UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Splitting methods for autonomous and non-autonomous perturbed equations

PhD Thesis

Author:

Muaz Seydaoğlu

Advisors:

Prof. Dr. Sergio Blanes Zamora
Dr. Philipp Bader

Valencia, September 2016

Summary

This thesis addresses the treatment of perturbed problems with splitting methods. After motivating these problems in Chapter 1, we give a thorough introduction in Chapter 2, which includes the objectives, several basic techniques and already existing methods.

In Chapter 3, we consider the numerical integration of non-autonomous separable parabolic equations using high order splitting methods with complex coefficients (methods with real coefficients of order greater than two necessarily have negative coefficients). We propose to consider a class of methods that allows us to evaluate all time dependent operators at real values of the time, leading to schemes which are stable and simple to implement. If the system can be considered as the perturbation of an exactly solvable problem and the flow of the dominant part is advanced using real coefficients, it is possible to build highly efficient methods for these problems. We show the performance of this class of methods for several numerical examples and present some new improved schemes.

In Chapter 4, we propose splitting methods for the computation of the exponential of perturbed matrices which can be written as the sum $A = D + \varepsilon B$ of a sparse and efficiently exponentiable matrix D with sparse exponential e^D and a dense matrix εB which is of small norm in comparison with D. The predominant algorithm is based on scaling the large matrix A by a small number 2^{-s} , which is then exponentiated by efficient Padé or Taylor methods and finally squared in order to obtain an approximation for the full exponential. In this setting, the main portion of the computational cost arises from dense-matrix multiplications and we present a modified squaring which takes advantage of the smallness of the perturbation matrix B in order to reduce the number of squarings necessary. Theoretical results on local error and error propagation for splitting methods are complemented with numerical experiments and show a clear improvement over existing methods when medium precision is sought.

In Chapter 5, we consider the numerical integration of the perturbed Hill's equation. Parametric resonances can appear and this property is of great interest in many different physical applications. Usually, the Hill's equations originate from a Hamiltonian function and the fundamental matrix solution is a symplectic matrix. This is a very important property to be preserved by the numerical integrators. In this chapter we present new sixth-and eighth-order symplectic exponential integrators that are tailored to the Hill's equation. The methods are based on an efficient symplectic approximation to the exponential of high dimensional coupled autonomous harmonic oscillators and yield accurate results for oscillatory problems at a low computational cost. Several numerical examples illustrate the performance of the new methods.

Conclusions and pointers to further research are detailed in Chapter 6.

Resumen

Esta tesis aborda el tratamiento de problemas perturbados con métodos de escisión (splitting). Tras motivar el origen de este tipo de problemas en el capítulo 1, introducimos los objetivos, varias técnicas básicas y métodos existentes en capítulo 2.

En el capítulo 3 consideramos la integración numérica de ecuaciones no autónomas separables y parabólicas usando métodos de splitting de orden mayor que dos usando coeficientes complejos (métodos con coeficientes reales de orden mayor de dos necesariamente tienen coeficientes negativos). Proponemos una clase de métodos que permite evaluar todos los operadores con dependencia temporal en valores reales del tiempo lo cual genera esquemas estables y fáciles de implementar. Si el sistema se puede considerar como una perturbación de un problema resoluble de forma exacta y si el flujo de la parte dominante se avanza usando coeficientes reales, es posible construir métodos altamente eficientes para este tipode problemas. Demostramos la eficiencia de estos métodos en varios ejemplos numéricos.

En el capítulo 4 proponemos métodos de splitting para el cálculo de la exponencial de matrices perturbadas que se pueden escribir como suma $A = D + \varepsilon B$ de una matriz dispersa y eficientemente exponenciable con exponencial dispersa e^D y una matriz densa εB de norma pequeña. El algoritmo predominante se basa en escalar la matriz grande con un número pequeño 2^{-s} para poder exponenciar el resultado con métodos eficientes de Padé o Taylor y finalmente obtener la aproximación a la exponencial elevando al cuadrado repetidamente. En este contexto, el coste computacional proviene de las multiplicaciones de matrices densas y presentamos una cuadratura modificada aprovechando la estructura perturbada para reducir el número de productos. Resultados teóricos sobre errores locales y propagación de error para métodos de splitting son complementados con experimentos numéricos y muestran una clara mejora sobre métodos existentes a precisión media.

En el capítulo 5, consideramos la integración numérica de la ecuación de Hill perturbada. Resonancias paramétricas pueden aparecer y esta propiedad es de gran interés en muchas aplicaciones físicas. Habitualmente, las ecuaciones de Hill provienen de una función hamiltoniana y la solución fundamental es una matriz simpléctica, una propiedad muy importante que preservar con los integradores numéricos. Presentamos nuevos integradores simplécticos exponenciales de orden seis y ocho adaptados a la ecuación de Hills. Estos métodos se basan en una aproximación simpléctica eficiente a la exponencial de osciladores armónicos acoplados de dimensión alta y dan lugar a resultados precisos para problemas oscilatorios a un coste computacional bajo y varios ejemplos numéricos ilustran su rendimiento.

Conclusiones e indicadores para futuros estudios se detallan en el capítulo 6.

Resum

La present tesi està enfocada al tractament de problemes perturbats utilitzant, entre altres, mètodes d'escisió (splitting). Comencem motivant l'oritge d'aquest tipus de problems al capítol 1, i a continuació introduïm els objectius, diferents tècniques bàsiques i alguns mètodes existents al capítol 2.

Al capítol 3, considerem la integració numèrica d'equacions no autònomes separables i parabòliques utilitzant mètodes d'splitting d'ordre major que dos utilitzant coeficients complexos (mètodes amb coeficients reials d'ordre major que dos necesariament tenen coeficients negatius). Proposem una clase de mètodes que permeten evaluar tots els operadors amb dependència temporal explícita amb valors reials del temps. Esta forma de procedir genera esquemes estables i fàcils d'implementar. Si el sistema es pot considerar com una perturbació d'un problema exactament resoluble, i la part dominant s'avança utilitzant coeficients reials, es posible construir mètodes altament eficients per aquest tipus de problemes. Demostrem la eficiència d'estos mètodes per a diferents exemples numèrics.

Al capítol 4, proposem mètodes d'splitting per al càcul de la exponencial de matrius pertorbades que es poden escriure com suma $A = D + \varepsilon B$ (una matriu que es pot exponenciar fàcilment i eficientement, com es el cas d'algunes matrius disperses e^D, i una matriu densa εB de norma menuda). L'algorisme predominant es basa en escalar la matriu gran amb un nombre menut 2^{-s} per a poder exponenciar el resultat amb mètodes eficients de Padé o Taylor i finalment obtindre la aproximació a la exponencial elevant al quadrat repetidament. En este context, el cost computacional prové de les multiplicacions de matrius denses i presentem una quadratura modificada aprofitant la estructura de matriu pertorbada per reduir el nombre de productes. Resultats teòrics sobre errors locals i propagació d'error per a mètodes d'splitting son analitzats i corroborats amb experiments numèrics, mostrant una clara millora respecte a mètodes existens quan es busca una precisió moderada.

Al capítol 5, considerem la integració numèrica de l'ecuació de Hill pertorbada. En este tipus d'equacions poden apareixer resonàncies paramètriques i esta propietat es de gran interés en moltes aplicacions físiques. Habitualment, les equacions de Hill provenen d'una función hamiltoniana i la solució fonamental es una matriu simplèctica, siguent esta una propietat molt important a preservar pels integradors numèrics. Presentem nous integradors simplèctics exponencials d'orden sis i huit construits especialmente per resoldre l'ecuació de Hill. Estos mètodes es basen en una aproxmiació simplèctica eficient a la exponencial d'osciladors harmònics acoplats de dimensió alta i donen lloc a resultats precisos per a problemas oscilatoris a un cost computacional baix. La eficiencia dels mètodes se il.lustra en diferents exemples numèrics.

Conclusions i indicadors per a futurs estudis es detallen al capítol 6.

CONTENTS

Chapter 1

MOTIVATION

In this thesis, we mainly consider the numerical integration of perturbed systems

$$\frac{d}{dt}u(t) = A(t,u) + \varepsilon B(t,u), \qquad u(0) = u_0 \in \mathbb{R}^d, \tag{1.0.1}$$

involving a small parameter ε, where both equations

$$\frac{d}{dt}u(t) = A(t,u), \qquad \frac{d}{dt}u(t) = \varepsilon B(t,u)$$

are exactly solvable (or easy to numerically solve). To take profit from the fact that ε is a small parameter and the equation for the dominant part can be efficiently solved, we consider splitting methods for near-integrable systems because they are tailored to this class of equations and the existing mehtods have shown a great performance in many relevant perturbed problems.

Perturbed equations may arise in many fields such as molecular dynamics, quantum mechanics, weakly-coupled systems, dynamical astronomy and celestial mechanics. For the numerical integration of the equations arising in such systems one can consider using symplectic integrators that preserve qualitative properties of exact solution [88]. Many differential equations that arise in dynamical astronomy are Hamiltonian systems and the solution of a Hamiltonian system is a symplectic transformation. The numerical solutions obtained by symplectic integrators have a property of area preservation. The solutions obtained by classical integrators are not symplectic transformations and the area can not be preserved, thus solutions are damped or excited [59]. Yoshida et al. [59] have presented symplectic integrators which are well suited for the long time integration of the Hamiltonian system. They also considered symplectic integrators for near-integrable Hamiltonian systems and the dynamical system of the solar system without dissipative effects which belongs to this class [59]. These integrators conserve the angular momentum vector of the N-body problem and the solution obtained by such integrators has the time reversibility property and so on. Symplectic integrators are widely used for the integrable Hamiltonian H_0 perturbed by a term H_1 of the near-integrable

form

$$H(p,q) = H_0(p,q) + \varepsilon H_1(p,q) \tag{1.0.2}$$

where $(p,q) \in \mathbb{R}^{2d}$, ε is small parameter and if $\varepsilon = 0$ then the system (1.0.2) is integrable. For example, the Hamiltonian of the N-body problem in celestial mechanics can be written of the form (1.0.2) by using Jacobian or heliocentric coordinates, where H_0 represents the Keplerian motion of the planets and εH_1 represents perturbations from planets interaction [104]. Furthermore, in the context of dynamical astronomy the second fundamental model of resonance with perturbation [50] is another example of the form (1.0.2). Saha and Tremaine [87] have proposed symplectic integrators to solve N-body problems in solar system dynamics which can originate from near-integrable Hamiltonian of the form (1.0.2) with $\varepsilon \ll 1$. They show theoretically and verify numerically that the linearly growing error in the phase space which originates from the error in frequency can be eliminated by a suitable starting procedure. Mclachlan [69] proposed numerical methods to solve the perturbed problem (1.0.1). These methods are based on compositions of certain elementary flows of the ordinary differential equations. If one uses these composition methods, then one can preserve the geometric properties of the true flows such as symplecticity for Hamiltonian vector fields , volume preservation for divergence free vector fields and so on [69]. Wisdom et al. [105] have presented symplectic methods in which the spurious oscillations in energy and state variables originate from the delta function formulation of symplectic integration can be removed by the canonical transformations. These methods can be used for the integration of perturbed dynamical systems, and they examined the performance of the methods on the N-body problem [105]. Mikkola [72] presented symplectic integrators for the investigation of the long time behavior of hierarchical few-body systems. These methods improved the efficiency of integrators constructed in [105] for few-body problems by making use of time transformations and extended phase space. Liao [64] has given the mixed implicit symplectic integrators for nearly-integrable Hamiltonian system of the form (1.0.2) and these integrators are time-reversible. Chambers and Murison [39] proposed new high-order symplectic integrators with fewer substeps per step for near-integrable Hamiltonian systems and they called them pseudo-order integrators. Blanes et al. [20] have used processing techniques to derive new high-order symplectic integrators for perturbed Hamiltonian systems and more recently, Blanes et al. [15] proposed new symplectic splitting methods designed to solve such systems with applications to the planetary N-body problems. They presented a systematic analysis, from which one can obtain a set of independent order conditions up to any desired order of accuracy by using Lyndon multi-indices. Farrés et al. [44] described different splitting symplectic methods and compared them to find the best splitting scheme for long term integration in the solar system.

The linear and nonlinear Schrödinger equations with perturbed harmonic oscillator potential are important perturbed problems in quantum mechanics. Recently, Bader et al. [3, 4] have proposed new high order splitting methods for a special class of Gross-Pitaveskii equations with the potential trap $V(x,t) = x^T M(t)x + \varepsilon V_\varepsilon(x,t)$ where $M(t) \in \mathbb{R}^{d\times d}$ is a positive-definite matrix and $\varepsilon V_\varepsilon(x,t)$ is a small perturbation. These splitting methods involve some negative coefficients and have been used with Fourier techniques to solve each sub-problem obtained by splitting the system. The authors in [5] have presented splitting methods to solve Schrödinger eigenvalue problem in which the Hamiltonian is considered as a perturbed system, i.e, $H = H_0 + \varepsilon V_\varepsilon(x)$ where the exactly solvable part is $H_0 = T + V_0(x)$. These high-order methods comprise complex coefficients with positive real part to overcome the positivity

requirement on high-order splitting methods for the integration of such classes of diffusive problems.

Motivated by all these results and the high performance of the splitting methods for perturbed systems, in this thesis we consider the following classes of perturbed problems that were not previously considered in the literature and the existing splitting methods either can not be directly applied or they show not a good performance:

- The numerical integration of non-reversible problems by splitting methods require forward time integration or, equivalently, methods with positive coefficients. Unfortunately, splitting methods of order greater than two with real coefficients necessarily have negative coefficients. One can avoid this order barrier by considering complex coefficients. However, if the problem is explicitly time-dependent, complex coefficients usually require to evaluate the time-dependent operators at complex times, and this can cause severe troubles in the implementation of the methods. For this reason, we propose to consider a new class of methods that involve to compute the evolution of the dominant part

  $$\frac{d}{dt}u(t) = A(t,u),$$

  along fractional real time intervals (in general, for explicitly time-dependent problems, one has to solve this problems numerically using an efficient integrator), followed by the integration of the autonomous problem

  $$\frac{d}{dt}u(t) = \varepsilon B(\tau,u)$$

  where τ corresponds to the time frozen at an appropriate instant. We study how to obtain the order conditions to be satisfied by the coefficients of the methods, we analyze which conditions carry the dominant contributions to the error, and we build a set of new methods that are tested on different PDEs, where the high performance of the new methods is manifest [26, 90].

- Splitting methods are based on the exact solution of some differential equations. When these equations are linear, this requires the computation of exponential of matrices or their action on vectors. There exist many algorithms to compute numerically the exponential of a matrix. Some of the most efficient algorithms are based on the splitting-squaring procedure jointly with a Padé or Taylor approximations. However, in some applications one has to compute

  $$e^{D+\varepsilon B}$$

  where D is a diagonal or sparse matrix such that e^D can be easily and efficiently computed but B is a dense matrix. We analyze a new class of splitting schemes tailored to this problem that shows a clearly superior performance (accuracy versus computational cost) for medium accuracies to either Padé or Taylor approximations as used by most commercial algorithms [8].

- Finally, we consider the numerical integration of second order differential equations of the form

$$x'' = f(t, x)$$

when $f(t, x)$ is a quasilinear vector field, i.e. it can be written as

$$x'' = M(t)x + \varepsilon g(t, x). \tag{1.0.3}$$

To use splitting methods for perturbed problems requires to use an efficient integrator to solve the linear problem

$$x'' = M(t)x.$$

When $M(t)$ is a periodic function, this problem corresponds to the Hill's equation and the system can be stable or unstable depending on some parameters in the matrix M (parametric resonance). The parametric resonances allows to have some control on many systems in order to keep it stable or unstable, and has many physical applications. In addition, if M is a symmetric matrix, the system evolves through a symplectic transformation. For this reason, we first study the efficient integration of this problem using tailored symplectic integrators. We propose several new methods whose performance is tested on several numerical examples [7]. The new methods can be used as the integrators of the dominant part for the perturbed problem (1.0.3).

Chapter 2

INTRODUCTION

Splitting methods appear when a vector field can be split into several pieces and each of them is simpler to integrate than the original. Important applications of this class of integrators are in geometric integration, that is, the integration of vector fields that preserves structural properties of the system. Some of the geometric properties are being Hamiltonian, or divergence-free, or providing time-reversible symmetry, or first integrals. They are exploited independently in different applied fields. Thus we have dimensional splitting for parabolic PDEs, fractional-step and operator splitting methods for the Navier Stokes equations and reaction diffusion systems, split-step methods in optics and acoustics, split-Hamiltonian methods in chemical physics, the mapping method in celestial mechanics, and Lie-Trotter-Kato formula in quantum statistical mechanics [12, 16, 70].

2.1 Splitting methods

The main idea of splitting is the process of breaking down a vector field into its basic elements such that each element is integrable and easy to handle separately [47]. The framework of splitting methods for the time integration of ordinary differential equations (ODEs) can be exhibited as follows. Let us consider the initial value problem

$$x' = \frac{dx}{dt} = F(x), \qquad x(0) = x_0 \in \mathbb{R}^d, \tag{2.1.1}$$

with $F : \mathbb{R}^d \to \mathbb{R}^d$ and exact solution $x(t) = \varphi_t(x_0)$. Consider the function F can be written as $F = \sum_{i=1}^{n} F^{[i]}$ with functions $F^{[i]} : \mathbb{R}^d \to \mathbb{R}^d$, then each equation

$$x' = F^{[i]}(x), \qquad x(0) = x_0 \in \mathbb{R}^d, \qquad i = 1, \dots, n,$$

can be integrated at time step $t = h$ with solutions $\varphi_h^{[i]}(x_0)$. If one takes the composition of these flows as follows,

$$\psi_h = \varphi_h^{[n]} \circ \dots \circ \varphi_h^{[2]} \circ \varphi_h^{[1]}, \tag{2.1.2}$$

then one obtains a first order approximation to the exact solution, that is,

$$\psi_h(x_0) = \varphi_h(x_0) + O(h^2),$$

and this can be easily seen by Taylor expanding ψ [16]. On the other hand, one might equivalently consider F as a vector field on phase space $\mathbb{R}^d$ which can be expressed as a sum of vector fields $F^{[i]}$ as $F = \sum_{i=1}^{n} F^{[i]}$. In this case, we can write the flow of the ODE (2.1.1) as $x(t) = \exp(tF)(x(0))$ and combining each of these flows $\exp(tF^{[i]}$, we obtain the first order accurate integrator below,

$$\psi_h = \exp(hF^{[1]}) \exp(hF^{[2]}) \dots \exp(hF^{[n]}) \qquad i = 1, \dots, n. \tag{2.1.3}$$

So, for time step h

$$\psi_h = \exp(h \sum_{i=1}^{n} F^{[i]}) + \mathcal{O}(()\, h^2).$$

Recognize that when comparing equation (2.1.3) with (2.1.2), the order is reversed. In subsection 2.2.2 we will see this fact stems from the Lie derivative associated to the ODE system (2.1.1) and this phenomenon referred to the *Vertauschungssatz* (see Section 2.2.2. In all this process one important assumption is that the equation $x' = F^{[i]}(x)$ should be easier to integrate or treat numerically [16, 70]. Motivated by these facts, splitting methods involve three steps [16, 70]:

(1) find vector fields $F^{[i]}$ such that F can be written as $F = \sum F^{[i]}$

(2) solve each equation $x' = F^{[i]}(x)$ or equivalently integrate each vector field $F^{[i]}$ either exactly or approximately

(3) combine these solutions to get an approximation for (2.1.1) or equivalently an integrator for F

Splitting methods yield some advantages in the sense of computational mathematics such as speed, accuracy and stability. They are in general explicit and the implementation of methods is easier. Furthermore, they form an important class of geometric numerical integrators due to preserving structural properties of the exact solution. Examples of these properties are conservation of first integrals, energy or angular momentum, time-symmetry, reversibility, volume preservation [16, 55, 70]. Naturally, numerical integration of the ODE (2.1.1) is based upon well-known techniques of discretizing the equation in a way to keep the global error, namely the Euclidean norm of the difference between the numerical solution x_n and the exact solution $x(t_n)$ at time step $t_n = nh$ as $\epsilon > 0$ below a desired tolerance $\|x_n - x(t_n\| < \epsilon$, as small as possible. This process can be done by choosing any standard methods (e.g one-step, multistep, extrapolation), the order (fixed or adaptive) and the time step (constant or variable), in the latter case, the phase portrait of the system will be defective. In contrast, one can choose geometric numerical integrators and evaluate solutions for very long times for several initial conditions with fixed time step in order to get an approximate phase portrait of the system. Its phase portrait will be similar to the original system [16]. Geometric numerical integrators are designed as numerical methods which have the same qualitative properties of the system, namely recreate the qualitative properties of the discretised solution of the differential

equation [47, 71]. There are several algorithms based upon preserving structural properties in many different areas of research such as molecular dynamics, quantum (statistical) mechanics, astronomy, accelerator physics [16, 47, 55, 63, 70].

Symplectic Euler and The Störmer-Verlet method:
Let us consider $H(q,p)$ as a function defined in the phase space Γ which is a domain in the oriented Euclidean space $\mathbb{R}^{2d}$ of the points $(q,p) = (q_1,\dots,q_d,p_1,\dots,p_d)$. Suppose a Hamiltonian system has the form $H(q,p) = T(p) + V(q)$ where $q \in \mathbb{R}^d$ are generalized coordinates, $p \in \mathbb{R}^d$ are the conjugated generalized momenta and H is the total mechanical energy as sum of kinetic energy T and potential energy V. Then the corresponding equations of motion are

$$\frac{d}{dt}q = +\nabla_p T(p), \qquad \frac{d}{dt}p = -\nabla_q V(q), \qquad q(0) = q_0, \quad p(0) = p_0, \tag{2.1.4}$$

where $\nabla_q = (\frac{\partial}{\partial q_1},\dots,\frac{\partial}{\partial q_d})$, $\nabla_p = (\frac{\partial}{\partial p_1},\dots,\frac{\partial}{\partial p_d})$. One can combine all the dependent variables in (2.1.4) in a $2d$-dimensional vector $x = (q,p)$. Then equation (2.1.4) can be written in the form (2.1.1)

$$\frac{dx}{dt} = J\nabla H$$

Here ∇ is gradient operator $(\frac{\partial}{\partial q_1},\dots,\frac{\partial}{\partial q_d},\frac{\partial}{\partial p_1},\dots,\frac{\partial}{\partial p_d})$ and J corresponds to the $2d \times 2d$ canonical symplectic matrix

$$J = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}$$

where I and 0 are $d \times d$, identity and zero matrices respectively. The exact flow of this Hamiltonian system is symplectic [2, 16, 63, 88]. Let us approximate the solution of the system (2.1.4) with Euler's method. Then, we obtain the first order approximation as below

$$\begin{aligned} q_{n+1} &= q_n + h\nabla_p T(p_n), \\ p_{n+1} &= p_n - h\nabla_q V(q_n), \end{aligned}$$

where $n = 0,1,\dots$ and $h > 0$ is the time step. H is the sum of two solvable Hamiltonian equations since the kinetic energy depends only on momenta and the potential energy only on coordinates, the split Hamiltonian system corresponds to the equations:

$$\begin{cases} \frac{d}{dt}q = +\nabla_p T(p) \\ \frac{d}{dt}p = 0 \end{cases} \quad \text{and} \quad \begin{cases} \frac{d}{dt}q = 0 \\ \frac{d}{dt}p = -\nabla_q V(q). \end{cases}$$

The flows of the above equations with initial condition (q_0,p_0) are

$$\varphi_t^{[T]} : \begin{cases} q(t) = q_0 + t\nabla_p T(p_0) \\ p(t) = p_0 \end{cases} \quad \text{and} \quad \varphi_t^{[V]} : \begin{cases} q(t) = q_0 \\ p(t) = p_0 - t\nabla_q V(q), \end{cases}$$

respectively. If we choose the step size h and grid points $t_n = t_0 + nh$ $(n = 0,1,\dots)$, then the composition of the scheme $\varphi_h^{[V]}$ with $\varphi_h^{[T]}$ as in (2.1.3), will give the method

$$\psi_h \equiv \varphi_h^{[T]} \circ \varphi_h^{[V]} : \begin{cases} q_{n+1} = q_n + h\nabla_p T(p_{n+1}) \\ p_{n+1} = p_n - h\nabla_q V(q_n). \end{cases}$$

This first order accurate method is a composition of two simple symplectic flows of the Hamiltonian systems. It is well known that the composition of two symplectic maps is also symplectic. Thus, ψ_h is a symplectic integrator and called symplectic Euler method [47]. Let us consider the definition of an adjoint of a given integrator ψ_h. The method ψ_h^* defined by

$$\psi_h^* = [\psi_{-h}]^{-1},$$

is called the adjoint method of ψ_h. From the above definition it is clear that adjoint of the adjoint method is the original method. A method ψ_h is called self-adjoint or (time) symmetric if it is equal to its own adjoint [16, 63], namely

$$\psi_h = \psi_h^* \qquad or \qquad \psi_{-h} \circ \psi_h = id. \tag{2.1.5}$$

One can consider composition of the flows in the reverse order as $\varphi_h^{[V]} \circ \varphi_h^{[T]}$ with the adjoint of ψ_h, that is,

$$\psi_h^* \equiv \varphi_h^{[V]} \circ \varphi_h^{[T]} : \begin{cases} q_{n+1} = q_n + h\nabla_p T(p_n) \\ p_{n+1} = p_n - h\nabla_q V(q_{n+1}). \end{cases}$$

We can go a step further and consider a symmetric version of the composition, a time step $h/2$ of ψ with its adjoint as $\Theta_h^{[2]} = \psi_{h/2} \circ \psi_{h/2}^*$, that is,

$$\Theta_h^{[2]} \equiv \varphi_{h/2}^{[V]} \circ \varphi_h^{[T]} \circ \varphi_{h/2}^{[V]} : \begin{cases} p_{n+1/2} &= p_n - \frac{h}{2}\nabla_q V(q_n) \\ q_{n+1} &= q_n + h\nabla_p T(p_{n+1/2}) \\ p_{n+1} &= p_{n+1/2} - \frac{h}{2}\nabla_q V(q_{n+1}). \end{cases} \tag{2.1.6}$$

One can readily show that the second order accurate method $\Theta_h^{[2]}$ is time-symmetric as follows

$$\begin{aligned} \Theta_{-h}^{[2]} &= \psi_{-h/2} \circ \psi_{-h/2}^* \\ &= [\psi_{h/2}^*]^{-1} \circ [\psi_{h/2}]^{-1} \\ &= [\psi_{h/2}^* \circ \psi_{h/2}]^{-1} \\ &= [\Theta_h^{[2]}]^{-1}. \end{aligned}$$

This composition is often referred to as Strang splitting [94]. Different names of this method can also be found in the literature, such as the Störmer-Verlet method [101] or leapfrog [16, 47, 70].

2.2 Composition

2.2.1 Construction of higher order integrators by composition

The construction of higher order integrators is simple using compositions of a basic integrator of low order. We have seen in the previous section that a second order integrator Störmer-Verlet method (2.1.6) can be obtained by a symmetric composition of a first order integrator and its adjoint. Then, a fourth order integrator $\Theta_h^{[4]} : \mathbb{R}^{2d} \to \mathbb{R}^{2d}$ is obtained by a symmetric composition of the second order integrator $\Theta_h^{[2]} : \mathbb{R}^{2d} \to \mathbb{R}^{2d}$ as [95, 106]

$$\Theta_h^{[4]} \equiv \Theta_{\alpha h}^{[2]} \circ \Theta_{\beta h}^{[2]} \circ \Theta_{\alpha h}^{[2]}.$$

where

$$2\alpha + \beta = 1,$$
$$2\alpha^3 + \beta^3 = 0.$$

If we go further, one might recursively define $\Theta_h^{[2k+2]} : \mathbb{R}^{2d} \to \mathbb{R}^{2d}$ for $k = 1, 2, \dots$ in the form

$$\Theta_h^{[2k+2]} \equiv \Theta_{\alpha h}^{[2k]} \circ \Theta_{\beta h}^{[2k]} \circ \Theta_{\alpha h}^{[2k]}. \tag{2.2.1}$$

where

$$\alpha = \frac{1}{2 - 2^{\frac{1}{2k+1}}}, \qquad \beta = 1 - 2\alpha, \tag{2.2.2}$$

This is called the triple jump or Yoshida's composition [106]. Then, the integrators $\Theta_h^{[2k]}$ are of order $2k$ for $k \geq 1$ [16, 95, 106]. Additionally, $\Theta_h^{[2k]}$ $(k = 1, 2, \dots)$ defined by (2.2.1) and (2.2.2) are time-symmetric integrators of order $(2k)$ [16]. Especially, suppose that $F(x)$ in the initial value problem (2.1.1) can be split in the form

$$F(x) = \sum_{i=1}^{n} F^{[i]}(x).$$

Then, basic first order integrator and its adjoint,

$$\psi_h = \varphi_h^{[n]} \circ \dots \circ \varphi_h^{[2]} \circ \varphi_h^{[1]}$$
$$\psi_h^* = \psi_{-h}^{-1} = \varphi_h^{[1]} \circ \varphi_h^{[2]} \dots \circ \varphi_h^{[n]}.$$

give a way to construct time-symmetric integrators $\Theta_h^{[2k]}$ $(k = 1, 2, \dots)$ of order $2k$. This process is a simple way to construct arbitrarily higher order methods. The iterative scheme (2.2.1) involve a large number of evaluations and usually create huge truncation errors even if they are easy to construct [16].
Recognize that $2k$th order integrators $\Theta_h^{[2k]}$ $(k = 1, 2, \dots)$ can be expressed as follows

$$\Phi_h = \psi_{\alpha_{2s}h} \circ \psi^*_{\alpha_{2s-1}h} \circ \dots \circ \psi_{\alpha_2 h} \circ \psi^*_{\alpha_1 h}, \tag{2.2.3}$$

where $s = 3^{k-1}$ and with constant coefficients $(\alpha_1, \dots \alpha_{2s}) \in \mathbb{R}^{2s}$. Then, arbitrarily higher order methods can be constructed by determined coefficients $(\alpha_1, \dots \alpha_{2s}) \in \mathbb{R}^{2s}$ in the composition integrators of the form (2.2.3). Let us consider ODE (2.1.1) can be split in two pieces as

$$\frac{d}{dt}x = F^{[A]}(x), \qquad \frac{d}{dt}x = F^{[B]}(x), \tag{2.2.4}$$

with corresponding solutions $\varphi_h^{[A]}(x_0)$ and $\varphi_h^{[B]}(x_0)$, respectively. One can rewrite the composition integrator (2.2.3) by taking composition of solutions as $\psi_h = \varphi_h^{[A]} \circ \varphi_h^{[B]}$ and inserting those into the integrator as below

$$\Phi_h = \varphi_{b_{s+1}h}^{[B]} \circ \varphi_{a_s h}^{[A]} \circ \varphi_{b_s h}^{[B]} \circ \dots \circ \varphi_{b_2 h}^{[B]} \circ \varphi_{a_1 h}^{[A]} \circ \varphi_{b_1 h}^{[B]} \tag{2.2.5}$$

where

$$b_1 = \alpha_1, \qquad a_i = \alpha_{2i-1} + \alpha_{2i}, \qquad b_{i+1} = \alpha_{2i} + \alpha_{2i+1}, \qquad i = 1, \dots, s,$$

(with $\alpha_{2s+1} = 0$) [16].

2.2.2 Lie derivative and integrators

Let us consider the relation between integrators and first-order differential operators. One can write the flow of a differential equation formally as the exponential of a Lie derivative. This process provides a way to determine the order conditions of the integration scheme [47]. Let $\varphi_t(x_0)$ be the exact solution of the differential equation (2.1.1). Then, the Lie derivative associated with F is given by

$$L_F = \sum_{i=1}^{n} F_i \frac{\partial}{\partial x_i},$$

and the linear differential operator L_F acts on differentiable functions $g : \mathbb{R}^n \to \mathbb{R}$ as

$$L_F[g](x) = \sum_{i=1}^{n} F_i(x) \frac{\partial g}{\partial x_i}(x),$$

where $x = (x_1, \dots, x_n) \in \mathbb{R}^n$. The Lie derivative L_F has the following properties. Let Z_1, Z_2 be arbitrary vector functions. Then,

$$\begin{aligned} L_F(\alpha Z_1 + \beta Z_2) &= \alpha L_F Z_1 + \beta L_F Z_2, \qquad \alpha, \beta \in \mathbb{R} \\ L_F(Z_1 Z_2) &= (L_F Z_1) Z_2 + Z_1 (L_F Z_2), \\ L_F^k(Z_1 Z_2) &= \sum_{i=1}^{k} \binom{k}{i} (L_F^i Z_1)(L_F^{k-i} Z_2), \end{aligned}$$

where $L_F^i Z = L_F(L_F^{i-1} Z)$ and $L_F^0 Z = Z$. The last property is known as the Leibniz rule and its iteration L_F^k can be easily proved by induction. On the other hand, let F and V be two arbitrary vector fields. For $\alpha, \beta \in \mathbb{R}$, we have

$$\begin{aligned} \alpha L_F + \beta L_V &= L_{\alpha F + \beta V} \\ [L_F, L_V] &= L_F L_V - L_V L_F = L_W, \end{aligned}$$

where W is another vector field obtained by the Lie bracket of the vector fields F and V, denoted by $W = (F, V)$, and its components read

$$W_i = (F, V)_i = L_F V_i - L_V F_i = \sum_{j=1}^{n} F_j \frac{\partial V_i}{\partial x_j} - V_j \frac{\partial F_i}{\partial x_j}.$$

The action of a Lie derivative on a differentiable function $G : \mathbb{R}^n \to \mathbb{R}^m$ can be written in the form

$$W = L_F G(y) = G'(y) F(y), \quad W : \mathbb{R}^n \to \mathbb{R}^m.$$

Here, $G'(y)$ indicates the Jacobian matrix of G [2]. One can write derivative of the G acting on the solution $\varphi_t(x_0)$ of the initial value problem (2.1.1) with applying the chain rule, that is,

$$\frac{d}{dt} G(\varphi_t(x_0)) = (L_F G)(\varphi_t(x_0)).$$

If one repeats differentiations over and over, say l times, then

$$\frac{d^l}{dt^l} G(\varphi_t(x_0)) = (L_F^l G)(\varphi_t(x_0)), \qquad l \geq 1.$$

From the above equation, one gets the Taylor series of $G(\varphi_t(x_0))$ at $t = 0$ as

$$G(\varphi_t(x_0)) = \sum_l \frac{t^l}{l!}(L_F^l G(x_0)) = \exp(tL_F)[G](x_0). \tag{2.2.6}$$

Let G be the identity map, that is, $G(y) = \mathrm{I}(y) = y$. Inserting it into (2.2.6) gives the Taylor series of the solution itself as follows

$$\varphi_t(x_0) = \sum_l \frac{t^l}{l!}(L_F^l \mathrm{I})(x_0) = \exp(tL_F)[\mathrm{I}](x_0).$$

Let $\varphi_s^{[A]}$ and $\varphi_t^{[B]}$ be the corresponding flows of the differential equations $\frac{d}{dt}x = F^{[A]}(x)$, $\frac{d}{dt}x = F^{[B]}(x)$. Then, by composing these flows and using (2.2.6) one gets

$$g(\varphi_t^{[B]} \circ \varphi_s^{[A]})(x_0) = \exp(sL_A)\exp(tL_B)[g](x_0). \tag{2.2.7}$$

The above composition is obtained from formula (2.2.6) by considering $F = A$, replacing t with s and $G(y) = g(\varphi_t^{[B]}(y)) = \exp(tL_B)[g](y)$. Notice that the order of Lie transforms in l.h.s of the (2.2.7) becomes reversed to their corresponding maps and this phenomenon is called "Vertauschungssatz" [18, 47, 79]. For the sake of simplicity and if not explicitly stated, otherwise we will replace hereafter the symbol of the Lie derivative associated with F in ODE (2.1.1) L_F with F.

Let an integrator $\psi_h : \mathbb{R}^d \to \mathbb{R}^d$ be of order r for the ODE (2.1.1) with h-flow ψ_h. Then

$$\psi_h(x) = \varphi_h(x_0) + \mathcal{O}(()\, h^{r+1}) \qquad as \qquad h \to 0.$$

We know that, for any smooth function $g : \mathbb{R}^d \to \mathbb{R}$,

$$g(\varphi_h(x)) = \exp(hF)[g](x) = g(x) + \sum_{n\geq 1} \frac{h^n}{n!} F^n[g](x), \tag{2.2.8}$$

is satisfied from equation (2.2.6). Let us consider any basic integrator $\phi_h : \mathbb{R}^d \to \mathbb{R}^d$. Set [16]

$$X_n[g](x) = \frac{1}{n!}\frac{d^n}{dh^n} g(\phi_h(x))\,|_{h=0} \tag{2.2.9}$$

and let

$$X(h) = I + \sum_{n\geq 1} h^n X_n. \tag{2.2.10}$$

Here I is the identity operator. Then, it formally holds that $g(\phi_h(x)) = X(h)[g](x)$. According to (2.2.8), for the integrator ϕ_h to be of order r, it requires verifying

$$X_n = \frac{1}{n!}F^n, \qquad 1 \leq n \leq r.$$

On the other hand, let us denote the series of vector fields as

$$Y(h) = \sum_{n\geq 1} h^n Y_n = \log(X(h)) = \sum_{m\geq 1} \frac{(-1)^{m+1}}{m}(hX_1 + h^2X_2 + \ldots)^m,$$

that is,

$$Y_n = \sum_{m\geq 1}^{n} \frac{(-1)^{m+1}}{m} \sum_{j_1+\ldots+j_m=n} (X_{j_1} \ldots X_{j_m})^m.$$

Then, $X(h) = \exp(Y(h))$ and formally $g(\phi_h(x)) = \exp(Y(h))[g](x)$[16]. According to (2.2.8), the integrator ϕ_h is of order r when

$$Y_1 = F \qquad Y_n = 0 \qquad \text{for} \qquad 2 \leq n \leq r. \tag{2.2.11}$$

The condition for the time-symmetry of ϕ_h is $\phi_h^*(x) = \exp(-Y(-h))$ and is satisfied if and only if $Y(h) = hY_1 + h^3Y_3 + \cdots$. Thus, that time symmetric methods are of even order[16]. Let us deal with the symmetric integrators $\Theta_h^{[2k]}$ given by (2.2.1) with (2.2.2) and check that they are of order $2k$ on condition that $\Theta_h^{[2]}$ is a symmetric second order integrator. For the symmetric of order $2k$ integrators $\Theta_h^{[2k]}$ one has

$$g(\Theta_h^{[2k]}(x)) = \exp(F^{[2k]}(h))[g](x)$$

where the series of differential operators are in the form

$$F^{[2k]}(h) = hF + h^{2k+1}F_{2k+1}^{[2k]} + h^{2k+3}F_{2k+3}^{[2k]} + \ldots$$

and according to (2.2.1),

$$\exp(F^{[2k+2]}(h)) = \exp(F^{[2k]}(\alpha h)) \exp(F^{[2k]}(\beta h)) \exp(F^{[2k]}(\alpha h)).$$

One thus deduces that

$$F^{[2k+2]}(h) = h(2\alpha + \beta)F + h^{2k+1}(2\alpha^{2k+1} + \beta^{2k+1})F_{2k+1}^{[2k]} + O(h^{2k+3}),$$

and $\Theta_h^{[2k+2]}$ is of order $2k + 2$ on condition that $\Theta_h^{[2k]}$ is of order $2k$ [16]. Let us check the order of general composition of the form (2.2.3). It is possible to write that

$$g(\Phi_h(x)) = \Xi(h)[g](x), \tag{2.2.12}$$

where $\Xi(h)$ is a series of differential operator in the form

$$\Xi(h)[g](x) = I + \sum_{n\geq 1} h^n \Xi_n.$$

Then, it verifies that

$$\Xi(h) = X(-\alpha_1 h)^{-1} X(\alpha_2 h) \ldots X(-\alpha_{2s-1} h)^{-1} X(\alpha_{2s} h) \tag{2.2.13}$$

where $X(h)$ is given by (2.2.9)-(2.2.10) with

$$X(h)^{-1} = I + \sum_{m\geq 1} (-1)^m (hX_1 + h^2 X_2 + \ldots)^m.$$

Comparing (2.2.12) with (2.2.8), the order of the integrator (2.2.3) exceeds r if

$$\Xi_n = \frac{1}{n!} F^n \qquad 1 \leq n \leq r.$$

To get the terms Ξ_n of the series $\Xi(h)$, it is convenient to take the formal equality

$$\Xi(h) = \exp(-Y(-\alpha_1 h)) \exp(Y(\alpha_2 h)) \dots \exp(-Y(-\alpha_{2s-1} h)) \exp(Y(\alpha_{2s} h)) \tag{2.2.14}$$

in place of (2.2.13) to get the series expansion of $\log(\Xi(h)) = \sum_{n\geq 1} h^n F_n$. Then, clearly general compositions methods of order r require to verify the following conditions

$$F_1 = F, \qquad F_n = 0 \qquad \text{for} \qquad 2 \leq n \leq r.$$

It is perfectly possible to write a formal expression of the series $\Xi(h)$ of differential operators associated to the integrator Φ_h in (2.2.5) as below

$$\Xi(h) = e^{b_1 h L_B} e^{a_1 h L_A} \dots e^{b_s h L_B} e^{a_s h L_A} e^{b_{s+1} h L_B}, \tag{2.2.15}$$

where L_A and L_B are Lie derivatives corresponding to $F^{[A]}$ and $F^{[B]}$ in (2.2.4) respectively, that is,

$$L_A[g](x) = \sum_{j=1}^{d} F_j^{[A]}(x) \frac{\partial g}{\partial x_j}(x), \qquad L_B[g](x) = \sum_{j=1}^{d} F_j^{[B]}(x) \frac{\partial g}{\partial x_j}(x). \tag{2.2.16}$$

2.3 Order conditions via BCH formula

In this chapter, we present order conditions of the splitting and composition methods. There are different ways to obtain these order conditions for prescribed order of accuracy in the literature. They usually yields polynomial equations in the coefficients. The best known process to have these polynomial equations can be found in [106] and summarized briefly in [16, 47], that is, applying the Baker-Campbell-Hausdorff (BCH) formula in a recursive manner to the $\log(\Xi(h)) = \sum_{n\geq 1} h^n L_n$. Another alternative technique can be found in [78] based upon the theory of rooted trees similar to the idea of B-series used in the analysis of Runge-Kutta methods (see also [47]). We show two procedures presented in [16]. The first procedure uses the BCH formula recursively and the second procedure replaces rooted trees presented in [78] with Lyndon words in [40] to get polynomial equations in the coefficients. Let $\pounds(X, Y)$ be a Lie algebra generated by two non-commuting operators with the commutator $[X, Y] = XY - YX$ as Lie bracket [100]. The goal of the BCH formula is to express $C \in \pounds(X, Y)$ which satisfies

$$\exp(X) \exp(Y) = \exp(C). \tag{2.3.1}$$

In order to get this idea, one can express the left hand side of (2.3.1) as follows,

$$C = \log(\mathrm{bch}(X, Y)) = X + Y + \sum_{k=2}^{\infty} C_k,$$

where $C_k(X, Y)$ indicates a homogeneous Lie polynomial of degree k, namely a linear combination of commutators of the form $[V_1, [V_2, \dots, [V_{k-1}, V_k] \dots]]$ with $V_i \in X, Y$ for $1 \leq i \leq k$

[16]. The first terms of the series C_k can be written as

$$\begin{aligned} C_2 &= \frac{1}{2}[X,Y], \\ C_3 &= -\frac{1}{12}[[X,Y],X] + \frac{1}{12}[[X,Y],Y], \\ C_4 &= \frac{1}{24}[[[X,Y],Y],X]. \end{aligned}$$

Explicit expressions for this series up to degree of $k = 20$ in a Hall basis of the free Lie algebra $\pounds(X,Y)$ can be found in [32].
Let us consider integrator (2.2.3) formally expressed as a product of exponentials of vector fields (2.2.14). Applying recursively the BCH formula to (2.2.14), one gets

$$\begin{aligned} \log(\Xi(h)) = hu_1Y_1 + h^2u_2Y_2 + h^3(u_3Y_3 + u_{12}[Y_1,Y_2]) \\ + h^4(u_4Y_4 + u_{13}[Y_1,Y_3] + u_{112}[Y_1,[Y_1,Y_2]]) + O(h^5) \end{aligned}$$

where $u_{i_1 i_2 \ldots i_m}$ are polynomials in the coefficients $\alpha_1, \ldots, \alpha_{2s}$ of the degree $n = i_1 + \ldots + i_m$, and whose first terms can be written as [16]

$$u_1 = \sum_{j=1}^{2s} \alpha_j, \qquad u_2 = \sum_{j=1}^{2s} (-1)^j \alpha_j^2, \qquad u_2 = \sum_{j=1}^{2s} \alpha_j^3.$$

By imposing equation (2.2.11) to satisfy that the composition integrator (2.2.3) is of order $r \geq 1$, one gets order conditions

$$u_1 = 1 \qquad \text{and} \qquad u_{i_1 i_2 \ldots i_m} = 0, \qquad \text{whenever} \qquad 2 \leq i_1 + \ldots + i_m \leq r.$$

On the other hand, one can have the order conditions of splitting method (2.2.5) expressed as (2.2.15) in the coefficients a_i, b_i with similar process as above. Applying recursively the BCH formula to the (2.2.15), one gets a series expansion of $\log(\Xi(h)) = \sum_{n\geq 1} h^n L_n$ in the form:

$$\begin{aligned} \log(\Xi(h)) = h(w_a L_A + w_b L_B) + h^2 w_{ab} L_{AB} + h^3(w_{abb} L_{ABB} + w_{aba} L_{ABA}) \\ + h^4(w_{abbb} L_{ABBB} + w_{abba} L_{ABBA} + w_{abaa} L_{ABAA}) + O(h^5), \end{aligned} \tag{2.3.2}$$

where

$$\begin{aligned} L_{AB} = [L_A, L_B], \quad L_{ABB} = [L_{AB}, L_B], \quad L_{ABA} = [L_{AB}, L_A], \\ L_{ABBB} = [L_{ABB}, L_B], \quad L_{ABBA} = [L_{ABB}, L_A], \quad L_{ABAA} = [L_{ABA}, L_A], \end{aligned}$$

and $w_a, w_{ab}, w_{abb}, w_{aba}, w_{abbb}, w_{abba}, w_{abaa}, \ldots$ indicate polynomials in the coefficients a_i, b_i of the splitting method (2.2.5). Some of these polynomials can be written as [16]

$$
\begin{aligned}
w_a &= \sum_{i=1}^{s} a_i, \\
w_b &= \sum_{i=1}^{s+1} b_i, \\
w_{ab} &= \frac{1}{2} w_a w_b - \sum_{1 \le i \le j \le s} b_i a_j, \\
w_{aba} &= -\frac{1}{12} w_a^2 w_b + \frac{1}{2} \left(\sum_{1 \le i < j \le k \le s} a_i b_j a_k \right), \\
w_{abb} &= \frac{1}{12} w_a w_b^2 - \frac{1}{2} \left(\sum_{1 \le i \le j < k \le s+1} a_i b_j a_k \right).
\end{aligned}
$$

To reach the desired order accuracy for the splitting method (2.2.5), one needs $w_a = w_b = 1$ and $w_{ab} = w_{aba} = w_{abb} = \ldots = 0$ in equation (2.3.2). The vector fields $L_A, L_B, L_{AB}, L_{ABA}$, L_{ABB} in (2.3.2) are considered as a basis of the free Lie algebra on the alphabet A, B. In this case the set of order conditions are independent. However a basis of the free Lie algebra in the equation (2.3.2) is the Hall basis (basis of P. Hall) with the Hall words A, B, AB, ABA, ABB, $ABBB, ABBA, ABAA, \ldots$ [16, 84]. For each Hall word j, the coefficients w_j can be obtained by making use of the results of [77]. $aaa\mathbb{RRRR}\mathcal{O}(a)$

2.3.1 Runge-Kutta-Nyström methods (RKN)

This class of methods appears in the design of numerical integrator for second-order differential systems

$$y'' = g(y), \qquad y \in \mathbb{R}^d \qquad \text{and} \qquad g : \mathbb{R}^d \to \mathbb{R}^d. \tag{2.3.3}$$

One can easily rewrite (2.3.3) as a first order system by bringing in a new variable $x = (y, z)$ with $y' = z$, that is

$$F^{[A]}(y, z) = (z, 0), \qquad F^{[B]}(y, z) = (0, g(y)),$$

where $F^{[A]} : \mathbb{R}^{2d} \to \mathbb{R}^{2d}$ and $F^{[B]} : \mathbb{R}^{2d} \to \mathbb{R}^{2d}$, then

$$x' = F^{[A]}(x) + F^{[B]}(x).$$

Splitting scheme (2.2.5) can be considered for this class of problems by defining exact flows for $x' = F^{[A]}(x), x' = F^{[B]}(x)$ as $\varphi^{[A]}, \varphi^{[B]}$ respectively. These methods have a common application for deriving numerical integrator with Hamiltonian problems of the form

$$M \frac{d^2}{dt^2} q = -\nabla_q V(q), \tag{2.3.4}$$

with $p = M \frac{d}{dt} q$. Notice that equation (2.3.4) corresponds in the form (2.3.3) with $y = Mq$, $y' = z = p$ and $g(y) = -\nabla_q V(q)$. In that case the Hamiltonian is written in the form $H(q, p) = T(p) + V(q)$, where $T(p)$ is quadratic in momenta

$$T(p) = \frac{1}{2} p^T M p,$$

where M is diagonal mass matrix and $V(q)$ is the potential. It is well known that the Lie algebra of Hamiltonian functions under the Poisson bracket is isomorphic the Lie algebra of Hamiltonian vector fields. Thus, for the RKN case L_B and L_{ABB} in (2.3.2) commute, namely

$$[[[L_A, L_B], L_B], L_B] = 0.$$

This reduces the number of order conditions in (2.3.2). In that case more efficient methods can be designed, in particular when the order of method $r \geq 4$ [16, 23]. On the other hand one can apply this class of methods to the general system

$$y'' = My' + g_1(y) + g_2(y).$$

Here RKN methods are more efficient when the part $F^{[A]}(y, z) = (z, Mz + g_1(y))$ which corresponds to $y'' = My' + g_1(y)$ is easily solvable. Now, for Hamiltonian systems, this generalization corresponds to $H(q, p) = T(q, p) + V(q)$, where

$$T(q, p) = \frac{1}{2} p^T M(q) p + f^T(q) p.$$

In addition, one can consider the generalized harmonic oscillator with corresponding Hamiltonian equation $H(q, p) = \frac{1}{2} p^T M p + V(q)$ where $V(q) = \frac{1}{2} q^T K q$ and $M, K \in \mathbb{R}^{d \times d}$.

In that case, increasing the order from $r = 2k - 1$ to $r = 2k$ needs only one independent condition and increasing the order from $r = 2k$ to $r = 2k + 1$ only two. Thus the order conditions can be rapidly reduced for the desired order of accuracy. RKN methods require detailed work to design efficient methods based on physical problems in their application area. For RKN methods it is important to note that the vector fields $F^{[A]}, F^{[B]}$ have qualitatively different properties and they are not exchangeable [16].

2.3.2 Near-integrable systems

Let us consider an ODE system with a small parameter $|\varepsilon| \ll 1$ in the form

$$\frac{d}{dt} x = F^{[A]}(x) + \varepsilon F^{[B]}(x), \qquad x(0) = x_0 \in \mathbb{R}^d,$$

which can be split in two parts as follows

$$\frac{d}{dt} x = F^{[A]}(x) \qquad \frac{d}{dt} x = \varepsilon F^{[B]}(x).$$

We denote the exact $h-$flows of each part $x' = F^{[A]}(x)$ and $x' = \varepsilon F^{[B]}(x)$ by $\varphi_h^{[A]}(x_0)$, $\varphi_h^{[B]}(x_0)$, respectively. Here, each flow of the system is exactly integrable or can be numerically computed at $t = h$, the time step. One can apply the integrator scheme (2.2.5) to the near-integrable system and can select arbitrary coefficients a_i, b_i. In this case a formal expression of the series $\Xi(h)$ of differential operators associated to the integrator Φ_h in (2.2.5) can be written as

$$\Xi(h) = e^{b_1 h \varepsilon L_B} e^{a_1 h L_A} \ldots e^{b_s h \varepsilon L_B} e^{a_s h L_A} e^{b_{s+1} h \varepsilon L_B}, \tag{2.3.5}$$

with Lie derivatives (2.2.16) of $F^{[A]}(x), F^{[B]}(x)$, respectively. Applying recursively the BCH formula to (2.3.5), one gets a similar expression as (2.3.2) with εL_B instead of L_B

$$\begin{aligned}\log(\Xi(h)) = hw_aL_A + \varepsilon(hw_bL_B + h^2w_{ab}L_{AB} + h^3w_{aba}L_{ABA} + h^4w_{abaa}L_{ABAA})\\ + \varepsilon^2(h^3w_{abb}L_{ABB} + h^4w_{abba}L_{ABBA}) + \varepsilon^3h^4w_{abbb}L_{ABBB} + O(\varepsilon h^5).\end{aligned}$$

Here ε (magnitude of perturbation) and h (magnitude of step) are small numbers, typically $\varepsilon \ll h$ or $\varepsilon \approx h$. Then one can consider to eliminate error terms in small powers of ε in place of h and built a method has order $(k_1, k_2, ...)$ if $log(\Xi(h)) - hL$ is of order $O(\sum \varepsilon^j h^{k_j+1})$ where $L = L_A + \varepsilon L_B$ [16]. The $\log(\Xi(h))$ consists a finite number of terms at each order in h and an infinite number of terms at each order in ε. Following [69], in the general case the number of independent Lie brackets of L_A, L_B at order $\varepsilon^j h^k$ in $\log(\Xi(h))$ is given as

$$\frac{1}{k}\sum_{d|j \atop d|(k-j)} \frac{(\frac{k}{d})!}{(\frac{j}{d})!\,(\frac{k-j}{d})!},$$

and tabulated in the matrix form

$$\begin{array}{c} \text{orders} \\ h^2 \\ h^3 \\ h^4 \\ h^5 \\ h^6 \\ h^7 \\ \\ \end{array} \begin{array}{c} \begin{array}{cccccc} \varepsilon^1 & \varepsilon^2 & \varepsilon^3 & \varepsilon^4 & \varepsilon^5 & \varepsilon^6 \end{array} \\ \left[\begin{array}{cccccc} 1 & & & & & \\ 1 & 1 & & & & \\ 1 & 1 & 1 & & & \\ 1 & 2 & 2 & 1 & & \\ 1 & 2 & 3 & 2 & 1 & \\ 1 & 3 & 5 & 5 & 3 & 1 \\ \vdots & & & & & \end{array}\right. \begin{array}{c} \\ \\ \\ \\ \\ \\ \ddots \end{array} \left.\begin{array}{c} \\ \\ \\ \\ \\ \\ \end{array}\right] \end{array} \tag{2.3.6}$$

Notice that there is only one term for each order k in (2.3.6) such that $\varepsilon h^k w_{aba\ldots a}L_{ABA\ldots A}$. Thus it is easy to annihilate errors of order εh^k and this depends commonly on the particular problems studied for eliminating dominant error in a given method [16, 69]. In consequence, various methods depend on orders in parameters h and ε can be found in the literature and some of them tabulated in [16].

2.4 The Magnus Expansion (ME)

The content of this chapter is the solution of the initial value problem associated with the linear ordinary differential equation

$$\frac{dY}{dt} = A(t)Y(t), \qquad Y(t_0) = Y_0, \tag{2.4.1}$$

where Y is a vector valued(or matrix)(real or complex) function, and A a matrix valued(real or complex) function. If $A(t_1)A(t_2) = A(t_2)A(t_1)$, for any values of t, t_1 and t_2, then one can write the solution as

$$Y(t) = \exp\left(\int_{t_0}^{t_1} A(s)ds\right) Y_0. \tag{2.4.2}$$

In the general case, $A(t_1)A(t_2) \neq A(t_2)A(t_1)$, and (2.4.2) does not give the solution. Magnus [65] proposed to represent the solution of the linear evolution equation

$$\frac{dY}{dt} = A(t)Y(t), \qquad Y(t_0) = I, \tag{2.4.3}$$

as the exponential of a certain function

$$Y(t) = \exp(\Omega(t, t_0))Y_0.$$

To find an expression for the solution with Magnus' proposal, one can use the $n \times n$ matrix $U(t, t_0)$ defined through

$$Y(t) = U(t, t_0)Y_0, \qquad U(t_0, t_0) = I. \tag{2.4.4}$$

Substituting (2.4.4) in (2.4.1), one gets linear matrix differential equations

$$\frac{dU}{dt}(t, t_0) = A(t)U(t, t_0), \qquad U(t_0, t_0) = I, \tag{2.4.5}$$

where I is $n \times n$ identity matrix. Here, $U(t, t_0)$ is called as the time evolution operator. Now, the solution of (2.4.5) can be written as a matrix exponential in the form

$$U(t, t_0) = \exp(\Omega(t, t_0)), \qquad \Omega(t_0, t_0) = 0,$$

and a series expansion for the matrix in the exponent

$$\Omega(t, t_0) = \sum_{k=1}^{\infty} \Omega_k(t, t_0), \tag{2.4.6}$$

which is called the Magnus expansion [18]. Three first terms of that series are given as follows

$$\Omega_1(t, t_0) = \int_{t_0}^{t} A(t_1)dt_1,$$
$$\Omega_2(t, t_0) = \frac{1}{2}\int_{t_0}^{t} dt_1 \int_{t_0}^{t_1} dt_2[A(t_1), A(t_2)],$$
$$\Omega_3(t, t_0) = \frac{1}{6}\int_{t_0}^{t} dt_1 \int_{t_0}^{t_1} dt_2 \int_{t_0}^{t_2} dt_3([A(t_1), [A(t_2), A(t_3)]] + [A(t_3), [A(t_2), A(t_1)]],$$

where $[A, B] = AB - BA$ stands for the matrix commutator of A and B. Notice that, $\Omega_1(t, t_0)$ corresponds the expression in the exponent of equation (2.4.2) [18].

2.4.1 Derivative of the exponential and its inverse

One can write matrix commutators for fixed Ω, $[\Omega, A] = \Omega A - A\Omega$, as a linear operator $A \longmapsto [\Omega, A]$

$$\mathrm{ad}_\Omega(A) = [\Omega, A], \qquad \mathrm{ad}^k_\Omega(A) = [\Omega, \mathrm{ad}^{k-1}_\Omega(A)], \qquad \mathrm{ad}^0_\Omega(A) = A, \qquad (k \in \mathbb{N})$$

which is called the adjoint operator[47, 100]. On the other hand, the exponential of this $\mathrm{ad}_\Omega(A)$ operator, $\mathrm{Ad}_\Omega(A) = \exp(\mathrm{ad}_\Omega(A))$, is given as [18]

$$\mathrm{Ad}_\Omega(A) = \exp(\Omega)A\exp(-\Omega) = \sum_{k=0}^{\infty} \frac{1}{k!}\mathrm{ad}^k_\Omega(A).$$

Lemma 2.4.1. *[18] The derivative of a matrix exponential given by*

(1) $\frac{d}{dt}\exp(\Omega(t,t_0)) = \mathrm{dexp}_{\Omega(t,t_0)}(\Omega'(t,t_0))\exp(\Omega(t,t_0)),$

(2) $\frac{d}{dt}\exp(\Omega(t,t_0)) = \exp(\Omega(t,t_0))\mathrm{dexp}_{-\Omega(t,t_0)}(\Omega'(t,t_0)),$

(3) $\frac{d}{dt}\exp(\Omega(t,t_0)) = \int_0^1 e^{x\Omega(t,t_0)}\Omega'(t,t_0)e^{(1-x)\Omega(t,t_0)}dx,$

where

$$\mathrm{dexp}_\Omega(A) = \sum_{k=0}^{\infty}\frac{1}{(k+1)!}\mathrm{ad}_\Omega^k(A) \equiv \frac{\exp(\mathrm{ad}_\Omega)-I}{\mathrm{ad}_\Omega}(A). \tag{2.4.7}$$

The series (2.4.7) converges everywhere.

Lemma 2.4.2. *[9] $dexp_\Omega$ is invertible if the eigenvalues of the linear operator ad_Ω are different from $2j\pi i$ with $j \in \{\pm1, \pm2, ...\}$. In addition, if $\|\Omega\| < \pi$ then, a convergent expansion $\mathrm{dexp}_\Omega^{-1}(A)$ is given as*

$$\mathrm{dexp}_\Omega^{-1}(A) = \sum_{k=0}^{\infty}\frac{B_k}{k!}\mathrm{ad}_\Omega^k(A) \equiv \frac{\mathrm{ad}_\Omega}{\exp(\mathrm{ad}_\Omega)-I}(A)$$

where B_k are the Bernoulli numbers, defined by

$$B_k = \sum_{j=0}^{\infty}\frac{B_j}{j!}x^k.$$

The values of the first few Bernoulli numbers are $B_0 = 1, B_1 = -\frac{1}{2}, B_2 = \frac{1}{6}, B_4 = -\frac{1}{30}$. In general $B_k = 0$ for all odd $k > 1$.

Theorem 2.4.3. *[18] The solution of the differential equation $Y' = A(t)Y$ with initial condition $Y(t_0) = Y_0$ can be written as $Y(t) = \exp(\Omega(t,t_0))Y(t_0)$ with $\Omega(t,t_0)$ defined by*

$$\Omega'(t,t_0) = \mathrm{dexp}^{-1}_{\Omega(t,t_0)}(A(t)), \qquad \Omega(t_0,t_0) = 0. \tag{2.4.8}$$

Proofs of the Lemma 2.4.1, Lemma 2.4.2 and Theorem 2.4.3 can be found in [18, 47] and references therein. Let us make use of the Lemma (2.4.2) to expand equation (2.4.8), that is,

$$\Omega'(t,t_0) = A(t) - \frac{1}{2}[\Omega(t,t_0),A(t)] + \frac{1}{12}[\Omega(t,t_0),[\Omega(t,t_0),A(t)]] + \dots. \tag{2.4.9}$$

The differential equation (2.4.9) is nonlinear in $\Omega(t,t_0$. By applying Picard iteration, one gets

$$\Omega^{[k]}(t,t_0) = \int_{t_0}^{t}\Big(A(t_1) - \frac{1}{2}[\Omega^{[k-1]}(t_1,t_0),A(t_1)] + \frac{1}{12}[\Omega^{[k-1]}(t_1,t_0),[\Omega^{[k-1]}(t_1,t_0),A(t_1)]] + \dots\Big)dt_1,$$

where

$$\Omega^{[0]}(t,t_0) = 0, \qquad \Omega^{[1]}(t,t_0) = \int_{t_0}^{t} A(t_1)dt_1.$$

Then the solution in a appropriate small neighborhood of the origin to (2.4.9) is given by the limit:

$$\lim_{k\to\infty} \Omega^{[k]}(t,t_0) = \Omega(t,t_0).$$

2.4.2 First few terms of the Magnus expansion

Following [18], we suppose that A is of first order in some parameter ϵ. Now replace A by ϵA in (2.4.3) and determine the series

$$\Omega(t,t_0) = \sum_{k=1}^{\infty} \epsilon^k \Omega_k(t,t_0) \tag{2.4.10}$$

where Ω_k is of order ϵ^k. Notice that for $\epsilon = 1$ one gets (2.4.6). However, the first few terms of the Magnus expansion can be obtained by substituting the series (2.4.10) in (2.4.8) and collecting terms with equal powers of ϵ and equating the corresponding terms. Then, denoting $A(t_i \equiv A_i$, the first three orders are obtained as follows,

1. $\Omega_1'(t,t_0) = A(t)$, so that
$$\Omega_1(t,t_0) = \int_{t_0}^{t} dt_1 A_1,$$

2. $\Omega_2'(t,t_0) = -\frac{1}{2}[\Omega_1(t,t_0), A(t)]$, so that
$$\Omega_2(t,t_0) = \frac{1}{2}\int_{t_0}^{t} dt_1 \int_{t_0}^{t_1} dt_2 [A_1, A_2],$$

3. $\Omega_3'(t,t_0) = -\frac{1}{2}[\Omega_1(t,t_0), A(t)] + \frac{1}{12}[\Omega_1(t,t_0), [\Omega_1(t,t_0), A(t)]]$, so that
$$\Omega_3(t,t_0) = \frac{1}{6}\int_{t_0}^{t} dt_1 \int_{t_0}^{t_1} dt_2 \int_{t_0}^{t_2} dt_3 ([A_1,[A_2,A_3]] + [A_3,[A_2,A_1]].$$

All the terms in the ME (2.4.6) can be written in a recursive manner as presented in [18]

$$\Omega_1(t,t_0) = \int_{t_0}^{t} A(t_1)dt_1, \tag{2.4.11}$$

$$\Omega_k(t,t_0) = \sum_{j=1}^{k-1} \frac{B_j}{j!} \sum_{\substack{k_1+\ldots+k_j=k-1 \\ k_1\geq 1 \ldots k_j \geq 1}} \int_{t_0}^{t} [\Omega_{k_1}(s,t_0), [\Omega_{k_2}(s,t_0), [\ldots, [\Omega_{k_j}(s,t_0), A(s)]\ldots]]] dt_s.$$

2.4.3 Time symmetry of the Magnus expansion

Let us consider the map $\varphi_t : Y(t_0) \to Y(t)$ associated to the linear differential equation (2.4.3) For any $t_0 \leq t_f$, integrating (2.4.3) from t_0 to t_f, followed by integrating from t_f back to t_0, recovers original initial value. Thus, the map φ_t satisfies (2.1.5) so that it is time symmetric. On the other hand, one can write $\varphi_{t_f}(Y_0) = U(t_f, t_0)Y_0$ from (2.4.4). Due to time symmetry property, the fundamental matrix verifies

$$U(t_0, t_f) = U^{-1}(t_f, t_0).$$

Then in the terms of the Magnus expansion,

$$\Omega(t_0, t_f) = -\Omega(t_f, t_0).$$

The main importance of the time symmetry of the Magnus expansion is that $\Omega(t, t_0)$ can be expanded in odd powers of t [56]. To illustrate this fact, the solution of (2.4.3) at the final time $t_f = t_0 + m$ can be written as

$$Y(t_{1/2} + \frac{m}{2}) = \exp(\Omega(t_{1/2} + \frac{m}{2}, t_{1/2} - \frac{m}{2}))Y(t_{1/2} - \frac{m}{2}), \tag{2.4.12}$$

where $t_{1/2} = (t_0 + t_f)/2$. By multiplying both sides of (2.4.12) from left with $\exp^{-1}$, one has

$$Y(t_{1/2} - \frac{m}{2}) = \exp(-\Omega(t_{1/2} + \frac{m}{2}, t_{1/2} - \frac{m}{2}))Y(t_{1/2} + \frac{m}{2}). \tag{2.4.13}$$

Morever, one can write the solution of (2.4.3) at t_0 as

$$Y(t_{1/2} - \frac{m}{2}) = \exp(\Omega(t_{1/2} - \frac{m}{2}, t_{1/2} + \frac{m}{2}))Y(t_{1/2} + \frac{m}{2}). \tag{2.4.14}$$

Then, by comparing (2.4.13) with (2.4.14), one arrives at

$$\Omega(t_{1/2} - \frac{m}{2}, t_{1/2} + \frac{m}{2}) = -\Omega(t_{1/2} + \frac{m}{2}, t_{1/2} - \frac{m}{2}),$$

and thus Ω contains only odd powers of m. Consequently, if one computes the Taylor series expansion of an analytic function $A(t)$ around to midpoint $t = t_{1/2}$, then each term in Ω_k will consist of odd powers of m [18].

2.4.4 Convergence of the Magnus expansion

In general, the Magnus series converges when A is small in a suitable sense. There are several bounds to the radius of convergence in terms of A which have been discovered by many authors. Before we review some of these results, let us consider $\Omega_k(t, t_0)$ given by (2.4.11). $\Omega(t, t_0) = \sum_{k=1}^{\infty} \Omega_k(t, t_0$ is absolutely convergent for $t_0 \leq t < T$, where

$$T = \max\left\{t \geq t_0 : \int_{t_0}^{t} \|A\|_2 \, ds < r_c\right\}.$$

The lower bounds for the convergence radius in the above formula has been computed as $r_c = \log 2 = 0.69314\ldots$ in [58, 82], $r_c = 0.57745\ldots$ in [37], $r_c = 1.08686\ldots$ in [17, 73].

For an improved radius of convergence result we refer to [31], where two different sufficient conditions for the convergence of the Magnus series are given. Here, we represent the first one for which the convergence of the series follows from the boundedness of the norm of the operator A.

Theorem 2.4.4. *[31] Let $A(t)$ be a bounded operator in a Hilbert space $\mathcal{H}$ for the differential equation $Y'(t) = A(t)Y(t)$ with $Y(t_0) = I$. The Magnus series $\Omega(t, t_0) = \sum_{k=1}^{\infty} \Omega_k(t, t_0)$, with $\Omega_k(t, t_0)$ given by* (2.4.11) *convergences on the interval $t_0 \leq t < T$ such that*

$$\int_{t_0}^{T} \|A\| \, ds < \pi.$$

and $Y(t) = \exp(\Omega(t, t_0))$. The statement also remains valid with a normal operator Y (in particular, with unitary Y).

2.4.5 Numerical integrators via Magnus expansion

The sufficient conditions for the convergence of the Magnus series given in Theorem (2.4.4) indicate that the series only converges locally. One can divide the time interval $[t_0, t_f]$ into N steps such that Magnus series convergences in each subinterval $[t_{n-1}, t_n]$, $n = 1, \dots N$, with $t_N = t_f$. Then, the solution of (2.4.1) at final time t_N can be written as

$$Y(t_N) = \prod_{n=1}^{N} \exp(\Omega(t_n, t_{n-1}))Y_0.$$

Numerical integration algorithms based on Magnus Expansion can be obtained with three steps given below [18].

1. the series $\Omega(t, t_0) = \sum_{k=1}^{\infty} \Omega_k$ is truncated at any order.

2. the multivariate integrals appear in the truncated series $\Omega^{[k]} = \sum_{i=1}^{k} \Omega_i$ are replaced by preferred approximations.

3. the exponential of the matrix $\Omega^{[k]}$ is computed.

As already mentioned, the Magnus Expansion is time symmetric. To take advantage of this symmetry one can consider the Taylor series expansion of $A(t)$ around the midpoint $t_{1/2} = t_n + h/2$,

$$A(t) = \sum_{j=0}^{\infty} \frac{1}{j!} \frac{d^j A(t)}{dt^j}\bigg|_{t=t_{1/2}} (t - t_{1/2})^j, \tag{2.4.15}$$

then substitute the series into (2.4.11). This allows one to obtain explicitly the expression of $\Omega_k(t_{n+1}, t_n)$ up to desired order in the Magnus expansion. For example, the expression of Ω_k

up to order h^8 can be written as,

$$\begin{aligned}
\Omega_1(t_{n+1},t_n) &= ha_0 + h^3\frac{1}{12}a_2 + h^5\frac{1}{80}a_4 + h^7\frac{1}{448}a_6 + \mathcal{O}(h^9),\\
\Omega_2(t_{n+1},t_n) &= h^3\frac{-1}{12}[a_0,a_1] + h^5\left(\frac{-1}{80}[a_0,a_3] + \frac{1}{240}[a_1,a_2]\right)\\
&\quad + h^7\left(\frac{-1}{448}[a_0,a_5] + \frac{1}{2240}[a_1,a_4] - \frac{1}{1344}[a_2,a_3]\right) + \mathcal{O}(h^9),\\
\Omega_3(t_{n+1},t_n) &= h^5\left(\frac{1}{360}[a_0,a_0,a_2] - \frac{1}{240}[a_1,a_0,a_1]\right) + h^7\Big(\frac{1}{1680}[a_0,a_0,a_4]\\
&\quad - \frac{1}{2240}[a_0,a_1,a_3] + \frac{1}{6720}[a_1,a_1,a_2]\\
&\quad + \frac{1}{6048}[a_2,a_0,a_2] - \frac{1}{840}[a_3,a_0,a_1]\Big) + \mathcal{O}(h^9),
\end{aligned} \tag{2.4.16}$$

$$\begin{aligned}
\Omega_4(t_{n+1},t_n) &= h^5\frac{1}{720}[a_0,a_0,a_0,a_1] + h^7\Big(\frac{1}{6720}[a_0,a_0,a_0,a_3]\\
&\quad - \frac{1}{7560}[a_0,a_0,a_1,a_2] + \frac{1}{4032}[a_0,a_2,a_0,a_1]\\
&\quad + \frac{11}{60480}[a_1,a_0,a_0,a_2] - \frac{1}{6720}[a_1,a_1,a_0,a_1]\Big) + \mathcal{O}(h^9),\\
\Omega_5(t_{n+1},t_n) &= h^7\Big(\frac{-1}{15120}[a_0,a_0,a_0,a_0,a_2] - \frac{1}{30240}[a_0,a_0,a_1,a_0,a_1]\\
&\quad + \frac{1}{7560}[a_1,a_0,a_0,a_0,a_1]\Big),\\
\Omega_6(t_{n+1},t_n) &= h^7\Big(\frac{-1}{30240}[a_0,a_0,a_0,a_0,a_0,a_1]\Big),
\end{aligned}$$

where $a_j = \frac{1}{j!}\frac{d^jA(t)}{dt^j}|_{t=t_{1/2}}$ and $[a_{i_1},a_{i_2},\dots,a_{i_{k-1}},a_{i_k}]$=$[a_{i_1},[a_{i_2},[\dots,[a_{i_{k-1}},a_{i_k}]\dots]]]$. Notice that, due to time symmetric property of the Magnus expansion, each term in Ω_k consist of odd powers of h. Let $\alpha_i \equiv h^i a_{i-1}$, then the commutator $[\alpha_{i_1},\alpha_{i_2},\dots,\alpha_{i_k}]$ becomes an element of order $h^{i_1+i_2+\cdots+i_k}$. The matrices $\{\alpha_i\}_{i\geq 1}$ can be thought of as the generators of a graded free Lie algebra $\mathscr{L}(\alpha_1,\alpha_2,\dots)$. Consequently, methods of order $p \equiv 2s$ can be obtained if one considers the series Ω only with terms $\alpha_1,\alpha_2,\dots\alpha_s$ such that these terms can be obtained by a quadrature of order $2s$ to approximate integrals [76]. For example, up to order four, six and eight, for a general linear system using the Lie algebra generated by $\{\alpha_1,\alpha_2\}$, $\{\alpha_1,\alpha_2,\alpha_3\}$ and $\{\alpha_1,\alpha_2,\alpha_3,\alpha_4\}$ respectively read,

$$\Omega^{[4]} = \alpha_1 - \frac{1}{12}[12] + \mathcal{O}(h^5), \tag{2.4.17}$$

$$\begin{aligned}
\Omega^{[6]} &= \alpha_1 + \frac{1}{12}\alpha_3 - \frac{1}{12}[12] + \frac{1}{240}[23] + \frac{1}{360}[113]\\
&\quad - \frac{1}{240}[212] + \frac{1}{720}[1112] + \mathcal{O}(h^7),
\end{aligned} \tag{2.4.18}$$

$$\begin{aligned}\Omega^{[8]} = \alpha_1 &+ \frac{1}{12}\alpha_3 - \frac{1}{12}[12] + \frac{1}{240}[23] + \frac{1}{360}[113] - \frac{1}{240}[212] \\ &+ \frac{1}{720}[1112] - \frac{1}{80}[14] - \frac{1}{1344}[34] - \frac{1}{2240}[124] + \frac{1}{6720}[223] \\ &+ \frac{1}{6048}[313] - \frac{1}{840}[412] + \frac{1}{6720}[1114] - \frac{1}{7560}[1123] + \frac{1}{4032}[1312] \\ &+ \frac{11}{60480}[2113] - \frac{1}{6720}[2212] - \frac{1}{15120}[11113] - \frac{1}{30240}[11212] \\ &+ \frac{1}{7560}[21112] - \frac{1}{30240}[111112] + \mathcal{O}(h^9), \end{aligned} \tag{2.4.19}$$

where $[ij \dots kl]$ represents the nested commutator $[\alpha_i, [\alpha_j, [\dots, [\alpha_k, \alpha_l] \dots]]]$. Let us now introduce the averaged matrices [18]

$$A^{(i)}(h) \equiv \frac{1}{h^i}\int_{t_n}^{t_n+h}(t - t_{1/2})^i A(t)dt \equiv \frac{1}{h^i}\int_{h/2}^{-h/2} t^i A(t + t_{1/2})dt, \tag{2.4.20}$$

where $i = 0, 1, \dots, s-1$. If one inserts the Taylor series (2.4.15) into the (2.4.20), one gets

$$\begin{aligned} A^{(0)}(h) &= a_0 + \frac{1}{12}h^2 a_2 + \frac{1}{80}h^4 a_4 + \frac{1}{448}h^6 a_6 + \cdots, \\ A^{(1)}(h) &= \frac{1}{12}h a_1 + \frac{1}{80}h^3 a_3 + \frac{1}{448}h^5 a_5 + \cdots, \\ A^{(2)}(h) &= \frac{1}{12}a_0 + \frac{1}{80}h^2 a_2 + \frac{1}{448}h^4 a_4 + \frac{1}{2304}h^6 a_6 + \cdots, \end{aligned}$$

and so on. On the other hand, one can easily rewrite the expression Ω_k given in (2.4.16) in the terms of the $A^{(i)}$ [19]. Let us consider the single integrals given by (2.4.20). Their exact evaluation may not be possible, or may be computationally more costly. In order to compute these integrals one can use a numerical quadrature. It is well known that a quadrature formula is determined by the specification of the weights and nodes. If one considers any set of nodes $c_1, c_2, \dots, c_k$ and weights $b_1, b_2, \dots, b_k$ of a particular quadrature rule such that it is of order p, one gets

$$A^{(0)}(h) = \int_{t_n}^{t_n+h} A(t)dt = h\sum_{i=1}^{k} b_i A_i + \mathcal{O}(h^{p+1}),$$

where $A_i \equiv A(t_n + c_i h)$ and

$$A^{(i)}(h) = h\sum_{j=1}^{k} b_j \left(c_j - \frac{1}{2}\right)^i A_j + \mathcal{O}(h^{p+1}), \qquad i = 0, 1, \dots, s-1,$$

or equivalently, $A^{(i)} = h\sum_{j=1}^{k}(Q_C^{(s,k)})_{ij}A_j$ with $(Q_C^{(s,k)})_{ij} = b_j(c_j - \frac{1}{2})^i$. Here the quadrature rule referred as C, which is specified by the weights and nodes. Let G be the fourth, sixth and eight order Gauss-Legendre quadrature rule which is taken instead of C with $s = k = 2$, $s = k = 3$ and $s = k = 4$ respectively. Gauss-Legendre quadrature nodes and weights given as for order four

$$b_1 = b_2 = \frac{1}{2}, \qquad c_1 = \frac{1}{2} - \frac{\sqrt{3}}{6}, \qquad c_2 = \frac{1}{2} + \frac{\sqrt{3}}{6}, \tag{2.4.21}$$

order six

$$b_1 = b_3 = \frac{5}{18}, \quad b_2 = \frac{4}{9} \quad c_1 = \frac{1}{2} - \frac{\sqrt{15}}{10}, \quad c_2 = \frac{1}{2}, \quad c_3 = \frac{1}{2} + \frac{\sqrt{15}}{10}, \tag{2.4.22}$$

order eight,

$$c_i = \frac{1}{2} - v_i,\ c_{5-i} = \frac{1}{2} + v_i, \quad b_i = \frac{1}{2} - w_i,\ b_{5-i} = \frac{1}{2} + w_i, \tag{2.4.23}$$

$i = 1, 2$, and where

$$v_1 = \frac{1}{2}\sqrt{\frac{3 + 2\sqrt{6/5}}{7}},\ \ v_2 = \frac{1}{2}\sqrt{\frac{3 - 2\sqrt{6/5}}{7}}, \quad w_1 = \frac{1}{2} - \frac{1}{6}\sqrt{\frac{5}{6}},\ \ w_2 = \frac{1}{2} + \frac{1}{6}\sqrt{\frac{5}{6}}.$$

so that

$$Q_G^{(2,2)} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ -\frac{\sqrt{3}}{12} & \frac{\sqrt{3}}{12} \end{pmatrix}, \tag{2.4.24}$$

$$Q_G^{(3,3)} = \begin{pmatrix} \frac{5}{18} & \frac{4}{9} & \frac{5}{18} \\ -\frac{\sqrt{15}}{36} & 0 & \frac{\sqrt{15}}{36} \\ \frac{1}{24} & 0 & \frac{1}{24} \end{pmatrix}, \tag{2.4.25}$$

$$Q_G^{(4,4)} = \begin{pmatrix} \frac{1}{6}\sqrt{\frac{5}{6}} & -\frac{1}{6}\sqrt{\frac{5}{6}} & 1+\frac{1}{6}\sqrt{\frac{5}{6}} & 1-\frac{1}{6}\sqrt{\frac{5}{6}} \\ -\frac{1}{12}\sqrt{\frac{1}{42}(15+2\sqrt{30})} & \frac{1}{12}\sqrt{\frac{1}{42}(15-2\sqrt{30})} & \frac{1}{12}\sqrt{\frac{1}{210}(2595-262\sqrt{30})} & \sqrt{\frac{173}{2016}+\frac{131}{504\sqrt{30}}} \\ \frac{1}{336}(4+\sqrt{30}) & \frac{1}{336}(4-\sqrt{30}) & \frac{2}{21}-\frac{19}{56\sqrt{30}} & \frac{2}{21}+\frac{19}{56\sqrt{30}} \\ -\frac{1}{336}\sqrt{\frac{1}{70}(585+106\sqrt{30})} & \frac{1}{336}\sqrt{\frac{1}{70}(585-106\sqrt{30})} & \frac{1}{1680}\sqrt{\frac{91125}{14}-\frac{8203\sqrt{30}}{7}} & \frac{1}{1680}\sqrt{\frac{91125}{14}+\frac{8203\sqrt{30}}{7}} \end{pmatrix}. \tag{2.4.26}$$

Morever, inserting (2.4.15) into (2.4.20) yields

$$A^{(i)} = \sum_{j=1}^{s} (K^{(s)})_{ij}\alpha_j \equiv \sum_{j=1}^{s} \frac{1 - (-1)^{i+j}}{(i+j)2^{i+j}}\alpha_j + \mathcal{O}(h^{p+1}), \qquad 0 \le i \le s-1. \tag{2.4.27}$$

Now, it is possible to write the expression of α_i in terms of $A^{(i)}$ or A_j as

$$\alpha_i = \sum_{j=1}^{s} \left(T^{(s)}\right)_{ij} A^{(j-1)} = h \sum_{j=1}^{k} \left(T^{(s)} Q_C^{(s,k)}\right)_{ij} A_j, \tag{2.4.28}$$

whith $T^{(s)} = (K^{(s)})^{-1}$. On the other hand, if one considers the expression of Ω_k in (2.4.16) then notices that the term $\frac{1}{12}\alpha_3$ is not included in (2.4.17) and $\frac{1}{80}\alpha_5$ and $-\frac{1}{80}[\alpha_1, \alpha_4]$ are not in (2.4.18) and $\frac{1}{80}\alpha_5$, $\frac{1}{2240}[\alpha_2, \alpha_5]$, $-\frac{1}{448}[\alpha_1, \alpha_6]$ and $\frac{1}{80}\alpha_7$ are not in (2.4.19). These neglected terms are reproduced when either $A^{(i)}$'s are computed analytically or numerically by using any symmetric quadrature rule of desired order or higher [18]. Let us consider Gauss-Legendre quadrature rule in which nodes and weights given in (2.4.21), (2.4.22), (2.4.23) to present Magnus integrator of order 4,6 and 8 respectively.

Fourth Order Magnus Integrator: In this case one has to compute α_i as

$$\alpha_i = \sum_{j=1}^{2} \left(T^{(2)}\right)_{ij} A^{(j-1)} = h \sum_{j=1}^{2} \left(T^{(2)} Q_G^{(2,2)}\right)_{ij} A_j,$$

where $Q_G^{(2,2)}$ given by (2.4.24), $A_1 = A\left(t_n + \left(\frac{1}{2} - \frac{\sqrt{3}}{6}\right)h\right)$, $A_2 = A\left(t_n + \left(\frac{1}{2} + \frac{\sqrt{3}}{6}\right)h\right)$ and

$$T^{(2)} = \begin{pmatrix} 1 & 0 \\ 0 & 12 \end{pmatrix},$$

obtained from inverting relation (2.4.27). With this procedure one gets

$$\alpha_1 = \frac{h}{2}(A_1 + A_2), \qquad \alpha_2 = \frac{h\sqrt{3}}{2}(A_2 - A_1),$$

and then inserting these into (2.4.17) one obtains

$$\Omega^{[4]}(h) = \frac{h}{2}(A_1 + A_2) - h^2 \frac{\sqrt{3}}{12}[A_1, A_2],$$
$$Y_{n+1} = \exp(\Omega^{[4]}(h)) Y_n.$$

Sixth Order Magnus Integrator: One can similarly compute the α_i's from

$$\alpha_i = \sum_{j=1}^{3} \left(T^{(3)}\right)_{ij} A^{(j-1)} = h \sum_{j=1}^{3} \left(T^{(3)} Q_G^{(3,3)}\right)_{ij} A_j,$$

where $Q_G^{(3,3)}$ given by (2.4.25),

$$A_1 = A\left(t_n + \left(\frac{1}{2} - \frac{\sqrt{15}}{10}\right)h\right), A_2 = A\left(t_n + \frac{1}{2}h\right), A_3 = A\left(t_n + \left(\frac{1}{2} + \frac{\sqrt{15}}{10}\right)h\right)$$

and

$$T^{(3)} = \begin{pmatrix} 9/4 & 0 & -15 \\ 0 & 12 & 0 \\ -15 & 0 & 180 \end{pmatrix},$$

as follows

$$\alpha_1 = hA_2, \qquad \alpha_2 = \frac{h\sqrt{15}}{3}(A_3 - A_1), \qquad \alpha_3 = \frac{10h}{3}(A_3 - 2A_2 + A_1).$$

Replacing these in (2.4.18) yields the sixth order approximation $Y_{n+1} = \exp(\Omega^{[6]}(h)) Y_n$

Eight Order Magnus Integrator: For the implementation of the eighth-order methods, the α_i's are computed by using formula

$$\alpha_i = \sum_{j=1}^{4} \left(T^{(4)}\right)_{ij} A^{(j-1)} = h \sum_{j=1}^{2} \left(T^{(4)} Q_G^{(4,4)}\right)_{ij} A_j,$$

where

$$T^{(4)} = \begin{pmatrix} 9/4 & 0 & -15 & 0 \\ 0 & 75 & 0 & -420 \\ -15 & 0 & 180 & 0 \\ 0 & -420 & 0 & 2800 \end{pmatrix},$$

and $Q_G^{(4,4)}$ given by (2.4.26). Then, one has

$$\begin{array}{lllllll} \alpha_1 & = & \frac{3}{4}\,(3A^{(0)} - 20A^{(2)}), & & \alpha_3 & = & -15\,(A^{(0)} - 12A^{(2)}), \\ \alpha_2 & = & 15\,(5A^{(1)} - 28A^{(3)}), & & \alpha_4 & = & -140\,(3A^{(1)} - 20A^{(3)}), \end{array} \tag{2.4.29}$$

and the $A^{(i)}$, $i = 1, 2, 3, 4$ are approximated with the 8th order Gauss-Legendre quadrature rule (2.4.23). Letting $S_1 = A_1 + A_4$, $S_2 = A_2 + A_3$, $R_1 = A_4 - A_1$, $R_2 = A_3 - A_2$, we reach the expression

$$A^{(0)} = \frac{1}{2}\left(w_1 S_1 + w_2 S_2\right), \qquad A^{(1)} = \frac{1}{2}\left(w_1 v_1 R_1 + w_2 v_2 R_2\right),$$

$$A^{(2)} = \frac{1}{2}\left(w_1 v_1^2 S_1 + w_2 v_2^2 S_2\right), \quad A^{(3)} = \frac{1}{2}\left(w_1 v_1^3 R_1 + w_2 v_2^3 R_2\right).$$

By using the values α_i in (2.4.29) into the (2.4.19), one reaches the eight order approximation $Y_{n+1} = \exp(\Omega^{[8]}(h))Y_n$.

2.4.6 Commutator-Free Magnus integrators

The Magnus integrators presented in the previous section contain several nested commutators of $A^{(i)}$. Computation of the exponential is the costly part of the algorithm. One can design the schemes up to same order which involve the product of exponentials of linear combination of the $A^{(i)}$ instead of nested commutators. These schemes avoid the presence of commutators and preserve the same qualitative properties of the system [18, 22]. One can consider the compositions of the form

$$\Psi_s^{[n]} \equiv \exp\left(\int_{t_n}^{t_{n+h}} r_s(\tau)A(\tau)d\tau\right)\cdots\exp\left(\int_{t_n}^{t_{n+h}} r_1(\tau)A(\tau)d\tau\right)$$

where $\Psi_s^{[n]} = e^{\Omega(t_n+h)} + \mathcal{O}(h^{n+1})$ is satisfied with the appropriate choices of the scalar functions $r_i(\tau)$. On the other hand, one can use some quadrature rules to define coefficients $\rho_{i,j}$, $i = 1, \ldots, s, j = 1, \ldots, k$ instead of functions $r_i(\tau)$ such that

$$\Psi_s^{[n]} \equiv e^{\tilde{A}_s}\cdots e^{\tilde{A}_1},$$

where $\tilde{A}_i = h\sum_{j=1}^{k}\rho_{i,j}A_j$. For simplicity, it is possible to work with the Lie algebra generated by the α_i. For example, one can obtain 4th and 6th order integrators by solving the equations

$$\Psi_s^{[4]} \equiv \prod_{i=1}^{s}\exp\left(\sum_{k=1}^{2} x_{i,k}\,\alpha_k\right) = \exp\left(\alpha_1 - \frac{1}{12}[12]\right) + \mathcal{O}(h^5), \tag{2.4.30}$$

$$\begin{aligned}\Psi_s^{[6]} \equiv \prod_{i=1}^{s}\exp\left(\sum_{k=1}^{3} x_{i,k}\,\alpha_k\right) &= \exp\Big(\alpha_1 + \frac{1}{12}\alpha_3 - \frac{1}{12}[12] + \frac{1}{240}[23] \\ &+ \frac{1}{360}[113] - \frac{1}{240}[212] + \frac{1}{720}[1112]\Big) + \mathcal{O}(h^7).\end{aligned} \tag{2.4.31}$$

One can obtain the order conditions for the coefficients $x_{i,k}$ by applying Baker-Campbell-Hausdorff formula to the left hand side of the equations (2.4.30) and (2.4.31). As already mentioned before, the Magnus Expansion is time-symmetric and these properties should be also preserved by the integrators $\Psi_s^{[n]}$. If one imposes the symmetry condition as

$$x_{s+1-i,j} = (-1)^{j+1} x_{i,j},$$

then the integrators $\Psi_s^{[n]}$ become time-symmetric. Some examples are the fourth order commutator free methods with two and three exponentials given in [22]

$$\Psi_2^{[4]} \equiv \exp\Big(\frac{1}{2}\alpha_1 + \frac{1}{6}\alpha_2\Big)\exp\Big(\frac{1}{2}\alpha_1 - \frac{1}{6}\alpha_2\Big), \tag{2.4.32}$$

$$\Psi_3^{[4]} \equiv \exp\Big(\frac{1}{12}\alpha_2\Big)\exp\Big(\alpha_1\Big)\exp\Big(-\frac{1}{12}\alpha_2\Big). \tag{2.4.33}$$

If one considers fourth order Gauss-Legendre quadrature rule (2.4.21) and the relation (2.4.28) to define α_1, α_2 in (2.4.32) and (2.4.33) then one obtains the schemes as

$$\Psi_2^{[4]} \equiv \exp\Big(h\rho_{2,1}A_1 + \rho_{2,2}A_2\Big)\exp\Big(h\rho_{1,1}A_1 + \rho_{1,2}A_2\Big),$$

$$\Psi_3^{[4]} \equiv \exp\Big(\frac{\sqrt{3}}{12}h(A_2 - A_1)\Big)\exp\Big(\frac{h}{2}(A_1 + A_2)\Big)\exp\Big(-\frac{\sqrt{3}}{12}h(A_2 - A_1)\Big),$$

with $\rho_{2,1} = \rho_{1,2} = \frac{3-2\sqrt{3}}{12}$ and $\rho_{2,2} = \rho_{1,1} = \frac{3+2\sqrt{3}}{12}$.

2.4.7 Different time-averaging

In this section we consider different techniques presented in [21] which allow one to design splitting methods for separable non-autonomous problems. Splittings methods have been usually designed for autonomous separable systems. However, on the non-autonomous problems, these methods can show poor performance and lead to lower overall order of accuracy against the corresponding autonomous cases. The techniques presented in this section allow to reach the high performance of the splitting methods using the same schemes as for the autonomous problems. Let us consider a non-autonomous separable problem in the form

$$x' = \frac{dx}{dt} = F(x,t), \qquad x(0) = x_0 \in \mathbb{R}^d, \tag{2.4.34}$$

which is separable into two parts

$$\frac{d}{dt}x = F^{[A]}(x,t), \qquad \frac{d}{dt}x = F^{[B]}(x,t). \tag{2.4.35}$$

There are two simple procedures presented in [21] which allow to use splitting method (2.2.5). The first procedure is given by replacing the maps $\varphi_{a_i h}^{[A]}, \varphi_{b_i h}^{[B]}$ by the maps associated to the exact flow of the equations

$$x' = F^{[A]}(x,t), \qquad t \in [t_0 + c_i h, t_0 + (c_i + a_i)h], \tag{2.4.36}$$

$$x' = F^{[B]}(x,t), \qquad t \in [t_0 + d_i h, t_0 + (d_i + b_i)h] \tag{2.4.37}$$

where $c_i = \sum_{j=0}^{i-1} a_j$, $d_i = \sum_{j=0}^{i-1} b_j$, $a_0 = 0$, $b_0 = 0$, and the initial conditions are considered as the solution obtained from the previous flow. This technique can be considered as a time-average on each stage of the composition. It is not always possible to find the exact solution of the non-autonomous equations (2.4.36) and (2.4.37) due to explicit time dependence. In particular, their formal solutions can be written by using the Magnus series expansion for nonlinear differential equations [18]. Another simple procedure called the "frozen" technique, in which the maps $\varphi^{[A]}_{a_i h}$, $\varphi^{[B]}_{b_i h}$ are considered to be $(a_i h)$-flow and $(b_i h)$-flow to the corresponding autonomous vector fields

$$\begin{aligned} x' &= F^{[A]}(x, t_0 + d_i h), \qquad t \in [t_0 + c_i h, t_0 + (c_i + a_i)h], \\ x' &= F^{[B]}(x, t_0 + c_i h), \qquad t \in [t_0 + d_i h, t_0 + (d_i + b_i)h]. \end{aligned}$$

The advantage of both techniques is to transform the non-autonomous equation into the autonomous equation by introduction of a new variable. If one considers t as the dependent variable, one transforms the general equation (2.4.34) into an autonomous form as follows:

$$\begin{cases} x' = F(x, t_1) \\ t_1' = 1 \end{cases}$$

or, equivalently,

$$z' = f(z), \qquad z(0) = (x_0, 0) \in \mathbb{R}^{d+1},$$

and $f(z) = (F(x, t_1), 1)$. Following this procedure with taking time as two independent coordinates and splitting the vector field in the extended system, the averaging technique reads

$$\frac{d}{dt}\begin{pmatrix} x \\ t_1 \\ t_2 \end{pmatrix} = \begin{pmatrix} F^{[A]}(x, t_2) \\ 0 \\ 1 \end{pmatrix} + \begin{pmatrix} F^{[B]}(x, t_1) \\ 1 \\ 0 \end{pmatrix}.$$

Now the new extended system is autonomous and separable into solvable parts provided that subproblems (2.4.35) have exact solutions or can be solved numerically by an efficient method. On the other hand, another split of the vector field in the extended autonomous system can be obtained by the freezing technique

$$\frac{d}{dt}\begin{pmatrix} x \\ t_1 \\ t_2 \end{pmatrix} = \begin{pmatrix} F^{[A]}(x, t_2) \\ 1 \\ 0 \end{pmatrix} + \begin{pmatrix} F^{[B]}(x, t_1) \\ 0 \\ 1 \end{pmatrix},$$

which is separable in solvable parts with the assumption that subproblems (2.4.35) are solvable when the time dependency in $F^{[A]}$ and $F^{[B]}$ is frozen [21]. If the vector fields present several time-dependent time scales, say $F^{[A]}(x, \epsilon_1 t, \dots, \epsilon_k t)$, $F^{[B]}(x, \varepsilon_1 t, \dots, \varepsilon_m t)$, then one can generalize above splittings by introducing new coordinates for the times, $\epsilon_1 t, \dots, \epsilon_k t$, $\varepsilon_1 t, \dots, \varepsilon_m t$ and treat them as in any of the previous way. However, the vector fields can be considered with only two different time-dependent sources, $F^{[A]}(x, \epsilon_1 t, \epsilon_2 t)$, $F^{[B]}(x, \varepsilon_1 t, \varepsilon_2 t)$ without losing generality. For simplicity in the presentation, one can consider one of the time dependencies in each time dependent vector field factorized in as

$$F^{[A]}(x, t) = A(t) f^{[A]}(x, t), \qquad F^{[B]}(x, t) = B(t) f^{[B]}(x, t),$$

where $A(t) \in \mathbb{R}^{d\times m_1}, B(t) \in \mathbb{R}^{d\times m_2}$ are assumed to be exactly integrable (or cheaply approximated) and $f^{[A]} : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^{m_1}, f^{[B]} : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^{m_2}$. One can obtain efficient numerical schemes by considering the time-dependent functions in $f^{[A]}(x,t), f^{[B]}(x,t)$ are always frozen and $A(t)$, $B(t)$ can be either frozen or exactly solved in the corresponding non-autonomous equations. If one considers to split the forcing term in as

$$F^{[A_1]}(z) = \begin{pmatrix} A(t_2)f^{[A]}(x,t_2) \\ 1 \\ 0 \end{pmatrix}, \qquad F^{[B_1]}(z) = \begin{pmatrix} B(t_1)f^{[B]}(x,t_1) \\ 0 \\ 1 \end{pmatrix},$$

where $z = (x, t_1, t_2)^T$, then one obtains the equivalent autonomous problem $z' = F^{[A_1]}(z) + F^{[B_1]}(z)$. Let consider the exact flows of each subproblems $z' = F^{[A_1]}(z)$ and $z' = F^{[B_1]}(z)$ as $\varphi_t^{[A_1]}, \varphi_t^{[B_1]}$, respectively. The approximate solution can be obtained by the composition

$$\psi_h^{[1]} = \varphi_{a_{s+1}h}^{[A_1]} \circ \varphi_{b_{s+1}h}^{[B_1]} \circ \varphi_{a_sh}^{[A_1]} \circ \varphi_{b_sh}^{[B_1]} \circ \varphi_{a_{s-1}h}^{[A_1]} \circ \ldots \circ \varphi_{b_2h}^{[B_1]} \circ \varphi_{a_1h}^{[A_1]} \circ \varphi_{b_1h}^{[B_1]},$$

in which the time proceeds with the step sizes a_ih, b_ih, $(i = 1, 2, \ldots, s+1)$ and the time variable is kept frozen in both forcing terms. One can also design different numerical schemes by taking into account that $A(t)$ or $B(t)$ can be exactly solvable. If one considers $B(t)$ is exactly solvable and split the forcing term as

$$F^{[A_2]}(z) = \begin{pmatrix} A(t_2)f^{[A]}(x,t_2) \\ 1 \\ 0 \end{pmatrix}, \qquad F^{[B_2]}(z) = \begin{pmatrix} B(t_2)f^{[B]}(x,t_1) \\ 0 \\ 1 \end{pmatrix},$$

then one obtains the equivalent autonomous system $z' = F^{[A_2]}(z) + F^{[B_2]}(z)$. In this case, one can design numerical scheme as follows:

$$\psi_h^{[2]} = \varphi_{a_{s+1}h}^{[A_2]} \circ \varphi_{b_{s+1}h}^{[B_2]} \circ \varphi_{a_sh}^{[A_2]} \circ \varphi_{b_sh}^{[B_2]} \circ \varphi_{a_{s-1}h}^{[A_2]} \circ \ldots \circ \varphi_{b_2h}^{[B_2]} \circ \varphi_{a_1h}^{[A_2]} \circ \varphi_{b_1h}^{[B_2]},$$

Here the term $B(t)$ advances in each step with $\varphi_{b_ih}^{[B_2]}$, which is the (b_ih)-flow of $z' = F^{[B_i]}(z)$ or, equivalently, the exact solution of $x' = B(t)f^{[B]}(x,t_1)$ for a time step (b_ih) and frozen time t_1. However, as already mentioned, the formal solution of this problem can be represented as a Magnus series expansion for nonlinear differential equations [18]. The flow $\varphi_{a_ih}^{[A_2]}$ can be computed for the frozen time t_2 in the forcing term $F^{[B_2]}(z)$. Conversely, one can assume that $A(t)$ is exactly integrable. In this case, the forcing term can be split as

$$F^{[A_3]}(z) = \begin{pmatrix} A(t_1)f^{[A]}(x,t_2) \\ 1 \\ 0 \end{pmatrix}, \qquad F^{[B_3]}(z) = \begin{pmatrix} B(t_1)f^{[B]}(x,t_1) \\ 0 \\ 1 \end{pmatrix},$$

in order to obtain equivalent system $z' = F^{[A_3]}(z) + F^{[B_3]}(z)$. The numerical scheme can be obtained by the composition of exact flows $\varphi_{a_ih}^{[A_3]}, \varphi_{b_ih}^{[B_3]}$ in the form:

$$\psi_h^{[3]} = \varphi_{a_{s+1}h}^{[A_3]} \circ \varphi_{b_{s+1}h}^{[B_3]} \circ \varphi_{a_sh}^{[A_3]} \circ \varphi_{b_sh}^{[B_3]} \circ \varphi_{a_{s-1}h}^{[A_3]} \circ \ldots \circ \varphi_{b_2h}^{[B_3]} \circ \varphi_{a_1h}^{[A_3]} \circ \varphi_{b_1h}^{[B_3]}.$$

Notice that during integration, the time variable is frozen at the first step for all terms and taken into account for time evolution in the alternating integration. Furthermore, the system

$z' = F^{[A_4]}(z) + F^{[B_4]}(z)$ where

$$F^{[A_4]}(z) = \begin{pmatrix} A(t_1)f^{[A]}(x,t_2) \\ 1 \\ 0 \end{pmatrix}, \qquad F^{[B_4]}(z) = \begin{pmatrix} B(t_2)f^{[B]}(x,t_1) \\ 0 \\ 1 \end{pmatrix},$$

can be considered if both $A(t)$ and $B(t)$ are exactly integrable. The corresponding numerical scheme reads

$$\psi_h^{[4]} = \varphi_{a_{s+1}h}^{[A_4]} \circ \varphi_{b_{s+1}h}^{[B_4]} \circ \varphi_{a_s h}^{[A_4]} \circ \varphi_{b_s h}^{[B_4]} \circ \varphi_{a_{s-1}h}^{[A_4]} \circ \ldots \circ \varphi_{b_2 h}^{[B_4]} \circ \varphi_{a_1 h}^{[A_4]} \circ \varphi_{b_1 h}^{[B_4]}, \qquad (2.4.38)$$

in which the vector field associated with $F^{[B_4]}(z)$ is first integrated where the term $B(t)$ is explicitly time dependent and then $A(t)$ being allowed to advance in time in the second integration associated with $F^{[A_4]}(z)$. Thus, the vector fields are fully integrated with respect to time in the scheme (2.4.38).

Near-seperable systems : Let us consider an example of the time-dependent system being separable with respect to the q and p in the form:

$$q' = A(t)f_2(p,t), \qquad p' = B(t)f_1(q,t), \qquad (2.4.39)$$

where $q \in \mathbb{R}^m, p \in \mathbb{R}^{d-m}$. One can rewrite the system as

$$\frac{d}{dt}\begin{pmatrix} q \\ p \end{pmatrix} = \begin{pmatrix} A(t)f_2(p,t) \\ 0 \end{pmatrix}, \quad \frac{d}{dt}\begin{pmatrix} q \\ p \end{pmatrix} = \begin{pmatrix} 0 \\ B(t)f_1(p,t) \end{pmatrix}.$$

The subproblems given in (2.4.39) are solvable and their solutions read

$$\begin{cases} q(t) = q_0 + \int_{t_0}^{t} A(\tau)f_2(p_0,\tau)d\tau \\ \\ p(t) = p_0 \end{cases} \quad \text{and} \quad \begin{cases} q(t) = q_0 \\ \\ p(t) = p_0 + \int_{t_0}^{t} B(\tau)f_1(q_0,\tau)d\tau. \end{cases}$$

In this presentation it has been assumed the time variable is frozen in both terms $f_1(q,t)$,$f_2(q,t)$. The algorithm of the first scheme $\psi_h^{[1]}$ for one time step $t_n \to t_n + h$ can be shown by

$$\begin{aligned} &t_0^1 := t_n, \quad t_0^2 := t_n, \quad Q_0 := q_n, \quad P_0 := p_n, \\ &\textbf{do } \ k = 1, \ldots, s+1 \\ &\quad P_k := P_{k-1} + b_k h B(t_{k-1}^1) f_1(P_{k-1}, t_{k-1}^1) \\ &\quad t_k^2 := t_{k-1}^2 + b_k h \\ &\quad Q_k := Q_{k-1} + a_k h A(t_k^2) f_2(Q_k, t_k^2) \\ &\quad t_k^1 := t_{k-1}^1 + a_k h \\ &\textbf{enddo} \\ &\quad t_{n+1} := t_{s+1}^1 := t_{s+1}^1 := t_n + h, \qquad q_{n+1} := Q_{s+1}, \qquad p_{n+1} := P_{s+1}. \end{aligned} \qquad (2.4.40)$$

where the time in both forcing terms kept frozen. The algorithm of the scheme $\psi_h^{[2]}$ which is assumed that $B(t)$ has exact integral can be obtained by changing P_i in (2.4.40) with

$$P_i = P_{i-1} + \int_{t_{k-1}^2}^{t_k^2} B(\tau)f_1(Q_{k-1}, t_{k-1}^1)d\tau. \qquad (2.4.41)$$

If one considers that $A(t)$ to be exactly solvable, then can write the algorithm of the scheme $\psi_h^{[3]}$ by replacing Q_i in (2.4.40) with

$$Q_i = Q_{i-1} + \int_{t_{k-1}^1}^{t_k^1} A(\tau) f_2(P_k, t_k^2) d\tau. \tag{2.4.42}$$

The last scheme $\psi_h^{[4]}$ is considered such that both $A(t)$ and $B(t)$ have exact intengral. In this case one can write the algorithm by replacing both P_i and Q_i in (2.4.40) with the (2.4.41) and (2.4.42) respectively. The interested reader is referred to [21] for further information.

Chapter 3

NON-REVERSIBLE SYSTEMS

Splitting methods with real coefficients for the numerical integration of differential equations necessarily have some negative coefficients a_i,b_i in order to achieve order three or more [13, 46, 92, 96]. The methods which involve such a stepping backwards in time cannot be used for the evolution equation in the context of semigroups, like parabolic (diffusion) equations. The semigroups are in general not well defined for the time $t < 0$. However, splitting methods with complex coefficients with positive real part exist, and some of these methods can provide a high performance in spite of having to solve the equations in the complex domain. Previous works with applications among other in celestial mechanics and quantum mechanics where splitting methods with complex coefficients are considered already exist [10, 11, 38, 70, 83, 95, 96, 97].

3.1 The separable non-autonomous parabolic equations

The exposition is based on the article [90].

We consider the numerical integration of non-autonomous separable parabolic equations using high order splitting methods with complex coefficients. This class of methods has been recently used for the numerical integration of the autonomous case, showing good performances [14, 34, 49].

A straightforward application of splitting methods with complex coefficients to non-autonomous problems require the evaluation of the time-dependent functions on the operators at complex times, and the corresponding flows in the numerical scheme are, in general, not well conditioned. In this chapter, we propose to consider a class of splitting methods in which one set of the coefficients belong to the class of real and positive numbers. This can allow to evaluate all time-dependent operators at real values of the time, leading to schemes which are stable and simple to implement.

If the system can be considered as the perturbation of an exactly solvable problem (or easy to numerically solve) and the flow of the dominant part is advanced using the real coefficients, it is possible to build highly efficient methods for these problems.

3.1.1 The problem

Let us consider the non-autonomous separable PDE

$$\frac{du}{dt} = A(t,u) + B(t,u), \qquad u(0) = u_0, \tag{3.1.1}$$

$u(x,t) \in \mathbb{R}^D$, and where the (possibly unbounded) operators A, B and $A+B$ generate C^0 semi-groups for positive t over a finite or infinite Banach space. Equations of this form are encountered in the context of *parabolic* partial differential equations, an example being the inhomogeneous non-autonomous *heat equation*

$$\frac{\partial u}{\partial t} = \alpha(t)\Delta u + V(x,t)u, \quad \text{or} \quad \frac{\partial u}{\partial t} = \nabla(a(x,t)\nabla u) + V(x,t)u$$

where $t \geq 0$, $x \in \mathbb{R}^d$ or $x \in \mathbb{T}^d$ and Δ denotes the Laplacian with respect to the spatial coordinates, x. Another example corresponds to reaction-diffusion equations of the form

$$\frac{\partial u}{\partial t} = D(t)\Delta u + B(t,u),$$

where $D(t)$ is a matrix of diffusion coefficients (typically a diagonal matrix) and B accounts for the reaction part. In general, $A(t,u), B(t,u)$ can also depend on x, ∇, etc., which are omitted for clarity in the presentation.

For simplicity, we write the non-linear equation (3.1.1) in the (apparently) linear form

$$\frac{du}{dt} = L_{A(t,u)}u + L_{B(t,u)}u,$$

where L_A, L_B are the Lie operators associated to A, B, i.e.

$$L_{A(t,u)} \equiv A(t,u)\frac{\partial}{\partial u}, \qquad L_{B(t,u)} \equiv B(t,u)\frac{\partial}{\partial u}$$

which act on functions of u

If the problem is autonomous, the formal solution is given by $u(t) = e^{t(L_{A(u)}+L_{B(u)})}u_0$, which is a short way to write

$$u(t) = e^{t(L_{A(u)}+L_{B(u)})}u_0 = \sum_{k=0}^{\infty}\frac{t^k}{k!}\left(A(u)\frac{\partial}{\partial u} + B(u)\frac{\partial}{\partial u}\right)^k u\Bigg|_{u=u_0}.$$

If the subproblems

$$\frac{du}{dt} = A(u) \qquad \text{and} \qquad \frac{du}{dt} = B(u) \tag{3.1.2}$$

have exact solutions or can efficiently be numerically solved, it is usual to consider splitting methods as numerical integrators. If we denote by $e^{hL_{A(u)}}$, $e^{hL_{B(u)}}$ the exact h-flows for each

problem in (3.1.2) (and for a sufficiently small time step, h) the simplest method within this class is the *Lie-Trotter splitting*

$$e^{hL_{A(u)}}\, e^{hL_{B(u)}} \qquad \text{or} \qquad e^{hL_{B(u)}}\, e^{hL_{A(u)}},$$

which is a first order approximation in the time step to the solution, while the *symmetrized* version

$$S(h) = e^{h/2\, L_{A(u)}}\, e^{hL_{B(u)}}\, e^{h/2\, L_{A(u)}} \qquad \text{or} \qquad S(h) = e^{h/2\, L_{B(u)}}\, e^{hL_{A(u)}}\, e^{h/2\, L_{B(u)}}$$

is referred to as *Strang splitting*, and is an approximation of order 2, i.e. $S(h) = e^{h(L_{A(u)+B(u)})} + \mathcal{O}(h^3)$. Upon using an appropriate sequence of steps, high-order approximations can be obtained as

$$\Psi(h) = e^{hb_1 L_B}\, e^{ha_1 L_A} \ldots e^{hb_s L_B}\, e^{ha_s L_A}\, e^{hb_{s+1} L_B}, \tag{3.1.3}$$

and methods with real coefficients at any order can be obtained [41, 95, 106]. However, as already mentioned, splitting methods of order greater than two (with real coefficients) have at least one of the coefficients a_i negative as well as at least one of the coefficients b_i so, the flows e^{tL_A} and/or e^{tL_B} may not be well defined (this is indeed the case, for instance, for the Laplacian operator) and this prevents the use of methods which embed negative coefficients. For this reason, exponential splitting methods of at most order $p = 2$ have been considered up to recently.

In order to circumvent this order-barrier, the papers [34] and [49] simultaneously presented a systematic analysis for a class of composition methods with complex coefficients having positive real parts. Using this extension from the real line to the complex plane, the authors of [34] and [49] built up methods of orders 3 to 14 by considering a technique known as *triple-jump composition*. More efficient high order methods are obtained in [14].

In this work we are interested, however, in the numerical integration of the non-autonomous problem (3.1.1) where the use of complex coefficients involve additional constraints as we will see. A method of choice for solving numerically (3.1.1) consists in advancing the solution alternatively along the exact (or numerical) solutions of the two problems

$$\frac{du}{dt} = A(t,u) \qquad \text{and} \qquad \frac{du}{dt} = B(t,u). \tag{3.1.4}$$

The exact flows are, in general, not known. This is the case, for example, if $[L_{A(t_i,u)}, L_{A(t_j,u)}] = L_{A(t_i,u)}L_{A(t_j,u)} - L_{A(t_j,u)}L_{A(t_i,u)} \neq 0$ (and similarly for $B(t,u)$). If the exact solution is not known, it can be replaced by a sufficiently accurate numerical approximation.

This procedure is equivalent to take the time as two new coordinates, t_1, t_2

$$\begin{cases} u' = A(t_1,u) + B(t_2,u) \\ t_1' = 1 \\ t_2' = 1, \end{cases}$$

with $' \equiv \frac{d}{dt}$, and to split the system in the extended space as follows [21]

$$\begin{cases} u' = A(t_1,u) \\ t_1' = 1 \\ t_2' = 0 \end{cases} \qquad \text{and} \qquad \begin{cases} u' = B(t_2,u) \\ t_1' = 0 \\ t_2' = 1. \end{cases}$$

A more convenient way to split the system which transforms the non-autonomous problems into autonomous is the following

$$\left\{\begin{array}{l} u' = A(t_1, u) \\ t_1' = 0 \\ t_2' = 1 \end{array}\right. \qquad \text{and} \qquad \left\{\begin{array}{l} u' = B(t_2, u) \\ t_1' = 1 \\ t_2' = 0. \end{array}\right. \tag{3.1.5}$$

Notice that the explicit time-dependency in A and B is frozen in each subproblem and the formal solution corresponds to the exponential of the Lie operators where the time dependency in the operators are frozen on each time interval. Unfortunately, to use splitting methods with complex coefficients for non-autonomous problems requires, in general, to compute $A(t,u), B(t,u)$ for $t \in \mathbb{C}$, leading, in general, to badly conditioned algorithms.

In this work we show that splitting method having one set of coefficients real and positive valued, i.e.

$$a_i \in \mathbb{R}^+, \qquad b_i \in \mathbb{C}^+, \qquad (\text{or} \quad a_i \in \mathbb{C}^+, \qquad b_i \in \mathbb{R}^+),$$

allow to build algorithms where the operators $A(t,u), B(t,u)$ are evaluated only for $t \in \mathbb{R}$, leading to well defined methods. Several splitting methods with this structure have already been constructed[1].

We will also explore the case in which $\|B\| \ll \|A\|$, which we refer as a perturbed problem. We first show how this class of methods has to be used in these problems and next we study how to build high order efficient methods for these problems.

3.2 Splitting methods for non-autonomous problems

Suppose we have a splitting method with say, $a_i \in \mathbb{R}^+$ and $b_i \in \mathbb{C}^+$. To solve the eq. (3.1.4) we propose to take the time as one new coordinate and split the system as follows [21]

$$\left\{\begin{array}{l} u' = A(t_1, u) \\ t_1' = 1 \end{array}\right. \qquad \text{and} \qquad \left\{\begin{array}{l} u' = B(t_1, u) \\ t_1' = 0. \end{array}\right.$$

Let us denote by $\Phi_A^{[a_i h]}$ the map associated to the exact solution (or a sufficiently accurate numerical approximation) of the non-autonomous equation

$$\frac{du}{dt} = A(t,u), \qquad t \in [t_n + c_{i-1}h, t_n + c_i h]$$

with

$$c_i = \sum_{j=0}^{i} a_j,$$

and $a_0 = 0$. Then, the splitting method (3.1.3) for the non-autonomous equation reads now

$$\Psi(h) = e^{hb_1 L_{B_m}} \Phi_A^{[a_1 h]} \cdots e^{hb_s L_{B_1}} \Phi_A^{[a_s h]} e^{hb_{s+1} L_{B_0}},$$

[1] In [34], a fourth-order method was obtained with $a_i \in \mathbb{R}^+$. In a similar way, in [14] sixth-order schemes were also explored with $a_i \in \mathbb{R}^+$. The coefficients can be found at: `http://www.gicas.uji.es/Research/splitting-complex.html`.

where $B_i = B(t_n + c_i h, u)$. Notice that in this scheme, since t_1 is advanced with the coefficients a_i (and then it takes real values) the operators $A(t,u)$ and $B(t,u)$ are evaluated on real values of t. On the other hand, if $A(t,u)$ is an unbounded operator and a numerical methods is used to approximate the flow $\Phi_A^{[h]}$, it must be well defined for $0 \le h < h^*$ for some positive h^*, and this is not guaranteed for general methods. For example, some commutator-free methods up to fourth-order can be used. Given the equation

$$\frac{du}{dt} = A(t,u), \qquad\qquad t \in [0,h]$$

we have that

$$\Phi_A^{[h]} = e^{hL_{A(t_n+h/2,u)}}$$

corresponds to a symmetric second order method, and

$$\begin{aligned}\Phi_A^{[h]} &= e^{\frac{h}{2}(\alpha L_{A_1} + \beta L_{A_2})} e^{\frac{h}{2}(\beta L_{A_1(u)} + \alpha L_{A_2(u)})} \\ &= \Phi^{[h]}_{\frac{1}{2}(\beta A_1(u) + \alpha A_2(u))} \circ \Phi^{[h]}_{\frac{1}{2}(\alpha A_1(u) + \beta A_2(u))}\end{aligned} \tag{3.2.1}$$

with $A_1(u) = A\left(t_n + \left(\frac{1}{2} - \frac{\sqrt{3}}{6}\right)h, u\right)$, $A_2(u) = A\left(t_n + \left(\frac{1}{2} + \frac{\sqrt{3}}{6}\right)h, u\right)$ and $\alpha = \frac{1}{2} - \frac{\sqrt{3}}{3}$, $\beta = 1 - \alpha$, corresponds to a fourth-order method [22, 98]. Notice that the Lie operators, since being derivatives, are written in the reverse order than the maps, and this is very important to keep in mind for non-linear non-autonomous problems in order to apply the method correctly (see [24] for more details on the Magnus series expansion and Magnus integrators for non-autonomous non-linear differential equations). This scheme corresponds to the composition of the 1-flow maps for the equations

$$\begin{aligned} u_1' &= \frac{h}{2}(\alpha A_1(u_1) + \beta A_2(u_1)), \qquad & u_1(0) &= u_0 \\ u_2' &= \frac{h}{2}(\beta A_1(u_2) + \alpha A_2(u_2)), \qquad & u_2(0) &= u_1(1). \end{aligned}$$

and the solution given by the map corresponds to $u_2(1)$. These second and fourth-order commutator-free methods can be used for unbounded operators and higher order commutator -free Magnus integrators for unbounded operators are under investigation at this moment.

3.2.1 Splitting methods for non-autonomous perturbed systems

In some cases, the system can be considered as the perturbation of an exactly solvable problem. In those cases, it is usually convenient to split into the dominant part and the perturbation and to build methods which take advantage of this relevant property. However, if the problem is non-autonomous and the time-dependency is not treated properly, the performance of the methods designed for perturbed problems deteriorate considerably.

Suppose that $\|B(t,u)\| \ll \|A(t,u)\|$. To make this fact more evident, we replace B by εB with $|\varepsilon| \ll 1$[2]. In the autonomous case, for example, the Lie-Trotter composition for this split satisfies

$$e^{h(L_{A+\varepsilon B})} = e^{h\,L_A} e^{h\varepsilon L_B} + \frac{1}{2}\varepsilon h^2 [L_A, L_B] + \mathcal{O}(\varepsilon h^3),$$

[2]In most cases, this split is also convenient for not necessarily very small perturbations, say $\varepsilon < 1/2$.

i.e., it has a local error of order $\mathcal{O}(\varepsilon h^2)$.

Since A and B are qualitatively different for perturbed problems, it is usual to consider ABA and BAB compositions.

An s-stage symmetric BAB compositions given by

$$\Psi(h) = e^{hb_1 \varepsilon L_B}\, e^{ha_1 L_A} \,\cdots\, e^{hb_s \varepsilon L_B}\, e^{ha_s L_A}\, e^{hb_{s+1} \varepsilon L_B}, \tag{3.2.2}$$

with $a_{s+1-i} = a_i,\ b_{s+2-1} = b_i,\ i = 1, 2, \ldots$, and ABA compositions are given by

$$\Psi(h) = e^{ha_1 L_A}\, e^{hb_1 \varepsilon L_B} \,\cdots\, e^{ha_s L_A}\, e^{hb_s \varepsilon L_B}\, e^{ha_{s+1} L_A}, \tag{3.2.3}$$

with $a_{s+2-i} = a_i,\ b_{s+1-1} = b_i,\ i = 1, 2, \ldots$. We will use the following short notation for these methods

$$(b_1, a_1, \cdots, b_s, a_s, b_{s+1}) \qquad \text{and} \qquad (a_1, b_1, \cdots, a_s, b_s, a_{s+1}).$$

Notice that eq. (3.2.2) is a BAB composition which, for the particular case where $b_1 = b_{s+1} = 0$ transforms into a ABA composition (but with a different computational cost). It seems then natural to consider separately the following four cases:

1. $BAB:\quad a_i \in \mathbb{R}^+, \quad b_i \in \mathbb{C}^+,$
2. $BAB:\quad a_i \in \mathbb{C}^+, \quad b_i \in \mathbb{R}^+,$
3. $ABA:\quad a_i \in \mathbb{R}^+, \quad b_i \in \mathbb{C}^+,$
4. $ABA:\quad a_i \in \mathbb{C}^+, \quad b_i \in \mathbb{R}^+.$

The cases 2 and 4 require to split the system as follows

$$\left\{ \begin{array}{l} u' = A(t_1, u) \\ t_1' = 0 \end{array} \right. \qquad \text{and} \qquad \left\{ \begin{array}{l} u' = \varepsilon B(t_1, u) \\ t_1' = 1, \end{array} \right.$$

so t_1 will take real values. This split is similar to the one shown in the previous section by changing the roles of A and B.

In the extended phase space these two systems are equivalent to solve separately the following system written in terms of Lie operators

$$\frac{d}{dt} \left\{ \begin{array}{c} u \\ t_1 \end{array} \right\} = \underbrace{\left(A(t_1, u) \frac{\partial}{\partial u} + 0 \cdot \frac{\partial}{\partial t_1} \right)}_{\mathscr{A}} \left\{ \begin{array}{c} u \\ t_1 \end{array} \right\}$$

$$\frac{d}{dt} \left\{ \begin{array}{c} u \\ t_1 \end{array} \right\} = \underbrace{\left(\varepsilon B(t_1, u) \frac{\partial}{\partial u} + 1 \cdot \frac{\partial}{\partial t_1} \right)}_{\mathscr{B}} \left\{ \begin{array}{c} u \\ t_1 \end{array} \right\}.$$

The commutators of the Lie operators $\mathscr{A}$ and $\mathscr{B}$, which measure the error of the splitting methods in the extended phase space is

$$[h\mathscr{A}, h\mathscr{B}] = h^2(\mathscr{A}\mathscr{B} - \mathscr{B}\mathscr{A}) = h^2\left(\varepsilon[A,B] - \frac{dA(t_1,u)}{dt_1}\right)\frac{\partial}{\partial u} = \mathcal{O}(h^2)$$

which is not proportional to ε due to the term $\frac{dA(t_1,u)}{dt_1}$, and this also happens with higher order commutators.

The cases 1 and 3 are associated to the split

$$\left\{\begin{array}{l} u' = A(t_1,u) \\ t_1' = 1 \end{array}\right. \qquad \text{and} \qquad \left\{\begin{array}{l} u' = \varepsilon B(t_1,u) \\ t_1' = 0. \end{array}\right. \tag{3.2.4}$$

This system can be written in the extended phase space as

$$\frac{d}{dt}\left\{\begin{array}{c} u \\ t_1 \end{array}\right\} = \underbrace{\left(A(t_1,u)\frac{\partial}{\partial u} + 1\cdot\frac{\partial}{\partial t_1}\right)}_{\mathscr{A}}\left\{\begin{array}{c} u \\ t_1 \end{array}\right\},$$

$$\frac{d}{dt}\left\{\begin{array}{c} u \\ t_1 \end{array}\right\} = \underbrace{\left(\varepsilon B(t_1,u)\frac{\partial}{\partial u} + 0\cdot\frac{\partial}{\partial t_1}\right)}_{\mathscr{B}}\left\{\begin{array}{c} u \\ t_1 \end{array}\right\},$$

where now

$$[h\mathscr{A}, h\mathscr{B}] = \varepsilon h^2\left([A,B] + \frac{dB(t_1,u)}{dt_1}\right)\frac{\partial}{\partial u} = \mathcal{O}(\varepsilon h^2)$$

which is proportional to the small parameter ε (see [21] for more details).

Obviously, this split makes sense if one can exactly solve the non-autonomous equation associated to the dominant part

$$\frac{du}{dt} = A(t,u)$$

at a relatively low computational cost (or one can numerically solve it up to sufficiently high accuracy and at a relatively low computational cost) being the commutator-free Magnus integrators an appropriate choice in most cases. A similar methods used in [4] for perturbed Schrödinger and Gross-Pitaevskii equations, but in those problems negative real coefficients are allowed.

3.2.2 Order conditions

For consistent symmetric methods we can formally write

$$\begin{aligned}\Psi(h) = \exp\big(&h(L_A + \varepsilon L_B) + h^3\left(\varepsilon\, p_{aba}[[L_A,L_B],L_A] + \varepsilon^2 p_{abb}[[L_A,L_B],L_B]\right) \\ &+h^5\left(\varepsilon\, p_{abaaa}[[[[L_A,L_B],L_A],L_A],L_A] + \mathcal{O}(\varepsilon^2)\right) + \mathcal{O}(\varepsilon h^7)\big).\end{aligned}$$

Following [15], for the composition (3.2.3) we have that

$$p_{aba} \sim \frac{1}{2}\sum_{i=1}^{s} b_i c_i (1-c_i) - \frac{1}{12}, \qquad p_{abb} \sim \sum_{i=1}^{s} \frac{1}{2} b_i^2 c_i + \sum_{1\le i<j\le s} b_i b_j c_j - \frac{1}{3},$$
$$p_{abaaa} \sim \sum_{i=1}^{s} b_i c_i^4 - \frac{1}{5},$$

with $a_0 = 0$ and $c_{s+1} = 1$. The symbol $\sim$ indicates that, if the low order conditions are satisfied, both terms are proportional so, if the r.h.s of p_{aba} and p_{abb} vanish then $p_{aba} = p_{abb} = 0$. Here, the polynomial p_{abaaa} corresponds to the dominant error term in fourth-order methods for perturbed problem. This algebraic analysis remains also valid for unbounded operators under appropriate conditions on the operators (see [48] for more details).

If we take $a_1 = a_{s+1} = 0$ we obtain a *BAB* composition, and the equations of p_{aba} and p_{abb} can be easily adjusted to obtain *BAB* compositions.

Second order symmetric methods which cancel the terms of order $h^{2p+1}\varepsilon$ for $p = 1, 2, \dots, m$ and for different values of m exist with positive and real coefficients [69]. The error of these methods is of order $\mathcal{O}(h^{2m+1}\varepsilon + h^3\varepsilon^2)$ and we say the methods have effective order $(2m, 2)$. For instance, a method which satisfies $p_{aba} = p_{abaaa} = 0$ has effective order (6,2), and this can be attained with the sequence [69]

$$(b_1, a_1, b_2, a_2, b_2, a_1, b_1) = \left(\frac{1}{12}, \frac{5-\sqrt{5}}{10}, \frac{5}{12}, \frac{1}{\sqrt{5}}, \frac{5}{12}, \frac{5-\sqrt{5}}{10}, \frac{1}{12}\right). \tag{3.2.5}$$

Fourth-order methods require to satisfy $p_{aba} = 0,\ p_{abb} = 0$, and this can not be accomplished with a_i, b_i real and positive valued coefficients. We are then interested on the existence of methods in which $a_i \in \mathbb{R}^+$ and $b_i \in \mathbb{C}^+$. To get splitting methods where the coefficients satisfy these constraints we fix the values of the coefficients $a_i \in (0, 1)$ such that consistency and symmetry is satisfied, and leave the coefficients b_i to solve the order conditions. Obviously, since the coefficients are chosen real and positive, the equations only admit complex solutions for the coefficients b_i. Among all solutions obtained we will choose solutions with positive real part, i.e. $b_i \in \mathbb{C}^+$ from the set of all solutions found (in case these solutions exist).

Let us now analyse the number of free parameters and computational cost of *ABA* and *BAB* compositions in order to choose the most appropriate sequence: A symmetric $(2k)$-stage *BAB* composition has k coefficients a_i and $k + 1$ coefficients b_i while an *ABA* sequence has $k + 1$ coefficients a_i and k coefficients b_i so, the *BAB* composition has one more free parameter to solve the equations. In addition, since the dominant part is associated to the coefficients a_i and requires the numerical solution of a non-autonomous differential equation, it is not usually possible to concatenate the last map in one step with the first one in the following step, and in practice an *ABA* composition with the same number of stages as a *BAB* composition can be computationally more costly up to one additional stage. For these reasons (number of free parameters to solve the equations and the computational cost) we only consider in this work *BAB* compositions.

3.2.3 Fourth-order methods

Fourth-order methods can be obtained with a 4-stage composition

$$(b_1\, a_1\, b_2\, a_2\, b_3\, a_2\, b_2\, a_1\, b_1)$$

which satisfy the consistency conditions $a_1 + a_2 = 1/2,\ \ 2(b_1 + b_2) + b_3 = 1$. We can fix the values of a_1 such that $a_1 \in (0, 1/2)$ and take, e.g. b_1, b_2 to solve the equations $p_{aba} = p_{abb} = 0$. The choice $a_1 = \frac{1}{4}$ leads to the solution obtained in [34]. However, we can take a_1 as a free parameter to minimise the dominant error term[3]

$$\min_{a_1 \in (0,1/2)} |Re(p_{abaaa})| = \min_{a_1 \in (0,1/2)} \left| \sum_{i=1}^{4} Re(b_i)\, c_i^4 - \frac{1}{5} \right|.$$

The corresponding system of polynomial equations with two unknowns have only two solutions (complex conjugate to each other) for each choice of a_1 and with this process we obtain following method:

$$\begin{aligned}
b_1 &= 0.018329102861074364 - 0.10677008344599524i,\\
a_1 &= 0.13505265889288437,\\
b_2 &= 0.2784394345454581 + 0.20041452008768607i,\\
a_2 &= 0.36494734110711563,\\
b_3 &= 0.40646292518693505 - 0.18728887328338165i.
\end{aligned} \tag{3.2.6}$$

A 5-stage *BAB* composition has the same number of coefficient b_i and for this reason we have not considered it. To vanish the dominant error term at order 6 we need at least a 6-stage composition

$$(b_1\, a_1\, b_2\, a_2\, b_3\, a_3\, b_4\, a_3\, b_3\, a_2\, b_2\, a_1\, b_1)$$

where the coefficients b_i are used to satisfy the conditions $p_{aba} = p_{abb} = p_{abaaa} = 0.$ in addition to consistency.

The goal of this work is not to make an exhaustive search of methods but to show this class of methods are of interest for non-autonomous problems and to indicate how highly efficient methods could be obtained, and the optimal method can depend on the algebraic structure of each problem[4]. Then, just as an illustration we take $a_1 = a_2 = a_3 = \frac{1}{6}$. We have obtained one complex solution (and its complex conjugate) with coefficients:

$$\begin{aligned}
a_1 &= a_2 = a_3 = 1/6,\\
b_1 &= 0.05753968253968254 - 0.007886748775536424i,\\
b_2 &= 0.20476190476190473 + 0.04732049265321855i,\\
b_3 &= 0.16309523809523818 - 0.11830123163304637i,\\
b_4 &= 0.14920634920634912 + 0.15773497551072851i.
\end{aligned} \tag{3.2.7}$$

[3] We minimise the real part of the dominant error because after each time step we will remove the imaginary part of the numerical solution, i.e. $u_{n+1} = Re(\Psi(h)u_n)$.

[4] Higher order and more efficient methods require a considerably deeper analysis, and methods belonging to this class as well as more general methods are being considered by the authors of Ref. [14].

3.3 Numerical examples

To analyze the performance of the new methods we first consider a simple non-autonomous ODE as a test bench of the methods and next we apply the methods to a linear non-autonomous PDE and a non-linear non-autonomous PDE. We compare the performance of the methods with complex coefficients versus other methods which involve real coefficients. We choose the (6,2) splitting method (3.2.5) which is a method of second order. As a fourth-order method we consider extrapolation (which involves substraction of quantities) where the Strang splitting symmetric second order method is used as the basic scheme to raise the order. To be more precise, we consider

$$S(h) = e^{h/2\, L_{B_1}}\, e^{hL_{A_0}}\, e^{h/2\, L_{B_0}}$$

where we denote $A_i = A(t_n + ih, u)$, $B_i = B(t_n + ih, u)$. This scheme can be considered as the standard Strang decomposition applied to the non-autonomous system, but if we split it as shown in (3.1.5). If we take $S(h)$ as the basic method, high order methods by extrapolation can be obtained and they only involve positive time steps. A fourth-order method is given by the composition

$$\Phi^{[4]}(h) = \frac{4}{3} S\left(\frac{h}{2}\right) S\left(\frac{h}{2}\right) - \frac{1}{3} S(h)$$

which in our case it can be written as

$$\Phi^{[4]}(h) = \frac{4}{3} e^{\frac{h}{4} L_{B_1}}\, e^{\frac{h}{2} L_{A_{1/2}}}\, e^{\frac{h}{2} L_{B_{1/2}}}\, e^{\frac{h}{2} L_{A_0}}\, e^{\frac{h}{4} L_{B_0}} - \frac{1}{3} e^{\frac{h}{2} L_{B_1}}\, e^{hL_{A_0}}\, e^{\frac{h}{2} L_{B_0}}. \tag{3.3.1}$$

The following schemes with real coefficients are then considered:

- Strang: The second-order symmetric Strang splitting method (as a reference method);
- (6,2): The symmetric splitting method of effective order (6,2) whose coefficients are given in (3.2.5);
- (EXT4): The fourth-order extrapolation method (3.3.1);

and the following schemes with real coefficients and $a_i \in \mathbb{R}^+$ are considered:

- (RC4): The 4-stage fourth-order method from [34];
- (O4): The 4-stage fourth-order method built in [14], whose coefficients are available at `http://www.gicas.uji.es/Research/splitting-complex.html`, and referred as "Order 4 (optimized)";
- (SM4): The new optimized 4-stage fourth-order method given in (3.2.6);
- (SM(6,4)): The new 6-stage fourth-order method whose coefficients are given in (3.2.7);

The numerical approximations u_n obtained by a given method, $\Psi(h)$, which involve complex coefficients are computed as $u_n = \Re(\Psi(h)u_{n-1})$, i.e. we project on the real axis after completing each time step. To measure the performance of the methods we compute the error of

each method at the end of the time integration (we take as the exact solution a numerical approximation computed to a high precision) and we take as the cost of the method the number of evaluations of $\Phi_A^{[h]}$ which usually carries most of the computational cost.

Example 1 Let us consider the non-autonomous and non-linear perturbed equation

$$q'' + \Omega(t)^2 q = -\varepsilon \sum_{j=1}^{s} \sin(q - \omega_j t), \qquad q \in \mathbb{R}.$$

When Ω is a constant, the system describes the motion of a charged particle in a magnetic field perturbed by s electrostatic plane waves, each with the same wavenumber and amplitude, but with different temporal frequencies ω_j [30]. This equation can be written as a first order system of equations

$$\frac{d}{dt}\left\{ \begin{array}{c} q \\ p \end{array} \right\} = \left(\begin{array}{cc} 0 & 1 \\ -\Omega(t)^2 & 0 \end{array} \right) \left\{ \begin{array}{c} q \\ p \end{array} \right\} + \varepsilon \left\{ \begin{array}{c} 0 \\ -\sum_{j=1}^{s} \sin(q - \omega_j t) \end{array} \right\}$$

which we split as follows

$$\frac{d}{dt}\left\{ \begin{array}{c} q \\ p \end{array} \right\} = \left(\begin{array}{cc} 0 & 1 \\ -\Omega(t_1)^2 & 0 \end{array} \right) \left\{ \begin{array}{c} q \\ p \end{array} \right\}, \qquad \frac{dt_1}{dt} = 1$$

and

$$\frac{d}{dt}\left\{ \begin{array}{c} q \\ p \end{array} \right\} = \varepsilon \left\{ \begin{array}{c} 0 \\ -\sum_{j=1}^{s} \sin(q - \omega_j t_1) \end{array} \right\}.$$

The linear part has, in general, no solution in closed form and we approximate its flow using the 4th-order commutator-free Magnus integrator (3.2.1) which for this problem reads[5]

$$\begin{aligned} \Phi_A^{[a_i h]} &= e^{\frac{a_i h}{2}(\beta A_1 + \alpha A_2)} e^{\frac{a_i h}{2}(\alpha A_1 + \beta A_2)} \\ &= \exp\left[\frac{a_i h}{2} \left(\begin{array}{cc} 0 & 1 \\ -(\beta\Omega_1^2 + \alpha\Omega_2^2) & 0 \end{array} \right) \right] \exp\left[\frac{a_i h}{2} \left(\begin{array}{cc} 0 & 1 \\ -(\alpha\Omega_1^2 + \beta\Omega_2^2) & 0 \end{array} \right) \right] \end{aligned}$$

where $\Omega_i = \Omega(t_n + c_i h)$ and the exponential of each matrix can be easily computed taking into account that

$$\exp\left[\tau \left(\begin{array}{cc} 0 & 1 \\ -\Omega^2 & 0 \end{array} \right) \right] = \left(\begin{array}{cc} \cos(\tau\Omega) & \frac{1}{\Omega}\sin(\tau\Omega) \\ -\Omega\sin(\tau\Omega) & \cos(\tau\Omega) \end{array} \right).$$

The evolution for the perturbation is immediate since both q and t are frozen.

Notice that if the computational cost is dominated by the evaluation of the time-dependent functions and the rotation matrix, then since $a_i \in \mathbb{R}$ the overall cost does not change considerably either if b_i is real or complex.

For the numerical experiments we take $\Omega(t) = 1 + \frac{1}{2}\cos(\frac{3}{2}t)$ and the same initial conditions and parameters as given in [30]: $(q_0, p_0, t_0) = (0, 11.2075, 0)$, $s = 3$ $\omega_j = 7j$. We integrate for

[5] Here, the method is written in terms of exponentials of matrices, i.e. maps, so they appear in the reverse order as the Lie operators in (3.2.1).

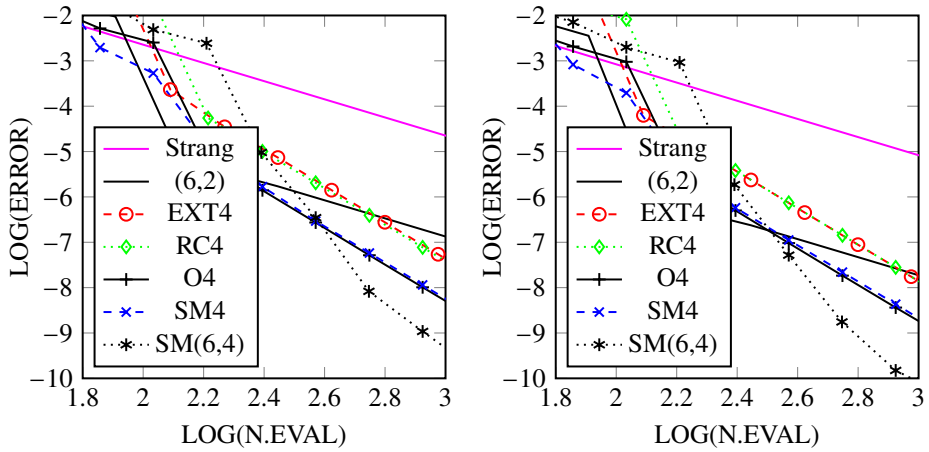


Figure 3.1: Error versus number of evaluations of $\Phi_A^{[h]}$ for the numerical integration in Example 1 at $t = 2\pi$ for $\varepsilon = \frac{1}{4}$ (left panel) and $\varepsilon = \frac{1}{10}$ (right panel).

$t \in [0, 2\pi]$ and measure the error at the final time. All the computations are done for $\varepsilon = 1/4$ and $\varepsilon = 1/10$. Fig. 3.1 shows the error versus the number of evaluations for different methods. We clearly observe the superiority of the methods which consider complex coefficients versus the lower order splitting methods with real coefficients or extrapolation when high accuracy is desired as well as the high performance of the new methods.

Example 2: A linear parabolic equation. The next test-problem is the following scalar parabolic equation in one-dimension

$$\frac{\partial u(x,t)}{\partial t} = \alpha(t)^2 \Delta u(x,t) + V(x,t)u(x,t), \qquad u(x,0) = u_0(x), \tag{3.3.2}$$

with $u_0(x) = \sin(2\pi x)$ and periodic boundary conditions in the space domain $[0,1]$. We take $\alpha(t) = \frac{1}{4} + \mu\cos(wt)$, $V(x,t) = \frac{1}{10}\left(3(1-e^{-t}) + \sin(2\pi x)\right)$ and discretize in space

$$x_j = j(\delta x), \qquad j = 1, \ldots, N \quad \text{with} \quad \delta x = 1/N,$$

thus arriving at the differential equation

$$\frac{dU}{dt} = \alpha(t)^2 AU + B(t)U,$$

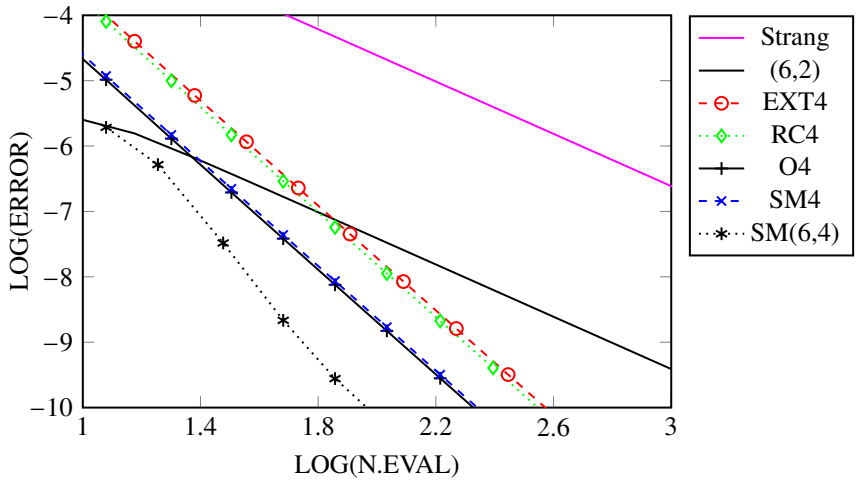


Figure 3.2: Efficiency comparison between algorithms for the linear parabolic equation (3.3.2) with parameters $\mu = 1/6, w = 2$ at final time $t = 1$.

where $U = (U_1, \dots, U_N) = (u_1, \dots, u_N) \in \mathbb{R}^N$. The Laplacian Δ has been approximated by the matrix A of size $N \times N$ given by[6]

$$A = \frac{1}{(\delta x)^2} \begin{pmatrix} -2 & 1 & & & 1 \\ 1 & -2 & 1 & & \\ & 1 & -2 & 1 & \\ & & \ddots & \ddots & \ddots \\ 1 & & & 1 & -2 \end{pmatrix}, \tag{3.3.3}$$

and $B(t) = \mathrm{diag}(V(x_1, t), \dots, V(x_N, t))$. We take $\mu = 1/6$, $w = 2, N = 100$ points and compare different composition methods by computing the corresponding approximate solution on the time interval $[0, 1]$. We compute the 2-norm error of the numerical solution with respect to the exact solution of the semidiscretised equation (computed numerically up to a sufficiently high accuracy) at time $t = 1$. The results are collected in Fig. 3.2 where the superiority of the splitting methods with complex coefficients is also manifest.

Example 3: The semi-linear reaction-diffusion equation of Fisher. Our final test-problem is the following non-linear parabolic scalar equation in one-dimension

$$\frac{\partial u}{\partial t} = \alpha(t)^2 \Delta u + F(u, t), \qquad u(x, 0) = u_0(x),$$

with periodic boundary conditions in the space domain $[0, 1]$. We take, in particular, the Fisher's potential

$$F(u) = \gamma(t) u(1 - u),$$

[6]Our main purpose here is just to illustrate the performance of the new splitting methods. In this sense, the particular scheme used to discretize in space is irrelevant. For that reason, and to keep the treatment as simple as possible, we have applied a simple second-order finite difference scheme in space.

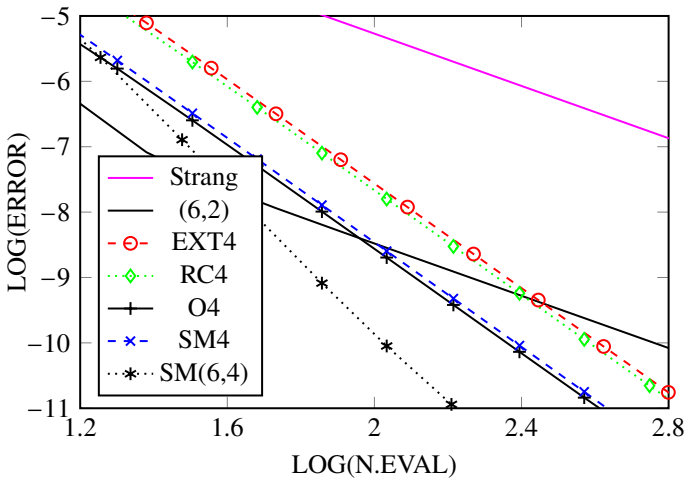


Figure 3.3: Efficiency comparison between algorithms for the equation of Fisher with parameters $\beta = 1, \mu = 1/6, w = 2$ at final time $t = 1$.

with $\gamma(t) = (2 - e^{-\beta t})/100$ and $\alpha(t) = \frac{1}{4} + \mu\cos(wt)$.

The splitting considered here corresponds to solving, on one hand, the linear equation with A given by (3.3.3) and on the other hand, the nonlinear ordinary differential equation

$$\frac{\partial u}{\partial t} = \gamma(t)u(1-u)$$

with initial condition $u(x,0) = u_0(x)$. After discretization in space, we arrive at the differential equation

$$\frac{dU}{dt} = \alpha(t)^2 AU + F(U,t),$$

where $U = (U_1, \dots, U_N) = (u_1, \dots, u_N) \in \mathbb{R}^N$, A is a circulant matrix of size $N \times N$ as in the the previous linear case and $F(U,t)$ is now defined by

$$F(U,t) = \gamma(t)(U_1(1-U_1), \dots, U_N(1-U_N)).$$

Here we consider splitting technique (3.2.4) as

$$\begin{cases} U' = \alpha(t_1)^2 AU \\ t_1' = 1 \end{cases} \quad \text{and} \quad \begin{cases} U' = \gamma_1 U(1-U) \\ t_1' = 0, \end{cases}$$

where $\gamma_1 = \gamma(t_1)$. Since $\gamma(t)$ is frozen at real values of t, it must be considered as a constant, and the scalar equations can be solved analytically

$$u(x,h) = u_0(x)\frac{e^{\gamma_1 h}}{1 + u_0(x)(e^{\gamma_1 h} - 1)},$$

which is well defined for small complex time h. We proceed in the same way as for the previous linear case, starting with $u_0(x) = \sin(2\pi x)$.

We choose $\beta = 1, \mu = 1/6,\ w = 2, N = 100$ and compute the error at the final time $t = 1$ by applying the same composition methods as in the linear case. The results are collected in Fig. 3.3.

Chapter 4

EXPONENTIAL OF PERTURBED MATRICES

The efficient computation of matrix exponentials has been extensively considered in the literature and the *scaling and squaring method* is perhaps the most widely used method for matrices of dimension $N \times N$ with N as large as a few hundred (see [53, 75, 93] and references therein). For example, Matlab and Mathematica compute numerically the exponential of matrices using this method where highly efficient algorithms for general matrices exist [52, 53, 54, 74]. The predominant algorithm is based on scaling the large matrix A by a small number 2^{-s}, which is then exponentiated by efficient Padé or Taylor methods and finally squared in order to obtain an approximation for the full exponential. We propose splitting methods for the computation of the exponential of perturbed matrices which can be written as the sum $A = D + \varepsilon B$ of a sparse and efficiently exponentiable matrix D with sparse exponential e^D and a dense matrix εB which is of small norm in comparison with D. After proper design and application, higher order splitting methods have been found to be superior to standard methods for certain classes of perturbed matrices.

4.1 The scaling, splitting and squaring method

The exposition is based on the article [8].

Given $A \in \mathbb{C}^{N\times N}$, the method is based on the property

$$e^A = \left(e^{A/2^s}\right)^{2^s}, \quad s \in \mathbb{N}$$

where typically $e^{A/2^s}$ is replaced by a polynomial approximation (e.g. a mth-order Taylor method, $T_m(A/2^s)$) or a rational approximation (e.g. an $2m$th-order diagonal Padé method, $r_{2m}(A/2^s)$) [53, 54, 89]. The optimal choice of both s and the algorithms to compute $e^{A/2^s}$ usually depend on the value of $\|A\|$ and the desired tolerance, and have been deeply analyzed.

The computational cost, $c(\cdot)$, is usually measured by the number of matrix–matrix products, so $c(e^A) = s + c(e^{A/2^s})$, where $c(e^{A/2^s})$ has to be replaced by the cost of its numerical approximation, e.g. $c(T_m(A/2^s))$ or $c(r_{2m}(A/2^s))$. Given a tolerance, one has to look for the scheme which provides such accuracy with the minimum number of products (see [53, 54] and references therein).

In some cases, if the matrix A has a given structure, more efficient methods can be obtained [35, 36] . For example, to compute the exponential of upper or lower triangular matrices, in Ref. [74], the authors show that it is advantageous to exploit the fact that the diagonal elements of the exponential are exactly known. It is then more efficient to replace the diagonal elements obtained using, e.g., Taylor or Padé approximations by the exact solution before squaring the matrix (this technique can also be extended to the first super (or sub-)diagonal elements). On the other hand, in many cases the matrix A can be considered as a small perturbation of a sparse matrix D, i.e., $A = D + B$ with $\|B\| < \|D\|$ (and frequently $\|B\| \ll \|D\|$) where e^D is sparse and exactly solvable (or can be accurately and cheaply approximated numerically), and B is a dense matrix. This is the case, for example, if D is diagonal (or block diagonal with small matrices along the diagonal), or if it is diagonalizable using only a few elementary transforms. Another example with similar structure is given by

$$D = \begin{pmatrix} 0 & I \\ -\Omega^2 & 0 \end{pmatrix} \in \mathbb{R}^{N\times N}, \quad n = 2k,$$

where I is the $k \times k$ identity matrix and Ω is a diagonal matrix. Then, the exponential e^D is also sparse and trivial to compute. This problem can originate from a semi-discretization of a hyperbolic PDE or from a set of k linearly coupled oscillators.

As a motivational example, let us consider the linear time-dependent system of differential equations for the evolution operator (or fundamental matrix solution) $X(t, t_0)$ which propagates any initial vector $x(t_0)$ as $x(t) = X(t, t_0)x(t_0)$,

$$\frac{d}{dt}X = M(\varepsilon t)X, \qquad X(t_0, t_0) = I \in \mathbb{C}^{N\times N},$$

with $M \in \mathbb{C}^{N\times N}$ and $|\varepsilon| \ll 1$, i.e., $M(\varepsilon t)$ evolves adiabatically with the variable t. Suppose that $M(\varepsilon t)$ is instantaneously diagonalizable, i.e., $M(\varepsilon t) = Q(\varepsilon t)D(\varepsilon t)Q^{-1}(\varepsilon t)$ with D a diagonal matrix. Then, we can consider what is usually called the *adiabatic picture* in quantum mechanics (if M is a skew-Hermitian matrix) for the treatment of the time-dependent Schrödinger equation [57, 61, 102] which is obtained by the change of variables, $X = Q(\varepsilon t)Y$, that leads to the following differential equation for Y,

$$\frac{d}{dt}Y = \left(D - Q^{-1}\frac{d}{dt}Q\right)Y, \qquad Y(t_0, t_0) = Q^{-1}(\varepsilon t_0)X(t_0, t_0).$$

We remark that in order for the diagonalization or exponentiation to be computationally feasible in quantum mechanics, the dimensionality of the Hamiltonian (here $-iM(\varepsilon t)$) might have to be reduced by considering the subspace spanned by a sufficiently large number of the lower eigenvalues [57, 102]. Our focus lies on the computation of perturbed exponentials, such as the mentioned evolution operator $Y(t, t_0)$ and thus, we exclusively mention a procedure[1] that directly motivates our problem setting: A second order method in the time step h

[1] Cf. Refs. [57, 61] and citations thereof for alternative approaches.

which advances the solution from t_i to $t_i + h$, where $Y_i \approx Y(t_i, t_0)$, is given by

$$Y_{i+1} = e^{h(D_{1/2}+\varepsilon B_{1/2})} Y_i, \tag{4.1.1}$$

where

$$D_{1/2} = D\left(\varepsilon\, t_{i+1/2}\right), \qquad \varepsilon B_{1/2} = -Q^{-1}\left(\varepsilon\, t_{i+1/2}\right) \frac{d}{dt} Q\left(\varepsilon\, t_{i+1/2}\right),$$

with $t_{i+1/2} = t_i + \frac{h}{2}$. Notice that $\varepsilon B_{1/2}$ is, in general, a dense matrix with a small norm (proportional to ε) due to the term $\frac{d}{dt}Q(\varepsilon t)$.

It is then natural to look for methods that approximate the exponential (4.1.1) at a low computational cost while providing sufficient accuracy. Notice that in most cases in practice it is not necessary to approximate the exponential up to round-off accuracy since the model/method itself does not reproduce the exact solution within round-off precision.

The aim of this work is the exploration of new and more efficient algorithms which take advantage of the fact that e^D is sparse and can be computed cheaply and that B has a small norm.

The new schemes are constructed and analyzed using splitting and composition techniques which will be tailored for this particular problem.

For clarity in the presentation, let us write the exponent as $A = D + \varepsilon B$, assuming that $\|B\| \sim \|D\|$. We take the partition $s = s_1 + s_2$, we set $h = 2^{-s_2}, n = 1/h = 2^{s_2}$, and we propose a new recursive procedure that we refer to as modified squaring

$$X_0 = e^{bh\varepsilon B}, \qquad X_k = X_{k-1} e^{a_k h D} X_{k-1}, \qquad k = 1, \dots, s_1 \tag{4.1.2}$$

and $Y_{s_1} = e^{a_{s_1+1} hD} X_{s_1} e^{a_{s_1+1} hD}$ where $b = 1/2^{s_1}$ and the parameters a_k will be chosen properly to improve accuracy. The total cost is

$$c\left(\left(Y_{s_1}\right)^{2^{s_2}}\right) = s_1 + s_2 + c(e^{bh\varepsilon B}),$$

where $c(e^{bh\varepsilon B}) = c(e^{\varepsilon B/2^s})$ is the cost to approximate this exponential. Since $\|h\varepsilon B\|$ is very small, a low-order diagonal Padé approximation can provide sufficient accuracy (for most problems it will suffice to just consider r_2 or r_4 which only require one inversion or one inversion and one product, respectively; a low-order Taylor approximation can also be used).

Notice that this modified squaring algorithm will preserve the same qualitative properties (e.g. a certain Lie group structure) as the underlying diagonal Padé scheme which is used to approximate the small exponential when both parts D and B belong to such a Lie algebra [55].

The choice $s_1 = 0$ corresponds to the Leapfrog or Strang method,

$$e^{h(D+\varepsilon B)} \approx e^{hD/2} e^{h\varepsilon B} e^{hD/2}, \tag{4.1.3}$$

where, as already mentioned, $e^{hD/2}$ can be accurately and cheaply computed.

More accurate methods can be obtained using a general composition

$$S_p^{[m]} = \prod_{i=1}^{m} e^{ha_i D} e^{hb_i \varepsilon B} \approx e^{h(D+\varepsilon B)}, \tag{4.1.4}$$

where the coefficients a_i, b_i are chosen such that $S_p^{[m]}$ is an approximation to the exact solution up to a given order, p, in the parameter h, i.e., $S_p^{[m]} = e^{h(D+\varepsilon B)} + \mathcal{O}(h^{p+1})$. However, to get efficient methods it is crucial to reduce the computational cost. Since the cost is dominated by the exponentials $e^{hb_i\varepsilon B}$, it is advisable to reuse as many exponentials as possible, e.g., letting $b_i = 1/m$, only one exponentiation is necessary. However, this class of methods has some limitations since for orders greater than 2, at least one of the coefficients a_i and one of the b_i must be negative and thus might jeopardize the re-utilization of the exponentials. However, for small perturbations, very accurate results can still be obtained with positive coefficients.

In the particular situation when $A \in \mathbb{C}^{N\times N}$, complex coefficients, $a_i \in \mathbb{C}$, can be used without increasing the computational cost, and then fourth-order methods with all b_i real and equal are achievable. The proposed recursive algorithm (4.1.2) corresponds to a particular case of a splitting method where the cost has been reduced while still leaving some free parameters for optimisation.

In this work, we assume that the product B^2 requires $\mathcal{O}\left(N^3\right)$ operations but DB requires only $\mathcal{O}\left(kN^2\right)$ with $k \ll N$ (e.g., $c(B^2) = 1$, $c(DB) = \delta$, with $\delta \ll 1$). Then, the commutator $\varepsilon[D,B] = \varepsilon(DB - BD)$ can be computed at considerably smaller cost than the product of two dense matrices while retaining a small norm due to the factor ε. It then makes sense to consider the recursive algorithm (4.1.2) where the exponential $e^{bh\varepsilon B}$ is replaced by

$$e^{bh\varepsilon B + \alpha h^3 \varepsilon [D,[D,B]]},$$

whose computational cost is similar, but more accurate results can be obtained if the scalar parameter α is properly chosen. Further exploiting this approach leads to the inclusion of the term $\beta h^5 \varepsilon[D,[D,[D,[D,B]]]]$ in the central exponential, which again, for an appropriate choice of the parameter β, decreases the error at a similar computational cost. The analysis presented in this work is also extended to the case in which not all parameters b_i are taken equal.

4.2 Computational cost of matrix exponentiation

4.2.1 Computational cost of Taylor and Padé methods

We first review the computational cost of the optimized Taylor and Padé methods which are used in the literature and that are used as reference in the numerical examples.

Taylor methods We use the Paterson-Stockmeyer scheme (see [51, 54, 80]) to evaluate $T_m = \sum_{k=0}^{m} A^k/k!$ which considerably reduces the number of required products.

From the Horner-scheme-like computation, given a number of matrix products $2k$, the maximal attainable order is $m = (k+1)^2$. In Ref. [54], it is indicated that the optimal choice for most cases corresponds to $k = 3$, i.e., order $m = 16$ with just 6 products given by: $A^2 = AA$, $A^3 = A^2A$, $A^4 = A^2A^2$ and

$$T_{16}(A) = g_0 + (g_1 + (g_2 + (g_3 + g_4A^4)A^4)A^4)A^4,$$

where g_i are linear combinations of already computed matrices, $g_i = \sum_{k=0}^{4} c_{i,k} A^k$, with $c_{i,k} = 1/(4i+k)!$ for $i = 0, 1, 2, 3$ and $g_4 = I/16$ proportional to the identity (matrix).

Diagonal Padé methods Diagonal Padé methods are given by the rational approximant

$$r_{2m}(A) = \frac{p_m(A)}{p_m(-A)},$$

provided the polynomials p_m are generated by the recurrence

$$\begin{aligned} p_0(A) &= I, \qquad p_1(A) = 2I + A \\ p_m(A) &= 2(2m-1)p_{m-1}(A) + A^2 p_{m-2}(A). \end{aligned} \tag{4.2.1}$$

Moreover, $r_{2m}(A) = e^A + \mathcal{O}(A^{2m+1})$, whereas for $m = 1, 2$ we have

$$r_2(A) = \frac{I + A/2}{I - A/2}, \qquad r_4(A) = \frac{I + A/2 + A^2/12}{I - A/2 + A^2/12}. \tag{4.2.2}$$

The recursive algorithm (4.2.1) is, however, not an efficient way to compute $r_{2m}(A)$. For example, the method $r_{26}(A)$ is considered among the optimal choices (with respect to accuracy and computational cost) of diagonal Padé methods when round off accuracy is desired and $\|A\|$ takes relatively large values. The algorithm to compute it is given by

$$(-u_{13} + v_{13}) r_{26}(A) = (u_{13} + v_{13}), \tag{4.2.3}$$

with

$$\begin{aligned} u_{13} &= A[A_6(b_{13}A_6 + b_{11}A_4 + b_9 A_2) + b_7 A_6 + b_5 A_4 + b_3 A_2 + b_1 I], \\ v_{13} &= A_6(b_{12}A_6 + b_{10}A_4 + b_8 A_2) + b_6 A_6 + b_4 A_4 + b_2 A_2 + b_0 I, \end{aligned}$$

where $A_2 = A^2, A_4 = A_2^2, A_6 = A_2 A_4$. Written in this form, it is evident that only six matrix multiplications and one inversion are required. In a similar way, the method $r_{10}(A)$, which will be used in this work, only requires 3 products and one inversion.

4.2.2 Computational cost of splitting methods

Recall that we are considering a sparse and sparsely exponentiable matrix D, while B is a dense matrix and responsible for the numerical complexity. In order to build competitive algorithms, it is important to analyze - under these assumptions - the computational cost of all operations involved in the different classes of splitting and composition methods.

Let X, Y be two dense $N \times N$ matrices and denote by $c(\cdot)$ the cost of the operations in brackets as the number of matrix–matrix products of dense matrices, e.g., $c(XY) = 1$ and $c(X + Y) = \delta$, with $\delta \ll 1$, thereby neglecting operations with a lower complexity in the number of operations. According to this criterion, we derive Table 4.1, where the dominant terms are highlighted in boldface (the cost for the inverse of a matrix is taken as 4/3 the cost of a matrix-matrix product [54]).

Based on this analysis, we examine the splitting method (4.1.4) to identify the computationally relevant aspects. In this work, we assume $\delta \ll 1$ and in our computations we will take $\delta = 0$ for

Table 4.1: Computational cost of matrix operations for the sparse and sparsely exponentiable matrix D and arbitrary dense matrices $X, Y \in \mathbb{C}^{N\times N}$. The inversion, $X^{-1}Y$, refers to solving the linear system $XZ = Y$ which appears in every Padé method, cf. (4.2.3). The factor w in $c(e^D)$ is assumed to be small, $w \ll 1$.

	Operation	Effort
Sum	$c(D+D) \approx 0$	$\mathcal{O}(kN)$, with $k \ll N$
	$c(X+Y) = \delta$	$\mathcal{O}(N^2)$
Product	$c(XY) = \mathbf{1}$	$\mathcal{O}(N^3)$
	$c(DD) = 0$	$\mathcal{O}(k^2N)$
	$c(DX) = k\delta$	$\mathcal{O}(kN^2)$
Inversion	$c(X^{-1}Y) = \mathbf{1} + \frac{1}{3}$	$c(X^{-1}Y) = \frac{4}{3}c(XY)$
Commutation	$c([D,X]) = c(DX - XD) = 2k\delta$	$\mathcal{O}(kN^2)$
	$c([D,[D,\dots,[D,X]\cdots]]) = 2rk\delta$	$\mathcal{O}(kN^2)$
Exponentiation	$c(e^D) = wk\delta$	$\mathcal{O}(k^2N)$
	$c(r_2(X)) = \mathbf{1} + \frac{1}{3}$	$\mathcal{O}(N^3)$
	$c(r_4(X)) = \mathbf{2} + \frac{1}{3}$	$\mathcal{O}(N^3)$

simplicity. First, we have to choose how to approximate the exponentials $e^{h\varepsilon b_i B}$. The methods from (4.2.2) have accuracy

$$r_2(h\varepsilon b_i B) = e^{h\varepsilon b_i B} + \mathcal{O}(h^3\varepsilon^3), \qquad r_4(h\varepsilon b_i B) = e^{h\varepsilon b_i B} + \mathcal{O}(h^5\varepsilon^5).$$

A rough estimate for the composition (4.1.4), assuming all coefficients b_i different, and taking into account the cost from Table 4.1, we have

$$c(S_p^{[m]}, r_2) = m\frac{4}{3} + m - 1 = \frac{7}{3}m - 1, \qquad c(S_p^{[m]}, r_4) = m\frac{7}{3} + m - 1 = \frac{10}{3}m - 1,$$

where $c(S_p^{[m]}, r_i)$ denotes the cost of the method $S_p^{[m]}$ when the exponentials $e^{\varepsilon B}$ are approximated by $r_i(\varepsilon B)$. Repeating the coefficients b_i, i.e., $b_i = 1/m,\ i = 1,\dots,m$, the computational cost can be reduced considerably and in this case, one gets

$$c(S_p^{[m]}, r_2) = \frac{4}{3} + (m-1) = m + \frac{1}{3}, \qquad c(S_p^{[m]}, r_4) = m + \frac{4}{3}.$$

Further simplifications are applicable and will be discussed in Sect. 4.4.

4.3 The Lie algebra of a perturbed system: (p_1, p_2) methods

Following the terminology of Ref. [69], we introduce a modified error concept which is suitable for the near-integrable structure of the matrix A at hand.

Letting $S_p^{[m]}$ be a pth-order m-stage consistent ($\sum_i a_i = \sum_i b_i = 1$) splitting method (4.1.4), we expand its error as

$$S_p^{[m]} - e^{hA} = \sum_{i=p+1} \sum_{j=1} e_{i,j}\varepsilon^j h^i C_{i,j},$$

where $e_{i,j}$ is a polynomial in the splitting coefficients a_k, b_k and $C_{i,j}$ is a sum of matrix products consisting of all combinations containing $(i-j)$ sparse elements D and j times B. Notice that

in addition to the scaling h, we also expand in powers of the small parameter ε. The method is said to be of effective order $p = (p_1, p_2, \ldots)$ if $e_{i_1,1} = e_{i_2,2} = \ldots = 0$ for all $i_k \le p_k$ and $p_1 \ge p_2 \ge \cdots$.

Designing a method now consists of identifying the dominant error terms $e_{i,j}\varepsilon^j h^i$ and finding coefficients a_j, b_j to zero the polynomials $e_{i,j}$. The main tool in this endeavor is the Baker-Campbell-Hausdorff formula which provides a series expansion of the single exponential that has been actually computed when multiplying two matrix exponentials,

$$e^{hA}e^{hB} = e^{\mathrm{bch}(hA,hB)}, \quad \mathrm{bch}(hA, hB) = h(A+B) + \frac{h^2}{2}[A, B] + \mathcal{O}(h^3).$$

Recursive application of this formula to a symmetric splitting (4.1.4) establishes the concept of a modified matrix $h\tilde{A}$, along the lines of backward-error-analysis,

$$\begin{aligned}\log(S_p^{[m]}) = h\tilde{A} = hA + \tilde{e}_{3,1}\varepsilon h^3[D,[D,B]] + \tilde{e}_{3,2}\varepsilon^2 h^3[B,[D,B]] \\ + \tilde{e}_{5,1}\varepsilon h^5[D,[D,[D,[D,B]]]] + \tilde{e}_{5,2}\varepsilon^2 h^5[[D,[D,B]],[D,B]] \\ +\tilde{e}_{5,3}\varepsilon^2 h^5[B,[D,[D,[D,B]]]]+\tilde{e}_{7,1}\varepsilon h^7[D,[D,[D,[D,[D,[D,B]]]]]]+\mathcal{O}\left(\varepsilon^3 h^5 + \varepsilon^2 h^7\right),\end{aligned} \tag{4.3.1}$$

where the $\tilde{e}_{i,j}$ are also polynomials in the splitting coefficients a_k, b_k which multiply elements of the Lie algebra and are different from the coefficients $e_{i,j}$. Higher-order terms can be computed by efficient algorithms [33].

4.3.1 Error propagation by squaring

The splitting method (4.3.1) can also formally be written as

$$S_{(p_1,p_2)}^{[m]} = \exp\left(h(D+\varepsilon B) + \varepsilon \sum_{k>p_1} c_k h^k \,\mathrm{ad}_D^{k-1}(B) + \mathcal{O}\left(\varepsilon^2 h^{p_2+1}\right)\right),$$

where the adjoint operator is recursively defined as $\mathrm{ad}_D(B) = [D, B]$, $\mathrm{ad}_D^k(B) = [D, \mathrm{ad}_D^{k-1}(B)]$ and there is only one term proportional to ε at each power of h. We can then define a *processor*, a map which is close to identity,

$$P = \exp\left(-\varepsilon \sum_{k>p_1} c_k h^{k-1} \,\mathrm{ad}_D^{k-2}(B)\right),$$

such that the method can be written as

$$S_{(p_1,p_2)}^{[m]} = PKP^{-1}, \tag{4.3.2}$$

with

$$K = \exp\left(h(D+\varepsilon B) + \mathcal{O}\left(h^{p_2+1}\varepsilon^2\right)\right).$$

Suppose now that the matrix A can be diagonalized, $A = QD_AQ^{-1}$, then clearly

$$e^A = Qe^{D_A}Q^{-1}.$$

The *kernel* K of the numerical method is also close to identity and can be diagonalized for sufficiently small $h = 1/n$ and ε using

$$\hat{Q} = Q + \mathcal{O}\left(h^{p_2+1}\varepsilon^2\right), \qquad \hat{D}_A = hD_A + \mathcal{O}\left(h^{p_2+1}\varepsilon^2\right),$$

such that, after n integration steps, we obtain

$$K^n = \hat{Q}e^{\tilde{D}_A}\hat{Q}^{-1}, \tag{4.3.3}$$

with $\tilde{D}_A = D_A + \mathcal{O}\left(nh^{p_2+1}\varepsilon^2\right)$. The size estimates of the above considerations lead to a favorable error propagation result which is stated in the following theorem.

Theorem 4.3.1. *Let $A = D + \varepsilon B$ a diagonalizable matrix and let $S^{[m]}_{(p_1,p_2)}$ be an m-stage splitting method of order (p_1, p_2) that approximates the scaled exponential e^{hA} with $h = 1/n$. Then, for sufficiently small values of h and ε we have that*

$$\left\| e^A - \left(S^{[m]}_{(p_1,p_2)}\right)^n \right\| \leq C_1 h^{p_1+1}\varepsilon + nC_2 h^{p_2+1}\varepsilon^2,$$

where C_1, C_2 are constants which depend on the norm $\|e^A\|$ but neither on h nor on ε.

Proof. From (4.3.2) and (4.3.3) we have that

$$\left(S^{[m]}_{(p_1,p_2)}\right)^n = P\hat{Q}e^{\tilde{D}_A}\hat{Q}^{-1}P^{-1} = \tilde{Q}e^{\tilde{D}_A}\tilde{Q}^{-1},$$

where now $\tilde{Q} = P\hat{Q} = Q + \mathcal{O}\left(h^{p_1+1}\varepsilon\right)$. Then

$$\begin{aligned}
\left\| e^A - \left(S^{[m]}_{(p_1,p_2)}\right)^n \right\| &= \left\| Qe^{D_A}Q^{-1} - \tilde{Q}e^{\tilde{D}_A}\tilde{Q}^{-1} \right\| \\
&= \left\| Qe^{D_A}Q^{-1} - \tilde{Q}e^{D_A}Q^{-1} + \tilde{Q}e^{D_A}Q^{-1} - \tilde{Q}e^{\tilde{D}_A}\tilde{Q}^{-1} \right\| \\
&\leq \|Q - \tilde{Q}\|\,\|e^{D_A}Q^{-1}\| + \|\tilde{Q}\|\,\|e^{D_A}Q^{-1} - e^{\tilde{D}_A}\tilde{Q}^{-1}\|.
\end{aligned}$$

The right summand is expanded in a similar way to

$$\begin{aligned}
\|e^{D_A}Q^{-1} - e^{\tilde{D}_A}\tilde{Q}^{-1}\| &= \|e^{D_A}Q^{-1} - e^{\tilde{D}_A}Q^{-1} + e^{\tilde{D}_A}Q^{-1} - e^{\tilde{D}_A}\tilde{Q}^{-1}\| \\
&\leq \|e^{D_A} - e^{\tilde{D}_A}\|\,\|Q^{-1}\| + \|e^{\tilde{D}_A}\|\,\|Q^{-1} - \tilde{Q}^{-1}\|.
\end{aligned}$$

Taking into account that $\tilde{D}_A = D_A + \mathcal{O}\left(nh^{p_2+1}\varepsilon^2\right)$, $\tilde{Q} = Q + \mathcal{O}\left(h^{p_1+1}\varepsilon\right)$, and subsuming $\|e^A\|$ into the constants, we obtained the desired result for sufficiently small values of h and ε. □

This result indicates that the error is the sum of a local error of effective order $\mathcal{O}(\varepsilon)$ plus a global error of order $\mathcal{O}\left(\varepsilon^2\right)$. For problems which require a relatively large number of squarings (a large value of $n = 2^s$) the dominant error of the splitting methods is proportional to ε^2. Then, to build methods which are accurate for different values of s, it seems convenient to look for methods of effective order (p_1, p_2) with $p_1 > p_2$.

The following numerical example illustrates the results obtained.

Example Let

$$A = \begin{pmatrix} \varepsilon & 1+\varepsilon \\ -1+\varepsilon & -\varepsilon \end{pmatrix}, \qquad D = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \tag{4.3.4}$$

with $\varepsilon = 10^{-1}, 10^{-3}$, and approximate $e^{2^s A} = \left(\cdots \left(e^A\right)^2 \cdots\right)^2$ to a relatively low accuracy. To approximate e^A, we consider a fourth-order Taylor method, $T_4(A)$ (that only requires 2 products) and a fourth-order Padé approximation, $r_4(A)$ (with a cost of one product and one inversion, equivalent to $1 + 4/3$ products). We compare the obtained results with the second-order splitting method (4.1.3), which we denote by $S_2^{[2,a]}$ or, since in this case $p_1 = p_2 = 2$, $S_{(2,2)}^{[2,a]}$, where the exponential e^D is computed exactly and εB is approximated with the second order diagonal Padé method, $r_2(\varepsilon B)$. The exact solution is given by

$$e^{2^s A} = \begin{pmatrix} \cos(2^s\mu) + \frac{\varepsilon}{\mu}\sin(2^s\mu) & \frac{1+\varepsilon}{\mu}\sin(2^s\mu) \\ -\frac{1-\varepsilon}{\mu}\sin(2^s\mu) & \cos(2^s\mu) - \frac{\varepsilon}{\mu}\sin(2^s\mu) \end{pmatrix}$$

with $\mu = \sqrt{1-2\varepsilon^2}$ and we analyze the error growth due to the squaring process in Fig. 4.1. We observe that neither Padé nor Taylor methods are sensitive w.r.t. the small parameter, whereas the splitting method drastically improves when decreasing ε. The splitting method is only of second order and thus used with the second order Padé method r_2 (using the fourth order method r_4 leaves error plot unchanged). Notice that for the small perturbation $\varepsilon = 10^{-3}$, the splitting with $r_2(\varepsilon B)$ is more accurate than the fourth-order Padé $r_4(A)$ which comes at nearly twice the computational cost (1 inversion vs. 1 inversion and 1 dense product). According to Theorem 4.3.1, the error of $S_{(2,2)}^{[2,a]}$ is the sum of a local error proportional to $h^3\varepsilon$ and a global error proportional to $nh^3\varepsilon^2$, with $n = 2^s$. Figure 4.1 shows the results obtained for different values of ε and s which clearly show both sources of error.

4.4 Splitting methods for scaling and squaring

Taking into account the numerical effort established in the introduction, we derive methods which are optimized for the problem at hand. The optimization principle becomes clear at the example of the two versions of Strang's second-order splitting method

$$\begin{aligned} S_2^{[2,a]} &= e^{\frac{h}{2}D}\, e^{h\varepsilon B}\, e^{\frac{h}{2}D} = \mathscr{D}_{h/2}\mathscr{B}_h\mathscr{D}_{h/2} \\ \text{and} \qquad S_2^{[2,b]} &= e^{\frac{h}{2}\varepsilon B}\, e^{hD}\, e^{\frac{h}{2}\varepsilon B} = \mathscr{B}_{h/2}\mathscr{D}_h\mathscr{B}_{h/2}, \end{aligned} \tag{4.4.1}$$

which differ in computational cost: using the notation $\mathscr{D}_h = e^{hD}$, $\mathscr{B}_h = e^{h\varepsilon B}$, and keeping in mind that $\mathscr{D}_h$ is a sparse matrix while $\mathscr{B}_h$ is dense, the dominant numerical cost amounts to a single exponential with $c(S_2^{[2,a]}) = c(\mathscr{B}_h)$ for the first version, whereas the latter requires an additional matrix product, $c(S_2^{[2,b]}) = c(\mathscr{B}_{h/2}) + c(\mathscr{B}\mathscr{B})$.

Furthermore, the large dominant part D is multiplied by $1/2$ before exponentiation in the cheaper variant which is advantageous in the sense of the scaling process.

We follow a variety of strategies in order to develop new methods and group them according to the splitting terminology, keeping in mind that the costly parts are products and exponentials of the dense matrices $\mathscr{B}$ and B, respectively.

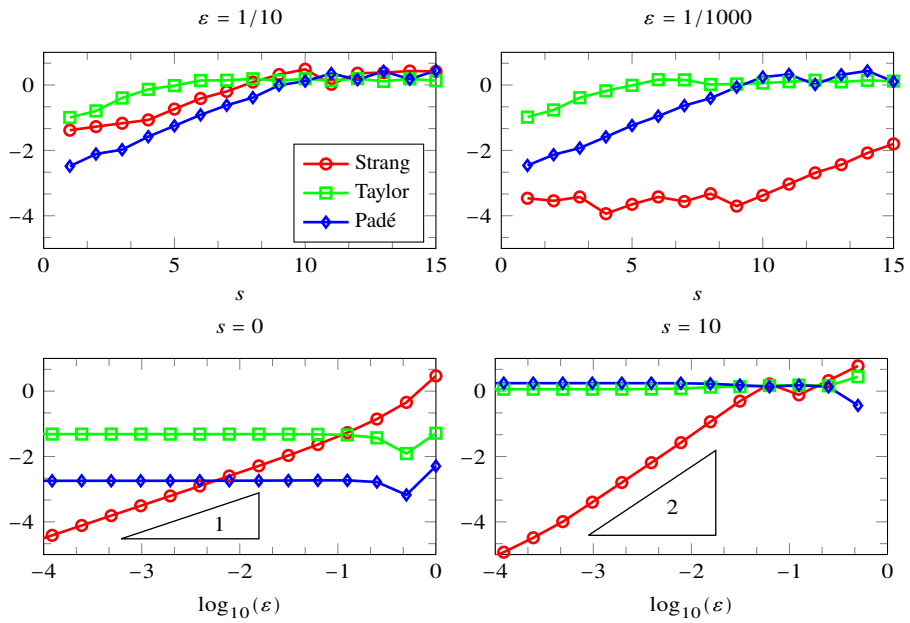


Figure 4.1: Error in the approximation to $e^{2^s A}$ with A given by (4.3.4) for different values of ε and number of squarings, s, in double-logarithmic scale. The bottom figures show that the error of the splitting methods is proportional to ε for small s (local error) and proportional to ε^2 for large values of s (global error).

4.4.1 Standard splittings

As we have discussed for the Strang splitting $S_2^{[2,b]}$, despite the appearance of B in two exponents, only one exponential actually has to be computed which is then stored and reused for the second identical exponent.

Generalizing this principle, we search for splitting methods a_i, b_j where all $b_j = b$ are identical to reduce the computational effort which now comes solely from the dense-matrix multiplications. A composition that is also symmetric in the coefficients a_j will reduce the number of error terms (since even powers in h disappear) and the amount of (cheap) exponentials $\mathscr{D}$ to be computed.

Next, we derive a particular family of splittings which can be understood in analogy to squarings and allow us to reduce the necessary products.

Modified squarings

We propose to replace a given number of squarings by a one-step splitting method which has the benefit of free parameters to minimize the error. For illustration, let us compute a squaring step, $h = 2^{-1}$, of the standard Strang method

$$(e^{h/2A}e^{h\varepsilon B}e^{h/2A})^2 = e^{\frac{1}{4}A}e^{\frac{1}{2}\varepsilon B}e^{\frac{1}{2}A}e^{\frac{1}{2}\varepsilon B}e^{\frac{1}{4}A}, \tag{4.4.2}$$

which we then contrast with a general splitting method at the same cost (one exponential and one product) without squaring ($h = 1$),

$$e^{a_2 A}e^{\frac{1}{2}\varepsilon B}e^{a_1 A}e^{\frac{1}{2}\varepsilon B}e^{a_2 A}. \tag{4.4.3}$$

It is evident that (4.4.3) includes (4.4.2) as a special case (choosing $a_1 = 1/2, a_2 = 1/4$) and we use the example (4.3.4) to illustrate the gains in accuracy. Fig. 4.2 shows that the performance is very sensitive to the choice of the free parameter and the method of effective order $(4, 2)$ (choosing $a_1 = 1/\sqrt{3}, a_2 = (1 - a_1)/2$) is very close to the optimal one. A larger

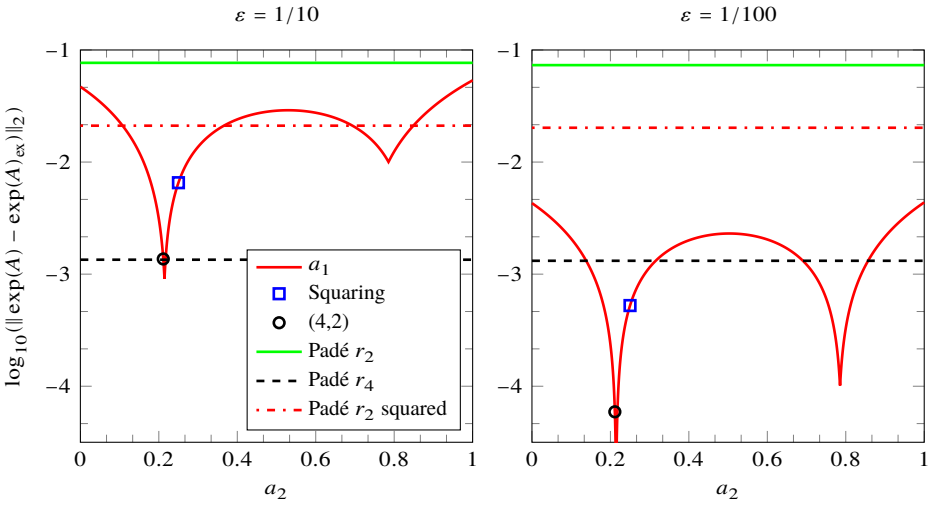


Figure 4.2: (color online) Modified squarings. All methods apart from the least accurate, $r_2(A)$ (green solid), have approximately the same numerical cost since the split uses 2nd order Padé for the B part.

number of squarings s can be replaced by a recursive procedure,

$$X_0 = e^{hb\varepsilon B}, \qquad X_k = X_{k-1}e^{a_k hD}X_{k-1}, \qquad k = 1, \ldots, s$$

and $Y_s = e^{a_{s+1}h}X_s e^{a_{s+1}h}$ where $b = 1/2^s$. The costly multiplications occur in the consecutive steps, X_k, where we recycle already computed blocks while introducing free parameters a_k at negligible extra effort. As a result, the cost of the algorithm is

$$c(Y_s) = s + c(e^{hb\varepsilon B}),$$

where it usually suffices to approximate $e^{hb\varepsilon B}$ with the second or fourth-order Padé method (4.2.2), so $c(e^{hb\varepsilon B}, r_2) = \frac{4}{3}$ and $c(e^{hb\varepsilon B}, r_4) = 1 + \frac{4}{3}$. For consistency, the coefficients a_k have to satisfy

$$\left(2^{s-1}a_1 + \cdots + 2a_{s-1} + a_s\right) + 2a_{s+1} = \sum_{k=1}^{s} 2^{s-k}a_k + 2a_{s+1} = 1.$$

Notice that the choice $a_{s+1} = 1/2^{s+1}$, $a_k = 1/2^s$ for $k = 1, \ldots, s$, corresponds to the standard scaling and squaring applied to the Strang method (4.4.1). In the following, we have collected the most efficient splitting methods for an increasing numbers of products $s = 0, 1, 2, 3, 4$. We have observed in the numerical experiments that for $s > 4$, the gain w.r.t. to standard scaling and squaring is marginal, and they are not considered in this work.

However, the parameter h demonstrates how any such method can be combined with standard scaling and squaring.

This procedure is equivalent to consider the partition $s = s_1 + s_2$, where the first s_1 squarings are carried out with the recursive algorithm with $b = 1/2^{s_1}$ and we continue with the remaining standard s_2 squarings with $h = 1/2^{s_2}$.

$s_1 = 0$. Strang $S_2^{[2,a]}$ with local error $\mathcal{O}(\varepsilon h^3)$.

$s_1 = 1$. After imposing symmetry, one free parameter remains and is used to obtain (4,2) methods [62, 69],

$$Y_1 = \mathscr{D}_{ha_2}\mathscr{B}_{h/2}\mathscr{D}_{ha_1}\mathscr{B}_{h/2}\mathscr{D}_{ha_2}, \tag{4.4.4}$$

where $a_1 = 1/\sqrt{3}$, $a_2 = (1 - a_1)/2$ and with local error $\mathcal{O}(\varepsilon h^5 + \varepsilon^2 h^3)$.

$s_1 = 2$. Allowing an additional product, at $b = 1/4$, we have

$$Y_2 = \mathscr{D}_{a_3 h}(\mathscr{B}_{h/4}\mathscr{D}_{a_2 h}\mathscr{B}_{h/4})\mathscr{D}_{a_1 h}(\mathscr{B}_{h/4}\mathscr{D}_{a_2 h}\mathscr{B}_{h/4})\mathscr{D}_{a_3 h}.$$

Optimizing the free parameters a_3, a_2, (where for consistency $a_1 = 1 - 2(a_3 + a_2)$) we can construct fourth-order methods, although complex-valued, with $a_3 = \frac{1}{10}(1 - i/3), a_2 = \frac{2}{15}(2 + i)$ and their complex conjugates a_i^* [34]. Alternatively, there are six real-valued (6,2) methods, the best of which is given in Table 4.2.

$s_1 = 3$. The three parameters for Y_3 can be used to produce complex-valued methods of order (6,4) or real-valued methods of order (8,2), the ones with smallest error coefficients can be found in Table 4.2.

$s_1 = 4$. The next iteration yields a 17-stage method Y_4. Its four parameters can be used to cancel the error coefficients $e_{3,1}, e_{3,2}, e_{5,1}, e_{7,1}$ for 48 complex (8,4) methods, or a $(10, 2)$ method with positive real coefficients, see Table 4.2.

Table 4.2: Modified squarings with and without commutators. In the right column, the corresponding computational cost is given together with the number of omitted solutions of the order conditions.

Y_2, order (6,2)	$c(\mathscr{B}_{h/4}) + 2c(\mathscr{B}\mathscr{B})$
$a_1 = \sqrt{(5-\sqrt{5})/30},\ a_2 = \sqrt{(5-2\sqrt{5})/15}$	[7 solutions omitted]
Y_3, order (8,2)	$c(\mathscr{B}_{h/8}) + 3c(\mathscr{B}\mathscr{B})$
$a_1 = 0.15394202084115342013479021 3164$	only positive solution
$a_2 = 0.08999923764546260567963098 6655$	[47 omitted]
$a_3 = 0.10224455429143755862716103 0779$	
$a_4 = \frac{1}{2} - (4a_1 + 2a_2 + a_3)/2.$	
Y_3, order (6,4)	
$a_1 = 0.13534452760420860194 + 0.06201309787740406230i$	[7 omitted]
$a_2 = 0.13027125534284511606 - 0.10310039626441585374i$	
$a_3 = 0.09906233274082533725 1 - 0.01588542476623739072 4i$	
$a_4 = \frac{1}{2} - (4a_1 + 2a_2 + a_3)$	
Y_4, order (10,2)	$c(\mathscr{B}_{h/16}) + 4c(\mathscr{B}\mathscr{B})$
$a_1 = 0.07725593304829713720207789 3145$	only positive solution
$a_2 = 0.04449263223932042451890593 70354$	[383 omitted]
$a_3 = 0.05108077361369342943802798 6467$	
$a_5 = 0.02545536598413089904583906 46508$	
$a_4 = 1 - 8a_1 - 4a_2 - 2a_3 - 2a_5$	
Y_4, order (8,4)	
$a_1 = 0.06782965853562196485274129 + 0.03038453954138687801299186i$	[47 omitted]
$a_2 = 0.06477414774829711915884478 - 0.05170904068177844632921239i$	
$a_3 = 0.04963134399080347125041612 + 0.00584283681423207753349501i$	
$a_5 = 0.02474856149827627051056177 - 0.00610084851840072905292033i$	
$a_4 = 1 - 8a_1 - 4a_2 - 2a_3 - 2a_5$	
$\tilde{Y}_2$, order (6,4), minimizing $\mathcal{O}(\varepsilon^2 h^5)$	$c(\mathscr{B}_{h/4}) + 2c(\mathscr{B}\mathscr{B})$
$a_1 = (1 - a_2 - 2a_3)/2$	
$a_2 = 0.47071989362081947165$	
$a_3 = 0.04898669326146179875$	
$\beta = -0.002320917859694561351$	
$\gamma = 0.000032954671822820378 2$	
$\tilde{Y}_2$, order (8,4)	[47 omitted]
$a_1 = 0.36022581463894912207346 47$	
$a_2 = 1 - 2(a_3 + a_1)$	
$a_3 = 0.07661021300692938614830 05$	
$\beta = -0.00103637077918270398691258$	
$\gamma = 0.00001024048253259859441 1391$	

4.4.2 Modified splittings

A drastic improvement on the previous methods can be achieved through the use of commutators. The special structure of the matrix implies the fast computability of certain commutators, namely the ones that contain the matrix B only once. The inclusion of these commutators in the scheme will not only allow to reduce the number of error terms but also to reach order 4 using only real coefficients. Since we are interested in symmetric methods of up to order

(6,4), the relevant terms are

$$[D,[D,B]] = DDB - 2DBD + BDD,$$
$$[D,[D,[D,[D,B]]]] = DDDDB - 4DDDBD + 6DDBDD - 4DBDDD + BDDDD,$$

and neglecting the numerical cost of summation and multiplication by a sparse matrix D, it is clear that the exponential

$$e^{\alpha h\varepsilon B+\beta h^3\varepsilon[D,[D,B]]+\gamma h^5\varepsilon[D,[D,[D,[D,B]]]]} = \tilde{\mathscr{B}}_{\alpha,\beta,\gamma} \tag{4.4.5}$$

can be evaluated at the same cost as $\mathscr{B}_{\alpha h}$. Along the lines of the modified squarings, we have derived the following compositions which require only one exponential $\tilde{B}$ at a fixed number of products. The substitution $Y_s \to \tilde{Y}_s$ indicates the replacement of B by $\tilde{B}$.

$s = 0$. Strang's method can be made into a (6,2) scheme with

$$\tilde{Y}_0 = \mathscr{D}_{h/2}\tilde{\mathscr{B}}_{1,1/24,1/1920}\mathscr{D}_{h/2}. \tag{4.4.6}$$

We stress that, in principle, a method of order $(2n, 2)$ can be constructed using only a single exponential, however, at the expense of increasingly complicated commutators, $[D,[D,[\dots,[D,B]]\cdots]$ whose computational complexity cannot be neglected anymore.

$s = 1$. Replacing $\mathscr{B}_{h/2}$ by $\tilde{\mathscr{B}}$ in (4.4.4), we obtain the (6,4) method

$$\tilde{Y}_1 = \mathscr{D}_{ha_2}\tilde{\mathscr{B}}\mathscr{D}_{ha_1}\tilde{\mathscr{B}}\mathscr{D}_{ha_2}, \tag{4.4.7}$$

where $a_2 = 1/6$, $a_1 = 2/3$ and $\tilde{\mathscr{B}}_{1/2,-1/144,121/311040}$ with unchanged effort $c(\mathscr{B}_{h/2}) + c(\mathscr{B}\mathscr{B})$.

$s = 2$. Using one additional multiplication, we reach $\tilde{Y}_2$, which can be tuned to be of order (8,4) or (6,4) while minimizing the error at $\mathscr{O}(\varepsilon^2 h^5)$, see Table 4.2.

We have also analyzed other classes of splitting and composition methods. Those methods showed a worse performance in the numerical examples tested in this work. However, we have included the corresponding approaches for completeness in the Appendix A.1.

4.5 Error analysis

Our methods have proven successful for a low to medium accuracy since the high-order Padé methods are hard to beat at round-off precision. In a first step, we derive new scaling estimates for Padé methods for lower precision requirements following Ref. [53]. Let $\theta_{2m}(u)$ be the largest value of $\|A\|$ s.t. the Padé scheme r_{2m} has backward-error smaller than u, i.e.,

$$\forall A, \|A\| \le \theta_{2m} \;:\; r_{2m}(A) = e^{A+E}, \text{ s.t. } \|E\| \le u.$$

The new θ_{2m} are given in Table 4.3. It is clear that the number of necessary squarings for a sought precision is $s = \lceil \log_2(\|A\|/\theta_{2m}) \rceil \in \mathbb{N}_0$ and taking into account the number of

Table 4.3: Theta values for diagonal Padé of order $2m$ with minimum number of products. The numbers highlighted in boldface correspond to the minimal cost $\pi_{2m} - \log_2(\theta_{2m})$. The first row was already derived in [52].

$u\backslash m$	1	2	3	5	7	9	13	17
$\leq 2^{-53}$	3.65E-8	5.32E-4	1.50E-2	2.54E-1	9.50E-1	2.10	**5.37**	9.44
$\leq$ 1E-10	3.46E-5	1.64E-2	1.47E-1	9.98E-1	**2.51**	4.44	8.94	1.38E1
$\leq$ 1E-6	3.46E-3	1.64E-1	6.80E-1	**2.48**	4.76	7.24	1.24E1	1.77E1
$\leq$ 1E-4	3.46E-2	5.16E-1	1.45	**3.85**	6.47	9.15	1.45E1	1.99E1

multiplications π_{2m} needed for each method, a global minimum $s + \pi_{2m}$ can be found at each precision.

We will focus our attention on the medium precision range $u \leq 10^{-6}$, where the 10th order method r_{10} is optimal among the Padé schemes. In analogy to the error control for Padé methods, we discuss the backward error of the previously obtained splitting methods. The BCH formula, in the form (4.3.1), already gives us a series expansion of the remainder E,

$$E = \sum_{i=p+1} \sum_{j=1} h^i f_{i,j} \mathbf{C}_{i,j}. \tag{4.5.1}$$

However, the expansion is difficult to compute for $i > 15$ with exponentially growing effort in the symbolic computation. Further complications arise from the nature of the expansion: it involves commutators $\mathbf{C}_{i,j}$ in D, B which we have to estimate. For most cases, the roughest (although sharp) estimate

$$\|[D, B]\| = \|DB - BD\| \leq 2\|D\|^2, \qquad \|B\| = \|D\|, \tag{4.5.2}$$

is way to loose to give accurate results. Having in mind matrices with asymmetric spectra, i.e., small positive and large negative eigenvalues, the following estimate is more useful [60, Theorem 4],

$$\|[D, B]\| \leq \|B\|(d^+ - d^-),$$

where the numerical range of D (or easier: the eigenvalues) lies within $[d^-, d^+]$, which corresponds to a factor 2 gain in the estimate. In any case, we can refine the estimate by recycling the calculations for the modified splittings, $[D, [D, B]]$, $[D, [D, [D, [D, B]]]]$ and intermediate steps, $[D, B]$, etc. Then, we estimate the most relevant commutators

$$\begin{aligned}
\|[B, [D, B]]\| &\leq 2\|[D, B]\|\|B\|, \\
\|[B, [D, [D, [D, B]]]]\| &\leq 2\|[D, B]\|\,\|[D, [D, B]]\|, \\
\|[D, [B, [D, [D, B]]]]\| &\leq 2\|[D, B]\|\,\|[D, [D, B]]\|, \\
\|[B, [B, [D, [D, B]]]]\| &\leq 4\|B\|^2\,\|[D, [D, B]]\|, \\
\|[D, [D, [D, [D, [D, B]]]]]\| &\leq (d^+ - d^-)^2\|[D, [D, [D, B]]]\|.
\end{aligned}$$

The splitting methods studied in this work can be classified by their order and the leading error commutators are collected in Table 4.4.

In principle, one could use the error terms at the next larger power in h to estimate the quality of this truncation, but for practical purposes and $h \ll 1$, numerical experiments show that the

simpler bounds are sufficient to get a reasonable recommendation for the number of squarings. For illustration, we print the expansion (4.5.1) for the method $\tilde{Y}_0$ (4.4.6),

$$\begin{aligned} E^{[6,2]}(h) \leq \tilde{E}^{[6,2]} = {} & 3.11\text{E-}6\varepsilon h^7 \| \operatorname{ad}_D^6(B)\| + 8.33\text{E-}2\varepsilon^2 h^3 (\|[B,[D,B]]\| \\ & + h^5 (1.39\text{E-}3\|[B, \operatorname{ad}_D^3(B)]\| + 5.56\text{E-}3\|[[B,D], \operatorname{ad}_D^2(B)]\|)) \\ & + \varepsilon^3 h^5 (5.56\text{E-}3\| \operatorname{ad}_B^2(\operatorname{ad}_D^2(B))\| + 2.78\text{E-}3\|[[B,D], \operatorname{ad}_B^2(D)]\|) \\ & + \mathcal{O}\left(\varepsilon h^9 + \varepsilon^2 h^7 + \varepsilon^3 h^7\right) \end{aligned} \tag{4.5.3}$$

and for method $\tilde{Y}_1$ from (4.4.7),

$$\begin{aligned} E^{[6,4]}(h) \leq \tilde{E}^{[6,4]} = {} & 3.49\text{E-}5\varepsilon h^7 \| \operatorname{ad}_D^6(B)\| \\ & + \varepsilon^2 h^5 (1.70\text{E-}3\|[B, \operatorname{ad}_D^3(B)]\| + 1.39\text{E-}3\|[[B,D], \operatorname{ad}_D^2(B)]\|) \\ & + \varepsilon^3 h^5 (1.39\text{E-}3\|[B^2, \operatorname{ad}_D^2(B)]\| + 4.63\text{E-}4\|[[B,D], \operatorname{ad}_B^2(D)]\|) \\ & + \mathcal{O}\left(\varepsilon h^9 + \varepsilon^2 h^7 + \varepsilon^3 h^7\right). \end{aligned} \tag{4.5.4}$$

Then, the following algorithm suggests itself: compute the commutators needed for the modified squarings, estimate their norms and finally evaluate the polynomials $\tilde{E}(h)$ to find an upper bound for h such that the local error remains below the given accuracy u. This h translates directly to the number of external squarings $s_2 = \lceil \log_2(h) \rceil$ and now, it only remains to sum the computational cost originating from the number of dense products and exponentials to find the overall most efficient method for a particular set of matrices D, B. In contrast to the static Padé case, where there is a single best method by just fixing the precision, this procedure is more flexible and chooses - at virtually no extra cost - the best method for the given matrix algebra structure.

Furthermore, we can establish a threshold for the size of the small parameter ε in order to decide when splittings should be preferred over Padé methods. For example, let $u = 10^{-6}(10^{-4})$ be the desired precision, we then know that r_{10} (r_{10}) is optimal and the largest value the norm $\theta = \|A\|$ can take is $\theta_{10} = 2.48(\theta_{10} = 3.85)$. Given that r_{10} requires three multiplications, we use the splitting method $\tilde{Y}_0$ with three squarings to yield a method of the same computational cost. In (4.5.3), this corresponds to taking $h = 2^{-3}$. Applying the roughest possible estimate (4.5.2) to $\tilde{E}^{[6,2]}(2^{-3})$, we obtain a polynomial in ε which takes values below u for $\varepsilon \leq 0.01(0.05)$. In practice, the norm estimates are sharper since we can use the commutators that have been computed in the algorithm and we expect an even larger threshold for ε.

Table 4.4: Leading error commutators at given order.

order	ε^1	ε^2	ε^3
$(2n, 2)$	$\operatorname{ad}_D^{2n}(B)$	$[B,[D,B]]$	$[B,[B,[D,[D,B]]]]$
$(2n, 4)$	$\operatorname{ad}_D^{2n}(B)$	$[B,[D,[D,[D,B]]]]$, $[D,[B,[D,[D,B]]]]$	$[B,[B,[D,[D,B]]]]$

4.6 Numerical results

In a couple of test scenarios, we attempt to provide an idea about when our new methods are superior to standard Padé methods. In each setting, we let $A = D + \varepsilon B$ and define a matrix D which will be perturbed by a matrix B, s.t.

$$B_{i,j} = \frac{\|D\|_1}{\|G\|_1} G, \qquad \text{with} \quad G_{i,j} = (i-j)/(i+j),$$

thus $\|B\|_1 = \|D\|_1$ and the perturbation size is chosen from the parameter set $\varepsilon = 10^{-1}, 10^{-2}, 10^{-3}$. We measure the relative error in the 1-norm, $\|S_p^{[m]} - e^A\|_1 / \|e^A\|_1$ for all methods where the exact solution is computed by a high-order Padé method and all splittings use the second-order scheme r_2 to approximate the exponential $\exp(2^{-s}\varepsilon B)$.

4.6.1 Rotations

Letting

$$D = i\,\mathrm{diag}\{-25, -24.5, \dots, 24.5, 25\},$$

with $i = \sqrt{-1}$, the performance of Padé methods of order 10 and 26, together with the 16th-order Taylor method is studied. Figure 4.3 shows the relative error (in logarithmic scale) versus cost (number of dense-matrix multiplications[2]) for different choices of the scaling parameter, s. The horizontal line shows the tolerance desired for the numerical experiments. It is evident that, as expected, the Padé method r_{10} is the most efficient among these standard schemes and will be used for reference in later experiments. For illustration, Fig. 4.3 also includes two modified squaring methods without commutators (Y_2, order (6,2) and Y_3, order (6,4) from Table 4.2), both of which are more efficient than r_{10} in the lower precision range. Notice that, since A is a complex matrix, the use of splitting methods with complex coefficients does not increase the cost of the algorithms in this case. Furthermore, the standard methods are insensitive w.r.t. the small parameter ε, whereas the splitting methods improve as ε decreases.

In a second experiment in Fig. 4.4, we use the same matrices as before but choose the most efficient splitting methods with commutators, $\tilde{Y}_0$ and $\tilde{Y}_1$. Using the local error estimates in (4.5.3) and (4.5.4), we indicate the point which corresponds to the optimal number of squarings for the splitting methods and compare it with the recommended squaring parameter for Padé r_{10}. For a relatively large parameter ε in the left panel of Fig. 4.4, the method r_{10} is still superior but is already equaled in terms of computational cost for a smaller perturbation in the center plot, but at higher accuracy. As ε becomes smaller in the right panel, we achieve higher accuracy at lower computational cost, saving one product for $\tilde{Y}_1$ and two products for $\tilde{Y}_2$, respectively.

In the next plot, Fig. 4.5, we increase the norm of the matrix and set $D_2 = 100D$, and B is scaled accordingly to maintain the equality $\|B\|_1 = \|D_2\|_1$. This implies a substantial increase in the number of necessary squarings with prior scaling and corresponds to a long-time integration in which we observe the favorable behavior expected from Fig. 4.1. The gain with respect to Padé's method is striking as ε decreases.

[2] The effort for matrix inversions is accounted for as $4/3$ matrix multiplications.

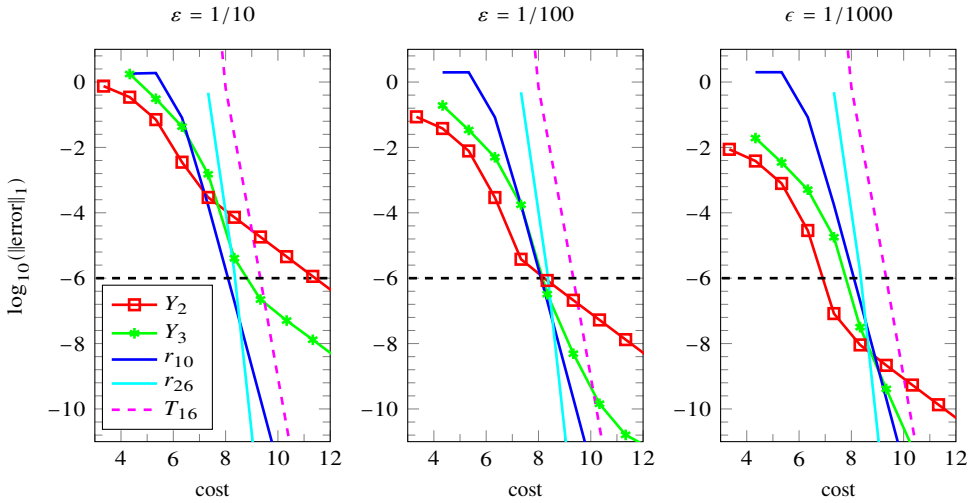


Figure 4.3: Relative error (in logarithmic scale) versus computational cost given by the number of dense matrix-matrix products for the standard Padé and Taylor methods r_{10}, r_{26}, T_{16}, and the splitting methods Y_2 and Y_3 of order (6,2) and (6,4), respectively, without commutators from Table 4.2.

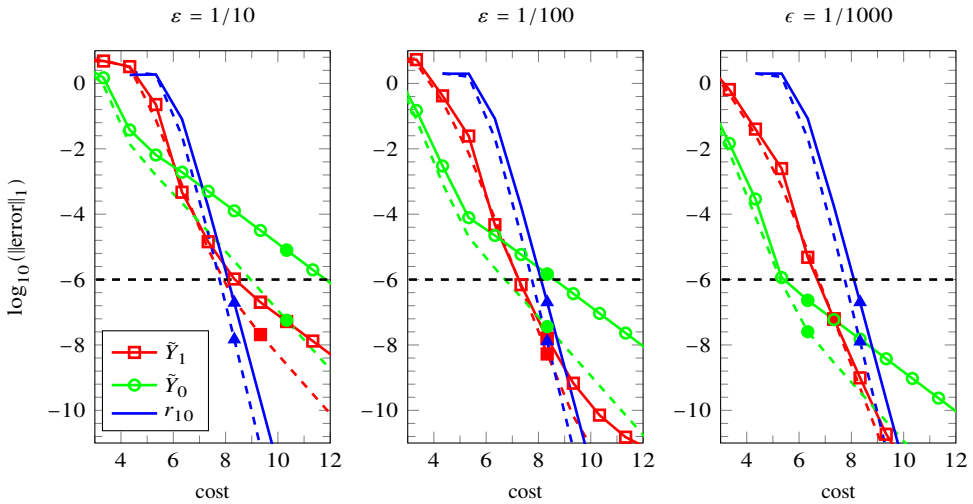


Figure 4.4: The solid lines show the relative global error e^A after squaring versus the overall computational cost and the dashed curves depict the relative local error in $e^{2^{-s}A}$ (before squaring) which is used for the error estimate, both for Padé and the splittings. The filled markers indicate the position of the recommended (automatic) algorithm.

4.6.2 Dissipation

A less favorable problem for our algorithm is given using a stiff matrix with large positive and negative eigenvalues,

$$D = \text{diag}\{15, 14.5, \ldots, -14.5, -15\}.$$

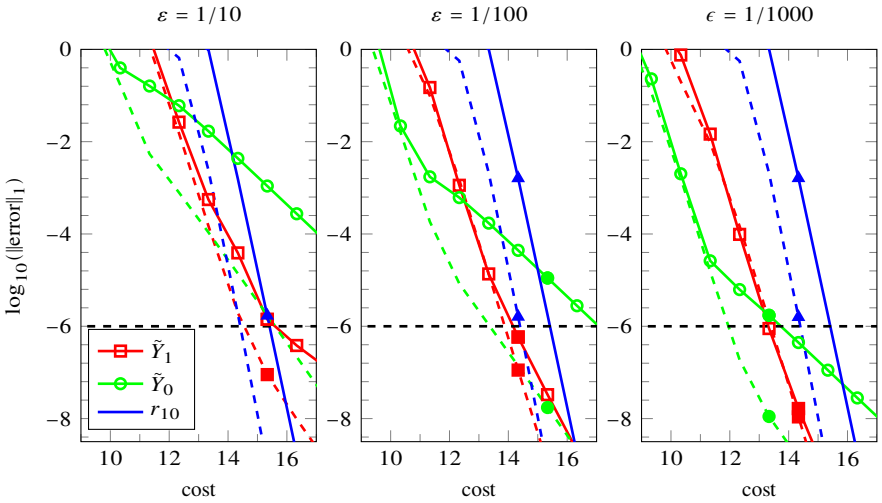


Figure 4.5: Same as Fig. 4.4 for an exponential of a large norm matrix, with diagonal part $D_2 = 100D$.

The perturbation B is scaled as before to $\|B\| = \|D\|$. Figure 4.6 shows the results obtained. Again, our methods perform well for low accuracies for not too large perturbations and improve as ε becomes smaller.

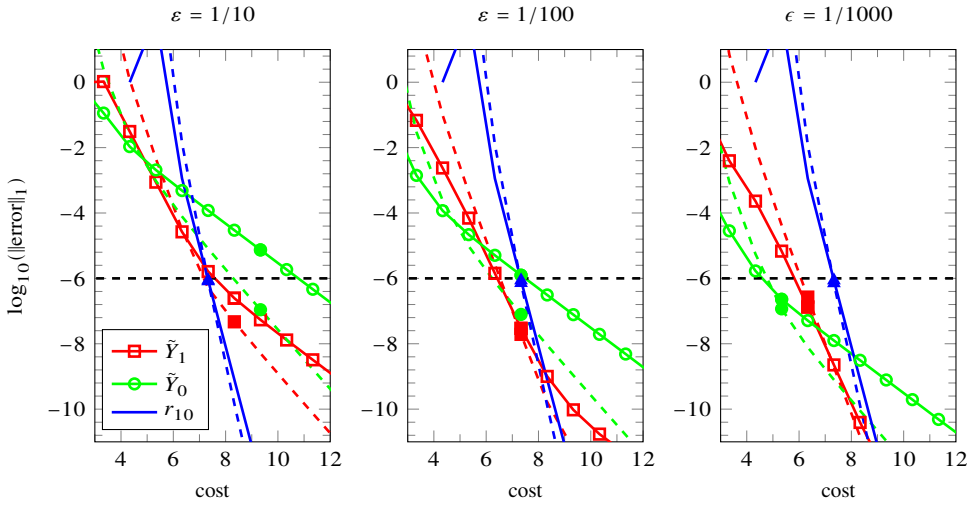


Figure 4.6: Same as Fig. 4.4 but for the stiff matrix case $D = \mathrm{diag}\{15, 14.5, \ldots, -14.5, -15\}$.

Chapter 5

MATRIX HILL'S EQUATION

The Hill's equation has many applications in practical periodically variable systems like the study of quadrupole mass filters and quadrupole devices [43, 81], microelectromechanical systems [99], parametric resonances in Bose-Einstein condensates [29, 45], spatially linear electric fields, dynamic buckling of structures, electrons in crystal lattices, waves in periodic media, etc. (see also [66, 67, 68, 85, 103] and references therein). Parametric resonances can appear and this property is of great interest in many different physical applications. Usually, the Hill's equations originate from a Hamiltonian function and the fundamental matrix solution is a symplectic matrix. This is a very important property to be preserved by the numerical integrators. On the other hand, if the system is weekly damped and/or it has a week non-linear interaction then using the numerical integrators derived for the perturbed system is most advantageous to integrate the whole perturbed systems over long time intervals. Thus, splitting methods for perturbed systems can be used with the constraint that the time-dependent Hill's equation must be solved (numerically) to high accuracy and the perturbation is integrated with the time frozen and it is exactly solved.

5.1 Symplectic integrators for the matrix Hill's equation

The exposition is based on the article [7].

The study of the potential of a charged particle moving in the electric field of a quadrupole, without considering the effects of the induced magnetic field, leads to the equations of motion

$$\frac{d^2}{dt^2}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} -\frac{2e}{md}\left(E_0 + E_1\cos(t)\right) & 0 & 0 \\ 0 & \frac{2e}{md}\left(E_0 + E_1\cos(t)\right) & 0 \\ 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix},$$

where e is the charge of the particle, m is the mass and d is the minimum distance from the electrode to the x_3-axis (the direction in which the particle is traveling), and $E(t) = E_0 + E_1 \cos(t)$ is the electric field which is radio-frequency-modulated, see [86].

The decoupled equations for x_i, $i = 1, 2$, are the well-known Mathieu equations

$$x''(t) + (\alpha - \beta \cos(t))\, x(t) = 0\,;$$

where α and β are constant parameters and each prime denotes a derivative in time.

Stable trajectories are guaranteed only in certain regions in the $\alpha - \beta$ plane and hence, the motion of the particles can be regulated by E_0 and E_1 by placing them within or outside of said stability regions.

This property allows to filter particles and is the underlying principle of quadrupole mass spectrometry. The setup is also known as Paul's trap [81]. This important application motivated a great interest in the study of periodically time-dependent systems and the effect of parametric resonances.

In this work, we consider the more general problem which is a matrix version of the so called Hill's equation

$$x'' + M(t)x = 0\,, \tag{5.1.1}$$

where $t \in \mathbb{R}$, $x \in \mathbb{C}^r$ and $M(t)$ is a periodic $r \times r$ matrix valued function with period T. Frequently, eq. (5.1.1) is written as

$$x'' + \left(A + \sum_{n=1}^{\infty} B_n \cos(nt) \right) x = 0\,,$$

which reduces - when truncating after the first term - to the Mathieu equation.

Most works from the literature concern on analytical methods to find the stability regions. However, the computation of the trajectories is of great interest in many cases, and this has to be carried numerically. If, in addition, the system is weekly damped and/or it has a week non-linear interaction, i.e.

$$x''(t) + \delta x'(t) + M(t)x(t) = \kappa f(t, x)\,,$$

with δ, κ small parameters and $f(t, x)$ a periodic function with period T, the numerical integration of the differential equations over long time intervals is the usual procedure to study the systems.

There exists efficient numerical integrators tailored to solve perturbed time-dependent linear problems [4, 21, 90]. These methods require to split the system into the dominant linear part and the perturbation as follows

$$\begin{bmatrix} x \\ x' \end{bmatrix}' = \begin{bmatrix} 0_{r\times r} & I_{r\times r} \\ -M(t) & 0_{r\times r} \end{bmatrix} \begin{bmatrix} x \\ x' \end{bmatrix} + \begin{bmatrix} 0_{r\times 1} \\ -\epsilon x' + \varepsilon f(t, x) \end{bmatrix} = A(t)z + B(t, z),$$

where $z = [x, x']^T$ and $I_{r\times r}$ denotes the $r \times r$ identity matrix. Then, splitting methods for perturbed systems can be used with the constraint that the time-dependent linear equation,

Table 5.1: Two-exponential sixth-order symplectic method using the Gauss-Legendre quadrature rule for solving the Hill's equation (5.2.1).

$$c_1 = \tfrac{5-\sqrt{15}}{10},\ c_2 = \tfrac{1}{2},\ c_3 = \tfrac{5+\sqrt{15}}{10}$$

$$M_1 = M(t_n + c_1 h), \qquad M_2 = M(t_n + c_2 h), \qquad M_3 = M(t_n + c_3 h),$$

$$K = M_1 - M_3, \qquad L = -M_1 + 2M_2 - M_3, \qquad F = h^2 K^2,$$

$$C_1 = -\tfrac{\sqrt{15}}{180}K + \tfrac{1}{18}L + \tfrac{1}{12960}F, \quad D_1 = -M_2 - \tfrac{4}{3\sqrt{15}}K + \tfrac{1}{6}L$$

$$C_2 = +\tfrac{\sqrt{15}}{180}K + \tfrac{1}{18}L + \tfrac{1}{12960}F, \quad D_2 = -M_2 + \tfrac{4}{3\sqrt{15}}K + \tfrac{1}{6}L$$

$$\Psi_n = \begin{pmatrix} I & 0 \\ hC_2 & I \end{pmatrix} \exp\left(\frac{h}{2}\begin{pmatrix} 0 & I \\ D_2 & 0 \end{pmatrix}\right) \exp\left(\frac{h}{2}\begin{pmatrix} 0 & I \\ D_1 & 0 \end{pmatrix}\right) \begin{pmatrix} I & 0 \\ hC_1 & I \end{pmatrix}$$

$z' = A(t)z$ must be solved (numerically) to high accuracy and the perturbation is integrated with the time frozen and it is exactly solved.

In [42], the theory of Floquet is applied in order to study equation (5.1.1).

The majority of published works discusses analytical methods to find the stability regions. However, the computation of the trajectories is of great interest in many cases, and this has to be carried out numerically. In most cases, Hill's equation originates from a Hamiltonian system where $M^T = M$, and the fundamental matrix solution is a symplectic matrix. This property plays a fundamental role on the stability of the system.

For Hamiltonian systems it is essential to preserve the symplectic structure of the exact solution for both the study of stability regions and for very long time numerical integrations. Therefore, our goal in this work is to design symplectic methods based on Magnus expansions which will efficiently integrate Hill's equation. We analyze new methods that are closely related to commutator-free methods [1, 23] and integrators for Nth-order linear systems [6] but they are tailored for solving the relevant matrix Hill's equation. The first step in this undertaking is to write Hill's eq. (5.1.1) as a first-order system

$$\frac{d}{dt}\begin{pmatrix} x \\ x' \end{pmatrix} = \begin{pmatrix} 0 & I \\ -M(t) & 0 \end{pmatrix}\begin{pmatrix} x \\ x' \end{pmatrix},$$

and we propose new sixth- and eighth-order symplectic methods for its solution. The most efficient sixth-order symplectic method that we have found is summarized in Table 5.1 which describes the evolution operator Ψ_n for a step of size h from t_n to $t_{n+1} = t_n + h$

$$\begin{pmatrix} x \\ x' \end{pmatrix}_{n+1} = \Psi_n \begin{pmatrix} x \\ x' \end{pmatrix}_n .$$

Notice that $M(t)$ is evaluated at the nodes of the sixth-order Gauss-Legendre quadrature rule,

but the method can easily be adapted to any other quadrature rule.

Furthermore, the matrices $C_i, D_i,\ i = 1, 2$ in the algorithm require only three evaluations of $M(t)$, several linear combinations of them and one matrix-matrix product. Each exponential can be computed up to an accuracy of order h^{2m} using $m-1$ matrix-matrix products and symplecticity can be preserved at an equivalent extra cost of $2+\frac{1}{3}$ products (see the Appendix A.2). This method provides very high accuracy for oscillatory problems while requiring much less evaluations of $M(t)$ per step at a slightly higher computational cost per step when compared to the most efficient explicit symplectic Runge-Kutta-Nyström methods from the literature. Moreover, the method provides the exact solution in the autonomous case.

In this work, we analyze sixth-order methods with one to three exponentials and eighth-order methods with four and five exponentials.

We also remark that for the non-homogeneous linear problem

$$x''(t) + M(t)x(t) = f(t),$$

the stability of the system is independent of the non-homogeneous term $f(t)$ (although the solution can strongly depend on it). This equation is frequently solved by variation of constants, but a more efficient procedure is to write the system as an homogeneous one by extending it as follows [25]

$$\frac{d}{dt}\begin{pmatrix} x \\ x' \\ 1 \end{pmatrix} = \begin{pmatrix} 0_{r\times r} & I_{r\times r} & 0_{r\times 1} \\ -M(t) & 0_{r\times r} & f(t) \\ 0_{1\times r} & 0_{1\times r} & 0_{1\times 1} \end{pmatrix}\begin{pmatrix} x \\ x' \\ 1 \end{pmatrix}.$$

Now it is obvious that the eigenvalues of the fundamental matrix solution will not depend on $f(t)$. The methods presented in this work can be applied to this system with minor changes and the computational cost remains essentially the same since it increases only by a low number of vector-matrix multiplications due to the non-homogeneous term. The method allows for a different treatment of the matrix $M(t)$ and the vector $f(t)$ [25] if required (they can even be evaluated at different nodes).

5.2 Numerical integration for one period

Let us consider eq. (5.1.1) and the equivalent first order system

$$z'(t) = A(t)z(t), \quad A(t) = \begin{pmatrix} 0 & I \\ -M(t) & 0 \end{pmatrix}, \tag{5.2.1}$$

with $z(0) = z_0 \in \mathbb{C}^{2r}$ and periodic $A(t+T) = A(t)$. Let $\Phi(t)$ denote the fundamental matrix solution of (5.2.1), i.e., $z(t) = \Phi(t)z(0)$, then

$$\Phi'(t) = A(t)\Phi(t), \quad \Phi(0) = I_{2r\times 2r}. \tag{5.2.2}$$

Floquet theory tells us that $\Phi(t+T) = \Phi(t)\Phi(T)$. Hence,

$$z(T) = \Phi(T)z_0, \ \ z(2T) = \Phi(T)z(T) = \Phi^2(T)z_0, \ \dots, z(nT) = \Phi^n(T)z_0,$$

and the system is stable if the eigenvalues $\{\lambda_1, \dots, \lambda_{2r}\}$ of $\Phi(T)$ lie in the unit disk, i.e., $|\lambda_i| \leq 1,\ i = 1, \dots, 2r$. Notice that for $|\lambda_i| < 1$, the system is asymptotically stable in the direction of the eigenvector associated to this eigenvalue.

Most systems are Hamiltonian (then $M^T = M$) and $A(t)$ defined by (5.2.1) belongs to the symplectic Lie algebra, i.e., $A(t)^T J + J A(t) = 0,\ \forall t$ where

$$J = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}$$

is the fundamental symplectic matrix. Furthermore, $\Phi(t)$ is a symplectic matrix, i.e., $\Phi(t)^T J \Phi(t) = J, \forall t$, and $\det \Phi(t) = 1$. The eigenvalues of $\Phi(T)$ occur in reciprocal pairs, $\{\lambda, \lambda^*, 1/\lambda, 1/\lambda^*\}$, where * denotes the complex conjugate. This implies that, for stable systems, all of the eigenvalues must lie in the unit circle (see for example [27] for more details on symplectic matrices and their properties).

If we solve numerically eq. (5.2.1) using standard methods like say, a Runge-Kutta scheme of order p, to obtain $\widetilde{\Phi}(T)$ as an approximation to $\Phi(T)$ we find that

$$\widetilde{\Phi}^T J \widetilde{\Phi} = J + \mathcal{O}(h^p)$$

where h is the time step used in the numerical integration, and $\det \widetilde{\Phi}^T = 1 + \mathcal{O}(h^p)$ and consequently, volume preservation is not guaranteed. The eigenvalues of $\widetilde{\Phi}(T)$ will not, in general, occur in pairs and, what is even worse, in general we will find that

$$|\widetilde{\lambda}_k| = 1 + \mathcal{O}(h^p), \qquad k = 1, 2, \dots, 2r.$$

Even if the exact solution is stable, the numerical solution can provide a fundamental matrix solution with eigenvalues inside and/or outside the unit circle. Then, one should use a very small time step to avoid these undesirable effects which is similar to the step size restrictions that occur when using explicit RK methods to solve stiff equations.

For this reason, it is of great interest to study the numerical integration of the Hill's equation using symplectic integrators.

5.2.1 Symplectic methods

In the following, we consider different classes of symplectic integrators from the literature and we analyze their performance when applied to the Hill's equation. This analysis will be used to build new symplectic integrators tailored to the Hill's equation in the following sections.

Implicit symplectic Runge–Kutta methods

It is well known that standard explicit Runge–Kutta (RK) methods are not symplectic. However, the s-stage implicit Runge–Kutta–Gauss–Legendre (RKGL) methods are symplectic and of maximal order $2s$. RK methods are characterized by the real numbers $a_{ij}, b_i\ (i, j = 1, \dots, s)$

and $c_i = \sum_{j=1}^{s} a_{ij}$, and for this linear problem they take the form

$$\begin{aligned} Z_i &= z_n + h \sum_{j=1}^{s} a_{ij} A_j Z_j, \qquad i = 1, \dots, s \\ z_{n+1} &= z_n + h \sum_{i=1}^{s} b_i A_i Z_i, \end{aligned} \tag{5.2.3}$$

where $A_i = A(t_n + c_i h)$. This linear system can be exactly solved, however, from the computational point of view it is not advisable to use direct methods. Notice that given $Q, P \in \mathbb{C}^{r\times r}$, the computational cost to multiply these matrices is

$$\mathscr{C} = c(QP) = 2r^3 - r^2 \text{ flops.}$$

We will take the cost of a method based in units of $\mathscr{C}$. For example, the cost to solve the system $QX = P$ is $c(Q^{-1}P) \simeq \frac{4}{3}\mathscr{C}$. Then, naïve counting of multiplications results in a cost of $(2 \times s)^3\mathscr{C}$ to solve the system (5.2.3) using a direct method, which would render the method uncompetitive versus explicit methods. Obviously, this can be improved using iterative methods to solve the implicit equations (5.2.3), where the cost of each iteration is only $2s\mathscr{C}$ (this number corresponds to the products $A_i Z_i$, $i = 1, \dots, s$ where each product, due to the sparse structure of the matrix A, involves two products of matrices of dimension $r \times r$).

High order methods are useful to get accurate results while using relatively large time steps. On the other hand, large time steps can reduce that rate of convergence of the implicit methods, and in turn require more iterations. In order to preserve the symplecticity, the algorithm should be used with a very small tolerance. One can use a fixed point iteration that - for second order equations - increases the convergence by a factor h^2 at each iteration. In order to preserve symplecticity to nearly round-off accuracy, typically 5–7 iteration will be necessary.

Splitting methods

To overcome the difficulties encountered with implicit methods, symplectic splitting methods can be used in their stead. For this purpose, we express the matrix equation (5.2.2) as an equivalent Hamiltonian system with time-dependent Hamiltonian function

$$H(q,p,t) = \frac{1}{2}p^T p + \frac{1}{2}q^T M(t) q \tag{5.2.4}$$

with $q, p \in \mathbb{R}^r$. Highly efficient symplectic Runge–Kutta-Nyström methods for this Hamiltonian can be found in [23]. One step from t_n to $t_n + h$, of an s-stage method applied to solve (5.2.4) is given by

$\tau_0 := t_n, \quad Q_0 := q_n, \quad P_0 := p_n,$
do $k = 1, \dots, s$
  $Q_k := Q_{k-1} + a_k h P_{k-1}$
  $\tau_k := \tau_{k-1} + a_k h$
  $P_k := P_{k-1} - b_k h M(\tau_k) Q_k$
enddo
$t_{n+1} := \tau_s = t_n + h, \qquad q_{n+1} := Q_s, \qquad p_{n+1} := P_s,$

where a_k, b_k are appropriate coefficients. Notice that the variable t is advanced with the coefficients a_i associated to the kinetic part. This algorithm can be used for solving (5.2.2) as follows

$$\Phi_{n+1} = \begin{pmatrix} I & 0 \\ -b_s\, hM_s & I \end{pmatrix} \begin{pmatrix} I & a_s hI \\ 0 & I \end{pmatrix} \quad \cdots \quad \begin{pmatrix} I & 0 \\ -b_1 hM_1 & I \end{pmatrix} \begin{pmatrix} I & a_1 hI \\ 0 & I \end{pmatrix} \Phi_n,$$

where $\Phi_n \simeq \Phi(t_n)$ and $M_k = M(\tau_k)$. Taking into account that, in general, Φ_n is a $2r \times 2r$ dense matrix and M_k is a $r \times r$ dense matrix, one step requires s evaluations of the time-dependent matrix $M(t)$ (usually of low computational cost for most Hill's equations of practical interest) and $2s$ products of $r \times r$ dense matrices, i.e. the cost for one step would be $2s\mathscr{C}$.

In the present setting, the problem is explicitly time-dependent and highly oscillatory for most values of the parameters and symplectic methods based on the Magnus expansion can be more efficient in many cases.

Magnus integrators

The Magnus expansion [65] expresses the solution to (5.2.2) in the form of a single exponential

$$\Phi(t,0) = \exp\left(\Omega(t,0)\right), \qquad \Omega(t,0) = \sum_{k=1}^{\infty} \Omega_k(t,0), \tag{5.2.5}$$

where the first terms of the Magnus series $\{\Omega_k\}$ are given by

$$\Omega_1(t,0) = \int_0^t A(t_1)\, dt_1, \quad \Omega_2(t,0) = \frac{1}{2}\int_0^t \int_{t_0}^{t_1} \left[A(t_1), A(t_2)\right]\, dt_2\, dt_1, \ \ldots$$

where $[P,Q] = PQ - QP$ is the matrix commutator of P and Q. Here Ω as well as any truncation of the series belong to the Lie algebra, and the symplectic property is preserved.

In order to obtain an approximation to Ω defined by (5.2.5) for a time step from t_n to $t_{n+1} = t_n + h$, it is convenient to express it in the graded free algebra generated by $\{\alpha_1, \alpha_2, \ldots\}$ where

$$\alpha_{i+1} = \frac{h^{i+1}}{i!} \left.\frac{d^i A(t)}{dt^i}\right|_{t=t_n+\frac{h}{2}}, \tag{5.2.6}$$

therefore $\alpha_i = \mathcal{O}(h^i)$. The Magnus expansion Ω can be approximated to arbitrary order in this algebra. We first consider sixth-order methods, and up to this order it suffices to take into account the Lie algebra generated by $\{\alpha_1, \alpha_2, \alpha_3\}$ (see [1] for a relatively simple proof to this result). By Taylor expanding $A(t)$ around the midpoint $t_h + \frac{h}{2}$ (see [18] for a comprehensive review) it follows that we obtain a sixth-order approximation to Ω by

$$\Omega^{[6]} = \alpha_1 + \frac{1}{12}\alpha_3 - \frac{1}{12}[12] + \frac{1}{240}[23] + \frac{1}{360}[113] - \frac{1}{240}[212] + \frac{1}{720}[1112], \tag{5.2.7}$$

where $[ij \ldots kl]$ represents the nested commutator $[\alpha_i, [\alpha_j, [\ldots, [\alpha_k, \alpha_l] \ldots]]]$ and $\Omega^{[6]} = \Omega + \mathcal{O}(h^7)$. Let $c_1, \ldots, c_m$ denote the nodes of a quadrature rule of order six or higher, it is then possible to replace each α_i, $i = 1,2,3$ by a linear combination of $A_k \equiv A(t_n + c_k h)$, $k =$

$1, \dots, m$ such that $\Omega^{[6]}$ is still an approximation to order six. Letting $b_1, \dots, b_m$ denote the weights of the same quadrature rule, the α_i can be written, for example, as

$$\alpha_1 = \frac{9}{4}A^{(0)} - 15A^{(2)}, \quad \alpha_2 = 12A^{(1)}, \quad \alpha_3 = -15A^{(0)} + 180A^{(2)}, \tag{5.2.8}$$

where

$$A^{(i)} = h \sum_{j=1}^{m} b_j \left(c_j - \frac{1}{2}\right)^i A_j, \qquad i = 0, 1, 2. \tag{5.2.9}$$

If we consider, for example, Gauss–Legendre (GL) collocation points where $c_i, b_i,\ i = 1, 2, 3$ are given by:

$$c_1 = \frac{1}{2} - \frac{\sqrt{15}}{10},\ c_2 = \frac{1}{2},\ c_3 = \frac{1}{2} + \frac{\sqrt{15}}{10},\ b_1 = \frac{5}{18},\ b_2 = \frac{4}{9},\ b_3 = \frac{5}{18}$$

and substituting into (5.2.9) and (5.2.8) we have (see also [18] and references therein)

$$\alpha_1 = hA_2, \quad \alpha_2 = \frac{\sqrt{15}h}{3}(A_3 - A_1), \quad \alpha_3 = \frac{10h}{3}(A_3 - 2A_2 + A_1), \tag{5.2.10}$$

where it is easy to see that $\alpha_1 = \mathcal{O}(h),\ \alpha_2 = \mathcal{O}(h^2),\ \alpha_3 = \mathcal{O}(h^3)$. Inserting this result into (5.2.7) yields a straight-forward method $\exp(\Omega^{[6]})$.

The computational cost of exponential of matrices The matrix $A(t)$ is a relatively sparse matrix and using commutators in (5.2.7) reduces this sparsity. As a result, the computational cost to compute $\exp(\Omega^{[6]}$ is considerably higher than to compute $\exp(\widetilde{A})$, where $\widetilde{A} = h\sum_i \beta_i A_i$ (with constants β_i) denotes a linear combination of $A(t)$ evaluated at different instants. This becomes obvious when examining the structure of the symplectic matrices

$$E_1 = \exp\begin{pmatrix} D & B \\ C & -D^T \end{pmatrix},$$

with $B^T = B, C^T = C$, versus the also symplectic

$$E_2 = \exp\begin{pmatrix} 0 & I \\ C & 0 \end{pmatrix}, \qquad \text{or} \qquad E_3 = \exp\begin{pmatrix} 0 & 0 \\ C & 0 \end{pmatrix} = \begin{pmatrix} I & 0 \\ C & I \end{pmatrix}.$$

The exponential E_3 is trivial and the simple structure of E_2 allows to write the exponential in a simple closed form where computations can be reused and an approximation to order $2m$ can be reached at the cost of only $(m-1)\mathscr{C}$ and a symplecticity can be preserved too at an extra cost of $(2 + \frac{1}{3})\mathscr{C}$ totaling $(m + \frac{4}{3})\mathscr{C}$ (see the Appendix A.2), while this is not possible in the computation of E_1. In consequence, we find that

$$c(E_3) < c(E_2) \ll c(E_1)$$

and we will look for composition methods that require mostly (symplectic) exponentials of matrices with structure E_3 while keeping the number of exponentials of the class E_2 to a minimum. This task requires a profound analysis of the Lie algebra associated to the Hill's equation.

Commutator-free Magnus integrators A standard way to avoid the computation of commutators in (5.2.7) which give rise to dense matrices E_1 is given by commutator-free (CF) Magnus integrators [1, 22]. A sixth-order example is the following composition

$$\exp\left(\Omega^{[6]}\right) = \prod_{i=1}^{s} \exp\left(\sum_{k=1}^{3} x_{i,k}\, \alpha_k\right) = \prod_{i=1}^{s} \exp\left(h\gamma_i \begin{pmatrix} 0 & \delta_i I \\ C_i & 0 \end{pmatrix}\right),$$

where the coefficients $x_{i,k}$ satisfy a set of polynomial equations. Here, C_k are linear combinations of $M(t)$ evaluated at quadrature nodes and, if $x_{i,1} \neq 0$ then $\delta_i = 1,\ \gamma_i = x_{i,1}$ otherwise $\delta_i = 0, \gamma_i = 1$. Sixth-order methods need to take $s \geq 5$ (a four-exponential sixth-order method exists, but it shows a very poor performance and it is not recommended in practice). In addition, some coefficients γ_i are negative.

5.3 Exponential symplectic methods for the Hill's equation

5.3.1 Sixth-order methods

The additional structure of the Hill's equation makes it possible to build new methods that improve the performance of the existing ones. The idea is similar to the schemes proposed for Nth-order time-dependent linear systems [6], but tailored for the Hill's equation.

The key point is to exploit the algebraic structure of α_1, α_2 and α_3[1], i.e.,

$$\alpha_1 = h\begin{pmatrix} 0 & I \\ P & 0 \end{pmatrix}, \quad \alpha_2 = h^2\begin{pmatrix} 0 & 0 \\ Q & 0 \end{pmatrix}, \quad \alpha_3 = h^3\begin{pmatrix} 0 & 0 \\ R & 0 \end{pmatrix},$$

where, in the case of the GL quadrature rule (5.2.10) we have that

$$P = -M_2, \quad hQ = -\frac{\sqrt{15}}{3}(M_3 - M_1), \quad h^2R = -\frac{10}{3}(M_3 - 2M_2 + M_1).$$

Since α_2 and α_3 are nilpotent matrices of degree two, they can be exponentiated trivially,

$$\exp(x\alpha_2 + y\alpha_3) = I + x\alpha_2 + y\alpha_3, \quad x, y \in \mathbb{C}$$

and the exponential of α_1 has a considerably lower computational cost than for a full matrix. In addition, we observe that

$$[\alpha_2, \alpha_3] = [23] = 0 \qquad \text{and} \qquad [212] = h^5\begin{pmatrix} 0 & 0 \\ 2Q^2 & 0 \end{pmatrix},$$

i.e., $[212]$ is also nilpotent with similar structure to α_2 and α_3. Since the exponentials of matrices that contain α_1 will be the most costly part of the new schemes, we analyze new compositions by the number of such exponentials involved.

[1] Recall that the α_{i+1} correspond to the ith derivative of $A(t)$, see (5.2.6).

One-α_1-exponential method In [6], the following fourth-order composition was proposed

$$\Phi_1^{[4]} = \exp\left(\frac{1}{24}(2\alpha_2 + \alpha_3)\right) \exp(\alpha_1) \exp\left(\frac{1}{24}(-2\alpha_2 + \alpha_3)\right).$$

As we have just established, [212] can be added to the first and last exponentials without significantly increasing the cost and furthermore

$$[1112] = \begin{pmatrix} W & 0 \\ 0 & -W^T \end{pmatrix},$$

with $W = h^5(3QP + PQ)$ is very small and its exponential can be approximated with two products and one inversion for the symplectic case ($QP = (PQ)^T$), i.e., cost $(2 + \frac{4}{3})\mathscr{C}$ by

$$\exp([1112]) = \begin{pmatrix} \Lambda & 0 \\ 0 & \Lambda^{-T} \end{pmatrix} + \mathcal{O}(h^{15}),$$

where $\Lambda = I + W + \frac{1}{2}W^2$. The computation of the inverse ensures symplecticity. Using these observations, we propose the composition

$$\begin{aligned}\Phi_1^{[6]} = \exp(x_6[1112]) \exp(x_3\alpha_2 + x_4\alpha_3 + x_5[212]) \exp(x_1\alpha_1 + x_2\alpha_3) \\ \times \exp(-x_3\alpha_2 + x_4\alpha_3 + x_5[212]) \exp(x_6[1112]).\end{aligned}$$

There is only one solution for the coefficients x_i given by

$$x_1 = 1, \quad x_2 = \frac{1}{20}, \quad x_3 = \frac{1}{12}, \quad x_4 = \frac{1}{60}, \quad x_5 = -\frac{1}{2880}, \quad x_6 = \frac{1}{720}.$$

It is important to remark that, while the cost of $\exp(x_6[1112])$ is $(2 + \frac{4}{3})\mathscr{C}$, in the flow of the algorithm this matrix is multiplied with a dense matrix Φ (i.e. $\exp(x_6[1112])\Phi$) which increases the total cost to $(2 + \frac{4}{3})\mathscr{C} + 4\mathscr{C}$. The computational effort can be reduced by generalizing the (first same as last) FSAL property: We concatenate two steps, the last exponential of one step and the first one in the following one[2], and since the exponents are small, we commit an error $e^{h^5V}e^{h^5W} = e^{h^5(V+W)+\mathcal{O}(h^{10})}$ and count the cost of only one exponential $\exp(x_6[1112])$ per step.

The total cost per step is $27 + \frac{2}{3}$,

$$c(\Phi_1^{[6]}) = \underbrace{6 + \frac{4}{3}}_{c(\exp([1112])\Phi} + \underbrace{(1+2)+2}_{2\,c(\exp(\alpha_2+\alpha_3+[212]))\Phi} + \underbrace{(7+\frac{1}{3})+8}_{c(\exp(\alpha_1+\alpha_2+\alpha_3))\Phi},$$

where the $c(\exp(\alpha_2 + \alpha_3 + [212]))\Phi = 1 + 2$ comes from one product for [212] (is computed only once and stored to be used in the second one) and a matrix multiplication with a sparse matrix and $c(\exp(\alpha_1 + \alpha_2 + \alpha_3))\Phi = (7 + \frac{1}{3}) + 8$ stems from an approximation to order 13 in (A.2.1) (5 products) and preservation of symplecticity ($2+\frac{1}{3}$ products).

[2]For the usual FSAL property, the two exponentials are identical and can be concatenated without error.

Two-α_1-exponential method The next composition has enough parameters to reach order six,

$$\Phi_2^{[6]} = \exp\left(x_1\alpha_2 + x_2\alpha_3 + x_3\left[\alpha_2,\alpha_1,\alpha_2\right]\right)\exp\left(x_4\alpha_1 + x_5\alpha_2 + x_6\alpha_3\right) \\ \times \exp\left(x_4\alpha_1 - x_5\alpha_2 + x_6\alpha_3\right)\exp\left(-x_1\alpha_2 + x_2\alpha_3 + x_3\left[\alpha_2,\alpha_1,\alpha_2\right]\right).$$

There is only one solution for the coefficients x_i,

$$x_1 = \frac{1}{60},\ x_2 = \frac{1}{60},\ x_3 = \frac{1}{43200},\ x_4 = \frac{1}{2},\ x_5 = \frac{2}{15},\ \ x_6 = \frac{1}{40},$$

and the composition takes the form

$$\Phi_2^{[6]} = \begin{pmatrix} I & 0 \\ hC_4 & I \end{pmatrix}\exp\left(\frac{h}{2}\begin{pmatrix} 0 & I \\ C_3 & 0 \end{pmatrix}\right)\exp\left(\frac{h}{2}\begin{pmatrix} 0 & I \\ C_2 & 0 \end{pmatrix}\right)\begin{pmatrix} I & 0 \\ hC_1 & I \end{pmatrix},$$

where $C_i,\ i = 1,2,3,4$ are linear combinations of $M(t)$ evaluated in a set of quadrature points and C_1, C_4 additionally contain one product of such linear combinations. If the sixth-order Gaussian quadrature rule is used, the method given in Table 5.1 is obtained, but any other quadrature rule of order six or higher can also be used.

The total cost preserving symplecticity and up to order 13 ($7 + \frac{1}{3}$ products) in (A.2.1) is

$$c(\Phi_2^{[6]}) = \underbrace{(1+2)}_{c(\exp(\alpha_2+\alpha_3+[212]))\Phi} + \underbrace{2((7+\frac{1}{3})+8)}_{2\,c(\exp(\alpha_1+\alpha_2+\alpha_3))\Phi} = 33 + \frac{2}{3}.$$

Here, we exploited that the last exponential can be concatenated with the first one in the following step at no extra cost, and it is not counted, a property which we call *first commutes with last* (FCWL).

Three-α_1-exponential method We analyze now the following composition with three exponentials

$$\Phi_3^{[6]} = \exp\left(x_1\alpha_1 + x_2\alpha_2 + x_3\alpha_3\right)\exp\left(x_4\alpha_1 + x_5\alpha_3 + x_6\left[\alpha_2,\alpha_1,\alpha_2\right]\right) \\ \times \exp\left(x_1\alpha_1 - x_2\alpha_2 + x_3\alpha_3\right).$$

There are only two solutions, one of them with $x_1, x_4 > 0$ given by:

$$x_1 = \frac{1}{10}(5-\sqrt{5}), \qquad x_2 = \frac{1}{24}(5-\sqrt{5}), \qquad x_3 = \frac{1}{60}(5-\sqrt{5}),$$
$$x_4 = \frac{1}{\sqrt{5}}, \qquad x_5 = \frac{1}{60}(-5+2\sqrt{5}), \quad x_6 = \frac{1}{8640}(-11+5\sqrt{5}).$$

This composition has the structure

$$\exp\left(\gamma_3 h\begin{pmatrix} 0 & I \\ C_3 & 0 \end{pmatrix}\right)\exp\left(\gamma_2 h\begin{pmatrix} 0 & I \\ C_2 & 0 \end{pmatrix}\right)\exp\left(\gamma_1 h\begin{pmatrix} 0 & I \\ C_1 & 0 \end{pmatrix}\right),$$

where $\gamma_1 = \gamma_3 = x_1,\ \gamma_2 = x_4$.

The total cost is

$$c(\Phi_3^{[6]}) = \underbrace{1 + (7+\frac{1}{3}) + 8}_{c(\exp(\alpha_1+\alpha_3+[212]))\Phi} + \underbrace{2((7+\frac{1}{3})+8)}_{2\,c(\exp(\alpha_1+\alpha_2+\alpha_3))\Phi} = 47.$$

5.3.2 Eigth-order methods

We extend the previous analysis to build eight-order methods. The Magnus expansion up to order eight for a general linear system using the Lie algebra generated by $\{\alpha_1, \alpha_2, \alpha_1, \alpha_4\}$ reads

$$\begin{aligned}\Omega^{[8]} = \alpha_1 &+ \frac{1}{12}\alpha_3 - \frac{1}{12}[12] + \frac{1}{240}[23] + \frac{1}{360}[113] - \frac{1}{240}[212] + \frac{1}{720}[1112] \\ &- \frac{1}{80}[14] - \frac{1}{1344}[34] - \frac{1}{2240}[124] + \frac{1}{6720}[223] + \frac{1}{6048}[313] - \frac{1}{840}[412] \\ &+ \frac{1}{6720}[1114] - \frac{1}{7560}[1123] + \frac{1}{4032}[1312] + \frac{11}{60480}[2113] - \frac{1}{6720}[2212] \\ &- \frac{1}{15120}[11113] - \frac{1}{30240}[11212] + \frac{1}{7560}[21112] - \frac{1}{30240}[111112].\end{aligned}$$

Further significant simplifications occur for the Hill's problem since the following commutators cancel

$$[23] = [34] = [24] = [223] = [1123] = [2212] = 0.$$

In addition, the matrices $[212]$, $[313]$, $[412]$ are nilpotent and these elements can be included in the exponents at a minor extra cost. We can also use other nilpotent matrices of even order, such as $[312]$, if they are distributed into the composition skew-symmetrically (the element $[213]$ is not included because of the Jacobi identity $[123] + [231] + [312] = 0$ and $[123] = 0$ so that $[312] = [213]$).

We consider a number of compositions with enough independent parameters to solve the order conditions using four and five exponentials. In each case, there are many different possible compositions leading in some cases to complex-valued solutions, to one or several real solutions or to families of solutions in terms of free parameters. In the following we present the method that provided the best performance in practice among the methods studied.

Five-α_1-exponential method The following composition

$$\begin{aligned}\Phi_5^{[8]} = \exp\,&(x_7\alpha_1 + x_8\alpha_2 + x_9\alpha_3 + x_{10}\alpha_4 + x_{11}[212]) \\ &\exp\,(x_{12}\alpha_2 + x_{13}\alpha_3 + x_{14}\alpha_4 + x_{15}[212] + x_{16}[313]) \\ &\exp\,(x_3\alpha_1 + x_4\alpha_2 + x_5\alpha_3 + x_6\alpha_4) \\ &\exp\,(x_1\alpha_1 + x_2\alpha_3) \\ &\exp\,(x_3\alpha_1 - x_4\alpha_2 + x_5\alpha_3 - x_6\alpha_4) \\ &\exp\,(-x_{12}\alpha_2 + x_{13}\alpha_3 - x_{14}\alpha_4 + x_{15}[212] + x_{16}[313]) \\ &\exp\,(x_7\alpha_1 - x_8\alpha_2 + x_9\alpha_3 - x_{10}\alpha_4 + x_{11}[212])\end{aligned}$$

has two real valued solutions at order six, and the one with smallest coefficients is

$$\begin{aligned}x_1 &= 0.6403363286379515, & x_2 &= 0.0433501199827269, \\ x_3 &= -0.4017895263297271, & x_4 &= -0.1170180583697493, \\ x_5 &= -0.1038563759039891, & x_6 &= -0.0376728349617945 \\ x_7 &= 0.5816213620107513, & x_8 &= 0.2609350592183406, \\ x_9 &= 0.1157777422250884, & x_{10} &= 0.0506748377294480, \\ x_{11} &= -0.0000936846387697, & x_{12} &= -0.0127292796833454, \\ x_{13} &= 0.0080702403542039, & x_{14} &= -0.0017487133111753, \\ x_{15} &= -0.0000928250351798, & x_{16} &= 0.0001835812673590\end{aligned}$$

The computational cost of this method is: $2\mathscr{C}$ to compute [212], [313], $2 \times 2\mathscr{C}$ to compute the products of nilpotent matrices, and $5 \times ((7 + \frac{1}{3}) + 8)\mathscr{C}$ to approximate the exponentials up to 12th-order, so the total cost is $82 + \frac{2}{3}\mathscr{C}$.

The computational cost of the different 8th-order methods we considered is about 70-85 $\mathscr{C}$ where the exponentials are approximated up to 12th order), it is approximately half the cost of the commutator-free methods of the same order obtained in [1] which require 11 exponentials, that is $168 + \frac{2}{3}\mathscr{C}$.

For the implementation of the eighth-order methods, the derivatives α_i are replaced by momentum integrals using the substitution rules from (2.4.29)

$$\begin{array}{rclcrcl} \alpha_1 &=& \frac{3}{4}\,(3A^{(0)} - 20A^{(2)}), & & \alpha_3 &=& -15\,(A^{(0)} - 12A^{(2)}), \\ \alpha_2 &=& 15\,(5A^{(1)} - 28A^{(3)}), & & \alpha_4 &=& -140\,(3A^{(1)} - 20A^{(3)}), \end{array} \tag{2.4.29}$$

and the $A^{(i)}$, $i = 1, 2, 3, 4$ are approximated with a standard rule following (5.2.9). For instance, using the 8th order Gauss-Legendre quadrature rule, we obtain

$$c_i = \frac{1}{2} - v_i,\ c_{5-i} = \frac{1}{2} + v_i, \quad b_i = b_{5-i} = \frac{w_i}{2},$$

$i = 1, 2$, and where

$$v_1 = \frac{1}{2}\sqrt{\frac{3 + 2\sqrt{6/5}}{7}},\ \ v_2 = \frac{1}{2}\sqrt{\frac{3 - 2\sqrt{6/5}}{7}}, \quad w_1 = \frac{1}{2} - \frac{1}{6}\sqrt{\frac{5}{6}},\ \ w_2 = \frac{1}{2} + \frac{1}{6}\sqrt{\frac{5}{6}}.$$

Letting $S_1 = A_1 + A_4$, $S_2 = A_2 + A_3$, $R_1 = A_4 - A_1$, $R_2 = A_3 - A_2$, we reach the expression

$$A^{(0)} = \frac{1}{2}\left(w_1 S_1 + w_2 S_2\right), \qquad A^{(1)} = \frac{1}{2}\left(w_1 v_1 R_1 + w_2 v_2 R_2\right),$$

$$A^{(2)} = \frac{1}{2}\left(w_1 v_1^2 S_1 + w_2 v_2^2 S_2\right), \quad A^{(3)} = \frac{1}{2}\left(w_1 v_1^3 R_1 + w_2 v_2^3 R_2\right).$$

5.4 Numerical examples

We now study the performance of the new methods on the numerical integration of different Hill's equations. The number of $r \times r$ matrix-matrix products, k, to compute the product $\Psi\Phi$, where Ψ denotes the method and Φ the fundamental matrix solution, is given in parenthesis as $k\,\mathscr{C}$, and it is taken as the cost of the method. We use 12th-order symplectic approximations to evaluate the exponentials E_2 at the cost of $(7 + \frac{1}{3})\,\mathscr{C}$ per exponential. The following methods are considered:

- $\mathrm{RK}_7^{[6]}$: A 7-stage explicit (non symplectic) RK method that only requires 3 new evaluations of $A(t)$ per step given in [28, p. 203-205] (14 $\mathscr{C}$).
- $\mathrm{RKGL}^{[6]}$: The 3-stage implicit symplectic RKGL method (we count an average of 6 iterations per step for a total cost of 36 $\mathscr{C}$).

- $\text{RKN}_{11}^{[6]}$: The 11-stage explicit symplectic RKN method given in [23] (22 $\mathscr{C}$). This method requires 11 new evaluations of $A(t)$ per step that are not counted into the cost.
- $\Phi_5^{[6]}$: The five-exponential commutator-free Magnus integrator from [22] ($(76 + \frac{2}{3})\mathscr{C}$).
- $\Phi_i^{[6]}$: The new $\Phi_i^{[6]}$, $i = 1, 2, 3$ methods ($(27 + \frac{2}{3})\mathscr{C}$, $(33 + \frac{2}{3})\mathscr{C}$, $47\mathscr{C}$, respectively).
- $\Phi_5^{[8]}$: The new $\Phi_5^{[8]}$ method ($(82 + \frac{2}{3})\mathscr{C}$).

5.4.1 The Mathieu equation

As a test bench to compare the performance of the methods we employ Mathieu's equation

$$x'' + \left(\omega^2 + \varepsilon \cos(2t)\right) x = 0.$$

In a first experiment to test the qualitative behavior, we compute the fundamental matrix solution for one period (at $T = \pi$) with the identity matrix as the initial conditions using the time step $h = \pi/10$ for a set of values ω (parametric resonances occur about $\omega = k$, $k \in \mathbb{N}$. We let $\varepsilon = 5$ and $\omega = j\Delta\omega = j\frac{1}{200}$, $j = 0, 1, \ldots, 1020$, and consider the following methods: $\text{RK}_7^{[6]}$, $\text{RKGL}^{[6]}$, $\text{RKN}_{11}^{[6]}$ and $\Phi_2^{[6]}$.

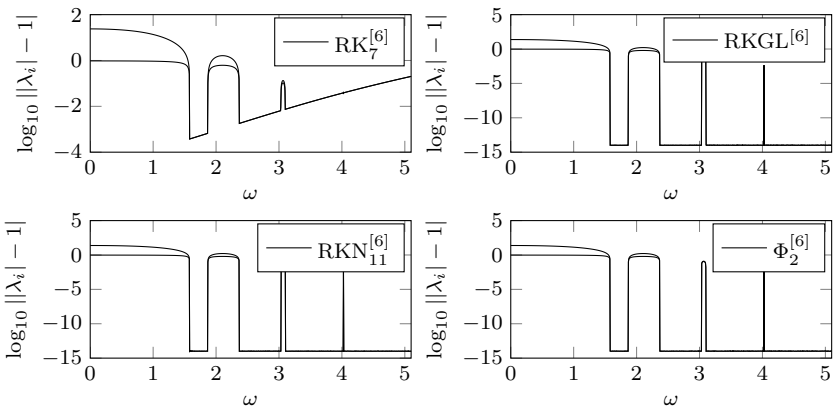
We compute the eigenvalues and their distance to the unit circle, $|\lambda_1| - 1$ and $|\lambda_2| - 1$. The results, in logarithmic scale[3], are shown in Figure 5.1a (a zoom about $\omega = 5$, taking smaller values of $\Delta\omega$, is shown in Figure 5.1b together with the reference solution, dashed lines, that was computed numerically to very high accuracy). From the results, it is clear that preservation of symplecticity is crucial. The non symplectic $\text{RK}_7^{[6]}$ method is about 1.5–2.5 times faster than the symplectic ones for the same time step, but requires a much smaller time step to reach similar accuracy.

To gain insight on how the accuracy depends on the frequency of the oscillatory solution, we repeat the same numerical experiment for $h = \pi/20$ and measure the L_1-norm of the error in the fundamental matrix solution (where the reference solution was obtained with a sufficiently small time step). The results are shown in Figure 5.2. We observe that the new exponential method shows a smaller error growth with ω. As a result, the same time-dependent function evaluations can be used for many different values of ω in the exponential method while the $\text{RKGL}^{[6]}$ and $\text{RKN}_{11}^{[6]}$ methods should be used with smaller time steps as ω increases and the time-dependent functions need to be recalculated.

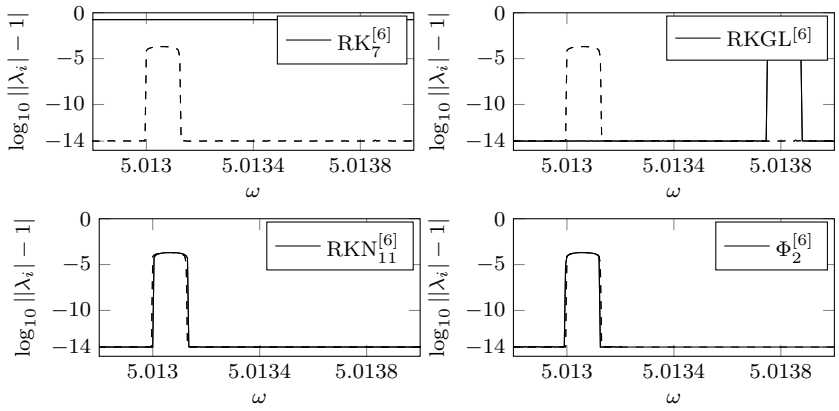
We now analyze the efficiency of the integrators. We choose a moderate value of $\omega = 5$, and $\varepsilon = 1$ and integrate for $t \in [0, \pi]$. We measure the L_1-norm of the error of the fundamental matrix at the final time versus the computational cost (in units of $\mathscr{C}$). The results are shown in Figure 5.3, where we observe that the new sixth-order method with two exponentials is superior for all accuracies.

For very small values of ω (non oscillatory problems) the symplectic $\text{RKN}_{11}^{[6]}$ integrator is the method of choice but as ω grows and the system becomes oscillatory, the new methods show the best performance.

[3] We compute $\log ||\lambda_i| - 1 + \delta|$ with $\delta = 10^{-14}$ to avoid singularities.



(a) Each line corresponds to one of the two eigenvalues.



(b) Zoom of Figure 5.1a about $\omega = 5$. The dashed line correspond to the reference solution obtained numerically to sufficiently high accuracy

Figure 5.1: Distance of the eigenvalues to the unit circle versus ω of the explicit non-symplectic method $\mathrm{RK}_7^{[6]}$, and the symplectic methods the $\mathrm{RKGL}^{[6]}$, $\mathrm{RKN}_{11}^{[6]}$ and $\Phi_2^{[6]}$.

5.4.2 Matrix Hill's equation

Let us now consider the following matrix Hill's equation

$$x'' + (A + B_1 \cos(2t) + B_2 \cos(4t))\, x = 0$$

with $A, B_1, B_2 \in \mathbb{R}^{r\times r}$. We take $A = r^2 I + D$ where D is the Pascal matrix

$$D_{1i} = D_{i1} = 1, \quad i = 1, \dots, r, \qquad D_{ij} = D_{i-1,j} + D_{i,j-1}, \quad 1 < i,j \le r,$$

which is a symmetric positive definite dense matrix. We set $B_1 = \varepsilon I$, $B_2 = \frac{1}{10}\varepsilon I$, $\varepsilon = r$ and $\varepsilon = \frac{1}{10} r$ and compute the solution for $r = 5$ and $r = 7$.

The error of the fundamental matrix solution is measured in the L_1-norm. The results are shown in Figure 5.4.

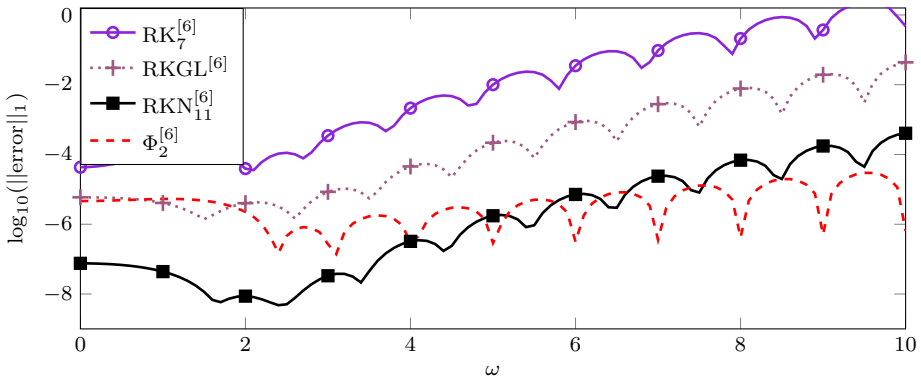


Figure 5.2: Error of the fundamental matrix solution at one period of integration versus ω with $\varepsilon = 5$.

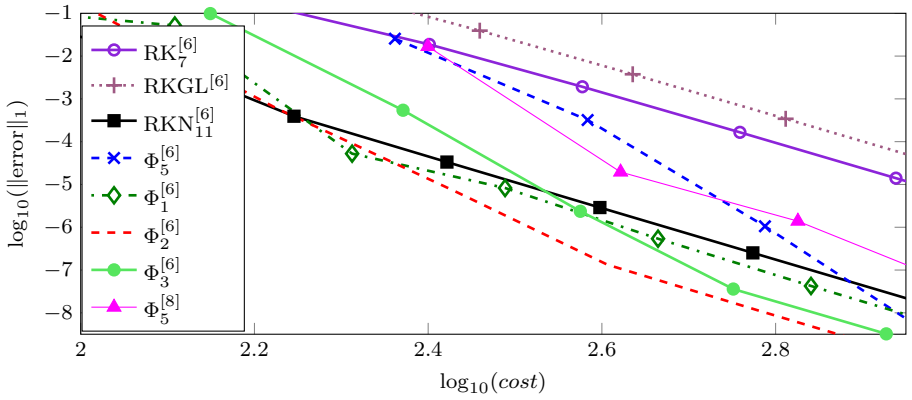


Figure 5.3: Error at the final time $t = \pi$ versus the number of products $\mathscr{C}$ in double logarithmic scale.

We clearly observe that as r increases (oscillatory problem) as well as ε decreases (approaching the autonomous problem) the new exponential methods show the best performance. The eighth-order method is only superior when very accurate results are desired being an open problem to know if there exist other composition leading to more efficient methods that are superior for medium to high accuracy.

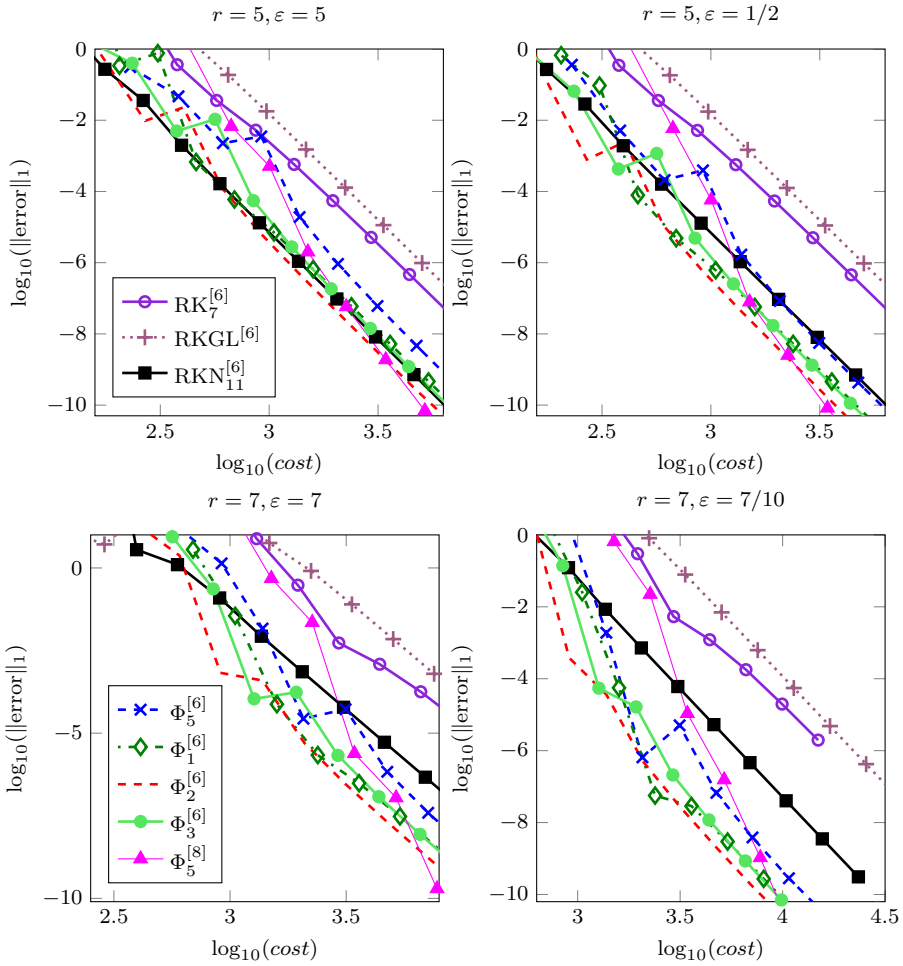


Figure 5.4: Efficiency plot for Example 5.4.2 The legend has been split over two panels.

5.4.3 The damped Mathieu equation

The next problem is the following Mathieu equation with extra small positive damping and forcing term,

$$x'' + \delta x' + \left(\omega^2 + \varepsilon \cos(2t)\right) x = \kappa \cos(2t) \qquad x(0) = 1, x'(0) = 0,$$

where δ, κ are small parameters. One can write this equation as a first order system of equations

$$\frac{d}{dt} \begin{Bmatrix} x \\ x' \end{Bmatrix} = \begin{pmatrix} 0 & 1 \\ -\left(\omega^2 + \varepsilon \cos(2t)\right) & 0 \end{pmatrix} \begin{Bmatrix} x \\ x' \end{Bmatrix} + \begin{Bmatrix} 0 \\ -\delta x' + \kappa \cos(2t) \end{Bmatrix}$$

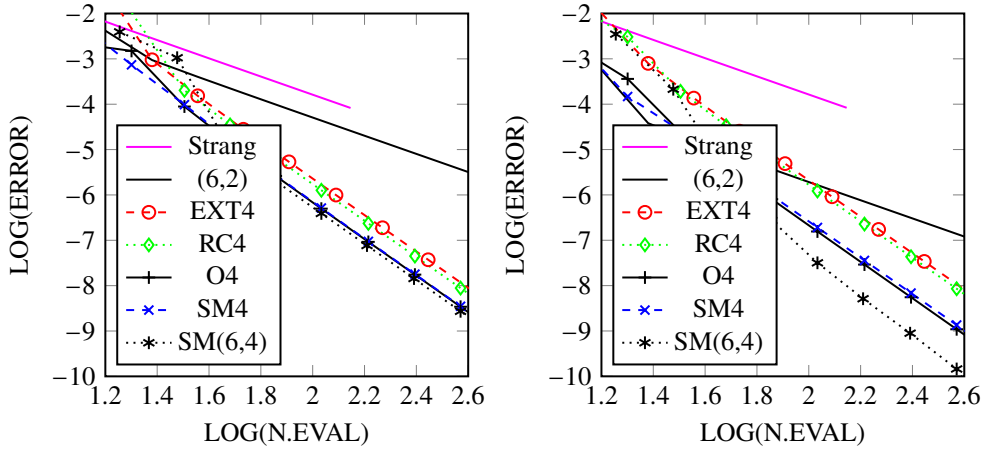


Figure 5.5: Error versus number of evaluations of $\Phi_A^{[h]}$ for the numerical integration of the damped Mathieu equation at $t = \pi$ for $\kappa = 1$ and $\delta = \frac{1}{10}$ (left panel) and $\delta = \frac{1}{100}$ (right panel).

and can take the time as a new coordinate and split the system as follows

$$\frac{d}{dt}\begin{Bmatrix} x \\ x' \end{Bmatrix} = \begin{pmatrix} 0 & 1 \\ -(\omega^2 + \varepsilon\cos(2t_1)) & 0 \end{pmatrix}\begin{Bmatrix} x \\ x' \end{Bmatrix}, \qquad \frac{dt_1}{dt} = 1,$$

and

$$\frac{d}{dt}\begin{Bmatrix} x \\ x' \end{Bmatrix} = \begin{Bmatrix} 0 \\ -\delta x' + \kappa\cos(2t_1) \end{Bmatrix}.$$

The first part above is Mathieu's equation and its solution, in general, can not be expressed in closed form and thus has to be solved numerically. Therefore, efficient numerical schemes for solving the damped Mathieu equation require highly accurate integrators for the Mathieu equation. Here we follow the same steps as for Example 1 in Chapter 3. Since the time is frozen, the evaluation for the perturbation is immediate. We show that the sixth-order two-exponential method $\Phi_2^{[6]}$ derived in this chapter shows high accuracy and we use this method for solving this part to get an efficient solution of the whole system. Now, we analyze the efficiency of the splitting schemes obtained for the perturbed systems derived in Chapter 3. We choose $\omega = 5$, and $\varepsilon = 1$ and integrate for $t \in [0, \pi]$. To measure the performance of the methods, we compute the error of each method at the end of the time integration (we take as the exact solution a numerical approximation computed to a high precision) and we take as the cost of the method the number of evaluations of Mathieu part ($\Phi_A^{[h]}$ considered in Chapter 3) which usually carries most of the computational cost. We measure the L_1-norm of the error at the final time versus the computational cost. The results are shown in Figure 5.5. For this problem, the splitting methods with complex coefficients perform better than splitting methods with real coefficients and than extrapolation method as shown in the figure.

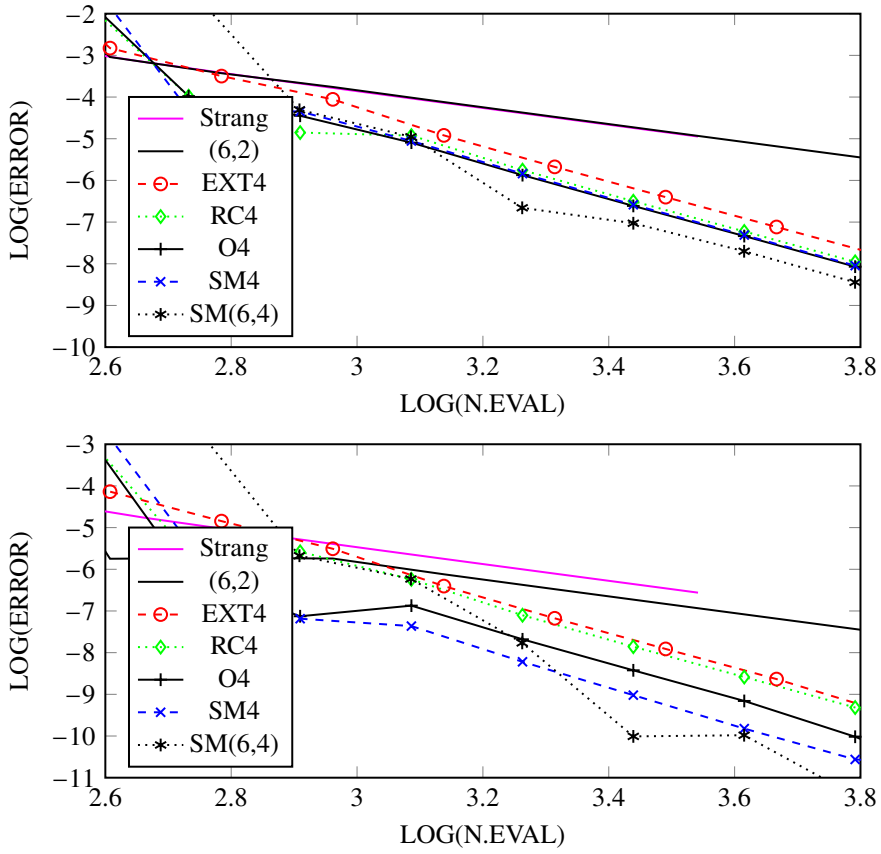


Figure 5.6: Error versus number of evaluations of $\Phi_A^{[h]}$ for the numerical integration of the damped Mathieu equation at $t = 20\pi$ for $\alpha = \frac{1}{10}$ (top panel) and $\alpha = \frac{1}{100}$ (bottompanel).

5.4.4 The non-linear Mathieu equation

The last test-problem is the following non-linear Mathieu equation

$$x'' + \left(\omega^2 + \varepsilon \cos(2t)\right) x + \alpha x^3 \cos(2t) = 0 \qquad x(0) = 1, x'(0) = 0,$$

where α is small parameter. We follow similar steps as for the damped Mathieu equation. We write this equation as a first order system of equations and take the time as a new coordinate and split the system as done for the damped Mathieu equation. We choose values of parameters as $\omega = 5$, and $\varepsilon = 1$ and compute the error at the final time $t \in [0, 20\pi]$ by using the same splitting methods as in the damped case. The results are shown in Figure 5.6. The new splitting methods with complex coefficients show better performance than the lower order splitting methods with real coefficients or extrapolation.

Chapter 6

CONCLUSIONS

We summarize the contributions and conclusions of each chapter and end with a brief introduction to a future line of work for which this thesis serves as foundation.

6.1 Non-reversible systems

We have considered the numerical integration of non-autonomous separable parabolic equations using high order splitting methods with complex coefficients. A straightforward application of splitting methods with complex coefficients to non-autonomous problems require the evaluation of the time-dependent functions in the operators at complex times, and the corresponding flows in the numerical scheme are, in general, not well conditioned. To avoid this trouble, in this work we consider a class of methods in which one set of the coefficients belong to the class of real and positive numbers. Taking the time as a new coordinate and an appropriate splitting of the system allows us to build numerical schemes where all time-dependent operators are evaluated at real values of the time. This technique shows a great interest for perturbed systems, and this problem is analyzed in more detail. In this case, the flow of the dominant part has to be advanced using the real coefficients. We have analyzed the algebraic structure of the problem and the cost of the algorithms in order to build efficient high order methods, and some few new methods are reported as an illustration. Higher order and more efficient methods require a considerably deeper analysis, and methods belonging to this class as well as more general methods are being considered by the authors of Ref. [14] and will be published elsewhere. Several numerical examples are considered where the good performance of this class of methods is shown . We have shown that splitting methods with complex coefficients can also be used on non-autonomous non-linear parabolic problems and they can show a good performance so, high order and more efficient schemes following the guidelines presented in this work can be of great interest. This class of method has proven to be successful for some partial differential equations [91]. We must also remark that order reductions are expected for problems with Dirichlet or Newman boundary conditions on bounded domains,

and the performance of high order methods on these problems diminishes, being an interesting problem that needs further investigation.

6.2 Exponential of perturbed matrices

We have proposed a new recursive algorithm based on splitting methods for the computation of the exponential of perturbed matrices which can be written as the sum $A = D + \varepsilon B$ of a sparse and efficiently exponentiable matrix D with sparse exponential e^D and a dense matrix εB which is of small norm in comparison with D. The algorithm is based on the scaling and squaring technique where the Padé or Taylor methods to compute the exponential of the scaled matrix have been replaced by appropriate splitting methods tailored for this class of matrices. An important feature of splitting methods for perturbed problems is that the error is a sum of a local error of order $\mathcal{O}(\varepsilon)$ plus a global error of order $\mathcal{O}(\varepsilon^2)$ and this allows to build new methods with high performance when low to medium accuracy is desired. Our algorithm entangles the splitting methods with conventional squarings in a way that allows to save computational cost and still leaves some free parameters for optimization. By using a recursive formulation, the dominant computational cost arising from the computation of dense-matrix products is minimized while posing only modest memory requirements and the free parameters are chosen to exploit the smallness of the perturbed matrix B. Theoretical results on local error and error propagation for splitting methods are verified by numerical experiments and precise notions of the smallness of the perturbation and tolerance requirements are derived. A clear improvement over existing and highly optimized Padé methods was observed for small perturbations when low to medium precision is sought.

6.3 Symplectic integrators for the matrix Hill's equation

We have studied the numerical integration of the matrix Hill's equation using methods that accurately reproduce the parametric resonances of the exact solution. We are mainly interested in the Hamiltonian case which is the most frequent one in practice, namely when the Hill's equations originate from a Hamiltonian function. In this case the fundamental matrix solution is a symplectic matrix and we illustrate the importance of the preservation of this property by the numerical integrators.

We have presented new symplectic sixth- and eighth-order symplectic exponential integrators that are tailored to the matrix Hill's equation. Exponential integrators usually show high accuracy for stiff and oscillatory problems but at a relatively high computational cost that made them uncompetitive versus existing explicit sympletic Runge-Kutta-Nyström methods when applied to the matrix Hill's equation. However, we show that a class of matrix exponentials can be very efficiently approximated while preserving the symplectic structure, and we have built new families of methods based on exponentials of this type. Several sixth- and eighth-order methods using compositions of one to five exponentials are considered. The numerical experiments showed the high performance of the new methods. Among the methods obtained, a sixth-order two-exponential method showed the best performance. The eighth-order methods obtained using different compositions had large coefficients that turned into relatively large truncation errors and, in addition, required a higher order truncation for the approximation

to the exponentials. It is left as an open question if different eighth-order compositions with small coefficients exist which could show a higher performance.

The methods obtained in this chapter are used for the numerical integration of the weekly damped Hill's equation with a week non-linear interaction as the basic method to solve perturbed time-dependent linear problems as shown in [4, 21, 90].

Appendix A

ALGEBRAIC TOOLS

A.1 Further approaches

This appendix uses the notation of Chapter 4. We collect results on approaches that are successful in the context of splittings for ordinary differential equations, however, have been found less efficient on the numerical experiments than the methods presented before.

A.1.1 On processing

A basic property of the adjoint action,

$$e^P Y e^{-P} = e^{\mathrm{ad}_P} Y = Y + [P, Y] + \frac{1}{2}[P, [P, Y]] + \cdots$$

together with the cheap computability of the commutator $[D, B] = DB - BD$ motivates the use of *processing techniques*, well-known for the numerical integration of differential equations, to eliminate error terms. The idea is now based on the observation that $(XYX^{-1})^N = XY^NX^{-1}$ and essentially corresponds to a change of basis in which the error propagation (recall that large s can be regarded as a (long-) time integration using a small time-step $h = 1/2^s$) is expected to be less severe.

The modified Strang algorithm (4.4.6) has leading error proportional to

$$[B, [D, B]], \quad [B, [D, [D, [D, B]]]], \quad [D, [D, [B, [D, B]]]].$$

The first two of which can be eliminated using a processor with

$$P = \alpha\varepsilon h^2 [D, B] + \beta\varepsilon h^4 [D, [D, [D, B]]],$$

thus motivating the ansatz

$$e^{\alpha\varepsilon h^2[D,B]+\beta\varepsilon h^4[D,[D,[D,B]]]}\tilde{Y}_s e^{-\alpha\varepsilon h^2[D,B]-\beta\varepsilon h^4[D,[D,[D,B]]]}.$$

The norm of the outer exponents is small and a low order Padé approximation, say $r_2(P)$, usually provides sufficient accuracy. Therefore, at the expense of one exponential, one multiplication and one inversion (which is performed together with the multiplication, as for the Padé methods, $(\mathscr{B}\mathscr{D})\tilde{\mathscr{B}}^{-1}$), we get two free parameters, α, β. Using the kernel $\tilde{Y}_0$, we reach order (6,4), whereas $\tilde{Y}_1$ is sufficient for order (10,4) and (6,6,4), see Table A.1.

A.1.2 More exponentials

For problems where complex coefficients a_j lead to a substantial increase in computational complexity (e.g., when $A, B \in \mathbb{R}^{N\times N}$) or matrix commutators are not desirable, it could be advantageous to allow negative values for some b_j.

A first example is the four-stage method

$$S_4^{[4]} = \mathscr{D}_{ha_1}\mathscr{B}_{hb_1}\mathscr{D}_{ha_2}\mathscr{B}_{hb_2}\mathscr{D}_{ha_2}\mathscr{B}_{hb_1}\mathscr{D}_{ha_1}. \tag{A.1.1}$$

This scheme requires two exponentials, two products and has two free parameters which can produce a fourth-order method with real coefficients a_j, b_j, known as triple jump [41, 95, 106], see Table A.1.

Another product is necessary to compute the six-stage composition

$$S_{(6,4)}^{[6]} = \mathscr{D}_{ha_1}(\mathscr{B}_{hb_1}\mathscr{D}_{ha_2}\mathscr{B}_{hb_1})\mathscr{D}_{ha_3}\mathscr{B}_{hb_2}\mathscr{D}_{ha_3}(\mathscr{B}_{hb_1}\mathscr{D}_{ha_2}\mathscr{B}_{hb_1})\mathscr{D}_{ha_1}.$$

Three free parameters are sufficient to construct (6,4) methods, however, with complex time-steps. The real-valued fourth-order method minimizing the error at $\mathcal{O}(\varepsilon h^5)$ can be found in Table A.1. An additional stage with a grouping similar to the modified splittings,

$$S_{(6,4)}^{[7]} = \mathscr{D}_{ha_1}(\mathscr{B}_{hb_1}\mathscr{D}_{ha_2}\mathscr{B}_{hb_2}\mathscr{D}_{ha_2}\mathscr{B}_{hb_1})\mathscr{D}_{ha_3}(\mathscr{B}_{hb_1}\mathscr{D}_{ha_2}\mathscr{B}_{hb_2}\mathscr{D}_{ha_2}\mathscr{B}_{hb_1})\mathscr{D}_{ha_1},$$

requires the same number of products but has real solutions of order (6,4). Among the four real-valued solutions, the one minimizing the error at $\mathcal{O}(\varepsilon h^7)$ is printed in Table A.1. We have found that supposedly clever re-utilization of exponentials by setting b_j to be a rational multiple of an already computed exponent b_k are not competitive since – at its very best – one can save the computation of an exponential at the cost of an inversion ($b_j = -b_k$) or a matrix product ($b_j = 2b_k$), however, the direct use of the sufficiently accurate r_2-Padé method needs only one inversion.

A.1.3 Splitting for low-order Padé

Technically, the stated splitting orders assume the exact computation of all exponentials, but in practice, the cheap underlying Padé scheme r_2 has accuracy limit $\mathcal{O}(\varepsilon^3 h^3)$. Since we assumed ε to be a small parameter, comparable to h^2, it could be regarded as $\mathcal{O}(\varepsilon h^7)$. Instead of switching to the more precise r_4 method ($\mathcal{O}(\varepsilon^5 h^5)$) for the exponential $\mathscr{B}$, (using r_2 for the processor has error $\mathcal{O}(h^6\varepsilon^3)$ and is therefore sufficient), we attempt to use a free parameter to decrease the r_2-related error in $\mathscr{B}$ to $h^5\varepsilon^5$.

The procedure is based on the observation that the approximant $r_2(h\varepsilon B)$ can be expressed as a single exponential

$$r_2(h\varepsilon B) = e^{h\varepsilon B + h^3\varepsilon^3 C + \mathcal{O}(h^5\varepsilon^5)}$$

for some matrix C. Notice that the exponent can be expanded in odd powers of h since r_2 is symmetric. Now, we simply add the (unknown) matrix C to the algebra and in addition to the previous order conditions, we have to solve $\sum_{i=1}^{m} b_i^3 = 0$. It is clear that condition $b_i = 1/m$ has to be dropped and at least three exponentials $\mathscr{B}_{b_j h}$ are necessary. We embark by modifying (A.1.1) to

$$\Psi^{[4,mod]} = \mathscr{D}_{ha_1}\tilde{\mathscr{B}}_{b_1,\beta_1,\gamma_1}\mathscr{D}_{ha_2}\tilde{\mathscr{B}}_{b_2,\beta_2,\gamma_2}\mathscr{D}_{ha_2}\tilde{\mathscr{B}}_{b_1,\beta_1,\gamma_1}\mathscr{D}_{ha_1},$$

with the notation (4.4.5). Using two exponentials (inversions) and two multiplications, we have six free parameters and only one additional equation. The freedom in the parameters allows to construct real-coefficient methods of order (10,4) and alternatively, at order (8,4), a method minimizing the squared error polynomials $e_{5,2}$ at $\varepsilon^2 h^5$, see Table A.1. Experiments show that the additional cost due to a further exponential (inversion) and a multiplication renders those methods uncompetitive in comparison with the modified squarings using a higher-order method r_4 for the single exponential $\tilde{B}$ which comes at just one additional product.

A.2 Efficient symplectic approximation of E_2

This appendix uses the notation of Chapter 5. We seek a symplectic approximation to

$$E_2 = \exp\left(\gamma h \begin{pmatrix} 0 & I \\ C & 0 \end{pmatrix}\right) = \exp\left(\tau \begin{pmatrix} 0 & I \\ C & 0 \end{pmatrix}\right) = \begin{pmatrix} \sigma & \mu \\ -C\mu & \sigma \end{pmatrix} \tag{A.2.1}$$

with $\tau = \gamma h$ and

$$\sigma = \sum_{n=0}^{\infty} \frac{\tau^{2n}}{(2n)!} C^n, \qquad \mu = \sum_{n=0}^{\infty} \frac{\tau^{2n+1}}{(2n+1)!} C^n.$$

Using only m products, $C^2, C^3, \ldots, C^{m+1}$, we can compute the truncation

$$E_2^{[m]} = \begin{pmatrix} \sigma_m & \mu_m \\ -C\mu_m & \sigma_m \end{pmatrix} \text{ with } \sigma_m = \sum_{n=0}^{m+1} \frac{\tau^{2n}}{(2n)!} C^n,\ \mu_m = \sum_{n=0}^{m} \frac{\tau^{2n+1}}{(2n+1)!} C^n,$$

and commit an error of $E_2 - E_2^{[m]} = \mathcal{O}(\tau^{2m+3})$. The result is not a symplectic matrix, but can be made symplectic by adding a correction term δ_m as follows

$$E_2^{[m,s]} = \begin{pmatrix} \sigma_m & \mu_m + \delta_m \\ \nu_m & \sigma_m \end{pmatrix}$$

with δ_m a close-to-zero symmetric matrix. The symplecticity condition for $E_2^{[m,s]}$ reads

$$(E_2^{[m,s]})^T J E_2^{[m,s]} = J.$$

Since C is a symmetric matrix, σ_m and μ_m are symmetric and commute, $\sigma_m\mu_m - \mu_m\sigma_m$, and the symplectic condition simplifies to

$$\sigma_m^2 - (\mu_m + \delta_m)\nu_m = I \quad \Longrightarrow \quad \delta_m = (\sigma_m^2 - I)/\nu_m - \mu_m.$$

Hence, δ_m can be computed with a product and one inverse for a total extra cost of $(1 + \frac{4}{3})\mathscr{C}$. Summarizing, we can achieve a symplectic approximation of error $\mathcal{O}(\tau^{2m+1})$ with $m + \frac{4}{3}$ products.

Table A.1: Further splitting methods, including several exponentials and processing techniques.

$S^{[4]}$, 4 stages, order 4	2 exp, 2 prod
$a_1 = \frac{1}{6}(2 + 1/2^{1/3} + 2^{1/3})$, $b_1 = \frac{1}{3}(2 + 1/2^{1/3} + 2^{1/3})$	2 complex sol. omitted
$S^{[6]}$, 6 stages, order 4	2 exp, 3 prod
$a_1 = 0.19731107566242791631$, $a_2 = 0.38252646594731312955$, $a_3 = (1 - 2a_1 - 2a_2) = -0.079837541609741045862$, $b_1 = 0.42519341909910345071$, $b_2 = 1 - 4b_1 = -0.70077367639641380284$.	[minimizes $\mathcal{O}(\varepsilon h^5)$]
$S^{[7]}$, 7 stages, order (6,4)	2 exp, 3 prod
$a_1 = 0.35937529621978708941$, $a_2 = -0.098379231055234835826$, $a_3 = (1 - 2a_1 - 4a_2) = 0.67476633178136516448$, $b_1 = 0.67702963544760500586$, $b_2 = 1/2 - 2b_1 = -0.85405927089521001173$.	[minimizes $\mathcal{O}(\varepsilon h^7)$]
Processed $e^{xh^2[D,B]+yh^4[D,[D,[D,B]]]}\tilde{Y}_0 e^{-xh^2[D,B]-yh^4[D,[D,[D,B]]]}$	2 exp, 1 prod, 1 inv
Order (6,4)	
$a_1 = 1/2$, $\beta = -1/24$, $\gamma = 31/5760$ $x = -1/12$, $y = 1/120$.	
Processed $e^{xh^2[D,B]+yh^4[D,[D,[D,B]]]}\tilde{Y}_1 e^{-xh^2[D,B]-yh^4[D,[D,[D,B]]]}$	2 exp, 1 prod, 1 inv
Order (6,6,4)	
$a_2 = 0.25879773408334034345302 75$, $\beta = -0.00522768336458362542165392 5$, $\gamma = 0.00003295467182282037 82$, $x = -0.02303276685416841919659022$, $y = 0.00074999773723013624257778 40$.	
Order (10,4)	
$a_2 = 0.25022550128889438521392 0$, $\beta = -0.00520834604604115659057 84$, $\gamma = 0.00003295467182282037 82$, $x = -0.02088970865555692963681 43$, $y = 0.00005733718613393429177 44$	
$\mathscr{D}_{ha_1}\tilde{\mathscr{B}}_{b_1,\beta_1,\gamma_1}\mathscr{D}_{ha_2}\tilde{\mathscr{B}}_{b_2,\beta_2,\gamma_2}\mathscr{D}_{ha_2}\tilde{\mathscr{B}}_{b_1,\beta_1,\gamma_1}\mathscr{D}_{ha_1}$	2 exp, 2 prod
real (10,4) based on r_2, $\gamma_2 = 0.00085504917142122339094 56432$, $\beta_2 = 0.04943901973117414517945800$, $\gamma_1 = 0.00016048937953278607607 1329$, $\beta_1 = -0.02471950986558707258972900$, $b_2 = -1.70241438391931526809 5376$, $a_2 = -0.17560359597982881702 38439$	
(8,4) minimizing $e_{5,2}$, $\gamma_2 = -0.00051416591205551342029 2834$, $\beta_2 = -0.05712635251331025746146227$, $\gamma_1 = 0.00002356787118511231494 74813$, $\beta_1 = 0.02856317625665512873073114$, $b_2 = -1.70241438391931526809 5376$, $a_2 = -0.17560359597982881702 38439$	

Notice that the local truncation error is then

$$\mathcal{O}(\tau^{2m+1}) = \gamma^{2m+1}\mathcal{O}(h^{2m+1})$$

and methods with small values of the coefficients γ in the composition can approximate the exponentials using a lower order truncation and hence at lower computational cost.

REFERENCES

[1] A. Alvermann and H. Fehske, High-order commutator-free exponential time-propagation of driven quantum systems, *J. Comp. Phys.* **230** (2011), 5930–5956.

[2] V. I. Arnold, *Mathematical Methods of Classical Mechanics*, 2rd, New York: Springer-Verlag, 1989.

[3] P. Bader, "Geometric integrators for Schrödinger equations", PhD thesis, Universidad Politécnica de Valencia, Spain, 2014.

[4] P. Bader and S. Blanes, Fourier methods for the perturbed harmonic oscillator in linear and nonlinear Schrödinger equations, *Phys. Rev. E* **83** (2011), 046711.

[5] P. Bader, S. Blanes, and F. Casas, Solving the Schrödinger eigenvalue problem by the imaginary time propagation technique using splitting methods with complex coefficients, *J. Chem. Phys.* **139** (2013), 124117.

[6] P. Bader, S. Blanes, F. Casas, and E. Ponsoda, Efficient numerical integration of Nth-order non-autonomous linear differential equations, *J. Comp. Appl. Math.* **291** (2016), 380–390.

[7] P. Bader, S. Blanes, E. Ponsoda, and M. Seydaoğlu, Symplectic integrators for the matrix Hill's equation and its applications to engineering model, *J. Comp. Appl. Math.* (under minor revision).

[8] P. Bader, S. Blanes, and M. Seydaoğlu, The Scaling, Splitting, and Squaring Method for the Exponential of Perturbed Matrices, *SIAM J. Matrix Anal. Appl.* **36** (2015), 594–614.

[9] H. F. Baker, Alternants and Continuous groups, *Proc. London Math. Soc.* **3** (1905), 24–47.

[10] A. Bandrauk, E. Dehghanian, and H. Lu, Complex integration steps in decomposition of quantum exponential evolution operators, *Chem. Phys. Lett.* **419** (2006), 346–350.

[11] A. Bandrauk and H. Shen, Improved exponential split operator method for solving the time-dependent Schrödinger equation, *Chem. Phys. Lett.* **176** (1991), 428–432.

[12] S. Blanes and F. Casas, *A Concise Introduction to Geometric Numerical Integration*, vol. 23, CRC Press, 2016.

[13] S. Blanes and F. Casas, On the necessity of negative coefficients for operator splitting schemes of order higher than two, *Appl. Num. Math.* **54** (2005), 23–37.

[14] S. Blanes, F. Casas, P. Chartier, and A. Murua, Optimized high-order splitting methods for some classes of parabolic equations, *Math. Comput.* **82** (2013), 1559–1576.

[15] S. Blanes, F. Casas, A. Farrés, J. Laskar, J. Makazaga, and A. Murua, New families of symplectic splitting methods for numerical integration in dynamical astronomy, *Appl. Num. Math.* **68** (2013), 58–72.

[16] S. Blanes, F. Casas, and A. Murua, Splitting and composition methods in the numerical integration of differential equations, *Bol. Soc. Esp. Mat. Apl.* **45** (2008), 89–145.

[17] S. Blanes, F. Casas, J. A. Oteo, and J. Ros, Magnus and Fer expansions for matrix differential equations: the convergence problem, *J. Phys. A: Math. Gen.* **31** (1998), 259–268.

[18] S. Blanes, F. Casas, J. A. Oteo, and J. Ros, The Magnus expansion and some of its applications, *Phys. Rep.* **470** (2009), 151–238.

[19] S. Blanes, F. Casas, and J. Ros, Improved high order integrators based on the Magnus expansion, *BIT Numer. Math.* **40** (2000), 434–450.

[20] S. Blanes, F. Casas, and J. Ros, Processing symplectic methods for near-integrable Hamiltonian systems, *Cel. Mech. Dyn. Astron.* **77** (2000), 17–36.

[21] S. Blanes, F. Diele, C. Marangi, and S. Ragni, Splitting and composition methods for explicit time dependence in separable dynamical systems, *J. Comput. Appl. Math.* **235** (2010), 646–659.

[22] S. Blanes and P. C. Moan, Fourth- and sixth-order commutator-free Magnus integrators for linear and non-linear dynamical systems, *Appl. Numer. Math.* **56** (2006), 1519–1537.

[23] S. Blanes and P. C. Moan, Practical symplectic partitioned Runge–Kutta and Runge-Kutta-Nyström methods, *J. Comp. Appl. Math.* **142** (2002), 313–330.

[24] S. Blanes and P. C. Moan, Splitting methods for non-autonomous Hamiltonian equations, *J. Comp. Phys.* **170** (2001), 205–230.

[25] S. Blanes and E. Ponsoda, Time-averaging and exponential integrators for non-homogeneous linear IVPs and BVPs, *Appl. Numer. Math.* **62** (2012), 875–894.

[26] S. Blanes and M. Seydaoğlu, Splitting methods with real complex coefficients for seperable non-autonomous semi-linear reaction-diffusion equation of Fisher, *XXIII Congress on Differential Equations and Applications (CEDYA) XIII Congress on Applied Mathematics* (2013), ISBN: 978-84-8021-963-1, 491–498.

[27] J. C. Butcher, *Lie Methods for Nonlinear Dynamics with Applications to Accelerator Physics*, University of Maryland, 2015.

[28] J. C. Butcher, *The Numerical Analysis of Ordinary Differential Equations*, John Wiley, 1987.

[29] W. Cairncross and A. Pelster, Parametric resonance in Bose-Einstein condensates with periodic modulation of attractive interaction, *Eur. Phys. J. D* **68** (2014), 1–6.

[30] J. Candy and W. Rozmus, A symplectic integration algorithm for separable Hamiltonian functions, *J. Comp. Phys.* **92** (1991), 230–256.

[31] F. Casas, Sufficient conditions for the convergence of the Magnus expansion, *J. Phys. A: Math. Theor.* **40** (2007), 15001–15017.

[32] F. Casas and A. Murua, An efficient algorithm for computing the Baker– Campbell–Hausdorff series and some of its applications, *J. Math. Phys.* **50** (2009), 033513-1–033513-23.

[33] F. Casas and A. Murua, An efficient algorithm for computing the Baker-Campbell–Hausdorff series and some of its applications, *J. Math. Phys.* **50**, 033513 (2009).

[34] F. Castella, P. Chartier, S. Descombes, and G. Vilmart, Splitting methods with complex times for parabolic equations, *BIT Numer. Math.* **49** (2009), 487–508.

[35] E. Celledoni and A. Iserles, Approximating the exponential from a Lie algebra to a Lie group, *Math. Comput.* **69** (2000), 1457–1480.

[36] E. Celledoni and A. Iserles, Methods for the Approximation of the Matrix Exponential in a Lie-Algebraic Setting, *IMA J. Numer. Anal.* **21** (2001), 463–488.

[37] R. V. Chacon and A. T. Fomenko, Recursion formulas for the Lie integral, *Adv. Math.* **88** (1991), 200–257.

[38] J. E. Chambers, Symplectic integrators with complex time steps, *Astron. J.* **126** (2003), 1119–1126.

[39] J. E. Chambers and M. A. Murison, Pseudo-high-order symplectic integrators, *Astron. J.* **119** (2000), 425–433.

[40] P. Chartier and A. Murua, An algebraic theory of order, *ESAIM: Mathematical Modelling and Numerical Analysis* **43** (2009), 607–630.

[41] M. Creutz and A. Gocksch, Higher-order hybrid Monte Carlo algorithms, *Phys. Rev. Lett.* **63** (1989), 9–12.

[42] R. Denk, On the floquet exponents of Hill's equation systems, *Math. Nachr.* **172** (1995), 87–94.

[43] M. Drewsen and A. Brøner, Harmonic linear Paul trap: Stability diagram and effective potentials, *Phys. Rev. A* **62** (2000), 045401.

[44] A. Farrés, J. Laskar, S. Blanes, F. Casas, J. Makazaga, and A. Murua, High precision symplectic integrators for the solar system, *Cel. Mech. Dyn. Astron.* **116** (2013), 141–174.

[45] J. J. Garcia-Ripoll, V. M. Perez-Garcia, and P. Torres, Extended parametric resonances in nonlinear Schrödinger systems, *Phys. Rev. Lett.* **83** (1999), 1715–1718.

[46] D. Goldman and T. J. Kaper, *n*th-order operator splitting schemes and nonreversible systems, *SIAM J. Numer. Anal.* **33** (1996), 349–367.

[47] E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*, 2nd, Berlin: Springer Verlag, 2006.

[48] E. Hansen and A. Ostermann, Exponential splitting for unbounded operators, *Math. Comput.* **78** (2009), 1485–1496.

[49] E. Hansen and A. Ostermann, High order splitting methods for analytic semigroups exist, *BIT Numer. Math.* **49** (2009), 527–542.

[50] J. Henrard and A. Lemaitre, A second fundamental model for resonance, *Cel. Mec.* **30** (1983), 197–218.

[51] N. J. Higham, *Functions of Matrices: Theory and Computation*, Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2008, xx+425.

[52] N. J. Higham, The Scaling and Squaring Method for the Matrix Exponential Revisited, *SIAM J. Matrix Anal. Appl.* **26** (2005), 1179–1193.

[53] N. J. Higham, The Scaling and Squaring Method for the Matrix Exponential Revisited, *SIAM Rev.* **51** (2009), 747–764.

[54] N. J. Higham and A. H. Al-Mohy, Computing matrix functions, *Act. Num.* **19** (2010), 159–208.

[55] A. Iserles, H. Z. Munthe-Kaas, S.P. Nørsett, and A. Zanna, Lie group methods, *Act. Num.* **9** (2000), 215–365.

[56] A. Iserles, S. P. Nørsett, and A. F. Rasmussen, Time symmetry and high-order Magnus methods, *Appl. Numer. Math.* **39** (2001), 379–401.

[57] T. Jahnke and C. Lubich, Numerical integrators for quantum dynamics close to the adiabatic limit, *Numer. Math.* **94** (2003), 289–314.

[58] M. V. Karasev and M. V. Mosolova, Infinite products and T products of exponentials, *J. Theor. Math. Phys.* **28** (1977), 721–730.

[59] H. Kinoshita, H. Yoshida, and H. Nakai, Symplectic integrators and their application to dynamical astronomy, *Cel. Mech. Dyn. Astron.* **50** (1990), 59–71.

[60] F. Kittaneh, "Norm inequalities for commutators of normal operators", in: *Inequalities and Applications*, Birkhäuser Basel, 2009, 147–154.

[61] S. Klarsfeld and J. A. Oteo, Magnus approximation in the adiabatic picture, *Phys. Rev. A* **45** (1992), 3329–3332.

[62] J. Laskar and P. Robutel, High order symplectic integrators for perturbed Hamiltonian systems, *Cel. Mech. Dyn. Astron.* **80** (2001), 39–62.

[63] B. Leimkuhler and S. Reich, *Simulating Hamiltonian Dynamics*, New York: Cambridge University Press, 2004.

[64] X. Liao, Symplectic integrator for general near-integrable Hamiltonian system, *Cel. Mech. Dyn. Astron.* **66** (1996), 243–253.

[65] W. Magnus, On the exponential solution of differential equations for a linear operator, *Commun. Pure Appl. Math.* **7** (1954), 649–673.

[66] W. Magnus and S. Winkler, *Hill's equation*, New York: Wiley, 1966.

[67] F. G. Major, V. N. Gheorghe, and G. Werth, *Charged Particle Traps. Physics and Techniques of Charged Particle Field Confinement*, Springer, 2005.

[68] N. W. McLachlan, *Theory and application of Mathieu functions*, New York: Dover, 1964.

[69] R. I. McLachlan, Composition methods in the presence of small parameters, *BIT Numer. Math.* **35** (1995), 258–268.

[70] R. I. McLachlan and G. R. W. Quispel, Splitting methods, *Act. Num.* **11** (2002), 341–434.

[71] R.I. McLachlan and G. R. W. Quispel, *Six lectures on the geometric integration of ODEs*, London: Cambridge University Press, 2001.

[72] S. Mikkola, Practical symplectic methods with time transformation for the few-body problem, *Cel. Mech. Dyn. Astron.* **67** (1997), 145–165.

[73] P. C. Moan, *Efficient Approximation of Sturm-Liouville Problems Using Lie-group Methods*, Numerical analysis reports, University of Cambridge, Department of Applied Mathematics and Theoretical Physics, 1998.

[74] A. H. Al-Mohy and N. J. Higham, A new Scaling and Squaring Algorithm for the Matrix Exponential, *SIAM J. Matrix Anal. Appl.* **31** (2009), 970–989.

[75] C. Moler and C. V. Loan, Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later, *SIAM rev.* **45** (2003), 3–49.

[76] H. Munthe–Kaas and B. Owren, Computations in a free Lie algebra, *Phil. Trans. R. Soc. A* **357** (1999), 957–981.

[77] A. Murua, The Hopf Algebra of Rooted Trees, Free Lie Algebras, and Lie Series, English, *Found. Comput. Math.* **6** (2006), 387–426.

[78] A. Murua and J. M. Sanz-Serna, Order conditions for numerical integrators obtained by composing simpler integrators, *Philos. Trans. Royal Soc. London ser. A* **357** (1999), 1079–1100.

[79] P. J. Olver, *Applications of Lie Groups to Differential Equations*, 2rd, New York: Springer-Verlag, 1993.

[80] M. S. Paterson and L. J. Stockmeyer, On the number of nonscalar multiplications necessary to evaluate polynomials, *SIAM J. Comput.* **2** (1973), 60–66.

[81] W. Paul, Electromagnetic traps for charged and neutral particles, *Rev. Mod. Phys.* **62** (1990), 531–540.

[82] P. Pechukas and J. C. Light, On the exponential form of time-displacement operators in quantum mechanics, *J. Chem. Phys.* **44** (1966), 3897–3912.

[83] T. Prosen and I. Pižorn, High order non-unitary split-step decomposition of unitary operators, *J. Phys. A: math. Gen.* **39** (2006), 5957–5964.

[84] C. Reutenauer, *Free Lie Algebras*, London Maths Soc. Monographs **7**, Oxford: Oxford University Press, 1993.

[85] J. A. Richards, *Analysis of periodically time-varying systems*, New York: Springer-Verlag, 1983.

[86] L. Ruby, Applications of the Mathieu equation, *AJP* **64** (1996), 39–44.

[87] P. Saha and S. Tremaine, Symplectic integrators for solar system dynamics, *Astron. J.* **104** (1992), 1633–1640.

[88] J. M. Sanz-Serna and M. P. Calvo, *Numerical Hamiltonian Problems*, 1rd, London: Chapman and Hall, 1994.

[89] J. Sastre, J. Ibáñez, P. Ruiz, and E. Defez, Accurate and efficient matrix exponential computation, *Int. J. Comput. Math.* **91** (2014), 97–112.

[90] M. Seydaoğlu and S. Blanes, High-order splitting methods for separable non-autonomous parabolic equations, *Appl. Numer. Math.* **84** (2014), 22–32.

[91] M. Seydaoğlu, U. Erdoğan, and T. Öziş, Numerical solution of Burgers' equation with high order splitting methods, *J. Comp. Appl. Math.* **291** (2016), 410–421.

[92] Q. Sheng, Solving linear partial differential equations by exponential splitting, *IMA J. Numer. Anal.* **9** (1989), 199–212.

[93] R. B. Sidje, Expokit: a software package for computing matrix exponentials, *ACM Transactions on Mathematical Software (TOMS)* **24** (1998), 130–156.

[94] G. Strang, On the construction and comparison of difference schemes, *SIAM J. Numer. Anal.* **5** (1968), 506–517.

[95] M. Suzuki, Fractal decomposition of exponential operators with applications to many-body theories and Monte Carlo simulations, *Phys. Lett. A* **146** (1990), 319–323.

[96] M. Suzuki, General theory of fractal path integrals with applications to many-body theories and statistical physics, *J. Math. Phys.* **32** (1991), 400–407.

[97] M. Suzuki, Hybrid exponential product formulas for unbounded operators with possible applications to Monte Carlo simulations, *Phys. Lett. A* **201** (1995), 425–428.

[98] M. Thalhammer, A Fourth-order Commutator-free Exponential Integrator for Nonautonomous Differential Equations, *SIAM J. Numer. Anal.* **44** (2006), 851–864.

[99] K. L. Turner, S. A. Miller, P. G. Hartwell, N. C. MacDonald, S. H. Strogatz, and S. G. Adams, Five parametric resonances in a microelectromechanical system, *Nature* **396** (1998), 149–152.

[100] V. S. Varadarajan, *Lie Groups, Lie Algebras, and Their Representations*, 2nd, New York: Springer Verlag, 1984.

[101] L. Verlet, Computer "experiments" on classical fluids. I. Thermodynamical properties of Lennard–Jones molecules, *Phys. Rev.* **159** (1967), 95–103.

[102] N. V. Vitanov, T. Halfmann, B. W. Shore, and K. Bergmann, Laser-induced population transfer by adiabatic passage techniques, *Annu. Rev. Phys. Chem.* **52** (2001), 763–809.

[103] G. Werth, V. N. Gheorghe, and F. G. Major, *Charged Particle Traps II. Applications*, Springer, 2009.

[104] J. Wisdom and M. Holman, Symplectic maps for the n-body problem, *Astron. J.* **102** (1991), 1528–1538.

[105] J. Wisdom, M. Holman, and J. Touma, Symplectic correctors, *Integration Algorithms and Classical Mechanics* **10** (1996), 217–244.

[106] H. Yoshida, Construction of higher order symplectic integrators, *Phys. Lett. A* **150** (1990), 262–268.