du|.$$

By definition, $u^{(1)} \circ x =: du_1$ is an irreducible derivative tree. Thus, $du = du_1 \otimes du_0^{(1)}$. Now, if $du_0^{(1)}$ is also irreducible then we obtain the desired result. Assuming that $du_0^{(1)} = u^{(2)} \circ du_0^{(2)}$ is reducible, we re-apply the above argument to find

$$du = du_1 \otimes du_2 \otimes du_0^{(2)}, \qquad \text{where} \qquad |du_0^{(2)}| = |du_0^{(1)}| - |u^{(2)}| < |du_0^{(1)}|.$$

Again, if $du_0^{(1)}$ is irreducible we are finished. Otherwise, we continue repeating the above argument until $du_0^{(k)}$ is irreducible. Since $0 \leq |du_0^{(k+1)}| < |du_0^{(k)}|$, it follows that there exists some $k \leq |du|$ such that $du_0^{(k)}$ will be irreducible. Thus, $du = du_1 \otimes du_2 \otimes \cdots \otimes du_k$.

For uniqueness, suppose there exist irreducible trees $dv_1, \ldots, dv_l$, $l \leq k$ such that $du_1 \otimes \cdots \otimes du_k = dv_1 \otimes \cdots \otimes dv_l$, and there exists an $i \in \{1, \ldots, l\}$ such that $du_i \neq dv_i$. Then, writing $\widetilde{du} = du_2 \otimes \cdots \otimes du_k$ and $\widetilde{dv} = dv_2 \otimes \cdots \otimes dv_l$, we observe that

$$du_1 \otimes \widetilde{du} = (du_1 \backslash \times) \circ \widetilde{du} = (dv_1 \backslash \times) \circ \widetilde{dv} = dv_1 \otimes \widetilde{dv}.$$

Now, suppose we prune $\widetilde{du}$ from both sides of the above equation. This leaves a single rooted tree on the LHS, and it follows that the RHS operation must also leave a single rooted tree, i.e. it must remove $\widetilde{dv}$ from the RHS. This can only occur if $\widetilde{du} = \widetilde{dv}$. Comparing the remaining terms, we deduce that $du_1 = dv_1$.

Recursively applying the above argument on $\widetilde{du} = \widetilde{dv}$, we conclude that $du_j = dv_j$ for $j = 1, \ldots, l$. Now, if $l = k$, then we arrive at a contradiction since there is no $i \in \{1, \ldots, k\}$ such that $du_i \neq dv_i$.

If $l < k$, then we also arrive at a contradiction since

$$|dv| = |dv_1 \otimes \cdots \otimes dv_l| < |dv_1 \otimes \cdots \otimes dv_l| + |du_{l+1} \otimes \cdots \otimes du_k| = |du_1 \otimes \cdots \otimes du_k| = |du|,$$

which implies that $du_1 \otimes \cdots \otimes du_k \neq dv_1 \otimes \cdots \otimes dv_l$.

□

3.2.3 Derivative B-series and its properties

With the set of derivative trees now defined, we move on to consider derivatives of B-series. A key component of this will be the *elementary Jacobian*; an object related to the differentiated form of an elementary differential $F(\tau)(y)$.

Definition 3.17 (**Elementary Jacobian**)**.** For a tree $d\tau \in DT$, the *elementary Jacobian* is a mapping $J(d\tau) : (X, X) \to X$, defined recursively by

$$\begin{aligned} J(\times)(y, v) &= v, \\ J(d\tau)(y, v) &= f^{(m)}(y)(F(\tau_1)(y), \ldots, F(\tau_{m-1})(y), J(d\tau_0)(y, v)), \end{aligned}$$

for $d\tau = [\tau_1, \ldots, \tau_{m-1}, d\tau_0]$.

An elementary Jacobian without a fixed v-argument is essentially a square matrix with elements depending on y. Consequently, a product of elementary Jacobians can be written as

$$J(d\tau_1)(y, J(d\tau_2)(y, v)) = J(d\tau_1)(y, \cdot)J(d\tau_2)(y, \cdot)v.$$

In other words, an elementary Jacobian can be seen as a linear operator. Unless further clarification is required, we hereafter adopt the notation $J(d\tau) = J(d\tau)(y, \cdot)$ and $F(\tau) = F(\tau)(y)$, i.e. omitting all arguments.

Lemma 3.18. *The elementary Jacobian of a reducible tree $d\tau \in DT$, can be written as*

$$J(d\tau) = J(d\tau_1)J(d\tau_2) \cdots J(d\tau_k), \qquad \textit{where} \qquad d\tau = d\tau_1 \otimes d\tau_2 \otimes \cdots d\tau_k.$$

Proof. From Lemma 3.16, we know $d\tau_1, \ldots, d\tau_k$ are all irreducible. Now, we write $d\tau = d\tau_1 \otimes du_2$ where $d\tau_1 = [\tau_1, \ldots, \tau_{m-1}, \times]$, $du_2 = d\tau_2 \otimes \cdots \otimes d\tau_k$. Then, from the definitions of the substitution operation and the elementary Jacobian,

$$\begin{aligned} J(d\tau_1 \otimes du_2) &= J((d\tau_1 \setminus \times) \circ (\times \otimes du_2)) = J((d\tau_1 \setminus \times) \circ du_2), \\ &= f^{(m)}(F(\tau_1), \ldots, F(\tau_{m-1}), J(du_2)) = f^{(m)}(F(\tau_1), \ldots, F(\tau_{m-1}), \cdot)J(du_2), \\ &= J(\tau \circ \times)J(du_2) = J(d\tau_1)J(du_2). \end{aligned}$$

If $k = 2$, then $du_2 = d\tau_2$ is irreducible and we are finished. Otherwise, du_2 is trivially reducible, and we can reapply the above argument $k - 2$ more times until we find $du_k = d\tau_k$, at which point we obtain the desired result. □

A simple application of Lemmas 3.16 and 3.18 is given below:

$$J\left(\begin{array}{c}\bullet\\ \bullet\\ \bullet\end{array}\right) = J\left(\begin{array}{c}\bullet\\ \bullet\end{array}\otimes\begin{array}{c}\bullet\\ \bullet\end{array}\right) = J\left(\begin{array}{c}\bullet\\ \bullet\end{array}\right)J\left(\begin{array}{c}\bullet\\ \bullet\end{array}\right) = f'f''(f,\cdot).$$

Here, we see that the non-commutativity of the substitution operator is important as it ensures the non-commutativity of elementary Jacobian products.

Lemma 3.19. *Consider the tree $\tau = du \otimes v$, for some $du \in DT\backslash\{\times\}$, $v \in T$. Then,*

$$F(\tau) = J(du)F(v).$$

Proof. First, let us assume that du is irreducible. Then, for $du = u \circ \times$ where $u = [u_1, \ldots, u_{m-1}]$,

$$\begin{aligned} J(du)F(v) &= f^{(m)}(F(u_1), \ldots, F(u_{m-1}), \cdot)F(v), \\ &= f^{(m)}(F(u_1), \ldots, F(u_{m-1}), F(v)) = F(u \circ v). \end{aligned}$$

Now, by the definition of the substitution operator: $du \otimes v = (du\backslash\times) \circ (\times \otimes v) = (du\backslash\times) \circ v = u \circ v$. Thus, $F(u \circ v) = F(du \otimes v) = F(\tau)$.

Suppose now that du is reducible. Then, applying Lemma 3.18, we replace $J(du)$ in the above equation with $J(du_1)J(du_2)\cdots J(du_k)$ and apply the same argument k-times to find $F(du_1 \otimes \cdots \otimes du_k \otimes v) = F(du \otimes v) = F(\tau)$. □

Let us now consider more general combinations of elementary Jacobians, and in particular define the *derivative B-series*.

Definition 3.20. For a mapping $a : DT \to \mathbb{C}$, a formal series of the form

$$DB(a, y, v) = \sum_{d\tau \in DT} h^{|d\tau|} a(d\tau) J(d\tau)(y, v),$$

is called a *derivative B-series*, or more compactly, a *DB-series*.

Lemma 3.21. *The first and third arguments of a DB-series $DB(a, y, v)$ are linear.*

Proof. The method of proof for linearity of the first argument is the same as that given for Lemma 3.3. Linearity of the elementary Jacobian implies linearity in the third argument. □

As a consequence of linearity in the third argument, we can consider the product of a DB-series with either another DB-series or even a B-series.

Theorem 3.22. *The product of two DB-series is again a DB-series. That is, for the mappings $a, b, c : DT \to \mathbb{C}$, we have*

$$DB(c, y, v) = DB(a, y, DB(b, y, v)), \qquad \text{where} \qquad c(d\tau) := \sum_{du \otimes dv = d\tau} a(du)b(dv).$$

Proof. From the definition of a DB-series:

$$\begin{aligned} DB(a, y, DB(b, y, v)) &= \sum_{du \in DT} h^{|du|} a(du) J(du) DB(b, y, v), \\ &= \sum_{du \in DT} h^{|du|} a(du) J(du) \sum_{dv \in DT} h^{|dv|} b(dv) J(dv). \end{aligned}$$

Using linearity in the third argument and applying Lemma 3.18:

$$\begin{aligned} DB(a, y, DB(b, y, v)) &= \sum_{du \in DT} \sum_{dv \in DT} h^{|du|+|dv|} a(du)b(dv) J(du)J(dv), \\ &= \sum_{du \in DT} \sum_{dv \in DT} h^{|du \otimes dv|} a(du)b(dv) J(du \otimes dv). \end{aligned}$$

Making the substitution $d\tau = du \otimes dv$, then

$$DB(a, y, DB(b, y, v)) = \sum_{d\tau \in DT} h^{|d\tau|} c(d\tau) J(d\tau) = DB(c, y, v).$$

□

Theorem 3.23. *The product of a DB-series and B-series is a B-series. That is, for the mappings $a : DT \to \mathbb{C}$, $b, c : T \to \mathbb{C}$, where $b(\emptyset) = 0$, we have*

$$B(c, y) = DB(a, y, B(b, y)), \qquad \text{where} \qquad c(\emptyset) := 0, \quad \frac{c(\tau)}{\sigma(\tau)} := \sum_{du \otimes v = \tau} \frac{a(du)b(v)}{\sigma(v)}.$$

Proof. The proof is similar to that given above: By linearity and Lemma 3.19,

$$\begin{aligned} DB(a, y, B(b, y)) &= \sum_{du \in DT} \sum_{v \in T} h^{|du|+|v|} a(du) \frac{b(v)}{\sigma(v)} J(du) F(v), \\ &= \sum_{du \in DT} \sum_{v \in T} h^{|du \otimes v|} \frac{a(du)b(v)}{\sigma(v)} F(du \otimes v). \end{aligned}$$

$\lvert d\tau\rvert$	$d\tau$	$c\,(d\tau)$
0		$a\,(\bullet)\,b\,(\bullet)$
1		$a\,(\bullet)\,b\,(\bullet)$
2		$a\,(\bullet)\,b\,(\bullet)$
2		$a\,(\bullet)\,b\,(\bullet)$
3		$a\,(\bullet)\,b\,(\bullet)$
3		$a\,(\bullet)\,b\,(\bullet)$
3		$a\,(\bullet)\,b\,(\bullet)$
3		$a\,(\bullet)\,b\,(\bullet)$
3		$a\,(\bullet)\,b\,(\bullet)$

Table 3.3: Substitution formulae up to order 3 for two DB-series. (cf. Theorem 3.22).

Making the substitution $\tau = du \otimes v$, then

$$DB(a, y, B(b, y)) = \sum_{\tau \in T} \frac{h^{|\tau|}}{\sigma(\tau)} c(\tau) F(\tau) = B(c, y). \qquad \square$$

Remark 3.24. The assumption $b(\emptyset) = 0$ in the above theorem is important from both an algebraic and physical perspective as it prevents the generation of terms such as $f'(y)y$, $f''(y)(f(y), y)$, etc. These terms are not classified as elementary differentials nor do they make any sort of physical sense.

Tree formulae for the product with DB-series up to order 3, and for the product with B-series up to order 4 have been computed and are displayed respectively in Tables 3.3 and 3.4.

Now, we move onto the main result of this section: establishing the connection between the derivative of a B-series and the DB-series itself.

Theorem 3.25. *The derivative of a B-series is a DB-series. That is, for the mappings* $a : T \cup \{\emptyset\} \to \mathbb{C}$, $b : DT \to \mathbb{C}$. *Then,*

$$DB(b, y, \cdot) = \nabla_y B(a, y), \qquad \text{for} \qquad b(d\tau) := \frac{a(o_\tau(d\tau))}{\sigma(o_\tau(d\tau))} \nu(d\tau).$$

$\lvert\tau\rvert$	τ	$c(\tau)/\sigma(\tau)$
0	$\emptyset$	0
1		$a(\times)\, b(\bullet)/\sigma(\bullet)$
2		$a(\times)\, b\left(\right.$$\left.\right)/\sigma\left(\right.$$\left.\right) + a\left(\right.$$\left.\right) b(\bullet)/\sigma(\bullet)$
3		$a(\times)\, b\left(\right.$$\left.\right)/\sigma\left(\right.$$\left.\right) + a\left(\right.$$\left.\right) b(\bullet)/\sigma(\bullet)$
3		$a(\times)\, b\left(\right.$$\left.\right)/\sigma\left(\right.$$\left.\right) + a\left(\right.$$\left.\right) b\left(\right.$$\left.\right)/\sigma\left(\right.$$\left.\right) + a\left(\right.$$\left.\right) b(\bullet)/\sigma(\bullet)$
4		$a(\times)\, b\left(\right.$  $\left.\right)/\sigma\left(\right.$$\left.\right) + a\left(\right.$$\left.\right) b(\bullet)/\sigma(\bullet)$
4		$a(\times)\, b\left(\right.$  $\left.\right)/\sigma\left(\right.$$\left.\right) + a\left(\right.$$\left.\right) b\left(\right.$$\left.\right)/\sigma\left(\right.$$\left.\right) +$ $a\left(\right.$$\left.\right) b(\bullet)/\sigma(\bullet) + a\left(\right.$$\left.\right) b(\bullet)/\sigma(\bullet)$
4		$a(\times)\, b\left(\right.$  $\left.\right)/\sigma\left(\right.$$\left.\right) + a\left(\right.$$\left.\right) b\left(\right.$$\left.\right)/\sigma\left(\right.$$\left.\right) +$ $a\left(\right.$$\left.\right) b(\bullet)/\sigma(\bullet)$
4		$a(\times)\, b\left(\right.$$\left.\right)/\sigma\left(\right.$$\left.\right) + a\left(\right.$$\left.\right) b\left(\right.$$\left.\right)/\sigma\left(\right.$$\left.\right) +$ $a\left(\right.$$\left.\right) b\left(\right.$$\left.\right)/\sigma\left(\right.$$\left.\right) + a\left(\right.$$\left.\right) b(\bullet)/\sigma(\bullet)$

Table 3.4: Substitution formulae up to order 4 for DB-series and B-series. (cf. Theorem 3.23).

Proof. Let us first consider the differentiation of an elementary differential:

$$\begin{aligned}\nabla_y F(\tau) &= \nabla_y (f^{(m)}(F(\tau_1), \dots, F(\tau_m))), \\ &= f^{(m+1)}(F(\tau_1), \dots, F(\tau_m), \cdot), \\ &+ \sum_{j=1}^{m} f^{(m)}(F(\tau_1), \dots, F(\tau_{j-1}), F(\tau_{j+1}), \dots, F(\tau_m), \cdot) \nabla_y F(\tau_j).\end{aligned}$$

Recalling the definition of the elementary Jacobian, this can be written as

$$\nabla_y F(\tau) = \sum_{u_1 \in \text{ ch}(\tau)} J((\tau \backslash u_1) \circ \times) \nabla_y F(u_1).$$

Since $\nabla_y F(\emptyset) = I_X$ and $\nabla_y F(\bullet) = J(\overset{\times}{\bullet}) = f'$, it follows that the RHS is given as

$$\nabla_y F(\tau) = \sum_{u_1 \in \text{ ch}(\tau)} J((\tau \backslash u_1) \circ \times) \sum_{u_2 \in \text{ ch}(u_1)} J((u_1 \backslash u_2) \circ \times) \cdots \sum_{u_{k-1} \in \text{ ch}(u_k)} J((u_k \backslash u_{k-1}) \circ \times).$$

It now follows from Definition 3.7 and Lemma 3.18 that this sum can be compactly written as

$$\nabla_y F(\tau) = \sum_{d\tau \in D\tau(\tau)} \nu(d\tau) J(d\tau).$$

Here, the multiplicity $\nu(d\tau)$ accounts for the multiple occurrences the elementary Jacobian $J(d\tau)$ that may have arisen in the previous summation.

Now, we differentiate the B-series to find

$$\nabla_y B(a, y) = \sum_{\tau \in T} \frac{h^{|\tau|}}{\sigma(\tau)} a(\tau) \nabla_y F(\tau) = \sum_{\tau \in T} \frac{h^{|\tau|}}{\sigma(\tau)} a(\tau) \sum_{d\tau \in D\tau(\tau)} \nu(d\tau) J(d\tau).$$

As each $d\tau$ is associated with exactly one τ, and $|\tau| = |o_\tau(d\tau)| = |d\tau|$, we may re-write the above as

$$\nabla_y B(a, y) = \sum_{d\tau \in DT} h^{|d\tau|} b(d\tau) J(d\tau) = DB(b, y, \cdot),$$

as required. □

Example 3.26. At the beginning of Section 2.2, the derivative $h^3 \frac{\mathrm{d}^3 y}{\mathrm{d}t^3}$ was expressed as a B-series:

$$h^3 \frac{\mathrm{d}^3 y}{\mathrm{d}t^3} = h^3 f''(f,f) + h^3 f' f' f =: B(a,y),$$

where the corresponding B-series coefficients are

$a(\emptyset) = a(\bullet) = a($ $) = 0, \qquad a($ $) = 2, \qquad a($ $) = 1, \qquad a(\tau) = 0, \qquad \forall\ |\tau| > 3.$

Consider now the problem of computing $h^4 \frac{\mathrm{d}^4 y}{\mathrm{d}t^4}$ as a B-series. Since

$$h^4 \frac{\mathrm{d}^4 y}{\mathrm{d}t^4} = h \frac{\mathrm{d}}{\mathrm{d}t} B(a,y) = \nabla_y B(a,y) \cdot hf,$$

the B-series can be obtained using the formulae for differentiation and product with a B-series. Firstly, let us compute $DB(b,y,\cdot) = \nabla_y B(a,y)$; the non-zero coefficients are given below

$b($  $) = a($  $)\, \nu($  $)\, /\sigma($ $) = 1, \qquad b($  $) = a($ $)\, \nu($ $)\, /\sigma($ $) = 2,$

$b($  $) = a($  $)\, \nu($ $)\, /\sigma($ $) = 1, \qquad b($  $) = a($ $)\, \nu($ $)\, /\sigma($ $) = 1,$

$b($  $) = a($ $)\, \nu($ $)\, /\sigma($ $) = 1.$

Next, we compute the B-series $B(c,y) = DB(b,y,hf(y))$ using the product formulae given in Table 3.4. The non-zero coefficients are given below

$c($ $)\, /\sigma($ $) = b($ $) = 1, \qquad c($  $)\, /\sigma($ $) = b($  $) + b($ $) = 3,$

$c($  $)\, /\sigma($ $) = b($ $) = 1, \qquad c($  $)\, /\sigma($ $) = b($ $) = 1.$

Thus, these coefficients tell us that $h^4 \frac{\mathrm{d}^4 y}{\mathrm{d}t^4}$ can be written as

$$h^4 \frac{\mathrm{d}^4 y}{\mathrm{d}t^4} = h^4 f'''(f,f,f) + 3h^4 f''(f'f,f) + h^4 f' f''(f,f) + h^4 f' f' f' f.$$

◇

3.2.4 Extension to matrix DB-series

Similar to vector B-series, it will be also be convenient to consider DB-coefficients that map to $\mathbb{C}^{m\times n}$:

Definition 3.27 (**Matrix DB-series**)**.** For a mapping $a : DT \to \mathbb{C}^{m\times n}$, a formal series of the form

$$DB(a, y, v) = \sum_{d\tau\in DT} h^{|d\tau|} \left(a(d\tau) \otimes J(d\tau)(y, \cdot)\right) v, \qquad v \in X^n,$$

is called a $(m \times n)$-*matrix DB-series*.

All of the results in the previous sections extend straightforwardly to the matrix formulation of a DB-series. For example, the product of an $(m \times n)$-matrix DB-series with an n-vector B-series yields an m-vector B-series.

3.3 *A priori* parasitism analysis

In this section, we shall focus our attention on developing an *a priori* analysis of parasitism[1]. In particular, we shall address the following issues:

1. Modelling parasitism.
2. Bounding parasitic growth.
3. Deriving algebraic conditions that delay the onset of parasitism.

3.3.1 Modelling parasitism

The first problem to tackle in the modelling process is clarifying the definition for parasitism. In Section 2.4, we loosely defined this as follows:

> *[Parasitism] describes the unacceptable growth of perturbations made to the non-principal components of the method.*

Here, we see that the interpretation of *'perturbations'* is important as it will essentially determine the type of model we consider. For example, if we are concerned with the influence of rounding error on our numerical solution, then we assume perturbations are small and arbitrary, and apply them to our solution at some time $t = nh$. Alternatively, we might investigate the divergence of our numerical solution from some fixed

[1] Backward error analysis is used to obtain *a posteriori* bounds on the parasitic components of a method by studying a modified differential equation, the solution of which exactly satisfies the method. Here, we will consider a direct (*a priori*) approach that does not use the modified equation.

trajectory. In which case, the perturbation would be taken as the initial difference between trajectories at time $t = 0$.

Adopting the view that parasitism describes the divergence from some fixed trajectory, we would like to consider how the numerical solution of a GLM diverges from that which would be obtained using the underlying one-step method (UOSM). Recall that the UOSM Φ_h satisfies

$$\mathcal{M}_h \circ \mathcal{S}_h^*(y_0) = \mathcal{S}_h^* \circ \Phi_h(y_0).$$

Then, in other words, we seek an understanding of how

$$||\mathcal{M}_h^n \circ \mathcal{S}_h(y_0) - \mathcal{S}_h^* \circ \Phi_h^n(y_0)||, \qquad \text{and} \qquad ||\mathcal{F}_h \circ \mathcal{M}_h^n \circ \mathcal{S}_h(y_0) - \Phi_h^n(y_0)||,$$

grow with increasing n. As the UOSM is a one-step method, it does not suffer from parasitism. Thus, for as long as the GLM solution stays close to the UOSM solution, it will also remain parasitism-free.

For this model, we take the initial perturbation to be the difference in starting methods, i.e. $\delta_h(y_0) := \mathcal{S}_h(y_0) - \mathcal{S}_h^*(y_0)$ where $\mathcal{S}_h$ denotes the starting method used in practice. Since $\mathcal{S}_h^*$ exists as a formal series, $\delta_h(y_0)$ is generally non-zero. Furthermore, if the pair $(\mathcal{M}_h, \mathcal{S}_h)$ is of order p, then it follows from Corollary 2.28 that $\delta_h(y_0) = O(h^{p+1})$; significantly larger than rounding error.

Let us now consider the outcome of introducing the above perturbation at time $t = 0$: Writing $O(\delta^2) := O(||\delta_h(y_0)||^2)$, we have

$$\begin{aligned}
\mathcal{M}_h^n \circ \mathcal{S}_h(y_0) &= \mathcal{M}_h^n(\mathcal{S}_h^*(y_0) + \delta_h(y_0)), \\
&= \mathcal{M}_h^n \circ \mathcal{S}_h^*(y_0) + [\mathcal{M}_h^n(\mathcal{S}_h^*(y_0))]' \delta_h(y_0) + O(\delta^2), \\
&= \mathcal{S}_h^* \circ \Phi_h^n(y_0) + [\mathcal{M}_h^n(\mathcal{S}_h^*(y_0))]' \delta_h(y_0) + O(\delta^2),
\end{aligned}$$

where $'$ denotes a Fréchet derivative. Applying the chain rule, and using the notation

$$F'_{h,k} := \mathcal{F}'_h(\mathcal{S}_h^* \circ \Phi_h^k(y_0)), \qquad \text{and} \qquad M'_{h,k} := \mathcal{M}'_h(\mathcal{S}_h^* \circ \Phi_h^k(y_0)), \qquad k \in \mathbb{N}_0,$$

we find

$$\mathcal{M}_h^n \circ \mathcal{S}_h(y_0) = \mathcal{S}_h^* \circ \Phi_h^n(y_0) + M'_{h,n-1} M'_{h,n-2} \cdots M'_{h,0} \delta_h(y_0) + O(\delta^2), \tag{3.1}$$

$$\mathcal{F}_h \circ \mathcal{M}_h^n \circ \mathcal{S}_h(y_0) = \Phi_h^n(y_0) + F'_{h,k} M'_{h,n-1} M'_{h,n-2} \cdots M'_{h,0} \delta_h(y_0) + O(\delta^2). \tag{3.2}$$

Assuming terms of $O(\delta^2)$ can be neglected, then this model attributes parasitism to

the following product:

$$M'_{h,n-1}M'_{h,n-2}\cdots M'_{h,0}\delta_h(y_0). \tag{3.3}$$

3.3.2 Derivative UOSMs

Having settled on an appropriate model for parasitism, we would now like to apply the theory of derivative B-series to better understand the product (3.3). Let us begin by introducing the concept of *derivative UOSMs* - the derivative analogue of the UOSM.

Definition 3.28. The map $\Psi_h : (X, X) \to X$ is called a *derivative UOSM* (DUOSM) if

$$\mathcal{M}'_h(\mathcal{S}^*_h(y_0))S_h(y_0, v) = S_h(\Phi_h(y_0), \Psi_h(y_0, v)), \qquad \forall\ y_0, v \in X, \tag{3.4}$$

for some *derivative starting method* $S_h : (X, X) \to X^r$.

Here, we remark that both of the maps S_h and Ψ_h are linear in their second argument. We also note that to leading order (3.4) is equivalent to finding an eigenpair of V. Thus, in the case that V has distinct eigenvalues, there will be r distinct DUOSMs, i.e. let (ζ_P, u_P) be an eigenpair of V, then $\Psi^P_h(y_0, \cdot) = \zeta_P + O(h)$ and $S^P_h(y_0, \cdot) = u_P + O(h)$, for $P = 1, \ldots, r$.

Theorem 3.29. *The derivative of the UOSM Φ_h, is a DUOSM, i.e.*

$$\Psi^{(1)}_h(y_0, v) := \Phi'_h(y_0)v, \qquad \textit{and} \qquad S^{(1)}_h(y_0, v) := (\mathcal{S}^*_h(y_0))' v, \qquad v \in X,$$

satisfy (3.4). *Furthermore, for $F^{(1)}_h(y_0, \cdot) := \mathcal{F}'_h(\mathcal{S}^*_h(y_0))$, then*

$$F^{(1)}_h(y_0, S^{(1)}_h(y_0, v)) = v.$$

Proof. Differentiation of the UOSM condition (2.12) with respect to y_0, and post-multiplication by some $v \in X$ yields

$$\mathcal{M}'_h(\mathcal{S}^*_h(y_0))\,(\mathcal{S}^*_h(y_0))'\, v = (\mathcal{S}^*_h(\Phi_h(y_0)))'\,\Phi'_h(y_0)v.$$

Defining $\Psi^{(1)}_h(y_0, v) := \Phi'_h(y_0)v$ and $S^{(1)}_h(y_0, v) := (\mathcal{S}^*_h(y_0))' v$ gives the first result.

Differentiation of $\mathcal{F}_h \circ \mathcal{S}^*_h(y_0) = y_0$, with respect to y_0, and post-multiplication by some $v \in X$ gives the second result. □

The DUOSM of the above theorem shall be called the *trivial DUOSM* - that which is associated with the principal component of the method. The existence of non-trivial DUOSMs, i.e. those associated with the non-principal components of the method, is considered in the following theorem.

Theorem 3.30. *Consider a consistent GLM $\mathcal{M}_h$ such that V possesses distinct eigenvalues. Then, for each mapping $f_P : DT \to \mathbb{C}^{1\times r}$, $P \in \{1, \dots r\}$, that defines the (row-vector) DB-series*

$$F_h^P(y_0, \cdot) := DB(f_P, y_0, \cdot) = w_P^{\mathsf{H}} + O(h), \qquad \text{such that} \qquad w_P^{\mathsf{H}} u_P = 1,$$

where u_P is the right-eigenvector of V corresponding to eigenvalue ζ_P, there exist a unique $\psi_P : DT \to \mathbb{C}$ that defines the DB-series

$$\Psi_h^P(y_0, v) := DB(\psi_P, y_0, v) = \zeta_P v + h\psi_P(\bullet^{\times}_{\bullet})J(\bullet^{\times}_{\bullet})(y, v) + \dots,$$

and a unique $s_P : DT \to \mathbb{C}^r$ that defines the (vector) DB-series

$$S_h^P(y_0, v) := DB(s_P, y_0, v) = u_P v + h s_P(\bullet^{\times}_{\bullet}) \otimes J(\bullet^{\times}_{\bullet})(y, v) + \dots,$$

such that

$$\begin{aligned} \mathcal{M}_h'(\mathcal{S}_h^*(y_0))S_h^P(y_0, v) &= S_h^P(\Phi_h(y_0), \Psi_h(y_0, v)), \\ F_h^P(y_0, S_h^P(y_0, v)) &= v, \end{aligned}$$

hold for all $v \in X$, as formal power series in h.

Proof. The method of proof is similar to that applied to Theorem 2.27: Firstly, we expand $\mathcal{M}_h'(\mathcal{S}_h^*(y_0))$ about uy_0 to obtain an $(r \times r)$-matrix DB-series and $S_h^P(\Phi_h(y_0), \cdot)$ about y_0 to obtain an r-vector DB-series. Then, we perform a DB-series substitution (cf. Theorem 3.22) such that we can evaluate the expression $S_h^P(\Phi_h(y_0), \Psi_h(y_0, v)) - \mathcal{M}_h'(\mathcal{S}_h^*(y_0))S_h^P(y_0, v) = 0$ as an r-vector DB-series. Comparing the coefficients of elementary Jacobians, and re-arranging such that only the highest order unknowns appear on the LHS, we find

$$\begin{aligned}
O(1): &\quad (\zeta_P I_r - V)u_P = 0, \\
O(h): &\quad (\zeta_P I_r - V)s_P(\bullet^{\times}_{\bullet}) + u_P\psi_P(\bullet^{\times}_{\bullet}) = BUu_P, \\
O(h^2): &\quad (\zeta_P I_r - V)s_P\left(\bullet^{\times}_{\bullet}\right) + u_P\psi_P\left(\bullet^{\times}_{\bullet}\right) = BUs_P(\bullet^{\times}_{\bullet}) + BAUu_P - s_P(\bullet^{\times}_{\bullet})\psi_P(\bullet^{\times}_{\bullet}), \\
O(h^2): &\quad (\zeta_P I_r - V)s_P(\bullet^{\times}_{\bullet\!\!\diagup\!\!\diagdown}) + u_P\psi_P(\bullet^{\times}_{\bullet\!\!\diagup\!\!\diagdown}) = BCUu_P - \zeta_P s_P(\bullet^{\times}_{\bullet})\phi(\bullet), \\
&\qquad\qquad \vdots
\end{aligned}$$

where $C := \mathrm{diag}(c)$, $c := U\mathcal{S}^*(\bullet) + A\mathbb{1}$ and $\phi(\bullet)$, $\mathcal{S}^*(\bullet)$ respectively correspond to the hf-coefficient of the UOSM Φ_h and the ideal starting method $\mathcal{S}_h^*$.

We do the same for $F_h^P(y_0, S_h^P(y_0, v)) - v = 0$. Here, we need only apply the substitution formula from Theorem 3.22 and re-arrange:

$$
\begin{aligned}
O(1): &\quad w_P^{\mathsf{H}} u_P = 1, \\
O(h): &\quad w_P^{\mathsf{H}} s_P(\bullet^{\times}) = -f_P(\bullet^{\times}) u_P, \\
O(h^2): &\quad w_P^{\mathsf{H}} s_P\left(\bullet^{\times}\right) = -f_P(\bullet^{\times}) s_P(\bullet^{\times}) - f_P\left(\bullet^{\times}\right) u_P, \\
O(h^2): &\quad w_P^{\mathsf{H}} s_P(\bullet^{\times}) = -f_P(\bullet^{\times}) u_P, \\
&\quad \vdots
\end{aligned}
$$

As in the proof of Theorem 2.27, these comparisons generally lead to a system of equations of the form

$$
\begin{bmatrix} \zeta_P I_r - V & u_P \\ w_P^{\mathsf{H}} & 0 \end{bmatrix} \begin{bmatrix} s_P(d\tau) \\ \psi_P(d\tau) \end{bmatrix} = \begin{bmatrix} G(d\tau) \\ g(d\tau) \end{bmatrix}, \qquad |d\tau| \geq 1,
$$

where the RHS terms $G(d\tau)$ and $g(d\tau)$ depend on the known quantities $s_P(du)$, $\psi_P(du)$ for all $du \in DT$ such that $|du| < |d\tau|$.

Now, since the eigenvalues of V are distinct and $w_P^{\mathsf{H}} u_P = 1$, each system is uniquely solvable (by the ABCD Lemma [55]). Thus, the maps $S_h^P(y_0, v)$ and $\Psi_h^P(y_0, v)$ determined by s_P and ψ_P uniquely satisfy the DUOSM condition (3.4) and $F_h^P(y_0, S_h^P(y_0, v)) = v$, as required. □

3.3.3 Decomposition of parasitism product

It is now our intention to make a connection between the parasitism product (3.3) and DUOSMs by means of decomposition. An important part of this will be understanding the structure of the perturbation $\delta_h(y_0)$.

Theorem 3.31. *Let the assumptions of Theorem 3.30 hold and consider the mappings $f_P : DT \to \mathbb{C}^{1\times r}$, $P = 1, \ldots, r$, that define the (row-vector) DB-series:*

$$
F_h^P(y_0, \cdot) := DB(f_P, y_0, \cdot) = w_P^{\mathsf{H}} + O(h), \qquad \text{for } P = 1, \ldots, r,
$$

where w_P^{H} is chosen to be the left eigenvector of V corresponding to eigenvalue ζ_P. Then, there exist unique B-series $B(c_1, y_0), \ldots, B(c_r, y_0)$ such that

$$
\delta_h(y_0) = \sum_{P=1}^{r} S_h^P(y_0, B(c_P, y_0)). \tag{3.5}
$$

Proof. It follows from Theorem 3.30 that for each F_h^P, there exists a unique, formal, S_h^P that satisfies the DUOSM condition (3.4) and $F_h^P(y_0, S_h^P(y_0, v)) = v$. Now, suppose

$$\delta_h(y_0) = \sum_{P=1}^{r} S_h^P(y_0, B(c_P, y_0)),$$

for some B-series $B(c_1, y_0), \ldots, B(c_r, y_0)$ as yet to be determined. Then, for $P = 1, \ldots, r$, we have

$$F_h^P(y_0, \delta_h(y_0)) = F_h^P(y_0, S_h^{(1)}(y_0, B(c_1, y_0))) + \cdots + B(c_P, y_0) + \cdots + F_h^P(y_0, S_h^{(r)}(y_0, B(c_r, y_0))).$$

Recall that, to leading order, $F_h^P = w_P^{\mathsf{H}}$ and $S_h^P = u_P$, i.e. the left and right eigenvectors of V. It then follows that

$$F_h^P(y_0, \delta_h(y_0)) = B(c_P, y_0) + O(h),$$

for $P = 1, \ldots, r$. These r-many equations determine the following square system: Writing $F_h^{(i)}(y_0, S_h^{(j)}(y_0, v)) =: F_h^{(i)} S_h^{(j)}$, and $F_h^{(i)}(y_0, \delta_h(y_0)) =: F_h^{(i)} \delta_h(y_0)$ we then have

$$\begin{bmatrix} I_X & F_h^{(1)} S_h^{(2)} & \cdots & F_h^{(1)} S_h^{(r)} \\ F_h^{(2)} S_h^{(1)} & I_X & \cdots & F_h^{(2)} S_h^{(r)} \\ \vdots & \vdots & \ddots & \vdots \\ F_h^{(r)} S_h^{(1)} & F_h^{(r)} S_h^{(2)} & \cdots & I_X \end{bmatrix} \begin{bmatrix} B(c_1, y_0) \\ B(c_2, y_0) \\ \vdots \\ B(c_r, y_0) \end{bmatrix} = \begin{bmatrix} F_h^{(1)} \delta_h(y_0) \\ F_h^{(2)} \delta_h(y_0) \\ \vdots \\ F_h^{(r)} \delta_h(y_0) \end{bmatrix}.$$

As the LHS matrix is $I_{X_r} + O(h)$, it follows that it is invertible. Furthermore, since each element of the inverse matrix is a DB-series, and each element of the RHS vector is a B-series, it follows that we may uniquely solve for each $B(c_1, y_0), \ldots, B(c_r, y_0)$ using the substitution formula of Theorem 3.23.

□

Remark 3.32. The requirement that all w_P^{H} are left-eigenvectors of V is a sufficient one. For example, we can allow a single w_P^{H} to a be linear combination of left-eigenvectors and still find a unique decomposition for $\delta_h(y_0)$ of the form (3.5).

The existence of a $\delta_h(y_0)$ decomposition in terms of derivative starting methods is necessary for expressing the parasitism product in terms of DUOSMs.

Corollary 3.33. *The parasitism product* (3.3) *may be equivalently written as*

$$M'_{h,n-1}\cdots M'_{h,0}\delta_h(y_0) = \sum_{P=1}^{r} S^P_{h,0}\Psi^P_{h,n-1}\cdots\Psi^P_{h,0}B(c_P,y_0),$$

where, for some $k\in\mathbb{N}_0$,

$$M'_{h,k} := \mathcal{M}'_h(\mathcal{S}^*_h\circ\Phi^k_h(y_0)), \qquad S^P_{h,k} := S^P_h(\Phi^k_h(y_0),\cdot), \qquad \Psi^P_{h,k} := \Psi^P_h(\Phi^k_h(y_0),\cdot).$$

Proof. Consider a fixed $P\in\{1,\dots,r\}$ and note that

$$M'_{h,0}S^P_{h,0} = \mathcal{M}'_h(\mathcal{S}^*_h(y_0))S^P_h(y_0,\cdot) = S^P_h(\Phi_h(y_0),\Psi^P_h(y_0,\cdot)) = S^P_{h,1}\Psi^P_{h,0}.$$

Pre-multiplying through by $M'_{h,1}$ and applying the DUOSM condition with $y_0\mapsto\Phi_h(y_0)$, we find

$$M'_{h,1}M'_{h,0}S^P_{h,0} = M'_{h,1}S^P_{h,1}\Psi^P_{h,0} = S^P_{h,2}\Psi^P_{h,1}\Psi^P_{h,0}.$$

Repeating the above procedure, each time pre-multiplying by $M'_{h,k}$ and applying the DUOSM condition with $y_0\mapsto\Phi^k_h(y_0)$ for $k=2,\dots,n-1$, we obtain

$$M'_{h,n-1}M'_{h,n-2}\cdots M'_{h,0}S^P_{h,0} = S^P_{h,n}\Psi^P_{h,n-1}\Psi^P_{h,n-2}\cdots\Psi^P_{h,0}.$$

The result now follows from writing $\delta_h(y_0)$ in the form given by equation (3.5) and applying the above result for each $P\in\{1,\dots,r\}$. □

The above result is not yet complete as the sum still contains a term that evolves in the principal direction, i.e. in the direction of $S^{(1)}_h$. Throughout, we have insisted that perturbations made in the non-principal directions contribute are responsible for parasitism. In the following theorem, we demonstrate why perturbations in the principal direction can be essentially neglected.

Theorem 3.34. *Expansions* (3.1)-(3.2) *may be written as*

$$\mathcal{M}^n_h\circ\mathcal{S}_h(y_0) = \mathcal{S}^*_h\circ\Phi^n_h(y_0+B(c_1,y_0)) + \sum_{P=2}^{r} S^P_{h,n}\Psi^P_{h,n-1}\cdots\Psi^P_{h,0}B(c_P,y_0) + O(\delta^2),$$

$$\mathcal{F}_h\circ\mathcal{M}^n_h\circ\mathcal{S}_h(y_0) = \Phi^n_h(y_0+B(c_1,y_0)) + F^{(1)}_{h,n}\sum_{P=2}^{r} S^P_{h,n}\Psi^P_{h,n-1}\cdots\Psi^P_{h,0}B(c_P,y_0) + O(\delta^2).$$

Proof. Let us make the choice that $F_h^{(1)}(y_0,\cdot) := \mathcal{F}_h'(\mathcal{S}_h^*(y_0)) = w^{\mathsf{H}} + O(h)$. Then, from Theorems 3.29 and 3.30 it follows that $\Psi_h^{(1)}(y_0,\cdot) = \Phi_h'(y_0)$ and $S_h^{(1)}(y_0,\cdot) = (\mathcal{S}_h^*(y_0))'$.

For the remaining $F_h^P(y_0,\cdot)$, $P = 2,\dots,r$, define the mappings $f_P : DT \to \mathbb{R}^{1\times r}$ such that

$$f_P(\mathsf{x}) = w_P^{\mathsf{H}}, \qquad f_P(d\tau) = \text{arbitrary}, \qquad \forall\ |d\tau| > 0,$$

and let $F_h^P(y_0,\cdot) := DB(f_P, y_0, \cdot)$. Then, we apply Theorem 3.31 and Corollary 3.33 to give

$$\begin{aligned} M_{h,n-1}'\cdots M_{h,0}'\delta_h(y_0) &= \sum_{P=1}^{r} S_{h,n}^P \Psi_{h,n-1}^P \cdots \Psi_{h,0}^P B(c_P, y_0), \\ &= S_{h,n}^{(1)}\Psi_{h,n-1}^{(1)}\cdots\Psi_{h,0}^{(1)}B(c_1,y_0) + \\ &\qquad \sum_{P=2}^{r} S_{h,n}^P \Psi_{h,n-1}^P \cdots \Psi_{h,0}^P B(c_P, y_0). \end{aligned}$$

Now, substituting the above into (3.1) and writing $\Phi_{h,k}' := \Phi_h'(\Phi_h^k(y_0))$, we find

$$\begin{aligned} \mathcal{M}_h^n\circ\mathcal{S}_h(y_0) = \mathcal{S}_h^*\circ\Phi_h^n(y_0) + S_{h,n}^{(1)}\Phi_{h,n-1}'\cdots\Phi_{h,0}'B(c_1,y_0) + \\ \sum_{P=2}^{r} S_{h,n}^P \Psi_{h,n-1}^P \cdots \Psi_{h,0}^P B(c_P, y_0) + O(\delta^2), \end{aligned}$$

and since $B(c_1, y_0) = O(\delta)$, which follows from the proof of Theorem 3.31, we may write

$$\mathcal{M}_h^n\circ\mathcal{S}_h(y_0) = \mathcal{S}_h^*\circ\Phi_h^n(y_0 + B(c_1,y_0)) + \sum_{P=2}^{r} S_{h,n}^P \Psi_{h,n-1}^P \cdots \Psi_{h,0}^P B(c_P, y_0) + O(\delta^2),$$

to obtain the first result. The second result follows from applying the finishing method and noting that $F_{h,n}^{(1)}S_{h,n}^{(1)} = I$. □

3.3.4 Parasitic bounds

With Theorem 3.34 revealing that only the non-trivial DUOSMs contribute towards parasitism, we proceed now by determining a bound on the parasitic components.

Theorem 3.35. *Suppose that there exist constants $L > 0$ and $M \in \mathbb{N}_0$ such that*

$$\max_{2\leq P\leq r} ||\Psi_h^P(y,\cdot)|| \leq 1 + Lh^{M+1}, \qquad \text{holds for all} \qquad y \in X, \tag{3.6}$$

and there exist constants $C_0, C_1, C_2 > 0$, $k_1 \in \mathbb{N}_0$, and $k_2 \in \mathbb{N}$ such that

$$\max_{2\leq P\leq r} ||S_h^P(y,\cdot)|| \leq C_0, \ \max_{2\leq P\leq r} ||F_h'(y, S_h^P(y,\cdot))|| \leq C_1 h^{k_1}, \ \max_{2\leq P\leq r} ||B(c_P, y)|| \leq C_2 h^{k_2},$$

hold for all $y \in X$. Then, for $t = nh$,

$$\left|\left|\sum_{P=2}^{r} S_{h,n}^P \Psi_{h,n-1}^P \cdots \Psi_{h,0}^P B(c_P, y_0)\right|\right| \leq K_1(h,r)\exp(th^M L),$$
$$\left|\left|F_{h,n}' \sum_{P=2}^{r} S_{h,n}^P \Psi_{h,n-1}^P \cdots \Psi_{h,0}^P B(c_P, y_0)\right|\right| \leq K_2(h,r)\exp(th^M L),$$

where $K_1(h,r)$, $K_2(h,r)$ are constants dependent on h and r.

Proof. Taking norms we find

$$\begin{aligned}\left|\left|\sum_{P=2}^{r} S_{h,n}^P \Psi_{h,n-1}^P \cdots \Psi_{h,0}^P B(c_P, y_0)\right|\right| &\leq (r-1)\max_P \left|\left|S_{h,n}^P \Psi_{h,n-1}^P \cdots \Psi_{h,0}^P B(c_P, y_0)\right|\right|, \\ &\leq (r-1)\max_P \left|\left|S_{h,n}^P\right|\right| \cdot ||\Psi_{h,n-1}^P|| \cdots \\ &\qquad ||\Psi_{h,0}^P|| \cdot ||B(c_P, y_0)||, \\ &= (r-1)C_0C_2h^{k_2}(1+Lh^{M+1})^n.\end{aligned}$$

As $1 + x \leq \exp(x)$, for $x \geq 0$, we can put $K_1(h,r) = (r-1)C_0C_2h^{k_2}$ to obtain

$$\left|\left|\sum_{P=2}^{r} S_{h,n}^P \Psi_{h,n-1}^P \cdots \Psi_{h,0}^P B(c_P, y_0)\right|\right| \leq K_1(h,r)\exp(nh^{M+1}L).$$

Setting $t = nh$ gives the first result.

The second result can be deduced using the same approach given above. This yields

$$\left|\left|F_{h,n}' \sum_{P=2}^{r} S_{h,n}^P \Psi_{h,n-1}^P \cdots \Psi_{h,0}^P B(c_P, y_0)\right|\right| \leq K_2(h,r)\exp(th^M L),$$

where $K_2(h,r) = (r-1)C_1C_2h^{k_1+k_2}$. ∎

Remark 3.36. Below is a collection of essential observations and conclusions arising from the previous theorem:

- In order for the constants L, C_0, C_1, C_2 to exist, we require that the series for $\mathcal{S}_h^*, \Phi_h, S_h^P$ and Ψ_h^P all convergence for sufficiently small h. In some cases, convergence cannot be guaranteed (see e.g. [36, p.576]).

- Throughout we have assumed that nonlinear contributions from the $O(\delta^2)$ terms can be neglected. Note that under the regularity conditions of the above theorem, such an assumption does not compromise the validity of the results, i.e. it also possible to derive bounds on the $O(\delta^2)$ terms.

- The result implies that the GLM solution stays close to the UOSM solution (of a slightly perturbed problem) over the interval $0 \leq t \leq h^{-M}L^{-1}$:

$$\begin{aligned} ||y_n - y(nh)|| &\leq ||\Phi_h^n(y_0 + B(c_1, y_0)) - y(nh)|| + ||y_n - \Phi_h^n(y_0 + B(c_1, y_0))||, \\ &\leq ||\Phi_h^n(y_0 + B(c_1, y_0)) - y(nh)|| + K_2(h, r)\exp(th^M L), \\ &= ||\Phi_h^n(y_0 + B(c_1, y_0)) - y(nh)|| + O(h^{k_1+k_2}). \end{aligned}$$

  A similar result, using backward error analysis, is given in [23].

- If the IVP is Hamiltonian, $||H'(y)|| \leq C_3$, $C_3 > 0$ and $\mathcal{M}_h$ is either symmetric or G-symplectic, then for $0 \leq t \leq h^{-M}L^{-1}$, we have

$$\begin{aligned} |H(y_n) - H(y_0)| &\leq |H(\Phi_h^n(y_0 + B(c_1, y_0))) - H(y_0)| + \\ &\qquad |H(y_n) - H(\Phi_h^n(y_0 + B(c_1, y_0)))|, \\ &\leq |H(\Phi_h^n(y_0 + B(c_1, y_0))) - H(y_0)| + C_3 K_2(h, r)\exp(th^M L), \\ &\leq |H(y_0 + B(c_1, y_0)) - H(y_0)| + O(h^p) + O(h^{k_1+k_2}), \\ &\leq O(h^p) + O(h^{k_1+k_2}), \end{aligned}$$

  where the bound on $H(\Phi_h^n(\cdot))$ is discussed in [23, 24].

- If the pair $(\mathcal{M}_h, \mathcal{S}_h)$ is of order p, then the value of the k_2 exponent is at least $p+1$. This follows from the fact that each $B(c_P, y_0) = O(\delta) = O(h^{p+1})$.

- If the finishing method satisfies $\mathcal{F}_h(y) = w^{\mathsf{H}}y + O(h)$, where w^{H} is a left eigenvector of V, then $k_1 \geq 1$.

3.3.5 Higher-order parasitism-free conditions

It appears that the value of M in assumption (3.6) essentially determines the length of the interval over which the parasitic components remain bounded (cf. experiments in Chapter 7). This observation leads to the following definition.

Definition 3.37. A GLM is Mth-order parasitism-free if $\Psi_h^P(y_0, \cdot) = \zeta_P + O(h^{M+1})$ for all $P \in \{2, \dots, r\}$, where each ζ_P is a distinct eigenvalue of V satisfying $|\zeta_P| \leq 1$, and $\zeta_P \neq 1$.

Theorem 3.38. *For $P \in \{2, \dots, r\}$, define $D_P = (\zeta_P I_r - V + u_P w_P^{\mathsf{H}})^{-1}$, and let*

$$C := \operatorname{diag}(c), \qquad c := A\mathbb{1} + U\mathcal{S}^*(\bullet), \qquad C_2 := \operatorname{diag}(c_2), \qquad c_2 := Ac + U\mathcal{S}^*\left(\begin{smallmatrix}\bullet\\ |\\ \bullet\end{smallmatrix}\right),$$

where $\mathcal{S}^*(\bullet)$ and $\mathcal{S}^*\left(\begin{smallmatrix}\bullet\\ |\\ \bullet\end{smallmatrix}\right)$ respectively correspond to the hf-coefficient and the $h^2f'f$-coefficient of the ideal starting method. Then, a GLM is third-order parasitism-free if the following conditions are met

$$\begin{aligned}
\psi_P\left(\begin{smallmatrix}\times\\ |\\ \bullet\end{smallmatrix}\right) &:= w_P^{\mathsf{H}} BUu_P = 0,\\
\psi_P\left(\begin{smallmatrix}\times\\ |\\ \bullet\\ |\\ \bullet\end{smallmatrix}\right) &:= w_P^{\mathsf{H}} B(A + UD_PB)Uu_P = 0,\\
\psi_P\left(\begin{smallmatrix}\bullet & \times\\ & \diagdown\,\diagup\\ & \bullet\end{smallmatrix}\right) &:= w_P^{\mathsf{H}} BCUu_P = 0,\\
\psi_P\left(\begin{smallmatrix}\times\\ |\\ \bullet\\ |\\ \bullet\\ |\\ \bullet\end{smallmatrix}\right) &:= w_P^{\mathsf{H}} B(A + UD_PB)^2Uu_P = 0,\\
\psi_P\left(\begin{smallmatrix}\bullet & \times\\ & \diagdown\,\diagup\\ & \bullet\\ & |\\ & \bullet\end{smallmatrix}\right) &:= w_P^{\mathsf{H}} B(A + UD_PB)CUu_P - \zeta_P w_P^{\mathsf{H}} BUD_P^2BUu_P = 0,\\
\psi_P\left(\begin{smallmatrix}\times\\ |\\ \bullet & \bullet\\ & \diagdown\,\diagup\\ & \bullet\end{smallmatrix}\right) &:= w_P^{\mathsf{H}} BC(A + UD_PB)Uu_P = 0,\\
\psi_P\left(\begin{smallmatrix}\bullet & \bullet\\ & \diagdown\,\diagup\\ & \bullet\\ & |\\ & \times\end{smallmatrix}\right) &:= w_P^{\mathsf{H}} BC_2Uu_P = 0,\\
\psi_P\left(\begin{smallmatrix}\bullet & \bullet & \times\\ & \diagdown\,|\,\diagup\\ & \bullet\end{smallmatrix}\right) &:= w_P^{\mathsf{H}} BC^2Uu_P = 0,
\end{aligned} \tag{3.7}$$

for all $P \in \{2, \dots, r\}$.

Proof (sketch). For $P \in \{2, \dots, r\}$, let us make the arbitrary choice that $F_h^{(P)}(y_0, \cdot) = w_P^{\mathsf{H}}$, where w_P^{H} is the left eigenvector of V corresponding to eigenvalue ζ_P. Then, we follow the constructive proof of Theorem 3.30 to obtain a set of square systems in terms of $s_P(d\tau)$, $\psi_P(d\tau)$, for all $|d\tau| \leq 3$. As there are 9 derivative trees up to order 3, we must solve a total of 8 systems (since there is no condition corresponding to $d\tau = \times$).

Recall that each system is of the form

$$\begin{bmatrix} \zeta_P I_r - V & u_P \\ w_P^{\mathsf{H}} & 0 \end{bmatrix} \begin{bmatrix} s_P(d\tau) \\ \psi_P(d\tau) \end{bmatrix} = \begin{bmatrix} G(d\tau) \\ g(d\tau) \end{bmatrix}.$$

In particular, our choice of F_h^P ensures $g(d\tau) = 0$ for all $|d\tau| \geq 1$. By inverting the left-hand matrix, we find

$$\begin{bmatrix} s_P(d\tau) \\ \psi_P(d\tau) \end{bmatrix} = \begin{bmatrix} D_P & u_P \\ w_P^{\mathsf{H}} & 0 \end{bmatrix} \begin{bmatrix} G(d\tau) \\ 0 \end{bmatrix}.$$

Thus, $\psi_P(d\tau) = w_P^{\mathsf{H}} G(d\tau)$, where the $G(d\tau)$ are as determined in the proof of Theorem 3.30. Repeating the above process for each $P \in \{2, \ldots, r\}$ yields the complete set of conditions. □

The above result requires $8(r-1)$ conditions to be satisfied to ensure 3rd order parasitism-free behaviour. However, in many cases of practical interest, a reduction occurs.

Theorem 3.39. *Consider a consistent r-input GLM, with real coefficient matrices and V possessing distinct, uni-modular eigenvalues. Then, the total number of conditions required for 3rd order parasitism-free behaviour is equal to*

$$\begin{aligned} 4r \quad & \text{if} \quad r = \text{even}, \\ 4(r-1) \quad & \text{if} \quad r = \text{odd}. \end{aligned}$$

Proof. Since the coefficient matrices of the GLM are real, it follows that $\mathcal{M}_h'(\mathcal{S}_h^*(y_0)) = \overline{\mathcal{M}_h'(\mathcal{S}_h^*(y_0))}$. Thus, if the pair (S_h^P, Ψ_h^P) is a solution to the DUOSM condition (3.4), then so is the pair $(\overline{S_h^P}, \overline{\Psi_h^P})$. Furthermore, if $\Psi_h^P = \zeta_P + O(h^4)$ then we automatically have $\overline{\Psi_h^P} = \overline{\zeta_P} + O(h^4)$. In other words, for every pair of complex conjugate pairs of DUOSMs, we need only satisfy 8 parasitism-free conditions.

Now, consider the case that r is odd. As the eigenvalues of V are unimodular and distinct, it follows that there will be $(r-1)/2$ complex conjugate pairs, with the remaining eigenvalue at $\zeta_1 = 1$ by consistency. Consequently, there will be $(r-1)/2$ complex conjugate pairs of DUOSMs. Thus, we need only satisfy a total of $8(r-1)/2 = 4(r-1)$ conditions.

Similarly for the case r is even, there will be $(r-2)/2$ complex conjugate pairs of DUOSMs, and a single DUOSM corresponding to the eigenvalue $\zeta_2 = -1$. Thus, we will have to satisfy $8(r-2)/2 + 8 = 4r$ conditions.

□

Symmetry: The situation of real coefficient matrices and distinct, uni-modular eigenvalues arises frequently for symmetric GLMs (cf. Section 2.5). In this case, it can be shown that an additional reduction in the number of conditions required for Mth order parasitism-free behaviour occurs.

Theorem 3.40. *Consider a consistent, symmetric GLM with an involution matrix L, real coefficient matrices, and V possessing distinct eigenvalues. If the pair (S_h^P, Ψ_h^P), $P \in \{2, \dots, r\}$, is a DUOSM solution of (3.4), and*

$$L\mathcal{S}_h^*(y_0) = \mathcal{S}_{-h}^*(y_0), \qquad \text{and} \qquad LS_h^P(y_0, \cdot) = \lambda_L \overline{S_{-h}^P(y_0, \cdot)}, \qquad \lambda_L \in \mathbb{C}\backslash\{0\}, \tag{3.8}$$

then

$$\left(\Psi_{-h}^P(\Phi_h(y_0), \cdot)\right)^{-1} = \overline{\Psi_h^P(y_0, \cdot)}.$$

Proof. Recall from Section 2.5 that a GLM is symmetric if $\mathcal{M}_h$ satsifies

$$\mathcal{M}_{-h}^{-1}(y) = L\mathcal{M}_h(Ly), \qquad \text{or alternatively,} \qquad \mathcal{M}_{-h} \circ L\mathcal{M}_h(Ly) = y,$$

for some involution L. Differentiation with respect to y yields the following derivative identities:

$$\left(\mathcal{M}_{-h}^{-1}\right)'(y) = L\mathcal{M}_h'(Ly)L, \qquad \text{and} \qquad \mathcal{M}_{-h}'(L\mathcal{M}_h(Ly))L\mathcal{M}_h'(Ly)L = I.$$

Together, these imply

$$\left\{\left(\mathcal{M}_{-h}^{-1}(y)\right)'\right\}^{-1} = \mathcal{M}_{-h}'(\mathcal{M}_{-h}^{-1}(y)).$$

Now, using the above derivative identities, we find

$$\begin{aligned}
& \mathcal{M}_h'(\mathcal{S}_h^*(y_0))S_h^P(y_0, \cdot) = S_h^P(\Phi_h(y_0), \Psi_h^P(y_0, \cdot)), \\
\Rightarrow \quad & L\left(\mathcal{M}_{-h}^{-1}\right)'(L\mathcal{S}_h^*(y_0))LS_h^P(y_0, \cdot) = S_h^P(\Phi_h(y_0), \Psi_h^P(y_0, \cdot)), \\
\Rightarrow \quad & LS_h^P(y_0, \cdot)\left(\Psi_h^P(y_0, \cdot)\right)^{-1} = \mathcal{M}_{-h}'(\mathcal{M}_{-h}^{-1} \circ L\mathcal{S}_h^*(y_0))LS_h^P(\Phi_h(y_0), \cdot).
\end{aligned}$$

Using the symmetry assumptions (3.8), this becomes

$$\begin{aligned}
\lambda_L \overline{S_{-h}^P(y_0, \cdot)}\left(\Psi_h^P(y_0, \cdot)\right)^{-1} &= \lambda_L \mathcal{M}_{-h}'(\mathcal{M}_{-h}^{-1} \circ \mathcal{S}_{-h}^*(y_0))\overline{S_{-h}^P(\Phi_h(y_0), \cdot)}, \\
&= \lambda_L \mathcal{M}_{-h}'(\mathcal{S}_{-h}^* \circ \Phi_{-h}^{-1}(y_0))\overline{S_{-h}^P(\Phi_h(y_0), \cdot)},
\end{aligned}$$

and letting $h \mapsto -h$, $y_0 \mapsto \Phi_h(y_0)$, we obtain

$$\overline{S_h^P(\Phi_h(y_0),\cdot)}\left(\Psi_{-h}^P(\Phi_h(y_0),\cdot)\right)^{-1} = \mathcal{M}_h'(\mathcal{S}_h^*(y_0))\overline{S_h^P(\Phi_{-h}\circ\Phi_h(y_0),\cdot)}.$$

Symmetry of both the GLM and the ideal starting method implies that the UOSM is symmetric, i.e. $\Phi_{-h}\circ\Phi_h(y_0) = y_0$. Note also that since (S_h^P, Ψ_h^P) is a DUOSM solution, so is $(\overline{S_h^P}, \overline{\Psi_h^P})$, as the GLM coefficient matrices are real. Thus,

$$\overline{S_h^P(\Phi_h(y_0),\cdot)}\left(\Psi_{-h}^P(\Phi_h(y_0),\cdot)\right)^{-1} = \mathcal{M}_h'(\mathcal{S}_h^*(y_0))\overline{S_h^P(y_0,\cdot)} = \overline{S_h^P(\Phi_h(y_0),\cdot)}\ \overline{\Psi_h^P(y_0,\cdot)}.$$

The result then follows after applying $\overline{F_h^P(\Phi_h(y_0),\cdot)}$.

□

Corollary 3.41. *Let $\zeta_2 = -1$ be an eigenvalue of V. Then, $\Psi_h^{(2)}(y_0,\cdot)$ is real. Furthermore, if $\Psi_h^{(2)}(y_0,\cdot) = -1 + O(h^{M+1})$, then M is necessarily even.*

Proof. Since V and ζ_2 are both real, it follows that the corresponding left and right eigenvectors w_2, u_2 are also real. Now, from the proof of Theorem 3.30, the coefficients of $\Psi_h^{(2)}(y_0,\cdot)$ are found by solving the system

$$\begin{bmatrix} \zeta_2 I_r - V & u_2 \\ w_2^{\mathsf{H}} & 0 \end{bmatrix}\begin{bmatrix} s_2(d\tau) \\ \psi_2(d\tau) \end{bmatrix} = \begin{bmatrix} G(d\tau) \\ g(d\tau) \end{bmatrix},$$

for each $|d\tau| \geq 1$. Given that the left-hand matrix is real-valued, and that the coefficient matrices of the method are also real-valued, it follows that each $\psi_2(d\tau)$ is real (see e.g. the conditions for third-order parasitism-free behaviour (3.7)).

Now suppose that $\Psi_h^{(2)}(y_0,\cdot) = -1 + \psi_2^{(k+1)}(y_0)h^{k+1} + O(h^{k+2})$, where k is odd and $\psi_2^{(k+1)}(y_0)$ is a constant depending the elementary Jacobians corresponding to derivative trees of order $k+1$, evaluated at y_0. Then,

$$\begin{aligned} 1 = \Psi_{-h}^{(2)}(\Phi(y_0),\cdot)\Psi_h^{(2)}(y_0,\cdot) &= (-1 + \psi_2^{(k+1)}(\Phi_h(y_0))h^{k+1})\cdot \\ &\qquad (-1 + \psi_2^{(k+1)}(y_0)h^{k+1}) + O(h^{k+2}), \\ &= 1 - 2\psi_2^{(k+1)}(y_0)h^{k+1} + O(h^{k+2}), \end{aligned}$$

where we have used $\Phi_h(y_0) = y_0 + O(h)$. Comparing powers of h gives $\psi_2^{(k+1)}(y_0) = 0$. Thus, the result follows after we define $M = k+1$. □

The above theorem implies that for symmetric GLMs only conditions corresponding to derivative trees of odd order need to be satisfied. Furthermore, if the method has only two inputs, only 6 conditions are required for 4th order parasitism-free behaviour.

Example 3.42. Consider the symmetry matrices

$$L = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \qquad P_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \qquad P_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

The GLM given below is 4th order and (L, P_1)-symmetric:

$$\left[\begin{array}{cccc|cc} \frac{1}{2} & 0 & 0 & 0 & 1 & 1 \\ \frac{1}{2} & \frac{1}{2}6^{1/3} & 0 & 0 & 1 & 0 \\ \frac{1}{2} & 6^{1/3} & -6^{1/3} & 0 & 1 & 0 \\ \frac{1}{2} & 6^{1/3} & -2\cdot 6^{1/3} & \frac{1}{2}6^{1/3} & 1 & 0 \\ \hline 1 & 6^{1/3} & -2\cdot 6^{1/3} & 6^{1/3} & 1 & 0 \\ 0 & 6^{1/3} & -2\cdot 6^{1/3} & 6^{1/3} & 0 & -1 \end{array}\right]. \tag{3.9}$$

Another 4th order example, with rational coefficients, is given by the (L, P_2)-symmetric GLM below:

$$\left[\begin{array}{ccccc|cc} \frac{1}{2} & 0 & 0 & 0 & 0 & 1 & 1 \\ \frac{7}{12} & \frac{5}{24} & 0 & 0 & 0 & 1 & 0 \\ -\frac{1}{12} & \frac{1}{2} & \frac{1}{24} & 0 & 0 & 1 & 0 \\ \frac{13}{12} & \frac{1}{2} & -\frac{1}{2} & -\frac{13}{24} & 0 & 1 & 0 \\ \frac{5}{12} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{7}{24} & 1 & 0 \\ \hline 1 & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & 1 & 0 \\ 0 & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & 0 & -1 \end{array}\right]. \tag{3.10}$$

Substituting each set of coefficient matrices into (3.7), and noting that $w_2 = u_2 = [0, 1]^T$, we find that both methods satisfy the order conditions for 3rd-order parasitism-free behaviour. It follows from the symmetry of the methods and Corollary 3.41 that both are actually 4th-order parasitism-free.

In Chapter 7, several numerical experiments are performed on GLM (3.10) to demonstrate its 4th-order parasitism-free behaviour.

◇

Chapter 4

Practical toolkit

In this chapter we discuss the development of a set of computational tools for assisting the study of GLMs. These tools have been designed using an object-oriented approach to programming. An important aspect of this has been deciding suitable machine representations for the set of rooted and derivative trees; B-series and DB-series, and GLMs themselves. Similar developments in this area of computer-aided analysis include Ketcheson's *NodePy* Python package [41] for analysing and testing numerical methods on IVPs, and the Wolfram Mathematica *Numerical Differential Equation Analysis Package* [62] which is used in the study of RKMs. Note that while these packages provide similar tools for analysis, the code we have developed here differs in that it has been designed specifically for GLMs and the manipulation of B-series and DB-series. As a by-product of this approach, we automatically obtain computational tools for the analysis of RKMs, LMMs and other more exotic numerical methods.

4.1 Machine representation of trees

As a typical GLM analysis involves the manipulation of various B-series or DB-series expressions, we begin by discussing the machine representation of trees. In particular, those belonging to either the set of rooted trees T or the set of derivative trees DT.

Recall that the set T is defined recursively as follows:

$$\begin{aligned} \text{let} \quad & \tau = \bullet \in T, \\ \text{then also} \quad & \tau = [\tau_1, \ldots, \tau_m] \in T, \qquad \text{where} \qquad \tau_1, \ldots, \tau_m \in T. \end{aligned}$$

By this definition, it naturally follows that a rooted tree should be written as a string consisting of only commas, dots, and left and right square brackets. A similar observa-

tion can be made for derivative trees: Recall from Section 3.2.1 that, for a given $u \in T$, the set of derivative trees $D_\tau(u)$ is given by appending a $\times$ node to each $\bullet$ node of u. Thus, we see that a derivative tree string can be written using the same characters of a rooted tree with the addition of a single `'x'`.

These tree definitions also imply the following set of rules should be obeyed when constructing strings:

- All strings begin with `'['` and end with `']'`, except for `'.'`, or `'x'`.
- There must be an equal number of `'['` and `']'` characters.
- Each derivative tree must contain exactly one `'x'` character.
- `'.'` follows either `','` or `'['`.
- `','` follows either `'.'` or `'x'` or `']'`.
- `']'` follows either `'.'` or `'x'` or `']'`.
- `'['` follows either `','` or `'['`.
- `'x'` follows either `','` or `'['`.

Following these rules, we find that the set of rooted trees, up order 4, are given by the strings

```
''    '.'    '[.]'    '[[.]]'    '[.,.]'
'[[[.]]]'    '[[.,.]]'    '[.,[.]]'    '[.,.,.]'
```

where the empty string represents the empty tree. Similarly, the set of derivative trees, up to order 3, are given by the strings

```
'x'    '[x]'    '[[x]]'    '[x,.]'
'[[[x]]]'    '[[x,.]]'    '[[x],.]'    '[x,[.]]'    '[x,.,.]'
```

Note here that the action of appending a $\times$ node to a $\bullet$ node is equivalent to replacing either a `'['` by `'[x,'`, or a `'.'` by `'[x]'`.

Tree operations: Earlier in Section 2.2 and throughout Chapter 3 we introduced a variety of operations that are performed on trees. By taking the string representation of a tree, we have been able to implement these operations in MATLAB. A list of those common to both rooted and derivative trees is given in Table 4.1. Operations unique to rooted and derivative trees are given respectively in Tables 4.2 and 4.3.

Command	Description	Output
`trees.children(t)`	Extract the children of `t`.	cell (strings)
`trees.compare(t1,t2)`	Check whether `t1` and `t2` are equivalent. Note, two trees are equivalent if they share the same children.	logical
`trees.bprod(t1,t2)`	Perform a Butcher product on `t1` and `t2`.	string
`trees.order(t)`	Compute the order of `t`.	integer

Table 4.1: Fundamental tree operations implemented in MATLAB.

Command	Description	Output
`trees.symmetry(t)`	Compute the symmetry of `t`.	integer
`trees.density(t)`	Compute the density of `t`.	integer

Table 4.2: Rooted tree operations implemented in MATLAB.

Command	Description	Output
`trees.underlying(dt)`	Compute the underlying rooted tree of `dt`.	string
`trees.multiplicity(dt)`	Compute the multiplicity of `dt`.	integer
`trees.substitution(du,v)`	Perform a substitution of `v` into `du` .	string

Table 4.3: Derivative tree operations implemented in MATLAB.

Examples on the usage of these commands are given below:

Command:	Output:
`trees.children('[[.,[.]],[.,.]]')`	`{'[.,[.]]','[.,.]'}`
`trees.compare('[[.],.]','[.,[.]]')`	`1`
`trees.compare('.','[.,.]]')`	`0`
`trees.bprod('.','[.]')`	`'[[.]]'`
`trees.bprod('[.]','.')`	`'[.,.]'`
`trees.order('[[.,[.]],[.,.]]')`	`8`
`trees.symmetry('[[.,.]]')`	`2`
`trees.density('[[.,.]]')`	`12`
`trees.underlying('[[x,.]]')`	`'[[.]]'`
`trees.multiplicity('[[x],.]')`	`2`
`trees.substitution('[x,.]','[.]')`	`'[[.],.]'`

Lookup tables: Several of the tree operations mentioned above are defined recursively. As a result, they often repeat many of the same computations. For example, to determine the symmetry of the tree-string `'[[.],[.]]'` we must also determine the symmetry of the tree-string `'[.]'`, which ends up being computed twice.

In order to avoid this type of redundancy, we have constructed the lookup table `rooted_trees` that contains precomputed lists of the various quantities associated with rooted trees:

```
>> S = load('rooted_trees.mat'); disp(S)
           index: [487x1 double]
           order: [487x1 double]
            tree: {487x1 cell}
        children: {487x1 cell}
    children_idx: {487x1 cell}
        symmetry: [487x1 double]
         density: [487x1 double]
            fstr: {487x1 cell}
```

This approach has also been applied to derivative trees where precomputed data is stored in the lookup table `derivative_trees`:

```
>> S = load('derivative_trees.mat'); disp(S)
           index: [1141x1 double]
           order: [1141x1 double]
            tree: {1141x1 cell}
 underlying_tree: {1141x1 cell}
  underlying_idx: [1141x1 double]
   rchildren_idx: {1141x1 cell}
   dchildren_idx: {1141x1 cell}
        dbxb_idx: {1141x1 cell}
       dbxdb_idx: {1141x1 cell}
   transpose_idx: [1141x1 double]
    multiplicity: [1141x1 double]
            fstr: {1141x1 cell}
```

In the absence of a well-defined ordering on either set of trees, it was necessary that we build an index to uniquely associate each tree to a positive integer. This was achieved using an iterative procedure.

As is indicated in the tables above, we currently hold information on 487 rooted trees and 1141 derivative trees (that is, on rooted trees up to order 9 and derivative trees up to order 8). Note that equivalent trees have been removed from both sets.

4.2 Object representation for B-series and DB-series

Our approach to the machine representation of B-series and DB-series is to consider them both as objects, i.e. data-structures with certain *properties* and *methods*. This approach has many advantages, including the ability to customise programming syntax which can be used to simplify the process of translating mathematical expressions into the appropriate programming language.

4.2.1 Representation

First, let us consider the object representation of a B-series: Recall that, for some mapping $a : T \cup \{\emptyset\} \to \mathbb{C}$, a B-series is defined as

$$B(a, y) = a(\emptyset)y + \sum_{\tau \in T} \frac{h^{|\tau|}}{\sigma(\tau)} a(\tau) F(\tau)(y).$$

In practice, we are usually only interested in a finite set of coefficients and tend to neglect those associated with trees above a certain order. In other words, for some $q \in \mathbb{N}$, we consider a truncated B-series given by

$$B_q(a, y) = a(\emptyset)y + \sum_{|\tau| \leq q} \frac{h^{|\tau|}}{\sigma(\tau)} a(\tau) F(\tau)(y).$$

This series leads us to define a (truncated) B-series object as a data-structure with the following properties:

- `coeffs`: a complex-valued row vector corresponding to the coefficients $a(\tau)$.
- `sym_coeffs`: a complex-valued row vector corresponding to the coefficients $\dfrac{a(\tau)}{\sigma(\tau)}$.
- `truncation`: a positive integer specifying the order of truncation.
- `num_coeffs`: an integer value for the number of coefficients associated with the order of truncation.

Here, it is important to note that each element of a coefficient vector uniquely corresponds to a tree in $T \cup \{\emptyset\}$. For example, `coeffs(5)` is the B-series coefficient corresponding to the 5th tree in $T \cup \{\emptyset\}$, as ordered in the lookup table `rooted_trees`. That is, it corresponds to the coefficient $a($  $)$.

The approach used above also applies to derivative B-series, i.e. we seek an object representation of the truncated DB-series

$$DB_q(a, y, v) = \sum_{|d\tau| \leq q} h^{|\tau|} a(\tau) J(\tau)(y, v),$$

where $a : DT \to \mathbb{C}$ and $q \in \mathbb{N}$. This leads us to define the series as a data-structure with the properties `coeffs`, `truncation` and `num_coeffs`, as described above. Again, each element of the coefficient vector maps to a unique tree, this time in DT.

Construction: Displayed in Table 4.4 is a list of MATLAB commands for the construction of B-series and DB-series objects. This list also includes special constructors for the commonly occurring B-series such as

$$\text{zero: } B(a, y_0) = 0, \qquad \text{trivial: } B(a, y_0) = y_0, \qquad \text{evolution: } B(a, y_0) = \varphi_h(y_0),$$

and for DB-series such as

$$\text{zero: } DB(a, y_0, v) = 0, \qquad \text{trivial: } DB(a, y_0, v) = v.$$

Example 4.1. Suppose we would like to express the following B-series as an object:

$$y + hf(y) + \frac{1}{2}h^2 f'(y)f(y) + \frac{1}{6}h^3 f'(y)f'(y)f(y) + \frac{1}{6}h^3 f''(y)(f(y), f(y)).$$

While this expression is finite, it can also be interpreted as an infinite series with the higher order coefficients set to zero. Thus, we must still specify the order to which we truncate this series. Here, we arbitrarily set this to be order 4.

Next, we extract the B-series coefficients and store them in a row vector (ordered such that they agree with the index specified in `rooted_trees`). Since there are 9 trees up to order 4, we find that this vector is given by $[1, 1, 1/2, 1/6, 1/3, 0, 0, 0, 0]$. Now, we create the object using the following command

Command	Description	Output
`bseries(a)`	Construct a B-series from the row vector of coefficients `a`.	B-series
`bseries('zero',q)`	Construct the zero B-series truncated to order `q`.	B-series
`bseries('trivial',q)`	Construct the trivial B-series truncated to order `q`.	B-series
`bseries('evolution',q)`	Construct a B-series corresponding to the time-h evolution, truncated to order `q`.	B-series
`dbseries(a)`	Construct a DB-series from the row vector of coefficients `a`.	DB-series
`dbseries('zero',q)`	Construct the zero DB-series truncated to order `q`.	DB-series
`dbseries('trivial',q)`	Construct the trivial DB-series truncated to order `q`.	DB-series

Table 4.4: MATLAB Commands for constructing B-series and DB-series objects.

```
>> B = bseries([1,1,1/2,1/6,1/3,0,0,0,0]); disp(B)
  bseries with properties:

       coeffs: [1 1 1/2 1/6 1/3 0 0 0 0]
    sym_coeffs: [1 1 1/2 1/6 1/6 0 0 0 0]
    truncation: 4
    num_coeffs: 9
```

Note that if we specified a 3rd order truncation, we could have alternatively generated the B-series using the command `bseries('evolution',3)`.

◇

Example 4.2. Using the commands in Table 4.4 we can also represent vector B-series as objects: A trivial vector B-series example is given by

$$\begin{bmatrix} y \\ 0 \end{bmatrix}.$$

As in the previous example, we arbitrarily set the order of truncation to 4. Then, the object is built using the following commands:

```
>> B1=bseries('trivial',4);
>> B2=bseries('zero',4);
>> vecB = [B1;B2];
```

◇

4.2.2 Operations

The next stage in the development of an object is the implementation of its operations (or rather, its *methods* to use object-oriented terminology). For B-series and DB-series these include the elementary algebraic operations, expansions, compositions and inversion, to name but a few. Below, we discuss some of the implementation details behind these operations. The corresponding MATLAB commands can be found in Tables 4.5 and 4.6.

Algebraic operations: Since both series are linear in their first argument, we have chosen to refine the `+` and `-` operators to reflect this property. In addition, we have also redefined the `*` operator to allow for the left multiplication by a scalar or matrix. Note: caution is advised when dealing with several series where the y-arguments differ as there is currently no way to distinguish between them.

B-series composition: An expression of the form $B(a, B(b, y))$ is called a B-series composition and these operations typically arise when attempting to determine the order of a numerical method. The problem of implementing this operation for B-series objects can be approached in several of ways. For example, we could directly compute the coefficients from known composition formulae (see e.g. [36, p. 64]). However, to do this for all trees in the lookup table `rooted_trees` would require 487 individual formulae to be computed.

Instead, we choose to reformulate the composition operation as a matrix-vector multiplication: Let us begin by writing the first few terms of $B(a, B(b, y))$ out explicitly,

$$B(a, B(b, y)) = a(\emptyset)B(b, y) + h\frac{a(\bullet)}{\sigma(\bullet)}f(B(b, y)) + h^2\frac{a(\bullet\!\!-\!\!\bullet)}{\sigma(\bullet\!\!-\!\!\bullet)}f'(B(b, y))f(B(b, y)) + O(h^3).$$

Command	Description	Output
`scale(Ba,x)`	Scale h by the constant $x \in \mathbb{R}$.	B-series
`hf(Ba)`	Compute the B-series $hf(B(a,y))$. Note, we must have $a(\emptyset) = 1$.	B-series
`compose(Ba1,Ba2)`	Compute the B-series of the composition $B(a_1, B(a_2, y))$. Note, we must have $a_2(\emptyset) = 1$.	B-series
`inverse(Ba)`	Compute the inverse B-series $B(a^{-1}, y)$. Note, we must have $a(\emptyset) = 1$.	B-series
`diff(Ba)`	Differentiate a B-series, i.e. compute $\nabla_y B(a,y)$.	DB-series

Table 4.5: B-series operations in MATLAB.

Command	Description	Output
`scale(DBa,x)`	Scale h by the constant $x \in \mathbb{R}$.	DB-series
`hdf(Ba)`	Compute the DB-series $hf'(B(a,y))$. Note, we must have $a(\emptyset) = 1$.	DB-series
`compose(DBa1,Ba2)`	Compute the DB-series of the composition $DB(a_1, B(a_2, y), v)$. Note, we must have $a_2(\emptyset) = 1$.	DB-series
`inverse(DBa)`	Compute the inverse DB-series $DB(a^{-1}, y, v)$. Note, we must have $a(\bullet) = 1$.	DB-series
`sub(DBa1,DBa2)`	Compute the product of two DB-series $DB(a_1, y, DB(a_2, y, v))$.	DB-series
`sub(DBa1,Ba2)`	Compute the product of a DB-series and B-series, i.e. $DB(a_1, y, B(a_2, y))$. Note, we must have $a_2(\emptyset) = 0$.	B-series

Table 4.6: DB-series operations in MATLAB.

Observe that we can equivalently express this as the inner product

$$B(a, B(b,y)) = \begin{bmatrix} a(\emptyset) & a(\bullet) & a(\begin{smallmatrix}\bullet\\\bullet\end{smallmatrix}) & \cdots \end{bmatrix} \begin{bmatrix} B(b,y) \\ \frac{h}{\sigma(\bullet)} f(B(b,y)) \\ \frac{h^2}{\sigma(\begin{smallmatrix}\bullet\\\bullet\end{smallmatrix})} f'(B(b,y)) f(B(b,y)) \\ \vdots \end{bmatrix}. \tag{4.1}$$

For this composition to hold, we must have $B(b, y) = y + O(h)$. Assuming this to be true, we perform a Taylor series expansion on the elementary differentials, about y, to obtain

$$B(a, B(b, y)) = \begin{bmatrix} a(\emptyset) & a(\bullet) & a(\begin{smallmatrix}\bullet\\|\\\bullet\end{smallmatrix}) & \cdots \end{bmatrix} C(b) \begin{bmatrix} y \\ \frac{h}{\sigma(\bullet)} f(y) \\ \frac{h^2}{\sigma(\begin{smallmatrix}\bullet\\|\\\bullet\end{smallmatrix})} f'(y)f(y) \\ \vdots \end{bmatrix},$$

where $C(b)$ is an upper-triangular matrix with elements depending on the $b(\tau)$. The B-series coefficients of the composed method are then given by a multiplication of the vector of a-coefficients and the $C(b)$ matrix.

Constructing $C(b)$: Note the following observation: Consider the jth element in the column vector of (4.1). This element is a B-series, and may written as a kth-order derivative of f acting on k-many (possibly different) B-series. Each of which also belong to the column vector, and have an associated index that is strictly less than j. In other words, each row of $C(b)$ corresponds to a set of coefficients describing the B-series

$$h^k f^{(k)}(B(b, y))(B(c_1, y), \ldots, B(c_k, y)), \qquad \text{for some} \qquad k \in \mathbb{N}_0, \tag{4.2}$$

where the coefficients of $B(c_1, y), \ldots, B(c_k, y)$ are each given by some row in $C(b)$ (with row-index less than j).

This observation leads us to define the following algorithm for constructing $C(b)$:

---

1. Set the first row equal to the coefficients of b.
2. Determine the coefficients for the second row, i.e. for $hf(B(b, y))$, up to the order of truncation.
3. Loop over each remaining row in the order $j = 3, 4, 5, \ldots$, and
   (a) for row j, determine the arguments of (4.2), i.e. find the rows corresponding to the $c_1, \ldots, c_k$,
   (b) compute the coefficients of (4.2), up to the order of truncation.

For step 2, recall that the coefficients for this B-series are given by the formula described in Lemma 3.4. This result has been implemented for B-series objects and may be applied using the command `hf`.

For step 3(a), we note that the jth row corresponds to the jth tree in the lookup table `rooted_trees`. Thus, indexes for the children of this tree directly correspond to the row indexes for $c_1, \ldots, c_k$.

For step 3(b), we have written a code `DFxBk` that computes the B-series coefficients of $h^k f^{(k)}(y)(B(c_1, y), \ldots, B(c_k, y))$, where $B(c_1, y), \ldots, B(c_k, y)$ are given as inputs. Then, by performing a Taylor series expansion, we can compute (4.2) via

$$\sum_{i \geq 0} h^{i+k} \frac{1}{i!} f^{(i+k)}(y)(B(c_1, y), \ldots, B(c_k, y), B^i(c_0, y)),$$

where $B^i(c_0, y)$ should be read as i-many copies of $B(c_0, y) := B(b, y) - y$.

The above algorithm has been successfully implemented in MATLAB and can be used to construct the $C(b)$ matrix using the command `constructBCMatrix`. The complete composition operation is given by the command `compose`.

Example 4.3. Recall that the evolution operator of an ODE, φ_t, possesses the symmetry property $\varphi_{-t} \circ \varphi_t(y_0) = y_0$, $t \in \mathbb{R}$. This can be verified (up to the order of truncation) using the B-series operations given in Table 4.5:

```
>> eh = bseries('evolution',4);
>> emh = scale(eh,-1); disp(emh)
  bseries with properties:

        coeffs: [1 -1 1/2 -1/6 -1/3 1/24 1/12 1/8 1/4]
    sym_coeffs: [1 -1 1/2 -1/6 -1/6 1/24 1/24 1/8 1/24]
    truncation: 4
    num_coeffs: 9

>> C = constructBCMatrix(eh.coeffs); disp(C)
     1     1    1/2    1/6    1/3   1/24   1/12    1/8    1/4
     0     1     1    1/2     1    1/6    1/3    1/2     1
     0     0     1     1      2    1/2     1     3/2     3
     0     0     0     1      0     1      2      1      0
     0     0     0     0      1     0      0      1      3
     0     0     0     0      0     1      0      0      0
     0     0     0     0      0     0      1      0      0
     0     0     0     0      0     0      0      1      0
     0     0     0     0      0     0      0      0      1
```

```
>> disp(compose(emh,eh))
  bseries with properties:

        coeffs: [1 0 0 0 0 * * * 0]
    sym_coeffs: [1 0 0 0 0 * * * 0]
    truncation: 4
    num_coeffs: 9
```

Here, we notice that the vector `emh.coeffs` is orthogonal to columns $2-9$ of `C`. Thus, the composition `compose(emh,eh)` corresponds to the trivial B-series truncated to order 4, as expected. (MATLAB note: the `*` symbol seen above denotes a rational approximation to zero due to rounding error).

◇

DB-series composition: An expression of the form $DB(a, B(b, y), v)$ is called a DB-series composition. The approach we have taken in implementing this operation is identical to that described for B-series. That is, we reformulate the operation as a matrix-vector product where the matrix is constructed using the command `constructDBCMatix` and the vector is given by the a-coefficients.

DB-series substitution: Expressions of the form

$$DB(c, y) = DB(a, y, DB(b, y, v)) \qquad \text{and} \qquad B(c, y) = DB(a, y, B(b, y)),$$

are called substitutions or products. These were considered in Section 3.2.3, where the formulae for computing the coefficients were respectively found to be

$$c(d\tau) = \sum_{du \otimes dv = d\tau} a(du)b(dv) \qquad \text{and} \qquad \frac{c(\tau)}{\sigma(\tau)} = \sum_{du \otimes v = \tau} \frac{a(du)b(v)}{\sigma(v)}.$$

As with the composition operations, these substitutions can be viewed as a matrix-vector multiplication. Here, the matrix is constructed from special indexes stored in the lookup table `derivative_trees`. In particular, associated with each derivative tree is an index for building a single row of the matrix. If we are considering a $DB(a, y, DB(b, y, v))$ substitution, then these indexes are found in the field `dbxdb_idx`. For $DB(a, y, B(b, y, v))$ substitutions, these indexes are given in the field `dbxb_idx`. Either type of substitution can be applied using the command `sub(Ba,Bb)`.

B-series differentiation: The differentiation of a B-series $B(a, y)$ was considered in Section 3.2.3 where the formula for the DB-series coefficients was found to be

$$b(d\tau) = \frac{a(o_\tau(d\tau))}{\sigma(o_\tau(d\tau))}\nu(d\tau).$$

The implementation of this operation is relatively simple as, for each derivative tree, we only require information on the index and symmetry of its underlying rooted tree and its multiplicity. This information is readily accessible in lookup tables `rooted_trees` and `derivative_trees`.

Example 4.4. Recall from Example 4.3 that the ODE evolution operator satisfies the symmetry property $\varphi_{-t}\circ\varphi_t(y_0) = y_0$. Fixing $t = h$, and differentiating with respect to y_0, we find

$$\frac{\partial\varphi_{-h}}{\partial y_0}(\varphi_h(y_0))\frac{\partial\varphi_h}{\partial y_0}(y_0) = I.$$

This result can be verified (up to the order of truncation) using the operations described above: Choosing a B-series truncation of order 9 (and consequently a DB-series truncation of order 8), we find

```
>> eh = bseries('evolution',9);
>> emh = eh.scale(-1);
>> deh = diff(eh);
>> demh = diff(emh);
>> test = sub(compose(demh,eh),deh) - dbseries('trivial',8);
>> disp(norm(test.coeffs,1))
       1/152496068182400
```

Thus, neglecting the effects of rounding error, we confirm that the result holds (up to a truncation of order 8).

◇

Inverse B-series: An advantage to using the matrix-vector formulation for the composition operation is the ability to compute B-series inverses: Let $a^{-1} : T \cup \{\emptyset\} \to \mathbb{C}$ be a mapping describing the coefficients of the inverse of the B-series $B(a, y)$, where $a(\emptyset) = 1$. Then, $B(a^{-1}, B(a, y)) = y$. In matrix-vector form this corresponds to

$$\begin{bmatrix} a^{-1}(\emptyset) & a^{-1}(\bullet) & a^{-1}(\!\!\begin{smallmatrix}\bullet\\|\\\bullet\end{smallmatrix}\!\!) & \cdots \end{bmatrix} C(a) = \begin{bmatrix} 1 & 0 & 0 & \cdots \end{bmatrix}.$$

As both a and $C(a)$ are known, the inversion operation amounts to performing a back-solve on an upper-triangular matrix to determine the coefficients for $a^{-1}(\tau)$.

Inverse DB-series: While a DB-series inverse can be computed using a similar approach to that described above, we have found that a fixed-point iteration is more efficient, particularly when we consider the inverse of a matrix DB-series. This iteration is described as follows: For a given DB-series where $a(\times) \neq 0$, let `x = 1/coeffs(1)` and `rhs = dbseries('trivial',q)` for some $q \in \mathbb{N}$. Then, the following code sample will generate the DB-series inverse,

```
inv = x*rhs;
for i = 1:q
    e = sub(inv,obj)-rhs;
    inv = inv - x*e;
end
```

4.3 Object representation for GLMs

Having discussed the implementation of B-series and DB-series objects, we now move on to consider the representation of GLMs on a machine. Here, we continue to use the idea of objects for representing these methods.

4.3.1 Representation

As GLMs are characterised by their coefficient matrices, we define a GLM object to be a data-structure with the following properties:

- `A`: a real-valued square matrix corresponding to the A matrix.
- `U`: a complex-valued matrix corresponding to the U matrix.
- `B`: a complex-valued matrix corresponding to the B matrix.
- `V`: a complex-valued matrix corresponding to the V matrix.

In addition to the above, we also define several *dependent* properties, i.e. those that depend directly on the coefficient matrices:

- `stagetype`: a string describing the structure of the stage matrix, e.g.

  `'empty'` `'explicit'` `'diagonal'` `'implicit'`

Command	Description	Output
`glm(A,U,B,V)`	Construct a GLM object for the coefficient matrices (A, U, B, V).	`glm`
`starter(As,Bs,u)`	Construct a starting method object for the coefficient matrices (A_S, B_S, u).	`starter`
`starter(u)`	Construct the trivial starting method object for $\mathcal{S}_h = u$.	`starter`
`finisher(Af,Uf,Bf,w)`	Construct a finishing method object for the coefficient matrices (A_f, U_f, B_f, w).	`finisher`
`finisher(sh,w)`	Construct a finishing method object that satisfies the condition $\mathcal{F}_h \circ \mathcal{S}_h(y_0) = y_0$, where $\mathcal{S}_h$ is given by the starting method object `sh`. The vector `w` should be chosen to satisfy the preconsistency condition $w^{\mathsf{H}} u = 1$.	`finisher`
`finisher(w)`	Construct the trivial finishing method object for $\mathcal{F}_h = w^{\mathsf{H}}$.	`finisher`

Table 4.7: GLM constructors implemented in MATLAB.

- `stages`: a non-negative integer specifying the number of stages.
- `inputs`: a non-negative integer specifying the number of inputs.
- `outputs`: a non-negative integer specifying the number of outputs.

The creation of a GLM object is performed by a call to one of the constructors given in Table 4.7. Also in this table are constructors for starting and finishing methods. As objects, these methods fit the classification of a GLM, i.e. they share the properties given above. However, they differ in that their coefficient matrices are subject to certain restrictions. For example, a starting method object must have the `V` matrix as a column vector, and the `U` matrix as a column vector of ones.

Example 4.5. Let us consider the machine representation of the Leapfrog method (2.4) with its Euler starting method. Recall that the tableaux for these methods are respectively given as

$$\left[\begin{array}{c|cc} 0 & 0 & 1 \\ \hline 0 & 0 & 1 \\ 2 & 1 & 0 \end{array}\right], \qquad \left[\begin{array}{c|c} 0 & 1 \\ \hline 0 & 1 \\ 1 & 1 \end{array}\right].$$

Recall also that the finishing method is defined as $\mathcal{F}_h(y) = e_1^T y$, i.e. the first input. Now, we build the Leapfrog method by passing its coefficient matrices to the `glm` constructor:

```
>> mh = glm(0,[0,1],[0;2],[0,1;1,0]); disp(mh)
  glm with properties:

          A: 0
          U: [0 1]
          B: [2x1 double]
          V: [2x2 double]
  stagetype: 'explicit'
     stages: 1
     inputs: 2
    outputs: 2
```

Similarly for the starting method, we pass only its `A`, `B`, `V` matrices to the `starter` constructor:

```
>> sh = starter(0,[0;1],[1;1]); disp(sh)
  starter with properties:

            A: 0
            U: 1
            B: [2x1 double]
            V: [2x1 double]
    stagetype: 'explicit'
       stages: 1
       inputs: 1
      outputs: 2
```

Command	Description	Output
`scale(mh,x)`	Compute the GLM object for $\mathcal{M}_{xh}$, $x \in \mathbb{R}$.	GLM
`compose(mhn,...,mh1)`	Compute the GLM resulting from the composition $\mathcal{M}_h^{(n)} \circ \cdots \circ \mathcal{M}_h^{(1)}$.	GLM
`inverse(mh)`	Compute the GLM inverse. Note, V must be invertible.	GLM
`map(mh,y)`	Evaluate the expression $\mathcal{M}_h(y)$, for the given B-series input y.	(Vector) B-series
`diff(mh,y)`	Evaluate the Fréchet derivative $\mathcal{M}_h'(y)$, for the given B-series input y.	(Matrix) DB-series

Table 4.8: GLM operations in MATLAB.

Finally, the finishing method is created by passing e_1^T to the `finisher` constructor:

```
>> fh = finisher([1,0]); disp(fh)
  finisher with properties:

            A: []
            U: [0x2 double]
            B: [1x0 double]
            V: [1 0]
    stagetype: 'empty'
       stages: 0
       inputs: 2
      outputs: 1
```

◇

4.3.2 Operations

Below, we discuss the implementation of various operations that are performed on GLMs. A compact list of these operations is given in Table 4.8.

Algebraic operations: The only algebraic operation we consider is the left and right multiplication by a matrix. This is motivated by the T-equivalent representation of a GLM: Recall that for some invertible transformation $T \in \mathbb{C}^{r\times r}$, the maps

$$\mathcal{M}_h \qquad \text{and} \qquad T^{-1}\mathcal{M}_h \circ T$$

are said to be equivalent. Thus, in our implementation, the rules for multiplication are defined as follows: A right multiplication by some complex-valued matrix X implies that $V \mapsto VX$ and $U \mapsto UX$, and a left multiplication by X implies $B \mapsto XB$ and $V \mapsto XV$.

Evaluation: A common operation in GLM analysis is the evaluation of the method map $\mathcal{M}_h(y)$, where y is an approximation to some vector B-series. Our implementation of this operation is broken down into two steps: Firstly, we compute a B-series solution Y to the stage equation,

$$Y = hAF(Y) + Uy.$$

This is performed using a fixed-point iteration described by the code sample below:

```
q = y.truncation;
Y0 = U*y;
for i = 1:q
    Y = Y0 + A*hf(Y);
end
```

The second step is then to the compute the update, i.e. `y = V*y + B*hf(Y)`, which gives a vector B-series output. This entire operation can be performed using the command `map(mh,y)` where `mh` is a given GLM object and `y` is a vector B-series object.

Scale: Expressions that involve a re-scaling of the time-step, i.e. $\mathcal{M}_h \mapsto \mathcal{M}_{xh}$, $x \in \mathbb{R}$ can also be considered. Here, we note that scaling h implies that we scale the A and B coefficient matrices of the GLM. Thus, in our implementation we simply update these matrices by multiplying them by the specified scaling constant.

Composition: Recall that the composition of two GLMs, $\mathcal{M}_h^{(2)} \circ \mathcal{M}_h^{(1)}$ is again a GLM with coefficient matrices determined by the composition formula (2.11):

$$\left[\begin{array}{cc|c} A^{(1)} & 0 & U^{(1)} \\ U^{(2)}B^{(1)} & A^{(2)} & U^{(2)}V^{(1)} \\ \hline V^{(2)}B^{(1)} & B^{(2)} & V^{(2)}V^{(1)} \end{array}\right].$$

This operation is relatively straightforward to reproduce for GLM objects: First, we extract the coefficient matrices from the two objects to be composed. Then, we compute the new coefficient matrices from the formula given above. The output method is then created by calling the GLM constructor on these matrices.

Inverse: Recall that the inverse of a GLM is given by the tableau (2.19):

$$\left[\begin{array}{c|c} A - UV^{-1}B & UV^{-1} \\ \hline -V^{-1}B & V^{-1} \end{array}\right].$$

Our implementation of this operation is identical to the approach used for composition, i.e. compute the new matrices from the given formula, then call the constructor on these matrices to build the object output.

Differentiation: The Fréchet derivative of a GLM, $\mathcal{M}_h'(y)$, can also be evaluated for a given B-series input: The derivative is defined by the following equations

$$\begin{aligned} Y &= hAF(Y) + Uy, \\ \mathcal{M}_h'(y) &= V + hBF'(Y)\left(I - hAF'(Y)\right)^{-1} U. \end{aligned}$$

To construct this as a matrix DB-series output, we proceed as follows:

1. Find the B-series solution to the stage equation.
2. Compute the matrix DB-series of $hF'(Y)$.
3. Compute the DB-series inverse to $I - hAF'(Y)$.
4. Compute the update.

Here, the first step is equivalent to that used for the **evaluation** operation. The remaining steps are completed using the DB-series tools for composition, inversion and substitution.

4.4 Applications

In this section, we introduce several analytical tools that have been developed using the object representations of GLMs, B-series and DB-series. In particular, we cover how to

- determine the GLM order of the pair $(\mathcal{M}_h, \mathcal{S}_h)$.
- compute the UOSM and ideal starting method of a GLM.
- compute the derivative UOSMs of a GLM (and the corresponding derivative starting methods).

4.4.1 Computing the order of a GLM

Suppose we are given a pair $(\mathcal{M}_h, \mathcal{S}_h)$ and would like to determine the GLM order, i.e. find the largest $p \in \mathbb{N}$ such that

$$\mathcal{M}_h \circ \mathcal{S}_h(y_0) = \mathcal{S}_h \circ \varphi_h(y_0) + O(h^{p+1}),$$

holds. The following code sample demonstrates how we can use our computational tools to achieve this:

```
>> mh = glm(A,U,B,V);
>> sh = starter(As,Bs,u);
>> y0 = bseries('trivial',9);
>> phi = bseries('evolution',9);
>> test = map(compose(mh,sh),y0) - map(sh,phi);
```

The order of the method is then determined by searching for the index of first non-zero term in `test.coeffs` and then cross-referencing with the corresponding tree in the lookup table `rooted_trees`. Note, however, that due to the current size of the lookup table, we can only verify methods up to order 9.

The above process has been completely automated, and can be used on any pair of GLM and starting method objects, `mh`, `sh`, with the command

`analysis.find_order(mh,sh)`.

4.4.2 Generating the UOSM and ideal starting method

In [46], a practical iterative algorithm for approximating the UOSM Φ_h and ideal starting method $\mathcal{S}_h^*$ is given (see also [11] for a similar result). This algorithm is based on the constructive proof of Theorem 2.27 given earlier in Section 2.3 and is described as follows:

Let $\mathcal{S}_h^{[k]}(y_0)$, $k \in \mathbb{N}_0$, denote a kth-order approximation to $\mathcal{S}_h^*(y_0)$ and define

$$\begin{aligned}\eta_k(y_0) &= \mathcal{F}_h \circ \mathcal{S}_h^{[k]}(y_0) - y_0,\\ \varepsilon_k(y_0) &= \mathcal{M}_h \circ \mathcal{S}_h^{[k]}(y_0) - \mathcal{S}_h^{[k]} \circ \mathcal{F}_h \circ \mathcal{M}_h \circ \mathcal{S}_h^{[k]}(y_0).\end{aligned}$$

Then, the *iterative starting method* is given by

$$\begin{aligned}\mathcal{S}_h^{[0]}(y_0) &= uy_0,\\ \mathcal{S}_h^{[k+1]}(y_0) &= \mathcal{S}_h^{[k]}(y_0) + D\varepsilon_k(y_0) - u\eta_k(y_0), \qquad \forall\ k \geq 1,\end{aligned}$$

where $D = (I_r - uw^{\mathsf{H}})(I_r - V + uw^{\mathsf{H}}V)^{-1}$. The kth-order approximation to $\Phi_h(y_0)$, denoted $\Phi_h^{[k]}(y_0)$, is then given by the composition

$$\Phi_h^{[k]}(y_0) = \mathcal{F}_h \circ \mathcal{M}_h \circ \mathcal{S}_h^{[k]}(y_0).$$

While the iterative starting method was originally developed to obtain practical approximations to $\mathcal{S}_h^*(y_0)$ (i.e. a numerical vector in X^r), it may also be used to generate kth-order truncated B-series of both Φ_h and $\mathcal{S}_h^*$. Below, we discuss two implementations of the iterative starting method to obtain B-series outputs:

1. **Tableau**: Using the tableau composition formula (2.11) we can write the iteration in terms of the coefficient matrices of the method: Let $(A^{[k]}, U^{[k]}, B^{[k]}, u)$ denote the coefficient matrices corresponding to $\mathcal{S}_h^{[k]}$. Then, the tableau for $\mathcal{S}_h^{[k+1]}$ is given by

$$\left[\begin{array}{ccccc|c} A^{[k]} & 0 & 0 & 0 & 0 & U^{[k]} \\ UB^{[k]} & A & 0 & 0 & 0 & \mathbb{1} \\ U_F V B^{[k]} & U_F B & A_F & 0 & 0 & \mathbb{1}_F \\ U^{[k]} w^{\mathsf{H}} V B^{[k]} & U^{[k]} w^{\mathsf{H}} B & U^{[k]} B_F & A^{[k]} & 0 & U^{[k]} \\ U_F B^{[k]} & 0 & 0 & 0 & A_F & \mathbb{1}_F \\ \hline DB^{[k]} & DB & 0 & -DB^{[k]} & -uB_F & u \end{array}\right],$$

where A, U, B, V are the matrices for $\mathcal{M}_h$; A_F, U_F, B_F, w the matrices for $\mathcal{F}_h$; and we have used $Uu = \mathbb{1}$, $U_F u = \mathbb{1}_F$. These matrices can be built using the MATLAB command `tableauSk`.

The B-series for $\mathcal{S}_h^*(y_0)$ and $\Phi_h(y_0)$, truncated to order q, can then be computed using the following code sample:

```
[Aq,Bq] = tableauSk(q,A,U,B,V,AF,UF,BF,w,u);
y0 = bseries('trivial',q);
Sh = starter(Aq,Bq,u);
Mh = glm(A,U,B,V);
Fh = finisher(Sh,w);
Phi = map(compose(Fh,Mh,Sh),y0);
```

where `Aq`, `Bq` are the matrices corresponding to $A^{[q]}$ and $B^{[q]}$.

2. **Direct iteration**: An alternative approach is to directly implement the iteration:

```
% Initialise
D = (I-u*w)/(I-V+u*w*V);
y0 = bseries('trivial',q);
ISM = u*y0;

% Iterate
for i = 1:q
    eta = map(Fh,ISM) - y0;
    MISM = map(Mh,ISM);
    PHI = map(Fh,MISM);
    SPHI = compose(ISM,PHI);
    epsilon = MISM - SPHI;
    ISM = ISM + D*epsilon - u*eta;
end
```

Here, it is assumed that the finishing method object `Fh`, the GLM object `Mh` and the preconsistency vector `u`, are known. To apply the direct iteration to obtain both the UOSM and ideal starting method we use the command

`[ism,uosm] = analysis.computeUOSM(q,Fh,Mh,u)`.

Example 4.6. In Example 4.5, we considered the object representation of the Leapfrog method. Suppose now we would like to determine the coefficients of its UOSM, correct to an order 4 truncation. This can be achieved (using the direct iteration approach) as follows: Let `Fh`, `Mh` and `Sh` denote the objects corresponding to the finishing method, Leapfrog method and starting method. Then, the UOSM computed to order 4 is given by the command

```
>> u = Sh.V;
>> [ism,uosm] = analysis.computeUOSM(4,Fh,Mh,u); disp(uosm)
  bseries with properties:

    sym_coeffs: [1 1 1/2 0 1/8 -1/8 -1/16 0 1/48]
        coeffs: [1 1 1/2 0 1/4 -1/8 -1/8 0 1/8]
    num_coeffs: 9
    truncation: 4
```

Here, we observe that the coefficients belonging to `sym_coeffs` indicate that the method is of order 2, i.e. $\Phi_h(y_0) = \varphi_h(y_0) + O(h^3)$.

◇

4.4.3 Generation of derivative UOSMs

Recall that the map $\Psi_h : (X, X) \to X$ is called a *derivative UOSM* (DUOSM) of a GLM if

$$\mathcal{M}_h'(\mathcal{S}_h^*(y_0))S_h^P(y_0, v) = S_h^P(\Phi_h(y_0), \Psi_h^P(y_0, v)), \qquad \forall\ y_0, v \in X,$$

for some *derivative starting method* $S_h^P : (X, X) \to X^r$. It was shown in Theorem 3.30 that, for a given row vector DB-series $F_h^P(y_0, \cdot)$, there exists a unique pair $(S_h^P(y_0, v), \Psi_h^P(y_0, v))$ such that the above holds, and $F_h^P(y_0, S_h^P(y_0, v)) = v$. As was the case for the generation of the UOSM, we can use the constructive proof of Theorem 3.30 to form the basis of an iterative algorithm for determining the derivative UOSM:

Let w_P, u_P denote the left and right eigenvectors of V corresponding to the eigenvalue ζ_P, scaled such that $w_P^{\mathsf{H}} u_P = 1$. Also let $S_h^{P[k]}(y_0)$ denote the kth-order approximation to $S_h^P(y_0)$, and $\Psi_h^{P[k]}(y_0) := w_P^{\mathsf{H}} \mathcal{M}_h'(\mathcal{S}_h^*(y_0))S_h^{P[k]}$ denote the kth-order approximation to $\Psi_h^P(y_0)$, and define

$$\varepsilon_k(y_0) = \mathcal{M}_h'(\mathcal{S}_h^*(y_0))S_h^{P[k]}(y_0) - S_h^{P[k]}(\Phi_h(y_0))\Psi_h^{P[k]}(y_0).$$

Then, the iteration for computing the derivative starting method is given by

$$\begin{aligned} S_h^{P[0]}(y_0) &= u_P, \\ S_h^{P[k+1]}(y_0) &= S_h^{P[k]}(y_0) + D_P \varepsilon_k(y_0), \qquad \forall\ k \geq 1, \end{aligned}$$

where $D_P = (I_r - u_P w_P^{\mathsf{H}})(\zeta_P I_r - V + u_P w_P^{\mathsf{H}} V)^{-1}$.

Remark 4.7. Here, we have made the choice that $F_h^P(y_0, \cdot) = w_P^{\mathsf{H}}$. Note that this choice guarantees that $w_P^{\mathsf{H}} S_h^{P[k]}(y_0, v) = v$, for all k, since $w_P^{\mathsf{H}} D_P = 0$ and

$$w_P^{\mathsf{H}} S_h^{P[k]}(y_0, v) = w_P^{\mathsf{H}} S_h^{P[k-1]}(y_0, v) = \cdots = w_P^{\mathsf{H}} S_h^{P[0]}(y_0) = w_P^{\mathsf{H}} u_P = 1.$$

The above algorithm can be implemented as a direct iteration as is demonstrated in the following code sample:

```
% Compute the ideal starting method and UOSM
[ism,uosm] = analysis.computeUOSM(q+1,Fh,Mh,u)

% Initialise
Dp = (I-up*wp)/(zeta*I-V+up*wp*V);
DM = diff(Mh,ism);
DSM = up*dbseries('trivial',q);

% Iterate
for i = 1:q
    DMDSM = sub(DM,DSM);
    DUOSM = wp*DMDSM;
    DSDUOSM = sub(compose(DSM,uosm),DUOSM);
    epsilon = DMDSM - DSDUOSM;
    DSM = DSM + Dp*epsilon;
end
```

To apply the iteration in practice, we use the command

`[dsm,duosm] = analysis.computeDUOSM(q,Fh,Mh,u,up,wp,zeta)`.

Example 4.8. At the end of Chapter 3, we presented GLMs (3.9) and (3.10) that we claimed are parasitism-free to 4th order, i.e. their derivative UOSM satisfies $\Psi_h^{(2)}(y_0, v) = -v + O(h^5)$. In the following piece of code, we use the direct iteration described above to verify this claim for GLM (3.10):

```
% Construct GLM coefficient matrices:
A = zeros(5);
A(1,1)= 1/2;
A(2,1)= 7/12; A(2,2)= 5/24;
A(3,1)= -1/12; A(3,2)= 1/2; A(3,3)= 1/24;
A(4,1)= 13/12; A(4,2)= 1/2; A(4,3)= -1/2; A(4,4)= -13/24;
A(5,1)= 5/12; A(5,2)= 1/2; A(5,3)= -1/2; A(5,4)= -1/2; A(5,5)= 7/24;
B = zeros(2,5);
B(1,:)= [1,1/2,-1/2,-1/2,1/2];
B(2,:)= [0,1/2,-1/2,-1/2,1/2];
U= [1,1;1,0;1,0;1,0;1,0];
V= [1,0;0,-1];

% Build method objects; finisher is trivial
Mh= glm(A,U,B,V);    Fh= finisher([1,0]);    u= [1;0];

% Store parasitc eigenvalue and directions
zeta= -1;    up= [0;1];    wp= [0,1];

% Compute DUOSM to order 4
[dsm,duosm]= analysis.computeDUOSM(4,Fh,Mh,u,up,wp,zeta);

% Check for zero coefficients
test= duosm - zeta*dbseries('trivial',4); disp(norm(test.coeffs,1))
```

Running this test, we found `norm(test.coeffs,1)` evaluates to 3.3249e-16, i.e. zero to rounding error. This confirms that the coefficients of orders 0-4 of the DUOSM are zero.

Note: the above code can be easily modified to check that (3.9) is also parasitism-free to 4th order by changing the coefficient matrices for A, U and B. Doing this, we found that `norm(test.coeffs,1)` evaluates to exactly 0.

◇

Chapter 5

Composition

Composition is a technique applied to numerical methods to construct new methods with some desired property. For example, composition can be used to design a method of higher order [20, 27, 63, 58, 59, 45], control parasitism [13], and increase stability for stiff problems [29, 40]. High-order methods may also be constructed via extrapolation [37, Ch. II.9]. However, this approach tends not to preserve the underlying geometric properties of the base numerical method.

In this chapter, we shall focus on using composition to construct high-order GLMs. In particular, we build upon the theory of composition for one-step methods which is already well-developed (see e.g. [36, Ch II.4]).

5.1 Composition of one-step methods

Let $\Phi_h : X \to X$ denote a one-step method (OSM). Then, a composition method $\psi_h : X \to X$ is given by

$$\psi_h(y_0) = \Phi_{\alpha_k h} \circ \cdots \circ \Phi_{\alpha_2 h} \circ \Phi_{\alpha_1 h}(y_0), \qquad y_0 \in X, \tag{5.1}$$

where $k \in \mathbb{N}$, $\alpha_1, \ldots, \alpha_k \in \mathbb{R}\backslash\{0\}$ and it is assumed that $\alpha_1 + \cdots + \alpha_k = 1$ to ensure that ψ_h corresponds to a time-h evolution.

Composition methods are particularly useful in geometric integration as they tend to preserve the structure-preserving properties of the base numerical method Φ_h.

Theorem 5.1. *[36, p. 190] If Φ_h is symplectic, then the composition method ψ_h is also symplectic. If Φ_h is symmetric, then ψ_h is symmetric provided*

$$\alpha_j = \alpha_{k-j+1}, \qquad \textit{for} \qquad j = 1, \dots, k.$$

Proof. If $\Phi_h(y_0)$ is symplectic, then it satisfies

$$\left(\frac{\partial \Phi_h(y_0)}{\partial y_0}\right)^T J \left(\frac{\partial \Phi_h(y_0)}{\partial y_0}\right) = J, \qquad J = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix},$$

for any value of h and y_0. Now, defining

$$\Phi_j' := \left(\frac{\partial \Phi_{\alpha_j h}(y)}{\partial y}\right)\Bigg|_{y=\Phi_{\alpha_{j-1}h}\circ\cdots\circ\Phi_{\alpha_2 h}\circ\Phi_{\alpha_1 h}(y_0)}, \qquad j = 1, \dots, k,$$

we observe that

$$\left(\frac{\partial \psi_h(y_0)}{\partial y_0}\right)^T J \left(\frac{\partial \psi_h(y_0)}{\partial y_0}\right) = (\Phi_1')^T \cdots (\Phi_{k-1}')^T (\Phi_k')^T J \Phi_k' \Phi_{k-1}' \cdots \Phi_1'.$$

It follows from the symplecticity of Φ_h that $(\Phi_j')^T J \Phi_j' = J$, for $j = 1, \dots, k$. Thus,

$$\left(\frac{\partial \psi_h(y_0)}{\partial y_0}\right)^T J \left(\frac{\partial \psi_h(y_0)}{\partial y_0}\right) = J,$$

and the method is symplectic as required.

If Φ_h is symmetric, then $\Phi_h(y_0) = \Phi_{-h}^{-1}(y_0) =: \Phi_h^*(y_0)$ (cf. Section 2.5). Now, we observe that

$$\psi_h^*(y_0) = (\Phi_{\alpha_k h}\circ\cdots\circ\Phi_{\alpha_2 h}\circ\Phi_{\alpha_1 h}(y_0))^* = \Phi_{\alpha_1 h}^*\circ\cdots\circ\Phi_{\alpha_{k-1} h}^*\circ\Phi_{\alpha_k h}^*(y_0).$$

Using symmetry of Φ_h and recalling that $\alpha_j = \alpha_{k-j+1}$, for $j = 1, \dots, k$, we find that $\psi_h^*(y_0) = \psi_h(y_0)$, and the method is symmetric as required. □

5.1.1 Higher order methods

For certain choices of $\alpha_1, \dots, \alpha_k$, the composed method ψ_h is of higher order than the base method Φ_h. In particular, it has been shown (see e.g. [36, pp. 43–46]) that if Φ_h is of even order $p \in \mathbb{N}$, then the composed method is at least of order $p+1$ if

$$\begin{aligned} \alpha_1 + \cdots + \alpha_k &= 1, \\ \alpha_1^{p+1} + \cdots + \alpha_k^{p+1} &= 0. \end{aligned} \tag{5.2}$$

It should be noted that for odd p, no real solution exists to the above conditions.

Definition 5.2 (**Triple jump** [20, 27, 63, 58])**.** Consider a OSM Φ_h of even order $p \in \mathbb{N}$. Then, the *triple jump* composition method is given by

$$\psi_h(y_0) := \Phi_{\alpha_1 h} \circ \Phi_{\alpha_2 h} \circ \Phi_{\alpha_1 h}(y_0),$$

where

$$\alpha_1 = \frac{1}{2 - 2^{1/(p+1)}}, \qquad \alpha_2 = -\frac{2^{1/(p+1)}}{2 - 2^{1/(p+1)}}. \tag{5.3}$$

Definition 5.3 (**Suzuki 5-jump** [58])**.** Consider a OSM Φ_h of even order $p \in \mathbb{N}$. Then, the *Suzuki 5-jump* composition method is given by

$$\psi_h(y_0) := \Phi_{\alpha_1 h} \circ \Phi_{\alpha_1 h} \circ \Phi_{\alpha_2 h} \circ \Phi_{\alpha_1 h} \circ \Phi_{\alpha_1 h}(y_0),$$

where

$$\alpha_1 = \frac{1}{4 - 4^{1/(p+1)}}, \qquad \alpha_2 = -\frac{4^{1/(p+1)}}{4 - 4^{1/(p+1)}}. \tag{5.4}$$

Both of these compositions will yield a method of at least order $p+1$. In the case that Φ_h is symmetric, it follows from the preservation of symmetry and the necessity of even order that the composition method is also symmetric and must be at least of order $p+2$. Furthermore, we can use this new method as the basis for another composition. Continuing in this fashion we can generate methods of arbitrarily high order.

Example 5.4. Let Φ_h represent a consistent RKM with defining matrices (A, B). When viewed as a GLM, the method has a tableau given by

$$\left[\begin{array}{c|c} A & \mathbb{1} \\ \hline B & 1 \end{array}\right].$$

Recall that the tableau corresponding to the composition of two GLMs $\mathcal{M}_h^{(1)}$, $\mathcal{M}_h^{(2)}$, with coefficient matrices $(A^{(1)}, U^{(1)}, B^{(1)}, V^{(1)})$ and $(A^{(2)}, U^{(2)}, B^{(2)}, V^{(2)})$, can be found using the formula (2.11):

$$\left[\begin{array}{cc|c} A^{(1)} & 0 & U^{(1)} \\ U^{(2)}B^{(1)} & A^{(2)} & U^{(2)}V^{(1)} \\ \hline V^{(2)}B^{(1)} & B^{(2)} & V^{(2)}V^{(1)} \end{array}\right].$$

By recursively applying this formula, we can express the triple jump method as a GLM:

$$\left[\begin{array}{ccc|c} \alpha_1 A & 0 & 0 & \mathbb{1} \\ \alpha_1 \mathbb{1} B & \alpha_2 A & 0 & \mathbb{1} \\ \alpha_1 \mathbb{1} B & \alpha_2 \mathbb{1} B & \alpha_1 A & \mathbb{1} \\ \hline \alpha_1 B & \alpha_2 B & \alpha_1 B & 1 \end{array}\right].$$

Here, if we consider the implicit midpoint rule with $(A, B) = (\frac{1}{2}, 1)$, which is both symmetric and symplectic (see e.g. [36, pp. 3, 34]), as the base numerical method then an order increase to $p = 4$ is obtained. Furthermore, the composed method is also symmetric and symplectic (see Theorem 5.1). General compositions involving the implicit midpoint rule have been studied by Sanz-Serna and Abia [53], who have shown they essentially generate the family of the diagonally-implicit, symmetric and symplectic RKMs.

◇

Composition of arbitrary OSMs

Let us now consider an arbitrary, consistent OSM Φ_h that is of odd order p. As mentioned above, no real solution exists to (5.2), thus we cannot consider compositions of the form (5.1). However, if we consider a composition that involves a method and its adjoint $\Phi_h^*(y_0) := \Phi_{-h}^{-1}(y_0)$ (see e.g. [59, 45]), then we can overcome this restriction. In particular,

$$\psi_h(y_0) = \Phi^*_{\beta_k h} \circ \Phi_{\alpha_k h} \circ \cdots \circ \Phi_{\alpha_2 h} \circ \Phi^*_{\beta_1 h} \circ \Phi_{\alpha_1 h}(y_0), \qquad y_0 \in X, \tag{5.5}$$

where $\alpha_1 \ldots, \alpha_k, \beta_1, \ldots, \beta_k \in \mathbb{R}$ and it is assumed that $\alpha_1 + \beta_1 + \alpha_2 + \ldots + \alpha_k + \beta_k = 1$. It can be seen that this form of composition is a direct generalisation of (5.1), where the original is obtained for the choice $\beta_1 = \beta_2 = \cdots = \beta_k = 0$.

As with the original composition, conditions on α_j, β_j, $j = 1, \ldots, k$, can be found such that an order increase is obtained (see e.g. [36, p. 45]). In particular, if Φ_h is of order p and

$$\begin{aligned} \alpha_1 + \beta_1 + \alpha_2 + \cdots + \alpha_k + \beta_k &= 1, \\ \alpha_1^{p+1} + (-1)^p \beta_1^{p+1} + \cdots + \alpha_k^{p+1} + (-1)^p \beta_k^{p+1} &= 0, \end{aligned} \tag{5.6}$$

then $\psi_h(y_0)$ will be at least of order $p + 1$.

Example 5.5. Consider the composition

$$\psi_h(y_0) = \Phi_{\frac{h}{2}} \circ \Phi^*_{\frac{h}{2}}(y_0),$$

where Φ_h is the RKM corresponding to the first order backward Euler method with defining matrices $(A, B) = (1, 1)$, and the adjoint Φ^*_h is given by the Euler method with matrices $(A^*, B^*) = (0, 1)$. Viewing the methods as GLMs, and using the composition tableau formula (2.11) (see also Example 5.4), then ψ_h is described by the tableau

$$\left[\begin{array}{cc|c} \frac{1}{2} & 0 & 1 \\ \frac{1}{2} & 0 & 1 \\ \hline \frac{1}{2} & \frac{1}{2} & 1 \end{array}\right].$$

Notice that the first and second stages are the same. This means we can combine them to obtain the reduced, single-stage RKM (expressed as a GLM)

$$\left[\begin{array}{c|c} \frac{1}{2} & 1 \\ \hline 1 & 1 \end{array}\right],$$

which is the implicit midpoint rule, known to be of order $p = 2$.

◇

The order increase of the composed method in the above example can also be explained by symmetry: First note that

$$\psi^*_h = (\Phi^*_{\frac{h}{2}} \circ \Phi_{\frac{h}{2}})^* = \Phi^*_{\frac{h}{2}} \circ \Phi^{**}_{\frac{h}{2}} = \Phi^*_{\frac{h}{2}} \circ \Phi_{\frac{h}{2}} = \psi_h.$$

In other words, the action of composing a method with its adjoint (and with equally weighted time-steps) yields a symmetric method. It then follows from the necessity of even order that the composition method must also be of even order. Thus, any OSM of odd order p, composed in this way, will yield a symmetric method of even order $p + 1$.

5.2 Composition of GLMs

In the following sections, we explore generalisations of the composition formulae (5.1) and (5.5) to GLMs. An important part of this will involve understanding the rules for the composition of methods with different time-steps. In particular, we will cover two special cases:

1. GLMs with Nordsieck inputs,
2. GLMs with general inputs.

5.2.1 Composition of GLMs with Nordsieck inputs

Consider a GLM that takes Nordsieck inputs, that is, at step $n \in \mathbb{N}_0$ the GLM generates approximations to the Nordsieck vector $N_h^{[n]}$, where

$$N_h^{[n]} := \begin{bmatrix} y|_{t=nh} \\ h\frac{\mathrm{d}y}{\mathrm{d}t}|_{t=nh} \\ h^2\frac{\mathrm{d}^2y}{\mathrm{d}t^2}|_{t=nh} \\ \vdots \\ h^{r-1}\frac{\mathrm{d}^{r-1}y}{\mathrm{d}t^{r-1}}|_{t=nh} \end{bmatrix}.$$

Examples of methods that can be converted to Nordsieck form include the class of irreducible LMMs [54] and the class of diagonally implicit multistage integration methods (DIMSIMs) [12]. Methods of this type have been identified as suitable candidates for variable time-step implementations since only a scaling of the input vector is required. Thus, it is natural to consider this class of methods when attempting to generalise composition formulae (5.1) and (5.5) to GLMs.

Nordsieck composition: Let us now consider a GLM generalisation of composition formula (5.1) for methods with Nordsieck inputs. In particular, for an r-input GLM, we define the diagonal matrix

$$D(a,b) := \operatorname{diag}\left(1, \frac{a}{b}, \left(\frac{a}{b}\right)^2, \ldots, \left(\frac{a}{b}\right)^{r-1}\right), \qquad a,b \in \mathbb{R}\backslash\{0\}.$$

Then, for $\alpha_1, \ldots, \alpha_k \in \mathbb{R}\backslash\{0\}$, the *Nordsieck composition* formula is given as

$$\mathcal{M}_h^N := D(\alpha_1,\alpha_k)\mathcal{M}_{\alpha_k h} \circ D(\alpha_k,\alpha_{k-1})\mathcal{M}_{\alpha_{k-1}h} \circ \cdots \circ \mathcal{M}_{\alpha_2 h} \circ D(\alpha_2,\alpha_1)\mathcal{M}_{\alpha_1 h}. \tag{5.7}$$

Theorem 5.6. *Consider a consistent GLM $\mathcal{M}_h$, where $\zeta_1 = 1$ is a simple eigenvalue of V, and a corresponding starting method $\mathcal{S}_h$ such that the following assumptions are satisfied:*

(A1) $(\mathcal{M}_h, \mathcal{S}_h)$ *is of even order* $p \in \mathbb{N}$.

(A2) $\mathcal{S}_h(y_0) = N_h^{[0]} + O(h^{p+2})$ *and* $e_1^T \mathcal{S}_h(y_0) = y_0$, *where* $e_1 = [1, 0, \ldots, 0]^T \in \mathbb{R}^r$.

(A3) $e_1^T V = e_1^T$.

Furthermore, let $\alpha_1, \ldots, \alpha_k$ *be chosen such that* (5.2) *holds, i.e.*

$$\begin{aligned} \alpha_1 + \cdots + \alpha_k &= 1, \\ \alpha_1^{p+1} + \cdots + \alpha_k^{p+1} &= 0. \end{aligned}$$

Then, there exists a starting method $\mathcal{S}_h^N$ *such that the pair* $(\mathcal{M}_h^N, \mathcal{S}_h^N)$ *is at least of order* $p+1$.

Proof. Given (A1), it follows from the definition of GLM order (2.10) that

$$D(\alpha_i, \alpha_j)\mathcal{M}_{\alpha_j h} \circ \mathcal{S}_{\alpha_j h}(y_0) = D(\alpha_i, \alpha_j)\left(\mathcal{S}_{\alpha_j h}(\varphi_{\alpha_j h}(y_0)) + C(y_0)\alpha_j^{p+1} h^{p+1}\right) + O(h^{p+2}),$$

where $C(y_0)$ is some vector dependent on various derivatives of f evaluated at y_0. It follows from (A2) that $D(\alpha_i, \alpha_j)\mathcal{S}_{\alpha_j h} = \mathcal{S}_{\alpha_i h} + O(h^{p+2})$. Thus,

$$D(\alpha_i, \alpha_j)\mathcal{M}_{\alpha_j h} \circ \mathcal{S}_{\alpha_j h}(y_0) = \mathcal{S}_{\alpha_i h}(\varphi_{\alpha_j h}(y_0)) + D(\alpha_i, \alpha_j)C(y_0)\alpha_j^{p+1} h^{p+1} + O(h^{p+2}).$$

Using this result, and noting that $\mathcal{M}_h'(y)z = Vz + O(h||z||)$, it follows that

$$\begin{aligned} \mathcal{M}_h^N \circ \mathcal{S}_{\alpha_1 h}(y_0) &= \mathcal{S}_{\alpha_1 h}(\varphi_{(\alpha_k + \cdots + \alpha_1)h}(y_0)) + D(\alpha_1, \alpha_k)KC(y_0)h^{p+1} + O(h^{p+2}), \\ K &:= \sum_{m=1}^{k} VD(\alpha_k, \alpha_{k-1})VD(\alpha_{k-1}, \alpha_{k-2}) \cdots VD(\alpha_{m+1}, \alpha_m)\alpha_m^{p+1}. \end{aligned}$$

Note that (A3), together with (5.2), implies that

$$\begin{aligned} e_1^T K &= \sum_{m=1}^{k} e_1^T VD(\alpha_k, \alpha_{k-1})VD(\alpha_{k-1}, \alpha_{k-2}) \cdots VD(\alpha_{m+1}, \alpha_m)\alpha_m^{p+1}, \\ &= e_1^T \sum_{m=1}^{k} \alpha_m^{p+1} = 0^T. \end{aligned}$$

Now, define the matrix $G := (V - I + e_1 e_1^T)^{-1} D(\alpha_1, \alpha_s) K$. Here we note that since $\zeta_1 = 1$ is a simple eigenvalue of V, with e_1 the corresponding left and right eigenvector, G is well-defined. Furthermore, $(V - I)G = D(\alpha_1, \alpha_s)K$ since G satisfies $e_1^T G = 0^T$.

Next, we fix $\mathcal{S}_h^N(y_0) := \mathcal{S}_{\alpha_1 h}(y_0) - GC(y_0)h^{p+1}$, and observe that

$$\begin{aligned}\mathcal{M}_h^N \circ \mathcal{S}_h^N(y_0) - \mathcal{S}_h^N(\varphi_h(y_0)) &= \mathcal{M}_h^N \circ \mathcal{S}_{\alpha_1 h}(y_0) - \mathcal{S}_{\alpha_1 h}(\varphi_h(y_0)) - \\ &\qquad VGC(y_0)h^{p+1} + GC(\varphi_h(y_0))h^{p+1} + O(h^{p+2}), \\ &= D(\alpha_1, \alpha_s)KC(y_0)h^{p+1} - VGC(y_0)h^{p+1} + \\ &\qquad GC(\varphi_h(y_0))h^{p+1} + O(h^{p+2}).\end{aligned}$$

Using $\varphi_h(y_0) = y_0 + O(h)$ and $(V - I)G = D(\alpha_1, \alpha_s)K$, we find

$$\mathcal{M}_h^N \circ \mathcal{S}_h^N(y_0) - \mathcal{S}_h^N(\varphi_h(y_0)) = O(h^{p+2}).$$

By the definition of GLM order, this implies the pair $(\mathcal{M}_h^N, \mathcal{S}_h^N)$ is at least of order $p + 1$, as required.

□

Remark 5.7. Note in the above theorem that V can be singular. This implies that composition methods can be constructed from strictly-stable, Nordsieck-input, GLMs. As we shall see the following section, strictly-stable methods cannot be applied to the GLM-generalisation of (5.5) as this requires that the adjoint method exists, i.e. when V is invertible.

Example 5.8 (Nordsieck triple jump). Let α_1 and α_2 be given by the triple jump parameters (5.3), and define $D := D(\alpha_2, \alpha_1)$, then (5.7) with $k = 2$ yields

$$\mathcal{M}_h^N = \mathcal{M}_{\alpha_1 h} \circ D^{-1} \mathcal{M}_{\alpha_2 h} \circ D \mathcal{M}_{\alpha_1 h}. \tag{5.8}$$

Recursively applying the composition of GLM tableau formula (2.11) gives

$$\left[\begin{array}{ccc|c} \alpha_1 A & 0 & 0 & U \\ \alpha_1 UDB & \alpha_2 A & 0 & UDV \\ \alpha_1 UD^{-1}VDB & \alpha_2 UD^{-1}B & \alpha_1 A & UD^{-1}VDV \\ \hline \alpha_1 VD^{-1}VDB & \alpha_2 VD^{-1}B & \alpha_1 B & VD^{-1}VDV \end{array}\right].$$

◇

Preservation of symmetry: Recall that a composition of the form (5.2) involving a symmetric OSM yields a method that is also symmetric. Below, we consider the equivalent result for symmetric Nordsieck GLMs.

Theorem 5.9. *Let $\mathcal{M}_h$ be an (L,P)-symmetric GLM, where L is an $r\times r$ diagonal matrix with $L_{ii}=(-1)^{i+1}$, $i=1,\dots,r$, and assume that $\alpha_j=\alpha_{k-j+1}\in\mathbb{R}\backslash\{0\}$, for $j=1,\dots,k$. Then, composition (5.7) is also symmetric.*

Proof. Recall that a GLM is symmetric if it satisfies $\mathcal{M}_h^*=L\mathcal{M}_h\circ L$, where the adjoint method is defined as $\mathcal{M}_h^*:=\mathcal{M}_{-h}^{-1}$. Now, taking the adjoint of $\mathcal{M}_h^N$ we find

$$\begin{aligned}\left(\mathcal{M}_h^N\right)^* &= \left(D(\alpha_1,\alpha_k)\mathcal{M}_{\alpha_k h}\circ D(\alpha_k,\alpha_{k-1})\mathcal{M}_{\alpha_{k-1}h}\circ\cdots\circ\mathcal{M}_{\alpha_2 h}\circ D(\alpha_2,\alpha_1)\mathcal{M}_{\alpha_1 h}\right)^*,\\ &= \mathcal{M}_{\alpha_1 h}^*\circ D^{-1}(\alpha_2,\alpha_1)\mathcal{M}_{\alpha_2 h}^*\circ\cdots\circ\mathcal{M}_{\alpha_{k-1}h}^*\circ D^{-1}(\alpha_k,\alpha_{k-1})\mathcal{M}_{\alpha_k h}^*\circ D^{-1}(\alpha_1,\alpha_k).\end{aligned}$$

Noting that $D^{-1}(a,b)=D(b,a)$ for $a,b\in\mathbb{R}\backslash\{0\}$, this becomes

$$\left(\mathcal{M}_h^N\right)^* = \mathcal{M}_{\alpha_1 h}^*\circ D(\alpha_1,\alpha_2)\mathcal{M}_{\alpha_2 h}^*\circ\cdots\circ\mathcal{M}_{\alpha_{k-1}h}^*\circ D(\alpha_{k-1},\alpha_k)\mathcal{M}_{\alpha_k h}^*\circ D(\alpha_k,\alpha_1).$$

By assumption, we have that $\alpha_j=\alpha_{k-j+1}\in\mathbb{R}\backslash\{0\}$, for $j=1,\dots,k$. This implies that $D(\alpha_1,\alpha_k)=I$, and thus the expression above may be written as

$$\left(\mathcal{M}_h^N\right)^* = \mathcal{M}_{\alpha_k h}^*\circ D(\alpha_k,\alpha_{k-1})\mathcal{M}_{\alpha_{k-1}h}^*\circ\cdots\circ\mathcal{M}_{\alpha_2 h}^*\circ D(\alpha_2,\alpha_1)\mathcal{M}_{\alpha_1 h}^*.$$

Since $\mathcal{M}_h$ is symmetric, it follows that

$$\left(\mathcal{M}_h^N\right)^* = L\mathcal{M}_{\alpha_k h}\circ LD(\alpha_k,\alpha_{k-1})L\mathcal{M}_{\alpha_{k-1}h}\circ\cdots\circ\mathcal{M}_{\alpha_2 h}\circ LD(\alpha_2,\alpha_1)L\mathcal{M}_{\alpha_1 h}\circ L.$$

By the commutativity of diagonal matrices, we observe that $LD(a,b)L=D(a,b)$ for $a,b\in\mathbb{R}\backslash\{0\}$ since $L^2=I$. Thus, we deduce that

$$\left(\mathcal{M}_h^N\right)^* = L\mathcal{M}_{\alpha_k h}\circ D(\alpha_k,\alpha_{k-1})\mathcal{M}_{\alpha_{k-1}h}\circ\cdots\circ\mathcal{M}_{\alpha_2 h}\circ D(\alpha_2,\alpha_1)\mathcal{M}_{\alpha_1 h}\circ L = L\mathcal{M}_h^N\circ L,$$

and the method is symmetric as required. □

Having shown that the symmetry is preserved under the Nordsieck composition, we now move on to show that there exists a symmetric starting method that allows for an additional order increase to $p+2$ to be achieved.

Theorem 5.10. *Consider an (L,P)-symmetric GLM, where L is an $r \times r$ diagonal matrix with $L_{ii} = (-1)^{i+1}$, $i = 1,\dots,r$. Under the assumptions of Theorem 5.6, there exists a symmetric starting method,*

$$\widetilde{\mathcal{S}_h^N}(y_0) := \frac{1}{2}\left(\mathcal{S}_h^N(y_0) + L\mathcal{S}_{-h}^N(y_0)\right),$$

where $\mathcal{S}_h^N(y_0)$ is as given in the proof of Theorem 5.6, such that the pair $(\mathcal{M}_h^N, \widetilde{\mathcal{S}_h^N})$ is at least of order $p+2$.

Proof. Recall from Chapter 2, Theorem 2.43 that the ideal starting method $\mathcal{S}_h^*$ associated with a symmetric GLM and symmetric finishing method must also be symmetric, i.e. $\mathcal{S}_h^* = L\mathcal{S}_{-h}^*$. From Theorem 5.9 we know that $\mathcal{M}_h^N$ is symmetric, and since its finishing method $\mathcal{F}_h = e_1^T$ satisfies $e_1^T L = e_1^T$, it is also symmetric. Thus, we deduce that the ideal starting method associated with $\mathcal{M}_h^N$ must be symmetric.

Recall also from Chapter 2, Corollary 2.28 that if the pair $(\mathcal{M}_h, \mathcal{S}_h)$ is of order p, then $\mathcal{S}_h = \mathcal{S}_h^* + O(h^{p+1})$. From Theorem 5.6, we know that the pair $(\mathcal{M}_h^N, \mathcal{S}_h^N)$ is at least of order $p+1$. Thus, we deduce that $\mathcal{S}_h^N = \mathcal{S}_h^* + O(h^{p+2})$.

Combining these results, we find that

$$\mathcal{S}_h^* = \mathcal{S}_h^* - \frac{1}{2}\left(\mathcal{S}_h^* - L\mathcal{S}_{-h}^*\right) = \frac{1}{2}\left(\mathcal{S}_h^* + L\mathcal{S}_{-h}^*\right) = \frac{1}{2}\left(\mathcal{S}_h^N + L\mathcal{S}_{-h}^N\right) + O(h^{p+2}).$$

It now follows that $\mathcal{S}_h^N = \widetilde{\mathcal{S}_h^N} + O(h^{p+2})$, and we deduce that the pair $(\mathcal{M}_h^N, \widetilde{\mathcal{S}_h^N})$ must be at least of order $p+1$. However, since $\mathcal{F}_h$, $\mathcal{M}_h^N$, and $\widetilde{\mathcal{S}_h^N}$ are all symmetric, and $p+1$ is odd, it follows from the necessity of even order (cf. Theorem 2.42) that the pair $(\mathcal{M}_h^N, \widetilde{\mathcal{S}_h^N})$ must be at least of order $p+2$, as required.

□

Remark 5.11. It cannot be guaranteed that the starting method $\widetilde{\mathcal{S}_h^N}(y_0)$ will produce Nordsieck inputs of the form $\widetilde{\mathcal{S}_h^N}(y_0) = N_h^{[0]} + O(h^{p+4})$. Thus, for symmetric methods, repeated compositions cannot be used to obtain methods of arbitrarily high order.

i	h_i	$\Delta y_i := \|\|y_{T/h_i} - y_{T/h_{i+1}}\|\|_2$	$\log_{2/3}(\Delta y_i/\Delta y_{i+1})$
1	$\left(\frac{2}{3}\right)^4$	1.4542e-5	-6.0189
2	$\left(\frac{2}{3}\right)^5$	1.2670e-6	-5.9979
3	$\left(\frac{2}{3}\right)^6$	1.1132e-7	-6.0881
4	$\left(\frac{2}{3}\right)^7$	9.4304e-9	-6.4985
5	$\left(\frac{2}{3}\right)^8$	6.7640e-10	-

Table 5.1: Verification of 6th order using the Nordsieck-triple jump method in Example 5.12.

Example 5.12. Consider a Nordsieck-GLM triple jump (5.8) of the following 4th order, (L, P)-symmetric GLM:

$$\left[\begin{array}{ccc|cccc} \frac{113}{1608} & 0 & 0 & 1 & -\frac{515}{1608} & \frac{157}{3216} & -\frac{71}{14791} \\ \frac{9}{134} & \frac{71}{402} & 0 & 1 & -\frac{103}{402} & \frac{43}{804} & -\frac{43}{12864} \\ \frac{145}{402} & \frac{769}{804} & \frac{407}{1608} & 1 & -\frac{515}{1608} & \frac{247}{3216} & -\frac{51}{16715} \\ \hline \frac{2}{27} & \frac{23}{27} & \frac{2}{27} & 1 & 0 & 0 & 0 \\ \frac{4}{9} & \frac{10}{9} & \frac{4}{9} & 0 & -1 & 0 & 0 \\ \frac{1}{3} & -2 & -\frac{5}{3} & 0 & 0 & 0 & -\frac{1}{16} \\ \frac{80}{3} & -32 & -\frac{16}{3} & 0 & 0 & 16 & 0 \end{array}\right],$$

where

$$P = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \qquad L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}.$$

The preceding results indicate that we should obtain a 6th order method. To verify this, we apply the composed method to the simple pendulum problem (2.15):

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} p(t) \\ q(t) \end{bmatrix} = \begin{bmatrix} -\sin(q(t)) \\ p(t) \end{bmatrix}, \qquad \begin{bmatrix} p(0) \\ q(0) \end{bmatrix} = \begin{bmatrix} p_0 \\ q_0 \end{bmatrix}, \qquad t \in [0, T],$$

where we choose $T = 1024$, $(p_0, q_0) = (1, 0)$, for various choices of time-step h. The starting method was implemented using the iterative procedure introduced in [46] (see also Chapter 4.4.2). The results displayed in Table 5.1 clearly demonstrate that 6th order is achieved for this example.

◇

5.2.2 A canonical form for GLMs

In anticipation of our discussion on the composition of GLMs with arbitrary inputs, we introduce a *canonical form* for GLMs.

Definition 5.13. A GLM is said to be *canonical* if its starting and finishing methods are given by the preconsistency vectors u and w^{H}.

Canonical methods have the important property that their inputs are independent of h. Thus, we can compose multiple canonical methods of different time-steps provided only the preconsistency vectors agree. As is shown below, any method can transformed into canonical form provided that the tableaux for its starting method $\mathcal{S}_h$ and finishing method $\mathcal{F}_h$ are given by (2.7), i.e.

$$\left[\begin{array}{c|c} A_S & \mathbb{1}_S \\ \hline B_S & u \end{array}\right] \qquad \text{and} \qquad \left[\begin{array}{c|c} A_S - U_F B_S & U_F \\ \hline -w^{\mathsf{H}} B_S & w^{\mathsf{H}} \end{array}\right],$$

and that U_F satisfies $U_F u = \mathbb{1}_S$ (cf. (2.5)). In other words, a method can be transformed into canonical form provided that $\mathcal{F}_h \circ \mathcal{S}_h(y_0) = y_0$ exactly.

Theorem 5.14. *Every GLM $\mathcal{M}_h$, with starting and finishing methods, $\mathcal{S}_h$ and $\mathcal{F}_h$, determined by the tableaux (2.7) such that $\mathcal{F}_h \circ \mathcal{S}_h(y_0) = y_0$, is equivalent to a canonical GLM defined by the composition*

$$\mathcal{C}_h = T_h^{-1} \circ \mathcal{M}_h \circ T_h,$$

where $T_h, T_h^{-1} : X^r \to X^r$ are respectively determined by the GLM tableaux

$$\left[\begin{array}{c|c} A_S & U_F \\ \hline B_S & I \end{array}\right], \qquad \textit{and} \qquad \left[\begin{array}{c|c} A_S - U_F B_S & U_F \\ \hline -B_S & I \end{array}\right], \tag{5.9}$$

where A_S, U_F, B_S are the coefficient matrices of the starting and finshing methods, and I is the $r \times r$ identity matrix.

Proof. Let the maps $T_h, T_h^{-1} : X^r \to X^r$ be determined by the GLM tableaux (5.9). Then, it follows from the formula for the inverse of a GLM (2.19) that

$$T_h \circ T_h^{-1}(y) = T_h^{-1} \circ T_h(y) = y, \qquad \text{for any} \qquad y \in X^r.$$

Now, consider a nonlinear transformation of the numerical method as a whole, i.e.

$$\mathcal{F}_h \circ \mathcal{M}_h^n \circ \mathcal{S}_h(y_0) = (\mathcal{F}_h \circ T_h) \circ (T_h^{-1} \circ \mathcal{M}_h \circ T_h)^n \circ (T_h^{-1} \circ \mathcal{S}_h)(y_0) = \mathcal{F}_h^{\mathcal{C}} \circ \mathcal{C}_h^n \circ \mathcal{S}_h^{\mathcal{C}}(y_0).$$

Note that the corresponding starting and finishing methods of $\mathcal{C}_h$ are given by

$$\mathcal{S}_h^{\mathcal{C}} := T_h^{-1} \circ \mathcal{S}_h \qquad \text{and} \qquad \mathcal{F}_h^{\mathcal{C}} := \mathcal{F}_h \circ T_h,$$

where the tableaux for $\mathcal{S}_h$ and $\mathcal{F}_h$ are given by (2.7). Observe that the composition $T_h \circ u$ yields a tableau of the form

$$\left[\begin{array}{c|c} A_S & U_F u \\ \hline B_S & u \end{array}\right] = \left[\begin{array}{c|c} A_S & \mathbb{1}_S \\ \hline B_S & u \end{array}\right],$$

where we have used $U_F u = \mathbb{1}_S$ from (2.5). This agrees with the tableau for $\mathcal{S}_h$ and thus it follows that $\mathcal{S}_h^{\mathcal{C}}(y_0) = u \cdot y_0$. Also observe that $w^{\mathsf{H}} T_h^{-1}$ yields a tableau of the form

$$\left[\begin{array}{c|c} A_S - U_F B_S & U_F \\ \hline -w^{\mathsf{H}} B_S & w^{\mathsf{H}} \end{array}\right],$$

which agrees with the tableau for $\mathcal{F}_h$, thus $\mathcal{F}_h^{\mathcal{C}}(y) = w^{\mathsf{H}} y$. It now follows from Definition 5.13 that $\mathcal{C}_h$ is a canonical method. □

The tableau for the corresponding canonical method of a GLM may be obtained using the tableau composition formula (2.11):

$$\left[\begin{array}{ccc|c} A_S & 0 & 0 & U_F \\ UB_S & A & 0 & U \\ U_F V B_S & U_F B & A_S - U_F B_S & U_F V \\ \hline V B_S & B & -B_S & V \end{array}\right]. \tag{5.10}$$

In general, performing a nonlinear change of coordinates runs the risk of destroying certain properties of the base numerical method. Below, we show that a transformation to canonical form does not affect the order of the method.

Theorem 5.15. *Suppose $(\mathcal{M}_h, \mathcal{S}_h)$ is of order p. Then, $(\mathcal{C}_h, u)$ is also of order p.*

Proof. From order definition (2.10) we know $\mathcal{M}_h \circ \mathcal{S}_h(y_0) = \mathcal{S}_h \circ \varphi_h(y_0) + O(h^{p+1})$. After pre-multiplying by T_h^{-1} we find

$$\begin{aligned} & T_h^{-1} \circ \mathcal{M}_h \circ (T_h \circ T_h^{-1}) \circ \mathcal{S}_h(y_0) = T_h^{-1} \mathcal{S}_h(\varphi_h(y_0)) + O(h^{p+1}), \\ \implies \qquad & \mathcal{C}_h(u y_0) = u \varphi_h(y_0) + O(h^{p+1}). \end{aligned}$$

Thus $(\mathcal{C}_h, u)$ is also of order p. □

Preservation of symmetry: Symmetry is generally not preserved under the canonical transformation, unless the starting and finishing methods are also symmetric.

Theorem 5.16. *Suppose that $\mathcal{M}_h$ is an (L,P)-symmetric GLM. If $\mathcal{S}_h$ and $\mathcal{F}_h$ are symmetric, then $\mathcal{C}_h$ is symmetric.*

Proof. Since $\mathcal{S}_h$ and $\mathcal{F}_h$ are symmetric, this implies that their coefficient matrices satisfy conditions (2.24) and (2.25), i.e. there exists a permutation matrix P_S such that

$$A_S = -P_S A_S P_S, \qquad B_S = -L B_S P_S, \qquad Lu = u,$$
$$U_F = P_S U_F L, \qquad w^{\mathsf{H}} L = w^{\mathsf{H}}.$$

Upon substitution into the tableaux for T_h and T_h^{-1} we deduce that $T_h = LT_{-h}L$ and $T_h^{-1} = LT_{-h}^{-1}L$. Now, by the symmetry of $\mathcal{M}_h$, we observe that

$$\mathcal{C}_h = T_h^{-1}\circ\mathcal{M}_h\circ T_h = T_h^{-1}\circ L\mathcal{M}_h^*\circ LT_h = (T_h^{-1}\circ LT_h^{-*})\circ\mathcal{C}_h^*\circ(T_h^*\circ LT_h).$$

However,

$$T_h^*\circ LT_h = T_{-h}^{-1}\circ LT_h = T_{-h}^{-1}\circ T_{-h}\circ L = L.$$

Thus, $\mathcal{C}_h = L\mathcal{C}_h^*\circ L$ and the canonical method is symmetric as required. □

Example 5.17. It has been shown by Gragg [30] that the Leapfrog method, when initialised with the Euler starter, yields a global error expansion in even powers of h. In the context of symmetric GLMs, we cannot directly explain this result as the Euler starter is not symmetric, i.e. recall that this starter is given by

$$y^{[0]} = \begin{bmatrix} y_0 \\ y_0 + hf(y_0) \end{bmatrix}, \qquad \text{or in terms of GLM tableau} \qquad \left[\begin{array}{c|c} 0 & 1 \\ \hline 0 & 1 \\ 1 & 1 \end{array}\right],$$

then observe that this is not symmetric with respect to the L-involution of the Leapfrog method, where $L = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$.

However, we can explain Gragg's result using the canonical method: Recall that the Leapfrog tableau (2.4) is given by

$$\left[\begin{array}{c|cc} 0 & 0 & 1 \\ \hline 0 & 0 & 1 \\ 2 & 1 & 0 \end{array}\right].$$

Then using (5.10), we find the canonical form is given by

$$\left[\begin{array}{ccc|cc} 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ \hline 1 & 0 & 0 & 0 & 1 \\ 0 & 2 & -1 & 1 & 0 \end{array}\right].$$

Here, we observe that the second and third stage equations are equivalent, which implies there is a redundancy in the representation of the method. By removing one of these redundant stages (e.g. combining the second and third columns together, then removing the third row), we obtain the irreducible representation given by the tableau

$$\left[\begin{array}{cc|cc} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ \hline 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{array}\right].$$

It can now be verified using the symmetry conditions (2.21), that the canonical method is $(L_{\mathcal{C}}, P_{\mathcal{C}})$-symmetric where

$$L_{\mathcal{C}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad P_{\mathcal{C}} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

In addition, we observe that the starting and finishing methods, u and w^{H}, are trivially symmetric, i.e. $L_{\mathcal{C}} u = u$ and $w^{\mathsf{H}} L_{\mathcal{C}} = w^{\mathsf{H}}$. Thus, since the Leapfrog method written out in full is given by

$$\mathcal{F}_h \circ \mathcal{M}_h^n \circ \mathcal{S}_h(y_0) = w^{\mathsf{H}} \mathcal{C}_h^n \circ u y_0,$$

and w^{H}, $\mathcal{C}_h$ and u are all symmetric with respect to $L_{\mathcal{C}}$, we can apply the necessity of even order result (cf. Theorem 2.42) to conclude that the method has a global error expansion in even powers of h.

$\diamond$

5.2.3 Composition of canonical methods

Consider a canonical GLM $\mathcal{C}_h$ with an invertible matrix V. As mentioned earlier, the inputs to a canonical method are h-independent. Thus, we can consider a straightforward generalisation of the composition (5.5) to GLMs:

$$\mathcal{C}_h^A := \mathcal{C}^*_{\beta_k h} \circ \mathcal{C}_{\alpha_k h} \circ \cdots \circ \mathcal{C}^*_{\beta_2 h} \circ \mathcal{C}_{\alpha_2 h} \circ \mathcal{C}^*_{\beta_1 h} \circ \mathcal{C}_{\alpha_1 h}. \tag{5.11}$$

It can then be shown that the conditions on $\alpha_1, \ldots, \alpha_k, \beta_1, \ldots, \beta_k$ required for an order increase agree with those used for the composition of OSMs.

Theorem 5.18. *Suppose the pair $(\mathcal{C}_h, u)$ is of order $p \in \mathbb{N}$ and $\mathcal{C}_h$ is invertible. Then, the pair $(\mathcal{C}_h^A, u)$ is at least of order $p+1$ provided*

$$\alpha_1 + \beta_1 + \alpha_2 + \beta_2 + \cdots + \alpha_k + \beta_k = 1, \tag{5.12}$$

$$\alpha_1^{p+1} + (-1)^p \beta_1^{p+1} + \alpha_2^{p+1} + (-1)^p \beta_2^{p+1} + \cdots + \alpha_k^{p+1} + (-1)^p \beta_k^{p+1} = 0. \tag{5.13}$$

Proof. Since $\mathcal{C}_h$ is invertible, it follows that the adjoint method exists and, therefore, so does composition (5.11). Now, recall Lemma 2.39 which states that if the pair $(\mathcal{M}_h, \mathcal{S}_h)$ is of order p, i.e.

$$\mathcal{M}_h \circ \mathcal{S}_h(y_0) = \mathcal{S}_h \circ \varphi_h(y_0) + C(y_0) h^{p+1} + O(h^{p+2}),$$

then the pair $(\mathcal{M}_h^*, \mathcal{S}_{-h})$ satisfies

$$\mathcal{M}_h^* \circ \mathcal{S}_{-h}(y_0) = \mathcal{S}_{-h} \circ \varphi_h(y_0) + (-1)^p V^{-1} C(y_0) + O(h^{p+2}).$$

For canonical methods, $\mathcal{S}_h = \mathcal{S}_{-h} = u$. Thus, for each $j \in \{1, \ldots, k\}$, we have

$$\begin{aligned}
\mathcal{C}_{\alpha_j h}(u y_0) &= u \varphi_{\alpha_j h} + \alpha_j^{p+1} h^{p+1} C(y_0) + O(h^{p+2}), \\
\mathcal{C}^*_{\beta_j h}(u y_0) &= u \varphi_{\beta_j h} + (-1)^p \beta_j^{p+1} h^{p+1} V^{-1} C(y_0) + O(h^{p+2}).
\end{aligned}$$

Composing these expressions, we find that

$$\begin{aligned}
\mathcal{C}^*_{\beta_j h} \circ \mathcal{C}_{\alpha_j h}(u y_0) &= \mathcal{C}^*_{\beta_j h}(u \varphi_{\alpha_j h}(y_0) + C(y_0) h^{p+1} \alpha_j^{p+1} + O(h^{p+2})), \\
&= \mathcal{C}^*_{\beta_j h}(u \varphi_{\alpha_j h}(y_0)) + V^{-1} C(y_0) h^{p+1} \alpha_j^{p+1} + O(h^{p+2}), \\
&= u \varphi_{(\alpha_j + \beta_j) h}(y_0) + V^{-1} C(y_0) h^{p+1} \left(\alpha_j^{p+1} + (-1)^p \beta_j^{p+1} \right) + O(h^{p+2}),
\end{aligned}$$

where we have applied $(\mathcal{C}_h^*(y))' z = V^{-1} z + O(h||z||)$, and $C(\varphi_h(y_0)) = C(y_0) + O(h)$.

Recursively applying the above result to each $\mathcal{C}^*_{\beta_j h}\circ\mathcal{C}_{\alpha_j h}$ in the order of $j = 1, \ldots, k$ we find

$$\mathcal{C}^A_h(uy_0) = u\varphi_{(\sum_{j=1}^k \alpha_j+\beta_j)h}(y_0) + V^{-1}C(y_0)h^{p+1}\sum_{j=1}^{k}\left(\alpha_j^{p+1} + (-1)^p\beta_j^{p+1}\right) + O(h^{p+2}).$$

Thus, if (5.12) and (5.13) are satisfied, it follows that the pair $(\mathcal{C}^A_h, u)$ is at least of order $p+1$. □

To obtain an adjoint-free composition, i.e. a GLM-generalisation of (5.1), we set $\beta_j = 0$ for $j = 1, \ldots, k$ in (5.11). This choice replaces each $\mathcal{C}^*_{\beta_j h}$ by V^{-1} to give

$$\mathcal{C}^A_h = V^{-1}\mathcal{C}_{\alpha_k h}\circ\cdots\circ V^{-1}\mathcal{C}_{\alpha_2 h}\circ V^{-1}\mathcal{C}_{\alpha_1 h}.$$

Notice here that the final multiplication by V^{-1} will not affect the order of the method as $Vu = u$, i.e. if $(\mathcal{C}^A_h, u)$ is of order p, then $(V\mathcal{C}^A_h, u)$ is also of order p, since

$$V\mathcal{C}^A_h(uy_0) = V(u\varphi_h(y_0) + O(h^{p+1})) = u\varphi_h(y_0) + O(h^{p+1}).$$

Thus, we define the adjoint-free composition of canonical GLMs as

$$\mathcal{C}^B_h := \mathcal{C}_{\alpha_k h}\circ\cdots\circ V^{-1}\mathcal{C}_{\alpha_2 h}\circ V^{-1}\mathcal{C}_{\alpha_1 h}. \tag{5.14}$$

Theorem 5.19. *Let the assumptions of Theorem 5.18 hold. Then, the pair $(\mathcal{C}^B_h, u)$ is at least of order $p+1$ provided*

$$\alpha_1 + \alpha_2 + \cdots + \alpha_k = 1, \tag{5.15}$$
$$\alpha_1^{p+1} + \alpha_2^{p+1} + \cdots + \alpha_k^{p+1} = 0. \tag{5.16}$$

Proof. Follows from Theorem 5.18 with $\beta_1 = \cdots = \beta_s = 0$ and noting that $Vu = u$. □

Remark 5.20. The route we have taken in deriving (5.14) is important, as a direct application of (5.1) to canonical GLMs would fail to include the V^{-1} multiplications between methods. So while the composition would be valid, an order increase under conditions (5.15)-(5.16) would not necessarily be achieved.

Preservation of symmetry: Suppose now that the canonical method is (L, P)-symmetric. Without loss of generality, we restrict our attention to compositions of the form (5.14) as, by definition, a symmetric method is similar to its adjoint.

Theorem 5.21. *Let $\mathcal{C}_h$ be an (L,P)-symmetric, canonical GLM. Then, composition (5.14) is symmetric if $\alpha_j = \alpha_{k-j+1}$, for $j = 1,\dots,k$.*

Proof. Taking the adjoint of $\mathcal{C}_h^B$, we find

$$\left(\mathcal{C}_h^B\right)^* = \left(\mathcal{C}_{\alpha_k h}\circ\cdots\circ V^{-1}\mathcal{C}_{\alpha_2 h}\circ V^{-1}\mathcal{C}_{\alpha_1 h}\right)^* = \mathcal{C}^*_{\alpha_1 h}\circ\cdots\circ V\mathcal{C}^*_{\alpha_{k-1} h}\circ V\mathcal{C}^*_{\alpha_k h}.$$

By assumption, $\alpha_j = \alpha_{k-j+1}$, for $j = 1,\dots,k$. Thus, this becomes

$$\left(\mathcal{C}_h^B\right)^* = \mathcal{C}^*_{\alpha_k h}\circ\cdots\circ V\mathcal{C}^*_{\alpha_2 h}\circ V\mathcal{C}^*_{\alpha_1 h}.$$

Since $\mathcal{C}_h$ is symmetric, we have that $\mathcal{C}_h = L\mathcal{C}_h^*\circ L$ and $LVL = V^{-1}$. Therefore,

$$\begin{aligned}\left(\mathcal{C}_h^B\right)^* &= (L\mathcal{C}_{\alpha_k h}\circ L)\circ\cdots\circ VL\mathcal{C}_{\alpha_2 h}\circ LVL\mathcal{C}_{\alpha_1 h}\circ L,\\ &= L\mathcal{C}_{\alpha_k h}\circ\cdots\circ V^{-1}\mathcal{C}_{\alpha_2 h}\circ V^{-1}\mathcal{C}_{\alpha_1 h}\circ L = L\mathcal{C}_h^B\circ L,\end{aligned}$$

and the method is symmetric as required. □

5.2.4 Composition of non-canonical methods

Consider now a composition method based on an invertible GLM with arbitrary inputs. The corresponding composition formulae and results all extend straightforwardly from those given in the previous section after making the substitution $\mathcal{C}_h = T_h\circ\mathcal{M}_h\circ T_h^{-1}$. In particular, the general form of (5.11) is written as

$$\begin{aligned}\mathcal{M}_h^A := (T_{\alpha_1 h}\circ T^*_{\beta_k h})\circ\mathcal{M}^*_{\beta_k h}\circ(T^{-*}_{\beta_k h}\circ T^{-1}_{\alpha_k h})\circ\mathcal{M}_{\alpha_k h}\circ(T_{\alpha_k h}\circ T^*_{\beta_{k-1} h})\circ\cdots\circ\\ \mathcal{M}_{\alpha_2 h}\circ(T_{\alpha_2 h}\circ T^*_{\beta_1 h})\circ\mathcal{M}^*_{\beta_1 h}\circ(T^{-*}_{\beta_1 h}\circ T^{-1}_{\alpha_1 h})\circ\mathcal{M}_{\alpha_1 h},\end{aligned} \tag{5.17}$$

and for (5.14) this is

$$\begin{aligned}\mathcal{M}_h^B := (T_{\alpha_1 h}\circ T^{-1}_{\alpha_k h})\circ\mathcal{M}_{\alpha_k h}\circ(T_{\alpha_k h}\circ V^{-1}T^{-1}_{\alpha_{k-1} h})\circ\cdots\circ\\ \mathcal{M}_{\alpha_2 h}\circ(T_{\alpha_2 h}\circ V^{-1}T^{-1}_{\alpha_1 h})\circ\mathcal{M}_{\alpha_1 h}.\end{aligned} \tag{5.18}$$

In addition, the corresponding starting and finishing methods are given by

$$\mathcal{S}_h^A = \mathcal{S}_h^B = \mathcal{S}_{\alpha_1 h} \qquad \text{and} \qquad \mathcal{F}_h^A = \mathcal{F}_h^B = \mathcal{F}_{\alpha_1 h},$$

where $\mathcal{S}_h$ and $\mathcal{F}_h$ are the starting and finishing methods of the base GLM $\mathcal{M}_h$.

Composition of symmetric GLMs: Suppose the α_j are chosen to satisfy $\alpha_j = \alpha_{k-j+1}$, for $j = 1, \ldots, k$. Then, it follows from the discussion on canonical GLMs and Theorem 5.21 that the method $\mathcal{M}_h^B$ is also symmetric (provided the starting and finishing methods are also symmetric). Furthermore, if the α_j satisfy (5.15)-(5.16) then an additional order increase is obtained, i.e. from p to $p+2$ where p is necessarily even. As with OSMs, we can continue this composition indefinitely to construct methods of arbitrarily high order.

Example 5.22. Let us define the nonlinear map

$$R(\alpha_2, \alpha_1) := T_{\alpha_2 h} \circ V^{-1} T_{\alpha_1 h}^{-1}, \qquad \text{for} \qquad \alpha_1, \alpha_2 \in \mathbb{R} \backslash \{0\}.$$

Then, from composition (5.18), we can obtain the GLM version of the triple jump:

$$\mathcal{M}_h^T := \mathcal{M}_{\alpha_1 h} \circ R(\alpha_1, \alpha_2) \circ \mathcal{M}_{\alpha_2 h} \circ R(\alpha_2, \alpha_1) \circ \mathcal{M}_{\alpha_1 h}, \tag{5.19}$$

where, for $\mathcal{M}_h$ of order p,

$$\alpha_1 = \frac{1}{2 - 2^{1/(p+1)}}, \qquad \alpha_2 = -\frac{2^{1/(p+1)}}{2 - 2^{1/(p+1)}}.$$

Similarly, we obtain the GLM version of Suzuki 5-jump:

$$\begin{aligned} \mathcal{M}_h^S := \mathcal{M}_{\alpha_1 h} \circ R(\alpha_1, \alpha_1) \circ \mathcal{M}_{\alpha_1 h} \circ R(\alpha_1, \alpha_2) \circ \cdots \\ \mathcal{M}_{\alpha_2 h} \circ R(\alpha_2, \alpha_1) \circ \mathcal{M}_{\alpha_1 h} \circ R(\alpha_1, \alpha_1) \circ \mathcal{M}_{\alpha_1 h}, \end{aligned} \tag{5.20}$$

where

$$\alpha_1 = \frac{1}{4 - 4^{1/(p+1)}}, \qquad \alpha_2 = -\frac{4^{1/(p+1)}}{4 - 4^{1/(p+1)}}. \tag{5.21}$$

$\diamond$

It is possible to express the compositions in the previous example in terms of a GLM tableau. To do this for the triple jump, we first assume that the U_F matrix in the T_h and T_h^{-1} maps is defined as $U_F = \mathbb{1}_S w_1^{\mathsf{H}}$, where w_1 is the left eigenvector of V corresponding to eigenvalue $\zeta_1 = 1$. Then, through several applications of the tableau composition formula (2.11), the GLM triple jump is written as

$$\left[\begin{array}{ccccccc|c}
A\alpha_1 & 0 & 0 & 0 & 0 & 0 & 0 & U \\
U_F B\alpha_1 & (A_S - U_F B_S)\alpha_1 & 0 & 0 & 0 & 0 & 0 & U_F \\
U_F B\alpha_1 & -U_F B_S\alpha_1 & A_S\alpha_2 & 0 & 0 & 0 & 0 & U_F \\
UV^{-1}B\alpha_1 & -UV^{-1}B_S\alpha_1 & UB_S\alpha_2 & A\alpha_2 & 0 & 0 & 0 & U \\
U_F B\alpha_1 & -U_F B_S\alpha_1 & U_F B_S\alpha_2 & U_F B\alpha_2 & (A_S - U_F B_S)\alpha_2 & 0 & 0 & U_F \\
U_F B\alpha_1 & -U_F B_S\alpha_1 & U_F B_S\alpha_2 & U_F B\alpha_2 & -U_F B_S\alpha_2 & A_S\alpha_1 & 0 & U_F \\
UV^{-1}B\alpha_1 & -UV^{-1}B_S\alpha_1 & UB_S\alpha_2 & UV^{-1}B\alpha_2 & -UV^{-1}B_S\alpha_2 & UB_S\alpha_1 & A\alpha_1 & U \\
\hline
B\alpha_1 & -B_S\alpha_1 & VB_S\alpha_2 & B\alpha_2 & -B_S\alpha_2 & VB_S\alpha_1 & B\alpha_1 & V
\end{array}\right].$$

A similar approach can used to obtain the tableau for the Suzuki 5-jump.

Stage reductions: For an efficient implementation of a GLM composition method, we must ensure that any redundant stages are removed prior to integration. For example, the nonlinear map $R(a,b) := T_{ah} \circ V^{-1} T_{bh}^{-1}$ (cf. Example 5.22), performed between method evaluations is expressed as the GLM with tableau:

$$\left[\begin{array}{cc|c}
(A_S - U_F B_S)b & 0 & U_F \\
-U_F V^{-1} B_S b & A_S a & U_F V^{-1} \\
\hline
-V^{-1} B_S b & B_S a & V^{-1}
\end{array}\right].$$

Let A_S be an $\widetilde{s} \times \widetilde{s}$ matrix, then this tableau suggests that a total of $2\widetilde{s}$ stage equations must be solved for each $R(a,b)$-evaluation. However, assuming that $U_F = \mathbb{1}_S w_1^{\mathsf{H}}$, then in the case $a = b$ (see Suzuki composition (5.20)) we find a reduction to $\widetilde{s}$-many stages occurs, i.e. the tableau for $R(a,a)$ actually reads

$$\left[\begin{array}{c|c}
A_S a & U_F \\
\hline
(I - V^{-1}) B_S a & V^{-1}
\end{array}\right].$$

As reductions of this type are both method and composition dependent, we suggest that each (distinct) nonlinear map $R(a,b)$ is coded as an individual GLM, with redundant stages removed. Then, compositions such as the Suzuki 5-jump (5.20) should be performed in the fashion

$$\mathcal{M}_h^S := \mathcal{M}_{\alpha_1 h} \circ R_1 \circ \mathcal{M}_{\alpha_1 h} \circ R_3 \circ \mathcal{M}_{\alpha_2 h} \circ R_2 \circ \mathcal{M}_{\alpha_1 h} \circ R_1 \circ \mathcal{M}_{\alpha_1 h},$$

where $R_1 = R(\alpha_1, \alpha_1)$, $R_2 = R(\alpha_2, \alpha_1)$ and $R_3 = R(\alpha_1, \alpha_2)$ are each distinct, and irreducible GLMs.

Further comments: The overall efficiency of these composition methods will be determined by the level of implicitness in both the base numerical method $\mathcal{M}_h$, and in the maps T_h and T_h^{-1}. Methods that are likely to permit the most efficient implementations will be those with trivial finishing methods since then both T_h and T_h^{-1} can be designed to be entirely explicit.

Numerical experiments involving the non-canonical composition formulae can be found in Chapter 7. There, we apply the triple and Suzuki 5-jump compositions to construct symmetric methods of orders 6 and 8. In view of the above comments, these compositions are only performed for GLMs with trivial finishing methods.

Chapter 6

Decomposition

Method decomposition arises as the natural complement to composition. For example, a composition method written as $\mathcal{M}_h^{(c)} = \mathcal{M}_h^{(2)} \circ \mathcal{M}_h^{(1)}$ has a trivial decomposition into the methods $\mathcal{M}_h^{(1)}$ and $\mathcal{M}_h^{(2)}$. Here, the tableau for the composed method is computed using the composition formula (2.11):

$$\left[\begin{array}{c|c} A^{(c)} & U^{(c)} \\ \hline B^{(c)} & V^{(c)} \end{array}\right] = \left[\begin{array}{cc|c} A^{(1)} & 0 & U^{(1)} \\ U^{(2)}B^{(1)} & A^{(2)} & U^{(2)}V^{(1)} \\ \hline V^{(2)}B^{(1)} & B^{(2)} & V^{(2)}V^{(1)} \end{array}\right].$$

Notice that if the stage matrices $A^{(1)}$, $A^{(2)}$ are lower triangular, then it follows that the stage matrix $A^{(c)}$ of the composed method is also lower triangular. Suppose now that we are given a GLM with a lower triangular stage matrix. Then, we might ask under what conditions, on the coefficient matrices, does a decomposition hold?

In this chapter, we address this question and present a result on the decomposition of a structure-preserving GLM into several single-stage GLMs. The connection between single-stage GLMs and linear multistep methods (LMMs) is also explored and a representation for LMMs as GLMs in terms of growth parameters and characteristic roots is given.

6.1 GLM decomposition

Suppose we are given an s-stage GLM with a lower triangular stage matrix. Then, we are interested in the existence of a decomposition of the form

$$\mathcal{M}_h(y) = T_s\mathcal{M}_h^{(s)} \circ T_{s-1}\mathcal{M}_h^{(s-1)} \circ \cdots \circ \mathcal{M}_h^{(2)} \circ T_1\mathcal{M}_h^{(1)}(T_0 y), \tag{6.1}$$

where each $\mathcal{M}_h^{(i)}$ is a single-stage GLM and T_i a linear transformation.

In order to derive algebraic conditions such that the above decomposition holds, we first apply the tableau composition formula (2.11) to the RHS of (6.1). Then, we make a direct comparison to the coefficient matrices on the LHS. This approach yields the following decomposition conditions:

$$\begin{aligned}
A_{ii} &= a_i, && 1 \le i \le s, \\
A_{ij} &= u_i^{\mathsf{H}} T_{i-1} V_{i-1} T_{i-2} V_{i-2} \cdots T_{j+1} V_{j+1} T_j b_j, && 1 \le j < i \le s, \\
e_i^T U &= u_i^{\mathsf{H}} T_{i-1} V_{i-1} T_{i-2} V_{i-2} \cdots T_1 V_1 T_0, && 1 \le i \le s, \\
Be_i &= T_s V_s T_{s-1} V_{s-1} \cdots T_{i+1} V_{i+1} T_i b_i, && 1 \le i \le s, \\
V &= T_s V_s T_{s-1} V_{s-1} \cdots T_1 V_1 T_0,
\end{aligned} \tag{6.2}$$

where (A, U, B, V) are the coefficient matrices of the s-stage GLM, and $(a_i, u_i^{\mathsf{H}}, b_i, V_i)$ are the coefficient matrices of the ith single-stage GLM, for $i = 1, \ldots, s$.

Theorem 6.1. *Let $\mathcal{M}_h$ be an s-stage GLM with coefficient matrices (A, U, B, V). If A is lower triangular, and the method is either*

(a) (L, P)-symmetric, where P is the time-reversal permutation matrix,

(b) G-symplectic,

then, there exists a decomposition of the form

$$\mathcal{M}_h(y_0) = \mathcal{M}_h^{(s)} \circ V^{-1}\mathcal{M}_h^{(s-1)} \circ \cdots \circ V^{-1}\mathcal{M}_h^{(2)} \circ V^{-1}\mathcal{M}_h^{(1)}(y_0), \tag{6.3}$$

where each $\mathcal{M}_h^{(i)}$ is a single-stage method with coefficient matrices

$$\left[\begin{array}{c|c} a_i & u_i^{\mathsf{H}} \\ \hline b_i & V_i \end{array}\right] = \left[\begin{array}{c|c} e_i^T A e_i & e_i^T U \\ \hline Be_i & V \end{array}\right],$$

and where e_i denotes the ith canonical basis vector of dimension s.

Proof. The proposed decomposition (6.3) is of the form of (6.1). To see this, we set

$$T_0 = T_s = I_r, \qquad V_i = V, \qquad u_i^{\mathsf{H}} = e_i^T U, \qquad b_i = Be_i, \qquad a_i = e_i^T Ae_i,$$

for $i = 1, \ldots, s$, and we set $T_i = V^{-1}$, for $i = 1, \ldots, s-1$. Immediately, we notice that the first and last three decomposition conditions of (6.2) are satisfied, provided V is invertible. The second condition simplifies to

$$A_{ij} = u_i^{\mathsf{H}} V^{-1} b_j = e_i^T U V^{-1} B e_j, \quad 1 \leq j < i \leq s.$$

Case (a): Suppose that $\mathcal{M}_h$ is (L, P)-symmetric. Then, the symmetry condition $V = LV^{-1}L$ implies that V is invertible, and from the condition $A = UV^{-1}B - PAP$ we deduce that

$$A_{ij} = e_i^T Ae_j = e_i^T UV^{-1}Be_j - e_i^T PAPe_j = e_i^T UV^{-1}Be_j, \tag{6.4}$$

since A is lower triangular and P is the time-reversal permutation. Thus, the second decomposition condition is automatically satisfied.

Case (b): Suppose that $\mathcal{M}_h$ is G-symplectic. Then, the G-symplectic condition $G = V^{\mathsf{H}}GV$ with G non-singular implies that V is invertible. Combining conditions $DA + A^T D = B^{\mathsf{H}}GB$ and $DU = B^{\mathsf{H}}GV$ we deduce

$$\begin{aligned} A_{ij} = e_i^T Ae_j &= e_i^T D^{-1} B^{\mathsf{H}} GBe_j - e_i^T D^{-1} A^T De_j, \\ &= e_i^T UV^{-1}Be_j - \frac{d_{jj}}{d_{ii}} e_j^T Ae_i = e_i^T UV^{-1}Be_j, \end{aligned}$$

which satisfies the final decomposition condition since A is lower triangular. □

Corollary 6.2. *If $\mathcal{M}_h$ is G-symplectic, then each $\mathcal{M}_h^{(i)}$ is G-symplectic, where G is the same for each method. If $\mathcal{M}_h$ is (L, P)-symmetric, then*

$$\mathcal{M}_{-h}^{(i)\,-1} = L\mathcal{M}_h^{(s-i+1)}L, \qquad \textit{for} \qquad i = 1, \ldots, s.$$

Proof. Consider the case $\mathcal{M}_h$ is G-symplectic, then

$$\begin{aligned} V_i^{\mathsf{H}} GV_i &= V^{\mathsf{H}} GV = G, \\ b_i^{\mathsf{H}} GV_i &= e_i^T (DD^{-1}) B^{\mathsf{H}} GV = e_i^T DU = d_{ii} u_i^{\mathsf{H}}, \\ b_i^{\mathsf{H}} Gb_i &= e_i^T B^{\mathsf{H}} GBe_i = e_i^T (DA + A^T D)e_i = 2d_{ii}a_i. \end{aligned}$$

Thus, each $\mathcal{M}_h^{(i)}$ is G-symplectic. Suppose now that $\mathcal{M}_h$ is symmetric, then

$$\begin{aligned} V_i^{-1} &= V^{-1} = LVL = LV_{s-i+1}L, \\ V_i^{-1}b_i &= V^{-1}Be_i = LBPe_i = LBe_{s-i+1} = Lb_{s-i+1}, \\ u_i^T V_i^{-1} &= e_i^T UV^{-1} = e_i^T PUL = e_{s-i+1}^T UL = u_{s-i+1}^T L, \\ a_i - u_i^T V_i^{-1} b_i &= e_i^T (A - UV^{-1}B)e_i = e_i^T PAPe_i = a_{s-i+1}, \end{aligned}$$

which implies that ${\mathcal{M}_{-h}^{(i)}}^{-1} = L\mathcal{M}_h^{(s-i+1)}L$. □

6.1.1 Practical considerations of decomposition

In a naïve implementation of a GLM, we require the storage of r-inputs $y_1^{[n]}, \ldots, y_r^{[n]}$ at every step. In addition, we must also evaluate and store each $f(Y_1), \ldots, f(Y_s)$ to enable computation of the outputs. Thus, at every step, we must effectively store $s+r$ vectors of size $\dim(X)$. For problems with only a few degrees of freedom, this is not usually an issue. However, for large problems, the feasibility of the computation is governed by available memory.

GLMs permitting a decomposition are only as expensive (in terms of storage) as an r-input, single-stage GLM. An update is given by s-many applications of a single-stage method with an effective storage cost of $r+1$ vectors of size $\dim(X)$. For GLMs with a large number of stages this can be particularly beneficial.

6.2 Connection to linear multistep methods

It has been shown that an irreducible[1], r-step LMM may be equivalently expressed as a single-stage, r-input GLM [54, 15]. Given the above the decomposition results, it is possible that some structure-preserving GLMs may viewed as the composition of LMMs. This connection is explored further in the remaining sections of this chapter.

6.2.1 Linear multistep methods as GLMs

Recall that an r-step LMM is given by

$$\sum_{j=0}^{r} \alpha_j y_{n+j} = h \sum_{j=0}^{r} \beta_j f(y_{n+j}),$$

[1] LMMs are said to be reducible, in the sense of Dahlquist, if their characteristic polynomials share a common root. Otherwise, they are irreducible.

where $\alpha_j, \beta_j \in \mathbb{R}$, $\alpha_r \neq 0$ and $|\alpha_0| + |\beta_0| > 0$, and its corresponding characteristic polynomials are given by

$$\rho(\zeta) = \sum_{j=0}^{r} \alpha_j \zeta^j, \qquad \sigma(\zeta) = \sum_{j=0}^{r} \beta_j \zeta^j.$$

An equivalent formulation may be given in the form of a single-stage GLM [54, 15]. In particular, an r-step LMM in GLM form (abbreviated to LMM–GLM), denoted by the map $\mathcal{L}_h : X^r \to X^r$, is determined by the tableau

$$\left[\begin{array}{c|c} \frac{\beta_r}{\alpha_r} & e_r^T \\ \hline b & C^T \end{array}\right], \tag{6.5}$$

where

$$e_r = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \qquad b = \begin{bmatrix} \frac{\alpha_r\beta_0 - \beta_r\alpha_0}{\alpha_r^2} \\ \frac{\alpha_r\beta_1 - \beta_r\alpha_1}{\alpha_r^2} \\ \vdots \\ \vdots \\ \frac{\alpha_r\beta_{r-2} - \beta_r\alpha_{r-2}}{\alpha_r^2} \\ \frac{\alpha_r\beta_{r-1} - \beta_r\alpha_{r-1}}{\alpha_r^2} \end{bmatrix}, \qquad C = \begin{bmatrix} 0 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & \cdots & \cdots & 1 \\ -\frac{\alpha_0}{\alpha_r} & -\frac{\alpha_1}{\alpha_r} & -\frac{\alpha_2}{\alpha_r} & \cdots & \cdots & -\frac{\alpha_{r-1}}{\alpha_r} \end{bmatrix}.$$

Note that since C is a companion matrix, its characteristic polynomial is given by $\frac{\rho(\zeta)}{\alpha_r}$. Thus, the eigenvalues of C are given by the roots of $\rho(\zeta)$.

Example 6.3. The family of 2-step, symmetric, second order, LMMs is written as

$$\alpha_2(y_{n+2} - y_n) = h(\beta_2 f(y_{n+2}) + 2(\alpha_2 - \beta_2) f(y_{n+1}) + \beta_2 f(y_n)).$$

Expressed as a GLM, this family is written as

$$\left[\begin{array}{c|cc} \frac{\beta_2}{\alpha_2} & 0 & 1 \\ \hline \frac{2\beta_2}{\alpha_2} & 0 & 1 \\ \frac{2(\alpha_2 - \beta_2)}{\alpha_2} & 1 & 0 \end{array}\right].$$

Note, that if we fix $\beta_2 = 0$, $\alpha_2 = \frac{1}{2}$ we obtain the familiar Leapfrog method (2.4). For the choice $\beta_2 = \frac{1}{6}$, $\alpha_2 = \frac{1}{2}$, we obtain Simpson's rule.

$\diamond$

6.2.2 A diagonal form for the LMM–GLM

Suppose that the roots of the characteristic polynomial are distinct and non-zero (as is the case with symmetric LMMs). Then, there exists an invertible transformation such that the companion matrix C can be diagonalised. By the T-equivalence of a GLM, this then implies there exists an equivalent GLM formulation in diagonal form, i.e. the corresponding V matrix is diagonal.

Theorem 6.4. *Consider an r-step LMM where the roots of $\rho(\zeta)$ are distinct and non-zero. Then, there exists an invertible transformation $T \in \mathbb{C}^{r\times r}$ such that the LMM–GLM (6.5), is T-equivalent to the GLM $\mathcal{L}_h^D := T^{-1}\mathcal{L}_h \circ T$ with tableau*

$$\left[\begin{array}{c|cccc} \frac{\beta_r}{\alpha_r} & \zeta_1 & \zeta_2 & \cdots & \zeta_r \\ \hline \mu_1 & \zeta_1 & 0 & \cdots & 0 \\ \mu_2 & 0 & \zeta_2 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ \mu_r & 0 & \cdots & 0 & \zeta_r \end{array}\right], \tag{6.6}$$

where each ζ_i, $i = 1, \ldots, r$, is a root of $\rho(\zeta)$ and

$$\mu_i = \frac{\sigma(\zeta_i)}{\zeta_i \rho'(\zeta_i)}, \qquad 1 \le i \le r, \tag{6.7}$$

are the growth parameters of LMM stability theory (see e.g. [36, p. 592]).

Proof. Let $T \in \mathbb{C}^{r\times r}$ be written as the matrix product $T = W^{-1}D$ where W is a Vandermonde matrix and D is a diagonal matrix defined as

$$W = \begin{bmatrix} 1 & \zeta_1 & \zeta_1^2 & \cdots & \zeta_1^{r-1} \\ 1 & \zeta_2 & \zeta_2^2 & \cdots & \zeta_2^{r-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \zeta_r & \zeta_r^2 & \cdots & \zeta_r^{r-1} \end{bmatrix}, \qquad D_{ij} = \begin{cases} \frac{1}{\alpha_r}\zeta_i \cdot \rho'(\zeta_i), & \text{for } i = j, \\ 0 & \text{for } i \ne j. \end{cases}$$

Since the ζ_i are distinct and non-zero, it follows that both W and D are invertible, and therefore, T is invertible also.

Now, let us consider the tableau of $T^{-1}\mathcal{L}_h \circ T$:

$$\left[\begin{array}{c|c} \frac{\beta_r}{\alpha_r} & e_r^T T \\ \hline T^{-1}b & T^{-1}C^T T \end{array}\right].$$

To show that this tableau simplifies to that given in (6.6), we first verify that $T^{-1}C^TT = \Sigma_r$ where $\Sigma_r := \mathrm{diag}(\zeta_1, \ldots, \zeta_r)$: Since C is a companion matrix, it is diagonalisable by the Vandermonde matrix W. Thus, we deduce that

$$T^{-1}C^TT = D^{-1}WC^TW^{-1}D = D^{-1}\Sigma_r D = \Sigma_r.$$

Next, we verify that $e_r^TT = [\zeta_1, \ldots, \zeta_r]$: Recall that the elements of an inverse Vandermonde matrix $(W^{-1})_{ij}$ are given by the ζ^{i-1}-coefficients of the jth Lagrange basis polynomial

$$l_j(\zeta) = \frac{\alpha_r}{\rho'(\zeta_j)} \prod_{\substack{i=1 \\ i\neq j}}^{r} (\zeta - \zeta_i), \qquad j = 1, \ldots, r.$$

Thus, it follows that $e_r^TT = e_r^TW^{-1}D = \left[\frac{\alpha_r}{\rho'(\zeta_1)}, \ldots, \frac{\alpha_r}{\rho'(\zeta_r)}\right] D = [\zeta_1, \ldots, \zeta_r]$.

Finally, we verify that $T^{-1}b = \mu$ where $\mu = [\mu_1, \ldots, \mu_r]^T$ is the vector of growth parameters: Note that we may equivalently express (6.7) as an $r \times r$ linear Vandermonde system:

$$\begin{bmatrix} 1 & \zeta_1 & \zeta_1^2 & \cdots & \zeta_1^{r-1} \\ 1 & \zeta_2 & \zeta_2^2 & \cdots & \zeta_2^{r-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \zeta_r & \zeta_r^2 & \cdots & \zeta_r^{r-1} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{r-1} \end{bmatrix} = \begin{bmatrix} \zeta_1\rho'(\zeta_1)\mu_1 - \zeta_1^r\beta_r \\ \zeta_2\rho'(\zeta_2)\mu_2 - \zeta_2^r\beta_r \\ \vdots \\ \zeta_r\rho'(\zeta_r)\mu_r - \zeta_r^r\beta_r \end{bmatrix},$$

or more compactly as

$$W\beta = \alpha_r D\mu - \Sigma_r^r\beta_r, \qquad \text{where} \qquad \beta = [\beta_0, \ldots, \beta_{r-1}]^T.$$

Writing b as

$$b = \frac{1}{\alpha_r}\beta - \frac{\beta_r}{\alpha_r^2}\alpha, \qquad \text{where} \qquad \alpha = [\alpha_0, \ldots, \alpha_{r-1}]^T,$$

we find

$$T^{-1}b = \frac{1}{\alpha_r}D^{-1}W\beta - \frac{\beta_r}{\alpha_r^2}D^{-1}W\alpha = \frac{1}{\alpha_r}D^{-1}(\alpha_r D\mu - \Sigma_r^r\beta_r) - \frac{\beta_r}{\alpha_r^2}D^{-1}W\alpha.$$

Since $W\alpha = [\rho(\zeta_1) - \alpha_r\zeta_1^r, \ldots, \rho(\zeta_r) - \alpha_r\zeta_r^r]^T = -\alpha_r\Sigma_r^r$, it follows that

$$T^{-1}b = \mu - \frac{\beta_r}{\alpha_r}D^{-1}\Sigma_r^r - \frac{\beta_r}{\alpha_r^2}D^{-1}W\alpha = \mu.$$

Thus, by a comparison of tableaux, (6.5) is T-equivalent to (6.6).

□

Example 6.5. Reconsider the family of LMMs given in Example 6.3. The characteristic polynomials are given by

$$\rho(\zeta) = \alpha_2(\zeta^2 - 1), \qquad \sigma(\zeta) = \beta_2\zeta^2 + 2(\alpha_2 - \beta_2)\zeta + \beta_2.$$

The roots of ρ are given by $\zeta_1 = 1, \zeta_2 = -1$, and the associated growth parameters are computed to be $\mu_1 = 1$ and $\mu_2 = \frac{2(\beta_2 - \alpha_2)}{\alpha_2}$. Thus, this family may be equivalently expressed in diagonal form as

$$\left[\begin{array}{c|cc} \frac{\beta_2}{\alpha_2} & 1 & -1 \\ \hline 1 & 1 & 0 \\ \frac{2\beta_2 - \alpha_2}{\alpha_2} & 0 & -1 \end{array}\right].$$

◇

6.2.3 Reducibility

An important concept in the representation of numerical methods is reducibility. For GLMs, we may question whether a given tableau can be equivalently expressed as one with fewer stages or inputs. For single-stage GLMs, and similarly LMMs, we are only concerned with the potential reducibility of their inputs.

Definition 6.6. A GLM with coefficient matrices (A, U, B, V) is said to be *U-reducible* if there exists a $u_i \in \mathbb{C}^r$ such that u_i is a right-eigenvector of V and $Uu_i = 0$. Otherwise it is *U-irreducible*.

Definition 6.7. A GLM with coefficient matrices (A, U, B, V) is said to be *B-reducible* if there exists a $w_i \in \mathbb{C}^r$ such that w_i is a left-eigenvector of V and $B^{\mathsf{H}}w_i = 0$. Otherwise it is *B-irreducible*.

For a single-stage method GLM with diagonal V, U-irreducibility is equivalent to imposing that U has no zero element. Similarly, B must have no zero element to avoid B-reducibility.

Example 6.8. Let us reconsider Example 6.5, and suppose we make the choice that $\beta_2 = \frac{1}{2}\alpha_2$. Then, the second component of μ is zero and the method is B-reducible. Removing this redundant input, we obtain the method

$$\left[\begin{array}{c|c} \frac{1}{2} & 1 \\ \hline 1 & 1 \end{array}\right],$$

which is the GLM form of the implicit midpoint rule.

◇

From the diagonal GLM formulation of a LMM (6.6), we observe that a LMM with distinct, non-zero roots can never yield a U-reducible GLM. Similarly, a LMM with non-zero growth parameters will never yield a B-reducible GLM.

Notice that if a LMM possesses a zero growth parameter, then (6.7) implies that $\rho(\zeta)$ and $\sigma(\zeta)$ must share a common root. This is precisely Dahlquist's definition of a reducible LMM (see e.g. [36, Chap. XV]). Thus, a reducible LMM with distinct, non-zero roots implies B-reducibility of the diagonal GLM representation.

6.2.4 Decomposition into LMMs

It has been shown that a LMM can be expressed as a single-stage GLM. Let us now consider the converse result.

Theorem 6.9. *Consider a U-irreducible, single-stage GLM $\mathcal{M}_h$ where V has distinct, non-zero eigenvalues. Then, there exists an invertible matrix $T \in \mathbb{C}^{r\times r}$, such that the method $\mathcal{M}_h$ is T-equivalent to a diagonal LMM–GLM of the form (6.6).*

Proof. Since V has distinct eigenvalues, there exists an equivalent method where V is diagonal. Thus, without loss of generality, we assume that V is diagonal.

Now, since $\mathcal{M}_h$ is U-irreducible it follows that $\mathrm{diag}(u_1, \ldots, u_r)$, where $u_i = Ue_i$ for $i = 1, \ldots, r$ and e_i denotes the ith canonical basis vector of dimension r, is invertible. Writing

$$T = \mathrm{diag}(u_1, \ldots, u_r)^{-1}\mathrm{diag}(\zeta_1, \ldots, \zeta_r),$$

where $\zeta_1, \ldots, \zeta_r$ are the eigenvalues of V, we find that $\mathcal{M}_h$ is T-equivalent to a GLM

with tableau

$$\left[\begin{array}{c|c} A & UT \\ \hline T^{-1}B & T^{-1}\mathrm{diag}(\zeta_1,\dots,\zeta_r)T \end{array}\right] = \left[\begin{array}{c|cccc} A & \zeta_1 & \zeta_2 & \cdots & \zeta_r \\ \hline \frac{1}{\zeta_1}e_1^TBUe_1 & \zeta_1 & 0 & \cdots & 0 \\ \frac{1}{\zeta_2}e_2^TBUe_2 & 0 & \zeta_2 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ \frac{1}{\zeta_r}e_r^TBUe_r & 0 & \cdots & 0 & \zeta_r \end{array}\right].$$

Defining $\beta_r := A\alpha_r$ and $\mu_i := \frac{1}{\zeta_i}e_i^TBUe_i$ for $i = 1,\dots,r$, we find the above tableau agrees with (6.6), as required. □

Using the above result, we can now state the conditions under which a structure-preserving GLM yields a decomposition into s-many LMMs.

Theorem 6.10. *Let $\mathcal{M}_h$ be an s-stage GLM where*

(A1) A is lower-triangular,

(A2) $\mathcal{M}_h$ is either

- *(a) (L,P)-symmetric, where P is the time-reversal permutation matrix,*
- *(b) G-symplectic,*

(A3) U has no zero element,

(A4) V has distinct eigenvalues and is diagonal.

Then, there exists a decomposition of the form

$$\mathcal{M}_h(y_0) = D^{(s)}\mathcal{L}_h^{(s)} \circ D^{(s-1)}\mathcal{L}_h^{(s-1)} \circ \cdots \circ D^{(2)}\mathcal{L}_h^{(2)} \circ D^{(1)}\mathcal{L}_h^{(1)}(D^{(0)}y_0), \tag{6.8}$$

where each $\mathcal{L}_h^{(i)}$ corresponds to a diagonal LMM–GLM with tableau of the form (6.6), and

$$\begin{aligned} D^{(0)} &:= \mathrm{diag}\left(\frac{U_{1,1}}{\zeta_1},\dots,\frac{U_{1,r}}{\zeta_r}\right), \\ D^{(i)} &:= \mathrm{diag}\left(\frac{U_{i+1,1}}{\zeta_1 U_{i,1}},\dots,\frac{U_{i+1,r}}{\zeta_r U_{i,r}}\right), \qquad \textit{for} \qquad i = 1,\dots,s-1, \\ D^{(s)} &:= \mathrm{diag}\left(\frac{\zeta_1}{U_{s,1}},\dots,\frac{\zeta_r}{U_{s,r}}\right). \end{aligned}$$

Proof. Given assumptions $(A1)$ and $(A2)$, it follows from Theorem 6.1 that $\mathcal{M}_h$ has a decomposition of the form

$$\mathcal{M}_h(y_0) = \mathcal{M}_h^{(s)} \circ V^{-1} \mathcal{M}_h^{(s-1)} \circ \cdots \circ V^{-1} \mathcal{M}_h^{(2)} \circ V^{-1} \mathcal{M}_h^{(1)}(y_0),$$

where each $\mathcal{M}_h^{(i)}$ is a single-stage GLM. Now, we deduce from assumptions $(A3)$ and $(A4)$ that each $\mathcal{M}_h^{(i)}$ is U-irreducible, and it then follows from Theorem 6.9 that each is equivalent to a diagonal LMM–GLM, denoted by the map $\mathcal{L}_h^{(i)}$. Thus,

$$\begin{aligned}\mathcal{M}_h(y_0) &= \mathcal{M}_h^{(s)} \circ V^{-1} \mathcal{M}_h^{(s-1)} \circ \cdots \circ V^{-1} \mathcal{M}_h^{(2)} \circ V^{-1} \mathcal{M}_h^{(1)}(y_0), \\ &= T^{(s)} \mathcal{L}_h^{(s)} \circ {T^{(s)}}^{-1} V^{-1} T^{(s-1)} \mathcal{L}_h^{(s-1)} \circ {T^{(s-1)}}^{-1} \circ \cdots \\ &\qquad {T^{(3)}}^{-1} V^{-1} T^{(2)} \mathcal{L}_h^{(2)} \circ {T^{(2)}}^{-1} V^{-1} T^{(1)} \mathcal{L}_h^{(1)}({T^{(1)}}^{-1} y_0),\end{aligned}$$

where $T^{(i)} = \operatorname{diag}(U_{i,1}, \ldots, U_{i,r})^{-1} \operatorname{diag}(\zeta_1, \ldots, \zeta_r)$, for $i = 1, \ldots, s$. By the definition of the $D^{(i)}$ matrices, it follows that

$$\mathcal{M}_h(y_0) = D^{(s)} \mathcal{L}_h^{(s)} \circ D^{(s-1)} \mathcal{L}_h^{(s-1)} \circ \cdots \circ D^{(2)} \mathcal{L}_h^{(2)} \circ D^{(1)} \mathcal{L}_h^{(1)}(D^{(0)} y_0),$$

as required. □

Example 6.11. Consider the G-symplectic and symmetric GLM given in [13]:

$$\left[\begin{array}{cccc|cc} \frac{1}{12} & 0 & 0 & 0 & 1 & \frac{1}{2} \\ -\frac{1}{3} & \frac{1}{6} & 0 & 0 & 1 & 1 \\ \frac{5}{3} & -\frac{2}{3} & \frac{1}{6} & 0 & 1 & -1 \\ \frac{7}{6} & -\frac{5}{12} & \frac{1}{12} & \frac{1}{12} & 1 & -\frac{1}{2} \\ \hline \frac{2}{3} & -\frac{1}{6} & -\frac{1}{6} & \frac{2}{3} & 1 & 0 \\ 1 & -\frac{1}{2} & \frac{1}{2} & -1 & 0 & -1 \end{array}\right].$$

As this method satisfies the assumptions of Theorem 6.10 it can be decomposed into several LMMs. In particular,

$$\mathcal{M}_h(y_0) = V D^{-1} \mathcal{L}_h^{(1)} \circ D \mathcal{L}_h^{(2)} \circ \mathcal{L}_h^{(2)} \circ D^{-1} \mathcal{L}_h^{(1)}(D y_0),$$

where $D = \operatorname{diag}(1, -\frac{1}{2})$ and

$$\begin{array}{c|cc} \frac{1}{12} & 1 & -1 \\ \hline \frac{2}{3} & 1 & 0 \\ -\frac{1}{2} & 0 & -1 \end{array} \qquad \begin{array}{c|cc} \frac{1}{6} & 1 & -1 \\ \hline -\frac{1}{6} & 1 & 0 \\ \frac{1}{2} & 0 & -1 \end{array}.$$

$$\mathcal{L}_h^{(1)} \qquad\qquad \mathcal{L}_h^{(2)}$$

Here, we note that a simplification arises from T-equivalence (where $T = D^{-1}$). The result of which is that

$$\mathcal{M}_h(y_0) \sim V\mathcal{L}_h^{(1)} \circ D\mathcal{L}_h^{(2)} \circ \mathcal{L}_h^{(2)} \circ D^{-1}\mathcal{L}_h^{(1)}(y_0).$$

Upon closer inspection of the tableaux for $\mathcal{L}_h^{(1)}$ and $\mathcal{L}_h^{(2)}$, we find that each method belongs to the family of 2-step, symmetric, second-order LMMs. To see this, we rescale the time-step in both methods such that they correspond to a time-h evolution, i.e.

$$\begin{array}{c|cc} \frac{1}{8} & 1 & -1 \\ \hline 1 & 1 & 0 \\ -\frac{3}{4} & 0 & -1 \end{array} \qquad \begin{array}{c|cc} -1 & 1 & -1 \\ \hline 1 & 1 & 0 \\ -3 & 0 & -1 \end{array}.$$

$$\mathcal{L}_{\frac{3}{2}h}^{(1)} =: L_h^{(1)} \qquad\qquad \mathcal{L}_{-6h}^{(2)} =: L_h^{(2)}$$

Then, we make a comparison to the tableau given in Example 6.5. Doing so, we find the corresponding LMMs are given by

$$\begin{aligned} L_h^{(1)} \qquad &\Longrightarrow \qquad y_{n+2} - y_n = \frac{h}{8}(f_{n+2} + 14f_{n+2} + f_n), \\ L_h^{(2)} \qquad &\Longrightarrow \qquad y_{n+2} - y_n = -h(f_{n+2} - 4f_{n+2} + f_n). \end{aligned}$$

Finally, we note that the decomposition for $\mathcal{M}_h$ may also be written in terms of $L_h^{(1)}$ and $L_h^{(2)}$:

$$\mathcal{M}_h(y_0) \sim \; VL_{\frac{2}{3}h}^{(1)} \circ DL_{-\frac{1}{6}h}^{(2)} \circ L_{-\frac{1}{6}h}^{(2)} \circ D^{-1}L_{\frac{2}{3}h}^{(1)}(y_0).$$

In this form, it is clearer to see that the composition on the RHS corresponds to a single time-h evolution.

$\diamond$

Concluding remarks: The decomposition result presented here provides an alternative approach to the implementation of structure-preserving GLMs, namely, as the composition of several single-stage GLMs. Under certain conditions on the coefficient matrices of the method, these single-stage GLMs can be equivalently expressed as LMMs. Interestingly, this situation closely resembles that considered by Donelson and Hansen [25] where the cyclic composition of LMMs was considered as a way to overcome the first Dahlquist barrier of multistep methods. Thus, further developments to the decomposition theory could look at incorporating the ideas behind this cyclic composition in the design of high-order structure-preserving GLMs.

Chapter 7

Numerical experiments

In this chapter, we perform a variety of numerical experiments that demonstrate some of the key results presented in the thesis. In particular, we will consider the

- verification of the predicted parasitism-free behaviour,
- implementation of higher-order composition methods,
- comparison of work/efficiency of GLMs with implicit RKMs,
- long-time preservation properties of symmetric/G-symplectic GLMs.

7.1 Geometric problems

In the following section, we introduce various classical problems that we will consider for our numerical experiments. The chosen problems each possess at least one invariant so as to provide a simple measure for the effectiveness of G-symplectic/symmetric methods as structure-preserving integrators.

7.1.1 Hamiltonian

Recall the $2m$-dimensional, $m \in \mathbb{N}$, Hamiltonian IVP (2.2):

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} p(t) \\ q(t) \end{bmatrix} = \begin{bmatrix} -\nabla_q H(p(t), q(t)) \\ \nabla_p H(p(t), q(t)) \end{bmatrix}, \qquad \begin{bmatrix} p(0) \\ q(0) \end{bmatrix} = \begin{bmatrix} p_0 \\ q_0 \end{bmatrix}, \qquad t \in [-T, T],$$

where $H : X \to \mathbb{R}$ is the Hamiltonian, $q, p : [-T, T] \to \mathbb{R}^m$, and $p_0, q_0 \in \mathbb{R}^m$.

Simple pendulum (SP): This problem describes the motion of a pendulum with unit mass and length, and with time scaled such that gravity $g = 1$ (see e.g. [36, p. 5]). The corresponding Hamiltonian is written as

$$H(p, q) = \frac{1}{2}p^2 - \cos(q).$$

Double pendulum (DP): The double pendulum problem describes the motion, under gravity, of two connected pendulums (see e.g. [36, p. 233]). Here, we assume both pendulums are of unit mass and length, with gravity rescaled to $g = 1$. The corresponding Hamiltonian is written as

$$H(p_1, p_2, q_1, q_2) = \frac{p_1^2 + 2p_2^2 - 2p_1p_2\cos(q_1 - q_2)}{2[1 + \sin^2(q_1 - q_2)]} - \cos(q_2) - 2\cos(q_1).$$

Kepler (KPL): The Kepler problem describes the motion of two celestial bodies under mutual gravitational attraction (see e.g. [36, pp. 8–12]). Here, we centre our coordinate system about the centre of mass, and assume unit masses and a scalar potential of the form $V(r) = -\frac{1}{r}$ such that the Hamiltonian may be written as

$$H(p_1, p_2, q_1, q_2) = \frac{1}{2}\left(p_1^2 + p_2^2\right) + V\left(||q_2 - q_1||_2\right).$$

In addition to the Hamiltonian, a second invariant is the angular momentum which is defined as the quadratic function

$$L(p_1, p_2, q_1, q_2) = q_1p_2 - q_2p_1.$$

Hénon–Heiles (HH): The Hénon–Heiles model (see e.g. [36, p. 15]) describes stellar motion inside a gravitational potential of a galaxy, with cylindrical symmetry. The defining Hamiltonian is given by

$$H(p_1, p_2, q_1, q_2) = \frac{1}{2}\left(p_1^2 + p_2^2\right) + \frac{1}{2}\left(q_1^2 + q_2^2\right) + q_1^2q_2 - \frac{1}{3}q_2^3.$$

Bead on a wire (BOW): The motion of a bead on a rigid wire can be described as a Hamiltonian system (see e.g. [2]) with

$$H(p, q) = \frac{1}{2(1 + U'(q)^2)}p^2 + U(q), \qquad \text{where} \qquad U(q) = \frac{1}{10}q^2(q - 2)^2 + \frac{8}{1000}q^3.$$

Galactic dynamics (GD): The galactic dynamics problem describes the motion of a single star in a galaxy under the potential of all the remaining stars (see e.g. [2],[38, pp. 319–325]). The corresponding Hamiltonian is given by

$$H(p_1,p_2,p_3,q_1,q_2,q_3) = \frac{1}{2}\left(p_1^2+p_2^2+p_3^2\right)+\Omega(p_1q_2-p_2q_1)+A\ln\left(C+\frac{q_1^2}{a^2}+\frac{q_2^2}{b^2}+\frac{q_3^2}{c^2}\right),$$

where

$$\Omega = 0.25, \qquad A=1, \quad C=1, \quad a=1.25, \quad b=1, \quad c=0.75.$$

7.1.2 Non-Hamiltonian

The following problem is not Hamiltonian in the usual sense, but can be reformulated as a constrained Hamiltonian system [36, Ch. VII.5].

Rigid Body (RB): This problem describes the motion of a rigid body, with its mass centred at the origin (see e.g. [36, pp. 99–100]). Here, the governing equations are given as

$$\begin{aligned}
\frac{\mathrm{d}y_1}{\mathrm{d}t} &= a_1y_2y_3, \qquad a_1 = \frac{(I_2-I_3)}{I_2I_3},\\
\frac{\mathrm{d}y_2}{\mathrm{d}t} &= a_2y_3y_1, \qquad a_2 = \frac{(I_3-I_1)}{I_3I_1},\\
\frac{\mathrm{d}y_3}{\mathrm{d}t} &= a_3y_1y_2, \qquad a_3 = \frac{(I_1-I_2)}{I_1I_2},
\end{aligned}$$

where we have chosen the principal moments of inertia to take the values of

$$I_1 = 2, \qquad I_2 = 1, \qquad I_3 = \frac{2}{3}.$$

The problem possesses two quadratic invariants of the form

$$\begin{aligned}
Q_1(y_1,y_2,y_3) &= y_1^2+y_2^2+y_3^2,\\
Q_2(y_1,y_2,y_3) &= \frac{1}{2}\left(\frac{y_1^2}{I_1}+\frac{y_2^2}{I_2}+\frac{y_3^2}{I_3}\right).
\end{aligned}$$

7.2 Numerical methods for experiments

For our numerical experiments several GLMs have been chosen, each of which possesses at least one structure-preserving property. We have also chosen several RKMs to serve as comparison methods. We note that symmetric LMMs for first order ODEs will not be considered for comparison as these are known to suffer from parasitism on $t = O(1)$ intervals. Also, we do not consider multistep methods for second order differential equations as these are not necessarily applicable to all of the test problems described in the previous section.

Each method will be referred to by a four-digit number of the form $pqrs$ which corresponds to the order, stage order, number of inputs and stages of the method, followed by additional characters describing the properties of the methods:

- G - denotes a G-symplectic/symplectic method,
- S - denotes a symmetric method.

7.2.1 GLMs

We have selected the following 4 GLMs for our experiments:

- **GLM-4123G**: G-symplectic GLM given in [17].
- **GLM-4123S**: Symmetric GLM with a single implicit stage (cf. (2.26)).
- **GLM-4125S**: Symmetric GLM that is parasitism-free to 4th order (cf. (3.10)).
- **GLM-4124GS**: G-symplectic and symmetric GLM given in [13].

The (A, U, B) coefficient matrices for each method are given in Table 7.1, and the (A_S, B_S) coefficient matrices for each starting method are given in Table 7.2. Each method shares the same V matrix, and preconsistency vectors u and w:

$$V = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \qquad u = w = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

Remark 7.1. The principal component of the starting method for **GLM-4123G** is non-trivial, i.e. $w^{\mathsf{H}} B_S \neq 0$. Thus, we also require a finishing method. This is given by the tableau below:

$$\left[\begin{array}{ccc|cc} 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 & 0 \\ -\frac{25}{81} & -\frac{20}{81} & 0 & 1 & 0 \\ \hline -\frac{1}{60} & -\frac{1}{48} & \frac{3}{80} & 1 & 0 \end{array}\right].$$

The corresponding starting method was constructed such that this finishing method is explicit and $\mathcal{F}_h \circ \mathcal{S}_h(y_0) = y_0$ exactly. As a consequence, the stage equations of the starting method suffer from a higher degree of implicitness. However, since this is only applied once, the additional computational cost is often negligible over long-time integrations.

It should be noted that if we allowed $\mathcal{F}_h \circ \mathcal{S}_h(y_0) = y_0 + O(h^{p+1})$, then we could have also constructed an explicit starting method. As this case is not covered elsewhere in the thesis, we have not considered it for practical computations.

Verification of method properties: In Table 7.3, we list the (D, G) matrices of the G-symplectic GLMs and the (L, P)-matrices of the symmetric GLMs, along with the P_S matrix corresponding to the (L, P_S)-symmetric starting methods.

7.2.2 RKMs

For comparison, we have selected the following RKMs:

- **RK-4212GS**: Gauss method of order 4 (see e.g. [36, p. 34]).
- **RK-4113GS**: Triple-jump of the implicit midpoint rule (see e.g. [52]).
- **RK-4113S**: Lobatto IIIB method of order 4 (see e.g. [36, p. 37]).
- **RK-6117**: Explicit, 6th order method given in [4].
- **RK-6313GS**: Gauss method of order 6 (see e.g. [36, p. 34]).

The (A, B) coefficient matrices of each method are given in Table 7.4. Recall that for an RKM, $V = 1$ and $U = \mathbb{1}$.

Method	A	U	B	G-symplectic	Symmetric
GLM-4123G	$\begin{bmatrix} \frac{1}{3} & 0 & 0 \\ \frac{5}{6} & \frac{1}{6} & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & \frac{1}{3} \\ 1 & -\frac{1}{3} \\ 1 & 1 \end{bmatrix}$	$\begin{bmatrix} \frac{3}{4} & \frac{3}{8} & -\frac{1}{8} \\ \frac{1}{4} & -\frac{1}{8} & -\frac{1}{8} \end{bmatrix}$	✓	×
GLM-4123S	$\begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{3}{2} & \frac{1}{2} & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 \\ 1 & -2 \\ 1 & -2 \end{bmatrix}$	$\begin{bmatrix} \frac{2}{3} & \frac{1}{6} & \frac{1}{6} \\ \frac{2}{3} & \frac{1}{6} & \frac{1}{6} \end{bmatrix}$	×	✓
GLM-4125S	$\begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 & 0 \\ \frac{7}{12} & \frac{5}{24} & 0 & 0 & 0 \\ -\frac{1}{12} & \frac{1}{2} & \frac{1}{24} & 0 & 0 \\ \frac{13}{12} & \frac{1}{2} & -\frac{1}{2} & -\frac{13}{24} & 0 \\ \frac{5}{12} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{7}{24} \end{bmatrix}$	$\begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \\ 0 & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{bmatrix}$	×	✓
GLM-4124GS	$\begin{bmatrix} \frac{1}{12} & 0 & 0 & 0 \\ -\frac{1}{3} & \frac{1}{6} & 0 & 0 \\ \frac{5}{3} & -\frac{2}{3} & \frac{1}{6} & 0 \\ \frac{7}{6} & -\frac{5}{12} & \frac{1}{12} & \frac{1}{12} \end{bmatrix}$	$\begin{bmatrix} 1 & \frac{1}{2} \\ 1 & 1 \\ 1 & -1 \\ 1 & -\frac{1}{2} \end{bmatrix}$	$\begin{bmatrix} \frac{2}{3} & -\frac{1}{6} & -\frac{1}{6} & \frac{2}{3} \\ 1 & -\frac{1}{2} & \frac{1}{2} & -1 \end{bmatrix}$	✓	✓

Table 7.1: Coefficient matrices of the chosen GLMs and their properties.

Method	A_S	B_S	Symmetric
GLM-4123G	$\begin{bmatrix} \frac{1}{60} & \frac{1}{48} & -\frac{3}{80} & 0 \\ -\frac{59}{60} & \frac{1}{48} & -\frac{3}{80} & 0 \\ -\frac{473}{1620} & -\frac{293}{1296} & -\frac{3}{80} & 0 \\ -\frac{688}{2825} & -\frac{491}{8064} & -\frac{5531}{362880} & 0 \end{bmatrix}$	$\begin{bmatrix} \frac{1}{60} & \frac{1}{48} & -\frac{3}{80} & 0 \\ -\frac{541}{1020} & -\frac{11}{444} & -\frac{18}{65} & \frac{6804}{8177} \end{bmatrix}$	$\times$
GLM-4123S	$\begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 \\ -\frac{1}{2} & 0 & 0 & 0 \\ 0 & -\frac{1}{10} & \frac{1}{10} & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{5}{12} & -\frac{1}{6} & -\frac{1}{6} & \frac{5}{12} \end{bmatrix}$	✓
GLM-4125S	$\begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{1}{4} & 0 & 0 & 0 \\ -\frac{1}{4} & 0 & 0 & 0 \\ 0 & \frac{1}{4} & -\frac{1}{4} & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 \\ -\frac{1}{3} & \frac{1}{3} & \frac{1}{3} & -\frac{1}{3} \end{bmatrix}$	✓
GLM-4124GS	$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{373}{550} & \frac{177}{550} & 0 & 0 & 0 & 0 & 0 \\ \frac{8233}{50976} & -\frac{30749}{152928} & \frac{3025}{76464} & 0 & 0 & 0 & 0 \\ -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{373}{550} & 0 & 0 & 0 & -\frac{177}{550} & 0 & 0 \\ -\frac{8233}{50976} & 0 & 0 & 0 & \frac{30749}{152928} & -\frac{3025}{76464} & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{383}{1296} & \frac{275}{2592} & \frac{1}{2} & \frac{383}{1296} & -\frac{275}{2592} & -\frac{1}{2} \end{bmatrix}$	✓

Table 7.2: Starting methods and their properties for the GLMs in Table 7.1.

Method	D	G	L	P	P_S
GLM-4123G	$\begin{bmatrix} \frac{3}{4} & 0 & 0 \\ 0 & \frac{3}{8} & 0 \\ 0 & 0 & -\frac{1}{8} \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$	$\times$	$\times$	$\times$
GLM-4123S	$\times$	$\times$	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
GLM-4125S	$\times$	$\times$	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
GLM-4124GS	$\begin{bmatrix} \frac{2}{3} & 0 & 0 & 0 \\ 0 & -\frac{1}{6} & 0 & 0 \\ 0 & 0 & -\frac{1}{6} & 0 \\ 0 & 0 & 0 & \frac{2}{3} \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & -\frac{1}{3} \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$

Table 7.3: Structure-preserving matrices for the GLMs of Table 7.1.

Method	A	B	Symplectic	Symmetric
RK-4212GS	$\begin{bmatrix} \frac{1}{4} & \frac{1}{4}-\frac{\sqrt{3}}{6} \\ \frac{1}{4}+\frac{\sqrt{3}}{6} & \frac{1}{4} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{2} & \frac{1}{2} \end{bmatrix}$	✓	✓
RK-4113GS	$\begin{bmatrix} \frac{1}{4-2^{4/3}} & 0 & 0 \\ \frac{1}{2-2^{1/3}} & -\frac{2^{1/3}}{4-2^{4/3}} & 0 \\ \frac{1}{2-2^{1/3}} & -\frac{2^{1/3}}{2-2^{1/3}} & \frac{1}{4-2^{4/3}} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{2-2^{1/3}} & -\frac{2^{1/3}}{2-2^{1/3}} & \frac{1}{2-2^{1/3}} \end{bmatrix}$	✓	✓
RK-4113S	$\begin{bmatrix} \frac{1}{6} & -\frac{1}{6} & 0 \\ \frac{1}{6} & \frac{1}{3} & 0 \\ \frac{1}{6} & \frac{5}{6} & 0 \end{bmatrix}$	$\begin{bmatrix} \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{bmatrix}$	×	✓
RK-6117	$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{2}{9} & \frac{4}{9} & 0 & 0 & 0 & 0 & 0 \\ \frac{7}{36} & \frac{2}{9} & -\frac{1}{12} & 0 & 0 & 0 & 0 \\ -\frac{35}{144} & -\frac{55}{36} & \frac{35}{48} & \frac{15}{8} & 0 & 0 & 0 \\ -\frac{1}{360} & -\frac{11}{36} & -\frac{1}{8} & \frac{1}{2} & \frac{1}{10} & 0 & 0 \\ -\frac{41}{260} & \frac{22}{13} & \frac{43}{156} & -\frac{118}{39} & \frac{32}{195} & \frac{80}{39} & 0 \end{bmatrix}$	$\begin{bmatrix} \frac{13}{200} & 0 & \frac{11}{40} & \frac{11}{40} & \frac{4}{25} & \frac{4}{25} & \frac{13}{200} \end{bmatrix}$	×	×
RK-6313GS	$\begin{bmatrix} \frac{5}{36} & \frac{2}{9}-\frac{\sqrt{15}}{15} & \frac{5}{36}-\frac{\sqrt{15}}{30} \\ \frac{5}{36}+\frac{\sqrt{15}}{24} & \frac{2}{9} & \frac{5}{36}-\frac{\sqrt{15}}{24} \\ \frac{5}{36}+\frac{\sqrt{15}}{30} & \frac{2}{9}+\frac{\sqrt{15}}{15} & \frac{5}{36} \end{bmatrix}$	$\begin{bmatrix} \frac{5}{18} & \frac{4}{9} & \frac{5}{18} \end{bmatrix}$	✓	✓

Table 7.4: Comparison RKMs and their properties.

7.3 Parasitism

In the following experiments we look to confirm the theoretical parasitism results of Chapter 3, i.e. for a given method, we estimate the interval over which the numerical solution is computationally parasitism-free. From the conclusions of Theorem 3.35, we expect this interval to be $t = O(h^{-M})$ for an Mth-order parasitism-free GLM. It is also expected that the invariants of the problems considered are to be preserved to $O(h^p)$, where p denotes the order of the method, over this interval (see also [23]).

Experiments are performed on either the simple pendulum **(SP)** problem with initial data $(p_0, q_0) = (1, 2)$, or the bead on a wire **(BOW)** problem with initial data $(p_0, q_0) = (0.49, 0)$. As output, we monitor the error in the Hamiltonian, $H(p_n, q_n) - H(p_0, q_0)$, at every step.

GLM-4124GS: Using the practical toolkit (cf. Chapter 4), we find that the DUOSM corresponding to the eigenvalue $\zeta_2 = -1$ is given by

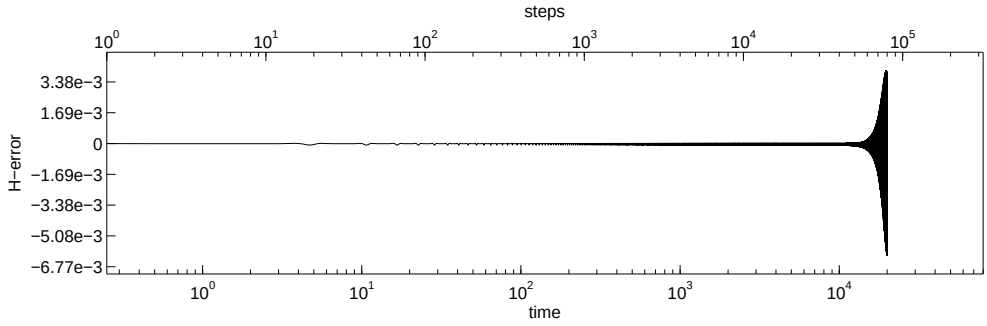
$$\Psi_h^{(2)}(y, v) = -v - \frac{h^3}{144}\Big(f'(y)f'(y)f'(y)v - 5f'(y)f''(y)(f(y), v) + 5f''(y)(f(y), f'(y)v) + 3f''(y)(f'(y)f(y), v) + \frac{39}{2}f'''(y)(f(y), f(y), v)\Big) + O(h^4).$$

This implies that the method is 2nd-order parasitism-free, i.e. $M = 2$, and thus we expect a parasitism-free interval of $t = O(h^{-2})$. In order to numerically confirm this, we first integrate the simple pendulum problem with a time-step $h = 0.25$ over the interval $[0, 2\times10^4]$. This is sufficient to capture the onset of parasitic growth, as is demonstrated in Figure 7-1a. Next, we halve the time-step and repeat the experiment over the longer interval $[0, 8 \times 10^4]$, as is shown in Figure 7-1b. These results demonstrate that halving the time-step approximately increases the interval of parasitism-free behaviour by 4, and therefore agrees with the predicted $t = O(h^{-2})$. A similar experiment with this method has been performed in [23] which also arrives at the same conclusion.

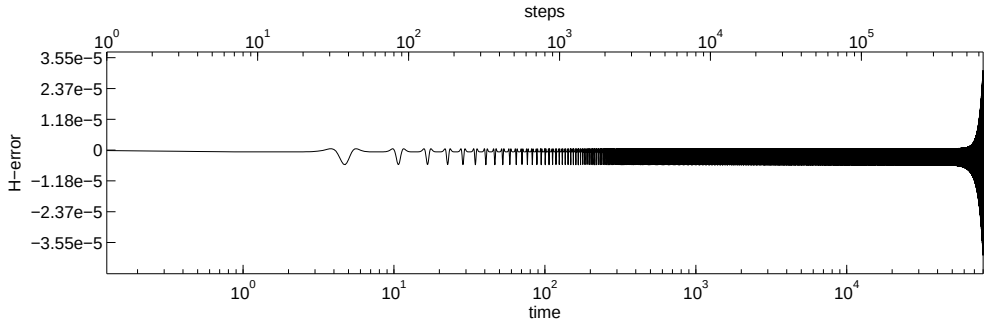
GLM-4123G: The DUOSM corresponding to the eigenvalue $\zeta_2 = -1$ is given by

$$\Psi_h^{(2)}(y_0, v) = -v - \frac{h^2}{18}f''(y_0)(f(y_0), v) + O(h^3).$$

This suggests that the method should remain parasitism-free for an interval of $t = O(h^{-1})$. However, the results in Figure 7-2 seem to suggest that it is actually closer to $t = O(h^{-2})$. To explain this, we make the following observation:



(a)



(b)

Figure 7-1: Estimation of the interval of parasitism-free behaviour for GLM-4124GS applied to **(SP)** with initial data $(p_0, q_0) = (1, 2)$. (a) Hamiltonian error with $h = 0.25$ over $[0, 2 \times 10^4]$ (b) Hamiltonian error with $h = 0.125$ over $[0, 8 \times 10^4]$.

Recall that there exists a unique pair $(S_h^P(y_0, v), \Psi_h^P(y_0, v))$, expressed as formal DB-series, such that

$$\mathcal{M}_h'(\mathcal{S}_h^*(y_0))S_h^P(y_0, v) = S_h^P(\Phi_h(y_0), \Psi_h^P(y_0, v)), \qquad \text{and} \qquad F_h^P(y_0, S_h^P(y_0, v)) = v,$$

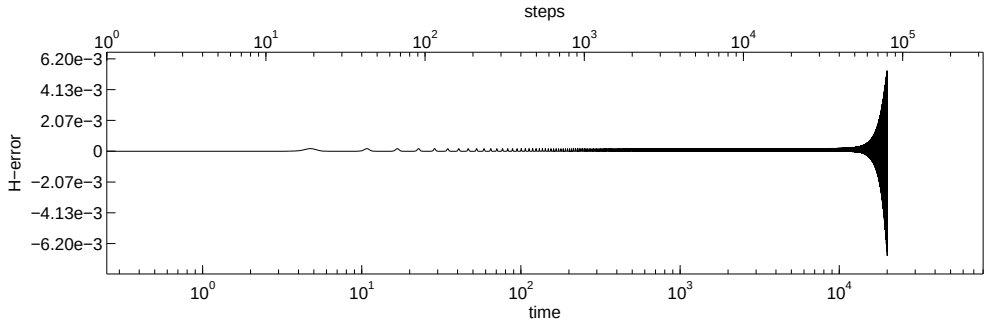
hold for some fixed (row-vector) DB-series $F_h^P(y_0, \cdot)$. Now, let $\chi_h^P(y_0, v)$ be an arbitrary invertible DB-series. Then, the following pair,

$$S_h^P(y_0, \cdot)[\chi_h^P(y_0, v)]^{-1}, \qquad \text{and} \qquad \chi_h^P(\Phi_h(y_0), \cdot)\Psi_h^P(y_0, \cdot)[\chi_h^P(y_0, v)]^{-1},$$

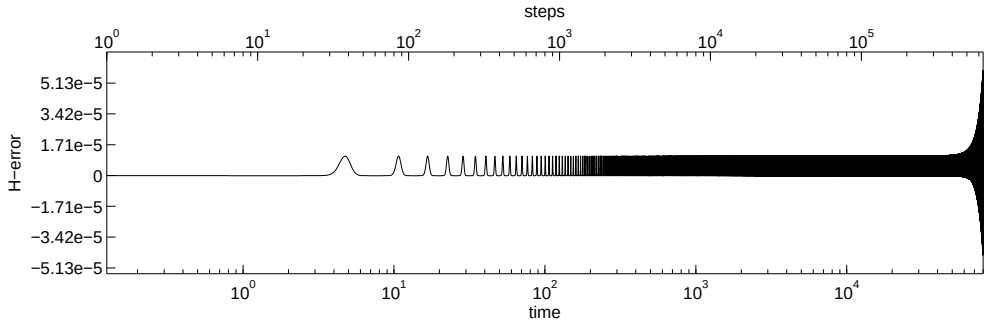
is the corresponding unique solution when $F_h^P(y_0, \cdot) \mapsto \chi_h^P(y_0, \cdot)F_h^P(y_0, \cdot)$.

Suppose now that we fix $\chi_h^{(2)}(y_0, v) = v - \frac{h}{18}f'(y_0)v$, then

$$\chi_h^{(2)}(\Phi_h(y_0), v) = v - \frac{h}{18}f'(y_0)v - \frac{h^2}{18}f''(y_0)(f(y_0), v) + O(h^3),$$
$$[\chi_h^{(2)}(y_0, v)]^{-1} = v + \frac{h}{18}f'(y_0)v + \frac{h^2}{18^2}f'(y_0)f'(y_0)v + O(h^3),$$



(a)



(b)

Figure 7-2: Estimation of the interval of parasitism-free behaviour for GLM-4123G applied to **(SP)** with initial data $(p_0, q_0) = (1, 2)$. (a) Hamiltonian error with $h = 0.25$ over $[0, 2 \times 10^4]$ (b) Hamiltonian error with $h = 0.125$ over $[0, 8 \times 10^4]$.

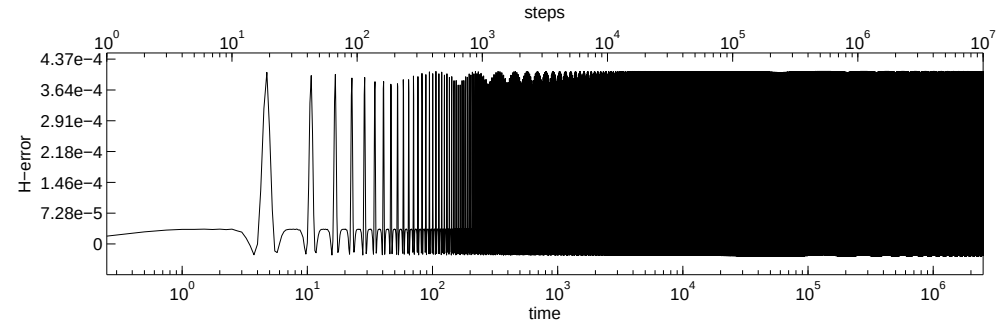
and

$$\chi_h^{(2)}(\Phi_h(y_0), \cdot)\Psi_h^{(2)}(y_0, \cdot)[\chi_h^{(2)}(y_0, v)]^{-1} = -v + O(h^3).$$

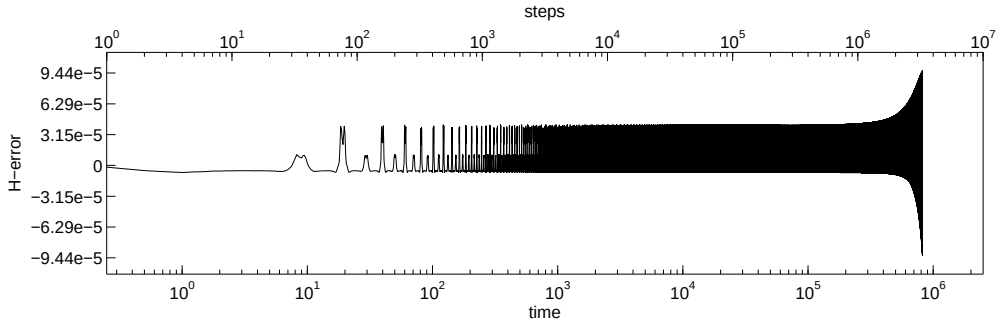
Thus, if we instead choose to define $F_h^{(2)}(y_0, \cdot) = \chi_h^{(2)}(y_0, \cdot)w_2^{\mathsf{H}}$ (as opposed to the original choice of $F_h^{(2)}(y_0, \cdot) = w_2^{\mathsf{H}}$) we find that method has a DUOSM that is parasitism-free to order 2, which now agrees with the computational results.

Remark 7.2. Note that the modification made to $F_h^{(2)}(y_0, \cdot)$ above does not imply that we have to alter the practical finishing method to achieve second-order parasitism-free behaviour.

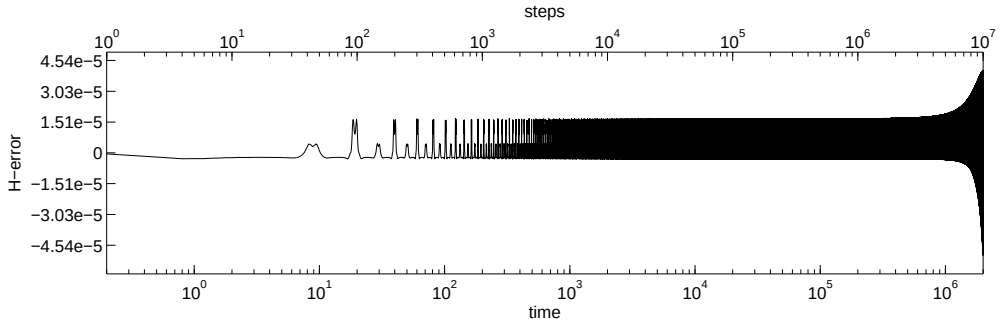
GLM-4125S: Recall from the end of Chapter 3 that this method was designed to be parasitism-free to order 4. Thus, by performing a similar experiment to those given above, we expect to observe a parasitism-free interval of $t = O(h^{-4})$: Consider the simple pendulum problem where the method is applied with a fixed time-step of $h = 0.25$ over an interval of $[0, 2.5 \times 10^6]$. The results in Figure 7-3a show no observable



(a)

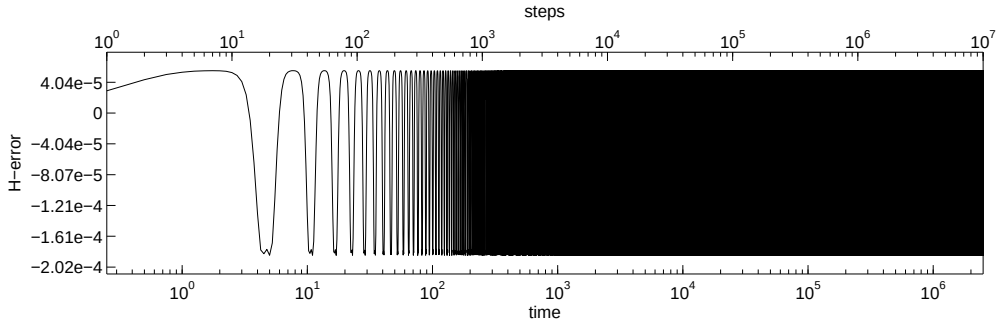


(b)

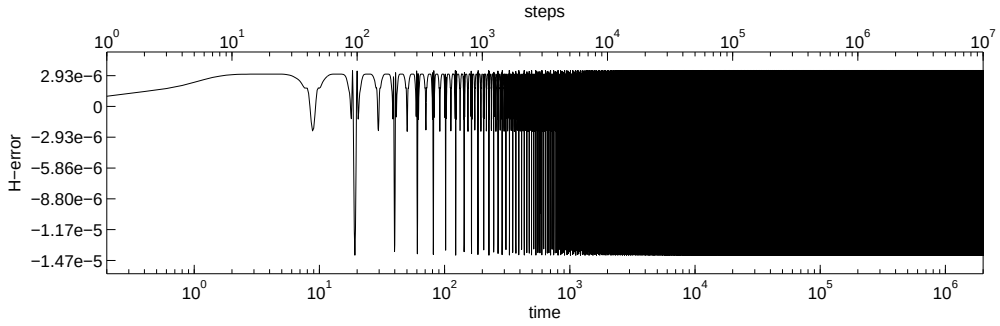


(c)

Figure 7-3: Estimation of the interval of parasitism-free behaviour for GLM-4125S. (a) Hamiltonian error for **(SP)** with initial data $(p_0, q_0) = (1, 2)$, time-step $h = 0.25$ and an integration interval of $[0, 2.5 \times 10^6]$. (b) Hamiltonian error for **(BOW)** with initial data $(p_0, q_0) = (0.49, 0)$, time-step $h = 0.25$ and an integration interval of $[0, 8.2 \times 10^5]$. (c) Hamiltonian error for **(BOW)** with initial data $(p_0, q_0) = (0.49, 0)$, time-step $h = 0.2$ and an integration interval of $[0, 2 \times 10^6]$.



(a)



(b)

Figure 7-4: Demonstration of long-time parasitism-free behaviour for GLM-4123S. (a) Hamiltonian error for **(SP)** with initial data $(p_0, q_0) = (1, 2)$, time-step $h = 0.25$ and an integration interval of $[0, 2.5 \times 10^6]$. (b) Hamiltonian error for **(BOW)** with initial data $(p_0, q_0) = (0.49, 0)$, time-step $h = 0.25$ and an integration interval of $[0, 2.5 \times 10^6]$.

parasitic growth and we are therefore unable to quantify the interval of parasitism-free behaviour for this problem.

Next, we consider the bead on the wire problem with a fixed time-step of $h = 0.25$ over an interval of $[0, 8.2 \times 10^5]$. In this case, the onset of parasitism can be observed in the plot of Figure 7-3b. Repeating this experiment with a fixed time-step of $h = 0.2$ over the interval $[0, 2 \times 10^6]$ we find that the onset occurs approximately 2.44 times later. This agrees with the $t = O(h^{-4})$ estimate since $(0.2/0.25)^{-4} \approx 2.44$.

GLM-4123S: The DUOSM corresponding to eigenvalue $\zeta_2 = -1$ is given by

$$\Psi_h^{(2)}(y, v) = -v - \frac{h^3}{12}\Big(f'(y)f'(y)f'(y)v - 2f'(y)f''(y)(f(y), v) - f''(y)(f(y), f'(y)v) + 4f''(y)(f'(y)f(y), v) + f'''(y)(f(y), f(y), v) \Big) + O(h^4),$$

which suggests a parasitism-free interval of $t = O(h^{-2})$. However, experiments on both the simple pendulum and bead on a wire have failed to capture the onset of parasitic behaviour (see Figures 7-4a and 7-4b). Due to the limitations of our computational resources we have been unable to verify the length of the parasitism-free interval, though the results seem to imply that this has been achieved.

We remark that, unlike GLM-4123G, no choice of $\chi_h^{(2)}(y_0, v)$ will yield an alternative DUOSM that is parasitism-free to a higher order. Thus, we suspect that the good behaviour of this method is due to a significantly small constant in the $t = O(h^{-2})$ term.

7.4 Composition

In the following set of experiments, we apply the composition results of Chapter 5 to construct higher order methods and then numerically verify that the theoretical order increase has been achieved. The compositions we consider are the triple jump (5.19):

$$\mathcal{M}_{\alpha_1 h} \circ R(\alpha_1, \alpha_2) \circ \mathcal{M}_{\alpha_2 h} \circ R(\alpha_2, \alpha_1) \circ \mathcal{M}_{\alpha_1 h},$$

where

$$\alpha_1 = \frac{1}{2 - 2^{1/(p+1)}}, \qquad \alpha_2 = -\frac{2^{1/(p+1)}}{2 - 2^{1/(p+1)}},$$

and the Suzuki 5-jump (5.20):

$$\mathcal{M}_{\alpha_1 h} \circ R(\alpha_1, \alpha_1) \circ \mathcal{M}_{\alpha_1 h} \circ R(\alpha_1, \alpha_2) \circ \mathcal{M}_{\alpha_2 h} \circ R(\alpha_2, \alpha_1) \circ \mathcal{M}_{\alpha_1 h} \circ R(\alpha_1, \alpha_1) \circ \mathcal{M}_{\alpha_1 h},$$

where

$$\alpha_1 = \frac{1}{4 - 4^{1/(p+1)}}, \qquad \alpha_2 = -\frac{4^{1/(p+1)}}{4 - 4^{1/(p+1)}}.$$

It is assumed that the base method, $\mathcal{M}_h$, is symmetric. Given that all the symmetric methods in Table 7.1 have trivial finishing methods, the transformations $R(\alpha_2, \alpha_1)$ and $R(\alpha_1, \alpha_1)$ are described by the GLM tableaux

$$\left[\begin{array}{cc|c} A_S\alpha_2 & 0 & \mathbb{1}_S w^{\mathsf{H}} \\ 0 & A_S\alpha_1 & \mathbb{1}_S w^{\mathsf{H}} \\ \hline -V^{-1}B_S\alpha_2 & B_S\alpha_1 & V^{-1} \end{array}\right], \qquad \left[\begin{array}{c|c} A_S\alpha_1 & \mathbb{1}_S w^{\mathsf{H}} \\ \hline (I - V^{-1})B_S\alpha_1 & V^{-1} \end{array}\right].$$

$R(\alpha_2, \alpha_1)$ $\qquad\qquad$ $R(\alpha_1, \alpha_1)$

The tableau for $R(\alpha_1, \alpha_2)$ is found by swapping the α_1 and α_2 in the tableau of $R(\alpha_2, \alpha_1)$. For each of these transformations, we identify and remove any reducible stages prior to integration (cf. the end of Chapter 5 for details on stage reductions).

The numerical verification of order is performed on the Kepler problem **(KPL)** with initial data

$$[p_1(0), p_2(0), q_1(0), q_2(0)] = \left[0, \sqrt{\frac{1+e}{1-e}}, 1-e, 0\right], \qquad e = 0.6,$$

over an integration interval of $[0, 7.5]$. A similar composition experiment has been conducted in [36, p. 46] where the solution at time $t = 7.5$, in quadruple precision, is given as

$$\begin{aligned}
p_1(7.5) &= -0.85638471534339535151524486215030,\\
p_2(7.5) &= -0.16055215079983843525441910410 2,\\
q_1(7.5) &= -0.82816440269077081820475758537 0,\\
q_2(7.5) &= \quad 0.77889809565863544708165448079 6.
\end{aligned}$$

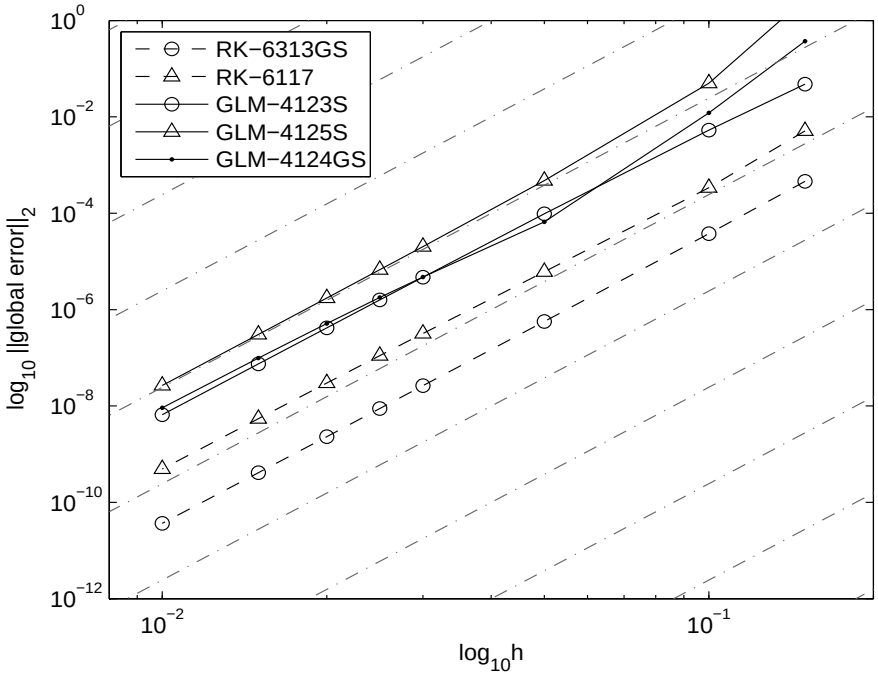
This will be used as our reference solution for obtaining values on global error.

Triple jump: An order plot for the triple jump composition of the methods GLM-4123S, GLM-4125S and GLM-4124GS is given in Figure 7-5a. On the y-axis we have the $\log_{10}$ of the 2-norm of the global error evaluated at $t = 7.5$, and on the x-axis we have the $\log_{10}$ of the time-step. As reference, we have also included order plots for RK-6117 and RK-6313GS.

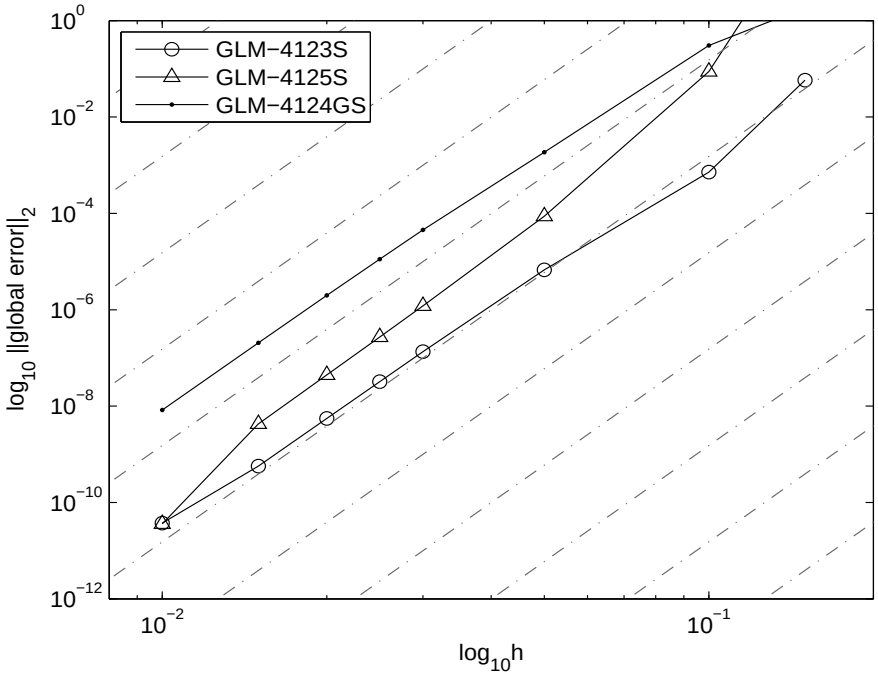
These results indicate that the composed methods achieve order $p = 6$. This agrees with the symmetric composition theory covered in Chapter 5 since each method (and its corresponding starting and finishing method) is known to be symmetric.

In Figure 7-5b, results of a triple jump of a triple jump are given where the composed methods reach an order of $p = 8$. This again agrees with the symmetric composition theory and demonstrates that the composition technique may be used to construct methods of arbitrarily high order.

Suzuki 5-jump: Figures 7-6a and 7-6b give similar composition results for a Suzuki 5-jump. While a greater number of stage evaluations are required for these methods, we note that the value of global error is significantly smaller than with the triple jump. For the 8th order Suzuki methods, we observe that method accuracy is limited by machine precision for time-steps $h < 0.04$.



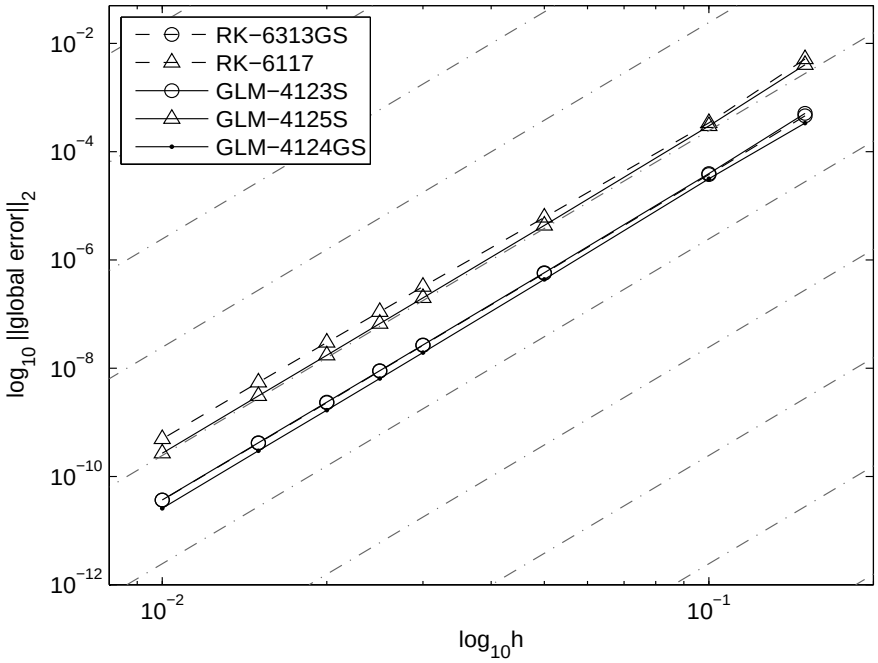
(a)



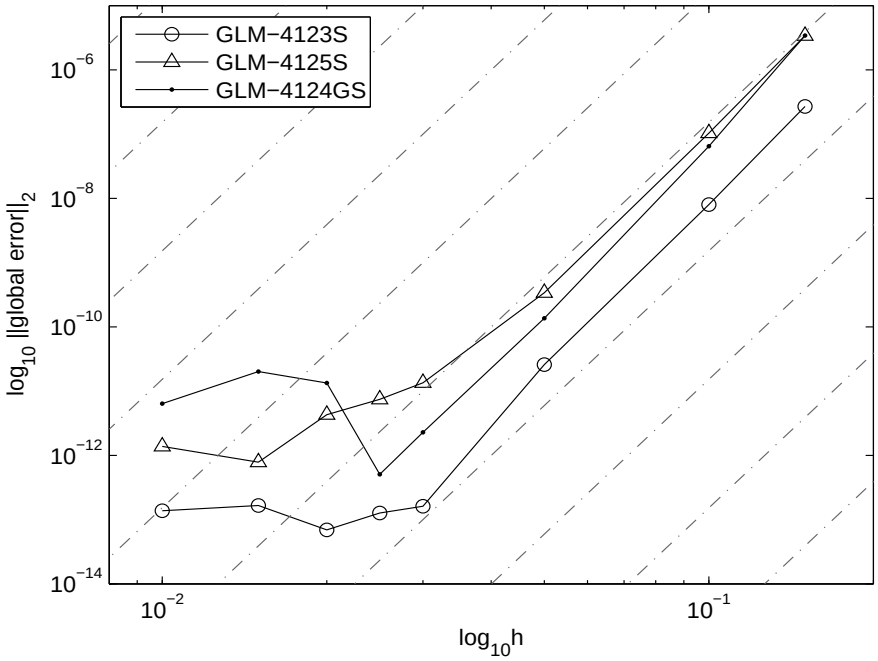
(b)

Figure 7-5: Order demonstration for 6th and 8th order triple jump methods on **(KPL)** (a) triple jump for 6th order. (b) triple jump of a triple jump for 8th order. Reference lines of gradients 6 and 8 are given by the grey dash-dot lines.

.



(a)



(b)

Figure 7-6: Order demonstration of 6th and 8th order Suzuki 5-jump methods on **(KPL)** (a) Suzuki for 6th order. (b) A double Suzuki for 8th order. Reference lines of gradients 6 and 8 are given by the grey dash-dot lines.

.

7.5 Efficiency

In the following experiments, we make an efficiency comparison of GLMs to RKMs. Here, efficiency is defined as the cost of the method versus the global error, where cost is taken to be the total number of function evaluations made over an integration. For fair comparison, the stage equations for each method are solved using a fixed point iteration where termination occurs when the (absolute) difference of two successive iterates is no greater than 10^{-12}.

The problems we will consider are the simple pendulum **(SP)**, Hénon–Heiles **(HH)** and galactic dynamics **(GD)**. These have been chosen so that we might also investigate the impact of a problem's dimension on the efficiency of a method.

Simple pendulum: For initial data $(p(0), q(0)) = (1, 2)$, we compute the solution at time $t = 15$ using RK-6117 with a time-step of $h = 0.005$:

$$p(15) = -0.661387597436204, \qquad q(15) = 2.342601503807022.$$

Here, the final time corresponds to a little over one period.

In Figure 7-7, we have efficiency plots for all 4th order methods applied to **(SP)**. The most efficient appears to be the RK-4212GS (Gauss) method which gives the best global error for a fixed number of function evaluations. Of the GLMs, we find that GLM-4123S performs the best due to having only one implicit stage to solve each iteration. In contrast, GLM-4125S is poorest as result of having to solve 5 implicit stage equations every iteration.

Hénon–Heiles: For initial data $(p_1(0), p_2(0), q_1(0), q_2(0)) = \left(\frac{1}{3}, \frac{1}{10}, 0, \frac{1}{4}\right)$, we compute the solution at time $t = 38$ using RKM-6117 with a time-step of $h = 0.005$:

$$\begin{aligned} p_1(38) &= 0.284229861508927, & q_1(38) &= 0.142467969999536, \\ p_2(38) &= 0.009550106944305, & q_2(38) &= 0.272937263697556. \end{aligned}$$

This particular choice of initial data ensures that the solution is non-chaotic and, with the given integration interval, almost forms a closed orbit when projected onto the (q_1, q_2) plane.

The results given in Figure 7-8 show that RK-4212GS performs best overall, though now only marginally when compared to GLM-4123S. Another notable observation is that RK-4113GS performs poorest in terms of efficiency.

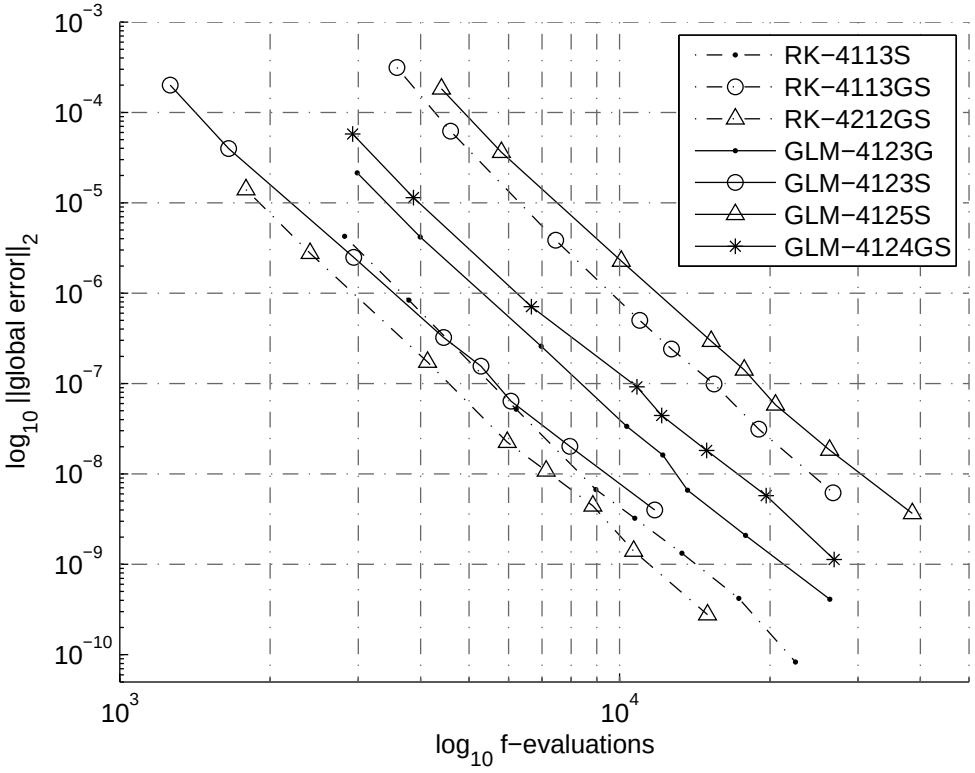


Figure 7-7: Efficiency plots of 4th order methods applied to **(SP)**.

.

Galactic Dynamics: For initial data $(p_1(0), p_2(0), p_3(0), q_1(0), q_2(0), q_3(0)) = \left(0, \frac{711}{421}, \frac{1}{5}, \frac{5}{2}, 0, 0\right)$, we compute the solution at time $t = 10$ using RKM-6117 with a time-step of $h = 0.005$:

$$\begin{aligned} p_1(10) &= -0.962784812534641, & q_1(10) &= -2.861219736261032, \\ p_2(10) &= -0.528445761422120, & q_2(10) &= \ \ 0.425411261094871, \\ p_3(10) &= -0.003013803762495, & q_3(10) &= \ \ 0.254721799516517. \end{aligned}$$

The results of this experiment are given in Figure 7-9. Here we see a strong similarity to the results of Figure 7-8, with the exception of GLM-4124GS which is now less efficient than RK-4113S (Lobatto IIIB).

All results considered, it appears that the dimension of a problem is an important factor on a method's efficiency. We expect that, for larger problems, GLM-4123S will hold an advantage over RK-4212GS, though the size of such problems has yet to be determined.

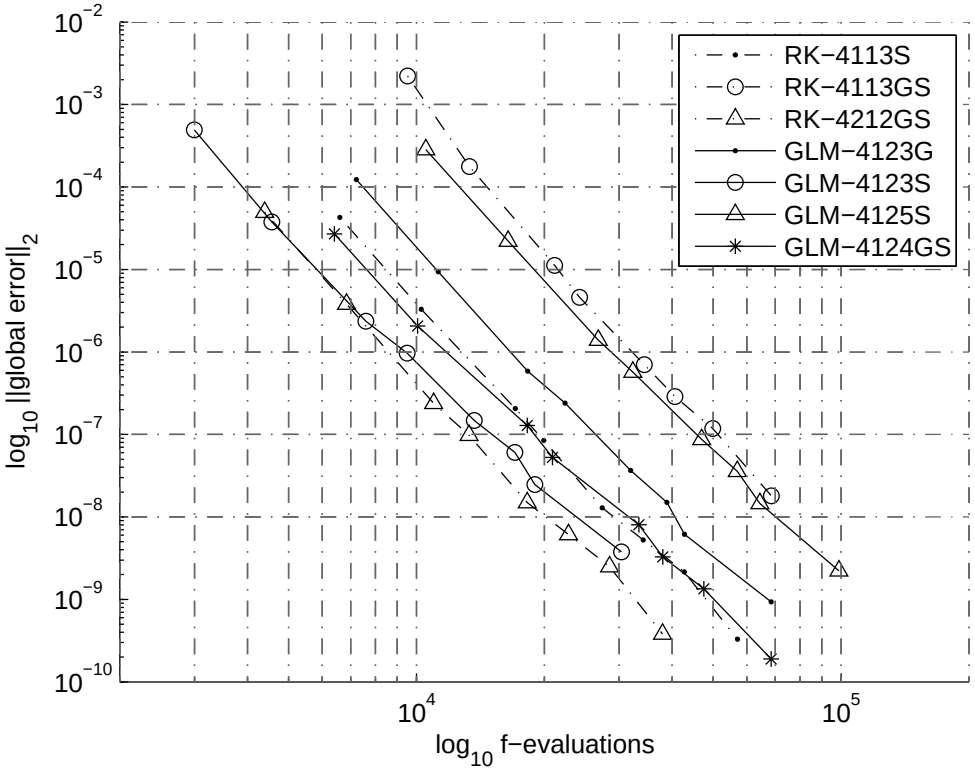


Figure 7-8: Efficiency plots of 4th order methods applied to **(HH)**.

.

7.6 Long-time integration

In the following experiments, we investigate the long-time preservation properties of our GLMs on various geometric problems. In light of the earlier parasitism results, we will only consider GLM-4123S and GLM-4125S for these simulations as these were found to possess the best parasitism-free behaviour. For comparison, we have chosen RK-4212GS (Gauss) and RK-4113S (Lobatto IIIB).

Hénon-Heiles: For initial data $(p_1(0), p_2(0), q_1(0), q_2(0)) = (\frac{561}{1346}, \frac{1}{5}, 0, \frac{3}{10})$ and an integration interval of $[0, 2.5 \times 10^6]$, we have a chaotic solution. In Figure 7-10, plots for the Hamiltonian error, using a time-step of $h = 0.25$, are given for each method. In each case, the Hamiltonian is well-preserved over the interval.

Kepler: Here, we use same initial data as given in Section 7.4, and consider an integration interval of $[0, 10^5]$. For each method, a small time-step of $h = 0.01$ is applied such that the moderately large time-derivatives, arising from the close approach, are

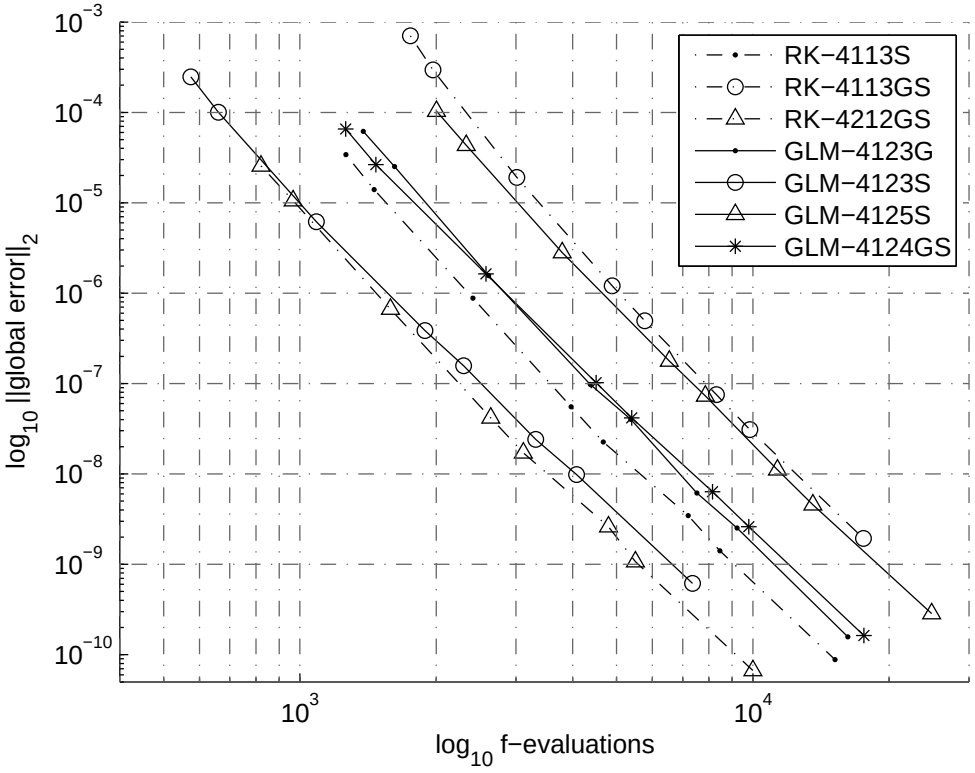


Figure 7-9: Efficiency plots of 4th order methods applied to **(GD)**.

.

adequately resolved. The results in Figure 7-11 generally demonstrate good preservation of the Hamiltonian across all methods. The exception is GLM-4123S where it appears that either parasitism or a significant accumulation of rounding error has resulted in an observable loss of preservation towards the end of the integration.

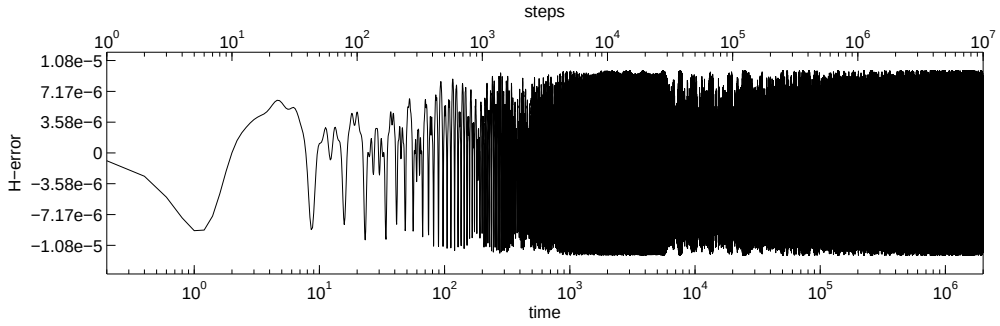
Similar conclusions can be made in Figure 7-12 where the preservation of angular momentum is considered. We note that since RK-4212GS is a symplectic method, it should exactly preserve quadratic invariants (in exact arithmetic). However, rounding error inevitably dominates these computations over long-times. Hence, the results appear to demonstrate a lack of preservation.

Double pendulum: For initial data $(p_1(0), p_2(0), q_1(0), q_2(0)) = (0, 0, 3.14, -3.1)$ and an integration interval of $[0, 10^5]$, we have a chaotic solution. As with the Kepler problem, we consider a small time-step of $h = 0.01$ to adequately resolve the large time-derivatives. The Hamiltonian preservation results of this experiment are given in Figure 7-13. Comparing the plots of RK-4113S and RK-4212GS, we deduce that the eventual drift in the Hamiltonian when using methods GLM-4123S, GLM-4125S and

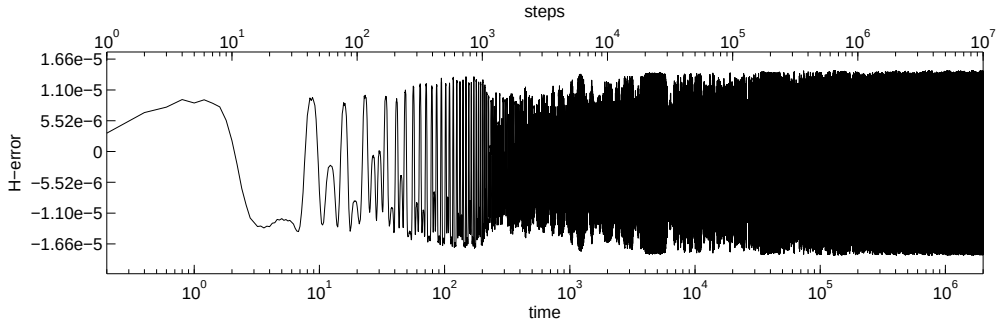
RK-4113S is caused by the lack of symplecticity of the method rather than parasitism. That being said, integration with GLM-4123S was terminated early as this drift grew sufficiently large to trigger parasitism.

Galactic Dynamics: Here, we consider the same initial data as given Section 7.5 and an integration interval of $[0, 10^6]$. Each method uses a time-step of $h = 0.1$. The plots given in Figure 7-14 demonstrate that the Hamiltonian is well-preserved for all methods.

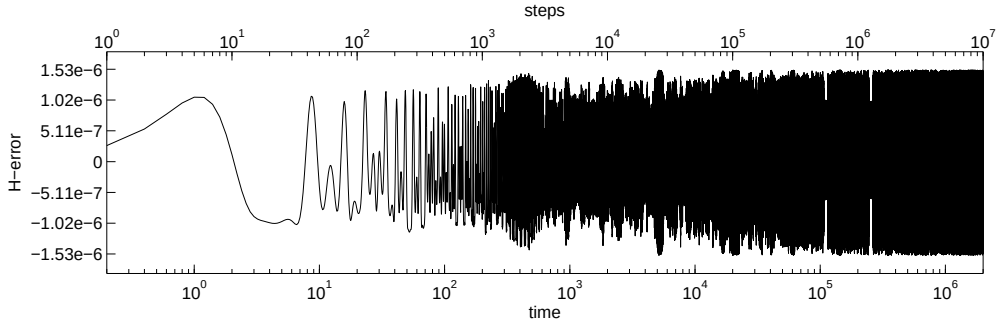
Rigid Body: Here, we take initial data $(y_1(0), y_2(0), y_3(0)) = (\cos(1.1), 0, \sin(1.1))$, an integration interval of $[0, 2 \times 10^6]$, and apply each method using a fixed time-step of $h = 0.2$. The results detailing the preservation of the quadratics invariants Q_1 and Q_2 are respectively given in Figures 7-15 and 7-16. Both GLM-4123S and RK-4113S demonstrate good preservation of these invariants over the interval. As with the angular momentum results in the Kepler computations, RK-4212GS should exactly preserve these invariants. However, we again find that a significant accumulation of rounding error gives the perception of a lack of preservation. The results for GLM-4125S show a slight loss preservation towards the end of the integration. It is possible that this is attributed to either parasitism or rounding error. However, given the limitations of our computational resources, we are unable to give to a definitive conclusion as to which one.



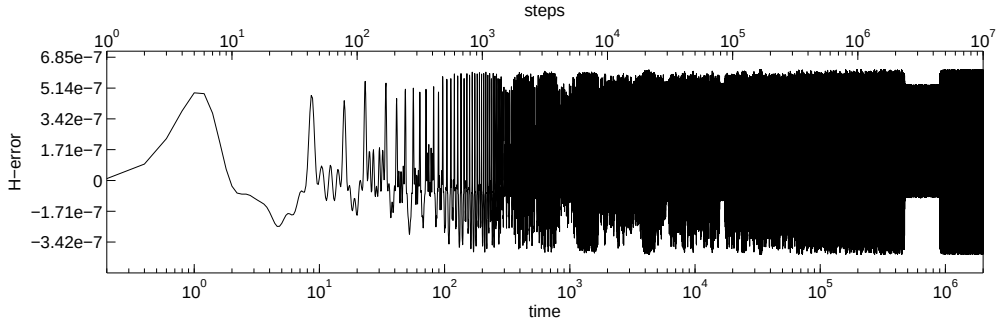
(a) GLM-4123S



(b) GLM-4125S

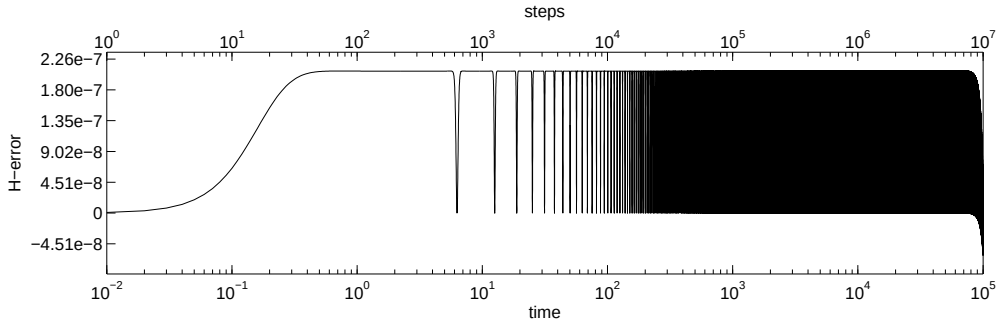


(c) RK-4113S

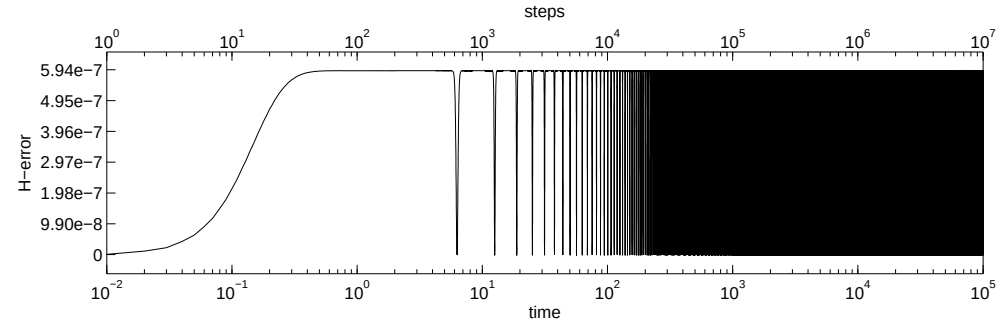


(d) RK-4212GS

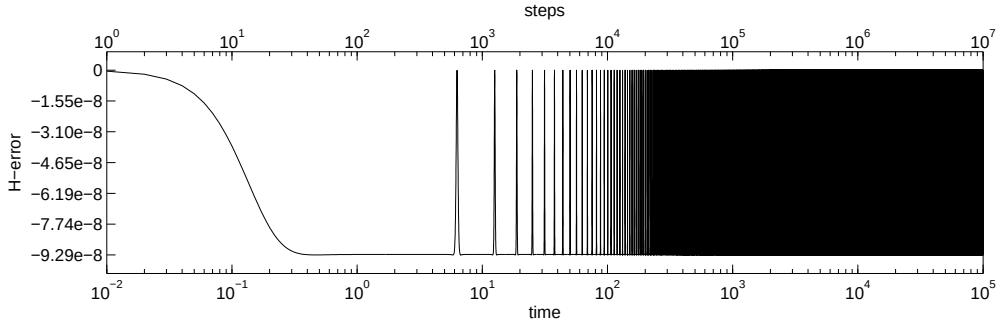
Figure 7-10: Hamiltonian preservation for **(HH)**. Each method is applied using a time-step of $h = 0.25$ over the interval $[0, 2.5 \times 10^6]$.



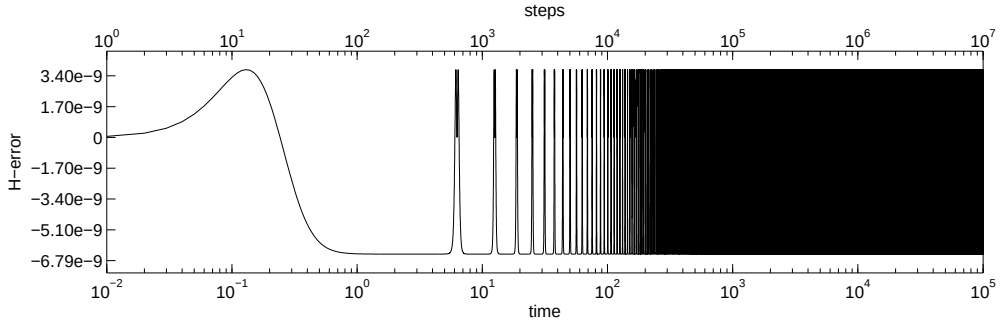
(a) GLM-4123S



(b) GLM-4125S

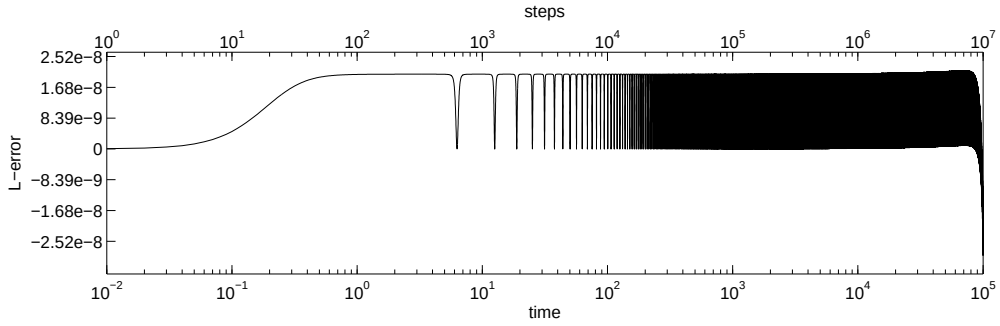


(c) RK-4113S

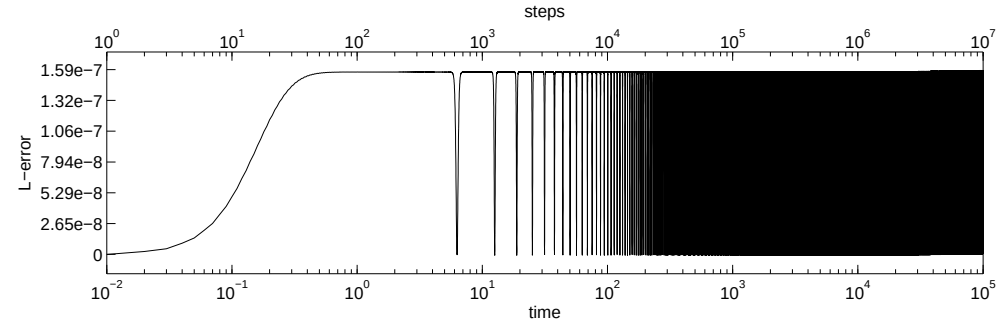


(d) RK-4212GS

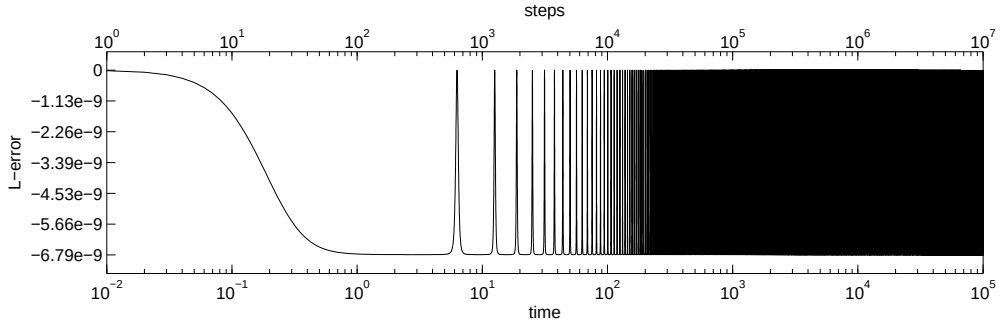
Figure 7-11: Hamiltonian preservation for **(KPL)**. Each method is applied using a time-step of $h = 0.01$ over the interval $[0, 10^5]$.



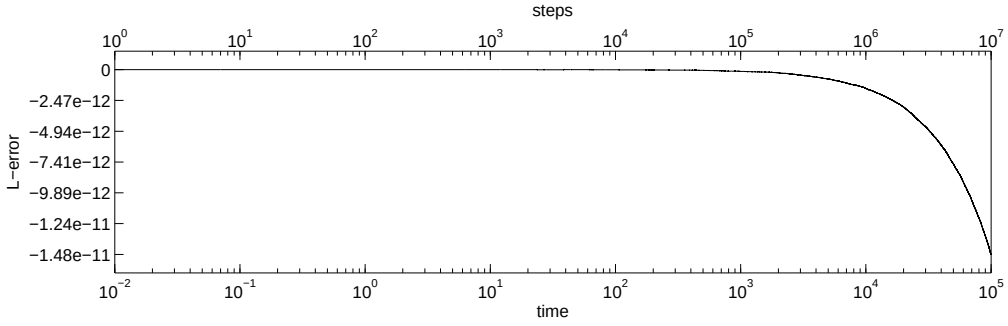
(a) GLM-4123S



(b) GLM-4125S

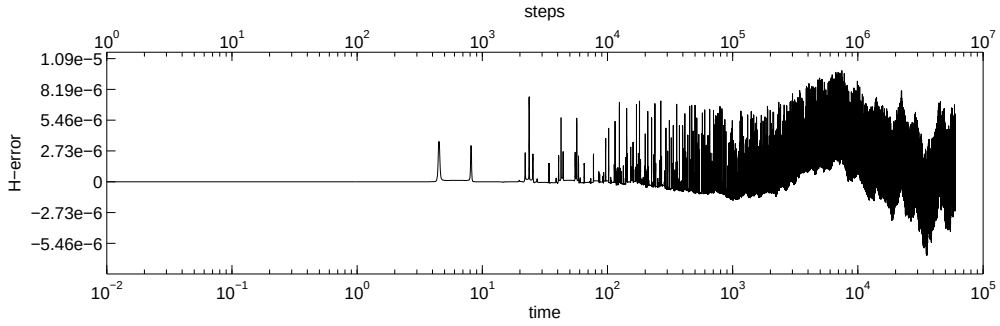


(c) RK-4113S

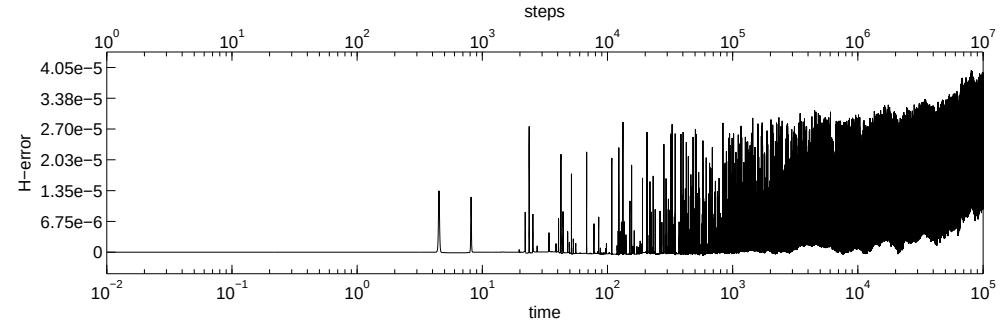


(d) RK-4212GS

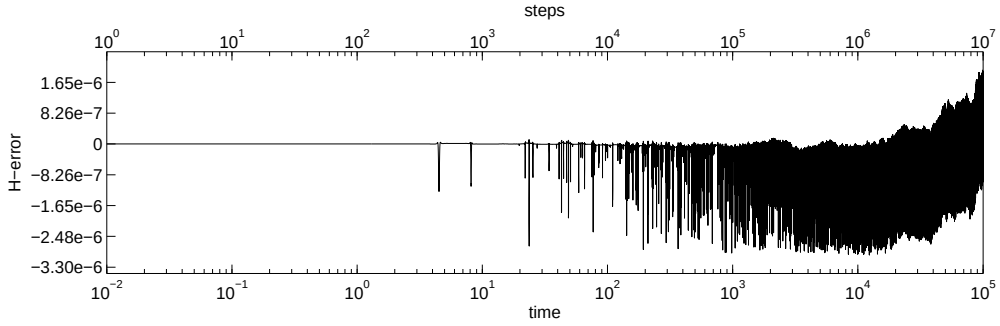
Figure 7-12: Angular momentum preservation for **(KPL)**. Each method is applied using a time-step of $h = 0.01$ over the interval $[0, 10^5]$.



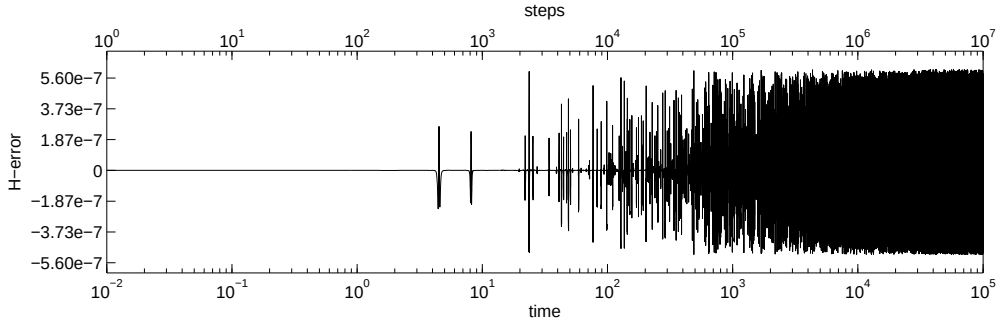
(a) GLM-4123S



(b) GLM-4125S

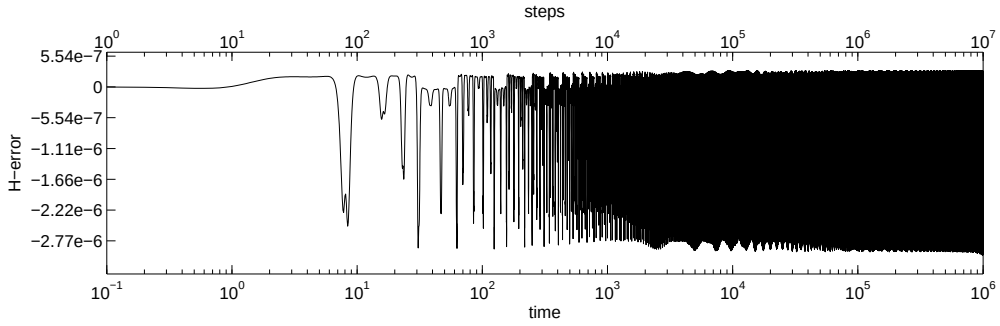


(c) RK-4113S

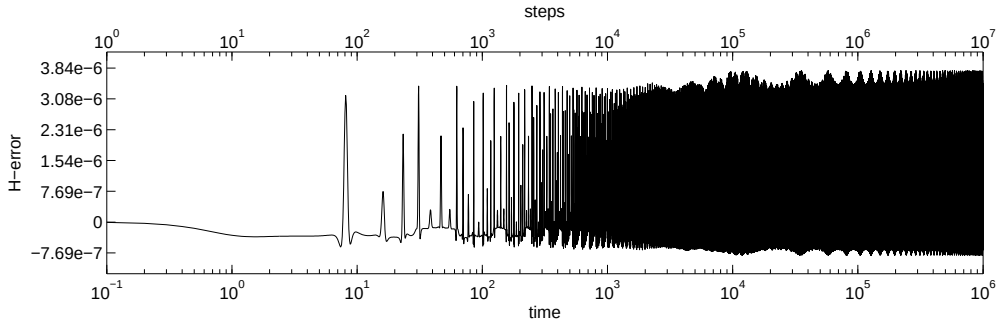


(d) RK-4212GS

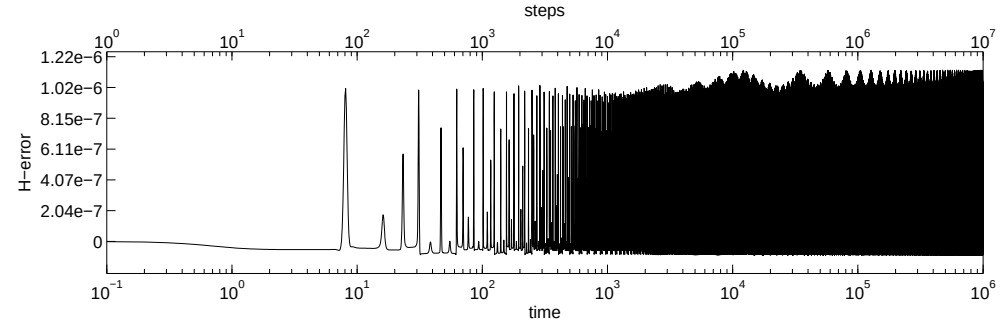
Figure 7-13: Hamiltonian preservation for **(DP)**. Each method is applied using a time-step of $h = 0.01$ over the interval $[0, 10^5]$.



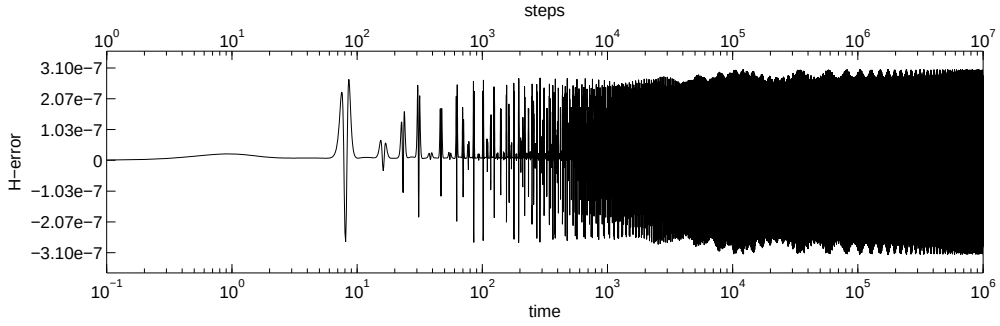
(a) GLM-4123S



(b) GLM-4125S

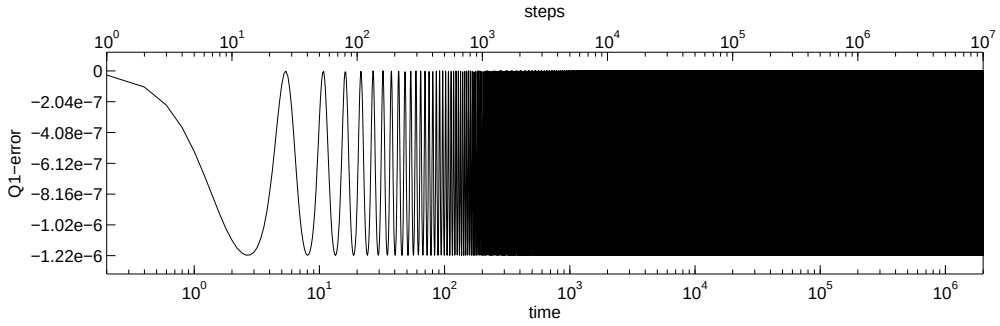


(c) RK-4113S

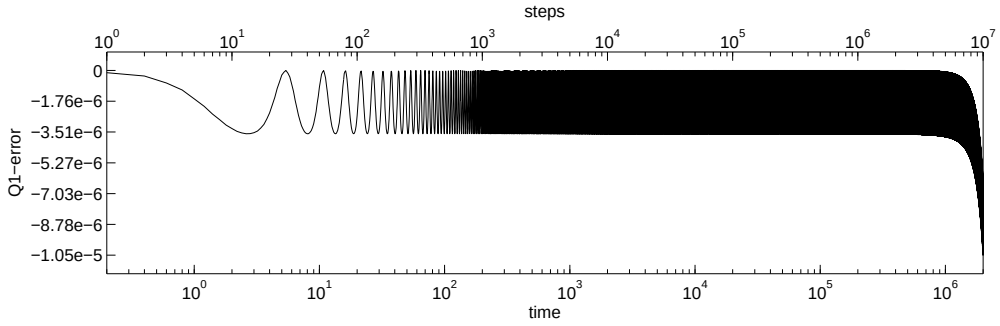


(d) RK-4212GS

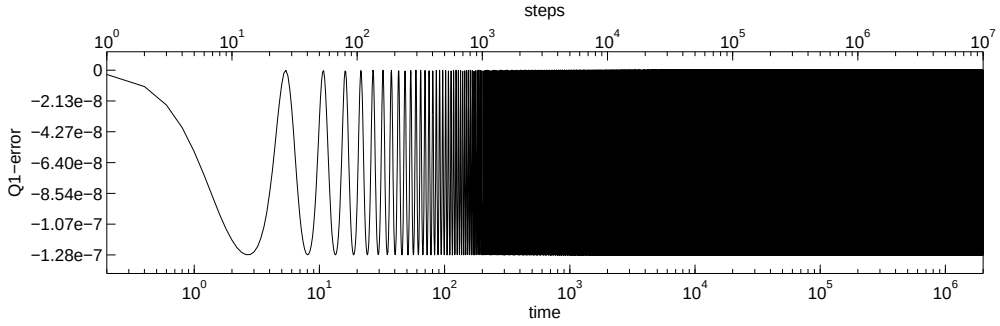
Figure 7-14: Hamiltonian preservation for **(GD)**. Each method is applied using a time-step of $h = 0.1$ over the interval $[0, 10^6]$.



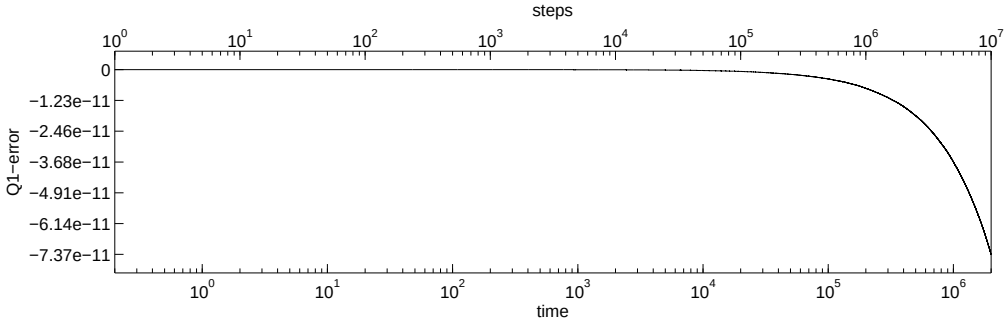
(a) GLM-4123S



(b) GLM-4125S

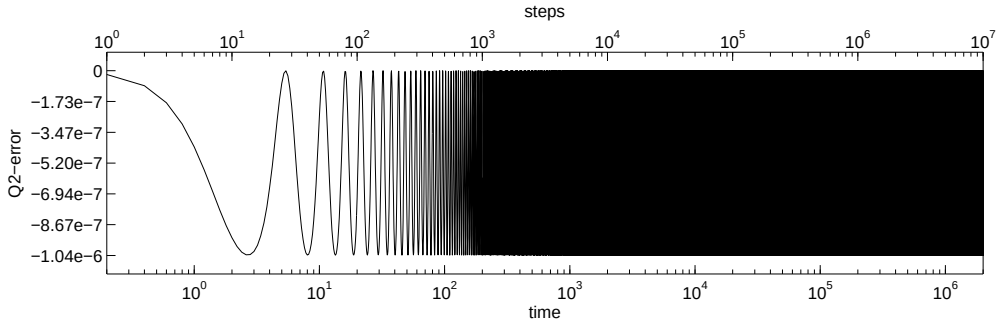


(c) RK-4113S

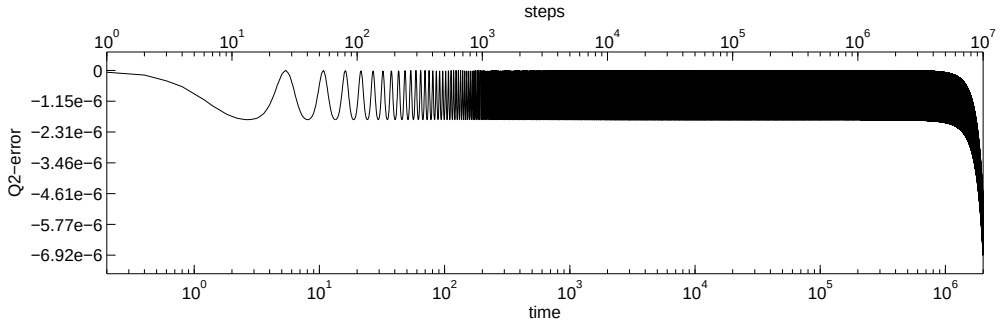


(d) RK-4212GS

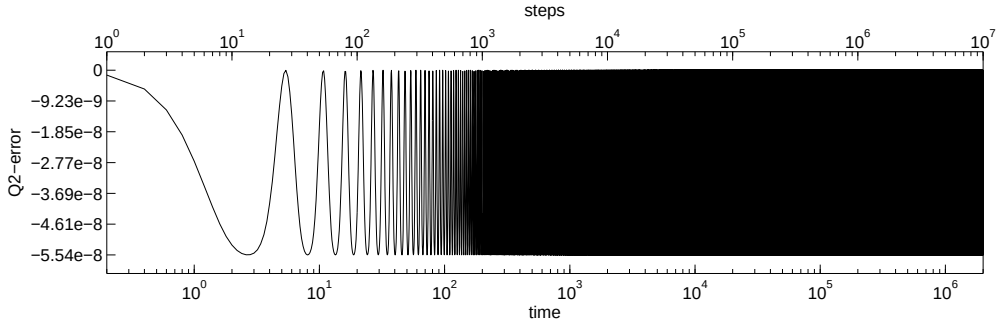
Figure 7-15: Q_1-preservation for **(RB)**. Each method is applied using a time-step of $h = 0.2$ over the interval $[0, 2 \times 10^6]$.



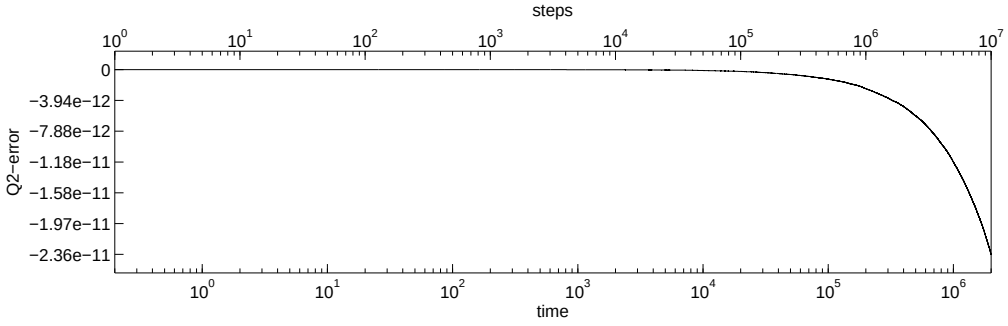
(a) GLM-4123S



(b) GLM-4125S



(c) RK-4113S



(d) RK-4212GS

Figure 7-16: Q_2-preservation for **(RB)**. Each method is applied using a time-step of $h = 0.2$ over the interval $[0, 2 \times 10^6]$.

Chapter 8

Conclusions

This thesis has considered the construction and analysis of structure-preserving general linear methods (GLMs). Main contributions include:

1. The development of a **theoretical toolkit** for analysing derivatives of B-series. Here, we have defined a set of derivative trees which acts as an analogue to the set of rooted trees in B-series analysis. These have helped to construct algebraic formulae for the various operations that are performed on derivative B-series. The toolkit has been applied to develop an ***a priori* theory of parasitism** for GLMs. Here, we have extended the idea of the underlying one-step method (UOSM) to derivative UOSMs. The research outcomes of this work include a bound on the parasitic components of the method that demonstrates the potential applicability of these methods for long-time integrations. Furthermore, we have derived higher-order parasitism-free conditions guaranteeing that the parasitic components remain bounded over longer intervals. In addition, we give the conditions for 3rd order parasitism-free behaviour and show that this increases to 4th order with symmetric, 2-input methods. This has led to the construction of new symmetric methods that are 4th order parasitism-free. A variety of numerical experiments have been performed at the end of the thesis that show agreement with the theory.

2. The development of a **computational toolkit** for assisting the analysis of GLMs. Here, we have taken an object-oriented approach to programming and explained how to represent rooted trees, derivative trees, B-series, derivative B-series, and GLMs as objects. In addition, we have implemented numerous operations that are typically performed on these objects, e.g. composition, inversion, product, addition, subtraction, to name but a few. Several important applications have

been described that demonstrate the usefulness of this toolkit. In particular, we have shown how to compute the order of a GLM with respect to a given starting method. We have also shown how to derive the UOSM and the ideal starting method of a given GLM. Finally, we have shown how to perform a parasitism analysis on the methods by deriving the (parasitic) derivative UOSMs of a given method.

3. A **theory of composition** for GLMs. Here, we have considered two approaches: The first concerns GLMs that take Nordsieck inputs. The main advantage here is that only a scaling of the inputs is required between compositions of the methods. However, the drawback is that this approach cannot be applied repeatedly to obtain methods of arbitrarily high order. The second approach concerns a generalisation of the composition formulae used for one-step methods to GLMs. Here, if the method is symmetric, then it is possible to obtain methods of arbitrarily high order. Several numerical experiments have been performed that computationally confirm this result. In particular, we have constructed methods of order 6 and 8, and computationally confirmed that the theoretical order increase has been attained.

4. A **decomposition result** on structure-preserving GLMs. Here, we have shown that many structure-preserving GLMs permit a decomposition into single-stage GLMs. An important consequence of this is that methods can be implemented in a memory-efficient manner. We have also explored the connection between single-stage GLMs and linear multistep methods (LMMs) which has revealed that certain stucture-preserving GLMs can be expressed as a composition of (possibly symmetric) LMMs.

8.1 Considerations for future work

Below, we list possible directions for future work:

- The full breadth of the parasitism theory has yet to be explored:
  - By considering alternative derivative finishing methods, i.e. different from being strictly a left-eigenvector of V, it may be possible to achieve a reduction in the total number of 3rd order parasitism-free conditions. If this is true, more efficient methods could be designed, i.e. with fewer/explicit stages.

  - Being able to express the parasitic contributions as formal derivative B-series motivates the investigation of filters for effective parasitism removal. A simple idea would be to approximate the leading terms of these derivative B-series by finite differences and then to subtract this contribution from the numerical solution.

- The computational toolkit can be further developed to include other pieces of analysis. In particular, the auto-detection of the (L, P) matrices of symmetric methods or the (D, G) matrices of G-symplectic methods would be useful. The implementation of the tools used in backward error analysis would also be invaluable to the general analysis of these methods.

- Some measure of the efficiency of composition methods should be performed to assess their impact in practical applications. This could be in the form of the efficiency experiments performed on the GLMs at the end of this thesis.

- Other consequences of the decomposition theory into LMMs could include an alternative approach to the construction of high-order GLMs, i.e. by combining this theory with the ideas used in the cyclic composition of LMMs. This would provide a third approach to the composition theory we have developed.

Bibliography

[1] V. I. Arnold. *Mathematical Methods of Classical Mechanics*, volume 60. Springer Science & Business Media, 1989.

[2] A. Aubry and P. Chartier. Pseudo-symplectic Runge-Kutta methods. *BIT*, 38(3):439–461, 1998.

[3] K. Burrage and J. C. Butcher. Stability criteria for implicit Runge-Kutta methods. *SIAM J. Numer. Anal.*, 16(1):46–57, 1979.

[4] J. C. Butcher. On Runge-Kutta processes of high order. *J. Aust. Math. Soc.*, 4(02):179–194, 1964.

[5] J. C. Butcher. A Modified Multistep Method for the Numerical Integration of Ordinary Differential Equations. *J. ACM*, 12(1):124–135, 1965.

[6] J. C. Butcher. On the Convergence of Numerical Solutions to Ordinary Differential Equations. *Math. of Comp.*, 20(93):1–10, 1966.

[7] J. C. Butcher. An introduction to DIMSIMs. *Comput. Appl. Math.*, 14:59–72, 1995.

[8] J. C. Butcher. An introduction to Almost Runge-Kutta methods. *Appl. Numer. Math.*, 24(2):331–342, 1997.

[9] J. C. Butcher. General linear methods. *Acta Numer.*, 15:157–256, 2006.

[10] J. C. Butcher. *Numerical Methods for Ordinary Differential Equations*. Wiley, 2nd edition, 2008.

[11] J. C. Butcher. The cohesiveness of G-symplectic methods. *Numer. Algorithms*, pages 1–18, 2015.

[12] J. C. Butcher, P. Chartier, and Z. Jackiewicz. Nordsieck representation of DIMSIMs. *Numer. Algorithms*, 16(2):209–230, 1997.

[13] J. C. Butcher, Y. Habib, A. T. Hill, and T. J. T. Norton. The control of parasitism in G-symplectic methods. *SIAM J. on Numer. Anal.*, 52(5):2440–2465, 2014.

[14] J. C. Butcher and L. L. Hewitt. The existence of symplectic general linear methods. *Numer. Algorithms*, 51(1):77–84, 2009.

[15] J. C. Butcher and A. T. Hill. Linear multistep methods as irreducible general linear methods. *BIT*, 46(1):5–19, 2006.

[16] J. C. Butcher, A. T. Hill, and T. J. T. Norton. Symmetric general linear methods. Submitted to BIT.

[17] J. C. Butcher and G. Imran. Order conditions for G-symplectic methods. *BIT*, pages 1–22, 2015.

[18] G. D. Byrne and R. J. Lambert. Pseudo-Runge-Kutta methods involving two points. *J. ACM*, 13(1):114–123, 1966.

[19] R. P. Chan and A. Gorgey. Active and passive symmetrization of Runge–Kutta Gauss methods. *Appl. Numer. Math.*, 67:64–77, 2013.

[20] M. Creutz and A. Gocksch. Higher-order hybrid Monte Carlo algorithms. *Phys. Rev. Lett.*, 63(1):9, 1989.

[21] G. Dahlquist. Fehlerabschätzungen bei Differenzenmethoden zur numerischen Integration gewöhnlicher Differentialgleichungen. *Z. Angew Math. Mech.*, 31(8-9):239–240, 1951.

[22] G. Dahlquist. Error analysis for a class of methods for stiff non-linear initial value problems. In *Numerical analysis*, pages 60–72. Springer, 1976.

[23] R. D'Ambrosio and E. Hairer. Long-term stability of multi-value methods for ordinary differential equations. *J. Sci. Comput.*, 60(3):627–640, 2014.

[24] R. D'Ambrosio, E. Hairer, and C. J. Zbinden. G-symplecticity implies conjugate-symplecticity of the underlying one-step method. *BIT*, 2013.

[25] J. Donelson and E. Hansen. Cyclic composite multistep predictor-corrector methods. *SIAM J. Numer. Anal.*, 8(1):137–157, 1971.

[26] T. Eirola and J. M. Sanz-Serna. Conservation of integrals and symplectic structure in the integration of differential equations by multistep methods. *Numer. Math.*, 61:281–290, 1992.

[27] E. Forest. Canonical integrators as tracking codes. In *Physics of Particle Accelerators*, volume 184, pages 1106–1136. AIP Conference Proceedings, 1989.

[28] C. W. Gear. Hybrid methods for initial value problems in ordinary differential equations. *SIAM J. Num. Anal. ser. B*, 2(1):69–86, 1965.

[29] W. Gentzsch and A. Schlüter. Über ein Einschrittverfahren mit zyklischer Schrittweitenänderung zur Lösung parabolischer Differentialgleichungen. *Z. Angew. Math. Mech.*, 58:T415–T416, 1978.

[30] W. B. Gragg. On extrapolation algorithms for ordinary initial value problems. *SIAM J. Num. Anal. ser. B*, 2:384–403, 1965.

[31] W. B. Gragg and H. J. Stetter. Generalized multistep predictor-corrector methods. *J. ACM*, 11(2):188–209, 1964.

[32] E. Hairer. Backward error analysis for multistep methods. *Numer. Math.*, 84(2):199–232, 1999.

[33] E. Hairer. Symmetric linear multistep methods. *BIT*, 46:515–524, 2006.

[34] E. Hairer. Conjugate-symplecticity of linear multistep methods. *J. Comput. Math.*, 26(5):657–659, 2008.

[35] E. Hairer and C. Lubich. Symmetric multistep methods over long times. *Numer. Math.*, 97:699–723, 2004.

[36] E. Hairer, C. Lubich, and G. Wanner. *Geometric Numerical Integration. Structure-Preserving Algorithms for Ordinary Differential Equations*. Springer, 2nd edition, 2006.

[37] E. Hairer, S. P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems*. Springer, 1st edition, 1987.

[38] E. Hairer, S. P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems*. Springer, 2nd edition, 1991.

[39] A. T. Hill. Unpublished notes.

[40] A. Iserles. Solving linear ordinary differential equations by exponentials of iterated commutators. *Numer. Math.*, 45(2):183–199, 1984.

[41] D. I. Ketcheson. NodePy. Version 0.6. http://github.com/ketch/nodepy/, 2015.

[42] U. Kirchgraber. Multi-step methods are essentially one-step methods. *Numer. Math.*, 48(1):85–90, 1986.

[43] J. D. Lambert and I. A. Watson. Symmetric multistep methods for periodic initial value problems. *IMA J. Appl. Math.*, 18(2):189–202, 1976.

[44] F. M. Lasagni. Canonical Runge-Kutta methods. *Z. Angew. Math. Phys.*, 39(6):952–953, 1988.

[45] R. I. McLachlan. On the numerical integration of ordinary differential equations by symmetric composition methods. *SIAM J. Sci. Comput.*, 16(1):151–168, 1995.

[46] T. J. T. Norton and A. T. Hill. An iterative starting method to control parasitism for the Leapfrog method. *Appl. Numer. Math.*, 87:145–156, 2015.

[47] H. Poincaré. Les méthodes nouvelles de la méchanique céleste. *Paris: Gauthier-Villars, 1892, 1893,— c1899*, 1, 1899.

[48] G. D. Quinlan and S. Tremaine. Symmetric multistep methods for the numerical integration of planetary orbits. *The Astronomical Journal*, 100:1694–1700, 1990.

[49] N. Rattenbury. *Almost Runge-Kutta methods for stiff and non-stiff problems*. PhD thesis, ResearchSpace@ Auckland, 2005.

[50] J. M. Sanz-Serna. Runge-Kutta schemes for Hamiltonian systems. *BIT*, 28(4):877–883, 1988.

[51] J. M. Sanz-Serna. Symplectic integrators for Hamiltonian problems: an overview. *Acta Numer.*, 1:243–286, 1992.

[52] J. M. Sanz-Serna. Symplectic Runge-Kutta and related methods: recent results. *Phys. D*, 60(1):293–302, 1992.

[53] J. M. Sanz-Serna and L. Abia. Order conditions for canonical Runge-Kutta schemes. *SIAM J. Numer. Anal.*, 28(4):1081–1096, 1991.

[54] R. D. Skeel. Equivalent forms of multistep formulas. *Math. Comp.*, 33(148):1229–1250, 1979.

[55] A. Spence and I. G. Graham. Numerical methods for bifurcation problems. In *The Graduate Students Guide to Numerical Analysis 98*, pages 177–216. Springer, 1999.

[56] D. Stoffer. On reversible and canonical integration methods. *SAM-Report*, 88(05), 1988.

[57] D. Stoffer. General linear methods: connection to one step methods and invariant curves. *Numer. Math.*, 64(1):395–408, 1993.

[58] M. Suzuki. Fractal decomposition of exponential operators with applications to many-body theories and Monte Carlo simulations. *Phys. Lett. A*, 146(6):319–323, 1990.

[59] M. Suzuki. Fractal path integrals with applications to quantum many-body systems. *Phys. A*, 191(1):501–515, 1992.

[60] Y.-F. Tang. The symplecticity of multi-step methods. *Comput. Math. with Appl.*, 25(3):83–90, 1993.

[61] P. D. Williams. A Proposed Modification to the Robert-Asselin Time Filter. *Mon. Wea. Rev.*, 137(8):2538–2546, 2009.

[62] Wolfram Research, Inc. *Mathematica*. Version 10.2. Champaign, Illinois, 2015.

[63] H. Yoshida. Construction of higher order symplectic integrators. *Phys. Lett. A*, 150(5):262–268, 1990.