Håvard Berland

Lie group and exponential integrators: Theory, implementation, and applications

Doctoral thesis
for the degree of philosophiae doctor

Trondheim, July 2006

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

NTNU
Norwegian University of Science and Technology

Doctoral thesis
for the degree of philosophiae doctor

Faculty of Information Technology, Mathematics and Electrical Engineering
Department of Mathematical Sciences

ISBN 82-471-7969-5 (printed version)
ISBN 82-471-7968-7 (electronic version)
ISSN 1503-8181

Doctoral theses at NTNU, 2006:106

Printed by NTNU-trykk

Summary

This PhD-thesis contains an introduction and six research papers sorted chronologically, of which the first four are accepted for publication. The introduction aims at giving a very brief summary of the background theory needed for the following papers. Also, some motivation of the issues addressed by the papers is given. Paper I discusses algebraic structures of ordered rooted trees and their applications to Lie group integrators. Results from Hopf algebra theory on elementary differentials for Lie group integrators are used, and applications to order analysis and backward error analysis are given. Paper II, III, IV, and V are primarily on exponential integrators, a class of numerical schemes tailored the solution of stiff systems of systems of ordinary differential equations. Paper II discusses classical order analysis and gives some theoretical results on the form of the integrators, applicable for the construction of high order exponential integrators. Paper III is on an implementation of exponential integrators on computers, and source code, available electronically, accompanies the paper. Paper IV includes an analytical and numerical study of the performance of two classes of exponential integrators on the nonlinear Schrödinger equation. Paper V is a numerical study of behaviour over long integration invervals on the nonlinear Schrödinger equation, using nonlinear spectral theory for determining validity of the numerical solution and thereby jugdging the numerical integrators. At last, in Paper VI, properties of a class of exponential like functions, essential in exponential integrators, are derived, using an approach based on Lie group theory.

Acknowledgements

This thesis is the result of my work for the degree of Philosophiae Doctor at Department of Mathematical Sciences, Norwegian University of Science and Technology (NTNU).

First of all, I would like to thank my supervisor Professor Brynjulf Owren for continuous support and guidance. My PhD-scholarship and travel grants have been kindly provided by the Norwegian University of Science and Technology.

Spring 2003 was spent at Centre for Advanced Study in Oslo during "Special year in geometric integration", where I got the opportunity to meet numerous people within the field of numerical analysis, and received additional funding from the Norwegian Academy of Science and Letters. The numerical analysis groups in Trondheim and Bergen have organized yearly MaGiC meetings at picturesque mountain locations in Norway, and I appreciate the acquaintance with Elena Celledoni, Stein Krogstad, Borislav Minchev, Per Christian Moan, Hans Z. Munthe-Kaas, Jitse Niesen, Niklas Säfström, and Antonella Zanna, many of whom I have also met at various other conferences and meetings. Bård Skaflestad and Will Wright get special attention being very pleasant to coauthor with. I would also like to thank Department of Mathematical Sciences, NTNU, for providing a state of the art computer system, maintained by Per Kristian Hove. I thank friends in Trondheim for making lunchtime and life pleasant, Hugo Hammer, Steinar Kragset, and Kjetil Midthun.

Special attention also goes to Professor Constance Schober and Dr Alvaro Islas at University of Central Florida, Orlando, USA, for their warm hospitality and kindliness during my stay at University of Central Florida October 2005 to February 2006. I am thankful for their willingness to coauthor Paper V.

Finally, I am for ever grateful to my Marte.

Håvard Berland

Trondheim, July 2006

Contents

Chapter 1

Introduction

1.1 Ordinary differential equations and numerical approximations

Ordinary differential equations (ODEs) can be written as

$$\frac{\mathrm{d}y}{\mathrm{d}t} = f(y,t), \qquad y(0) = y_0 \tag{1.1}$$

and describe how a certain quantity y (let it be a scalar or a vector) evolves, for instance in time, given its initial state y_0 and a description, the function f, of the evolution given the current state y and t. ODEs are often used to describe physical systems, and depending on the complexity of the mathematical model, an explicit expression for the solution $y(t)$ may or may not be feasible. For this reason, numerical methods which can provide a numerical approximation to the true solution $y(t)$ are an invaluable tool in several situations.

The numerical solution of ODEs has been studied since Euler in 1768 who proposed a simple way to solve it, the forward Euler scheme,

$$y_{n+1} = y_n + hf(y_n, t_n) \tag{1.2}$$

in which y_n is an approximation to the exact solution $y(t)$ at time t_n and h is the time step $h = t_{n+1} - t_n$. The scheme produces y_{n+1} which is an approximation of $y(t_{n+1})$. Simplicity almost always has some inherent limitations, and circumventing the limitations motivates further research in numerical analysis. The numerical solution of ODEs is usually obtained by a Runge–Kutta scheme or a multistep scheme, both being generalizations of the forward Euler scheme and both are special cases of general linear methods. An extensive reference for numerical integration of ordinary differential equations is the monograph by Hairer, Wanner, and Nørsett [24].

This thesis focuses on two aspects. First, there could be properties or constraints known to the solution y known a priori, typically originating from the mathematical model used to obtain (1.1), that should or could be enforced in the numerical solution. One approach for enforcing a priori information in the

solution is to employ Lie group integrators. This methodology is applicable in situations where one is able to formulate the equation on a manifold representing the constraints or information in a natural way. This is further introduced and exemplified in Section 1.2.

The second aspect is to be able to choose a suitable time step h. In general, the larger the time step h, the larger the numerically committed error will be. However, the forward Euler scheme, among many, will in certain situations exhibit an upper limit on h. For larger h, the scheme will be unstable and the numerical solution will diverge. Exponential integrators, Section 1.4, is one class of numerical methods for ODEs which aim to avoid such restrictions on h. This is the subject of stability of integrators related to "*stiff*" problems.

This introduction aims at giving a brief summary and glimpses of the background theory providing the necessary basics for the subsequent papers. For a rigorous and unabridged presentation, the reader is referred to monographs and papers cited.

1.2 Lie group integrators

In the forward Euler integrator (1.2) the phase space value y_n is updated by adding a component $hf(y_n)$, where the addition is performed in the vector space in which the solution $y(t)$ lives, $\mathbf{R}^n$. The "basic movements" of the integrator in the ambient space is provided by translations, using the familiar "+" in (the trivial manifold) $\mathbf{R}^n$.

Lie group integrators generalize this by making it possible to use other types of basic movements, which replace vector space additions. The differential equation is now formulated on the tangent space of a general manifold M,

$$\frac{\mathrm{d}y}{\mathrm{d}t} = F(y), \qquad y(t) \in M, \quad F(y(t)) \in T_{y(t)}M. \tag{1.3}$$

For our purpose, it is sufficient to know that a manifold with dimension n is a mathematical object which "looks like" $\mathbf{R}^n$ in a neighborhood around every point, but is globally fundamentally different. An ant placed on the surface of a balloon may experience the surface as a plane, not a sphere. The tangent space at a point describes the space of *directions* it is possible to move in from the point within the manifold.

The two-dimensional sphere $S^2 \subset \mathbf{R}^3$ provides a good example of a manifold suitable for Lie group integrators. The tangent space at every point on the sphere is a copy of the vector space $\mathbf{R}^2$, as the directions in which one is able to move, being confined on a sphere's surface, is spanned by a plane.

In general, a *homogeneous space* is a natural environment in which a Lie group integrator is applied.

Definition 1.1. A homogeneous space is denoted by the triple (M, G, Λ) where

- M is any manifold
- G is a Lie group
- Λ is a transitive Lie group action which for elements $m \in M$ and $g \in G$ will produce another element $\Lambda(g, m) = m^* \in M$.

The group action Λ is the generalization of the "+" in classical integrators. It represents the fundamental building block of any movement on the manifold, replacing addition in $\mathbf{R}^n$ as used in classical integrators. Transitivity ensures that the group and the group action can reach all over the manifold. By trivial choices of M, G, and the action Λ, namely $\mathbf{R}^n$, $\mathbf{R}^n$, and "+", Lie group method reduces to classical integrators in the familiar sense.

The next step is to determine the updates which should be used for the group action, $hf(y_n)$ for the trivial Euler case. This comes down to transforming the vector field $F(y)$ on the tangent space of the manifold, to the Lie algebra associated to the Lie group G, and then computing the update (a group element $g \in G$). Early contributions to this field were given by Lewis and Simo in [35, 36]. The first systematic generalizations of Runge–Kutta methods to Lie groups are the Crouch–Grossman methods [19], where in each step the final value is computed by consecutive applications of the group action Λ. However, additional order conditions were needed, compared to the classical Runge–Kutta methods, as also shown in [46]. Another idea is due to Munthe-Kaas in [39], where the entire differential equation is transferred to the Lie algebra, and computations for the inner stages in the underlying Runge–Kutta scheme are performed entirely in the Lie algebra. Using this approach, no additional order conditions are needed and all existing Runge–Kutta schemes could be used directly, however the computations inside the Lie algebra may have additional issues, notably with stability and complexity of computing commutators in the algebra. The paper [40] further analyzed computations in a Lie algebra, important for subsequent analysis and schemes. A generalization of both Crouch–Grossman and Runge–Kutta–Munthe-Kaas methods is presented in [13], in which one is able to circumvent the necessity of commutators. A commutator-free scheme of order 4 originates from this work, and is used extensively together with the Affine Lie group in Paper V.

As an example of a simple Lie group integrator and its beneficial properties, we consider a comparison of Lie–Euler and forward Euler on the vector field in $\mathbf{R}^3$,

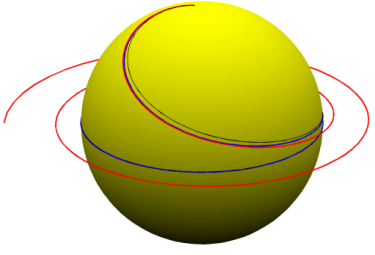
$$\dot{\boldsymbol{y}} = \begin{pmatrix} \dot{y}_1 \\ \dot{y}_2 \\ \dot{y}_3 \end{pmatrix} = \begin{pmatrix} -y_2 + y_1 y_3^2 \\ y_1 + y_2 y_3^2 \\ -y_3(y_1^2 + y_2^2) \end{pmatrix}. \tag{1.4}$$

One may confirm that $\frac{\mathrm{d}}{\mathrm{d}t}(y_1^2 + y_2^2 + y_3^2) = 0$, which means that the length of the vector $\boldsymbol{y} = (y_1, y_2, y_3)^T$ is time invariant in the exact solution. That means that the exact solution will stay on a sphere S^2 of radius $\|\boldsymbol{y}_0\|$ for all time. A Lie group integrator will need to formulate (1.4) on the homogeneous space of a sphere, the manifold, the matrix group SO(3) (3×3 matrices with determinant 1) and the group action multiplying SO(3) matrices with vectors in $\mathbf{R}^3$.

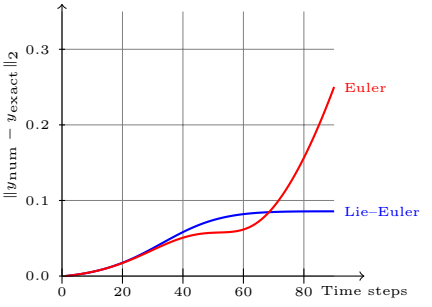
Lie–Euler for this problem is the update

$$\boldsymbol{y}_{n+1} = \operatorname{Exp}\left(h \begin{pmatrix} 0 & -(y_1^2 + y_2^2) & y_3(y_1 - y_2) \\ y_1^2 + y_2^2 & 0 & y_3(y_1 + y_2) \\ -y_3(y_1 - y_2) & -y_3(y_1 + y_2) & 0 \end{pmatrix}\right) \boldsymbol{y}_n \tag{1.5}$$

Supplying Lie–Euler and forward Euler with a step size h (chosen sufficiently large for visual impact), reveals the difference shown in Figure 1.1, where the forward Euler solution spirals outwards and away from the sphere, while Lie–Euler stays on the sphere to machine precision, but still deviates from the exact solution as according to being a first order integrator.



(a) Phase space solution. Thin almost indistinguishable black line is the exact solution, blue line is Lie–Euler, red line is forward Euler, both solvers with time step $h = 0.05$.



(b) The error committed by the numerical methods when compared to an exact solution computed using MATLABS `ode45`. Euler quickly blows up, while Lie–Euler stabilizes near the exact solution. The norm used is in $\mathbf{R}^3$.

Figure 1.1: Comparison of Lie–Euler and Euler on the $\mathbf{R}^3$ problem (1.4). The computational complexity of both schemes is roughly equivalent for this problem.

An extensive survey paper on Lie group methods is provided by [30]. Paper I discusses algebraic theory underlying order analysis for Lie group integrators, and Paper VI uses the theory from Lie group integrators in order to obtain further results for exponential integrators, Section 1.4.

1.3 Order analysis and trees

Order analysis for numerical integrators is a tool used to assess how fast a numerical integrator converges towards the exact solution as the time step h approaches zero. The analysis is initiated by repeated differentiation of the ODE (1.1) using the Leibniz and chain rule.

$$\dot{y} = f(y) \tag{1.6}$$

$$\ddot{y} = f'(y)\dot{y} \qquad = f'(y)f(y) \tag{1.7}$$

$$y^{(3)} = f''(y)\dot{y}^2 + f'(y)\ddot{y} \qquad = f''(y)f(y)^2 + (f'(y))^2 f(y) \tag{1.8}$$

The exact solution can be expressed in terms of a Taylor series in the step size h for $y(t_n + h)$, and this is compared term by term to a Taylor series in h for the numerical solution $y_{n+1} \approx y(t_n + h)$. The discrepancy then gives the order of the numerical scheme, for a scheme of (local) order p, we write

$$y_{n+1} = y(t_n + h) + \mathcal{O}(h^{p+1}). \tag{1.9}$$

assuming $y_n = y(t_n)$.

The Forward Euler scheme (1.2) is an order 1 scheme. Runge–Kutta schemes presented at the beginning of the 20th century solved precisely the conditions obtained from this analysis. An important scheme that is still much used is the fourth order scheme of Kutta, *"the Runge–Kutta scheme"*, with coefficients

$$\begin{array}{c|cccc} 0 & & & & \\ 1/2 & 1/2 & & & \\ 1/2 & & 1/2 & & \\ 1 & & & 1 & \\ \hline & 1/6 & 1/3 & 1/3 & 1/6 \end{array} \tag{1.10}$$

when written in a "Butcher-tableau". This very scheme is also used as a basis for many fourth order exponential integrators applied in Paper II, III, IV, and V.

Higher order methods require further differentiation as in (1.6)–(1.8). But the number of terms increase *exponentially*, and for high order, this strategy is cumbersome. The corresponding Taylor series for Runge–Kutta schemes have even more terms. Butcher realized in [10] that the trees Cayley used in 1857 [12] for representing elementary differentials could be used for order analysis of Runge–Kutta scheme. Subsequent analysis and further development were significantly simplified using the language of trees, in which all necessary operations and conditions can be described.

Assign the rooted one-node tree • to the right hand side of (1.6). Then define a differentiation process as "graft a node on each of the n nodes in n copies of the n-node tree given". This results in the two-node tree  corresponding to (1.7), which

again can be processed according to the same rule yielding a sum of two trees,  . Here we assume a vector space on the trees, in which the addition operator corresponds to addition in (1.8). Further differentiation is now significantly less tedious than the cumbersome application of the Leibniz and chain rule on (1.8). Table 1.1 lists the first elementary differentials and their corresponding trees. The differentiation process on trees described here is called the "Natural growth operator" on trees, and is often denoted $N(\tau)$ for a tree τ.

Order $\lvert\tau\rvert$	Tree τ	Elementary differential $F(\tau)$
1		f
2		$f'f$
3		$f''f^2$
3		$(f')^2 f$

Table 1.1: Trees and corresponding elementary differentials. The argument y in $f(y)$ is omitted. A numerical scheme of order 3 must satisfy a condition for each of the trees depicted here.

On the vector space of trees, one can construct an algebra, consisting of a unit and a product. The empty tree $\emptyset$ is the unit, and the product of trees is juxtaposition of trees into forests, for example $\mu($  $\otimes$  $) =$. These trees are unordered, so the product is commutative. On the same set of trees, one defines a coalgebra, that is a counit and a coproduct Δ. The coproduct yields sums of tensor products of trees, $\Delta(\tau) = \sum_{\text{cuts } C \text{ of } \tau} P^C(\tau) \otimes R^C(\tau)$. A cut C in a tree τ is a collection of edges removed from the tree in such a way that along a path from the root node to a leaf node, at most one edge is removed. $R^C(\tau)$ is the part containing the original root node after the cut, and $P^C(\tau)$ is the remaining forest of trees. The product and the coproduct are compatible in such a way that this defines a *bialgebra* on the set of trees. Together with yet another operation on the trees, the *antipode*, this constitutes a *Hopf* algebra on the set of unordered rooted trees. We refer to [47] for details on Hopf algebraic structures in general.

B-series is a fundamental object in Butcher's analysis of Runge–Kutta schemes, introduced by Hairer and Wanner in [25],

$$B(\boldsymbol{a}, y) = \boldsymbol{a}(\emptyset) + \sum_{\tau \in T} \frac{h^{|\tau|}}{\sigma(\tau)} \boldsymbol{a}(\tau) F(\tau)(y) \tag{1.11}$$

where in this definition, $\boldsymbol{a}$ is a map from the set of trees T to the real numbers. h is a step size parameter, σ is a symmetry coefficient, and $F(\tau)$ represents the elementary differential associated to the tree τ, as in Table 1.1. Both the exact

solution and the numerical solution provided by a Runge–Kutta scheme have a B-series expansion, that is, some map $\boldsymbol{a}\colon T \to \mathbf{R}$. A particularly remarkable result by Butcher [11] is that the Runge–Kutta schemes are dense in the set of maps $\boldsymbol{a}\colon T \to \mathbf{R}$. Among these maps, there is also one map representing the exact solution.

The Butcher group consisted of such maps $\boldsymbol{a}$, from unordered trees into real numbers, and on this group, one is able to define an algebra structure as well. This is the dual of the tree algebra. It was Dür in [21] that pointed out that Butcher's group was in fact an instance of a Hopf algebra. The product of two maps representing B-series is calculated according to the coproduct in the tree-algebra, and because Runge–Kutta schemes are dense in this space, the result is another Runge–Kutta method, the *composed* scheme (see also [43]). The antipode in the tree algebra, will represent an inverse on the space of maps $T \to \mathbf{R}$, and has its application as the adjoint Runge–Kutta method.

For Lie group methods, one difference for the order analysis is that vector fields on a manifold will not in general commute. A consequence of this is that the trees need to be ordered.  is no longer equivalent to  as for unordered trees. A Hopf algebra can be constructed as before, with necessary modifications. B-series are replaced with "Lie–series". The dual of this space contains the maps from trees to real numbers constituting the Lie–series, and this (dual) is the Grossman–Larson algebra [23] on ordered rooted trees discussed in Paper I.

Parallel to the development in numerical analysis, Kreimer [32] used a Hopf algebra of rooted trees to simplify combinatorics of renormalization in quantum field theory and Connes and Moscovici [17] used a Hopf algebra in non-commutative geometry. The algebras were shown to be equivalent to the Hopf algebra used for analyzing Runge–Kutta schemes by Brouder in [8] and later reviewed in [9].

Paper I discusses the algebraic structure on ordered rooted trees, and show implications to Lie group integrators. The Lie group methods of Crouch and Grossman [19] were analyzed in terms of frozen vector fields in [37], and these frozen vector fields are connected to the Grossman–Larson Hopf algebra. Applications of the theory mentioned in Paper I include counting the number of order conditions using the theory on free Lie algebras, and also backward error analysis of Lie group integrators. A recent paper [41] goes deep into the algebra structures of ordered rooted trees, providing explicit and recursive formulas for the products, coproducts and antipodes for the algebra. The algebra of unordered rooted trees is also proved to be a sub-algebra, realized as the image of a tree symmetrization map. The algebra on trees, and the concept of B-series, has been extended and modified for greater applicability, see [42] for an overview. Recently generalizations have been used to characterize conditions on B-series for the preservation of invariants of motion, [15, 16].

1.4 Stiff equations and exponential integrators

Solving stiff equations has been a challenge ever since Curtiss and Hirschfelder first gave meaning to the notion in 1952 [20]. Though never precisely defined, the term *stiff* refers to the fact that for some problems, explicit Runge–Kutta methods "do not work" [26]. By not working, it is meant that the temporal step size h must be excessively small, rendering a numerical computation unfeasible, even though the solution itself may be smooth in time.

Section 1.3 discussed asymptotic analysis and expressed accuracy in terms of order of a numerical scheme (1.9), in principle only valid for the limit $h \to 0$. In addition to asymptotic order, there is a requirement of stability, which provides an upper bound for h, and this upper bound is what makes explicit integrators face problems for stiff equations.

As an example, forward Euler (1.2) is known to be unstable on the scalar complex equation

$$\dot{y}(t) = \lambda y(t), \qquad t \geq 0, \quad y(0) = 1, \quad \lambda \in \mathbf{C} \tag{1.12}$$

if $|1 + h\lambda| > 1$, or, if λ is real, for $h\lambda \notin [-2, 0]$. Depending on the magnitude of λ this may or may not be a severe restriction on h. More important is the generalizations to system of ODEs, in which the integrator must be stable for all eigenmodes of the system at the same time. The eigenvalues of the systems are in many cases so large in magnitude that the stability requirement renders the forward Euler practically useless.

When semi-discretizing a partial differential equation of evolution into a system of ordinary differential equations, the stability properties can usually be determined by extracting the linear part of the system. If the partial differential equation involves space differentiation of order r, the condition number for this linear part will be $\mathcal{O}(n^r)$ where n is the number of equations. High spatial accuracy is obtained by increasing n, the problem is that for stability, the upper limit for the time step h is $\mathcal{O}(1/n^r)$, which quickly becomes unacceptable in most situations. This argument is valid both for the forward Euler (1.2) and Kutta's fourth order scheme (1.10), as the size of their stability domains are both of magnitude $\mathcal{O}(1)$.

Curtiss and Hirschfelder [20] proposed backward differentiation formulae for the numerical solution of stiff equations. These schemes are a subset of multistep methods, and reduce to implicit Euler (or backward Euler). The important point here for the remedy is *implicitness*. For implicit schemes, there is a system of equations (possibly nonlinear) that needs to be solved for every time step. This requires the use of some iterative procedure, normally Newton iterations or a variation thereof. The downside is naturally computational complexity, but the step size restriction seen in forward Euler, is no longer present.

Being able to avoid *both* the step size requirement and implicitness is a feature of exponential integrators. This is achieved by extracting the linear part of the differential equation, often the cause of the step size requirement, and treating that part in an exact manner. Linear (systems of) first order differential equations, like (1.12), has an exact solution provided by the (matrix) exponential. The numerical effort spent in solving a nonlinear system of equations for implicit integrators, is now transferred to the evaluation of the matrix exponential.

Split the equation as

$$\dot{y} = f(y,t) = Ly + N(y,t), \qquad y(0) = y_0, \quad y(t) \in \mathbf{R}^n. \tag{1.13}$$

where Ly is a linear part extracted from $f(y)$ (not necessarily the full Jacobian of $f(y)$) and $N(y,t)$ the remainder, possibly nonlinear. The advantage gained by using exponential integrators compared to other choices lies in a successful choice of this splitting. As more information is inserted into the linear part, the more expensive the computations involving the matrix exponential of L will be. Often when solving PDEs, there is a natural splitting already in the PDE formulation, and many examples are provided in Paper III.

The generalization of the forward Euler scheme to an exponential integrator for (1.13) is ambiguous. There are two candidates, first the Lawson–Euler [34]

$$y_{n+1} = \mathrm{e}^{hL} y_n + h\mathrm{e}^{hL} N(y_n, t_n)$$

and then Nørsett–Euler [44]

$$y_{n+1} = \mathrm{e}^{hL} y_n + \frac{\mathrm{e}^{hL} - 1}{hL} hN(y_n, t_n)$$

also coined ETD-Euler, Lie–Euler, exponential Euler, exponentially fitted Euler etc. The expression $(\mathrm{e}^{hL} - 1)/(hL)$ is typically written $\varphi_1(hL)$ and is an example of what we later will call a φ function, a function class essential for exponential integrators, and due to the singularity at $h = 0$, there are also numerical issues with their computation.

Exponential integrators have roots at least back to Certaine [14] and Lawson [34]. Nørsett introduced exponential integrators of Adams–Bashforth type in [44], rediscovered recently by Cox and Matthews in [18]. The initial work was done despite the knowledge that the evaluation of the matrix exponential was computationally expensive and unfeasible in many contexts. In spite of this, theory of the methods was developed in [29] and [22]. It is the recent progress in numerical linear algebra, that has lead the way for a revived interest in exponential integrators, see [27].

Through the splitting into a linear and a nonlinear part, the features of the differential equation that have inherent bound on step sizes h, are (hopefully)

separated from the remainder. Exponential integrators then solve the equation $\dot{y} = Ly$ exactly by exponentiating the matrix L, intermixed with treatment of the remainder N in such a way that the two following conditions are fulfilled;

1. If $N(y,t) = 0$, an exponential integrator yields the exact solution.

2. If $L = 0$, an exponential integrator reduces to the *underlying* Runge–Kutta scheme.

The classical order analysis is valid for a finite discretization of a partial differential equation when $h\|L\| \to 0$. However, there might be no upper bound on $\|L\|$ for increasing spatial resolution in some PDE settings, with the result that the truncation error committed by the exponential integrator can not be bounded independently of the spatial resolution parameter n. In [28] *stiff order conditions* for exponential integrators are introduced, having classical order conditions as a subset. For semi-linear parabolic differential equations, the error can now be bounded according to the *stiff* order of the exponential integrator. Quite a few integrators with a given classical order, have a lower stiff order and will exhibit order reduction in certain cases, but fulfilling an additional number of order conditions incurs a higher computational demand. Paper IV and V include some discussion on the implications of schemes with different stiff order based on numerical results for the nonlinear Schrödinger equation. For the construction of schemes with high stiff order, the extension of exponential integrators of Runge–Kutta type to the general linear method type, as done in [33], seems viable. See also [45, 48] for numerous examples.

The papers II–VI are concerned with various aspects of exponential integrators. Paper II analyzes order conditions in a classical way, similar to the work of Friedli in [22], using tools mentioned in Section 1.3 but with bicolored trees. The results are applicable for constructing high order schemes using the least number of exponential-like evaluations within the format of the schemes. Paper III describes a software framework for implementing exponential integrators modularly in MATLAB. The accompanying source code has been used in all computations in this thesis. The numerical computation of the φ functions have issues at $z = 0$ as previously mentioned, and the code presented solves this by scaling the argument z and modified squaring of the result. Paper IV and V discuss the performance of exponential integrators on the nonlinear Schrödinger equation. Compared to other types of integrators, exponential integrators perform well with regard to accuracy vs. efficiency. Paper VI is a contribution to research on numerical approximations of the φ functions, as all the numerical properties of the current implementation are not known. A framework for solving non-autonmous differential equations is constructed using methodology from Lie group integrators and a particular homogeneous space, in which the φ functions play a central role. Some

properties of the functions are then derived, and also it is shown how to use this to construct further types of exponential integrators.

1.5 Summary of papers

This section summarizes the papers less technically than the abstract printed initally in each chapter.

Paper I — Algebraic Structures on Ordered Rooted Trees and Their Significance to Lie Group Integrators, page 19

Authors: Håvard Berland and Brynjulf Owren
Published in Group theory and numerical analysis, CRM Proceedings and Lecture Notes, 39:49–63, 2005. [3]

The Grossman–Larson algebra on the set of ordered rooted trees is discussed. The product in the algebra is associated to composition of vector fields on manifolds, using the notion of frozen and unfrozen operators. For frozen operators, the Grossman–Larson product simplifies to a shuffle product. Both structures can be equipped with the same coalgebra structure and an antipode, and results therefrom are mentioned. It is shown how it is possible to count the number of order conditions that must be fulfilled for a Lie group integrator to be of some order, the antipode is stated, and an application of the logarithm is used to facilitate backward error analysis.

Paper II — *B*-series and order conditions for exponential integrators, page 41

Authors: Håvard Berland, Brynjulf Owren, and Bård Skaflestad
Published in SIAM Journal of Numerical Analysis, 43(4):1715–1727, 2005. [4]

Exponential integrators of Runge–Kutta type are presented in a uniform format with arbitrary coefficient functions. Using classical asymptotic order analysis including B-series, order conditions for the coefficient functions are listed. Using an approach of Zennaro, we present a way to construct exponential integrators, and thereby prove the classical order of which some recently schemes. To limit the number of coefficient function evaluations necessary, the coefficient function can be chosen from finite dimensional function spaces and the paper provides lower bounds on these spaces' dimensions. An algorithm for constructing arbitrary high order exponential integrator is given together with a fifth order example based on Fehlberg's scheme.

Paper III — EXPINT — A MATLAB package for exponential integrators, page 61

Authors: Håvard Berland, Bård Skaflestad, and Will M. Wright
To appear in ACM Transactions on Mathematical Software 2007. [6]

This publication is a collection and description of a MATLAB framework made for easy implementation and testing of exponential integrators. Each exponential integrator is described by its coefficient functions, and this package allows for easy modification of existing schemes and modularity in designing an application using exponential integrators. The paper and software contributes an extensive set of currently known exponential integrators and their coefficient functions. The format for exponential integrators presented in [4] is also generalized to cater for exponential general linear methods, which enriches the class of exponential integrators significantly. A major contribution of the publication is the approach and code for computing so called φ functions, a class of functions present in most exponential integrator, but which presents numerical issues when implemented on a computer. The approach taken consists of scaling the function argument, calculating the scaled argument using a diagonal Padé approximation, and then use corrected squaring to obtain the final result. The accompanying software and README-file include numerous examples of known PDEs solved using exponential integrators.

Paper IV — Solving the nonlinear Schrödinger equation using exponential integrators, page 87

Authors: Håvard Berland, Brynjulf Owren, and Bård Skaflestad
To appear in Modeling, Identification and Control, 27(4), 2006. [5]

Behavior of exponential integrators on the nonlinear Schrödinger equation is analyzed. In particular, the differences between exponential integrators due to Lawson and the early fourth order integrator by Cox and Matthews are examined. The parameters varied in the study are mainly the smoothness of the initial condition, and also the potential in the Schrödinger equation. The Lawson integrator, of stiff order only one, exhibits numerical problems and order reduction for a non-smooth potential. This is proved to be related to the low stiff order of the scheme. On the other hand, the Cox and Matthews scheme of stiff order two exhibits order reduction on low-regularity on non smooth initial condition, which again is partly described by a linearized analysis.

Paper V — Conservation of phase space properties for the cubic Schrödinger equation, page 109

Authors: Håvard Berland, Alvaro L. Islas, and Constance M. Schober
Preprint Numerics No. 1/06, Department of Mathematical Sciences, Norwegian University of Science and Technology. Submitted to Journal of Computational Physics. [2]

The nonlinear Schrödinger equation, with zero potential when compared with the one used in Paper IV, is used as a benchmark problem for testing exponential integrators alongside a multisymplectic solver. Multisymplectic solvers, described in [7], have recently exhibited advantageous numerical properties when compared to classical integrators [31]. The results indicate that a commutator-free exponential integrator of order 4 is best suited for this equation, in terms of accuracy and computational time. Included are also a Lawson exponential integrator of order 4 and a split-step scheme of order 4 obtained by Yoshida's method. The performance of the integrators are assessed by monitoring their ability to conserve the associated nonlinear spectrum of the initial condition, which is invariant in time. The initial condition chosen is a small perturbation of a solution which has unstable eigenmodes, and is thus a numerical challenge for all types of integrators.

Paper VI — Generalized affine groups in exponential integrators, page 137

Author: Håvard Berland
Preprint Numerics No. 2/06, Department of Mathematical Sciences, Norwegian University of Science and Technology. [1]

The affine matrix group together with $\mathbf{R}^n$ is a simple example of a homogeneous space, on which one is able to apply any Lie group integrator. This was already mentioned in the early paper [39], and the methodology provides one important source of recipes for constructing exponential integrators. This paper extends the homogeneous space by adding an arbitrary number of additional vectors to the group and develops a full framework for constructing Lie group integrators using the generalized affine groups for solving non-autonomous ordinary differential equations. The work can be seen as an extension to work done in [38], although the main objective for this paper is to reveal properties of the so called φ functions. φ functions are key objects in exponential integrators, and possess some numerical difficulties, but have not been analyzed in full yet. Some properties of φ functions arise as a by product of the development of the framework, and it is hoped that these result may help further analysis on φ functions.

Bibliography

[1] Berland, H.: Generalized affine groups in exponential integrators. Tech. Rep. 2/06, Department of Mathematical Sciences, NTNU, Norway (2006). `http://www.math.ntnu.no/preprint/`

[2] Berland, H., Islas, A.L., Schober, C.M.: Conservation of phase space properties for the cubic Schrödinger equation. Tech. Rep. 1/06, Department of Mathematical Sciences, NTNU, Norway (2006). `http://www.math.ntnu.no/preprint/`

[3] Berland, H., Owren, B.: Algebraic structures on ordered rooted trees and their significance to Lie group integrators. In: Group theory and numerical analysis, *CRM Proc. Lecture Notes*, vol. 39, pp. 49–63. Amer. Math. Soc., Providence, RI (2005)

[4] Berland, H., Owren, B., Skaflestad, B.: B-series and order conditions for exponential integrators. SIAM J. Numer. Anal. **43**(4), 1715–1727 (2005)

[5] Berland, H., Owren, B., Skaflestad, B.: Solving the nonlinear Schrödinger equation using exponential integrators. Modeling, Identification and Control **27**(4) (2006)

[6] Berland, H., Skaflestad, B., Wright, W.M.: Expint — A Matlab package for exponential integrators. ACM Trans. on Math. Soft. (2006). To appear

[7] Bridges, T.J., Reich, S.: Multi-symplectic integrators: numerical schemes for Hamiltonian PDEs that conserve symplecticity. Phys. Lett. A **284**(4-5), 184–193 (2001)

[8] Brouder, C.: Runge–Kutta methods and renormalization. Eur. Phys. J. C **12**, 521–534 (2000)

[9] Brouder, C.: Trees, renormalization and differential equations. BIT **44**(3), 425–438 (2004)

[10] Butcher, J.C.: Coefficients for the study of Runge–Kutta integration processes. J. Austral. Math. Soc. **3**(185–201) (1963)

[11] Butcher, J.C.: An algebraic theory of integration methods. Math. Comp. **26**, 79–106 (1972)

[12] Cayley, A.: On the Theory of Analytic Forms Called Trees. Philos. Mag. **13**, 19–30 (1857)

[13] Celledoni, E., Marthinsen, A., Owren, B.: Commutator-free Lie group methods. FGCS **19**(3), 341–352 (2003)

[14] Certaine, J.: The solution of ordinary differential equations with large time constants. In: Mathematical methods for digital computers, pp. 128–132. Wiley, New York (1960)

[15] Chartier, P., Faou, E., Murua, A.: An algebraic approac to invariant preserving integrators: The case of quadratic and Hamiltonian invariants (2006). To appear in Numerische Mathematik

[16] Chartier, P., Murua, A.: Preserving first integrals and volume forms of additively split systems (2006). Submitted to IMA Journal of Numerical Analysis

[17] Connes, A., Moscovici, H.: Hopf algebras, cyclic cohomology and the transverse index theorem. Comm. Math. Phys. **198**(1), 199–246 (1998)

[18] Cox, S.M., Matthews, P.C.: Exponential time differencing for stiff systems. J. Comput. Phys. **176**(2), 430–455 (2002)

[19] Crouch, P.E., Grossman, R.: Numerical integration of ordinary differential equations on manifolds. J. Nonlinear Sci. **3**, 1–33 (1993)

[20] Curtiss, C.F., Hirschfelder, J.O.: Integration of stiff equations. Proc. Nat. Acad. Sci. **38**, 235–243 (1952)

[21] Dür, A.: Möbius functions, incidence algebras and power series representations, *Lecture Notes in Mathematics*, vol. 1202. Springer-Verlag, Berlin (1986)

[22] Friedli, A.: Verallgemeinerte Runge–Kutta Verfahren zur Lösung steifer Differentialgleichungssysteme. In: Numerical treatment of differential equations, pp. 35–50. Lecture Notes in Math., Vol. 631. Springer, Berlin (1978)

[23] Grossman, R., Larson, R.G.: Hopf-algebraic structure of families of trees. J. Algebra **126**(1), 184–210 (1989)

[24] Hairer, E., Nørsett, S.P., Wanner, G.: Solving ordinary differential equations. I, *Springer Series in Computational Mathematics*, vol. 8, second edn. Springer-Verlag, Berlin (1993). Nonstiff problems

[25] Hairer, E., Wanner, G.: On the Butcher group and general multivalue methods. Computing **13**, 1–15 (1974)

[26] Hairer, E., Wanner, G.: Solving Ordinary Differential Equations. II, *Springer Series in Computational Mathematics*, vol. 14. Springer-Verlag, Berlin (1991). Stiff and differential-algebraic problems

[27] Hochbruck, M., Lubich, C.: On Krylov subspace approximations to the matrix exponential operator. SIAM J. Numer. Anal. **34**(5), 1911–1925 (1997)

[28] Hochbruck, M., Ostermann, A.: Explicit exponential Runge–Kutta methods for semilinear parabolic problems. SIAM J. Numer. Anal. **43**(3), 1069–1090 (2005)

[29] van der Houwen, P.J., Verwer, J.G.: Generalized linear multistep methods. I. Development of algorithms with zero-parasitic roots. Mathematisch Centrum, Amsterdam (1974). Mathematisch Centrum, Afdeling Numerieke Wiskunde NW 10/74

[30] Iserles, A., Munthe-Kaas, H.Z., Nørsett, S.P., Zanna, A.: Lie-group methods. Acta Numerica **9**, 215–365 (2000)

[31] Islas, A.L., Schober, C.M.: On the preservation of phase space structure under multisymplectic discretization. J. of Comp. Phys. **197**, 585–609 (2004)

[32] Kreimer, D.: On the Hopf algebra structure of perturbative quantum field theories. Adv. Theor. Math. Phys. **2**, 303–334 (1998)

[33] Krogstad, S.: Generalized integrating factor methods for stiff PDEs. J. of Comp. Phys. **203**(1), 72–88 (2005)

[34] Lawson, J.D.: Generalized Runge–Kutta processes for stable systems with large Lipschitz constants. SIAM J. Numer. Anal. **4**, 372–380 (1967)

[35] Lewis, D., Simo, J.C.: Conserving algorithms for the dynamics of Hamiltonian systems of Lie groups. J. Nonlinear Sci. **4**, 253–299 (1994)

[36] Lewis, D., Simo, J.C.: Conserving Algorithms for the N Dimensional Rigid Body. In: J.E. Mardsen, G.W. Patrick, W.F. Shadwick (eds.) Integration Algorithms and Classical Mechanics, pp. 121–140. American Mathematical Society (1996)

[37] Marthinsen, A., Owren, B.: A note on the construction of Crouch–Grossman methods. BIT **41**(1), 207–214 (2001)

[38] Minchev, B.V.: Lie group integrators with nonautonomous frozen vector fields. Tech. rep., Department of Computer Science, University of Bergen (2004). Preprint submitted to Elsevier Science

[39] Munthe-Kaas, H.: High order Runge–Kutta methods on manifolds. In: Proceedings of the NSF/CBMS Regional Conference on Numerical Analysis of Hamiltonian Differential Equations (Golden, CO, 1997), vol. 29,1, pp. 115–127 (1999)

[40] Munthe-Kaas, H., Owren, B.: Computations in a free Lie algebra. R. Soc. Lond. Philos. Trans. Ser. A Math. Phys. Eng. Sci. **357**(1754), 957–981 (1999)

[41] Munthe-Kaas, H., Wright, W.M.: On the Hopf Algebraic Structure of Lie Group Integrators. Tech. rep., University of Bergen, Department of Mathematics (2006)

[42] Murua, A.: Formal series and numerical integrators. I. Systems of ODEs and symplectic integrators. Appl. Numer. Math. **29**(2), 221–251 (1999)

[43] Murua, A., Sanz-Serna, J.M.: Order conditions for numerical integrators obtained by composing simpler integrators. R. Soc. Lond. Philos. Trans. Ser. A Math. Phys. Eng. Sci. **357**(1754), 1079–1100 (1999)

[44] Nørsett, S.P.: An A-stable modification of the Adams–Bashforth methods. In: Conf. on Numerical Solution of Differential Equations (Dundee, Scotland, 1969), pp. 214–219. Springer, Berlin (1969)

[45] Ostermann, A., Thalhammer, M., Wright, W.M.: A class of explicit exponential general linear methods. BIT **46**(2), 409–432 (2006)

[46] Owren, B., Marthinsen, A.: Runge–Kutta methods adapted to manifolds and based on rigid frames. BIT **39**(1), 116–142 (1999)

[47] Sweedler, M.: Hopf Algebras. W. A. Benjamin, Inc. (1969)

[48] Thalhammer, M., Wright, W.M.: Generalized Lawson methods (2006). In preparation

Paper I

Håvard Berland and Brynjulf Owren:

Algebraic Structures on Ordered Rooted Trees and Their Significance to Lie Group Integrators

Published in Group theory and numerical analysis, CRM Proceedings and Lecture Notes, 39:49–63, 2005.

Chapter 2

Algebraic structures on ordered rooted trees and their significance to Lie group integrators

Abstract:

Most Lie group integrators can be expanded in series indexed by the set of ordered rooted trees. One can associate to each tree two distinct higher order derivation operators, which we call frozen and unfrozen operators. Composition of frozen operators induces a concatenation product on the trees, whereas composition of unfrozen operators induces a somewhat more complicated product known as the Grossman–Larson product. Both of these algebraic structures can be supplemented by the same coalgebraic structure and an antipode, the result being two distinct cocommutative graded Hopf algebras. We discuss the use of these structures and characterize subsets of the Hopf algebras corresponding to vector fields and mappings on manifolds. This is relevant for deriving order conditions for a general class of Lie group integrators and for deriving the modified vector field in backward error analysis for these integrators.

2.1 Introduction

The derivation of high order Runge–Kutta methods was revolutionized by Butcher's discovery of the beautiful connection between their series expansion in terms of the stepsize and the set T of rooted trees [2]. Virtually overnight, long and tedious calculations were replaced by elegant recursion formulas expressed in terms of trees. Later on, in 1972 Butcher published a paper [3] where he showed that Runge–Kutta methods form a group under composition, and derived explicit expressions for the group operations as induced on the trees. Hairer and Wanner named it the Butcher group and contributed substantially to the theory in [10]. The group is defined on the dual of the tree space by using a bialgebra structure on the space of rooted trees.

More recently, Kreimer [13] used a Hopf algebra of rooted trees to deal with the combinatorics of renormalization in quantum field theory, the connection to the work of Butcher was nicely presented in [1].

In the last few years, new classes of integrators have been subjected to order

analysis by means of trees. In [19] order conditions for composition methods are studied by means of so called ∞-trees, see also [8]. Another class of novel schemes is the one based on Lie group actions on manifolds. Such integrators, which generalize classical Runge–Kutta methods, are now commonly referred to as Lie group integrators. Early contributors to this class of schemes include Crouch and Grossman [6] as well as Lewis and Simo [14, 15]. The Lie group schemes were later subjected to a more systematic treatment by many authors, see the survey [12]. In [16] Munthe-Kaas showed how a certain subclass of the Lie group integrators could be expanded in a series for the purpose of order analysis, but his approach was not pursued any further at that time since he discovered in [17] that a suitable change of variable would allow him to use the classical Butcher theory for deriving the order conditions. But the schemes of Crouch and Grossman [6] did not fit into this framework, and so Owren and Marthinsen [21] developed a slightly different framework based on *ordered rooted trees* for deriving the general order conditions for such schemes.

In this note, we will see how the algebraic structure introduced by Butcher can be extended to the set of ordered rooted trees. In particular we will present two different Hopf algebras, the first one was introduced by Grossman and Larson in [7] and the second is related to the one presented by Reutenauer in [22]. We will discuss their relevance to order conditions and backward error analysis for a general class of Lie group integrators.

2.2 Lie group integrators

An ordinary differential equation on a manifold has various different formulations,

$$y' = F(y) = f(y) \cdot y = \sum_i f^i(y)\, E_i(y). \tag{2.1}$$

F is here a smooth vector field on $\mathcal{M}$. The second equality tacitly refers to a transitive action on $\mathcal{M}$ by a Lie group G, and $f : \mathcal{M} \to \mathcal{V} \subset \mathfrak{g}$ where $\mathfrak{g}$ is the Lie algebra of G, and $\mathcal{V}$ is a subspace of $\mathfrak{g}$. The notation $v \cdot p$, $v \in \mathfrak{g}$, $p \in \mathcal{M}$ signifies the derivative of the group action in the sense that

$$v \cdot p = \left.\frac{\mathrm{d}}{\mathrm{d}t}\right|_{t=0} \exp(tv) \cdot p.$$

In the last equality of (2.1), we have used a set of frame vector fields $E_1, \ldots, E_d$ that may be defined as $E_i(p) = e_i \cdot p$ for some basis $e_1, \ldots, e_d$ of $\mathfrak{g}$ (or $\mathcal{V}$). The functions $f^i \colon \mathcal{M} \to k$ are then given such that $f(y) = \sum f^i(y)\, e_i \cdot y$. To the end of this note, we shall always assume that the field k is either $\mathbf{R}$ or $\mathbf{C}$. Note that the action is usually not assumed to be free. For our purposes, the language of

actions and frames can be used interchangeably, but for this present exposition we find it slightly advantageous to use frames. By a minor abuse of notation, we shall therefore denote by $\mathcal{V}$ also the linear span of the frame vector fields

$$\mathcal{V} = \operatorname{span}\{E_1(y), \ldots, E_d(y)\}.$$

We will as usual interpret vector fields as derivations acting on functions $\psi : \mathcal{M} \to k$, and denote this action by $F[\psi]$ for any $F \in \mathfrak{X}(\mathcal{M})$.

In order to develop numerical integrators, we approximate the vector field by the *freeze* map $\operatorname{Fr}\colon \mathcal{M} \times \mathfrak{X}(\mathcal{M}) \to \mathcal{V}$, in which the vector fields E_i are assumed to be simpler to integrate than F,

$$\left(p, \sum_i f^i(y) E_i(y)\right) \mapsto \sum_i f^i(p) E_i(y). \tag{2.2}$$

We propose a scheme for Lie group integrators which generalizes both Crouch–Grossman (CG) type of schemes [6, 21], and the Runge–Kutta–Munthe-Kaas (RKMK) methods [16, 17]. The scheme uses the point $p \in \mathcal{M}$ as input, together with a stepsize $h \in \mathbf{R}$ and produces as output a point $y_1 \in \mathcal{M}$ such that $y_1 \approx \exp(hF)\, p$.

$$g_i = \exp\left(\sum_r \alpha^r_{i,J} K_r\right) \cdots \exp\left(\sum_r \alpha^r_{i,1} K_r\right) p \tag{2.3}$$

$$\bar{K}_i = h \operatorname{Fr}(g_i, F) = h \sum_\ell f^\ell(g_i) E_\ell \tag{2.4}$$

$$K_i = P_i\left(\bar{K}_1, \ldots, \bar{K}_s\right) \quad \text{(Lie polynomial)} \tag{2.5}$$

$$y_1 = \exp\left(\sum_r \beta^r_J K_r\right) \cdots \exp\left(\sum_r \beta^r_1 K_r\right) p \tag{2.6}$$

In (2.3) to (2.5) the index i runs from 1 to s, where s is the number of stages in the step.

This scheme belongs to the RKMK class if $J = 1$ in (2.3) and (2.6) (only one exponential used for g_i and y_1). If $\alpha^r_{i,j} = 0$ when $r \neq j$ in (2.3), $\beta^r_j = 0$ when $r \neq j$ in (2.6), and $K_i = \bar{K}_i$ for all i, then this is a Crouch–Grossman scheme. Requiring only $\bar{K}_i = K_i$ for all i, one obtains a commutator-free Lie group scheme [4].

The scheme is explicit if P_i in (2.5) depends only on $\bar{K}_1, \ldots, \bar{K}_i$ and $\alpha^r_{i,j} = 0$ when $r \geq i,\ 1 \leq j \leq J$.

In the sequel we shall introduce an algebraic formalism which will provide a unified setting for analysis of Lie group integrators of the form (2.3)—(2.6).

2.3 Algebraic structures on trees

In this section we shall impose algebraic structures on the space of ordered rooted trees. We will define two distinct associative products and a unit element and thereby introduce two different algebra structures. We next define a coassociative coproduct with counit to obtain one coalgebra which can be used with each of the two algebras to form two distinct bialgebras. Each of these can be equipped with an antipode, the result being two distinct cocommutative graded Hopf algebras. A good reference for the theory of Hopf algebras is Sweedler's book [23], to which we will refer frequently.

The set of all ordered rooted trees is denoted by T_O, see e.g. [5] for a rigorous treatment of such trees and their combinatorial properties. We shall work recursively with trees, and our notation is based on the fact that a tree $t \in T_O$ is either the one-node tree $\bullet \in T_O$ (the identity tree), or obtained by connecting the roots of an ordered set of trees $t_1, \ldots, t_\mu$ to a new common root, where each $t_i \in T_O$. We use the notation $t = B_+(t_1, \ldots, t_\mu)$ for this operation. Conversely, $B_-(t)$ will denote the (ordered) set of trees obtained by deleting the root of t, and we call this set the subtrees of t. Let $\sigma(t)$ be the underlying set of nodes of t. The number of nodes in a tree t is denoted $|t|$, and we let the grading be $\nu(t) = |t| - 1$. So $\nu(\bullet) = 0$, and for any other tree $t = B_+(t_1, \ldots, t_\mu)$ one has $\nu(t) = \mu - 1 + \sum_{i=1}^{\mu} \nu(t_i)$. It is well known, see e.g. [5], that the number ν_r of trees such that $\nu(t) = r$ is given by the Catalan number

$$\nu_r = \frac{1}{r+1}\binom{2r}{r}. \tag{2.7}$$

The linear space kT_O is obtained by forming finite linear combinations of trees over the field k. The homogeneous components of grade m is the subspace of kT_O spanned by the trees $\{t \in T_O : \nu(t) = m\}$.

We shall later make use of the subset of T_O consisting of trees with only one subtree at the first level, we denote this subset by T_O^1, and let kT_O^1 be the corresponding subspace of kT_O.

$$T_O^1 = \{t \in T_O : t = B_+(t'), \quad t' \in T_O\}.$$

Grossman–Larson product on trees

An *attachment map* is a map which associates to any element of an ordered finite subset S of T_O a unique element of another set M, which will typically be the nodes of a tree, $\sigma(w)$. We write $d\colon S \to \sigma(w)$ for this map. By the notation

$$w \leftharpoonup_d S \tag{2.8}$$

we mean an augmented tree t where each tree $s \in S$ has been attached to the node $d(s)$ of w by adding an edge from the root of s to $d(s)$. Thus, s becomes a subtree of the tree rooted at $d(s)$. This subtree will be ordered *before* any of the existing subtrees at $d(s)$. Moreover, if $d(s) = d(s')$ where $s < s'$ then s will be ordered before s' as subtrees at $d(s)$ in t. By convention, we depict this ordering by grafting elements of S to the left of the already existing subtrees at nodes in t. An example attachment is

$$\bullet \leftharpoonup_d \{s_1, s_2\} =$$

where we have $d(s_1) = \bullet$ and $d(s_2) = \star$. s_1 and s_2 are arbitrary trees in T_O.

The sum over all possible attachment maps d from S to $\sigma(w)$ is written as

$$w \leftharpoonup S := \sum_d w \leftharpoonup_d S.$$

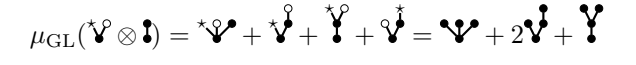
Definition 2.1 (Grossman–Larson algebra)**.** The identity element is the one-node tree $\bullet$, and the product of two trees is

$$\mu_{\mathrm{GL}}(v \otimes w) = w \leftharpoonup B_-(v), \quad v, w \in T_O$$

The unit in the algebra is $u : k \to kT_O$ given by $u(\alpha) = \alpha\bullet$ for $\alpha \in k$.

This product is non-commutative. An example is

$$\mu_{\mathrm{GL}}(\ \otimes\ ) =\ +\ +\ +\ =\ + 2\ +\ \tag{2.9}$$



where we have temporarily inserted a star and a circle for the grafted nodes to indicate the various attachment maps d involved in the sum.

Grossman and Larson prove in [7, Section 3] that this product is indeed associative.

Concatenation algebra on trees

We define a simpler product on the trees in T_O.

Definition 2.2 (Concatenation algebra)**.** The concatenation product of two trees v and w results from joining all the subtrees of both v and w to a new common root,

$$\mu_{\mathrm{M}}(v \otimes w) = B_+(B_-(v) \cup B_-(w)), \quad v, w \in T_O$$

such that the order is preserved. The identity is the one-node tree $\bullet$.

Alternatively, if $v = B_+(v_1, \ldots, v_\mu)$ and $w = B_+(w_1, \ldots, w_\nu)$, then $\mu_{\mathrm{M}}(v \otimes w) = B_+(v_1, \ldots v_\mu, w_1, \ldots w_\nu)$.

Note that this product has a subset of the terms arising from the product in the Grossman–Larson algebra. We may write $\mu_M(v \otimes w) = w \leftharpoonup_{d_0} B_-(v)$ where d_0 is the map that sends all elements of the set $B_-(v)$ to the root $r \in \sigma(w)$.

Coalgebra on trees

The two Hopf algebras on trees we will present, share the same coproduct $\Delta : kT_O \to kT_O \otimes kT_O$. We here follow the presentation in [7], but we refer to [23] for the basic theory on algebras and coalgebras.

Definition 2.3 (Coalgebra)**.** The coproduct on the trees in T_O is given by

$$\Delta(t) = \sum_{\mathcal{X} \subseteq B_-(t)} B_+(\mathcal{X}) \otimes B_+(\mathcal{X}^c)$$

which extends linearly to kT_O. The subsets $\mathcal{X}$ inherit the ordering from t as do the complements $\mathcal{X}^c$. We include the empty set $\emptyset$ as well as $B_-(t)$ in the sum, using the convention $B_+(\emptyset) = \bullet$. The counit is the linear map $\epsilon\colon kT_O \to k$ such that

$$\epsilon(\bullet) = 1 \quad \text{and} \quad \epsilon(t) = 0, \quad t \in T_O, t \neq \bullet.$$

We refer to [7] for a proof that the coproduct defined above is coassociative, meaning that $(I \otimes \Delta) \circ \Delta = (\Delta \otimes I) \circ \Delta$.

A bialgebra has both an algebra structure and a coalgebra structure which are compatible in the sense that the coproduct must be an algebra homomorphism (or equivalently, the product is a coalgebra homomorphism). That is

$$\Delta(\mu_{\mathrm{GL}}(v \otimes w)) = \mu_{\mathrm{GL}\otimes\mathrm{GL}}(\Delta(v) \otimes \Delta(w)). \tag{2.10}$$

We refer to Grossman and Larson [7, Section 3] for a proof of Equation (2.10). Note that the product structure $\mu_{\mathrm{GL}\otimes\mathrm{GL}}$ on $kT_O \otimes kT_O$ is naturally constructed using $\mu_{\mathrm{GL}\otimes\mathrm{GL}} = (\mu_{\mathrm{GL}} \otimes \mu_{\mathrm{GL}}) \circ (I \otimes T \otimes I)$ where $T\colon a \otimes b \mapsto b \otimes a$ is the twist map.

It is easier to prove that Δ is an algebra homomorphism with respect to the concatenation algebra.

Proposition 2.4. *The coproduct Δ given in Definition 2.3 is a concatenation algebra homomorphism, that is*

$$\Delta(\mu_{\mathrm{M}}(t_1 \otimes t_2)) = \mu_{\mathrm{M}\otimes\mathrm{M}}(\Delta(t_1) \otimes \Delta(t_2))$$

Proof. From the left we have

$$\begin{aligned} \Delta(\mu_{\mathrm{M}}(t_1 \otimes t_2)) &= \Delta(B_+(B_-(t_1) \cup B_-(t_2)) \\ &= \sum_{\mathcal{X} \subseteq B_-(t_1) \cup B_-(t_2)} B_+(\mathcal{X}) \otimes B_+(\mathcal{X}^c) \end{aligned}$$

and from the right, using $\mu_{\mathrm{M}\otimes\mathrm{M}} = (\mu_{\mathrm{M}} \otimes \mu_{\mathrm{M}}) \circ (I \circ T \circ I)$.

$$\begin{aligned}
&\mu_{\mathrm{M}\otimes\mathrm{M}}(\Delta(t_1) \otimes \Delta(t_2)) \\
&\quad = \mu_{\mathrm{M}\otimes\mathrm{M}} \left(\sum_{\mathcal{X}_1,\mathcal{X}_2} B_+(\mathcal{X}_1) \otimes B_+(\mathcal{X}_1^c) \otimes B_+(\mathcal{X}_2) \otimes B_+(\mathcal{X}_2^c) \right) \\
&\quad = (\mu_{\mathrm{M}} \otimes \mu_{\mathrm{M}}) \left(\sum_{\mathcal{X}_1,\mathcal{X}_2} B_+(\mathcal{X}_1) \otimes B_+(\mathcal{X}_2) \otimes B_+(\mathcal{X}_1^c) \otimes B_+(\mathcal{X}_2^c) \right) \\
&\quad = \sum_{\mathcal{X}_1,\mathcal{X}_2} B_+(\mathcal{X}_1 \cup \mathcal{X}_2) \otimes B_+(\mathcal{X}_1^c \cup \mathcal{X}_2^c)
\end{aligned}$$

which is a sum equivalent to the one above. □

It is also apparent from the definition of the coproduct that the coalgebra is cocommutative, that is $\Delta = \Delta \circ T$.

Grossman–Larson Hopf algebra

A mapping $\mathcal{S}\colon kT_O \to kT_O$ in a bialgebra, is an antipode [23] if it satisfies

$$\mu \circ (\mathcal{S} \otimes \mathrm{Id}) \circ \Delta = u \circ \epsilon = \mu \circ (\mathrm{Id} \otimes \mathcal{S}) \circ \Delta \tag{2.11}$$

Note that $u \circ \epsilon$ is zero on all trees in T_O except from $\bullet$, where one has $u \circ \epsilon(\bullet) = \bullet$. One may apply (2.11) with μ replaced by μ_{GL} recursively to obtain $\mathcal{S}_{\mathrm{GL}}(t)$ for any $t \in T_O$, noting that one gets $\mathcal{S}_{\mathrm{GL}}(\bullet) = \bullet$ as well as identities of the form

$$\mathcal{S}_{\mathrm{GL}}(t) = -t - \sum \mu_{\mathrm{GL}}(\mathcal{S}_{\mathrm{GL}}(t_{i_1}) \otimes t_{i_2}), \quad |t_{i_1}| < |t|.$$

An explicit formula for $\mathcal{S}_{\mathrm{GL}}(t)$ seems hard to derive or use in general. But there are some simple cases. For instance,

$$\mathcal{S}_{\mathrm{GL}}(t) = -t, \quad \text{for } t \in T_O^1. \tag{2.12}$$

For trees with exactly two branches emanating from the root, one has

$$\mathcal{S}_{\mathrm{GL}}(B_+(t_1, t_2)) = B_+(t_2, t_1) + B_+(t_1 \leftharpoonup t_2 + t_2 \leftharpoonup t_1)$$

where B_+ has been extended to a linear map.

Example 2.5 (Grossman–Larson antipode)**.**

$\mathcal{S}_{\mathrm{GL}}(\bullet) = \bullet$ $\qquad$ $\mathcal{S}_{\mathrm{GL}}($$) = -$

$\mathcal{S}_{\mathrm{GL}}($$) =$ $+ 2$

$\mathcal{S}_{\mathrm{GL}}($$) =$ $+ 2$ $+$ 

Concatenation Hopf algebra

We define the antipode for the concatenation algebra in the same way, but the trivial product provides an easy factorization which facilitates a formula for the antipode.

The antipode of a tree $t = B_+(t_1, \ldots, t_\mu)$ in the concatenation algebra takes the form

$$\mathcal{S}(t) = (-1)^\mu B_+(t_\mu, \ldots, t_1)$$

which follows from the fact that the antipode of a Hopf algebra is an antiautomorphism [23, Proposition 4.0.1, page 74] and from Equation (2.12).

Because the coproduct for both Hopf algebras is cocommutative, the antipodes have the property that $\mathcal{S}^2$ is the identity map on kT_O [23, Proposition 4.0.1, page 74].

Infinite series and their subsets

We may define formal series on T_O as infinite sums

$$S = \sum_{t \in T_O} S_t \, t$$

where $S_t \in k$ is the coefficient of the tree t in the series S. The set of all such series is denoted $k_\infty T_O$ of which kT_O is clearly a subspace. We can extend the bialgebras to this space by setting the coefficients of the product of two series S and T to

$$(\mu(S \otimes T))_t = \sum_{\mu(v \otimes w) = t} S_v T_w \tag{2.13}$$

the sum being finite since the homogeneous components of kT_O are finite dimensional. The coproduct Δ is extended in a similar way. We may now define the commutator between two series in either algebra as

$$[S, T] = \mu(S \otimes T - T \otimes S).$$

The primitive elements of the extended Hopf algebras are those which satisfy

$$\Delta(S) = S \otimes \bullet + \bullet \otimes S. \tag{2.14}$$

The linear space of primitive elements is from now on denoted by $\mathfrak{g}$, it is closed under the commutator and thus forms a Lie algebra. Later, we shall see that $\mathfrak{g}$ plays an important role as its members represent formal expansions of the vector fields used in the integration schemes. In particular, we observe from the

definition of the coproduct that $k_\infty T_O^1 \subset \mathfrak{g}$. Moreover, it follows immediately from (2.14) and the general definition of the antipode that

$$\mathcal{S}(S) = -S, \quad S \in \mathfrak{g}.$$

The Milnor–Moore theorem ensures that the universal enveloping algebra of $\mathfrak{g}$ is isomorphic to $k_\infty T_O$.

The series $T \in k_\infty T_O$ with coefficient $T_\bullet = \langle T, \bullet \rangle = 1$ form a group under the product in (2.13). Taking the formal exponential of all Lie series $\mathfrak{g}$, we obtain a subgroup $G = \exp(\mathfrak{g})$ with the property

$$\Delta(T) = T \otimes T. \tag{2.15}$$

if $T = \exp(S)$ for $S \in \mathfrak{g}$. The proof is simple, and can be found in [22, Thm. 3.2].

The group G is also invariant under the antipode, the antipode represents the group inverse, as we get

$$\mu(\mathcal{S}(T) \otimes T) = \bullet, \quad \text{for all } T \in G = \exp(\mathfrak{g}) \tag{2.16}$$

from the defining equation (2.11) of the antipode. Later, we shall see that the elements of G represent expansions of maps defined as part of the integration schemes.

2.4 Elementary high order derivations

In this section we return to the Lie group integration schemes. Suppose that a frame set (action) is given together with the vector field F in (2.1). Letting vector fields be derivation operators acting on the functions on the manifold, we shall recursively define high order derivation operators which will feature in the expansions of the various quantities appearing in the Lie group integrators. The algebra obtained by composing these operators will be seen to coincide with the tree algebras of the previous section.

High order derivations

Definition 2.6 (Elementary High Order Derivation, EHOD)**.** Let $F \in \mathfrak{X}(\mathcal{M})$ and $E_1, \dots, E_d \in \mathfrak{X}(\mathcal{M})$ be given. We define $\mathbf{F}$ to be the k-linear map from $k_\infty T_O$ to the vector space of high order derivations on the manifold such that for $t \in T_O$

$$\mathbf{F}(t) \colon \psi \mapsto \mathbf{F}(t)[\psi]$$

where $\mathbf{F}(\bullet) = \mathrm{Id}$ and if $t = B_+(t_1, \dots, t_\mu)$,

$$\mathbf{F}(t) = \sum_{i_1, \dots, i_\mu} \mathbf{F}(t_1)[f^{i_1}] \cdots \mathbf{F}(t_\mu)[f^{i_\mu}] E_{i_1} \cdots E_{i_\mu}.$$

We now give some motivation for the above definition. Formally, one has the expansion of the flow of $F \in \mathfrak{X}(\mathcal{M})$ relative to the point $p \in \mathcal{M}$,

$$\psi(\exp(hF)p) = \psi(p) + F[\psi](p) + \frac{1}{2}F^2[\psi](p) + \cdots = \exp(hF)[\psi](p). \tag{2.17}$$

The powers of F are obtained by repeated application of F as a derivation operator. The expansion (2.17) is sometimes called a *pullback series*.

We may substitute Equation (2.1) ($F = f^i E_i$) into (2.17) and use Leibniz' rule repeatedly to generate terms that are EHODs. We use the convention that indices appearing more than once are summed over, and from [21] we find

$$\begin{aligned}
F^1 &= F = f^i E_i = \mathbf{F}(\bullet) \\
F^2 &= F[F] = f^i E_i[f^j E_j] = f^i E_i[f^j]E_j + f^i f^j E_i[E_j] \\
&= \mathbf{F}(\bullet\!\!\bullet) + \mathbf{F}(\vee) \\
F^3 &= f^i f^j f^k E_i E_j E_k + f^i f^k E_i[f^j]E_j E_k + 2f^i f^j E_i[f^k]E_j E_k + \\
&\quad f^i f^j E_i E_j[f^k]E_k + f^i E_i[f^j]E_j[f^k]E_k \\
&= \mathbf{F}(\wedge\!\bullet) + \mathbf{F}(\curlyvee) + 2\mathbf{F}(\vee\!\!\bullet) + \mathbf{F}(\vee\!\!\bullet) + \mathbf{F}(\bullet\!\!\bullet\!\!\bullet).
\end{aligned} \tag{2.18}$$

Proposition 2.7. *The map* $\mathbf{F}$ *in Definition 2.6 is an algebra homomorphism from the Grossman–Larson algebra to the algebra of EHODs under composition,*

$$\mathbf{F}(\mu_{\mathrm{GL}}(v \otimes w)) = \mathbf{F}(v) \circ \mathbf{F}(w).$$

We first prove the following lemma.

Lemma 2.8. *Let* $v = B_+(t) \in T_O^1,\ t \in T_O$ *and* $w \in T_O$. *Then*

$$\mathbf{F}(v) \circ \mathbf{F}(w) = \mathbf{F}(w \leftharpoonup t)$$

Proof. The proof is by induction on $|w|$. Suppose first that $w = \bullet$, then $\mathbf{F}(v) \circ \mathbf{F}(\bullet) = \mathbf{F}(\bullet \leftharpoonup_{d_\bullet} \{t\}) = \mathbf{F}(v)$. Next suppose that the lemma holds for each w such that $|w| \leq k$ where $k \geq 1$. Any tree with $k+1$ nodes is of the form $w = B_+(w_1, \ldots, w_\omega)$ where each $|w_i| \leq k$. We calculate

$$\mathbf{F}(v) \circ \mathbf{F}(w) = \sum_i \mathbf{F}(t)[f^i]E_i \left[\sum_{\mathbf{j}} \mathbf{F}(w_1)[f^{j_1}] \cdots \mathbf{F}(w_\omega)[f^{j_\omega}]E_{j_1} \cdots E_{j_\omega} \right]$$

We use the Leibniz rule and split the result in two parts, $\mathbf{F}(v) \circ \mathbf{F}(w) = T_1 + T_2$ where

$$T_1 = \mathbf{F}(B_+(t, w_1, \ldots, w_\omega)) = \mathbf{F}(w \leftharpoonup_{d_r} \{t\})$$

where r is the root of w. T_1 is the part where E_i above acts as a derivation on the part $E_{j_1}\cdots E_{j_\omega}$. The second part is where E_i acts on the coefficient functions,

$$\begin{aligned}
T_2 &= \sum_{q=1}^{\omega}\sum_{i,\mathbf{j}} \mathbf{F}(t)[f^i]E_i\left[\mathbf{F}(w_q)[f^{j_q}]\right]\prod_{k\neq q}\mathbf{F}(w_k)[f^{j_k}]E_{j_1}\cdots E_{j_\omega}\\
&= \sum_{q=1}^{\omega}\sum_{\mathbf{j}} \mathbf{F}(v)\circ\mathbf{F}(w_q)[f^{j_q}]\prod_{k\neq q}\mathbf{F}(w_k)[f^{j_k}]E_{j_1}\cdots E_{j_\omega}\\
&= \sum_{\mathbf{j}}\sum_{q=1}^{\omega}\sum_{x_q\in\sigma(w_q)} \mathbf{F}(w_q \leftharpoonup_{d_{x_q}} \{t\})[f^{j_q}]\prod_{k\neq q}\mathbf{F}(w_k)[f^{j_k}]E_{j_1}\cdots E_{j_\omega}\\
&= \sum_{x\in\sigma(w)\setminus\{r\}} \mathbf{F}(w \leftharpoonup_{d_x} \{t\})
\end{aligned}$$

□

Proof of Proposition 2.7. If v is the unit tree $\bullet$, the result is obvious. Suppose that $v = B_+(v_1,\ldots,v_\nu)$, each $v_i \in T_O$, and let y be the independent variable for the EHODs. Then

$$\begin{aligned}
\mathbf{F}(v)\circ\mathbf{F}(w) &= \sum_{\mathbf{i}} \mathbf{F}_s(v_1)[f^{i_1}]\cdots\mathbf{F}_s(v_1)[f^{i_\nu}]\Big|_{s=y} E_{i_1}\cdots E_{i_\nu}[\mathbf{F}(w)]\\
&= \sum_{\mathbf{i}} (\mathbf{F}_s(B_+(v_1))[f^{i_1}]E_{i_1})\cdots(\mathbf{F}_s(B_+(v_\nu))[f^{i_\nu}]E_{i_\nu})[\mathbf{F}(w)]\Big|_{s=y}
\end{aligned}$$

Each of the ν trees $B_+(v_i)$ are of the form $v = B_+(t)$, as required by Lemma 2.8. The trees are all attached only to the nodes of w, there is no accumulation since the coefficient functions of the attached subtrees do not depend on y. We get according to the lemma that

$$\mathbf{F}(v)\circ\mathbf{F}(w) = \sum_{k=1}^{\nu}\sum_{x\in\sigma(w)} \mathbf{F}(w \leftharpoonup_{d_x} \{v_k\}) = \sum_{d} \mathbf{F}(w \leftharpoonup_{d} B_-(v))$$

□

Example 2.9. We use the same trees as in the example in Equation (2.9)

$$\mathbf{F}(\tau)\circ\mathbf{F}(\bullet) = \mathbf{F}(\tau^2 + 2\tau' + \tau^3).$$

The corresponding EHODs composed with each other results in

$$\begin{aligned}
f^i f^j E_i E_j \circ f^k E_k &= f^i f^j E_i E_j[f^k E_k]\\
&= f^i f^j E_i E_j[f^k]E_k + f^i f^j E_j[f^k]E_i E_k +\\
&\quad f^i f^j E_i[f^k]E_j E_k + f^i f^j f^k E_i E_j E_k
\end{aligned}$$

which we see correspond to the correct trees as in Equation (2.18).

Frozen Elementary High Order Derivations

In (2.2) we introduced the freeze map which assigns to a pair (F, p) the vector field $F_p \in \mathrm{span}\{E_1, \ldots, E_d\}$ which coincides with F at p, $F|_p = F_p|_p$. This amounts to freezing the coefficient functions f^i at the point p in the representation $F = f^i E_i$ in terms of the frame.

Definition 2.10 (Frozen Elementary High Order Derivation, FEHOD)**.** Let $F \in \mathfrak{X}(\mathcal{M})$, $p \in \mathcal{M}$ and $E_1, \ldots, E_d \in \mathfrak{X}(\mathcal{M})$ be given. We define $\mathbf{F}_p$ to be the k-linear map from $k_\infty T_O$ to the vector space of high order derivations on the manifold such that for $t \in T_O$, one has: $\mathbf{F}_p(\bullet) = \mathrm{Id}$, and if $t = B_+(t_1, \ldots, t_\mu)$, then

$$\mathbf{F}_p(t) = \sum_{i_1,\ldots,i_\mu} \mathbf{F}(t_1)[f^{i_1}](p) \cdots \mathbf{F}(t_\mu)[f^{i_\mu}](p)\, E_{i_1} \cdots E_{i_\mu}$$

Proposition 2.11. *The map $\mathbf{F}_p$ is an algebra homomorphism from the concatenation algebra to the algebra of FEHODs under composition,*

$$\mathbf{F}_p(\mu_{\mathrm{M}}(v \otimes w)) = \mathbf{F}_p(v) \circ \mathbf{F}_p(w).$$

Proof. Let $v = B_+(t_1, \ldots, t_\mu)$ and $w = B_+(t_{\mu+1}, \ldots, t_\nu)$, then

$$\begin{aligned}
\mathbf{F}_p(v) \circ \mathbf{F}_p(w) &= \sum_{i_1,\ldots,i_\mu} \mathbf{F}(t_1)[f^{i_1}](p) \cdots \mathbf{F}(t_\mu)[f^{i_\mu}](p)\, E_{i_1} \cdots E_{i_\mu} \circ \\
&\qquad \left(\sum_{i_{\mu+1},\ldots,i_\nu} \mathbf{F}(t_{\mu+1})[f^{i_{\mu+1}}](p) \cdots \mathbf{F}(t_\nu)[f^{i_\nu}](p)\, E_{i_{\mu+1}} \cdots E_{i_\nu} \right) \\
&= \sum_{i_1,\ldots,i_\nu} \mathbf{F}(t_1)[f^{i_1}](p) \cdots \mathbf{F}(t_\nu)[f^{i_\nu}](p)\, E_{i_1} \cdots E_{i_\nu} \\
&= \mathbf{F}_p(\mu_{\mathrm{M}}(v \otimes w))
\end{aligned}$$

as each of the $\mathbf{F}(t_j)[f^{i_j}](p)$ is just a constant and thereby unaffected by the E_i's. □

2.5 *B*-series

Let kT_O^* be the algebraic dual of the space kT_O. For any $\mathbf{a} \in kT_O^*, p \in \mathcal{M}$ we associate a formal series of operators

$$B_p(\mathbf{a}) = \sum_{t \in T_O} h^{|t|-1} \langle \mathbf{a}, t \rangle\, \mathbf{F}_p(t) \tag{2.19}$$

where $\langle \cdot, \cdot \rangle$ is the duality pairing. We can think of (2.19) as a generalization of the B-series discussed in [9]. Some authors include symmetry coefficients $\sigma(t)$, and then take $\langle \mathbf{a}', t \rangle = \langle \mathbf{a}, t \rangle / \sigma(t)$ instead of $\langle \mathbf{a}, t \rangle$ in the definition.

Keeping the algebra homomorphism $\mathbf{F}_p$ from Definition 2.10 in mind, we employ the shorthand notation

$$\sum_{t \in T_O} \langle \mathbf{a}, t \rangle t \tag{2.20}$$

for the B-series (2.19) and this is now a series in $k_\infty T_O$. This formula also shows the natural identification of kT_O^* with $k_\infty T_O$.

Suppose that a map $\phi_{\mathbf{a}}$ has a B-series expansion with coefficients from $\mathbf{a} \in kT_O^*$ and relative to the point $p \in \mathcal{M}$. This means that formally

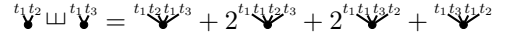
$$\psi(\phi_{\mathbf{a}}(y)) = B_p(\mathbf{a})[\psi](y) \tag{2.21}$$

for any function $\psi \in C^\infty(\mathcal{M}, k)$ and $y \in \mathcal{M}$. Suppose that we freeze the vector field F at the point $\phi_{\mathbf{a}}(p) \in \mathcal{M}$, we then compute the series of $hF_{\phi_{\mathbf{a}}(p)}$ using (2.21)

$$\begin{aligned} h \sum_i f^i(\phi_{\mathbf{a}}(p)) E_i &= h \sum_i \sum_{t \in T_O} h^{|t|-1} \langle \mathbf{a}, t \rangle \, \mathbf{F}_p(t)[f^i](p) E_i \\ &= h \sum_{t \in T_O} h^{|t|-1} \langle \mathbf{a}, t \rangle \mathbf{F}_p(B_+(t)) = \sum_{t \in T_O^1} h^{|t|-1} \langle \mathbf{a}, B_-(t) \rangle \mathbf{F}_p(t). \end{aligned}$$

In other words, the B-series of a frozen vector field is associated to the space $k_\infty T_O^1$. In Section 2.3 we defined the Lie algebra $\mathfrak{g} \subset k_\infty T_O$, that is, those $S \in \mathfrak{g}$ which satisfy $\Delta(S) = S \otimes \bullet + \bullet \otimes S$. This Lie algebra contains $k_\infty T_O^1$. We may therefore conclude that the commuted vector fields $K_i = P_i(\bar{K}_1, \ldots, \bar{K}_s)$ for each of the Lie polynomials P_i in (2.5) belongs to $\mathfrak{g}$. Now, the quantities g_i (2.3) and y_1 (2.6) are, owing to the Baker–Campbell–Hausdorff (BCH) formula, exponentials of elements in $\mathfrak{g}$ and therefore belong to the group G. Note that the BCH formula can also be used in a formal way here so that no convergence criterion is needed.

The fact that series in $\mathfrak{g}$ satisfy (2.14) imposes restrictions on the dual elements $\mathbf{a}$ which represent such series. From [22, Theorem 3.1] we find that $\langle \mathbf{a}, v \sqcup\!\sqcup w \rangle = 0$ for all $v, w \in T_O \backslash \{\bullet\}$ where $v \sqcup\!\sqcup w$ is the shuffle product denoting the sum of all possible ordered insertions of w into v on the first subtree level [22, p. 23-24]. For instance, the shuffle product of the two trees $B_+(t_1, t_2)$ and $B_+(t_1, t_3)$ is



where $t_1, t_2, t_3 \in T_O$.

Similarly, (2.15) can now be used to characterize the coefficient forms in kT_O^* of a series which belongs to G. Suppose that such a $\mathbf{b} \in kT_O^*$ is representing a series in G. We find that $\langle \mathbf{b}, \bullet \rangle = 1$ and that $\langle \mathbf{b}, v \sqcup\!\sqcup w \rangle = \langle \mathbf{b}, v \rangle \langle \mathbf{b}, w \rangle$, which is

called a shuffle relation. These relations were derived in a different way in [21], see also [24].

From [21] we find that the exact flow $\psi(\exp(hF)y)$ of the differential equation $\dot{y} = F(y)$ can be expressed in a B-series $B_p(\mathbf{a})$ where $\langle \mathbf{a}, t\rangle = \alpha(t)/(|t|-1)!$ and where $\alpha(t)$ is defined recursively as

$$\alpha(\bullet) = 1 \qquad \text{and} \qquad \alpha(B_+(t_1,\ldots,t_\mu)) = \prod_{\ell=1}^{\mu} \binom{\sum_{i=1}^{\ell}|t_i|-1}{|t_\ell|-1}\alpha(t_\ell). \tag{2.22}$$

The linear form $\mathbf{a} \in kT_O^*$ obeys the shuffle relation $(\mathbf{a}, v \sqcup\!\sqcup w) = (\mathbf{a}, v)(\mathbf{a}, w)$.

As argued above, the numerical integration schemes presented in Section 2.2 admit a B-series expansion as well. Thus, for each such scheme, there exists a $\mathbf{b} \in kT_O^*$ with which (2.19) holds formally. In [20] one can find the details of how the B-series is obtained for commutator-free schemes.

Given two mappings on a manifold, say $\phi_{\mathbf{a}}, \phi_{\mathbf{b}} \colon \mathcal{M} \to \mathcal{M}$ with corresponding B-series $B(\mathbf{a})$ and $B(\mathbf{b})$ in $k_\infty T_O$. The composition $\phi_{\mathbf{c}} = \phi_{\mathbf{b}} \circ \phi_{\mathbf{a}}$ also has a B-series, with coefficients from $\mathbf{c}$. By applying (2.21) twice we get that $B_p(\mathbf{c}) = B_p(\mathbf{a}) \circ B_p(\mathbf{b})$ (note the usual reversal of order, passing from composition of mappings to composition of operators). The concatenation product on $k_\infty T_O$ now yields

$$\begin{aligned}\mu_{\mathrm{M}}\left(\sum_{v\in T_O}\langle \mathbf{a}, v\rangle\, v \otimes \sum_{w\in T_O}\langle \mathbf{b}, w\rangle\, w\right) &= \sum_{v,w\in T_O}\langle \mathbf{a}, v\rangle\,\langle \mathbf{b}, w\rangle\,\mu_{\mathrm{M}}(v\otimes w)\\ &= \sum_{t\in T_O}\sum_{\mu_{\mathrm{M}}(v\otimes w)=t}\langle \mathbf{a}, v\rangle\,\langle \mathbf{b}, w\rangle\, t = \sum_{t\in T_O}\langle \mathbf{c}, t\rangle\, t\end{aligned} \tag{2.23}$$

The resulting B-series has coefficients $\langle \mathbf{c}, \bullet\rangle = 1$ and for $t = B_+(t_1,\ldots,t_\mu)$,

$$\langle \mathbf{c}, t\rangle = \sum_{k=0}^{\mu}\langle \mathbf{a}, B_+(t_1,\ldots,t_k)\rangle\,\langle \mathbf{b}, B_+(t_{k+1},\ldots,t_\mu)\rangle \tag{2.24}$$

In view of the identification of $k_\infty T_O$ with kT_O^*, we can think of μ_{M} as a product on kT_O^* and simply write

$$\mathbf{c} = \mu_{\mathrm{M}}(\mathbf{a}\otimes\mathbf{b}).$$

Taking the adjoint operator $\Delta_{\mathrm{M}} := \mu_{\mathrm{M}}^* : kT_O \to kT_O \otimes kT_O$ (using the usual identification of $(kT_O^* \otimes kT_O^*)^*$ with $kT_O \otimes kT_O$) we obtain

$$\Delta_{\mathrm{M}}(B_+(t_1,\ldots,t_\mu)) = \sum_{k=0}^{\mu} B_+(t_1,\ldots,t_k)\otimes B_+(t_{k+1},\ldots,t_\mu),$$

and $\langle \mathbf{c}, t\rangle = \langle \mu_{\mathrm{M}}(\mathbf{a}\otimes\mathbf{b}), t\rangle = \langle \mathbf{a}\otimes\mathbf{b}, \Delta_{\mathrm{M}}(t)\rangle$ which is precisely what (2.24) says.

The antipode in the concatenation algebra $\mathcal{S}_{\mathrm{M}}$ has an immediate application to the B-series $B_p(\mathbf{a})$ of maps $\phi_{\mathbf{a}}$ on the manifold. The antipode of a series $T \in G = \exp(\mathfrak{g})$ is the corresponding series of the inverse map $\phi_{\mathbf{a}}^{-1}$. This is evident from Equation (2.16) and from the fact that the B-series for the identity map on $\mathcal{M}$ has its coefficients from the counit $\epsilon \in kT_O^*$ where $\langle \epsilon, \bullet\rangle = 1$, $\langle \epsilon, t\rangle = 0$ for all other trees $t \neq \bullet$.

2.6 Applications

Order conditions for integration schemes

The paper [20] presents the order conditions for a subclass of the schemes introduced in Section 2.2. We will show here how one can use the algebra on trees to count the order conditions for any order of the numerical method.

The algebra kT_O is graded by $\nu(t) = |t|-1$ as in the introduction to Section 2.3. We can decompose the algebra kT_O and the Lie algebra $\mathfrak{g}$ of primitive elements in kT_O into their respective graded components

$$kT_O = \sum_{k=0}^{\infty} \mathcal{B}_k \quad \text{and} \quad \mathfrak{g} = \sum_{k=0}^{\infty} \mathfrak{g}_k$$

The dimensions of each graded component in $\mathfrak{g}$ up to grade q are added to yield the number of order conditions for a qth-order integration scheme. From (2.7) we have $\dim \mathcal{B}_k = \nu_k = \frac{1}{k+1}\binom{2k}{k}$. It is well known [25, Theorem 3.2.8] that

$$kT_O = \mathrm{UEA}(\mathfrak{g})$$

where $\mathrm{UEA}(\mathfrak{g})$ is the universal enveloping algebra of $\mathfrak{g}$. Comparing generating functions for the dimensions of kT_O and $\mathrm{UEA}(\mathfrak{g})$ we get

$$\prod_{n=1}^{\infty} (1-T^n)^{-\dim \mathfrak{g}_n} = \sum_{k=0}^{\infty} \nu_k T^k$$

When solved for $\dim \mathfrak{g}_k$ one obtains

$$\dim \mathfrak{g}_k = \frac{1}{2k} \sum_{d|k} \mu(k) \binom{2k/d}{k/d} \tag{2.25}$$

where $\mu(k)$ is the Möbius function defined for any positive integer as $\mu(1) = 1$, $\mu(k) = (-1)^p$ when k is the product of p distinct primes, and $\mu(k) = 0$ otherwise. The sum is over all positive integers d which divide k, including 1 and k.

The number of order conditions of an integration scheme of order of consistency q is then

$$N_q = \sum_{k=1}^{q} \dim \mathfrak{g}_k.$$

and the first ten numbers are

$$1, 1, 3, 8, 25, 75, 245, 800, 2700, 9225.$$

The formula (2.25) is well-known in the literature. For example, it counts the number of balanced Lyndon words [18] and also has an application to double bracket flows [11].

Backward error analysis

Suppose a numerical method applied to the differential equation

$$y' = F(y) \tag{2.26}$$

is the map $\phi_{h,F} \colon \mathcal{M} \to \mathcal{M}$. If there exists an h-dependent vector field $\tilde{F}$ such that

$$\phi_{h,F} = \exp(h\tilde{F}),$$

then the numerical method for Equation (2.26) solves the differential equation $y' = \tilde{F}(y)$ exactly. We call $\tilde{F}$ the *modified vector field* for $\phi_{h,F}$.

One may follow for example Hairer, Lubich and Wanner [8] and formally expand the modified vector field in powers of h

$$\tilde{F} = F + hF_2 + h^2F_3 + \cdots . \tag{2.27}$$

The numerical method $\phi_{h,F}$ as a mapping on the manifold has an expansion in a B-series, say $B(\mathbf{a})$. Then we require that our modified vector field $\tilde{F}$ must obey

$$\psi(\phi_{h,F}(p)) = \psi(\exp(h\tilde{F})p) = B(\mathbf{a})[\psi](p).$$

where $B(\mathbf{a})$ is the series defined in (2.19).

We calculate $\exp(h\tilde{F})$ according to the expansion in (2.27) and get

$$\exp(h\tilde{F}) = I + hF + h^2(F_2 + \tfrac{1}{2}F^2) + h^3(F_3 + \tfrac{1}{2}(FF_2 + F_2F) + \tfrac{1}{6}F^3) + \cdots$$

To calculate F_2 we compare coefficients of h^2 in the equation above and in $B(\mathbf{a})$. We get

$$\mathbf{a}(\bullet\!\!-\!\!\bullet\!\!-\!\!\bullet)\mathbf{F}(\bullet\!\!-\!\!\bullet\!\!-\!\!\bullet) + \mathbf{a}(\begin{smallmatrix}\bullet & \bullet\\ & \diagdown\!\diagup\\ & \bullet\end{smallmatrix})\mathbf{F}(\begin{smallmatrix}\bullet & \bullet\\ & \diagdown\!\diagup\\ & \bullet\end{smallmatrix}) = F_2 + \tfrac{1}{2}F^2 = F_2 + \tfrac{1}{2}\mathbf{F}(\bullet\!\!-\!\!\bullet\!\!-\!\!\bullet)^2$$
$$= F_2 + \frac{1}{2}\left(\mathbf{F}(\bullet\!\!-\!\!\bullet\!\!-\!\!\bullet) + \mathbf{F}(\begin{smallmatrix}\bullet & \bullet\\ & \diagdown\!\diagup\\ & \bullet\end{smallmatrix})\right)$$

where we have used consistency of the method, $F = \mathbf{F}(\bullet)$ and the Grossman–Larson product of $\bullet$ and $\bullet$. Consistency of the numerical method requires $\mathbf{a}(\bullet) = 1$, and by a shuffle relation we also have that $\mathbf{a}(\bullet\!\!-\!\!\bullet) = \frac{1}{2}\mathbf{a}(\bullet)^2 = \frac{1}{2}$. In the end we solve for F_2 and get

$$F_2 = \left(\mathbf{a}(\bullet\!\!-\!\!\bullet) - \frac{1}{2}\right)\mathbf{F}(\bullet\!\!-\!\!\bullet) \tag{2.28}$$

The same approach will for the h^3 coefficients lead to

$$\begin{aligned} F_3 = &\left(\mathbf{a}(\bullet\!\!-\!\!\bullet\!\!-\!\!\bullet) - \frac{1}{2}\mathbf{a}(\bullet\!\!-\!\!\bullet) + \frac{1}{12}\right)\mathbf{F}(\bullet\!\!-\!\!\bullet\!\!-\!\!\bullet) + \left(\mathbf{a}(\bullet\!\!-\!\!\bullet\!\!-\!\!\bullet) - \frac{1}{2}\mathbf{a}(\bullet\!\!-\!\!\bullet) - \frac{1}{12}\right)\mathbf{F}(\bullet\!\!-\!\!\bullet\!\!-\!\!\bullet) \\ &+ \left(\mathbf{a}(\bullet\!\!-\!\!\bullet\!\!-\!\!\bullet) - \frac{1}{2}\mathbf{a}(\bullet\!\!-\!\!\bullet) + \frac{1}{12}\right)\mathbf{F}(\bullet\!\!-\!\!\bullet\!\!-\!\!\bullet) + \left(\mathbf{a}(\bullet\!\!-\!\!\bullet) - \mathbf{a}(\bullet\!\!-\!\!\bullet) + \frac{1}{3}\right)\mathbf{F}(\bullet\!\!-\!\!\bullet) \end{aligned} \tag{2.29}$$

Again, putting this in a more general perspective, let $\epsilon + \mathbf{a} \in kT_O^*$ represent the B-series of the numerical method, where ϵ is the counit in the Grossman–Larson Hopf algebra, and $\langle \mathbf{a}, \bullet \rangle = 0$. We now find that the modified vector field $\tilde{F}$ has a B-series in $\mathfrak{g}$ associated to $\mathbf{b} \in kT_O^*$ such that

$$\mathbf{b} = \log_{\mathrm{GL}}(\epsilon + \mathbf{a}) = \mathbf{a} - \frac{1}{2}\mu_{\mathrm{GL}}(\mathbf{a} \otimes \mathbf{a}) + \cdots$$

Here, μ_{GL} is induced on kT_O^* from its identification with $k_\infty T_O$.

The expression for the vector fields F_k are obtained by projecting $\mathbf{b}$ onto the kth graded component of $k_\infty T_O$. One obtains that the series associated to $\mathbf{b}$ belongs to $\mathfrak{g}$.

2.7 Conclusion

We have discussed Hopf algebra structures on the space of ordered rooted trees, and shown how these structures are related to the expansions of Lie group integration schemes for ordinary differential equations on manifolds. The theory presented here is fairly general, in the sense that it accounts for most of the known Lie group integrators which are based on exponentials. It also allows for the analysis of schemes that are hybrids between for instance the RKMK methods [17] and the commutator-free schemes [4]. Although we have emphasized the algebraic aspects of the theory, we believe that there is a potential for using these aspects in developing integration schemes with high accuracy and good long-time behaviour.

Bibliography

[1] Brouder, C.: Runge–Kutta methods and renormalization. Eur. Phys. J. C **12**, 521–534 (2000)

[2] Butcher, J.C.: Coefficients for the study of Runge–Kutta integration processes. J. Austral. Math. Soc. **3**(185–201) (1963)

[3] Butcher, J.C.: An algebraic theory of integration methods. Math. Comp. **26**, 79–106 (1972)

[4] Celledoni, E., Marthinsen, A., Owren, B.: Commutator-free Lie group methods. FGCS **19**(3), 341–352 (2003)

[5] Cohen, D.I.A.: Basic Techniques of Combinatorial Theory. John Wiley & Sons (1978)

[6] Crouch, P.E., Grossman, R.: Numerical integration of ordinary differential equations on manifolds. J. Nonlinear Sci. **3**, 1–33 (1993)

[7] Grossman, R., Larson, R.G.: Hopf-algebraic structure of families of trees. J. Algebra **126**(1), 184–210 (1989)

[8] Hairer, E., Lubich, C., Wanner, G.: Geometric Numerical Integration, *Springer Series in Computational Mathematics*, vol. 31. Springer-Verlag, Berlin (2002). Structure-preserving algorithms for ordinary differential equations

[9] Hairer, E., Nørsett, S., Wanner, G.: Solving Ordinary Differential Equations I, Nonstiff Problems. Springer, Berlin (1987)

[10] Hairer, E., Wanner, G.: On the Butcher group and general multivalue methods. Computing **13**, 1–15 (1974)

[11] Iserles, A.: On the discretization of double-bracket flows. Found. Comput. Math. **2**, 305–329 (2002)

[12] Iserles, A., Munthe-Kaas, H.Z., Nørsett, S.P., Zanna, A.: Lie-group methods. Acta Numerica **9**, 215–365 (2000)

[13] Kreimer, D.: On the Hopf algebra structure of perturbative quantum field theories. Adv. Theor. Math. Phys. **2**, 303–334 (1998)

[14] Lewis, D., Simo, J.C.: Conserving algorithms for the dynamics of Hamiltonian systems of Lie groups. J. Nonlinear Sci. **4**, 253–299 (1994)

[15] Lewis, D., Simo, J.C.: Conserving Algorithms for the N Dimensional Rigid Body. In: J.E. Mardsen, G.W. Patrick, W.F. Shadwick (eds.) Integration Algorithms and Classical Mechanics, pp. 121–140. American Mathematical Society (1996)

[16] Munthe-Kaas, H.: Runge–Kutta methods on Lie groups. BIT **38**(1), 92–111 (1998)

[17] Munthe-Kaas, H.: High order Runge–Kutta methods on manifolds. In: Proceedings of the NSF/CBMS Regional Conference on Numerical Analysis of Hamiltonian Differential Equations (Golden, CO, 1997), vol. 29,1, pp. 115–127 (1999)

[18] Munthe-Kaas, H., Krogstad, S.: On enumeration problems in Lie–Butcher theory. FGCS **19**(7), 1197–1205 (2003)

[19] Murua, A., Sanz-Serna, J.M.: Order conditions for numerical integrators obtained by composing simpler integrators. R. Soc. Lond. Philos. Trans. Ser. A Math. Phys. Eng. Sci. **357**(1754), 1079–1100 (1999)

[20] Owren, B.: Order conditions for commutator-free Lie group methods. Tech. Rep. 7/05, Department of Mathematical Sciences, NTNU, Norway (2005). `http://www.math.ntnu.no/preprint/`

[21] Owren, B., Marthinsen, A.: Runge–Kutta methods adapted to manifolds and based on rigid frames. BIT **39**(1), 116–142 (1999)

[22] Reutenauer, C.: Free Lie Algebras. Oxford University Press (1993)

[23] Sweedler, M.: Hopf Algebras. W. A. Benjamin, Inc. (1969)

[24] Tracogna, S., Welfert, B.: Notes on riffle shuffles (1998). Available upon request

[25] Varadarajan, V.S.: Lie Groups, Lie Algebras, and Their Representations. GTM 102. Springer-Verlag (1984)

Paper II

Håvard Berland, Brynjulf Owren, and Bård Skaflestad:

B-series and order conditions for exponential integrators

Published in SIAM Journal of Numerical Analysis, 43(4):1715–1727, 2005.

Chapter 3

B-series and order conditions for exponential integrators

Abstract:

We introduce a general format of numerical ODE-solvers which include many of the recently proposed exponential integrators. We derive a general order theory for these schemes in terms of *B*-series and bicolored rooted trees. To ease the construction of specific schemes we generalize an idea of Zennaro and define Natural Continuous Extensions in the context of exponential integrators. This leads to a relatively easy derivation of some of the most popular recently proposed schemes. The general format of schemes considered here makes use of coefficient functions which will usually be selected from some finite dimensional function spaces. We will derive lower bounds for the dimension of these spaces in terms of the order of the resulting schemes. Finally we illustrate the presented ideas by giving examples of new exponential integrators of orders 4 and 5.

3.1 Introduction

Numerical integration schemes which use the matrix exponential go back all the way to Certaine [4], but there are also early papers by Lawson [15], Nørsett [20], Ehle and Lawson [6], and Friedli [7] to mention just a few. Recently there has been a revived interest in these schemes, in particular for the solution of nonlinear partial differential equations, see for instance [11, 17, 5, 3, 14, 13]. For a thorough review of the history of exponential integrators, see [16] and the references therein. The integrators found in these papers are derived in rather different ways, and they are formulated for different types of systems of differential equations. In this note, we consider the autonomous nonlinear system of ordinary differential equations

$$\dot{u} = Lu + N(u), \qquad u(0) = u_0. \tag{3.1}$$

Here L is a matrix and $N(u)$ a nonlinear mapping. The order theory we consider is valid for a large class of exponential integrators, including the Runge–Kutta–Munthe-Kaas (RKMK) schemes [17], the commutator-free Lie group integrators [3], and those schemes of Cox and Matthews [5] as well as Krogstad [14] which reduce to classical Runge–Kutta schemes when $L = 0$.

We present the general format for integrators of (3.1) as

$$N_r = N\big(\exp(c_r hL)\, u_0 + h \sum_{j=1}^{s} a_r^j(hL)\, N_j\big), \quad r = 1, \dots, s \tag{3.2}$$

$$u_1 = \exp(hL)\, u_0 + h \sum_{r=1}^{s} b^r(hL)\, N_r. \tag{3.3}$$

Here we assume that the functions $a_r^j(z)$ and $b^r(z)$ are at least p times continuously differentiable at $z = 0$ for integration schemes of order p.

0				
$\frac{1}{2}$	$\frac{1}{2}\phi_0(z/2)$			
$\frac{1}{2}$	$\frac{z}{8}\phi_0(z/2)$	$\frac{1}{2}(1-\frac{z}{4})\phi_0(z/2)$		
1			$\phi_0(z)$	
	$\frac{1}{6}\phi_0(z)(1+\frac{z}{2})$	$\frac{1}{3}\phi_0(z)$	$\frac{1}{3}\phi_0(z)$	$\frac{1}{6}\phi_0(z)(1-\frac{z}{2})$

(a) RKMK, order 4

0				
$\frac{1}{2}$	$\frac{1}{2}\phi_0(z/2)$			
$\frac{1}{2}$		$\frac{1}{2}\phi_0(z/2)$		
1	$\frac{z}{4}\phi_0(z/2)^2$		$\phi_0(z/2)$	
	$\frac{1}{2}\phi_0(z) - \frac{1}{3}\phi_0(z/2)$	$\frac{1}{3}\phi_0(z)$	$\frac{1}{3}\phi_0(z)$	$-\frac{1}{6}\phi_0(z) + \frac{1}{3}\phi_0(z/2)$

(b) Commutator-free, order 4

Table 3.1: Examples of schemes in general format for exponential integrators

Table 3.1 gives the coefficient functions $a_r^j(z)$ and $b^r(z)$ for the fourth order RKMK scheme introduced in [18] in this general format when applied to the problem (3.1) with an affine Lie group action, and the commutator-free scheme of order 4 from [3]. In both tables $\phi_0(z) = (\mathrm{e}^z - 1)/z$.

For deriving order conditions, we expand the coefficient functions in powers of z,

$$a_r^j(z) = \sum_{k \geq 0} \alpha_r^{j,k} z^k \quad \text{and} \quad b^r(z) = \sum_{k \geq 0} \beta^{r,k} z^k$$

where the sum may terminate with a remainder term. For the schemes we consider here, these functions are in fact all entire. If $N(u) = 0$ in (3.1), then any scheme in the above class will reproduce the exact solution in every step. Whereas if $L = 0$, the scheme (3.2)–(3.3) reduces to a classical Runge–Kutta method with coefficients $a_r^j = \alpha_r^{j,0}$ and $b^r = \beta^{r,0}$. This scheme is henceforth called *the underlying Runge–Kutta scheme*. We will always assume that $c_r = \sum_j \alpha_r^{j,0},\ 1 \le r \le s$.

The schemes proposed by Friedli [7] closely resemble the format (3.2)–(3.3), the difference being that the coefficient functions a_{rj} (resp b_r) are evaluated in $c_r hL$ rather than in hL, thus a non-trivial discrepancy may occur whenever $c_r = 0$. And even though the author of this paper explicitly requires that the functions $a_{rj}(z)$ and $b_r(z)$ be of the form

$$\int_0^1 \mathrm{e}^{(1-\theta)z} p(\theta)\,\mathrm{d}\theta, \qquad p(\theta) \text{ polynomial}$$

his analysis holds also for the case of more general coefficient functions, so that the order conditions he obtains for $p \le 4$ are almost identical to those derived in Section 2 here. However the order theory presented here is general.

We will discuss conditions on the coefficients $\alpha_r^{j,k}$ and $\beta^{r,k}$ under which the scheme (3.2)–(3.3) has order of consistency p for problems of the type (3.1). We will use the well known approach involving rooted trees, see for instance [9, 2]. The conditions we find will only depend on the first $\alpha_r^{j,k}$ for $k \le p-2$ and on $\beta^{j,k}$ for $k \le p-1$. In this note we will not address issues related to the behavior of the coefficient functions $a_r^j(z)$ and $b^r(z)$ for large values of z.

In the recent paper [12] an order theory for explicit exponential integrators is presented and its application to semilinear parabolic problems is discussed. While classical or nonstiff order conditions are usually derived by assuming that a Lipschitz constant exists, one needs to account for the unboundedness of the operator L whenever PDEs are considered. It is found that a set of additional order conditions must be satisfied to guarantee convergence order p under suitable assumptions; one requires the linear operator L to be the infinitesimal generator of an analytic semigroup, and that the nonlinear function satisfies a Lipschitz condition. The authors are also able to give an example where order reduction is seen numerically for schemes not satisfying the additional conditions. But the conditions are rather restrictive, and in [13] exponential integrators of (nonstiff) order four are tried out numerically for a number of well-known semilinear PDEs, and no order reduction is seen, despite the fact that these integrators do not satisfy all the required conditions for order four as given in [12]. This shows that the issue of determining the order behavior of exponential integrators for PDEs is indeed a subtle one, and remains today in an unsatisfactory state of nonresolution.

3.2 *B*-series and order conditions

Repeated differentiation of (3.1) with respect to time yields

$$\begin{aligned}
\frac{\mathrm{d}^2 u}{\mathrm{d}t^2} &= L\dot{u} + N'(\dot{u}) \\
&= L^2 u + LN + N'(Lu) + N'(N) \\
\frac{\mathrm{d}^3 u}{\mathrm{d}t^3} &= L^3 u + L^2 N + LN'(Lu) + LN'(N) \\
&\quad + N''(Lu, Lu) + 2N''(Lu, N) + N'(L^2 u) \\
&\quad + N'(LN) + N''(N, N) + N'N'(Lu) + N'N'(N)
\end{aligned}$$

etc. The exact solution of (3.1) has a formal expansion

$$u(h) = \sum_{q=0}^{\infty} \frac{h^q}{q!} \left.\frac{\mathrm{d}^q}{\mathrm{d}h^q}\right|_{h=0} u(h)$$

where each term in the qth derivative corresponds in an obvious way to a rooted bicolored tree. Let for instance $\bullet \sim F(\bullet) = N(u)$ and $\circ \sim F(\circ) = Lu$ be the two trees with one node. Next define B_+ as the operation which takes a finite set of trees $\{\tau_1, \ldots, \tau_\mu\}$ and connects their roots to a new common black root. Similarly, $\tau = W_+(\tau')$ connects the root of τ' to a new white root resulting in the tree τ associated to $F(\tau) = L \cdot F(\tau')$. It suffices here to allow W_+ to act on a single tree and not on a set of trees. To each tree τ with q nodes formed this way, there exists precisely one term, $F(\tau)$ called an elementary differential, in the qth derivative of the solution of (3.1). For $q > 1$ it is defined recursively as

$$F(B_+(\tau_1, \ldots, \tau_\mu))(u) = N^{(\mu)}(F(\tau_1), \ldots, F(\tau_\mu))(u) \tag{3.4}$$

$$F(W_+(\tau'))(u) = LF(\tau')(u) \tag{3.5}$$

We may denote by T the set of all bicolored trees such that each white node has at most one child, and set $T = T_b \cup T_w$ the union of trees with black and white roots respectively. Introducing the empty set $\emptyset$, and using the convention $B_+(\emptyset) = \bullet$, $W_+(\emptyset) = \circ$, we may write

$$T \cup \emptyset = \bigcup_{m \geq 0} W_+^m(T_b \cup \emptyset), \quad T_w = \bigcup_{m \geq 1} W_+^m(T_b \cup \emptyset) \tag{3.6}$$

The same bicolored trees used here do also appear in the linearly implicit W-methods, see Steihaug and Wolfbrandt [21] as well as the text [10] by Hairer and Wanner. Following for instance the text by Hairer, Lubich, and Wanner [8], we

may work with formal B-series. For an arbitrary map $\boldsymbol{c} : T \cup \emptyset \to \mathbf{R}$, we let the formal series

$$B(\boldsymbol{c}, u) = \boldsymbol{c}(\emptyset)u + \sum_{\tau \in T} \frac{h^{|\tau|}}{\sigma(\tau)} \boldsymbol{c}(\tau) F(\tau)(u) \tag{3.7}$$

be a B-series, where $\sigma(\tau)$ is the symmetry coefficient defined as $\sigma(\bullet) = \sigma(\circ) = 1$, and for $\tau = B_+(\tau_1, \ldots, \tau_\mu)$,

$$\sigma(\tau) = \sigma(\tau_1) \cdots \sigma(\tau_\mu)\, m_1! \cdot m_2! \cdots$$

where the m_i's count the number of equal trees among $\tau_1, \ldots, \tau_\mu$.

The further derivation of order conditions is based on the assumption that both the exact and numerical solution possess B-series of the form (3.7), say $B(\boldsymbol{e}, u_0)$ and $B(\boldsymbol{u}_1, u_0)$ respectively. We refer to [1] for details, and present only the final result.

Theorem 3.1. *Let $T' \subset T$ be the set of bicolored rooted trees such that every white node has precisely one child. An exponential integrator defined by* (3.2)–(3.3) *has order of consistency p if*

$$\boldsymbol{u}_1(\tau) = \frac{1}{\gamma(\tau)}, \quad \textit{for all } \tau \in T' \textit{ such that } |\tau| \le p,$$

where

$$\begin{aligned}
\boldsymbol{u}_1(\emptyset) = \boldsymbol{U}_r(\emptyset) &= 1, \quad 1 \le r \le s, \\
\boldsymbol{u}_1(W_+^m B_+(\tau_1, \ldots, \tau_\mu)) &= \sum_{r=1}^{s} \beta^{r,m} \boldsymbol{U}_r(\tau_1) \ldots \boldsymbol{U}_r(\tau_\mu) \\
\boldsymbol{U}_r(W_+^m B_+(\tau_1, \ldots, \tau_\mu)) &= \sum_{j=1}^{s} \alpha_r^{j,m} \boldsymbol{U}_j(\tau_1) \ldots \boldsymbol{U}_j(\tau_\mu)
\end{aligned}$$

Note that even though all trees in the set T feature in the B-series for the exact and numerical solutions, it suffices to consider a subset T' consisting of all trees in T except those with a terminal white node. There is an interesting connection between the set of trees T' and the trees used to develop the order theory for composition methods in [19]. White nodes appear as connected strings of nodes which, except from the root, have exactly one parent and one child, and always terminating in a black node. Therefore one can remove all white nodes and assign to the terminating black node the number of removed nodes plus one. Black nodes not connected to a white node is assigned the number one. These multilabelled trees are precisely those appearing in [19], they can also be

identified as the set of rooted trees of nonempty sets. The generating function for these trees is well-known,

$$M(x) = \frac{x}{1-x}\ \exp\left(M(x) + \frac{M(x^2)}{2} + \frac{M(x^3)}{3} + \cdots\right).$$

The number of order conditions for each order 1 to 9 is 1, 2, 5, 13, 37, 108, 332, 1042, 3360.

3.3 Construction of exponential integrators

The schemes of Lawson [15] are exponential integrators derived simply by introducing a change of variable, $w(t) = \mathrm{e}^{-tL}\, u(t)$ in (3.1), and by applying a standard Runge–Kutta scheme to the resulting ODE. This approach results in a formula for w_1 in terms of w_0. By setting $u_n = \mathrm{e}^{tL}\, w_n$ one gets a scheme of the form (3.2)–(3.3) in which

$$a_r^j(z) = \alpha_r^{j,0}\, \mathrm{e}^{(c_r - c_j)z} \quad \text{and} \quad b^r(z) = \beta^{r,0}\, \mathrm{e}^{(1-c_r)z}.$$

as noted by Lawson in [15].

This scheme has order p if the underlying scheme determined by $\alpha_r^{j,0}$ and $\beta^{r,0}$ is of order p. This gives us a very useful tool for constructing exponential integrators with given underlying Runge–Kutta schemes. We express this in a proposition.

Proposition 3.2. *Suppose that the coefficients $\alpha_r^{j,0}$ and $\beta^{r,0}$, $1 \le r, j \le s$ define a Runge–Kutta scheme of order p. Then, any exponential integrator of the form* (3.2)–(3.3) *satisfying*

$$\alpha_r^{j,m} = \frac{1}{m!}(a_r^j)^{(m)}(0) = \frac{1}{m!}\alpha_r^{j,0}(c_r - c_j)^m, \quad 0 \le m \le p-2, \tag{3.8}$$

$$\beta^{r,m} = \frac{1}{m!}(b_r^j)^{(m)}(0) = \frac{1}{m!}\beta^{r,0}(1 - c_r)^m, \quad 0 \le m \le p-1, \tag{3.9}$$

is of order p. In the above expression we use $0^0 := 1$.

Proof. Order conditions for exponential integrators of order p involve $\alpha_r^{j,m}$ for $0 \le m \le p-2$ and $\beta^{r,m}$ for $0 \le m \le p-1$. On the other hand, the Lawson schemes must satisfy the order conditions for exponential integrators, and their values for these coefficients a precisely those specified in the proposition. □

It is convenient to introduce finite dimensional function spaces V_a and V_b to which the respective coefficient functions $a_r^j(z)$ and $b^r(z)$ will belong. For the

	$\|\tau\|$	Tree	$F(\tau)$	$\gamma(\tau)$	$\boldsymbol{u}_1(\tau)$	$\sigma(\tau)$
1	1		N	1	$\sum_r \beta^{r,0}$	1
2	2		$N'N$	2	$\sum_r \beta^{r,0} c_r$	1
3	2		LN	2	$\sum_r \beta^{r,1}$	1
4	3		$N''(N,N)$	3	$\sum_r \beta^{r,0} c_r^2$	2
5	3		$N'N'N$	6	$\sum_{r,j} \beta^{r,0} \alpha_r^{j,0} c_j$	1
6	3		$N'(LN)$	6	$\sum_{r,j} \beta^{r,0} \alpha_r^{j,1}$	1
7	3		$LN'N$	6	$\sum_r \beta^{r,1} c_r$	1
8	3		L^2N	6	$\sum_r \beta^{r,2}$	1
9	4		$N'''(N,N,N)$	4	$\sum_r \beta^{r,0} c_r^3$	6
10	4		$N''(N'N,N)$	8	$\sum_{r,j} \beta^{r,0} \alpha_r^{j,0} c_j c_r$	1
11	4		$N''(LN,N)$	8	$\sum_{r,j} \beta^{r,0} \alpha_r^{j,1} c_r$	1
12	4		$N'N''(N,N)$	12	$\sum_{r,j} \beta^{r,0} \alpha_r^{j,0} c_j^2$	2
13	4		$LN''(N,N)$	12	$\sum_r \beta^{r,1} c_r^2$	2
14	4		$N'N'N'N$	24	$\sum_{r,j,k} \beta^{r,0} \alpha_r^{j,0} \alpha_j^{k,0} c_k$	1
15	4		$N'N'(LN)$	24	$\sum_{r,j,k} \beta^{r,0} \alpha_r^{j,0} \alpha_j^{k,1}$	1
16	4		$N'(LN'N)$	24	$\sum_{r,j} \beta^{r,0} \alpha_r^{j,1} c_j$	1
17	4		$N'(L^2N)$	24	$\sum_{r,j} \beta^{r,0} \alpha_r^{j,2}$	1
18	4		$LN'N'N$	24	$\sum_{r,j} \beta^{r,1} \alpha_r^{j,0} c_j$	1
19	4		$LN'(LN)$	24	$\sum_{r,j} \beta^{r,1} \alpha_r^{j,1}$	1
20	4		$L^2N'N$	24	$\sum_r \beta^{r,2} c_r$	1
21	4		L^3N	24	$\sum_r \beta^{r,3}$	1

Table 3.2: Trees, elementary differentials and coefficients for $\tau \in T'$ with $|\tau| \leq 4$.

purpose of calculations, it is also useful to work with basis functions ψ_k for these spaces,

$$a_r^j(z) = \sum_{k=0}^{K_a-1} A_r^{j,k}\psi_k(z) \quad \text{and} \quad b^r(z) = \sum_{k=0}^{K_b-1} B^{r,k}\psi_k(z) \tag{3.10}$$

where $K_a = \dim(V_a)$ and $K_b = \dim(V_b)$. There is a technical assumption that we will adopt to the end of this note.

Assumption 3.3. Any finite dimensional function space V of dimension K used for coefficient functions $a_r^j(z)$ or $b^r(z)$ has the property that the map from V to $\mathbf{R}^K$ defined by

$$f \mapsto \ (f(0), f'(0), \ldots, f^{(K-1)}(0))^T$$

is injective. Equivalently, any function in V is uniquely determined by its first K Taylor coefficients.

Deriving schemes with Natural Continuous Extensions

The approach of Krogstad in [14] is to approximate the nonlinear function

$$N(u(t_0 + \theta h)), \qquad 0 < \theta < 1$$

with a polynomial in θ. Assuming that the functions $a_r^j(z)$ for the internal stages are given, one lets $N(u(t_n + \theta h))$ be approximated by

$$\bar{N}(t_0 + \theta h) = \sum_{r=1}^{s} w_r'(\theta) N_r. \tag{3.11}$$

where $N_r = N(U_r)$ are the stage derivatives and $w_r(\theta)$ are polynomials of degree d, with $w(0) = 0$, such that $\bar{N}(t_0 + \theta h)$ approximates $N(u(t_0 + \theta h))$ uniformly for $0 < \theta < 1$ to a given order. Replacing the exact problem with the approximate one, $\dot{v} = Lv + \bar{N}(t),\ v(t_0) = u_0$ one finds

$$u_1 := v(t_0 + h) = \mathrm{e}^{hL}u_0 + \sum_{r=1}^{s} b^r(hL) N_r,$$

where

$$b^r(z) = \int_0^1 \mathrm{e}^{(1-\theta)z} w_r'(\theta)\, \mathrm{d}\theta.$$

We define the functions

$$\phi_k(z) = \int_0^1 \mathrm{e}^{(1-\theta)z}\theta^k \, \mathrm{d}\theta, \ k = 0, 1, \ldots. \tag{3.12}$$

Thus, here the function space $V_b = \text{span}\{\phi_0, \ldots, \phi_{d-1}\}$, so $\psi_k = \phi_k$ and $K_b = d$ in (3.10). Cox and Matthews [5] presented a fourth order scheme using these basis functions with $K_b = 3$. Krogstad [14] also derived a variant of their method by using a continuous extension as just explained. In [22] Zennaro developed a theory which generalizes the collocation polynomial idea to arbitrary Runge–Kutta schemes. The approach was called Natural Continuous Extensions (NCE). By making a slight modification to the approach of Zennaro, one can find a useful way of deriving exponential integrators as well as providing them with a continuous extension.

Suppose $w_1(\theta), \ldots, w_s(\theta)$ are given polynomials of degree d, and that the stage derivatives $N_1, \ldots, N_s$ of an exponential integrator are given from (3.2). We define the $d-1$ degree polynomial $\bar{N}(t)$ by (3.11).

Definition 3.4. We call $\bar{N}(t)$ of (3.11) a Natural Continuous N-Extension (NCNE) of degree d of the exponential integrator (3.2)–(3.3) if

1.
$$w_r(0) = 0, \quad w_r(1) = b^r(0), \qquad r = 1, \ldots, s.$$

2.
$$\max_{t_0 \le t \le t_1} |N(u(t)) - \bar{N}(t)| = \mathcal{O}(h^{d-1}) \tag{3.13}$$
where $u(t)$ is the exact solution of (3.1) satisfying $u(t_0) = u_0$.

3.
$$\int_{t_0}^{t_1} G(t)(N(u(t)) - \bar{N}(t))\,\mathrm{d}t = \mathcal{O}(h^{p+1}) \tag{3.14}$$
for every smooth matrix-valued function $G(t)$.

It is important to note that the polynomial $\bar{N}(t)$ only depends on the stages N_r and the weights $b^r(0) = \beta^{r,0}$ corresponding to the underlying Runge–Kutta scheme. We also observe that since the $w_r(\theta)$ do not depend on L, an NCNE as defined above is also an NCE in the sense of Zennaro for the system $\dot{u} = N(u)$. Before discussing the existence of NCNEs, we motivate their usefulness in designing exponential integrators. Suppose an underlying Runge–Kutta method has been chosen, and that an NCNE has been found. Then we can determine the functions $b^r(z)$ in order to obtain an exponential Runge–Kutta method of the same order as the underlying scheme.

Theorem 3.5. *If $\bar{N}(t)$ defined from* (3.11) *is an NCNE of degree d for a pth order scheme, then the functions*

$$b^r(z) = \int_0^1 \mathrm{e}^{(1-\theta)z} w'(\theta)\,\mathrm{d}\theta = \beta^{r,0} + z\int_0^1 \mathrm{e}^{(1-\theta)z} w(\theta)\,\mathrm{d}\theta,$$

define the weights of an exponential integrator of order p.

Proof. The exponential integrator we consider, is obtained by replacing (3.1) by

$$\dot{v} = Lv + \bar{N}(t), \qquad v(t_0) = u_0 \tag{3.15}$$

over the interval $[t_0, t_1]$ and by solving (3.15) exactly. We subtract (3.15) from (3.1) to obtain

$$\dot{u} - \dot{v} = L(\dot{u} - \dot{v}) + \big(N(u) - \bar{N}(t)\big)$$

We may solve this equation to obtain

$$u(t_1) - v(t_1) = \int_{t_0}^{t_1} \mathrm{e}^{(t_1 - t)L}\big(N(u(t)) - \bar{N}(t)\big)\,\mathrm{d}t = \mathcal{O}(h^{p+1}),$$

the last equality is thanks to (3.14). □

A reinterpretation of a result by Zennaro [22] combined with Proposition 3.2 leads to the following statement.

Theorem 3.6. *Suppose that an underlying Runge–Kutta scheme with coefficients $\alpha_r^{j,0}$ and $\beta^{r,0}$ of order p is given. Then it is possible to find a set of coefficient functions $a_r^j(z)$ with $a_r^j(0) = \alpha_r^{j,0}$ such that an NCNE of degree $d = \lfloor \frac{p+1}{2} \rfloor$ exists. Moreover, if $\bar{N}(t)$ is a NCNE of degree d then*

$$\left\lfloor \frac{p+1}{2} \right\rfloor \le d \le \min(\nu^*, p)$$

where ν^* is the number of distinct elements among $c_1, \ldots, c_s$.

Corollary 3.7. *For every underlying Runge–Kutta scheme, there exists an exponential integrator whose coefficient functions $b^r(z)$ are in the linear span of the functions $\{\phi_0(z), \ldots, \phi_{d-1}(z)\}$ where $d = \lfloor \frac{p+1}{2} \rfloor$.*

Note in particular that one can derive fourth order exponential integrators using linear combinations of just $\phi_0(z)$ and $\phi_1(z)$ for $b^r(z)$, which is one less than what Cox and Matthews used, we present a specific example in Section 3.4.

Lower bounds for K_a and K_b

We start establishing lower bounds for the number of necessary basis functions ψ_k by proving an ancillary result.

Lemma 3.8. *Let $q \ge 0$ be an integer. The matrix $T_q \in \mathbf{R}^{d\times d}$ with elements*

$$(T_q)_{m+1,k+1} = \frac{1}{(q+m+k+1)!}, \quad 0 \le m, k \le d-1.$$

is invertible.

Proof. Let $w = (w_1, \dots, w_d)^T \in \mathbf{R}^d$ be arbitrary, and consider the polynomial

$$p(x) = \sum_{k=0}^{d-1} w_{k+1} \frac{x^{q+d+k}}{(q+d+k)!}.$$

We compute

$$p^{(d-m-1)}(1) = \sum_{k=0}^{d-1} w_{k+1} \frac{1}{(q+m+k+1)!} = (T_q w)_{m+1}, \quad 0 \le m \le d-1.$$

So $T_q w = 0$ is equivalent to $p^{(j)}(1) = 0$ for $0 \le j \le d-1$. Since $p(x)$ is of the form $x^{q+d} r(x)$ where $r(x)$ is a polynomial of degree at most $d-1$, it follows that $p(x) \equiv 0$ so that $w = 0$. □

As $\phi_k^{(m)}(0) = m!k!/(m+k+1)!$ for ϕ_k defined by (3.12), we get as an immediate consequence of this lemma that the function spaces $V = \operatorname{span}(\phi_q, \dots, \phi_{q+K-1})$, $q \ge 0$ satisfy Assumption 3.3.

Theorem 3.9. *For an exponential integrator of order p, the dimension of the function spaces V_a and V_b are bounded from below as follows*

$$K_a = \dim V_a \ge \left\lfloor \frac{p}{2} \right\rfloor, \quad K_b = \dim V_b \ge \left\lfloor \frac{p+1}{2} \right\rfloor. \tag{3.16}$$

Proof. We will show that using smaller values of K_a or K_b than dictated by (3.16) is incompatible with the order conditions for a scheme of order p. Let V_a and V_b be arbitrary function spaces, satisfying Assumption 3.3, let V denote either of them, and let $d = \dim V$. If $f \in V$, then there are numbers $w_0, \dots, w_{d-1}$ such that

$$f^{(d)}(0) = \sum_{m=0}^{d-1} w_m f^{(m)}(0) \tag{3.17}$$

Suppose now that $d_a := \dim V_a = \lfloor p/2 \rfloor - 1$ and $d_b := \dim V_b = \lfloor (p+1)/2 \rfloor - 1$.

Consider the bicolored trees $\tau_q^{m,k}$ defined by

$$\tau_q^{m,k} = B_+^q \big(W_+^m B_+ (\overbrace{\bullet, \dots, \bullet}^{k}) \big)$$

which consist of a string of $q \ge 0$ black nodes followed by a string of $m > 0$ white nodes with a bushy tree of $k+1$ black nodes grafted onto the topmost leaf of the white nodes. We shall use these trees with $q = 0$ for proving the bound on K_b and with $q = 1$ for K_a. The density of $\tau_q^{m,k}$ is given by

$$\gamma(\tau_q^{m,k}) = \frac{(q+m+k+1)!}{k!}.$$

The trees corresponding to order conditions for a scheme of order p have at most p nodes, $|\tau_q^{m,k}| = q+m+k+1 \le p \Rightarrow 0 \le k \le p-m-1-q$. The definition of d_a and d_b implies that $p-2 \ge 2d_a$ and $p-1 \ge 2d_b$. If we set $q=1$, $m=d_a$ we thus obtain conditions for $0 \le k \le d_a$, whereas $q=0$, $m=d_b$ results in $0 \le k \le d_b$.

The conditions corresponding to $\tau_1^{d_a,k}$ can be expressed as

$$\frac{1}{d_a!}\sum_{r,j=1}^{s} \beta^{r,0}(a_r^j)^{(d_a)}(0)c_j^k = \frac{k!}{(d_a+k+2)!}, \quad 0 \le k \le d_a$$

which, upon insertion of $(a_r^j)^{(d_a)}(0) = \sum w_m (a_r^j)^{(m)}(0)$ as in (3.17) yields

$$\frac{k!\,d_a!}{(k+d_a+2)!} = \sum_{m=0}^{d_a-1} w_m \Big(\sum_{r=1}^{s} \beta^{r,0}(a_r^j)^{(m)}(0)c_j^k\Big) = \sum_{m=0}^{d_a-1} w_m \,\frac{m!\,k!}{(m+k+2)!}.$$

The conditions for $\tau_0^{d_b,k}$ similarly yield

$$\frac{k!\,d_b!}{(k+d_b+1)!} = \sum_{m=0}^{d_b-1} w_m \Big(\sum_{r=1}^{s} (b^r)^{(m)}(0)\,c_r^k\Big) = \sum_{m=0}^{d_b-1} w_m \frac{m!\,k!}{(m+k+1)!}.$$

In both cases ($d = d_a$ or d_b) we end up with a $(d+1)\times d$ linear system of equations for determining w_m, $m = 0, \dots, d-1$. This system is of the form

$$\sum_{m=0}^{d-1} \frac{m!\,k!}{(q+m+k+1)!}\, w_m = \frac{k!\,d!}{(q+k+d+1)!}, \quad 0 \le k \le d$$

for $q \in \{0,1\}$ and is solvable only if the matrix with elements

$$(T_q)_{m+1,k+1} = \frac{m!\,k!}{(q+m+k+1)!}, \quad 0 \le m,k \le d$$

is singular. However, Lemma 3.8 implies that the matrix T_q is invertible so the linear system is inconsistent. It is hence not possible to choose $K_a = d_a$ or $K_b = d_b$. □

Some remarks regarding the implications of Theorem 3.9 are in order. First, note that the bounds in the theorem are not proved to be sharp, however Theorem 3.6 ensures that the lower bound is attainable for the dimension of V_b if a basis is given by the functions ϕ_k of (3.12). However, this result does not apply to the space V_a of the functions $a_r^j(z)$. For instance, in the case $p=5$, one can prove that it is indeed possible to take $K_a = 2$, but V_a cannot be the span of ϕ_0 and ϕ_1. But an example of a feasible two-dimensional space is that with basis

$\psi_0(z) = \phi_1(z)$ and $\psi_1(z) = \phi_1(\frac{3}{5}z)$. A particular scheme is given in Table 3.4, though the usefulness of the bounds are questionable in this particular example. Using, say $V_b = \text{span}\{\phi_0, \phi_1, \phi_2\}$ combined with the above choice of V_a requires the computation with a total of 4 basis functions, whereas only 3 are necessary if one instead chooses $V_a = V_b$.

We furthermore note that the minimum attainable value of the parameters K_a and K_b depend only on the order p of the underlying Runge–Kutta scheme and the choice of the basis functions ψ_k. Specifically, the coefficients of the underlying Runge–Kutta scheme do not influence the minimum values of K_a and K_b.

3.4 Examples of exponential integrators

In this section we will present examples of exponential integrators. For fourth order methods, one will notice that some well-known schemes are obtained for particular choices of the free parameters, suggesting that a search on the entire space of parameters may result in schemes which in some sense may have better properties than the known methods. The scheme of order 5 presented at the end is only included as an illustration of the proposed procedure for solving the order conditions. It remains a subject of future research to establish to which extent higher order exponential integrators are useful for practical purposes.

The procedure we have used in constructing schemes may be summarized as follows

1. Choose an underlying Runge–Kutta scheme. This determines $\alpha_r^{j,0}$ and $\beta^{r,0}$.

2. Choose basis functions $\psi_k(z)$ for the coefficient functions and determine K_a and K_b.

3. Use the order conditions for the trees of the form $W_+^m(\tau_C)$ where τ_C is a tree with only black nodes, and determine $\beta^{r,m}$, for $1 \le m \le K_b - 1$. See also (3.9).

4. Identify order conditions which are linear in $c'_r = \sum_{j=1}^s \alpha_r^{j,1}$ and which otherwise depend only on $\beta_r^{j,m}$, $0 \le m \le K_b - 1$ and $\alpha_r^{j,0}$, and solve for c'_r.

5. Identify remaining conditions which depend linearly on $\alpha_r^{j,1}$. Solve for $\alpha_r^{j,1}$ together with $c'_r = \sum_{j=1}^s \alpha_r^{j,1}$. Repeat this procedure to solve for $\alpha_r^{j,m}$, $2 \le m \le K_a - 1$.

6. $\beta^{r,m}$ are now uniquely determined for $m \ge K_b$ and $\alpha_r^{j,m}$ for $m \ge K_a$ by (3.17). Verify all remaining order conditions for $\beta^{r,m}$, $K_b \le m \le p - 1$ and for $\alpha_r^{j,m}$, $K_a \le m \le p-2$. If inconsistencies appear, the basis functions are not feasible.

7. Verify all remaining order conditions.

In most cases we have considered, once $\alpha_r^{j,0}$ and $\beta^{r,0}$ have been chosen, one can find the remaining $\alpha_r^{j,m}$ independently of the $\beta^{r,m}$. Most of the exponential integrators we find in the literature are based on the classical fourth order scheme of Kutta, and it is typical that one can combine $a_r^j(z)$ from one scheme with $b^r(z)$ from another scheme and still get overall order four.

In the class of ETD schemes, proposed by Cox and Matthews in [5] and Krogstad in [14], the space V_b is spanned by the three functions ϕ_0, ϕ_1, and ϕ_2 of (3.12). However in the former reference they use $\dim V_a = 2$ with a basis $\{\phi_0(z/2), z\phi_0(z/2)^2\}$. This V_a coincides with the one used in [3], given in Table 3.1b.

Another choice is to use $\phi_k(z)$ of (3.12) both for V_a and V_b. In Table 3.3 we characterize all resulting schemes with $K_a = 2$ and $K_b = 3$. It is interesting to note that Theorem 3.9 predicts $K_a \geq 2$ and $K_b \geq 2$, and indeed, by choosing $\gamma_1 = \frac{1}{3}$ and $\gamma_2 = -\frac{1}{3}$, we see that ϕ_2 disappears from the $b^r(z)$ functions. Choosing $\gamma_1 = \gamma_2 = 0$, we recover the $b^r(z)$ functions obtained in [5]. Letting V_b, be

$$
\begin{aligned}
a_2^1(z) &= -(\tfrac{1}{2} + \rho_1)\phi_0(z) + (2\rho_1 + 2)\phi_1(z) \\
a_3^1(z) &= (1 + \rho_1 - \tfrac{1}{4}(\rho_2 + \rho_3))\phi_0(z) + (-2 - 2\rho_1 + \tfrac{1}{2}(\rho_2 + \rho_3))\phi_1(z) \\
a_3^2(z) &= (-1 + \tfrac{1}{4}(\rho_2 + \rho_3))\phi_0(z) + (3 - \tfrac{1}{2}(\rho_2 + \rho_3)\phi_1(z) \\
a_4^1(z) &= \tfrac{1}{2}(\rho_2 + \rho_3)\phi_0(z) - (\rho_2 + \rho_3)\phi_1(z) \\
a_4^2(z) &= -\frac{\rho_2}{2}\phi_0(z) + \rho_2\phi_1(z) \\
a_4^3(z) &= (1 - \tfrac{1}{2}\rho_3)\phi_0(z) + \rho_3\phi_1(z) \\
b^1(z) &= (1 + \gamma_2)\,\phi_0(z) + (-3 - 6\gamma_2)\,\phi_1(z) + (6\gamma_2 + 2)\,\phi_2(z) \\
b^2(z) &= (-\gamma_1 - 2\gamma_2)\,\phi_0(z) + (6\gamma_1 + 12\gamma_2 + 2)\phi_1(z) + (-6\gamma_1 - 12\gamma_2 - 2)\,\phi_2(z) \\
b^3(z) &= \gamma_1\,\phi_0(z) + (-6\gamma_1 + 2)\,\phi_1(z) + (6\gamma_1 - 2)\,\phi_2(z) \\
b^4(z) &= \gamma_2\,\phi_0(z) + (-6\gamma_2 - 1)\,\phi_1(z) + (6\gamma_2 + 2)\,\phi_2(z)
\end{aligned}
$$

Table 3.3: Coefficient function for a fourth order ETD scheme with classical RK4 as underlying scheme. Basis functions given by (3.12).

spanned by $\psi_0(z) = \phi_0(z)$ and $\psi_1(z) = \phi_0(z/2)$, one obtains the unique solution

$$
\begin{aligned}
b^1(z) &= \tfrac{1}{2}\phi_0(z) - \tfrac{1}{3}\phi_0(z/2) \\
b^2(z) &= b^3(z) = \tfrac{1}{3}\phi_0(z) \\
b^4(z) &= -\tfrac{1}{6}\phi_0(z) + \tfrac{1}{3}\phi_0(z/2).
\end{aligned}
\tag{3.18}
$$

These weights coincide with the ones derived in the fourth order scheme in [3],

given in Table 3.1b. Yet another choice is to let V_b consist of functions of the form $p(z)\phi_0(z)$ where $p(z)$ is a polynomial of degree 1, and we recover $b^r(z)$ as in Table 3.1a.

Finally, we give an example of a fifth order exponential integrator based on a scheme of Fehlberg. As indicated in Section 3.3 we take $\dim V_a = 2$ with basis $\psi_0(z) = \phi_1(z)$ and $\psi_1(z) = \phi_1(\frac{3}{5}z)$. For V_b we use the basis $\psi_k(z) = \phi_k(z)$ for $k = 0, 1, 2$. The resulting coefficient functions are given in Table 3.4.

In summary, this paper presents a complete order theory for exponential integrators of the form (3.2)–(3.3). From deriving order conditions by means of bicolored trees to proving bounds for the lowest possible number of basis functions, the results presented herein provide a general framework for constructing schemes of this type. A number of issues are, however, not addressed in the present paper. These include systematically choosing basis functions ψ_k, and how to construct schemes with low error constants.

Exponential integrators are interesting from the point of view of handling unbounded or stiff operators, yet the order theory does not say anything about what happens for large eigenmodes of L in (3.1). Determining conditions for favorable behavior in light of such operators should be an arena for future work.

3.5 Acknowledgements

Thanks to Christian Bower for advice regarding the enumeration of bicolored trees, and to an anonymous referee for suggesting a simpler proof of Lemma 3.7.

$$
\begin{array}{c|ccccc}
0 & & & & & \\
\frac{2}{9} & a_2^1(z) & & & & \\
\frac{1}{3} & a_3^1(z) & a_3^2(z) & & & \\
\frac{3}{4} & a_4^1(z) & a_4^2(z) & a_4^3(z) & & \\
1 & a_5^1(z) & a_5^2(z) & a_5^3(z) & a_5^4(z) & \\
\frac{5}{6} & a_6^1(z) & a_6^2(z) & a_6^3(z) & a_6^4(z) & a_6^5(z) \\
\hline
 & b^1(z) & b^2(z) & b^3(z) & b^4(z) & b^5(z) \quad b^6(z)
\end{array}
,
$$

$$
\begin{aligned}
b^1(z) &= \tfrac{47}{150}\phi_0 - \tfrac{188}{75}\phi_1 + \tfrac{47}{15}\phi_2 \\
b^2(z) &= 0 \\
b^3(z) &= -\tfrac{43}{25}\phi_0 + \tfrac{132}{5}\phi_1 - 33\phi_2 \\
b^4(z) &= \tfrac{4124}{75}\phi_0 - \tfrac{6152}{15}\phi_1 + \tfrac{1352}{3}\phi_2 \\
b^5(z) &= \tfrac{189}{10}\phi_0 - \tfrac{662}{5}\phi_1 + 142\phi_2 \\
b^6(z) &= -\tfrac{1787}{25}\phi_0 + \tfrac{12966}{25}\phi_1 - \tfrac{2814}{5}\phi_2
\end{aligned}
$$

(i,j)	$(2,1)$	$(3,1)$	$(3,2)$	$(4,1)$	$(4,2)$	$(4,3)$	$(5,1)$	$(5,2)$
a_i^j	$-\frac{2}{3}$	$\frac{569}{11544}$	$-\frac{831}{3848}$	$-\frac{77157}{61568}$	$\frac{587979}{61568}$	$-\frac{405}{64}$	$\frac{655263}{7696}$	$-\frac{1148769}{7696}$
$\hat{a}_i^j$	$\frac{10}{9}$	$\frac{1355}{11544}$	$\frac{2755}{3848}$	$\frac{143535}{61568}$	$-\frac{821745}{61568}$	$\frac{675}{64}$	$-\frac{2031205}{23088}$	$\frac{1252665}{7696}$

(i,j)	$(5,3)$	$(5,4)$	$(6,1)$	$(6,2)$	$(6,3)$	$(6,4)$	$(6,5)$	—
a_i^j	$\frac{1593}{40}$	$\frac{144}{5}$	$-\frac{2212835}{277056}$	$\frac{477285}{30784}$	$-\frac{39}{16}$	$-\frac{4}{9}$	$-\frac{185}{96}$	—
$\hat{a}_i^j$	$-\frac{405}{8}$	$-\frac{80}{3}$	$\frac{6888625}{831168}$	$-\frac{496525}{30784}$	$\frac{65}{16}$	$\frac{20}{27}$	$\frac{575}{288}$	—

Table 3.4: Coefficient functions for a fifth order exponential integrator with Fehlberg's fifth order RK as the underlying scheme. Here $a_i^j(z) = a_i^j\phi_1(z) + \hat{a}_i^j\phi_1(\frac{3}{5}z)$.

Bibliography

[1] Berland, H., Owren, B., Skaflestad, B.: B-series and order conditions for exponential integrators. Tech. Rep. 5/04, Department of Mathematical Sciences, NTNU, Norway (2004). `http://www.math.ntnu.no/preprint/`

[2] Butcher, J.C.: Numerical methods for ordinary differential equations. John Wiley & Sons Ltd., Chichester, UK (2003)

[3] Celledoni, E., Marthinsen, A., Owren, B.: Commutator-free Lie group methods. FGCS **19**(3), 341–352 (2003)

[4] Certaine, J.: The solution of ordinary differential equations with large time constants. In: Mathematical methods for digital computers, pp. 128–132. Wiley, New York (1960)

[5] Cox, S.M., Matthews, P.C.: Exponential time differencing for stiff systems. J. Comput. Phys. **176**(2), 430–455 (2002)

[6] Ehle, B.L., Lawson, J.D.: Generalized Runge–Kutta processes for stiff initial-value problems. J. Inst. Maths. Applics. **16**, 11–21 (1975)

[7] Friedli, A.: Verallgemeinerte Runge–Kutta Verfahren zur Lösung steifer Differentialgleichungssysteme. In: Numerical treatment of differential equations, pp. 35–50. Lecture Notes in Math., Vol. 631. Springer, Berlin (1978)

[8] Hairer, E., Lubich, C., Wanner, G.: Geometric Numerical Integration, *Springer Series in Computational Mathematics*, vol. 31. Springer-Verlag, Berlin (2002). Structure-preserving algorithms for ordinary differential equations

[9] Hairer, E., Nørsett, S.P., Wanner, G.: Solving ordinary differential equations. I, *Springer Series in Computational Mathematics*, vol. 8, second edn. Springer-Verlag, Berlin (1993). Nonstiff problems

[10] Hairer, E., Wanner, G.: Solving ordinary differential equations. II, *Springer Series in Computational Mathematics*, vol. 14, second edn. Springer-Verlag, Berlin (1996). Stiff and differential-algebraic problems

[11] Hochbruck, M., Lubich, C., Selhofer, H.: Exponential integrators for large systems of differential equations. SIAM J. Sci. Comput. **19**(5), 1552–1574 (1998)

[12] Hochbruck, M., Ostermann, A.: Explicit exponential Runge–Kutta methods for semilinear parabolic problems. SIAM J. Numer. Anal. **43**(3), 1069–1090 (2005)

[13] Kassam, A.K., Trefethen, L.N.: Fourth-order time-stepping for stiff PDEs. SIAM J. Sci. Comput. **26**(4), 1214–1233 (electronic) (2005)

[14] Krogstad, S.: Generalized integrating factor methods for stiff PDEs. J. of Comp. Phys. **203**(1), 72–88 (2005)

[15] Lawson, J.D.: Generalized Runge–Kutta processes for stable systems with large Lipschitz constants. SIAM J. Numer. Anal. **4**, 372–380 (1967)

[16] Minchev, B., Wright, W.M.: A review of exponential integrators for semilinear problems. Tech. Rep. 2/05, Department of Mathematical Sciences, NTNU, Norway (2005). `http://www.math.ntnu.no/preprint/`

[17] Munthe-Kaas, H.: High order Runge–Kutta methods on manifolds. In: Proceedings of the NSF/CBMS Regional Conference on Numerical Analysis of Hamiltonian Differential Equations (Golden, CO, 1997), vol. 29,1, pp. 115–127 (1999)

[18] Munthe-Kaas, H., Owren, B.: Computations in a free Lie algebra. R. Soc. Lond. Philos. Trans. Ser. A Math. Phys. Eng. Sci. **357**(1754), 957–981 (1999)

[19] Murua, A., Sanz-Serna, J.M.: Order conditions for numerical integrators obtained by composing simpler integrators. R. Soc. Lond. Philos. Trans. Ser. A Math. Phys. Eng. Sci. **357**(1754), 1079–1100 (1999)

[20] Nørsett, S.P.: An *A*-stable modification of the Adams–Bashforth methods. In: Conf. on Numerical Solution of Differential Equations (Dundee, Scotland, 1969), pp. 214–219. Springer, Berlin (1969)

[21] Steihaug, T., Wolfbrandt, A.: An attempt to avoid exact Jacobian and nonlinear equations in the numerical solution of stiff differential equations. Math. Comp. **33**(146), 521–534 (1979)

[22] Zennaro, M.: Natural continuous extensions of Runge–Kutta methods. Math. Comp. **46**(173), 119–133 (1986)

Paper III

Håvard Berland, Bård Skaflestad, and Will M. Wright:

EXPINT — A MATLAB package for exponential integrators

To appear in ACM Transactions on Mathematical Software 2007.

Chapter 4

EXPINT — A MATLAB package for exponential integrators

Abstract:

Recently, a great deal of attention has been focused on the construction of exponential integrators for semi-linear problems. In this paper we describe a MATLAB package which aims to facilitate the quick deployment and testing of exponential integrators, of Runge–Kutta, multistep and general linear type. A large number of integrators are included in this package along with several well-known examples. The so-called φ functions and their evaluation is crucial for accuracy, stability and efficiency of exponential integrators, and the approach taken here is through a modification of the scaling and squaring technique; the most common approach used for computing the matrix exponential.

4.1 Introduction

EXPINT is a MATLAB package designed as a tool to facilitate easy testing of various exponential integrators. Exponential integrators are a class of numerical methods specifically designed for the numerical solution of semi-linear problems and have recently had a renewed interest in their development, generally tailored to provide efficient time integration technology for semi-discretized PDEs. Exponential integrators are essentially an alternative to implicit methods for the numerical solution of stiff or highly oscillatory differential equations. Assuming that the substantial portion of the problem's inherent dynamics may be ascribed to the linear term, the methods combine exact solution of the linear part with numerical resolution of the remaining, nonlinear part. The integrators are designed to be explicit and the coefficients of the method are functions of the linear term. The aim of the EXPINT package is threefold: Create a uniform environment which enables the comparison of various integrators; Provide tools for easy visualization of numerical behaviour; To be easily modified so that users can include problems and integrators of their own. We set out to provide as efficient an implementation as possible without compromising generality and ease of comparison and usability.

The outline of the paper is as follows. In the first section we outline the system

requirements for the EXPINT package. Section 4.2 describes the meaning of exponential integrators and how they are formatted in the present context. Exponential integrators of Runge–Kutta and multistep type are generalized using general linear methods, and our way of writing the integrators is tailored thusly. In Section 4.3 we outline installation of this package and describe the package structure. Equations and boundary conditions are represented in a problem structure described in Section 4.4, where also some details regarding an example included in this package can be found. Formulae for the diagonal Padé approximations and a generalization of the squaring process for the φ functions are given in Section 4.5. Functions essential to the package are briefly described in a user-oriented way in Section 4.6. In the last two sections some numerical experiments are recorded and we address future work. As a supplement to this paper, a README file is contained in the main directory of the package, which outlines the examples and lists the integrators included.

The README file and the MATLAB files can be found on the webpage

`http://www.math.ntnu.no/num/expint/matlab.php`

System requirements for the EXPINT package

In terms of computing power, the system requirements are quite modest. Any computer capable of running a moderately up-to-date version of MATLAB is sufficient for experimenting with exponential integrators by means of the EXPINT package. On the other hand, faster central processing units and memory systems decrease actual runtime.

However, the package places more stringent demands on the actual MATLAB environment. In particular, some of the package's fundamental routines use MATLAB language features and core functions which were introduced relatively recently. As a consequence, the package will not run without modification in MATLAB environments preceding the release 13 series (i.e. MATLAB 6.5) released in June of 2002.

MATLAB 7, also known as the release 14 series, offers additional language features which will make the package more efficient; most notably more powerful function handles compared to version 6.5. However, to not limit its portability any further the package does not at present take advantage of these features.

4.2 Exponential integrators

Exponential integrators are numerical schemes specifically designed for solving differential equations where it is possible to split the problem into a linear and a nonlinear part

$$\dot{y} = Ly + N(y,t), \qquad y(t_{n-1}) = y_{n-1}, \tag{4.1}$$

where $y \in \mathbf{C}^d$, $L \in \mathbf{C}^{d\times d}$ and $N\colon \mathbf{C}^d \times \mathbf{R} \to \mathbf{C}^d$. Typically in the applications of interest (discretizations of PDEs), the matrix L comes from the spatial discretization of a differential operator. We will assume that the matrix L contains the significant part of the dynamics of the problem and the nonlinear term is slowly varying. A bound on L depends on the discretization parameter; when this parameter tends to zero, it is generally not possible to obtain a uniform bound on L. The standard approach used to solve such problems is by implicit schemes. The aim of exponential integrators is to treat the linear term exactly and thereby removing the need for implicitness. The nonlinear part can then be treated in an explicit and computationally more efficient manner. Exponential integrators have a long and interesting history which is discussed in [23].

An exponential integrator has two main features:

(A) If $L = 0$, then the scheme reduces to a standard general linear scheme. This is often called the underlying general linear scheme.

(B) If $N(y, t) = 0$ for all y and t, then the scheme reproduces the exact solution of (4.1).

To ensure that (B) is satisfied, the exponential function must be used within the numerical scheme. Despite the fact that the bound on L is dependent on the discretization parameter, typically the coefficients of the scheme will be bounded independently of the discretization parameter. Generally it is not possible to obtain efficient schemes by solely using the exponential function within the scheme. Functions closely related to the exponential, known as φ functions, are also needed.

The method by which exponential integrators are implemented in this MATLAB package allows for a relaxation of feature (B), in which some approximation of the exponential can be used. An example included in this package is the Crank–Nicolson scheme. In the remainder of this section we will first describe how to implement exponential integrators of Runge–Kutta type, followed by a discussion of general linear schemes which generalize both Runge–Kutta and multistep schemes.

The format of an exponential Runge–Kutta scheme

For an s-stage exponential integrator of Runge–Kutta type, we define the internal stages and output approximation as follows:

$$\begin{aligned} Y_i &= h\sum_{j=1}^{s} a_{ij}(hL)\, N(Y_j, t_{n-1} + c_j h) + u_{i1}(hL) y_{n-1}, \quad i = 1, \ldots, s, \\ y_n &= h\sum_{i=1}^{s} b_i(hL) N(Y_i, t_{n-1} + c_i h) + v_1(hL) y_{n-1}. \end{aligned} \tag{4.2}$$

It is worthwhile to point out the lack of the collocation points c_i in the coefficient functions arguments, as opposed to the formats presented in [11, 30]. This allows for more generality within the schemes.

In order to fulfill feature (A) above, we require $u_{i1}(0) = 1$, $a_{ij}(0) = a_{ij}$, $v_1(0) = 1$, and $b_i(0) = b_i$, where the real numbers a_{ij} and b_i are the coefficients of the underlying Runge–Kutta scheme. The functions used in (4.2) are conveniently represented in an extended Butcher tableau

$$\begin{array}{c|ccc|c} c_1 & a_{11}(z) & \cdots & a_{1s}(z) & u_{11}(z) \\ \vdots & \vdots & & \vdots & \vdots \\ c_s & a_{s1}(z) & \cdots & a_{ss}(z) & u_{s1}(z) \\ \hline & b_1(z) & \cdots & b_s(z) & v_1(z) \end{array} \tag{4.3}$$

The MATLAB functions defining an exponential integrator in the `schemes` directory returns precisely these functions given in this tableau.

The extension from a traditional integrator to an exponential integrator is not unique. We give two well-known examples of how the forward Euler scheme can be extended to the exponential setting. The first scheme is most commonly known as Lawson–Euler

$$y_n = h\mathrm{e}^{hL}N(y_{n-1}, t_{n-1}) + \mathrm{e}^{hL}y_{n-1}, \qquad \begin{array}{c|c|c} 0 & 0 & 1 \\ \hline & \mathrm{e}^{hL} & \mathrm{e}^{hL} \end{array} \tag{4.4}$$

The second scheme is Nørsett–Euler

$$y_n = h\varphi_1(hL)N(y_{n-1}, t_{n-1}) + \mathrm{e}^{hL}y_{n-1}, \qquad \begin{array}{c|c|c} 0 & 0 & 1 \\ \hline & \varphi_1(hL) & \mathrm{e}^{hL} \end{array} \tag{4.5}$$

The latter scheme has been reinvented several times and is also known as the ETD Euler, filtered Euler, Lie–Euler and exponentially fitted Euler. The function φ_1 will be defined later.

Extension to general linear schemes

We explain the necessary notation for the construction of exponential general linear schemes, which were introduced in [23]. The notation follows closely with that developed for the standard schemes in [4].

To perform a step of length h in an exponential general linear scheme, we need to import r approximations into the step, denoted as $y_i^{[n-1]}$, $i = 1, \ldots, r$. These quantities can be of a very general nature. The internal stages (as in the Runge–Kutta case) are written as Y_i, $i = 1, \ldots, s$. After the step is completed, r updated approximations are computed. These are then used in the next step.

Each step in an exponential general linear scheme can be written as

$$\begin{aligned} Y_i &= h\sum_{j=1}^{s} a_{ij}(hL)N(Y_j, t_{n-1}+c_jh) + \sum_{j=1}^{r} u_{ij}(hL)y_j^{[n-1]}, \qquad i = 1,\dots,s,\\ y_i^{[n]} &= h\sum_{j=1}^{s} b_{ij}(hL)N(Y_j, t_{n-1}+c_jh) + \sum_{j=1}^{r} v_{ij}(hL)y_j^{[n-1]}, \qquad i = 1,\dots,r. \end{aligned} \tag{4.6}$$

The coefficient functions are grouped into matrices,

$$\begin{array}{c|ccc|ccc} c_1 & a_{11}(z) & \cdots & a_{1s}(z) & u_{11}(z) & \cdots & u_{1r}(z)\\ \vdots & \vdots & & \vdots & \vdots & & \vdots\\ c_s & a_{s1}(z) & \cdots & a_{ss}(z) & u_{s1}(z) & \cdots & u_{sr}(z)\\ \hline & b_{11}(z) & \cdots & b_{1s}(z) & v_{11}(z) & \cdots & v_{1r}(z)\\ & \vdots & & \vdots & \vdots & & \vdots\\ & b_{r1}(z) & \cdots & b_{rs}(z) & v_{r1}(z) & \cdots & v_{rr}(z) \end{array} \tag{4.7}$$

The manner in which exponential general linear schemes are implemented in this package assumes a special structure of the vector $y^{[n-1]}$, the quantities which are passed from step to step

$$y^{[n-1]} = \begin{bmatrix} y_{n-1} & hN_{n-2} & hN_{n-3} & \cdots & hN_{n-r} \end{bmatrix}^T,$$

where $N_{n-i} = N(y_{n-i}, t_{n-i})$. This choice enables both the ETD Adams–Bashforth [3, 9, 26] and generalized Lawson schemes [20, 31] to be conveniently represented in a single framework. Restricting ourselves to schemes of this form removes the need for complicated starting procedures. Instead, an exponential Runge–Kutta scheme can be used for the first $r-1$ steps, recording the approximation to the nonlinear term. In our MATLAB code for exponential general linear methods, the scheme `hochost4` is used for the first $r-1$ steps, but this is easily changed in the file `expglm.m`.

To our knowledge, the only exponential integrators that do not fit into this framework are the methods developed in [6], which from a stability point of view are similar to the ETD Adams–Bashforth schemes [27]. If integrators are developed with, for example, a Nordsieck vector being passed from step to step, it is not difficult to change the source code to allow for this option, but it will break compatibility with the current implementation of the schemes.

In this paper we will use various terms of order. Firstly, the observed order is the order the numerical scheme achieves in a certain numerical example. Secondly, the non-stiff order refers to the instance when the operator L is uniformly bounded. Such conditions were derived in [1, 23]. Finally, the stiff order refers to the case when L is not uniformly bounded; see [6, 17, 27] for more details.

Families of exponential integrators

Several families of exponential integrators may be derived from various problem transformations or approximation strategies. The following brief sections review some of the common classes presented in the literature.

Lawson schemes

Lawson [21] used a change of variables $v(t) = \mathrm{e}^{(t-t_{n-1})L} y(t)$, to construct a modified differential equation in the v variable. This modified differential equation is then integrated using a standard explicit solver. The result is then back-transformed to provide an approximation in the y variable. The overall scheme in the original variables was pointed out in [10]. All Lawson schemes have stiff order one.

ETD schemes

ETD (Exponential Time Differencing) schemes are the most common class of exponential integrators and they date back to 1960 [8]. These schemes, based on Adams–Bashforth methods were first discovered by Nørsett [26]. ETD schemes based on Runge–Kutta schemes were independently discovered by several authors; we cite the papers [11, 29, 18]. The main idea behind the ETD schemes is to approximate the nonlinear term in the variation of constants formula by a suitable algebraic polynomial. The ETD Adams–Bashforth schemes obtain the same stiff order as the classical order [6, 27], whereas the ETD Runge–Kutta schemes are required to satisfy complicated stiff order conditions. These were recently constructed up to order four in [17].

Affine Lie group schemes

Munthe-Kaas [25] pointed out that the affine Lie group could be used to solve semi-linear problems using Lie group schemes. Unfortunately, the schemes of RKMK type were shown to exhibit instabilities due to the use of commutators. The commutator-free schemes [7] overcome this problem but the overall stiff order achieved is limited to two.

Generalized Lawson schemes

Krogstad [20] proposed a more detailed transformation than that of Lawson. The resulting schemes are exponential general linear schemes, which achieve stiff order $r+1$ for a transformation of order r. These schemes and modifications are discussed in detail in [31].

Name	Nonstiff p	Stiff p	Stages s	Output r	$\#\varphi$	matvecs
Lawson–Euler	1	1	1	1	1	1
ABLawson4	4	1	1	4	4	4
Lawson4	4	1	4	1	2	6
Nørsett–Euler	1	1	1	1	2	2
ABNørsett4	4	4	1	4	5	5
ETD4RK	4	2	4	1	6	10
Krogstad	4	3	4	1	7	11
Hochbruck–Osterm.	4	4	5	1	8	13
Cfree4	4	2	4	1	4	9
RKMK4t	4	2	4	1	4	9
GenLawson43	4	4	4	3	8	16
ModGenLawson43	4	4	4	3	9	17
PEC423	4	4	2	3	5	8
PECEC433	4	4	3	3	5	10

Table 4.1: Selected integrators included in the package, along with relevant figures describing their properties. $\#\varphi$ is the number of distinct φ functions needed to be evaluated for each scheme. Note that counting the number of φ functions and matrix-vector products does not give a complete description of the efficiency of the scheme.

4.3 Installation and quick-start

This section describes how you can install the EXPINT package and perform numerical experiments as soon as possible.

Installation

Fetch the archived file

`http://www.math.ntnu.no/num/expint/matlab/expint.tar.gz`

and unpack the file somewhere on your computer. In a UNIX shell you would typically write

```
$ tar zxvf expint.tar.gz
```

This will unpack the archive and will give you a directory named `expint` containing the directories `problems`, `schemes`, `common` and `tests` and the file `startup.m`.

A quick test, local error plot

In the main directory of the package, that is in the directory where the mentioned directories are present, launch MATLAB from the command line. This ensures that the `startup.m` file is being read. If MATLAB was not opened in this directory move to the correct directory and type `startup` at the prompt; this will ensure that the correct paths are set. The commands in Listing 4.1 may be typed in directly, or run via a script (`m`-file). For the moment we regard this example code to be self-explanatory, but details will be explained further in subsequent sections.

Directory structure

`problems` Each equation or problem you wish to solve needs to be defined by a "problem" structure. Files in this directory are helper functions for generating this structure (a MATLAB structure). Each problem is actually a system of ODEs, so typically the semi-discretization is what is provided by the "problem" when solving PDEs. The problem-structure is dealt with in Section 4.4.

`schemes` Files in this directory represent a specific scheme or integrator in the format for exponential integrators which were introduced in Section 4.2.

`common` All kinds of utility functions are inserted in here. Important functions are described in Section 4.6.

Listing 4.1: An example script to do a local error computation with plotting.

```
% Define problem to be solved:
problem = nls('ND', 256, 'IC', 'smooth', 'Potential', 'zero', 'lambda', 1);

% We should make a suitable vector of timesteps. The smallest timestep
% must divide all other timesteps (if not, the computation would be
% unnecessarily slower, so it is not supported)
steplist = unique(round(10.^(3:-0.1:0)));
dt = 10^-3 .* steplist;

% Choose schemes to be used:
schemes = setupschemes('etd4rk', 'lawson4', 'genlawson43');

% Do computational test with lawson4 as a reference solver (with
% smaller timestep):
schemes = localorder(problem, dt, schemes, 'lawson4');

% Make plot:
orderplot(schemes, 'Timestep_h', 'Local_error', problem.problemname);
```

`tests` In here we have put example scripts to facilitate testing. You could also put your own scripts here if so desired.

Choosing schemes to test

It is a typical situation to test many schemes at once in order to do comparisons. For this, one defines an array of structures describing the schemes to be used. Let `schemes` be the array; each element is a structure describing one scheme. The structure elements are

`coffunc` A string naming a MATLAB function which is able to return the coefficient functions in the format of (4.3) or (4.7).

`name` A describing name for the scheme in question. If possible, it is advisable to have this field equal the `coffunc` field. This field is used for legends in plots.

`linestyle` This is a string that will be used by the plotting routines to determine the line style for the scheme in question, in accordance with the MATLAB plot function.

A function called `setupschemes` is added for convenience in order to easily build up this structure, where you only list the names of the schemes, and the structure will be returned. See Listing 4.1 for an example. In addition, the scripts `localorder` and `globalorder` add computational results to this structure for use within the script `orderplot`.

An optional field in the scheme structure is

`relstages` A positive integer which can be used as a weight. If `relstages` $= 4$ for a scheme, the actual timestep used by `expglm` will be one fourth of your supplied timestep. This could be useful in some circumstances for producing fairer comparisons.

4.4 Defining equations to solve

The approach taken to tell MATLAB about the current differential equation to solve is done using a data structure, which first must be defined in every computation. Assembling this structure is best done in its own function file, and it will also contain references to other functions representing the problem in question.

The problem structure

The following fields are mandatory for defining a problem representing a differential equation.

`ND` A discretization parameter. It is up to the problem to interpret this parameter in a suitable way. It could be the number of Fourier modes or the number of inner points in a finite difference discretization.

`y0` The initial condition for the corresponding system of ODEs. The package assumes that this is a column vector.

`L` The matrix L in (4.1).

`N` A string naming a MATLAB-function implementing the function $N(y, t)$ in (4.1). The calling syntax is assumed to be `N(y, t, problem)` where `y` is the phase value, `t` is the time point, and `problem` is the entire problem structure.

`problemname` A textual description of the problem.

In addition, we have found it useful to include these fields as well, but these should be treated as optional by all code:

`postprocessing` Function to be used for post-processing the data, for example inverse fourier transform if the problem is spectrally discretized. If the problem is of a multidimensional nature, it is appropriate to do a conversion from vector to matrix form of the data more suited for plotting.

`LplusN` A string naming a MATLAB-function implementing $Ly + N(y, t)$. This is needed in order to be able to use MATLAB's `ode15s` as a reference solver.

Included problems

In the `problems` directory a bundle of example problems is included. The included problems serve mainly as examples on how to implement flexible problem structures. Beyond returning a structure fulfilling the requirements of Section 4.4, the actual implementation is entirely up to the user. As an illustrative example we include a brief discription of the 1D nonlinear Schrödinger equation represented by the file `nls.m`.

This problem was introduced in Listing 4.1 and takes the form

$$\mathrm{i}y_t = -y_{xx} + (V(x) + \lambda|y|^2)y, \qquad x \in [-\pi, \pi].$$

with some initial condition and with periodic boundary conditions. The problem file includes the (spectral) semi-discretization of the problem. Several different potential functions and initial conditions are implemented and may be chosen when initializing the problem structure. The nonlinear constant λ may be similarly specified by the user. A typical example of a calling sequence is the one in Listing 4.1,

```
problem = nls('ND', 256, 'IC', 'smooth', 'Potential', 'zero', 'lambda', 1);
```

The way arguments are supplied to the problem functions is by no means mandatory for this package; it is merely for convenience of the user. What exactly is meant by a "*smooth*" initial condition as in this example is easily found in the source code. The user is encouraged to make changes for him/herself.

The README file included in the main directory of the package contains more information regarding the problems implemented in this package, including the Allen–Cahn, Burgers, complex Ginzburg–Landau, Gray–Scott, Kuramoto–Sivashinsky, KdV and sine-Gordon equations.

4.5 φ functions

Central to the efficient implementation of exponential integrators is the evaluation of exponential-like functions, commonly denoted by φ functions in recent literature. We define the functions by the integral representation

$$\varphi_\ell(z) = \frac{1}{(\ell-1)!}\int_0^1 \mathrm{e}^{(1-\theta)z}\theta^{\ell-1}\,\mathrm{d}\theta, \qquad \ell = 1, 2, \ldots. \tag{4.8}$$

For small values of ℓ, with $z \neq 0$, these are

$$\varphi_1(z) = \frac{\mathrm{e}^z - 1}{z}, \quad \varphi_2(z) = \frac{\mathrm{e}^z - z - 1}{z^2}, \quad \text{and} \quad \varphi_3(z) = \frac{\mathrm{e}^z - z^2/2 - z - 1}{z^3}.$$

It is convenient to define $\varphi_0(z) = \mathrm{e}^z$, in which case the functions obey the recurrence relation

$$\varphi_\ell(z) = \frac{\varphi_{\ell-1}(z) - \frac{1}{(\ell-1)!}}{z}, \qquad \varphi_\ell(0) = \frac{1}{\ell!}, \qquad \ell = 1, 2, \ldots.$$

Evaluating these expressions numerically is not a trivial task. The accurate computation of φ_1 is a well-known problem in the numerical literature see [14, 16, 22], and problems arise from small values of z producing cancellation errors. For small values of z, expanding in Taylor series leads to accurate approximations, while for large values of z, the direct formula can be used. This approach was used by Cox and Matthews [9], but they noticed in some cases that in an interval close to the switching point, neither approximation was accurate.

Kassam and Trefethen [19] proposed to approximate with a contour integral, which worked well as long as the contour of integration was suitably chosen. However, the contour is generally problem dependent and difficult to determine in advance.

We propose evaluating the φ functions by an extension of the well-known matrix exponential scaling and squaring method, (method 3 of [24]). Using a scaled z sufficiently close to the origin, the φ function is calculated using a diagonal Padé approximant. The initial scaling is then undone by means of certain functional relations. This approach was previously used to compute φ_1 in [16] and the accompanying C and MATLAB software. The general form of the (d, d) Padé approximant of φ_ℓ is

$$\varphi_\ell(z) = \frac{N_d^\ell(z)}{D_d^\ell(z)} + \mathcal{O}(z^{2d+1}), \qquad z \neq 0,$$

where the unique polynomials N_d^ℓ and D_d^ℓ are

$$\begin{aligned} N_d^\ell(z) &= \frac{d!}{(2d+\ell)!} \sum_{i=0}^{d} \left[\sum_{j=0}^{i} \frac{(2d+\ell-j)!(-1)^j}{j!(d-j)!(\ell+i-j)!} \right] z^i, \\ D_d^\ell(z) &= \frac{d!}{(2d+\ell)!} \sum_{i=0}^{d} \frac{(2d+\ell-i)!}{i!(d-i)!} (-z)^i. \end{aligned} \tag{4.9}$$

When $\ell = 0$, these reduce to the well-known diagonal Padé approximations of the exponential function, see for example [5].

Reversing the scaling for $\ell > 0$ is more subtle than the case $\ell = 0$. In [3] a list of scaling relations for $\ell = 1, \ldots, 6$, is given. In general, one finds that the φ

functions obey the scaling relations

$$\varphi_{2\ell}(2z) = \frac{1}{2^{2\ell}}\left[\varphi_\ell(z)\varphi_\ell(z) + \sum_{j=\ell+1}^{2\ell}\frac{2}{(2\ell-j)!}\varphi_j(z)\right],$$
$$\varphi_{2\ell+1}(2z) = \frac{1}{2^{2\ell+1}}\left[\varphi_\ell(z)\varphi_{\ell+1}(z) + \sum_{j=\ell+2}^{2\ell+1}\frac{2}{(2\ell+1-j)!}\varphi_j(z) + \frac{1}{\ell!}\varphi_{\ell+1}(z)\right]. \tag{4.10}$$

The overall algorithm for computing φ functions may be summarized as follows:

1. Let p be the smallest integer such that $2^p \geq \|z\|_\infty$.
2. Set $z_{\text{scaled}} = z/2^{\max(0,p+1)}$.
3. Compute $\varphi_\ell(z_{\text{scaled}})$ using a (7,7) Padé approximant from (4.9).
4. Undo scaling by (4.10).

In Step 3, we choose the $(7,7)$ Padé approximation. This is chosen because it was more accurate than the $(6,6)$ choice promoted in [16] for φ_1. Note that the evaluation of φ_ℓ by scaling and modified squaring requires φ_j, for $j < \ell$, therefore the function implemented in this package will only take in ranges of φ functions to compute, starting with φ_1. A detailed analysis of the φ functions is a topic pursued in [2].

4.6 Included functions and scripts

We discuss here selected functions included in the package, and their usage. The header documentation in each script is usually more in-depth than presented here, but may be of a more technical nature.

`expglm.m`

The fundamental stepper routine for exponential integrators.

[t, u, varargout] = **expglm**(problem, tspan, h, cof)
[t, u, varargout] = **expglm**(problem, tspan, h, cof, timepoints);

This is the key interface function in this package. It is the function that integrates the differential equation (or the "problem") throughout a timespan with constant step size. If an additional vector of timepoints is supplied, the numerical solution at those points will be returned. Each element in `timepoints` should be an integer multiple of `h`. The optional third output variable is a processed solution value, determined by the function named in the problems structure-variable

`postprocessing`. `expglm` also has a callback functionality in case one wants to calculate a function, or plot some function (energy) of the solution as `expglm` integrates through time. This uses the field `outputfcn` in the problem-structure, Section 4.4. The `nls`-problem contains two examples on usage of this callback functionality.

`phipade.m`

Computes approximations to φ functions.

```
phi_k = phipade(z, k);
[phi_1, phi_2, ..., phi_k] = phipade(z, k);
```

This is a function crucial to all schemes using φ functions. It computes φ_ℓ for $\ell \geq 1$ (4.8) by implementing the algorithm sketched in Section 4.5. For efficiency, especially in order-tests, and also for compatibility with variable step size implementation, additional memory resources are expended in caching recently computed function values. The function value cache is organized in an LRU (Least Recently Used) fashion and, unfortunately, increases the code complexity of `phipade`. Caching is turned on by default but may be toggled with the use of the `wantcache` function or by setting a variable `wantcache` within `phipade`. The function is entirely self-contained with respect to use of other functions and thus, it may be pulled out of the EXPINT package and used independently if necessary.

`globalorder.m`

Performs a global order test.

```
schemes = globalorder(problem, tspan, dt, schemes, varargin)
```

Input arguments include the problem defining the equation, the timespan, a vector of timesteps to test and the schemes to test. Optionally, one may provide a string with a scheme-name which will be used for the reference solution. The default is to use `ode15s`. It is important to build up a sensible vector of timesteps to be used, *all timesteps used should divide the length of the timespan so that all stepsizes are constant throughout each test.* An example is

```
tspan = [0, 1];
steplist = round(10.^(4:-0.1:0));
dt = unique(diff(tspan)./steplist);
```

The function returns the cell-structure of schemes, but with error-data added (and also timings, measured using `cputime`). This data is used by `orderplot` and `timingplot`.

localorder.m

Performs a local order test.

schemes = **localorder**(problem, dt, schemes, varargin)

This file should only be used for exponential Runge–Kutta type schemes. Arguments are as for `globalorder`, but now the vector of timesteps must be built up differently. *The smallest timestep must divide all other timesteps used*, or else the computation will take an unnecessarily long time. An example is

```
steplist  = round(10.^(4:-0.1:1));
dt = unique(10^-4 .* steplist);
```

Error and timing data is added as for `globalorder`.

orderplot.m

Makes a plot of experimental order.

varargout = **orderplot**(schemes)
varargout = **orderplot**(schemes, thexlabel, theylabel, thetitle)

This function produces a plot of local or global error versus timestep. Both error and timesteps are read from the `schemes` array.

timingplot.m

Makes a plot of timing results.

varargout = **timingplot**(schemes)
varargout = **timingplot**(schemes, thexlabel, theylabel, thetitle)

This function works almost identically to `orderplot.m`, although it plots (global) error as a function of time spent for each time step and for each scheme.

orderline.m

Plots a dotted straight line to show order.

orderline(**gradient**)
orderline(**gradient**, [h, err])

This is a small utility function for plotting a dotted straight line in a loglog-plot along with a number representing the gradient of the line. This is useful to add in an orderplot to visualize the observed order of the schemes. With only one argument, the function is interactive, waiting for the user to click with the mouse in a plot specifying one point the line is to go through. The function prints out a vector that can be used as `[h, err]` the next time the function is used. The function is non-interactive when `[h, err]` is supplied.

surfplot.m

Surface plot of numerical solution.

varargout = **surfplot**(data, timepoints, problem, varargin)

Produces a surface plot of the numerical solution. In this case, one should use the "processed" solution of the problem if applicable, see `expglm`, Section 4.6. A typical usage could be:

```
[t, u, up] = expglm(problem, tspan, h, 'lawson4', timepoints);
surfplot(up, timepoints, problem);
```

4.7 Illustrative examples

In this section we illustrate some typical uses of the EXPINT package.

Global order with timing plot

The global order test provides a simple way of producing plots which can be used for comparing the quality of certain schemes on a particular problem. We show here one representative example of a global error plot, the Kuramoto–Sivashinsky equation in 1D. Both global error as a function of timestep and error as a function of time spent are shown in Figure 4.1.

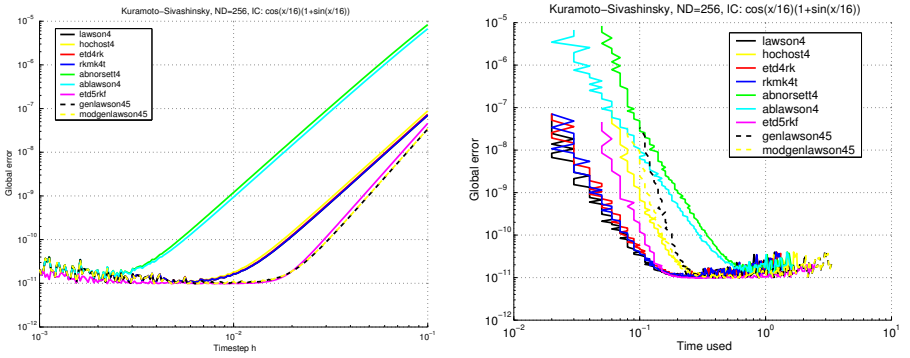
The global error plot in Figure 4.1a displays an instability of magnitude `eps`$/h$. This is due to roundoff error accumulation from T/h repeated multiplications of $\exp(hL)$. This instability can be isolated to the repeated multiplications of $\exp(hL)$, as it also occurs in the Lawson4 scheme, which only uses the exponential function (evaluated using the MATLAB built in function `expm`). We note that as with the φ functions, evaluating the quantity $\exp(hL)$ is only done once, at process startup. This larger accumulation of roundoff error compared to classical numerical integrator, is, therefore, a numerical artifact of all exponential integrators. Minimizing the effect of roundoff is the subject of ongoing investigations.

Surface plot

The following sequence of commands generates the plot in Figure 4.2a.

```
problem = kursiv('ND', 256);
[t, u, up] = expglm(problem, [0 100], 0.1, 'lawson4',  [0:0.2:100]);
surfplot(up, t, problem);
```

Here we use the third optional output argument from `expglm`, which returns post-processed output determined by the problem. In this case for the Kuramoto–Sivashinsky equation, it does the inverse fourier transform and selects only the



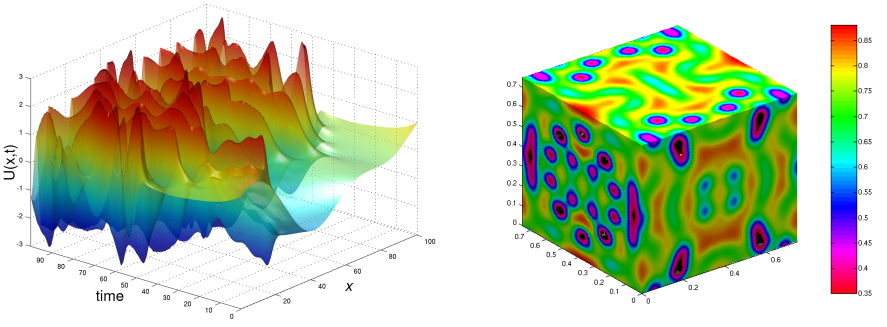
(a) Global error

(b) Error as a function of computational time

Figure 4.1: Example test run for the Kuramoto–Sivashinsky equation. Code used is given in the file `tests/globorderkursiv.m` in the package distribution

real part. We also specifically instruct `expglm` to return the solution values at certain points.

A similar procedure produces the plot in Figure 4.2b. However, as this procedure requires a bit of extra manipulation to generate the desired graphic effects, we refer the reader to the `testgrayscott3d` script implemented in `tests/` directory for additional details.



(a) Kuramoto–Sivashinsky 1D $ND = 256$, Initial condition: $\cos(x/16)(1 + \sin(x/16))$.

(b) 3D plot, Gray–Scott, $ND = 48$.

Figure 4.2: Example surface plots

4.8 Discussion

The EXPINT package was designed as a tool for facilitating easy testing and comparison of exponential integrators (and approximations to them) applied to semilinear problems. The function which performs the integration, known as `expglm`, implements the exponential general linear method described in Section 4.2. While such an implementation comes to some extent at the cost of efficiency, we believe this to be justified given our aims of providing usability. The package includes many well known exponential integrators (and approximations to them) and some not so familiar ones. Additionally, several semi-discretizations of well-known PDEs have been included. We encourage the user to add their own integrators and problems.

An important part of the EXPINT package is the evaluation of the φ functions. There are various problems associated with alternative approaches used to evaluate these functions. Combining scaling and squaring with Padé-approximations has become the standard approach in numerical software like MATLAB for computing the matrix exponential. In the self-contained function `phipade`, a scaling and corrected squaring combined with diagonal Padé approximations enables efficient evaluation of the φ functions. The Padé coefficients in Section 4.5 have not been published elsewhere as far as we can determine.

Our implementation of `expglm` is with a fixed step size only. A variable step size implementation is possible by replacing this file. The evaluation of the φ functions quickly becomes the crux of the computation for variable step size. A sensible implementation should make use of the scaling relations for φ functions given in (4.10), possibly through modifications for specialized efficiency in `phipade`. It should also be noted that `phipade` already supports variable step size through a Least Recently Used-cache functionality, but due to possible memory constraints this should be used with care. The cache size is easily changed in the source code.

A way to overcome the higher computational cost arising from the change of the stepsize is to take advantage of the fact that when implementing any exponential integrator, we do not really need to compute the φ functions. What we require is their action on a given state vector v. Krylov subspace approximations to the exponential and various related functions have been studied by many authors, see for example [13, 15, 28]. The main idea is to approximately project the action of the function $\varphi_\ell(hL)$ on a state vector v, to a smaller Krylov subspace. The computations are then performed in this subspace, typically of much smaller dimension than the original problem. This technique has been used effectively for exponential integrators; we cite the papers [12, 16]. Special considerations should be taken in both the design of the schemes and the choice of stepsizes to ensure an efficient implementation.

It is an area of future research to find out whichever approach to computing φ

functions that provides the most versatile platform for exponential integrators, both in terms of computational speed, memory requirements, ease of implementation and restrictions in choice of stepsizes.

Our hope is that this package is useful in testing and comparing exponential integrators and will encourage the use of these novel integrators in the future.

Bibliography

[1] Berland, H., Owren, B., Skaflestad, B.: *B*-series and order conditions for exponential integrators. SIAM J. Numer. Anal. **43**(4), 1715–1727 (2005)

[2] Berland, H., Skaflestad, B., Wright, W.M.: Scaling and squaring of φ functions in exponential integrators (2006). In preparation

[3] Beylkin, G., Keiser, J.M., Vozovoi, L.: A new class of time discretization schemes for the solution of nonlinear PDEs. J. of Comp. Phys. **147**, 362–387 (1998)

[4] Burrage, K., Butcher, J.C.: Nonlinear stability for a general class of differential equation methods. BIT **20**, 185–203 (1980)

[5] Butcher, J.C.: Numerical methods for ordinary differential equations. John Wiley & Sons Ltd., Chichester, UK (2003)

[6] Calvo, M.P., Palencia, C.: A class of multistep exponential integrators for semilinear problems. Numer. Math. **102**, 367–381 (2006)

[7] Celledoni, E., Marthinsen, A., Owren, B.: Commutator-free Lie group methods. FGCS **19**(3), 341–352 (2003)

[8] Certaine, J.: The solution of ordinary differential equations with large time constants. In: Mathematical methods for digital computers, pp. 128–132. Wiley, New York (1960)

[9] Cox, S.M., Matthews, P.C.: Exponential time differencing for stiff systems. J. Comput. Phys. **176**(2), 430–455 (2002)

[10] Ehle, B.L., Lawson, J.D.: Generalized Runge–Kutta processes for stiff initial-value problems. J. Inst. Maths. Applics. **16**, 11–21 (1975)

[11] Friedli, A.: Verallgemeinerte Runge–Kutta Verfahren zur Lösung steifer Differentialgleichungssysteme. In: Numerical treatment of differential equations, pp. 35–50. Lecture Notes in Math., Vol. 631. Springer, Berlin (1978)

[12] Friesner, R.A., Tuckerman, L.S., Dornblaser, B.C., Russo, T.V.: A method for the exponential propagation of large stiff nonlinear differential equations. J. Sci. Comput. **4**(4), 327–354 (1989)

[13] Gallopoulos, E., Saad, Y.: Efficient solution of parabolic equations by Krylov approximation methods. SIAM J. Sci. Statist. Comput. **13**, 1236–1264 (1992)

[14] Higham, N.J.: Accuracy and stability of numerical algorithms, second edn. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA (2002)

[15] Hochbruck, M., Lubich, C.: On Krylov subspace approximations to the matrix exponential operator. SIAM J. Numer. Anal. **34**(5), 1911–1925 (1997)

[16] Hochbruck, M., Lubich, C., Selhofer, H.: Exponential integrators for large systems of differential equations. SIAM J. Sci. Comput. **19**(5), 1552–1574 (1998)

[17] Hochbruck, M., Ostermann, A.: Explicit exponential Runge–Kutta methods for semilinear parabolic problems. SIAM J. Numer. Anal. **43**(3), 1069–1090 (2005)

[18] van der Houwen, P.J.: Construction of integration formulas for initial value problems. North-Holland Publishing Co., Amsterdam (1977). North-Holland Series in Applied Mathematics and Mechanics, Vol. 19

[19] Kassam, A.K., Trefethen, L.N.: Fourth-order time-stepping for stiff PDEs. SIAM J. Sci. Comput. **26**(4), 1214–1233 (electronic) (2005)

[20] Krogstad, S.: Generalized integrating factor methods for stiff PDEs. J. of Comp. Phys. **203**(1), 72–88 (2005)

[21] Lawson, J.D.: Generalized Runge–Kutta processes for stable systems with large Lipschitz constants. SIAM J. Numer. Anal. **4**, 372–380 (1967)

[22] Lu, Y.Y.: Computing a matrix function for exponential integrators. J. Comput. Appl. Math **161**(1), 203–216 (2003)

[23] Minchev, B., Wright, W.M.: A review of exponential integrators for semilinear problems. Tech. Rep. 2/05, Department of Mathematical Sciences, NTNU, Norway (2005). `http://www.math.ntnu.no/preprint/`

[24] Moler, C.B., van Loan, C.F.: Nineteen dubious ways to compute the exponential of a matrix, twenty five years later. SIAM Review **45**(1), 3–49 (2003)

[25] Munthe-Kaas, H.: High order Runge–Kutta methods on manifolds. In: Proceedings of the NSF/CBMS Regional Conference on Numerical Analysis of Hamiltonian Differential Equations (Golden, CO, 1997), vol. 29,1, pp. 115–127 (1999)

[26] Nørsett, S.P.: An A-stable modification of the Adams–Bashforth methods. In: Conf. on Numerical Solution of Differential Equations (Dundee, Scotland, 1969), pp. 214–219. Springer, Berlin (1969)

[27] Ostermann, A., Thalhammer, M., Wright, W.M.: A class of explicit exponential general linear methods. BIT **46**(2), 409–432 (2006)

[28] Saad, Y.: Krylov subspace methods for solving large unsymmetric linear systems. Math. Comp. **37**, 105–126 (1981)

[29] Strehmel, K., Weiner, R.: Behandlung steifer Anfangswertprobleme gewöhnlicher Differentialgleichungen mit adaptiven Runge–Kutta-Methoden. Computing **29**(2), 153–165 (1982)

[30] Strehmel, K., Weiner, R.: B-convergence results for linearly implicit one step methods. BIT **27**, 264–281 (1987)

[31] Thalhammer, M., Wright, W.M.: Generalized Lawson methods (2006). In preparation

Paper IV

Håvard Berland, Brynjulf Owren, and Bård Skaflestad:

Solving the nonlinear Schrödinger equation using exponential integrators

To appear in Modeling, Identification and Control, 27(4), 2006.

Chapter 5

Solving the nonlinear Schrödinger equation using exponential integrators

Abstract:

Using the notion of integrating factors, Lawson developed a class of numerical methods for solving stiff systems of ordinary differential equations. However, the performance of these "Generalized Runge–Kutta processes" was demonstrably poorer when compared to the ETD schemes of Certaine and Nørsett, recently rediscovered by Cox and Matthews. The deficit is particularly pronounced when the schemes are applied to parabolic problems. In this paper we compare a fourth order Lawson scheme and a fourth order ETD scheme due to Cox and Matthews, using the nonlinear Schrödinger equation as the test problem. The primary testing parameters are degree of regularity of the potential function and the initial condition, and numerical performance is heavily dependent upon these values. The Lawson and ETD schemes exhibit significant performance differences in our tests, and we present some analysis on this.

5.1 Introduction

Although not new, exponential integrators were not considered a practical means of resolving systems of ordinary differential equations until very recently. Exponential integrators are especially designed to handle stiff systems, and accomplish this goal by constructing exact integral curves for the linear part of the differential operator. Constructing the integral curves entails the application of the matrix exponential and related functions.

The class of integrators henceforth termed exponential integrators first appeared in Certaine [5] and Nørsett [16]. These schemes are both of exponential time differencing (ETD) type. Then Lawson [14] constructed the integrating factor type in 1967. Recent reports on exponential integrators show that especially for parabolic semi-linear problems, the ETD type of exponential integrators outperform integrators of Lawson type [13, 15, 17]. However, few results are available with respect to the performance on non-parabolic problems like the Schrödinger equation.

In this paper we test a fourth order Lawson integrating factor scheme against

a fourth order ETD scheme, ETD4RK in [6]. Most other similar exponential integrator schemes perform very similarly to the ETD4RK-scheme. Exponential integrators are introduced in Section 5.2 and some analysis and numerical results are presented in Sections 5.4 and 5.5.

The equation we will use for numerical tests in this paper is the nonlinear Schrödinger equation in one space dimension

$$\begin{aligned} \mathrm{i}\psi_t &= -\nabla^2\psi + (V(x) + \lambda|\psi|^2)\psi \\ \psi(-\pi, t) &= \psi(\pi, t), \qquad \text{for all } t \geq 0 \\ \psi(x, 0) &= \psi_0(x), \qquad x \in [-\pi, \pi]. \end{aligned} \tag{5.1}$$

This Schrödinger equation arises in several different areas of physics of which we mention multiscale perturbation theory, gravity waves in water, and propagation of intense optical pulses in fibers. The nonlinearity constant λ controls the ratio of dispersive effects to nonlinear effects, and may give a focusing version of the equation. The equation may be both parabolic and hyperbolic, it has some smoothing effects, but time-reversibility prevents it from generating an analytic semigroup, which is fundamental for the stiff order analysis in Section 5.3. An introduction to the mathematical theory of the nonlinear Schrödinger equation is given in [4].

We would like to point out that we do not try to directly preserve any invariants of the equations in question, as opposed to many other specialized schemes for the Schrödinger equation. In this work, we test the given schemes on a limited time scale, and focus on reporting the observed order. The Schrödinger equation possesses several conservation laws, notably conservation of density, energy and momentum. For long-time integration where stability and preservation of invariants is an important factor, multisymplectic schemes may be a viable choice [3, 10]. The benefits of preservation of invariants must be weighted against the additional cost necessary for multisymplectic schemes.

For our Schrödinger equation (5.1) we will employ a discrete Fourier transform with $N_{\mathscr{F}}$ modes. Upon semi-discretizing the physical problem in space, we obtain a system of ordinary differential equations given by

$$\dot{y} = Ly + N(y) \tag{5.2}$$

in which $y \in \mathbf{C}^n$ is the Fourier transform of ψ, $L \in \mathbf{C}^{n\times n}$, and $N : \mathbf{C}^n \to \mathbf{C}^n$. For the Schrödinger equation (5.1) the L matrix becomes diagonal with entries

$$L_{kk} = -\mathrm{i}k^2, \qquad k = -\tfrac{N_{\mathscr{F}}}{2} + 1, \ldots, \tfrac{N_{\mathscr{F}}}{2} \tag{5.3}$$

and the nonlinear function $N(y)$ becomes

$$N(y) = -\mathrm{i} \cdot \mathscr{F}\big((V(x) + \lambda|\mathscr{F}^{-1}(y)|^2)\mathscr{F}^{-1}(y)\big) \tag{5.4}$$

in which each component of y represents a particular Fourier mode, k.

5.2 Exponential integrators

Exponential integrators are explicit schemes which recover the exact solution to linear problems. As such, this class of schemes is well suited to problems which can be split into a linear and a nonlinear part, and for which the linear part is either stiff or unbounded and the nonlinear part grows more slowly than the linear part. When semi-discretizing PDEs, this happens if spatial derivatives in the linear part are of higher order than in the nonlinear part. We note that the Schrödinger equation, whether semi-discretized as in (5.2) or in its original PDE form (5.1), satisfies these requirements although the matrix L of (5.2) is unbounded only in terms of the parameter $N_{\mathscr{F}}$.

In the following, we consider systems of ordinary differential equations split into a linear and a nonlinear part as

$$\dot{y} = Ly + N(y,t), \qquad y(0) = y_0. \tag{5.5}$$

As alluded to in the above paragraph, exponential integrators applied to this problem possess two primary features

1. If $L = 0$, the integrator reduces to a classical Runge–Kutta or linear multistep method.

2. If $N(y,t) = 0$ for all y and t, the integrator reproduces the exact solution to (5.5).

The nonlinear function N may depend on time, but the linear part should not be explicitly time dependent in order for the exponential integrator to be computationally competitive. Moreover, exponential integrators implicitly assume that most of the system's inherent dynamic behaviour can be ascribed to the linear operator L.

Classical integrators are divided into two classes; linear multistep methods and one-step Runge–Kutta methods. This paper considers only exponential Runge–Kutta methods. We note that the framework of general linear methods, a generalization of both linear multistep methods and Runge–Kutta methods, may also be extended to define exponential integrators as in [2].

Exponential integrators of Runge–Kutta type are written as

$$Y_i = \sum_{j=1}^{s} a_{ij}(hL)\, hN(Y_j, t_0 + c_j h) + \exp(c_i hL) y_0 \tag{5.6a}$$

$$y_1 = \sum_{i=1}^{s} b_i(hL)\, hN(Y_j, t_0 + c_j h) + \exp(hL) y_0 \tag{5.6b}$$

in which Y_i, $i = 1, \ldots, s$ are internal stages and y_1 is the final approximation of $y(t_1) = y(t_0 + h)$. This format extends the common format of Runge–Kutta schemes in that the coefficients a_{ij} and b_i are now analytic functions of the linear operator L.

In order to fulfill the two features of an exponential integrator, $a_{ij}(0)$ and $b_i(0)$ must be the coefficients of some *underlying* Runge–Kutta-method. It is evident that this scheme will solve linear equations ($N(y,t) = 0$) exactly. Extending the notation of Butcher, the coefficient functions and collocation nodes are written up in the tableau

$$\begin{array}{c|ccc} c_1 & a_{11}(z) & \cdots & a_{1s}(z) \\ \vdots & \vdots & & \vdots \\ c_s & a_{s1}(z) & \cdots & a_{ss}(z) \\ \hline & b_1(z) & \cdots & b_s(z) \end{array} \tag{5.7}$$

where we have used $z = hL$ for convenience.

The two simplest choices of exponential integrators of Runge–Kutta type are the Lawson–Euler

$$y_n = \exp(hL)y_{n-1} + h\exp(hL)N(y_{n-1}, t_{n-1}), \tag{5.8}$$

$$\begin{array}{c|c} 0 & 0 \\ \hline & \mathrm{e}^z \end{array}$$

and Nørsett–Euler

$$y_n = \exp(hL)y_{n-1} + h\varphi_1(hL)N(y_{n-1}, t_{n-1}), \tag{5.9}$$

$$\begin{array}{c|c} 0 & 0 \\ \hline & \varphi_1(z) \end{array}$$

schemes. The function $\varphi_1(z)$ in the Nørsett–Euler scheme is given by $\varphi_1(z) = (\mathrm{e}^z - 1)/z$. The latter scheme has been reinvented several times, and is also known as ETD Euler, filtered Euler, Lie–Euler (using the affine Lie group) and exponentially fitted Euler.

Lawson schemes

The Lawson exponential integrators, of which Lawson–Euler is a special case, are derived by introducing a change of variables involving an integrating factor and applying a classical Runge–Kutta scheme to the transformed equation. Given an underlying Runge–Kutta scheme with coefficients $\tilde{a}_{ij}$, $\tilde{b}_i$ and corresponding quadrature nodes c_i, the Lawson exponential integrator coefficient functions are as given in [14]

$$a_{ij}(z) = \tilde{a}_{ij}\mathrm{e}^{(c_i - c_j)z} \quad \text{and} \quad b_i(z) = \tilde{b}_i\mathrm{e}^{(1-c_i)z}. \tag{5.10}$$

Lawson schemes are particularly simple to implement, but have some disadvantages as reported early in the history of exponential integrators. For example, they do not preserve fixed points of the differential equation, and are also known for rather large error constants.

The aim of this paper is to elaborate on the performance of a Lawson exponential integrator based on Kutta's classical fourth order method. This scheme is given by the tableau

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2}\mathrm{e}^{z/2} & & & \\ \frac{1}{2} & & \frac{1}{2} & & \\ 1 & & & \mathrm{e}^{z/2} & \\ \hline & \frac{1}{6}\mathrm{e}^{z} & \frac{1}{3}\mathrm{e}^{z/2} & \frac{1}{3}\mathrm{e}^{z/2} & \frac{1}{6} \end{array} \tag{5.11}$$

and will be denoted "Lawson4".

Ehle and Lawson modified the Lawson schemes in their paper [7] and introduced another fourth order exponential integrator also using the φ_1 function, thereby slightly improving the performance for parabolic applications and regaining fixed point preservation. Their modification was in the direction of ETD-schemes, but it is not competitive to the now known ETD-schemes.

Exponential time differencing (ETD)

Rather than using integrating factors, we may approximate the nonlinear function $N(y,t)$ by some polynomial in t, and integrate the approximate equation exactly. The resulting schemes are known in recent literature as the "exponential time differencing" (ETD) schemes, although this name is not entirely descriptive. The polynomial approximation may be calculated using previous steps of the integration process, thus producing multistep ETD schemes, or by Runge–Kutta-like stages, resulting in ETD schemes of Runge–Kutta type. We refer the reader to the review paper [15] for a thorough review of exponential integrators of these types.

For notational simplicity, and without loss of generality, we consider only autonomous problems $N(y) = N(y(t))$ in the remainder of this paper.

Lemma 5.1. *The exact solution of the initial value problem*

$$\dot{y}(t) = Ly(t) + N(y(t)), \qquad y(0) = y_0,$$

has the expansion

$$y(t) = \mathrm{e}^{tL}y_0 + \sum_{i=1}^{\infty} \varphi_i(tL)t^i N^{(i-1)}(y_0).$$

where

$$\varphi_i(z) = \frac{1}{(i-1)!}\int_0^1 \mathrm{e}^{(1-\theta)z}\,\theta^{i-1}\,\mathrm{d}\theta. \tag{5.12}$$

Proof. The basic idea is just a Taylor expansion of the nonlinear function $N(y(t))$ and the variation of constants formula. A proof may be found in [13, Lemma 1.1]. □

We will in this paper compare the Lawson4 scheme (5.11) against the most commonly used fourth order ETD scheme, ETD4RK, due to Cox and Matthews [6]. The coefficients of ETD4RK are given by

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2}\varphi_1(\frac{z}{2}) & & & \\ \frac{1}{2} & & \frac{1}{2}\varphi_1(\frac{z}{2}) & & \\ 1 & \varphi_1(\frac{z}{2})(\mathrm{e}^{z/2}-1) & & \varphi_1(\frac{z}{2}) & \\ \hline & b_1(z) & b_2(z) & b_3(z) & b_4(z) \end{array} \tag{5.13}$$

in which

$$\begin{aligned} b_1(z) &= \varphi_1(z) - 3\varphi_2(z) + 4\varphi_3(z) \\ b_2(z) = b_3(z) &= 2\varphi_2(z) - 4\varphi_3(z) \\ b_4(z) &= -\varphi_2(z) + 4\varphi_3(z). \end{aligned}$$

Computationally, the Lawson4 scheme (5.11) is much cheaper and easier to implement on a computer than ETD4RK. The evaluation of φ functions in (5.12) has numerical issues, and we believe this is best dealt with using scaling and corrected squaring together with Padé approximants. Details on this may be found in [2].

5.3 Order conditions

Classical order analysis for numerical integrators develops Taylor expansions for all quantities. The analysis, however, is rigorous and valid only in the limit as $hL \to 0$. If L is defined by spatially semi-discretizing an unbounded differential operator $\mathcal{L}$, L may be unbounded in terms of a parameter, typically the spatial resolution. Thus, $hL \to 0$ cannot generally be guaranteed independently of the parameter. As such, classical order analysis is of somewhat limited use in the study of exponential integrators applied to unbounded semi-linear problems. Nevertheless, classical order conditions must be satisfied for exponential integrators and traditional Runge–Kutta integrators alike. The Lawson4 (5.11) and ETD4RK (5.13) schemes are methods with classical order four. Details on

classical order analysis for exponential integrators using B-series may be found in the paper [1].

A recent paper of Hochbruck and Ostermann [9] studies exponential integrators applied to infinite dimensional semi-linear parabolic Cauchy problems. Conditions under which the integrators converge in this abstract setting are rather restrictive, and give rise to the notion of stiff order. Including the classical order conditions as special cases, these "stiff order conditions" constitute an extended set of requirements which must be satisfied to guarantee high convergence rates. In this context Lawson4 (5.11) has stiff order only 1 and ETD4RK (5.13) has stiff order only 2. The use of φ functions in the coefficient functions $a_{ij}(z)$ and $b_i(z)$ of (5.6) is required to attain high stiff order.

However, the applicability of stiff order analysis to the nonlinear Schrödinger equation remains an open issue. Integrators of stiff order four, examples of which are listed in [2], perform similarly to ETD4RK in this study. This suggests that high stiff order is not critical to achieving efficient schemes in all cases, and these high stiff order schemes are therefore omitted in all plots.

The first stiff order condition for an exponential integrator is easily obtained by comparing the numerical solution given in (5.6) to the exact solution from Lemma 5.1. For the first order in h we get the equation

$$\varphi_1(z)\, hN(y(t_0)) - \sum_{i=1}^{s} b_i(z)\, hN(y(t_0)) = 0,$$

which we rewrite as

$$\psi_1(z)\, hN(y(t_0)) = 0. \tag{5.14}$$

Based on this, the first stiff order conditions reads

$$\psi_1(z) = \sum_{i=1}^{s} b_i(z) - \varphi_1(z) = 0. \tag{5.15}$$

The Lawson integrators do *not* satisfy this condition exactly, but the integrators nevertheless satisfy the condition to a sufficient degree of accuracy, a notion which will be explained in Section 5.4. There we study the solution's dependence upon the Schrödinger equation potential function $V(x)$.

An easy route to deriving two stiff order conditions is considering preservation of fixed points. Exact preservation of fixed points is important in many applications, and hence a desirable property of exponential integrators. Requiring $Ly = -N(y)$ and $y_1 = y_0$, equation (5.6b) gives

$$y_0 = -\sum_{i=1}^{s} b_i(z) z + \mathrm{e}^z y_0$$

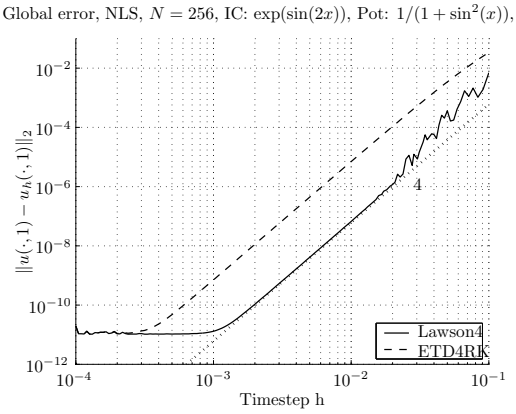


Figure 5.1: A global order test. Both exponential integrators in this study perform as order 4 integrators. The dotted line is only an indicator line showing how order 4 looks like.

equivalent to

$$\sum_{i=1}^{s} b_i(z) = \varphi_1(z). \tag{5.16}$$

For equation (5.6a) we require $Y_i = y_0$ for all i, and we obtain

$$\sum_{j=1}^{s} a_{ij}(z) = c_i \varphi_1(c_i z) \quad \text{for each } i. \tag{5.17}$$

These are precisely the first and third stiff order conditions in [9]. Lawson integrators fulfill neither of these conditions, and thus do not preserve fixed points. ETD4RK, however, fulfills both conditions for fixed point preservation.

Despite their low stiff order, Lawson4 (stiff order 1) and ETD4RK (stiff order 2) still behave as fourth order schemes on our problem, given smooth initial condition and smooth potential. See Figure 5.1.

5.4 Potential function dependency

The first stiff order condition (5.15) is not satisfied by the Lawson schemes. The significance of the stiff order conditions in the case of non-parabolic problems like the Schrödinger equation is unclear, but the conditions still affect the numerical performance in some cases. Figure 5.1 shows that the Lawson scheme is roughly

100 times more accurate than ETD4RK at comparable step sizes, however, as we will justify, the performance results in Figure 5.1 are strongly influenced by the smoothness of the potential function used in this particular test.

In this section we study how the regularity of the potential function $V(x)$ affects the numerical performance of the Lawson4 integrator.

Order estimates

The analysis will be based on the rate of decay for the Fourier coefficients of input functions. The relationship between Fourier decay and differentiability is taken from a well-known result in Fourier analysis.

Lemma 5.2. *If a function f is r times differentiable, that is, $f^{(r)} \in L^1$, then*

$$|\hat{f}(k)| \leq \frac{\|f^{(r)}\|_{L^1}}{|k|^r}, \qquad k \in \mathbf{Z}\backslash\{0\} \tag{5.18}$$

and $|\hat{f}(0)| \leq \|f^{(r)}\|_{L^1}$.

We estimate the error contribution from the first stiff order condition (5.15) in Fourier space. By substitution of the $b_i(z)$ of Lawson4 into (5.15) we obtain

$$\psi_1(z) = \frac{\mathrm{e}^z - 1}{z} - \frac{1}{6}\mathrm{e}^z - \frac{2}{3}\mathrm{e}^{z/2} - \frac{1}{6}$$

which, when z is small, has the Taylor expansion

$$-\frac{1}{2880}z^4 + \mathcal{O}(z^5). \tag{5.19}$$

For $x \in \mathbf{R}$ we have $|\psi_1(x\mathrm{i})| \leq 2$, and we use this to construct an upper bound for ψ_1 for high Fourier modes where the Taylor expansion (5.19) is not valid. Let the bound be the function

$$\psi_{1,\mathrm{env}} = \begin{cases} \frac{(hk^2)^4}{2880} & |k| \leq (2 \cdot 2880)^{1/8} h^{-1/2} \\ 2 & |k| > (2 \cdot 2880)^{1/8} h^{-1/2} \end{cases} \tag{5.20}$$

which is sufficiently sharp for our purpose.

Proposition 5.3. *If*

1. *$\psi_1(z) = -Cz^p + \mathcal{O}(z^{p+1})$ for z small.*
2. *$N(y(t))$ (in Fourier space) in (5.2) has a decay rate of at least r for all time and $\frac{1}{2} < r \leq 2p$, that is*

$$N(y(t))(k) \leq \frac{K_N}{|k|^r}$$

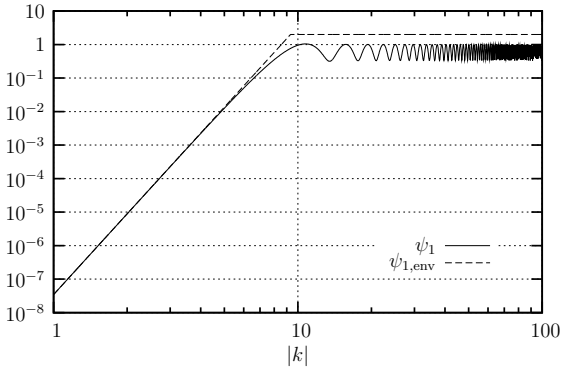


Figure 5.2: The error in the first stiff order condition for Lawson4, $h = 0.1$ in this plot.

3. $hk_{max}^2 \gg 1$

then the local error contribution for the first stiff order condition is

$$\|h\psi_1(-\mathrm{i}k^2)N(k)\|_2 = C^* h^{1+\frac{r}{2}-\frac{1}{4}} + \mathcal{O}(h^{1+\frac{r}{2}+\frac{1}{4}}) \tag{5.21}$$

Proof. We bound the $\psi_1(-\mathrm{i}k^2)$ function by

$$\psi_{1,\mathrm{env}}(-\mathrm{i}k^2) = \begin{cases} C(hk^2)^p & |k| \le k_c \\ 2 & |k| > k_c \end{cases}$$

where the critical mode value is $k_c = (2C)^{\frac{1}{2p}} h^{-1/2}$. To estimate the error, we sum over k in the first stiff order condition

$$\|\psi_1\, hN\|_2^2 \le h^2 \sum_{|k|\le k_c} C^2 (hk^2)^{2p} K_N^2 k^{-2r} + h^2 \sum_{k_c<|k|\le k_{max}} 4K_N^2 k^{-2r},$$

in which we estimate the sums using Euler–MacLaurin's summation formula $\sum_{k=1}^{n} f(k) = \int_0^n f(x)\,\mathrm{d}x + \frac{1}{2}\left(f(n) - f(0)\right) + \tilde{R}_1$ where $|\tilde{R}_1| \le \frac{1}{2}\int_0^n |f'(x)|\,\mathrm{d}x$, for any function $f \in C^1([0,n])$, so

$$\|\psi_1\, hN\|_2^2 \le 2h^2 K_N^2 \Big(C^2 h^{2p} \big(\frac{k_c^{4p-2r+1}}{4p-2r+1} + k_c^{4p-2r} \big) + 8\big(\frac{k_{max}^{1-2r}}{1-2r} - \frac{k_c^{1-2r}}{1-2r} + k_{max}^{-2r} - k_c^{-2r} \big) \Big).$$

Inserting $k_c = (2C)^{\frac{1}{2p}} h^{-1/2}$ we get the dependency on h, and the square root of the dominating term is $h^{1+\frac{r}{2}-\frac{1}{4}}$ as long as $N_{\mathscr{F}} = 2k_{max}$ is large enough and $1/2 < r \leq 2p$. □

If $r > 2p$, the scheme is not accurate enough to capture the "non-smoothness" of the N-function, and the first order condition does not contribute to any error of order less than the classical error.

Thus, as long as the nonlinear function is smooth enough, we can also include Lawson4 as one of the schemes that obey the first stiff order condition, although only accurately enough so that its main features as a fourth order classical method is conserved. Looking at only the first stiff order condition is sufficient for explaining the observed numerical behaviour in this paper.

Numerical results

In the following experiments, we have used an artificially constructed potential with a prescribed decay rate r. This means constructing the potential by letting its Fourier modes be $1/(\mathrm{i}k^r)$ multiplied with a complex number in which both the real and the imaginary part are normally distributed with mean zero and variance one. Then we have used MATLAB's inverse discrete Fourier transform to get an example function for use. We note that in particular the hat function has a decay rate of 2, although Lemma 5.2 only predicts 1. This is due to bounded variation of the hat function.

Figures 5.3 and 5.4 show observed error behaviour when solving the nonlinear Schrödinger equation subject to a smooth initial condition and potential functions of regularity 2 and 4 respectively. Low regularity potential functions lower the regularity of the nonlinear function $N(y(x,t))$. Assuming N is no more regular than $V(x)$, Proposition 5.3 then predicts orders 1.75 and 2.75 respectively for the Lawson4 scheme in these cases. We conclude that the observed order corresponds fairly well to what is predicted by the proposition. Moreover, we see from the plots that for the Lawson schemes, the global error as a function of time step oscillates rather wildly when not all eigenmodes are resolved by a small enough h. These oscillations are smooth on a zoomed plot and are due to some resonance effect. This is further discussed in Section 5.6.

5.5 Initial condition dependency

In this section we will see that the ETD4RK scheme is more influenced by the regularity of the initial condition than is the Lawson4 scheme. A crucial introductory numerical observation is that the dependency on the initial condition is

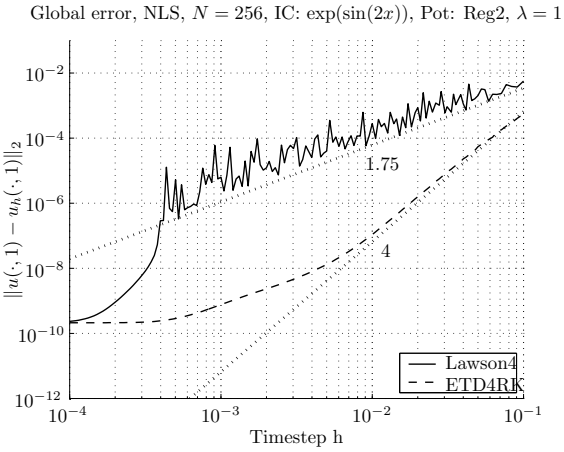


Figure 5.3: Global error when the potential has regularity 2.

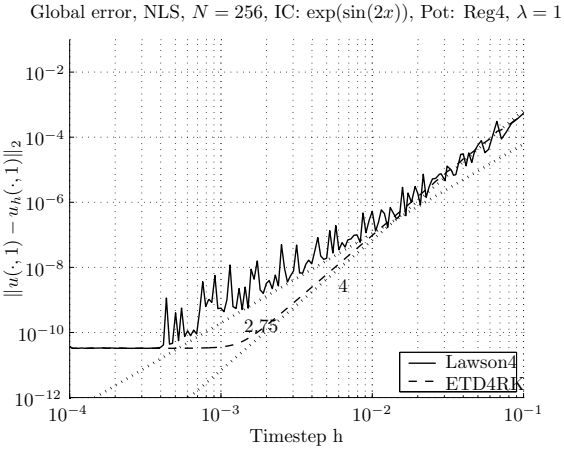


Figure 5.4: Global error when the potential has regularity 4.

present in the linearized version as well as in the nonlinear version, that is when $\lambda = 0$ in (5.1). This facilitates substantially simplified analysis.

Analysis for the linear problem

Consider the Fourier domain linear problem

$$\dot{y} = Ly + Wy \tag{5.22}$$

where L is the Laplacian in Fourier domain as before (diagonal, $\mathrm{i}k^2$) and W is a circulant matrix stemming from a Fourier transform of the potential in the Schrödinger equation (5.1). The fact that the matrices L and W in general do not commute is the source of the order reduction observed in the Lawson scheme as we shall see. If, on the other hand, the potential function is a constant, L and W will commute, and order reduction is not observed.

The presentation here resembles the Strang splitting analysis of Jahnke and Lubich [11] on (5.22) when the linear operator in the differential equation is unbounded. They found order reduction due to the same phenomena that we will see here.

Applying an explicit exponential integrator to (5.22) we get

$$y_1 = \mathrm{e}^{hL}y_0 + h\Big(\sum_{i=1}^{s} b_i(hL)W\mathrm{e}^{c_i hL}\Big)y_0 + \mathcal{O}(h^2). \tag{5.23}$$

Then, by way of the variation of constants formula, the exact solution to (5.22) may be represented as

$$\mathrm{e}^{h(L+W)}y_0 = \mathrm{e}^{hL}y_0 + \int_0^h \mathrm{e}^{sL}W\mathrm{e}^{(h-s)(L+W)}y_0\,\mathrm{d}s.$$

For our fourth order schemes, we iterate the variation of constants formula four times for the exact solution resulting in a sum including up to five-dimensional integrals. Applying the variation of constants formula once more to remove W from the exponential, and substituting $\theta = (h-s)/h$, it is clear that a second order scheme must satisfy

$$\sum_{i=1}^{s} b_i(hL)W\mathrm{e}^{c_i hL}y_0 = \int_0^1 \mathrm{e}^{(1-\theta)hL}W\mathrm{e}^{\theta hL}y_0\,\mathrm{d}\theta. \tag{5.24}$$

Regularity requirement for the Lawson scheme

Inserting the Lawson4 scheme $b_i(hL)$ coefficients allows interpretation of (5.24). The left hand side of (5.24) is Simpson's quadrature of the function $f(\theta) =$

$\mathrm{e}^{(1-\theta)hL} W \mathrm{e}^{\theta hL}$. The error of Simpson's quadrature is known to be $f^{(4)}(\xi)/2880$ for some $\xi \in [0,1]$, and in this case

$$f^{(4)}(\xi) = h^4 \mathrm{e}^{(1-\xi)hL}[L,[L,[L,[L,W]]]]\mathrm{e}^{\xi hL} = h^4 \mathrm{e}^{(1-\xi)hL} \mathrm{ad}_L^4(W) \mathrm{e}^{\xi hL}. \quad (5.25)$$

Transforming from Fourier space to phase space, L becomes $\mathrm{d}^2/\mathrm{d}x^2$ and W becomes a multiplication operator denoted by V. One may verify the formula

$$\mathrm{ad}^m_{\mathrm{d}^2/\mathrm{d}x^2}(V)_0 = \sum_{i=0}^{m} 2^i \binom{m}{i} V^{(2m-i)} \psi_0^{(i)}. \quad (5.26)$$

When $m = 4$, one observes that the Lawson4 scheme satisfies condition (5.24) to a sufficient degree of accuracy if the initial condition in phase space $\psi_0(x) \in C^4(-\pi,\pi)$ and the potential $V(x) \in C^8(-\pi,\pi)$.

Iterating the variation of constants formula further, one obtains additional iterated integrals. As these integrals involve only lower derivatives of the appropriate $f(\theta_1,\theta_2,\dots)$ function, equating to lowered regularity requirements for V and y_0, we omit the details in this exposition.

Regularity requirement for ETD4RK

We interpret (5.24) in a Gauss quadrature sense with the weight function $w(\theta) = \mathrm{e}^{(1-\theta)hL}$. Requiring the quadrature formula to be exact for fourth degree polynomials gives four stiff order conditions.

$$\sum_{i=1}^{s} b_i(hL) c_i^k = k! \varphi_{k+1}(hL), \quad \text{for } k = 0,1,2,3. \quad (5.27)$$

For ETD4RK this is not in general satisfied when $k = 4$, and we expect the principal quadrature error term to depend on $g^{(4)}(\theta)$ where $g(\theta) = W\mathrm{e}^{\theta hL} u_0$. Differentiating this function, we get

$$g^{(4)}(\theta) = h^4 W L^4 \mathrm{e}^{\theta hL} y_0 \quad (5.28)$$

an upper bound for which translates to y_0 being at least 8 times continuously differentiable in space. Thus, we should expect ETD4RK to demand more regularity for the initial condition than Lawson4. On the other hand, ETD4RK makes no demand on the regularity of the potential function, as opposed to the Lawson4 scheme.

Numerical results

Figures 5.5 and 5.6 show global error plots with both Lawson and ETD4RK. The potential is smooth while the regularity of the initial condition is low (Fourier

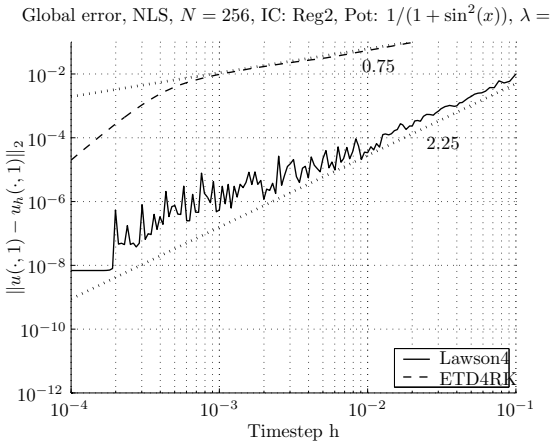


Figure 5.5: Smooth potential, initial condition of regularity 2.

decay rates of 2 and 4). It is apparent that ETD4RK suffers drastically from the low regularity, and based on experiments, it has order $h^{r/2-1/4}$ when r is the regularity, independent of linear problem or not.

5.6 Discussion

Computational speed

In terms of construction and implementation, Lawson type exponential integrators are more immediate than ETD type schemes. Particularly, the coefficient functions of Lawson schemes are given explicitly by (5.10), whereas derivation of ETD type coefficient functions is typically more cumbersome. Additionally, Lawson type schemes require one or more matrix exponentials for which acceptable algorithms are well known. The ETD type schemes require the evaluation of multiple φ-functions, a computational problem which is at least as difficult as computing matrix exponentials. In evaluating φ-functions, Kassam and Trefethen [12] discovered a stability problem which they solved by contour integral evaluation in the complex plane. This requires an a priori contour radius which in general is problem dependent and not trivially available. In our numerical experiments, we found a scaling and squaring technique together with Padé-approximations of the φ function to be a better option, inspired by a code from [8]. The actual implementation is discussed in [2].

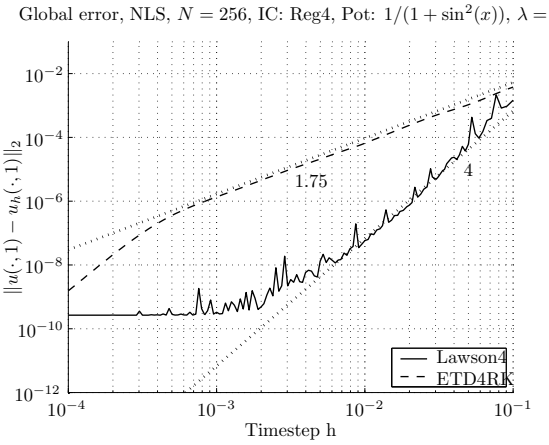


Figure 5.6: Smooth potential, initial condition of regularity 4.

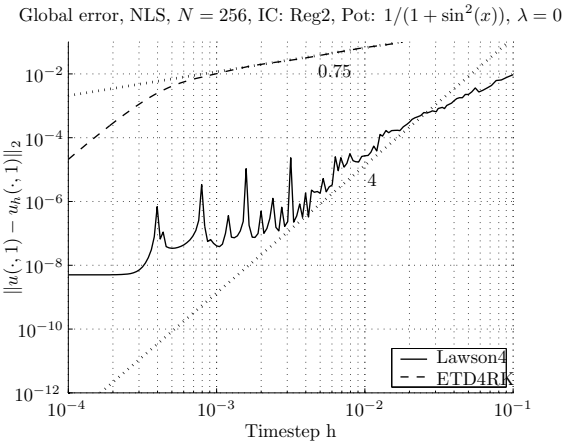


Figure 5.7: Lawson4 is close to order 4 but oscillating in the linear case, $\lambda = 0$, while ETD4RK suffers from the low regularity (2) of the initial function. The potential is smooth, $1/(1+\sin^2(x))$. In the corresponding local error plot, ETD4RK exhibits order 1.75 and Lawson order 5 with no oscillations.

Oscillations in observed order

Most order plots for the Lawson4 integrator show significant oscillations in observed accuracy as a function of time step h. Zooming in on each plot reveals that the the oscillations are smooth but quickly varying magnitudes of the highest eigenmode of L. These oscillations span roughly 2 orders of magnitude, and therefore represent a considerable error contribution at particular time step sizes.

The oscillations are due to some resonance effects, and that these are not damped as in the case of ETD schemes by dividing by z in the φ functions. To avoid these oscillations the Lawson schemes therefore must use φ-based coefficient functions. This, in turn, effectively renders the scheme into another type than what has been denoted Lawson schemes in this paper. Moreover, in a sense the resulting scheme is worse than Lawson's scheme as the modified scheme becomes more sensitive to the regularity of the initial condition.

Low regularity potential and initial condition

Using low regularity initial conditions and potential functions, we get the mixed case of undesirable behaviour from both types of schemes. Varying both the initial condition and the potential (one particular combination of which is shown in Figure 5.8), there is little actual gain from choosing one scheme over the other. However, due to the observed oscillations, ETD4RK might be a better choice in these cases.

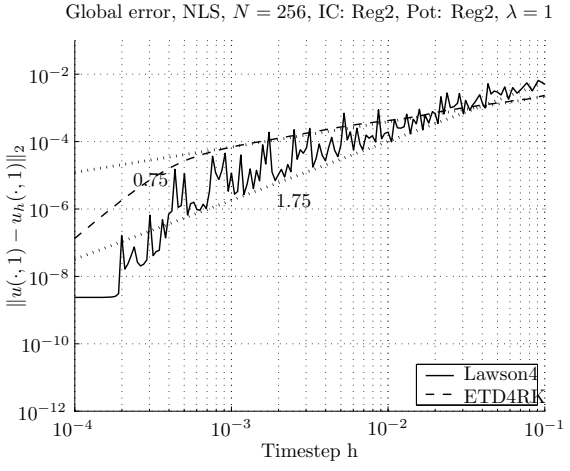


Figure 5.8: The mixed case: Both the initial condition and the potential has regularity 2.

Exponential general linear methods

General linear methods generalize Runge–Kutta integrators and multistep integrators. Exponential general linear methods thus generalize the integrators catered for in this paper, as reported in [2]. A class of exponential general linear methods known as the generalized Lawson schemes, see [17], mixes the Lawson and ETD schemes and give good results on parabolic problems, achieving high stiff order. However, in the experiments described in this paper, these schemes never perform better than the best of ETD4RK or Lawson4.

Rounding error accumulation

In closing we would like to comment on an important feature of our experiments. The measured error does not decrease further as a function of decreasing step size once the error reaches a level of about 10^{-10}. As this is several orders of magnitude larger than machine accuracy, it is clear that rounding errors introduced in the evaluation of the φ functions affect long-time accuracy of the exponential integrator. We still believe that the Padé approximation, as described in [2], is the best algorithm for evaluating φ functions, and that accuracy of exponential integrators may be increased by further research into this algorithm.

5.7 Conclusion

We have studied the numerical performance of the Lawson4 scheme compared to the ETD4RK scheme on a nonlinear Schrödinger test problem and observe that the actual performance is heavily influenced by the potential function and initial condition. In short, Lawson4 is dependent upon the regularity of the potential function while ETD4RK is dependent upon the regularity of the initial condition. Stiff order conditions are used as a tool for explaining the observed behaviour, although the general applicability of stiff order conditions to non-parabolic problems remains unclear. Further research is necessary to explain phenomena exhibited by exponential integrators on partial differential equations.

Bibliography

[1] Berland, H., Owren, B., Skaflestad, B.: B-series and order conditions for exponential integrators. SIAM J. Numer. Anal. **43**(4), 1715–1727 (2005)

[2] Berland, H., Skaflestad, B., Wright, W.M.: Expint — A Matlab package for exponential integrators. ACM Trans. on Math. Soft. (2006). To appear

[3] Bridges, T.J., Reich, S.: Multi-symplectic integrators: numerical schemes for Hamiltonian PDEs that conserve symplecticity. Phys. Lett. A **284**(4-5), 184–193 (2001)

[4] Cazenave, T.: An introduction to nonlinear Schrödinger equations, third edn. No. 26 in Textos de Métodos Mathemáticos. Universidade Federal do Rio de Janeiro, Instituto de Matemática (1996)

[5] Certaine, J.: The solution of ordinary differential equations with large time constants. In: Mathematical methods for digital computers, pp. 128–132. Wiley, New York (1960)

[6] Cox, S.M., Matthews, P.C.: Exponential time differencing for stiff systems. J. Comput. Phys. **176**(2), 430–455 (2002)

[7] Ehle, B.L., Lawson, J.D.: Generalized Runge–Kutta processes for stiff initial-value problems. J. Inst. Maths. Applics. **16**, 11–21 (1975)

[8] Hochbruck, M., Lubich, C., Selhofer, H.: Exponential integrators for large systems of differential equations. SIAM J. Sci. Comput. **19**(5), 1552–1574 (1998)

[9] Hochbruck, M., Ostermann, A.: Explicit exponential Runge–Kutta methods for semilinear parabolic problems. SIAM J. Numer. Anal. **43**(3), 1069–1090 (2005)

[10] Islas, A.L., Karpeev, D.A., Schober, C.M.: Geometric integrators for the nonlinear Schrödinger equation. J. of Comp. Phys. **173**, 116–148 (2001)

[11] Jahnke, T., Lubich, C.: Error bounds for exponential operator splittings. BIT **40**(4), 735–744 (2000)

[12] Kassam, A.K., Trefethen, L.N.: Fourth-order time-stepping for stiff PDEs. SIAM J. Sci. Comput. **26**(4), 1214–1233 (electronic) (2005)

[13] Krogstad, S.: Generalized integrating factor methods for stiff PDEs. J. of Comp. Phys. **203**(1), 72–88 (2005)

[14] Lawson, J.D.: Generalized Runge–Kutta processes for stable systems with large Lipschitz constants. SIAM J. Numer. Anal. **4**, 372–380 (1967)

[15] Minchev, B., Wright, W.M.: A review of exponential integrators for semi-linear problems. Tech. Rep. 2/05, Department of Mathematical Sciences, NTNU, Norway (2005). `http://www.math.ntnu.no/preprint/`

[16] Nørsett, S.P.: An A-stable modification of the Adams–Bashforth methods. In: Conf. on Numerical Solution of Differential Equations (Dundee, Scotland, 1969), pp. 214–219. Springer, Berlin (1969)

[17] Ostermann, A., Thalhammer, M., Wright, W.M.: A class of explicit exponential general linear methods. BIT **46**(2), 409–432 (2006)

Paper V

Håvard Berland, Alvaro L. Islas, and Constance M. Schober:

Conservation of phase space properties for the cubic Schrödinger equation

Preprint Numerics No. 1/06, Department of Mathematical Sciences, Norwegian University of Science and Technology. Submitted to Journal of Computational Physics.

Chapter 6

Conservation of phase space properties using exponential integrators on the cubic Schrödinger equation

Abstract:

The cubic nonlinear Schrödinger (NLS) equation with periodic boundary conditions is solvable using Inverse Spectral Theory. The "nonlinear" spectrum of the associated Lax pair reveals topological properties of the NLS phase space that are difficult to assess by other means. In this paper we use the invariance of the nonlinear spectrum to examine the long time behavior of exponential and multisymplectic integrators as compared with the most commonly used split step approach. The initial condition used is a perturbation of the unstable plane wave solution, which is difficult to numerically resolve. Our findings indicate that the exponential integrators from the viewpoint of efficiency and speed have an edge over split step, while a lower order multisymplectic is not as accurate and too slow to compete.

6.1 Introduction

Exponential integrators have been popular in recent literature and they seem especially attractive in PDE settings as a means of easily obtaining high order explicit schemes without step size restrictions for the time-integration. The notable property of exponential integrators is the use a splitting of the problem into a linear part, possibly rendering the problem stiff, and a nonlinear part. The linear part is treated exactly, in an attempt to ameliorate temporal step size restrictions imposed by possible stiffness inherent in the PDE problem. However, experience with exponential integrators on long time scales for various PDEs with a complicated phase space structure is not abundant. This paper examines whether exponential integrators are a viable alternative for the time integration of PDEs, by comparing them with two other classes of time-integrators and by using the NLS equation as a benchmark problem.

The NLS equation is completely integrable with a rich phase space structure (i.e. stable as well as unstable solutions with homoclinic orbits). Typically the performance of an integrator has been determined by examining the preservation of global invariants such as energy and momentum. The nonlinear spectrum of

the associated Lax pair of the PDE provides a detailed description of the phase space and is invariant in time. Following [17], we monitor the spectrum to assess how well the phase space is preserved by the various integrators. Section 6.2 outlines the necessary properties of the inverse spectral theory.

For Hamiltonian PDEs possessing a multisymplectic structure (symplectic in both space and time), multisymplectic integrators preserve exactly a discrete version of the multisymplectic structure [10]. For quadratic Hamiltonians, multisymplectic integrators preserve the local conservation laws exactly. Even though this is not the case for the NLS equation, multisymplectic integrators were shown to provide improved resolution of the local conservation laws and dynamical invariants [16, 17]. Compared to classical integrators (Runge–Kutta), multisymplectic integrators exhibited less drift in the conservation laws over long time periods [17]. We include an implicit second order multisymplectic integrator for comparison with the exponential integrators and it is found to be the most computationally demanding integrator among those tested. Whether improved preservation of the structural and geometric properties of the PDE by multisymplectic integrators justifies their additional computational cost remains an open question.

Section 6.3 gives the necessary details of the spectral space discretization, which will be equivalent for all schemes included in this study. In Section 6.3 exponential integrators are presented in a format including the majority of known exponential integrators, and specifications of the exponential integrators used are given in coefficient function tableaus. We restrict our attention to explicit exponential integrators to facilitate speed and ease of implementation, and test the two integrators CFREE4 and LAWSON4. The implementation has used the MATLAB package described in [7] directly. A fourth order split step scheme obtained by Yoshida's technique is included for comparison, as it is, in many ways, related to exponential integrators and it is also used fairly extensively in the physics community. We only included a multisymplectic integrator of order two as, even at this order, it turned out to be computationally more intensive than the other integrators. Our final criteria for comparing integrators is based on accuracy obtained for fixed CPU time, and thus it makes sense to compare integrators with differing asymptotic order.

The numerical results are given in Section 6.4 where we examine how well the most critical features of the nonlinear spectrum are preserved. Knowledge of the properties of the nonlinear spectrum makes it possible to give simple criteria for determining whether a numerical result is acceptable or not. We present results in table form and conclude that among our integrators, CFREE4 seems to be the most reliable and CPU effective choice for our problem.

The nonlinear Schrödinger equation

The periodic focusing nonlinear Schrödinger (NLS) equation

$$u_t = \mathrm{i}u_{xx} + \mathrm{i}2|u|^2u, \tag{6.1}$$

$u(x + D, t) = u(x, t)$, is an infinite dimensional integrable Hamiltonian system with Hamiltonian

$$H(u, u^*) = \int_0^D \left(|u_x|^2 - |u|^4\right) \mathrm{d}x. \tag{6.2}$$

The (complex) solution $u(x, t)$ represents the slow space-time evolution of the envelope of a carrier signal, and has important applications in nonlinear optics, deep water waves and also plasma physics. The wave equation (6.1) bears its name because it corresponds to the quantum Schrödinger equation with $2|u|^2$ as the potential.

An important physical prediction of the NLS equation is the Benjamin–Feir or modulational instability for periodic boundary conditions [4]. An example of this is provided by the plane wave solution,

$$u(x, t) = a\mathrm{e}^{2\mathrm{i}|a|^2t}. \tag{6.3}$$

which has M linearly unstable modes with growth rates $\sigma_n^2 = \mu_n^2(\mu_n^2 - 4a^2)$, $\mu_n = 2\pi n/D$, provided

$$0 < (\pi n/D)^2 < |a|^2 \tag{6.4}$$

is satisfied (the number M of unstable modes is the largest integer satisfying $0 < M < |a|D/\pi$). That is, the plane wave is unstable with respect to long wavelength perturbations.

In the numerical experiments we consider multi-phase solutions whose initial data is obtained by perturbing the plane wave solution,

$$u(x, 0) = a\left(1 + \epsilon\,\mathrm{e}^{\mathrm{i}\psi}\cos(\mu_n x)\right), \tag{6.5}$$

where $a = \frac{1}{2}$, $\mu_n = \frac{2\pi n}{D}$ and $D = 4\sqrt{2}\pi$ is the length of the spatial domain. Typically, the strength of the perturbation ϵ is chosen to be 0.1, $\psi = 0$ and $n = 1$. For the given spatial length D, the plane wave has two unstable modes, thereby the initial condition is coined the two-mode case. Increasing the spatial period results in more unstable modes, making the problem numerically harder.

6.2 The inverse spectral method

In [28], the NLS equation (6.1) was shown by Zakharov and Shabat to be completely integrable (with infinite sequences of commuting flows and common conservation laws) and solvable by inverse scattering theory for rapidly decreasing

initial data on the infinite line. Special solutions such as solitons, which are localized wave packets (envelope solutions for the NLS equation) that survive collisions with one another, were proven to exist in this case. More generally, inverse spectral theory provides a method, the analog of the whole-line inverse scattering theory, for solving the NLS equation on periodic domains. Its, Krichever and Kotlarov used inverse spectral theory and methods of algebraic geometry to obtain N-phase solutions of the periodic NLS equation, expressible in terms of Riemann theta functions [18, 20, 3].

Lax pair and characterization of the spectrum

The inverse scattering and spectral methods may be viewed as a generalization of Fourier methods for solving linear PDEs. Briefly, the complete integrability of the NLS equation is established by using the associated Lax pair of linear operators defined by [28]:

$$\mathcal{L}^{(x)}\phi = 0 \qquad \text{and} \qquad \mathcal{L}^{(t)}\phi = 0, \tag{6.6}$$

where

$$\mathcal{L}^{(x)} = \begin{pmatrix} \frac{\mathrm{d}}{\mathrm{d}x} + \mathrm{i}\lambda & -u \\ u^* & \frac{\mathrm{d}}{\mathrm{d}x} - \mathrm{i}\lambda \end{pmatrix} \text{ and } \mathcal{L}^{(t)} = \begin{pmatrix} \frac{\mathrm{d}}{\mathrm{d}t} + \mathrm{i}[2\lambda^2 - |u|^2] & -2\lambda u - \mathrm{i}u_x \\ 2\lambda u^* - \mathrm{i}u_x^* & \frac{\mathrm{d}}{\mathrm{d}t} - \mathrm{i}[2\lambda^2 - |u|^2] \end{pmatrix}. \tag{6.7}$$

The coefficients $u(x,t)$ are periodic in x, $u(x+D,t) = u(x,t)$, λ is the spectral parameter, and ϕ is the eigenfunction. The NLS equation arises as the solvability or compatibility condition for these operators, i.e., $\phi_{xt} = \phi_{tx}$ if and only if $u(x,t)$ satisfies the NLS equation (6.1).

The first step in solving the NLS equation is to calculate the direct spectral transform of the initial data, i.e. to determine the spectrum of $\mathcal{L}^{(x)}$, which is analogous to calculating the Fourier coefficients in Fourier theory. Next, time evolution is performed on the "spectral data" using $\mathcal{L}^{(t)}$. Finally the inverse spectral transform is calculated in order to recover the waveform at a later time.

The direct spectral transform and its inverse provide a one-to-one correspondence between solutions of the NLS equation and the spectral data. The spectral data consists of two types of spectrum of $\mathcal{L}^{(x)}$; the periodic eigenvalues λ_j and the Dirichlet eigenvalues μ_j (to be defined below). The direct spectral transform consists of computing (λ_j, μ_j) from $u(x,t)$. A fundamental property is that the periodic eigenvalues λ_j are invariants of the NLS flow whereas the Dirichlet spectrum flows in both x and t. The time dependence of $\mu_j(x,t)$ is generated by $\mathcal{L}^{(t)}$ and is equivalent to the dynamics of the NLS flow since $\mathcal{L}^{(x)}$ and $\mathcal{L}^{(t)}$ are compatible. The inversion formula for N-phase solutions of the NLS is given by

the trace formula [18]:

$$\frac{\mathrm{i}u_x}{u} = \sum_{j=1}^{2N} \lambda_j - 2\sum_{j=1}^{N-1} \mu_j.$$

For periodic potentials the spectrum is obtained using Floquet theory. This spectral analysis is similar to that of the time-independent Hill's operator, with the main difference that $\mathcal{L}^{(x)}$ is not self-adjoint. The spectrum of u,

$$\sigma(u) = \left\{ \lambda \in \mathbf{C} \mid \mathcal{L}^{(x)}\phi = 0, |\phi| \text{ bounded } \forall x \right\}, \tag{6.8}$$

can be expressed in terms of the transfer matrix $M(x+D;u,\lambda)$ across a period, where $M(x;u,\lambda)$ is a fundamental solution matrix of the Lax pair (6.7). Introducing the Floquet discriminant $\Delta(u,\lambda) = \mathrm{Trace}\,[M(x+L;u,\lambda)]$, one obtains

$$\sigma(u) = \{\lambda \in \mathbf{C} \mid \Delta(u,\lambda) \in \mathbf{R},\ -2 \le \Delta(u,\lambda) \le 2\}. \tag{6.9}$$

The distinguished points of the periodic/antiperiodic spectrum, where $\Delta(u,\lambda_j) = \pm 2$, are:

(a) Simple points $\{\lambda_j^s \mid \frac{\mathrm{d}\Delta}{\mathrm{d}\lambda} \neq 0\}$,

(b) Double points $\{\lambda_j^d \mid \frac{\mathrm{d}\Delta}{\mathrm{d}\lambda} = 0,\ \frac{\mathrm{d}^2\Delta}{\mathrm{d}\lambda^2} \neq 0\}$.

The Floquet discriminant functional $\Delta(u,\lambda)$ is invariant under the NLS flow and encodes the infinite family of constants of motion of the NLS (parametrized by the λ_j's). While the λ_j's are equivalent to a set of invariant action variables, the μ_j's (which are the zeros of the M_{12} entry of the transfer matrix and are not invariant) provide the conjugate angle variables.

The Floquet spectrum (6.9) of a generic NLS potential consists of the entire real axis plus additional curves (called bands) of continuous spectrum in the complex plane which terminate at the simple points λ_j^s. N-phase solutions are those with a finite number of bands of continuous spectrum. Double points can be thought of as the coalescence of two simple points and their location is particularly significant. The order and location of the λ_j completely determine the spatial structure and nonlinear mode content of NLS solutions as well as the dynamical stability as follows [12, 13]:

(a) Simple points correspond to stable active degrees of freedom.

(b) Double points label all additional potentially active degrees of freedom.

Real double points correspond to stable inactive (zero amplitude) modes. Complex double points correspond to the unstable active modes and parametrize the associated homoclinic orbits.

The N-phase quasiperiodic solutions of the NLS equation have the form [3, 18, 20],

$$u(x,t) = u_0 \mathrm{e}^{\mathrm{i}(k_0 x - \omega_0 t)} \frac{\Theta(\mathbf{W}^-|\tau)}{\Theta(\mathbf{W}^+|\tau)}, \tag{6.10}$$

where Θ is the Riemann theta function, $\mathbf{W}^\pm = (W_1^\pm, \ldots, W_N^\pm)$, and the phases evolve according to $W_j^\pm = (\kappa_j x + \Omega_j t + \theta_j^\pm)$. The external phase as well as the wavenumbers κ_j and frequencies Ω_j are expressible in terms of algebraic-geometric data including the branch points of the associated Riemann surface (namely, the simple points λ_j^s of the Floquet spectrum). The entire x and t dependence of an N-phase solution is captured by ODEs for the μ_j's. Essentially these ODEs linearize via the classical Abel–Jacobi map associated with the Riemann surface. The phases $W_j^\pm$ in the action-angle description of these solutions are the images of $\mu_j(x,t)$ under the Abel–Jacobi map.

In terms of the NLS phase space, the values of the actions λ_j fix a particular level set. The level set defined by u is then given by, $\mathcal{M}_u \equiv \{v \,|\, \Delta(v,\lambda) = \Delta(u,\lambda), \lambda \in \mathbf{C}\}$. Typically, $\mathcal{M}_u$ is an infinite dimensional stable torus. However, the NLS phase space also contains degenerate tori which may be unstable. If a torus is unstable, its invariant level set consists of the torus and an orbit homoclinic to the torus. These invariant level sets, consisting of an unstable component, are represented, in general, by complex double points in the spectrum [13].

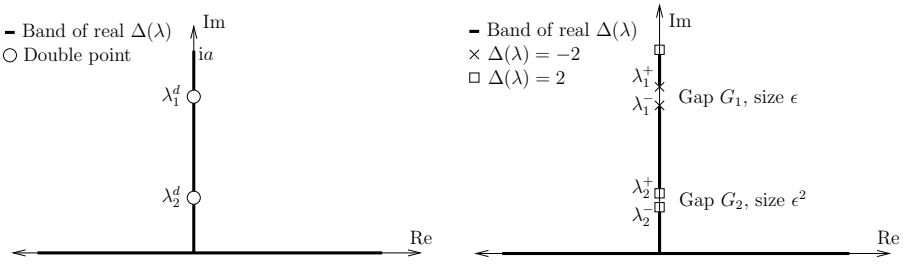
In our experiments we implement the first step of the inverse spectral method and appeal to the invariance of the λ_j's to evaluate the ability of the various numerical schemes to preserve the NLS phase space structure. The nonlinear spectrum for the Lax pair (6.7) is computed numerically using Fortran software at each timevalue of the solution. In our case, only the λ_j's where $\Delta = \pm 2$ are necessary. The discriminant function Δ is obtained by a direct nonlinear spectral transform, solving the overdetermined system of ODEs (in the variable x) given by the Lax pair for the given numerical solution $u(x,t)$. The zeros of $\Delta = \pm 2$ are then obtained with a root solver using Muller's method as in [21]. The spectrum itself is calculated with an accuracy of $\mathcal{O}(10^{-8})$ which is sufficient for our results.

Perturbation of the plane wave

The simplest example of an N-phase solution of the NLS equation is the plane wave solution (6.3). In this case, the Floquet discriminant is given by $\Delta(a,\lambda) = 2\cos(\sqrt{a^2+\lambda^2}D)$ and thus the spectrum consists of the entire real axis and the band $[-\mathrm{i}a, \mathrm{i}a]$. At $\pm\mathrm{i}a$ there is a pair of simple periodic/antiperiodic eigenvalues and there is an infinite sequence of double points,

$$\lambda_j^2 = (j\pi/D)^2 - a^2 \tag{6.11}$$

for $j \in \mathbf{Z}$, of which $2\lfloor aD/\pi \rfloor = 2M$ are pure imaginary double points and the remaining are real. The condition for a complex double point is exactly the condition for a mode to be linearly unstable (compare with (6.4)). However, in contrast to inverse spectral theory, linear analysis does not provide any answers to what happens to the solution on a long time-scale.



(a) Plane wave, unperturbed. (b) Perturbed, two-mode solution.

Figure 6.1: The spectrum (6.9) for (a) the plane wave (6.3) and (b) the perturbed plane wave data (6.5). The perturbation of strength ϵ opens up the two double points of the plane wave spectrum (a) and yields the perturbed spectrum (b) with two gaps corresponding to two semi-stable modes in the solution (see Figure 6.2). Real double points are not shown.

When $a = 0.5$ and $D = 4\sqrt{2}\pi$ in (6.3), $M = 2$ and λ_1 and λ_2 are complex double points (since the spectrum is symmetric with respect to the real axis we restrict our consideration to $\operatorname{Im}\lambda \geq 0$). It is difficult in advance to predict the time of failure using only physical data or a few global constants. For the family of initial data (6.5), as the energy levels are varied slightly there are alternating windows of stable and unstable tori. As a consequence, we have found that the λ_j's, which determine the geometry of the level sets, are the significant quantities to preserve [17].

Assuming $u = u^{(0)} + \epsilon u^{(1)} + \cdots$, $\lambda = \lambda^{(0)} + \epsilon\lambda^{(1)} + \cdots$, and $\phi = \phi^{(0)} + \epsilon\phi^{(1)} + \cdots$, the spectrum of initial data (6.5) can be calculated via perturbation analysis. Substituting these expansions into (6.7), the $\lambda_j^{(0)}$ are given by (6.11) and the corrections at $\mathcal{O}(\epsilon)$ are [2]

$$\left(\lambda_j^{(1)}\right)^2 = \begin{cases} \frac{a^2}{16\lambda_j^2}\left(\mathrm{e}^{-\mathrm{i}\phi} - \frac{1}{a^2}\left(j\pi/D + \lambda_j\right)^2 \mathrm{e}^{\mathrm{i}\phi}\right) \\ \qquad \cdot \left(\mathrm{e}^{-\mathrm{i}\phi} - \frac{1}{a^2}\left(-j\pi/D + \lambda_j\right)^2 \mathrm{e}^{\mathrm{i}\phi}\right) & j = n \\ 0 & j \neq n \end{cases} \tag{6.12}$$

At $\mathcal{O}(\epsilon)$ there is a correction only to the double point $\lambda_j^{(0)}$, $(j = n)$, which

resonates with the perturbation. The other double points do not experience an $\mathcal{O}(\epsilon)$ correction.

The behavior of the correction $\lambda_j^{(1)}$, $(j = n)$, depends on whether the double point $\lambda_j^{(0)}$ is real or imaginary:

$$\left(\lambda_j^{(1)}\right)^2 = \begin{cases} -\frac{a^2}{4\lambda_j^2}\sin(\phi+\theta)\sin(\phi-\theta) & \text{for } \lambda_j \text{ imaginary} \\ -\frac{a^2}{8\lambda_j^2}\left[\cos 2\phi + 1 - 2(j\pi/D)^2/a^2\right] & \text{for } \lambda_j \text{ real} \end{cases}$$

where $\tan\theta = \mathrm{Im}(\lambda_j)D/(j\pi)$. Since initial data (6.5) is for a solution even in x, the spectrum has an additional symmetry with respect to the imaginary axis. This is reflected in the correction $\lambda_j^{(1)}$ which, for imaginary double points, is either real, zero or pure imaginary [2] depending on the choice of ψ in the perturbed potential, i.e. imaginary double points can split into either crosses or gaps in the spectrum. This is a realization of the saddle structure of the real part $\Delta(u, \lambda_j)$ when λ_j is imaginary. More generally, if the potential is noneven in x, the correction $\lambda_j^{(1)}$ is fully complex and the double point can split in any direction, yielding an asymmetric version of a gap state.

For real double points, the correction $\lambda_j^{(1)}$ can only be imaginary. Gaps cannot appear on the real axis in the spectrum of $\mathcal{L}^{(x)}$. Hence the situation with real double points is very different from that of imaginary double points. Splitting of real double points only introduces additional degrees of freedom into the spatial structure but does not introduce an instability as homoclinic manifolds are not associated with them. At the next order only the double points λ_j, $(j = 2n)$, experience an $\mathcal{O}(\epsilon^2)$-splitting. A full determination of the splitting at higher order is given in [2] and in the non-even case in [1].

The Floquet spectrum for initial data (6.5), with $\psi = 0$ and $n = 1$ is shown in Figure 6.1b. The corrections to λ_1 and λ_2, as determined by (6.12), are pure imaginary and gaps G_1 and G_2 open in the spectrum with sizes of order ϵ and ϵ^2, respectively [2]. The double point λ_j splits into $\lambda_j^{\pm}$ and the gap $G_j = \lambda_j^+ - \lambda_j^-$. The first gap corresponds to the center mode in Figure 6.2 and from $\lambda_1^{\pm}$ the spatial and temporal frequencies of the mode are determined. Similarly $\lambda_2^{\pm}$ (corresponding to the smaller gap of size ϵ^2) determine the wave number and frequency of the second mode which appears symmetrically on both sides. If the ratio of the two temporal frequencies is a rational number, the initial condition will recur in finite time.

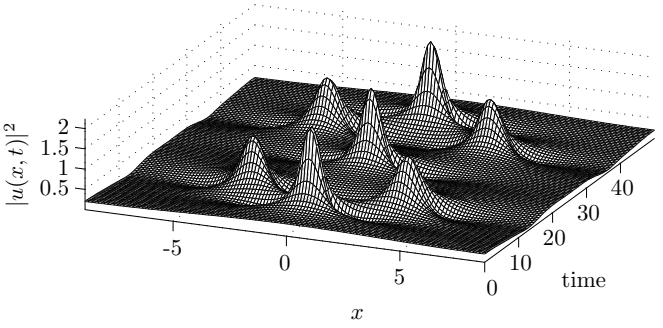


Figure 6.2: The surface $|u(x,t)|^2$ of the nonlinear Schrödinger equation with initial condition (6.5). The center mode around $x = 0$ corresponds to gap G_1 in Figure 6.1b, and the second mode appearing around $x = \pm 3$ in the $|u(x,t)|^2$-plot corresponds to gap G_2 in Figure 6.1b.

6.3 Numerical integrators

Space discretization

The instability of the NLS equation which we are examining occurs only for periodic boundary condition, thus it is reasonable to restrict our attention to spectral spatial discretizations making use of the fast Fourier transform. The spatial resolution is equivalent for all of the schemes considered, so the differences in performance are attributable to the time integrators.

Let

$$\hat{u}(k,t) = \mathscr{F}(u(x,t)) = \frac{1}{D}\int_{-D/2}^{D/2} u(x,t)\mathrm{e}^{-\nu_k x}\,\mathrm{d}x$$

with $k \in \mathbf{Z}$ and $\nu_k = \frac{2\pi \mathrm{i}}{D}k$ be the Fourier transform of $u(x,t)$. The inverse Fourier transform is

$$u(x,t) = \mathcal{F}^{-1}(\hat{u}(k,t)) = \sum_{k=-\infty}^{\infty} \hat{u}(k,t)\mathrm{e}^{\nu_k x}.$$

In Fourier space, equation (6.1) then takes the form of an infinite system of first order ODEs

$$\begin{aligned} \hat{u}_t(k,t) &= \mathrm{i}\nu_k^2\hat{u}(k,t) + 2\mathrm{i}\left|\mathcal{F}^{-1}(\hat{u}(k,t))\right|^2\hat{u}(k,t) \qquad k \in \mathbf{Z}, \quad t \geq 0 \\ \hat{u}(k,0) &= \mathscr{F}(u(x,0)) \end{aligned}. \tag{6.13}$$

The spatial discretization now consists of truncating (6.13) to $N_{\mathcal{F}}/2 \leq k \leq$

$N_{\mathcal{F}}/2 - 1$, and the resulting system of dimension $N_{\mathcal{F}}$ is then solved by the integrators mentioned below.

Exponential integrators

Exponential integrators are a class of integrators that emerged from an improved ability in numerical software to compute the matrix exponential in a reasonable amount of time. Recent developments [5, 11, 15, 22] have yielded an abundance of possibilities with regards to choice of scheme (see for instance [7] and the accompanying source code for a listing). In this study we examine the performance of the class of exponential integrators; thus, the particular choice of exponential integrator is of less importance. After numerous initial numerical tests, we have chosen CFREE4 as "the" exponential integrator in this report. It gives the best overall performance for our problem as measured by the global error and the preservation of invariants over long integration intervals.

An s-stage explicit exponential integrator of Runge–Kutta type for the system of ordinary differential equations

$$\dot{y}(t) = Ly(t) + N(y,t), \qquad y(0) = y_0 \in \mathbf{R}^n \tag{6.14}$$

where L is linear and $N(y,t)$ is a (possibly) nonlinear function, is the procedure

$$Y_i = \sum_{j=1}^{s} a_{ij}(hL)\, hN(Y_j, t_0 + c_j h) + \exp(c_i hL) y_0 \tag{6.15a}$$

$$y_1 = \sum_{i=1}^{s} b_i(hL)\, hN(Y_i, t_0 + c_i h) + \exp(hL) y_0 \tag{6.15b}$$

in which Y_i, $i = 1, \ldots, s$, are internal stages and y_1 is the final approximation of $y(t_1) = y(t_0 + h)$. This format extends the format for Runge–Kutta schemes in that the coefficients a_{ij} and b_i are now analytic functions in the matrix L.

In short, the main properties of an exponential integrator are (i) when the linear part L is zero, we recover the *underlying* Runge–Kutta-scheme, and (ii) when $N \equiv 0$, exponential integrators compute the exact solution. The coefficient functions $a_{ij}(z)$ and $b_i(z)$ ($z = hL$) are usually listed in a tableau similar to the Butcher tableau for Runge–Kutta schemes. For CFREE4, the functions are

$$\begin{array}{c|cccc|c}
0 & & & & & 1 \\
\frac{1}{2} & \frac{1}{2}\varphi_{1,2} & & & & \mathrm{e}^{z/2} \\
\frac{1}{2} & & \frac{1}{2}\varphi_{1,2} & & & \mathrm{e}^{z/2} \\
1 & \frac{1}{2}\varphi_{1,2}(\mathrm{e}^{z/2} - 1) & & \varphi_{1,2} & & \mathrm{e}^{z} \\
\hline
 & \frac{1}{2}\varphi_1 - \frac{1}{3}\varphi_{1,2} & \frac{1}{3}\varphi_1 & \frac{1}{3}\varphi_1 & -\frac{1}{6}\varphi_1 + \frac{1}{3}\varphi_{1,2} & \mathrm{e}^{z}
\end{array} \tag{6.16}$$

where

$$\varphi_\ell(z) = \frac{1}{(\ell-1)!}\int_0^1 \mathrm{e}^{(1-\theta)z}\theta^{\ell-1}\,\mathrm{d}\theta, \qquad \ell = 1, 2, \ldots \tag{6.17}$$

and

$$\varphi_{i,j} = \varphi_i(c_j z), \qquad i = 1, \ldots \text{ and } j = 1, \ldots, s. \tag{6.18}$$

For $\ell = 1, 2, 3$ (and for $z \neq 0$) the functions are

$$\varphi_1(z) = \frac{\mathrm{e}^z - 1}{z}, \quad \varphi_2(z) = \frac{\mathrm{e}^z - z - 1}{z^2}, \quad \text{and} \quad \varphi_3(z) = \frac{\mathrm{e}^z - z^2/2 - z - 1}{z^3}.$$

There is a singularity at $z = 0$ that could potentially lead to cancellation errors for small time steps. In order to avoid this in our experiments, these so called φ functions are evaluated by scaling the argument z, using a Padé approximant of φ_ℓ, and squaring the result. Details for this are presented in [7] and further analyzed in [8, 19].

Stiff order analysis of exponential integrators was introduced as an analytical tool in [15] which addresses the convergence of exponential integrators for semilinear parabolic partial differential equations. To obtain order p convergence in that case, independently of spatial resolution, stiff order p for the exponential integrator is required. Stiff order conditions represent an additional set of order conditions, including the classical order conditions as a special case. The CFREE4 scheme is only of stiff order 2, but still outperformed the other schemes in the initial study, including expensive schemes with high stiff order, though only marginally in some cases.

The report [6] is a study of two fourth order exponential integrators on the nonlinear Schrödinger equation, one Lawson integrator with stiff order 1, and ETD4RK from [11] with stiff order 2. Some effects (exhibited through order reduction) were attributable to low stiff order of the Lawson integrators, possibly also connected to preservation of fixed points, a condition equivalent to stiff order 2. Given sufficient spatial smoothness of the nonlinear function, and sufficient smoothness of the initial condition, high stiff order seemed less important. Also in [6], connections between the Lawson type and split step integrators are indicated, and it is in this respect that the we include the Lawson integrator of order 4 but stiff order 1, LAWSON4, with tableau

$$\begin{array}{c|cccc|c}
0 & & & & & 1 \\
\frac{1}{2} & \frac{1}{2}\mathrm{e}^{z/2} & & & & \mathrm{e}^{z/2} \\
\frac{1}{2} & & \frac{1}{2} & & & \mathrm{e}^{z/2} \\
1 & & & \mathrm{e}^{z/2} & & \mathrm{e}^{z} \\
\hline
 & \frac{1}{6}\mathrm{e}^{z} & \frac{1}{3}\mathrm{e}^{z/2} & \frac{1}{3}\mathrm{e}^{z/2} & \frac{1}{6} & \mathrm{e}^{z}
\end{array}. \tag{6.19}$$

Both CFREE4 and LAWSON4 reduce to the same classical fourth order Runge–Kutta scheme when the linear part is zero.

In the spectral space discretization, the matrix L in (6.14) for (6.13) becomes diagonal with elements

$$L_{kk} = \mathrm{i}\nu_k^2 = -\mathrm{i}k^2/8, \qquad k = -N_{\mathcal{F}}/2, \ldots, N_{\mathcal{F}}/2 - 1. \tag{6.20}$$

The nonlinear function N in (6.14) becomes

$$N(\hat{u}(k,t),t) = 2\mathrm{i}\left|\mathcal{F}^{-1}(\hat{u}(k,t))\right|^2 \hat{u}(k,t). \tag{6.21}$$

Our implementation uses fixed step sizes for all schemes, and as the linear part is time-independent, the functions $\mathrm{e}^{c_i z}$ and $\varphi_{i,j}$ are computed only at time zero of the integration and cached for the subsequent integration steps. Thus, the cost of evaluating the exponential and the φ functions are amortized over long integration intervals.

Second order multisymplectic spectral

The multisymplectic formulation for the nonlinear Schrödinger equation was introduced in [10], as an integrator being symplectic in both time and space. Multisymplectic integrators preserve exactly a discrete version of multisymplecticity. If the Hamiltonian function $S(z)$ is space and time independent, the PDE will possess local energy and momentum conservation laws. In addition, if $S(z)$ is quadratic in z, a multisymplectic integrator will preserve exactly these local conservation laws and, in the periodic case, the associated global conservation laws.

Although the Hamiltonian for the NLS equation is space and time independent, it is not quadratic and thus only multisymplecticity is exactly conserved in this case. Nevertheless, multisymplectic integrators have been reported to preserve the conservation laws better than classical Runge–Kutta schemes [17].

We will first show the multisymplectic formulation for a finite difference discretization of a PDE, then we see how it simplifies when using a spectral discretization in space.

Equation (6.1) can be written in multisymplectic form by letting $u = p + \mathrm{i}q$ and augmenting the phase space with the variables $v = p_x$ and $w = q_x$. The system obtained is

$$\begin{aligned} q_t - v_x &= 2(p^2+q^2)p \\ -p_t - w_x &= 2(p^2+q^2)q \\ p_x &= v \\ q_x &= w \end{aligned} \tag{6.22}$$

A Hamiltonian PDE is defined as multisymplectic if it can be written in the form

$$Mz_t + Kz_x = \nabla_z S(z)$$

where M and K are skew-symmetric matrices, and S is a smooth function of the state variable z. Formulation (6.22) of the nonlinear Schrödinger equation is multisymplectic provided

$$z = \begin{pmatrix} p \\ q \\ v \\ w \end{pmatrix}, \qquad M = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \qquad K = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix},$$

and $S(z) = \frac{1}{2}\left((p^2+q^2)^2 + v^2 + w^2\right)$.

The multisymplectic conservation law is

$$\omega_t + \kappa_x = 0 \tag{6.23}$$

where ω and κ are alternating forms

$$\omega(U,V) = \langle MU, V\rangle = V^T MU \quad \text{and} \quad \kappa(U,V) = \langle KU, V\rangle = V^T KU \tag{6.24}$$

for two solutions U and V of the variational equation associated to the PDE (6.22) [10]. An integrator of the system (6.22) is multisymplectic if and only if a discretized version of (6.23) is preserved exactly.

The centered cell discretization [10] is a centered difference approximation in space to (6.22), and implicit midpoint in space, which can be proven to be a multisymplectic scheme. However, in our periodic case, the finite difference discretization in space is inferior to a spectral space discretization, and thus, it has not been included in the numerical tests in this report.

Instead, the centered cell discretization is modified to use a spectral discretization in space. Approximation of space differentiation can be seen as multiplication by the spectral differentiation matrix $\mathcal{D}$ [14],

$$\frac{\mathrm{d}u}{\mathrm{d}x} \approx \mathcal{D}\{u_k\}, \qquad u_k \approx u(x_k), \quad \mathcal{D} = \begin{cases} (-1)^{j-k}\frac{\pi}{D}\cot\frac{\pi}{D}(x_j - x_k) & \text{if } j \neq k \\ 0 & \text{if } j = k \end{cases} \tag{6.25}$$

Replacing all space derivatives in (6.22) with spectral differentiation using $\mathcal{D}$, and modifying the multisymplectic conservation law (6.23) accordingly, [17] proves that this together with implicit midpoint in time yields a multisymplectic scheme for the nonlinear Schrödinger equation. We denote this scheme MSSPECTRAL2. The implicit equation at each time step is solved numerically using a simplified Newton iteration.

Fourth order split step scheme

Split step schemes have been used for a long time for integrating the nonlinear Schrödinger equation in physical applications. In our context, [25] is an early reference to the type of scheme. Taha and Ablowitz [24] give an extensive survey of the prime integrators for the numerical solution of the nonlinear Schrödinger equation, and conclude that Tappert's split step Fourier scheme [25] is in most cases superior to the other schemes considered. The paper [26] investigates numerical aspects of the basic first order split step Fourier method using linearization techniques.

For the split step Fourier method to be comparable to the fourth order exponential integrators used, we first construct a second order scheme by using a Strang splitting technique [23] and the same splitting as for exponential integrators. Let $\Phi^h_{\mathrm{RK4C}}(y(t_0))$ be the approximation to $y(t_0+h)$ produced by Kutta's classical fourth order scheme for the problem $\dot{y}(t) = N(y,t)$. Then a Strang splitting scheme for (6.14) is

$$\Phi^h_{\mathrm{SS2}}(y_0) = \Phi^{h/2}_{\mathrm{RK4C}} \circ \exp(hL) \circ \Phi^{h/2}_{\mathrm{RK4C}}(y_0) \tag{6.26}$$

with L and N given by (6.20) and (6.21). Then we use Yoshida's formula

$$\Phi_{\mathrm{SS4}}(y_0) = \Phi^{c_1 h}_{\mathrm{SS2}} \circ \Phi^{c_0 h}_{\mathrm{SS2}} \circ \Phi^{c_1 h}_{\mathrm{SS2}}(y_0) \tag{6.27}$$

to construct a fourth order split step scheme from the second order Strang scheme, where $c_0 = \frac{-2^{1/3}}{2-2^{1/3}}$ and $c_1 = \frac{1}{2-2^{1/3}}$ [27]. Scheme (6.27) will be denoted SPLITSTEP4 in the remainder of this paper.

SPLITSTEP4 is picked as a representative and well-studied scheme within the class of split step schemes for the nonlinear Schrödinger equation. However, recently symplectic partitioned Runge–Kutta schemes have been constructed for the nonlinear Schrödinger equation which maybe could outperform the SPLITSTEP4 scheme used here, as reported in [9]. In further studies, these new split step schemes should also be compared to exponential integrators.

6.4 Numerical results

The NLS (6.1) equation with a perturbed plane wave solution as the initial condition (6.5) has been integrated with various choices of the spatial discretization parameter $N_{\mathcal{F}}$ and the temporal step size h. Our aim is to determine the time length for which the numerical solution is valid, where the validity is determined by how well the nonlinear spectrum is preserved, and thereby judge the chosen integrators. Table 6.1 indicates the length of time for which the numerical solution was accepted according to criteria based entirely on the nonlinear spectrum and constitutes the main result from this work.

	$h = 10^{-1}$	$h = 10^{-2}$	$h = 10^{-3}$	$h = 10^{-4}$
CFREE4	6.9 (1)	6.9 (1)	6.9 (1)	6.9 (1)
LAWSON4	6.9 (1)	6.9 (1)	6.9 (1)	6.9 (1)
SPLITSTEP4	6.7 (1,2)	6.9 (1)	6.9 (1)	6.9 (1)
MSSPECTRAL2	26.2 (1)	6.9 (1)	6.9 (1)	6.9 (1)

(a) $N_{\mathcal{F}} = 64$

	$h = 10^{-1}$	$h = 10^{-2}$	$h = 10^{-3}$	$h = 10^{-4}$
CFREE4	90.1 (1)	157.9 (1)	157.9 (1)	157.9 (1)
LAWSON4	6.8 (2)	157.9 (1)	157.9 (1)	157.9 (1)
SPLITSTEP4	7 (1,2)	2049 (r)	157.9 (1)	157.9 (1)
MSSPECTRAL2	5.9 (2)	207.8 (1)	158 (1)	157.9 (1)

(b) $N_{\mathcal{F}} = 128$

	$h = 10^{-1}$	$h = 10^{-2}$	$h = 10^{-3}$	$h = 10^{-4}$
CFREE4	90.3 (1)	3165 (r)	> 10000	> 10000
LAWSON4	6.7 (2)	1891 (r)	> 10000	> 10000
SPLITSTEP4	7.1 (1,2)	2400 (r)	> 10000	> 10000
MSSPECTRAL2	5.9 (2)	207.6 (2)	1426 (r)	> 500

(c) $N_{\mathcal{F}} = 256$

	$h = 10^{-1}$	$h = 10^{-2}$	$h = 10^{-3}$	$h = 10^{-4}$
CFREE4	90.3 (1)	3166 (r)	> 10000	> 10000
LAWSON4	6.7 (2)	1993 (r)	> 10000	> 10000
SPLITSTEP4	7.1 (1,2)	907.4 (r)	1479 (r)	> 10000
MSSPECTRAL2	5.9 (2)	109.8 (2)	1426 (r)	n/a

(d) $N_{\mathcal{F}} = 512$

Table 6.1: Time until numerical solution fails. Symbols next to numbers denote reason for failure.

Preservation of nonlinear spectrum

The spectrum of the NLS equation is invariant in time, but the truncation errors incurred by the numerical schemes result in numerical solutions that can be viewed as the exact solutions of corresponding perturbed equations, for which the spectra evolve in time. Thus, the positions of the single/double points and the bands of real discriminant $\Delta(u, \lambda)$, computed by software briefly described in Section 6.2, will in general vary in time. It has been customary in similar studies to monitor conservation of momentum, energy and norm. These three quantities are expressible in terms of the nonlinear spectrum, but alone fail to describe all the properties of the spectrum.

The initial condition is a perturbation of an unstable state, and the numerical challenge is to balance near the border of this instability. We allow for discrepancies that are small enough not to excite any of the unstable modes, or said differently, we avoid *topological* changes to the spectral configuration.

If the gap size becomes zero (gap closure) we obtain a degenerate double point, as in the plane wave solution $\epsilon = 0$, Figure 6.1a, and the solution may then undergo a homoclinic crossing, entering a completely different state. A typical scenario after a gap closure is that eigenvalues at the end of the bands $\lambda_j^{\pm}$ obtain a nonzero real part and the mode enters a "cross" state. This corresponds to the solution having different spatial structure. Spatial symmetry is also typically lost as the modes may start to drift spatially. As symmetry is not enforced in the numerical solutions, accumulation of non-symmetric round-off errors may eventually ruin the numerical solution. In the non-symmetric (non-even) case, it is seen in [1] that the real part of the eigenvalues may grow without being initiated by a homoclinic crossing. The gap sizes and the magnitude of the real part of the single points have therefore been used as the main indicators of the validity of the numerical solution.

The numerical solution at time T is accepted given that the topological properties of the spectrum have remained unchanged for $0 \leq t \leq T$. Let

$$G_j(t) = \lambda_j^+(t) - \lambda_j^-(t), \qquad j = 1, 2, \tag{6.28}$$

denote the gap (complex valued) G_j at time t with reference to Figure 6.1a.

We monitor the deviations in the spectrum $\lambda_j^{\pm}$, examining the real and imaginary components independently. The precise requirements used are

$$\begin{aligned} &\frac{1}{100}|\mathrm{Im}\, G_j(0)| < |\mathrm{Im}\, G_j(t)| < 100|\mathrm{Im}\, G_j(0)| \qquad \text{for } j = 1, 2, \\ &|\mathrm{Re}\, G_j(0)| < 5 \cdot 10^{-5}. \end{aligned} \tag{6.29}$$

The smallest t for which one of these requirements are not met in a given scenario is printed in Table 6.1. The specific numbers 100 and $5 \cdot 10^{-5}$ have been

chosen to reveal the significant differences between the integrators in preserving the vital features of the phase space structure, that is, allowing everything but *topological* changes in the spectrum. The numbers must also be comparable to the eigenvalue accuracy of the spectrum computation, which in our case has been $\mathcal{O}(10^{-8})$. Also note that $|\mathrm{Im}\, G_j(t)|$ never reached its upper limit before the solution was invalidated by the other requirements in our experiments.

Based on the numerical solutions and the accompanying spectra produced, we find that the nonlinear spectrum evolves in four different ways.

(1) Gap 1 closes ($|\mathrm{Im}\, G_1| \approx 0$). Subsequently, $|\mathrm{Re}\, G_1|$ is nonzero. The numerical solution is said to "cross" a homoclinic orbit and enter a different state, often with the center mode spatially shifted half the domain length. This is exemplified in Figure 6.3.

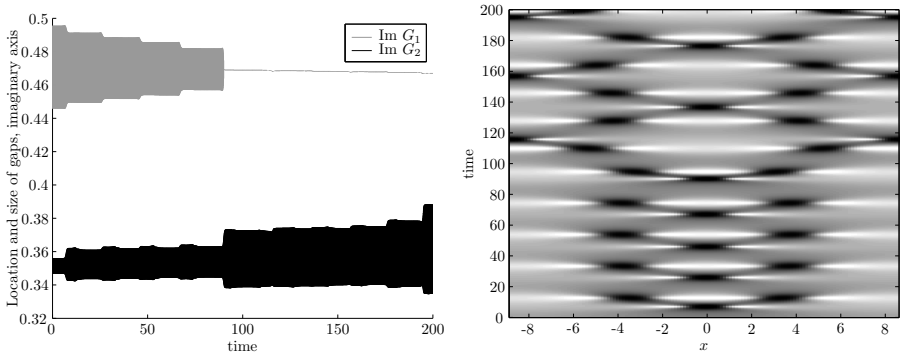
(2) Gap 2 closes ($|\mathrm{Im}\, G_2| \approx 0$). Subsequently, $|\mathrm{Re}\, G_2|$ is nonzero. Gap 2 corresponds to the antiperiodic mode appearing on both sides of the center mode, and also experiences spatial shift during gap closure and homoclinic crossing. This is exemplified in Figure 6.4.

(1,2) Both gaps close. This is similar to the two cases above, but here both gaps close during the time span defined by one peak of a mode. In the scenarios included here, this has been the onset of computational chaos in the numerical solution. This is exemplified in Figure 6.5.

(r) $|\mathrm{Re}\, G_2|$ becomes nonzero, without any gap closures. This only happens on long time scales, and is due to accumulation of non-symmetric round-off errors. The sign of λ_2^+ and λ_2^- is always different, and the sign of λ_2^+ determines the direction (right or left) the corresponding mode will travel in phase space. The real extent determine the speed of spatial drift. In some of the cases, this case is followed by $|\mathrm{Re}\, G_1|$ becoming non-zero as well. This is exemplified in Figure 6.6.

It is evident from the data in Table 6.1 that sufficient spatial resolution is a first prerequisite for a valid long-time computation. Using only $N_{\mathcal{F}} = 64$, G_1 closes during the first peak of the center mode for all integrators independent of temporal step size h, except for MSSPECTRAL2, which fails at the second center mode peak.

Increasing spatial accuracy to $N_{\mathcal{F}} = 128$ one can integrate for a longer period of time, but still, for most configurations G_1 closes early. Of special interest is the exceptional result of SPLITSTEP4 for $h = 10^{-2}$, where gap closures are avoided but the accumulation of non-symmetric round-off error eventually grows large enough to destroy the solution. However, finer experiments indicated that there is a small window for h in which better spectrum preservation is achieved for



(a) Gap evolution (b) $|u(x,t)|$

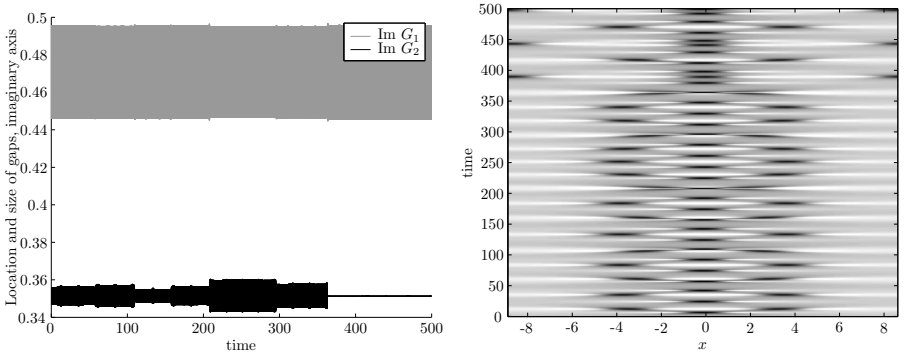
Figure 6.3: Example of failure due to closure of G_1 for CFREE4, $h = 0.1$, $N_{\mathcal{F}} = 128$. At each time slice, the points $\lambda_j^{\pm}$ are located at the end of the bars drawn, and the length of the bar denotes the extent of the gap. In the surface plot to the right, a darker tone indicates higher value, compare with Figure 6.2.

this integrator. In general, for $N_{\mathcal{F}} = 128$ and $h \leq 0.01$, there are no significant differences in the performance of the various schemes.

At $N_{\mathcal{F}} = 256$ the spatial resolution is sufficient to reveal differences in the time-integration. If $h \leq 0.001$, all of the schemes are able to integrate for as long as we tested. CFREE4, LAWSON4, and SPLITSTEP4 all suffer from a growth in the real part for $h = 0.01$, but all are good for smaller step sizes. This non zero real part is visible in the surface plots as a spatial drift in the modes. MSSPECTRAL2 behaves well for $N_{\mathcal{F}} = 128$ for $h = 0.01$ but suffers from a growth in the real part for $h = 0.001$ where the other integrators do not experience as much growth. Due to the computational complexity, MSSPECTRAL2 has not been computed over as long time intervals as the others.

Not much is gained by increasing the spatial resolution to $N_{\mathcal{F}} = 512$, but one should note that this turns out to be more difficult for SPLITSTEP4, as we observe a spatial drift for $h = 0.001$ which does not occur at $N_{\mathcal{F}} = 256$. Further, the validity time is smaller for the higher resolution of $N_{\mathcal{F}} = 512$ for $h = 0.01$ and $h = 0.001$. It is only in this case that SPLITSTEP4 is significantly less appealing than the exponential integrators.

The experiments have all been run on Intel Pentium IV processors using MATLAB Release 14SP3 (Version 7.1). During the experiments we noted that in some scenarios Release 14SP3 gave different numerical results (usually slightly worse) than those obtained with the same code and processor but using MATLAB Release



(a) Gap evolution (b) $|u(x,t)|$

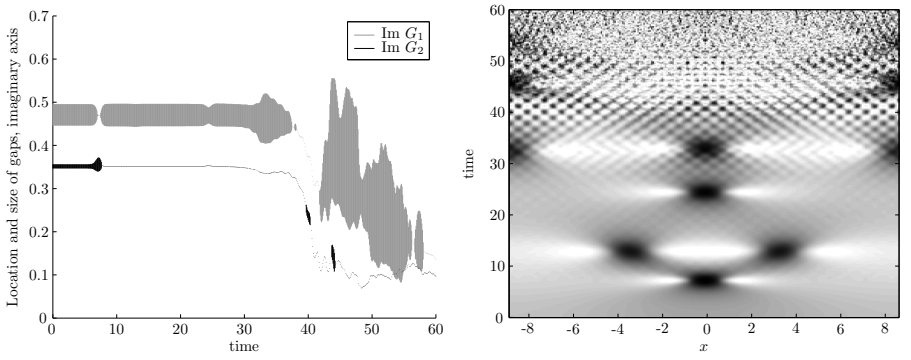
Figure 6.4: Example of failure due to closure of G_2. MSSPECTRAL2, $h = 0.01$, $N_{\mathcal{F}} = 256$.

13 (Version 6.5). Effects due to round-off errors, as in this case with the growth in real part, is more prone to differ between releases of MATLAB, and also possibly differ with the specific hardware used.

Computational time

Measuring computational complexity is a difficult task, and the results in this section should only be taken as an indication. Time has been measured by the built-in `cputime` command in MATLAB. The exponential integrators have been implemented using the EXPINT-package, and thus it incurs an overhead in that the package is designed modularly. A specific exponential integrator applied to a specific problem could be hand-crafted and would results in speedup for that exponential integrator and problem. In the exponential integrators, the time step is constant, which facilitates caching of the exponential function of the linear part and the φ functions. This is automatically taken care of by the EXPINT-package, and is also crucial for an exponential integrator implementation of this type. The multisymplectic code is to a certain degree already tailored to the problem in question, but nevertheless, this code is probably the one which could gain the most relative performance increase from optimization and tuning in the root solver. However, it is not believed that any optimization performed on the code for MSSPECTRAL2 will make any substantial changes to the results obtained in this work.

Table 6.2 contains timing data measured in steps per second with varying time step and integrator. Exponential integrators should not be significantly



(a) Imaginary part of gap sizes as a function of time

(b) $|u(x,t)|$

Figure 6.5: Example of failure due to closure of G_1 and G_2. SPLITSTEP4, $h = 0.01$, $N_{\mathcal{F}} = 256$. Development of computational chaos as observed here is typical for SPLITSTEP4.

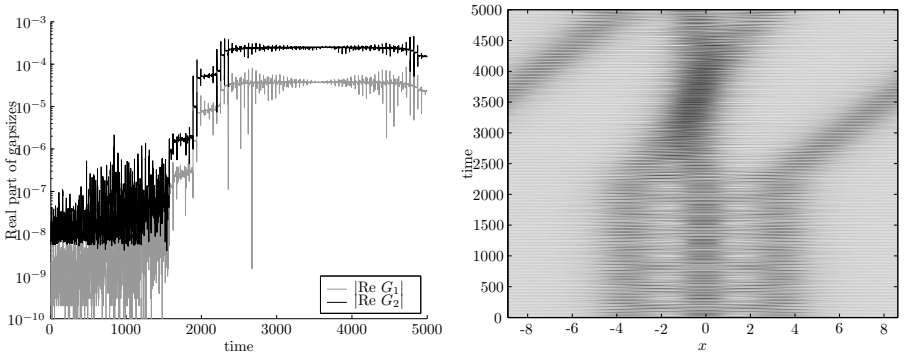
dependent on time step, but the multisymplectic integrator is, due to easier solvability of the root problem for decreasing h.

6.5 Discussion

In this study, we have used inverse spectral method as a tool to determine whether a solution obtained numerically for different integrators and discretization parameters is acceptable. We have integrated initial conditions ϵ-close to unstable states, which makes the problem hard numerically, as truncation errors errors from the space discretization, truncation errors from time-integration and round-off errors in the computer may eventually force the numerical solution to enter another state and then diverge from the exact solution. An unacceptable solution in this context means that the spectrum of the solution has changed topologically from its initial state, possibly through homoclinic crossings.

We tested the integrators CFREE4, LAWSON4, SPLITSTEP4, and MSSPECTRAL2, the last one being a second order implicit multisymplectic integrator. In short, CFREE4 was shown to exhibit the most stable properties in terms of being able to integrate for a long time avoiding topological changes to the spectrum. In addition, it is the computationally fastest integrator for given discretization parameters.

The two exponential integrators outperformed the other schemes. CFREE4 appeared slightly more stable than LAWSON4, perhaps attributable to higher stiff order, or its preservation of fixed points of the differential equation. SPLITSTEP4,



(a) Real part of gap sizes as a function of time

(b) Plot of $|u(x,t)|$

Figure 6.6: Example of failure due to breakdown of symmetry, nonzero real part of single points $\lambda_j^{\pm}$. LAWSON4, $h = 0.01$, $N_{\mathcal{F}} = 256$. The imaginary extent of the gaps is well preserved in this case.

being related to LAWSON4, was comparable, but performed less reliably than the exponential integrators. Its performance was not monotone in terms of spatial and temporal resolution.

The multisymplectic integrator MSSPECTRAL2, which gave good results on the one-mode case in [17], was not able to match the other schemes in this study, both in terms of preservation of spectrum and especially in terms of computational complexity.

The nature and computational demands of these experiments dictated that all possibilities could not be tested, and not all scenarios could be integrated until breakdown. The experiments could have been performed with additional configurations, possibly revealing more information on for instance SPLITSTEP4's peak performance on $N_{\mathcal{F}} = 128$ and $h = 0.01$. Also, there is a multitude of alternative exponential integrators that probably would have performed along the lines of CFREE4, at least those with stiff order at least 2. The conclusion here is more to advocate the use of exponential integrators, more than to advocate the use of the specific CFREE4 scheme.

6.6 Acknowledgements

Håvard Berland is grateful for the opportunity to stay at Department of Mathematics, University of Central Florida, Orlando US, with Dr. Constance Schober and Dr. Alvaro Islas, during autumn 2005.

Integrator	$h = 0.1$	$h = 0.01$	$h = 0.001$
CFREE4	1020.4	1089.9	1091.2
LAWSON4	913.2	1119.2	1114.1
SPLITSTEP4	423.7	435.4	443.7
MSSPECTRAL2	69.7	102.1	115.5

(a) 64 Fourier modes

Integrator	$h = 0.1$	$h = 0.01$	$h = 0.001$
CFREE4	593.5	649.4	656.8
LAWSON4	630.9	674.3	678.2
SPLITSTEP4	211.2	215.4	226.5
MSSPECTRAL2	2.6	4.3	5.1

(b) 256 Fourier modes

Table 6.2: Number of integration steps per CPU second, 2.4GHz Intel Pentium IV.

Bibliography

[1] Ablowitz, M.J., Herbst, B.M., Schober, C.M.: Computational chaos in the nonlinear schrödinger equation without homoclinic crossings. Physica A **228**, 212–235 (1996)

[2] Ablowitz, M.J., Schober, C.M.: Effective chaos in the nonlinear schrödinger equation. Contemporary Mathematics **172**, 253–268 (1994)

[3] Belokolos, E.D., Bobenko, A.I., Enol'skii, V.Z., Its, A.R., Matveev, V.B.: Algebro-Geometric Approach to Nonlinear Integrable Equations. Springer series in nonlinear dynamics. Springer-Verlag, Berlin (1994)

[4] Benjamin, T.B., Feir, J.E.: The disintegration of wave trains on deep water. J. Fluid Mech. **27**(3), 417–430 (1967)

[5] Berland, H., Owren, B., Skaflestad, B.: B-series and order conditions for exponential integrators. SIAM J. Numer. Anal. **43**(4), 1715–1727 (2005)

[6] Berland, H., Owren, B., Skaflestad, B.: Solving the nonlinear Schrödinger equation using exponential integrators. Modeling, Identification and Control **27**(4) (2006)

[7] Berland, H., Skaflestad, B., Wright, W.M.: Expint — A Matlab package for exponential integrators. ACM Trans. on Math. Soft. (2006). To appear

[8] Berland, H., Skaflestad, B., Wright, W.M.: Scaling and squaring of φ functions in exponential integrators (2006). In preparation

[9] Blanes, S., Moan, P.C.: Practical symplectic partitioned Runge–Kutta and Runge–Kutta–Nyström methods. J. Comput. Appl. Math. **142**(2), 313–330 (2002)

[10] Bridges, T.J., Reich, S.: Multi-symplectic integrators: numerical schemes for Hamiltonian PDEs that conserve symplecticity. Phys. Lett. A **284**(4-5), 184–193 (2001)

[11] Cox, S.M., Matthews, P.C.: Exponential time differencing for stiff systems. J. Comput. Phys. **176**(2), 430–455 (2002)

[12] Ercolani, N.M., Forest, M.G., McLaughlin, D.W.: Geometry of the modulational instability III. Homoclinic orbits. Physica D **43**(349) (1990)

[13] Ercolani, N.M., McLaughlin, D.W.: Toward a topological classification of integrable PDEs. In: R. Devaney, H. Flaschka, W. Meyer, T. Ratiu (eds.) MSRI Proc. Workshop on Symplectic Geometry (1990)

[14] Fornberg, B.: A practical guide to pseudospectral methods, *Cambridge Monographs on Applied and Computational Mathematics*, vol. 1. Cambridge University Press, Cambridge (1996)

[15] Hochbruck, M., Ostermann, A.: Explicit exponential Runge–Kutta methods for semilinear parabolic problems. SIAM J. Numer. Anal. **43**(3), 1069–1090 (2005)

[16] Islas, A.L., Karpeev, D.A., Schober, C.M.: Geometric integrators for the nonlinear Schrödinger equation. J. of Comp. Phys. **173**, 116–148 (2001)

[17] Islas, A.L., Schober, C.M.: On the preservation of phase space structure under multisymplectic discretization. J. of Comp. Phys. **197**, 585–609 (2004)

[18] Its, A.R., Kotljarov, V.P.: Explicit formulas for solutions of a nonlinear schrödinger equation. Dokl. Akad. Nauk Ukrain. SSR Ser. A **1051**, 965–968 (1976)

[19] Koikari, S.: An error analysis of the modified scaling and squaring method (2005). Submitted to Computers Math. Applic.

[20] Krichever, I.M.: Methods of algebraic geometry in the theory of nonlinear equations. Russian Math. Surv. **32**, 185–213 (1977)

[21] McLaughlin, D.W., Overman, E.A.: Whiskered tori for integrable PDEs: chaotic behavior in near integrable PDEs. In: Surveys in applied mathematics, Vol. 1, *Surveys Appl. Math.*, vol. 1, pp. 83–203. Plenum, New York (1995)

[22] Ostermann, A., Thalhammer, M., Wright, W.M.: A class of explicit exponential general linear methods. BIT **46**(2), 409–432 (2006)

[23] Strang, G.: On the construction and comparison of difference schemes. SIAM J. Numer. Anal. **5**, 506–517 (1968)

[24] Taha, T.R., Ablowitz, M.J.: Analytical and numerical aspects of certain nonlinear evolution equations. II. Numerical, nonlinear Schrödinger equation. J. Comput. Phys. **55**(2), 203–230 (1984)

[25] Tappert, F.D.: Numerical solutions of the Korteweg–de Vries equation and its generalizations by the split-step Fourier method. Lect. Appl. Math. Am. Math. Soc. **15**, 215–216 (1974)

[26] Weideman, J.A.C., Herbst, B.M.: Split-step methods for the solution of the nonlinear Schrödinger equation. SIAM J. Numer. Anal. **23**(3), 485–507 (1986)

[27] Yoshida, H.: Construction of higher order symplectic integrators. Physics Letters A **150**, 262–268 (1990)

[28] Zakharov, V.E., Shabat, A.B.: Exact theory of two-dimensional self-focusing and one-dimensional self-modulation of waves in nonlinear media. Ž. Èksper. Teoret. Fiz. **61**(1), 118–134 (1971)

Paper VI

Håvard Berland:

Generalized affine groups in exponential integrators

Preprint Numerics No. 2/06, Department of Mathematical Sciences, Norwegian University of Science and Technology.

Chapter 7

Generalized affine groups in exponential integrators

Abstract:

Most exponential integrators use functions similar to the exponential in their implementation. This report describes a structure in which these function play a central role, and the framework developed herein is believed to be applicable to further studies of these functions. The framework is constructed using ideas from Lie group methods, with generalized affine groups as the fundamental building block, for the solution of non-autonomous systems of ordinary differential equations. This work can be seen as an extension to work by Minchev [10].

7.1 Introduction

Exponential integrators are numerical schemes tailored for systems of ordinary differential equations of the type

$$y'(t) = Ly + N(y,t), \qquad y(t_0) = y(0) \tag{7.1}$$

in which L is a linear operator and $N(y,t)$ is a (possibly nonlinear) function. If, for instance, L is such that (7.1) is stiff, classical explicit Runge–Kutta integrators encounter step size restrictions, and the aim of exponential integrators is to enable the use of explicit schemes without step size restrictions.

A building block of exponential integrators is the "φ functions", which we define by the integral representation

$$\varphi_j(z) = \frac{1}{(j-1)!}\int_0^1 \mathrm{e}^{(\theta-1)z}\theta^{j-1}\,\mathrm{d}\theta, \qquad j = 1,2,\ldots, \tag{7.2}$$

which for $j = 1,2,3$ (and for $z \neq 0$) can be calculated as

$$\varphi_1(z) = \frac{\mathrm{e}^z - 1}{z}, \quad \varphi_2(z) = \frac{\mathrm{e}^z - z - 1}{z^2} \quad \text{and} \quad \varphi_3(z) = \frac{\mathrm{e}^z - z^2/2 - z - 1}{z^3}.$$

Numerous exponential integrators exist for the equation (7.1), we refer to [3] and its accompanying software for a list. Most of them make use of φ functions, but the Lawson integrators are examples that do not [8, 2].

Lie group integrators may be extended to exponential integrators through the affine group (of degree one) $\mathrm{GL}_n(\mathbf{R}) \rtimes \mathbf{R}^n$ and its affine action on $\mathbf{R}^n$. This was already mentioned in [11]. The simplest affine Lie group integrator for (7.1) is then

$$y_{n+1} = \mathrm{e}^{hL} y_n + h\varphi_1(hL)N(y_n, t_n) \tag{7.3}$$

where $y_n \approx y(t_n)$. This scheme can be called Lie–Euler in our context, but is also known as Nørsett–Euler, ETD Euler, filtered Euler, exponentially fitted Euler etc.

The important feature of (7.3) in this context is the use of the function φ_1. ϕ functions (7.2) is a class of functions that are all exponential-like, but has not been as extensively studied in mathematical literature such as the exponential function itself.

The motivation for this work was to construct a group around the φ functions in order to reveal further properties. All details on these generalized affine groups are developed in Section 7.2. A direct application of Corollary 7.8 herein already exists for the implementation of the scaling and squaring approach of evaluating φ functions in [3].

The report [10] has the same framework up to degree $d = 2$, and focuses mainly on constructing Lie group integrators using the higher order affine groups. This report owes a lot to [10] and [7] for ideas and inspiration. Section 7.3 exemplifies how this framework can be applied for $d = 2$ in order to construct Lie group integrators.

7.2 Generalized affine groups and their Lie algebras

The well-known affine group, $\mathrm{GL}_n(\mathbf{R}) \rtimes \mathbf{R}^n$ and its corresponding group action on $\mathbf{R}^n$, $(A, b) \cdot y = Ay + b$, is in this section generalized to higher degree by incorporating further $\mathbf{R}^n$-vectors into the group. The basic affine group is denoted degree $d = 1$, as the group has one $\mathbf{R}^n$ vector. One perhaps unexpected necessity for higher order groups, is the introduction of an additional scalar, λ, which is needed for this framework to be able to solve non-autonomous differential equations. The parameter λ is used in the algebra to maintain a notion of unity time.

We start with Lemma 7.1 which is well known in recent texts on exponential integrators. φ functions appear in this lemma as a convenient symbol for a recurring expression. Then an affine algebra, an affine group, an exponential and logarithm connecting the algebra and the group is defined. Lemma 7.5 then proves that this framework is able to solve the differential equation (7.1).

Lemma 7.7 is an application of the group structure which gives meaning to φ functions on sums. It has the immediate Corollary 7.8 which is already useful for the current numerical implementation of φ functions, and also Corollary 7.9 which is yet to be applied for future backward error analysis of φ functions.

Lemma 7.1. *The nonautonomous differential equation*

$$y'(t) = \alpha y + \sum_{j=0}^{d-1} \frac{t^j}{j!}\beta^{[j]}, \qquad y(t_0) = y_0 \tag{7.4}$$

for $y(t) \in \mathbf{R}^n$, $\alpha \in \mathbf{R}^{n\times n}$, $\beta^{[j]} \in \mathbf{R}^n$, *has the exact solution*

$$y(t_0 + h) = \mathrm{e}^{h\alpha} y_0 + \sum_{j=0}^{d-1} \sum_{k=0}^{j} \frac{t_0^{j-k}}{(j-k)!} h^{k+1} \varphi_{k+1}(h\alpha)\beta^{[j]} \tag{7.5}$$

with φ_k *defined in* (7.2).

Proof. Differentiate $y(t_0 + h)$ in (7.5) with respect to h and set $h = 0$. □

Definition 7.2 (Lie affine algebra)**.** We define the Lie affine algebra of degree d as containing the elements $(\alpha, \beta^{[0]}, \ldots, \beta^{[d-1]}, \lambda) \in \mathrm{M}_n(\mathbf{R}) \times (\mathbf{R}^n)^d \times \mathbf{R}$. Addition and scalar multiplication are defined trivially elementwise. The bracket operation is

$$\Big[(\alpha_1, \beta_1^{[0]}, \ldots, \beta_1^{[d-1]}, \lambda_1), (\alpha_2, \beta_2^{[0]}, \ldots, \beta_2^{[d-1]}, \lambda_2)\Big] = ([\alpha_1, \alpha_2], \gamma^{[0]}, \ldots, \gamma^{[d-1]}, 0) \tag{7.6}$$

where

$$\gamma^{[j]} = \alpha_1\beta_2^{[j]} - \alpha_2\beta_1^{[j]} + \sum_{k=1}^{d-1-j} \left(\beta_1^{[j+k]} \frac{\lambda_2^k}{k!} - \beta_2^{[j+k]} \frac{\lambda_1^k}{k!}\right).$$

This algebra is a Lie algebra as both the skew-symmetry and the Jacobi identity can be proved. This is not surprising due to the connection with the vector field in (7.4) which will be presented in Lemma 7.5.

The scalar λ in the algebra is necessary for $d \geq 2$ to have a notion of unity time when this algebra is used to solve non-autonomous differential equations. Choosing $\lambda \neq 1$ and rescaling $\beta^{[j]}$'s accordingly, is equivalent to scaling the time parameter in (7.4), as will be demonstrated in Remark 7.6. For $d = 1$, the parameter λ is insignificant and could be removed from the definitions and expressions.

Definition 7.3 (Affine group)**.** The affine group of degree d is defined as containing the elements $(A, b^{[0]}, \ldots, b^{[d-1]}, \lambda) \in \mathrm{GL}_n(\mathbf{R}) \rtimes (\mathbf{R}^n)^d \times \mathbf{R}$.

The group product is

$$(A_2, b_2^{[0]}, \ldots, b_2^{[d-1]}, \lambda_2) \cdot (A_1, b_1^{[0]}, \ldots, b_1^{[d-1]}, \lambda_1) = (A_2 A_1, \xi^{[0]}, \ldots, \xi^{[d-1]}, \lambda_1 + \lambda_2) \tag{7.7}$$

where

$$\xi^{[j]} = A_2 b_1^{[j]} + \sum_{k=0}^{d-1-j} b_2^{[j+k]} \frac{\lambda_1^k}{k!}.$$

The group identity is written

$$\mathrm{Id} = (I, \mathbf{0}, \ldots, \mathbf{0}, 0).$$

Moreover the group inverse is

$$(A, b^{[0]}, \ldots, b^{[d-1]}, \lambda)^{-1} = (A^{-1}, \chi^{[0]}, \ldots, \chi^{[d-1]}, -\lambda) \tag{7.8}$$

where

$$\chi^{[j]} = -A^{-1} \sum_{k=0}^{d-1-j} b^{[j+k]} \frac{(-\lambda)^k}{k!}.$$

Elements of the affine group act on $\mathbf{R}^n \times \mathbf{R}$ as follows

$$(A, b^{[0]}, \ldots, b^{[d-1]}, \lambda) \cdot (y, t) = \left(Ay + \sum_{j=0}^{d-1} b^{[j]} \frac{t^j}{j!}, t + \lambda \right). \tag{7.9}$$

We define the following map from the affine algebra to the affine group,

$$\mathrm{Exp}(\alpha, \beta^{[0]}, \ldots, \beta^{[d-1]}, \lambda) = (\mathrm{e}^{\alpha}, b^{[0]}, \ldots, b^{[d-1]}, \lambda) \tag{7.10}$$

where

$$b^{[j]} = \sum_{k=0}^{d-1-j} \lambda^k \varphi_{k+1}(\alpha) \beta^{[k+j]}.$$

Lemma 7.5 establishes that this map really is the exponential as it represents the one-parameter subgroups in the affine group.

Lemma 7.4 (Logarithm map)**.**

$$\mathrm{Log}(A, b^{[0]}, \ldots, b^{[d-1]}, \lambda) = (\log(A), \beta^{[0]}, \ldots, \beta^{[d-1]}, \lambda)$$

where

$$\beta^{[j]} = \sum_{k=0}^{d-1-j} \lambda^k c_{k+1} b^{[k+j]}.$$

The coefficients c_k obey the recursive formula

$$c_{k+1} = -\varphi_1^{-1}(\log A)\sum_{i=0}^{k-1}\varphi_{k+1-i}(\log A)c_{i+1}, \qquad k \geq 1, \quad c_1 = \varphi_1^{-1}(\log A). \tag{7.11}$$

A must be sufficiently close to the identity $I \in \mathrm{GL}_n(\mathbf{R})$.

The first few c_k coefficients are

$$\begin{aligned}
c_1 &= \varphi_1^{-1}\\
c_2 &= -\varphi_1^{-2}\varphi_2\\
c_3 &= \varphi_1^{-3}\varphi_2^2 - \varphi_1^{-2}\varphi_3\\
c_4 &= -\varphi_1^{-2}\varphi_4 + 2\varphi_1^{-3}\varphi_2\varphi_3 - \varphi_1^{-4}\varphi_2^3\\
c_5 &= -\varphi_1^{-2}\varphi_5 + \varphi_1^{-3}\varphi_2\varphi_4 - \varphi_1^{-4}\varphi_2^2\varphi_3 + \varphi_1^{-3}\varphi_3^2 + \varphi_1^{-5}\varphi_2\varphi_4 - 2\varphi_1^{-4}\varphi_2^2\varphi_3 + \varphi_1^{-2}\varphi_2^4
\end{aligned}$$

where all φ functions and their inverses are evaluated at $\log A$.

Proof. Assume d fixed. We will prove that the recursion (7.11) is necessary for $\mathrm{Exp}\circ\mathrm{Log} = \mathrm{Id}$ to hold. In this proof, all φ functions are evaluated at $\log A$.

$$\begin{aligned}
\mathrm{Exp}\circ\mathrm{Log}(A, b^{[0]}, \ldots, b^{[d-1]}, \lambda) &= (A, b^{[0]}, \ldots, b^{[d-1]}, \lambda)\\
\mathrm{Exp}(\log A, \beta^{[0]}, \ldots, \beta^{[d-1]}, \lambda) &= (A, b^{[0]}, \ldots, b^{[d-1]}, \lambda)\\
(A, \bar{b}^{[0]}, \ldots, \bar{b}^{[d-1]}, \lambda) &= (A, b^{[0]}, \ldots, b^{[d-1]}, \lambda)
\end{aligned}$$

which means that $\bar{b}^{[j]} = b^{[j]}$ must hold for $0 \leq j \leq d-1$. Furthermore

$$\begin{aligned}
\bar{b}^{[j]} &= \sum_{k=0}^{d-1-j}\lambda^k\varphi_{k+1}\beta^{[j+k]}\\
&= \sum_{k=0}^{d-1-j}\lambda^k\varphi_{k+1}\sum_{i=0}^{d-1-(j+k)}\lambda^i c_{i+1}b^{[i+j+k]}\\
&= \sum_{m=0}^{d-1-j}\lambda^m\left[\sum_{i=0}^{m}\varphi_{m-i+1}c_{i+1}\right]b^{[j+m]}
\end{aligned}$$

using $m = k+i$. Letting $j = d-1$, this immediately yields $c_1 = \varphi_1^{-1}$ if $\bar{b}^{[d-1]} = b^{[d-1]}$. Require $\bar{b}^{[d-1-k]} = b^{[d-1-k]}$ for $k = 1, 2, \ldots, d-1$. In

$$\bar{b}^{[d-1-k]} = \sum_{m=0}^{k}\lambda^m\sum_{i=0}^{m}\varphi_{m-i+1}c_{i+1}b^{[d-1-k+m]}$$

the coefficient in front of λ^k must be zero. This means

$$0 = \sum_{i=0}^{k} \varphi_{k+1-i} c_{i+1} = \varphi_1 c_{k+1} + \sum_{i=0}^{k-1} \varphi_{k+1-i} c_{i+1}$$

and immediately gives the recursion (7.11). A similar calculation gives that (7.11) must also hold for $\mathrm{Log} \circ \mathrm{Exp} = \mathrm{Id}$. We have used that all φ functions and their inverses commute with each other. □

The proof of Lemma 7.4 tacitly assumes that the affine algebra of degree $d-1$ is a subalgebra of the affine algebra of degree d, and similarly for the groups. The proof of this is omitted, but is easily seen in the matrix representation in Remark 7.10.

Lemma 7.5.

$$\mathrm{Exp}(h(\alpha, \beta^{[0]}, \ldots, \beta^{[d-1]}, 1)) \cdot (y(t_0), t_0) = (y(t_0 + h), t_0 + h)$$

where $y(t)$ satisfies (7.4).

Proof. Using the exponential map (7.10) and the group action (7.9) we write

$$\mathrm{Exp}(h(\alpha, \beta^{[0]}, \ldots, \beta^{[d-1]}, 1)) \cdot (y(t_0), t_0) = \left(\mathrm{e}^{h\alpha} y(t_0) + \sum_{j=0}^{d-1} \sum_{k=0}^{d-1-j} h^k \varphi_{k+1}(h\alpha) h \beta^{[k+j]} \frac{t_0^j}{j!} \,,\, t_0 + h \right)$$

Changing outer summation variable from j to $i = k + j$ we obtain

$$\left(\mathrm{e}^{h\alpha} y(t_0) + \sum_{i=0}^{d-1} \sum_{k=0}^{i} \frac{t_0^{i-k}}{(i-k)!} h^{k+1} \varphi_{k+1}(h\alpha) \beta^{[i]} \,,\, t_0 + h \right)$$

in which the first element corresponds to $y(t_0 + h)$ in (7.5). □

Remark 7.6. If we scale time in (7.4) by $t = \gamma\tau$, we obtain the equation

$$\tilde{y}'(\tau) = \gamma\alpha y + \sum_{j=0}^{d-1} \frac{\tau^j}{j!} \gamma^{j+1} \beta^{[j]}, \qquad \tilde{y}(\tau_0) = y(t_0) = y_0$$

Using this and Lemma 7.5, one sees that

$$\mathrm{Exp}(h(\alpha, \beta^{[0]}, \gamma\beta^{[1]}, \ldots, \gamma^{d-1}\beta^{[d-1]}, 1/\gamma)) \cdot (\tilde{y}(\tau_0), \tau_0) = (y(t_0 + h), t_0 + h)$$

thus illustrating a one-parameter isotropy subalgebra [1, 9] parametrized by γ.

Lemma 7.7.

$$\varphi_\ell((\gamma_1+\gamma_2)\alpha) = \frac{1}{(\gamma_1+\gamma_2)^\ell}\left(\mathrm{e}^{\gamma_1\alpha}\gamma_2^\ell\varphi_\ell(\gamma_2\alpha) + \sum_{k=1}^{\ell}\frac{\gamma_1^{\ell-k}\gamma_2^k}{(\ell-k)!}\varphi_k(\gamma_2\alpha)\right)$$

where γ_1 and γ_2 are real numbers.

Proof. Let $\bar\gamma = \gamma_1+\gamma_2$. By the homomorphism property of the exponential, we have

$$\begin{aligned}\mathrm{Exp}(\bar\gamma(\alpha,\beta^{[0]},\ldots,\beta^{[d-1]},\lambda)) = \\ \mathrm{Exp}(\gamma_1(\alpha,\beta^{[0]},\ldots,\beta^{[d-1]},\lambda))\cdot\mathrm{Exp}(\gamma_2(\alpha,\beta^{[0]},\ldots,\beta^{[d-1]},\lambda))\end{aligned} \quad (7.12)$$

The left hand side of (7.12) is the group element $\left(\mathrm{e}^{\bar\gamma\alpha}, b_L^{[0]},\ldots,b_L^{[d-1]},\bar\gamma\lambda\right)$ where

$$b_L^{[j]} = \sum_{k=0}^{d-1-j}(\bar\gamma\lambda)^k\varphi_{k+1}(\bar\gamma\alpha)\bar\gamma\beta^{[k+j]}$$

and the right hand side of (7.12) will be $(\mathrm{e}^{\bar\gamma\alpha}, b_R^{[1]},\ldots,b_R^{[d-1]},\bar\gamma\lambda)$ where

$$\begin{aligned}b_R^{[j]} = \mathrm{e}^{\gamma_1\alpha}&\left(\sum_{k=0}^{d-1-j}(\gamma_2\lambda)^k\varphi_{k+1}(\gamma_2\alpha)\gamma_2\beta^{[k+j]}\right)\\ &+\sum_{k=0}^{d-1-j}\frac{(\gamma_1\lambda)^k}{k!}\sum_{i=0}^{d-1-(j+k)}(\gamma_2\lambda)^i\varphi_{i+1}(\gamma_2\alpha)\gamma_2\beta^{[i+j+k]}\end{aligned}$$

in which the double sum is rearranged using $l=i+k$ to obtain

$$\begin{aligned}b_R^{[j]} = \mathrm{e}^{\gamma_1\alpha}&\left(\sum_{k=0}^{d-1-j}(\gamma_2\lambda)^k\varphi_{k+1}(\gamma_2\alpha)\gamma_2\beta^{[k+j]}\right)\\ &+\sum_{l=0}^{d-1}\frac{(\gamma_1\lambda)^{(l-i)}}{(l-i)!}\sum_{i=0}^{d-1-l}(\gamma_2\lambda)^i\varphi_{i+1}(\gamma_2\alpha)\gamma_2\beta^{[i+l]}.\end{aligned}$$

The relation for φ_ℓ, $1\le\ell\le d$ is now obtained by comparing the coefficients of λ^ℓ in $b_L^{[0]} = b_R^{[0]}$. □

The φ functions possess a double-angle relation. This relation is a building block of the scaling and corrected squaring approach of calculating the φ functions numerically.

Corollary 7.8 (Squaring of φ functions). *The φ functions* (7.2) *have the following squaring property,*

$$\varphi_\ell(2\alpha) = \frac{1}{2^\ell}\left(\mathrm{e}^\alpha \varphi_\ell(\alpha) + \sum_{k=1}^{\ell} \frac{1}{(\ell-k)!}\varphi_k(\alpha)\right)$$

Proof. Set $\gamma_1 = \gamma_2 = 1$ in Lemma 7.7. □

This squaring property is also attainable by dividing the integration interval in (7.2) in half and then shifting the integration variable in the second half, as outlined in [6].

The following corollary will be applied in further studies on φ functions.

Corollary 7.9.

$$\mathrm{e}^{-\alpha}\varphi_\ell(\alpha) = \sum_{k=1}^{\ell} \frac{(-1)^{k+1}}{(\ell-k)!}\varphi_k(-\alpha)$$

Proof. Set $\gamma_1 = -1$ and $\gamma_2 = 1$ in Lemma 7.7. □

Remark 7.10. According to Ado's theorem, we have that the Lie affine algebra and affine group are isomorphic to a subalgebra of matrices in $\mathrm{M}_{n+d}(\mathbf{R})$ and to a subgroup of $\mathrm{GL}_{n+d}(\mathbf{R})$. The isomorphism for the Lie affine algebra can be represented by the map

$$(\alpha, \beta^{[0]}, \ldots, \beta^{[d-1]}, \lambda) \mapsto \begin{pmatrix} \alpha & \beta^{[0]} & \beta^{[1]} & \cdots & \beta^{[d-1]} \\ \mathbf{0}^T & 0 & 0 & \cdots & 0 \\ \mathbf{0}^T & \lambda & \ddots & & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ \mathbf{0}^T & 0 & \cdots & \lambda & 0 \end{pmatrix} \tag{7.13}$$

where $\mathbf{0}$ is a column vector of size n.

The isomorphism for the affine group can be represented by the map

$$(A, b^{[0]}, \ldots, b^{[d-1]}, \lambda) \mapsto \begin{pmatrix} A & b^{[0]} & b^{[1]} & \ldots & & b^{[d-1]} \\ \mathbf{0}^T & 1 & 0 & \ldots & & 0 \\ \mathbf{0}^T & \lambda & \ddots & & & \\ \vdots & & \ddots & \ddots & & \\ \mathbf{0}^T & \frac{\lambda^{d-1}}{(d-1)!} & & \ddots & \ddots & 0 \\ \mathbf{0}^T & \frac{\lambda^d}{d!} & \frac{\lambda^{d-1}}{(d-1)!} & \ldots & \lambda & 1 \end{pmatrix} \tag{7.14}$$

where elements below the diagonal are

$$c_{i,j} = \frac{\lambda^{i-j}}{(i-j)!} \qquad \text{for } 2 \le j \le d+1 \text{ and } i \ge j.$$

7.3 Affine Lie group integrators

In this section, we extend the Lie group integrators presented in [11] using a second degree affine group on a scalar equation. This is an exemplification of the results presented in Section 7.2.

Let a non-autonomous differential equation be given by

$$y'(t) = \alpha y + \beta^{[0]} + t\beta^{[1]} \qquad y(t_0) = y_0 \in \mathbf{R}^n, \quad \alpha \in \mathbf{R}^{n\times n}, \quad \beta^{[j]} \in \mathbf{R}^n \tag{7.15}$$

In order to construct a Runge–Kutta–Munthe-Kaas (RKMK) scheme, one needs to define a configuration space (a manifold M) for the solution, a map from the configuration space to an algebra $\mathfrak{g}$, and an action from the algebra on the configuration space. Our problem is non-autonomous, so the time parameter t must be included in the configuration space, we write $M = \mathbf{R}^n \times \mathbf{R}$. The map $f\colon M \to \mathfrak{g}$ can be chosen to be

$$f\colon (y,t) \mapsto (\alpha, \beta^{[0]}, \beta^{[1]}, 1) \tag{7.16}$$

Alternative maps using the isotropy subalgebra are given in Remark 7.6.

In RKMK schemes, calculations (inner stages) are performed in the algebra $\mathfrak{g}$ and the result is subsequently obtained via the algebra action which factors into the exponential map and the group action. If we want to implement Lie–Euler for the problem (7.15) using (7.16), we would get the scheme

$$(y_{n+1}, t_{n+1}) = \operatorname{Exp}(h(\alpha, \beta^{[0]}, \beta^{[1]}, 1)) \cdot (y_n, t_n) \tag{7.17}$$

and by inserting explicit expressions from (7.10) and (7.9) (or from Section 7.A), one obtains

$$(y_{n+1}, t_{n+1}) = \left(\mathrm{e}^{h\alpha} y_n + h\varphi_1(h\alpha)\beta^{[0]} + h\varphi_1(h\alpha)\beta^{[1]} + h^2\varphi_2(h\alpha)\beta^{[1]}, t_n + h\right) \tag{7.18}$$

which can be coined Lie–Euler of degree 2.

The report [10] constructs a fourth order commutator free scheme using a second degree ($d = 2$) affine group for a non-autonomous problem. The approach there to find $f\colon M \to \mathfrak{g}$ is to linearize the nonlinear function,

$$N(y,t) \approx N_n + t\frac{N(y,t) - N_n}{t} = N^{[0]} + tN^{[1]}$$

where $N_n = N(y_n, t_n)$. Given this, a commutator free scheme of degree 2 is constructed using the exponential map and group action that corresponds to the group chosen. This scheme is compared numerically with the fourth order commutator free scheme obtained by using only $N^{[0]}$ above and the affine group of degree $d = 1$. The numerical results therein indicate that the commutator free scheme using $d = 2$ has a slight advantage over the scheme with $d = 1$.

For arbitrary nonlinear function, a Taylor expansion in t is needed to be able to apply arbitrary degree affine Lie group integrators. This may or may not be feasible depending on the way $N(y, t)$ is presented. For this reason, it is not believed that arbitrary degree affine Lie group integrators can contribute much to efficient numerical solution of differential equations.

7.4 Discussion

Generalized affine groups of arbitrary degree d have been constructed and the role of higher degree φ functions have been exemplified. It is not believed that Lie group integrators using high degree affine groups can benefit substantially with regard to numerical performance compared to using the first degree group, given a general $N(y, t)$ in (7.4), but it is hoped that this generalization can provide a tool for further analysis on exponential integrators.

The intention of this study was rather to reveal a structure in which the φ functions appear and to possibly be able to use this structure in future studies of φ functions. A natural step forward is to see if backward error analysis of the φ functions is possible analogously to work of N. Higham in [5] on backward error analysis of Padé approximants to the exponential function, in which precisely the group and algebra structure of general linear groups are used. This is the task of the forthcoming paper [4].

Acknowledgements

The author would like to thank Brynjulf Owren, Alexei Rudakov, Bård Skaflestad and Will M. Wright for valuable comments.

7.A Explicit low-degree groups

This appendix lists the definitions and operations specified in Section 7.2 explicitly when the group degree d is fixed.

First degree, $d = 1$

The parameter λ plays no role for $d = 1$ and is thus superflous. It has been kept in these expression in order to match precisely with the general results in Section 7.2.

Group:	$\mathrm{GL}_n(\mathbf{R}) \rtimes \mathbf{R}^n \times \mathbf{R}$
Group identity:	$(I, \mathbf{0}, 0)$
Group product:	$(A_2, b_2, \lambda_2) \cdot (A_1, b_1, \lambda_1) = (A_2 A_1, A_2 b_1 + b_2, \lambda_1 + \lambda_2)$
Group inverse:	$(A, b, \lambda)^{-1} = (A^{-1}, -A^{-1} b, -\lambda)$
Algebra:	$\mathrm{M}_n \times \mathbf{R}^n \times \mathbf{R}$
Algebra addition:	$(\alpha_1, \beta_1, \lambda_1) + (\alpha_2, \beta_2, \lambda_2) = (\alpha_1 + \alpha_2, \beta_1 + \beta_2, \lambda_1 + \lambda_2)$
Algebra bracket:	$[(\alpha_1, \beta_1, \lambda_1), (\alpha_2, \beta_2, \lambda_2)] = ([\alpha_1, \alpha_2], \alpha_1\beta_2 - \alpha_2\beta_1, 0)$
Exponential map:	$\mathrm{Exp}(\alpha, \beta, \lambda) = (\mathrm{e}^\alpha, \phi_1(\alpha)\beta, \lambda)$
Logarithm:	$\mathrm{Log}(A, b, \lambda) = \left(\log A, \frac{\log A}{A-I} b, \lambda\right)$
Group action:	$(A, b, \lambda) \cdot (y, t) = (Ay + b, t + \lambda)$

Second degree, $d = 2$

Group:	$\mathrm{GL}_n(\mathbf{R}) \rtimes (\mathbf{R}^n \times \mathbf{R}^n) \times \mathbf{R}$
Group identity:	$(I, \mathbf{0}, \mathbf{0}, 0)$
Group product:	$(A_2, b_2^{[0]}, b_2^{[1]}, \lambda_2) \cdot (A_1, b_1^{[0]}, b_1^{[1]}, \lambda_1)$ $= \left(A_2 A_1, A_2 b_1^{[0]} + b_2^{[0]}, A_2 b_1^{[1]} + b_2^{[1]}, \lambda_1 + \lambda_2\right)$
Group inverse:	$(A, b^{[0]}, b^{[1]})^{-1} = \big(A^{-1}, -A^{-1}(b^{[0]} - b^{[1]}\lambda),$ $-A^{-1} b^{[1]}, -\lambda\big)$
Algebra:	$\mathrm{M}_n \times \mathbf{R}^n \times \mathbf{R}^n \times \mathbf{R}$
Algebra addition:	$(\alpha_1, \beta_1^{[0]}, \beta_1^{[1]}, \lambda_1) + (\alpha_2, \beta_2^{[0]}, \beta_2^{[1]}, \lambda_2)$ $= \left(\alpha_1 + \alpha_2, \beta_1^{[0]} + \beta_2^{[0]}, \beta_1^{[1]} + \beta_2^{[1]} \lambda_1 + \lambda_2\right)$
Algebra bracket:	$[(\alpha_1, \beta_1^{[0]}, \beta_1^{[1]}, \lambda_1), (\alpha_2, \beta_2^{[0]}, \beta_2^{[1]}, \lambda_2)]$ $= ([\alpha_1, \alpha_2], \alpha_1\beta_2^{[0]} - \alpha_2\beta_1^{[0]} + \beta_1^{[1]}\lambda_2 - \beta_2^{[1]}\lambda_1,$ $\alpha_1\beta_2^{[1]} - \alpha_2\beta_1^{[1]}, 0)$
Exponential map:	$\mathrm{Exp}(\alpha, \beta^{[0]}, \beta^{[1]}, \lambda)$ $= (\mathrm{e}^\alpha, \phi_1(\alpha)\beta^{[0]} + \lambda\phi_2(\alpha)\beta^{[1]}, \phi_1(\alpha)\beta^{[1]}, \lambda)$
Logarithm:	$\mathrm{Log}(A, b^{[0]}, b^{[1]}, \lambda)$ $= (\log(A), \frac{(A\log A - \log A)b^{[0]} + (\log A - A + 1)b^{[1]}\lambda)}{(A-I)^2},$ $\frac{\log A}{A-I} b^{[1]}, \lambda)$ $= (\log A, \varphi_1^{-1}(\log A)b^{[0]} -$ $\varphi_1^{-2}(\log A)\varphi_2(\log A)b^{[1]}\lambda, \varphi_1^{-1}(\log A)b^{[1]}, \lambda)$
Group action:	$(A, b^{[0]}, b^{[1]}, \lambda) \cdot (y, t) = (Ay + b^{[0]} + tb^{[1]}, t + \lambda)$

Bibliography

[1] Berland, H.: Isotropy in geometric integration. Master's thesis, Norwegian University of Science and Technology (2002)

[2] Berland, H., Owren, B., Skaflestad, B.: B-series and order conditions for exponential integrators. SIAM J. Numer. Anal. **43**(4), 1715–1727 (2005)

[3] Berland, H., Skaflestad, B., Wright, W.M.: Expint — A Matlab package for exponential integrators. ACM Trans. on Math. Soft. (2006). To appear

[4] Berland, H., Skaflestad, B., Wright, W.M.: Scaling and squaring of φ functions in exponential integrators (2006). In preparation

[5] Higham, N.J.: The scaling and squaring method for the matrix exponential revisited. SIAM J. Matrix Anal. Appl. **26**(4), 1179–1193 (electronic) (2005)

[6] Koikari, S.: An error analysis of the modified scaling and squaring method (2005). Submitted to Computers Math. Applic.

[7] Krogstad, S.: RKMK-related methods for stiff PDEs. Tech. rep., University of Bergen (2003)

[8] Lawson, J.D.: Generalized Runge–Kutta processes for stable systems with large Lipschitz constants. SIAM J. Numer. Anal. **4**, 372–380 (1967)

[9] Lewis, D., Olver, P.J.: Geometric integration algorithms on homogeneous manifolds. Found. Comput. Math. **2**(4), 363–392 (2002)

[10] Minchev, B.V.: Lie group integrators with nonautonomous frozen vector fields. Tech. rep., Department of Computer Science, University of Bergen (2004). Preprint submitted to Elsevier Science

[11] Munthe-Kaas, H.: High order Runge–Kutta methods on manifolds. In: Proceedings of the NSF/CBMS Regional Conference on Numerical Analysis of Hamiltonian Differential Equations (Golden, CO, 1997), vol. 29,1, pp. 115–127 (1999)