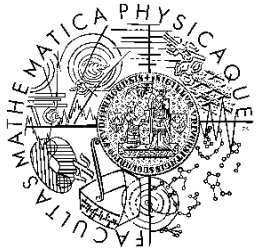


FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

MASTER THESIS

Lorenzo Lazzarino

Numerical Approximation of the Time-Ordered Exponential for Spin Dynamic Simulation

Department of Numerical Mathematics

Supervisor of the master thesis: Stefano Pozza, Dr., Ph.D.
Study programme: Mathematics
Study branch: Computational Mathematics

Prague 2023

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In ..... date ....

Author's signature

Title: Numerical Approximation of the Time-Ordered Exponential for Spin Dynamic Simulation

Author: Lorenzo Lazzarino

Department: Department of Numerical Mathematics

Supervisor: Stefano Pozza, Dr., Ph.D., Department of Numerical Mathematics

Abstract: We describe, discuss, and compare classes of methods for the numerical solution of non-autonomous linear ODEs using problems coming from Nuclear magnetic resonance (NMR) spectroscopy with Magic-angle spinning (MAS) as case study. The newly introduced $\star$-product approach uses a convolution-like product to express the time-ordered exponential to then expand it in a Legendre polynomials basis so to transform the original ODE problem into a linear algebra problem. The aim is to compare the numerical performance of this method with other commonly used methods. Therefore, we take into account geometric numerical integrators. This group of integrators are frequently used in many different areas of research as, for example, quantum mechanics, molecular dynamics and particle accelerators physics. Their approach can either approximate the solution of the non-autonomous ODE by means of a single time-independent exponential (Magnus Integrators) or by means of a product of time-independent exponentials (Splitting Methods, Commutator-Free Exponential Integrators). Finally, numerical experiments on the NMR MAS case study are performed to test the numerical behaviour of the $\star$-process and compare it with the already established alternatives.

Keywords: Time-ordered exponential, $\star$ - product, Magnus expansion, Geometrical numerical integrators, MAS NMR

Contents

1. Introduction

The objective of this thesis is to describe, analyse and compare well established and more recently introduced numerical methods for the solution of non-autonomous ODEs. In particular, we will focus on the performance of a new algorithm based on the so-called $\star$-product, [28]. To do so, we will take as case study a problem in quantum Chemistry that represents an application of particular interest for these methods.

Case study description We take into account the problem of understanding the spin dynamic of systems of particles such us proteins, membrane systems or macromolecular complexes. Following [3], [30], we consider the Nuclear magnetic resonance (NMR) spectroscopy, a powerful and vastly used atomic-level method that, together to being able to capture the motion of particles, allows to understand the chemical and three-dimensional structure of them. In particular, we focus on solid-state NMR spectroscopy with magic-angle spinning (MAS). In this procedure, the sample is rapidly rotated around an axis inclined by the magic-angle $\theta_M = \cos^{-1}(1/\sqrt{3}) \approx 54.7^\circ$ with respect to the static magnetic field, see figure (1.1) for a schematic representation. The spin dynamic in the MAS NMR experiment can be described by a Schrödinger equation

$$\begin{cases} \frac{\mathrm{d}}{\mathrm{d}t}\psi(t) = -i\mathcal{H}(t)\psi(t), \quad t \geqslant 0 \\ \psi(0) = \psi_0 \end{cases} \tag{1.1}$$

where $\mathcal{H}$ represents the Hamiltonian operator and the wave function $\psi(t)$ represents the state of the system at time t. Notice that the Hamiltonian $\mathcal{H}$ is time-dependent. Moreover, there is not spatial dependency. Indeed, the rapid sample spinning given from the MAS averages out the components that depend on the sample orientation with respect to the direction of the magnetic field. In other words, the Schrödinger equation (1.1) is a system of non-autonomous ODEs. Moreover, the Hamiltonian is of size 2^{N_s}, where N_s is the number of spins involved in the system. Hence, to a relatively small number of spins corresponds a big system of ODEs.

General problem description Before focusing on the MAS NMR problem, we are going to describe and analyse methods for the numerical approximation of general non-autonomous linear ODEs of the form

$$\begin{cases} \frac{\mathrm{d}}{\mathrm{d}t}u(t) = A(t)u(t), \quad t \in (t_0, T) \\ u(t_0) = u_0 \end{cases}, \tag{1.2}$$

where A is a $N \times N$ time-dependent matrix. While in the autonomous case (i.e., with time-independent matrix of coefficients) the solution is given by the analytic expression:

$$u(t) = e^{(t-t_0)A}u_0, \tag{1.3}$$

in the case of the equation (1.2) an analytic expression is not always possible. However, if the matrix of coefficients commutes with itself at all times, i.e.,

$$A(t_1)A(t_2) = A(t_2)A(t_1), \quad \forall t_1, t_2 \in (t_0, T), \tag{1.4}$$

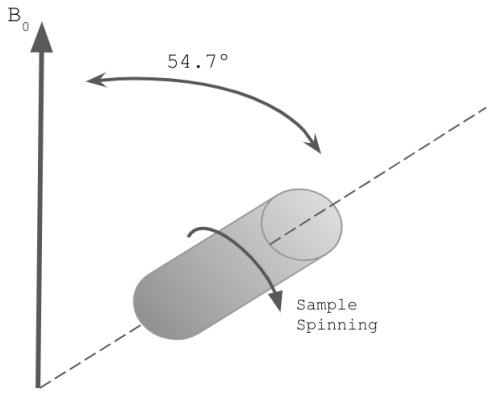


Figure 1.1: Schematic representation of MAS, as in [30]

then the solution of (1.2) is given by [7]

$$u(t) = exp\left(\int_{t_0}^{t} A(\tau)d\tau\right) u_0. \tag{1.5}$$

However, in general an analytic formula in terms of A might not exists. Instead, the solution is given through the notation

$$u(t) = \mathcal{T}exp\left(\int_{t_0}^{t} A(\tau)d\tau\right) u_0, \tag{1.6}$$

where $\mathcal{T}$ is the time-ordering (Dyson) operator, e.g., [22]. It is important to notice that, in general, it is really hard to describe explicitly the action of the time-ordering operator and so to explicitly express the solution to (1.2). For these reasons, the role of fast and reliable numerical approaches to solve such problem is crucial.

Thesis structure In Chapter 2, we describe and discuss a newly introduced procedure: the $\star$-product approach. This is based on an expression of the solution of (1.2) trough a convolution-like product that is expanded into the Legendre polynomial basis so to transform the original ODE into an infinite matrix problem. As we will see in Section 2.2, this problem can be divided into three matrix sub-problems. In Section 2.3, we discuss how to construct and implement the coefficient matrix deriving from a Legendre expansion. Then, an infinite system needs to be solved and so we describe how to find an accurate finite dimensional approximation to its solution (Section 2.4). To obtain the coefficient of the Legendre expansion of the solution of (1.2) a matrix-vector product is computed (Section 2.5). In Chapter 3, the focus will be on already established alternatives. In particular, we analyse geometric numerical integrators that, unlike the standard time integrators such as implicit or explicit Runge-Kutta methods, preserve the main qualitative features of the exact solution. The majority of geometric integrators is based on two alternative strategies. We can split the matrix $A(t)$ into several matrices chosen in such a way that their exponential can be easily evaluated and then express the solution of (1.2) as a product of exponentials. An example is the class of Splitting Methods (Section 3.1). Alternatively, as described in Section 3.2, the use of the so-called Magnus Expansion allows us to represent

the solution of (1.2) as an exponential of an infinite series and so to evaluate it by means of a single exponential. An example is the class of Magnus Integrators (Section 3.3). In Section 3.4, we describe a class of methods that generalizes the Splitting Methods and that is strictly related to the Magnus Expansion. Such combination of the two approaches leads to the so-called Commutator-Free Exponential Integrators. To implement these methods quadrature formulas (Section 3.5) and techniques for the matrix exponential approximation (Section 3.6) are needed.

In conclusion, in Chapter 4, we report tests and comparisons on the numerical behaviour of the $\star$-approach and of the other presented methods on MAS NMR test cases.

2. $\star$-Product approach

In this Chapter we describe the $\star$- product approach for solving ODEs following the description in [28]. We show how with this new process we can express the solution of (1.2) through a convolution-like product and then transform, by an expansion in the Legendre polynomials basis, the original ODE problem into a matrix problem. For the description of this method, we consider the scalar non-autonomous ODE

$$\frac{d}{dt}\tilde{u}(t) = \tilde{f}(t)\tilde{u}(t), \quad \tilde{u}(-1) = 1, \quad t \in [-1, 1]\,, \tag{2.1}$$

where $\tilde{f}$ is an analytic function over $[-1, 1]$. The more general case (1.2) will follow straightforwardly, see [29]. Notice that, since Legendre polynomials are usually defined on the interval $[-1, 1]$, for the sake of simplicity, without loss of generality, we set $t_0 = -1, T = 1$.

2.1 Theoretical background

We recall multiple tools that will be necessary for the description of the $\star$-approach.

Definitions First of all, let us denote with $\Theta(t-s)$ the Heaviside step function:

$$\Theta(t-s) = \begin{cases} 0, & \text{if } t < s \\ 1, & \text{otherwise} \end{cases}. \tag{2.2}$$

Then, we can define a set $\mathcal{A}_\Theta$ of all scalar distribution that can be written as $f(t,s) = \tilde{f}(t,s)\Theta(t-s)$, where $\tilde{f}$ is an analytic function. Moreover, if $f, g \in \mathcal{A}_\Theta$ then we can define the $\star$-product as

$$f(t,s) \star g(t,s) := \int_{-1}^{1} g(t,\tau) f(\tau,s) d\tau, \quad \text{with } f, g \in \mathcal{A}_\Theta, \quad t, s \in [-1, 1]. \tag{2.3}$$

Notice that this definition is easily generalizable to the matrix-matrix, matrix-vector, matrix-scalar, and vector-scalar cases by simply modifying appropriately the product into the integrand. Further, we have that the identity of the $\star$-product is the Dirac delta distribution $\delta(t-s)$.

New expression of the ODE solution Considering $f(t,s) = \tilde{f}(t)\Theta(t-s)$ we can rewrite the ODE (2.1) in the $\star$-framework as follow:

$$\frac{d}{dt}u(t,s) = f(t,s)u(t,s), \quad u(s,s) = 1, \quad t, s \in [-1, 1]\,. \tag{2.4}$$

Then, the solution of (2.1) is $\tilde{u}(t) = u(t,-1)$. By the results presented in [15], we can express the solution of the ODE (2.4) in the form

$$u(t,s) = \Theta(t-s) \star R_\star(f)(t,s), \tag{2.5}$$

where $R_\star(f)(t,s)$ is the $\star$-resolvent

$$R_\star(f)(t,s) = \delta(t-s) + \sum_{k \geq 1} f^{\star k}(t,s). \tag{2.6}$$

2.2 Connection with matrix algebra

In order to devise a numerical method, we aim to transform the computation of (2.5) into a linear algebra problem. To this end, we exploit the relations that the $\star$-algebraic structure has with the matrix algebra, [28].

Expansion in Legendre polynomials A needed tool is the orthonormal sequence of Legendre polynomials $\{p_k\}_{k\geq 0}$. Each p_k is a polynomial of exact degree k that satisfies the orthonormality condition with respect to all other Legendre polynomials, i.e,

$$\int_{-1}^{1} p_k(t)p_\ell(t)dt = \begin{cases} 0, & \text{if } k \neq \ell \\ 1, & \text{if } k = \ell \end{cases}. \tag{2.7}$$

These polynomials are the key for the connection between the $\star$-environment and the matrix one. Indeed, we can expand a function $\tilde{f}(t)$ into the Legendre polynomials basis

$$\tilde{f}(t) = \sum_{d=0}^{\infty} \alpha_d p_d(t), \quad \text{with } \alpha_d = \int_{-1}^{1} \tilde{f}(t)p_d(t)dt. \tag{2.8}$$

It is important to notice that the Fourier coefficients decay with a rate that depends on the smoothness of $\tilde{f}(t)$. In particular, since $\tilde{f}$ is analytic, for some constant $C > 0$ we have:

$$|a_d| \leq C\rho^{-d-1}, \tag{2.9}$$

for some $\rho > 1$ small enough, [33, Theorem 2.1]. Then, if the Fourier coefficients $\alpha_N, \alpha_{N+1}, \ldots$ are smaller than a prescribed tolerance, the truncated expansion $\hat{f}_N(t) := \sum_{d=0}^{N} \alpha_d p_d(t)$ is a good approximation to $\tilde{f}(t)$:

$$\|\tilde{f}(t) - \hat{f}_N(t)\|_\infty = \sum_{d=N+1}^{\infty} \alpha_d p_k(t) \leq \sum_{d=N+1}^{\infty} |\alpha_d| \sqrt{\frac{2d+1}{2}}. \tag{2.10}$$

For a distribution $f \in \mathcal{A}_\Theta$, the expansion in the Legendre polynomial basis is

$$f(t,s) = \sum_{k=0}^{\infty}\sum_{\ell=0}^{\infty} f_{k,\ell} p_k(t)p_\ell(s), \quad \text{for every } t \neq s, \quad t,s \in [-1,1]\,, \tag{2.11}$$

where

$$f_{k,\ell} = \int_{-1}^{1}\int_{-1}^{1} f(\tau,\rho)p_k(\tau)p_\ell(\rho)d\rho d\tau. \tag{2.12}$$

A crucial role in the $\star$-procedure will be played by this coefficients, and in particular by the *coefficient matrix* F defined as the following infinite matrix

$$F := \left[f_{k,\ell}\right]_{k,\ell=0}^{\infty} = \begin{bmatrix} f_{0,0} & f_{0,1} & f_{0,2} & \cdots \\ f_{1,0} & f_{1,1} & f_{1,2} & \cdots \\ f_{2,0} & f_{2,1} & f_{2,2} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}. \tag{2.13}$$

Notice that even if the presence of the Heaviside creates discontinuity, there is no need to make use of strategies to overcome the Gibbs phenomenon ([18]) that arises when the series (2.11) is truncated. Indeed, for the $\star$-procedure there is no need to compute the reconstruction of the function values on the entire domain but only for $t \in [-1,1]$ and $s = -1$ that is where f is analytic in t.

Formulation of the equivalent matrix problem The aim now is to formulate a linear algebra problem that is equivalent to the ODE problem (2.1). In other words, we want an expression that relates the $\star$-resolvent $R_\star$ in (2.6) with the coefficient matrix F in (2.13). First, let us notice that from the definition of $\star$-product (2.3) and the orthonormality property of the Legendre polynomials (2.7), it follows that for $f, g \in \mathcal{A}_\Theta$ and their corresponding coefficient matrices F, G we have

(CM1) $h = f + g \in \mathcal{A}_\Theta$ and its coefficient matrix H satisfy $H = F + G$;

(CM2) $h = f \star g \in \mathcal{A}_\Theta$ and its coefficient matrix H satisfy $H = FG$.

For the proof of these statements see [28, Lemma 2.1]. Moreover, we stress that it is possible to show that the matrix-product in the second result is well defined, i.e., each element of FG exists, see [28, Section 3.3]. Now, let us focus on the $\star$-resolvent. The following holds:

(SR1) If $f(t,s)$ is bounded for $t, s \in [-1, 1]$, then its $\star$-resolvent exists;

(SR2) The $\star$-resolvent is the $\star$-inverse of $(\delta - f)$. In other words,

$$R_\star(f) = (\delta - f)^{-\star}. \tag{2.14}$$

As shown in [28], the first result follows from uniform convergence in $\mathcal{A}_\Theta$ of $\sum_{k\geq 1} f^{\star k}$, while the second one follows from

$$R_\star(f) \star (\delta - f) = \left(\delta + \sum_{k\geq 1} f^{\star k}\right) \star (\delta - f) = \delta. \tag{2.15}$$

Finally, we can combine the above results to obtain an expression of the $\star$-resolvent in terms of F. Let $g = \sum_{k\geq 1} f^{\star k}$, then by (CM1) and (CM2), $g \in \mathcal{A}_\Theta$ and therefore we can consider its coefficient matrix G. Moreover, using the same properties and (SR1) we have

$$R_\star(f) = \delta + \sum_{k\geq 1} f^{\star k} = \phi(t)^T (\mathbb{I} + G)\, \phi(s), \text{ with } \phi(\tau) := \begin{bmatrix} p_0(\tau) \\ p_1(\tau) \\ \vdots \end{bmatrix}. \tag{2.16}$$

If we denote with $\mathbb{I}$ the identity with infinite dimension, the equation (2.16) and (SR2) enable us to show the relation between $(\mathbb{I} + G)$ and $(\mathbb{I} - F)$,

$$\begin{aligned} \delta &= (R_\star(f) \star (\delta - f))(t,s) \\ &= \left(\phi(t)^T (\mathbb{I} + G)\, \phi(s)\right) \star \left(\phi(t)^T (\mathbb{I} - F)\, \phi(s)\right) \\ &= \phi(t)^T (\mathbb{I} + G)(\mathbb{I} - F)\, \phi(s) = \phi(t)^T \mathbb{I}\, \phi(s). \end{aligned} \tag{2.17}$$

Therefore, we obtained the wanted relation, i.e., if the inverse of $(\mathbb{I} - F)$ exists, then

$$R_\star(f) = \begin{bmatrix} p_0(t) & p_1(t) & \dots \end{bmatrix} (\mathbb{I} - F)^{-1} \begin{bmatrix} p_0(s) \\ p_1(s) \\ \vdots \end{bmatrix}. \tag{2.18}$$

The last step to transform the ODE (2.1) into an equivalent matrix problem is to define an expression for the Legendre coefficients $\{c_k\}_{k\geq 0}$ of $\tilde{u}(t)$ expressed by coefficient matrices. This follows immediately from the results described above. Indeed, if we denote with T the coefficient matrix of the Heaviside function and we assume that $(\mathbb{I} - F)$ is invertible, then

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \end{bmatrix} = T(\mathbb{I} - F)^{-1} \begin{bmatrix} p_0(-1) \\ p_1(-1) \\ p_2(-1) \\ \vdots \end{bmatrix}. \tag{2.19}$$

Summing up, we have defined an equivalent linear algebra problem to the original ODE problem by exploiting the expression of the solution $\tilde{u}(t)$ in the $\star$-framework, some results on the $\star$-resolvent, and the properties of the coefficient matrices coming from a Legendre expansion. We can define the found infinite matrix problem as follow:

Infinite matrix problem *Aim*: Given a smooth function $\tilde{f}(t)$ compute the coefficients $\{c_k\}_{k\geq 0}$ of the solution $\tilde{u}(t)$ of the scalar ODE (2.1) expanded in the Legendre polynomials basis.
Procedure: Let $\phi(-1) = \begin{bmatrix} p_k(-1) \end{bmatrix}_{k=0}^{\infty}$ and let T be the coefficient matrix of the Heaviside function.

(IM1) Construct $F = [f_{k,l}]_{k,\ell=0}^{\infty}$, i.e., construct the infinite matrix of Fourier coefficients;

(IM2) Solve $(\mathbb{I} - F)x = \phi(-1)$, i.e., solve an infinite linear system;

(IM3) Compute $Tx = \begin{bmatrix} c_0 & c_1 & c_2 & \cdots \end{bmatrix}^T$, i.e., compute a matrix-vector product.

In the rest of the chapter, we describe how to tackle this sub-problems. In particular, we discuss how to transform each infinite dimensional problem into a finite dimensional one and how to numerically solve it.

2.3 Construct the coefficient matrix

In this section we describe how to deal with the sub-problem of constructing the coefficient matrix (IM1). In other words, we want to find an analytic expression of the entries of this matrix. Moreover, we discuss how to obtain a convenient approximation of F using the banded property of matrices in $\mathcal{A}_\Theta$, we define a finite dimensional approximation and describe a possible implementation.

Analytical expression of Fourier coefficients We recall two properties of the Legendre polynomials that enables us to define a formula for the Fourier coefficients, [28]. Let p_ℓ be the Legendre polynomial of exact degree ℓ.

(LP1) Then, for $\ell = 0$,

$$\int_{-1}^{\tau} p_0(\rho)d\rho = \frac{1}{\sqrt{3}}p_1(\tau) + p_0(\tau), \tag{2.20}$$

and for $\ell > 0$

$$\int_{-1}^{\tau} p_\ell(\rho)d\rho = \frac{1}{\sqrt{2\ell+1}}\left(\frac{1}{\sqrt{2\ell+3}}p_{\ell+1}(\tau) - \frac{1}{\sqrt{2\ell-1}}p_{\ell-1}(\tau)\right). \tag{2.21}$$

(LP2) Let a, b, c be non-negative integers and let $s := \frac{a+b+c}{2}$ and $\alpha := |b-c|$. Then,

$$\begin{aligned}\mathcal{F}_{a,b,c} &:= \int_{-1}^{1} p_a(\rho)p_b(\rho)p_c(\rho)d\rho \\ &= \begin{cases} 0, & \text{if } a+b+c \text{ odd}, \\ 0, & \text{if } s < \max(a,b,c), \\ 0, & \text{if } a < |b-c|, \\ \frac{\sqrt{(2a+1)(2b+1)(2c+1)}}{\sqrt{2}(a+b+c+1)}\binom{2s-2a}{s-a}\binom{2s-2b}{s-b}\binom{2s-2c}{s-c}\binom{2s}{s}^{-1} & \text{else} \end{cases}\end{aligned} \tag{2.22}$$

We want to define a formula for the Fourier coefficients of $f(t,s) = \tilde{f}(t)\Theta(t-s) \in \mathcal{A}_\Theta$. Let us recall that the expansion of $\tilde{f}(t)$ in Legendre polynomials reads $\tilde{f}(t) = \sum_{d=0}^{\infty} \alpha_d p_d(t)$. Therefore, our aim is to find an analytic expression for the Fourier coefficients $b_{k,\ell}^{(d)}$ of the Legendre expansion of $p_d(t)\Theta(t-s) \in \mathcal{A}_\Theta$. Following [28, Theorem 3.3] and considering the orthonormality property of the Legendre polynomials (2.7), we have, for $\ell > 0$:

$$\begin{aligned} b_{k,\ell}^{(d)} &:= \int_{-1}^{1}\int_{-1}^{1} p_d(\tau)\Theta(\tau-\rho)p_k(\tau)p_\ell(\rho)d\rho d\tau \\ &= \int_{-1}^{1} p_d(\tau)p_k(\tau)\left(\int_{-1}^{1}\Theta(\tau-\rho)p_\ell(\rho)d\rho\right)d\tau \\ &= \int_{-1}^{1} p_d(\tau)p_k(\tau)\underbrace{\left(\int_{-1}^{\tau} p_\ell(\rho)d\rho\right)}_{\text{Apply Property (LP1)}}d\tau \\ &= \frac{1}{\sqrt{2\ell+1}}\left[\frac{1}{\sqrt{2\ell+3}}\mathcal{F}_{d,k,\ell+1} - \frac{1}{\sqrt{2\ell-1}}\mathcal{F}_{d,k,\ell-1}\right]. \end{aligned} \tag{2.23}$$

For $\ell = 0$ similar computation can be done obtaining:

$$b_{k,0}^{(d)} = \frac{1}{\sqrt{3}}\mathcal{F}_{d,k,1} + \mathcal{F}_{d,k,0}. \tag{2.24}$$

Moreover, from the property (LP2) and the expression (2.23), it is possible to show that the entries along a diagonal of the matrix $B^{(d)} := \left[b_{k,\ell}^{(d)}\right]_{k,\ell=0}^{\infty}$ decay linearly, see [28, Lemma 3.4].

We have seen that it is possible to formulate an analytic expression for the Fourier coefficients of $p_d(t)\Theta(t-s) \in \mathcal{A}_\Theta$ and we know that we can express $\tilde{f}(t)$ in the basis of Legendre polynomials. Therefore, we can write the coefficient matrix $F := [f_{k,\ell}]_{k,\ell=0}^{\infty}$ of $f(t,s) = \tilde{f}(t)\Theta(t-s)$ as

$$F = \sum_{d=0}^{\infty} \alpha_d B^{(d)}, \tag{2.25}$$

where α_d are the coefficients of the Legendre expansion of $\tilde{f}(t)$. By (2.25), the properties of $B^{(d)}$ can be related to properties of F. In particular, we have that along a diagonal of F there is a linear decay of the entries.

Approximation by a banded matrix We have expressed the coefficient matrix F as a infinite series (2.25). Now we truncate it obtaining the approximation

$$F^{(N)} := \sum_{d=0}^{N} \alpha_d B^{(d)}. \tag{2.26}$$

Let us analyse the maximum norm of the error. We have

$$\|F - F^{(N)}\|_\infty = \left\| \sum_{d=N+1}^{\infty} \alpha_d B^{(d)} \right\|_\infty \leq \sum_{d=N+1}^{\infty} |\alpha_d| \|B^{(d)}\|_\infty. \tag{2.27}$$

By (2.9) and by the bound in [28, Lemma 3.7]:

$$\|B^{(d)}\|_\infty \leq 3d + 2, \tag{2.28}$$

we obtain

$$\|F - F^{(N)}\|_\infty \leq \sum_{d=N+1}^{\infty} C\rho^{-d-1}(3d+2). \tag{2.29}$$

This means that, for a given tolerance δ_{tol}, there exists N such that $\|F - F^{(N)}\|_\infty \leq \delta_{tol}$, i.e., there exists N for which $F^{(N)}$ is a good approximation of F. Moreover, notice that the coefficient matrix $B^{(d)}$ of $p_d(t)\Theta(t-s) \in \mathcal{A}_\Theta$ is a $(d+1)$-banded matrix, i.e.,

$$b_{k,\ell}^{(d)} = 0, \quad \text{for } |k - \ell| > d + 1. \tag{2.30}$$

Therefore, the truncated series $F^{(N)}$ is a $(N+1)$-banded matrix that, for a large N, is sufficiently close to F.

Computation of a finite dimensional approximation Even if $F^{(N)}$ is expressed through a finite sum, this matrix is still an infinite matrix due to the infinite dimension of the matrices $B^{(d)}$. Thus, in order to describe a numerical procedure, an approximation by a finite matrix is needed. We consider the $M \times M$ leading principal submatrix $F_M^{(N)}$ of $F^{(N)}$. This choice will be justified in the next section where we will see how it leads to a good enough approximation of the solution of the infinite linear system in (IM2). Since

$$F_M^{(N)} = \sum_{d=0}^{N} \alpha_d B_M^{(d)}, \tag{2.31}$$

to compute the finite dimensional coefficient matrix we need to efficiently compute the coefficients α_d and the matrix $\{B_M^{(d)}\}_{d\geq 0}$. The first task can be accomplished by making use of the MATLAB package `chebfun`, [12]. Indeed, in [28, Section 4.4] the proposed numerical procedure is the following:

1. Compute using `chebfun` the interpolating Chebyshev coefficient $\{\hat{\beta}_d\}_{d=0}^N$ such that
$$\tilde{f}(t) \approx \sum_{k=0}^{N} \hat{\beta}_d T_d(t). \tag{2.32}$$

2. Transform the Chebyshev coefficient $\{\hat{\beta}_d\}_{d=0}^N$ into Legendre coefficients $\{\hat{\alpha}_d\}_{d=0}^N$ using `cheb2leg` in `chebfun`.

The coefficient matrices $B_M^{(d)}$ of the basis have to be constructed just once. Therefore, the second task is to efficiently compute and store them. A possible strategy for it is based on an expression of the entries $b_{k,\ell}^{(d)}$ in terms of structured matrices. Previously in this chapter, with the equation (2.23), we have expressed $b_{k,\ell}^{(d)}$ in terms of integrals of triple products of Legendre polynomials. Considering a Hankel matrix H_M, a Toeplitz matrix T_M and $C_M = \left[\sqrt{(2b+1)}\sqrt{(2c+1)}\right]_{b,c=0}^{M-1}$, we can write such integrals as

$$[\mathcal{F}_{a,b,c}]_{b,c=0}^{M-1} = \sqrt{2a+1}\,(C_M \circ H_M \circ T_M)\,. \tag{2.33}$$

This allows us to express the finite coefficient matrix of the basis $B_M^{(d)}$ as a composition of structured matrices. Indeed, if we denote with $\underline{H}_M$ and $\underline{T}_M$ the matrices H_{M+1} and T_{M+1} with the last row removed and

$$Z_M := \begin{bmatrix} 1 & -1 & & & & \\ 1 & 0 & -1 & & & \\ & 1 & 0 & -1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & 0 & -1 \\ & & & & 1 & 0 \\ & & & & & 1 \end{bmatrix} \in \mathbb{R}^{(M+1)\times M}, \tag{2.34}$$

then,

$$B_M^{(d)} = \left[b_{k,l}^{(d)}\right]_{k,l=0}^{M-1} = \sqrt{2d+1}\left(\tilde{C}_M \circ ((\underline{H}_M \circ \underline{T}_M)Z_M)\right), \tag{2.35}$$

where $\tilde{C}_M := \left[\frac{\sqrt{2k+1}}{\sqrt{2\ell+1}}\right]_{k,\ell=0}^{M-1} \in \mathbb{R}^{M\times M}$. This representation allows us to use the computational tools of efficient storing and computing for structured matrices. For details on the employed matrices H_M and T_M and a derivation of the formula (2.35) see [28, Section 4.4].

2.4 Approximate solution of the infinite linear system

In this section, we analyse the second sub-problem (IM2):

$$(\mathbb{I} - F)x = \phi(-1). \tag{2.36}$$

In Section 2.3, we have seen that the coefficient matrix F can be approximate by a banded matrix $F^{(N)}$. Therefore, for an appropriate choice of N, the solution of (2.36) will be arbitrarily close to the solution of

$$(\mathbb{I} - F^{(N)})x = \phi(-1). \tag{2.37}$$

Hence, our first goal is to show that the resolvent $(\mathbb{I} - F^{(N)})^{-1}$ exists. Moreover, we have seen that we can approximate $F^{(N)}$ by its $M \times M$ leading principal submatrix $F_M^{(N)}$. This lead to the finite linear system

$$(\mathbb{I}_M - F_M^{(N)})\dot{x} = \phi_M(-1), \tag{2.38}$$

where $\phi_M(-1) = \begin{bmatrix} p_0(-1) & p_1(-1) & \dots & p_{M-1}(-1) \end{bmatrix}^T$. Our second objective will be to show that the solution of (2.38) is an accurate approximation to the solution of (2.37) so to justify the choice of considering $F_M^{(N)}$. Further, we discuss a possible strategy to chose the value of M.

Resolvent existence We are interested in the existence of $(\mathbb{I} - F^{(N)})^{-1}$ and, therefore, in the invertibility of an operator. Indeed, the entries of the (banded) matrix $F^{(N)}$ define the operator $\mathcal{R} : \mathcal{H} \to \mathcal{H}$ where $\mathcal{H}$ is an Hilbert space with orthonormal basis $\{\dot{e}_0, \dot{e}_1, \dot{e}_2, \dots\}$:

$$\mathcal{R}\,\dot{e}_\ell = \sum_{k=\max\{\ell-N,0\}}^{\ell+N} f_{k,\ell}\,\dot{e}_k. \tag{2.39}$$

If we project this operator to the finite dimensional Hilbert space $\mathcal{H}_M = span\{\dot{e}_0, \dots, \dot{e}_{M-1}\}$ by the orthogonal projection $\mathcal{P}_M : \mathcal{H} \to \mathcal{H}_M$ then we obtain a finite dimensional operator $\mathcal{R}_M = \mathcal{P}_M \mathcal{R} \mathcal{P}_M$ that is represented by the matrix $F_M^{(N)}$. By [23, Theorem 3.1], the operator $\mathcal{R}$ is invertible if $F^{(N)}$ is banded and if for all $M = 1, 2, \dots$ and $j = 1, \dots, N+1$, there exist K_j, L_j positive constant such that

$$\left\| \left(\mathbb{I}_M - F_M^{(N)}\right)^{-1} e_{M-j} \right\|_2 \le K_j, \quad \left\| \left(\mathbb{I}_M - \left(F_M^{(N)}\right)^H\right)^{-1} e_{M-j} \right\|_2 \le L_j. \tag{2.40}$$

We have already shown that $F^{(N)}$ is a banded matrix, so it is sufficient to show that (2.40) holds. Lemma 4.1 in [28] shows that for an N banded matrix A if

$$W(A) := \{v^H A\, v, \|v\|_2 = 1\} \subset D(0, r), \tag{2.41}$$

where $D(0, r)$ is disk centered at the origin and with radius $r < 1$, then, there exists $\mu < 1$ and constant C so that

$$\left|(\mathbb{I} - A)^{-1}_{k,\ell}\right| \le C\mu^{d(k,\ell)}, \quad d(k,\ell) := \frac{|k-\ell|}{(N+1)}. \tag{2.42}$$

Therefore, if, for a fixed $r < 1$ and $M \ge M_0$ with M_0 large enough, $W(F_M^{(N)}) \subset D(0, r)$ then

$$\begin{aligned} \left\| \left(\mathbb{I}_M - F_M^{(N)}\right)^{-1} e_\ell \right\|_2^2 &= \sum_{k=0}^{M} \left| \left(\mathbb{I}_M - F_M^{(N)}\right)^{-1}_{k,\ell} \right|^2 \\ &\le C^2 \sum_{k=0}^{M} \mu^{2d(k,\ell)} \le C^2 \sum_{k=0}^{M} \tau^{|k-\ell|}, \quad \tau = \mu^{2/(N+1)} < 1, \\ &\le C^2 \sum_{k=0}^{\infty} \tau^{|k-\ell|} =: K_\ell < \infty, \end{aligned} \tag{2.43}$$

where K_ℓ is independent from M. This means that, under the assumption above, (2.40) is satisfied and so the operator $\mathcal{R}$ in (2.39) is invertible. Further, if for some y there exists x such that $\mathcal{R}x = y$ and $\mathcal{R}_M x_M = \mathcal{P}_M y$ then $x_M \to x$ in the topology norm. Rigorous conditions on $\tilde{f}(t)$ such that the condition $W(F_M^{(N)}) \subset D(0, r)$ holds are subject of an ongoing research. However, numerically based results can be found in [28, Section 4.1].

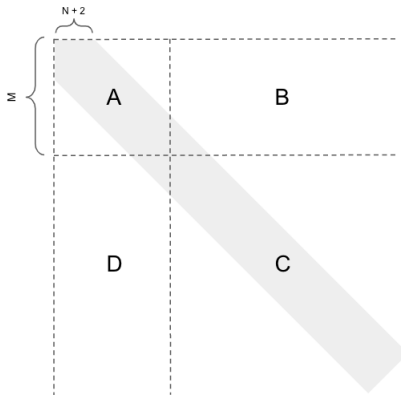


Figure 2.1: Block division of $(\mathbb{I}_M - F^{(N)})$, as in [28].
The coloured region indicate the non-zero entries and the white region the zero ones.

Truncation error Let us denote with x_M the first M entries of the infinite solution of the infinite linear system (2.37). Then, we analyse the error between the solution $\dot{x}$ of the finite dimensional linear system (2.38) and x_M. Let us divide $(\mathbb{I}_M - F^{(N)})$ into four blocks according to figure (2.1). Notice that we have $A := (\mathbb{I}_M - F_M^{(N)}) \in \mathbb{C}^{M\times M}$. As a consequence we have the following:

$$(\mathbb{I} - F^{(N)})x = \phi(-1) \Leftrightarrow \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x_M \\ z \end{bmatrix} = \begin{bmatrix} \phi_M(-1) \\ v \end{bmatrix}, \tag{2.44}$$

where $v = \begin{bmatrix} p_M(-1) & p_{M+1}(-1) & \dots \end{bmatrix}^T$ (recall $\phi_M(-1) = \begin{bmatrix} p_k(-1) \end{bmatrix}_{k=0}^{M-1}$). Then, $\dot{x} = A^{-1}\phi_M(-1)$ and

$$x_M = (\mathbb{I}_M - A^{-1}BD^{-1}C)^{-1}A^{-1}\phi_M(-1) - (\mathbb{I}_M - A^{-1}BD^{-1}C)^{-1}A^{-1}BD^{-1}v. \tag{2.45}$$

Therefore, the error $|x_M - \dot{x}|$ can be written as

$$\left| \left[(\mathbb{I}_M - A^{-1}BD^{-1}C)^{-1} - \mathbb{I}_M \right] \dot{x} - (I_M - A^{-1}BD^{-1}C)^{-1}A^{-1}BD^{-1}v \right|. \tag{2.46}$$

If we truncate all the matrix entries that have magnitude below a fixed threshold, then the matrix A^{-1} can be seen as a K-banded matrix. Studying the structure of the matrices $A^{-1}BD^{-1}$ and $A^{-1}BD^{-1}C$ appearing in (2.46) it is possible to show that by considering the finite dimensional system we are computing accurately (at least) $M - N - K - 2$ entries. However, due to the decay phenomenon, in practice more than $M - N - K - 2$ entries are in general computed with high accuracy. For more details on the truncation error see [28, Section 4.2].

Truncation parameter choice In the previous chapter, we have described some directions on how to efficiently compute the finite dimensional coefficient matrix. This enables us to consider a strategy to select a good value of the truncation parameter that is employed in `chebfun`. Firstly, the Legendre coefficient are computed for a selected truncation parameter M and until a given tolerance is achieved the same process is repeated doubling the truncation parameter at each step. Other possible technique are presented in [28, Section 4.3].

2.5 Controlling the truncation error

The last sub-problem (IM3) in the infinite matrix problem is to control the truncation error selecting an appropriate number of Legendre coefficients of the solution. We recall that to compute these coefficients we perform the matrix-vector product

$$Tx = \begin{bmatrix} c_0 & c_1 & c_2 & \cdots \end{bmatrix}^T. \tag{2.47}$$

In the last section, we saw that instead of the infinite linear system (2.36), we can consider the finite linear system (2.38) with solution $\dot{x}$. Therefore, to compute the Legendre coefficients of the solution $\tilde{u}(t)$ of the ODE (2.1) we could perform the following finite matrix-vector product:

$$T_M \dot{x} = \hat{c}, \tag{2.48}$$

where $\{\hat{c}_k\}_{k=0}^M$ are the approximate Legendre coefficients of $\tilde{u}(t)$. However, numerical experiments have shown that, to better determine how many of these approximate coefficients should be taken into account to reconstruct the solution $\tilde{u}(t)$, a slightly different system and consequent matrix-vector product can be considered. Let $\underline{\hat{F}}_M^{(N)}$ be a matrix equal to $F_M^{(N)}$ except for the $N+1$ rows that are set to zero. Analogously, let $\underline{T}_M$ be equal to T_M with the last row omitted. Then, we consider the system

$$(\mathbb{I}_M - \underline{\hat{F}}_M^{(N)})\dot{\underline{x}} = \phi_M(-1), \tag{2.49}$$

and to determine the approximate coefficients of the solution $\tilde{u}(t)$ we perform the following matrix-vector product

$$\underline{T}_M \dot{\underline{x}} = \underline{\hat{c}}. \tag{2.50}$$

Compared to the coefficients $\hat{c}$, the coefficients $\underline{\hat{c}}$ have a shape that allows us to easily eliminate the coefficients that are more affected from the truncation error, setting them to zero. Hence, they allows us to easily reconstruct the solution $\tilde{u}(t)$ with a prescribed accuracy, i.e., such that

$$\|\tilde{u}(t) - \sum_{d=0}^{\hat{M}} \underline{\hat{c}}_d p_d(t)\|_\infty \lesssim \delta_{\text{sol}}. \tag{2.51}$$

2.6 System of ODEs and their numerical solution

As underlined in the previous sections, the generalization of the $\star$-approach to the matrix case is a direct consequence of the strategies presented for the scalar case. For a fixed initial time $s = -1$, consider the problem

$$\frac{\mathrm{d}}{\mathrm{d}t}u(t) = \tilde{A}(t)\Theta(t-s)u(t), \quad u(-1) = u_0, \quad t \in [-1,1], \tag{2.52}$$

where $\tilde{A}(t)$ is an analytic matrix-valued function over $[-1,1]$. Then, as shown in [29], the approximated solution is given by

$$\begin{aligned} u(t) &\approx (\mathbb{I}_N \otimes \phi_M(t)^T T_M)(\mathbb{I}_{MN} - \mathcal{A}_M)^{-1}(\mathbb{I}_N \otimes \phi_M(-1))u_0 \\ &\approx (\mathbb{I}_N \otimes \phi_M(t)^T T_M)(\mathbb{I}_{MN} - \mathcal{A}_M)^{-1}(u_0 \otimes \phi_M(-1)), \end{aligned} \tag{2.53}$$

where $\mathcal{A}_M$ is the equivalent for the matrix case of the coefficient matrix $F_M^{(N)}$ of the scalar case. Therefore, as in the scalar case, we need to solve the linear system

$$(\mathbb{I}_{MN} - \mathcal{A}_M)x = u_0 \otimes \phi_M(-1). \tag{2.54}$$

Suppose that $\tilde{A}(t)$ in (2.52) can be written in the form

$$\tilde{A}(t) = \sum_{k=1}^{d} A_k \tilde{f}_k(t), \tag{2.55}$$

where $A_1, \ldots, A_d$ are constant matrices and $\tilde{f}_1, \ldots, \tilde{f}_d$ are distinct scalar functions. This is the case for NMR simulations where each matrix-function product represent the evolution in time of different kind of interactions between nuclei in the spin system; see Chapter 4 for more details. In this case, the coefficient matrix $\mathcal{A}_M$ of $A(t,s) = \tilde{A}(t)\Theta(t-s)$ can be expressed as

$$\mathcal{A}_M = \sum_{k=1}^{d} A_k \otimes F_M^{(k)}, \tag{2.56}$$

where $F_M^{(k)}$ is the coefficient matrix of $\tilde{f}_k(t)$,[29]. Thus, we can rewrite the solution of (2.54) in terms of the solution X of the matrix equation

$$X - \sum_{k=1}^{d} F_M^{(k)} X A_k^T = \phi_M(-1)u_0^T, \quad x = vec(X), \tag{2.57}$$

where we denoted the vectorization of X by $vec(X)$. In other words, once the solution of the matrix equation is obtained, we can compute the solution of the linear system (2.52) by stacking into a single vector the columns of X. Exploiting the equation (2.57), we can construct the iterative method

$$X_{n+1} = \sum_{k=1}^{d} F_M^{(k)} X_n A_k^T + \phi_M(-1)u_0^T. \tag{2.58}$$

Then, for a given tolerance δ_{tol}, the iterations can be stopped by the criterion

$$\|X_{n+1} - X_n\| \leq \delta_{tol}. \tag{2.59}$$

Finally, it has been experimentally noted that the matrices X_n are low-rank, see [29]. Hence, to efficiently compute the solution of the matrix equation, and so of the linear system, it is suggested to use low-rank approximations, i.e.,

$$X_n \approx V_n W_n^H, \quad V_n, W_n \in \mathbb{C}^{N \times r}, \tag{2.60}$$

with $r << M$, the numerical rank of X_n.

Table 2.1: A schematic summary of the $\star$-approach
For this procedure, we have considered the scalar ODE problem (2.1) that can be written in the $\star$-framework, see Equation (2.4). Then, the solution is expressed in terms of the $\star$-resolvent which can be written in terms of the coefficient matrix coming from an expansion in Legendre polynomials. In this way, to solve the original ODE problem we can solve a matrix problem. Here, a schematic summary on how to do it.

Infinite matrix problem	**Banded matrices**	**Finite dimensional problem**	**Implementation proposed in [28] (MATLAB)**
Construct the infinite coefficient matrix $F = [f_{k,\ell}]_{k,\ell=0}^{\infty}$	F can be approximated by a banded matrix $F^{(N)} = \sum_{d=0}^{N} \alpha_d B^{(d)}$	Compute, for an appropriate N, $\{\hat{\alpha}_k\}_{k=0}^{N}$, i.e., the interpolating Legendre coefficients of $\tilde{f}(t)$	Use `chebfun` to compute the coefficients $\hat{\beta}_d$ of the interpolating Chebyshev series
			Use `cheb2leg` to transform $\hat{\beta}_d$ into interpolating Legendre coefficients $\hat{\alpha}_k$
		Select the size M of F in order to achieve the given tolerance, i.e., such that enough Legendre coefficients are computed	Compute the Legendre coefficients for M and if the tolerance is not reached than do the same for $2M$. `chebfun` can be used
		Construct the finite dimensional approximation of the coefficient matrix $\hat{F}_M^{(N)} = \sum_{d=0}^{N} \hat{\alpha}_d B_M^{(d)}$	Use the analytic formula (2.35)
Solve an infinite linear system $(\mathbb{I} - F)x = \phi(-1)$	Since, for large N, $F^{(N)}$ is a good approximation of F, the solution of $(\mathbb{I} - F^{(N)})x = \phi(-1)$ is sufficiently close to the solution of the infinite system	We have a finite linear system $(\mathbb{I}_M - \hat{F}_M^{(N)})\dot{x} = \phi_M(-1)$ that can be solved by standard linear algebra techniques	Use the MATLAB function `backslash` on the system $(\mathbb{I}_M - \underline{\hat{F}}_M^{(N)})\underline{\dot{x}} = \phi_M(-1)$, where $\underline{F}_M^{(N)}$ is formed by setting to zero the last $N+1$ rows of $F_M^{(N)}$ (For the matrix case use low-rank approximations)
Compute the infinite matrix-vector product $Tx = c$		Compute the finite matrix-vector product $T_M\dot{x} = \hat{c}$	Compute the product $\underline{T}_M\underline{\dot{x}} = \underline{\hat{c}}$ where $\underline{T}_M$ is formed by setting to zero the last row of T_M

3. Geometric numerical integrators

In this Chapter, we describe and discuss already established alternatives to the $\star$-process to solve non-autonomous ODEs. In particular, we focus on geometric numerical integrators. We consider the problem (1.2) where, without loss of generality, we set $t_0 = 0$, i.e.,

$$\begin{cases} \frac{\mathrm{d}}{\mathrm{d}t}u(t) = A(t)u(t), \quad t \in (0,T) \\ u(0) = u_0 \end{cases}, \tag{3.1}$$

The discussed approach is based on the approximation of a time-dependent matrix exponential. We can identify two main strategies to do so. The Splitting Methods in Section 3.1 and, more generally, the Commutator-free exponential integrators in Section 3.4, approximate the solution of (3.1) through a product of time-independent matrix exponentials that acts on the initial condition vector u_0. Instead, the Magnus Integrators in Section 3.3 expresses the solution of (3.1) by a truncated Magnus expansion, Section 3.2. As we will explain, for these methods the role of quadrature formulas and technique for the approximation of a time-independent matrix exponential will be crucial.

3.1 Splitting methods

The main idea behind the class of Splitting Methods is to evaluate the solution of the problem (3.1) by means of a product of exponentials. If we are able to split the matrix $A(t)$ into multiple suitable matrices whose evaluation result easier, than we can compute the wanted solution by multiplying the solutions founded for the selected matrices.

Autonomous ODE First, let us describe how this approach works for a linear system of ODEs with time-independent coefficients A. Suppose that A can be split into several parts which are easier to deal with. For simplicity, consider two non-commutative square matrices X and Y such that $A = X + Y$ and such that e^{tX} and e^{tY} are easier to compute compare to e^{tA}. Consider the linear ODE

$$\begin{aligned} &\frac{\mathrm{d}}{\mathrm{d}t}u(t) = (X+Y)u(t), \quad t \in (0,T), \\ &u(0) = u_0, \end{aligned} \tag{3.2}$$

whose exact solution is $u(t) = e^{t(X+Y)}u_0$. Then, a standard strategy is to define the approximated solution as a composition of the exact solutions of the equations $\frac{\mathrm{d}}{\mathrm{d}t}u = Xu$ at times $t = a_i\tau$ and $\frac{\mathrm{d}}{\mathrm{d}t}u = Yu$ at times $t = b_i\tau$, for suitable real coefficients a_i, b_i. We have:

$$e^{\tau(X+Y)} \approx e^{\tau b_L Y} e^{\tau a_L X} \cdots e^{\tau b_1 Y} e^{\tau a_1 X} = \prod_{l=1}^{L} \left(e^{\tau b_l Y} e^{\tau a_l X} \right). \tag{3.3}$$

For different values of L and of the coefficients a_i, b_i we obtain different methods. The most common are the Lie-Trotter Splitting with $L = 1, a_1 = 1, b_1 = 1$ or $L = 1, a_1 = 0, a_2 = 1, b_1 = 1, b_2 = 0$ and the Strang splitting method with $L = 2, a_1 = a_2 = 1/2, b_1 = 1, b_2 = 0$ or $L = 2, a_1 = 0, a_2 = 1, b_1 = b_2 = 1/2$. For further details see [27].

Non-autonomous ODE Now, let us consider non-autonomous ODEs. Analogously to the autonomous case, we consider the equation:

$$\frac{\mathrm{d}}{\mathrm{d}t}u(t) = (X(t) + Y(t))u(t), \quad u(0) = u_0. \tag{3.4}$$

Following [9], we present two approaches to find an approximated solution. In both of them, we define a set of ODEs, solve them and define the approximated solution of (3.4) as the product of the founded solutions.

Time-average technique The first approach can be considered as a time-average technique. We define the set of ODEs as follow:

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}u(t) &= X(t)u(t), \quad t \in [c_k\tau, (c_k + a_k)\tau], \\ \frac{\mathrm{d}}{\mathrm{d}t}v(t) &= Y(t)v(t), \quad t \in [d_k\tau, (d_k + b_k)\tau], \end{aligned} \tag{3.5}$$

where $a_0 = 0$, $c_k = \sum_{i=0}^{k-1} a_i$, $b_0 = 0$, $d_k = \sum_{i=0}^{k-1} b_i$ and the solution obtained in the previous step $k - 1$ are taken as the initial condition for the next. The explicit solutions for the equations (3.5) are not always trivial. However, there are some strategies to obtain them, for example the Magnus Expansion (see Section 3.2).

Frozen technique The second approach is the frozen technique and it can be seen as a simpler alternative to the time-average technique. The set of ODEs is defined as follow:

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}u(t) &= X(d_k\tau)u(t), \quad t \in [c_k\tau, (c_k + a_k)\tau], \\ \frac{\mathrm{d}}{\mathrm{d}t}v(t) &= Y(c_k\tau)v(t), \quad t \in [d_k\tau, (d_k + b_k)\tau]. \end{aligned} \tag{3.6}$$

The Splitting Methods are usually simple to implement and their storage requirement are quite modest, [6]. However, as proven in [16], methods with order higher than two always contain negative coefficients a_i, b_i. This can lead to poor accuracy in practice. An alternative is to employ also complex coefficients, [10].

3.2 Magnus expansion

The Magnus Expansion provides an expression of the solution of the problem (3.1) as the exponential of an infinite series. The series is based on commutators of matrices, i.e., given two square matrices A, B, the commutator is defined as

$$[A, B] := AB - BA. \tag{3.7}$$

Commutators can also be nested, e.g.,

$$[A,[B,C]]\,,\ [[A,B],C]\,,\ [[A,B],[C,B]]\,. \tag{3.8}$$

In particular, we can define the following iterative commutators of A and B:

$$[A,B]_m = \begin{cases} B & if \quad m=0, \\ [A,B] & if \quad m=1, \\ [A,[A,B]_{m-1}] & otherwise. \end{cases} \tag{3.9}$$

Derivation Deriving the Magnus Expansion in its general form means aiming at expressing the solution $u(t)$ of (3.1) as

$$u(t) = e^{\Omega(t)}u_0, \qquad \Omega(t) = \sum_{n=1}^{\infty}\Omega_n(t) \tag{3.10}$$

where each $\Omega_n(t)$ is defined as n nested integrals containing $n-1$ nested commutators of $A(t)$. The first three $\Omega_n(t)$ are given by the expressions:

$$\begin{aligned} \Omega_1(t) &= \int_0^t dt_1 A(t_1), \\ \Omega_2(t) &= \frac{1}{2}\int_0^t dt_1 \int_0^{t_1} dt_2 [A(t_1),A(t_2)], \\ \Omega_3(t) &= \frac{1}{6}\int_0^t dt \int_0^{t_1} dt_2 \int_0^{t_2} dt_3 \,[A(t_1),[A(t_2),A(t_3)]] \\ &\qquad + [[A(t_1),A(t_2)],A(t_3)]. \end{aligned} \tag{3.11}$$

In the next part, we will derive a general expression for $\Omega_n(t)$ in terms of $A(t)$ following the description in [4]. By definition, $e^{\Omega(t)}$ solves the differential equation:

$$\frac{\mathrm{d}}{\mathrm{d}t}e^{\Omega(t)} = A(t)e^{\Omega(t)}, \quad \Omega(0)=0. \tag{3.12}$$

Considering the function $f(s,t) = \frac{\mathrm{d}}{\mathrm{d}t}e^{s\Omega(t)}$, we can evaluate the derivative of the matrix exponential on the left hand side. Indeed, $f(s,t)$ satisfies the differential equation

$$\frac{\mathrm{d}}{\mathrm{d}s}f(s,t) = \frac{\mathrm{d}}{\mathrm{d}t}\frac{\mathrm{d}}{\mathrm{d}s}e^{s\Omega(t)}\Omega(t) = f(s,t)\Omega(t) + e^{s\Omega(t)}\dot{\Omega}(t), \quad f(0,t)=0 \tag{3.13}$$

that has solution $f(s,t) = \int_0^s e^{r\Omega(t)}\dot{\Omega}(t)e^{(s-r)\Omega(t)}dr$. Note that here $\dot{\Omega}(t)$ stands for the derivate of Ω.
Therefore, for $s=1$, we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t}e^{\Omega(t)} = \left(\int_0^1 e^{r\Omega(t)}\dot{\Omega}(t)e^{-r\Omega(t)}dr\right)e^{\Omega(t)}. \tag{3.14}$$

We can now use the identity $e^X Y e^{-X} = \sum_{m=0}^{\infty}(1/m!)[X,Y]_m$ which follows from comparison of the derivatives of $s \mapsto e^{sX}Ye^{-sX}$ and $s \mapsto \sum_{m=0}^{\infty}(s^m/m!)[X,Y]_m$. Therefore, we obtain

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}e^{\Omega(t)} &= \sum_{m=0}^{\infty}\left(\int_0^1 \frac{r^m}{m!}[\Omega(t),\dot{\Omega}(t)]_m dr\right)e^{\Omega(t)} \\ &= \sum_{m=0}^{\infty}\frac{1}{(m+1)!}[\Omega(t),\dot{\Omega}(t)]_m e^{\Omega(t)}. \end{aligned} \tag{3.15}$$

Substituting the expression (3.15) in the equation (3.12) we obtain a relation between $A(t)$ and $\Omega(t)$, i.e.,

$$A(t) = (\frac{\mathrm{d}}{\mathrm{d}t}e^{\Omega(t)})e^{-\Omega(t)} = \sum_{m=0}^{\infty} \frac{1}{(m+1)!}[\Omega(t), \dot{\Omega}(t)]_m. \tag{3.16}$$

Finally, to obtain a general expression describing each Ω_n in terms of $A(t)$, we insert the expression for Ω (3.10) in (3.16):

$$\begin{aligned} A(t) &= \sum_{m=0}^{\infty} \frac{1}{(m+1)!} \left[\sum_{n=1}^{\infty} \Omega_n, \sum_{k=0}^{\infty} \dot{\Omega}_k\right]_m \\ &= \sum_{m=0}^{\infty} \frac{1}{(m+1)!} \sum_{n_1,\ldots,n_m=1}^{\infty} \sum_{k=0}^{\infty} [\Omega_{n_1}, [\Omega_{n_2}, \ldots, [\Omega_{n_m}, \dot{\Omega}_k] \ldots]. \end{aligned} \tag{3.17}$$

Solving for $\dot{\Omega}_n$ and integrating, we have the desired explicit expressions

$$\begin{aligned} \Omega_1(t) &= \int_0^t dt' A(t'), \\ \Omega_n(t) &= -\sum_{m=1}^{n-1} \frac{1}{(m+1)!} \sum_{\substack{\{n_i\}_{i=1}^m, k \geqslant 1 \\ \sum_{i=1}^m n_i + k = n}} \int_0^t dt' [\Omega_{n_1}(t'), [\Omega_{n_2}(t'), \ldots, [\Omega_{n_m}(t'), \dot{\Omega}_k(t')] \ldots]. \end{aligned} \tag{3.18}$$

For further details on the derivation of the Magnus expansion see [4].

3.3 Magnus integrators

We can approximate the solution of (3.1) by means of the Magnus Expansion. In other words, we can construct a class of methods, the Magnus Integrators, truncating the infinite series (3.10) and using quadrature formulas to estimate the integrals in (3.18).

For a better understanding, we report an example presented in [8]. Let us consider the two principal terms in (3.10), $\Omega_1(t) + \Omega_2(t)$. In addition, let us approximate the related integrals by a quadrature formula with K nodes. Lastly, let us fix a time-step τ. In this way, we obtain the forth-order Magnus Integrator:

$$\begin{aligned} &u_1 = e^{\Omega^{[4]}(\tau)} u_0 = u(t_1) + \mathcal{O}(\tau^5), \\ &\Omega^{[4]}(\tau) = \tilde{B}_1(\tau) + [\tilde{B}_2(\tau), \tilde{B}_3(\tau)], \\ &\tilde{B}_j(\tau) = \tau \sum_{k=1}^{K} \tilde{a}_{jk} A(c_k \tau), \qquad j \in \{1,2,3\}, \quad c_k \in [0,1], \quad k \in \{1,\ldots,K\} \end{aligned} \tag{3.19}$$

3.3.1 Magnus integrators and Legendre polynomials

For large n, the computation of Ω_n can become cumbersome making hard to explicitly define high order Magnus Integrators. The main issues are related to the presence of nested integrals and nested commutators in the definition of Ω_n. However, there are strategies to simplify the computations. One of these is presented by Alvermann and Fehske in [4]. The idea is to replace the nested

integrals by single integrals and to calculate only the necessary commutators.

We consider again the already presented Legendre polynomials (see Section 2.2). However, this time we take into account the shifted version of them that is defined on the interval $[0,1]$ and, to avoid confusions, we identify them with P_k:

$$\begin{aligned} &P_0(x) = 1, \quad P_1(x) = 2x - 1, \\ &P_{n+1}(x) = \frac{2n+1}{n+1}(2x-1)P_n(x) - \frac{n}{n+1}P_{n-1}(x). \end{aligned} \tag{3.20}$$

Note that these polynomials are symmetric with respect to $x = 1/2$ and they form a complete set of orthogonal functions on the interval $[0,1]$. Fixing a time step τ, we can expand $A(t)$ on the interval $[0,\tau]$ into a series of Legendre polynomials in the following way:

$$A(t) = \frac{1}{\tau}\sum_{n=1}^{N} A_n P_{n-1}\left(\frac{t}{\tau}\right) + \mathcal{O}(\tau^{N+1}), \quad t \in [0,\tau], \tag{3.21}$$

where

$$A_n = (2n-1)\int_0^\tau A(t)P_{n-1}\left(\frac{t}{\tau}\right)dt. \tag{3.22}$$

Then, $\Omega(\tau)$ can be express as a sum of nested commutators of the expansion coefficients A_n by inserting the expansion (3.21) into the recursion (3.18). Hence, because of the mentioned properties of Legendre polynomials, we can reduce the nested integrals to single integrals as desired. To see how this actually works, let us consider the case for $n = 2$. We have:

$$\begin{aligned} \Omega_2(\tau) &= \frac{1}{2\tau^2}\int_0^\tau dt_1 \int_0^{t_1} dt_2 \left[\sum_{n_1\geqslant 1} A_{n_1}P_{n_1-1}\left(\frac{t_1}{\tau}\right), \sum_{n_2\geqslant 1} A_{n_2}P_{n_2-1}\left(\frac{t_2}{\tau}\right)\right] \\ &= \frac{1}{2}\sum_{n_1,n_2\geqslant 1}\left(\int_0^1 dx_1 \int_0^{x_1} dx_2 P_{n_1-1}(x_1)P_{n_2-1}(x_2)\right)[A_{n_1}, A_{n_2}]\,. \end{aligned} \tag{3.23}$$

By orthogonality, the integral of $P_{n_1-1}(x_1)P_{n_2-1}(x_2)$ vanishes for every choice of n_1 and n_2 except for $n_1 = 2, n_2 = 1$ and $n_1 = 1, n_2 = 2$. We obtain

$$\begin{aligned} \Omega_2(\tau) &= \frac{1}{2}\left(\int_0^1 dx_1 \int_0^{x_1} dx_2 P_0(x_1)P_1(x_2) - P_1(x_1)P_0(x_2)\right)[A_1, A_2] \\ &= -\frac{1}{6}[A_1, A_2]. \end{aligned} \tag{3.24}$$

We have seen that, for $n = 2$, not all commutators contribute to $\Omega(\tau)$. This can be generalized for all n as a consequence of the two following properties of the expansion. Let $[A_{i_1}, \ldots, A_{i_n}]$ represent an arbitrary nested commutator composed by the matrices $A_{i_1}, \ldots, A_{i_n}$. We say that a nested commutator $[A_{i_1}, \ldots, A_{i_n}]$ has even (odd) order if $i_1 + \cdots + i_n$ is even (odd). Then:

(ML1) Since the Legendre polynomials are symmetric with respect to $x = 1/2$, i.e., $P_n(1-x) = (-1)^n P_n(x)$, it follows that even order terms in the expansion vanish.

(ML2) As a consequence of the Legendre polynomials orthogonality, a commutator $[A_{n_1}, \ldots, A_{n_m}]$ vanishes if $n_k > 1 + \sum_{i\neq k} n_i$, for some $k = 1, \ldots, m$.

The property (ML2) underlines the importance of choosing the Legendre expansion instead of, for example, the apparently simpler Taylor expansion. Indeed, even if (ML1) would still holds for an expansion in centered powers $(t-\tau/2)^n$, (ML2) is a direct consequence of the orthogonality of the Legendre polynomials.

3.3.2 Drawbacks

The presence of commutators can imply serious consequences. Indeed, the properties of $A(t)$ are usually not passed on to commutators and therefore is difficult to determine how to characterize the well-definiteness and the stability of the method. Moreover, high-order Magnus Integrators may require a large computational effort. Following [8], we denote by

$$\mathscr{C}(e^{a_0 A(t)}\xi) \tag{3.25}$$

the computational cost needed to evaluate by Krylov-type methods the action of a matrix-exponential $e^{a_0 A(t)}$, with a_0 a real or complex coefficient, on a vector ξ. Notice that, typically, a Krylov subspace method requires one or two matrix-vector products per iteration. This will serve as a way to measure and compare different Magnus Integrators. Let us consider, for example, the forth-order Magnus Integrator (3.19) implemented by Krylov techniques (see Section 3.6 for more details). We have

$$\Omega^{[4]}(\tau)u_0 = \tilde{B}_1(\tau)u_0 + \left(\tilde{B}_2(\tau)\tilde{B}_3(\tau)u_0 - \tilde{B}_3(\tau)\tilde{B}_2(\tau)u_0\right). \tag{3.26}$$

Therefore, for each time step τ, the computation of $e^{\Omega^{[4]}(\tau)}u_0$ requires a cost of five matrix-vector multiplication. In other words, we have

$$\mathscr{C}(e^{\Omega^{[4]}(\tau)}u_0) \simeq 5\mathscr{C}(e^{A(t)}u_0). \tag{3.27}$$

As a further example, a sixth-order Magnus Integrator requires, for each time-step τ, at least 13 matrix-vector multiplications, consequently

$$\mathscr{C}(e^{\Omega^{[6]}(\tau)}u_0) \simeq 13\mathscr{C}(e^{A(t)}u_0). \tag{3.28}$$

3.4 Commutator-free exponential integrators

The Commutator-Free Exponential (CFE) Integrators can be seen as a generalization of the Splitting Methods (Section [3.1]), but with significantly reduced number of exponentials involved. The aim is to circumvent the difficulties of Magnus Integrators, in particular the presence of commutators.
We consider the expression

$$\tilde{U}^{(N)}_{CF}(\tau) = e^{\Omega_1}e^{\Omega_2}\dots e^{\Omega_s} \tag{3.29}$$

where each of the exponent is a linear combination

$$\Omega_i = \sum_{n=1}^{N} f_{i,n}A_n \quad \forall\, i = 1,\dots,s \tag{3.30}$$

of the coefficient (3.22) $A_1, \ldots, A_N$ of the Legendre expansion of $A(t)$. Note that the coefficients $f_{i,n}$ determine the CFE Integrator. Their choice is made once and independently of the specific $A(t)$.
The formula (3.29) can be seen as the exact integrator for an auxiliary problem with fictitious, stepwise constant $\tilde{A}(t)$. Therefore, we aim to determine coefficients so that the error introduced by the replacement of the original problem with this auxiliary problem is proportional to τ^{N+1}, independently of $A(t)$.

The simplest CFE Integrator is the so-called second-order midpoint rule:

$$\tilde{U}^{(2)}_{CF2:1}(\tau) = exp\left[\int_0^\tau dt\, A(t)\right] \approx exp\left[\tau A(\tau/2)\right] \tag{3.31}$$

corresponding to $s = 1, f_{1,1} = 1$, [4].

3.4.1 Order conditions

For higher-order CFE Integrators the task of finding the suitable coefficients can be challenging. Moreover, there is not a systematic procedure to accomplish such task. However, there exist strategies that lead to conditions on the coefficients that guarantee a certain order of accuracy of the integrator. As an example, we present the approach by Alvermann and Fehske [4]. This is based on the use of the so-called Baker–Campbell– Hausdorff (BCH) formula.

The BCH formula is a special case of Magnus Expansion

$$e^{X_1} \ldots e^{X_s} = exp\left[\sum_{i=1}^{s} X_i + \frac{1}{2}\sum_{1\le i<j\le s}[X_i, X_j] + \ldots\right] \tag{3.32}$$

that allows us to combine the s exponentials so to obtain $\tilde{U}^{(N)}_{CF}(\tau) = e^{\tilde{\Omega}}$, where

$$\tilde{\Omega} = \sum_{i=1}^{s} f_{1,i}A_1 + \sum_{1\le i<j\le s}\frac{f_{i,1}f_{j,2} - f_{j,1}f_{i,2}}{2}[A_1, A_2] + \ldots \tag{3.33}$$

We obtain the order conditions by demanding equality between $\tilde{\Omega}$ and Ω from the Magnus expansion (3.10), for all the terms of order at most τ^N. The following properties hold, [4, Section 4.1]:

(OC1) A CFE Integrator is time-reversal symmetric if it holds that

$$f_{s-i+1,n} = (-1)^{n+1} f_{i,n}.$$

Thus, for a time-symmetric CFE Integrator it is sufficient to specify the $f_{i,n}$ for $i \le s/2$.

(OC2) By property (ML2), the order condition involving higher-order A_n can be satisfied by setting $f_{i,n} = 0$ for $n > N/2$.

Hence, by (OC1) the number of relevant coefficients and order conditions has been approximately halved. While, (OC2) allows us to take the sum of the first $N/2$ terms in (3.30).

As already mentioned, there exist other ways to construct order conditions. An example is the derivation by means of a particular subset of multi-indices, the so-called Lyndon word. The idea behind this strategy is similar to the idea we just described. First, we compute an expansion of the product exponentials $e^{X_1} \dots e^{X_s}$, obtaining an exponent $\bar{\Omega}$. Then, we take into account only the terms corresponding to a Lyndon word of both $\bar{\Omega}$ and Ω. Finally, comparing the selected terms, we have the desired order conditions. For more details on this approach see [8].

3.5 Quadrature formulas

For the implementation of the methods we have seen until now, we need to approximate integrals. Indeed, for both Magnus Integrators and CFE Integrators, we need to estimate the integrals in the coefficients (3.22) from the Legendre expansion (3.21). We make use of a quadrature rule of the kind:

$$I(f) = \int_0^1 f(x)dx = \sum_{j=1}^{m} w_j f(x_j) + R[f] = I_m(f) + R[f], \tag{3.34}$$

where $x_1, \dots, x_m$ are the quadrature nodes, $w_0, \dots, w_m$ are the corresponding weights and $R[f]$ is the reminder. The remainder, which will be the error of the approximation, is usually unknown. A quadrature rule is said to be exact for the degree d when $R[p] = 0$ for all polynomials p of degree d and there is at least one polynomial q of degree $d+1$ for which $R[q] \neq 0$. Note that for a arbitrary set of nodes $\{x_j\}_{j=1}^m$ there exists a unique set of weights $\{w_j\}_{j=1}^m$ for which the quadrature formula in (3.34) is exact for polynomials of degree $\leqslant m-1$, [32, Theorem 37.2].

We consider the so-called *Gauss-Legendre quadrature formula*. The idea is to optimally pick both the m nodes and the weights in order to increase as much as possible the degree d for which the formula is exact. It is known that there exists a unique set of nodes and weights that satisfy this request and it consists of the zeros of the (shifted) Legendre polynomial $P_m(x)$. This formula is exact for polynomials of degree at most $2m-1$. This quadrature is commonly used because no other quadrature formula (3.34) has order of accuracy higher than $2m-1$.

To apply the Gauss-Legendre quadrature to the methods we previously described, consider $m = N/2+1$. Then we can approximate the Legendre coefficients $A_n(t)$ as follow:

$$A_n \approx (2n-1)\tau \sum_{m=1}^{N/2+1} w_m P_{n-1}(x_m) A(x_m \tau). \tag{3.35}$$

Thus, as stated in [4], for the particular case of CFE Integrators, we can write Ω_i as a linear combination of $A(t)$ at different time $x_m\tau$

$$\Omega_i = \tau \sum_{m=1}^{N/2+1} g_{i,m} A(x_m \tau), \tag{3.36}$$

with new coefficients

$$g_{i,m} = w_m \sum_{n=1}^{N/2+1} (2n-1) P_{n-1}(x_m) f_{i,n}. \tag{3.37}$$

3.6 Matrix exponential approximations

In several of the methods discussed in this thesis, the solution of the ODE (3.1) is approximated by computing one or more matrix exponentials. For example, for the CFE Integrators, the most expensive numerical task of formula (3.29) is the evaluation of e^{Ω_i}. In this section, we will present several of the most commonly used strategies to compute a matrix exponential. The methods can be divided in two main categories: the ones for computing the entire matrix exponential, and the ones for computing the action of a matrix exponential on a vector.

3.6.1 Approximation of the entire matrix exponential

In the first case, given a matrix M, the goal is to find an approximation of the exponential e^M. Following [26], we present the Padé Approximation and the Scaling and Squaring strategy.

Padé approximation The (p, q) Padé approximation to e^M is defined by

$$R_{pq}(M) = [D_{pq}(M)]^{-1} N_{pq}(M), \tag{3.38}$$

where

$$\begin{aligned} N_{pq}(M) &= \sum_{j=0}^{p} \frac{(p+q-j)!p!}{(p+q)!j!(p-j)!} M^j, \\ D_{pq}(M) &= \sum_{j=0}^{q} \frac{(p+q-j)!q!}{(p+q)!j!(q-j)!} (-M)^j. \end{aligned} \tag{3.39}$$

The nonsingularity of $D_{pq}(M)$ is guaranteed if p and q are large enough or if the eigenvalues of M are negative.
We use the Padé approximation when $\|M\|$ is not too large. In this case, a key role is played by the choice of p and q. The diagonal approximations ($p = q$) are, in general, preferred over the off-diagonal approximations for two main reasons. First, even if the amount of work needed to compute $R_{pq}(M)$ and $R_{qq}(M)$ is almost the same, in the first case the approximation has order $p + q$ while in the second $2q > p + q$. Second, if all eigenvalues of M are in the left half plane, then for $p > q$ cancellation occurs while for $p < q$ the denominator matrix $D_{pq}(M)$ is badly conditioned. In both cases, we have larger rounding errors and so the most reasonable choice is a diagonal approximation. In general, the main disadvantage of this method is due roundoff error that leads to unreliability of the approximation. Let us consider a diagonal[1] approximation. Then, for large q, $D_{qq}(M)$ tends to $e^{-M/2}$, while $N_{qq}(M)$ tends to $e^{M/2}$. Therefore, cancellation error occurs. Moreover, the matrix $D_{pq}(M)$ could be poorly conditioned with respect to the inversion, especially when M has widely spread eigenvalues. However, these obstacles do not prevent this technique from being a powerful tool.

Scaling and Squaring We can overcome the issues of the Padé approximation by using the Scaling and Squaring strategy. Consider the following property of the exponential function:

$$e^M = \left(e^{\frac{M}{n}}\right)^n. \tag{3.40}$$

[1]For the case with $p \neq q$ similar comments apply.

The idea behind the Scaling and Squaring strategy is to set $n = 2^j$ in such a way $e^{M/n}$ can be computed efficiently and in a reliable way. Then, to compute the approximation, we perform repeating squaring. An important role is played by the choice of n. Usually, it is taken to be the smallest power of two for which $\|M\|/n \leqslant 1$. In this way, the norm $\|M/n\|$ is small enough to perform the Padé Approximation for $e^{\frac{M}{n}}$, i.e., $R_{qq}(M/2^j)$ for some q and j that have to be chosen. As shown in [26], if $\|M\| \leqslant 2^{j-1}$, then

$$\left[R_{qq}(M/2^j)\right]^{2^j} = e^{M+E}, \tag{3.41}$$

where

$$\frac{\|E\|}{\|M\|} \leqslant 8 \left[\frac{\|M\|}{2^j}\right]^{2q} \left(\frac{(q!)^2}{(2q)!(2q+1)!}\right). \tag{3.42}$$

This results enables us to determine q and j in various ways. For example, considering an error tolerance ϵ, it is possible to select any pair (q, j) such that (3.42) implies

$$\frac{\|E\|}{\|M\|} \leqslant \varepsilon. \tag{3.43}$$

When properly implemented, this strategy is one of the most effective we know. Moreover, in [1],[19], Al-Mohy and Higham have presented an algorithms starting from this strategy on which the build-in MATLAB function `expm` is based.

3.6.2 Approximation of the action of the matrix exponential on a vector

In the case we are studying, we do not need to compute the entire exponential, but the matrix exponential times a vector. Below, we present several methods able to approximate this quantity.

Scaling and Squaring-like strategy Because of the efficiency of Scaling and Squaring for the computation of a matrix exponential, we first describe a way to adapt this strategy to the case of computing the action of a matrix exponential on a vector. Following [2], we consider a rational approximation r_m of the exponential function and we assume r_m to be good near the origin. Then, we select an integer $s \geqslant 1$ in such a way that $r_m(s^{-1}M)$ is a good approximation of $e^{s^{-1}M}$. Finally, we get approximation $Z_s \approx e^M\xi$ by exploiting the relation

$$e^M\xi = (e^{s^{-1}M})^s\xi = \underbrace{e^{s^{-1}M}e^{s^{-1}M}\cdots e^{s^{-1}M}}_{s\,times}\xi, \tag{3.44}$$

and defining the recurrence

$$Z_{i+1} = r_m(s^{-1}M)Z_i, \quad i = 0, \ldots, s-1, \quad Z_0 = \xi. \tag{3.45}$$

A good choice of r_m can be a truncated Taylor series

$$r_m = T_m(s^{-1}M) = \sum_{j=0}^{m} \frac{(s^{-1}M)^j}{j!}. \tag{3.46}$$

This choice enables us to determinate s and m in an optimal way. This approach shares the same form of approximation and backward error analysis of the Scaling and Squaring method for the computation of e^M. However, it does not make use of the repeating squaring technique in the final steps.

Krylov subspace methods Another strategy is based on the so-called Krylov subspace methods. The main idea is to approximate the action of a matrix exponential on a vector in the following way

$$e^M \xi \approx p_{m-1}(M)\xi, \tag{3.47}$$

where p_{m-1} is a polynomial of degree $m-1$. Let us consider the Krylov subspace

$$K_m(M, \xi) = span\{\xi, M\xi, \ldots, M^{m-1}\xi\}. \tag{3.48}$$

Then, the problem can be seen as finding an element of $K_m(M, \xi)$ that approximate $e^M \xi$. Following [31], we will present three different approaches to accomplish such task.

The first strategy we describe makes use of the *Arnoldi algorithm*. This process aim to construct and compute an orthonormal basis of $K_m(M, \xi)$. The idea is to apply the Gram-Schmidt orthogonalization process starting with $v_1 = \xi/\|\xi\|$. Therefore, according to [17], we first compute the recurrence coefficients, then we subtract from Mv_j the components of Mv_j on the previous basis vectors and finally, if possible, we normalize the resulting new basis vector. We obtain the following algorithm:

Algorithm 1 Arnoldi Algorithm

1: Compute $v_1 = \xi/\|\xi\|$
2: **For** $j = 1, 2, \ldots, m$ **do**
3: Compute $h_{i,j} = (Mv_j, v_i)$ for $i = 1, \ldots, j$
4: Compute $\omega_j = Mv_j - \sum_{i=1}^{j} h_{i,j} v_i$
5: $h_{j+1,j} = \|\omega_j\|$. If $h_{j+1,j} = 0$ then Stop
6: $v_{j+1} = \frac{\omega_j}{h_{j+1,j}}$
7: **End**

It is not hard to verify that the vectors v_j form an orthonormal basis for the Krylov subspace $K_m(M, \xi)$. It is really useful to look at this process through its matrix formulation. To this end, let's define the matrix V_m whose columns are the vectors v_j, with $j = 1, \ldots, m$, computed with the algorithm just described. Then, we have that the recurrence relations defining the vector v_{m+1} of the Arnoldi algorithm can be written as

$$MV_m = V_m H_m + h_{m+1,m} v_{m+1} e_m^T, \tag{3.49}$$

where H_m is an upper Hessenberg matrix with entries $h_{i,j}$ and e_m stands for the m-th column of the identity matrix I_m. From (3.49) we get $H_m = V_m^T M V_m$. Note that H_m represent the projection of the linear transformation M onto the subspace $K_m(M, \xi)$ with respect to V_m. We look for an approximation $x = V_m y \in$

$K_m(M,\xi)$ that minimizes the norm $\|e^M\xi - V_m y\|_2$. The optimal solution to this problem is known to be

$$y_{opt} = V_m^T e^M \xi. \tag{3.50}$$

However, y_{opt} contain the unknown vector $e^M\xi$ and so it is not computable. Nevertheless, we have

$$y_{opt} = \|\xi\|_2 V_m^T e^M V_m e_1. \tag{3.51}$$

Considering the series expansion of the exponential we get

$$V_m^T e^M V_m = V_m^T \left(\sum_{k=0}^{\infty} \frac{M^k}{k!}\right) V_m = \sum_{k=0}^{\infty} \frac{V_m^T M^k V_m}{k!} \approx \sum_{k=0}^{\infty} \frac{H_m^k}{k!} = e^{H_m}. \tag{3.52}$$

As suggested in [13], we can use this approximation to obtain

$$e^M\xi \approx \|\xi\|_2 V_m e^{H_m} e_1. \tag{3.53}$$

Note that $V_m^T M^k V_m \xi = H_m^k e_1$ for $k = 0, \ldots, m-1$. Moreover, the approximation will be exact for all m greater or equal to the degree of the minimal polynomial of $v_1 = \xi/\|\xi\|_2$ with respect to M, [13, Section 3]. However, in practice this is highly improbable to happen. Nevertheless, we observe that even with relatively small m we can obtain extremely accurate approximations. In addition, since for an arbitrary scalar τ the Krylov subspace associated with M and τM are identical and $V_m^T(\tau M)V_m = \tau H_m$, we have

$$e^{\tau M}\xi \approx \|\xi\|_2 V_m e^{\tau H_m} e_1. \tag{3.54}$$

The described strategy can be seen as a model reduction. Indeed, it considerably reduces the dimension of the matrix to approximate. Therefore, for m small enough, we can perform other methods to approximate $e^{H_m}e_1$. Due to the simple structure and the small dimension of the matrix H_m this last task will be significantly cheaper then the original one of approximating $e^M\xi$. At the end of this section we will discuss it more in detail.

The second strategy we describe makes use of the *Lanczos alghorithm* and it is strictly linked to the strategy just presented. In this case, we consider a symmetric matrix M. Then, from the Arnoldi iteration we get the matrices V_m, H_m such that

$$H_m = V_m^T M V_m. \tag{3.55}$$

This implies that also H_m is symmetric and, since it is an upper Hessenberg matrix, we conclude that H_m is a symmetric tridiagonal matrix. This matrix is called the Jacobi matrix and it will be denoted by

$$T_m = \begin{bmatrix} \alpha_1 & \beta_2 & & \\ \beta_2 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_m \\ & & \beta_m & \alpha_m \end{bmatrix}. \tag{3.56}$$

The Arnoldi recurrence in Algorithm 1 becomes

$$\omega_j = Mv_j - \sum_{i=1}^{j} h_{i,j} v_i = Mv_j - \alpha_j v_j - \beta_j v_{j-1}. \tag{3.57}$$

Algorithm 2 Lanczos Algorithm

1: Compute $v_1 = \xi/\|\xi\|$
2: **For** $j = 1, 2, \ldots, m$ **do**
3: $\quad \omega_j = Mv_j - \beta_j v_{j-1}$
4: $\quad \alpha_j = (\omega_j, v_j)$
5: $\quad \omega_j = \omega_j - \alpha_j v_j$
6: $\quad \beta_{j+1} = \|\omega_j\|_2$. If $\beta_{j+1} = 0$ then Stop
7: $\quad v_{j+1} = \omega_j/\beta_{j+1}$
8: **End**

We can express this algorithm trough its matrix formulation, as we have done for the Arnoldi algorithm obtaining:

$$MV_m = V_m T_m + \beta_{m+1} v_{m+1} e_m^T. \tag{3.58}$$

It is also possible to use a similar approach for the case with M non-symmetric. The main advantage is the preservation of the property of having short-recurrence of the Lanczos algorithm. In the other hand, this technique can leads to serious break downs. Starting from the vector ξ and from a second vector $\bar{\xi}$ orthogonal to the first, the idea is to construct a biorthogonal basis of the subspaces $K_m(M, \xi)$ and $K_m(M^T, \bar{\xi})$. The algorithm proceed analogously to the (symmetric) Lanczos algorithm. It produce $V_m = [v_1, \ldots, v_m]$ and $W_m = [w_1, \ldots, w_m]$ basis of, respectively, $K_m(M, \xi)$ and $K_m(M^T, \bar{\xi})$ so that $W_m^T V_m = I_m$. We obtain the following algorithm.

Algorithm 3 Non-Symmetric Lanczos Algorithm

1: Compute $v_1 = \xi/\|\xi\|$ and select ω_1 such that $(v_1, \omega_1) = 1$.
2: **For** $j = 1, 2, \ldots, m$ **do**
3: $\quad \alpha_j = (Mv_j, \omega_j)$
4: $\quad \hat{v}_{j+1} = Mv_j - \alpha_j v_j - \beta_j v_{j-1}$
5: $\quad \hat{\omega}_{j+1} = M^T \omega_j - \alpha_j \omega_j - \delta_j \omega_{j-1}$
6: $\quad \delta_{j+1} = |(\hat{v}_{j+1}, \hat{\omega}_{j+1})|^{\frac{1}{2}}$. If $\delta_{j+1} = 0$ then Stop
7: $\quad \beta_{j+1} = (\hat{v}_{j+1}, \hat{\omega}_{j+1})/\delta_{j+1}$
8: $\quad \omega_{j+1} = \hat{\omega}_{j+1}/\beta_{j+1}$
9: $\quad v_{j+1} = \hat{v}_{j+1}/\delta_{j+1}$
10: **End**

The corresponding matrix formulation is

$$\begin{aligned} MV_m &= V_m T_m + \delta_{m+1} v_{m+1} e_m^T, \\ M^T W_m &= W_m T_m^T + \beta_{m+1} w_{m+1} e_m^T, \end{aligned} \tag{3.59}$$

where

$$T_m = \begin{bmatrix} \alpha_1 & \beta_2 & & \\ \delta_2 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_m \\ & & \delta_m & \alpha_m \end{bmatrix} \tag{3.60}$$

is still a tridiagonal matrix but it not symmetric any more. In both symmetric and non-symmetric Lanczos, we can compute an analogous approximation to (3.53) of the action of the matrix exponential on a vector:

$$e^M \xi \approx \|\xi\|_2 V_m e^{T_m} e_1. \tag{3.61}$$

However, in the non-symmetric case, the fact that V_m is no longer orthogonal could lead to numerical difficulties caused by the possibly very large norm of V_m and by the instability due to non-orthogonal basis.

The third strategy we describe can be seen as an improvement of the first two. We will use the notation in (3.53) for identify both the Arnoldi based approximation (3.53) and the Lanczos based approximation (3.61). Note that both of them have been defined by means of the first m vectors even if v_{m+1} was also available. Therefore, the natural question is if it is possible to use all the generated vector to obtain a more precise approximation. Let us consider the function

$$\phi(z) = \frac{e^z - 1}{z}. \tag{3.62}$$

Then, we can write

$$e^M \xi = \xi + M\phi(M)\xi, \tag{3.63}$$

Note that the process we did to approximate $e^M \xi$ by the Arnoldi (or Lanczos) algorithm can be applied to other for a general matrix functions, for more details see [20, Section 13.2.2]. In this case, we can approximate $\phi(M)\xi$ by

$$\phi(M)\xi \approx V_m \phi(H_m)\|\xi\|_2 e_1. \tag{3.64}$$

Denoting by s_m the error $s_m = \phi(M)v_1 - V_m\phi(H_m)e_1$ and using the recurrence (3.49), we obtain

$$\begin{aligned} e^M v_1 &= v_1 + M\phi(M)v_1 \\ &= v_1 + M(V_m\phi(H_m)e_1 + s_m) \\ &= v_1 + (V_m H_m + h_{m+1,m} v_{m+1} e_m^T)\phi(H_m)e_1 + Ms_m \\ &= V_m[e_1 + H_m\phi(H_m)e_1] + h_{m+1,m} e_m^T \phi(H_m) e_1 v_{m+1} + Ms_m \\ &= V_m e^{H_m} e_1 + h_{m+1,m} e_m^T \phi(H_m) e_1 v_{m+1} + Ms_m. \end{aligned} \tag{3.65}$$

The results from (3.65) prompt us to use the following approximation

$$e^M \xi \approx \|\xi\|_2 [V_m e^{H_m} e_1 + h_{m+1,m} e_m^T \phi(H_m) e_1 v_{m+1}]. \tag{3.66}$$

As indicated in (3.65), the error in this approximation is $\|\xi\|_2 M s_m$. The a-priori error analysis in [31, Section 4.3] shows that the upper bound of the error of the m-approximation of the improved schemes based on Arnoldi (Lanczos) is the same of the one obtained for the $m+1$-approximation of the standard Arnoldi (Lanczos) schemes.

The remaining issue to be addressed is the computation of $e^{H_m} e_1$ or $\phi(H_m)e_1$. In [13],[31], it is shown that is possible to construct rational approximations extremely accurate even with relatively small degree. One reason is the simple structure of the matrix H_m that is either upper Hessenberg or Tridiagonal. Moreover, we recall that the dimension of H_m is significantly smaller than the one of M. These two properties contribute to considerably lower the computational costs.

Chebyshev technique The last method we present make use of Chebyshev polynomials. Even if this method can be used to compute both a matrix exponential and the action of a matrix exponential on a vector, we decided to present it in this section because, as stated in [7] , it is typically used in numerical quantum dynamics to compute $e^{-i\tau H}\psi_0$ over very long times. Therefore, putting ourself in this setting, we consider an Hermitian matrix H and $\tau > 0$. The main idea is to truncate the Chebyshev polynomials expansion of the exponential function $e^{-i\tau H}$. To apply this procedure we need to previously bound the extreme eigenvalues $\lambda_{min}, \lambda_{max}$ of H. Then, we can consider the truncated Chebyshev expansion of e^{-ix} on the interval $[\tau\lambda_{min}, \tau\lambda_{max}]$

$$e^{-ix} \approx \sum_{n=0}^{m} c_n Q_n(x), \tag{3.67}$$

with appropriately chosen coefficients c_n and with

$$Q_n(x) = T_n\left(\frac{2x - \tau\lambda_{max} - \tau\lambda_{min}}{\tau\lambda_{max} - \tau\lambda_{min}}\right). \tag{3.68}$$

Here, T_n are the Chebyshev polynomials on the interval $[-1, 1]$ determined by the recurrence

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_{n+1} = 2xT_n(x) - T_{n-1}(x). \tag{3.69}$$

In conclusion, we can use the approximation

$$e^{-i\tau H} \approx \sum_{n=0}^{m} c_n Q_n(\tau H). \tag{3.70}$$

As reported in [4], the advantages are the low memory demands, simple implementation, unconditional stability and accuracy for arbitrary large propagation times. However, especially for time-dependent Hamiltonian operators, the need to determinate a-priori bounds on the eigenvalues of the matrix H represents an important disadvantage.

4. Numerical comparison

As described in [3], [30], NMR consists of the oscillatory response to resonant excitation of nuclei with non-zero spins in a magnetic field. This excitation is performed by radio-frequency irradiations, that are reported as the so-called chemical shift, i.e., the ratio between the frequency of a nucleus and the nucleus of a standard compound (e.g., tetramethylsilane). We can characterize the NMR interaction depending on whether they depend on the sample orientation with respect to the direction of the magnetic field (anisotropic) or not (isotropic). In solid-state NMR spectroscopy, anisotropic interactions as chemical shift and dipolar couplings (that depend on internuclear distances) dominates, [3]. In most cases, we use magic-angle spinning (MAS), [30]. The magic angle θ_M is given from the fact that the anisotropy NMR interactions can be expressed as a tensor whose time average vanishes at θ_M. Thus, the spacial dependencies are average out over time. However, this does not mean that the anisotropy components does not contribute to the system but only that their contribution is soften from the rapid sample spinning. Therefore, in the description of the dynamic system we need to include also the anisotropy components.

In order to describe the Hamiltonian in (1.1) for the problems we consider in the following sections, we need to first recall the definition of Pauli matrices. Therefore, following [11], we define

$$\sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \sigma_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad \sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{4.1}$$

These are Hermitian matrices with the same characteristic equation: $\lambda^2 - 1 = 0$. Moreover, it can be easily verified that together with the 2×2 identity matrix $\mathbb{I}_2$, they form a basis for the space of 2×2 complex matrices,[11, Complément A(IV)]. They are useful to describe the components of a spin. In other words, if we consider a system with N_s spin-1/2, we can define the following matrix for the spins $k = 1, \cdots, N_s$:

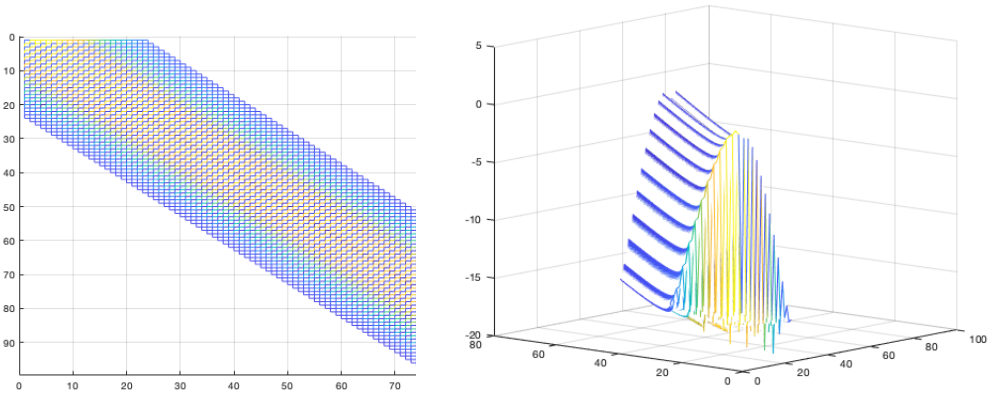
$$I_{k_x} := \mathbb{I}_{2^{k-1}} \otimes \sigma_x \otimes \mathbb{I}_{2^{N_s-k}}. \tag{4.2}$$

Analogously we define I_{k_y} and I_{k_z}. Notice that all these matrices have size 2^{N_s}. In this Chapter, for the sake of simplicity, we consider homonuclear systems, i.e., composed of atoms of the same elements.

4.1 Bloch-Seigert dynamics

As first example, we take into account a small but significant case to show the characteristic of the $\star$-approach discussed in Chapter 2. We have selected a problem that can be seen as one of the simplest model where it is possible to see physical effects due to the time-dependency of the Hamiltonian, [14]. We take into account a system composed of a single spin-1/2 whose dynamic can be described by the Schrödinger equation

$$\frac{\mathrm{d}}{\mathrm{d}t}u(t) = -i\mathcal{H}_{BS}(t)u(t), \quad u(0) = u_0, \quad t \in [0, T], \tag{4.3}$$



(a) Banded Property **(b)** Decay phenomenon

Figure 4.1: Coefficient Matrix Properties
On the left, the structure of the finite coefficient matrix $F_{100}^{(24)}$. The coloured parts represent non-zero entries, while the white part represent the zero entries. It is clear that this matrix is banded. *On the right*, the magnitude of the elements of the coefficient matrix $F_{100}^{(24)}$ in logarithmic scale. We notice the decay phenomenon.

where the Hamiltonian is the Bloch-Seigert Hamiltonian:

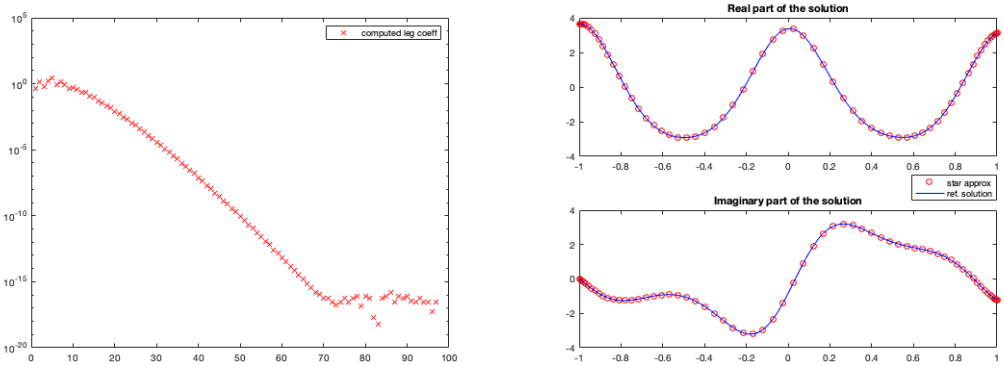
$$\mathcal{H}_{BS} = \begin{bmatrix} \omega_0/2 & 2\beta\cos(\omega_r t) \\ 2\beta\cos(\omega_r t) & -\omega_0/2 \end{bmatrix} \tag{4.4}$$

Recalling the definition (4.1) of the Pauli matrices, we can split the Hamiltonian $\mathcal{H}_{BS}$ into a time-independent component and a time-dependent one, i.e.,

$$\mathcal{H}_{BS} = \frac{\omega_0}{2}\sigma_z + 2\beta\cos(\omega_r t)\sigma_x. \tag{4.5}$$

We consider the resonant ($\omega_0 = \omega_r$) strong-coupling ($\beta/\omega_r > 1$) case. In particular, we consider the rotator frame frequency $\omega_r = 20000$ and $\beta = 1.2\omega_r$. We make use of the $\star$-process generalized to the matrix case, where the system (2.54) is solved using the iterative method (2.57) with low-rank approximations [1]; see Section 2.6. The aim is to show the main characteristic of the objects involved in the $\star$-process. The time interval is considered to be exactly the length of one period of the Hamiltonian, i.e., $t \in [0, 2\pi/\omega_r]$. Then, we scale the time-dependent component of $\mathcal{H}_{BS}$ in order to study it with $t \in [-1, 1]$. As initial vector, we set a random vector of the appropriate size using the MATLAB function `randn`. We take tolerance $\delta_{tol} = 10^{-12}$ for the stopping criterion (2.59). Moreover, performing some tests, we came to the conclusion that a good choice for the truncation parameter for this problem is $M = 100$. In Section 2.3, we described the importance of banded matrix in the $\star$-process and the decay phenomenon of the elements in the coefficient matrix. Thus, we analyse the computed matrix in order to check whether it satisfies these properties. In Figure 4.1a, we show the structure of the matrix $F_M^{(N)}$ corresponding to our problem. We can clearly see that $F_M^{(N)}$ is a banded matrix as expected from the theory and that $N = 24$. Moreover,

[1] The author thanks Dr.Pozza for providing the MATLAB code corresponding to the full implementation of the $\star$-process.



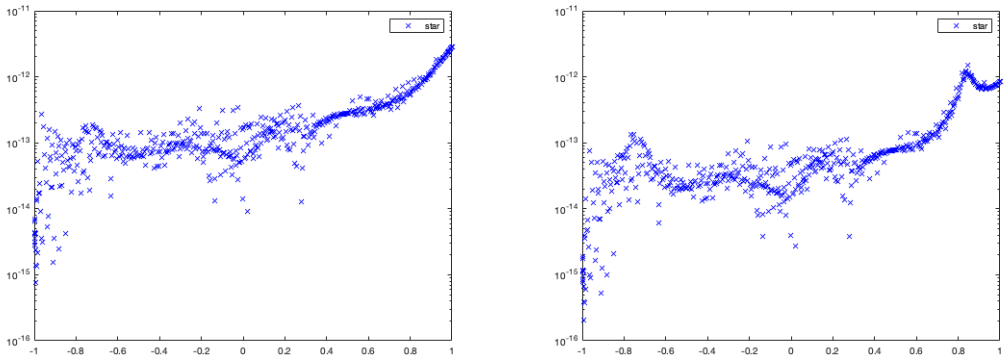
(a) Decay of the Legendre coefficients **(b)** Solution

Figure 4.2: Solution and its Legendre coefficients
On the left, Legendre coefficients of the solution of (4.3) computed by the $\star$-process. We notice that there is a decay phenomenon. *On the right*, comparison between the solution of problem (4.3) computed by the $\star$-algorithm (red) and by the reference method RK89 (blue).

in Figure 4.1b, by plotting the magnitude of the elements of the coefficient matrix in logarithmic scale, we notice the described decay phenomenon. Once the computations are completed, we obtain the approximated Legendre coefficients of the solution of (4.3). As the solution of the ODE is analytic, we expect the coefficients to decay superexponentially. In Figure 4.2a, we can see that, indeed, the computed coefficients decay. Of course, the decay stops when the machine precision is reached.

The computed solution is compared with a reference method to show the reliability of the $\star$-process. As reference, we choose the Runge-Kutta $(8, 9)$ (RK89) method with tolerance 10^{-12} implemented by the MATLAB function `ode89`. In Figure 4.2b, a comparison of the real and imaginary parts of the solution computed from the two methods is shown. We can see that the solution given by the $\star$-approach is perfectly aligned with the one from the reference method. Moreover, we compute the relative error for the $\star$-process as the difference in absolute value of the solution of the reference method (RK89 with tolerance 10^{-12}) with the solution of the $\star$-algorithm in the same points where the reference solution was computed. Further, we compute the absolute error analogously. The results can be seen in Figure 4.3. We can clearly see that the $\star$-process is reliable. To compare the $\star$-approach with the Runge-Kutta methods, we perform the same test as before (with the same initial condition vector) but taking into account the $\star$-process with tolerance $\delta_{tol} = 10^{-4}$ and the RK89 and Runge-Kutta[2] $(7, 8)$ (RK78) methods with tolerance 10^{-3}. Then, we analyse the resultant errors taking as reference the $\star$-solution previously computed (with tolerance 10^{-12}). In Figure 4.4, we can see that the $\star$-process has a smaller error compared to the Runge-Kutta methods. We have repeated the same experiment 20 times so to fairly compare also computational times. The average of the total computational times are reported in Table 4.1 together with the other main results of this experiment. At first sight, we notice that the $\star$-algorithm is slower than the Runge-Kutta meth-

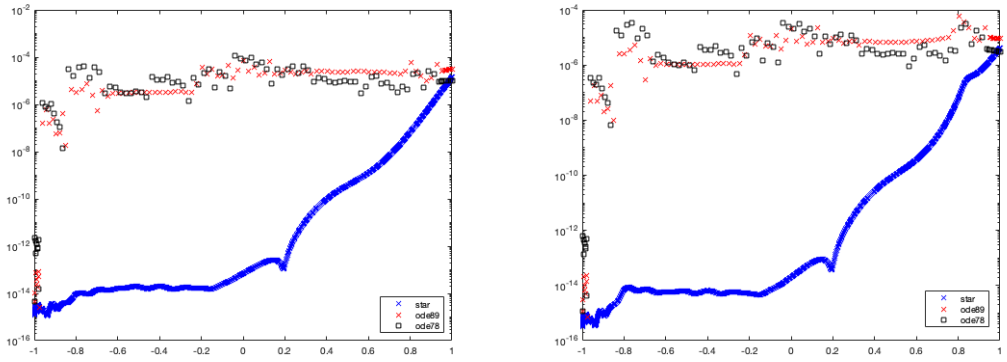
[2]RK78 can be implemented with the function `ode78` in MATLAB.



(a) Absolute Error **(b)** Relative Error

Figure 4.3: Comparison of Errors
Absolute (*left*) and Relative (*right*) errors of the $\star$-process with tolerance $\delta_{tol} = 10^{-12}$. Reference solution obtained with RK89 with tolerance 10^{-12}.

ods. However, the computational times taken from the $\star$-process to compute the solution of the linear system are closer to the total computational times of `ode89` and `ode78`. This is crucial because, for larger problems, the total computational time of the $\star$-algorithm will be dominated by the time taken to solve the linear system and not by the discretization time as for the small problems. Therefore, we can predict that for large dimension ODEs the $\star$-process might be a valid alternative to the classical ODE solvers also in terms of computational time. We design the next experiment with the purpose of verifying it.



(a) Absolute Error **(b)** Relative Error

Figure 4.4: Comparison of Errors
Absolute (*left*) and Relative (*right*) errors of the reference $\star$-process with tolerance $\delta_{tol} = 10^{-4}$ (blue), RK89 (red crosses) and RK78 (black squares) both with tolerance 10^{-3}. Reference solution obtained with $\star$-process with tolerance 10^{-12}.

Table 4.1: Comparison on Bloch-Seigert dynamics The main results of the Bloch-Seigert experiment are reported. As total computational times we report the average of the total computational times over 20 attempts.

Method	Tolerance	Relative error	Absolute error	Total computational time	Linear systems total time	Discretization time
RK89 (ref.)	10^{-12}	/	/	$0.0480s$	/	/
$\star$-process ($M = 100$)	10^{-12}	10^{-13}	10^{-11}	$0.2526s$	$0.1769s$	$0.0613s$
$\star$-process ($M = 100$)	10^{-4}	10^{-7}	10^{-6}	$0.1136s$	$0.0568s$	$0.0419s$
RK89	10^{-3}	10^{-6}	10^{-4}	$0.0279s$	/	/
RK78	10^{-3}	10^{-6}	10^{-4}	$0.0238s$	/	/

4.2 Dipolar Interactions in the rotor-frame

As a second example, we consider a homonuclear system described by the simplified, but yet descriptive, expression of the Hamiltonian $\mathcal{H}$ through the isotropic chemical shifts and the dipolar interactions:

$$\mathcal{H}(t) = \mathcal{H}_{ICS} + \mathcal{H}_{DD}(t), \tag{4.6}$$

We omitted the anisotropic chemical shifts, the radio-frequency pulses, quadripolar interactions and the indirect spin-spin couplings. In the following, the considered terms are defined according to [5], and a more detailed expression of $\mathcal{H}$ can be found in the same source.

Dipole-dipole coupling Let us consider the MAS frequency $\omega_r = 2\pi/\tau_r$, where τ_r is the MAS period. Moreover, consider the so-called Euler angles, i.e., three angles α, β and γ that enable us to rotate the axis of reference from the orientation of a molecule in the MAS rotor to a fixed system of axis, for more details see [25]. Then, we can define the dipole-dipole coupling component of the Hamiltonian through its Fourier expansion:

$$\mathcal{H}_{DD} = \sum_{k=1}^{N_s-1} \sum_{q=k+1}^{N_s} \left(\sum_{n=-2}^{2} \omega_{kq}^{(n)}(\alpha, \beta, \gamma) e^{in\omega_r t} \right) \cdot \left[2I_{k_z} I_{q_z} - (I_{k_x} I_{q_x} + I_{k_y} I_{q_y}) \right]. \tag{4.7}$$

Depending on the reference system of choice the expression for the coefficients $\omega_{kq}^{(n)}(\alpha, \beta, \gamma)$ changes. Here, we consider the rotator frame system. In this case, we can express the homonuclear dipolar Hamiltonian as in [21]:

$$\mathcal{H}_{DD} = \delta \sum_{k=1}^{N_s-1} \sum_{q=k+1}^{N_s} \left(\sum_{n=-2}^{2} e^{in(\gamma+\omega_r t)} C_n(\beta) \right) \cdot \frac{1}{r_{kq}^3} \left[2I_{k_z} I_{q_z} - (I_{k_x} I_{q_x} + I_{k_y} I_{q_y}) \right], \tag{4.8}$$

where r_{kq} is the interspin distance between spins k and q,

$$\begin{aligned} C_1(\beta) = C_{-1}(\beta) &= -\frac{1}{2\sqrt{2}} \sin(2\beta), \\ C_2(\beta) = C_{-2}(\beta) &= \frac{1}{4} \sin^2(\beta), \end{aligned} \tag{4.9}$$

Table 4.2: Coefficients of CF6:5Opt, as in [4]
To find the coefficients that are not listed use the property (OC1).

$f_{1,1} = 0.1714$	$f_{1,2} = 0.15409059414309687213$	$f_{1,3} = 0.11947178242929061641$
$f_{2,1} = 0.37496374319946236513$	$f_{2,2} = 0.13813675394387646682$	$f_{2,3} = -0.13090674649282935743$
$f_{3,1} = 1 - 2f_{2,1} - 2f_{1,1}$	$f_{3,2} = 0$	$f_{3,3} = -2f_{2,3}$

and the constant δ is taken to be as in [24, Supplementary Information], i.e.,

$$\delta = -\frac{1}{2}\frac{\mu_0}{4\pi}\frac{\gamma_h \hbar}{10^{-30}\pi}, \tag{4.10}$$

where $\hbar \approx 1.05457266 \cdot 10^{-34}$ is the reduced Planck constant, $\gamma_h \approx 2.67522128 \cdot 10^8$ is the proton gyromagnetic ratio and $\mu_0/4\pi = 10^{-7}$ is the vacuum permeability. Notice that in the rotator frame system we can loose the dependency from the Euler angle α. If we plug the constants $C_m(\beta)$ in (4.8), then we can rewrite it as

$$\begin{aligned}\mathcal{H}_{DD} = \delta &\left(-\frac{1}{2\sqrt{2}}\sin(2\beta)\cos(\gamma + \omega_r t) + \frac{1}{4}\sin^2(\beta)\cos(2\gamma + 2\omega_r t)\right) \\ &\cdot \left(\sum_{k=1}^{N_s-1}\sum_{q=k+1}^{N_s}\frac{1}{r_{kq}^3}\left[2I_{k_z}I_{q_z} - (I_{k_x}I_{q_x} + I_{k_y}I_{q_y})\right]\right).\end{aligned} \tag{4.11}$$

Isotropic chemical shift This is the time-independent part of the Hamiltonian and it is not modulated by the MAS rotation, [24]. It can be defined as follow:

$$\mathcal{H}_{CS} = \sum_{k=1}^{N_s}\Omega_k I_{k_z}, \tag{4.12}$$

where Ω_k are values representing chemical shift differences (dependent by the magnetic field) and they are given by the physical property of the sample.

To maintain the link with a real-world problem, we consider raw molecular data of the cationic tin oxo-cluster [3]. In other words, as input molecule, we select a subset of this database composed by the first N_s protons so to have a system with N_s spins. We report the first 13 of them in Appendix A. In solid-state NMR it is, in general, hard to precisely define the chemical shifts of the nuclei. As a consequence, we decide to define the chemical shifts differences Ω_k as a fix randomly generated vector. We select $\omega_r = 20000$ and we consider the Schrödinger equation with Hamiltonian as in (4.6) for $t \in [0, \frac{1}{10\omega_r}]$. Moreover, we select $\beta = \pi/4$ and $\gamma = \pi$. As initial condition vector u_0 we take a random vector of the appropriate size.

We want to compare the $\star$-process against other already established methods for solving non-autonomous ODEs. For the $\star$-process we consider the same implementation used in the Bloch-Seigert experiment that makes use of the strategy presented in Section 2.6. While, to represent the class of geometric numerical

[3]The data (by Dr.F.Ribot) are taken form the website of P.-L. Giscard: http://www-lmpa.univ-littoral.fr/ plgiscard/CationicOxoCluster.txt

integrators presented in Chapter 3, we consider the methods described below. For all of them, we make use of Gauss-Legendre quadrature (Section 3.5) to evaluate integrals and, since all the involved matrices are Hermitian, the Lanczos algorithm (Section 3.6.2) for the computation of the matrix exponentials.

MI6 We implement in MATLAB the 6-th order Magnus Integrators according to the definition in [7]. If, for a time step τ, we define

$$\begin{aligned}\mathcal{H}^{(1)} &= \mathcal{H}\left(n\tau + \left(\frac{1}{2} - \frac{\sqrt{15}}{10}\right)\tau\right),\\ \mathcal{H}^{(2)} &= \mathcal{H}\left(n\tau + \frac{1}{2}\tau\right),\\ \mathcal{H}^{(3)} &= \mathcal{H}\left(n\tau + \left(\frac{1}{2} + \frac{\sqrt{15}}{10}\right)\tau\right),\end{aligned} \tag{4.13}$$

and

$$\alpha_1 = \tau\mathcal{H}^{(2)}, \quad \alpha_2 = \frac{\sqrt{15}\tau}{3}(\mathcal{H}^{(3)} - \mathcal{H}^{(1)}), \quad \alpha_3 = \frac{10\tau}{3}(\mathcal{H}^{(3)} - 2\mathcal{H}^{(2)} + \mathcal{H}^{(1)}) \tag{4.14}$$

then, we have

$$\Omega^{[6]} = \alpha_1 + \frac{1}{12}\alpha_3 + \frac{1}{240}[-20\alpha_1 - \alpha_3 + C_1, \alpha_2 + C_2], \tag{4.15}$$

where $C_1 = [\alpha_1, \alpha_2]$ and $C_2 = -1/60[\alpha_1, 2\alpha_3 + C_1]$. Finally, the solution is found by

$$u_{n+1} = e^{-i\Omega^{[6]}}u_n. \tag{4.16}$$

CF4:2 The 4-th order optimized Commutator-free exponential integrator with 2 exponentials can be easily implemented considering the Gauss-Legendre quadrature formula, [4]. Indeed, we have

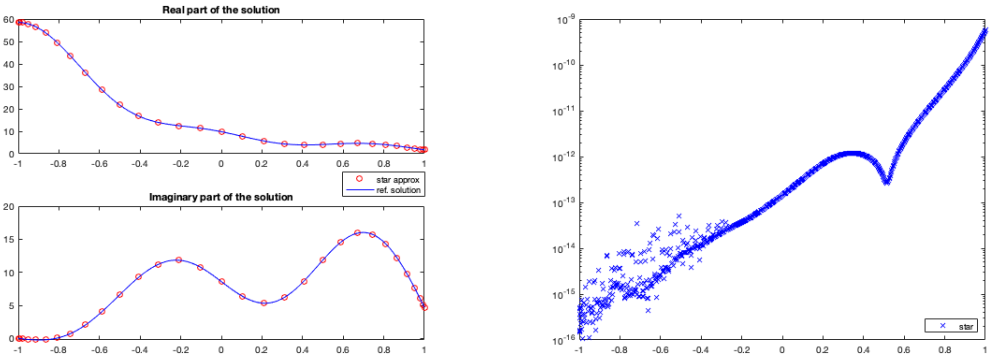
$$\begin{aligned}\Omega_1 &= \frac{3 - 2\sqrt{3}}{12}\mathcal{H}^{(1)} + \frac{3 + 2\sqrt{3}}{12}\mathcal{H}^{(2)},\\ \Omega_2 &= \frac{3 + 2\sqrt{3}}{12}\mathcal{H}^{(1)} + \frac{3 - 2\sqrt{3}}{12}\mathcal{H}^{(2)},\end{aligned} \tag{4.17}$$

where $\mathcal{H}^{(1)} = \mathcal{H}[\tau(1/2 - \sqrt{3}/6)]$ and $\mathcal{H}^{(2)} = \mathcal{H}[\tau(1/2 + \sqrt{3}/6)]$. Then,

$$\tilde{U}_{CF4:2} = e^{-i\tau\Omega_1}e^{-i\tau\Omega_2}. \tag{4.18}$$

CF6:5Opt As for the other CFE integrators, the 6-th order optimized Commutator-free exponential integrator with 5 exponentials is uniquely defined by its coefficients $f_{i,n}$; see equation (3.30). We have reported them, according to [4], in Table 4.2. Notice that for this method at each iteration we need to compute five action of matrix exponential on a vector.

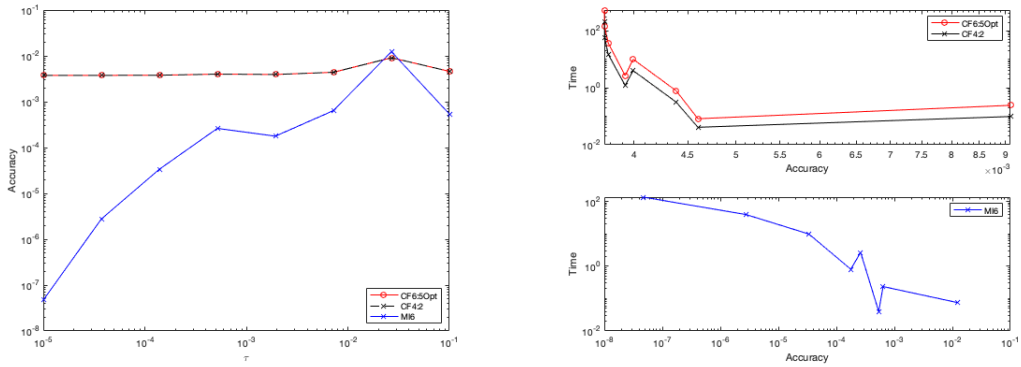
The accuracy of a Magnus Integrator and of a CFE integrator is controlled by the choice of the time-step τ. Indeed, for a ℓ-th order CFE integrator the



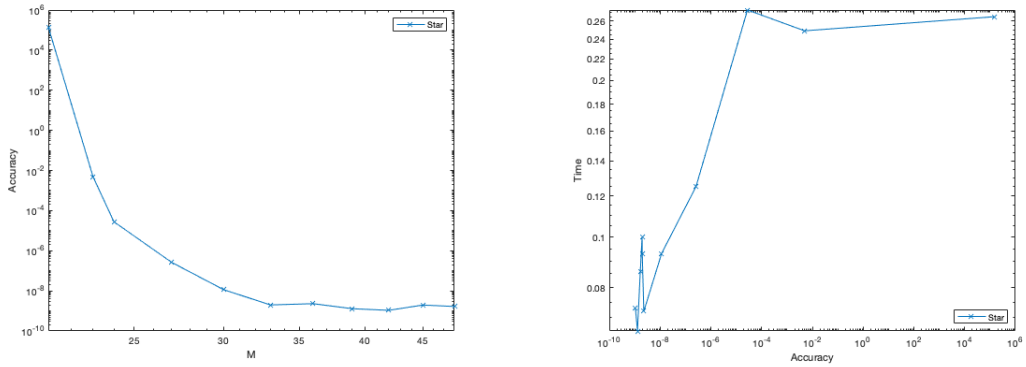
(a) Computed solutions **(b)** Relative errors

Figure 4.5: Reliability of $\star$-solution
On the left, the solutions computed by the reference RK89 (blue) and by the $\star$-algorithm (red). The solutions are perfectly aligned. *On the right*, the relative errors of the $\star$-algorithm took as comparison with reference RK89 with tolerance 10^{-13}.

error can be written as $\epsilon = cT\tau^{\ell}$ with c some constant linked to the particular problem, [4]. For the $\star$-process a similar role is played by the truncation parameter M. Therefore, we compare the described methods and the $\star$-process on the necessary value of, respectively, τ and M to reach a fixed accuracy and on the total computational time to achieve it. To this end, we consider a system of $N_s = 6$ spins, and so with Hamiltonian of dimension $2^6 \times 2^6$. First of all, as in the Bloch-Seigert experiment, we select an appropriate reference solution. We initially consider RK89 with tolerance 10^{-13} as reference and we test the $\star$-approach with tolerance $\delta_{tol} = 10^{-13}$ and $M = 40$. Analysing the computed solutions in Figure 4.5a and the relative errors in Figure 4.5b, we can conclude that the $\star$-algorithm with tolerance 10^{-13} produces a solution that can be taken as reference solution for accuracies of order up to 10^{-9}. As a consequence, we can use it to see the dependency between the parameters M and τ and the reached accuracy of the respective methods. In Figure 4.6a and 4.6c, we report the behaviour of the methods with respect to some sample parameters. We notice that for the Commutator-free exponential integrators even with really small time step (e.g., 10^{-5}) we are not able to reach good accuracies. For the Magnus integrator, we see that for a step size smaller than 10^{-3} we can reach higher accuracies. However, as shown in Figure 4.6b, this has a price in term of computational time. For the $\star$-process, the dependencies of the accuracy on the choice of the truncation parameter is clearly visible and even with small values of M we reach accuracies of the order of 10^{-8}. It is also important to notice that for $M \geq 35$ the accuracy remains approximately the same. Therefore, to gain computational time, one should consider $M = 35$. In Figure 4.6b and 4.6d, we show how the computational time is affected by the reached accuracy.



(a) Accuracy vs τ **(b)** GNI: Accuracy vs Time



(c) Accuracy vs M **(d)** $\star$-process: Accuracy vs Time

Figure 4.6: Accuracies
On the left, accuracies with respect to the reference $\star$-solution depending on the choice of the parameters τ and M. *On the right*, dependences between achieved accuracy and computational time.

We notice that, for the Geometric numerical integrators, the smaller time step is taken the longer the method will take to compute the solution. While, for the $\star$-process, it can happen that, even if we are reaching higher accuracies, the computational time is reduced. To compare the behaviour of the methods we fix the aimed accuracy and perform the same test 20 times. We report the needed parameter M or τ and the average computational time. We have chosen the accuracies $10^{-3}, 10^{-5}$, and 10^{-8}. It is important to underline that all of the methods are able to reach even higher accuracies. However, if for the $\star$-process this might not take significantly more time, for the other methods this represents an important increase in cost and time. The results are reported in Table 4.3. We notice that the higher is the aimed accuracy the more we can see a difference in computational times between methods. Moreover, we stress that for the $\star$-process to reach a significantly higher accuracy a slightly bigger values for M is needed. Instead, for the other methods we are force to heavily decrease the time step to gain more accuracy.

Table 4.3: Comparison on Dipolar interaction experiment
The main results of the Dipolar interaction experiment are reported. As parameters we selected the minimal (maximal) value of M (τ) such that the fixed accuracy has been reached. The average times over 20 attempts are taken in accordance to the computation with the selected parameter.

Accuracy	Method	Parameter	Total computational time
10^{-3}	$\star$ -process	$M = 23$	$0.1969s$
	MI6	$\tau = 10^{-\frac{20}{21}}$	$0.0463s$
	CF6:5Opt	$\tau = 10^{-1}$	$0.1324s$
	CF4:2	$\tau = 10^{-1}$	$0.0701s$
10^{-5}	$\star$ -process	$M = 24$	$0.1370s$
	MI6	$\tau = 1.4 \cdot 10^{-4}$	$10.3845s$
	CF6:5Opt	$\tau < 10^{-5}$	$> 540s$
	CF4:2	$\tau < 10^{-5}$	$> 220s$
10^{-8}	$\star$ -process	$M = 30$	$0.0766s$
	MI6	$\tau = 10^{-5}$	$149.9658s$

As a last test, we analyse the dependencies between the total computational time and the size of the problem. In particular, considering the dipolar interaction problem given by the Hamiltonian (4.6), we show the increments in computational times depending on the number of considered spins, and so on the dimension of the Hamiltonian. We compare the $\star$-process and the MI6 that, from the last test, appears to be the Geometric numerical integrator that achieves higher accuracies with bigger step sizes.

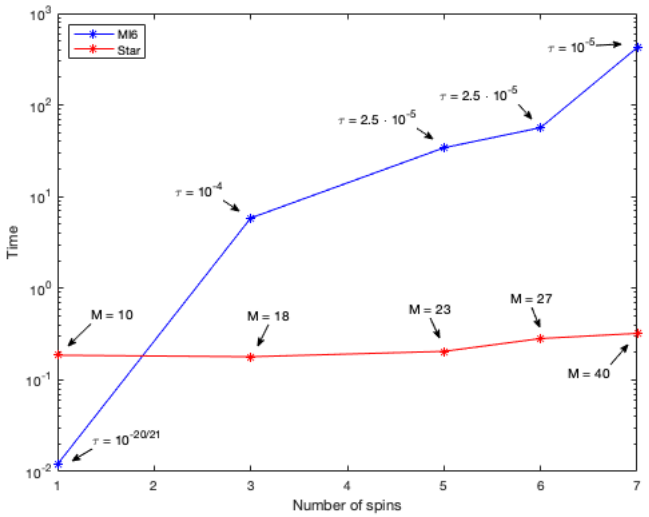


Figure 4.7: Time vs Number of spins
The parameter M and τ are chosen in such a way that a relative error of order at least 10^{-7} is reached.

As for the previous experiment, we consider the $\star$-solution computed with tolerance 10^{-13} as reference. We fix randomly generated initial condition vectors of the appropriate size and we select parameters M and τ in such a way that an accuracy of at least 10^{-7} is reached. In Figure 4.7, we show the results. We notice that for the smallest problem the MI6 is faster than the $\star$-algorithm. However, in the $\star$-process, when the total computational time starts to be dominated by the time for the computation of the linear system solution (instead of discretization time), the $\star$-approach clearly outperform our implementation of MI6.

Conclusion

In this work we have described and analysed a newly introduced method for the solution of non-autonomous ODEs and its established alternatives. First, we discussed how, through a convolution-like product, it is possible to express the solution of a non-autonomous ODE in such a way that, by its expansion in Legendre polynomials, the original problem is transformed into a linear algebra problem, the infinite matrix problem. This lead us to analyse how to tackle each subproblem of it. We described how to approximate the coefficient matrix by a banded matrix and, then, by a finite matrix. As a consequence, we were able to show how to numerically solve the infinite linear system and how to recover enough coefficients of the Legendre expansion of the solution of the original problem.

We discussed the Geometric numerical integrators. The considered methods are based on a expression of the solution of the non-autonomous ODE through the Magnus Expansion, which has been described and derived. From it, the Magnus integrators follow. To give a complete description of them, we recalled the concepts of quadrature formulas and methods for the approximation of a time-independent matrix exponential. Then, we analysed strategies to overcome the difficulties that could arise with the design of higher order Magnus integrators. Therefore, the Commutator-free exponential integrators were introduced. This class of methods can be seen as a generalization of the Splitting methods, which has also been briefly described. However, they are still strictly linked to the Magnus expansion as we analysed when discussing on order conditions.

We have performed experiments to understand whether the $\star$-algorithm can be seen as a valid option to solve non-autonomous ODEs. To this end, we have taken a study case on which we have compared the methods. The MAS NMR dynamics is an important application of the described methods. As a first experiment, we have considered one of simplest, but still significant, problem in NMR and we have shown that the characteristics of the $\star$-process described in the theory can be found also in practice. Then, we have considered a more descriptive Hamiltonian for an homonuclear system, and we have implemented in MATLAB the codes for some representatives of the Geometric numerical integrators. The results underline how the $\star$-process is suitable for large problems rather than small one. We are aware that our code of the Geometric numerical integrators can not represent their optimal implementation and that there are other methods to be tested. However, from our experiments, we deduce that the $\star$-process is not only a reliable and valid alternative to the well-established methods, but, especially with the increasing of the dimension of the problem, it might be achieving higher accuracy with less efforts.

This work can be seen as a starting point for the analysis of the comparison between the $\star$-process and the commonly used methods for non-autonomous ODEs. In conclusion, while we acknowledge the limitations of this thesis, we firmly believe that it offers valuable insights that can be further explored and expanded upon.

Acknowledgements The author would like to express deep appreciation to Dr. Stefano Pozza, the supervisor of this work, for providing invaluable guidance and support, both academic and personal, throughout the course of this project. The author would also like to thank Prof. Christian Bonhomme for the assistance provided in comprehending NMR theory.

Bibliography

[1] A. Al-Mohy and N. Higham. A new scaling and squaring algorithm for the matrix exponential. *SIAM Journal on Matrix Analysis and Applications*, 31, 01 2009.

[2] A. Al-Mohy and N. Higham. Computing the action of the matrix exponential, with an application to exponential integrators. *SIAM Journal on Scientific Computing*, 33, 01 2011.

[3] A. Alia, S. Ganapathy, and H.J. de Groot. Magic angle spinning (MAS) NMR: a new tool to study the spatial and electronic structure of photosynthetic complexes. *Photosynthesis Research*, 102:415–25, 2009.

[4] A. Alvermann and H. Fehske. High-order commutator-free exponential time-propagation of driven quantum systems. *Journal of Computational Physics*, 230(15):5930–5956, July 2011.

[5] M. Bak, J.T. Rasmussen, and N.C. Nielsen. SIMPSON: A general simulation program for solid-state NMR spectroscopy. *Journal of Magnetic Resonance*, 147(2):296–330, 2000.

[6] S. Blanes, F. Casas, and A. Murua. Splitting and composition methods in the numerical integration of differential equations. *Boletín de la Sociedad Española de Matemática Aplicada*, page 89–145, 2008.

[7] S. Blanes, F. Casas, J.A. Oteo, and J. Ros. The Magnus expansion and some of its applications. *Physics Reports*, 470(5-6):151–238, January 2009.

[8] S. Blanes, F. Casas, and M Thalhammer. High-order commutator-free quasi-Magnus exponential integrators for non-autonomous linear evolution equations. *Computer Physics Communications*, 220:243–262, November 2017.

[9] S. Blanes, F. Diele, C. Marangi, and S. Ragni. Splitting and composition methods for explicit time dependence in separable dynamical systems. *Journal of Computational and Applied Mathematics*, 235(3):646–659, 2010.

[10] F. Castella, P. Chartier, S. Descombes, and G. Vilmart. Splitting methods with complex times for parabolic equations. *BIT Numerical Mathematics*, 49:487–508, 09 2009.

[11] C. Cohen-Tannoudji, B. Diu, and F. Laloë. *Mécanique Quantique - Tome 1: Nouvelle édition*. Les Ulis: EDP Sciences, 2018.

[12] T. A. Driscoll, N. Hale, and L. N. Trefethen. Chebfun guide. *Pafnuty Publications*, Oxford, 2014.

[13] E. Gallopoulos and Y. Saad. On the parallel solution of parabolic equations. In *Proceedings of the 3rd International Conference on Supercomputing*, ICS '89, page 17–28, New York, NY, USA, 1989. Association for Computing Machinery.

[14] P.-L. Giscard and C. Bonhomme. Dynamics of quantum systems driven by time-varying Hamiltonians: Solution for the Bloch-Siegert Hamiltonian and applications to NMR. *Physical Review Research*, 2:023081, Apr 2020.

[15] P.-L. Giscard, K. Lui, S. J. Thwaite, and D. Jaksch. An exact formulation of the Time-Ordered Exponential using Path-Sums. *Journal of Mathematical Physics*, 56(5):053503, 2015.

[16] D. Goldman and Kaper. T.J. Nth-order operator splitting schemes and nonreversible systems. *SIAM Journal on Numerical Analysis*, 33(1):349–367, 1996.

[17] G.H. Golub and G. Meurant. *Matrices, moments and quadrature with applications*. Princeton University Press, 2010.

[18] D. Gottlieb and C.-W. Shu. On the Gibbs phenomenon and its resolution. *SIAM Review*, 39(4):644–668, 1997.

[19] N. Higham. The scaling and squaring method for the matrix exponential revisited. *SIAM Journal on Matrix Analysis and Applications*, 26(4):1179–1193, 2005.

[20] N. Higham. *Functions of matrices*. SIAM, 2008.

[21] T. Kobayashi, Z. Wang, and M. Pruski. Homonuclear dipolar recoupling of arbitrary pairs in multi-spin systems under magic angle spinning: A double-frequency-selective ZQ-SEASHORE experiment. *Solid State Nuclear Magnetic Resonance*, 101:76–81, 2019.

[22] Yu. N. Kosovtsov. The introduction to the operator method for solving differential equations. First-order DE. arXiv : math-ph/0202040 [math-ph], 2002.

[23] S.H. Kulkarni, R. Radha, and K. Sarvesh. Solution of an infinite band matrix equation. *Banach Journal of Mathematical Analysis*, 17:Paper No. 14, 2023.

[24] A. Malaer, S. Penzel, G.-M. Camenisch, T. Wiegand, A. Samoson, A. Böckmann, M. Ernst, and B. Meier. Quantifying proton NMR coherent linewidth in proteins under fast MAS conditions: A second moment approach. *Physical Chemistry Chemical Physics*, 21, 08 2019.

[25] P. P. Man. Wigner active and passive rotation matrices applied to NMR tensor. *Concepts in Magnetic Resonance Part A*, 45A(1):e21385, 2016.

[26] C. Moler and C. Van Loan. Nineteen dubious ways to compute the exponential of a matrix, Twenty-five years later. *SIAM Review*, 45(1):3–49, 2003.

[27] R. E.E. Omer, E. B.M. Bashier, and A. I. Arbab. Numerical solutions of a system of ODEs based on Lie-Trotter and Strang operator-splitting methods. *Universal Journal of Computational Mathematics*, 5(2):20–24, April 2017.

[28] S. Pozza and N. Van Buggenhout. A new Legendre polynomial-based approach for non-autonomous linear ODEs. arXiv:2303.11284 [math.NA], 2023.

[29] S. Pozza and N. Van Buggenhout. A new matrix equation expression for the solution of non-autonomous linear systems of ODEs. *PAMM*, 22(1):e202200117, 2023.

[30] B. Reif, S.E. Ashbrook, L. Emsley, and M. Hong. Solid-state NMR spectroscopy. *Nature Reviews Methods Primers*, 2021.

[31] Y. Saad. Analysis of some Krylov subspace approximations to the matrix exponential operator. *SIAM Journal on Numerical Analysis*, 29(1):209–228, 1992.

[32] L. N. Trefethen and D. Bau. *Numerical Linear Algebra*. SIAM, 1997.

[33] H. Wang and S. Xiang. On the convergence rates of Legendre approximation. *Mathematics of computation*, 81(278):861–877, 2012.

List of Figures

List of Tables

A. Appendix

We report the first 13 protons Cartesian coordinates of cationic tin oxo-cluster[1].

Table A.1: First 13 protons of cationic tin oxo-cluster.

Label	Type symbol	x	y	z
H231	H	0.97729998827	0.960699975491	0.879599988461
H241	H	0.045400001109	0.0	0.879599988461
H341	H	0.97729998827	0.039299998432	0.879599988461
H561	H	0.022700000554	0.960699975491	0.120399996638
H151	H	0.95459997654	0.0	0.120399996638
H161	H	0.022700000554	0.039299998432	0.120399996638
H21	H	0.04300000146	0.925499975681	0.87349998951
H31	H	0.913999974728	0.0	0.87349998951
H41	H	0.04300000146	0.074500001967	0.87349998951
H51	H	0.957000017166	0.925499975681	0.126499995589
H61	H	0.086000002921	0.0	0.126499995589
H11	H	0.957000017166	0.074500001967	0.126499995589
H52	H	0.927600026131	0.918600022793	0.094400003552

[1]The data (by Dr.F.Ribot) are taken form the website of P.-L. Giscard: http://www-lmpa.univ-littoral.fr/ plgiscard/CationicOxoCluster.txt