NUMERICAL SOLUTION OF HIGHLY OSCILLATORY DIFFERENTIAL EQUATIONS BY MAGNUS SERIES METHOD

A Thesis Submitted to
the Graduate School of Engineering and Sciences of
İzmir Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE

in Mathematics

by
Bengi KANAT

November 2006
İZMİR

We approve the thesis of **Bengi KANAT**

Date of Signature

.... **30 November 2006**

Assist. Prof. Gamze TANOĞLU
Supervisor
Department of Mathematics
İzmir Institute of Technology

.... **30 November 2006**

Prof. Dr. Gökmen TAYFUR
Department of Electronic Engineering
İzmir Institute of Technology

.... **30 November 2006**

Assoc. Prof. Ali İhsan NESLİTÜRK
Department of Mathematics
İzmir Institute of Technology

.... **30 November 2006**

Prof. Dr. İ. Hakkı DURU
Head of Department
Department of Mathematics
İzmir Institute of Technology

....

Assoc. Prof. M.Barış ÖZERDEM
Head of the Graduate School

ACKNOWLEDGEMENTS

Firstly I would like to thank my advisor Asst. Prof. Dr. Gamze Tanoğlu for her guidance and encouragement. This project is supported by TUBITAK/ SLOVENIA (TBAG-U/104) and BAP 2005 IYTE12. I would like to express my thanks to Bojan Orel from University of Ljubljana, Slovenia, for his various suggestions and to the people from the Workshop NA06 (Numerical Analysis Day Workshop, Ljubljana, Slovenia); Arieh Iserles and Syvert P. Norsett and et al for their ideas for the future works.

Many thanks to my mother, my brother and my husband for their support in every area of my life. This thesis is dedicated to my dear father,...

ABSTRACT

NUMERICAL SOLUTION OF HIGHLY OSCILLATORY DIFFERENTIAL EQUATIONS BY MAGNUS SERIES METHOD

In this study, the differential equation known as Lie-type equation where the solutions of the equation stay in the Lie-Group is considered. The solution of this equation can be represented as an infinite series whose terms consist of integrals and commutators, based on the Magnus Series. This expansion is used as a numerical geometrical integrator called Magnus Series Method, to solve this type of equations. This method which is also one of the Lie-Group methods, has slower error accumulation and more efficient computation results during the long time interval than classical numerical methods such as Runge-Kutta, since it preserves the qualitative features of the exact solutions. Several examples are considered including linear and nonlinear oscillatory problems to illustrate the efficiency of the method.

ÖZET

MAGNUS SERİ METODU KULLANARAK YÜKSEK SALINIMLI DİFERANSİYEL DENKLEMLERİN SAYISAL OLARAK ÇÖZÜMÜ

Bu çalışmada Lie-tipi denklemler olarak bilinen, çözümleri Lie-Grup'ta bulunan diferansiyel denklemler irdelenmiştir. Bu denklemin çözümü Magnus Serisine dayanan ve terimleri integraller ve komütatörler olan sonsuz bir seri olarak ifade edilebilir. Bir çeşit sayısal geometrik integratör olan Magnus Seri Metodu Lie-tipi denklemleri sayısal olarak çözer. Aynı zamanda Lie-grup metodlardan biri olan bu metod az hata ve Runge-Kutta gibi klasik nümerik metodlardan daha etkili hesaplama sonuçlarına sahiptir çünkü tam çözümlerin niteliksel özelliklerini korur. Lineer ve lineer olmayan salınım problemlerini içeren birkaç örneğe metod, etkiyi vurgulamak için uygulanmıştır.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Most of the physical systems, from very large (galaxies) to the very small (molecules), to the very complicated (the weather), deal with motion. It is possible to write down the equations of the motion. These equations are in the form of ordinary or partial differential equations, but it might not be possible to solve all of them exactly. Some approaches which are improved by the numerical analysts are used to solve them numerically ('integrated') on computers (WEB-6, 2005). Up to today's intelligence, quantitative character of the problems are used but today qualitative feature is discovered: Geometric Integration.

Geometric integration is the numerical integration of a differential equation while preserving one or more "geometric properties" of the exact flow. Many of these features are important in physical applications such as preservation of energy, momentum, angular momentum, phase space volume, symmetries, time-reversal symmetry, symplectic structure, etc. In recent decades, these qualitative features have been found to be important in every branch of numerical analysis because of i) appropriateness for methods which are faster, simpler and more stable, ii) appropriateness for methods which can give qualitatively better results than standard methods, iii) possibility to calculate some of ODEs; example in the long-time integration of Hamiltonian systems.

Most geometric properties are not preserved by classical numerical methods (McLachlan et al. 2006). In geometric integration the geometric properties are built into the numerical method, which gives the method markedly superior performance, especially during long-time simulations. By sharing geometric structure and invariants with the exact flow, the method is more reliable, faster, accurate, simpler and cheaper than classical approaches and it is used in the areas such as the structure of liquids, biomolecules, quantum mechanics, nanodevices , etc. (WEB-6, 2005).

Geometric integrators include symplectic and multisymplectic integrators that preserve the Hamiltonian or Poisson structure; variational integrators that utilize the variational character of Lagrangian and canonical Hamiltonian systems, conservative integrators that preserve first integrals or conservation laws and symmetric integrators that preserve symmetries of the system (WEB-5, 2006).

This new approach either connects pure mathematics with its computation or produces closer solutions since the solution evolves in the same geometrical structure. The main types of methods used in geometric integration of ODEs are splitting and composition methods, projection methods, Runge-Kutta's variants and Lie Group Methods (McLachlan et al. 2006).

The integrations of ODEs in Lie-groups started with Sophus Lie (Ibragimov 1999). As he said in 1895, he claimed that he was the first who used the concept of groups for an integration theory of differential equations (Ibragimov 1999). The linear object he obtained was originally called the "infinitesimal group" by himself and was later renamed as "Lie algebra" by H.Weyl (Hsiang 2000). The first numerical method to integrate ODEs on Lie-groups appeared in 1993 by Crouch and Grossman (Zanna 1997). Briefly, Iserles, Norsett and their "numerical analysis group" rendered numerically efficient numerical approaches. They improved several types of Lie-group methods such as: Runge-Kutta Munthe-Kaas, Magnus Series Method, etc. They applied these methods to some typical equations like Airy (Iserles 2002), Mathieu (Iserles et al. 1999) and Bessel-like, proved the convergency of the Magnus Series and analysed the global error of Magnus Series Method. Still today, the studies on methods are being continued.

In this thesis, Magnus Series that was developed by Wilhelm Magnus in 1954, is applied to the first order system of ordinary differential equation. It is used as one of the Lie-group solvers and solve the following differential system:

$$y' = A(t, y)y, \quad t \geq 0, \quad y(0) = y_0 \tag{1.1}$$

which is called as Lie-type equation. The numerical solution obtained by this method is compared with the classical numerical Runge-Kutta Method.

Outline of the thesis is as follows: In chapter 2 we start by reviewing some main ideas of interest us regarding Lie groups, Lie algebras and their relationship. In Chapter 3 and 4, the motivation and implementation of Magnus Series Method for linear and nonlinear differential equations are given. In chapter 5 and 6, the applications of Magnus Series Method to the different linear and nonlinear ordinary differential equations are presented and some results obtained from this method are compared with the classical Runge-Kutta Method. Finally, discussion and future works are presented in the last chapter.

CHAPTER 2

THEORY AND BACKGROUND

Consider the first order system of ordinary differential equation,

$$y' = f(t, y), \qquad t \geq 0, \qquad y(0) = y_0, \qquad y(t) \in \mathbb{R}^{\mathbb{N}} \tag{2.1}$$

where f is a vector field on $\mathbb{R}^{+} \times \mathbb{R}^{\mathbb{N}}$, can be solved with the given initial conditions by using one of the well-known numerical integrators such as Runge-Kutta and multistep methods; these are known as classical integrators. Lie-group methods give possibility of replacing the domain $\mathbb{R}^{\mathbb{N}}$ by more general configuration spaces, this is the main motivation and the advantage of the manifold rather than the entire $\mathbb{R}^{\mathbb{N}}$ is an insensible configuration space which has crucial geometric features of the underlying differential system for instance conservation laws, symmetries or symplectic structure and in numerical meaning, as we will see later, slower error accumulation (WEB-3, 2006).

There are several types of Lie-group methods such as Croach-Grossman Method, Runge-Kutta-Munthe-Kaas Method and Magnus Series Method. In this study the oscillatory differential equations are solved by Magnus Series Method. In the following sections oscillatory equations and Magnus Series Method are given theoretically with the basic definitions in order to make connections between Lie-group structure and Magnus Series.

2.1. Oscillatory Equations

The initial-value ordinary differential system considered in Eq.(2.1) exhibits oscillatory solutions with a timescale much shorter than the integration interval. This kind of dynamical systems is referred as highly oscillatory ones. In this section, we will give a brief review about this concept.

Consider the equation:

$$y'' + a(t)y = 0 \tag{2.2}$$

where $a(t)$ is a real-valued and continuous function on $t_0 \leq t < \infty$ (WEB-2, 2006). If all the nontrivial solutions of (2.1) have an infinite number of zeros on $t_0 \leq t < \infty$, then

(2.1) is referred to as an oscillatory equation and these nontrivial solutions are termed oscillatory solutions. For any equation, if some solutions are oscillatory and remaining are nonoscillatory, the the equation is an nonoscillatory equation. Consider the equation $u'' + m^2 u = 0$ where m is a real constant, then this equation is an oscillatory and all the solutions which are in the form of $\cos(mt)$, $\sin(mt)$ are oscillatory solutions. However if we consider the other equation $u''' - u'' + u' - u = 0$, this is an nonoscillatory equation since one of its nontrivial solutions, namely e^t, is not oscillatory.

The following result help us to get some oscillatory properties of the solutions of second order linear differential equations.

Theorem 2.1 *If all the nontrivial solutions of (2.2) are oscillatory, $b(t)$ is continuous, and $b(t) \geq a(t)$, $t_0 \leq t < \infty$, then all the nontrivial solutions of*

$$u'' + b(t)u = 0 \tag{2.3}$$

are oscillatory.

Proof: Let $y(t)$ and $u(t)$ be the nontrivial solutions of (2.2) and (2.3) respectively. Multiplying (2.3) by y, (2.2) by u, and subtracting, we get

$$yu'' - uy'' + (b(t) - a(t))yu = 0, \tag{2.4}$$

that is,

$$d(yu' - uy') + (b(t) - a(t))yu = 0 \tag{2.5}$$

Let t_1 and t_2 be any two zeros of $y(t)$ and assume that $t_0 \leq t_1 < t_2$ and that $y(t) \geq 0$ on the interval $t_1 \leq t \leq t_2$. By integrating (2.5) from t_1 to t_2, we obtain

$$y(t_2)u'(t_2) - u(t_2)y'(t_2) - y(t_1)u'(t_1) + u(t_1)y'(t_1) \tag{2.6}$$

$$+ \int_{t_1}^{t_2} [b(s) - a(s)]y(s)u(s)ds = 0 \tag{2.7}$$

Since t_1 and t_2 are two zeros of $y(t)$ and $y'(t_1) > 0$, $y'(t_2) < 0$, we obtain:

$$u(t_1)y'(t_1) - u(t_2)y'(t_2) + \int_{t_1}^{t_2} [b(s) - a(s)]y(s)u(s)ds = 0 \tag{2.8}$$

We claim that $u(t)$ has a zero on $[t_1, t_2]$, since $y'(t_1) > 0$, $y'(t_2) < 0$ and $y(t)$, $b(t) - a(t)$ are nonnegative on $t_1 \leq t \leq t_2$. Hence $u(t)$ changes sign in the interval $[t_1, t_2]$. This shows that between any two zeros of $y(t)$ there is a zero of $u(t)$.

Corollary 2.2 *The nontrivial solutions of*

$$y'' + (1 + \phi(t))y = 0 \tag{2.9}$$

where $\phi(t) \to 0$ *as* $t \to \infty$ *are oscillatory.*

Proof: Since $\lim_{t\to\infty} \phi(t) = 0$, for sufficiently large t_0 we have

$$|\phi(t)| \leq \epsilon \qquad fort \geq t_0$$

that is, $-\epsilon \leq \phi(t) \leq \epsilon$. Therefore, $1 - \epsilon \leq 1 + \phi(t) \leq 1 + \epsilon$. Choose $\epsilon = \frac{1}{2}$, then we have $1 + \phi(t) \geq \frac{1}{2}$ for $t \geq t_0$. Since all the nontrivial solutions of $y'' + \frac{1}{2}y = 0$ are oscillatory.

Corollary 2.3 *The nontrivial solutions of*

$$y'' + \phi(t)y = 0 \tag{2.10}$$

are oscillatory if $\phi(t) \geq m^2 > 0$ *for all t.*

Proof: Since all the nontrivial solutions of $y'' + m^2 y = 0$ are oscillatory the result follows from Theorem (2.1)

Corollary 2.4 *if* $\lim_{t\to\infty} a(t) = \infty$ *monotonically, then all the nontrivial solutions of (2.2) are oscillatory*

Proof: From the hypothesis, it is clear that $a(t) > \epsilon > 0$ for all t greater than some t_0. Since all the nontrivial solutions of $y'' + \in y = 0$, $\in > 0$, are oscillatory, the result follows from Theorem (2.1).

Briefly we can introduce the linear oscillator (2.1) where $a(t) > 0$, $\lim_{t\to\infty} a(t) = \infty$ and $a'(t) = o(a(t))$, $t \gg 1$.

2.2. Lie-Group Methods

The most known numerical algorithms in literature are formulated in vector spaces. The new approach developed lately proposes that the solution of the differential equation evolves on the differential manifold. In order to clarify the concept, the following useful definitions and theorems are stated in this section.

Definition 2.1 *A* m*-dimensional manifold* M *is a smooth surface* $M \subset \mathbb{R}^N$ *for some* $N \geq m$.

As we know from the advanced calculus that the function $f : \mathbb{R}^n \to \mathbb{R}^m$ which is differentiable is smooth if f is differentiable infinitely and denoted by $f \in C^\infty$.

In algebraic meaning, (Olver 1993) M is a set which has countable collection of subsets $u_\alpha \subset M$ (called coordinate charts) and one-to-one functions $x_\alpha : u_\alpha \to v_\alpha$ onto connected open subsets $v_\alpha \subset \mathbb{R}^m$ (called local coordinate maps) and this set satisfies following properties;

i) The coordinate charts cover M

$$\cup_\alpha u_\alpha = M,$$

ii) On the overlap of any pair of coordinate charts $u_\alpha \cap u_\beta$, the composite map

$$x_\beta \circ x_\alpha^{-1} : x_\alpha(u_\alpha \cap u_\beta) \to x_\beta(u_\alpha \cap u_\beta)$$

is a smooth function,

iii) If $x \in u_\alpha$, $\tilde{x} \in u_\beta$ are distinct points of M, then there exist open subsets $\omega \subset v_\alpha$, $\tilde{\omega} \subset v_\beta$ with $x_\alpha(x) \in \omega$, $x_\beta(\tilde{x}) \in \tilde{\omega}$ satisfying

$$x_\alpha^{-1}(\omega) \cap x_\beta^{-1}(\tilde{\omega}) = \varnothing.$$

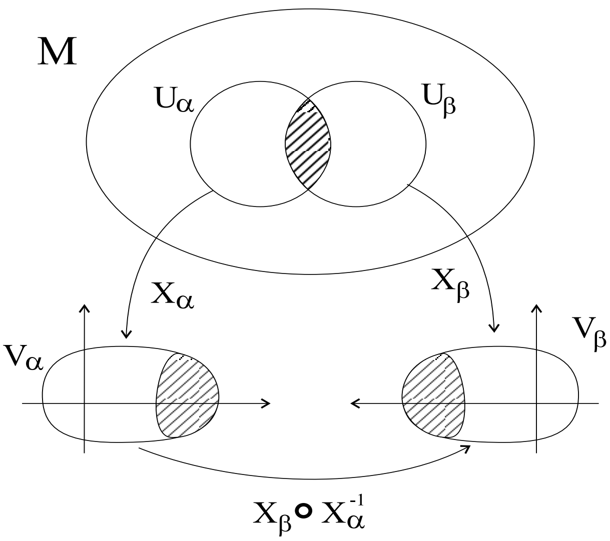

Figure 2.1: Manifold

(Source: Olver 1993)

The Fig.(2.1) illustrates the above properties. Note that $u \subset M$ is an open set if and only if for each $x \in u$ there is a neighbourhood of x contained in u.

Examples are as follows:

1. $\mathbb{R}^n$, n dimensional Euclidean space,

2. An easier example is the unit circle:

$$S^2 = \{(x, y, z) : x^2 + y^2 = 1\}$$

which is one-dimensional manifold,

3. $\mathbb{T}^n$, the n-dimensioanal torus. Fig.(2.2) shows the 2-dimensional torus,

$$\mathbb{T}^n = S^1 \times S^1 \times ...... \times S^1 \qquad (n - times)$$

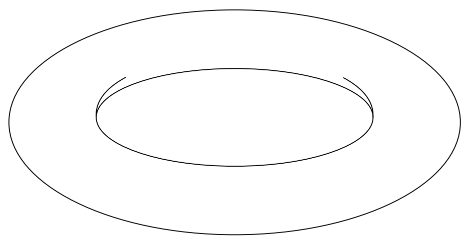

Figure 2.2: Torus

(Source: James 1987)

4. $gl(n, \mathbb{R})$, real nxn matrices. This is an n^2-dimensional manifold homeomorphic to $\mathbb{R}^{n^2}$,

5. $Gl(n, \mathbb{R})$, general linear group, real nxn matrices with non-zero determinant. It is a manifold as an open subset of $\mathbb{R}^{n^2} \approx Gl(n, \mathbb{R})$,

6. $SO(n, \mathbb{R})$, special orthogonal group, real nxn orthogonal matrices ($AA^T = I$) with det=1.

Most manifolds of practical relevance are homogeneous spaces: For the concept of 'homogeneous space', we need related definitions as follows

Definition 2.2 *Suppose that, given a smooth manifold M, there exists a Lie-Group $\mathcal{G}$ and a map $\lambda : \mathcal{G} \times M \to M$ such that $\lambda(X_1, \lambda(X_2, y)) = \lambda(X_1 X_2, y)$ for every $X_1, X_2 \in \mathcal{G}$ and $\lambda(1, y) = y$ for all $y \in M$. In that case, it is said that $\mathcal{G}$ acts on M and that λ is a group action*

Definition 2.3 *A group action is transitive if for every y_1, $y_2 \in M$ there exists $X \in \mathcal{G}$ such that $\lambda(X, y_1) = y_2$. A manifold that admits a transitive group action is called a homogeneous space.*

Many important conservation laws can be described as an evolution restricted to a homogeneous space. The simplest example is when M is itself a Lie-Group, whereby $\mathcal{G} = M$ acts upon itself and λ is the usual (left or right) multiplication. The n-sphere and the n-torus are also homogeneous spaces with respect to the action of $O(n; \mathbb{R})$. Stiefel manifold (the set of all $n \times r$ real matrices X, $r \leq n$ such that $X^T X = I$), the Grassmann manifold (the set of all r-dimensional linear subspaces of $\mathbb{R}^n$, alternatively all equivalence classes in a Stiefel manifold with respect to a product by a member of $O(n; \mathbb{R})$), the projective space of all lines through the origin in $\mathbb{R}$ or $\mathbb{C}$, and the set of all elements of $gl(n; \mathbb{R})$ similar to a given matrix.

The most important property of a manifold is the existence of tangents to the manifold at any point $p \in M$.

Definition 2.4 *Let M be a n-dimensional manifold and $\alpha(t)$ be a smooth curve on M, ($\alpha(0) = p$)*

$$a = \frac{d\alpha(t)}{dt} \mid_{t=0}$$

is the tangent vector at p.

Definition 2.5 *Some curves on M can have the same tangent (velocity) vectors at a point, if we define a set $C_p(M) = \{\alpha : \mathcal{I} \rightarrow M : \alpha \text{ smooth and } \alpha(0) = p\}$ (WEB-4, 2005). Curves in $C_p(M)$ are equivalent if their derivatives at p agree.*

Consider a tangent vector to be an equivalence class $[\alpha]_p$. The set of all tangents (equivalence classes) $C_p(M)/[\ ]_p$) is the tangent space at p to M and denoted by TM_p.

Fig.(2.3) shows the equivalent and not equivalent class on the manifold M.

Definition 2.6 *If all the tangent spaces at all points $p \in M$ put together, then it forms tangent bundle which is also another manifold and denoted by*

$$TM = \cup_{p \in M} TM_p$$

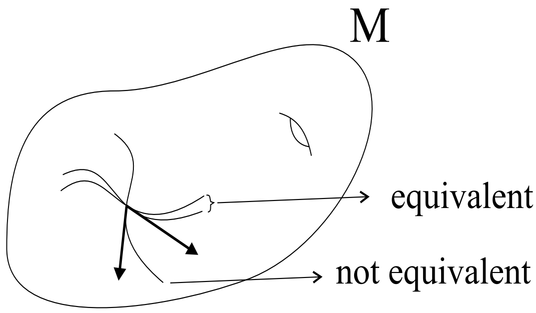

Figure 2.3: Equivalence Class

(Source: James 1987)

Definition 2.7 *A (tangent) vector field on M is a smooth function $F : M \to TM$ such that $F \in TM \mid_p$ for all $p \in M$. The collection of all vector fields on M is denoted by $\Xi(M)$.*

Definition 2.8 *Let F be a tangent vector field on M and suppose the differential equation on M*

$$y' = F(t, y), \quad t \geq 0, \quad y(0) \in M \tag{2.11}$$

The flow of F is the solution operator $\Psi_{t,F} : M \to M$ such that

$$y(t) = \Psi_{t,F}(y_0)$$

solves the Eq.(2.11).

The general form of differential equation (2.11) is the main type of differential equations, Lie-type, with small differences such as

Definition 2.9 *Every differential equation considered in (2.11) can be written in following the form:*

$$y' = A(t, y)y, \qquad , t \geq 0, \qquad y(0) = y_0 \in \mathcal{G} \tag{2.12}$$

if it evolves in a Lie Group. This equation is called Lie type equation.

Here $\mathcal{G}$ is the Lie-group which can be thought as the special case of manifold, $A(t, y) : \mathbb{R}^+ \to$ g, g is the Lie-algebra to the corresponding Lie-group which can be thought as the tangent space of the Lie-group $\mathcal{G}$. It is time to give the basic two definitions of this study; Lie-group and Lie-algebra (Iserles et al. 2000).

Definition 2.10 *The differential manifold $\mathcal{G}$ is a Lie Group if it has a group structure with respect to a binary operation, usually multiplication, $\mathcal{G}x\mathcal{G} \to \mathcal{G}$:*

$p \cdot (q \cdot r) = (p \cdot q) \cdot r$ *for all $p, q, r \in \mathcal{G}$* *(associativity)*

$\exists I \in \mathcal{G}$ *such that* $I \cdot p = p \cdot I = p \quad \forall p \in \mathcal{G}$ *(identity element)*

$\forall p \in \mathcal{G} \quad \exists p^{-1} \in \mathcal{G}$ *such that* $p^{-1} \cdot p = I$ *(inverse)*

the maps $(p, r) \mapsto p \cdot r$ and $p \mapsto p^{-1}$ *(smoothness)*

are smooth functions

so that multiplication and inverse are continuous in the topology of the manifold.

Note that a group is an ordered pair $\langle G, \circ \rangle$, where G is a set ($\circ$ is a binary operation) satisfying the following conditions;

i) closure, $x \in G$ and $y \in G \Rightarrow x \circ y \in G$,

ii) $\circ$ is associative, $x \circ (y \circ z) = (x \circ y) \circ z$,

iii) $e \in G$ is an identity (two-sided) for the operation $\circ$,

iv) for every element $x \in G$, there is an element y that is an inverse of x, with respect to $\circ$ and identity e,

G is an abelian group if $\circ$ is commutative. Important examples of Lie-groups are as follows:

1. The general linear group $GL(n; \mathbb{F})$ of all nonsingular nxn matrices over the field $\mathbb{F}$ (typically $\mathbb{F} = \mathbb{R}$ or $\mathbb{F} = \mathbb{C}$),
2. The special linear group $SL(n; \mathbb{F})$ of all $X \in GL(n; \mathbb{F})$ such that $detX = 1$,
3. The orthogonal group $O(n; \mathbb{F})$ of all $X \in GL(n; \mathbb{F})$ such that $XX^T = 1$,
4. The special orthogonal group $SO(n; \mathbb{F}) = O(n; \mathbb{F}) \cap SL(n; \mathbb{F})$,
5. The symplectic group $Sp(n)$ of all matrices $X \in GL(2n; \mathbb{R})$ such that $XJX^T = J$, where $J = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix}$,
6. $\mathbb{R}^n$ is an abelian Lie group under vector addition, with Lie algebra just $\mathbb{R}^n$,
7. S^1 is an abelian Lie group. The group structure is defined by considering S^1 as the quotient group,
8. $\mathbb{T}^n$ is an abelian Lie group with group structure induced from S^1. Lie algebra is $\mathbb{R}^n$.

Lie group seems to be a marriage between algebraic concept of a group and differential-geometric notion of a manifold. This combination of algebra and calculus create a powerful techniques, as we will see soon.

Definition 2.11 *A Lie algebra is a linear space V equipped with a Lie bracket, a bilinear, skew-symmetric mapping:*

$$[\cdot,\cdot] : V \times V \rightarrow V$$

This mapping, usually called commutator or Lie bracket, obeys the following identities:

1. $[a, b] = -[b, a]$ (skew-symmetry)
2. $[\alpha a, b] = \alpha[a, b]$ for $\alpha \in \mathcal{R}$
3. $[a + b, c] = [a, c] + [b, c]$ (bilinearity)
4. $[a, [b, c]] + [b, [c, a]] + [c, [a, b]] = 0$ (Jacobi's identity)

Definition 2.12 *A Lie group homomorphism is a smooth map $\phi : G \rightarrow H$ between two groups which respects the group operations*

$$\phi(g.h) = \phi(g).\phi(h) \qquad g, h \in G \tag{2.13}$$

If ϕ has a smooth inverse, then it is also an isomorphism between G and H

An important feature of a Lie-group is that there exists a natural map taking g to $\mathcal{G}$ and this map is usually called exponential map (WEB-4, 2005).

Definition 2.13 *Let $\mathcal{G}$ be a Lie group and* g *its Lie algebra. The exponential mapping exp :* g *$\rightarrow \mathcal{G}$ is defined as $exp(a) = \rho(1)$ where $\rho(t) \in \mathcal{G}$ satisfies the differential equation $\rho'(t) = a\rho(t)$, $\rho(0) = I$.*

In the Fig.(2.4), the mission of the exponential map is shown and here ϕ is the derivative map.

$$\begin{tikzcd}
\mathrm{H} \arrow[r, "\phi"] & \mathrm{G} \\
\mathrm{h} \arrow[u, "\exp"] \arrow[r, "d\phi"'] & \mathrm{g} \arrow[u, "\exp"']
\end{tikzcd}$$

Figure 2.4: Exponential Mapping

(Source: James 1987)

Properties of the exponential map:

1. $exp(ta) = \rho(t), \qquad \forall t \in \mathbb{R}$,

2. $exp(t_1 + t_2)a = exp(t_1 a) + exp(t_2 a), \qquad \forall t_1, t_2 \in \mathbb{R}$,

3. $exp(-ta) = (exp(ta))^{-1}, \qquad \forall t \in \mathbb{R}$,

4. $exp : \mathbf{g} \to \mathcal{G}$ is smooth, and $exp_* : T_0(\mathbf{g}) \to T_e(\mathcal{G})$ is the identity map. Hence by the inverse function theorem, exp gives a diffeomorphism of a neighbourhood of 0 in g onto a neighbourhood e of $\mathcal{G}$,

5. If $[a, b] = 0$ then $exp(a + b) = exp(a)exp(b)$,

Next, we will give the definition of the adjoint representation.

Definition 2.14 *Let $p \in \mathcal{G}$ and let $\rho(t)$ be a smooth curve on $\mathcal{G}$ such that $\rho(0) = I$ and $\rho'(0) = b \in \mathbf{g}$. The adjoint representation is defined as:*

$$Ad_p(b) = \frac{d}{dt} p\rho(t)p^{-1} \mid_{t=0} \tag{2.14}$$

The derivative of Ad with respect to the first argument is denoted as ad. Let $\rho(s)$ be a smooth curve on $\mathcal{G}$ such that $\rho(0) = I$ and $\rho'(0) = a$;

$$ad_a(b) \equiv \frac{d}{ds} Ad_{\rho(s)}(b) \mid_{s=0} = [a, b] \tag{2.15}$$

The following properties show that Ad is a linear group action and that for a fixed argument p it is a Lie-algebra isomorphism of g onto itself:

$$Ad_p(a) \in \mathbf{g}, \quad for\ all\ p \in \mathcal{G}, a \in \mathbf{g} \tag{2.16}$$

$$Ad_p \circ Ad_q = Ad_{pq}, \tag{2.17}$$

$$Ad_p(a + b) = Ad_p(a) + Ad_p(b), \tag{2.18}$$

$$Ad_p([a, b]) = [Ad_p(a), Ad_p(b)]. \tag{2.19}$$

Both Ad_p and ad_a may be regarded as matrices acting on the linear space g since they are linear in their second argument. This gives meaning to the following important formula which we will use in the next sections:

$$Ad_{exp(a)} = expm(ad_a)$$

CHAPTER 3

MAGNUS SERIES METHOD FOR LINEAR DIFFERENTIAL EQUATIONS

In this section one of the Lie-Group methods, Magnus series Method is introduced to solve the highly oscillatory differential equations. The main goal is to show the transitions between Lie-group structure and Magnus Series and then the formation of the Magnus Series Method.

3.1. Motivation

The linear differential equation on a matrix Lie-group is an equation of the form

$$Y' = A(t)Y, \qquad t \geq 0, \quad Y(0) = Y_0 \in \mathcal{G} \tag{3.1}$$

where $A(t) : \mathbb{R} \to \mathbf{g}$ is the matrix function, $\mathcal{G}$ is the Lie-group, $\mathbf{g}$ is the Lie algebra of the corresponding Lie-group $\mathcal{G}$ and AY is the usual matrix product between $A \in \mathbf{g}$ and $Y \in \mathcal{G}$. These equations are called Lie-type equations or linear type Lie-group equations.

As a motivation first consider (3.1) as a scalar differential equation (for the common notation lower case $a(t)$ is used for scalar functions and upper case $A(t)$ for the matrix functions) such that:

$$y' = a(t)y, \qquad t \geq 0, \quad y(0) = y_0 \tag{3.2}$$

then the solution is:

$$y(t) = exp\Big(\int_0^t a(s)ds\Big)y(0)$$

as we know from the basic ODEs courses. However in the matrix form we can not write the solution as it is in the scalar form since $A(t_1)$ and $A(t_2)$ does not commute with each other for all $t_1, t_2 \geq 0$.

The observation will be for an interval of time $[t_0, t_1] \cup [t_1, t_2]$. Suppose that the solution of the Eq.(3.1) is the same as in the scalar form, then the solution is

$$Y(t) = exp\Big(\int_{t_0}^{t} A(s)ds\Big)Y(t_0), \tag{3.3}$$

We can write it for t_2 and t_1

$$Y(t_2) = exp\Big(\int_{t_1}^{t_2} A(s)ds\Big)Y(t_1), \tag{3.4}$$

$$Y(t_1) = exp\Big(\int_{t_0}^{t_1})A(s)ds\Big)Y(t_0), \tag{3.5}$$

and substitute it in the (3.4), we have

$$Y(t_2) = exp\Big(\int_{t_1}^{t_2} A(s)ds\Big)exp\Big(\int_{t_0}^{t_1} A(s)ds\Big)Y(t_0). \tag{3.6}$$

On the other hand, we can divide the interval into two parts in the eq.(3.3):

$$Y(t_2) = exp\Big(\int_{t_0}^{t_2} A(s)ds\Big)Y(t_0) = exp\Big(\int_{t_0}^{t_1} A(s)ds\Big)exp\Big(\int_{t_1}^{t_2} A(s)ds\Big)Y(t_0) \tag{3.7}$$

If we call $exp(\int_{t_0}^{t_1} A(s)ds) = B$ and $exp(\int_{t_1}^{t_2} A(s)ds) = C$, in the (3.7) $Y(t_2) = BCY(t_0)$, but in the (3.6) $Y(t_2) = CBY(t_0)$. As we all know in the matrix form $BC \neq CB$. We can not use the solution for the scalar linear differential equations in the matrix linear systems.

3.2. Magnus series Method

In the context of the Lie groups, the tangent space is the Lie algebra g and required map is the matrix exponential, thus exponential map carries the elements of algebra to the corresponding group. Since, in the previous section, the solution of the Eq.(3.1) in general case can not be written as a single matrix of a single integral unless the group is not commutative.

To introduce the solution of Eq.(3.1) we need to define the exponential map for the matrix form.

Definition 3.1 *The exponential mapping $expm : \mathrm{g} \to \mathcal{G}$ is defined as*

$$expmA = \sum_{j=0}^{\infty} \frac{A^j}{j!} \tag{3.8}$$

$expm(O) = I$, and a is sufficiently near $O \in \mathrm{g}$ since exponential has a smooth inverse given by the matrix logarithm $logm : \mathcal{G} \to \mathrm{g}$.

After this definition we can give the solution of Eq.(3.1) as

$$Y = e^{\delta(t)} Y_0 \tag{3.9}$$

In order to show this representation is the solution of Eq.(3.1), we need to take the derivative of Eq.(3.9), so we need to define the derivative of the exponential map.

Definition 3.2 *The differential of the exponential mapping is defined as a function* $dexp$: $\mathbf{g} \times \mathbf{g} \rightarrow \mathbf{g}$ *such that*

$$\frac{d}{dt} exp(A((t)) = dexp_{A(t)}(A'(t)) exp(A(t)). \tag{3.10}$$

Since $dexp_A$ is linear in its second argument for a fixed A and $dexp_A$ is an analytic function of the matrix transformation ad_A:

$$dexp_A = \frac{expm(ad_A) - I}{ad_A} \tag{3.11}$$

where ad is the derivative of the adjoint representation Ad which is given by the formula

$$\begin{aligned} Ad_P A &= PAP^{-1}, \\ ad_A B &= AB - BA = [A, B] \end{aligned}$$

as it was defined in (2.14).

The formula (3.11) should be read as a power series in the following manner. Since

$$\frac{e^x - 1}{x} = 1 + \frac{1}{2!}x + \frac{1}{3!}x^2 + ... + \frac{1}{(j+1)!}x^j + ... \tag{3.12}$$

we obtain

$$\begin{aligned} dexp_a(c) &= c + \frac{1}{2!}[a, c] + \frac{1}{3!}[a, [a, c]] + \frac{1}{4!}[a, [a, [a, c]]].... \\ &= \sum_{j=0}^{\infty} \frac{1}{(j+1)!} ad_a^j c \end{aligned} \tag{3.13}$$

where ad^j is the adjoint operator defined as

$$\begin{aligned} ad_A^j A &= A, & j = 0 \\ &= [ad_A^{j-1} C, C] & j \geq 1 \end{aligned}$$

The fact that $dexp_A$ is an analytic function in ad_A makes it easy to invert the matrix $dexp_A$ simply by inverting the analytic function,

$$dexp_A^{-1} = \frac{ad_A}{expm(ad_A) - I} \tag{3.14}$$

Recall that:

$$\frac{x}{e^x - 1} = 1 - \frac{1}{2}x + \frac{1}{12}x^2 - \frac{1}{720}x^4 + ... = \sum_{j=0}^{\infty} \frac{B_j}{j!}x^j \tag{3.15}$$

where B_j are $Bernoulli\ numbers$ (Abramowitz and Stegun 1970). Thus

$$dexp_A^{-1}(C) = C - \frac{1}{2}[A, C] + \frac{1}{12}[A, [A, C]] + ... = \sum_{j=0}^{\infty} \frac{B_j}{j!}ad_A^j(c) \tag{3.16}$$

Note that, except for B_1, all odd-indexed Bernoulli numbers vanish.

Now, we can return to our main equations. The goal is only to take derivative of Eq.(3.9) and then to embed it into Eq.(3.1). The derivative of the Eq.(3.9) is

$$Y' = (e^{\delta(t)})'Y_0. \tag{3.17}$$

By using the definition (3.2), we obtain

$$Y' = dexp_{\delta(t)}(\delta'(t))e^{\delta(t)}Y_0 \tag{3.18}$$

After using representation (3.9), the Eq.(3.18) becomes

$$Y' = dexp_{\delta(t)}(\delta'(t))Y. \tag{3.19}$$

After substituting Y' in the Eq.(3.1), we have

$$dexp_{\delta(t)}(\delta'(t))Y = A(t)Y \tag{3.20}$$

or

$$dexp_{\delta(t)}(\delta'(t)) = A(t). \tag{3.21}$$

In order to find $\delta(t)$, we need to use the inverse of $dexp$.

$$dexp^{-1}_{\delta(t)}(dexp_{\delta(t)}(\delta'(t)) = dexp^{-1}_{\delta(t)}(A(t)) \tag{3.22}$$

$$\delta'(t) = dexp^{-1}_{\delta(t)}(A(t)) \tag{3.23}$$

The following lemma proves that the representation (3.23) produces a solution of the Eq.(3.1)

Lemma 3.1 *For small $t \geq 0$ the solution of Eq.(3.1) is given by*

$$Y(t) = expm(\delta(t))Y_0 \tag{3.24}$$

where $\delta \in$ g satisfies the differential equation:

$$\delta'(t) = dexp^{-1}_{\delta(t)}(A(t)); \qquad \delta(0) = O \tag{3.25}$$

Eq.(3.24) was originally stated by Felix Hausdorff (1906), although some attribute it to John Edward Campbell, who might have published it a few years earlier (Iserles et al. 2000).

Next, find the more usefull representation for the right hand side of the Eq.(3.25) by using the definition (3.2)

$$\begin{aligned}\delta'(t) &= dexp^{-1}_{\delta(t)}(A(t)) = A(t) - \frac{1}{2}[\delta(t), A(t)] + \frac{1}{12}[\delta(t), [\delta(t), A(t)]] + ... \\ &= \sum_{j=0}^{\infty} \frac{B_j}{j!} ad^j_{\delta(t)}(A(t))\end{aligned} \tag{3.26}$$

where B_j are $Bernoulli\ numbers$.

Finally, we will use the Picard Iteration to solve the differential equation (3.26).

3.3. Picard Iteration Method

Let $Y(x)$ be the solution (if it exists) to the initial value problem;

$$y' = f(x, y) \qquad y(x_0) = Y_0 \tag{3.27}$$

By integrating

$$Y(x) = Y_0 + \int_{x_0}^{x} f(t, Y(t))dt \tag{3.28}$$

This integral equation is solved, at least in theory, by using the iteration

$$Y_{m+1}(x) = Y_0 + \int_{x_0}^{x} f(t, Y_m(t))dt, \qquad x_0 \leq x \leq b \tag{3.29}$$

for $m \geq 0$, with $Y_0 \equiv Y_0$. This is called Picard Iteration, and under suitable assumptions, the iterates $\{Y_m(x)\}$ can be shown to converge uniformly to $Y(x)$.

After this brief terminology we can now solve the equation

$$\delta'(t) = dexp^{-1}_{\delta(t)} A(t) = \sum_{j=0}^{\infty} \frac{B_j}{j!} ad^j_{\delta(t)} A(t) \tag{3.30}$$

by Picard Iteration

$$\begin{aligned} \delta^{[0]}(t) &\equiv O \\ \delta^{[m+1]}(t) &= \int_0^t dexp^{-1}_{\delta^{[m]}(\xi)} A(\xi) d\xi \\ &= \sum_{j=0}^{\infty} \frac{B_j}{j!} \int_0^t ad^j_{\delta^{[m]}(\xi)} A(\xi) d\xi, \qquad m = 0, 1, ... \end{aligned}$$

Rearranging terms for simplicity, we obtain

$$\delta^{[1]}(t) = \int_0^t A(\xi_1) d\xi_1 \tag{3.31}$$

$$\begin{aligned} \delta^{[2]}(t) &= \int_0^t A(\xi_1) d\xi_1 - \frac{1}{2} \int_0^t \Big[\int_0^{\xi_1} A(\xi_2) d\xi_2, A(\xi_1) \Big] d\xi_1 \\ &+ \frac{1}{12} \int_0^t \Big[\int_0^{\xi_1} A(\xi_2) d\xi_2, \Big[\int_0^{\xi_1} A(\xi_2) d\xi_2, A(\xi_1) \Big] \Big] d\xi_1 + ..., \end{aligned} \tag{3.32}$$

$$\begin{aligned} \delta^{[3]}(t) &= \int_0^t A(\xi_1) d\xi_1 - \frac{1}{2} \int_0^t \Big[\int_0^{\xi_1} A(\xi_2) d\xi_2, A(\xi_1) \Big] d\xi_1 \\ &+ \frac{1}{12} \int_0^t \Big[\int_0^{\xi_1} A(\xi_2) d\xi_2, \Big[\int_0^{\xi_1} A(\xi_2) d\xi_2, A(\xi_1) \Big] \Big] d\xi_1 \\ &+ \frac{1}{4} \int_0^t \Big[\int_0^{\xi_1} \Big[\int_0^{\xi_2} A(\xi_3) d\xi_3, A(\xi_2) \Big] d\xi_2, A(\xi_1) \Big] d\xi_1 \\ &- \frac{1}{24} \int_0^t \Big[\int_0^{\xi_1} \Big[\int_0^{\xi_2} A(\xi_3) d\xi_3, \Big[\int_0^{\xi_2} A(\xi_3) d\xi_3, A(\xi_2 \Big] \Big] d\xi_2, A(\xi_1) \Big] d\xi_1 \\ &- \frac{1}{24} \int_0^t \Big[\int_0^{\xi_1} \Big[\int_0^{\xi_2} A(\xi_3) d\xi_3, A(\xi_2) \Big] d\xi_2, \Big[\int_0^{\xi_1} A(\xi_2) d\xi_2, A(\xi_1) \Big] \Big] d\xi_1 \\ &- \frac{1}{24} \int_0^t \Big[\int_0^{\xi_1} A(\xi_2) d\xi_2, \Big[\int_0^{\xi_1} \Big[\int_0^{\xi_2} A(\xi_3) d\xi_3, A(\xi_2) \Big] d\xi_2, A(\xi_1) \Big] \Big] d\xi_1 \\ &+ ... \end{aligned} \tag{3.33}$$

and so on. The Picard theorem implies that $\delta(t) = \lim_{m\to\infty} \delta^{[m]}(t)$ exists in a suitably small neighbourhood of the origin and the above first few iterations indicate that it can be expanded as a linear combination of terms that are composed from integrals and commutators acting recursively on the matrix A. This is called as the Magnus expansion.

$$\delta(t) = \sum_{j=0}^{\infty} H_j(t) \tag{3.34}$$

where each H_j is a linear combination of terms that include exactly $(j+1)$ integrals (or j commutators). Thus;

$$\begin{aligned}
H_0(t) &= \int_0^t A(\xi_1)d\xi_1 & (3.35)\\
H_1(t) &= -\frac{1}{2}\int_0^t \Big[\int_0^{\xi_1} A(\xi_2)d\xi_2, A(\xi_1)\Big]d\xi_1 & (3.36)\\
H_2(t) &= \frac{1}{12}\int_0^t \Big[\int_0^{\xi_1} A(\xi_2)d\xi_2, \Big[\int_0^{\xi_1} A(\xi_2)d\xi_2, A(\xi_1)\Big]\Big]d\xi_1 & (3.37)\\
&+ \frac{1}{4}\int_0^t \Big[\int_0^{\xi_1} \Big[\int_0^{\xi_2} A(\xi_3)d\xi_3, A(\xi_2)\Big]d\xi_2, A(\xi_1)\Big]d\xi_1
\end{aligned}$$

$$\begin{aligned}
H_3(t) &= -\frac{1}{24}\int_0^t \Big[\int_0^{\xi_1} \Big[\int_0^{\xi_2} A(\xi_3)d\xi_3, \Big[\int_0^{\xi_2} A(\xi_3)d\xi_3, A(\xi_2\Big]\Big]d\xi_2, A(\xi_1)\Big]d\xi_1\\
&- \frac{1}{24}\int_0^t \Big[\int_0^{\xi_1} \Big[\int_0^{\xi_2} A(\xi_3)d\xi_3, A(\xi_2)]d\xi_2, \Big[\int_0^{\xi_1} A(\xi_2)d\xi_2, A(\xi_1)\Big]\Big]d\xi_1\\
&- \frac{1}{24}\int_0^t \Big[\int_0^{\xi_1} A(\xi_2)d\xi_2, \Big[\int_0^{\xi_1} \Big[\int_0^{\xi_2} A(\xi_3)d\xi_3, A(\xi_2)\Big]d\xi_2, A(\xi_1)\Big]\Big]d\xi_1\\
&- \frac{1}{8}\int_0^t \Big[\int_0^{\xi_1} A(\xi_2)d\xi_2, \Big[\int_0^{\xi_1} A(\xi_2)d\xi_2, \Big[\int_0^{\xi_1} A(\xi_2)d\xi_2, A(\xi_1)\Big]\Big]\Big]d\xi_1
\end{aligned} \tag{3.38}$$

Firstly Willhelm Magnus generalized the Baker-Campbell-Hausdorff formula

$$log(e^x e^y) = x + y + \frac{1}{2}[x,y] + \frac{1}{12}([x,[x,y]] - [y,[x,y]]) + ... \tag{3.39}$$

and derived an asymptotic expansion for $\delta(t)$ as $t \to 0$ as:

$$\begin{aligned}
\delta(t) &= \int_0^t A(\xi_1)d\xi_1 - \frac{1}{2}\int_0^t\int_0^{\xi_1} \Big[A(\xi_2), A(\xi_1)\Big]d\xi_2 d\xi_1\\
&+ \frac{1}{4}\int_0^t\int_0^{\xi_1}\int_0^{\xi_2} \Big[\Big[A(\xi_3), A(\xi_2)\Big], A(\xi_1)\Big]d\xi_3 d\xi_2 d\xi_1\\
&+ \frac{1}{12}\int_0^t\int_0^{\xi_1}\int_0^{\xi_2} \Big[A(\xi_3), \Big[A(\xi_2), A(\xi_1)\Big]\Big]d\xi_3 d\xi_2 d\xi_1 + ...
\end{aligned} \tag{3.40}$$

Although Magnus carried the expansion further, to terms consisting of fourfold integrals, he neither presented a general formula nor convergence proof (Iserles and Norsett 1997b). This did not prevent the Magnus expansion from being used in literally hundreds of papers in theoretical physics and quantum chemistry. Comprehensive analysis of the Magnus expansion within the context of numerical analysis has been carried out recently by Iserles and Norsett.

Note that all individual terms in Eq.(3.40) are in Lie algebra g, so does their linear combination. As we will see in the section about numerical experiments, this gives many advantages like less error since the numerical solution stays in the same manifold as the exact flow.

3.4. Convergence of the Magnus Series Method

Theorem 3.2 *Suppose that the Lie algebra* g *is equipped with the norm* $\|\cdot\|$. *The Magnus expansion (3.40) absolutely converges in this norm for every* $t \geq 0$ *such that*

$$\int_0^t \|a(\xi)\| d\xi \leq \int_0^{2\Pi} \frac{d\xi}{4+\xi[1-cot(\xi/2)]} \approx 1.086868702 \tag{3.41}$$

Proof: Integrating (3.31) and taking norms, we have by the triangular inequality and the trivial bound $\|ad_B^j A\| \leq (2\|B\|)^j \|A\|$ that

$$\begin{aligned} \|\delta(t)\| &= \| \int_0^t dexp_{\delta(\xi)}^{-1} a(\xi) d\xi \| \leq \int_0^t \|dexp_{\delta(\xi)}^{-1} a(\xi)\| d\xi \\ &\leq \int_0^t \sum_{j=0}^{\infty} \frac{|B_j|}{j!} (2\|\delta(\xi)\|)^j \|a(\xi)\| d\xi = \int_0^t g(2\|\delta(\xi)\|) \|a(\xi)\| d\xi, \end{aligned}$$

where

$$g(x) = 2 + \frac{x}{2}(1 - cot\frac{x}{2})$$

Now we use a Bihari-type inequality from Moan(1998): suppose that $h, g, , v \in \mathbf{C}(0, t^*)$ are positive and that g is nondecreasing. Then

$$h(t) \leq \int_0^t g(h(\xi)) v(\xi) d\xi \qquad t \in (0, t^*)$$

implies that

$$h(t) \leq \tilde{g}^{-1}(\int_0^t v(\xi)d\xi), \quad t \in (0, t^{**}), \quad where \quad \tilde{g} = \int_0^x \frac{d\xi}{g(\xi)}$$

and $t^{**} \in (0, t^*]$ is such that $\tilde{g}(\int_0^t v(\xi)d\xi)$is bounded in $(0, t^{**})$. In our cases $h(t) = 2\|\delta(t)\|$, $v(t) = \|a(t)\|$ and $g(t)$ are all positive and the letter is nondecreasing for $t \in (0, 2\pi)$. Therefore,

$$\|\delta(t)\| \leq \frac{1}{2}\tilde{g}^{-1}(\int_0^t \|a(\xi)\|d\xi)$$

and $\|\delta(t)\|$ is bounded, provided that $\tilde{g}(\int_0^t \|a(\xi)\|d\xi)$ is bounded. $\square$

Differential equations evolving on a homogeneous space M can be locally written as

$$y' = \lambda_y(f(t, y)), \qquad t \geq 0, \qquad y(t_0) = y_0 \in M \tag{3.42}$$

where $f : \mathbb{R} \times M \to \mathbf{g}$ is Lipschitz ($\mathbf{g}$ being the Lie-algebra of $\mathcal{G}$), while the map $\lambda_y : \mathbf{g} \to TM$ is defined as

$$\lambda_y(X) = \frac{d}{d\varepsilon}\lambda(exp(\varepsilon X), y)|_{\varepsilon=0}$$

Lie-group solvers, in that manner, Magnus Series Method, can be extended to homogeneous spaces (Munthe-Kaas, 1999). The homogeneous space (3.1) is translated to the Lie-Group $\mathcal{G}$, then to the Lie-algebra g, solved there and the solution is mapped to again M (Celledoni et al. 2002), as it is shown in the following Figure (3.1).

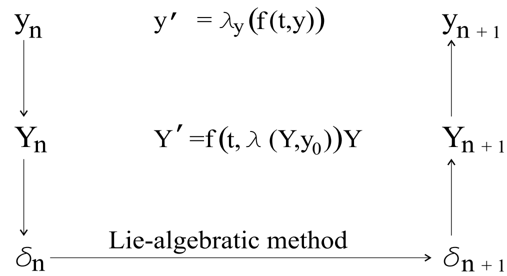

Figure 3.1: Illustration of the Method

(Source: Celledoni et al. 2002)

After the proof of the last convergency theorem, we can introduce the use of the Magnus Series in the numerical analysis.

3.5. Implementation of Magnus Series Method

Magnus Methods, obtained by applying symmetric quadrature formulas to the Magnus Series:

$$\begin{aligned}
\delta(t) &= \int_0^t A(\xi_1)d\xi_1 - \frac{1}{2}\int_0^t \int_0^{\xi_1} [A(\xi_2), A(\xi_1)]d\xi_2 d\xi_1 \\
&+ \frac{1}{4}\int_0^t \int_0^{\xi_1} \int_0^{\xi_2} [[A(\xi_3), A(\xi_2)], A(\xi_1)]d\xi_3 d\xi_2 d\xi_1 \\
&+ \frac{1}{12}\int_0^t \int_0^{\xi_1} \int_0^{\xi_2} [A(\xi_3), [A(\xi_2), A(\xi_1)]]d\xi_3 d\xi_2 d\xi_1 + ...
\end{aligned} \tag{3.43}$$

The simplest formulas for numerical solution of Eq.(3.43) based on the Magnus expansion, are

$$\begin{aligned}
\delta_{n+1} &= hA(t_n + h/2), \\
y_{n+1} &= e^{\delta_{n+1}} y_n
\end{aligned}$$

known as **MG2** (second order Magnus Method) and it is based on Gauss-Legendre points. The other one:

$$\begin{aligned}
\delta_{n+1} &= h(A(t_n) + A(t_n + h))/2 \\
y_{n+1} &= e^{\delta_{n+1}} y_n
\end{aligned}$$

which is called **MG2L** and is based on Gauss-Lobatto points (Orel 2001).

Iserles and Norsett (Iserles and Norsett 1997a) introduced the fourth-order method based on the Gauss-Legendre points. The forth order formula for the first four integrals in the Magnus expansion (3.43) are

$$\begin{aligned}
\int_0^h A(t_1)dt_1 &\approx \frac{1}{2}h(A_1 + A_2), \\
\int_0^h \int_0^{t_1} [A(t_1), A(t_2)]dt_2 dt_1 &\approx -\frac{\sqrt{3}}{6}h^2[A_1, A_2], \\
\int_0^h \int_0^{t_1} \int_0^{t_2} [A(t_1), [A(t_2), A(t_3)]]dt_3 dt_2 dt_1 &\approx h^3[(\frac{3}{80} - \frac{\sqrt{3}}{48})A_1 - (\frac{3}{80} + \frac{\sqrt{3}}{48})A_2, [A_1, A_2]], \\
\int_0^h \int_0^{t_1} \int_0^{t_2} [[A(t_1), A(t_2)], A(t_3)]dt_3 dt_2 dt_1 &\approx h^3[(\frac{3}{80} + \frac{\sqrt{3}}{16})A_1 - (\frac{3}{80} - \frac{\sqrt{3}}{16})A_2, [A_1, A_2]].
\end{aligned}$$

where

$$A_1 = A((\frac{1}{2} - \frac{\sqrt{3}}{6})h),$$
$$A_2 = A((\frac{1}{2} + \frac{\sqrt{3}}{6})h).$$

Assembling the above quadrature results in a fourth-order method for the linear equation which respects arbitrary Lie-group structure,

$$\begin{aligned}
A_1 &= A(t_n + (\frac{1}{2} - \frac{\sqrt{3}}{6})h), \qquad A_2 = A(t_n + (\frac{1}{2} + \frac{\sqrt{3}}{6})h) \\
\Sigma &= \frac{1}{2}h[A_1 + A_2] - \frac{\sqrt{3}}{12}h^2[A_1, A_2] + \frac{1}{80}h^3[A_1 - A_2, [A_1, A_2]] \\
y_{n+1} &= e^{\Sigma}y_n
\end{aligned}$$

Higher order quadrature requires the evaluation of larger number of commutators which costs expensively in the manner of a numerical criterion.

CHAPTER 4

MAGNUS SERIES METHOD FOR NONLINEAR DIFFERENTIAL EQUATIONS

In Chapter 3, we deal with the linear Lie-type equation, the same procedure is valid except in some details for the nonlinear Lie-type equation

$$Y' = A(t, Y)Y, \qquad Y(0) = Y_0 \in \mathcal{G} \tag{4.1}$$

where $\mathcal{G}$ is a matrix Lie-Group, $A : \mathbb{R}_+ \times \mathcal{G} \to$ g and g is the corresponding Lie algebra.

As we told in the previous chapters, the technique to solve the Eq.(4.1) is to carry $Y(t)$ from $\mathcal{G}$ to the corresponding Lie-algebra g (the tangent space of Lie Group at the identity of $\mathcal{G}$ (Iserles et al. 1998)) by using the exponential map, then solve it there and finally lift the solution back to $\mathcal{G}$. So, the first step is to map the solution to g such that

$$Y(t) = e^{\Omega(t)} Y_0. \tag{4.2}$$

As we obtained in the linear case Ω is the solution of the following differential equation

$$\Omega' = dexp_{\Omega}^{-1}(A(t, Y)), \qquad \Omega(0) = O.$$

If we substitute mapping into the $A(t, Y)$, then

$$\Omega' = dexp_{\Omega}^{-1}(A(t, e^{\Omega} Y_0)), \qquad \Omega(0) = O. \tag{4.3}$$

We have to remind the reader some operators

$$dexp_{\Omega}^{-1}(C) = \sum_{k=0}^{\infty} \frac{B_k}{k!} ad_{\Omega}^{k} C$$

$\{B_k\}_{m \in \mathbb{Z}_+}$ are the Bernoulli numbers and ad^m is the adjoint representation:

$$ad_{\Omega}^{0} C = C, \qquad ad_{\Omega}^{m+1} C = [\Omega, ad_{\Omega}^{m} C], \qquad m \geq 0$$

and some Bernoulli numbers are:

$$\begin{aligned} B_0 &= 1,\ B_1 = \frac{-1}{2},\ B_2 = \frac{1}{6} \\ B_3 &= 0,\ B_4 = -\frac{1}{30},\ B_5 = 0 \\ B_6 &= \frac{1}{42} \dots \end{aligned}$$

and so on. Note that for all odd integers $k \geq 3$, B_k's are all zero.

In the linear case we use Picard's Iteration to solve the eq.(4.3), we can use the same procedure as follows (WEB-1, 2005)

$$\begin{aligned}
\Omega^{[0]}(t) &\equiv O, \\
\Omega^{[m+1]}(t) &= \int_0^t dexp^{-1}_{\Omega^{[m]}(s)} A(s, e^{\Omega^{[m]}(s)} Y_0) ds, \\
&= \int_0^t \sum_{k=0}^{\infty} \frac{B_k}{k!} ad^k_{\Omega^{[m]}(s)} A(s, e^{\Omega^{[m]}(s)} Y_0) ds., \qquad m \geq 0
\end{aligned}$$

The power series of $\Omega^{[k]}(t)$ and $\Omega^{[k+1]}(t)$ differ in terms of only $O(t^{m+1})$, then we can omit all terms of $\Omega^{[k]}(t)$ with the order greater than $O(t^m)$. If $\Omega^{[0]} = O$ and if we choose $m = 0$ for the first step, then

$$\begin{aligned}
\Omega^{[1]} &= \int_0^t \sum_{k=0}^{\infty} \frac{B_k}{k!} ad^k_{\Omega^{[0]}=O} A(s, e^{\Omega^{[0]}=O} Y_0) ds \\
\Omega^{[1]} &= \int_0^t \frac{B_0}{0!} ad^0_O A(s, Y_0) + \frac{B_1}{1!} ad^1_O A(s, Y_0) + \frac{B_2}{2!} ad^2_O A(s, Y_0) + \ldots ds
\end{aligned}$$

which we can see that except first term in the integral, all the terms give zero because of the zero multiplication of the adjoint representation. Then $\Omega^{[1]}$ is equal to only to the first term in the integral, therefore

$$\Omega^{[1]} = \int_0^t A(s, Y_0) ds = \Omega(t) + O(t^2) \tag{4.4}$$

Now, let us look at the second term of Picard's Ω. If we choose $m = 2$, then

$$\begin{aligned}
\Omega^{[2]} &= \int_0^t \sum_{k=0}^{\infty} \frac{B_k}{k!} ad^k_{\Omega^{[1]}} A(s, e^{\Omega^{[1]}} Y_0) ds \\
&= \int_0^t \frac{B_0}{0!} ad^0_{\Omega^{[1]}} A(s, e^{\Omega^{[1]}} Y_0) + \frac{B_1}{1!} ad^1_{\Omega^{[1]}} A(s, e^{\Omega^{[1]}} Y_0) + \frac{B_2}{2!} ad^2_{\Omega^{[1]}} A(s, e^{\Omega^{[1]}} Y_0) + \ldots ds \\
&= \int_0^t A(s, e^{\Omega^{[1]}} Y_0) - \frac{1}{2} \int_0^t [\Omega^{[1]}, A(s, e^{\Omega^{[1]}} Y_0)] ds + O(t^3)
\end{aligned}$$

but here,

$$-\frac{1}{2} \int_0^t [\Omega^{[1]}, A(s, e^{\Omega^{[1]}} Y_0)] ds = O(t^3)$$

When $\Omega^{[3]}$ is computed, it is seen that $\Omega^{[3]}$ produces $\Omega^{[2]}$ up to $O(t^2)$, this is valid for all indexes of $\Omega^{[k]}$, this means that we can truncate $dexp^{-1}$ at $k = m - 2$ term for $\Omega^{[m]}$ and take generally

$$
\begin{aligned}
\Omega^{[1]} &= \int_0^t A(s, Y_0)ds, &(4.5)\\
\Omega^{[m]} &= \sum_{k=0}^{m-2} \frac{B_k}{k!} \int_0^t ad^k_{\Omega^{[m-1]}} A(s, e^{\Omega^{[m-1]}} Y_0)ds. \qquad m \geq 2 &(4.6)
\end{aligned}
$$

Detailed discussion can be found in the reference (WEB-1, 2005). Now, it is time to compute the values of the integrals by using numerical quadrature formulas. If we use Euler's formula for $\Omega^{[1]}$, then

Order 1:

$$
\Omega^{[1]} = \int_0^t A(s, Y_0)ds = hA(0, Y_0) + O(h^2), \tag{4.7}
$$

Order 2: Discretization of the $\Omega^{[2]}$ by the trapezoidal rule is

$$
\Omega^{[2]} = \int_0^t A(s, e^{\Omega^{[1]}} Y_0)ds = \frac{h}{2}\Big(A(0, Y_0) + A(s, e^{\Omega^{[1]}} Y_0)\Big) + O(h^3), \tag{4.8}
$$

Order 3:

$$
\Omega^{[3]} = \int_0^t \Big(A_2(s) - \frac{1}{2}[\Omega^{[2]}(s), A_2(s)]\Big)ds, \tag{4.9}
$$

where $A_2(s) \equiv A(s, e^{\Omega^{[3]}} Y_0)$. If we use Simpson's rule, we have

$$
\begin{aligned}
\Omega^{[3]} &= \frac{h}{6}\Big(A(0, Y_0) + 4A_2(h/2) + A_2(h)\Big) - \frac{h}{3}[\Omega^{[2]}(h/2), A_2(h/2)] &(4.10)\\
&- \frac{h}{12}[\Omega^{[2]}(h), A_2(h)] + O(h^4).
\end{aligned}
$$

In literature, this is a recent research on the Magnus Series Method for non-linear differential equations. Fernando Casas and Arieh Iserles published an article about this in 2005 (WEB-1, 2005), however they did not applied this method to any differential equation as a numerical simulation in their article, and they did not introduced the method with the order higher than 3 (fourth order is given in the appendix section as an algorithm). Therefore, this is a topic of on going research. In our study, we obtained the method with the order 4 and 5 and applied the results to a nonlinear differential equation.

Order 4:

$$
\Omega^{[4]}(t) = \sum_{k=0}^{2} \frac{B_k}{k!} \int_0^t ad^k_{\Omega^{[3]}(s)} A_3(s)ds,
$$

where $A_3(s) = A(s, e^{\Omega^{[3]}(s)}Y_0,$

$$= \int_0^t \Big(\frac{B_0}{0!}A_3(s)ds + \frac{B_1}{1!}[\Omega^{[3]}, A_3]ds + \frac{B_2}{2!}[\Omega^{[3]}, [\Omega^{[3]}, A_3]]\Big)ds,$$

If we call all the function under the integral as e.g $K(s)$ and use Simpson's rule

$$\int_0^t K(s)ds = \frac{h}{6}[K(0) + 4K(\frac{h}{2}) + K(h)].$$

Then fourth-order method is

$$\begin{aligned} \Omega^{[4]}(h) &= \frac{h}{6}A(0, Y_0) + \frac{h}{6}4A_3(\frac{h}{2}) - \frac{h}{3}\Big[\Omega^{[3]}(\frac{h}{2}), A_3(\frac{h}{2})\Big] \\ &+ \frac{h}{18}\Big[\Omega^{[3]}(\frac{h}{2}), \Big[\Omega^{[3]}(\frac{h}{2}), A_3(\frac{h}{2})\Big]\Big] + \frac{h}{6}A_3(h) \\ &- \frac{h}{12}\Big[\Omega^{[3]}(h), A_3(h)\Big] + \frac{h}{72}\Big[\Omega^{[3]}(h), \Big[\Omega^{[3]}(h), A_3(h)\Big]\Big]. \end{aligned} \tag{4.11}$$

Order 5: In order 5, the coefficients makes it easier to compute the integrals because the fourth Bernoulli number $B_3 = 0$ as we will see.

$$\Omega^{[5]}(t) = \sum_{k=0}^{3} \frac{B_k}{k!} \int_0^t ad^k_{\Omega^{[4]}(s)} A_4(s)ds,$$

where $A_4(s) = A(s, e^{\Omega^{[4]}(s)}Y_0,$

$$= \int_0^t \Big(\frac{B_0}{0!}A_4(s)ds + \frac{B_1}{1!}[\Omega^{[4]}, A_3]ds + \frac{B_2}{2!}[\Omega^{[4]}, [\Omega^{[4]}, A_4]]\Big)ds.$$

Here $B_3 = 0$, and if we again call the function under the integral as e.g $M(s)$, then

$$\int_0^t M(s)ds = \frac{h}{6}[M(0) + 4M(\frac{h}{2}) + M(h)].$$

Then the fifth-order method is

$$\begin{aligned} \Omega^{[5]}(h) &= \frac{h}{6}A(0, Y_0) + \frac{h}{3}A_4(\frac{h}{2}) - \frac{h}{3}\Big[\Omega^{[4]}(\frac{h}{2}), A_4(\frac{h}{2})\Big] \\ &+ \frac{h}{18}\Big[\Omega^{[4]}(\frac{h}{2}), \Big[\Omega^{[4]}(\frac{h}{2}), A_4(\frac{h}{2})\Big]\Big] + \frac{h}{6}A_4(h) \\ &- \frac{h}{12}\Big[\Omega^{[4]}(h), A_4(h)\Big] + \frac{h}{72}\Big[\Omega^{[4]}(h), \Big[\Omega^{[4]}(h), A_4(h)\Big]\Big]. \end{aligned} \tag{4.12}$$

In the next chapter, we will apply Magnus Series Method to the both linear and nonlinear initial value problems in order to show the accuracy of the method.

CHAPTER 5

APPLICATIONS OF MAGNUS SERIES METHOD

In this chapter several linear and nonlinear differential equations are considered and the results are compared with a classical numerical method, Runge-Kutta. This Method is based on the series so it would be the best choice to choose this method for comparison.

5.1. Brief Introduction to Runge-Kutta Method

The Runge-Kutta Method is based on the Taylor Series expansion of $y(x)$ (Atkinson 1988), where $y(x)$ is the solution of the initial value problem given in Eq.(5.1)

$$y' = f(t, y) \qquad y(x_0) = y_0 \tag{5.1}$$

After expanding $y(x_1)$ about x_0 using Taylor's theorem:

$$y(x_1) = y(x_0) + hy'(x_0) + ... + \frac{h^r}{r!}y^{(r)}(x_0) + \frac{h^{r+1}}{(r+1)!}y^{(r+1)}(\xi)$$

for some $x_0 \leq \xi_0 \leq x_1$, we have an approximation for $y(x_1)$ by dropping the remainder term, provided we can calculate $y''(x_0)$,...$y^{(r)}(x_1)$ by differentiating $y'(x) = f(x, y(x))$. This Taylor Series Method can give excellent results but we need to differentiate f. The Runge-Kutta Method is closely related to the Taylor Series expansion, but no differentiation of f is necessary.

All RK (Runge-Kutta is abbreviated to RK) methods can be written in the form

$$y_{n+1} = y_n + \sum_{i=1}^{m} b_i f_i \tag{5.2}$$

where

$$f_k = hf(t_n + c_k h, \theta_k) \tag{5.3}$$

$$\theta_k = y_n + \sum_{i=1}^{m} a_{k,i} f_i \tag{5.4}$$

for $k = 1, ..., m$ and the coefficients can be obtained by using Butcher Table (Hairer et al. 1993)

$$\begin{array}{c|cccc}
c_1 & a_{1,1} & a_{1,2} & \dots & a_{1,m} \\
c_2 & a_{2,1} & a_{2,2} & \dots & a_{2,m} \\
\vdots & \vdots & \vdots & & \vdots \\
c_m & a_{m,1} & a_{m,2} & \dots & a_{m,m} \\
\hline
 & b_1 & b_2 & \dots & b_m
\end{array}$$

Fourth-order RK, RK4, is used as the classical method, especially for being in the same order with the Magnus Series Method's order. If RK4 is applied to the equation with the coefficients from the Butcher Table:

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

from t_0 to $t_{n+1} = t_0 + (n+1)h$, then the numerical solution is

$$y_{n+1} = y_n + \frac{1}{6}h(k_1 + 2k_2 + 2k_3 + k_4) \tag{5.5}$$

where

$$\begin{aligned}
k_1 &= f(t_n, y_n), \\
k_2 &= f(t_n + \frac{1}{2}h, y_n + \frac{1}{2}hk_1), \\
k_3 &= f(t_n + \frac{1}{2}h, y_n + \frac{1}{2}hk_2), \\
k_4 &= f(t_n + h, y_n + hk_3).
\end{aligned}$$

5.2. Application of Magnus Method to Linear Oscillatory Problems

Consider the general form of the oscillatory problems

$$y'' + g(t)y = 0 \tag{5.6}$$

the following transformation is used to obtain matrix Lie-type equations

$$y = y_1$$

$$y_1' = y_2$$

then the equation is in the form

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix}' = \begin{pmatrix} 0 & 1 \\ -g(t) & 0 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$$

If we call $\begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$ as Y and the matrix function $\begin{pmatrix} 0 & 1 \\ -g(t) & 0 \end{pmatrix}$ as A, we can write the equation as

$$Y' = A(t)Y \tag{5.7}$$

which is the linear Lie-type equation.

In this section, the various functions $g(t)$ in the Eq.(5.6) are chosen as $g(t) = 1$, $g(t) = t$ and $g(t) = t^2$. First, the linear oscillator equation is considered where $g(t) = 1$.

$$y'' + y = 0 \qquad y(0) = 1; \quad y'(0) = 0 \tag{5.8}$$

First step is to write this equation as a Lie-type equation which is also a linear system. If we apply the following transformation to obtain the system of equations in the form (5.6):

$$y = y_1$$

$$y_1' = y_2$$

then,

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix}' = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$$

with the initial condition $\begin{pmatrix} y_1(0) \\ y_2(0) \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$. We can find the exact solution by the classical ordinary differential equations method, here the exact solution is $y(t) = \cos(t)$.

We solved the Eq.(5.8) by both fourth order Magnus Series Method and Runge-Kutta Method. The time interval is taken as $[0, 100]$ and the step size is $h = 1/16$. In Fig.(5.1) and Fig.(5.2) the errors between the exact and Runge-Kutta and Magnus Series numerical methods are plotted as a function of time, respectively. As can be seen in these figures, Runge-Kutta has 10^{-5} while Magnus Series Method has 10^{-14}, thus 3 times better result is obtained by Magnus Series Method.

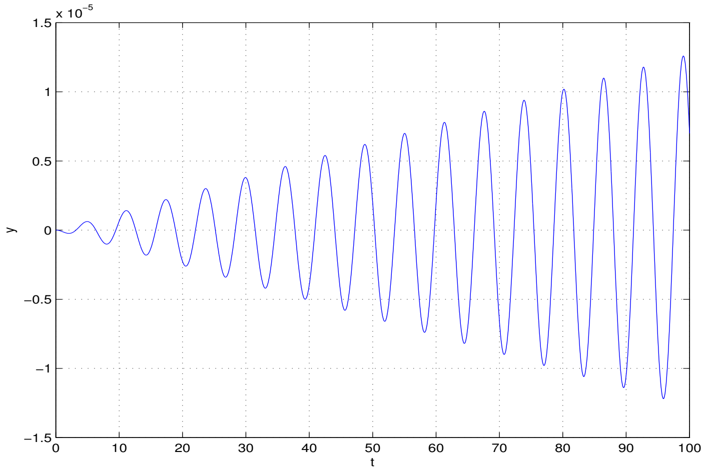

Figure 5.1: Global Error of the Runge-Kutta Method by solving the Linear Oscillatory Equation for the time interval (0,100) with the step size h=1/16.

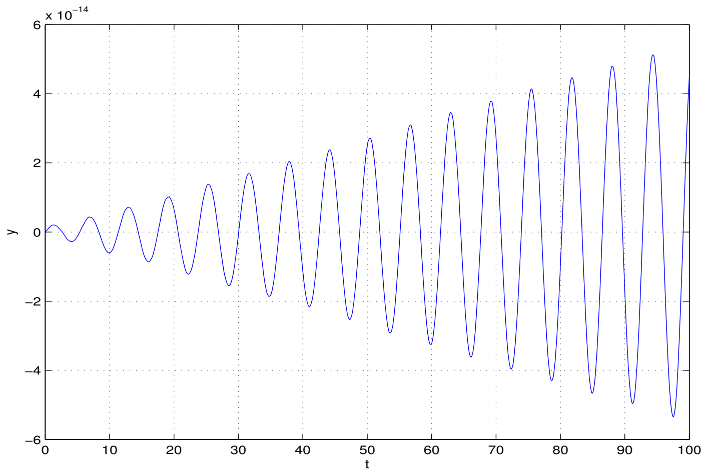

Figure 5.2: Global Error of the Magnus Series Method by solving the Linear Oscillatory Equation for the time interval (0,100) with the step size h=1/16.

Next equation we consider is the Airy equation

$$y'' + ty = 0 \tag{5.9}$$

with the initial conditions $y(0) = 1$ and $y'(0) = 0$. In general, it has an analytic solution $y(t) = c_1 Ai(t) + c_2 Bi(t)$, where c_1 and c_2 can be evaluated to yield the analytic result given in the following equation (Diele and Ragni 2002).

$$y(t) = \frac{Ai(-t)Bi'(0) - Ai'(0)Bi(-t)}{Ai(0)Bi'(0) - Ai'(0)Bi(0)}$$

If $t < 0$ then, it does not have an oscillatory behavior.

When we apply the necessary transformation as it has done in the previous section, the Eq.(5.9) becomes

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix}' = \begin{pmatrix} 0 & 1 \\ -t & 0 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \tag{5.10}$$

with the initial condition

$$\begin{pmatrix} y_1(0) \\ y_2(0) \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

In this case the matrix function $A(t)$ becomes

$$A(t) = \begin{pmatrix} 0 & 1 \\ -t & 0 \end{pmatrix}.$$

Figures (5.3) and (5.4) display the solution of the Airy Equation (5.8) with fourth order Runge-Kutta and Magnus Series Method respectively. In these computations, the time interval is taken as $(0, 1000)$ and the step size as $h = 1/16$. It is evident in Fig.(5.3) that Runge-Kutta Method loses accuracy for the longer integral interval. Fig.(5.5) and Fig.(5.6) show the global error of the Runge-Kutta and Magnus Series Method respectively. As it is seen in these figures that Runge-Kutta has 10^{-1} error while Magnus Series has 10^{-7}. Magnus Series Method performs better than Runge-Kutta Method when applied to the Airy Equation.

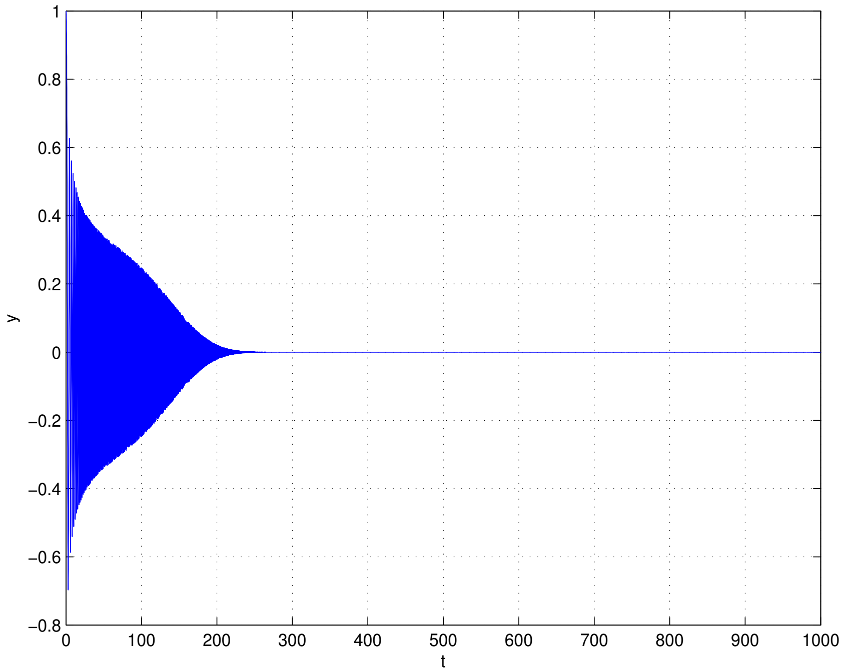

Figure 5.3: The Solution of the Airy Equation by Runge-Kutta Method for the time interval (0,1000) with the step size h=1/16.

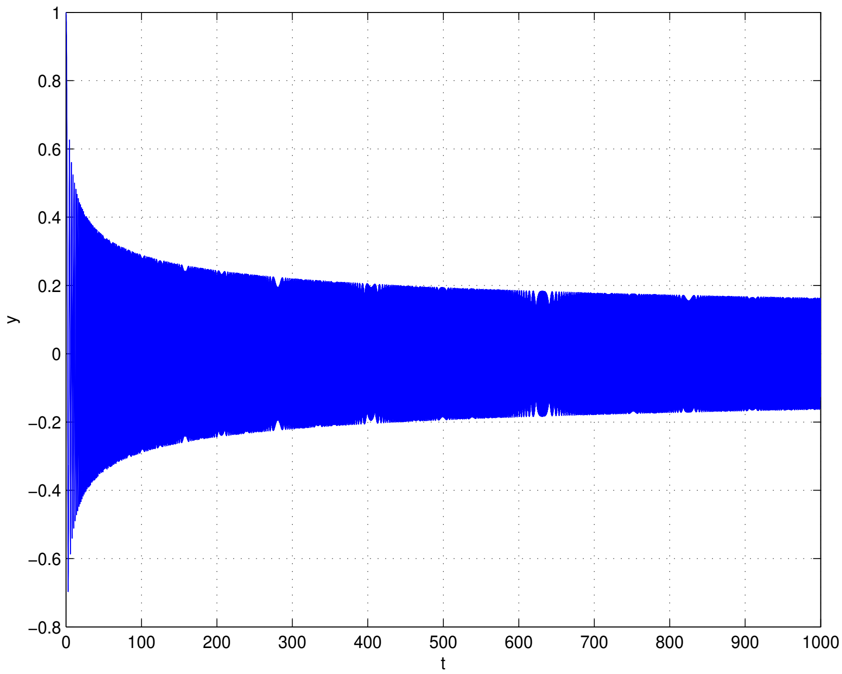

Figure 5.4: The Solution of the Airy Equation by Magnus Series Method for the time interval (0,1000) with the step size h=1/16.

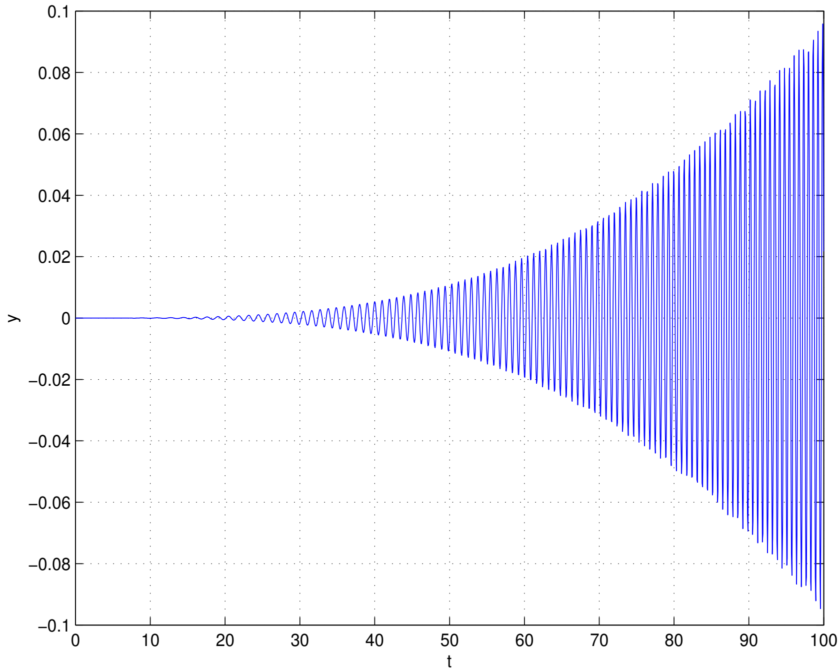

Figure 5.5: Global Error of the Runge-Kutta Method obtained by solving the Airy Equation for the interval (0,100) with the step size h=1/16.

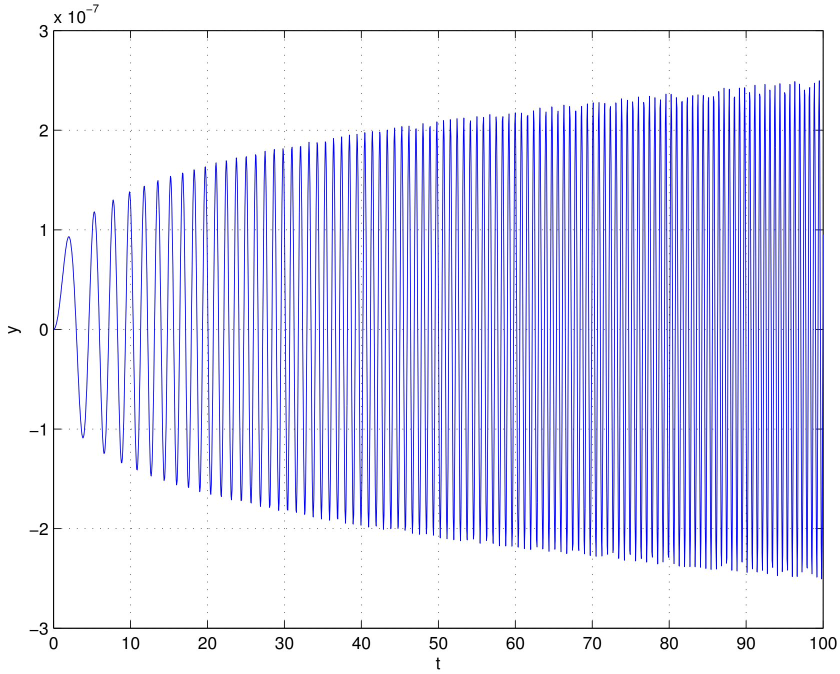

Figure 5.6: Global Error of the Magnus Series Method obtained by solving the Airy Equation for the interval (0,100) with the step size h=1/16.

Finally the following equation is considered with the given initial conditions

$$y'' + t^2 y = 0 \quad y(0) = 0 \quad y'(0) = 1. \tag{5.11}$$

After transformation, the equation is in the form of a Lie-type equation

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix}' = \begin{pmatrix} 0 & 1 \\ -t^2 & 0 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \tag{5.12}$$

The methods, Magnus and Runge-Kutta are applied to the Eq.(5.10) for the time interval $(0, 100)$ with the time step $h = 1/16$. The results are shown in Figures (5.7) and (5.8) respectively. As can be seen in these figures, although Magnus Series Method works for the Eq.(5.10), the other method does not work. In addition, error accumulation is higher than the other cases which can be seen in Fig.(5.7).

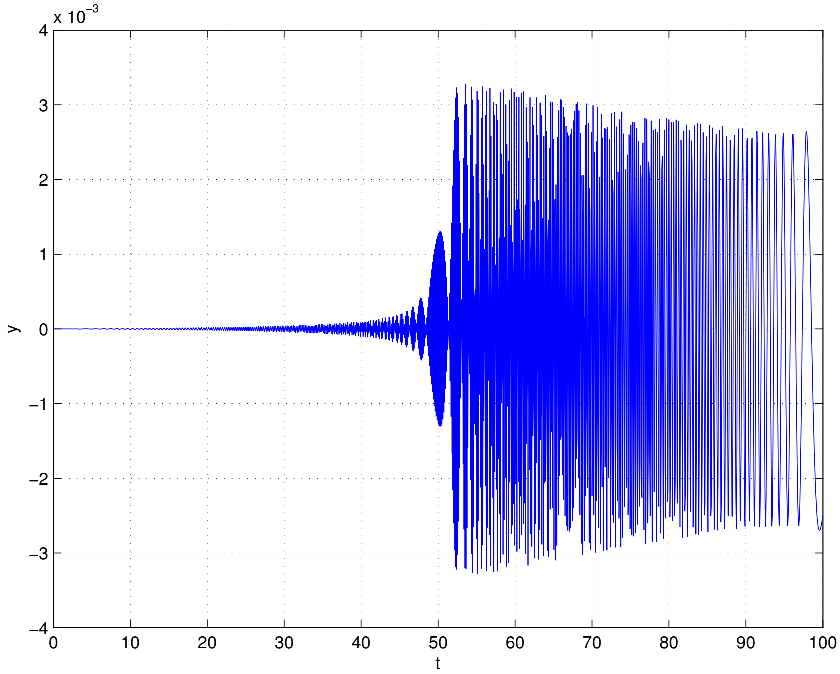

Figure 5.7: Global Error of the Magnus Series Method obtained by solving the Eq.(5.10) for the interval (0,100) with the step size h=1/16.

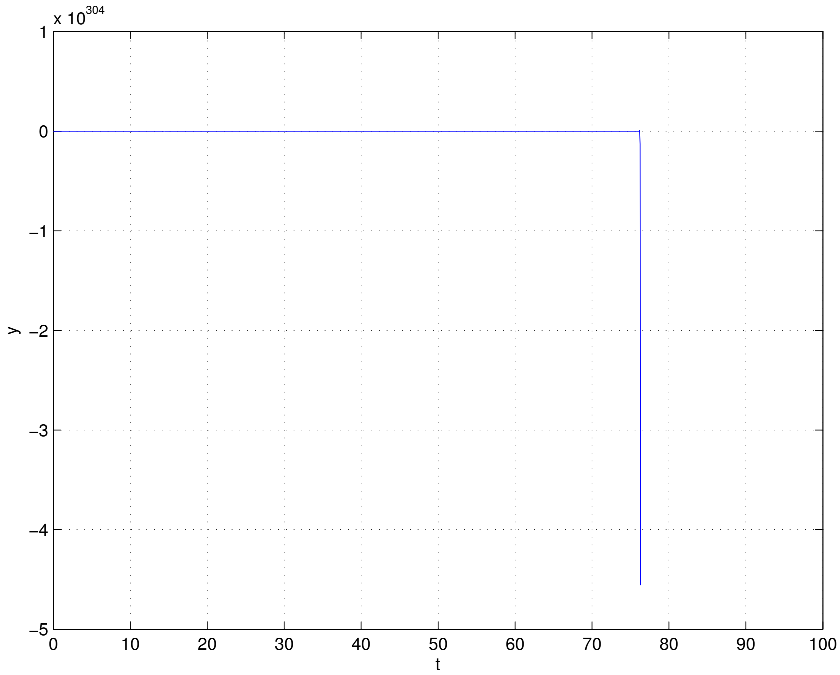

Figure 5.8: Global Error of the Runge-Kutta Method obtained by solving the Eq.(5.10) for the interval (0,100) with the step size h=1/16.

5.3. Application of Magnus Method to Nonlinear Oscillatory Problems

In this section, our aim is to validate the effectiveness of the Magnus Series Method in the field of nonlinear oscillatory system by applying it to the Duffing Equation (5.13) which we consider as test problem,

$$y'' + y + \varepsilon y^3 = 0, \qquad y(0) = 1; \qquad y'(0) = 0 \tag{5.13}$$

The exact solution for this equation is given by Eq.(5.14) obtaining from Multiple-scale Method (Bender and Orszag 1999) as

$$y(t) \approx \cos(t) + \varepsilon\Big(\frac{\cos(3t)}{32} - \frac{\cos(t)}{32} - \frac{3t\sin(t)}{8}\Big) \tag{5.14}$$

As can be seen in previous sections, after the transformation, we obtain the following Lie-type equation

$$Y' = A(t, y)Y \tag{5.15}$$

where the matrix function $A(t, y)$

$$A(t, y) = \begin{pmatrix} 0 & 1 \\ (1 + \varepsilon y^2) & 0 \end{pmatrix}$$

The Magnus Series and Runge-Kutta Methods are applied to the Duffing Equation (5.12) for the interval $(0, 1000)$ with the step size $h = 1/16$. The Figures (5.9) and (5.10) show the global error obtained by fourth-order Magnus Series Method and Runge-Kutta Method respectively. It is evident that Magnus Series Method has 10^{-6} error while Runge-Kutta has 10^{-4}. Thus Magnus Series Method gives better result than Runge-Kutta Method.

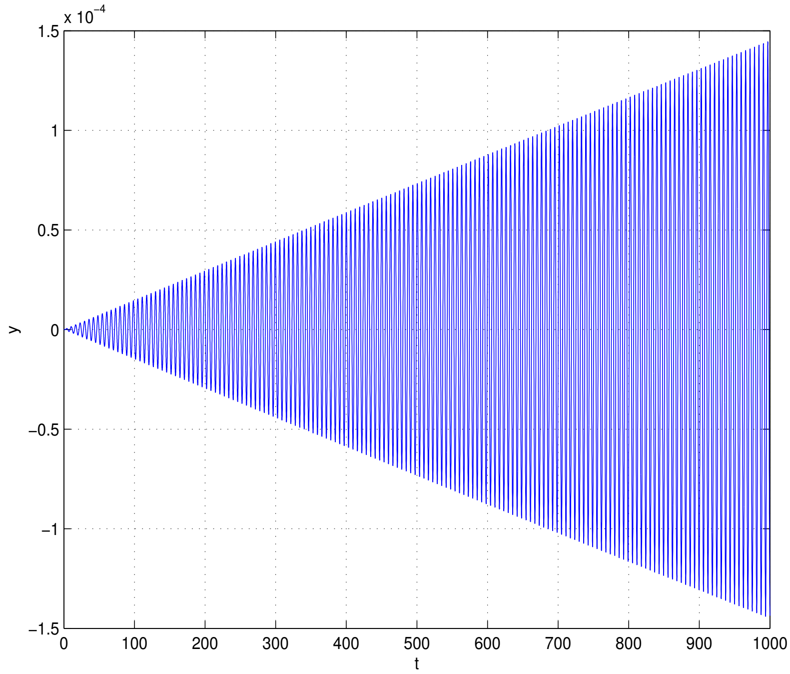

Figure 5.9: Global Error of the Runge-Kutta Method obtained by solving Duffing Equation for the interval (0,1000) with the step size h=1/16.

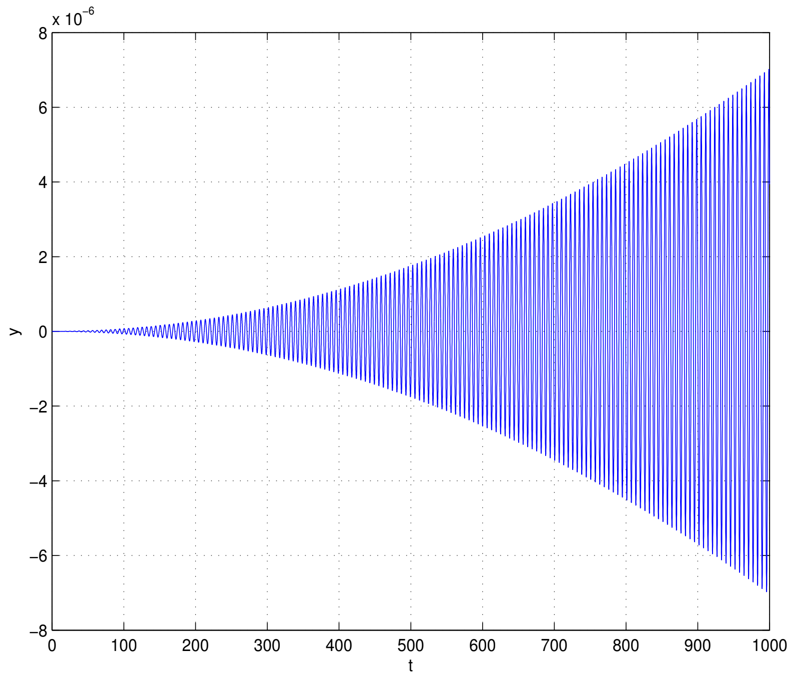

Figure 5.10: Global Error of the Magnus Series Method obtained by solving Duffing Equation for the interval (0,1000) with the step size h=1/16.

5.4. Magnus Method for Hamiltonian System

Conservative problems, particularly in dynamics, can be expressed in the form:

$$\frac{dp_i}{dt} = -\frac{\partial H}{\partial q_i} \tag{5.16}$$

$$\frac{dq_i}{dt} = +\frac{\partial H}{\partial p_i} \qquad i = 1, ..., d \tag{5.17}$$

where H is a given function called Hamiltonian of the system, q_i is a generalized coordinate, and p_i a generalized momentum and $(p, q) = (p_1, ..., p_d, q_1, ..., q_d) \in \omega$, the oriented Euclidean space $\mathbb{R}^{2d}$. The integer d is called the number of degrees of freedom and ω is the phase space. The Hamiltonian is defined by:

$$H(p, q) = T(p, q) + V(q) \tag{5.18}$$

where T is the kinetic energy and V the potential energy. The Hamiltonian systems have two important properties: First, the Hamiltonian is an energy integral of the motion, second, Hamiltonian systems conserve volumes in phase space. This means that if we draw the trajectories that originate from all of the points inside a region of volume V in phase space at $t = 0$, then the endpoints of these trajectories at time t fill a region with the same volume V for all t. This condition is Jacobian mathematically. Some examples are as follows:

The well-known system with $d = 1$ (one degree of freedom) is the Harmonic oscillator such that (Sanz-Serna and Calvo 1994):

$$H = T + V, \qquad T = \frac{p_1^2}{2m}, \qquad V = \frac{kq_1^2}{2}$$

m and k are positive constants that respectively correspond to mass and spring constant. The system is such that:

$$p' = -kq \qquad q' = \frac{p}{m}$$

The general solution for q is an oscillation $q(t) = C_1 \sin(\sqrt{((k/m))}t + C_2)$ where C_1 and C_2 are integration constants. Similarly $p(t) = m\sqrt{((k/m))}C_1 \cos(\sqrt{((k/m))}t + C_2)$. When plotted in the phase (p, q)-plane, the parametric curves corresponds to the ellipses.

The other example is the $pendulum$. If the units are chosen in such a way that the mass of the blob, the length of the rod and the acceleration of gravity are all unity, then

$$H = T + V, \qquad T = \frac{p^2}{2}, \qquad V = -\cos(q) \tag{5.19}$$

where q is the angle between the rod and a vertical, downward oriented axis. The equations of the motion are then:

$$p' = -\sin q \qquad q' = p \tag{5.20}$$

The third example and the one that is an application of the Magnus Method is the double Harmonic oscillator. This has two degrees of freedom and:

$$H = T + V, \qquad T = \frac{p_1^2 + p_2^2}{2}, \qquad V = \frac{q_1^2 + q_2^2}{2}$$

In the equations of the motion $p_1' = -q_1$, $q_1' = p_1$, $p_2' = -q_2$ and $q_2' = p_2$. The projections of the solutions onto the (p_i, q_i)-plane corresponds to the circles. If we write it as a linear lie-type system as it defined in the second section it would be:

$$\begin{pmatrix} p_1 \\ p_2 \\ q_1 \\ q_2 \end{pmatrix}' = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ q_1 \\ q_2 \end{pmatrix} \tag{5.21}$$

with the initial condition $[1\ 1\ 0\ 0]^T$. Magnus Series Method is applied to the Double Harmonic Oscillator Equation (5.21) for the interval $(-1, 1)$ for the time step $h = 1/16$. The Figures (5.11) and (5.12) display the solutions as functions of coordinate q_1 and q_2 on the (p_1, q_1) and (p_2, q_2) phase-plane respectively. The projections are circles as it is expected.

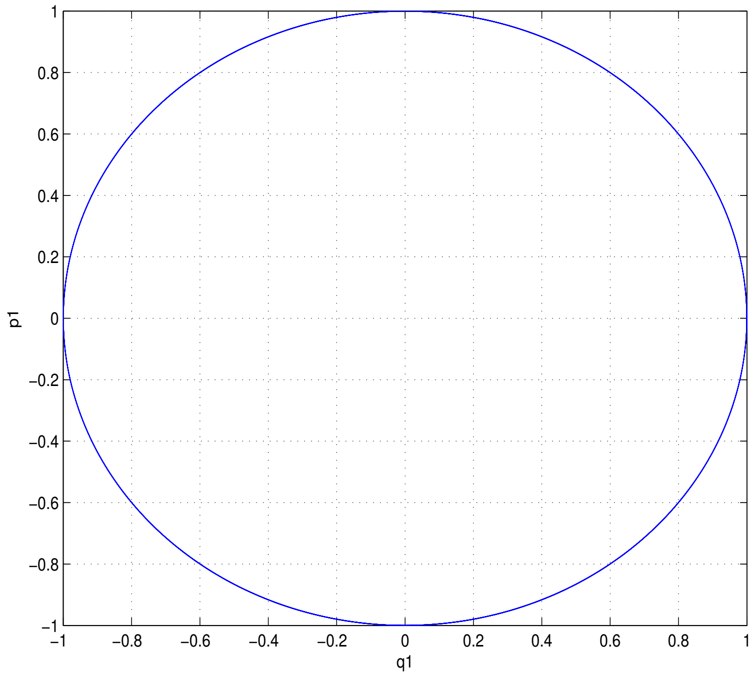

Figure 5.11: Projection of the Double Harmonic Oscillator for the first phase-plane by Magnus Method for the interval (-1,1) with the time step h=1/16.

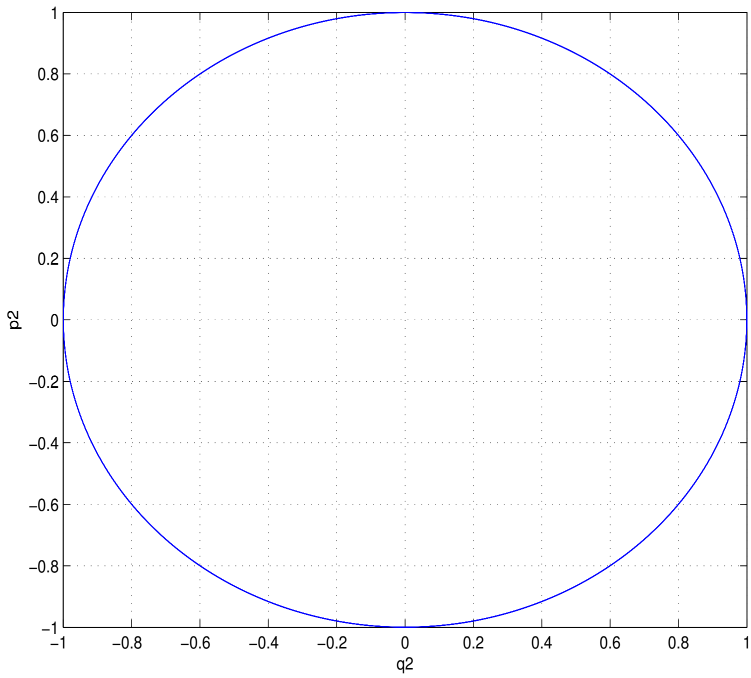

Figure 5.12: Projection of the Double Harmonic Oscillator for the second phase-plane by Magnus Method for the interval (-1,1) with the time step h=1/16.

5.5. Magnus Method for Boundary Value Problems

In this section; linear second order boundary value problem is solved by using Magnus Series Method. First, boundary value problem is converted to two initial value problems with the help of the Shooting Method, then two initial value problems are solved by Magnus Series Method.

Consider the following linear boundary value problem:

$$x'' = p(t)x' + q(t)x + r(t) \quad x(a) = \alpha \quad x(b) = \beta \tag{5.22}$$

which has a unique solution $x = x(t)$ over $a \leq t \leq b$. The solution of the Eq.(5.22) can be found by considering the following two special initial value problems.

$$\begin{aligned} u'' &= p(t)u' + q(t)u + r(t) \qquad u(a) = \alpha \quad u'(a) = 0 && (5.23) \\ v'' &= p(t)v' + q(t)v \qquad v(a) = 0 \quad v'(a) = 1 && (5.24) \end{aligned}$$

After the solution of each initial value problems are found, the linear combinations become the solution of the boundary value problem given in Eq.(5.22) as

$$x(t) = u(t) + Cv(t) \tag{5.25}$$

where $C = (\beta - u(b))/v(b)$.

As an illustration, the method is applied to the following simple linear oscillatory boundary value problem

$$x'' = -x \qquad x(0) = 1 \quad x(90) = 0.4481 \tag{5.26}$$

which has the exact solution $\cos(t)$. After problem is converted to the following initial value problem

$$\begin{aligned} u'' &= -u \quad u(0) = 1 \quad u'(0) = 0 \\ v'' &= -v \quad v(0) = 0 \quad v'(0) = 1 \end{aligned}$$

the Runge-Kutta and Magnus Series Method are applied to the each problem separately. Figure (5.13) and Figure (5.15) displays the solution of the boundary value problem (5.26) for the time interval $(0, 100)$ with the time step $h = 1/16$ by Runge-Kutta Method and Magnus Series Method respectively while Figure (5.14) and Figure (5.16) shows the error of Runge-Kutta and Magnus Series Method respectively for the same time interval and time step. As can be seen in these figures Magnus Series has less error than Runge-Kutta Method.

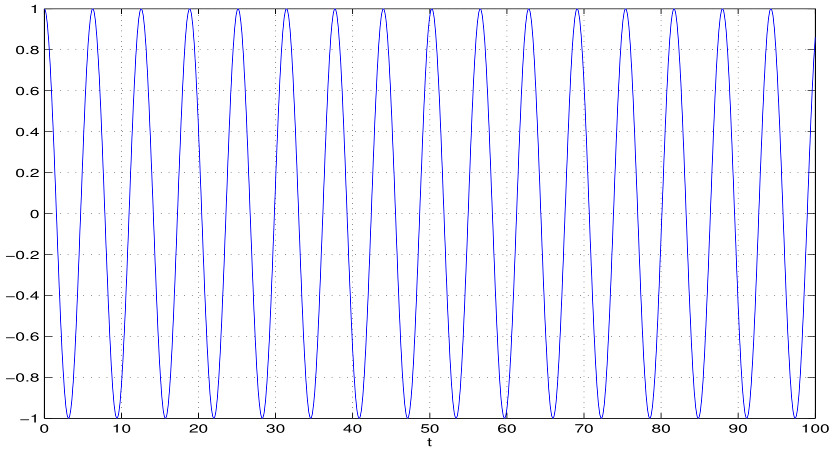

Figure 5.13: Solution of the Boundary Value Problem (5.26) by Runge-Kutta Method for the time interval (0,100) with the time step h=1/16.

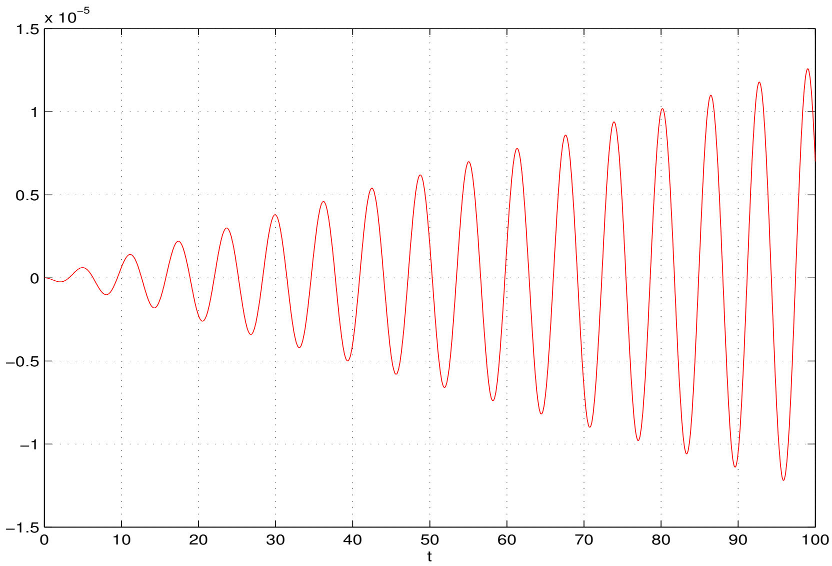

Figure 5.14: Global Error of the Boundary Value Problem (5.26) by Runge-Kutta Method for the time interval (0,100) with the time step h=1/16.

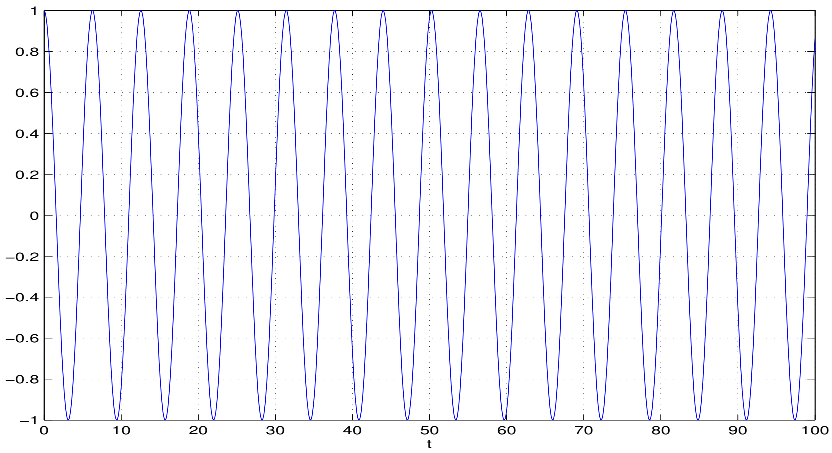

Figure 5.15: Solution of the Boundary Value Problem (5.26) by Magnus Series Method for the time interval (0,100) with the time step h=1/16.

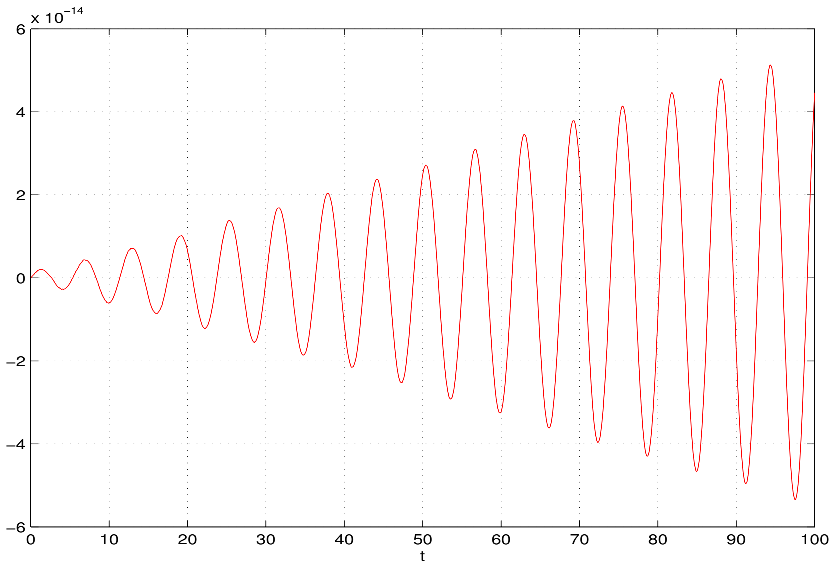

Figure 5.16: Global Error of the Boundary Value Problem (5.26) by Magnus Series Method for the time interval (0,100) with the time step h=1/16.

5.6. Magnus Method for Second Order System of Equations

In this section we consider the second order linear system as follows

$$Y'' = A(t)Y \qquad Y(0) = Y_0, \quad Y'(0) = \dot{Y}_0 \tag{5.27}$$

Magnus Series Method is applied to find the appropriate solution of this form of matrix differential equation. First consider $Y = \begin{pmatrix} x \\ y \end{pmatrix}$ such that

$$\begin{pmatrix} x \\ y \end{pmatrix}'' = \begin{pmatrix} a(t) & b(t) \\ c(t) & d(t) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \tag{5.28}$$

After the transformation $x = x_1$, $x_1' = x_2$ for x and $y = y_1$, $y_1' = y_2$ for y, the Lie- type equation (5.29) is obtained.

$$\begin{pmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \end{pmatrix}' = \begin{pmatrix} 0 & 1 & 0 & 0 \\ a(t) & 0 & b(t) & 0 \\ 0 & 0 & 0 & 1 \\ c(t) & 0 & d(t) & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \end{pmatrix} \tag{5.29}$$

As an illustration, the following equation is considered with the given initial conditions.

$$\begin{pmatrix} x \\ y \end{pmatrix}'' = \begin{pmatrix} -t & 0 \\ -1 & -t \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}, \quad Y(0) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad Y'(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{5.30}$$

which has the exact solution Airy equation for x and the derivative of the Airy equation for y.

In Fig.(5.17) first and second figures show the global error of the Runge-Kutta Method by solving the Eq.(5.30) for x and y respectively in the interval $(0, 100)$ with the step size $h = 1/16$. In Fig.(5.18) first figure displays the global error of the Magnus Series Method for x in the Eq.(5.30) while second displays for y in the same equation. It is evident in these figures that Magnus Series Method has less error than Runge-Kutta Method for x in the Eq.(5.30), however, the global error of both methods for y in the Eq.(5.30) are approximately the same.

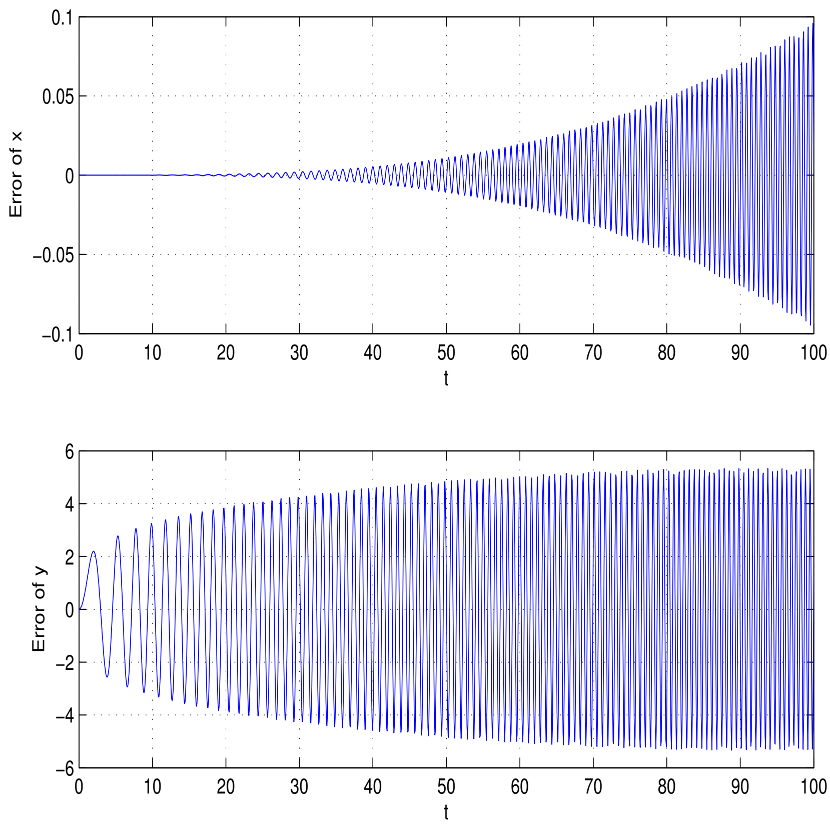

Figure 5.17: Global Error of the Runge-Kutta Method for x and y in the Eq.(5.29) respectively in the interval (0,100) with the time step h=1/16.

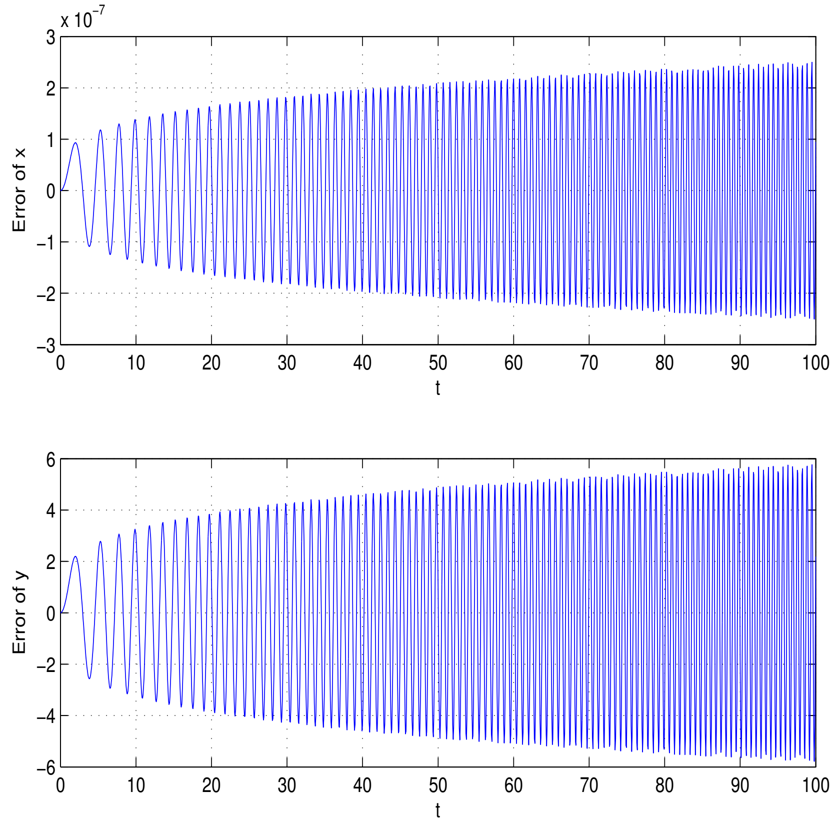

Figure 5.18: Global Error of the Magnus Series Method for x and y in the Eq.(5.29) respectively in the interval (0,100) with the time step h=1/16.

CHAPTER 6

CONCLUSION

6.1. Summary

In this thesis, we present numerical solutions of the several types of ordinary differential equations including linear, nonlinear initial value problem, hamiltonian system, based on the magnus series methods, one of the geometric integrator. We also extend the method to solve boundary value problem and second order system of ordinary differential equations. All the differential equations we considered exhibits the highly oscillatory behaviours in their solutions.

Starting with the Magnus integrators for linear systems, we have considered three cases by choosing the coefficient of y in Eq.(2.2) as a polynomial functions.

First, we choose $a(t)$ as a constant $a(t) = 1$ and applied both Magnus Series and Runge-Kutta Methods. After computation, we observed that Magnus Series Method works three times better than the Runge-Kutta Method.

Next, we choose $a(t) = t$, which is called Airy Equation. Both Magnus Series and Runge-Kutta Methods are applied again. At this time not only Magnus Series has less error than Runge-Kutta, but also we observed that Magnus resumes to give results for the long time integration while traditional method Runge-Kutta does not work after a fixed time.

The other polynomial function we considered is $a(t) = t^2$. Classical method Runge-Kutta does not work for this equation, although Magnus works. For that equation, we observed that the error accumulation of the Magnus Series Method increases when the degree of the polynomial function $a(t)$ increases.

Next, to illustrate the main features of Lie-type equation, we applied the method to the Duffing equation. The new algorithm, we developed, based on fourth and fifth order expansion of the magnus series is applied to solve duffing equation. After computation, we observed that the error accumulation is less than the one we obtained by Runge-Kutta method.

Finally, double Harmonic Oscillator problem as Hamiltonian System is considered. Since in future we would like to applly the method for molecular dynamic simulations. This problem can be thought as a simpler case for such computations. In order to show the methods is also applicable for the boundary value problems, we solve one particular example. We extend the method for the linear system of second order differential equation as well. All those computations we observed that the magnus series method has better performance than the classical Runge-Kutta Method based on the error plots for each cases.

As a result, all numerical experiments show that the validity and the effectiveness of Magnus Method by solving highly oscillatory differential equations when it is comparing the classical Runge-Kutta method. Since the geometric integrators led the solutions stay on the same manifold when the integration time evolves. We have written computer programs in MATLAB 6.5 to simulate the methods for all problems.

6.2. Future Work

Geometric integration is a new area for the numerical analysis and so Magnus Series Method. The method is applied to several types of equations in the literature and this remind us a question ”Which types of equations can be solved by this method?”. Computational cost is an important criteria in applied sciences and the researches on computational cost for Magnus Series Method still continue, then the other question that we have to ask is ”how can the method be cheaper computationally?”. Also, we do not still know why the Magnus Series Method can give great results for highly oscillatory differential equations. ”Higher dimensional molecular dynamics simulation by Magnus Series Method” and ”solution of nonhomogeneous Lie-type equations by Magnus Series Method” are the topics intended to research.

REFERENCES

Atkinson, K.E.,1988. *"An Introduction to Nomerical Analysis, 2nd Edition"*, John Wiley, New York, pp.418-424.

Bender and Orszag, 1999. *"Advanced Mathematical Methods for Scientists and Engineers"*, Springer-Verlag, New York, pp.546

Celledoni et al., 2002. *"Complexity Theory for Lie-Group Solvers"*, Journal of Complexity 18, pp.242-286.

Diele and Ragni, 2002. *"Numerical Comparison between Different Lie-Group Methods for Solving Linear Oscillatory ODEs"*, Springer-Verlag, ICCS2002, LNCS2331, pp.476-485.

Hsiang, W.Y., 2000. *"Lecture on Lie Groups/Lie Groups and Lie Algebras"*, Series on University Mathematic-vol2, World Scientific.

Ibragimov, N.H.,1999. *"Elementary Lie Group Analysis and Ordinary Differential Equations"*, John Wiley, New York, pp.100

Iserles, A., 2002. *"Think globally, act locally: Solving highly-oscillatory ordinary differential equations"*, Applied Numerical Mathematics 43, pp.145-160.

Iserles and Norsett, 1997a. *"Linear ODEs in Lie Goups"*, Numerical Analysis Reports, University of Cambridge, DAMTP 1997/NA9.

Iserles and Norsett, 1997b. *"On the Solutionof Linear Differential Equations in Lie Groups"*, Numerical Analysis Reports, University of Cambridge, DAMTP 1997/NA3.

Iserles et al., 2000. *"Lie-group Methods"*,(Cambridge University Press),Acta Numerica(2000), pp.215-365.

Iserles et al., 1999. *"On the Implementation of the Method of Magnus Series for Linear Differential Equations"*, BIT, vol.39, No.2, pp.281-304.

Iserles et al., 1998. *"Time-Symmetry and Higher-Order Magnus Methods"*, Numerical Analysis Reports, University of Cambridge, DAMTP 1998/NA06.

McLachlan et al., 2006. *"Geometric Integrators for ODEs"*, J.Physics A, 39:5251-5285.

Olver, P.J., 1993. *"Applications of Lie Groups to Differential Equations"*, Springer, pp.2-4.

Orel, B., 2001. *"Extrapolated Magnus Methods"*, BIT, Vol.41, No.5, pp.1089-1100.

Sanz-Serna and Calvo, 1994. *"Numerical Hamiltonian Problems"*, Chapman and Hall, pp.3-6.

Zanna, A., 1997. *"Lie-Group Methods for Isospectral Flows"*, Numerical Analysis Reports, University of Cambridge, DAMTP 1997/NA02.

WEB-1, 2005. Geometric Integration Preprint Server, 20/12/2005. *"Explicit Magnus Expansions for Nonlinear Equations"*, http://www.focm.net/gi/gips/.

WEB-2, 2006. Geometric Integration Preprint Server, 10/01/2006. *"Brief Introduction to Lie-Group Methods"*, http://www.focm.net/gi/gips/.

WEB-3, 2006. Geometric Integration Preprint Server, 10/01/2006. *"Numerical Analysis in Lie Groups"*, http://www.focm.net/gi/gips/.

WEB-4, 2005. The Institute for Systems Research (ISR) Technical Report, 12/06/2005. *"Introduction to Manifolds, Lie-Groups and Estimation on the Circle"*, http://techreports.isr.umd.edu/ARCHIVE.

WEB-5, 2006. CiteSeer, 04/05/2006. *"Geometric Integration Algorithms on Homogeneous Manifolds"*, http://citeseer.ist.psu.edu/lewis01geometric.html.

WEB-6, 2005. McLachlan, R., 14/08/2005. *"The Emerging Science of Geometric Integration"*, http://www.massey.ac.nz/ rmclachl/SciRevMcL.html.

APPENDIX A

MATLAB CODES FOR THE APPLICATIONS OF MAGNUS SERIES METHOD

MAGNUS SERIES METHOD-MATLAB CODE FOR LINEAR OSCILLATORY PROBLEM 1

```
%Magnus for y''+y=0,y(0)=1,y'(0)=0
Y=[1;0]; h=1/16; t=0:h:100; n=100/h+1; for i=1:n
    ye(i)=cos(t(i));
end for i=1:n
    y(i)=Y(1,1);
    k1=-t(i)+h*(1/2-(sqrt(3)/6));
    A1=[0 1;-1 0];
    k2=-t(i)+h*(1/2+(sqrt(3)/6));
    A2=[0 1;-1 0];
    b=A1+A2;
    c=A1*A2-A2*A1;
    d=A1-A2;
    e=d*c-c*d;
    sigma=((1/2)*h*b)-((sqrt(3)/12)*h^2*c)+((1/80)*h^3*e);
    Y=expm(sigma)*Y;
    em(i)=ye(i)-y(i);
end plot(t,em,'b');grid;xlabel('t');ylabel('y');
```

RUNGE-KUTTA METHOD-MATLAB CODE FOR LINEAR OSCILLATORY PROBLEM 1

```
%RK4 for y''+y=0,y(0)=1,y'(0)=0
Y=[1;0]; h=1/16; t=0:h:100; n=100/h+1; A=[0 1;-1 0]; for i=1:n
    ye(i)=cos(t(i));
end for i=1:n
```

```
    y(i)=Y(1,1);
    V1=A*Y;
    Y2=Y+((h/2)*V1);
    V2=A*Y2;
    Y3=Y+((h/2)*V2);
    V3=A*Y3;
    Y4=Y+(h*V3);
    V4=A*Y4;
    Y=Y+((h/6)*(V1+2*V2+2*V3+V4));
    er(i)=ye(i)-y(i);
end plot(t,er,'b');grid;xlabel('t');ylabel('y');
```

RUNGE-KUTTA METHOD-MATLAB CODE FOR LINEAR OSCILLATORY PROBLEM 1

```
%Magnus for the Airy Equation y''+ty=0,y(0)=1,y'(0)=0
Y=[1;0]; h=1/16; t=0:h:100; n=100/h+1; for i=1:n
    ye(i)=(airy(3,0)*airy(-t(i))-airy(1,0)*airy(2,-t(i)))/
    (airy(0)*airy(3,0)-airy(1,0)*airy(2,0));
end for i=1:n
    y(i)=Y(1,1);
    k1=-t(i)-h*(1/2-(sqrt(3)/6));
    A1=[0 1;k1 0];
    k2=-t(i)-h*(1/2+(sqrt(3)/6));
    A2=[0 1;k2 0];
    b=A1+A2;
    c=A1*A2-A2*A1;
    d=A1-A2;
    e=d*c-c*d;
    sigma=((1/2)*h*b)-((sqrt(3)/12)*h^2*c)+((1/80)*h^3*e);
    Y=expm(sigma)*Y;
    em(i)=ye(i)-y(i);
end plot(t,em,'b');grid;xlabel('t');ylabel('y');
%plot(t,y,'b');grid;xlabel('t');ylabel('y');
```

RUNGE-KUTTA METHOD-MATLAB CODE FOR LINEAR OSCILLATORY PROBLEM 2

```
%RK4 for the Airy Equation y’’+ty=0,y(0)=1,y’(0)=0
Y=[1;0]; h=1/16; t=0:h:100; n=100/h+1; for i=1:n
    ye(i)=(airy(3,0)*airy(-t(i))-airy(1,0)*airy(2,-t(i)))/
    (airy(0)*airy(3,0)-airy(1,0)*airy(2,0));
end for i=1:n
    y(i)=Y(1,1);
    A1=[0 1;-t(i) 0];
    V1=A1*Y;
    A2=[0 1;-t(i)-h/2 0];
    Y2=Y+(h/2)*V1;
    V2=A2*Y2;
    A3=[0 1;-t(i)-h/2 0];
    Y3=Y+(h/2)*V2;
    V3=A3*Y3;
    A4=[0 1;-t(i)-h 0];
    Y4=Y+h*V3;
    V4=A4*Y4;
    Y=Y+((h/6)*(V1+2*V2+2*V3+V4));
    er(i)=ye(i)-y(i);
end plot(t,er,’b’);grid;xlabel(’t’);ylabel(’y’);
%plot(t,y,’b’);grid;xlabel(’t’);ylabel(’y’);
```

MAGNUS SERIES METHOD-MATLAB CODE FOR LINEAR OSCILLATORY PROBLEM 3

```
%Magnus for the Equation y’’+t^2y=0,y(0)=0,y’(0)=1
Y=[0;1]; h=1/16; t=0:h:100; n=100/h+1; for i=1:n
    ye(i)=1/2*pi/gamma(3/4)*t(i)^(1/2)*besselj(1/4,1/2*t(i)^2);
end for i=1:n
    y(i)=Y(1,1);
    k1=t(i)+h*(1/2-(sqrt(3)/6));
```

```
    A1=[0 1;-k1^2 0];
    k2=t(i)+h*(1/2+(sqrt(3)/6));
    A2=[0 1;-k2^2 0];
    b=A1+A2;
    c=A1*A2-A2*A1;
    d=A1-A2;
    e=d*c-c*d;
    sigma=((1/2)*h*b)-((sqrt(3)/12)*h^2*c)+((1/80)*h^3*e);
    Y=expm(sigma)*Y;
    em(i)=ye(i)-y(i);
end
plot(t,em,'b');grid;xlabel('t');ylabel('y');
```

RUNGE-KUTTA METHOD-MATLAB CODE FOR LINEAR OSCILLATORY PROBLEM 3

```
%RK4 for the Equation y''+t^2y=0,y(0)=0,y'(0)=1
Y=[0;1]; h=1/16; t=0:h:100; n=100/h+1; for i=1:n
    ye(i)=1/2*pi/gamma(3/4)*t(i)^(1/2)*besselj(1/4,1/2*t(i)^2);
end for i=1:n
    y(i)=Y(1,1);
    A1=[0 1;-t(i)^2 0];
    V1=A1*Y;
    A2=[0 1;-(t(i)+h/2)^2 0];
    Y2=Y+(h/2)*V1;
    V2=A2*Y2;
    A3=[0 1;-(t(i)+h/2)^2 0];
    Y3=Y+(h/2)*V2;
    V3=A3*Y3;
    A4=[0 1;-(t(i)+h)^2 0];
    Y4=Y+h*V3;
    V4=A4*Y4;
    Y=Y+((h/6)*(V1+2*V2+2*V3+V4));
    er(i)=ye(i)-y(i);
```

```
end plot(t,er,'b');grid;xlabel('t');ylabel('y');
```

MAGNUS SERIES METHOD-MATLAB CODE FOR NONLINEAR OSCILLATORY PROBLEM

```
%Magnus for the Duffing Equation y''+y+ep*y^3=0,y(0)=1,y'(0)=0
Y=[1;0]; h=1/16; t=0:h:1000; n=1000/h+1; ep=0.00001; for i=1:n
    ye(i)=cos(t(i))+ep*(cos(3*t(i))/32-
    cos(t(i))/32-(3*t(i)*sin(t(i)))/8);
end for i=1:n
    y(i)=Y(1,1);
    a0=[0 1;-(1+ep*y(i)^2) 0];
    k0=h*a0;
    k1=(h/2)*a0;
    b1=expm(k0)*Y;
    a1=[0 1;-(1+ep*b1(1,1)^2) 0];
    m0=(h/2)*(a0+a1);
    b2=expm(k1)*Y;
    a2=[0 1;-(1+ep*b2(1,1)^2) 0];
    m1=(h/4)*(a0+a2);
    b3=expm(m1)*Y;
    a3=[0 1;-(1+ep*b3(1,1)^2) 0];
    b4=expm(m0)*Y;
    a4=[0 1;-(1+ep*b4(1,1)^2) 0];
    n0=(h/6)*(a0+4*a3+a4)-(h/3)*(m1*a3-a3*m1)-
    (h/12)*(m0*a4-a4*m0);
    k2=(h/4)*a0;
    b5=expm(k2)*Y;
    a6=[0 1;-(1+ep*b5(1,1)^2) 0];
    m2=(h/8)*(a0+a6);
    b6=expm(m2)*Y;
    a5=[0 1;-(1+ep*b6(1,1)^2) 0];
    n1=(h/12)*(a0+4*a5+a3)-(h/6)*(m2*a5-a5*m2)-
    (h/24)*(m1*a3-a3*m1);
```

```
    b7=expm(n1)*Y;
    a7=[0 1;-(1+ep*b7(1,1)^2) 0];
    b8=expm(n0)*Y;
    a8=[0 1;-(1+ep*b8(1,1)^2) 0];
    c1=n1*a7-a7*n1;
    c2=n0*a8-a8*n0;
    fi4=a0*(h/6)+4*(h/6)*a7-(h/3)*c1+(h/18)*
    (n1*c1-c1*n1)+(h/6)*a8-
    (h/12)*c2+(h/72)*(n0*c2-c2*n0);
    Y=expm(fi4)*Y;
    em(i)=ye(i)-y(i);
end plot(t,em,'b');grid;xlabel('t');ylabel('y');
%plot(t,y,'b');grid;xlabel('t');ylabel('y');
```

RUNGE-KUTTA METHOD-MATLAB CODE FOR NONLINEAR OSCILLATORY PROBLEM

```
%RK4 for the Duffing Equation y''+y+ep*y^3=0,y(0)=1,y'(0)=0
Y=[1;0]; h=1/16; t=0:h:1000; n=1000/h+1; ep=0.00001; for i=1:n
    ye(i)=cos(t(i))+ep*(cos(3*t(i))/32-
    cos(t(i))/32-(3*t(i)*sin(t(i)))/8);
end for i=1:n
    y(i)=Y(1,1);
    A=[0 1;-(1+ep*y(i)^2) 0];
    V1=A*Y;
    Y2=Y+(h/2)*V1;
    V2=A*Y2;
    Y3=Y+(h/2)*V2;
    V3=A*Y3;
    Y4=Y+h*V3;
    V4=A*Y4;
    Y=Y+((h/6)*(V1+2*V2+2*V3+V4));
    er(i)=ye(i)-y(i);
end plot(t,er,'b');grid;xlabel('t');ylabel('y');
```

```
%plot(t,y,'b');grid;xlabel('t');ylabel('y');
```

MAGNUS SERIES METHOD-MATLAB CODE FOR HAMILTONIAN SYSTEM

```
%Magnus for Double Harmonic Osillator (hamioltanian system)
Y=[1;1;0;0]; h=1/16; n=100/h+1;
for i=1:n
    p1(i)=Y(1,1);
    q1(i)=Y(3,1);
    p2(i)=Y(2,1);
    q2(i)=Y(4,1);
    A1=[0 0 -1 0;0 0 0 -1;1 0 0 0;0 1 0 0];
    A2=[0 0 -1 0;0 0 0 -1;1 0 0 0;0 1 0 0];
    b=A1+A2;
    c=A1*A2-A2*A1;
    d=A1-A2;
    e=d*c-c*d;
    sigma=((1/2)*h*b)-((sqrt(3)/12)*h^2*c)+((1/80)*h^3*e);
    Y=expm(sigma)*Y;
end
subplot(211); plot(q1,p1,'b');grid;xlabel('q1');ylabel('p1');
subplot(212); plot(q2,p2,'b');grid;xlabel('q2');ylabel('p2');
```

MAGNUS SERIES METHOD-MATLAB CODE FOR BOUNDARY VALUE PROBLEM

```
%Magnus Method for the Boundary Value Problems
(linear Oscillator)
h=1/16; t=0:h:100; n=100/h+1; beta=90; betax=-0.4481; U=[1;0];
V=[0;1]; for j=1:n
    ye(j)=cos(t(j));
end for i=1:n
    u(i)=U(1,1);
    if t(i)==beta
```

```
        ku=u(i);
    end
    A1U=[0 1;-1 0];
    A2U=[0 1;-1 0];
    b=A1U+A2U;
    c=A1U*A2U-A2U*A1U;
    d=A1U-A2U;
    e=d*c-c*d;
    sigma=((1/2)*h*b)-((sqrt(3)/12)*h^2*c)+((1/80)*h^3*e);
    U=expm(sigma)*U;
end for i=1:n
    v(i)=V(1,1);
    if t(i)==beta
        kv=v(i);
    end
    A1V=[0 1;-1 0];
    A2V=[0 1;-1 0];
    b=A1V+A2V;
    c=A1V*A2V-A2V*A1V;
    d=A1V-A2V;
    e=d*c-c*d;
    sigma=((1/2)*h*b)-((sqrt(3)/12)*h^2*c)+((1/80)*h^3*e);
    V=expm(sigma)*V;
end
c1=(betax-ku)/kv;
for i=1:n
    y(i)=u(i)+c1*v(i);
    em(i)=ye(i)-y(i);
end
subplot(211) plot(t,y,'b');grid;xlabel('t');ylabel('x-Magnus');
subplot(212);plot(t,em,'r');grid;xlabel('t');ylabel('Error');
```

RUNGE-KUTTA METHOD-MATLAB CODE FOR BOUNDARY VALUE PROBLEM

```
%RK for the Boundary Value Problems(linear Oscillator)
h=1/16; t=0:h:100; n=100/h+1; beta=90; betax=-0.4481; U=[1;0];
V=[0;1]; for j=1:n
    ye(j)=cos(t(j));
end AU=[0 1;-1 0]; AV=[0 1;-1 0]; for i=1:n
    u(i)=U(1,1);
    if t(i)==beta
        ku=u(i);
    end
    V1U=AU*U;
    U2=U+((h/2)*V1U);
    V2U=AU*U2;
    U3=U+((h/2)*V2U);
    V3U=AU*U3;
    U4=U+(h*V3U);
    V4U=AU*U4;
    U=U+((h/6)*(V1U+2*V2U+2*V3U+V4U));
end for i=1:n
    v(i)=V(1,1);
    if t(i)==beta
        kv=v(i);
    end
    V1V=AV*V;
    V2=V+((h/2)*V1V);
    V2V=AV*V2;
    V3=V+((h/2)*V2V);
    V3V=AV*V3;
    V4=V+(h*V3V);
    V4V=AV*V4;
    V=V+((h/6)*(V1V+2*V2V+2*V3V+V4V));
```

```
end
c1=(betax-ku)/kv;
for i=1:n
    y(i)=u(i)+c1*v(i);
    em(i)=ye(i)-y(i);
end
subplot(211) plot(t,y,'b');grid;xlabel('t');ylabel('x-RK4');
subplot(212); plot(t,em,'r');grid;xlabel('t');ylabel('Error');
```

MAGNUS SERIES METHOD-MATLAB CODE FOR SECOND ORDER SYSTEM

```
%Magnus for the Second Order System
M=[1;0;0;0]; h=1/16; t=0:h:200; n=200/h+1; for i=1:n
    xe(i)=(airy(3,0)*airy(-t(i))-airy(1,0)*airy(2,-t(i)))/
    (airy(0)*airy(3,0)-airy(1,0)*airy(2,0));
end for i=1:n
    ye(i)=(airy(3,0)*airy(1,-t(i))-airy(1,0)*airy(3,-t(i)))/
    (airy(0)*airy(3,0)-airy(1,0)*airy(2,0));
end for i=1:n
    x(i)=M(1,1);
    y(i)=M(3,1);
    k1=t(i)+h*(1/2-(sqrt(3)/6));
    A1=[0 1 0 0;-k1 0 0 0;0 0 0 1;-1 0 -k1 0];
    k2=t(i)+h*(1/2+(sqrt(3)/6));
    A2=[0 1 0 0;-k2 0 0 0;0 0 0 1;-1 0 -k2 0];
    b=A1+A2;
    c=A1*A2-A2*A1;
    d=A1-A2;
    e=d*c-c*d;
    sigma=((1/2)*h*b)-((sqrt(3)/12)*h^2*c)+((1/80)*h^3*e);
    M=expm(sigma)*M;
    ex(i)=xe(i)-x(i);
    ey(i)=ye(i)-y(i);
```

```
end subplot(211); plot(t,ex,'b');grid;xlabel('t');subplot(212);
plot(t,ey,'b');grid;xlabel('t');ylabel('Error of y');
```

RUNGE-KUTTA METHOD-MATLAB CODE FOR SECOND ORDER SYSTEM

```
%RK4 for the Second Order System
M=[1;0;0;0]; h=1/16; t=0:h:100; n=100/h+1; for i=1:n
    xe(i)=(airy(3,0)*airy(-t(i))-airy(1,0)*airy(2,-t(i)))/
    (airy(0)*airy(3,0)-airy(1,0)*airy(2,0));
end for i=1:n
    ye(i)=(airy(3,0)*airy(1,-t(i))-airy(1,0)*airy(3,-t(i)))/
    (airy(0)*airy(3,0)-airy(1,0)*airy(2,0));
end for i=1:n
    x(i)=M(1,1);
    y(i)=M(3,1);
    A1=[0 1 0 0;-t(i) 0 0 0;0 0 0 1;-1 0 -t(i) 0];
    V1=A1*M;
    A2=[0 1 0 0;-t(i)-h/2 0 0 0;0 0 0 1;-1 0 -t(i)-h/2 0];
    M2=M+(h/2)*V1;
    V2=A2*M2;
    A3=[0 1 0 0;-t(i)-h/2 0 0 0;0 0 0 1;-1 0 -t(i)-h/2 0];
    M3=M+(h/2)*V2;
    V3=A3*M3;
    A4=[0 1 0 0;-t(i)-h 0 0 0;0 0 0 1;-1 0 -t(i)-h 0];
    M4=M+h*V3;
    V4=A4*M4;
    M=M+((h/6)*(V1+2*V2+2*V3+V4));
    ex(i)=xe(i)-x(i);
    ey(i)=ye(i)-y(i);end subplot(211); plot(t,ex,'b');grid;
xlabel('t');ylabel('Error of x');subplot(212);
plot(t,ey,'b');grid;xlabel('t');ylabel('Error of y');
```