FINAL PROJECT THESIS

MASTER OF ADVANCED MATHEMATICS

FACULTAT DE MATEMÀTIQUES

UNIVERSITAT DE BARCELONA

On time Delay Differential Equations

---

by

JOAN GIMENO I ALQUÉZAR

Advisor: Àngel Jorba i Monte
Department: Matemàtica aplicada i anàlisi
June 28, 2015.

2010 *Mathematics Subject Classification.* Primary 34K09, 34K28, 34K13; Secondary 65P99, 68N19, 65G99

Key words and phrases. Delay Differential Equations, Floquet Theory, Automatic Differentiation, Integration methods

Abstract. The current final project belongs to a subject in the master's degree in Advanced Mathematics at the University of Barcelona. It deals with time delays which usually are arisen in differential equations.
Firstly, the project develops the main important known results of Delay Differential Equations, which are a specific case of Functional Differential Equations. In particular, we shall also focus on Delay Differential Equation with a constant delay.
Secondly, different integrators of a general Delay Differential Equation with a constant delay are explained and their numerical results are exposed according to the made implementation scripts written in C and C++. By the way, an introduction to Automatic Differentiation Theory is also presented in order to be able to compute derivatives until a prefixed order of some suitable functions.
Finally, a new method of computing periodic delayed orbits of a Delay Differential Equation with a constant delay is posed and a test of that new method is explained with some comments of the results.

acknowledgements. Àngel Jorba by his supervision in whole this project and Carles Simó by his explanations in the master's subject, Simulation Methods.

Contents

Nomenclature

(a_n)	Sequence	6
$\approx$	Relation of "approximately equals to"	37
$\exists\, P(x);\, Q$	There exists x verifying a property P such that Q holds	
$\forall\, P(x),\, Q$	For all x verifying a property P, then Q holds	
$\lceil\cdot\rceil$	Ceiling map	58
$\leftarrow$	Relation of "Assignment to"	51
$\leftrightarrow$	Swap of pointers	61
$\mathbb{IR}$	Set of real intervals.	43
$\mathbb{R}(m,n)$	Vector space of matrices of m rows and n columns over $\mathbb{R}$	51
$\mathcal{B}(X,Y)$	Susbset of the bounded and continuous maps from X to Y	5
$\mathcal{L}(X,Y)$	Set of linear and continuous mappings	9
$\mathbb{1}_A$	Characteristic function on the set A.	42
$\rightarrow$	Logic implication	
$\mathcal{C}(X,Y)$	Set of continuous maps from X to Y	3
$\overline{A}$	Closure of a set in a topology	5
$\restriction$	Partial restriction relation	5
$\varphi^{[j]}$	Normalized j-th derivative, i.e. $\frac{\varphi^{(j)}}{j!}$	
O	Big "O". Complexity notation	
o	Small "o"	31
$R[x]_{\leq n}$	n-th degree truncated polynomial over R	27
$X \subset Y$	X is contained or equal to Y, i.e. X is a subset of Y	

Introduction

The current project deals with delays that may be arisen in the time of a dynamical system. Firstly, we are going to focus on an abstract definition using Functional Analysis tools. That first stage is contained in Chapters **I** and **II**. Essentially, time Delay Differential Equations (DDE) are studied and some closed results to the Ordinary Differential Equation (ODE) Theory are also proved. For instance, the existence and uniqueness of an Initial Value Problem (IVP) of a DDE, its continuity and differentiability with respect to the initial condition, ...
The time Delay occupies a place of central importance in all areas of science such as biological sciences (e.g. population dynamics), celestial mechanics (e.g. relativistic N-body problem), ... and, in general, in any system which the delays (also called lags) have effects in the dynamical system. It must be said that if the delays are small, their effect may be omitted whereas they are not small, they may be had an important role in the dynamic of the system.
Specific Delay Differential Equations are stood out by to be a first approach of that generalization of ODE Theory. They have a formal expression:

$$\dot{x}(t) = f(t, x(t), x(t-\tau)), \qquad \tau \geq 0.$$

Thus Delay Differential Equations with a constant delay τ differ from Ordinary Differential Equations in that the derivative at any time depends on the solution at prior times.

The second stage of the thesis is to study how a Delay Differential Equation with a constant delay may be integrated it using similar methods that one can found in ODE Theory. For instance, Runge-Kutta methods, Runge-Kutta-Fehlberg methods and Taylor method. These have been developed in Chapter **IV**. A fast search of the state of art tells us that the former family of methods are the most common implemented in available software like dde-biftool[1], dde_solver[2]. However, almost every one is implemented in either FORTRAN or MATLAB.
The integration methods implemented for us have been:

- The Delayed Runge-Kutta 4.
- The Delayed Runge-Kutta-Fehlberg 78.
- The Delayed Taylor with different control steps.

All of them have been written in C. The strategy followed for testing them has been to consider tests with a known solution and then their errors have been plotted in different Figures.

The Chapter **III** contains modern results of Automatic Differentiation. They are able to compute all the derivatives easily and quickly up to a prefixed order whatever initial derivable function being admits a decomposition in elementary maps. It allows us to consider interpolation of Taylor expansions. However, that kind of interpolation should be treated with caution because error propagations may be arisen.

Finally, a new method periodic orbit computation of a Delay Differential Equation can be found in the last Chapter **V**. The current methods used so as to compute periodic orbits of a DDE are called collocation methods (see [**14**]) and they are inspired by collocation methods in an ODE. Even so, the original method, which we pose, is also inspired by the computation of periodic orbits of an Ordinary Differential Equation, i.e. the Poincaré map. Hence, the Delayed Poincaré map is defined and the Newton's method is used in order to find a fixed point of that map. In particular, the Automatic Differentiation and the Delayed Runge-Kutta 4, developed

[1] See http://www.cs.kuleuven.ac.be/cwis/research/twr/research/software/delay/ddebiftool.shtml.
[2] See http://www.radford.edu/~thompson/ffddes/index.html.

respectively in the Chapters **III** and **IV**, has been used in an implementation written in C++.

According to the preceding explanation we shall assume that basic (and maybe some fancy) C and C++ programming language is well-known by the reader. Other elementary results of Functional Analysis like Banach spaces, Differentiability on Banach spaces, ... shall also be well-known by the reader. Although some Appendices have been written in order to do use them in some theoretical parts of the Chapters **I** and **II**.

The last issue that should explicitly be commented is the references used in the development of the current final project. In the theoretical part, Chapters **I** and **II**, the main references have been [**8**] and [**11**]. [**7**] for Chapter **III**. The notes of the master's subject "Simulation methods" (2014-2015 course) and [**1**] for Chapter **IV**. And, finally, any reference has been used in the last Chapter **V**.

CHAPTER I

Delay Differential Equations

In many applications, one assumes that the future state of a process does not depend on the past states and is determined by the present. If one suppose that the system is governed by an equation which involves the state and the rate of change of the state, it is usually to consider either ordinary or partial differential equations. However, in other situations that assumptions becomes apparent a first approximation to the true situation and that a more realistic model would include some of the past states of the system. Also, in some cases where one does not imagine a dependence on the past.
This chapter deals with the notion of Functional Differential Equations (FDE), more particularly, the notion of a Delay Differential Equation (DDE). We introduce the basic results of that Theory like existence, uniqueness, continuation, continuous dependence for retarded equations, ...
The main references used in this Chapter have been [**10**], [**9**] and [**11**].

DEFINITION I.1. Let $t_0 \in \mathbb{R}$, $a \geq 0$ and $r \geq 0$. If

$$x \in \mathcal{C}([t_0 - r, t_0 + a], \mathbb{R}^n),$$

then for any $t \in [t_0, t_0 + a]$, we define $C := \mathcal{C}([0, r], \mathbb{R}^n)$ and $x_t \in C$ by

$$x_t(\tau) = x(t - \tau).$$

Now, let Ω be a subset of $\mathbb{R} \times C$ and $f \colon \Omega \to \mathbb{R}^n$ a function. A *Delay Differential Equation on* Ω is the relation

$$\dot{x} = f(t, x_t) \tag{1.1}$$

where $\dot{x}$ only represents the right-hand derivative of x.

Remark I.2. One can also consider the m-dimensional case putting $\Omega \subset \mathbb{R} \times C^m$. Then

$$\dot{x} = f(t, x_t(\tau_1), \ldots, x_t(\tau_m)).$$

For simplicity on the notations, we shall use $m = 1$.

DEFINITION I.3. A map x is a solution of (1.1) on $[t_0 - r, t_0 + a)$ when there are $t_0 \in \mathbb{R}$ and $a > 0$ such that

i. $x \in \mathcal{C}([t_0 - r, t_0 + a), \mathbb{R}^n)$.
ii. $(t, x_t) \in \Omega$.
iii. For any $t \in [t_0, t_0 + a)$,

$$\dot{x}(t) = f(t, x_t).$$

DEFINITION I.4. If $t_0 \in \mathbb{R}$ and $u \in C$, then $x(t_0, u)$ is a solution of (1.1) with initial condition u at t_0 when there is $a > 0$ such that

i. $x(t_0, u)$ is a solution of (1.1) on $[t_0 - r, t_0 + a)$.
ii. $x_{t_0}(t_0, u) \equiv u$.

Sometimes, one simply says that $x(t_0, u)$ is a solution through (t_0, u).

As in the ODE case, we have in DDE an integral equation:

LEMMA I.5. *Let $(t_0, u) \in \Omega \subset \mathbb{R} \times C$ and $f\colon \Omega \to \mathbb{R}^n$ continuous. Finding a solution of*

$$\begin{cases} \dot{x} = f(t, x_t) \\ x_{t_0} \equiv u. \end{cases}$$

is equivalent to solving the integral equation

$$\begin{aligned} x(t) &= u(0) + \int_{t_0}^{t} f(s, x_s)\, ds, \qquad t \geq t_0 \\ x_{t_0} &\equiv u. \end{aligned}$$

PROOF. Let us prove the two implications.

$\Rightarrow$) Let $x(t_0, u)$ be a solution of the initial value problem, that is,

$$\begin{cases} \dfrac{\partial x(t_0, u)}{\partial t}(t) = f(t, x_t(t_0, u)) \\ x_{t_0}(t_0, u) \equiv u. \end{cases}$$

Then by the Fundamental Theorem of calculus,

$$\begin{aligned} x(t_0, u)(t) - u(0) &= x(t_0, u)(t) - x_{t_0}(t_0, u)(0) = x(t_0, u)(t) - x(t_0, u)(t_0) \\ &= \int_{t_0}^{t} x'(t_0, u)(s)\, ds = \int_{t_0}^{t} f(s, x_s(t_0, u))\, ds. \end{aligned}$$

$\Leftarrow$) Again, by the Fundamental Theorem of calculus,

$$x'(t) = \frac{d}{dt}\left(u(0) + \int_{t_0}^{t} f(s, x_s)\, ds \right) = f(t, x_t(t_0, u)). \qquad \square$$

Let us show that any Initial Value Problem can be modified so that the initial time is t_0.

NOTATION. Given $(t_0, u) \in \mathbb{R} \times C$, let $\widetilde{u} \in \mathcal{C}([t_0 - r, +\infty), \mathbb{R}^n)$ be defined by

$$\begin{aligned} \widetilde{u}_{t_0} &\equiv u \\ \widetilde{u}(t_0 + t) &= u(0) \qquad \forall\, t \geq 0. \end{aligned}$$

LEMMA I.6. *Let $(t_0, u) \in \Omega \subset \mathbb{R} \times C$ and $f\colon \Omega \to \mathbb{R}^n$ be continuous.*

$$\begin{cases} \dot{x} = f(t, x_t) \\ x_{t_0} \equiv u. \end{cases} \qquad \textit{and} \qquad \begin{cases} \dot{y} = f(t_0 + t, \widetilde{u}_{t_0 + t} + y_t) \\ y_0 \equiv 0. \end{cases}$$

have the same solutions. Equivalently,

$$\begin{aligned} x(t) &= u(0) + \int_{t_0}^{t} f(s, x_s)\, ds, \quad t \geq t_0 \\ x_{t_0} &\equiv u \end{aligned} \qquad \begin{aligned} y(t) &= \int_{0}^{t} f(t_0 + s, \widetilde{u}_{t_0 + s} + y_s)\, ds, \quad t \geq 0 \\ y_0 &\equiv 0. \end{aligned}$$

are equivalent.

PROOF. By Lemma **I.5**, if $x(t)$ is a solution for the initial conditions (t_0, u), then

$$\begin{aligned} x(t) &= u(0) + \int_{t_0}^{t} f(s, x_s)\, ds, \qquad t \geq t_0 \\ x_{t_0} &\equiv u. \end{aligned}$$

Let us consider the change $y(t) = x(t_0 + t) - \widetilde{u}(t_0 + t)$ for $t \geq -r$.
If $t \geq 0$, then

$$x(t_0 + t) - \widetilde{u}(t_0 + t) = x(t_0 + t) - u(0) = \int_{t_0}^{t_0 + t} f(s, x_s)\, ds = \int_{0}^{t} f(t_0 + s, x_{t_0 + s})\, ds$$

If $-r \leq t \leq 0$, then

$$y_0(-t) = y(t) = x(t_0+t) - \widetilde{u}(t_0+t) = x_{t_0}(-t) - \widetilde{u}_{t_0}(-t) = u(-t) - u(-t) = 0.$$

Therefore, $y(t)$ solves the integral equation

$$\begin{aligned} y(t) &= \int_0^t f(t_0+s, \widetilde{u}_{t_0+s} + y_s)\, ds \qquad \forall\, t \geq 0, \\ y_0 &\equiv 0. \end{aligned}$$

By Lemma **I.5**, the result follows. □

1. Existence and uniqueness

As happens in Ordinary Differential Equation, there are Theorems for the uniqueness and existence of an Initial Value Problem of a Delay Differential Equation. In fact, if in the Definition **I.1** the $r = 0$, a Delay Differential Equation is exactly an Ordinary Differential Equation.

LEMMA I.7. *If $x \in \mathcal{C}([t_0 - r, t_0 + a], \mathbb{R}^n)$, then x_t is a continuous map of t for $t \in [t_0, t_0 + a]$.*

PROOF. Since x is continuous, it is uniformly continuous on $I = [t_0 - r, t_0 + a]$, so

$$\forall\, \varepsilon > 0, \exists\, \delta > 0;\ \forall\, t, s \in I,\ |t - s| < \delta \Rightarrow |x(t) - x(s)| < \varepsilon.$$

For $t \in [t_0, t_0 + a]$ and $|t - s| < \delta$,

$$|x_t(\tau) - x_s(\tau)| = |x(t - \tau) - x(s - \tau)| < \varepsilon.$$

for all $\tau \in [0, r]$. □

NOTATION. Let a, b, r be positive real numbers,

$$\begin{aligned} \overline{I}_a &= [0, a], \\ \overline{B}_b &= \{v \in C\colon |v| \leq b\}, \\ A(a, b) &= \{v \in \mathcal{C}([-r, a], \mathbb{R}^n)\colon v_0 \equiv 0,\ v_t \in \overline{B}_b,\ t \in \overline{I}_a\}. \end{aligned}$$

LEMMA I.8. *Let $\Omega \subset \mathbb{R} \times C$ be open, $K \subset \Omega$ compact and $f\colon \Omega \to \mathbb{R}^n$ continuous. There are*

i. *$V \subset \Omega$ neighbourhood of K such that $f \restriction V \in \mathcal{B}(V, \mathbb{R}^n)$.*
ii. *$U \subset \mathcal{B}(V, \mathbb{R}^n)$ neighbourhood of f and constants $M, a, b > 0$ such that*

$$|g(t, v)| < M \qquad \forall\, (t, v) \in V \text{ and } \forall\, g \in U. \tag{1.2}$$

Moreover, for each $(t_0, u) \in K$,

$$(t_0 + t, \widetilde{u}_{t_0+t} + v_t) \in V \qquad \forall\, t \in \overline{I}_a \text{ and } \forall\, v \in A(a, b).$$

PROOF. Since f is continuous and K is compact, there is $M > 0$ such that

$$|f(t_0, u)| < M \qquad \forall\, (t_0, u) \in K.$$

Again, by compactness, there are α, β and ε positives such that,

$$|f(t_0 + t, u + v)| < M - \varepsilon \qquad \forall\, (t_0, u) \in K \text{ and } \forall\, (t, v) \in \overline{I}_\alpha \times \overline{B}_\beta.$$

Taking $V = \{(t_0 + t, u + v)\colon (t_0, u) \in K \text{ and } (t, v) \in \overline{I}_\alpha \times \overline{B}_\beta\}$, then $f \in \mathcal{B}(V, \mathbb{R}^n)$. And there is a neighbourhood $U \subset \mathcal{B}(V, \mathbb{R}^n)$ of f such that the condition (1.2) holds.
Now, since K is compact, we choose $a < \alpha$ and $0 < b < \beta$ such that

$$\|\widetilde{u}_{t_0+t} - u\| < \beta - b \qquad \forall\, (t_0, u) \in K \text{ and } \forall\, t \in \overline{I}_a.$$

So by the construction of V,

$$\|v_t + \widetilde{u}_{t_0+t} - u\| < b + \beta - b = \beta \qquad \forall\, v \in A(a, b).$$

□

LEMMA I.9. *Let $M, b > 0$ be reals. The set*

$$W = \{v \in \mathcal{C}(I, \mathbb{R}^n) \colon \|v\| \leq b \text{ and } |v(t) - v(s)| \leq M|t - s| \text{ for all } t, s \in I\}$$

is compact in $\mathcal{C}(I, \mathbb{R}^n)$ whatever compact subset I of $\mathbb{R}^m$.

PROOF. We shall apply the Corollary **A.6**.

- Closed. Let $(v^k) \subset W$ be a sequence such that $v^k \to v$. Then $v \in W$. Indeed,

$$\|v\| < \varepsilon + b \qquad \text{and} \qquad |v(t) - v(s)| < 2\varepsilon + M|t - s|.$$

  So $\|v\| \leq b$ and $|v(t) - v(s)| \leq M|t - s|$ as $\varepsilon \to 0$.
- Uniformly bounded. Immediate.
- Equicontinuous. It is straightforward, we must show that

$$\forall\, \varepsilon > 0,\ \exists\, \delta > 0;\ \forall\, v \in W \text{ and } \forall\, x, y \in I,\ |x - y| < \delta \Rightarrow |v(x) - v(y)| < \varepsilon.$$

  So given $\varepsilon > 0$, we take $\delta < \frac{\varepsilon}{M}$.

Therefore W is compact. It only remains to show the convexity, that means $(1 - \lambda)u + \lambda v \in W$. Indeed,

$$|(1 - \lambda)\big(u(t) - u(s)\big) + \lambda\big(v(t) - v(s)\big)| \leq (1 - \lambda)|u(t) - u(s)| + \lambda|v(t) - v(s)| \leq M|t - s|$$

and

$$|(1 - \lambda)u + \lambda v| \leq (1 - \lambda)|u| + \lambda|v| \leq (1 - \lambda)b + \lambda b = b. \qquad \square$$

LEMMA I.10. *Let $\Omega \subset \mathbb{R} \times C$ be open, $K \subset \Omega$ compact and $f \colon \Omega \to \mathbb{R}^n$ continuous. Given U, V neighbourhoods and positive constants M, a, b obtained by Lemma **I.8**. The map*

$$T \colon K \times U \times A(a, b) \to \mathcal{C}([-r, a], \mathbb{R}^n)$$

defined by

$$T(t_0, u, g, v)(t) = \begin{cases} 0 & t \in [-r, 0] \\ \displaystyle\int_0^t g(t_0 + s, \widetilde{u}_{t_0+s} + v_s)\, ds & t \in \overline{I}_a. \end{cases}$$

is continuous and there is a compact set W in $\mathcal{C}([-r, a], \mathbb{R}^n)$ such that

$$T \colon K \times U \times A(a, b) \to W.$$

Moreover, if $Ma \leq b$, then

$$T \colon K \times U \times A(a, b) \to A(a, b).$$

PROOF. A map $T \colon K \times U \times A(a, b) \to \mathcal{C}([-r, a], \mathbb{R}^n)$ is well defined in the sense that $T(t_0, u, g, v) \in \mathcal{C}([-r, a], \mathbb{R}^n)$. The condition (1.2) tells us that for each $t, s \in \overline{I}_a$,

$$|T(t_0, u, g, v)(t) - T(t_0, u, g, v)(s)| \leq M|t - s|$$
$$|T(t_0, u, g, v)(t)| \leq Ma.$$

Let us consider

$$W = \{v \in \mathcal{C}([-r, a], \mathbb{R}^n) \colon |v(t) - v(s)| \leq M|t - s| \text{ and } |v(t)| \leq Ma\}.$$

By Lemma **I.9**, it is compact. Thus, $T \colon K \times U \times A(a, b) \to W$.
If $Ma \leq b$, then $W \subset A(a, b)$ and $T \colon K \times U \times A(a, b) \to A(a, b)$. Indeed, $v \in A(a, b)$ if, and only if, for each $\tau \in [0, r]$ and for each $t \in [0, a]$,

$$v(\tau) = 0 \qquad \text{and} \qquad |v(t - \tau)| \leq b.$$

That is, $|v(t)| \leq b$ with $t \in [-r, a]$. This condition holds if $v \in K$ and $Ma \leq b$.
Finally, we must show the continuity of T. Let us consider $((t^k, u^k, g^k, v^k))_k$ a sequence on $K \times U \times A(a, b)$ such that

$$(t^k, u^k, g^k, v^k) \to (t_0, u, g, v) \qquad \text{as } k \to \infty.$$

with $(t_0, u, g, v) \in K \times U \times A(a,b)$. We know that

$$T(t^k, u^k, g^k, v^k) \in W$$

and since W is compact, there is a convergent subsequence that we will designate with the same index,

$$T(t^k, u^k, g^k, v^k) \to h \qquad \text{as } k \to \infty$$

now with $h \in W$. Since

$$g^k(t^k + s, \widetilde{u}^k_{t^k+s} + v^k_s) \to g(t_0 + s, \widetilde{u}_{t_0+s} + v_s) \qquad \text{as } k \to \infty$$

whenever $t \in \overline{I}_a$. Moreover, all g^k and g are uniformly bounded by Lemma **I.8**. Hence, by the Dominated Convergence Theorem, for all $t \in \overline{I}_a$,

$$h(t) = \lim_{k\to\infty} \int_0^t g^k(t^k + s, \widetilde{u}^k_{t^k+s} + v^k_s)\, ds = \int_0^t g(t_0 + s, \widetilde{u}_{t_0+s} + v_s)\, ds = T(t_0, u, g, v)(t).$$

We have proved that any convergent subsequence is not dependent of the subsequence. This implies the convergence of the sequence. Therefore T is continuous. □

LEMMA I.11. *$A(a,b)$ is closed, bounded and convex set on $\mathcal{C}([-r, a], \mathbb{R}^n)$.*

PROOF.

- Closed. Let $(v^k) \subset A(a,b)$ be a sequence so that $v^k \to v$ as $k \to \infty$. We must show $u \in A(a,b)$. Indeed
$$|v(\tau)| \le |v(\tau) - v^k(\tau)| + |v^k(\tau)| < \varepsilon$$
for all $0 \le \tau \le r$ and for all $\varepsilon > 0$. So $v_0 = 0$.
$$|v(t-\tau)| \le |v(t-\tau) - v^k(t-\tau)| + |v^k(t-\tau)| < \varepsilon + b$$
for all $0 \le t \le a$, for all $0 \le \tau \le r$ and for all $\varepsilon > 0$. So $v_t \in \overline{B}_b$ for any $t \in \overline{I}_a$.
- Uniformly bounded. If $v \in A(a,b)$,
$$\sup_{0\le\tau\le r} |v(t-\tau)| \le b \qquad \forall\, 0 \le t \le a.$$
Therefore, $A(a,b)$ is uniformly bounded.
- Convex. Clear. □

THEOREM I.12. *Let $\Omega \subset \mathbb{R} \times C$ be open set and $f\colon \Omega \to \mathbb{R}^n$ continuous. If $K \subset \Omega$ is compact, there are*

i. *$V \subset \Omega$ neighbourhood of K such that $f \restriction V \in \mathcal{B}(V, \mathbb{R}^n)$.*
ii. *$U \subset \mathcal{B}(V, \mathbb{R}^n)$ neighbourhood of $f \restriction V$.*
iii. *a positive real number.*

such that for any $(t_0, u) \in K$ and any $g \in U$, there is a solution $x(t; t_0, u, g)$ of

$$\begin{cases} \dot{x} = g(t, x_t) \\ x_{t_0} \equiv u \end{cases}$$

that exists on $[t_0 - r, t_0 + a]$.
Moreover, if $g(t, v)$ is Lipschitz in v in each compact subset in Ω, the solution is unique.

PROOF. Fixed $g \in U$, let us take $W = \{(t_0, u)\}$. Applying Lemma **I.10**, $T(t_0, u, g, \cdot)$ has a fixed point in the closed bounded and convex set $A(a,b)$ by Corollary **B.3**. By Lemmas **I.5** and **I.6**, we have a solution that exists on $[t_0 - r, t_0 + a]$.
Now, if x and y are solutions on $[t_0 - r, t_0 + a]$, by Lemma **I.5**,

$$x_{t_0} - y_{t_0} \equiv 0$$
$$x(t) - y(t) = \int_{t_0}^t \big(g(s, x_s) - g(s, y_s)\big)\, ds \qquad t \ge t_0.$$

If L is a Lipschitz constant of $g(t,v)$ in any compact subset in Ω containing the trajectories $\{(t,x_t)\}$ and $\{(t,y_t)\}$ with $t\in\overline{I}_a$, then we can choose α so that $0<(\alpha-t_0)L<1$. Thus, for each $t\in\overline{I}_\alpha$,

$$|x(t)-y(t)|\le L\int_{t_0}^{t}\|x_s-y_s\|\,ds\le(\alpha-t_0)L\sup_{t_0\le s\le t}|x_s-y_s|.$$

Let us observe that

$$\begin{aligned}\sup_{t_0\le s\le t}|x_s-y_s| &= \sup_{t_0\le s\le t}\sup_{0\le\tau\le r}|x(s-\tau)-y(s-\tau)|\\ &\le \sup_{t_0-r\le s\le t}|x(s)-y(s)|\le\sup_{t_0-r\le s\le\alpha}|x(s)-y(s)|.\end{aligned}$$

Therefore we have proved that the mapping

$$\begin{aligned}\mathcal{C}([t_0-r,t_0+a],\mathbb{R}^n)&\longrightarrow\mathcal{C}([t_0-r,t_0+a],\mathbb{R}^n)\\ x(t)&\longmapsto\begin{cases}u(t_0-t) & t\in[t_0-r,t_0]\\ u(0)+\displaystyle\int_{t_0}^{t}g(s,x_s)\,ds & t\in[t_0,t_0+a].\end{cases}\end{aligned}$$

is a contraction for $t\in\overline{I}_\alpha$. So $x(t)=y(t)$. □

In $\mathbb{R}^n$, we already know that a map is locally Lipschitz if, and only if, it is Lipschitz in each compact subset. However, it could be not true in a Banach space, basically, it is because a closed ball is not always a compact set. Therefore, we are going to prove one of the implication in Proposition **I.13**.
As a consequence, if a mapping is locally Lipschitz, it is Lipschitz for each compact subset. On the other hand, in $\mathbb{R}^n$ we already know that if a mapping is $\mathcal{C}^1$ in an open set Ω, it is locally Lipschitz in Ω. However, we shall see that a $\mathcal{C}^1$ mappping in an open subset of a real Banach space is Lipschitz on each compact subset of that open.
Summarizing, given an initial value problem of $\dot{x}=f(t,x_t)$. If f is $\mathcal{C}^1$ with respect to the second variable, a solution exists and it is unique.

PROPOSITION I.13. *Let $f\colon X\to Y$ be a locally Lipschitz mapping between real Banach spaces.*

If X is compact, then f is Lipschitz.

PROOF. There are open sets $U_1,\dots,U_n$ in X such that

$$X=U_1\cup\cdots\cup U_n$$

and $f\restriction U_i$ is Lipschitz with value L_i. Let $\delta>0$ be a Lebesgue number associated to this open covering of X. Therefore, for all $x,y\in X$,

$$\|x-y\|<\delta\Rightarrow x,y\in U_i$$

for some i. Let us define

$$M=\sup_{x\in X}\|f(x)\|\qquad\text{and}\qquad L=\max\left\{L_1,\dots,L_n,\frac{2M}{\delta}\right\}.$$

For each $x,y\in X$, there are two possible cases:

i. If $\|x-y\|<\delta$, then $\|f(x)-f(y)\|\le L_i\|x-y\|\le L\|x-y\|$.
ii. If $\|x-y\|\ge\delta$, then $\|f(x)-f(y)\|\le 2M=\dfrac{2M\delta}{\delta}\le L\|x-y\|$.

Thus, f is Lipschitz on X. □

PROPOSITION I.14. *Let Ω be an open set of a real Banach space.*

If $f\colon\Omega\to\mathbb{R}^n$ is $\mathcal{C}^1$, it is Lipschitz on each compact subset in Ω.

PROOF. Let $K \subset \Omega$ be a compact. Since f is $\mathcal{C}^1$ in Ω, then

$$Df\colon \Omega \to \mathcal{L}(\Omega, \mathbb{R}^n), \qquad x \mapsto Df(x)$$

is continuous. So $Df \restriction K$ has a bounded called L. By Proposition **I.13**, if $f \restriction K$ is locally Lipschitz, it will be Lipschitz.
Let $x_0 \in K$ and $\overline{B(x_0;\varepsilon)} \subset K$. Since each closed ball in a normed space is convex, given $x, y \in \overline{B(x_0;\varepsilon)}$, we define $u = x - y$ and

$$g\colon [0,1] \to \mathbb{R}^n, \qquad t \mapsto f(x + tu).$$

Clearly g is continuous and by the chain rule $g'(t) = Df(x + tu)u$. Thus, by the Fundamental Theorem of Calculus,

$$f(x) - f(y) = g(1) - g(0) = \int_0^1 g'(t)\,dt = \int_0^1 Df(x + tu)u\,dt$$

Therefore, $|f(x) - f(y)| \leq L\|x - y\|$. □

THEOREM I.15 (Globally uniqueness). *Let $\Omega \subset \mathbb{R}\times C$ be open, $(t_0, u) \in \Omega$, $f\colon \Omega \to \mathbb{R}^n$ continuous and Lipschitz in each compact subset with respect to the second variable. If $x\colon [t_0 - r, a] \to \mathbb{R}^n$ and $y\colon [t_0 - r, b] \to \mathbb{R}^n$ are solutions of*

$$\begin{cases} \dot{x} = f(t, x_t) \\ x_{t_0} \equiv u. \end{cases} \tag{1.3}$$

Then $x_t \equiv y_t$ for any $t \in [t_0, c]$ with $c = \min\{a, b\}$.

PROOF. Suppose that there is $t_0 < t_1$ so that $x_{t_1} \not\equiv y_{t_1}$. Let us define

$$t_* = \inf\{t \in [t_0, c]\colon x_t \not\equiv y_t\}.$$

Thus, $x_t \equiv y_t$ for any $t \in [t_0, t_*]$. Let $v \equiv x_{t_*} \equiv y_{t_*}$. By Theorem **I.12** at $(t_*, v) \in \Omega$, there is $z\colon I_* \to \mathbb{R}^n$ solution of the initial value problem (1.3). Contradiction with the election of t_*. □

2. Maximal solution

DEFINITION I.16. Let $\Omega \subset \mathbb{R} \times C$ be open, $f\colon \Omega \to \mathbb{R}^n$ continuous and x solution on $[t_0 - r, a)$ of

$$\begin{cases} \dot{x} = f(t, x_t) \\ x_{t_0} \equiv u. \end{cases}$$

x is a maximal solution when for any other solution y on $[t_0 - r, b)$ with $a < b$,

$$y \restriction [t_0, a) = x \Rightarrow a = b.$$

THEOREM I.17 (Existence and uniqueness of maximal solutions). *Let $\Omega \subset \mathbb{R} \times C$ be open set and $f\colon \Omega \to \mathbb{R}^n$ continuous and Lipschitz in each compact subset with respect to the second variable. If $(t_0, u) \in \Omega$, there is maximal solution $x\colon I(t_0, u) \to \mathbb{R}^n$ of*

$$\begin{cases} \dot{x} = f(t, x_t) \\ x_{t_0} \equiv u. \end{cases} \tag{1.4}$$

Moreover, $I(t_0, u) = [t_0 - r, a)$ with $t_0 < a$.

PROOF. Let $\mathcal{S}(t_0, u) = \{I_y \xrightarrow{y} \mathbb{R}^n\colon y \text{ is solution of } (1.4)\}$. Let us define

$$I(t_0, u) = \bigcup_{y \in \mathcal{S}(t_0,u)} I_y.$$

and $x\colon I(t_0, u) \to \mathbb{R}^n$ defined by

$$x(t) = y(t) \qquad \text{if } t \in I_y.$$

By Theorem **I.15**, x is well-defined because it does not depend on the solution y chosen.
Clearly, x is maximal.
If $I(t_0, u) = [t_0 - r, a]$, then $(a, x(a)) \in \Omega$ and there is a solution with initial condition $(a, x(a))$ which is an left extension of x. Contradiction with the maximality of x. □

3. Continuation of solutions

THEOREM I.18. *Let $\Omega \subset \mathbb{R} \times C$ be open set and $f\colon \Omega \to \mathbb{R}^n$ continuous. If x is a maximal solution on $[t_0 - r, a)$ of $\dot{x} = f(t, x_t)$, then*

$$\forall\, K \subset \Omega \text{ compact}, \exists\, t_K \in [t_0 - r, a);\ \forall\, t \in [t_K, a),\ (t, x_t) \notin K.$$

PROOF. First of all, if $a = +\infty$, the result is trivially true.
If $r = 0$, it corresponds to the case of an Ordinary Differential Equation.
If the conclusion is not true for $r > 0$, there are a sequence $t_k \to a^-$ as $k \to \infty$ and $v \in C$ such that

$$(t_k, x_{t_k}) \to (a, v) \qquad \text{as } k \to \infty$$

with $(t_k, x_{t_k}) \in W$. Thus, for any $\varepsilon > 0$,

$$\sup_{\tau \in [\varepsilon, r]} |x_{t_k}(\tau) - v(\tau)| \to 0 \qquad \text{as } k \to \infty.$$

So $x(a - \tau) = v(\tau)$ with $0 < \tau \leq r$. Hence, x can be extended continuously as follows:

$$\hat{x}(t) = \begin{cases} x(t) & t \in [t_0 - r, a) \\ v(0) & t = a. \end{cases}$$

Since now $(a, \hat{x}_a) \in \Omega$, one can find a solutions trough $(a, \hat{x}_a)$ to the right of a. However, it is a contraction with the maximality assumption of x. □

THEOREM I.19. *Let $\Omega \subset \mathbb{R} \times C$ open and $f\colon \Omega \to \mathbb{R}^n$ continuous verifying:*

$$K \subset \Omega \text{ closed and bounded set implies } f(K) \text{ is bounded set.}$$

If x is a maximal solution on $[t_0 - r, a)$ of $\dot{x} = f(t, x_t)$, then

$$\forall\, K \subset \Omega \text{ closed and bounded}, \exists\, t_K \in [t_0 - r, a);\ \forall\, t \in [t_K, a),\ (t, x_t) \notin K.$$

PROOF. First of all, if $a = +\infty$, the result is true.
If $r = 0$, it corresponds to the case of an Ordinary Differential Equation.
If $r > 0$ and the conclusion is not true, there is a sequence $(t_k) \subset \mathbb{R}$ such that

$$t_k \to a^- \qquad \text{as } k \to \infty$$

and $(t_k, x_{t_k}) \in K$. Since $r > 0$, then $\{x(t)\colon t \in [t_0 - r, b)\}$ is bounded. Therefore, there is $M > 0$ such that

$$|f(t, u)| \leq M \qquad \forall\, (t, u) \in \overline{\{(t, x_t)\colon t \in [t_0, a)\}}.$$

By Lemma **I.5**,

$$|x(t + b) - x(t)| \leq \int_t^{t+b} |f(s, x_s)|\, ds \leq Mb$$

for any t with $t + b < a$. Hence, x is uniformly continuous on $[t_0 - r, a)$. □

4. Continuity and differentiability of solutions

We want to obtain results about the continuity and differentiability of a solution of an initial value problem of a Delay Differential Equation. The first Theorem **I.20** is a little bit technical and we will skip the proof.

THEOREM I.20 (Continuity of initial conditions). *Let $\Omega \subset \mathbb{R} \times C$ open set and $f\colon \Omega \to \mathbb{R}^n$ continuous. The solution $x(t_0, u, f)$ of*

$$\begin{cases} \dot{x} = f(t, x_t) \\ x_{t_0} \equiv u \end{cases}$$

is continuous with respect to t_0, u and f.

THEOREM I.21 (Differentiability of initial conditions). *Let $\Omega \subset \mathbb{R} \times C$ open set and $f \in \mathcal{C}^p(\Omega, \mathbb{R}^n)$ with $p \geq 1$. The solution $x(t_0, u, f)$ of*

$$\begin{cases} \dot{x} = f(t, x_t) \\ x_{t_0} \equiv u \end{cases} \tag{1.5}$$

is unique and $\mathcal{C}^p$ with respect to u, f for t in any compact set in the domain of $x(t_0, u, f)$.

PROOF. By Theorem **I.12** and Proposition **I.14**, the solution of (1.5) is unique. Let the maximal interval of existence of $x(t_0, u, f)$ be $[t_0 - r, t_0 + \beta)$.
Firstly, we must show that $x(t_0, u, f)$ is $\mathcal{C}^1$ with respect to u on $[t_0 - r, t_0 + \alpha]$, of course, with $\alpha < \beta$. There is an open neighbourhood U of u such that $x(t_0, v, f)$ is defined for any $v \in U$ on $[t_0 - r, t_0 + \alpha]$. If

$$K = \{(t, x_t)\colon t \in [t_0, t_0 + \beta)\},$$

it is compact. By Lemma **I.8**, we obtain M, a, b, U and V. We choose a so that

$$Ma \leq b \qquad \text{and} \qquad 0 < La < 1. \tag{1.6}$$

with L a bound of the derivative of f with respect to u on Ω.
Let us consider the solution change used in Lemma **I.6**, that is

$$y(t) = x(t_0 + t) - \widetilde{u}(t_0 + t) \qquad t \in \overline{I}_a$$

and the map $T(t_0, u, f)$ defined in Lemma **I.10**. By Lemma **I.6**, $y(t)$ is a fixed point of $T(t_0, u, f)$. The restriction (1.6) on a and b implies that $T(t_0, u, f)$ takes $A(a, b)$ into itself for each a, b and it is a contraction.
Moreover, the contraction constant does not depend on $(t_0, u, f) \in V \times U$. Since $T(t_0, u, f)$ is $\mathcal{C}^p$ in Ω, by Theorem **C.3**, the fixed point $y(t_0, u, f)$ is $\mathcal{C}^p$ in Ω.
A very similar proof shows that $x(t_0, u, f)(t)$ is $\mathcal{C}^1$ in f for $t \in [t_0, t_0 + a]$. □

4.1. Linear variational equations. According to Theorem **I.21**, given an initial value problem as (1.5) which is $\mathcal{C}^p$ and it has solution $x(t_0, u, f)$, the linear variational equations are:

Variational equation for t_0: For any $t \geq t_0$, $D_{t_0} x(t_0, u, f)\colon \mathbb{R} \to \mathbb{R}^n$ is linear and continuous and for any $t \in \mathbb{R}$,$D_{t_0} x(t_0, u, f)t$ verifies

$$\begin{cases} \dot{y} = D_1 f(t, x_t(t_0, u, f)) + D_2 f(t, x_t(t_0, u, f)) y_t \\ y_{t_0} \equiv 0. \end{cases}$$

Variational equation for u: For any $t \geq t_0$, $D_u x(t_0, u, f)(t)\colon C \to \mathbb{R}^n$ is linear and continuous and for any $v \in C$, $D_u x(t_0, u, f)v(t)$ verifies

$$\begin{cases} \dot{y} = D_2 f(t, x_t(t_0, u, f)) y_t \\ y_{t_0} \equiv id. \end{cases}$$

Variational equation for f: For any $t \geq t_0$, $D_f x(t_0, u, f)(t)\colon \mathcal{C}^p(\Omega, \mathbb{R}^n) \to \mathbb{R}^n$ is linear and continuous and for any $g \in \mathcal{C}^p(\Omega, \mathbb{R}^n)$, $D_f x(t_0, u, f)g(t)$ verifies

$$\begin{cases} \dot{z} = D_2 f(t, x_t(t_0, u, f)) z_t + g(t, x_t(t_0, u, f)) \\ z_{t_0} \equiv 0. \end{cases}$$

5. The solution map

DEFINITION I.22. Let $\Omega \subset \mathbb{R} \times C$ be open set and $f\colon \Omega \to \mathbb{R}^n$ be continuous. The solution map is defined by

$$\begin{aligned} T(t,t_0)\colon C &\longrightarrow C \\ u &\longmapsto x_t(t_0,u) \end{aligned}$$

where $x_t(t_0,u)$ is the solution of the Delay Differential Equation with a unique solution

$$\begin{cases} \dot{x} = f(t,x_t) \\ x_{t_0} \equiv u. \end{cases}$$

By Theorem **I.20**, T is continuous. If now, we consider that the Delay Differential Equation is autonomous, i.e. $\dot{x} = f(x_t)$, the parameter t_0 has not any role, so we can assume $t_0 = 0$ and consider $T(t)$ instead of $T(t,t_0)$. Then

PROPOSITION I.23. *Let $\Omega \subset \mathbb{R} \times C$ be an open set and $f\colon \Omega \to \mathbb{R}^n$ be a continuous map. The solution map*

$$\begin{aligned} T(t)\colon C &\longrightarrow C \\ u &\longmapsto x_t(u), \end{aligned}$$

verifies

i. $T(0) = id$.
ii. $T(t) \circ T(s) = T(t+s)$
iii. $T(t)(u)$ *is continuous in* (t,u).

where it is understood that t and s are allowed to range over an interval may depend on u.

PROOF.

i. $T(0)(u) = x_0(u) = u = id(u)$ for any u.
ii. It follows by the uniqueness assumption of solutions.
iii. By Theorem **I.20**. □

For Ordinary Differential Equation, the solution map defines an homeomorphism. However, for Delay Differential Equation may not be true. Let us show a short collection of properties of the solution map $T(t,t_0)$ valid for any equation whose unique solution for any given initial condition.

Firstly, let us start with general facts:

DEFINITION I.24.

- A *bounded set* in a metric space is a subset contained in a ball.
- A mapping between metric spaces is *bounded* when it takes closed bounded sets into bounded sets.
- A mapping from a topological space to a metric space is *locally bounded* when it takes some neighbourhood of each point into a bounded set.
- A mapping from a metric space to a topological space is *compact* when it takes bounded sets to a relatively compact sets.
- A mapping from a metric space to a topological space is *locally compact* when it takes some bounded neighbourhood of each point into a relatively compact set.

Remark I.25. A locally bounded map may not be locally compact. Indeed, the identity map $id\colon \mathbb{R} \to \mathbb{R}$ is locally bounded from $\mathbb{R}$ with the euclidian topology to $\mathbb{R}$ with the discrete topology. But it is not locally compact because $id(B(0;2)) = \mathbb{R}$ is not relatively compact.

LEMMA I.26. *Any continuous map f from a topological space to a metric space is locally bounded.*

PROOF. By continuity, $f^{-1}(B(f(x),1))$ is open and f restricted in that open is bounded. □

COROLLARY I.27. *$T(t,t_0)$ is locally bounded for $t \geq t_0$.*

PROOF. Since $T(t,t_0)u$ is continuous in (t,t_0,u), it follows that for any $t \geq t_0$ and $u \in C$ for which $(t_0,u) \in \Omega$ and $T(t,t_0)u$ is defined, there is a neighbourhood U of u in C which depends on (t,t_0,u) such that $T(t,t_0)(U)$ is bounded. □

LEMMA I.28. *$T(t,t_0)$ is locally compact for $t \geq t_0 + r$.*

PROOF. By continuity of f and $T(s,t_0)u$, for any $t \geq t_0$, there is a neighbourhood $U(t,t_0,u)$ of u and a constant M such that for any $t_0 \leq s \leq t$,

$$\|T(s,t_0)\big(U(t,t_0,u)\big)\| \leq M \qquad \text{and} \qquad |f\big(s,T(s,t_0)(U(t,t_0,u)\big)| \leq M.$$

That means $|\dot{x}(t_0,U(t,t_0,u))(s)| \leq M$ for any $t_0 \leq s \leq t$. Thus, the family of mappings

$$\{x_t(t_0,v)\colon v \in U(t,t_0,v)\}$$

is precompact for $t \geq t_0 + r$ and by Lemma **A.2**, it is relatively compact. □

Now, we shall show that with some extra conditions, the solution map will be compact.

DEFINITION I.29.

- A map $T(\lambda)\colon X \to Y$ between metric spaces depending on a parameter in a metric space Λ is said to be *bounded uniformly on compact sets of* Λ when

$$\forall\, \Lambda_0 \subset \Lambda \text{ compact}, \forall\, U \subset X \text{ bounded}, \exists\, V \subset Y \text{ bounded}; \forall\, \lambda \in \Lambda_0,\ T(\lambda)(U) \subset V.$$

- A map $T(t,t_0)\colon X \to Y$ from a topological space to a metric spaces with $t \geq t_0$ is said to be *conditionally compact* when $T(t,t_0)(u)$ is continuous in (t,t_0,u) and

$$\forall\, V \subset Y \text{ bounded}, \exists\, K \subset Y \text{ compact}; \forall\, t_0 \leq s \leq t,\ T(s,t_0)(u) \in V \Rightarrow T(t,t_0)(u) \in K.$$

LEMMA I.30. *Let $T(t,t_0)\colon X \to Y$ be a map between metric spaces defined for $t \geq t_0$ and bounded uniformly on compact sets of $[t_0,+\infty)$.*

If $T(t,t_0)$ is conditionally compact, it is compact for $t \geq t_0$.

PROOF. Given $U \subset X$ bounded and $t \geq t_0$. There is $V \subset Y$ bounded such that

$$T(s,t_0)(U) \subset V \qquad \text{with } t \geq s \geq t_0.$$

There is also $K \subset Y$ compact such that $T(t,t_0)(U) \subset K$. So $\overline{T(t,t_0)(U)}$ is compact. □

THEOREM I.31 (Solution map representation). *Let $f\colon \mathbb{R} \times C \to \mathbb{R}^n$ be a bounded continuous map. The solution map can be written as*

$$T(t,t_0) = \Phi(t-t_0) + \Psi(t,t_0), \qquad t \geq t_0$$

where $\Phi(t-t_0)\colon C \to C$ is defined by

$$u(\tau) \mapsto \begin{cases} u(t-\tau) - u(0) & \text{if } t-\tau < 0 \\ 0 & \text{if } t-\tau \geq 0 \end{cases}$$

and $\Psi(t,t_0)\colon C \to C$ is conditionally compact. Thus, $T(t,t_0)$ is a contraction for $t > t_0$ and is conditionally compact for $t \geq t_0 + r$.

PROOF. The map $\Phi(t-t_0)$ is linear and continuous for $t \geq t_0$. Then we define the continuous map $\Psi(t,t_0) = T(t,t_0) - \Phi(t-t_0)$. Concretely,

$$\Psi(t,t_0)(u)(\tau) = \begin{cases} u(0) & \text{if } t-\tau < t_0 \\ u(0) + \displaystyle\int_{t_0}^{t+t_0} f\big(s,T(s,t_0)(u)\big)\,ds & \text{if } t-\tau \geq t_0. \end{cases}$$

Since $\Phi(t-t_0)$ is bounded and linear, given $U \subset C$ bounded set, there is $V \subset C$ bounded set such that

$$T(s,t_0)(u) \in V \qquad t \geq s \geq t_0$$

provided that $\Psi(s,t_0)(u) \in U$ for $t \geq s \geq t_0$. Since f is bounded and continuous, there is M depending only on U, t_0 and t such that for $t \geq s \geq t_0$

$$\Psi(t,t_0)(u) \in U \Rightarrow |f\big(s,T(s,t_0)(u)\big)| \leq M.$$

Therefore,

$$\int_a^b |f\big(s,T(s,t_0)(u)\big)|\,ds \leq M(b-a) \qquad t \geq b \geq a \geq t_0.$$

By Lemma **I.9**, we define the compact

$$K = \{v \in C\colon v \in U \text{ and } |v(\tau)-v(\sigma)| \leq M|\tau-\sigma| \text{ with } \tau,\sigma \in [0,r]\}.$$

Then $\Psi(t,t_0)(u) \in K$ because $\Psi(t,t_0)(u) = u(0)$ for $t-\tau \leq t_0$. Finally, we conclude the proof observing that $\Phi(t-t_0) = 0$ for $t - t_0 \geq r$. □

The next Corollary is straightforward using first Theorem **I.31** and then Lemma **I.30**.

COROLLARY I.32. *If $f\colon \mathbb{R} \times C \to \mathbb{R}^n$ is a bounded continuous map and the solution map $T(t,t_0)\colon C \to C$ with $t \geq t_0$ is a map bounded uniformly on compact sets of $[t_0,+\infty)$, then*

i. *The map $\Psi(t,t_0)$ of Theorem **I.31** is compact for $t \geq t_0$.*
ii. *$T(t,t_0)$ is compact for $t \geq t_0 + r$.*

6. Linear systems

A special case of Delay Differential Equation is the linear Delay Differential Equation. That is given $(t_0,u) \in \mathbb{R} \times C$,

$$\begin{cases} \dot{x}(t) = A(t)x_t + h(t) & t \geq t_0 \\ x_{t_0} \equiv u. \end{cases} \tag{1.7}$$

where h is continuous and A is linear and continuous.

THEOREM I.33. *There exists a unique solution $x(t_0,u)$ of (1.7) defined on $[t_0-r,+\infty)$.*

PROOF. Since $A(t)$ is linear and continuous, it is Lipschitz. So we have a locally unique solution by Theorem **I.12**. Now, let x be a maximal solution of (1.7) on $[t_0-r,+\infty)$. Integrating the system,

$$|x(t)| \leq |u(0)| + \int_{t_0}^t |A(s)x_s|\,ds + \int_{t_0}^t |h(s)|\,ds$$

for any $t \in [t_0,a)$. Thus

$$\|x_t\| \leq \|u\| + \int_{t_0}^t \|A(s)\|\|x_s\|\,ds + \int_{t_0}^t |h(s)|\,ds.$$

By Gronwall's Lemma,

$$\|x_t\| \leq \left(\|u\| + \int_{t_0}^t |h(s)|\,ds\right) \exp \int_{t_0}^t \|A(s)\|\,ds$$

for any $t \in [t_0,a)$. The right hand side in the above inequality is locally bounded for $t \in [t_0,+\infty)$. Hence, we obtain

$$\sup_{t_0 \leq t < a} \|x_t\| = M < +\infty.$$

Moreover, x is uniformly continuous on $[t_0,a)$ by the inequality

$$|x(t)-x(t')| \leq M\int_t^{t'} \|A(s)\|\,ds + \int_t^{t'} |h(s)|\,ds \qquad t_0 \leq t < t' < a.$$

So, $\{(t,x_t)\colon t_0 \leq t < a\}$ belongs to a compact set in $\mathbb{R} \times C$. This contradicts Theorem **I.18**. □

COROLLARY I.34. *Let $x(t_0,u,h)$ be the solution of (1.7). Then*

$$x(t_0,u,h)=x(t_0,u,0)+x(t_0,0,h)$$

where

$$\begin{aligned} x(t_0,\cdot,0)\colon C &\longrightarrow \mathcal{C}([t_0-r,+\infty),\mathbb{R}^n) \\ u &\longmapsto x(t_0,u,0) \end{aligned} \qquad \begin{aligned} x(t_0,0,\cdot)\colon \mathcal{C}([0,t],\mathbb{R}^n) &\longrightarrow \mathcal{C}([t_0-r,+\infty),\mathbb{R}^n) \\ h &\longmapsto x(t_0,0,h) \end{aligned}$$

are linear and continuous maps. Moreover, for $t\geq t_0$,

$$|x(t_0,u,0)(t)|\leq \|u\|\exp\int_{t_0}^{t}\|A(s)\|\,ds$$

and

$$|x(t_0,0,h)(t)|\leq \int_{t_0}^{t}|h(s)|\,ds\exp\int_{t_0}^{t}\|A(s)\|\,ds.$$

PROOF. $x(t_0,u,0)$ and $x(t_0,0,h)$ are solution, respectively, of

$$\begin{cases}\dot{x}(t)=A(t)x_t & t\geq t_0\\ x_{t_0}\equiv u\end{cases} \qquad \text{and} \qquad \begin{cases}\dot{x}(t)=A(t)x_t+h(t) & t\geq t_0\\ x_{t_0}\equiv 0.\end{cases}$$

So by linearity of $A(t)$,

$$\begin{aligned}\frac{\partial\big(x(t_0,u,0)+x(t_0,u,0)\big)}{\partial t}(t)&=\frac{\partial x(t_0,u,0)}{\partial t}(t)+\frac{\partial x(t_0,u,0)}{\partial t}(t)\\ &=A(t)x_t(t_0,u,0)+A(t)x_t(t_0,u,0)+h(t)\\ &=A(t)\big(x_t(t_0,u,0)+x_t(t_0,u,0)\big)+h(t),\end{aligned}$$

and $x_{t_0}(t_0,u,0)+x_{t_0}(t_0,0,h)\equiv u$. By uniqueness, $x(t_0,u,h)=x(t_0,u,0)+x(t_0,0,h)$. The linearity of $u(t_0,\cdot,0)$ and $x(t_0,0,\cdot)$ follows again by uniqueness of the solutions. The continuity of them follows from Theorem **I.20** and the inequalities from Gronwall's Lemma. □

7. Delay Differential Equations with a constant delay

In the previous Sections we have considered the Banach space $C=\mathcal{C}([0,r],\mathbb{R}^n)$ with uniform topology, $\Omega\subset\mathbb{R}\times C^2$ an open subset and $f\colon\Omega\to\mathbb{R}^n$ a continuous map. Then an initial value problem is

$$\begin{cases}\dot{x}(t)=f(t,x_t(\tau_1),x_t(\tau_2))\\ x_{t_0}\equiv u\end{cases} \tag{1.8}$$

where $x_t\in C$ is defined by $x_t(\tau)=x(t-\tau)$.
Now, we want to focus on a particular case of (1.8), that is, a Delay Differential Equation with a unique constant delay. In order to do so, let us take the next differential equation:

$$\begin{cases}\dot{x}(t)=f\big(t,x_t(\tau_1(t)),x_t(\tau_2(t))\big)\\ \dot{\tau}_1=0\\ \dot{\tau}_2=0\\ \tau_1(0)=0\\ \tau_2(0)=1.\end{cases}$$

It becomes to

$$\dot{x}(t)=f(t,x(t),x(t-1)). \tag{1.9}$$

The differential equation (1.9) can be viewed as an Ordinary Differential Equation (ODE) if it is expressed by $\dot{x}(t)=f(t,x(t),\varphi(t))$. Thus, the results proved in ODE's Theory are straightforward applied at each interval $[t_0+k-1,t_0+k]$ with $k\geq 0$ an integer. In particular, if f is of class $\mathcal{C}^p$, then the solution is of class $\mathcal{C}^p$ at each (t_0+k-1,t_0+k). Moreover, at each point $t=t_0+k$ we obtain an extra order of differentiability until $\mathcal{C}^p$.

EXAMPLE I.35. One of the easiest example is

$$\begin{cases} \dot{x}(t) = x(t-1) \\ x_0 \equiv 1. \end{cases}$$

That initial value problem can be computed explicitly in each interval $[k-1, k]$ with $k \geq 0$ an integer. Indeed,

$$\begin{aligned} t \in [-1, 0], \qquad & x(t) = 1 \\ t \in [0, 1], \qquad & x(t) = t + 1 \\ t \in [1, 2], \qquad & x(t) = \frac{t^2}{2} + t + \frac{1}{2} \\ t \in [2, 3], \qquad & x(t) = \frac{t^3}{6} + \frac{t^2}{2} + \frac{t}{2} + \frac{1}{6} \\ t \in [3, 4], \qquad & x(t) = \frac{t^4}{24} + \frac{t^3}{6} + \frac{t^2}{4} + \frac{t}{6} + \frac{1}{24} \\ & \vdots \end{aligned}$$

In Figure **I.1**, $\dot{x}(t) = x(t-1)$ and $\dot{x}(t) = x(t)$ are compared.

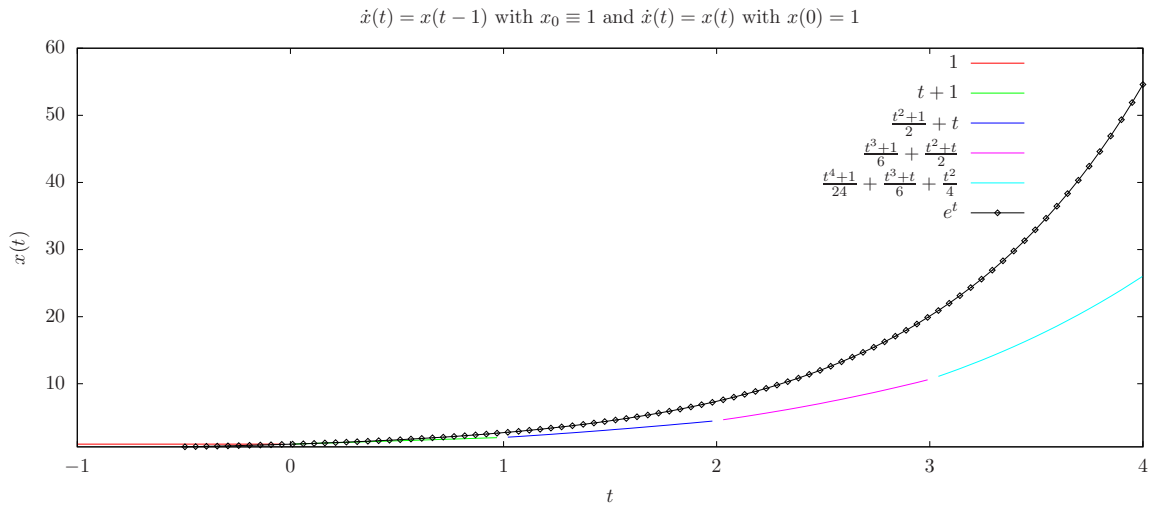

Figure I.1. Plot of a Delay Differential Equation and its corresponding Ordinary Differential Equation.

7.1. Linear differential equation with a constant delay. The Equation (1.7) with a constant delay becomes to

$$\dot{x}(t) = Ax(t) + Bx(t-1) + h(t). \tag{1.10}$$

where A, B and $\tau \geq 0$ are constants and h is a given continuous map. As an immediate consequence of the Theorem **I.33** is:

THEOREM I.36. *If u is a given continuous function on $[0, 1]$, there is a unique map $x(u, h)$ defined on $[-1, +\infty)$ that coincides with u on $[-1, 0]$ and satisfies the Equation (1.10) for $t \geq 0$. Moreover, $x(u, h)(t)$ is $\mathcal{C}^1$ for all $t > 0$ and it is $\mathcal{C}^1$ at $t = 0$ if, and only if, $u(\tau)$ has a derivative at $\tau = 0$ with*

$$\dot{u}(0) = Au(0) + Bu(-1) + h(0).$$

If h has derivatives of all orders, then $x(u, h)$ becomes smoother with increasing values of t.

PROOF. If x is a solution of (1.10) which coincides with u on $[-1,0]$, then the ordinary-variation-of-constants formula implies that x must satisfy

$$\begin{aligned} &x(t) = u(t), \qquad t \in [-1,0], \\ &x(t) = e^{At}u(0) + \int_0^t e^{A(t-s)}\big(Bx(s-1) + h(s)\big)\,ds, \qquad t \geq 0. \end{aligned} \tag{1.11}$$

Also, if x satisfies (1.11), it must satisfy (1.10). The uniqueness part follows the fact that it is unique in each interval $[k, k+1]$ for any integer $k \geq 0$.
The remainder statements follow from Theorem **I.21**. □

7.2. Characteristic equation of a homogeneous linear differential equation with a constant delay. Let us consider the linear Delay Differential Equation with a fixed delay $\tau \geq 0$,

$$\dot{x}(t) = Ax(t) + Bx(t-\tau) \tag{1.12}$$

It has a non-trivial solution $e^{\lambda t}c$ if, and only if,

$$\lambda - A - Be^{-\lambda\tau} = 0.$$

The map $h(\lambda) = \lambda - A - Be^{-\lambda\tau}$ is called characteristic map of (1.12). For any solution λ,

$$|\lambda - A| = |B|e^{-\tau \operatorname{Re}\lambda}.$$

So if $|\lambda| \to +\infty$, then $e^{-\tau \operatorname{Re}\lambda} \to \infty$. Besides, h is an entire map, so there is a real number α such that there can be only a finite number of zeros of $h(\lambda)$ in any compact set. Thus, there are only a finite number in any vertical strip in the complex plane. All that can be summarize in the following Lemma **I.37**.

LEMMA I.37. *Let $\dot{x}(t) = Ax(t) + Bx(t-\tau)$ be a linear Delay Differential Equation. It has a non-trivial solution $e^{\lambda t}c$ if, and only if,*

$$\lambda - A - Be^{-\lambda\tau} = 0. \tag{1.13}$$

If there is a sequence (λ_j) of solutions such that $|\lambda_j| \to +\infty$ as $j \to \infty$, then

$$\operatorname{Re}\lambda_j \to -\infty \quad \text{as } j \to \infty.$$

Therefore, there is a real number α such that all the solutions of (1.13) verify $\operatorname{Re}\lambda < \alpha$ and there are only a finite number of solutions in any vertical strip in the complex plane.

THEOREM I.38. *Let $\dot{x}(t) = Ax(t) + Bx(t-\tau)$ be a linear Delay Differential Equation. Let λ be a root of multiplicity m of the characteristic equation*

$$h(\lambda) = \lambda - A - Be^{-\lambda\tau} = 0.$$

Then $t^k e^{\lambda t}$ with $k = 0, \dots, m-1$ is a solution of the differential equation. Since it is linear, any finite sum of such solution is also a solution and infinite sums are also solutions under suitable conditions to ensure convergence.

PROOF. If $x(t) = t^k e^{\lambda t}$, then

$$\begin{aligned} e^{-\lambda t}\big(\dot{x}(t) - Ax(t) - Bx(t-\tau)\big) &= t^k\lambda + kt^{k-1} - At^k - B(t-\tau)^k e^{-\lambda\tau} \\ &= t^k\lambda + kt^{k-1} - At^k - Be^{-\lambda\tau}\sum_{j=0}^{k}\binom{k}{j}t^{k-j}\tau^j \\ &= \sum_{j=0}^{k}\binom{k}{j}t^{k-j}h^{(j)}(\lambda) \end{aligned}$$

If now λ is a zero of $h(\lambda)$ of multiplicity m, then $h^{(j)}(\lambda) = 0$ for $j = 0, \dots, m-1$. Therefore, $x(t) = t^k e^{\lambda t}$ is a solution for $k = 0, \dots, m-1$. □

CHAPTER II

Stability and Floquet Theory

1. Stability of solutions

DEFINITION II.1. Let $\Omega \subset \mathbb{R} \times C$ be open, $f\colon \Omega \to \mathbb{R}^n$ continuous and $x = 0$ so that

$$f(t, 0) = 0 \qquad \forall\, t.$$

Let us denote $x(t_0, u)$ solution of of the initial value problem of $\dot{x} = f(t, x_t)$. We say that:

- $x = 0$ is *stable* when
$$\forall\, t_0,\, \forall\, \varepsilon > 0,\, \exists\, \delta > 0;\, \forall\, t \geq t_0,\, \|u\| < \delta \Rightarrow \|x_t(t_0, u)\| < \varepsilon.$$
- $x = 0$ is *uniformly stable* when
$$\forall\, \varepsilon > 0,\, \exists\, \delta > 0;\, \forall\, t_0,\, \forall\, t \geq t_0,\, \|u\| < \delta \Rightarrow \|x_t(t_0, u)\| < \varepsilon.$$
- $x = 0$ is *asymptotically stable* when it is stable and
$$\forall\, t_0,\, \exists\, \eta > 0;\, \|u\| < \delta \Rightarrow x(t_0, u)(t) \xrightarrow[t \to +\infty]{} 0.$$
- $x = 0$ is *uniformly asymptotically stable* when it is asymptotically stable and
$$\forall\, t_0,\, \exists\, \delta;\, \forall\, \eta > 0,\, \exists\, t_1;\, \forall\, t \geq t_0 + t_1,\, \|u\| \leq \delta \Rightarrow \|x_t(t_0, u)\| \leq \eta.$$

Remark II.2. Stability notions has been explained in Definition **II.1** for a solution $x = 0$. The general case is also defined if we do the next comment: Given $\dot{x} = f(t, x_t)$, a solution $x(t)$ verifies one of the definitions in **II.1** when the solution of $y = 0$ of the new equation

$$\dot{y} = f(t, y_t + x_t) - f(t, x_t)$$

verifies the same condition.

PROPOSITION II.3. *Let $\Omega \subset \mathbb{R} \times C$ be open, $f\colon \Omega \to \mathbb{R}^n$ continuous and $\omega > 0$ so that*

$$f(t + \omega, v) = f(t, v) \qquad \forall\, (t, v) \in \Omega.$$

If $\dot{x} = f(t, x_t)$ has unique solutions, then the solution $x = 0$ is stable when it is uniformly stable.

PROOF.

$\Rightarrow$) By periodicity, for any $t \geq t_0$, any integer k and any $u \in C$,

$$x_t(t_0, u) = x_{t+k\omega}(t_0 + k\omega, u).$$

Indeed, $x_{t+k\omega}(t_0 + k\omega, u)$ verifies

$$\begin{aligned}\frac{\partial x(t_0 + k\omega, u)}{\partial t}(t + k\omega) &= f(t + k\omega, x_{t+k\omega}(t_0 + k\omega, u)) \\ &= f(t, x_{t+k\omega}(t_0 + k\omega, u))\end{aligned}$$

and $x_{t_0+\omega} \equiv u$. By uniqueness of solutions, $x_{t+t_0}(t_0, u) = x_{t+t_0+k\omega}(t_0 + k\omega, u)$. Thus, it is enough to show that

$$\forall\, \varepsilon > 0,\, \exists\, \delta > 0;\, \forall\, t_0 \in [0, \omega],\, \forall\, t \geq t_0,\, \|u\| < \delta \Rightarrow \|x_t(t_0, u)\| < \varepsilon.$$

For $0 \leq t_0 \leq \omega$ and $t \geq t_0$, $x_{t+\omega}(t_0, u) = x_{t+\omega}(\omega, x_\omega(t_0, u))$

$\Leftarrow$) Clear. □

2. Floquet Theory for Delay Differential Equations

For a moment, let us recall the Floquet representation of a homogeneous linear periodic ordinary differential equation. Consider the homogeneous linear ω-periodic system

$$\dot{x} = A(t)x \qquad \text{and} \qquad A(t+\omega) = A(t) \tag{2.1}$$

where $A(t)$ is a continuous n-by-n real or complex matrix function of t. Then every fundamental matrix solution $X(t)$ of (2.1) has the form

$$X(t) = P(t)e^{Bt}$$

where $P(t)$ and B are n-by-n matrices, $P(t+\omega) = P(t)$ for all t and B is constant.

In functional differential equations, does not exist a complete Floquet theory. Although we have a Floquet representation in some cases.

2.1. Indexed families on a Banach space. In order to use the Spectral Theory of a linear and continuous map, we introduce the notion of periodic family of a Banach space.

DEFINITION II.4. Let X be a Banach space. An indexed family by $\mathbb{R}\times\mathbb{R}$ of linear and continuous maps

$$T(t,t_0)\colon X \to X, \qquad t \geq t_0$$

is a family on X when

i. $T(t_0,t_0) = id$.
ii. $T(t,t_0)\circ T(t_0,s) = T(t,s)$ for all $t \geq t_0 \geq s$.

It is called ω-periodic family on X when there is $\omega > 0$ verifying:

iii. $T(t+\omega, t_0+\omega) = T(t,t_0)$ for all $t \geq t_0$.
iv. There is $M > 0$ such that for all $0 \leq t_0 \leq \omega$ and $t_0 \leq t \leq t_0+\omega$,

$$\|T(t,t_0)\| \leq M.$$

DEFINITION II.5. Let $\{T(t,t_0)\}_{t\geq t_0}$ be an ω-periodic family. The period map is defined by

$$\begin{aligned} P(t_0)\colon X &\longrightarrow X \\ u &\longmapsto T(t_0+\omega,t_0)(u). \end{aligned}$$

LEMMA II.6. *Let $P(t_0)$ be a period map of an ω-period family $\{T(t,t_0)\}_{t\geq t_0}$. Then*

i. *$P(t_0)$ is linear and continuous.*
ii. $P(t_0+\omega) = P(t_0)$.
iii. $P^k(t_0) = T(t_0+k\omega, t_0)$.
iv. $T(t,t_0)\circ P^k(t_0) = P^k(t)\circ T(t,t_0)$.

PROOF.

i. Since $T(t_0+\omega,t_0)$ is linear and continuous.
ii. $P(t_0+\omega) = T(t_0+2\omega, t_0+\omega) = T(t_0+\omega,t_0) = P(t_0)$.
iii. We apply induction on k.
   - $k = 1$. It is just the definition.
   - It $P^k(t_0) = T(t_0+k\omega,t_0)$, then

$$\begin{aligned} T(t_0+k\omega+\omega,t_0) &= T(t_0+k\omega+\omega, t_0+\omega)\circ T(t_0+\omega,t_0) \\ &= P^k(t_0+\omega)\circ P(t_0) = P^k(t_0)\circ P(t_0) = P^{k+1}(t_0). \end{aligned}$$

iv. We have

$$\begin{aligned} T(t,t_0)\circ P^k(t_0) &= T(t,t_0)\circ T(t_0+k\omega,t_0) \\ &= T(t+k\omega,t_0+k\omega)\circ T(t_0+k\omega,t_0) = T(t+k\omega,t_0) \end{aligned}$$

and $P^k(t)\circ T(t,t_0) = T(t+k\omega,t)\circ T(t,t_0) = T(t+k\omega,t_0)$. □

PROPOSITION II.7. *Let $P(t_0)$ be a period map of an ω-period family $\{T(t,t_0)\}_{t\geq t_0}$. Then*

The non-zero point spectrum is independent of t_0.

In particular, the dimension of the eigenspace of a non-zero eigenvalue is independent of t_0.

PROOF. Let us consider $\sigma_p^*(P(t_0))$ the point spectrum without 0, we must show that

$$\sigma_p^*(P(t_0)) = \sigma_p^*(P(t)).$$

That means two inclusions:

$\subset$) Let $\lambda \neq 0$ and u so that $P(t_0)(u) = \lambda u$. By Lemma **II.6**,

$$0 = \big(T(t,t_0) \circ (P(t_0) - \lambda \cdot id)\big)(u) = \big((P(t) - \lambda \cdot id) \circ T(t,t_0)\big)(u)$$

whenever $t \geq t_0$. We observe that $T(t,t_0)(u)$ is λ-eigenvalue for $t \geq t_0$. Indeed, if it is zero, there is $k \neq 0$ such that $t + k\omega \geq t_0$, then by Lemma **II.6**,

$$0 = \big(T(t+k\omega,t_0) \circ T(t,t_0)\big)(u) = T(t+k\omega,t_0)(u) = P^k(t_0)(u) = \lambda^k u.$$

It contradicts the conditions $\lambda \neq 0$ or $u \neq 0$.

$\supset$) If $t_0 + k\omega > t$, by Lemma **II.6**, $\sigma_p^*(P(t)) \subset \sigma_p^*(P(t_0 + k\omega)) \subset \sigma_p^*(P(t_0))$. □

DEFINITION II.8. Let $P(t_0)$ be a period map of an ω-period family $\{T(t,t_0)\}_{t\geq t_0}$.

- The non-zero point spectrum is denoted by

$$\sigma_p^*(P).$$

their elements are called characteristic multipliers or Floquet multipliers.
- A λ is called a characteristic exponent when $e^{\lambda\omega}$ is a characteristic multiplier.

COROLLARY II.9. *λ is a characteristic multiplier when there is $u \neq 0$ such that for all $t \geq t_0$,*

$$T(t+\omega,t_0)u = \lambda T(t,t_0)u.$$

PROOF. We must show two implications:

$\Rightarrow$) If $\lambda \in \sigma_p^*(P)$, then $P(t_0)u = \lambda u$ for some $u \neq 0$. Then

$$T(t+\omega,t_0)u = T(t+\omega,t_0+\omega)T(t_0+\omega,t_0)u = T(t,t_0)P(t_0)u = \lambda T(t,t_0)u.$$

$\Leftarrow$) Take $t = t_0$ and apply Proposition **II.7**. □

LEMMA II.10. *Let A be a square matrix with a non-zero and unique eigenvalue. There is a matrix B such that*

$$A = e^B.$$

`N.B.:` There is a more general result which tells us that if a matrix A has a non-zero determinant, there is B verifying $A = e^B$ (see [**12**, ch. 4]*).*

PROOF. Let $\lambda \neq 0$ be the eigenvalue and $\log\lambda$ be a complex value so that $e^{\log\lambda} = \lambda$. The matrix $C = (\log\lambda)Id$ verifies,

$$e^C = \sum_{k\geq 0} \frac{C^k}{k!} = \sum_{k\geq 0} \frac{\log^k \lambda}{k!} Id = e^{\log\lambda} Id = \lambda Id.$$

Since A is diagonalizable, then $A = U(\lambda Id)U^{-1} = Ue^C U^{-1} = e^{UCU^{-1}}$. □

THEOREM II.11. *Let $P(t_0)$ be a period map of an ω-period family $\{T(t,t_0)\}_{t\geq t_0}$ on a Banach space X. If $P(t_0)$ is compact, then for any characteristic multiplier λ, there are $\varphi_1,\ldots,\varphi_d$ in X, a constant d-by-d matrix B and a d-row vector $\varphi(t)$ in X such that*

i. $\sigma(e^{B\omega}) = \{\lambda\}$.
ii. $\varphi(t_0) = (\varphi_1,\ldots,\varphi_d)$.
iii. $\varphi(t+\omega) = \varphi(t)$ *for all t in $\mathbb{R}$.*

iv. *For all* $t \geq t_0$,

$$T(t,t_0)\varphi(t_0) = \varphi(t)e^{B(t-t_0)}.$$

If ψ *is any* d*-vector, then*

$$T(t,t_0)\varphi(t_0)\psi = \varphi(t)e^{B(t-t_0)}\psi.$$

Moreover, the generalized eigenspace of $P(t)$ *for* λ *has the same rank* $k \geq 1$ *and basis* $\varphi(t)$*. And its dimension is independent of* $t \in \mathbb{R}$*.*

Proof. By Theorem **D.15**, we can obtain the decomposition

$$X = N(\lambda) \oplus F(\lambda).$$

with $N(\lambda)$ finite dimensional subspace. Let $\varphi_1, \ldots, \varphi_d$ be a basis of $N(\lambda)$, denoted $\varphi(t_0)$. Since $P(0)\big(N(\lambda)\big) \subset N(\lambda)$, by Linear Algebra, there is a d-by-d matrix M such that

$$P(0)\varphi(t_0) = \varphi(t_0)M.$$

Since $\sigma(u \restriction N(\lambda)) = \{\lambda\}$, the only eigenvalue of M is $\lambda \neq 0$. By Lemma **II.10**, there is a d-by-d matrix B such that $M = e^{B\omega}$. Let us define

$$\varphi(t) = T(t,t_0)\varphi(t_0)e^{-B(t-t_0)}, \qquad t \geq t_0.$$

Then for $t \geq t_0$,

$$\begin{aligned}
\varphi(t+\omega) &= T(t+\omega,t_0)\varphi(t_0)e^{-B(t+\omega-t_0)} \\
&= T(t+\omega,t_0+\omega)T(t_0+\omega,t_0)\varphi(t_0)e^{-B\omega}e^{-B(t-t_0)} \\
&= T(t,t_0)T(t_0+\omega,t_0)\varphi(t_0)e^{-B\omega}e^{-B(t-t_0)} \\
&= T(t,t_0)P(t_0)\varphi(t_0)e^{-B\omega}e^{-B(t-t_0)} \\
&= T(t,t_0)\varphi(t_0)e^{B\omega}e^{-B\omega}e^{-B(t-t_0)} \\
&= T(t,t_0)\varphi(t_0)e^{-B(t-t_0)} = \varphi(t).
\end{aligned}$$

We can extend $\varphi(t)$ for $t \in \mathbb{R}$ in the following way:

$$t \mapsto \varphi(t+m\omega), \qquad \text{for any integer } m \text{ such that } t+m\omega \geq t_0.$$

Then the new $\varphi(t)$ is ω-periodic in $t \in \mathbb{R}$. Now, we claim that

$$T(t,t_0)\ker\big((P(t_0)-\lambda\cdot id)^j\big) = \ker\big((P(t)-\lambda\cdot id)^j\big), \qquad t \geq t_0 \text{ and } j \geq 1. \tag{2.2}$$

Let us show it by induction on j.

- $j = 1$. By Lemma **II.6**,

  $$T(t,t_0)\circ(P(t_0)-\lambda\cdot id) = (P(t)-\lambda\cdot id)\circ T(t,t_0),$$

  so it follows (2.2) for $j = 1$.
- Again by Lemma **II.6** and by induction hypothesis, (2.2) is proved.

Relation (2.2) tells us that $T(t,t_0)$ with $t \geq t_0$ maps the generalized eigenspace $P(t_0)$ for λ onto the generalized eigenspace $P(t)$ for λ. Let us prove that the restriction of $T(t,t_0)$ with $t \geq t_0$ to the generalized eigenspace of $P(t_0)$ for λ is injective. Indeed, if u is an element of the generalized eigenspace of $P(t_0)$ for λ and $T(t,t_0)u = 0$ for some $t_0 \leq t \leq t_0 + m\omega$ and some integer m, then

$$(P(t_0)-\lambda\cdot id)^k u = 0 \qquad \text{for some integer } k$$

moreover, $P^m(t_0)u = T(t_0+m\omega,t_0)u = T(t_0+m\omega,t)T(t,t_0)u = 0$. Since $\lambda \neq 0$, the polynomials $(x-\lambda)^k$ and x^m are coprimes, so by Bézout's identity,

$$a(x)(x-\lambda)^k + b(x)x^m = 1$$

for some polynomials $a(x)$ and $b(x)$. Therefore,

$$u = a(P(t_0))(P(t_0)-\lambda\cdot id)^k u + b(P(t_0))P^m(t_0)u = 0$$

and the injectivity has been proved. We also see that the generalized eigenspace of $P(t)$ for λ is equal to $\ker\big((P(t)-\lambda\cdot id)^k\big)$ with the same k as the one for $P(t_0)$. So $\varphi(t)$ defines a basis of $\ker\big((P(t)-\lambda\cdot id)^k\big)$. □

2.2. Floquet representation for linear Delay Differential Equations.

PROPOSITION II.12. *Let $A(t)\colon C\to\mathbb{R}^n$ be a linear and continuous map and $\omega>0$ so that for any t,*

$$A(t+\omega)=A(t).$$

The indexed family of maps

$$\begin{aligned} T(t,t_0)\colon C &\longrightarrow C \qquad\qquad t\geq t_0\\ u &\longmapsto x_t(t_0,u)\end{aligned} \tag{2.3}$$

is an ω-periodic family on C where $x_t(t_0,u)$ is the solution defined on $[t_0-r,+\infty)$ of

$$\begin{cases}\dot{x}=A(t)x_t\\ x_{t_0}\equiv u.\end{cases}$$

In particular, its period map

$$\begin{aligned} P(t_0)\colon C &\longrightarrow C\\ u &\longmapsto x_{t_0+\omega}(t_0,u)\end{aligned} \tag{2.4}$$

is ω-periodic with respect to t_0 and it is also linear and continuous.

PROOF. By Theorem **I.33**, for any $t_0\in\mathbb{R}$ and $u\in C$, there is a solution $x(t_0,u)$ of (2.3) defined on $[t_0-r,+\infty)$. By Theorem **I.20**, $x(t_0,u)$ is continuous and by Lemma **I.7** $x_t(t_0,u)$ is also continuous. By Corollary **I.34**, $x(t_0,\cdot)$ is linear, so $x_t(t_0,\cdot)$ is linear. Thus, $T(t,t_0)$ is linear and continuous whenever $t\geq t_0$. We must check the other conditions of the Definition **II.4**:

i. $T(t_0,t_0)(u)=x_{t_0}(t_0,u)\equiv u$. So $T(t_0,t_0)=id$.
ii. $T(t,t_0)\circ T(t_0,s)=T(t,s)$. Indeed, by uniqueness of solutions, $x_t(t_0,x_{t_0}(s,u))=x_t(s,u)$ for $t\geq t_0\geq s$.
iii. $T(t+\omega,t_0+\omega)=T(t,t_0)$. Indeed, let $x(t_0+\omega,u)$ and $x(t_0,u)$ be solutions. Then

$$\frac{\partial x(t_0+\omega,u)}{\partial t}(t+\omega)=A(t+\omega)x_{t+\omega}(t_0+\omega,u)=A(t)x_{t+\omega}(t_0+\omega,u),$$
$$\frac{\partial x(t_0,u)}{\partial t}(t+\omega)=A(t+\omega)x_{t+\omega}(t_0,u)=A(t)x_{t+\omega}(t_0,u)$$

and

$$\frac{\partial x(t_0,u)}{\partial t}(t)=A(t)x_t(t_0,u).$$

By uniqueness, $x_{t+\omega}(t_0+\omega,u)=x_{t+\omega}(t_0,u)=x_t(t_0,u)$ for all u.
iv. It follows from Corollary **I.34**.

The period map comes from Definition **II.5** and Lemma **II.6**. □

Thanks to Theorem **II.11**, we can formulate the next Theorem **II.13**.

THEOREM II.13. *Let $A(t)\colon C\to\mathbb{R}^n$ be a linear and continuous map and $\omega>0$ so that for any t,*

$$A(t+\omega)=A(t).$$

Let $P(t_0)$ be its period map. For any λ characteristic multiplier, there are a d-dimensional basis $\varphi_1,\dots,\varphi_d$ of a $P(t_0)$-invariant vector subspace, a constant d-by-d matrix B and an n-by-d matrix function $\varphi(t)$ on C such that

i. $\sigma(e^{B\omega})=\{\lambda\}$.
ii. $\varphi(t_0)=(\varphi_1,\dots,\varphi_d)$.
iii. $\varphi(t+\omega)=\varphi(t)$ *for all t in $\mathbb{R}$.*

iv. *For all* $t \geq t_0$,

$$x_t(t_0, \varphi(t_0)) = \varphi(t)e^{B(t-t_0)}.$$

If ψ *is any d-vector, then*

$$x_t(t_0, \varphi(t_0)\psi) = \varphi(t)e^{B(t-t_0)}\psi.$$

Moreover,

v. *The characteristic multiplier is independent of* t_0.

vi. *The generalized eigenspace of* $P(t)$ *for* λ *has the same rank* $k \geq 1$ *and basis* $\varphi(t)$. *And its dimension is independent of* $t \in \mathbb{R}$.

vii. $\lambda = e^{\mu\omega}$ *is a characteristic multiplier when there is a non-zero solution of the Delay Differential Equation* $\dot{x} = A(t)x_t$ *of the form*

$$x(t) = p(t)e^{\mu t}$$

where $p(t + \omega) = p(t)$.

PROOF. Since $\omega > 0$, there is an integer $m > 0$ such that $m\omega \geq r$. Then $P^m(t_0) = T(t_0 + m\omega, t_0)$ by Lemma **II.6**. Therefore, by Corollary **I.32**, $P^m(t_0)$ is compact. There is not any problem if ω is substituted by $m\omega$. Then i, ii, iii, iv and vi follows by Theorem **II.11**. The v follows by Proposition **II.7**. Thus, it only remains to show vii.

vii. Let us suppose that $t_0 = 0$. Then

$$x_t(0, u)(\tau) = x(0, u)(t - \tau) = x_{t-\tau}(0, u)(0)$$

whenever $0 \leq \tau \leq r$. If $u \neq 0$ is an eigenvector of eigenvalue λ, for all $0 \leq \tau \leq r$,

$$\varphi(t)(\tau) = \varphi(t - \tau)(0)e^{B\tau}.$$

Putting $\widetilde{\varphi}(t - \tau) = \varphi(t - \tau)(0)$ and $u = \varphi(0)v$ for some v, then for all t in $\mathbb{R}$,

$$x(0, \varphi(0)v)(t) = \widetilde{\varphi}(t)e^{Bt}v.$$

$\Rightarrow$) Since $\sigma(e^{B\omega}) = \{\lambda\} = \{e^{\mu\omega}\}$, then $\sigma(B) = \{\mu\}$. Thus

$$x(0, \varphi(0)v)(t) = \widetilde{\varphi}(t)e^{\mu t}v.$$

$\Leftarrow$) We must show that $\lambda = e^{\mu\omega}$ is characteristic multiplier. Indeed,

$$P(0)(u)(0) = x_\omega(0, u)(0) = x(0, u)(\omega) = p(\omega)e^{\mu\omega} = p(0)\lambda.$$

By hypothesis $p(0) \neq 0$. So λ is a characteristic multiplier. □

COROLLARY II.14. *Let* $A(t)\colon C \to \mathbb{R}^n$ *be a linear, continuous and* ω*-periodic map and* $P(t_0)$ *its periodic map with* λ *characteristic multiplier,*

i. *If* $|\lambda| < 1$ *for all* λ, *the solution* $x = 0$ *is uniformly asymptotically stable.*

ii. *If* $|\lambda| \leq 1$ *for all* λ, *the solution* $x = 0$ *is uniformly stable.*

CHAPTER III

Automatic differentiation

The recent sophisticated software tools and the high level programming languages have made the computation of some numerical issues, in particular, evaluation programs easier and more important. On one hand, some procedures of a program can now be determined more or less automatically, that is, without the user having to rewrite the function evaluation procedure. It is a quite common situation since mathematical models often depend on a possibly large number of parameters (some of them maybe unknown). On the other hand, optimization tasks are a key objective because qualitative and quantitative dependence are every time much complicated. In this context appears the automatic differentiation or also called algorithmic differentiation, in any case the acronym is AD. The main idea behind AD is the following:

"AD differentiates what you implement".

The Chapter starts establishing how one may decompose an evaluation procedure in composition of elemental functions. Then the chain's rule applied several times will give us the desired derivatives.

1. Evaluation procedure

Let us suppose that we have a function $F\colon \mathbb{R}^n \to \mathbb{R}^m$ which can be decomposed it into even smaller atomic operations which will called elemental functions and typically they are denoted by φ_i. A composition procedure gives us the function $y = F(x)$, that is,

$$\begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix} = F \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}.$$

The evaluation procedure can be expressed as:

$$\begin{aligned} v_{i-n} &= x_i && i = 1, \dots, n \\ v_i &= \varphi_i(v_j)_{j \prec i} && i = 1, \dots, s \\ y_{m-i} &= v_{s-i} && i = m-1, \dots, 0. \end{aligned}$$

That means,

- Firstly, the inputs x_i's are assigned to a new variables v_{i-n}'s.
- Secondly, $F(x)$ is computed in a finite way by composition of its elemental functions.
- Finally, the outputs y_{m-i}'s are assigned independently[1] by v_{s-i}'s.

The notation $\prec$ is an order relation called *dependence relation*. It is define as follows:

$$j \prec i \iff v_i \text{ depends directly on } v_j.$$

Typically, it will happen that $j < i$ as integer numbers.

As usual notation in computer science, we will write

$$j \prec^* i \iff j \prec i_1 \prec \cdots \prec i_r \prec i \text{ for some } r.$$

[1]This step avoids possible problems that may appear in a parallelism paradigm.

Important comment. We have assumed that $F = \varphi_s \circ \cdots \circ \varphi_1$. Although we will not express with all detail, we can have elemental functions

$$\varphi_i \colon \mathbb{R}^{n_i} \to \mathbb{R}^{m_i}, \qquad i = 1, \ldots, s$$

with $n_1 = n$ and $m_s = m$. In some cases, one can break vector-valued elemental functions into their scalar components. Hence, one can always achieve $m_i = 1$ for notational simplicity or the sake of conceptional. However, the drawback of this simplifying assumtion is that efficiency may be lost when several of the component functions involve common intermediate. Therefore, it will depend on the specific problem that we are codifying to choose which can be better; the vector-valued or the scalar-valued decomposition.

1.1. Overwrites. The evaluation procedure explained above has been interpreted from a mathematical point of view, that means, we have wanted to assume that each variable v_i's occurs exactly once time in the left-hand side and, of course, it does not affect any other variable when we are assigning it.
A reason of this point of view is that it allows to do a computational graph, that is, an acyclic graph whose vertices are simply the variables v_i's and an arc runs from v_j to v_i exactly when $j \prec i$. The roots of the graph represent the independent variables and the leaves the dependent variables. Typically, we draw the graph from the roots on the left to the leaves on the right. Optionally, one can also draw the inputs and the outputs (e.g. Figure **III.1**).

In a specific implementation for a specific problem, one want to be efficient with a minimum and adjacency use of memory. So one can try to overwrite some assignments and to use the minimum possible of variables for the combination of steps implemented. Authors as [**7**] defines the allocation function & assuming that a preprocessor or compiler generate an addressing scheme that maps the variables v_i's into subsets of an integer range.

EXAMPLE III.1. Let $F \colon \mathbb{R}^3 \to \mathbb{R}^2$ be a function defined by

$$(x, y, z) \mapsto (\cos(e^{x+y} + z), e^{x+y}).$$

The evaluation procedure can be expressed by

v_{-2}	$=$	x
v_{-1}	$=$	y
v_0	$=$	z
v_1	$=$	$v_{-2} + v_{-1}$
v_2	$=$	$\exp(v_1)$
v_3	$=$	$v_2 + v_0$
v_4	$=$	$\cos(v_3)$
v_5	$=$	v_2
y_1	$=$	v_4
y_2	$=$	v_5.

v_{-2}	$=$	x
v_{-1}	$=$	y
v_0	$=$	z
v_1	$=$	$v_{-2} + v_{-1}$
v_1	$=$	$\exp(v_1)$
v_2	$=$	$v_1 + v_0$
v_2	$=$	$\cos(v_2)$
y_1	$=$	v_2
y_2	$=$	v_1.

Table III.1. Evaluation procedure with and without overwriting.

The computational graph of this evaluation procedure is in Figure **III.1**.

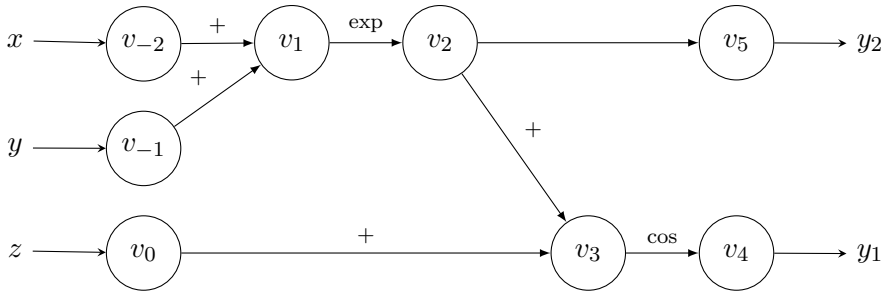

Figure III.1. Computational graph.

2. Univariate polynomial propagation

DEFINITION III.2. Given a ring R, the n-th degree truncated polynomial ring is defined as

$$R[x]_{\leq n} := R[x]/(x^{n+1}) \simeq \{a_n x^n + \cdots + a_0 \colon a_0, \ldots, a_n \in R\}.$$

Clearly, $R[x]_0 \simeq R$ and we have the ring ascending chain

$$R \subsetneq R[x]_1 \subsetneq R[x]_2 \subsetneq \cdots .$$

COROLLARY III.3. *$R[x]_{\leq n}$ is not an integral domain for any $n \geq 1$.*
In particular, it is neither a unique factorization domain nor a principal ideal domain nor an euclidian domain and nor a field.

PROOF. We must show that it has a zero divisor. Indeed, x^n and x are non-zero elements, but

$$x \cdot x^n = x^{n+1} = 0 \qquad \text{and} \qquad x^n \cdot x = x^{n+1} = 0. \qquad \square$$

If $R = \mathbb{R}$, then $\mathbb{R}[x]_{\leq n}$ can be endowed with a vectorial norm and inequalities like Triangular, Hölder and Minkowsky hold. Thus one can perform arithmetic just like on real numbers. In modern computer languages like C++, C#, Java, Python, ... one may simply overload real variables with n-th degree truncated polynomial variables. The parameter n can be fixed at compiler-time for an efficiency reason or be selected at runtime for a flexibility reason. One can also consider the free module

$$\mathbb{R}[x]^m_{\leq n} := \mathbb{R}[x]_{\leq n} \times \overset{(m}{\cdots} \times \mathbb{R}[x]_{\leq n}.$$

It is endowed with a norm associated to the norm of $\mathbb{R}[x]_{\leq n}$, typically the supremum of the norm of each variable.

LEMMA III.4. *Let $\varphi \colon \mathbb{R} \to \mathbb{R}$ be a derivable function verifying*

$$a(u) \cdot \varphi'(u) - b(u) \cdot \varphi(u) = c(u)$$

for $u \in \mathbb{R}[x]_{\leq n}$ and for some a, b and c. Then, $v = \varphi(u)$ may be obtained recurrently by

$$\begin{aligned} v_0 &= \varphi(u_0) \\ v_k &= \frac{1}{k a_0} \left(\sum_{j=1}^{k} \left(c_{k-j} + \sum_{i=0}^{k-j} b_i v_{k-i-j} \right) j u_j - \sum_{j=1}^{k-1} j a_{k-j} v_j \right), \qquad k = 1, \ldots, n. \end{aligned}$$

PROOF. By the chain's rule, $v' = \varphi'(u) \cdot u'$. That is, $a(u) \cdot v' = \big(c(u) + b(u) \cdot v\big) \cdot u'$. As we will see in detail in Theorem **III.5**, the product is just the discrete convolution. Thus,

$$a_0 k v_k + \sum_{j=1}^{k-1} j a_{k-j} v_j = \sum_{j=0}^{k} a_{k-j} j v_j = \sum_{j=0}^{k} \left(c_{k-j} + \sum_{i=0}^{k-j} b_{k-j-i} v_i \right) j u_j. \qquad \square$$

In particular, if $b(u) \equiv 0$ in Lemma **III.4**, then $\varphi(u)$ is simply a rational quadrature, i.e.

$$\varphi(u) = \int \frac{c(u)}{a(u)}\, du.$$

THEOREM III.5. *Let $u, v, w, s, c \in \mathbb{R}[x]_{\leq n}$, $\lambda, \alpha \in \mathbb{R}$ with $\alpha \neq 0$. Then*

	Recurrence for k up to n	*Complexity*
$v = u + w$	$v_k = u_k + w_k$	$n+1$
$v = \lambda u$	$v_k = \lambda u_k$	$n+1$
$v = u \cdot w$	$v_k = \sum_{j=0}^{k} u_j w_{k-j} = \sum_{j=0}^{k} u_{k-j} w_j$	$(n+1)^2$
$v = \frac{u}{w}$	$v_k = \frac{1}{w_0}\left(u_k - \sum_{j=1}^{k} w_j v_{k-j}\right)$	$(n+1)^2$
$v = u^2$	$v_k = \sum_{j=0}^{k} u_j u_{k-j}$	$\frac{1}{2}(n+1)^2$
$v = \sqrt{u}$	$v_0 = \sqrt{u_0}$ $v_k = \frac{1}{2v_0}\left(u_k - \sum_{j=1}^{k-1} v_j v_{k-j}\right)$	$\frac{1}{2}n^2$
$v = u^\alpha$	$v_0 = u_0^\alpha$ $v_k = \frac{1}{ku_0}\sum_{j=0}^{k-1}((k-j)\alpha - j)u_{k-j}v_j$	n^2
$v = \log(u)$	$v_0 = \log(u_0)$ $v_k = \frac{1}{u_0}\left(u_k - \frac{1}{k}\sum_{j=1}^{k-1}(k-j)u_j v_{k-j}\right)$	n^2
$v = e^u$	$v_0 = \exp(u_0)$ $v_k = \frac{1}{k}\sum_{j=0}^{k-1}(k-j)v_j u_{k-j}$	n^2
$s = \sin(u)$	$s_0 = \sin(u_0)$ $s_k = \frac{1}{k}\sum_{j=1}^{k} j c_{k-j} u_j$	$2n^2$
$c = \cos(u)$	$c_0 = \cos(u_0)$ $c_k = -\frac{1}{k}\sum_{j=1}^{k} j s_{k-j} u_j$	

Table III.2. Polynomial coefficient propagation through some univariate elemental functions.

PROOF.

✓ $v = u + w$ and $v = \lambda u$ are clear.

✓ $v = u \cdot w$ is just the Cauchy product. Indeed,

$$\left(\sum_{k=0}^{n} u_k x^k\right) \cdot \left(\sum_{k=0}^{n} w_k x^k\right) = \sum_{k=0}^{2n} x^k \sum_{j=0}^{k} u_j w_{k-j} - \sum_{k=0}^{n-1} x^k \left(u_k \sum_{j=n+1}^{2n-k} w_j x^j + w_k \sum_{j=n+1}^{2n-k} u_j x^j\right).$$

Modulo x^{n+1}, we obtain

$$\left(\sum_{k=0}^{n} u_k x^k\right) \cdot \left(\sum_{k=0}^{n} w_k x^k\right) = \sum_{k=0}^{n} x^k \sum_{j=0}^{k} u_j w_{k-j}.$$

It is clear that in $\mathbb{R}$ we also have

$$\sum_{j=0}^{k} u_j w_{k-j} = \sum_{j=0}^{k} u_{k-j} w_j.$$

✓ $v = \dfrac{u}{w}$ is equivalent to $w \cdot v = u$. Therefore

$$w_0 v_k + \sum_{j=1}^{k} w_j v_{k-j} = \sum_{j=0}^{k} w_j v_{k-j} = u_k \Rightarrow v_k = \frac{1}{w_0}\left(u_k - \sum_{j=1}^{k} w_j v_{k-j}\right).$$

✓ $v = u^2$ is clear because $u^2 = u \cdot u$.

✓ $v = \sqrt{u}$ is equivalent to $v \cdot v = u$. Therefore

$$v_0 v_k + \sum_{j=1}^{k-1} v_j v_{k-j} + v_k v_0 = \sum_{j=0}^{k} v_j v_{k-j} = u_k \Rightarrow v_k = \frac{1}{2v_0}\left(u_k - \sum_{j=1}^{k-1} v_j v_{k-j}\right).$$

✓ $v = u^\alpha$. Let us define $\varphi(x) = x^\alpha$. Then

$$u \cdot \left(u^\alpha \cdot \frac{\alpha}{u}\right) - \alpha u^\alpha = 0.$$

It follows now applying Lemma **III.4** with $a(u) = u$, $b(u) = \alpha$ and $c(u) = 0$.

✓ $v = \log(u)$. Let us define $\varphi(x) = \log(x)$. Then

$$u \cdot \frac{1}{u} - 0 \log(u) = 1.$$

To apply Lemma **III.4** with $a(u) = u$, $b(u) = 0$ and $c(u) = 1$.

✓ $v = e^u$. Let us define $\varphi(x) = e^x$. Then

$$1e^u - 1e^u = 0.$$

To apply Lemma **III.4** with $a(u) = 1$, $b(u) = 1$ and $c(u) = 0$.

✓ $v = \sin(u)$. Let us define $\varphi(x) = \sin(x)$. Then

$$1 \cos(u) - 0 \sin(u) = \cos(u).$$

To apply Lemma **III.4** with $a(u) = 1$, $b(u) = 0$ and $c(u) = \cos(u)$.

✓ $v = \cos(u)$. Let us define $\varphi(x) = \cos(x)$. Then

$$-1 \sin(u) - 0 \cos(u) = \sin(u).$$

To apply Lemma **III.4** with $a(u) = -1$, $b(u) = 0$ and $c(u) = \sin(u)$. □

Remark III.6. Some important observations are:

- One can use the symmetries of the expressions $v = u^2$ and $v = \sqrt{u}$. Indeed, we have

$$v_{2k} = u_k^2 + 2\sum_{j=0}^{k-1} u_j u_{2k-j} \qquad \text{and} \qquad v_{2k+1} = 2\sum_{j=0}^{k} u_j u_{2k+1-j}$$

for the first. It is just for this reason that the complexity is the reduced to the half.

- The formula of Theorem **III.5** for $v = u^\alpha$ may not work if $u_0 = 0$. However, if α is an odd integer, $v = u^\alpha$ has sense and one can computed by iteration of $u^\alpha = u^{\alpha-1} \cdot u$.
- The output of some expressions in Theorem **III.5** can be allocated in some input variable. For instance, $v = u \cdot w$, $v = \dfrac{u}{w}$ and $v = u^2$.

We can observe, in Script **III.1**, a possible declaration in a high-level programming language C++, using an Object Oriented Programming (OOP) paradigm.

```
#include <iostream>
#include <cmath>
using namespace std;

template <typename T>
class Polynomial
{
private:
  T *pol;
  unsigned int N;
  /* Private methods */
  ...
public:
  /* Constructors, public methods and destructor */
  ...
  /* Overloading operators */
  inline T& operator[](unsigned int i);
  inline Polynomial & operator=(const Polynomial &p);
  Polynomial operator+(const Polynomial &p);
  inline Polynomial operator+(const T &t);
  Polynomial& operator+=(Polynomial &p);
  inline Polynomial& operator+=(const T &t);
  Polynomial operator-(const Polynomial &p);
  inline Polynomial operator-(const T &t);
  Polynomial& operator-=(const Polynomial &p);
  inline Polynomial& operator-=(const T &t);
  Polynomial operator*(const Polynomial &p);
  inline T operator*(const T &t);
  Polynomial& operator*=(const Polynomial &p);
  inline Polynomial& operator*=(const T &t);
  Polynomial operator/(const Polynomial &p);
  inline Polynomial operator/(const T &t);
  Polynomial& operator/=(const Polynomial &p);
  inline Polynomial& operator/=(const T &t);

  friend std::ostream& operator<<(std::ostream &out, Polynomial &p);
}
template <typename T>
Polynomial<T> sqrt(const Polynomial<T> &p);


template <typename T>
Polynomial<T> pow(const Polynomial<T> &p, const T &t);


template <typename T>
Polynomial<T> exp(const Polynomial<T> &p);


template <typename T>
void sincos(const Polynomial<T>&p, Polynomial<T>&s, Polynomial<T>&c);
```

Script III.1. Arithmetic of univariate polynomial in C++.

3. Univariate Taylor's propagation

Let us recall the Taylor's expansion notion for a derivable map.

DEFINITION III.7. Let I be a non-trivial interval of $\mathbb{R}$. If $\varphi\colon I \to \mathbb{R}^m$ is a p-times derivable at $a \in I$, the p-th Taylor polynomial of φ at a is

$$P(\varphi)(t) = \sum_{j=0}^{p} \varphi^{[j]}(a)(t-a)^j \qquad \text{with } \varphi^{[j]} = \frac{\varphi^{(j)}}{j!} \text{ and } \varphi^{[0]} := \varphi.$$

It is clear that $P(\varphi)^{(j)}(a) = \varphi^{(j)}(a)$ for any $j = 0, \dots, p$.

NOTATION (Small "o").

$$g(x) = o(h(x)) \text{ as } x \to a \iff g(x) = h(x)h_0(x) \text{ and } h_0(x) \to 0 \text{ as } x \to a.$$

THEOREM III.8. *Let $\varphi\colon I \to \mathbb{R}^m$ be a p-times derivative map at $a \in I$. Then*

$$\varphi(t) = P(\varphi)(t) + o\big((t-a)^p\big) \text{ as } t \to a \Leftrightarrow \lim_{t\to a} \frac{\varphi(t) - P(\varphi)(t)}{(t-a)^p} = 0.$$

Moreover, $P(\varphi)(t)$ is the unique polynomial of degree $\leq p$ which satisfies that property.

The Theorem **III.8** tells us that any p-times derivative mapping $\varphi\colon I \to \mathbb{R}^m$ has a unique locally extension to $P_\varphi\colon \mathbb{R}[t]_{\leq p} \to \mathbb{R}[t]^m_{\leq p}$ defined by $u \mapsto P(\varphi \circ u)$. More abstractly, for any $p \geq 1$, there is a linear extension mapping

$$\begin{array}{ccc} \mathcal{C}^p(I, \mathbb{R}^m) & \longrightarrow & \mathcal{C}(\mathbb{R}[t]_{\leq p}, \mathbb{R}[t]^m_{\leq p}) \\ \varphi & \longmapsto & P_\varphi. \end{array}$$

One can generalize the above explanation as follows:

DEFINITION III.9. Let $U \subset \mathbb{R}^n$ be open. If $\varphi\colon U \to \mathbb{R}^m$ is a p-times differentiable map at $a \in U$, the p-th Taylor polynomial of φ at a is

$$P(\varphi)(t) = \sum_{j=0}^{p} D^{[j]}\varphi(a)(t-a)^\alpha \qquad \text{with } D^{[j]}\varphi = \frac{1}{\alpha!} D^j\varphi$$

where $D^j\varphi$ is defined by

$$D^j\varphi(a)(x) = \sum_{|\alpha|=j} \frac{|\alpha|!}{\alpha!} \frac{\partial^{|\alpha|}\varphi}{\partial^\alpha x}(a) x^\alpha$$

being $\alpha \in \mathbb{N}^n$ a mult-index[2].

THEOREM III.10. *Let $U \subset \mathbb{R}^n$ be open, $\varphi\colon U \to \mathbb{R}^m$ be a p-times differentiable map at $a \in U$. Then*

$$\varphi(t) = P(\varphi)(t) + o\big(|t-a|^p\big) \text{ as } t \to a \Leftrightarrow \lim_{t\to a} \frac{\varphi(t) - P(\varphi)(t)}{|t-a|^p} = 0.$$

Moreover, $P(\varphi)(t)$ is the unique polynomial of degree $\leq p$ which satisfies that property.

Hence each p-times differentiable mapping $\varphi\colon \mathbb{R}^n \to \mathbb{R}^m$ has a unique locally extension to $P_\varphi\colon \mathbb{R}[t]^n_{\leq p} \to \mathbb{R}[t]^m_{\leq p}$ defined by $u \mapsto P(\varphi \circ u)$. That is, for any $p \geq 1$, there is a linear extension mapping

$$\begin{array}{ccc} \mathcal{C}^p(\mathbb{R}^n, \mathbb{R}^m) & \longrightarrow & \mathcal{C}\big(\mathbb{R}[t]^n_{\leq p}, \mathbb{R}[t]^m_{\leq p}\big) \\ \varphi & \longmapsto & P_\varphi. \end{array}$$

Applying now also the Theorem **III.5**, the Theorem **III.11** is straightforward. Basically, because we are considering $\varphi(u_1(t), \dots, u_n(t))$ and the latter Theorem mentioned can be applied in each variable.

[2]Standard multi-index notation is used and $|\alpha|$ denotes the ℓ_1-norm of the multi-index α.

THEOREM III.11. *Let u, v, w, s, c be p-times derivatives maps, $\lambda, \alpha \in \mathbb{R}$ with $\alpha \neq 0$. Then*

	Recurrence for k up to p	*Complexity*
$v = u + w$	$v^{[k]}(t) = u^{[k]}(t) + w^{[k]}(t)$	$p+1$
$v = \lambda u$	$v^{[k]}(t) = \lambda u^{[k]}(t)$	$p+1$
$v = u \cdot w$	$v^{[k]}(t) = \sum_{j=0}^{k} u^{[j]}(t) w^{[k-j]}(t) = \sum_{j=0}^{k} u^{[k-j]}(t) w^{[j]}(t)$	$(p+1)^2$
$v = \frac{u}{w}$	$v^{[k]}(t) = \frac{1}{w^{[0]}(t)} \left(u^{[k]}(t) - \sum_{j=1}^{k} w^{[j]}(t) v^{[k-j]}(t) \right)$	$(p+1)^2$
$v = u^2$	$v^{[k]}(t) = \sum_{j=0}^{k} u^{[j]}(t) u^{[k-j]}(t)$	$\frac{1}{2}(p+1)^2$
$v = \sqrt{u}$	$v^{[0]}(t) = \sqrt{u^{[0]}(t)}$ $v^{[k]}(t) = \frac{1}{2v^{[0]}(t)} \left(u^{[k]}(t) - \sum_{j=1}^{k-1} v^{[j]}(t) v^{[k-j]}(t) \right)$	$\frac{1}{2}p^2$
$v = u^\alpha$	$v^{[0]}(t) = \left(u^{[0]}(t)\right)^\alpha$ $v^{[k]}(t) = \frac{1}{k u^{[0]}(t)} \sum_{j=0}^{k-1} ((k-j)\alpha - j) u^{[k-j]}(t) v^{[j]}(t)$	p^2
$v = \log(u)$	$v^{[0]}(t) = \log\left(u^{[0]}(t)\right)$ $v^{[k]}(t) = \frac{1}{u^{[0]}(t)} \left(u^{[k]}(t) - \frac{1}{k} \sum_{j=1}^{k-1} (k-j) u^{[j]}(t) v^{[k-j]}(t) \right)$	p^2
$v = e^u$	$v^{[0]}(t) = \exp\left(u^{[0]}(t)\right)$ $v^{[k]}(t) = \frac{1}{k} \sum_{j=0}^{k-1} (k-j) v^{[j]}(t) u^{[k-j]}(t)$	p^2
$s = \sin(u)$	$s^{[0]}(t) = \sin\left(u^{[0]}(t)\right)$ $s^{[k]}(t) = \frac{1}{k} \sum_{j=1}^{k} j c^{[k-j]}(t) u^{[j]}(t)$	$2p^2$
$c = \cos(u)$	$c^{[0]}(t) = \cos\left(u^{[0]}(t)\right)$ $c^{[k]}(t) = -\frac{1}{k} \sum_{j=1}^{k} j s^{[k-j]}(t) u^{[j]}(t)$	

Table III.3. Taylor coefficient propagation through some univariate elemental functions.

Propagation. The Taylor expansion tells us that a smooth map φ can be expressed locally as:

$$\varphi(t_0 + h) = \varphi^{[0]}(t_0) + \varphi^{[1]}(t_0)h + \varphi^{[2]}(t_0)h^2 + \varphi^{[3]}(t_0)h^3 + \cdots .$$

Therefore if we have an evaluation procedure $F = \varphi_s \circ \cdots \circ \varphi_1$, then using Table **III.3** we can obtain the Taylor expansion of F truncated to a prefixed order.

4. Gradient propagation

A multivariate Taylor propagation of a function $F\colon \mathbb{R}^n \to \mathbb{R}^m$ requires a monomial order prefixed. But, it does not matter in the case of the Gradient propagation. Indeed, let us assume, for instance, the monomial order

$$s_1 > \cdots > s_n > 1.$$

The gradient propagation of F will be an element in $\mathbb{R}[s_1, \ldots, s_n]_{\leq 1}$. The idea is to propagate

$$F(x_1 + s_1, \ldots, x_n + s_n)$$

where $s_1, \ldots, s_n$ will be symbols (i.e. indeterminates).
Let us formalize it with the notation of an evaluation procedure. Given

$$\begin{aligned} v_{i-n} &= x_i && i = 1, \ldots, n \\ v_i &= \varphi_i(v_j)_{j \prec i} && i = 1, \ldots, s \\ y_{m-i} &= v_{s-i} && i = m-1, \ldots, 0. \end{aligned}$$

an evaluation procedure for the function F. Introducing $s_1, \ldots, s_n$ symbols, the Gradient propagation is:

$$\begin{aligned} v_{i-n} &= x_i \\ \dot{v}_{i-n} &= s_i \end{aligned} \qquad i = 1, \ldots, n$$

$$\begin{aligned} v_i &= \varphi_i(v_j)_{j \prec i} \\ \dot{v}_i &= \sum_{j \prec i} \frac{\partial \varphi_i}{\partial v_j}\big((v_j)_{j \prec i}\big)\dot{v}_j \end{aligned} \qquad i = 1, \ldots, s$$

$$\begin{aligned} y_{m-i} &= v_{s-i} \\ \dot{y}_{m-i} &= \dot{v}_{s-i} \end{aligned} \qquad i = m-1, \ldots, 0.$$

Abbreviating $u_i = (v_j)_{j \prec i}$ and $\dot{u}_i = (\dot{v}_j)_{j \prec i}$ we write $\dot{v}_i = \dot{\varphi}_i(u_i)\dot{u}_i$.
When v_i shares a location with one of its arguments v_j, the derivative $\dot{v}_i$ will be incorporated after v_i has been updated. So it is a common notation to indicate $[v_i, \dot{v}_i]$ in order to indicate that they should be evaluated simultaneously sharing intermediate results.
Typically, given a function F, one want to evaluate F and DF at a given point. It rarely makes sense to evaluate DF without evaluating F at the same time. Thus, we are going to assume that a Gradient propagation procedure includes the evaluation of the underlying function F itself.

The next Proposition **III.12** is a straightforward result of an elementary course of Maths and the proof will be skipped.

PROPOSITION III.12. *Let u, v, w be derivable maps and $\lambda, \alpha \in \mathbb{R}$. Then*

φ	$\dot{\varphi}$
$v = \lambda$	$\dot{v} = 0$
$v = u \pm w$	$\dot{v} = \dot{u} \pm \dot{w}$
$v = uw$	$\dot{v} = \dot{u}w + u\dot{w}$
$v = \dfrac{1}{u}$	$\dot{v} = -vv\dot{u}$
$v = \sqrt{u}$	$\dot{u} = \dfrac{\dot{u}}{2v}$
$v = u^{\alpha}$	$\dot{v} = \dfrac{\dot{u}}{u}$ $\dot{v} = \alpha v\dot{v}$
$v = \exp(u)$	$\dot{v} = v\dot{u}$
$v = \log(u)$	$\dot{v} = \dfrac{\dot{u}}{u}$
$v = \cos(u)$	$\dot{v} = -\sin(u)\dot{u}$
$v = \sin(u)$	$\dot{v} = \cos(u)\dot{u}$

Table III.4. Some tangent elemental operations.

We observe that all procedures in Table **III.4** are still correct when two or even three of the variables u, v or w coincide. That fact, it is usually called "alias-safe".

EXAMPLE III.13. Let $F\colon \mathbb{R}^3 \to \mathbb{R}^2$ be a function defined by

$$(x, y, z) \mapsto (\cos(e^{x+y} + z), e^{x+y}).$$

The gradient evaluation procedure can be expressed by

$v_{-2} = x$	$\dot{v}_{-2} = s_x$
$v_{-1} = y$	$\dot{v}_{-1} = s_y$
$v_0 = z$	$\dot{v}_0 = s_z$
$v_1 = v_{-2} + v_{-1}$	$\dot{v}_1 = \dot{v}_{-2} + \dot{v}_{-1}$
$v_2 = \exp(v_1)$	$\dot{v}_2 = v_2\dot{v}_1$
$v_3 = v_2 + v_0$	$\dot{v}_3 = \dot{v}_2 + \dot{v}_0$
$v_4 = \cos(v_3)$	$\dot{v}_4 = -\sin(v_3)\dot{v}_3$
$v_5 = v_2$	$\dot{v}_5 = \dot{v}_2$
$y_1 = v_4$	$\dot{y}_1 = \dot{v}_4$
$y_2 = v_5$.	$\dot{y}_2 = \dot{v}_5$.

Table III.5. Gradient evaluation procedure without overwriting.

Explicitly, we have

$$\dot{y}_1 = -\sin(e^{x+y}+z)e^{x+y}s_x - \sin(e^{x+y}+z)e^{x+y}s_y - \sin(e^{x+y}+z)s_z$$
$$\dot{y}_2 = e^{x+y}s_x + e^{x+y}s_y.$$

Putting $s_x \leftrightarrow (1,0,0)$, $s_x \leftrightarrow (0,1,0)$ and $s_z \leftrightarrow (0,0,1)$ we have obtained

$$DF(x,y,z) = \begin{pmatrix} -\sin(e^{x+y}+z)e^{x+y} & -\sin(e^{x+y}+z)e^{x+y} & -\sin(e^{x+y}+z) \\ e^{x+y} & e^{x+y} & 0 \end{pmatrix}.$$

A possible declaration in a high-level programming language C++, using an Object Oriented Programming (OOP) paradigm is in Script **III.2**.

```
#include <iostream>
#include <cmath>
using namespace std;

template <typename T>
class Gradient
{
private:
  T *gra;
  unsigned int N;
  /* Private methods */
  ...
public:
  /* Constructors, public methods and destructor */
  ...
  /* Overloading operators */
  inline T& operator[](unsigned int i);
  inline Gradient & operator=(const Gradient &u);
  Gradient operator+(const Gradient &w);
  inline Gradient operator+(const T &t);
  Gradient& operator+=(Gradient &u);
  inline Gradient& operator+=(const T &t);
  Gradient operator-(const Gradient &w);
  inline Gradient operator-(const T &t);
  Gradient& operator-=(const Gradient &u);
  inline Gradient& operator-=(const T &t);
  Gradient operator*(const Gradient &w);
  inline T operator*(const T &t);
  Gradient& operator*=(const Gradient &u);
  inline Gradient& operator*=(const T &t);
  Gradient operator/(const Gradient &w);
  inline Gradient operator/(const T &t);
  Gradient& operator/=(const Gradient &w);
  inline Gradient& operator/=(const T &t);

  friend std::ostream& operator<<(std::ostream &out, Gradient &u);
}
template <typename T>
Gradient<T> sqrt(const Gradient<T> &u);
template <typename T>
Gradient<T> pow(const Gradient<T> &u, const T &t);
template <typename T>
Gradient<T> exp(const Gradient<T> &u);
template <typename T>
void sin(const Gradient<T>&u, Gradient<T>&s);
template <typename T>
void cos(const Gradient<T>&u, Gradient<T>&c);
```

Script III.2. Gradient propagation class in C++.

5. An application. Hermite's interpolation

Given two Taylor expansions at two different points, a question is whether the Taylor expansion at an intermediate point can be interpolated with a good accuracy. In order to do so, the next application has been studied.

Let $\varphi\colon [a,b] \to \mathbb{R}$ be a smooth map on (a,b). Given a table of values of $n+1$ points and its m_k first normalized derivatives. That is,

$$\begin{array}{c|cccc} x & \varphi^{[0]} & \varphi^{[1]} & \cdots & \\ \hline x_0 & \varphi_0^{[0]} & \varphi_0^{[1]} & \cdots & \varphi_0^{[m_0]} \\ \vdots & \vdots & \vdots & & \vdots \\ x_n & \varphi_n^{[0]} & \varphi_n^{[1]} & \cdots & \varphi_n^{[m_n]} \end{array} \qquad \text{with } \varphi^{[j]} = \frac{\varphi^{(j)}}{j!} \text{ and } \varphi^{(0)} := \varphi.$$

Assuming that $x_0 < \cdots < x_n$.
The Hermite's interpolation problem consists in finding a polynomial P of degree less or equals to

$$N = \sum_{k=0}^{n} (m_k + 1)$$

such that

$$P^{[j]}(x_k) = \varphi_k^{[j]} \qquad \text{for } j = 0, \ldots, m_k \text{ and } k = 0, \ldots, n.$$

By Rolle's Theorem, the error expression is well-known if φ is $\mathcal{C}^{N+1}(a,b)$ map

$$\varphi(x) - P(x) = \varphi^{[N+1]}(\xi(x))(x - x_0)^{m_0+1} \cdots (x - x_n)^{m_n+1}$$

with $x \in (a,b)$ and $\xi(x)$ between $x_0, \ldots, x_n, x$.

Construction of Hermite's polynomial. One can generalize the divided differences method used typically in the Newton's interpolation method. The divided differences are defined by:

$$\varphi[x_k^j] = \varphi_k^{[j]}$$

with $j = 0, \ldots, m_k$ and $k = 0, \ldots, n$. Then

$$\varphi[x_i^{k_i}, x_{i+1}^{k_{i+1}}, \ldots, x_{i+j}^{k_{i+j}}, x_{i+j+1}^{k_{i+j}}] = \frac{\varphi[x_i^{k_i - 1}, x_{i+1}^{k_{i+1}}, \ldots, x_{i+j+1}^{k_{i+j}}] - \varphi[x_i^{k_i}, \ldots, x_{i+j}^{k_{i+j}}, x_{i+j+1}^{k_{i+j}-1}]}{x_{i+j+1} - x_i} \tag{3.1}$$

with $i = 0, \ldots, n-j$ and $j = 0, \ldots, n-1$. If some exponent of (3.1) is outside of its range, the point will be removed. For instance,

$$\varphi[x_0^1, x_1^2, x_2^0] = \frac{\varphi[x_0^0, x_1^2, x_2^0] - \varphi[x_0^1, x_1^2]}{x_2 - x_0}.$$

Thus the Hermite's polynomial is

$$\begin{aligned} P(x) = {} & \varphi[x_0^0] + \varphi[x_0^1](x - x_0) + \cdots + \varphi[x_0^{m_0}](x - x_0)^{m_0} \\ & + \varphi[x_0^{m_0}, x_1^0](x - x_0)^{m_0+1} + \varphi[x_0^{m_0}, x_1^1](x - x_0)^{m_0+1}(x - x_1) \\ & + \cdots\cdots \\ & + \varphi[x_0^{m_0}, \ldots, x_{n-1}^{m_{n-1}}, x_n^{m_n}](x - x_0)^{m_0+1} \cdots (x - x_{n-1})^{m_{n-1}+1}(x - x_n)^{m_n}. \end{aligned} \tag{3.2}$$

Let us do an example:

x_0	$\boxed{\varphi_0^{[0]}}$							
		$\boxed{\varphi_0^{[1]}}$						
x_0	$\varphi_0^{[0]}$		$\boxed{\varphi_0^{[2]}}$					
		$\varphi_0^{[1]}$		$\boxed{\varphi_0^{[3]}}$				
x_0	$\varphi_0^{[0]}$		$\varphi_0^{[2]}$		$\boxed{\boldsymbol{\varphi[x_0^3, x_1^0]}}$			
		$\varphi_0^{[1]}$		$\boldsymbol{\varphi[x_0^2, x_1^0]}$		$\boxed{\boldsymbol{\varphi[x_0^3, x_1^1]}}$		
x_0	$\varphi_0^{[0]}$		$\boldsymbol{\varphi[x_0^1, x_1^0]}$		$\boldsymbol{\varphi[x_0^2, x_1^1]}$		$\boxed{\boldsymbol{\varphi[x_0^3, x_1^2]}}$	
		$\boldsymbol{\varphi[x_0^0, x_1^0]}$		$\boldsymbol{\varphi[x_0^1, x_1^1]}$		$\boldsymbol{\varphi[x_0^2, x_1^2]}$		$\boxed{\boldsymbol{\varphi[x_0^3, x_1^2, x_2^0]}}$
x_1	$\varphi_1^{[0]}$		$\boldsymbol{\varphi[x_0^0, x_1^1]}$		$\boldsymbol{\varphi[x_0^1, x_1^2]}$		$\boldsymbol{\varphi[x_0^2, x_1^2, x_2^0]}$	
		$\varphi_1^{[1]}$		$\boldsymbol{\varphi[x_0^0, x_1^2]}$		$\boldsymbol{\varphi[x_0^1, x_1^2, x_2^0]}$		
x_1	$\varphi_1^{[0]}$		$\varphi_1^{[2]}$		$\boldsymbol{\varphi[x_0^0, x_1^2, x_2^0]}$			
		$\varphi_1^{[1]}$		$\boldsymbol{\varphi[x_1^2, x_2^0]}$				
x_1	$\varphi_1^{[0]}$		$\boldsymbol{\varphi[x_1^1, x_2^0]}$					
		$\boldsymbol{\varphi[x_1^0, x_2^0]}$						
x_2	$\varphi_2^{[0]}$							

Table III.6. Table of generalized divided differences. The computation is in bold font and the Hermite's polynomial coefficients are boxed.

Hence, the Hermite's polynomial will be

$$\begin{aligned} P(x) = \varphi_0^{[0]} &+ \varphi_0^{[1]}(x - x_0) + \varphi_0^{[2]}(x - x_0)^2 + \varphi_0^{[3]}(x - x_0)^3 \\ &+ \varphi[x_0^3, x_1^0](x - x_0)^4 + \varphi[x_0^3, x_1^1](x - x_0)^4(x - x_1) \\ &+ \varphi[x_0^3, x_1^2](x - x_0)^4(x - x_1)^2 + \varphi[x_0^3, x_1^2, x_2^0](x - x_0)^4(x - x_1)^3. \end{aligned}$$

As the Hermite's polynomial P of φ satisfies $P^{(j)}(x_k) = \varphi^{(j)}(x_k)$. One wonder if P can be used locally for approximate the Taylor expansion of φ. That is,

$$\varphi(t_0 + h) \approx P^{[0]}(t_0) + P^{[1]}(t_0)h + P^{[2]}(t_0)h^2 + P^{[3]}(t_0)h^3 + \cdots .$$

A first naive approach is to consider iteratively the same initial problem but with lesser columns, i.e.

x	$\varphi^{[i]}$	$\varphi^{[i+1]}$	$\cdots$	
x_0	$\varphi_0^{[i]}$	$\varphi_0^{[i+1]}$	$\cdots$	$\varphi_0^{[m_0]}$
$\vdots$	$\vdots$	$\vdots$		$\vdots$
x_n	$\varphi_n^{[i]}$	$\varphi_n^{[i+1]}$	$\cdots$	$\varphi_n^{[m_n]}$

with $i > 0$ and $\varphi^{[j]} = \dfrac{\varphi^{(j)}}{j!}$.

At each iteration, one must compute the generalized divided differences and then evaluate at t_0. It is quite clear that each computed table can not be reused. Indeed, returning to our example:

x_0	$\boxed{\varphi_0^{[1]}}$				
		$\boxed{\varphi_0^{[2]}}$			
x_0	$\varphi_0^{[1]}$		$\boxed{\varphi_0^{[3]}}$		
		$\varphi_0^{[2]}$		$\boxed{\boldsymbol{\varphi^{[1]}[x_0^2, x_1^0]}}$	
x_0	$\varphi_0^{[1]}$		$\boldsymbol{\varphi^{[1]}[x_0^1, x_1^0]}$		$\boxed{\boldsymbol{\varphi^{[1]}[x_0^2, x_1^1]}}$
		$\boldsymbol{\varphi^{[1]}[x_0^0, x_1^0]}$		$\boldsymbol{\varphi^{[1]}[x_0^1, x_1^1]}$	
x_1	$\varphi_1^{[1]}$		$\boldsymbol{\varphi^{[1]}[x_0^0, x_1^1]}$		
		$\varphi_1^{[2]}$			
x_1	$\varphi_1^{[1]}$				

Table III.7. Derived generalized divided differences. The computation is in bold font and the Hermite's polynomial coefficients are boxed.

Hence, the Hermite's polynomial for $\varphi^{[1]}$ will be

$$\begin{aligned} P(x) = \varphi_0^{[1]} + \varphi_0^{[2]}(x - x_0) + \varphi_0^{[3]}(x - x_0)^2 \\ + \varphi^{[1]}[x_0^2, x_1^0](x - x_0)^3 + \varphi^{[1]}[x_0^2, x_1^1](x - x_0)^3(x - x_1). \end{aligned} \tag{3.3}$$

Clearly, each computation in Table **III.7** does not coincide with any computation of Table **III.6** (in a general case).

A second approach is to derivative the Hermite's polynomial and to evaluate it at the desired point t_0. One have two ways:

i. Symbolic derivation.
ii. Numerical derivation.

In the first case, we will have a general expression for each derivative of the Hermite's polynomial and then we should evaluate it at t_0.
In the second case, we can obtain directly the derivative at t_0. In order to do that, we shall use Automatic Differentiation because any polynomial can be expressed in an evaluation procedure. First of all, we need an efficient evaluation of a polynomial at a point. It is called the Horner's algorithm and has a linear complexity $O(N)$.

Horner's method. Given the coefficient list $(a_1, \ldots, a_n)$ of a univariate polynomial, i.e.

$$a_1 + \cdots + a_n x^n.$$

A Horner's evaluation algorithm has been codified in the Script **III.3** whose complexity is linear.

```
double horner_method(int n, double * const a, double x)
{
  typeof(n) i;
  typeof(x) s;

  s = a[n-1];
  for (i = n-2; i >= 0; i --)
    {s = s * x + a[i];}
  return s;
}
```

Script III.3. Horner's method codification in C.

A suitable modification can be considered for the evaluation of the Hermite's polynomial (3.2) at y. Indeed, let us define

$$\begin{aligned} m &= (m_0, \dots, m_{n-1}) \\ N &= \sum_{k=0}^{n-1} (m_k + 1) \\ x &= (x_0, \dots, x_{n-1}) \\ a &= \big(\varphi[x_0], \dots, \varphi[x_0^{m_0}, \dots, x_{n-1}^{m_{n-1}}]\big). \end{aligned}$$

Then a possible implementation is in the Script **III.4**.

```
double hermite_horner_method(int N, int n, int * const m, double *
    const a, double * const x, double y)
{
  typeof(n) k;
  typeof(N) i = N-2;
  typeof(y) h, s = a[N-1];

  for (k = n-1; k >= 0; k --)
  {
    h = y - x[k];
    for (N -= m[k]; i >= N; i --)
      {s = s * h + a[i];}
  }
  return s;
}
```

Script III.4. Horner's method codification for Hermite's polynomial in C.

For instance, the polynomial on the Equation (3.3) is expressed as

$$\big(\big(\big(\varphi^{[1]}[x_0^2, x_1^1](x - x_1) + \varphi^{[1]}[x_0^2, x_1^0]\big)(x - x_0) + \varphi_0^{[3]}\big)(x - x_0) + \varphi_0^{[2]}\big)(x - x_0) + \varphi_0^{[1]}.$$

Now the idea is to overload the arithmetic using Theorem **III.11**. In our particular case, the only issue that may become a problem is the product `s*h`, but it is not a trouble because `h` only have two components in its Taylor's expansion. Thus, the discrete convolution at most contains 2 sums in each order.

Let us see that the two approaches proposed are not exactly the same. For instance, given $\varphi(x) = (x+1)^4$ and the respective tables of derived generalized divided differences:

0	1			
		4		
0	1		**11**	
		15		**6**
1	16		**17**	
		32		
1	16			

Table III.8. Hermite derivatives approach.

0	4	
		15
1	32	

Table III.9. Naive approach.

From the left hand side we obtain the interpolated polynomial $P_1(x) = 1+4x+11x^2+6x^2(x-1)$ and from the right hand side $P_2(x) = 4 + 15x$. An immediate computation gives us

$$P_1'(x) = 4 + 22x + 12x(x-1) + 6x^2.$$

Clearly the degrees of P_1' and P_2 are not the same. Therefore each approach has a different error in its computation and apparently the derivation of the Hermite interpolated polynomial should be better, as one observe in Figure **III.2**.

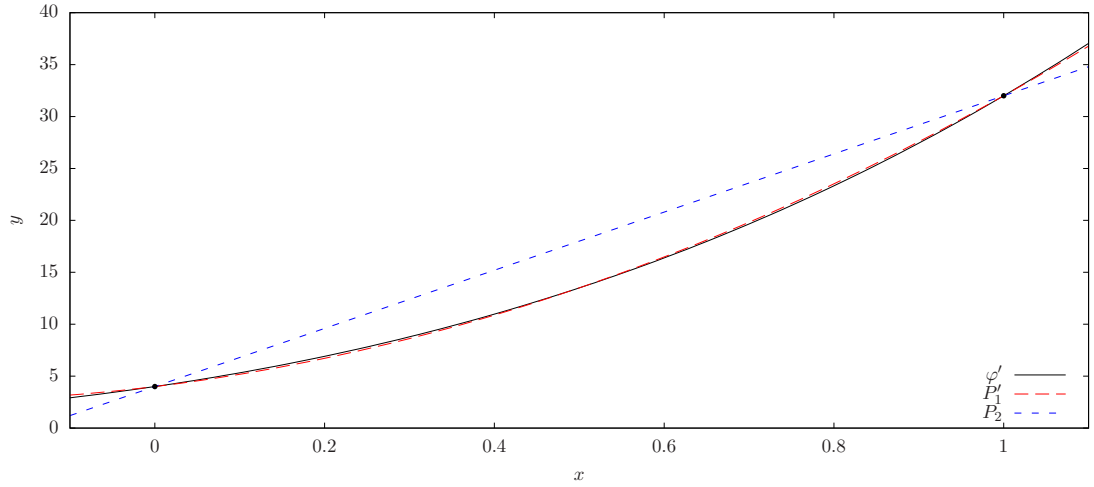

Figure III.2. The error of an intermediate point of P_1' will considerably be smaller than P_2.

5.1. Problem in higher orders. Let us consider $\varphi\colon \mathbb{R} \to \mathbb{R}$ be a function so that the next table of values has been given using Theorem **III.11**:

x	$\varphi^{[0]}$	$\varphi^{[1]}$	$\cdots$	$\varphi^{[m]}$
$x_0 = -\frac{1}{2}$	$\varphi^{[0]}_{-\frac{1}{2}}$	$\varphi^{[1]}_{-\frac{1}{2}}$	$\cdots$	$\varphi^{[m]}_{-\frac{1}{2}}$
$y = 0$	$\varphi^{[0]}_{0}$	$\varphi^{[1]}_{0}$	$\cdots$	$\varphi^{[m]}_{0}$
$x_1 = \frac{1}{2}$	$\varphi^{[0]}_{\frac{1}{2}}$	$\varphi^{[1]}_{\frac{1}{2}}$	$\cdots$	$\varphi^{[m]}_{\frac{1}{2}}$

with $\varphi^{[j]} = \dfrac{\varphi^{(j)}}{j!}$.

Let P be the Hermite interpolated polynomial on x_0 and x_1. We want to compare the error with the derivatives of P and φ at an intermediate point y. That is,

$$|P^{[j]}(y) - \varphi^{[j]}(y)|.$$

Moreover, let us introduce another value to be compared. As we have the truncated Taylor expansion at x_0, i.e.

$$\varphi(x_0 + h) = \varphi^{[0]}(x_0) + \varphi^{[1]}(x_0)h + \cdots + \varphi^{[m]}(x_0)h^m \tag{3.4}$$

Taking $h = y - x_0$, we obtain $\widetilde{\varphi}^{[j]}(y)$ using automatic differentiation (again using the Horner method in (3.4) with respect to h). Hence, the errors that we want to compare are

$$|P^{[j]}(y) - \varphi^{[j]}(y)| \qquad \text{and} \qquad |\widetilde{\varphi}^{[j]}(y) - \varphi^{[j]}(y)|$$

with $y \in [x_0, x_1]$.
The chosen test functions have been:

$$\varphi(x) = \frac{1}{1 + 25x^2}, \qquad \varphi(x) = \cos(x) \quad \text{and} \quad \varphi(x) = e^x.$$

As we can observe in Figure **III.3**, when m is near to 10 the derivatives of the Hermite polynomial are better than the derivatives in the Taylor expansion and there is no difference if $(x_0, x_1) = (-\frac{1}{2}, \frac{1}{2})$ or $(x_0, x_1) = (\frac{1}{2}, -\frac{1}{2})$. However, if $m > 10$, then some strange effects appear. For instance, the error of for $\varphi(x) = \exp(x)$ at $y = 0$ increases a lot. One can try to execute with a higher value of m but all those effects arise in an exaggerate way.

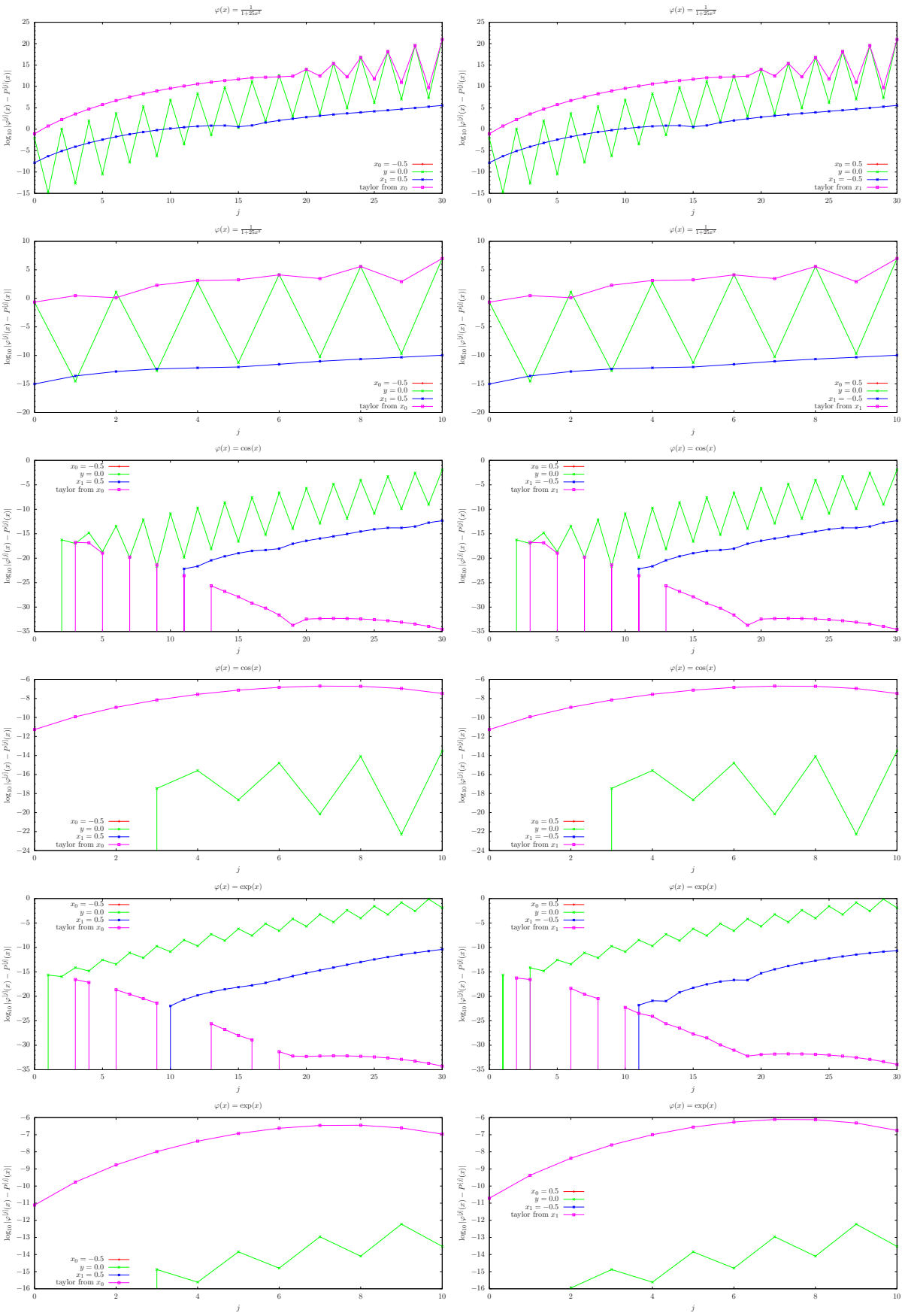

Figure III.3. Comparison of the derivatives of the Hermite's polynomial for x_0, x_1 and derivatives of the Taylor expansion for some test functions.

PROPOSITION III.14. *The table of generalized divided differences of two points $x_0 < x_1$ of the map $\varphi(x) = e^x$ always has positive terms $\varphi[x_0^i, x_1^j] > 0$ if $j = 0, 1$.*

PROOF. Let us suppose that $x_1 - x_0 = 1$. The data of the initial table are $e^{x_0}\frac{1}{j!}$ and $e^{x_1}\frac{1}{j!}$ with $j = 0, \dots, m$. By the Mean Value Theorem,

$$\begin{aligned}
\varphi[x_0^0, x_1^0] &= e^{x_1} - e^{x_0} = e^{\alpha_{00}} & \varphi[x_0^0, x_1^1] &= e^{x_1} - e^{\alpha_{00}} = e^{\alpha_{01}} \\
\varphi[x_0^1, x_1^0] &= e^{\alpha_{00}} - e^{x_0} = e^{\alpha_{10}} & \varphi[x_0^1, x_1^1] &= e^{\alpha_{01}} - e^{\alpha_{10}} = e^{\alpha_{11}} \\
\varphi[x_0^2, x_1^0] &= e^{\alpha_{10}} - e^{x_0 - \log 2!} = e^{\alpha_{20}} & \varphi[x_0^2, x_1^1] &= e^{\alpha_{11}} - e^{\alpha_{20}} = e^{\alpha_{21}} \\
\varphi[x_0^3, x_1^0] &= e^{\alpha_{20}} - e^{x_0 - \log 3!} = e^{\alpha_{30}} & \varphi[x_0^3, x_1^1] &= e^{\alpha_{21}} - e^{\alpha_{30}} = e^{\alpha_{31}} \\
&\vdots & &\vdots \\
\varphi[x_0^m, x_1^0] &= e^{\alpha_{(m-1)0}} - e^{x_0 - \log m!} = e^{\alpha_{m0}} & \varphi[x_0^m, x_1^1] &= e^{\alpha_{(m-1)1}} - e^{\alpha_{m0}} = e^{\alpha_{m1}}
\end{aligned}$$

Hence there is a set $\{\alpha_{ij} \colon i = 0, \dots, m \text{ and } j = 0, 1\}$ so that

$$\begin{aligned}
x_0 \leq \alpha_{00} &\leq x_1 \\
x_0 - \log i! \leq \alpha_{i0} &\leq \alpha_{(i-1)0} \qquad\qquad i = 1, \dots, m
\end{aligned}$$

and

$$\begin{aligned}
\alpha_{00} \leq \alpha_{01} &\leq x_1 \\
\alpha_{i0} \leq \alpha_{i1} &\leq \alpha_{(i-1)1} \qquad\qquad i = 1, \dots m.
\end{aligned}$$

Therefore $\varphi[x_0^i, x_1^j] > 0$ whenever $i = 0, \dots, m$ and $j = 0, 1$. □

According to the previous Proposition, the term of $\varphi[x_0^m, x_1^j]$ has to be strictly positive for $j = 0, 1$. But a numerical digit cancellation appears when the number of derivatives m is big enough. We have plotted the Hermite polynomial term $\varphi[x_0^m, x_1^j]$ for different values of m in Figure **III.4** where $(x_0, x_1) = (-\frac{1}{2}, \frac{1}{2})$.
Summing up, the derivatives of the Hermite's interpolated polynomial might be an appropriate method if, and only if, the number of the initial derivatives is not large.

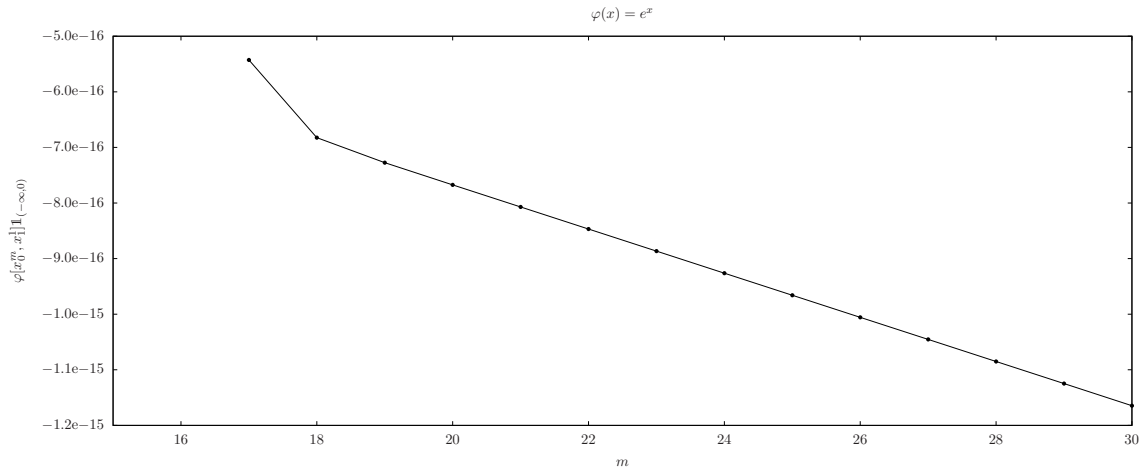

Figure III.4. Plot of negative values of $\varphi[x_0^m, x_0^1]$ for different values of m.

6. A computer-assisted proof

A computer-assisted proof is a "mathematical" proof that has been at least partially generated by computer. The aim of that section is to show how the Hermite's interpolation really has computational drawbacks. Firstly, let us recall the notion of floating number in order to introduce a brief introduction in interval arithmetic.

6.1. Floating-point number. A floating-point number, or float for short, is an arbitrary precision significand (also called mantissa) with a limited precision exponent. For instance, the double-precision floating-point number occupies 8 bytes in computer memory and according to the IEE 754 standard has a format:

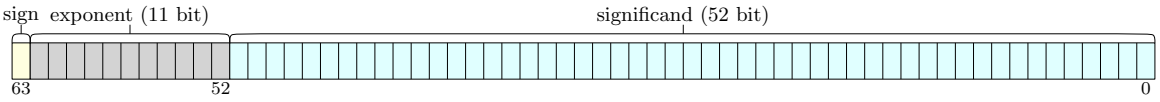

Figure III.5. IEEE 754 double-precision binary floating-point format.

Thus, the precision is the number of bits used to represent the significand of a floating-point number. The IEEE754 double-precision format has a precision of 53 bits, i.e. approximately 16 decimal digits because $53 \log_{10} 2 \approx 15.9546$.

6.2. Interval computations. The basic principle of interval arithmetic consists in enclosing every number by an interval containing it and being representable by machine numbers. Thus interval arithmetic is an arithmetic defined on sets of intervals denoted by $\mathbb{IR}$, rather than sets of real numbers.

The main operations in $\mathbb{IR}$ are defined as follows: If $\boldsymbol{x} = [\underline{x}, \overline{x}]$ and $\boldsymbol{y} = [\underline{y}, \overline{y}]$ are closed connected sets of real numbers, then

$$\boldsymbol{x} \operatorname{op} \boldsymbol{y} = \{x \operatorname{op} y \colon x \in \boldsymbol{x} \text{ and } y \in \boldsymbol{y}\} \qquad \text{for } \operatorname{op} \in \{+, -, \cdot, \div\}.$$

If $\varphi \colon \mathbb{R} \to \mathbb{R}$ is a continuous function, then $\boldsymbol{\varphi} \colon \mathbb{IR} \to \mathbb{IR}$ denotes an extension to the real intervals and it has to verify

$$\varphi(\boldsymbol{x}) \subset \boldsymbol{\varphi}(\boldsymbol{x}).$$

Clearly, if φ is monotone, an extension $\boldsymbol{\varphi}$ is quite immediate. In particular,

$$\begin{array}{c} \boldsymbol{e} \colon \mathbb{IR} \longrightarrow \mathbb{IR} \\ [\underline{x}, \overline{x}] \longmapsto [e^{\underline{x}}, e^{\overline{x}}] \end{array} \quad \text{extends} \quad \begin{array}{c} e \colon \mathbb{R} \longrightarrow (0, +\infty) \\ x \longmapsto e^x. \end{array}$$

$\boldsymbol{e}$ is the most natural and, in that case, is the tightest possible extension. But it is not unique.

6.3. The MPFR and MPFI libraries. Let us introduce two different libraries written in C which allow us to do computations with multiple precision and to change the arithmetic to an interval arithmetic.

6.3.1. *Multiple Precision Floating-Point Reliable.* The MPFR is a portable library written in C for arbitrary precision arithmetic on floating-point numbers. It aims to provide a class of floating-point numbers with precise semantics. The main characteristics of MPFR are

i. Its code is portable.
ii. The precision in bits can be set exactly to any valid value for each variables.
iii. It provides the four rounding modes from the IEEE 754-1985 standard.

In particular, with a precision of 53 bits, MPFR is able to exactly reproduce all computations with double-precision machine floating-point numbers.

6.3.2. *Multiple Precision Floating Interval.* The MPFI is intended to be a portable library written in C for arbitrary precision interval arithmetic with intervals represented using MPFR reliable floating-point numbers. It is based on the GNU MP library and on the MPFR library. The purpose of an arbitrary precision interval arithmetic is on the one hand to get guaranteed results, thanks to interval computation, and on the other hand to obtain accurate results, thanks to multiple precision arithmetic.

6.4. Lack of accuracy of the Hermite's interpolation method. The MPFR and MPFI libraries provides a powerful tool in order to implement a computer-assisted proof.
Let us focus on the Hermite's interpolation inaccuracy explained in Section **5**. The example that we are going to study is the exponential function $\varphi(x) = e^x$ at the points $x_0 = -\frac{1}{2}$ and $x_1 = \frac{1}{2}$. Since $x_1 - x_0 = 1$, the table of generalized divided differences is actually a table of generalized differences. And all the normalized derivatives $\varphi^{[j]}(x_i)$ are straightforward using Table **III.3**.
Let us suppose that our data has size n, i.e. $\varphi^{[0]}(x_i), \ldots, \varphi^{[n]}(x_i)$. Computing the table of generalized differences as in Table **III.6**, we obtain the coefficients of the Hermite's polynomial

$$\varphi^{[0]}(x_0), \ldots, \varphi^{[n]}(x_0), \boldsymbol{\varphi[x_0^n, x_1^0]}, \ldots, \boldsymbol{\varphi[x_0^n, x_1^{n-1}]}. \tag{3.5}$$

Firstly, (3.5) is computed with a double precision, i.e. IEEE754 double-precision floating-point number.
Secondly, (3.5) is computed with an interval arithmetic with a prefixed precision.
In Figure **III.6** has been plotted the error of the computed double precision and the multiple precision interval. That means, if $[a, b]$ is the interval and c is the double, the error will be bounded by

$$\text{error} \leq \max\{|a - c|, |b - c|\}.$$

The conclusions are:

- The error behaviour of the first $n + 1$ values is as one expects because it is just the input data which is the Taylor expansion of an analytic function.
- The rest has an extremely awful accuracy.

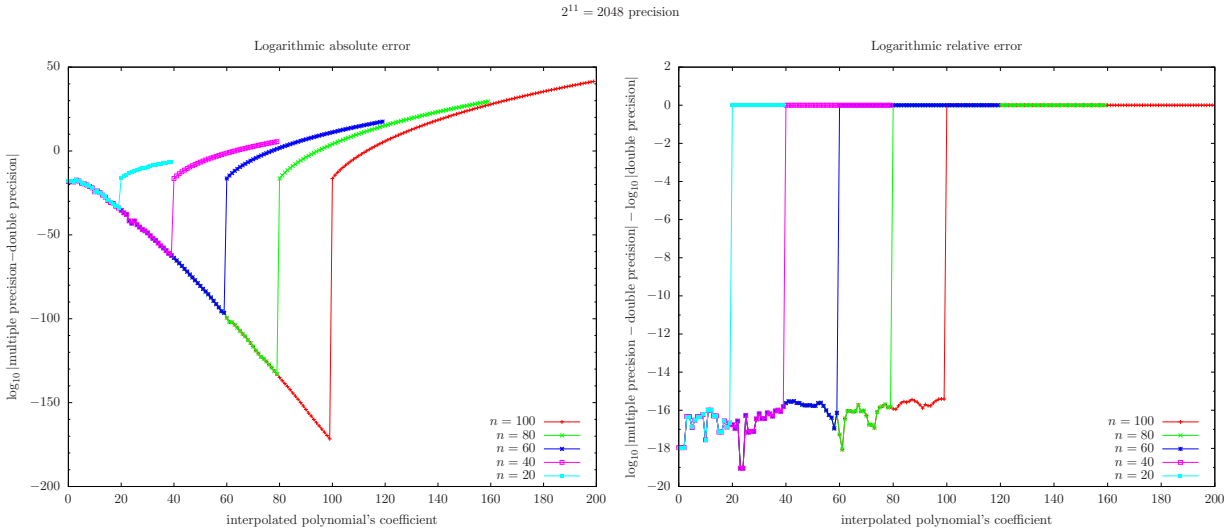

Figure III.6. Plot of the error of the Hermite polynomial's coefficients with a double precision computation and a multiple precision interval computation for different input data sizes. Computation time: 0.04 seconds.

In addition, the biggest interval diameter of (3.5) are in the Table **III.10**.

n	the biggest interval diameter
100	2.106078674434713e − 558
80	2.143634287380493e − 570
60	2.254879986115886e − 582
40	2.519891634836048e − 594
20	3.273297938271279e − 606

Table III.10. The biggest interval diameter for different values of n with 2048 precision.

CHAPTER IV

Integrators of Delay Differential Equations with a constant delay

Let us begin considering a Delay Differential Equation with a constant delay expressed by

$$\dot{x}(t) = f(t, x(t), x(t-\tau))$$

with τ is a fixed positive real number. If $\tau = 0$, we have an ordinary differential equation and integration methods can be applied. The aim of the Chapter is to translate some of those methods to delayed methods. That is, to impose delayed conditions and how they raise again the integration method.
First of all, we must fix the problem to be studied. It is called the Initial Value Problem and it suggests to find a solution $x(t_0, u)$ verifying

$$\begin{cases} \dot{x}(t) = f(t, x(t), \varphi(t)) \\ x_{t_0} \equiv u \end{cases} \qquad \text{with } \varphi(t) = x(t-\tau). \tag{4.1}$$

A first observation is that the map u will be discrete in a table of values whose size will be depend on each problem.

1. Euler's method

The first integration method of an Ordinary Differential Equation is the Euler's method.

Ordinary case. Fixed a step size h and an initial condition (t_0, x_0). Then

$$\begin{aligned} x_{n+1} &= x_n + hf(t_n, x_n) \\ t_{n+1} &= t_n + h \end{aligned}$$

is its iterative Euler's scheme. By Taylor's expansion,

$$x(t_0 + h; t_0, x_0) = x_0 + f(t_0, x_0)h + \frac{h^2}{2} D_t f(t_0, x_0) + \frac{h^2}{2} D_x f(t_0, x_0)\big(f(t_0, x_0)\big) + \cdots \tag{4.2}$$

So the error of the Euler's method is $O(h^2)$ and one says that it is a method of first order because it is exact up to the first order in h.

Delayed case. Fixed a step size $h = \frac{\tau}{N}$ and an initial condition (t_0, u). The iterative scheme is

$$\begin{aligned} x_{n+1} &= x_n + hf(t_n, x_n, \varphi(t_n)) \\ t_{n+1} &= t_n + h. \end{aligned}$$

It is also a first order method with error $O(h^2)$. Although its Taylor's expansion is now

$$\begin{aligned} x(t_0, u)(t_0 + h) = x_0 &+ f(t_0, x_0, \varphi(t_0))h + \frac{h^2}{2} D_t f(t_0, x_0, \varphi(t_0)) \\ &+ \frac{h^2}{2} D_x f(t_0, x_0, \varphi(t_0))(f(t_0, x_0, \varphi(t_0))) \\ &+ \frac{h^2}{2} D_\varphi f(t_0, x_0, \varphi(t_0))\big(f(t_0 - \tau, \varphi(t_0), \varphi(t_0 - \tau)\big) + \cdots \end{aligned}$$

The imposed condition on the step size is because we want to cross exactly at multiples of τ. The reason of that is because $\dot{x}$ only represents the right-hand derivative of x.

2. Runge-Kutta family of methods

A first attempt to improve the Euler's method is to compute the next iteration in two stages. Let us explain it for the first iteration:

i. Compute $f(t_0, x_0)$ and obtain $x_1^* = x_0 + hf(t_0, x_0)$.
ii. Compute $f(t_0 + h, x_1^*)$.
iii. Compute a better approximation of x_1^* by the average

$$x_1 = x_0 + h\frac{f(t_0, x_0) + f(t_0 + h, x_1^*)}{2}$$

and $t_1 = t_0 + h$.

More general,

i. Compute $f(t_0, x_0)$ and obtain $x_1^* = x_0 + a_1 hf(t_0, x_0)$.
ii. Compute $f(t_0 + h, x_1^*)$.
iii. Compute a better approximation of x_1^* by

$$x_1 = x_0 + h\big(b_{11}f(t_0, x_0) + b_{12}f(t_0 + h, x_1^*)\big)$$

and $t_1 = t_0 + h$.

where now the values a_1, b_{11} and b_{12} are parameters to determine. As we want to be more accurate, we impose a second order method. That is, the coefficients of (4.2), so $a_1 = 1$ and $b_{11} = b_{12} = \frac{1}{2}$.

Ordinary case. The generalization of the previous idea gives us the Runge-Kutta family. The idea is that if we want to pass from (t_0, x_0) to (t_1, x_1), we can try to compute f at different auxiliary points and then to use a linear combination of them in order to predict x_1.

$$\begin{aligned}
k_1 &= f(t_0 + a_1 h, x_0 + h\sum_{j=1}^{s} b_{1j}k_j)\\
&\vdots\\
k_s &= f(t_0 + a_s h, x_0 + h\sum_{j=1}^{s} b_{sj}k_j)\\
x_1 &= x_0 + h\sum_{j=1}^{s} c_j k_j\\
t_1 &= t_0 + h
\end{aligned}$$

Table IV.1. Runge-Kutta with s stages and h fixed step size.

We need $k_1, \ldots, k_s$ in the Table **IV.1** and if k_i wants to be computed, then $k_1, \ldots, k_s$ will be needed. Hence, a system of equations has to be solved in order to obtain $k_1, \ldots, k_s$. It can be done using Newton's method. Although the issue is avoided in an explicit version, i.e. the computation of k_i is conditioned to known $k_1, \ldots, k_{i-1}$.
Typically the Runge-Kutta coefficients are given in a Butcher's Tableau:

a_1	b_{11}	$\cdots$	b_{1s}
$\vdots$	$\vdots$		$\vdots$
a_s	b_{s1}	$\cdots$	b_{ss}
	c_1	$\cdots$	c_s

Table IV.2. Implicit Butcher's Tableau.

a_1					
a_2	b_{21}				
a_3	b_{31}	b_{32}			
$\vdots$	$\vdots$	$\vdots$	$\ddots$		
a_s	b_{s1}	b_{s2}	$\cdots$	$b_{s(s-1)}$	
	c_1	c_2	$\cdots$	c_{s-1}	c_s

Table IV.3. Explicit Butcher's Tableau.

2.0.1. *RK4.* A very popular explicit Runge-Kutta method with 4 stages and order 4 is the well-known RK4 whose Butcher's Tableau is:

0				
$\frac{1}{2}$	$\frac{1}{2}$			
$\frac{1}{2}$	0	$\frac{1}{2}$		
1	0	0	1	
	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$

$$
\begin{aligned}
k_1 &= f(t_0, x_0)\\
k_2 &= f(t_0 + \tfrac{h}{2}, x_0 + \tfrac{h}{2}k_1)\\
k_3 &= f(t_0 + \tfrac{h}{2}, x_0 + \tfrac{h}{2}k_2)\\
k_4 &= f(t_0 + h, x_0 + hk_3)\\
x_1 &= x_0 + \tfrac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)\\
t_1 &= t_0 + h.
\end{aligned}
$$

Table IV.4. RK4 equations.

Delayed case. The integration of a Delay Differential Equation with a constant delay by Delayed Runge-Kutta method is expressed in Table **IV.5**. It is a closed formulation to Table **IV.1**, the differences are in the terms $\varphi(t_0 + a_i h)$. Of course, a step size $h = \frac{\tau}{N}$ will also be considered.

$$
\begin{aligned}
k_1 &= f(t_0 + a_1 h, x_0 + h\sum_{j=1}^{s} b_{1j}k_j, \varphi(t_0 + a_1 h))\\
&\vdots\\
k_s &= f(t_0 + a_s h, x_0 + h\sum_{j=1}^{s} b_{sj}k_j, \varphi(t_0 + a_s h))\\
x_1 &= x_0 + h\sum_{j=1}^{s} c_j k_j\\
t_1 &= t_0 + h
\end{aligned}
$$

Table IV.5. Delayed Runge-Kutta with s stages and $h = \frac{\tau}{N}$ step size.

As the initial condition u is a function, which has been discretised, some values $\varphi(t_0 + a_i h)$ may be unknown. So it must be interpolated by the known data. It is just that fact which produces a more complicated implementation, at the same time that it adds other error source. Let us explain the delayed Runge-Kutta 4:

2.0.2. *DRK4.* The iterative scheme is

$$
\begin{aligned}
k_1 &= f(t_0, x_0, \varphi(t_0))\\
k_2 &= f(t_0 + \tfrac{h}{2}, x_0 + \tfrac{h}{2}k_1, \varphi(t_0 + \tfrac{h}{2}))\\
k_3 &= f(t_0 + \tfrac{h}{2}, x_0 + \tfrac{h}{2}k_2, \varphi(t_0 + \tfrac{h}{2}))\\
k_4 &= f(t_0 + h, x_0 + hk_3, \varphi(t_0 + h))\\
x_1 &= x_0 + \tfrac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)\\
t_1 &= t_0 + h.
\end{aligned}
\tag{4.3}
$$

In order to understand the difficulties of the method, let us consider specific values. For instance, $\tau = 1$, $N = 4$, $h = \frac{1}{N}$ and $t_0 = 0$. Then the initial condition u is discretised with 4 values u_0, u_1, u_2 and u_3. Thus, the input is two vectors called pt and px of size $N + 1$ and $n(N + 1)$ respectively. In fact, as the step h is constant in all the process one may only use px.
The first iteration is the computation of $t = 0$. Hence

- In k_1 the value at $t = -1$ is needed, which is known.
- In k_2 the value at $t = \frac{1}{8} - 1$ is needed, which is unknown.
- In k_3 the value at $t = \frac{1}{8} - 1$ is needed, which is unknown.
- In k_4 the value at $t = \frac{1}{4} - 1$ is needed, which is known.

Therefore, in DRK4 we need to compute the interpolated polynomial one time in each iteration. Clearly, the points used in that interpolation have to be $\leq N$ because the value at $t = 0$ is still unknown in px. That value at $t = 0$ ought to be stored in px too because we want to interpolate in k_2 and k_3 at $t = \frac{1}{4}$. A representation of that previous explanation can be found in Figure **IV.1**.

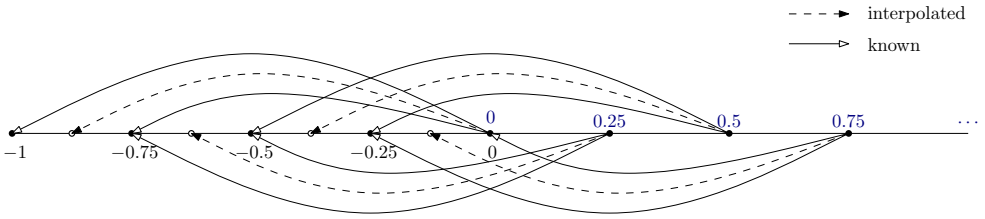

Figure IV.1. Points used in the computation of the first interval. The dashed arrows require an interpolation procedure of the known data. The boundary of the intervals should be shared in the known data and in the unknown data.

2.1. Comments for an implementation. As we have already commented the boundary of each interval has to be shared. The input data allocation have been represented in Figure **IV.2** and the new allocation that we have implemented in Figure **IV.3**.

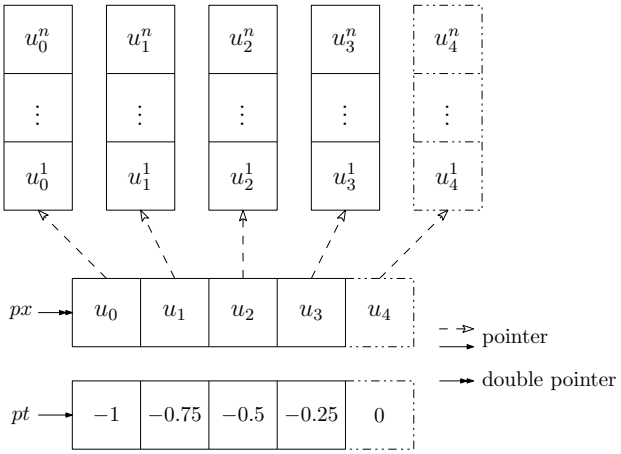

Figure IV.2. The input data with a double pointer px and a single pointer pt.

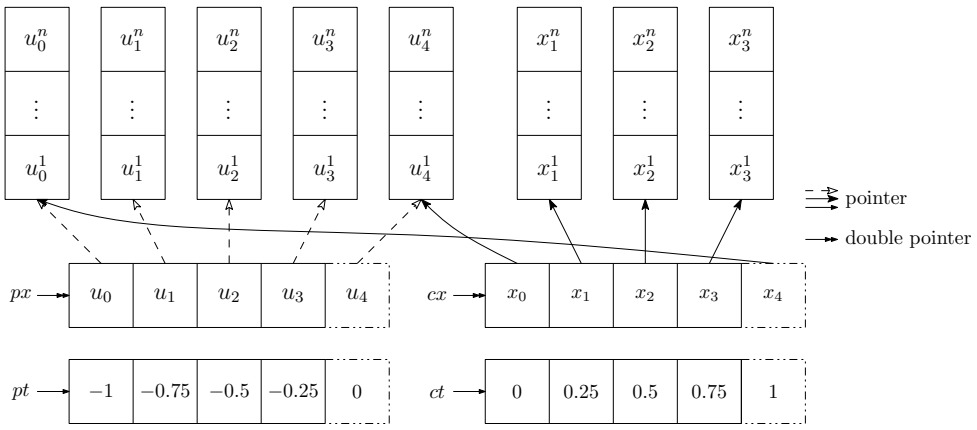

Figure IV.3. Previous interval allocation px, pt and current interval allocation cx, ct for a Delayed Runge-Kutta implementation.

The idea is to use a double pointer so that the first and the last pointers are going to be pointed to the opposite. It is a kind of relation called "head-tail and tail-head".
In fact, in that specific method with constant step h, the vectors pt and ct are only necessary for the interpolation step. As we are going to see in the next sections if the the step h is non constant, the time vectors will have an extra role. Thus, the struct used in our implementation is in Script **IV.1**. Of course, the number of points P used in the interpolation procedure will be a parameter given by the user.

```
struct dde_rk
{
  /**
   * n          Dimension of our problem
   * njets      Number of jets
   * idx        Index of the last jet used
   * pt         Previous time array
   * px         Previous x's array
   * ct         Current time array
   * cx         Current x's array
   */
  int n, njets, idx;

  double *pt, **px, *ct, **cx;
};
```

Script IV.1. Struct for an implementation of Delayed Runge-Kutta method in C.

After the memory allocation control in our implementation we propose the declaration of `drk4` function in Script **IV.2** whose arguments are:

- ◇ Struct `dde_rk` initialized.
- ◇ Dimension n of the problem.
- ◇ Pointer to the time t whose next value will be $t + h$.
- ◇ Pointer to the current x.
- ◇ Constant step h.
- ◇ Delay r.
- ◇ Pointer to the function of our Delay Differential Equation.
- ◇ Pointer to the interpolation function.
- ◇ Pointer to the k's and the number of k's used in (4.3).
- ◇ Auxiliary pointer for the interpolated polynomial with `inter_len` points.

```
int drk4(struct dde_rk *const dde,
         int n,
         double *const t,
         double *const x,
         double h,
         double r,
         void (*F)(int, int, double *const, double, double *const,
             double, double *const, double **const),
         int (*inter)(int, int, double, double *const, double **const,
              double *const, double *const),
         int nK, double **const K,
         int inter_len, double *const inter_pol);
```

Script IV.2. Declaration of the implemented function for the Delayed Runge-Kutta 4 in C.

2.2. Example. The first example that we have considered is the Delay Differential Equation with a constant delay

$$\dot{x}(t) = x(t-1).$$

As we want to be able to compare the error of the method, let us do a suitable modification in order to have a well-known solution. That is, let us introduce a new function $r(t)$

$$\dot{x}(t) = x(t-1) + r(t).$$

If we want $x(t) = \cos(t)$ to is the solution, we deduce that $r(t) = -\sin(t) - \cos(t-1)$. Thus, the Initial Value Problem becomes to

$$\begin{cases} \dot{x}(t) = x(t-1) - \sin(t) - \cos(t-1) \\ x_0(\tau) = \cos(-\tau) \end{cases} \tag{4.4}$$

with $0 \leq \tau \leq 1$. Since we are imposing the solution, the interpolation step can be done with the solution itself. A plot of that error has been plotted in Figure **IV.4**.

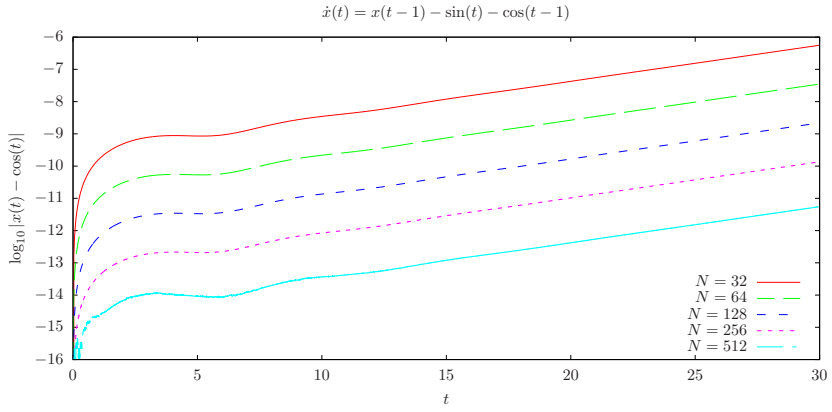

Figure IV.4. Delayed Runge Kutta 4 with step $h = \frac{1}{N}$ and without interpolation of (4.4).

We have two error sources:

- The Runge Kutta error which essentially depends on the constant step h.
- The interpolation error which essentially depends on the number of points used P.

We have also implemented two interpolation methods; the Newton interpolation method and the Lagrange interpolation method. In both case the result has been the same (as we expected). The plot with different values of h and P is in Figure **IV.5**.

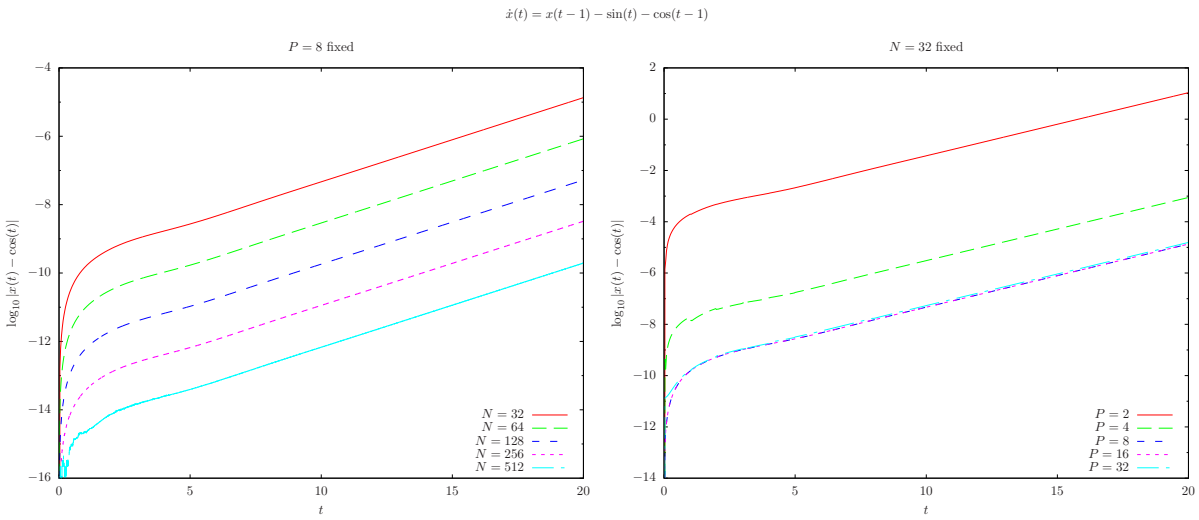

Figure IV.5. Error modifying $h = \frac{1}{N}$ in the left side and modifying P in the right side of (4.4).

3. Runge-Kutta-Fehlberg family of methods

Ordinary case. A Runge-Kutta-Fehlberg method is closed to a Runge-Kutta method with the difference that it allows to determine a suitable step to have a solution with a prearranged tolerance at each step. The idea is to compute two different approximations and then to try to estimate the step h for the next iteration.
Let us explain the Runge-Kutta-Fehlberg pq (typically $p < q$) with s stages.

- ⋆ Fixed a tolerance tol and values $h_{\min} \leq |h| \leq h_{\max}$.
- ⋆ Given a matrix $(b_{ij}) \in \mathbb{R}(s, s)$ and vectors $(a_1, \ldots, a_s)$, $(c_1, \ldots, c_{s_1})$ and $(d_1, \ldots, d_{s_2})$.
- ⋆ Given an initial point x_0 and an initial time t_0.
- Compute
$$k_i = f(t_0 + a_i h, x_0 + h \sum_{j=1}^{s} b_{ij} k_j) \qquad i = 1, \ldots, s.$$
- Compute approximations with respective orders p and q by
$$x_1^{(1)} = \sum_{i=1}^{s_1} c_i k_i$$
and
$$x_1^{(2)} = \sum_{i=1}^{s_2} d_i k_i.$$
- $\delta = \|x_1^{(1)} - x_1^{(2)}\|$.
- If $tol < \delta$, then
$$h \leftarrow 0.9h \left(\frac{tol}{\delta}\right)^{\frac{1}{q}}.$$
If $|h| \leq h_{\min}$, error message and exit.
Iterate the process with the new value h.
- If $\delta < tol$, then
$$x_0 \leftarrow x_0 + x_1^{(2)} \qquad \text{and} \qquad t_0 \leftarrow t_0 + h.$$
Moreover,
$$h \leftarrow 0.9h \min\left\{1.2, \left(\frac{tol}{\delta}\right)^{\frac{1}{q}}\right\}.$$
If $h_{\max} \leq |h|$, then
$$h \leftarrow \frac{h}{|h|} h_{\max}.$$
- Iterate until a final time.

Immediate comments of the previous procedure are: the Butcher's Tableau becomes to

a_1	b_{11}	$\cdots$	b_{1s}
$\vdots$	$\vdots$		$\vdots$
a_s	b_{s1}	$\cdots$	b_{ss}
	c_1	$\cdots$	c_{s_1}
	d_1	$\cdots$	d_{s_2}

Table IV.6. Implicit Butcher's Tableau.

a_1					
a_2	b_{21}				
a_3	b_{31}	b_{32}			
$\vdots$	$\vdots$	$\vdots$	$\ddots$		
a_s	b_{s1}	b_{s2}	$\cdots$	$b_{s(s-1)}$	
	c_1	c_2	$\cdots$	c_{s_1-1}	c_{s_1}
	d_1	d_2	$\cdots$	d_{s_2-1}	d_{s_2}

Table IV.7. Explicit Butcher's Tableau.

Each update of the step h will always be between $h_{\min}$ and $h_{\max}$ and the values 0.9 and 1.2 are only "security factors".

Delayed case. The delayed case for a Runge-Kutta-Fehlberg is exactly the same up to the computation of k_i's which is replaced by

$$k_i = f\big(t_0 + a_i h, x_0 + h\sum_{j=1}^{s} b_{ij}k_j, \varphi(t_0 + a_i h)\big) \qquad i = 1, \dots, s. \tag{4.5}$$

3.1. Comments for an implementation. The initial condition for a Delay Differential Equation has to be discretised. As in Delayed Runge-Kutta method some values $\varphi(t_0 + a_i h)$ may be unknown and we should interpolate the known data in order to obtain the unknown data. That fact generates a new error source.
A difference with the Delayed Runge-Kutta method is that the vectors for time pt and points px have not a fixed size. As a consequence, we must search information in our sorted stored data and we want to do it in an efficient way.

3.1.1. *Efficient search of a sorted array in C.* Given an element b and a vector $(a_1, \dots, a_n)$ so that $a_1 < \cdots < a_n$. We want to be able to find an index i such that

$$a_i \le b < a_{i+1}.$$

In order to avoid problems when $i = n$, let us assume that the initial vector is $(a_1, \dots, a_n, a_1)$. The programming language C has a function in its library `stdlib.h` called `bsearch`. That function implements a binary search of a sorted array whose declaration is

```
#include <stdlib.h>

void *bsearch(const void *key, const void *base,
              size_t nmemb, size_t size,
              int (*compar)(const void *, const void *));
```

According to the manual:

> **Description:** The `bsearch()` function searches an array of `nmemb` objects, the initial member of which is pointed to by `base`, for a member that matches the object pointed to by `key`. The size of each member of the array is specified by `size`.
>
> The contents of the array should be in ascending sorted order according to the comparison function referenced by `compar`. The `compar` routine is expected to have two arguments which point to the key object and to an array member, in that order, and should return an integer less than, equal to, or greater than zero if the `key` object is found, respectively, to be less than, to match, or be greater than the array member.
>
> **Return value:** The `bsearch()` function returns a pointer to a matching member of the array, or NULL if no match is found. If there are multiple elements that match the key, the element returned is unspecified.

It is well-known that the binary search of a sorted array of size n has a complexity $O(\log_2 n)$. Thus the problem is reduced to implement the pointer function `compar`.

```
int cmp(const void *b, const void *a) __attribute__((always_inline));
int cmp(const void *b, const void *a) {
  double t0, t, t1;
  t0 = ((typeof(&t0)) a)[0];
  t  = ((typeof(&t))  b)[0];
  t1 = ((typeof(&t1)) a)[1];

  if (t  < t0) {return -1;}
  if (t == t0) {return 0; }
  if (t1 <= t) {return 1; }
  return 0;
}
```

Script IV.3. `compar` function for `bsearch` function.

A possible implementation of that function can be found in Script **IV.3** whose first line is just an optimisation issue which says to the compiler `gcc` that the function has to be always inlined. Up to here, the posed problem has been solved. However, an extra optimisation can be done. Firstly, let us propose a new struct for a Delayed Runge-Kutta-Fehlberg method in Script **IV.4**. As we can observe we will use two new variables `plen` and `clen` in order to have a control of the used memory. Clearly, both values are going to be lesser than `njets`.

```
struct dde_rkf
{
  /**
   * n         Dimension of our problem
   * njets     Number of jets
   * plen      Length of the previous data
   * pidx      Index of the last jet used
   * clen      Length of the current data
   * pt        Previous time array
   * px        Previous x's array
   * ct        Current time array
   * cx        Current x's array
   */
  int n, njets, plen, pidx, clen;

  double *pt, **px, *ct, **cx;
};
```

Script IV.4. Struct for an implementation of Delayed Runge-Kutta-Fehlberg method in C.

Hence the memory allocation will be `n`·`njets` for px, `n`·(`njets`−1) for cx (see Section **3.1.2**) and `njets` for each pt and ct. The values `plen` and `clen` will be the real memory used in the previous interval (i.e. pt and px) and in the current interval (i.e. ct and cx).
Furthermore, the search only has to be in pt. The same index obtained will be the same index in px. But if we focus on that search, we realise that if we have obtained the index i, the possible next index in next searches will be $\geq i$. It is just that reason that the field `pidx` in Script **IV.4** has an extra role. In the one hand it saved the last index obtained in the last search (up to the boundary of the interval that we impose the value). On the other hand, it may optimise the next searches to do. Indeed, let us explain the Script **IV.5**.

```
int get_jet(int plen, double *const pt, double t, int *const pidx)
{
  typeof(t) *s;

  if (pidx == NULL)
    {
      s = (typeof(s))bsearch(&t,pt,plen,sizeof(*pt),&cmp);
      if (s != NULL)
        { return (s - pt); }
      return -1;
    }

  s = (typeof(s))bsearch(&t,pt+(*pidx),plen-(*pidx),sizeof(*pt),&cmp);
  if (s != NULL)
    {
      *pidx = s - pt;
      return *pidx;
    }
  return -1;
}
```

Script IV.5. Optimised search of previous jets.

The idea is to reduce the search space in successive calls to the function. For that reason, we are going to save the index of the last search done. And at each new call to the function, we will use that value. Graphically:

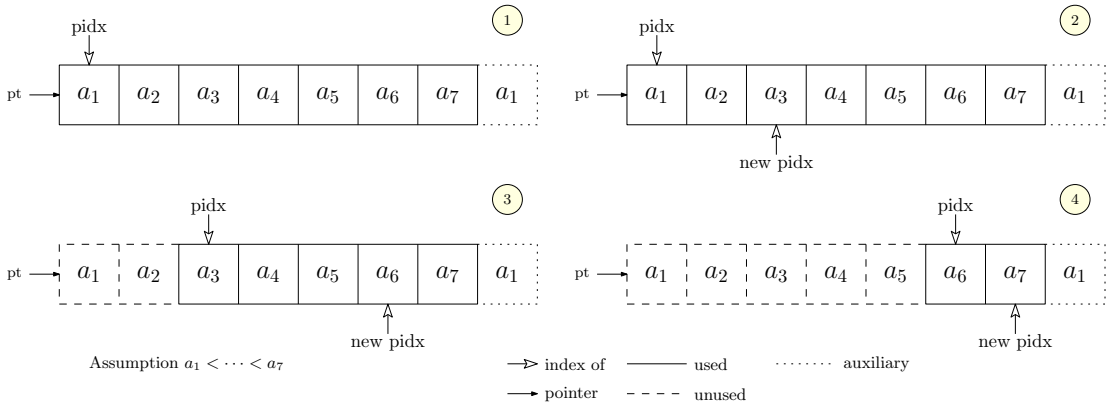

Figure IV.6. The search space is smaller in each call to the function `get_jet` in Script **IV.5**.

The conclusion is that the search that we need to do has a complexity of $O(\log_2(\texttt{plen}-\texttt{pidx}))$. It is true that one can wonder why we need to save all the previous interval if we can reuse the memory in that indexes smaller than `pidx`. Well, the answer is fast. We need to save it because we must interpolate and if we want to do that in a balanced way, we will need points enclosing the point indexed by `pidx`.

3.1.2. *Sharing data.* The memory strategy thought in Delayed Runge-Kutta does not work in Delayed Runge-Kutta-Fehlberg. The reason is that the real data used in the previous data interval pt, px and the current data interval ct, cx have not a constant size because the step h is non-constant. Therefore, a fixed number of times and points are allocated in the initialization of the struct in the Script **IV.4**.

Let us explain the Figure **IV.7** when the delay τ is equal to 1.

1: Let us assume that the initial interval has been introduced with $plen - 1$ elements in px, pt. The memory of u_{plen} and x_1 is shared and the computation of x_1 in cx puts the result in that shared memory. The other imposed condition is that the last time in pt is the first time in ct.
Now, let us suppose that the method goes on up the updated time $t + h$ is bigger than the boundary of the interval (in the Figure that means $t + h > 1$).

2: As $t + h > 1$, we replace the value h with a smaller h such that $t + h = 1$.
After that, px with cx and pt with ct will be swapped.

3: u_{plen} points to u_1 and u_1 points to x_{clen}.
At the same time that the last time on pt (now 1) will be the first time on ct.

4: Finally, the value u_1 will be computed with the data of the previous interval. As a consequence, the value x_{plen} will also be modified because it shares memory with u_1.
And the process goes on up to a final time given by the user.

The procedure explained for Figure **IV.7** has been chosen for a specific reason, the interpolation. Indeed, if we are in a boundary of the interval, let us say $t = 0$, then the Equation (4.5) tells us that the values $t + a_i h - \tau$ is needed. As the step size is not constant, the points are interpolated by the data in px. But if we want to compute the value x_{clen}, the value u_{plen}, which is exactly the same that x_1, will be required by the interpolation.

Thus the boundary of the intervals has to be shared and the implemented paradigm has only been "tail-head" in contrast to the relation "head-tail and tail-head" used in the Delayed Runge-Kutta method.

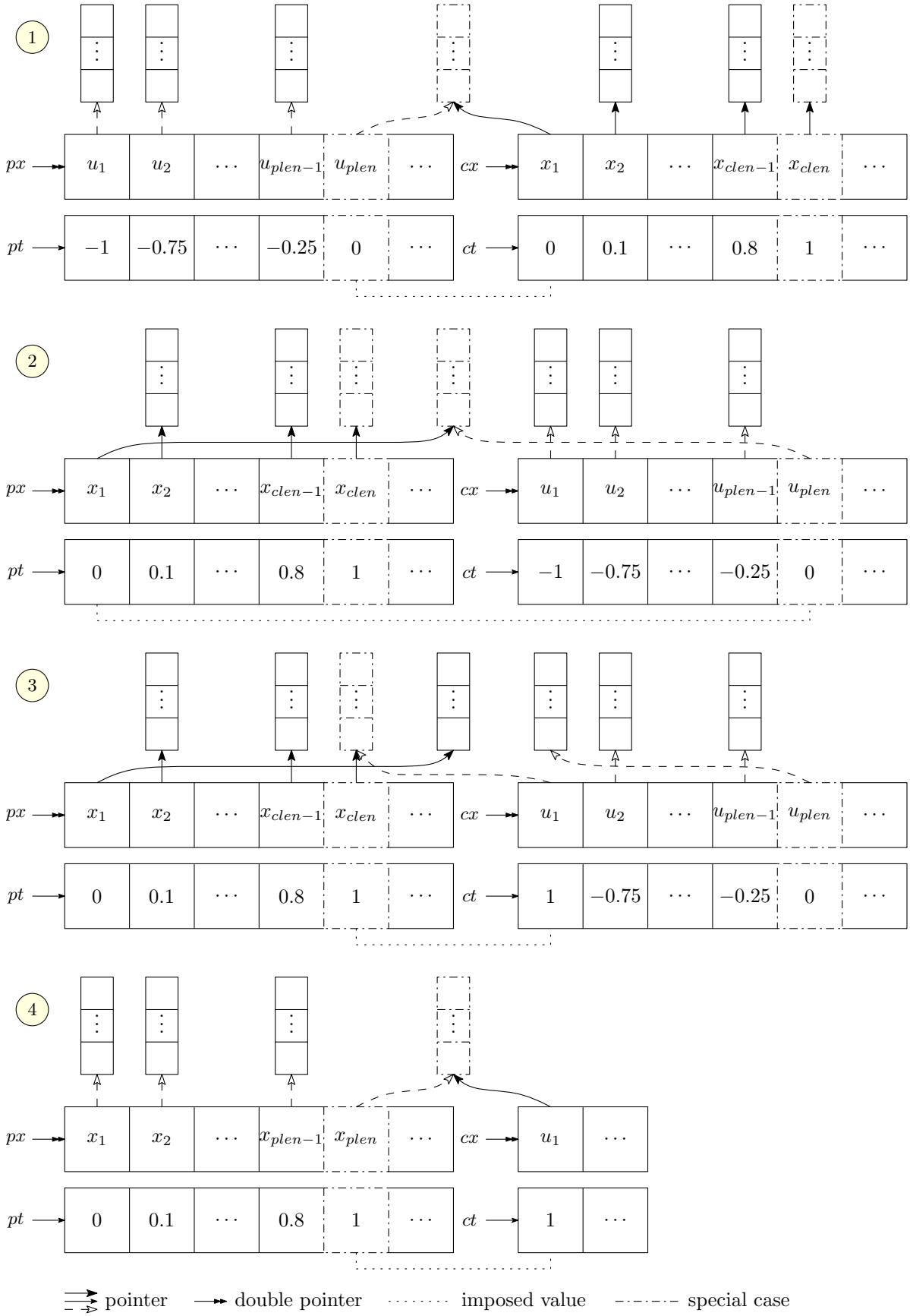

Figure IV.7. Previous interval allocation px, pt and current interval allocation cx, ct for a Delayed Runge-Kutta-Fehlberg implementation.

3.2. Example. Let us do the same example that in Section **2.2**. That is the Equation (4.4). First of all, we must fix the Butcher's Tableau of the Delayed Runge-Kutta-Fehlberg. We decided to implement when $p = 7$ and $q = 8$. It has Table **IV.8**.

	0												
	8	32800											
	12	12300	36900										
	18	18450	0	55350									
	45	184500	0	−691875	691875								
$\cdot\frac{1}{180}$	54	22140	0	0	110700	88560							$\cdot\frac{1}{442800}$
	90	−102500	0	0	512500	−1066000	1025000						
	18	45756	0	0	0	120048	−98400	6396					
	72	885600	0	0	−3911400	6927360	−5264400	329640	1328400				
	36	−373100	0	0	94300	−3201280	2550200	−140220	1254600	−36900			
	108	257364	0	0	−920700	1942272	−1625400	230364	243000	121500	194400		
	0	6480	0	0	0	0	−64800	−6480	−32400	32400	64800	0	
	108	−191916	0	0	−920700	1942272	−1560600	236844	275400	89100	129600	0	442800
$\cdot\frac{1}{840}$		41	0	0	0	0	272	216	216	27	27	41	
		0	0	0	0	0	272	216	216	27	27	0	41	41

Table IV.8. Butcher's Tableau of explicit Runge-Kutta-Fehlberg 78.

After the memory allocation control of our implementation we propose the declaration of `drk78` function in Script **IV.6** whose arguments are:

- ◇ Struct `dde_rkf` initialized.
- ◇ Dimension n of the problem.
- ◇ Pointer to the time t whose next value will be $t + h$.
- ◇ Pointer to the current x.
- ◇ The lowest allowed step $hmin$, pointer to the step h and the largest allowed step $hmax$.
- ◇ Pointer to the final time tf.
- ◇ Tolerance tol.
- ◇ Delay r, i.e. τ in (4.4).
- ◇ Pointer to the function of the Delay Differential Equation.
- ◇ Pointer to the interpolation function.
- ◇ Pointer to the k's and the number of k's used.
- ◇ Auxiliary pointer for the interpolated polynomial with `inter_len` points.

```
int drkf78(struct dde_rkf * const dde,
           int n,
           double * const t,
           double * const x,
           double hmin, double * const h, double hmax,
           double * const tf,
           double tol,
           double r,
           void (*F)(int, int, double *const, double, double *const,
               double, double *const, double **const),
           int (*inter)(int, int, double, double *const, double **
               const, double *const, double *const),
           int nK, double ** const K,
           int inter_len, double * const inter_pol)
```

Script IV.6. Declaration of the implemented function for the Delayed Runge-Kutta-Fehlberg 78 in C.

As one can observe in Table **IV.8**, some a_i's have the same value. That means, in particular, that some interpolated polynomials can be reused. The result of the integration of the Equation (4.4) with arguments

$$h_{\min} = 10^{-22}, \quad h = \frac{\tau}{2^9}, \quad h_{\max} = 1 \quad \text{and} \quad tol = 10^{-16}$$

has been plotted in Figure **IV.8**.

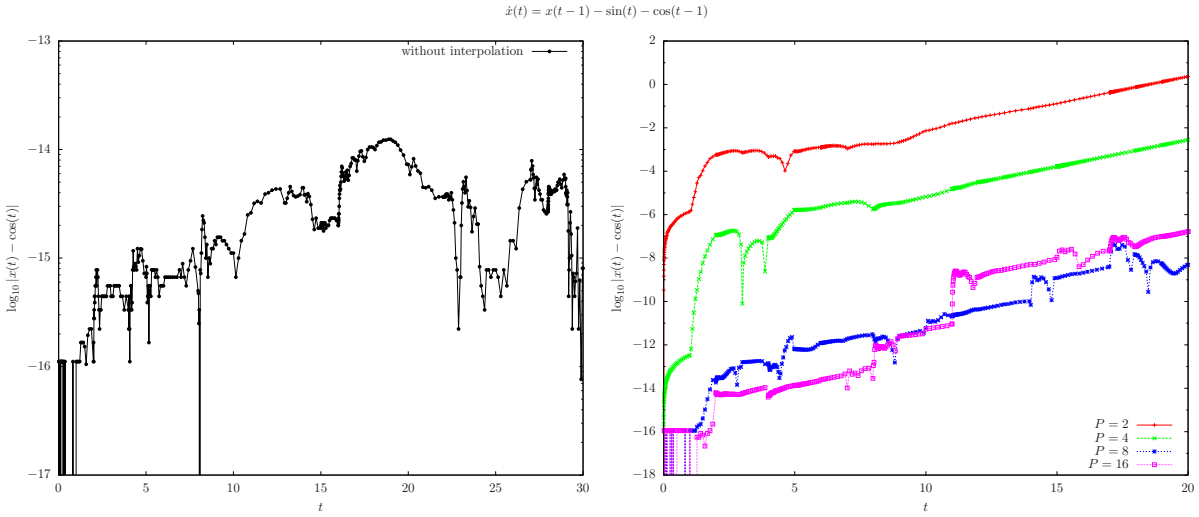

Figure IV.8. Plot of (4.4) without interpolation in the left hand side and with interpolation using P points in the other side.

In order to show the range of the non-constant step size we can saved the lowest and the largest values used in the computation of Figure **IV.8**.

	$\min h$	$\max h$	CPU time in seconds
without interpolation	1.599807e − 03	2.599525e − 01	0.00
$P = 2$	2.082941e − 03	2.518526e − 01	0.00
$P = 4$	2.109375e − 03	2.595972e − 01	0.01
$P = 8$	2.109375e − 03	2.159917e − 01	0.00
$P = 16$	2.109375e − 03	2.263812e − 01	0.01

Table IV.9. Range of the step h in the computation of Figure **IV.8**.

4. Taylor's method

Ordinary case. The method assumes that our initial map is smooth enough in the sense that there is the Taylor's expansion near to any point in the integration domain up to a prefixed truncation. That is, the solution of our Initial Value Problem can be expressed by

$$x(t+h) \approx x(t) + x^{[1]}(t)h + \cdots + x^{[p]}(t)h^p, \qquad \text{with } x^{[j]} = \frac{x^{(j)}}{j!}.$$

Thus the iterative scheme becomes to

$$\begin{aligned} x_{m+1} &= x_m + x_m^{[1]}(t_m)h_m + \cdots + x_m^{[p]}(t_m)h_m^p \\ t_{m+1} &= t_m + h_m. \end{aligned}$$

whenever $m \geq 0$. The h_m is called the m-th step and it should be a smaller value than a radius of convergence of a local Taylor's expansion at t_m of the solution $x(t)$.

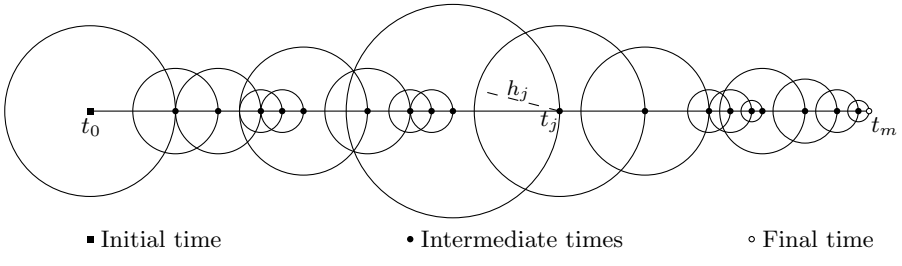

Figure IV.9. Idea of the Taylor's integration from an initial point to a final point.

In fact, there are two parameters that are needed to achieve a given level of accuracy. At the same time, that one try to minimize the total number of arithmetic operations so that the resulting computation ought to be as fast as possible.

4.1. Optimal selections. Step and Order size. According to [**15**] and [**1**], let us explain how we can estimate the step and order size of the Taylor's method. Let $x_m^{[j]}(t_m)$ be the jet of normalized derivative at t_m of the solution of our Initial Value Problem. It satisfies

$$x_m^{[0]}(t_m) = x_m(t_m) = x_m.$$

We want to choose a small enough value of h_m and a large enough value of p_m such that the values

$$\begin{aligned} x_{m+1} &= x_m + x_m^{[1]}(t_m)h_m + \cdots + x_m^{[p_m]}(t_m)h_m^{p_m} \\ t_{m+1} &= t_m + h_m \end{aligned}$$

satisfy

$$\|x_m(t_{m+1}) - x_{m+1}\| \leq \varepsilon$$

with ε a given level of accuracy. We also look for efficiency, i.e. the computational cost over the unit of time has to be minimum.
Let us assume that the total cost is proportional to p^2. In fact, we showed that applying automatic differentiation each operation in Table **III.3** has a complexity $O(p^2)$. So the assumption is not a really restriction.
Let us suppose that our solution $x(t)$ is locally analytic. Hence give an initial data (t_0, x_0), then

$$x(t_0 + h) = \sum_{j\geq 0} x_j h^j, \qquad x_j \in \mathbb{R}^n. \tag{4.6}$$

with a radius of convergence r. By Cauchy's inequality, the series (4.6) is convergent if

$$\forall\, \rho < r,\ \exists\, M > 0;\ \forall\, j \geq 0,\ \|x_j\| \leq \frac{M}{\rho^j}.$$

If we require $\|x_p\|h^p \leq \varepsilon$, then

$$\frac{h}{\rho} \leq \left(\frac{\varepsilon}{M}\right)^{\frac{1}{p}}. \tag{4.7}$$

Renaming the two fractions, $\hat{h} \leq \hat{\varepsilon}^{\frac{1}{p}}$ is obtained. Thus the function that has to be minimized is

$$\psi(p) = \frac{\text{Computational cost}}{\text{Cost per unit of time}} = \frac{ap^2}{\hat{\varepsilon}^{\frac{1}{p}}}$$

with a a suitable positive constant. Applying logarithmic derivative, the critical point is close to

$$\frac{d\log\psi}{dp}(p) \approx \frac{2}{p} + \frac{\log\hat{\varepsilon}}{p^2} = 0 \Rightarrow p \approx \left\lceil -\frac{\log\hat{\varepsilon}}{2} \right\rceil.$$

Applying now logarithm in both sides of (4.7), then

$$\log\hat{h} = \frac{1}{p}\log\hat{\varepsilon} = -\frac{2}{\log\hat{\varepsilon}}\log\hat{\varepsilon} \Rightarrow \hat{h}_{\text{optimal}} = e^2 \Rightarrow h_{\text{optimal}} = \frac{\rho}{e^2}.$$

Moreover, the remainder of the series (4.6) can be bounded by that values. Indeed,

$$\left\|\sum_{j>p} x_j h^j\right\| \leq \sum_{j>p}\|x_j\||h|^j \leq M\sum_{j>p}\frac{1}{\rho^j}\frac{\rho^j}{e^{2j}} = M\sum_{j>p}\left(\frac{1}{e^2}\right)^j = M\frac{e^{-2(p+1)}}{1-e^{-2}} \leq \varepsilon\frac{e^{-2p}}{e^2-1}.$$

The Proposition **IV.1** sums up the obtained results.

PROPOSITION IV.1. *Let $x(t)$ be the solution of an Initial Value Problem such that $z \mapsto x(t_m + z)$ is analytic on a disk of radius r_m. Then for any $\rho_m < r_m$, there is $M_m > 0$ such that*

$$\|x_m^{[j]}\| \leq \frac{M_m}{\rho_m^j}, \qquad j \geq 0. \tag{4.8}$$

If the required accuracy ε tends to 0, the values p_m and h_m that required the accuracy and minimize the global number of operations tend to

$$h_m = \frac{\rho_m}{e^2} \qquad \textit{and} \qquad p_m = -\frac{1}{2}\log\frac{\varepsilon}{M_m} - 1. \tag{4.9}$$

Noteworthy comments are found in [**1**]:

- The values in (4.9) are optimal only when the bound in (4.8) can not be improved. If, for instance, M_m can be reduced, the previous values are not optimal, i.e. other values of h_m or p_m could still deliver the required accuracy.
- The optimal step size does not depend on the level of accuracy and the optimal order guarantees the required tolerance once the step size has been selected.

4.1.1. *Strategies and estimations of order and step size.* Let us give different ways of that estimations.

Naive estimations: Fix the order p and the step h constants in all the integration.

Estimation of the Order: Given the absolute ε_a and the relative ε_r tolerances. Let us define

$$\varepsilon = \begin{cases} \varepsilon_a & \text{if } \varepsilon_r \|x_m^{[0]}\|_\infty \leq \varepsilon_a \\ \varepsilon_r & \text{otherwise.} \end{cases}$$

Then the order is

$$\left\lceil -\frac{1}{2}\log\varepsilon + 1 \right\rceil \leq p_m.$$

`N.B.:` In fact, that p_m can be independent of m, let us called it p.

Time-step estimation: Given an order p, absolute ε_a and relative ε_r tolerances. Let

$$\rho_m^{(j)} = \begin{cases} \left(\dfrac{1}{\|x_m^{[j]}\|}\right)^{\frac{1}{j}} & \text{if } \varepsilon_r \|x_m^{[0]}\|_\infty \leq \varepsilon_a \\ \\ \left(\dfrac{\|x_m^{[0]}\|}{\|x_m^{[j]}\|}\right)^{\frac{1}{j}} & \text{otherwise.} \end{cases}$$

If $\rho_m = \min\{\rho_m^{(p-1)}, \rho_m^{(p)}\}$, then the estimated time-step is

$$h_m = \frac{\rho_m}{e^2}.$$

Time-step estimation using absolute error: Given an order p and a tolerance ε. Let

$$\rho_0 = \left(\frac{\varepsilon}{\|x_m^{[p]}\|}\right)^{\frac{1}{p}} \qquad \text{and} \qquad \rho_1 = \left(\frac{\varepsilon}{\|x_m^{[p-1]}\|}\right)^{\frac{1}{p-1}}.$$

Then $h_m = \min\{\rho_0, \rho_1\}$.

Time-step estimation using relative error with respect to the 1st order: Given an order p and a tolerance ε. Let us define

$$\rho_0 = \left(\frac{\varepsilon\|x_m^{[1]}\|}{\|x_m^{[p]}\|}\right)^{\frac{1}{p}} \qquad \text{and} \qquad \rho_1 = \left(\frac{\varepsilon\|x_m^{[1]}\|}{\|x_m^{[p-1]}\|}\right)^{\frac{1}{p-1}}.$$

Then $h_m = \min\{\rho_0, \rho_1\}$.

Time-step estimation using absolute error and relative with the 1st order: Given an order p and a tolerance ε. Let us define

$$\rho_0 = \left(\frac{\varepsilon}{\|x_m^{[p]}\|}\right)^{\frac{1}{p}} \qquad \text{and} \qquad \rho_1 = \left(\frac{\varepsilon\|x_m^{[1]}\|}{\|x_m^{[p-1]}\|}\right)^{\frac{1}{p-1}}.$$

Then $h_m = \min\{\rho_0, \rho_1\}$.

It will depend on the function of our differential equation that we ought to choose one of the previous estimations (or others).

Delayed case. When we have a Delay Differential Equation with a constant delay τ, we can try to apply the Taylor's method in each interval of size τ. The boundary values are the only thing that has to be considered. Indeed, since $\dot{x}$ means the right-hand derivative of x, the Taylor's method computes that values up to the upper bound boundary value. In the interior of the interval the solution is as smoother as our Delay Differential Equation is. And in the boundary the smooth degree increases in each new computed interval.
As a consequence, in an implementation we should save the right-hand derivative and the left-hand derivative of any interval boundary value.
A graphical example of the method has been drawn in Figure **IV.10**. The values in $[-\tau, 0]$ are the discretised initial condition of our Initial Value Problem. Given the initial data, the ordinary Taylor method can be applied up to time τ, up to 2τ and so on. In fact, it is not exactly the ordinary Taylor method because our Delay Differential Equation has the form

$$\dot{x}(t) = f(t, x(t), x(t-\tau))$$

and the computed values of the previous interval has to be saved in order to be used in the computation of a new value. Consequently, we will need to look for values of the previous data.

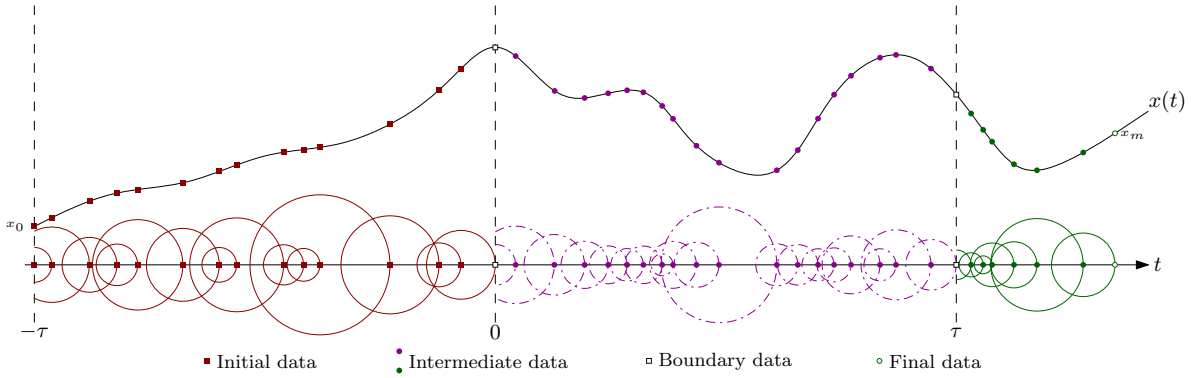

Figure IV.10. Graphical example of the Delayed Taylor method.

Let us also observe that the disks estimated by the radii ρ_j are in the time-line of the Figure **IV.10**.

4.2. Comments for an implementation. Like all the previous methods explained, the initial condition of an Initial Value Problem has to be discretised and the discretisation procedure will be depend on each Delay Differential Equation.
The Delayed Taylor method requires all the normalized derivatives at each point up to a prefixed order, that implies a triple pointer variable for the previous and current data.
If we want to work with a non-constant step size, we will need two new variables *plen* and *clen* which values means, respectively, the length of previous and current data.
Hence let us decide to use the Script **IV.7** which is quite closed to the Script **IV.4**. The differences are in the triple pointers of px, cx, the new variable N and $pidx$ for efficient searches (see Section **3.1.1**).

```c
struct dde_taylor
{
  /**
   * n          Dimension of our problem
   * N          Number of Taylor coefficients
   * njets      Number of jets
   * plen       Length of the previous data
   * pidx       Index of the last jet used
   * clen       Length of the current data
   * pt         Previous time array
   * px         Previous x's array
   * ct         Current time array
   * cx         Current x's array
   */
  int n, N, njets, plen, pidx, clen;

  double *pt, ***px, *ct, ***cx;
};
```

Script IV.7. Struct for an implementation of Delayed Taylor method in C.

Now the memory allocation can not be reused. The only issue that we can force is the final time of pt (i.e. indexed by $plen$) has to be the same that the first time in the current time ct. The procedure of the Figure **IV.11** is the next:

- ⋆ Given a previous data with length $plen$.
- Compute the current data searching on the previous data (see Section **3.1.1**) up to the time is either the final time or the upper bound of the boundary time-values.
- In the latter case, do:
  - $px \leftrightarrow cx$ and $pt \leftrightarrow ct$ (swapping).
  - $ct_1 \leftarrow pt_{clen}$.
  - $plen \leftarrow clen$ and $clen \leftarrow 0$.
- If at some moment the $clen$ is equal to $njets$, reallocate the memory.

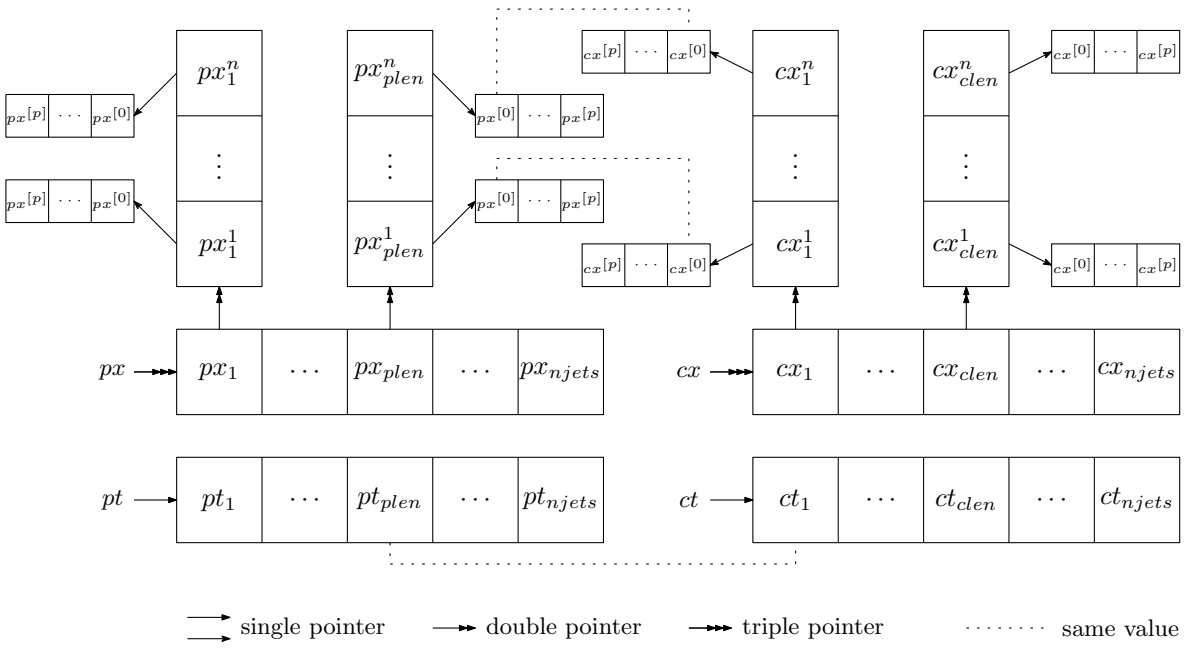

Figure IV.11. Previous interval allocation px, pt and current interval allocation cx, ct for a Delayed Taylor implementation.

Another important difference with the Delayed Runge-Kutta and Delayed Runge-Kutta-Fehlberg is that the interpolation is not needed because the convergence of the Taylor expansion can be used in order to obtain the next jet. Moreover, automatic differentiation and Horner's method are strictly recommended in this part. Indeed, let us consider the map

$$\varphi(h) = c_0 + c_1 h + \cdots + c_{N-1} h^{N-1}$$

with $c_j \in \mathbb{R}^n$. Since φ can be expressed by n polynomials, the evaluation procedure works and an implementation of complexity $O(nN)$ is obtained by the Horner's method. The Script **IV.8** codifies a possible implementation using the Theorem **III.5** and storing $\varphi^{[j]}$ in the variable `x`.

```
void diff_taylor_expansion(int N, int n, double ** const x, double h,
                           double ** const C)
{
  typeof(N) k, i;
  typeof(n) j;
  typeof(**x) s, t;

  for (j = 0; j < n; j ++)
    {
      x[j][1] = C[j][N-1];
      x[j][0] = x[j][1] * h;

      for (k = 2; k < N; k ++)
        {
          x[j][0] += C[j][N-k];

          t       = x[j][0];
          x[j][0] *= h;
          for (i = 1; i < k; i ++)
            {
              s       = x[j][i];
              x[j][i] = s * h + t;
              t       = s;
            }
          x[j][i]  = t;
        }
      x[j][0] += C[j][0];
    }
}
```

Script IV.8. Differentiation of a Taylor expansion in C.

4.3. Example. Let us consider again the example of the Equation (4.4), that is,

$$\dot{x}(t) = x(t-1) + r(t)$$

with $r(t)$ a suitable map such that if the initial condition is $x_0 \equiv \cos$, the solution is $x(t) = \cos(t)$. The initial input of the delayed Taylor method is a finite collection of jets with all the values at that points and, in addition, all the derivatives at that points until some prefixed order N. At least, the input size must have two jets and the latter has to be the left-hand derivative of the right-hand boundary interval (in our case, the left-hand derivative at $t = 0$).
Since the initial condition is an analytic function our discretisation has only two jets. One at $t = -1$ with a step size $h = 1$ and the second at $t = 0$.
In the delayed Taylor method, we need to be able to compute all the derivatives until some prefixed order N. In particular, the Taylor expansion of $r(t)$ can be computed using the Table **III.3**, i.e. the C++ class defined in Script **III.1**.
As the solution of the Initial Value Problem is known, the error can be computed. It has been plotted in Figure **IV.12** with an input absolute and an input relative error of 10^{-15}.

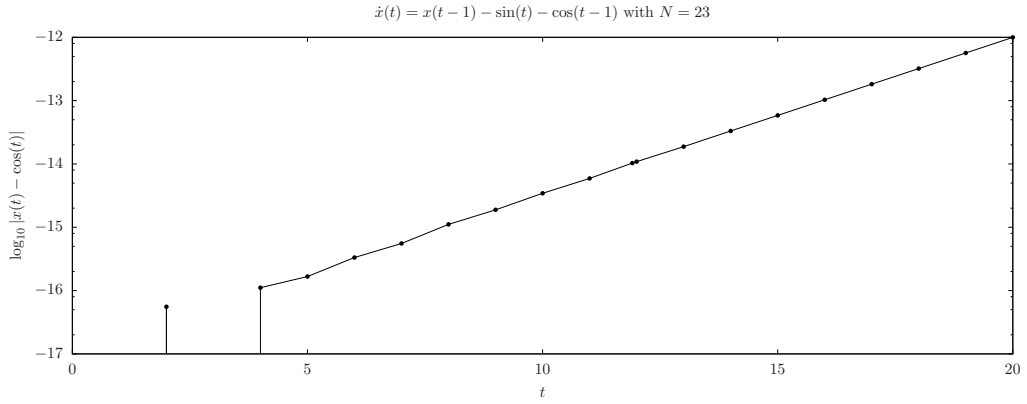

Figure IV.12. Logarithmic error of the delayed Taylor integration of the Equation (4.4).

5. Mackey-Glass equation

The Mackey-Glass equation is a non-linear time Delay Differential Equation with a constant delay $\tau > 0$. It has some parameters $\alpha, \beta, \gamma > 0$ and it is defined by

$$\dot{x} = \beta \frac{x(t-\tau)}{1 + x(t-\tau)^\alpha} - \gamma x.$$

We want it to be integrated and compared with a well-known solution. Let us do the same strategy that in the Equation (4.4). Adding a new map $r(t)$. If the solution is $x(t) = \cos(t)$, then

$$r(t) = -\sin(t) - \beta \frac{\cos(t-\tau)}{1 + \cos(t-\tau)^\alpha} + \gamma \cos(t). \tag{4.10}$$

verifies that

$$\dot{x} = \beta \frac{x(t-\tau)}{1 + x(t-\tau)^\alpha} - \gamma x + r(t)$$

has solution $x(t) = \cos(t)$. Hence the initial condition is

$$\begin{cases} \dot{x} = \beta \dfrac{x(t-\tau)}{1 + x(t-\tau)^\alpha} - \gamma x - \sin(t) - \beta \dfrac{\cos(t-\tau)}{1 + \cos(t-\tau)^\alpha} + \gamma \cos(t) \\ x_0 \equiv \cos\,. \end{cases} \tag{4.11}$$

DRK4 of the Mackey-Glass equation. Since the solution of the Equation (4.11) is $\cos(t)$, a first approach is to suppose that the interpolation step uses that fact, i.e. if some value needs to be interpolated at time t, then $\cos(t)$ will be the interpolated value. Thus, the error of the integration without interpolation has been plotted in Figure **IV.13**.

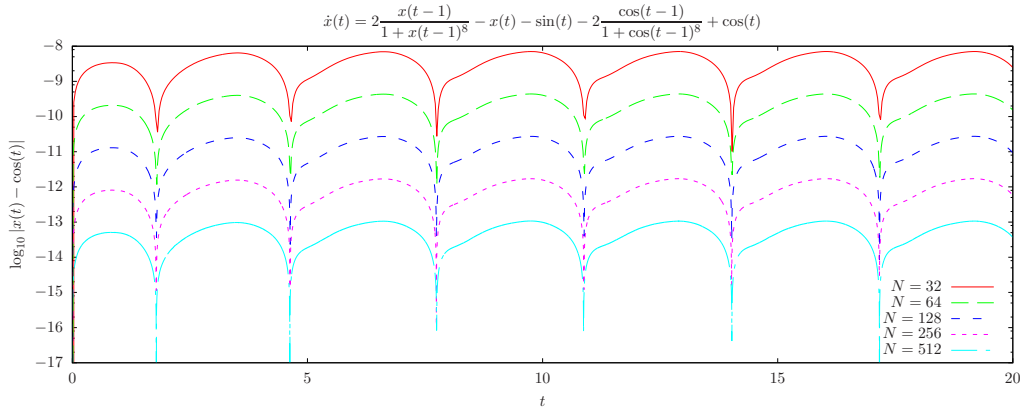

Figure IV.13. Delayed Runge-Kutta 4 with constant step $h = \frac{1}{N}$ of the Equation (4.11).

On the other hand, in Figure **IV.14** can be found the logarithmic error with different values of the step size and different interpolation sizes.

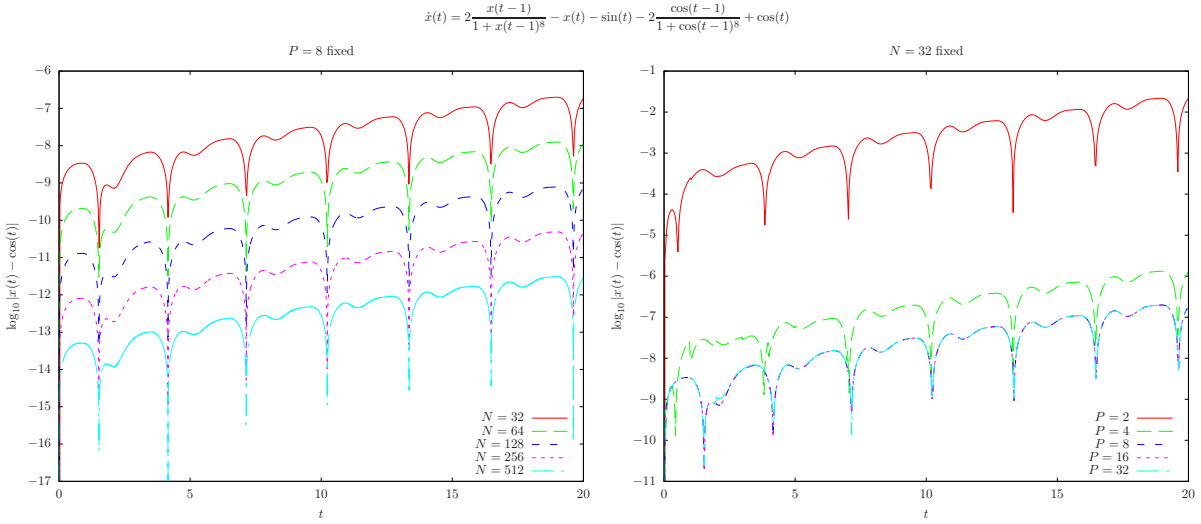

Figure IV.14. Delayed Runge-Kutta 4 of the Equation (4.11) with P interpolation size and $h = \frac{1}{N}$ step size.

DRKF78 of the Mackey-Glass equation. As we have done with DRK4, the solution of the Equation (4.11) is $\cos(t)$, so the interpolation can be done with the exact solution. Then different size in the interpolation step can be used in order to compare which has smaller error. The logarithmic error of the integration of the Equation (4.11) with arguments

$$h_{\min} = 10^{-22}, \quad h = \frac{\tau}{2^9}, \quad h_{\max} = 1 \quad \text{and} \quad tol = 10^{-16}.$$

has been plotted in Figure **IV.15**.

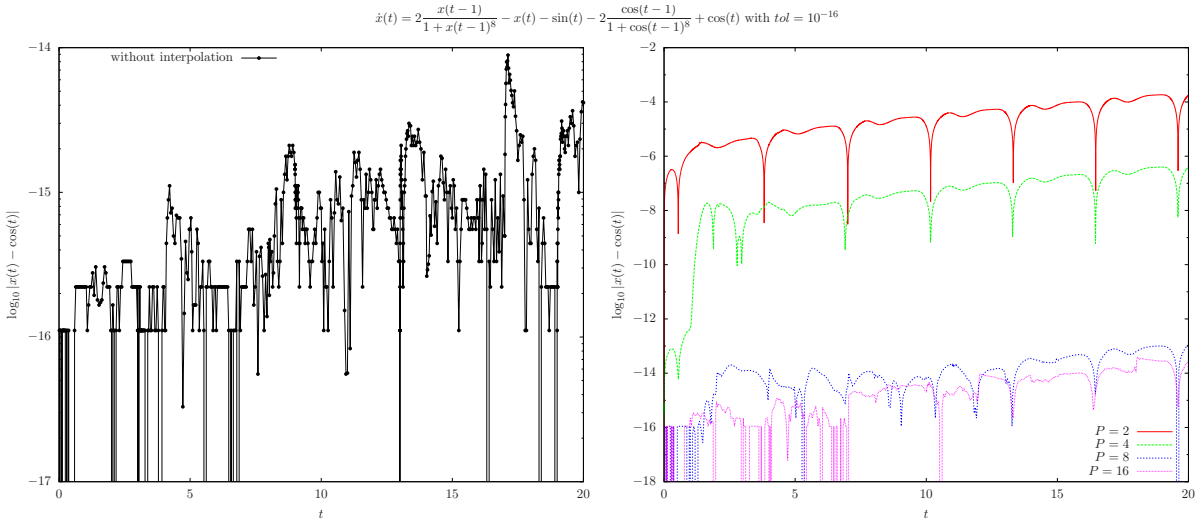

Figure IV.15. Delayed Runge-Kutta-Fehlberg 78 of the Equation (4.11) with P interpolation size and tolerance $tol = 10^{-16}$.

Furthermore, the maximum and the minimum step size used in that integration can be found in the following Table.

	$\min h$	$\max h$	CPU time in seconds
without interpolation	5.716693e − 04	6.688708e − 02	0.01
$P = 2$	1.722493e − 04	6.089603e − 03	0.12
$P = 4$	2.235153e − 04	4.956693e − 02	0.02
$P = 8$	7.705297e − 04	6.758960e − 02	0.00
$P = 16$	2.109375e − 03	6.744259e − 02	0.02

Table IV.10. Range of the step of h in the computation of Figure **IV.15**.

DTaylor of the Mackey-Glass equation. Finally, we can integrate the Equation (4.11) using the Delayed Taylor method with an input absolute and an input relative error of 10^{-15} and with an order in the local Taylor's expansion of $N = 23$ (see Figure **IV.16**). Again we can use Automatic Differentiation in the suitable $r(t)$ of (4.10) and using the Script **III.1**.

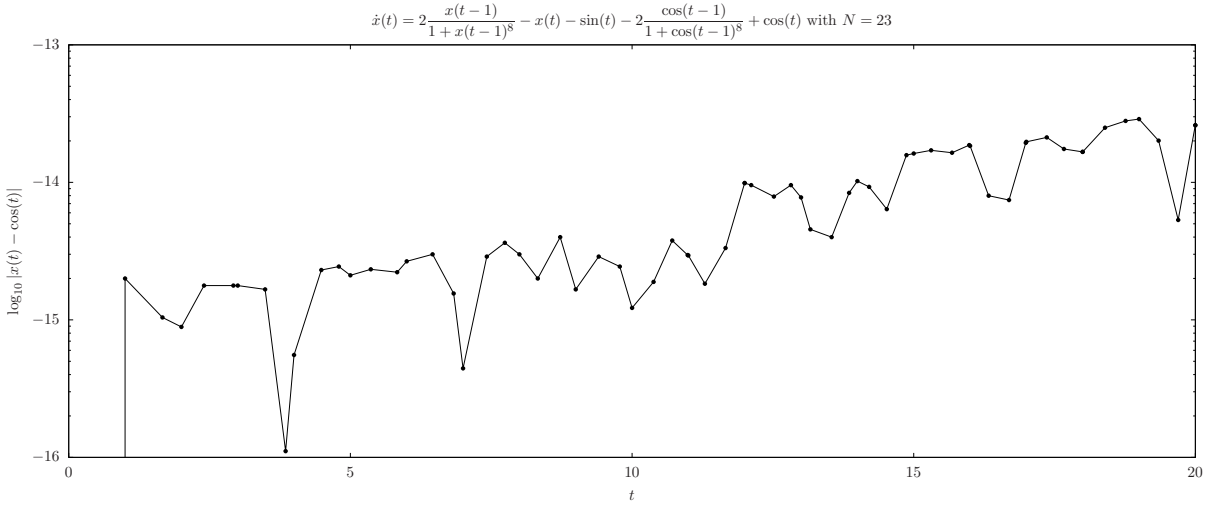

Figure IV.16. Logarithmic error of the Delayed Taylor method of the Equation (4.11).

CHAPTER V

Computation of periodic orbits

1. Ordinary periodic orbits

In Ordinary Differential Equations Theory the Poincaré map of an ODE

$$\dot{x} = f(t, x),$$

which has periodicity $\omega > 0$ with respect to the first variable, i.e. $f(t, x) = f(t + \omega, x)$, can be constructed using a suitable transversal section.
Firstly, let us recall the previous concept:

- A subvariety $\Sigma \subset \mathbb{R}^n$ is the image of a smooth map $\upsilon \colon U \to \mathbb{R}^m$ such that
  i. $D\upsilon(x)$ is injective for all $x \in U \subset \mathbb{R}^n$.
  ii. $\upsilon \colon U \to \Sigma$ is an homeomorphism.
- A transversal section $\Sigma \subset \mathbb{R}^{n-1}$ at x is a subvariety given by $\upsilon \colon U \to \Sigma$ such that
  i. $x \in \Sigma$.
  ii. $f(\upsilon(s))$ and $\operatorname{Im} D\upsilon(s)$ are a basis of $\mathbb{R}^n$ for all $s \in U$.

The Poincaré map at the transversal section Σ of the equation $\dot{x} = f(t, x)$ is the map

$$\begin{aligned} P \colon \Sigma &\longrightarrow \mathbb{R}^n \\ x_0 &\longmapsto x(\omega; 0, x_0). \end{aligned}$$

being $x(\cdot; 0, x_0)$ the flow of the differential equation with initial condition $x(0) = x_0$. Using the Inverse Function Theorem and the Implicit Function Theorem, the Poincaré map is, in fact, a diffeomorphism. If the differential equation is defined for all $x \in \mathbb{R}^n$ and the solutions for all $0 \leq t \leq \omega$, then P defines a discrete dynamical system. A fixed point of P is the initial condition of a periodic solution of period ω, and a periodic point of period k is the initial condition of a periodic solution of period $k\omega$.

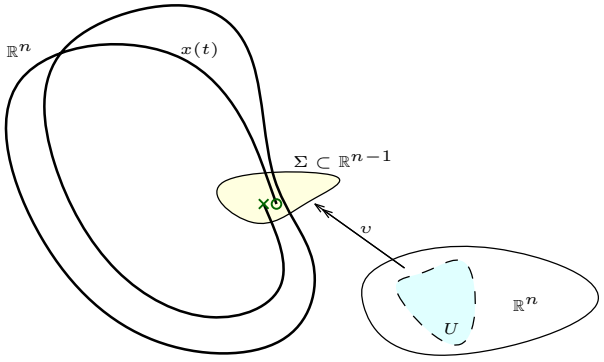

Figure V.1. A Poincaré map.

A fixed point of the Poincaré map can be tried to find it using the Newton's method which has an iterative scheme

$$\begin{aligned} \big(DP(x_k) - Id\big)h_k &= x_k - P(x_k) \\ x_{k+1} &= x_k + h_k \end{aligned}$$

beginning with a closed point x_0 to the zero desired.
By the way, the eigenvalues of $DP(x_k)$ gives us the stability of the found periodic orbit.

2. Delayed periodic orbits

Let us begin considering a Delay Differential Equation with a constant delay τ

$$\dot{x}(t) = f(t, x(t), x(t-\tau))$$

which has periodicity $\omega > 0$ with respect to the first variable, i.e.

$$f(t, x(t), x(t-\tau)) = f(t+\omega, x(t), x(t-\tau)).$$

The delayed Poincaré is also defined in a transversal section Σ which covers whole a subset of possible initial conditions. It is defined by

$$\begin{aligned} P(t_0)\colon C &\longrightarrow C \\ u &\longmapsto x_{t_0+\omega}(t_0, u) \end{aligned}$$

being $C = \mathcal{C}([-\tau, 0], \mathbb{R}^n)$ and $x(t_0, u)$ the solution of the Delay Differential Equation with initial condition $x_{t_0} \equiv u$.
The strategy that we pose is quite similar that in the ordinary case, a fixed point of the delayed Poincaré map may be a delayed periodic orbit of the Delay Differential Equation. Hence, the Newton's method may be used too.
Since we want to give numerical results, let us focus on how the differential used in the Newton's method. Firstly, the initial condition must be discretised obtaining a table of values. After that, an integration until $t_0 + \omega$ would be computed and for any of these discretised points and the Newton's method may be applied in order to search an initial discretised condition of a periodic orbit closed to the first seed used in the Newton's method. Graphically,

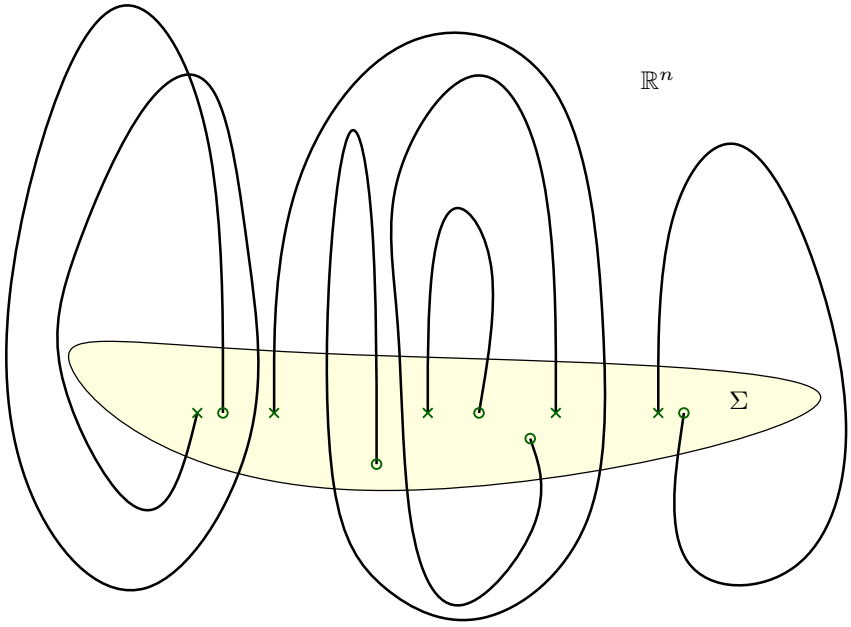

Figure V.2. Looking for fixed discretised points by a Delayed Poincaré map.

The Newton iterative scheme is, in that case,

$$\begin{aligned} \big(D(P(t_0))(u_k) - Id\big)h_k &= u_k - P(t_0)(u_k) \\ u_{k+1} &= u_k + h_k \end{aligned}$$

If the discretisation size is m, then u_k represents a matrix m-by-n and $D(P(t_0))$ has dimension m-by-m-by-n. The idea is that we are going to solve independently in each of the n coordinates.

2.1. Computation of the differential of a discretised delayed Poincaré map. Let u be a discretised initial condition of size $m+1$. That is, the values

$$u_0, \dots, u_m.$$

Let $x(t_0, u)$ be the solution of the Delayed Differential Equation. In particular, $x_{t_0}(t_0, u) \equiv u$, at least in the discretised points. If $v \equiv x_{t_0+\omega}(t_0, u)$, then

$$D(P(t_0)) = \begin{pmatrix} \dfrac{\partial v_0}{\partial u_0} & \cdots & \dfrac{\partial v_0}{\partial u_m} \\ \vdots & & \vdots \\ \dfrac{\partial v_m}{\partial u_0} & \cdots & \dfrac{\partial v_m}{\partial u_m} \end{pmatrix} \in \mathbb{R}(m+1, m+1).$$

That differential may be computed easily using automatic differentiation, in particular, the Table **III.4**. Concretely, we introduce symbols $s_0, \dots, s_m$ and the computation starts with

$$u_0 + s_0, \dots, u_m + s_m.$$

After the integration, the discretised $x_{t_0+\omega}(t_0, u)$ and the transposed differential are obtained by Gradient propagation. That propagation can be implemented overloading the arithmetic, such as doing use of the Script **III.2**.

3. Comments for an implementation

A first implementation approach is to use one of the integration methods exposed in the Chapter **IV**. Consequently, the Delayed Runge-Kutta 4 (see Chapter **IV**) is one of the most natural in a first stage. Nevertheless, that chosen implies that some objections must be made for an easier possible implementation

- the constant step must be $h = \frac{\tau}{m}$.
- ω must be a divisor of τ.

In addition, a linear system solver and an eigenvalue problem solver are needed. For this reason we have decided to use the next library:

LAPACK library. Linear Algebra PACKage is a standard software library for numerical linear algebra which has been written in Fortran 90 (version 3.2 and on). It provides routines for solving systems of simultaneous linear equations, eigenvalue problems, ...
However, a little bit drawback in a possible implementation is that the memory allocation of any matrix has to be pointed by a pointer instead of a double pointer. As a consequence, the shared memory used in Figure **IV.3** can not be applied. Let $A = ({}_k a_i^j)$ and $B = ({}_k b_i^j)$ be $(m+1)$-by-$(m+1)$-by-n 3D matrices contiguously allocated. Thus, the suggested memory allocation is in Figure **V.3**. We impose that the last values of px and pt are equals to the first values of cx and ct.

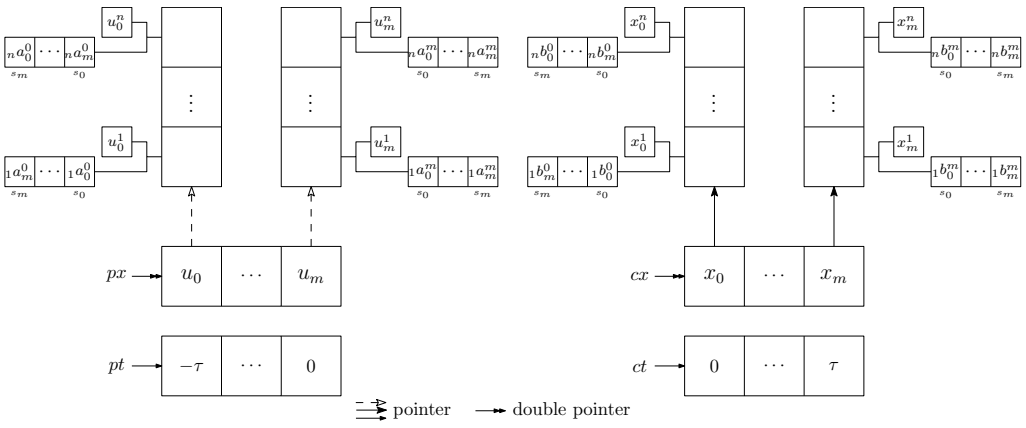

Figure V.3. Memory allocation when the LAPACK library is used.

The next code, written in C++, allows us to codify a Delay Runge-Kutta 4 overloading the arithmetic is in the Script **V.1**. It is quite similar to the Scripts **IV.1** and **IV.2** but in a fancy codification in Object Oriented Programming (OOP) paradigm.

```
template<class X, typename T=double>
class Drk
{
private:
  unsigned int n, njets, idx, nK;
  T *pt, *ct;
  X **px, **cx, **K;

  /* Private methods */
  ...

public:
  /* Constructors, public methods and destructor */
  ...

  int drk4(T &t, X * const x, T h, T r,
           void (*F)(int, int, X * const, T, X * const, T, X * const),
           int (*inter)(int, int, T, T * const, X ** const, T * const,
               X * const),
           int inter_len, T * const inter_pol);
};
```

Script V.1. Delayed Runge-Kutta class in C++.

Therefore an initialization of an object of the class `Drk` using the Script **III.2** is just

```
Drk< Gradient<double> > dde;
```

Script V.2. An initialization of Script **V.1** using the Script **III.2** in C++.

3.1. Example. Let us consider the next linear Delay Differential Equation with a constant delay $\tau = \pi$,

$$\dot{x}(t) = x(t-\pi) - \sin(t) - \cos(t-\pi).$$

If the initial condition is $x_0 \equiv \cos$, the solution will be $x(t) = \cos(t)$ which is a 2π-periodic solution.
We want to change a little bit the initial condition such that the Newton's method gives us the initial condition $x_0 \equiv \cos$. Therefore, let

$$\begin{cases} \dot{x}(t) = x(t-\pi) - \sin(t) - \cos(t-\pi) \\ x_0 \equiv \cos + 10^{-2}\sin \end{cases}$$

be Initial Value Problem. Applying the delayed Poincaré map with $\omega = 2\pi$ using the Delayed Runge-Kutta 4 with step $h = \frac{\pi}{N}$ and a tolerance 10^{-14} in the Newton's method. We have obtained the next Tables by different interpolation sizes P.

N	Newton error	Newton iterations
16	3.330669e − 16	1
32	9.992007e − 16	1
64	9.992007e − 16	1
128	1.443290e − 15	1

Table V.1. Execution summary with $P = 4$.

N	Newton error	Newton iterations
16	4.440892e − 16	1
32	2.220446e − 16	1
64	6.661338e − 16	1
128	2.442491e − 15	1

Table V.2. Execution summary with $P = 8$.

The logarithmic error with the initial condition $x_0 \equiv \cos$ has been plotted in Figure **V.4**.

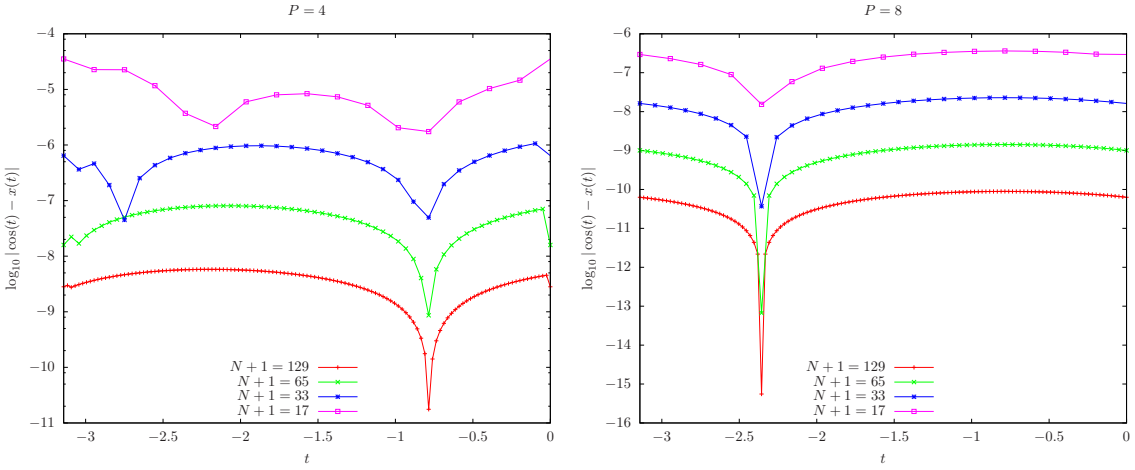

Figure V.4. Logarithmic error of the initial condition computed by the Newton's method with respect to the cosine initial condition.

By the way, the eigenvalues of the last discretised delayed Poincaré map have also computed and they have been plotted in Figure **V.5** when $P = 8$. In particular, all of them have modulus < 1 except one which is > 1. So the Corollary **II.14** does not work in that case.

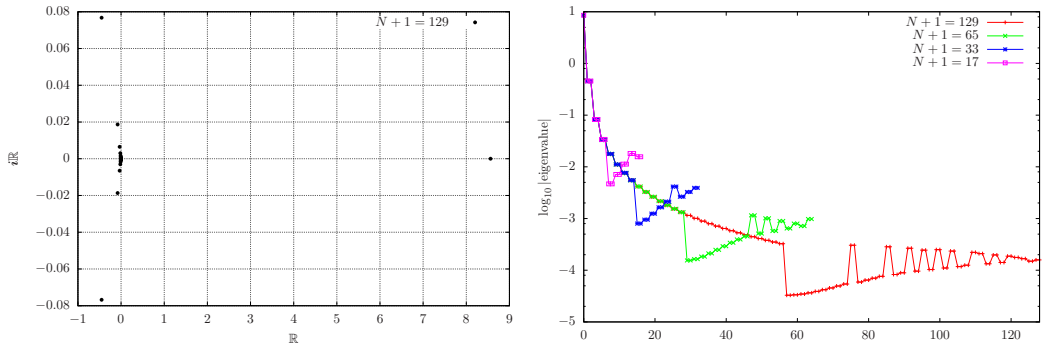

Figure V.5. Eigenvalues of the Delayed Poincaré plotted in the complex plane and their logarithmic moduli for different values of N.

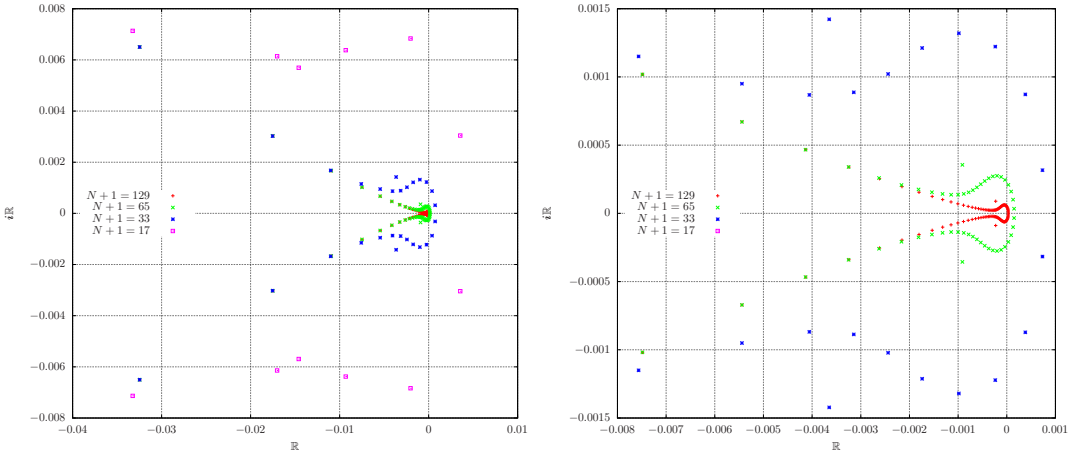

Figure V.6. Two zooms of the Eigenvalues of the Delayed Poincaré in the complex plane when the interpolation size is $P = 8$.

Conclusions

The main goal of the current project, which was to understand a first approach of the Delay Differential Equation (DDE) Theory, has been achieved.

Firstly, we have showed that the DDE Theory is more general than ODE Theory and, although, some Theorems are closed to the ordinary corresponding version, the proofs of them are quite different. Basically, because we are working in an infinite Banach space and, of course, the closed unit ball is not a compact set.
Furthermore, the linear Floquet Theory, exposed in Chapter **II**, has allowed us to use more Functional Analysis tools like Spectral Theory. However, we must say that the Delayed Floquet Theory is, nowadays, incomplete because there are cases which the Theory can not work.

Secondly, in all the rest Chapters **III** to **V** an exhaustive numerical development such as integrators of a DDE with a constant delay (Chapter **IV**) and a brief introduction to Automatic Differentiation Theory (Chapter **III**) have allowed us to introduce ourselves in a numerical computation usually used in some parts of dynamical systems. In addition, the tests with a known solution have allowed us to check if the integrator methods work well, at least, for our tests.

Thirdly, the meetings with my advisor have generated some new ideas. For instance, we have shown that Hermite's interpolation is not a good method to interpolate jets of smooth functions because it propagates huge errors in higher orders. That empirically facts have been proved by interval computations on a specific example in the last part of Chapter **III**. The second main new idea has been how a periodic orbit of a DDE with a constant delay may be computed using a Delayed Poincaré map, Chapter **V**. In addition, its stability can be studied easily. Currently, that method has not been found in any publication related with Delay Differential Equations and it looks like to be a good method according to some first results.

Personally, the project has allows me to learn a lot of in numerical computing and in dynamical systems. Moreover, a future work might be oriented in the next items:

- New numerical methods to compute periodic orbits and their stability.
- Delayed Poincaré maps implemented with different integrators.
- Integrators of DDE with multiple delays.
- Delayed Poincaré maps version with multiple delays.

APPENDIX A

Ascoli-Arzela's Theorem

Ascoli-Arzela's Theorem has been used in some subjects but in any of them has been proved, so the aim of this Appendix is to give a proof. Firstly, we need to fix some definitions and remember some previous well-known results (see [**2**]):

DEFINITION A.1. Let X be a metric space X.

- X is precompact when for any $\varepsilon > 0$, there is a finite covering of X with diameters $< \varepsilon$.
- A subset A of X is relatively compact when $\overline{A}$ is compact.

LEMMA A.2. *Let X be a metric space. There is equivalence between:*

i. *X is compact.*
ii. *Any sequence of X has at least a convergent subsequent.*
iii. *X is precompact and complet.*

Moreover,

i. *If $A \subset X$ is relatively compact, it is precompact.*
ii. *If X is complete and $A \subset X$ is precompact, it is relatively compact.*

DEFINITION A.3. Let H be a family of mappings from a metric space X to a metric space Y.

- H is equicontinuous at a point $x_0 \in X$ when

$$\forall\, \varepsilon > 0,\, \exists\, \delta > 0;\, \forall\, u \in H,\, d(x, x_0) < \delta \Rightarrow d\big(u(x), u(x_0)\big) < \varepsilon.$$

- H is equicontinuous when it is equicontinuous at any point of X.

ASCOLI-ARZELA'S THEOREM A.4. *Let H be a family of continuous maps from a compact metric space X to a Banach space Y. There is equivalence between:*

i. *H is relatively compact.*
ii. *H is equicontinuous and for any $x \in X$, $H(x) = \{u(x) \colon u \in H\}$ is relatively compact.*

PROOF.

i⇒ii) By compactness,

$$\overline{H} \subset B(u_1; \varepsilon) \cup \cdots \cup B(u_n; \varepsilon)$$

with $u_1, \ldots, u_n \in \overline{H}$. We can assume that $u_1, \ldots, u_n \in H$. Indeed, if for instance $u_1 \in \overline{H} \setminus H$, then $v_k \to u_1$ with v_k in H. By triangular inequality, $B(u_1; \varepsilon) \subset B(v_k; 2\varepsilon)$ for some k. So by arbitrariness of $\varepsilon > 0$, we can assume

$$H \subset \overline{H} \subset B(u_1; \varepsilon) \cup \cdots \cup B(u_n; \varepsilon)$$

with $u_1, \ldots, u_n \in H$.
In particular, for any $x \in X$,

$$H(x) \subset B(u_1(x); \varepsilon) \cup \cdots \cup B(u_n(x); \varepsilon).$$

The diameter of any of these balls is lesser than 2ε. So $H(x)$ is precompact and since Y is complete, it is relatively compact.
On the other hand, each $u_i \in H$ is uniformly continuous because X is compact, i.e.

$$\forall\, \varepsilon > 0,\, \exists\, \delta > 0;\, \forall\, x, y \in X,\, d(x, y) < \delta \Rightarrow \|u_i(x) - u_i(y)\| < \varepsilon.$$

We already know that given $u \in H$, then $\|u - u_i\| < \varepsilon$ for some i. So

$$\begin{aligned}\|u(x) - u(y)\| &\leq \|u(x) - u_i(x)\| + \|u_i(x) - u_i(y)\| + \|u_i(y) - u(y)\| \\ &\leq \|u - u_i\| + \|u_i(x) - u_i(y)\| + \|u_i - u\| < 3\varepsilon\end{aligned}$$

and H is equicontinuous.

ii⇒i) It is enough to prove that H is precompact because the set of continuous maps from X to Y is complete with the uniform convergence.

Take $\varepsilon > 0$. Since H is equicontinuous, for any $x \in X$, there is $\delta_x > 0$ such that

$$\forall\, x \in X,\, \exists\, \delta_x > 0;\, \forall\, u \in H,\, d(x,y) < \delta_x \Rightarrow \|u(x) - u(y)\| < \varepsilon.$$

By compactness, $X \subset B(x_1;\delta_1) \cup \cdots \cup B(x_n;\delta_n)$. In particular, for any $u \in H$,

$$x \in B(x_i;\delta_i) \Rightarrow \|u(x) - u(x_i)\| < \varepsilon.$$

Now, since, $H(x_i)$ is relatively compact, then $K = H(x_1) \cup \cdots \cup H(x_n)$ is also relatively compact. Therefore, there are $y_1, \ldots, y_m \in K$ such that

$$K \subset B(y_1;\varepsilon) \cup \cdots \cup B(y_m;\varepsilon).$$

For any map $\varphi\colon \{1,\ldots,n\} \to \{1,\ldots,m\}$, let us define

$$H_\varphi = \{u \in H\colon u(x_i) \in B(y_{\varphi(i)};\varepsilon) \text{ for all } i = 1,\ldots,n\}.$$

It is clear that there are a finite number of φ's and the union of H_φ's is a covering of H. We will finish the proof if we show that the diameter of H_φ is bounded by ε. Indeed, given $u, v \in H_\varphi$ and $x \in X$, then $x \in B(x_i;\delta_i)$ for some i and

$$\begin{aligned}\|u(x) - v(x)\| \leq\ & \|u(x) - u(x_i)\| + \|u(x_i) - y_{\varphi(i)}\| \\ & + \|y_{\varphi(i)} - v(x_i)\| + \|v(x_i) - v(x)\| \\ <\ & 4\varepsilon.\end{aligned}$$

So $\|u - v\| < 4\varepsilon$ and H is precompact. By Lemma **A.2**, it is relatively compact. □

PROPOSITION A.5. *Let X be a compact metric space, Y be a Banach space and (u_n) be an equicontinuous sequence in $\mathcal{C}(X,Y)$.*

If $u_n \to u$ pointwise on X, then $u_n \to u$ uniformly on X.

PROOF. By equicontinuity assumption, given $\varepsilon > 0$ and $x \in X$,

$$\exists\, \delta_x > 0;\, \forall\, n,\, d(x,y) < \delta_x \Rightarrow \|u_n(x) - u_n(y)\| < \varepsilon.$$

By compactness, $X = B(x_1;\delta_1) \cup \cdots \cup B(x_k;\delta_k)$. Since $u_n \to u$ pointwise on X,

$$\exists n_0;\, \forall\, n \geq n_0 \text{ and } \forall i,\, \|u_n(x_i) - u(x_i)\| < \varepsilon.$$

Therefore for each $n \geq n_0$ and for each $x \in X$, we have

$$\|u_n(x) - u(x)\| \leq \|u_n(x) - u_n(x_i)\| + \|u_n(x_i) - u(x_i)\| + \|u(x_i) - u(x)\| < 3\varepsilon.$$ □

COROLLARY A.6. *Any uniformly bounded equicontinuous sequence of continuous maps from a compact metric space X to a Banach space Y has a uniformly convergent subsequence on X.*

PROOF. If (u_n) is such sequence, by Ascoli-Arzela's Theorem **A.4**, $\overline{\{u_n\}}$ is compact and it has a convergent subsequence. By Proposition **A.5**, it converges uniformly on X. □

APPENDIX B

Fixed point Theorems

SCHAUDER'S FIXED POINT THEOREM B.1. *Let X be a real Banach space and K be a compact and convex subspace of X.*

If $u\colon K \to K$ is continuous, u has a fixed point in K.

PROOF. We will prove it in three steps:

- Fixed $\varepsilon > 0$. By compactness, let us consider
$$K \subset B(x_1;\varepsilon) \cup \cdots \cup B(x_n;\varepsilon). \tag{2.1}$$
Let K_ε be the closed convex hull of $x_1, \ldots, x_n$. That is,
$$K_\varepsilon = \left\{ \sum_{i=1}^{n} \lambda_i x_i \colon 0 \le \lambda_i \le 1 \text{ and } \sum_{i=1}^{n} \lambda_i = 1 \right\}.$$
By convexity, $K_\varepsilon \subset K$. Now, let us consider $P_\varepsilon\colon K \to K_\varepsilon$ defined by
$$x \mapsto \frac{\sum_{i=1}^{n} d(x, K \setminus B(x_i;\varepsilon))x_i}{\sum_{i=1}^{n} d(x, K \setminus B(x_i;\varepsilon))}$$
By (2.1), the denominator is non-zero. So P_ε is continuous. Moreover,
$$\|P_\varepsilon(x) - x\| \le \frac{\sum_{i=1}^{n} d(x, K \setminus B(x_i;\varepsilon))\|x_i - x\|}{\sum_{i=1}^{n} d(x, K \setminus B(x_i;\varepsilon))} \le \varepsilon.$$
for any $x \in K$.
- Let us consider $u_\varepsilon\colon K_\varepsilon \to K_\varepsilon$ defined by
$$u_\varepsilon(x) = (P_\varepsilon \circ u)(x)$$
Since K_ε is homeomorphic to the closed unit ball in $\mathbb{R}^m$ for some $m \le n$ and by Brouwer's fixed point Theorem, there is x_ε such that
$$u_\varepsilon(x_\varepsilon) = x_\varepsilon.$$
- By compactness, $x_{\varepsilon_j} \to x$ in X for some subsequence $\varepsilon_j \to 0$ and $x \in K$. Then
$$\|x_{\varepsilon_j} - u(x_{\varepsilon_j})\| = \|u_{\varepsilon_j}(x_{\varepsilon_j}) - u(x_{\varepsilon_j})\| = \|(P_{\varepsilon_j} \circ u)(x_{\varepsilon_j}) - u(x_{\varepsilon_j})\| \le \varepsilon_j.$$
By continuity, $u(x) = x$. □

DEFINITION B.2. Let U be a subset of a real Banach space X. A continuous map $u\colon U \to X$ is compact when for any bounded set $K \subset U$, $\overline{u(K)}$ is compact.

COROLLARY B.3. *Let K be a closed bounded and convex subset of a real Banach space X.*

If $u\colon K \to K$ is compact, it has a fixed point.

PROOF. Let W the convex closure of $u(K)$. By convexity of K, $W \subset u(K) \subset K$. By compactness of $\overline{u(K)} \subset K$, W is also compact. Then
$$W \subset K \Rightarrow u(W) \subset u(K) \subset K.$$
By Schauder's Fixed Point Theorem **B.1**, there is a fixed point in W. □

APPENDIX C

Uniform contractions

The aim of this part is to give a proof of differentiability of some types of contraction mappings under apparently easy conditions.
We assume that the contraction mapping principle is well-known, so the proof will be omitted (for details see [**2**] or [**6**]):

CONTRACTION MAPPING PRINCIPLE. *Let U be a closed subset of a complete metric space.*

If $u\colon U \to U$ is a contraction, it has a unique fixed point in U.

DEFINITION C.1. Let X, Y be Banach spaces, $U \subset X$ be open subset and $V \subset Y$. A map $u\colon U \times V \to V$ is a uniform contraction when there is $0 < \lambda < 1$ such that

i. u is continuous.
ii. For all $x \in U$ and for all $y, z \in V$,

$$\|u(x,y) - u(x,z)\| \leq \lambda \|y - z\|.$$

LEMMA C.2. *Given $u\colon U \times V \to V$ a uniform contraction with λ. If u is differentiable in the second variable, then*

$$\|D_2 u(x,y)\| \leq \lambda \qquad \forall\, (x,y) \in U \times V.$$

In particular, $id - D_2 u(x,y)$ is invertible.

PROOF. By the mean value Theorem,

$$\|u(x,y) - u(x,z)\| = \|D_2 u(x,y)(\xi)\| \|y - z\| \leq \lambda \|y - z\|.$$

Then $\|D_2 u(x,y)\| \leq \lambda < 1$. Indeed, since $D_2 u(x,y)$ is linear and continuous, by definition,

$$\|D_2 u(x,y)\| = \sup_{\|\xi\| \leq 1} \|D_2 u(x,y)(\xi)\|.$$

Let us distinct two possible cases:

- $\|\xi\| \leq 1$. Clearly $\|D_2 u(x,y)\| \leq \lambda$.
- $\|\xi\| > 1$. From $\frac{\lambda}{\|\xi\|} < \lambda$.

Finally, by Neumann's series, $id - D_2 u(x,y)$ is invertible because $\lambda < 1$. □

THEOREM C.3. *Let U be a subset of a Banach space, V be a closed subset of a Banach space. If $u\colon U \times V \to V$ is a uniform contraction and $g(x)$ is the unique fixed point of the mapping $u(x,\cdot)\colon V \to V$. Then*

i. *g is continuous.*
ii. *If u is $\mathcal{C}^1$, then g is locally Lipschitz.*
iii. *If u is $\mathcal{C}^p$, then g is $\mathcal{C}^p$. Moreover,*

$$Dg(x) = \big(id - D_2 u(x, g(x))\big)^{-1} \circ D_1 u(x, g(x)).$$

N.B. The conditions $\mathcal{C}^1$ and $\mathcal{C}^p$ are in open sets with closures contained in $U \times V$.

PROOF.

i. By continuity of u

$$\forall\, \varepsilon > 0,\ \exists\, \delta > 0;\ \|x - y\| < \delta \Rightarrow \|u(x, g(x)) - u(y, g(y))\| < \varepsilon.$$

By definition of g,

$$\begin{aligned}\|g(x)-g(y)\| &= \|u(x,g(x))-u(y,g(y))\| \\ &\le \|u(x,g(x))-u(x,g(y))\| + \|u(x,g(y))-u(y,g(y))\| \\ &\le \lambda\|g(x)-g(y)\| + \|u(x,g(y))-u(y,g(y))\|.\end{aligned}$$

So

$$\|g(x)-g(y)\| \le (1-\lambda)^{-1}\|u(x,g(y))-u(y,g(y))\| < (1-\lambda)^{-1}\varepsilon$$

and g is continuous.

ii. Fixed $x_0 \in U$, we choose $\varepsilon > 0$ and $C > 0$ such that

$$\left.\begin{array}{r}\|x-x_0\| < \varepsilon \\ \|y-g(x_0)\| < \varepsilon\end{array}\right\} \Rightarrow \|D_1u(x,y)\| \le C.$$

By continuity of g, there is $0 < \delta < \varepsilon$ such that

$$\|x-x_0\| < \delta \qquad \text{and} \qquad \|g(x)-g(x_0)\| < \varepsilon.$$

Thus, if $\|x-x_0\| < \delta$ and $\|y-x_0\| < \delta$, by the mean value Theorem,

$$\|g(x)-g(y)\| \le (1-\lambda)^{-1}\|u(x,g(y))-u(y,g(y))\| \le (1-\lambda)^{-1}C\|x-y\|.$$

Therefore g is locally Lipschitz.

iii. Since $g(x) = u(x,g(x))$, by chain's rule,

$$Dg(x) = D_1u(x,g(x)) + D_2u(x,g(x))Dg(x)$$

Let $A(x) = \big(id - D_2u(x,g(x))\big)^{-1}D_1u(x,g(x))$. We must show that $A(x) = Dg(x)$, that is

$$\lim_{h\to 0}\frac{g(x+h)-g(x)-A(x)h}{|h|} = 0.$$

Indeed, the numerator is equal to

$$\begin{aligned}&u(x+h,g(x+h)) - u(x,g(x)) - D_1u(x,g(x))h - D_2u(x,g(x))A(x)h \\ &= u(x+h,g(x+h)) - u(x+h,g(x)) - D_2u(x,g(x))A(x)h \\ &\qquad + u(x+h,g(x)) - u(x,g(x)) - D_1u(x,g(x))h \\ &= (g(x+h)-g(x))\int_0^1 D_2u\big(x+h,g(x)+s(g(x+h)-g(x))\big)\,ds \\ &\qquad - D_2u(x,g(x))A(x)h + h\int_0^1 D_1u(x+sh,g(x))\,ds - D_1u(x,g(x))h \\ &= (g(x+h)-g(x))\int_0^1 D_2u\big(x+h,g(x)+s(g(x+h)-g(x))\big) - D_2u(x,g(x))\,ds \\ &\qquad + D_2u(x,g(x))\big(g(x+h)-g(x)-A(x)h\big) \\ &\qquad + h\int_0^1 D_1u(x+sh,g(x)) - D_1u(x,g(x))\,ds.\end{aligned}$$

Therefore, $\|g(x+h)-g(x)-A(x)h\|$ is bounded by

$$\begin{aligned}&(1-\lambda)^{-1}\Bigg[|h|\sup_{0\le s\le 1}\|D_1u(x+sh,g(x)) - D_1u(x,g(x))\| \\ &+(g(x+h)-g(x))\sup_{0\le s\le 1}\|D_2u(x+h,g(x)+s\big(g(x+h)-g(x)\big) - D_2u(x,g(x))\|\Bigg].\end{aligned}$$

Since g is locally Lipschitz, then $\|g(x+h)-g(x)\| \le C|h|$. If $|h|$ is small enough, then

$$\|g(x+h)-g(x)-A(x)h\| \le (1-\lambda)^{-1}\varepsilon(C+1)|h|.$$

So $Dg(x) = A(x)$.

Now if $p = 1$, g is $\mathcal{C}^1$ because A is continuous. By induction, if u is $\mathcal{C}^p$, g is $\mathcal{C}^p$. □

APPENDIX D

Review of spectral theory

This section aims to fix notation and to give a fast review of an elemental spectral theory for linear and continuous maps between normed spaces with maybe some extra conditions. Some results are not going to prove. They can be found in [**16**, ch. VII] and [**2**, ch. XI].

1. Spectrum

DEFINITION D.1. Let X be a complex normed space and $u\colon X \to X$ be a linear and continuous map. Then

- A λ in $\mathbb{C}$ is a regular value when $u - \lambda \cdot id$ is a linear homeomorphism.
- A λ in $\mathbb{C}$ is a spectral value when it is not a regular value.
- The spectrum of u is defined by

$$\sigma(u) = \{\lambda \in \mathbb{C}\colon \lambda \text{ is a spectral value}\}.$$

- The point spectrum of u is the set of its eigenvalues, i.e.

$$\sigma_p(u) = \{\lambda \in \mathbb{C}\colon \ker(u - \lambda \cdot id) \neq \{0\}\}.$$

- The eigenspace of an eigenvalue λ is the set of its eigenvectors and the 0, i.e.

$$E(\lambda; u) = \ker(u - \lambda \cdot id).$$

Remark D.2. Clearly, $\sigma_p(u) \subset \sigma(u)$. If X is finite dimensional, then $\sigma_p(u) = \sigma(u)$.

NOTATION. $\sigma^*(u) := \sigma(u) \setminus \{0\}$ and $\sigma_p^*(u) := \sigma_p(u) \setminus \{0\}$.

2. Compact linear mappings

DEFINITION D.3. A linear map $u\colon X \to Y$ between normed spaces is compact when for any bounded subset $K \subset X$, $u(K)$ is relatively compact.

As immediate result:

COROLLARY D.4. *A linear map $u\colon X \to Y$ is compact when for any (x_n) bounded sequence of X, it has a subsequent (x_{n_k}) such that $(u(x_{n_k}))$ converges in Y.*

COROLLARY D.5. *Any compact linear map is continuous.*

PROOF. If $u\colon X \to Y$ is compact linear map, then $\|u\| < +\infty$ and it is continuous. □

PROPOSITION D.6. *Given u and v linear maps. $u \circ v$ is compact map whenever u or v is compact.*

PROOF. We distinct two cases:

- u is compact. Given K a bounded set, $v(K)$ is bounded by linearity. So $u(v(K))$ is relatively compact.
- v is compact. Given K a bounded set, $v(K)$ is relatively compact. So $u(v(K))$ is relatively compact by linearity. □

The next result tells us that any compact linear map whose infinite dimensional domain is not invertible.

COROLLARY D.7. *If $u\colon X \to X$ is an invertible compact linear map, then X has finite dimension. In particular, if u is compact linear map and X is infinite dimensional, $0 \in \sigma(u)$.*

PROOF. By Proposition **D.6**, *id* is compact. Then the unit ball is compact and X is finite dimensional. □

PROPOSITION D.8. *Let $u\colon X \to Y$ be a compact linear map and $E \subset X$ be a vector subspace. Then*

$$u \restriction E\colon E \longrightarrow \overline{u(E)} \text{ is compact.}$$

PROOF. Let K be a bounded subset of U. Then $u(K)$ is relatively compact in Y. Since $\overline{u(K)} \subset \overline{u(E)}$, then $u(K)$ is relatively compact in $\overline{u(E)}$. □

PROPOSITION D.9. *Let (u_n) be a sequence of compact maps from a normed space X to a Banach space Y. If $u_n \to u$, then u is compact.*

PROOF. Let K be a bounded subset of X. Since Y is complete, it is enough to show that $u(K)$ is precompact (Lemma **A.2**). By hypothesis,

$$K \subset B(0;\varepsilon).$$

By convergence, there is n such that $\|u_n - u\| < \varepsilon$. Since $u_n(K)$ is precompact,

$$u_n(K) \subset B(x_1;\varepsilon) \cup \cdots \cup B(x_r;\varepsilon).$$

So $u(K) \subset B(x_1;2\varepsilon) \cup \cdots \cup B(x_r;2\varepsilon)$. Indeed,

$$\|u(x) - x_i\| \leq \|u(x) - u_n(x)\| + \|u_n(x) - x_i\| < 2\varepsilon.$$ □

PROPOSITION D.10. *Let $u\colon X \to X$ be a linear map between a complex normed space. Given a polynomial $p(x) = a_n x^n + \cdots + a_0$. Then*

i. *$p(u)\colon X \to X$ is a linear map.*
ii. *$\sigma(p(u)) = p(\sigma(u))$. That is, $\sigma(p(u)) = \{\lambda\colon p(\mu) = \lambda \text{ for some } \mu \in \sigma(u)\}$.*

PROOF.

i. Clearly, $p(u) = a_n \cdot u^n + \cdots + a_0 \cdot id$ and it is linear if u is linear.
ii. Let us suppose that $n \geq 1$. Fixed λ, let the zeros of $p(x) - \lambda$ be $\alpha_1, \ldots, \alpha_n$, so that

$$p(u) - \lambda \cdot id = (u - \alpha_1 \cdot id) \cdots (u - \alpha_n \cdot id). \tag{4.1}$$

If $u - \alpha_1 \cdot id, \ldots, u - \alpha_n \cdot id$ have continuous inverses defined on all of X, then $p(u) - \lambda \cdot id$ also has continuous inverse defined on all of X. We must show two inclusions:

$\subset$) If $\lambda \in \sigma(p(u))$, there must be some α_k such that $\alpha_k \in \sigma(u)$. Since $p(\alpha_k) = \lambda$, this proves the inclusion.

$\supset$) Suppose now that some α_k is in $\sigma(u)$. Let us distinct two possible cases:

- $u - \alpha_k \cdot id$ has inverse. Its inverse is not continuous. So the inverse (when it exists) of $p(u) - \lambda \cdot id$ is not continuous by the relation (4.1). So $\lambda \in \sigma(p(u))$.
- $u - \alpha_k \cdot id$ has no inverse. Exchanging it with $u - \alpha_n \cdot id$ in (4.1), we obtain that $p(u) - \lambda \cdot id$ also has no inverse, so $\lambda \in \sigma(p(u))$. □

3. Topological direct sum

Any normed space is a topological vector space in the sense that the mappings of "sum of vectors" and "product by a scalar" are continuous with the topology induced by the norm. The assumption that a normed space is a vector space allows us to consider the direct sum of vector subspace. However, when we want to consider it a subtle notion appears and one should distinct the topological and algebraic point of views.

DEFINITION D.11.

- Z is an *algebraic* direct sum of vector subspaces X and Y when

i. $X \cap Y = \{0\}$.
ii. $X + Y = Z$.
It will be denoted by $Z = X \oplus Y$.

- Z is a *topological* direct sum of vector subspaces X and Y when
  i. $X \cap Y = \{0\}$.
  ii. $X + Y = Z$.
  iii. The map[1] $X \times Y \to X + Y$ defined by $(x, y) \mapsto x + y$ is a linear homeomorphism.
  It will also be denoted by $Z = X \oplus Y$.

PROPOSITION D.12. *Let X and Y be subspaces of a common normed space. There is equivalence between:*

i. *$X \oplus Y$ topologically.*
ii. *The projection map $\pi\colon X + Y \to X$ is linear, surjective and continuous.*

PROOF. Since any normed space is a topological vector space, the map $\rho\colon X \times Y \to X + Y$ is always continuous. It is also linear and $\|\rho\| \leq 2$. Its inverse is

$$\rho^{-1} = (\pi, id - \pi).$$

Therefore it remains to distinct if ρ has continuous inverse or π is continuous.

⇓) ρ^{-1} is continuous. So $\|\pi\| \leq \|\rho^{-1}\|$.
⇑) We have $\|\rho^{-1}\| \leq \|\pi\| + 1$, so ρ^{-1} is continuous. □

COROLLARY D.13. *Let X and Y be subspaces of a common normed space. If $X \oplus Y$ topologically, then X and Y are closed.*

PROOF. By Proposition **D.12**, the projection $\pi\colon X + Y \to X$ is linear and continuous. Then $X = \ker(id - \pi)$ is closed by continuity of π. □

THEOREM D.14. *Let X and Y be closed subspaces of a Banach space. Then*

$$X \oplus Y \text{ algebraically} \Rightarrow X \oplus Y \text{ topologically.}$$

PROOF. We will apply the open map Theorem. First of all, X and Y are also Banach space because they are closed in a Banach space. Let us consider two different norms in $X \times Y$.

$$(X \times Y)_1 \text{ with norm } \|(x, y)\|_1 = \|x\| + \|y\|.$$
$$(X \times Y)_\infty \text{ with norm } \|(x, y)\|_\infty = \sup\{\|x\|, \|y\|\}.$$

Let us consider now

$$(X \times Y)_\infty \xrightarrow{id} (X \times Y)_1 \xrightarrow{\rho} X + Y.$$

The assumption $X \oplus Y$ algebraically tells us that ρ is an isomorphism. Since ρ is always linear and continuous with $\|\rho\| = 1$, then ρ is a linear homeomorphism. On the other hand, id is bijective and continuous because $\|\cdot\|_1 \leq 2\|\cdot\|_\infty$. Thus, id is also a linear homeomorphism. □

4. Spectrum of a compact linear map

THEOREM D.15. *Let $u\colon X \to X$ be a compact linear map between a complex normed space. Then*

i. *$\sigma(u)$ is either finite or numerable in $\mathbb{C}$. Moreover, each element of $\sigma^*(u)$ is open.*
ii. *$\sigma^*(u) = \sigma_p^*(u)$.*
iii. *For each $\lambda \in \sigma^*(u)$, there are vector subspaces such that*
  a) *$X = N(\lambda) \oplus F(\lambda)$ topologically.*
  b) *$N(\lambda)$ is finite dimensional and $F(\lambda)$ is closed.*
  c) *$u\big(N(\lambda)\big) \subset N(\lambda)$ and there is an integer called rank of λ such that*

$$k = \min\{k \in \mathbb{Z}\colon k \geq 1 \text{ and } (u - \lambda \cdot id)^k\big(N(\lambda)\big) = \{0\}\}.$$

[1] Of course, $X \times Y$ is endowed with the supremum norm, that is, $\|(x, y)\| = \sup\{\|x\|, \|y\|\}$.

   d) $u\big(F(\lambda)\big) \subset F(\lambda)$ *and* $(u - \lambda \cdot id) \restriction F(\lambda)$ *is an homeomorphism.*
   e) $E(\lambda; u) \subset N(\lambda)$.
   f) $\sigma(u \restriction N(\lambda)) = \{\lambda\}$ *and* $\sigma(u \restriction F(\lambda)) = \sigma(u) \setminus \{\lambda\}$.
iv. *If* $\lambda \neq \mu$ *are in* $\sigma^*(u)$, *then* $N(\mu) \subset F(\lambda)$.

NOTATION. $\dim E(\lambda; u)$ is called geometric multiplicity and $\dim N(\lambda)$ algebraic multiplicity.

By Proposition **D.10** and Theorem **D.15** we deduce:

COROLLARY D.16. *Let* $u\colon X \to X$ *be a linear map between a complex normed space so that* u^n *is compact for some* n. *Then*

i. $\sigma(u)$ *is either finite or numerable in* $\mathbb{C}$. *Moreover, each element of* $\sigma^*(u)$ *is open.*
ii. $\sigma^*(u) = \sigma_p^*(u)$.
iii. *For each* $\lambda \in \sigma^*(u)$, *there are vector subspaces such that*
   a) $X = N(\lambda) \oplus F(\lambda)$ *topologically.*
   b) $N(\lambda)$ *is finite dimensional.*
   c) $u\big(N(\lambda)\big) \subset N(\lambda)$ *and* $u\big(F(\lambda)\big) \subset F(\lambda)$.
   d) $\sigma(u \restriction N(\lambda)) = \{\lambda\}$ *and* $\sigma(u \restriction F(\lambda)) = \sigma(u) \setminus \{\lambda\}$.

Bibliography

[1] À. Jorba and M. Zou. A software package for the numerical integration of ode by means of high-order taylor methods. *Experimental Mathematics*, (14):99–117, 2005.

[2] J. Dieudonne. *Fundamentos de análisis moderno*. Number v. 1. Reverté, 1979.

[3] L.C. Evans. *Partial Differential Equations*. Graduate studies in mathematics. American Mathematical Society, 2010.

[4] G.B. Folland. *Real Analysis: Modern Techniques and Their Applications*. Pure and Applied Mathematics: A Wiley Series of Texts, Monographs and Tracts. Wiley, 2013.

[5] George G. Hanrot and V. Lefèvre and P. Pélissier and P. Théveny and P. ZimmermannDoe. Multiple precision floating reliable (mpfr). http://www.mpfr.org/mpfr-current/mpfr.html, 2013.

[6] J. Gimeno. Aproximacions de corbes invariants per diffeomorfismes. Technical report, Universitat de Barcelona, 2014.

[7] A. Griewank and A. Walther. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation, Second Edition*. Society for Industrial and Applied Mathematics, 2008.

[8] J.K. Hale. *Functional differential equations*. Applied mathematical sciences. Springer-Verlag, 1971.

[9] J.K. Hale. *Ordinary Differential Equations*. Dover Books on Mathematics Series. Dover Publications, 2009.

[10] Sjoerd M. Hale, J.K. and Verduyn Lunel. *Introduction to Functional Differential Equations*, volume 99. Springer-Verlag, 1993.

[11] S. and Naitō T. Hino, Y. and Murakami. *Functional differential equations with infinite delay*. Lecture notes in mathematics. Springer-Verlag, 1991.

[12] N.G. Markley. *Principles of Differential Equations*. Pure and Applied Mathematics: A Wiley Series of Texts, Monographs and Tracts. Wiley, 2011.

[13] U. Naumann. *The Art of Differentiating Computer Programs: An Introduction to Algorithmic Differentiation*. Software, Environments, and Tools. Society for Industrial and Applied Mathematics, 2012.

[14] Robert Roose, Dirk and Szalai. Continuation and bifurcation analysis of delay differential equations. In *Numerical continuation methods for dynamical systems*, pages 359–399. Springer, 2007.

[15] C. Simó. Global dynamics and fast indicators. *Global analysis of dynamical systems*, pages 373–389, 2001.

[16] D.C. Taylor, A.E. and Lay. *Introduction to Functional Analysis*. R.E. Krieger Publishing Company, 1980.

[17] LAPACK team. Linear algebra package (lapack). http://www.netlib.org/lapack/, 1992.

Index