UNIVERSITAT DE BARCELONA

ADVANCED MATHEMATICS
MASTER'S FINAL PROJECT

Jet transport for General Linear methods

Author:
Philip Pita Forrier

Supervisor:
Àngel Jorba
Joan Gimeno

Facultat de Matemàtiques i Informàtica

June 28, 2023

Abstract

The following project deals with two main topics: General Linear methods (GLM) and jet transport. For their presentation, we have divided it in three chapters. In Chapter 1, we introduce the family of numerical integrators known as General Linear methods, which arise as a natural generalization of the so-known linear multistep (LMM) and Runge-Kutta (RK) methods. Throughout the chapter, we present the main properties of LMM and RK methods so that they can be compared with those obtained for GLM with greater generality. In Chapter 2, we introduce the technique known as jet transport for the numerical integration of variational equations. It is in this chapter where the main contributuion of this project is found: we prove that the numerical integration of an initial value problem using jet transport with General Linear methods is equivalent to the numerical integration of their variational equations with the same method. Not only that, but we also successfully derive the expressions that the higher order coefficients of the jets must satisfy to be a solution of an implicit system, thus allowing the effective implementation of implicit General Linear methods. In Chapter 3 we conclude this project by studying how implicit Runge-Kutta methods can be efficiently implemented using jet transport and we apply this implementation to study a few scenarios in the field of dynamical systems, where the computation of variational equations is of interest.

Contents

Chapter 1

General Linear Methods

In the study of the numerical solution of Ordinary Differential Equations (ODEs), different methods have been developed which are capable of exploiting distinct aspects of the problem to obtain better approximations. The main representatives in this field are the Euler method, linear multistep methods (LMM), Runge-Kutta (RK) methods and the Taylor method. It is interesting to observe that the last three are designed to be an improvement of the first, in the sense that to obtain a better solution, linear multistep methods reuse the information of previous steps; Runge-Kutta methods do more computations per step, known as stages; and the Taylor method computes high order derivatives of the solution. In this sense, these methods are respectively referred to as multistep, multistage and multiderivative. Unsurprisingly, it is of great value to combine these methods to obtain a more powerful one. While there are other interesting possibilities, we will focus on introducing the multistep-multistage methods, commonly known as General Linear methods. A practical way to illustrate these possible combinations is depicted in Figure 1.1, where moving to the left represents increasing the steps, to the right the stages and up the derivatives. Furthermore, we remark that this generalization is natural in the sense that; as we will see later, both LMM and RK methods suffer limitations respectively known as the Dahlquist and Butcher barriers, thus it is expected that by giving them more flexibility, similar methods will emerge that are not constrained by these restrictions.

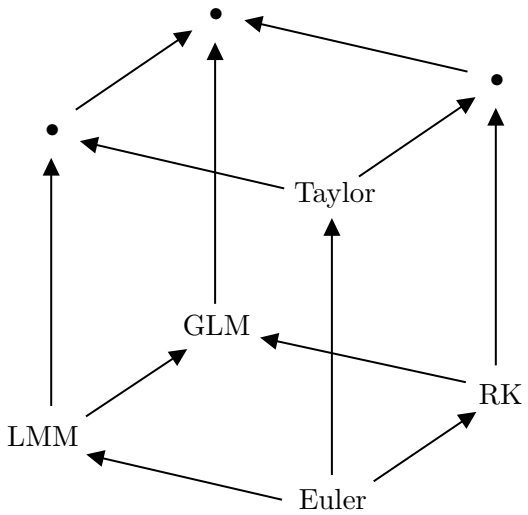

Figure 1.1: Visual representation (adapted from [But16]) of multistep, multistage and multiderivative methods and their combinations.

Another particularly interesting aspect of numerical integrators of ODEs is their ability to handle 'stiff' problems. Even though there is no globally accepted rigorous mathematical definition for stiffnes, the intuitive idea is that, as cited in [HW96], "stiff equations are equations where certain implicit methods [...] perform better, usually tremendously better, than explicit ones". Many mathematical tools have been developed in an attempt to solve these problems efficiently (the book [HW96] is mainly dedicated to them). We will present a few of them, like the properties of A-stability and L-stability, which are devised to ensure a sufficiently stable numerical solution to deal with stiffnes. In addition, in Chapters 2 and 3 we will study certain new possibilities for implicit methods.

1.1 Framework

Let us consider the initial value problem (IVP) or Cauchy problem

$$y'(x) = f(x, y(x)), \quad y(x_0) = y_0, \tag{1.1}$$

where $f : \mathbb{R} \times \mathbb{R}^N \to \mathbb{R}^N$. The independent variable x is often thought in physical systems as time, and the dependent variable $y(x)$ is the solution. In many situations, it will be useful to work with an autonomous version of the problem (1.1); that is, with no dependence of time. Observe that the IVP can always be presented in autonomous form by increasing the dimension of the problem by 1 and considering the following

$$u(x) = \begin{bmatrix} x \\ y(x) \end{bmatrix}, \qquad g(u(x)) = \begin{bmatrix} 1 \\ f(u(x)) \end{bmatrix}, \qquad u_0 = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}, \tag{1.2}$$

thus obtaining the autonomous initial value problem

$$u'(x) = g(u(x)), \quad u(x_0) = u_0, \tag{1.3}$$

where $g : \mathbb{R}^{N+1} \to \mathbb{R}^{N+1}$. We also remark that by studying these IVP we are in fact also studying higher order IVP problems like the following

$$\begin{aligned} &y^{(N+1)}(x) = f(y(x), y'(x), \ldots, y^{(N)}(x)), \\ &y(x_0) = y_{0,(0)},\ y'(x_0) = y_{0,(1)},\ \ldots,\ y^{(N)}(x_0) = y_{0,(N)}, \end{aligned}$$

because to obtain a formulation like the one of (1.3), we just need to consider

$$u(x) = \begin{bmatrix} u_1(x) \\ u_2(x) \\ \vdots \\ u_{N+1}(x) \end{bmatrix} = \begin{bmatrix} y(x) \\ y'(x) \\ \vdots \\ y^{(N)}(x) \end{bmatrix},\ g(u(x)) = \begin{bmatrix} u_2(x) \\ u_3(x) \\ \vdots \\ f(u(x)) \end{bmatrix},\ u_0 = \begin{bmatrix} y_{0,(0)} \\ y_{0,(1)} \\ \vdots \\ y_{0,(N)} \end{bmatrix}$$

As we will study the numerical solutions of problems of this kind, it is essential to consider if such a solution even exists and if it is unique. For answering that question for both exact and numerical solutions, we will refer to the Lipschitz condition.

Definition 1.1. A function $f : \mathbb{R}^N \to \mathbb{R}^N$ satisfies the Lipschitz condition if there is a constant L and a norm $\|\cdot\|$ such that for any $a, b \in \mathbb{R}^N$,

$$\|f(a) - f(b)\| \leq L\|a - b\| \ .$$

Theorem 1.1. There exists a unique solution for an IVP (1.1) when $f : \mathbb{R} \times \mathbb{R}^N \to \mathbb{R}^N$ is continuous in the independent variable and satisfies the Lipschitz condition in the dependent variable.

1.2 Linear Multistep Methods

A linear multistep method (LMM) with k-steps or linear k-step method is of the form

$$\sum_{i=0}^{k} \alpha_i y_{n-i} = h \sum_{i=0}^{k} \beta_i f(x_{n-i}, y_{n-i}) \ , \tag{1.4}$$

where the values y_{n-i} are approximations of $y(x_{n-i})$ for $i = 0, \ldots, k$. In order to be able to compute the value y_n from the formula (1.4), the values y_{n-i} for $i = 1, \ldots, k$ must be known and also, it should be $\alpha_0 \neq 0$. The sequences of coefficients $(\alpha_i)_{i=0}^k$, $(\beta_i)_{i=0}^k$ are characteristic of the method and we will require that $|\alpha_k| + |\beta_k| > 0$; that is, that α_k and β_k are not both 0, so that the method actually involves k steps, but that can always be achieved by reducing k if necessary.

For certain LMM, it is customary to use a different notation where the value y_n is in the left-hand side and all the others on the right-hand side. The idea is that, as we require $\alpha_0 \neq 0$, we can choose $\alpha_0 = 1$ and rescale the other coefficients accordingly. If we also take the coefficients α_i for $i = 1, \ldots, k$ with opposite sign, we can rewrite the expression of LMM on (1.4) as

$$y_n = \sum_{i=1}^{k} \alpha_i y_{n-i} + h \sum_{i=0}^{k} \beta_i f(x_{n-i}, y_{n-i}) \ . \tag{1.5}$$

As the value to compute y_n appears explicitly defined if $\beta_0 = 0$ and implicitly otherwise, the method is called explicit in the first case and implicit in the second.

In the study of the properties of LMM (1.4) it is convenient to introduce the so-called generating polynomials (ρ, σ), and it is even customary to identify LMM with the pair, where

$$\rho(w) = \sum_{i=0}^{k} \alpha_i w^{k-i}, \quad \sigma(w) = \sum_{i=0}^{k} \beta_i w^{k-i} \ .$$

Example. We introduce two of the most representative families of LMM methods: the Adams-Bashforth and Adams-Moulton methods. For simplicity, we display the coefficients in a table together with the number of steps k. These coefficients belong to the representation using notation (1.5), which is appropriate because they satisfy $\alpha_i = 0$ for $i > 1$.

We notice that the first family has $\beta_0 = 0$ and the second does not, thus the Adams-Bashforth methods are explicit and the Adams-Moulton methods are implicit. In particular, it is interesting to mention that the Adams-Bashforth with $k = 1$ is the explicit Euler method and the Adams-Moulton with $k = 0$ is the implicit Euler method.

Adams-Bashforth methods

k	α_0	α_1	α_2	α_3	α_4	β_0	β_1	β_2	β_3	β_4
1	1	1				0	1			
2	1	1	0			0	$\frac{3}{2}$	$-\frac{1}{2}$		
3	1	1	0	0		0	$\frac{23}{12}$	$-\frac{16}{12}$	$\frac{5}{12}$	
4	1	1	0	0	0	0	$\frac{55}{24}$	$-\frac{59}{24}$	$\frac{37}{24}$	$-\frac{9}{24}$

Adams-Moulton methods

k	α_0	α_1	α_2	α_3	β_0	β_1	β_2	β_3
0	1	1			1	0		
1	1	1	0		$\frac{1}{2}$	$\frac{1}{2}$		
2	1	1	0	0	$\frac{15}{12}$	$\frac{8}{12}$	$-\frac{1}{12}$	
3	1	1	0	0	$\frac{9}{24}$	$\frac{19}{24}$	$-\frac{5}{24}$	$\frac{1}{24}$

Let us also introduce another important family of methods, the 'Backward Differentiation Formulas' or BDF methods. Again, we give them in a table, displaying the number of steps k and their associated coefficients, but using the formulation (1.4) this time. It is interesting to remark how different is the usage of coefficients α_i and β_i between this family and the ones introduced before. Notice that, as $\beta_0 \neq 0$, the BDF methods are implicit.

BDF methods

k	α_0	α_1	α_2	α_3	α_4	α_5	α_6	β_0	β_1	β_2	β_3	β_4	β_5	β_6
1	1	-1						1	0					
2	$\frac{3}{2}$	-2	$\frac{1}{2}$					1	0	0				
3	$\frac{11}{6}$	-3	$\frac{3}{2}$	$-\frac{1}{3}$				1	0	0	0			
4	$\frac{25}{12}$	-4	3	$-\frac{4}{3}$	$\frac{1}{4}$			1	0	0	0	0		
5	$\frac{135}{60}$	-5	5	$-\frac{10}{3}$	$\frac{5}{4}$	$-\frac{1}{5}$		1	0	0	0	0	0	
6	$\frac{147}{60}$	-6	$\frac{15}{2}$	$-\frac{20}{3}$	$\frac{15}{4}$	$-\frac{6}{5}$	$\frac{1}{6}$	1	0	0	0	0	0	0

1.2.1 Local error and order

Definition 1.2. The local error of the LMM (1.4) is

$$y(x_n) - y_n$$

where the numerical solution y_n has been obtained from (1.4) using exact starting values; i.e. $y_{n-i} = y(x_{n-i})$, $i = 1, \dots, k$.

Observe that the assumption of knowing the exact starting values when defining the local error of a LMM is rather unusual. The idea of this definition is to consider only the error of the actual method and not that of the underlying method to approximate the required initial steps. When dealing with General Linear methods we will face the same problem, so the discussion on this topic will be delayed until they are introduced.

Definition 1.3. The LMM (1.4) is of order p if one of the following conditions is satisfied:

i) for all sufficiently regular functions $y(x)$, we have $L(y, x, h) = \mathcal{O}(h^{p+1})$, where

$$L(y, x, h) = \sum_{i=0}^{k} \Big(\alpha_i y(x + (k-i)h) - h\beta_i y'(x + (k-i)h) \Big).$$

ii) for all sufficiently regular differential equations (1.1), the local error of (1.4) is $\mathcal{O}(h^{p+1})$.

It can be easily shown (see [HWN93]) that both conditions above are equivalent. Studying them we can derive the so-called order conditions, stated at the following theorem.

Theorem 1.2. The LMM (1.4) is of order p if and only if one of the following equivalent conditions hold:

i) $\sum_{i=0}^{k} \alpha_i = 0$ and $\sum_{i=0}^{k} \alpha_i (k-i)^j = j \sum_{i=0}^{k} \beta_i (k-i)^{j-1}$ for all $j = 1, \dots, p$.

ii) $\rho(e^h) - h\sigma(e^h) = \mathcal{O}(h^{p+1})$ for $h \to 0$.

These order conditions can be easily derived (see [HWN93]) by inserting the Taylor series with respect to h of the exact solution and its derivative in the expression $L(y, x, h)$. We will later see that this is not as simple for Runge-Kutta methods nor General Linear methods.

Example.

1. The family of Adams-Bashforth methods has order $p = k$.
2. The family of Adams-Moulton methods has order $p = k + 1$.
3. The family of BDF methods has order $p = k$.

1.2.2 Convergence, stability and consistency

We now introduce the concept of convergence of a LMM, which states that the numerical solution actually approaches the exact solution as the step size decreases. Then, we will characterize it in terms of what we will call consistency and zero-stability. There are several equivalent ways of defining these properties, but we will do so in a manner that allows us to compare them with the ones of General Linear methods when they are introduced.

Definition 1.4. A LMM (1.4) is convergent if for any IVP (1.1) defined for $x \in [x_0, X]$ that satisfies the Lipschitz condition 1.1, we have that y_n computed using n steps of the LMM with $h = (x - x_0)/n$ converges to $y(x)$ for any $x \in [x_0, X]$, where y is a solution to (1.1).

Dahlquist famously studied in [Dah56] the situation in which LMM, even having high order and small local error had an 'unstable' solution; that is, the numerical solution given by the LMM was 'bad' even for very small step sizes h. One way of defining the condition for the method to be 'stable' is to require that the numerical solution y_n given by the LMM of the IVP

$$y' = 0 \ , \qquad y(x_0) = 0$$

is bounded as $n \to \infty$. Observe that for this IVP, the value y_n of the LMM (1.4) satisfies

$$\alpha_0 y_n + \alpha_1 y_{n-1} + \cdots + \alpha_k y_{n-k} = 0 \ ,$$

where this is related to the generating polynomial $\rho(w)$ by the substitution $y_{n-i} = w^{k-i}$. This fact is what motivates the definition of zero-stability. In the definition below, the condition given induces the boundedness of the solution y_n.

Definition 1.5. The LMM (1.4) is called zero-stable if the generating polynomial $\rho(w)$ satisfies the root condition; that is,

i) The roots of $\rho(w)$ lie in the closed unit disc $\{z \in \mathbb{C} \ ; \ |z| \leq 1\}$.

ii) The roots on the unit circle $\{z \in \mathbb{C} \ ; \ |z| = 1\}$ are simple.

In the same spirit as in the definition above, we can understand the concept of consistency of a LMM (1.4) as its capacity to solve an IVP. For that, we will first require the LMM to solve exactly the IVP

$$y' = 0 \ , \qquad y(x_0) = 1$$

starting from the exact initial value. It is clear that this happens when

$$\sum_{i=0}^{k} \alpha_i = 0 \ , \tag{1.6}$$

and in that case we will say that the method is preconsistent. If a preconsistent LMM method is also able to solve the IVP

$$y' = 1 \ , \qquad y(x_0) = 0$$

exactly starting from the exact initial value, we will say that it is consistent, and again, it can be checked that this happens when the LMM (1.4) satisfies (1.6) and also

$$\sum_{i=0}^{k} (k - i)\alpha_i = \sum_{i=0}^{k} \beta_i \ .$$

We observe that the conditions needed for a LMM (1.4) to be consistent are equivalent to the order conditions of order 1 given in Theorem 1.2 i), and also, that they can be stated in terms of the generating polynomials of the LMM. These facts motivate the following definitions.

Definition 1.6. A LMM (1.4) is preconsistent if it satisfies

$$\rho(1) = 0.$$

Definition 1.7. A LMM (1.4) is consistent if it is preconsistent and also satisfies

$$\rho'(1) = \sigma(1).$$

Using these concepts, we can state the culminating theorem on the convergence of LMM.

Theorem 1.3. A LMM (1.4) is convergent if and only if it is zero-stable and consistent.

Example.

1. The first generating polynomial of the Adams-Bashforth and Adams-Moulton families is $\rho(w) = w^k - w^{k-1}$, so 1 is a simple root and 0 is a root of multiplicity $k-1$, therefore all the methods of these families are zero-stable. It can also be checked that they all are consistent, so by the last theorem, they are convergent.

2. The first generating polynomial of the BDF family is $\rho(w) = \sum_{j=1}^{k} \frac{1}{j} w^{k-j}(w-1)^j$, so the study of its roots and thus of its zero-stability is more difficult. It can be proven (see [HWN93]) that BDF methods are zero-stable if and only if $k \leq 6$, thus the ones introduced in the Table at the beginning of the section are the only ones that are zero-stable. It can also be checked that they all are consistent, so by the last theorem, for $k \leq 6$ they are convergent.

1.2.3 Linear stability

As in the case of zero-stability, we are interested in the boundedness of the numerical solution y_n given by a LMM when applied to the linear test problem

$$y' = \lambda y \ , \quad \lambda \in \mathbb{C}$$

where we observe that the value of y_n given by the LMM (1.4) satisfies for this problem

$$\sum_{i=0}^{k} \alpha_i y_{n-i} - h\lambda \sum_{i=0}^{k} \beta_i y_{n-i} = 0 \ .$$

Using the substitution $y_{n-i} = w^{k-i}$, we notice that the expression is related with the generating polynomials (ρ, σ). Choosing $z = h\lambda$, the study of the resulting polynomial expression motivates the definition of the following function.

Definition 1.8. The stability function of a LMM (1.4) is

$$\Phi(w, z) = \rho(w) - z\sigma(w) \ .$$

Definition 1.9. A LMM (1.4) is said to be absolutely stable for a given $z \in \mathbb{C}$ if all roots $w_i = w_i(z)$ $i = 1, 2, \ldots, k$ of the stability function $\Phi(w, z)$ are inside the unit circle.

Definition 1.10. The region $\mathcal{A}$ of absolute stability of a LMM (1.4) is the set of all $z \in \mathbb{C}$ such that the method is absolutely stable; that is

$$\mathcal{A} = \Big\{ z \in \mathbb{C} \ \ ; \ \ |w_i(z)| < 1 \ , \ \ i = 1, 2, \ldots, k \Big\}.$$

Definition 1.11. A LMM (1.4) is A-stable if its region of absolute stability $\mathcal{A}$ contains the negative complex half-plane $\mathbb{C}^-$; that is,

$$\Big\{ z \in \mathbb{C} \ \ ; \ \ \mathrm{Re}(z) < 0 \Big\} \subset \mathcal{A}.$$

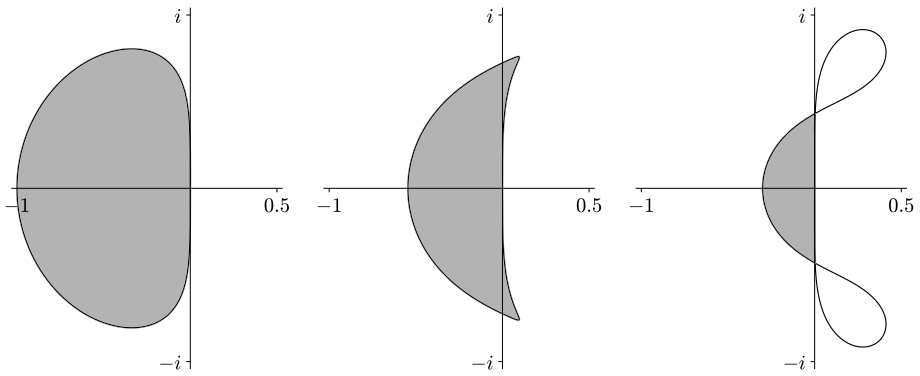

Figure 1.2: Stability region (shaded area) of the Adams-Bashforth methods for $k = 2, 3, 4$ respectively.

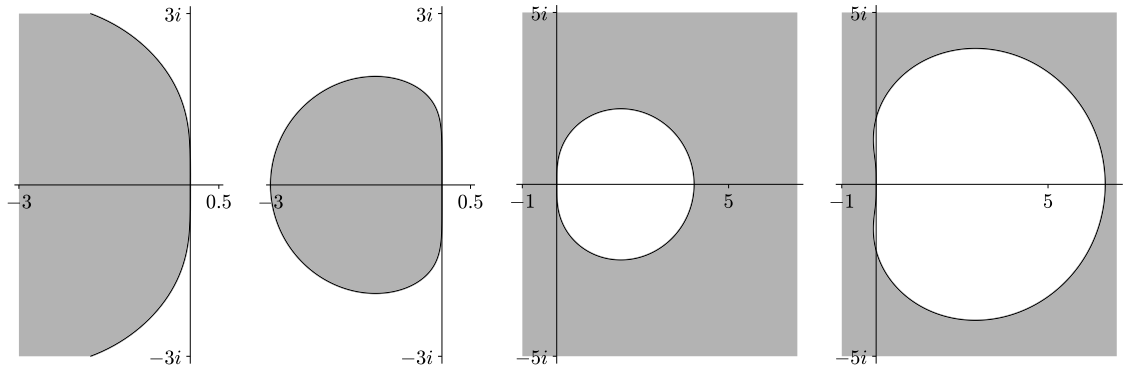

Figure 1.3: Stability region (shaded area) of the Adams-Moulton methods for $k = 2, 3$ and the BDF methods for $k = 2, 3$ respectively.

Example. Let us plot the stability regions of some of the methods introduced before. We observe from Figure 1.2 that the Adams-Bashforth methods for $k = 2, 3, 4$ are not A-stable, as the region of stability is bounded and thus it does not contain $\mathbb{C}^-$. Furthermore, we observe that the region shrinks as k increases.

Recalling their definition, the Adams-Moulton and BDF families are implicit so we expect them to perform better in terms of A-stability. We observe from Figure 1.3 that the Adams-Moulton methods for $k = 2, 3$ display a similar behavior to the Adams-Bashforth methods in Figure 1.2, and in particular they are not A-stable. Nevertheless, we can see that BDF methods do not have a bounded stability region and in particular the method for $k = 2$ is A-stable. Despite including almost all $\mathbb{C}^-$, the BDF method with $k = 3$ is not A-stable, because there is a small sector of the stability region around $-i/2$ and $i/2$ that is not contained in $\mathbb{C}^-$.

1.2.4 Order and stability barriers

Recalling the order conditions of Theorem 1.2, one can see that for a method of order p, the parameters have to satisfy $p + 1$ linear equations. We have seen that we can write the LMM (1.4) as in (1.5), so there are $2k$ free parameters, what suggests that the highest order that can be attained is $2k$. The following result, usually known as the first Dahlquist barrier, shows why such high order methods are not of interest.

Theorem 1.4. The order p of a zero-stable LMM with k steps is bounded by

$$p \leq \begin{cases} k+2 & \text{if } k \text{ is even,} \\ k+1 & \text{if } k \text{ is odd,} \\ k & \text{if } \beta_k/\alpha_k \leq 0 \text{ (in particular if the method is explicit).} \end{cases}$$

The next result, known as the second Dahlquist barrier, shows that there are no high-order A-stable LMM, proving what we observed in the last section when studying the stability regions of some LMM.

Theorem 1.5. The order of an A-stable linear multistep method cannot be greater than 2.

1.3 Runge-Kutta Methods

An s-stage Runge-Kutta method (RK) is of the form

$$k_i = f\Big(x_{n-1} + c_i h, y_{n-1} + h\sum_{j=1}^{s} a_{ij}k_j\Big) \ , \qquad i = 1, \dots, s, \tag{1.7}$$

$$y_n = y_{n-1} + h\sum_{i=1}^{s} b_i k_i \ . \tag{1.8}$$

Observe that the method is characterized by the coefficient matrix $A = [a_{ij}]$ and vectors $b = [b_i],\ c = [c_i]$, so it is customary to identify the method by what is known as the Butcher tableau

$$\begin{array}{c|ccc} c_1 & a_{11} & \dots & a_{1s} \\ \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & \dots & a_{ss} \\ \hline & b_1 & \dots & b_s \end{array} \quad = \quad \begin{array}{c|c} c & A \\ \hline & b^T \end{array} \ .$$

When the matrix A of coefficients is lower triangular with null diagonal elements; that is, $a_{ij} = 0$ for $i \leq j$, we say that the RK method is explicit (ERK), as the stages $k_1, \dots, k_s$ define an explicit system, and implicit otherwise. Some particular structures of the matrix A are of great interest, like diagonal implicit Runge-Kutta methods (DIRK), with A lower triangular with at least one non-zero diagonal element, or singly diagonal implicit Runge-Kutta methods (SDIRK), with A lower triangular with all identical diagonal elements.

In the literature, some authors use another representation for RK methods (1.7), given by considering

$$Y_i = y_{n-1} + h\sum_{j=1}^{s} a_{ij}k_j \ , \qquad i = 1, \dots, s,$$

and observing that we the RK method (1.7) can be rewritten as

$$\begin{aligned} Y_i &= y_{n-1} + h\sum_{j=1}^{s} a_{ij} f(x_{n-1} + c_j h, Y_j) \ , \qquad i = 1, \dots, s \\ y_n &= y_{n-1} + h\sum_{i=1}^{s} b_i f(x_{n-1} + c_i h, Y_i) \end{aligned} \tag{1.9}$$

Example. We introduce a few representatives of the multiple families of RK methods by giving their Butcher tableau. First, we show how the explicit and implicit Euler methods are formulated as RK methods. We also present the RK4 method, which is explicit and of order 4, and usually known as 'The' Runge-Kutta method or the 'Classic' Runge-Kutta method, which is probably the most popular RK method.

Explicit Euler

0	
	1

Implicit Euler

1	1
	1

RK4

0				
1/2	1/2			
1/2	0	1/2		
1	0	0	1	
	1/6	2/6	2/6	1/6

As an example of implicit RK methods, we have included the representative of order 4 of the Gauss methods, which are based on the Gaussian quadrature formulas, and the representative of order 5 of the Radau IIA methods which are based on the Radau quadrature formulas.

Gauss 4

$\frac{1}{2}-\frac{\sqrt{3}}{6}$	$\frac{1}{4}$	$\frac{1}{4}-\frac{\sqrt{3}}{6}$
$\frac{1}{2}+\frac{\sqrt{3}}{6}$	$\frac{1}{4}+\frac{\sqrt{3}}{6}$	$\frac{1}{4}$
	$\frac{1}{2}$	$\frac{1}{2}$

Radau IIA 5

$\frac{2}{5}-\frac{\sqrt{6}}{10}$	$\frac{11}{45}-\frac{7\sqrt{6}}{360}$	$\frac{37}{255}-\frac{169\sqrt{6}}{1800}$	$-\frac{2}{255}+\frac{\sqrt{6}}{75}$
$\frac{2}{5}+\frac{\sqrt{6}}{10}$	$\frac{37}{255}+\frac{169\sqrt{6}}{1800}$	$\frac{11}{45}+\frac{7\sqrt{6}}{360}$	$-\frac{2}{255}-\frac{\sqrt{6}}{75}$
1	$\frac{4}{9}-\frac{\sqrt{6}}{36}$	$\frac{4}{9}+\frac{\sqrt{6}}{36}$	$\frac{1}{9}$
	$\frac{4}{9}-\frac{\sqrt{6}}{36}$	$\frac{4}{9}+\frac{\sqrt{6}}{36}$	$\frac{1}{9}$

1.3.1 Local error and order

Definition 1.12. The local error of the RK method (1.7) is

$$y(x_{n-1}+h)-y_n\ .$$

Definition 1.13. A RK method (1.7) has order p if for sufficiently smooth problems (1.1)

$$y(x_{n-1}+h)-y_n=\mathcal{O}(h^{p+1})\ .$$

It is well known that deriving the order conditions for RK methods is not as straightforward as for LMM, as we need to compute high order derivatives of y to compare the coefficients of the Taylor expansion of the exact and numerical solutions. For that, we could use the elegant theory of rooted trees and B-series to give a general expression for the conditions, but that is out of the scope of this text (see [But06], [But16], [HWN93]). Nevertheless, we give the order conditions up to $p=4$ for reference:

$$\begin{aligned}
p=1:&\quad b^T\mathbf{e}=1,\\
p=2:&\quad (p=1),\ b^Tc=\frac{1}{2}\ ,\\
p=3:&\quad (p=2),\ b^TAc=\frac{1}{6},\ b^Tc^2=\frac{1}{3},\\
p=4:&\quad (p=3),\ b^TA^2c=\frac{1}{24},\ b^TAc^2=\frac{1}{12},\ b^TCAc=\frac{1}{8},\ b^Tc^3=\frac{1}{4}
\end{aligned}$$

where $\mathbf{e}=[1,\dots,1]^T\in\mathbb{R}^s$, $c^k=[c_1^k,\dots,c_s^k]^T$ for each $k\in\mathbb{N}$ and $C=\operatorname{diag}(c)$ is a diagonal matrix with c on the diagonal.

Example.

1. Although they have not been formally presented, the s-stage Gauss method is of order $2s$.
2. Similarly, the s-stage Radau IIA methods is of order $2s-1$.

1.3.2 Convergence, stability and consistency

In contrast with LMM, the concept of zero-stability is not an issue for RK methods as it is implied by their consistency. Moreover, we will see that consistency conditions are equivalent to the order conditions of order 1, thus deducing that convergence is not an issue for RK methods.

Analogously to LMM, a RK (1.7) method is said to be preconsistent if it is able to solve exactly the IVP

$$y' = 0 \ , \qquad y(x_0) = 1 \ .$$

Applying the RK method (1.7) to this problem, we notice that it is of the form

$$\begin{aligned} k_i &= 0 \ , \qquad i = 1, \dots, s \ , \\ y_n &= y_{n-1} + h \sum_{i=1}^{s} b_i k_i \end{aligned}$$

and thus it must be that $y_n = y_{n-1}$, obtaining the exact solution of the IVP. We then deduce that all RK methods are preconsistent. For consistency we will proceed in a similar fashion and consider the IVP

$$y' = 1 \ , \qquad y(x_0) = 0 \ , \tag{1.10}$$

and observe that the RK method (1.9) applied to it is of the form

$$y_n = y_{n-1} + h \sum_{i=1}^{s} b_i \ .$$

Being the exact solution of the IVP $y(x) = x$, the RK method is consistent if

$$x_{n-1} + h = x_{n-1} + h \sum_{i=1}^{s} b_i \implies b^T \mathbf{e} = 1 \ ,$$

which is the same as the order condition of order 1.

Another thing that we must have in consideration is that, by applying a RK method to a non-autonomous IVP, the vector of coefficients c is used, but it will not if we convert it to an autonomous IVP using (1.2). In order to obtain the same solution regardless of the formulation, let us study this 'phenomenon'. Applying the RK method (1.7) to the non-autonomous IVP (1.1), we obtain the following expression for the stages

$$Y_i = y_{n-1} + h \sum_{j=1}^{s} a_{ij} f(x_{n-1} + c_j h, Y_j) \ , \qquad i = 1, \dots, s$$

and when we apply it to the 'autonomization' of the problem (1.3), we obtain.

$$U_i = u_{n-1} + h \sum_{j=1}^{s} a_{ij} g(U_j) \ , \qquad i = 1, \dots, s$$

If we now rewrite the last expression in terms of (1.1), we get

$$\begin{bmatrix} U_{i,0} \\ Y_i \end{bmatrix} = \begin{bmatrix} x_{n-1} \\ y_{n-1} \end{bmatrix} + h \sum_{j=1}^{s} a_{ij} \begin{bmatrix} 1 \\ f(U_{j,0}, Y_j) \end{bmatrix} \ , \qquad i = 1, \dots, s$$

or equivalently, for all $i = 1, \dots, s$

$$\left.\begin{aligned} U_{i,0} &= x_{n-1} + h\sum_{j=1}^{s} a_{ij} \\ Y_i &= y_{n-1} + h\sum_{j=1}^{s} a_{ij} f(U_{j,0}, Y_j) \end{aligned}\right\} \implies Y_i = y_{n-1} + h\sum_{j=1}^{s} a_{ij} f(x_{n-1} + h\sum_{k=1}^{s} a_{jk}, Y_j) \ ,$$

so we deduce that for the expression of the stages Y_i to coincide regardless of the formulation, it happen that

$$c_j = \sum_{k=1}^{s} a_{jk} \quad \text{or equivalently,} \quad A\mathbf{e} = c \ .$$

This condition is usually known as stage-consistency, as in fact, we are requiring the stages Y_i to be approximations of order 1 to $y(x_{n-1}+c_i h)$; that is, that they are consistent. This condition is usually used for high order methods, as it simplifies the order conditions, but it is not necessary.

When we introduce General Linear methods, we will use for simplicity IVP in autonomous form, so in order to deal with this issue, we will also define the concept of stage-consistency and require the stages to be (possibly low order) approximations of the solution.

1.3.3 Linear stability

Let us study the boundedness of the numerical solution y_n when applied to the linear test problem

$$y' = \lambda y \ , \quad \lambda \in \mathbb{C} \ .$$

Observe that the RK method (1.9) applied to this problem is of the form

$$\begin{aligned} Y_i &= y_{n-1} + h\sum_{j=1}^{s} a_{ij}\lambda Y_j \ , \qquad i = 1, \dots, s, \\ y_n &= y_{n-1} + h\sum_{i=1}^{s} b_i \lambda Y_i \ , \end{aligned}$$

so by defining $Y = [Y_1, \dots, Y_s]^T$, $\mathbf{e} = [1, \dots, 1]^T \in \mathbb{R}^s$, we have that the first relation is

$$Y = y_{n-1}\mathbf{e} + h\lambda A Y \implies Y = y_{n-1}(I - hA)^{-1}\mathbf{e} \ ,$$

and substituting Y into the second and defining $z := h\lambda$, we obtain

$$y_n = y_{n-1} + h\lambda b^T y_{n-1}(I - h\lambda A)^{-1}\mathbf{e} = \Big(1 + zb^T(I - zA)^{-1}\mathbf{e}\Big)y_{n-1} \ ,$$

what motivates the following definitions.

Definition 1.14. The stability function of a RK method (1.9) is

$$R(z) = 1 + zb^T(I - zA)^{-1}\mathbf{e} \ .$$

Definition 1.15. The region $\mathcal{A}$ of stability of a RK method (1.7) is the set

$$\mathcal{A} = \Big\{ z \in \mathbb{C} \ ; \ |R(z)| \leq 1 \Big\}.$$

Definition 1.16. A RK method (1.7) is A-stable if its region of stability $\mathcal{A}$ contains the negative complex half-plane $\mathbb{C}^-$; that is,

$$\Big\{ z \in \mathbb{C} \ ; \ \mathrm{Re}(z) \leq 0 \Big\} \subset \mathcal{A}.$$

Definition 1.17. A RK method (1.7) is L-stable if its A-stable and

$$\lim_{z\to\infty} R(z) = 0 \ .$$

Example.

1. The stability function of the explicit and implicit Euler methods are respectively

$$R(z) = 1 + z \ , \qquad R(z) = \frac{1}{1-z} \ .$$

2. The stability function of the Radau IIA 5 method is

$$R(z) = \frac{1 + 2z/5 + z^2/20}{1 - 3z/5 + 3z^2/20 - z^3/60} \ .$$

Example.

1. All Gauss methods are A-stable (see [But16]).
2. All Radau IIA methods are L-stable (see [But16]).

1.3.4 Order barriers

When studying LMM we discussed the Dahlquist barriers, where the order of a stable method was limited by the number of steps. In a similar manner, we see in the following theorems (see [But16]), that the order of an explicit RK method is limited by its number of stages, known as the Butcher barriers.

Theorem 1.6. No explicit RK method exists of order p with $s < p$ stages.

In fact, this situation gets even worse as the order increases, as shown in the next theorem.

Theorem 1.7.

i) For $p \geq 5$ no explicit RK method exists of order p with $s = p$ stages.

ii) For $p \geq 7$ no explicit RK method exists of order p with $s = p + 1$ stages.

iii) For $p \geq 8$ no explicit RK method exists of order p with $s = p + 2$ stages.

1.4 General Linear Methods

An s-stage r-step General Linear method (GLM) is of the form

$$\begin{aligned} Y_i^{[n]} &= h\sum_{j=1}^{s} a_{ij} f(Y_j^{[n]}) + \sum_{j=1}^{r} u_{ij} y_j^{[n-1]}, \qquad i = 1, \dots, s, \\ y_i^{[n]} &= h\sum_{j=1}^{s} b_{ij} f(Y_j^{[n]}) + \sum_{j=1}^{r} v_{ij} y_j^{[n-1]}, \qquad i = 1, \dots, r, \end{aligned} \tag{1.11}$$

where the method is characterized by the vector $\mathbf{c} = [c_1, \dots, c_s]^T$ and four coefficient matrices

$$\mathbf{A} = [a_{ij}] \in \mathbb{R}^{s\times s} \ , \qquad \mathbf{U} = [u_{ij}] \in \mathbb{R}^{s\times r} \ , \qquad \mathbf{B} = [b_{ij}] \in \mathbb{R}^{r\times s} \ , \qquad \mathbf{V} = [v_{ij}] \in \mathbb{R}^{r\times r} \ .$$

The internal stages $Y_i^{[n]}$ are (possibly low order) approximations to $y(x_{n-1} + c_i h)$ and the external stages $y_i^{[n]}$ are approximations to the linear combination of the derivatives of y at x_n. It is customary to represent the GLM by the vector $\mathbf{c}$ and the partitioned $(s+r)\times(s+r)$ matrix

$$\left[\begin{array}{c|c} \mathbf{A} & \mathbf{U} \\ \hline \mathbf{B} & \mathbf{V} \end{array}\right],$$

as this is a simplification of the matrix that appears when introducing the notation

$$Y^{[n]} = \begin{bmatrix} {Y_1}^{[n]} \\ \vdots \\ {Y_s}^{[n]} \end{bmatrix}, \qquad F(Y^{[n]}) = \begin{bmatrix} f({Y_1}^{[n]}) \\ \vdots \\ f({Y_s}^{[n]}) \end{bmatrix}, \qquad y^{[n]} = \begin{bmatrix} {y_1}^{[n]} \\ \vdots \\ {y_s}^{[n]} \end{bmatrix},$$

and writing the GLM (1.11) in the form

$$\left[\begin{array}{c} Y^{[n]} \\ \hline y^{[n]} \end{array}\right] = \left[\begin{array}{c|c} \mathbf{A}\otimes\mathbf{I} & \mathbf{U}\otimes\mathbf{I} \\ \hline \mathbf{B}\otimes\mathbf{I} & \mathbf{V}\otimes\mathbf{I} \end{array}\right] \left[\begin{array}{c} hF(Y^{[n]}) \\ \hline y^{[n-1]} \end{array}\right],$$

where $\mathbf{I}$ here denotes the indentity matrix of dimension N, and $\otimes$ is the Kronecker (or tensor) product of two matrices, which for $\mathbf{Z} = [z_{ij}] \in \mathbb{R}^{\zeta_1\times\zeta_2}$, $\mathbf{W} \in \mathbb{R}^{\omega_1\times\omega_2}$ (where $\zeta_1, \zeta_2, \omega_1, \omega_2 \in \mathbb{N}$) is defined as the block matrix

$$\mathbf{Z}\otimes\mathbf{W} = \begin{bmatrix} z_{11}\mathbf{W} & \dots & z_{1\zeta_2}\mathbf{W} \\ \vdots & \ddots & \vdots \\ z_{\zeta_1 1}\mathbf{W} & \dots & z_{\zeta_1\zeta_2}\mathbf{W} \end{bmatrix} \in \mathbb{R}^{\zeta_1\omega_1\times\zeta_2\omega_2}.$$

Example. We understand GLM as a generalization of LMM and RK methods, so we expect that those are included as particular cases. As the notation suggests, we want LMM with k steps to be 1-stage k-step GLM, and RK methods with s stages to be s-stage 1-step GLM.

1. The most straightforward expression of a GLM is that of RK methods (1.9), as we just need to use the notation $Y_i^{[n]}$ for the stages and ${y_1}^{[n]}$ for the step to obtain

$$\begin{aligned} Y_i^{[n]} &= h\sum_{j=1}^{s} a_{ij} f(Y_j^{[n]}) + {y_1}^{[n-1]}, \qquad i = 1,\dots,s, \\ {y_1}^{[n]} &= h\sum_{j=1}^{s} b_{1j} f(Y_j^{[n]}) + {y_1}^{[n-1]}, \end{aligned}$$

   where, being $r = 1$, the matrices $\mathbf{B}, \mathbf{U}$ are vectors and $\mathbf{V}$ a scalar, so it is more intuitive to change the notation to give them a more fitting name. In this case, it will be

$$\mathbf{B} =: \mathbf{b}^T = [b_j]^T \in \mathbb{R}^{1\times s}, \qquad \mathbf{U} =: \mathbf{e} = [1,\dots,1] \in \mathbb{R}^{s\times 1}, \qquad \mathbf{V} =: 1 \in \mathbb{R}^{1\times 1}.$$

   and so the partitioned matrix that represents RK methods as GLM is

$$\left[\begin{array}{c|c} \mathbf{A} & \mathbf{e} \\ \hline \mathbf{b}^T & 1 \end{array}\right] = \left[\begin{array}{ccc|c} a_{11} & \dots & a_{1s} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ a_{s1} & \dots & a_{ss} & 1 \\ \hline b_1 & \dots & b_s & 1 \end{array}\right].$$

2. For LMM, the representation is not as immediate. Let us first try to construct it in the most natural way and then solve the issues that appear with that representation. To simplify the notation, we will use the representation of LMM given in (1.5) when studying them as GLM. As it is clear that the only stage of a LMM is that where we compute y_n, we can just define ${Y_1}^{[n]} = y_n$, and from (1.5) it is natural to define the steps $y^{[n]}$ as

$$y^{[n]} = \begin{bmatrix} y_n \\ y_{n-1} \\ \vdots \\ y_{n-k+1} \\ hf(y_n) \\ hf(y_{n-1}) \\ \vdots \\ hf(y_{n-k+1}) \end{bmatrix},$$

and so the LMM (1.5) can be written as

$$\begin{aligned} {Y_1}^{[n]} &= h\beta_0 f({Y_1}^{[n]}) + \sum_{j=1}^{k} \alpha_j {y_j}^{[n-1]} + \sum_{j=1}^{k} \beta_j {y_{k+j}}^{[n-1]} \ , \\ {y_1}^{[n]} &= {Y_1}^{[n]} \ , \\ {y_k}^{[n]} &= hf({Y_1}^{[n]}) \ , \\ {y_i}^{[n]} &= {y_{i-1}}^{[n-1]} \ , \qquad i = 2, \dots, k, k+2, \dots, 2k \ , \end{aligned}$$

where the first relation is what can be deduced from (1.5), where we shall observe that the term $h\beta_0 f({Y_1}^{[n]})$ is just $h\beta_0 y_n$ that has been separated from the other terms in the sum to correctly define the method as GLM; the second is due to the nature of the only stage of the method (commented above), the third is an implication of the second, and the last relation is immediate from the definition of $y^{[n]}$, so the LMM (1.5) has the following representation as a GLM

$$\left[\begin{array}{c|c} \mathbf{A} & \mathbf{U} \\ \hline \mathbf{B} & \mathbf{V} \end{array}\right] = \left[\begin{array}{c|ccccccc} \beta_0 & \alpha_1 & \dots & \alpha_{k-1} & \alpha_k & \beta_1 & \dots & \beta_{k-1} & \beta_k \\ \hline \beta_0 & \alpha_1 & \dots & \alpha_{k-1} & \alpha_k & \beta_1 & \dots & \beta_{k-1} & \beta_k \\ 0 & 1 & & & 0 & 0 & & 0 & 0 \\ \vdots & & \ddots & & \vdots & & \iddots & & \vdots \\ 0 & 0 & & 1 & 0 & 0 & & 0 & 0 \\ 1 & 0 & & 0 & 0 & 0 & & 0 & 0 \\ 0 & 0 & & 0 & 0 & 1 & & 0 & 0 \\ \vdots & & \iddots & & \vdots & & \ddots & & \vdots \\ 0 & 0 & & 0 & 0 & 0 & & 1 & 0 \end{array}\right].$$

Even though it is not necessarily an issue, we observe that with this representation we have accomplished $s = 1$ but we have $r = 2k$. The natural question now is whether we can find another representation for the LMM (1.5) as GLM with $s = 1$ and $r = k$. The answer is yes, but for that we must renounce to the intuitive definition for the steps $y^{[n]}$

given above and instead use the following

$$y_i^{[n-1]} = \sum_{j=k-i+1}^{k} \left(\alpha_j y_{n+k-i-j} + h\beta_j f(y_{n+k-i-j})\right) , \qquad i = 1, \dots, k \ .$$

Defining the steps this way, we can simply write (1.5) as

$$\begin{aligned} Y_1^{[n]} &= h\beta_0 f(y_n) + \sum_{j=1}^{k} \left(\alpha_j y_{n-j} + h\beta_j f(y_{n-j})\right) \\ &= h\beta_0 f(Y_1^{[n]}) + y_k^{[n-1]} \end{aligned}$$

where we have that for each $i = 1, \dots, k$, we can write

$$\begin{aligned} y_i^{[n]} &= \sum_{j=k-i+1}^{k} \left(\alpha_j y_{n+1+k-i-j} + h\beta_j f(y_{n+1+k-i-j})\right) = \\ &= \alpha_{k-i+1} y_n + h\beta_{k-i+1} f(y_n) + \sum_{j=k-i+2}^{k} \left(\alpha_j y_{n+1+k-i-j} + h\beta_j f(y_{n+1+k-i-j})\right) = \\ &= (\alpha_{k-i+1}\beta_0 + \beta_{k-i+1}) h f(Y_1^{[n]}) + \alpha_{k-i+1} y_k^{[n-1]} + y_{i-1}^{[n-1]} \ , \end{aligned}$$

and thus we obtain the desired alternative representation

$$\left[\begin{array}{c|c} \mathbf{A} & \mathbf{U} \\ \hline \mathbf{B} & \mathbf{V} \end{array}\right] = \left[\begin{array}{c|cccccc} \beta_0 & 0 & 0 & 0 & \dots & 0 & 1 \\ \hline \alpha_k\beta_0 + \beta_k & 0 & 0 & 0 & \dots & 0 & \alpha_k \\ \alpha_{k-1}\beta_0 + \beta_{k-1} & 1 & 0 & 0 & \dots & 0 & \alpha_{k-1} \\ \alpha_{k-2}\beta_0 + \beta_{k-2} & 0 & 1 & 0 & \dots & 0 & \alpha_{k-2} \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ \alpha_2\beta_0 + \beta_2 & 0 & 0 & 0 & \ddots & 0 & \alpha_2 \\ \alpha_1\beta_0 + \beta_1 & 0 & 0 & 0 & \dots & 1 & \alpha_1 \end{array}\right] .$$

1.4.1 Convergence, stability and consistency

Let us observe that in the formulation of GLM there is not necessarily a 'natural' interpretation of what the value $y^{[n]}$ represents. We have seen this in the formulation of LMM as GLM, where in the case $r = 2k$, $y^{[n]}$ represents the last k steps and their image by f; and in the case $r = k$ the definition of $y^{[n]}$ was even less intuitive. As mentioned in [But06], this is also related to the fact that, similarly to LMM, it is necessary to known some approximation of the first steps in order to be able to apply GLM. For that purpose, we introduce the concept of 'starting procedures' to obtain an initial vector $y^{[0]}$ from the initial condition y_0. In order to study the convergence of GLM, let us simply assume that such an starting procedure exists

$$\mathcal{S}_h : \mathbb{R}^N \to \mathbb{R}^{r \cdot N} \tag{1.12}$$

which associates with every step size h a starting vector $y^{[0]}$ such that

$$\lim_{h \to 0} S_h(y_0) = \lim_{h \to 0} y^{[0]} = (\mathbf{q}_0 \otimes \mathbf{I}) y(x_0)$$

where $\mathbf{q}_0 \in \mathbb{R}^r$ is a nonzero vector.

Definition 1.18. A GLM is convergent if for any IVP (1.1) satisfying the Lipschitz condition, there is a nonzero vector $\mathbf{q}_0 \in \mathbb{R}^r$ and a starting procedure $\mathcal{S}_h$ satisfying (1.12), such that the sequence of vectors $y^{[n]}$ computed using n steps of the GLM (1.11) with $y^{[0]} = \mathcal{S}_h(y_0)$ and $h = (\overline{x} - x_0)/n$ converges to $\mathbf{q}_0 y(\overline{x})$ for any $\overline{x} \in [x_0, X]$, where y is a solution to (1.1).

Let us recall that the convergence of LMM was characterized by the consistency and stability of the method, so the idea now is to define those properties for GLM and prove the equivalent version of Theorem 1.3. For that, we proceed analogously and say that a GLM (1.11) is preconsistent if it solves exactly the IVP

$$y' = 0 \ , \qquad y(x_0) = 1 \ ,$$

at the beginning and end of each step. For this problem, the GLM (1.11) has the form

$$\begin{aligned} Y^{[n]} &= \mathbf{U} y^{[n-1]} \ , \\ y^{[n]} &= \mathbf{V} y^{[n-1]} \ , \end{aligned}$$

and so taking into account the initial condition of the IVP, this motivates the next definition.

Definition 1.19. A GLM (1.11) is preconsistent if there is a vector $\mathbf{q}_0 \in \mathbb{R}^r$ such that

$$\mathbf{U}\mathbf{q}_0 = \mathbf{e} \ , \qquad \mathbf{V}\mathbf{q}_0 = \mathbf{q}_0 \ ,$$

where $\mathbf{e} = [1, \dots, 1]^T \in \mathbb{R}^s$. In that case, $\mathbf{q}_0$ is called preconsistency vector.

Analogously, we say that it is consistent if it is preconsistent and solves exactly the IVP

$$y' = 1 \ , \qquad y(x_0) = 0 \ ,$$

at the beginning and end of each step. For this problem the GLM (1.11) is

$$\begin{aligned} Y^{[n]} &= h\mathbf{A}\mathbf{e} + \mathbf{U} y^{[n-1]} \ , \\ y^{[n]} &= h\mathbf{B}\mathbf{e} + \mathbf{V} y^{[n-1]} \ , \end{aligned}$$

so this motivates the following definitions, where we distinguish between the consistency of the solution $y^{[n]}$ and that of the stage $Y^{[n]}$, as we did for RK methods.

Definition 1.20. A GLM (1.11) is consistent if it preconsistent with preconsistency vector $\mathbf{q}_0$ and there is a vector $\mathbf{q}_1 \in \mathbb{R}^r$ such that

$$\mathbf{B}\mathbf{e} + \mathbf{V}\mathbf{q}_1 = \mathbf{q}_0 + \mathbf{q}_1 \ .$$

In that case, $\mathbf{q}_1$ is called consistency vector.

Definition 1.21. A GLM (1.11) is stage-consistent if

$$\mathbf{A}\mathbf{e} + \mathbf{U}\mathbf{q}_1 = \mathbf{c} \ ,$$

Example. Let us compare these definitions with those given for LMM and RK methods.

1. For RK as GLM, observe that for the method to be preconsistent, consistent and stage-consistent, it must satisfy respectively

$$\mathbf{e}\mathbf{q}_0 = \mathbf{e} \ , \quad 1\mathbf{q}_0 = \mathbf{q}_0 \ , \qquad \mathbf{b}^T\mathbf{e} + 1\mathbf{q}_1 = \mathbf{q}_0 + \mathbf{q}_1 \qquad \mathbf{A}\mathbf{e} + \mathbf{e}\mathbf{q}_1 = \mathbf{c} \ ,$$

   where $r = 1$, so $\mathbf{q}_0, \mathbf{q}_1 \in \mathbb{R}$. That being the case, we choose $\mathbf{q}_0 = 1, \mathbf{q}_1 = 0$ and removing the redundant conditions, we are left with

$$\mathbf{b}^T\mathbf{e} = 1 \ , \qquad \mathbf{A}\mathbf{e} = \mathbf{c} \ .$$

2. For the case of LMM as GLM with $r = 2k$, we can choose

$$\mathbf{q}_0 = \left[\begin{array}{ccc|ccc} 1 & \overset{k}{\ldots} & 1 & 0 & \overset{k}{\ldots} & 0 \end{array}\right]^T \in \mathbb{R}^{2k} ,$$
$$\mathbf{q}_1 = \left[\begin{array}{ccc|ccc} 0 & -1 & \overset{k}{\ldots} & -(k-1) & \overset{k}{\ldots} & 1 \end{array}\right]^T \in \mathbb{R}^{2k} ,$$

and for the case with $r = k$,

$$\mathbf{q}_0 = \begin{bmatrix} \alpha_k \\ \alpha_{k-1} + \alpha_k \\ \vdots \\ \sum_{j=1}^{k} \alpha_j \end{bmatrix} \in \mathbb{R}^k , \qquad \mathbf{q}_1 = \begin{bmatrix} \beta_k \\ \beta_{k-1} + \beta_k - \alpha_k \\ \vdots \\ \sum_{j=1}^{k} \beta_j - \sum_{j=1}^{k} (j-1)\alpha_j \end{bmatrix} \in \mathbb{R}^k ,$$

and the conditions for the method to be preconsistent, consistent and stage-consistent become (notice that as $s = 1$, we have $\mathbf{e} = 1$)

$$\sum_{j=1}^{k} \alpha_j = 1 , \quad \sum_{j=1}^{k} \alpha_j = 1 , \quad \sum_{j=0}^{k} \beta_j - \sum_{j=2}^{k} (j-1)\alpha_j = 1 , \quad \sum_{j=0}^{k} \beta_j - \sum_{j=2}^{k} (j-1)\alpha_j = \mathbf{c} .$$

so combining them and removing the redundant ones, we obtain (recall we follow notation (1.5))

$$\sum_{j=1}^{k} \alpha_j = 1 , \qquad \sum_{j=1}^{k} j\alpha_j = \sum_{j=0}^{k} \beta_j , \qquad \mathbf{c} = 1 .$$

When considering the stability of a GLM (1.11), we will proceed as we did for LMM and study when the numerical solution $y^{[n]}$ of the GLM (1.11) is bounded as $n \to \infty$ for the IVP

$$y' = 0 , \qquad y(x_0) = 0 .$$

Observe that for this problem we have that the GLM (1.11) is

$$\begin{aligned} Y^{[n]} &= \mathbf{U} y^{[n-1]} \\ y^{[n]} &= \mathbf{V} y^{[n-1]} \end{aligned} \implies y^{[n]} = \mathbf{V}^n y^{[0]} ,$$

and so this motivates the next definition.

Definition 1.22. A GLM is zero-stable if there is a constant C such that for all $n \geq 0$,

$$\|\mathbf{V}^n\| \leq C.$$

In fact, we could have defined zero-stability as the following equivalent condition, but for convenience it is customary to state it as a theorem.

Theorem 1.8. A GLM is zero-stable if the minimal polynomial $\rho_{\mathbf{V}}$ of the coefficient matrix $\mathbf{V}$ has no zeros with magnitude greater than 1 and all zeros with magnitude equal to 1 are simple.

Example. By applying this definition of zero-stability to LMM and RK methods, we observe the following.

1. The minimal polynomial $\rho_{\mathbf{V}}$ of the matrix $\mathbf{V} = 1$ of the RK method as GLM is

$$\rho_{\mathbf{V}}(w) = w - 1 ,$$

so we can see that all RK methods as GLM are zero-stable.

2. In the case of LMM, the minimal polynomial of the matrix $\mathbf{V}$ of the LMM as GLM with $r = 2k$ and $r = k$ is of the form (see [Jac09])

$$\rho_{\mathbf{V}}(w) = w^k \rho(w) \qquad \text{or} \qquad \rho_{\mathbf{V}} = \rho(w) \ ,$$

where $\rho(w)$ is the first generating polynomial of the LMM (1.5), so we deduce that the zero-stability of LMM as GLM is equivalent to the root condition introduced for LMM.

As for LMM, we conclude this section by stating the theorem that characterizes the convergence of GLM in terms of zero-stability and consistency.

Theorem 1.9. A GLM (1.11) is convergent if and only if it is zero-stable and consistent.

1.4.2 Local error and order

For defining the convergence of a GLM we used the concept of starting procedures. Recalling their definition, as discussed in [But06], we can understand them as some sort of particular GLM with one 'input' and several 'outputs' that allows to approximate the initial vector $y^{[0]}$ from y_0. It is clear then that using such a 'loose' definition could affect how the order of a GLM is defined. For example, for a RK method as a GLM we could construct a starting procedure such that the resulting RK method as GLM has $s = p = 5$, and that is something that we do not expect for a RK method, as it contradicts the Butcher barriers. Also, when studying the order of a LMM we do not want it to be affected by the order of the approximations of the initial steps. Taking all this into account, what we will do is define the order of GLM relative to the starting procedure.

Let $\mathcal{E}_h$ denote the exact solution after a time step h and $\mathcal{M}$ the main GLM. The idea then is to study the difference of $\mathcal{M} \circ \mathcal{S}_h$ and $\mathcal{S}_h \circ \mathcal{E}$ (see Figure 1.4), motivating the following definition.

Definition 1.23. A GLM (1.11) has order p relative to a starting procedure $\mathcal{S}_h$ if

$$\mathcal{M} \circ \mathcal{S}_h - \mathcal{S}_h \circ \mathcal{E}_h = \mathcal{O}(h^{p+1}) \ .$$

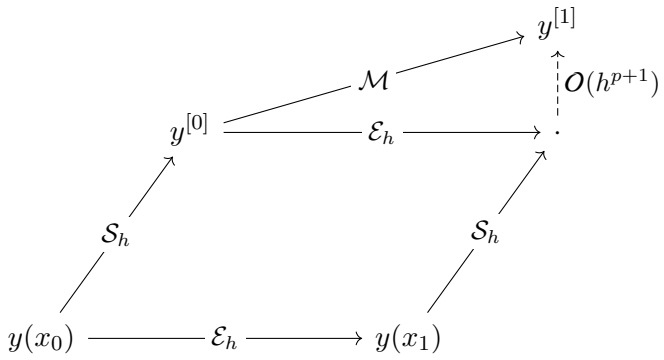

Figure 1.4: Visual representation of the definition of order of a GLM

As mentioned before, the order conditions for LMM can be easily derived, but those of RK where avoided because of their complexity. Being GLM a generalization of both, we expect that their order conditions are hard to derive, and in fact that will be the case.

For studying the order conditions of GLM, we first need to explicitly characterize the order of the values that appear in the formulation (1.11). For that, in order for p to be the order of the method and q the order of the stages, we must require that the $y^{[n-1]}$ and $y^{[n]}$ satisfy

$$y_i^{[n-1]} = \sum_{k=0}^{p} q_{ik} h^k y^{(k)}(t_{n-1}) + \mathcal{O}(h^{p+1}) \ , \qquad i = 1, \dots, r \tag{1.13}$$

$$y_i^{[n]} = \sum_{k=0}^{p} q_{ik} h^k y^{(k)}(t_n) + \mathcal{O}(h^{p+1}) \ , \qquad i = 1, 2, \dots, r \tag{1.14}$$

for some parameters $q_{ik} \in \mathbb{R}$ for $i = 1, \dots, r,\ k = 0, 1, \dots, p$. Also, for the stages $Y^{[n]}$ to be of order q, they must satisfy

$$Y_i^{[n]} = y(t_{n-1} + c_i h) + \mathcal{O}(h^{q+1}) \ , \qquad i = 1, \dots, s \tag{1.15}$$

Deriving the conditions of GLM for general values of order p and stage order q "is quite complicated requires sophisticated algebraic tools developed by Butcher" as mentioned in [Jac09]. Nevertheless, for methods with high stage order; that is, $q = p$ or $q = p - 1$ it is simpler and can be done using complex analysis. For developing those cases, it is necessary to define

$$\mathbf{q}_k = \begin{bmatrix} q_{1k} & q_{2k} & \dots & q_{rk} \end{bmatrix}^T \ , \qquad k = 0, 1, \dots, p \ ,$$

(notice that $\mathbf{q}_0$ and $\mathbf{q}_1$ are the preconsistency and consistency vectors), and also

$$e^{\mathbf{c}z} = \begin{bmatrix} e^{c_1 z} & e^{c_2 z} & \dots & e^{c_s z} \end{bmatrix}^T \ , \qquad \mathbf{w}(z) = \sum_{k=0}^{p} \mathbf{q}_k z^k \ .$$

Theorem 1.10. Assume that $y^{[n-1]}$ satisfies (1.13).

i) The GLM (1.11) of order p and stage order $q = p$ satisfies (1.15) and (1.14) iff

$$\begin{aligned} e^{\mathbf{c}z} &= z\mathbf{A}e^{\mathbf{c}z} + \mathbf{U}\mathbf{w}(z) + \mathcal{O}(z^{p+1}) \ , \\ e^{z}\mathbf{w}(z) &= z\mathbf{B}e^{\mathbf{c}z} + \mathbf{V}\mathbf{w}(z) + \mathcal{O}(z^{p+1}) \ . \end{aligned}$$

ii) The GLM (1.11) of order p and stage order $q = p - 1$ satisfies (1.15) and (1.14) iff

$$\begin{aligned} e^{\mathbf{c}z} &= z\mathbf{A}e^{\mathbf{c}z} + \mathbf{U}\mathbf{w}(z) + \left(\frac{\mathbf{c}^p}{p!} - \frac{\mathbf{A}\mathbf{c}^{p-1}}{(p-1)!} - \mathbf{U}\mathbf{q}_p \right) z^p + \mathcal{O}(z^{p+1}) \ , \\ e^{z}\mathbf{w}(z) &= z\mathbf{B}e^{\mathbf{c}z} + \mathbf{V}\mathbf{w}(z) + \mathcal{O}(z^{p+1}) \ . \end{aligned}$$

1.4.3 Linear stability

As for LMM and RK methods, the analysis of linear stability is based on the boundedness of the numerical solution $y^{[n]}$ when applied to the IVP

$$y' = \lambda y \ , \quad \lambda \in \mathbb{C} \ .$$

We want to study the influence of a single step of the method, so for that we want to find an $r \times r$ matrix such that

$$y^{[n]} = \mathbf{M} y^{[n-1]} \ .$$

Let us observe that the GLM (1.11) for this particular problem has the form

$$\begin{aligned} Y^{[n]} &= h\lambda \mathbf{A} Y^{[n]} + \mathbf{U} y^{[n-1]} \ , \\ y^{[n]} &= h\lambda \mathbf{B} Y^{[n]} + \mathbf{V} y^{[n-1]} \ , \end{aligned}$$

so if we define $z := h\lambda$, from the first equation we get

$$Y^{[n]} = (\mathbf{I} - z\mathbf{A})^{-1} \mathbf{U} y^{[n-1]} \ ,$$

and substituting it into the second equation, we obtain

$$y^{[n]} = \Big(\mathbf{V} + z\mathbf{B}(\mathbf{I} - z\mathbf{A})^{-1}\mathbf{U} \Big) y^{[n-1]} \ ,$$

thus motivating the following definition.

Definition 1.24. The stability matrix of a GLM (1.11) is

$$\mathbf{M}(z) = \mathbf{V} + z\mathbf{B}(\mathbf{I} - z\mathbf{A})^{-1}\mathbf{U} \ .$$

To study the boundedness of $y^{[n]}$ as $n \to \infty$ we consider the characteristic polynomial of the matrix $\mathbf{M}(z)$. It is interesting to observe that $\mathbf{M}(0) = \mathbf{V}$, so we can in fact understand zero-stability as a particular case of this study.

Definition 1.25. The stability function of a GLM (1.11) is

$$\Phi(w, z) = \det(w\mathbf{I} - \mathbf{M}(z)) \ .$$

Definition 1.26. A GLM (1.11) is said to be absolutely stable for a given $z \in \mathbb{C}$ if all roots $w_i = w_i(z)$ for $i = 1, 2, \ldots, r$ of the stability function $\Phi(w, z)$ are inside the unit circle.

Definition 1.27. The region $\mathcal{A}$ of absolute stability of a GLM (1.11) is the set of all $z \in \mathbb{C}$ such that the method is absolutely stable; that is

$$\mathcal{A} = \Big\{ z \in \mathbb{C} \ \ ; \ \ |w_i(z)| < 1 \ , \ \ i = 1, 2, \ldots, r \Big\}.$$

Definition 1.28. A GLM (1.11) is A-stable if its region of absolute stability $\mathcal{A}$ contains the negative complex half-plane $\mathbb{C}^-$; that is,

$$\Big\{ z \in \mathbb{C} \ \ ; \ \ \mathrm{Re}(z) < 0 \Big\} \subset \mathcal{A}.$$

Definition 1.29. A GLM (1.11) is L-stable if its A-stable and

$$\lim_{z \to \infty} \rho(\mathbf{M}(z)) = 0 \ ,$$

where $\rho(\mathbf{M}(z))$ is the spectral radius of the matrix $\mathbf{M}(z)$.

Example.

1. In the case of RK methods as GLM, being $r = 1$, the stability matrix $\mathbf{M}(z)$ is the function

$$\mathbf{M}(z) = 1 + z\mathbf{b}^T(\mathbf{I} - z\mathbf{A})^{-1}\mathbf{e} \ ,$$

which coincides with the function $R(z)$. As we can understand $\mathbf{M}(z)$ as 1×1 matrix, we have that $\Phi(w, z) = w - \mathbf{M}(z)$, with its only root being $\mathbf{M}(z)$, so we recover the definitions of the region of stability and A-stability of a RK method. In fact, as a 1×1 matrix, we could consider $R(z)$ to be the only eigenvalue of $\mathbf{M}(z)$, and thus we also recover the definition of L-stability.

2. For LMM as GLM, for $r = 2k$ it is proven in [Jac09] that the stability matrix is

$$\mathbf{M}(z) = \begin{bmatrix} \frac{\alpha_1}{1 - z\beta_0} & \cdots & \frac{\alpha_{k-1}}{1 - z\beta_0} & \frac{\alpha_k}{1 - z\beta_0} & \frac{\beta_1}{1 - z\beta_0} & \cdots & \frac{\beta_{k-1}}{1 - z\beta_0} & \frac{\beta_k}{1 - z\beta_0} \\ 1 & \ldots & 0 & 0 & 0 & \ldots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \ldots & 1 & 0 & 0 & \ldots & 0 & 0 \\ \frac{\alpha_1 z}{1 - z\beta_0} & \cdots & \frac{\alpha_{k-1} z}{1 - z\beta_0} & \frac{\alpha_k z}{1 - z\beta_0} & \frac{\beta_1 z}{1 - z\beta_0} & \cdots & \frac{\beta_{k-1} z}{1 - z\beta_0} & \frac{\beta_k z}{1 - z\beta_0} \\ 0 & \ldots & 0 & 0 & 1 & \ldots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \ldots & 0 & 0 & 0 & \ldots & 1 & 0 \end{bmatrix}$$

and the stability function

$$\Phi(w,z) = \frac{1}{\beta_0 z} w^k (\rho(w) - z\sigma(w)) \ ,$$

while for $r = k$ the stability matrix takes the from

$$\mathbf{M}(z) = \begin{bmatrix} 0 & \dots & 0 & \alpha_k + \dfrac{z}{1-\beta_0}(\alpha_k + \beta_0 + \beta_k) \\ 1 & \dots & 0 & \alpha_{k-1} + \dfrac{z}{1-\beta_0}(\alpha_{k-1} + \beta_0 + \beta_{k-1}) \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 1 & \alpha_1 + \dfrac{z}{1-\beta_0}(\alpha_1 + \beta_0 + \beta_1) \end{bmatrix}$$

and the stability function

$$\Phi(w,z) = \frac{1}{\beta_0 z} (\rho(w) - z\sigma(w)) \ ,$$

so we observe that the study of stability using the roots of such polynomials is in accordance with the theory introduced before for LMM.

Chapter 2

Jet transport

When studying dynamical systems given by the flow of an ODE there are several scenarios in which the ability to compute the variational equations is of great utility. Some of these, detailed in [Sim90], are the computation of periodic orbits, their stability, and their dependence with respect to parameters. In particular, the numerical integration of the variational equations can be used to compute the derivatives of the flow, and as we will see in Chapter 3, this is of great use to obtain the derivatives of the Poincaré map, which would otherwise be challenging because they usually lack of a closed expression.

For the numerical integration of the variational equations, we will follow [Gim+23] to present the technique called jet transport, based on the manipulation of formal power series. Furthermore, we will present our main contributions: Theorems 2.2 and 2.6. At the first one, we prove that the integration of an initial value problem using jet transport with General Linear methods is equivalent to the integration of its variational equations using the same method. In the second, we find the exact relations that must be satisfied by the high-order coefficients of the jets for them to be solutions of an implicit system. The results obtained in both theorems will therefore allow the effective implementation of jet transport for General Linear methods.

2.1 Framework

Let us recall the initial value problem (1.1) considered in Chapter 1; i.e.,

$$y'(x) = f(x, y(x)) \ , \qquad y(x_0) = y_0 \tag{2.1}$$

where $f : \mathbb{R} \times \mathbb{R}^N \to \mathbb{R}^N$. Let us denote by $y(x; x_0, y_0)$ the solution of the IVP (2.1) and suppose that the derivative of f with respect to y exists and is continuous. Then, $y(x; x_0, y_0)$ is differentiable with respect to y_0 with derivative

$$V(x) := D_{y_0} y(x; x_0, y_0) \ ,$$

and V satisfies the linear differential equation

$$V'(x) = D_y f(x, y(x)) V(x) \ , \qquad V(x_0) = \mathbf{I} \ ,$$

where $\mathbf{I}$ denotes the identity matrix of dimension N. It is customary to write this equation together with the IVP, obtaining the so-called first order Variational Equations (VE)

$$\begin{aligned} y'(x) &= f(x, y(x)) \ , & y(x_0) &= y_0 \\ V'(x) &= D_y f(x, y(x)) V(x) \ , & V(x_0) &= \mathbf{I} \ . \end{aligned} \tag{2.2}$$

We observe that there is no dependence between the columns of the matrix V, so in general, instead of working with the VE in the form (2.2), we will consider the equivalent formulation

$$\begin{aligned} y'(x) &= f(x, y(x)) \ , & y(x_0) &= y_0 \\ v'(x) &= D_y f(x, y(x)) v(x) \ , & v(x_0) &= v_0 \ . \end{aligned} \tag{2.3}$$

where v represents any column of the matrix V, so by setting the initial value v_0 to be the appropriate column of $\mathbf{I}$ we can recover the form (2.2). Moreover, if f is sufficiently differentiable, as detailed in [Sim90], by successive differentiation of the IVP we obtain the higher order variational equations.

Let us remark that the methods studied in Chapter 1 can be applied to the study of VE, as these can be written as an IVP by simply considering

$$u(x) = \begin{bmatrix} y(x) \\ v(x) \end{bmatrix}, \quad g(x, u(x)) = \begin{bmatrix} f(x, y(x)) \\ D_y f(x, y(x)) v(x) \end{bmatrix}, \quad u_0 = \begin{bmatrix} y_0 \\ v_0 \end{bmatrix}.$$

2.2 Automatic differentiation

Automatic differentiation is a technique to compute high-order derivatives of the output of an algorithm. It is customary to introduce this concept through the manipulation of formal power series. For that, we will follow [Gim19].

2.2.1 Formal power series in one variable

We define a formal power series in one variable s as

$$\sum_{k\geq 0} a_k s^k ,$$

where the coefficients a_k belong to a field. If we have a function $f \in C^\infty$ defined in a neighborhood of 0, we can consider the following power series

$$\sum_{k\geq 0} f^{[k]} s^k ,$$

where the coefficients are this time the k-th normalized derivative of f at 0, defined as

$$f^{[k]} := \frac{1}{k!} f^{(k)}(0) .$$

As formal power series are characterized by their coefficients, we can express their arithmetic in terms of the coefficients of the resulting formal power series.

Lemma 2.1. Let us consider the following formal power series

$$A := \sum_{k\geq 0} a_k s^k , \qquad B := \sum_{k\geq 0} b_k s^k , \qquad C := \sum_{k\geq 0} c_k s^k .$$

i) For the addition or subtraction $A \pm B = C$, the coefficients c_k are given by

$$c_k = a_k \pm b_k .$$

ii) For the multiplication $A \cdot B = C$, the coefficients c_k are given by

$$c_k = \sum_{j=0}^{k} a_j b_{k-j} .$$

iii) For the division $A/B = C$, if $b_0 \neq 0$, the coefficients c_k are given by

$$\begin{aligned} c_0 &= a_0/b_0 \\ c_k &= \frac{1}{b_0}\Big(a_k - \sum_{j=1}^{k} b_j c_{k-j}\Big) , \qquad k \geq 1 . \end{aligned}$$

iv) For the power $A^p = C$, if $p \in \mathbb{R}$, $p \neq 0, 1$ and $a_0 \neq 0$, the coefficients c_k are given by

$$\begin{aligned} c_0 &= a_0^p \\ c_k &= \frac{1}{ka_0}\sum_{j=0}^{k-1}(pk-(p+1)j)a_{k-j}c_j \ , \qquad k \geq 1 \ . \end{aligned}$$

v) Let h be a derivable function such that there are some mappings α, β, γ with

$$\alpha(A)\cdot h'(A) - \beta(A)\cdot h(A) = \gamma(A) \ .$$

If we denote by $\alpha_k, \beta_k, \gamma_k$ the coefficients of $\alpha(A), \beta(A), \gamma(A)$ respectively and if $a_0 \neq 0$, the coefficients c_k are given by

$$\begin{aligned} c_0 &= h(a_0) \\ c_k &= \frac{1}{k\alpha_0}\Big(\sum_{j=1}^{k}\Big(\gamma_{k-j} + \sum_{i=0}^{k-j}\beta_i c_{k-i-j}\Big)ja_j - \sum_{j=1}^{k-1} j\alpha_{k-j}c_j\Big) \ , \qquad k \geq 1 \ . \end{aligned}$$

Let us observe a few things in the last lemma. First, in cases iii), iv) and v), the coefficient c_k depends on other coefficients c_j for $j < k$, so they need to be computed recursively. Also, observe that case iv) includes interesting particular cases like the inversion for $p = -1$ and the square root for $p = 1/2$. Finally, even though not all functions fall under the scope of case v), as shown in [Gim19], the formula given can be used to obtain $h(A)$ for functions like the logarithm, exponential, sine, cosine and so on, so it indeed covers a broad family of functions. As an example, for $h(A) = \log(A)$ it is enough to choose $\alpha(A) = A$, $\beta(A) = 0$ and $\gamma(A) = 1$.

2.2.2 Formal power series in several variables

In an analogous way, we can define formal power series of K variables $s = (s_1, \ldots, s_K)$

$$\sum_{m\geq 0}\sum_{|k|=m} a_k s^k \ ,$$

where $k = (k_1, \ldots, k_K) \in \mathbb{N}^K$ is a multi-index and we define $|k| = k_1 + \cdots + k_K$ and $s^k = s_1^{k_1}\cdots s_K^{k_K}$. As before, being f a multivariate C^∞ function defined in a neighborhood of 0, we can consider the following formal power series

$$\sum_{m\geq 0}\sum_{|k|=m} f^{[k]}s^k \ , \quad \text{where } f^{[k]} = \frac{\partial^k f}{k_1!\ldots k_K!}(0) \ .$$

The arithmetic of formal multivariate power series is defined by reducing it to the one of formal power series in one variable, where in fact, we will use the expressions of Lemma 2.1 by proceeding as follows. Let us first consider a new symbolic variable z and replace s_i by $s_i z$ for all $i = 1, \ldots, K$. Observe that the formal multivariate power series above can be rewritten as

$$\sum_{m\geq 0} A_m z^m \ , \quad \text{where } A_m = \sum_{|k|=m} a_k s^k \ ;$$

that is, we have expressed the multivariate power series as a formal power series of only one variable, where the coefficients A_m are homogeneous polynomials of degree m. To obtain the arithmetic of multivariate power series, we then only need to refer to Lemma 2.1, where we just have to notice that the operations between the coefficients will be that of formal power series of one variable. Finally, setting $z = 1$ we recover the original formal multivariate power series.

2.2.3 Jet transport

Let us remark that when implementing the technique of automatic differentiation on a computer, we will limit the power series up to certain order; that is, we will in fact consider truncated power series. From now on, we fix such a maximum order to be M, and let us consider the fact that, when using equalities between truncated power series, we will understand that they coincide just up to order M.

In the literature, the set of derivatives of a function or the set of partial derivatives of a multivariate function, is usually called the jet of derivatives. Let us then notice that the formal series with coefficients $f^{[k]}$ of a function f 'codifies' its jet of derivatives, and in particular, for truncated power series of order M, it codifies them up to order M. We can understand that the power series of $h \circ f$ codifies the jet of derivatives of the composition, so the operations with power series can be seen as the 'transport' of the jet of derivatives, thus naming this chapter.

2.3 Jet transport for General Linear methods

It is shown in [Gim+23] that using jet transport; that is, using the arithmetic of truncated power series to integrate an IVP (2.1) using a Runge-Kutta method is equivalent to integrating the VE (2.2) using the same Runge-Kutta method. Following the same ideas, we will prove the analogous for General Linear methods.

In order to do so, we must first note that it is enough to prove that the statement holds after a single integration step. We also remark that it is enough to consider the equivalence on the case of first order VEs and jet transport of order 1 for the IVP, as higher order cases can be reduced to order 1 by increasing the dimensionality of the IVP.

Theorem 2.2. Let the external and internal stages of a GLM (1.11) applied to the VE (2.2) be respectively $(y_i^{[n]}, v_i^{[n]})$ and $(Y_i^{[n]}, V_i^{[n]})$, where the first component refers to the coordinates y and the second to v. When using jet transport of order 1 on the GLM (1.11) applied to the IVP (2.1), the external and internal stages obtained are respectively $y_i^{[n]} + v_i^{[n]} s$ and $Y_i^{[n]} + V_i^{[n]} s$.

Proof. First, observe that the GLM (1.11) applied to the VE (2.2) is of the form

$$\begin{aligned}
Y_i^{[n]} &= h \sum_{j=1}^{s} a_{ij} f(Y_j^{[n]}) + \sum_{j=1}^{r} u_{ij} y_j^{[n-1]}, \qquad i = 1, \dots, s, \\
V_i^{[n]} &= h \sum_{j=1}^{s} a_{ij} D_y f(Y_j^{[n]}) V_j^{[n]} + \sum_{j=1}^{r} u_{ij} v_j^{[n-1]}, \qquad i = 1, \dots, s, \\
y_i^{[n]} &= h \sum_{j=1}^{s} b_{ij} f(Y_j^{[n]}) + \sum_{j=1}^{r} v_{ij} y_j^{[n-1]}, \qquad i = 1, \dots, r, \\
v_i^{[n]} &= h \sum_{j=1}^{s} b_{ij} D_y f(Y_j^{[n]}) V_j^{[n]} + \sum_{j=1}^{r} v_{ij} v_j^{[n-1]}, \qquad i = 1, \dots, r,
\end{aligned} \tag{2.4}$$

Now, let us study the result of applying jet transport of order 1 to the GLM (1.11) when applied to the IVP (2.1); that is, to us study the method when using the following truncated power series of order 1 and their arithmetic

$$\begin{aligned}
&\overline{Y}_i^{[n]} + \overline{V}_i^{[n]} s \ , \qquad i = 1, \dots, s \\
&\overline{y}_i^{[n]} + \overline{v}_i^{[n]} s \ , \qquad i = 1, \dots, r \ .
\end{aligned}$$

The goal is then to show that these coefficients indeed coincide with the ones in (2.4). For that,

notice that the GLM (1.11) applied to the IVP (2.1) has the form

$$\overline{Y}_i{}^{[n]} + \overline{V}_i{}^{[n]} s = h \sum_{j=1}^{s} a_{ij} f(\overline{Y}_j{}^{[n]} + \overline{V}_j{}^{[n]} s) + \sum_{j=1}^{r} u_{ij} \Big(\overline{y}_j{}^{[n-1]} + \overline{v}_j{}^{[n-1]} s \Big), \qquad i = 1, \dots, s,$$

$$\overline{y}_i{}^{[n]} + \overline{v}_i{}^{[n]} s = h \sum_{j=1}^{s} b_{ij} f(\overline{Y}_j{}^{[n]} + \overline{V}_j{}^{[n]} s) + \sum_{j=1}^{r} v_{ij} \Big(\overline{y}_j{}^{[n-1]} + \overline{v}_j{}^{[n-1]} s \Big), \qquad i = 1, \dots, r,$$

where we notice that, expanding $f(\overline{Y}_j{}^{[n]} + \overline{V}_j{}^{[n]} s)$ as a truncated power series of order 1 around $\overline{Y}_j{}^{[n]}$, we obtain the relation

$$f(\overline{Y}_j{}^{[n]} + \overline{V}_j{}^{[n]} s) = f(\overline{Y}_j{}^{[n]}) + D_y f(\overline{Y}_j{}^{[n]}) \overline{V}_j{}^{[n]} s \ ,$$

thus the method above becomes

$$\overline{Y}_i{}^{[n]} + \overline{V}_i{}^{[n]} s = h \sum_{j=1}^{s} a_{ij} \Big(f(\overline{Y}_j{}^{[n]}) + D_y f(\overline{Y}_j{}^{[n]}) \overline{V}_j{}^{[n]} s \Big) + \sum_{j=1}^{r} u_{ij} \Big(\overline{y}_j{}^{[n-1]} + \overline{v}_j{}^{[n-1]} s \Big), \quad i = 1, \dots, s,$$

$$\overline{y}_i{}^{[n]} + \overline{v}_i{}^{[n]} s = h \sum_{j=1}^{s} b_{ij} \Big(f(\overline{Y}_j{}^{[n]}) + D_y f(\overline{Y}_j{}^{[n]}) \overline{V}_j{}^{[n]} s \Big) + \sum_{j=1}^{r} v_{ij} \Big(\overline{y}_j{}^{[n-1]} + \overline{v}_j{}^{[n-1]} s \Big), \quad i = 1, \dots, r,$$

and the statement follows from pairing the coefficients by their corresponding power of s and comparing with the expression in (2.4). □

2.3.1 Solving implicit systems

As implicit GLM (and in particular LMM and RK methods) require solving an implicit equation in order to compute the next step of the integration, we will need to study that issue in the context of this section; that is, when integrating using jet transport. For that, we need a technique that allows to solve implicit systems while dealing with the arithmetic of truncated power series.

Before doing so, let us start by introducing some notation to simplify the expressions that we will obtain. First, as it is customary in the context of numerical integration to use subindices to denote steps, we will use the notation of normalized derivatives to denote the coefficients of formal power series. For example, a truncated power series of order M and one symbol s will be denoted

$$a^{[0]} + a^{[1]} s + \cdots + a^{[M]} s^M \ ,$$

while a truncated power series of order 2 and 2 symbols s_1, s_2 will be

$$a^{[0]} + a^{[1,0]} s_1 + a^{[0,1]} s_2 + a^{[2,0]} s_1^2 + a^{[1,1]} s_1 s_2 + a^{[0,2]} s_2^2 \ .$$

In general, we will follow the next naming convention for truncated power series, as it displays all the information needed

$$[a]_K^M := \sum_{m=0}^{M} \sum_{|k|=m} a^{[k]} s^k \ ,$$

where M is the truncation order, K the number of symbols, and we use the multiindex notation $k = (k_1, \dots, k_K)$, $|k| = k_1 + \cdots + k_K$, where $s = (s_1, \dots, s_K)$ and $s^k = s_1^{k_1} \cdots s_K^{k_K}$. As an example, the two truncated power series above are, with this notation,

$$[a]_1^M \ , \qquad [a]_2^2 \ .$$

Also, as the jet of derivatives is codified in the power series, we will usually refer to them as jets.

Implicit Euler method

Let us consider the simplest setting, in which we are willing to integrate an IVP (1.3) using an implicit Euler method (or GLM with one step and one stage), where we have a known value a, an unknown value b and they related by the expression

$$F(a, b) = 0 \ , \tag{2.5}$$

where $F : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}^N$ depends on the method and on the function $f : \mathbb{R}^N \to \mathbb{R}^N$ of the IVP. In that case, the value b can be approximated by using an iterative method, like the Newton method. Nevertheless, in the context of jet transport of one symbol s and M, we have a known jet $[a]_1^M$ and an unknown jet $[b]_1^M$ related by the expression

$$F([a]_1^M, [b]_1^M) = 0 \ . \tag{2.6}$$

Let us study how can we obtain the coefficients that determine the unknown jet $[b]_1^M$. First, observe that truncating the jets at order 0; that is, for $M = 0$ we have that the jets $[a]_1^0 = a^{[0]}$, $[b]_1^0 = b^{[0]}$ are just values, so we are in the context of (2.5) and the coefficient of order 0 can be determined using a Newton method. For higher orders, the coefficients will be obtained iteratively using the next theorem.

Theorem 2.3. In the setting of (2.6), the coefficients of the jet $[b]_1^M$ are determined by

$$\begin{aligned} &D_b F(a^{[0]}, b^{[0]}) b^{[1]} = -D_a F(a^{[0]}, b^{[0]}) a^{[1]} \ , \\ &D_b F(a^{[0]}, b^{[0]}) b^{[m]} = -\Big(F([a]_1^{m-1}, [b]_1^{m-1})^{[m]} + D_a F(a^{[0]}, b^{[0]}) a^{[m]} \Big) \ , \quad m = 2, \dots, M \ , \end{aligned}$$

where $[a]_1^M$ is known and $b^{[0]}$ has been previously computed.

Proof. Let us proceed by induction on the order, computing the coefficient $b^{[m]}$ for each m.

- $m = 1$: To compute $b^{[1]}$, let us consider the jets only up to order 1. Taking the power expansion of $F([a]_1^1, [b]_1^1)$ around $([a]_1^0, [b]_1^0)$, we obtain

  $$F([a]_1^1, [b]_1^1) = F([a]_1^0, [b]_1^0) + D_a F([a]_1^0, [b]_1^0) a^{[1]} s + D_b F([a]_1^0, [b]_1^0) b^{[1]} s + \boldsymbol{O}(s^2)$$

  truncating and recalling $[a]_1^0 = a^{[0]}$, $[b]_1^0 = b^{[0]}$, we get

  $$= F(a^{[0]}, b^{[0]}) + D_a F(a^{[0]}, b^{[0]}) a^{[1]} s + D_b F(a^{[0]}, b^{[0]}) b^{[1]} s \ .$$

  We want this expression to be identically 0 (up to order 1), and by hypothesis the coefficient $b^{[0]}$ of order 0 has been chosen so that $F(a^{[0]}, b^{[0]}) = 0$. Hence, we must impose the coefficients of s to be 0, and thus $b^{[1]}$ must satisfy the relation

  $$D_b F(a^{[0]}, b^{[0]}) b^{[1]} = -D_a F(a^{[0]}, b^{[0]}) a^{[1]} \ .$$

- $m - 1 \to m$: Let us assume that we have computed the coefficients of order $1, \dots, m-1$, so we have determined $[b]_1^{m-1}$. To compute $b^{[m]}$, let us consider the jets only up to order m. Taking the power expansion of $F([a]_1^m, [b]_1^m)$ around $([a]_1^{m-1}, [b]_1^{m-1})$, we obtain

  $$\begin{aligned} F([a]_1^m, [b]_1^m) = F([a]_1^{m-1}, [b]_1^{m-1}) &+ D_a F([a]_1^{m-1}, [b]_1^{m-1}) a^{[m]} s^m + \\ &+ D_b F([a]_1^{m-1}, [b]_1^{m-1}) b^{[m]} s^m + \boldsymbol{O}(s^{m+1}) \end{aligned}$$

where, by induction hypothesis, $[b]_1^{m-1}$ has been chosen so that $F([a]_1^{m-1},[b]_1^{m-1})=0$ (up to order $m-1$). In that case, it has some coefficient of order m which will be named $F([a]_1^{m-1},[b]_1^{m-1})^{[m]}$ following the current notation (notice that higher order terms are not considered, as jets are truncated at order m). Expanding the remaining terms in power series around $(a^{[0]},b^{[0]})$, we get

$$\begin{aligned}= F([a]_1^{m-1},[b]_1^{m-1})^{[m]}s^m + \Big(D_aF(a^{[0]},b^{[0]})+\boldsymbol{O}(s)\Big)a^{[m]}s^m + \\ + \Big(D_bF(a^{[0]},b^{[0]})+\boldsymbol{O}(s)\Big)b^{[m]}s^m+\boldsymbol{O}(s^{m+1})\ ,\end{aligned}$$

and truncating at order m, we obtain the expression

$$\begin{aligned}= F([a]_1^{m-1},[b]_1^{m-1})^{[m]}s^m + D_aF(a^{[0]},b^{[0]})a^{[m]}s^m + \\ + D_bF(a^{[0]},b^{[0]})b^{[m]}s^m\ .\end{aligned}$$

Imposing the coefficients of s^m to be 0, $b^{[m]}$ must satisfy

$$D_bF(a^{[0]},b^{[0]})b^{[m]} = -\Big(F([a]_1^{m-1},[b]_1^{m-1})^{[m]}+D_aF(a^{[0]},b^{[0]})a^{[m]}\Big)\ .$$

□

In order to apply Theorem 2.3, it will be useful to have a way to obtain the derivatives that appear in the expression that determine the coefficients. Luckily, they can be easily obtained using jet transport as seen in the following Lemma.

Lemma 2.4. Let us consider jets of order 1 and the N symbols $s=(s_1,\dots,s_N)$. In the context of (2.6), the derivatives of Theorem 2.3 are given by

$$\begin{aligned}D_aF(a^{[0]},b^{[0]}) &= \left[\ F(a^{[0]}+s,b^{[0]})^{[e_1]}\ \middle|\ \dots\ \middle|\ F(a^{[0]}+s,b^{[0]})^{[e_N]}\ \right]\ ,\\ D_bF(a^{[0]},b^{[0]}) &= \left[\ F(a^{[0]},b^{[0]}+s)^{[e_1]}\ \middle|\ \dots\ \middle|\ F(a^{[0]},b^{[0]}+s)^{[e_N]}\ \right]\ ,\end{aligned}$$

where e_i is the i-th vector of the canonical basis of $\mathbb{R}^N$; that is, a vector in $\mathbb{R}^N$ with all zeros except for a one in the i-th position.

Proof. As $a,b\in\mathbb{R}^N$, let us denote $a=(\alpha_1,\dots,\alpha_N)$, $b=(\beta_1,\dots,\beta_N)$. Taking the power expansion of $F(a^{[0]}+s,b^{[0]})$ around $(a^{[0]},b^{[0]})$, we obtain

$$F(a^{[0]}+s,b^{[0]}) = F(a^{[0]},b^{[0]})+D_{\alpha_1}F(a^{[0]},b^{[0]})s_1+\dots+D_{\alpha_N}F(a^{[0]},b^{[0]})s_N+\boldsymbol{O}(|s|^2)\ ,$$

and so it is clear that the components of the derivative $D_aF(a^{[0]},b^{[0]})$ are coefficients of $s^{e_1},\dots,s^{e_N}$ of $F(a^{[0]}+s,b^{[0]})$. Analogously, taking the power expansion of $F(a^{[0]},b^{[0]}+s)$ around $(a^{[0]},b^{[0]})$,

$$F(a^{[0]},b^{[0]}+s) = F(a^{[0]},b^{[0]})+D_{\beta_1}F(a^{[0]},b^{[0]})s_1+\dots+D_{\beta_N}F(a^{[0]},b^{[0]})s_N+\boldsymbol{O}(|s|^2)\ ,$$

thus the components of $D_bF(a^{[0]},b^{[0]})$ are coefficients of $s^{e_1},\dots,s^{e_N}$ of $F(a^{[0]},b^{[0]}+s)$. □

Implicit General Linear methods

Once the simplest case has been studied, let us generalize it by considering the setting of implicit General Linear methods; that is, several known values $a_1,\dots,a_\varrho$ and several unknown values $b_1,\dots,b_\varsigma$ related by a system of ς equations $F=(F_1,\dots,F_\varsigma)$ as follows

$$F(a_1,\dots,a_\varrho,b_1,\dots,b_\varsigma)=\begin{cases}F_1(a_1,\dots,a_\varrho,b_1,\dots,b_\varsigma)=0\ ,\\ \vdots\\ F_\varsigma(a_1,\dots,a_\varrho,b_1,\dots,b_\varsigma)=0\ ,\end{cases}\tag{2.7}$$

where $F : (\mathbb{R}^N)^\varrho \times (\mathbb{R}^N)^\varsigma \to (\mathbb{R}^N)^\varsigma$ depends on the method and on the function $f : \mathbb{R}^N \to \mathbb{R}^N$ of the IVP. In the same way as for the case of the implicit Euler method, the values $b_1, \dots, b_\varsigma$ can be approximated by using an iterative method, like the Newton method. Nevertheless, in the context of jet transport of order M with K symbols $s = (s_1, \dots, s_K)$, we have several known jets $[a_1]_K^M, \dots, [a_\varrho]_K^M$ and several unknown jets $[b_1]_K^M, \dots, [b_\varsigma]_K^M$ related by the expression

$$F([a_1]_K^M, \dots, [a_\varrho]_K^M, [b_1]_K^M, \dots, [b_\varsigma]_K^M) = \begin{cases} F_1([a_1]_K^M, \dots, [a_\varrho]_K^M, [b_1]_K^M, \dots, [b_\varsigma]_K^M) = 0 \ , \\ \vdots \\ F_\varsigma([a_1]_K^M, \dots, [a_\varrho]_K^M, [b_1]_K^M, \dots, [b_\varsigma]_K^M) = 0 \ . \end{cases} \tag{2.8}$$

For convenience, we introduce the following notation to denote the several known and unknown variables and their related coefficients and jets.

$$\begin{aligned} &\mathbf{a} = (a_1, \dots, a_\varrho) \ , && \mathbf{b} = (b_1, \dots, b_\varsigma) \ , \\ &\mathbf{a}^{[k]} = ({a_1}^{[k]}, \dots, {a_\varrho}^{[k]}) \ , && \mathbf{b}^{[k]} = ({b_1}^{[k]}, \dots, {b_\varsigma}^{[k]}) \ , \\ &[\mathbf{a}]_K^M = ([a_1]_K^M, \dots, [a_\varrho]_K^M) \ , && [\mathbf{b}]_K^M = ([b_1]_K^M, \dots, [b_\varsigma]_K^M) \ . \end{aligned}$$

Let us study how can we obtain the coefficients to determine the unknown jets $[b_1]_K^M, \dots, [b_\varsigma]_K^M$. As before, observe that truncating the jets at order 0; that is, for $M = 0$ we have that the jets $[a_1]_K^0 = {a_1}^{[0]}, \dots, [a_\varrho]_K^0 = {a_\varrho}^{[0]}$ and $[b_1]_K^0 = {b_1}^{[0]}, \dots, [b_\varsigma]_K^0 = {b_\varsigma}^{[0]}$ are just values, so we are in the context of (2.7) and the coefficients of order 0 can be determined using a Newton method. For higher orders, let us notice that as we are using K symbols, there will no longer be one coefficient per order. In fact, for order m, there will be as many coefficients as multiindices $k = (k_1, \dots, k_K)$ satisfying $|k| = m$, which is exactly characterized in the following Lemma.

Lemma 2.5. The number of coefficients of order m of a formal power series of K symbols is

$$\binom{m + K - 1}{K - 1} = \frac{(m + K - 1)!}{m!(K - 1)!} \ .$$

Having this into account, the coefficients of higher order of the jets $[b_1]_K^M, \dots, [b_\varsigma]_K^M$ can be obtained iteratively using the next theorem.

Theorem 2.6. In the setting of (2.8), the coefficients of the jets $[b_1]_K^M, \dots, [b_\varsigma]_K^M$ are given by

$$D_{\mathbf{b}} F(\mathbf{a}^{[0]}, \mathbf{b}^{[0]}) \begin{bmatrix} {b_1}^{[k]} \\ \vdots \\ {b_\varsigma}^{[k]} \end{bmatrix} = -D_{\mathbf{a}} F(\mathbf{a}^{[0]}, \mathbf{b}^{[0]}) \begin{bmatrix} {a_1}^{[k]} \\ \vdots \\ {a_\varrho}^{[k]} \end{bmatrix} \ , \qquad |k| = 1$$

$$D_{\mathbf{b}} F(\mathbf{a}^{[0]}, \mathbf{b}^{[0]}) \begin{bmatrix} {b_1}^{[k]} \\ \vdots \\ {b_\varsigma}^{[k]} \end{bmatrix} = -F([\mathbf{a}]_K^{m-1}, [\mathbf{b}]_K^{m-1})^{[k]} - D_{\mathbf{a}} F(\mathbf{a}^{[0]}, \mathbf{b}^{[0]}) \begin{bmatrix} {a_1}^{[k]} \\ \vdots \\ {a_\varrho}^{[k]} \end{bmatrix} \ , \quad |k| = m \ , \quad m = 2, \dots, M$$

where $[a_1]_K^M, \dots, [a_\varrho]_K^M$ are known and ${b_1}^{[0]}, \dots, {b_\varsigma}^{[0]}$ have been previously computed.

Proof. Let us proceed by induction on the order, computing the coefficients for each order.

- $m = 1$: To compute ${b_1}^{[k]}, \dots, {b_\varsigma}^{[k]}$ for all k with $|k| = 1$, let us consider the jets only up to order 1. Taking the power expansion of $F([a_1]_K^1, \dots, [a_\varrho]_K^1, [b_1]_K^1, \dots, [b_\varsigma]_K^1)$ around

$([a_1]_K^0, \dots, [a_\varrho]_K^0, [b_1]_K^0, \dots, [b_\varsigma]_K^0)$, we obtain

$$F([a_1]_K^1, \dots, [a_\varrho]_K^1, [b_1]_K^1, \dots, [b_\varsigma]_K^1) = F([a_1]_K^0, \dots, [a_\varrho]_K^0, [b_1]_K^0, \dots, [b_\varsigma]_K^0) + \\ + \sum_{i=1}^{\varrho} \sum_{|k|=1} D_{a_i} F([a_1]_K^0, \dots, [a_\varrho]_K^0, [b_1]_K^0, \dots, [b_\varsigma]_K^0) {a_i}^{[k]} s^k + \\ + \sum_{i=1}^{\varsigma} \sum_{|k|=1} D_{b_i} F([a_1]_K^0, \dots, [a_\varrho]_K^0, [b_1]_K^0, \dots, [b_\varsigma]_K^0) {b_i}^{[k]} s^k + \mathcal{O}(|s|) \ .$$

truncating and recalling $[a_1]_K^0 = {a_1}^{[0]}, \dots, [a_\varrho]_K^0 = {a_\varrho}^{[0]}, [b_1]_K^0 = {b_1}^{[0]}, \dots, [b_\varsigma]_K^0 = {b_\varsigma}^{[0]}$,

$$F([a_1]_K^1, \dots, [a_\varrho]_K^1, [b_1]_K^1, \dots, [b_\varsigma]_K^1) = F({a_1}^{[0]}, \dots, {a_\varrho}^{[0]}, {b_1}^{[0]}, \dots, {b_\varsigma}^{[0]}) + \\ + \sum_{i=1}^{\varrho} \sum_{|k|=1} D_{a_i} F({a_1}^{[0]}, \dots, {a_\varrho}^{[0]}, {b_1}^{[0]}, \dots, {b_\varsigma}^{[0]}) {a_i}^{[k]} s^k + \\ + \sum_{i=1}^{\varsigma} \sum_{|k|=1} D_{b_i} F({a_1}^{[0]}, \dots, {a_\varrho}^{[0]}, {b_1}^{[0]}, \dots, {b_\varsigma}^{[0]}) {b_i}^{[k]} s^k + \mathcal{O}(|s|) \ .$$

We want this expression to be identically 0 (up to order 1), and by hypothesis the coefficients ${b_1}^{[0]}, \dots, {b_\varsigma}^{[0]}$ of order 0 have been chosen so that $F({a_1}^{[0]}, \dots, {a_\varrho}^{[0]}, {b_1}^{[0]}, \dots, {b_\varsigma}^{[0]}) = 0$. Hence we must impose the coefficients of s^k to be 0 for each k, and thus ${b_1}^{[k]}, \dots, {b_\varsigma}^{[k]}$ must satisfy the following relation for all k with $|k| = 1$

$$\sum_{i=1}^{\varsigma} D_{b_i} F({a_1}^{[0]}, \dots, {a_\varrho}^{[0]}, {b_1}^{[0]}, \dots, {b_\varsigma}^{[0]}) {b_i}^{[k]} = -\sum_{i=1}^{\varrho} D_{a_i} F({a_1}^{[0]}, \dots, {a_\varrho}^{[0]}, {b_1}^{[0]}, \dots, {b_\varsigma}^{[0]}) {a_i}^{[k]}$$

where we observe that the sum of derivatives on both sides can be rewritten as

$$\left[D_{b_1} F({a_1}^{[0]}, \dots, {a_\varrho}^{[0]}, {b_1}^{[0]}, \dots, {b_\varsigma}^{[0]}) \ \middle| \ \dots \ \middle| \ D_{b_\varsigma} F({a_1}^{[0]}, \dots, {a_\varrho}^{[0]}, {b_1}^{[0]}, \dots, {b_\varsigma}^{[0]}) \right] \begin{bmatrix} {b_1}^{[k]} \\ \vdots \\ {b_\varsigma}^{[k]} \end{bmatrix} = \\ - \left[D_{a_1} F({a_1}^{[0]}, \dots, {a_\varrho}^{[0]}, {b_1}^{[0]}, \dots, {b_\varsigma}^{[0]}) \ \middle| \ \dots \ \middle| \ D_{a_\varrho} F({a_1}^{[0]}, \dots, {a_\varrho}^{[0]}, {b_1}^{[0]}, \dots, {b_\varsigma}^{[0]}) \right] \begin{bmatrix} {a_1}^{[k]} \\ \vdots \\ {a_\varrho}^{[k]} \end{bmatrix}$$

where we identify the constructed matrix with the Jacobian, obtaining the relation

$$D_{\mathbf{b}} F({a_1}^{[0]}, \dots, {a_\varrho}^{[0]}, {b_1}^{[0]}, \dots, {b_\varsigma}^{[0]}) \begin{bmatrix} {b_1}^{[k]} \\ \vdots \\ {b_\varsigma}^{[k]} \end{bmatrix} = \\ - D_{\mathbf{a}} F({a_1}^{[0]}, \dots, {a_\varrho}^{[0]}, {b_1}^{[0]}, \dots, {b_\varsigma}^{[0]}) \begin{bmatrix} {a_1}^{[k]} \\ \vdots \\ {a_\varrho}^{[k]} \end{bmatrix} \ .$$

- $m - 1 \to m$: Let us assume that we have computed the coefficients of order $1, \dots, m-1$, so we have determined $[b_1]_K^{m-1}, \dots, [b_\varsigma]_K^{m-1}$. To compute the coefficients of order m, let us consider the jets only up to order m. Taking the power expansion of the expression $F([a_1]_K^m, \dots, [a_\varrho]_K^m, [b_1]_K^m, \dots, [b_\varsigma]_K^m)$ around $([a_1]_K^{m-1}, \dots, [a_\varrho]_K^{m-1}, [b_1]_K^{m-1}, \dots, [b_\varsigma]_K^{m-1})$, we obtain

$$F([a_1]_K^m,\ldots,[a_\varrho]_K^m,[b_1]_K^m,\ldots,[b_\varsigma]_K^m) = F([a_1]_K^{m-1},\ldots,[a_\varrho]_K^{m-1},[b_1]_K^{m-1},\ldots,[b_\varsigma]_K^{m-1}) + \\ + \sum_{i=1}^{\varrho}\sum_{|k|=m} D_{a_i}F([a_1]_K^{m-1},\ldots,[a_\varrho]_K^{m-1},[b_1]_K^{m-1},\ldots,[b_\varsigma]_K^{m-1}){a_i}^{[k]}s^k + \\ + \sum_{i=1}^{\varsigma}\sum_{|k|=m} D_{b_i}F([a_1]_K^{m-1},\ldots,[a_\varrho]_K^{m-1},[b_1]_K^{m-1},\ldots,[b_\varsigma]_K^{m-1}){b_i}^{[k]}s^k + \boldsymbol{\mathcal{O}}(|s|^{m+1}) \ ,$$

where, by induction hypothesis, $[b_1]_K^{m-1},\ldots,[b_\varsigma]_K^{m-1}$ have been chosen so that we have $F([a_1]_K^{m-1},\ldots,[a_\varrho]_K^{m-1},[b_1]_K^{m-1},\ldots,[b_\varsigma]_K^{m-1}) = 0$ (up to order $m-1$). In that case, it has some coefficient of order m for each k with $|k| = m$, which will be named (following the current notation) $F([a_1]_K^{m-1},\ldots,[a_\varrho]_K^{m-1},[b_1]_K^{m-1},\ldots,[b_\varsigma]_K^{m-1})^{[k]}$, and notice that higher order terms are not considered, as jets are truncated at order m. Expanding the remaining terms in power series around $({a_1}^{[0]},\ldots,{a_\varrho}^{[0]},{b_1}^{[0]},\ldots,{b_\varsigma}^{[0]})$, we get

$$F([a_1]_K^m,\ldots,[a_\varrho]_K^m,[b_1]_K^m,\ldots,[b_\varsigma]_K^m) = F([a_1]_K^{m-1},\ldots,[a_\varrho]_K^{m-1},[b_1]_K^{m-1},\ldots,[b_\varsigma]_K^{m-1})^{[k]}s^k + \\ + \sum_{i=1}^{\varrho}\sum_{|k|=m} \Big(D_{a_i}F({a_1}^{[0]},\ldots,{a_\varrho}^{[0]},{b_1}^{[0]},\ldots,{b_\varsigma}^{[0]}) + \boldsymbol{\mathcal{O}}(|s|)\Big){a_i}^{[k]}s^k + \\ + \sum_{i=1}^{\varsigma}\sum_{|k|=m} \Big(D_{b_i}F({a_1}^{[0]},\ldots,{a_\varrho}^{[0]},{b_1}^{[0]},\ldots,{b_\varsigma}^{[0]}) + \boldsymbol{\mathcal{O}}(|s|)\Big){b_i}^{[k]}s^k + \boldsymbol{\mathcal{O}}(|s|^{m+1}) \ ,$$

and truncating at order m, we obtain the expression

$$F([a_1]_K^m,\ldots,[a_\varrho]_K^m,[b_1]_K^m,\ldots,[b_\varsigma]_K^m) = F([a_1]_K^{m-1},\ldots,[a_\varrho]_K^{m-1},[b_1]_K^{m-1},\ldots,[b_\varsigma]_K^{m-1})^{[k]}s^k + \\ + \sum_{i=1}^{\varrho}\sum_{|k|=m} D_{a_i}F({a_1}^{[0]},\ldots,{a_\varrho}^{[0]},{b_1}^{[0]},\ldots,{b_\varsigma}^{[0]}){a_i}^{[k]}s^k + \\ + \sum_{i=1}^{\varsigma}\sum_{|k|=m} D_{b_i}F({a_1}^{[0]},\ldots,{a_\varrho}^{[0]},{b_1}^{[0]},\ldots,{b_\varsigma}^{[0]}){b_i}^{[k]}s^k \ .$$

Imposing the coefficients of s^k to be 0 for each k, the values ${b_1}^{[k]},\ldots,{b_\varsigma}^{[k]}$ must satisfy the following relation for all k with $|k| = m$

$$\sum_{i=1}^{\varsigma} D_{b_i}F({a_1}^{[0]},\ldots,{a_\varrho}^{[0]},{b_1}^{[0]},\ldots,{b_\varsigma}^{[0]}){b_i}^{[k]} = \\ - F([a_1]_K^{m-1},\ldots,[a_\varrho]_K^{m-1},[b_1]_K^{m-1},\ldots,[b_\varsigma]_K^{m-1})^{[k]} - \sum_{i=1}^{\varrho} D_{a_i}F({a_1}^{[0]},\ldots,{a_\varrho}^{[0]},{b_1}^{[0]},\ldots,{b_\varsigma}^{[0]}){a_i}^{[k]}$$

and if we proceed as before, this can be written in terms of Jacobian matrices as

$$D_{\mathbf{b}}F({a_1}^{[0]},\ldots,{a_\varrho}^{[0]},{b_1}^{[0]},\ldots,{b_\varsigma}^{[0]})\begin{bmatrix}{b_1}^{[k]}\\ \vdots \\ {b_\varsigma}^{[k]}\end{bmatrix} = \\ - F([a_1]_K^{m-1},\ldots,[a_\varrho]_K^{m-1},[b_1]_K^{m-1},\ldots,[b_\varsigma]_K^{m-1})^{[k]} - D_{\mathbf{a}}F({a_1}^{[0]},\ldots,{a_\varrho}^{[0]},{b_1}^{[0]},\ldots,{b_\varsigma}^{[0]})\begin{bmatrix}{a_1}^{[k]}\\ \vdots \\ {a_\varrho}^{[k]}\end{bmatrix} \ .$$

□

As in the last section, in order to apply Theorem 2.6, it will be useful to have a way to obtain the derivatives that appear in the expression that determine the coefficients. Again, they can be easily obtained using jet transport as seen in the following Lemma.

Lemma 2.7. Let us consider jets of order 1 and the $\varrho \cdot N$ and $\varsigma \cdot N$ symbols $\mathbf{u}$, $\mathbf{v}$ given by

$$\begin{aligned} \mathbf{u} &= (u_1, \dots, u_\varrho) \ , & u_i &= (u_{i1}, \dots, u_{iN}), \ i = 1, \dots, \varrho \ , \\ \mathbf{v} &= (v_1, \dots, v_\varsigma) \ , & v_i &= (v_{i1}, \dots, v_{iN}), \ i = 1, \dots, \varsigma \ . \end{aligned}$$

In the context of (2.8), the derivatives of Theorem 2.3 are given by

$$D_{\mathbf{a}} F(\mathbf{a}^{[0]}, \mathbf{b}^{[0]}) = \\ \left[F(\mathbf{a}^{[0]} + \mathbf{u}, \mathbf{b}^{[0]})^{[U_{1,1}]} \middle| \dots \middle| F(\mathbf{a}^{[0]} + \mathbf{u}, \mathbf{b}^{[0]})^{[U_{1,N}]} \middle| \dots \middle| F(\mathbf{a}^{[0]} + \mathbf{u}, \mathbf{b}^{[0]})^{[U_{\varrho,1}]} \middle| \dots \middle| F(\mathbf{a}^{[0]} + \mathbf{u}, \mathbf{b}^{[0]})^{[U_{\varrho,N}]} \right] \ ,$$

$$D_{\mathbf{b}} F(\mathbf{a}^{[0]}, \mathbf{b}^{[0]}) = \\ \left[F(\mathbf{a}^{[0]}, \mathbf{b}^{[0]} + \mathbf{v})^{[V_{1,1}]} \middle| \dots \middle| F(\mathbf{a}^{[0]}, \mathbf{b}^{[0]} + \mathbf{v})^{[V_{1,N}]} \middle| \dots \middle| F(\mathbf{a}^{[0]}, \mathbf{b}^{[0]} + \mathbf{v})^{[V_{\varsigma,1}]} \middle| \dots \middle| F(\mathbf{a}^{[0]}, \mathbf{b}^{[0]} + \mathbf{v})^{[V_{\varsigma,N}]} \right] \ ,$$

where $U_{i,j}$ and $V_{i,j}$ are the $((i-1)N + j)$-th vector of the canonical basis of $\mathbb{R}^{\varrho N}$ and $\mathbb{R}^{\varsigma N}$ respectively; that is, vectors with all zeros except for a one in the $((i-1)N + j)$-th position.

Proof. As $a_1, \dots, a_\varrho \in \mathbb{R}^N$, let us denote $a_i = (\alpha_{i1}, \dots, \alpha_{iN})$ for each $i = 1, \dots, \varrho$. Being $F = (F_1, \dots, F_\varsigma)$, let us study each F_ℓ for $\ell = 1, \dots, \varsigma$ individually. Taking the power expansion of $F_\ell({a_1}^{[0]} + u_1, \dots, {a_\varrho}^{[0]} + u_\varrho, \mathbf{b}^{[0]})$ around $(\mathbf{a}^{[0]}, \mathbf{b}^{[0]})$, we obtain

$$\begin{aligned} F_\ell({a_1}^{[0]} + u_1, \dots, {a_\varrho}^{[0]} + u_\varrho, \mathbf{b}^{[0]}) = F_\ell(\mathbf{a}^{[0]}, \mathbf{b}^{[0]}) + \\ + D_{\alpha_{11}} F_\ell(\mathbf{a}^{[0]}, \mathbf{b}^{[0]}) u_{11} + \dots + D_{\alpha_{1N}} F_\ell(\mathbf{a}^{[0]}, \mathbf{b}^{[0]}) u_{1N} + \\ \vdots \\ + D_{\alpha_{\varrho 1}} F_\ell(\mathbf{a}^{[0]}, \mathbf{b}^{[0]}) u_{\varrho 1} + \dots + D_{\alpha_{\varrho N}} F_\ell(\mathbf{a}^{[0]}, \mathbf{b}^{[0]}) u_{\varrho N} + \mathcal{O}(|\mathbf{u}|^2) \ , \end{aligned}$$

and so it is clear that the components of the derivative $D_{\mathbf{a}} F_\ell(\mathbf{a}^{[0]}, \mathbf{b}^{[0]})$ are the coefficients of $\mathbf{u}^{e_1}, \dots, \mathbf{u}^{e_N}, \dots, \mathbf{u}^{e_{(\varrho-1)N+1}}, \dots, \mathbf{u}^{e_{(\varrho-1)N+N}}$ of $F_\ell(\mathbf{a}^{[0]} + \mathbf{u}, \mathbf{b}^{[0]})$. The expression of $D_{\mathbf{a}} F(\mathbf{a}^{[0]}, \mathbf{b}^{[0]})$ follows from the fact that its rows are given by $D_{\mathbf{a}} F_1(\mathbf{a}^{[0]}, \mathbf{b}^{[0]}), \dots, D_{\mathbf{a}} F_\varsigma(\mathbf{a}^{[0]}, \mathbf{b}^{[0]})$.

Analogously, let us denote $b_i = (\beta_{i1}, \dots, \beta_{iN})$ for each $i = 1, \dots, \varsigma$ and let us study each F_ℓ for $\ell = 1, \dots, \varsigma$ individually. Taking the power expansion of $F_\ell(\mathbf{a}^{[0]}, {b_1}^{[0]} + v_1, \dots, {b_\varsigma}^{[0]} + v_\varsigma)$ around $(\mathbf{a}^{[0]}, \mathbf{b}^{[0]})$, we obtain

$$\begin{aligned} F_\ell(\mathbf{a}^{[0]}, {b_1}^{[0]} + v_1, \dots, {b_\varsigma}^{[0]} + v_\varsigma) = F_\ell(\mathbf{a}^{[0]}, \mathbf{b}^{[0]}) + \\ + D_{\beta_{11}} F_\ell(\mathbf{a}^{[0]}, \mathbf{b}^{[0]}) v_{11} + \dots + D_{\beta_{1N}} F_\ell(\mathbf{a}^{[0]}, \mathbf{b}^{[0]}) v_{1N} + \\ \vdots \\ + D_{\beta_{\varsigma 1}} F_\ell(\mathbf{a}^{[0]}, \mathbf{b}^{[0]}) v_\varsigma + \dots + D_{\beta_{\varsigma N}} F_\ell(\mathbf{a}^{[0]}, \mathbf{b}^{[0]}) v_{\varsigma N} + \mathcal{O}(|\mathbf{v}|^2) \ , \end{aligned}$$

and proceeding as before, the statement follows. □

Chapter 3

Applications, implementation and numerical experiments

Given the nature of the ideas developed in the preceding chapters, it is of interest to study some of their applications and to perform some numerical experimentation. The efficient implementation of some implicit General Linear methods is discussed in [Jac09], and is out of the scope of this text. With this in mind, we have considered illustrative the particular case of implicit Runge-Kutta methods. This section therefore has two goals: first, to discuss how to implement implicit RK methods using jet transport and perform some tests with that implementation; and second, to study implicit RK methods when applied to some problems of interest in dynamical systems. Once documented, the code developed for this section will be available on GitHub[1].

For that purpose, we start by introducing the van der Pol problem, which we will use as a representative example for the results of our numerical experiments.

3.1 The van der Pol problem

Balthazar van der Pol proposed in the 1920's the problem that now bears his own name. It originates from electronics, but it has become one of the most famous tests for ODE solvers, as it depends on a parameter μ which, if 'large', makes the problem alternate from stiff to non-stiff with a rapidly changing solution. It has been widely studied and appears in standard test sets for IVP, like the 'Geneva test set' [HW96] and the 'Bari test set' [MM08].

The van der Pol problem, henceforth referred to as VDPOL, is simply stated for $z \in \mathbb{R}$ and $\mu > 0$ as

$$z''(x) = \mu(1 - z^2(x))z'(x) - z(x) \ ,$$

and as mentioned in Section 1.1, increasing by one its dimension we can rewrite it as system of order 2 of the form

$$\begin{bmatrix} y_1(x) \\ y_2(x) \end{bmatrix}' = \begin{bmatrix} y_2(x) \\ \mu(1 - y_1^2(x))y_2(x) - y_1(x) \end{bmatrix} ,$$

As mentioned in [HWN93], there exists a unique limit cycle for the VDPOL problem; that is, a stable periodic solution to which all other solutions converge. This property together with its stiffness make the VDPOL problem the ideal example for the upcoming sections.

[1] `https://github.com/PhilPF/Jet-Transport-Implicit-RK`

3.2 The `taylor` package

The first version of the `taylor` package was introduced in [JZ05] as "an ODE solver generator. It reads a system of ODEs and outputs an ANSI C routine that performs a single step of a numerical integration using the Taylor method [...] meant to be called from a user main program", as described in the user's manual [GJZ23]. The latest version of `taylor` also includes support of jet transport which will be the usage we will focus on, as it allows to generate files specifying a suitable polynomial arithmetic that can then be used in another program. In our case, these will be the files that will allow to implement implicit RK methods with the appropriate arithmetic for jet transport.

In order to use them later in this text, let us review the usage of certain features of this package by providing some examples (see [GJZ23]).

- To use `taylor`, we must create an input file with the system of ODEs. For example, the file `vdpol.eq` specifies the ODE of the VDPOL problem

  `vdpol.eq`
  ```
  extern double mu;

  x' = y;
  y' = mu*(1-x^2)*y-x;
  ```

  where `eps` is declared as an undefined parameter using a statement of the form

  ```
  extern [MY_FLOAT|double|float|int|short|char] [var_list];
  ```

  so that it can be initialized later on our main program.

- To declare jet variables, we can use an statement of the form

  ```
  jet [var_list] symbols [number_of_symbols] deg [degree];
  ```

  so, for example, to treat the variables x and y as jets of 2 symbols of order 3 in the VDPOL problem, we could use the file `vdpol_2_3.eq`

  `vdpol_2_3.eq`
  ```
  extern double mu;

  x' = y;
  y' = mu*(1-x^2)*y-x;

  jet x,y symbols 2 deg 3;
  ```

- For generating the necessary files to integrate, for example, the VDPOL problem with the arithmetic specified in `vdpol_2_3.eq`, we can execute the next commands in our terminal (once `taylor` is installed, and from the folder containing `vdpol_2_3.eq`)

  ```
  taylor -header -o [name].h -name [name] vdpol_2_3.eq
  taylor -header_name [name].h -jet -name [name] -step -o [name].c vdpol_2_3.eq
  ```

  which will create, when specifying `[name]`, the files `[name].c` and `[name].h`.

- We remark that the header file `[name].h` must be included in our main file, say `main.c`, as it contains the macro definitions for the arithmetic operations and function calls to `[name].c`. Finally, the file `[name].c` must be compiled together with our main file, using for example

```
gcc main.c [name].c -lm
```

3.3 Implementation of jet transport for implicit Runge-Kutta methods

As in Section 2.3, let us consider the setting of an autonomous IVP (1.3). In that case, the formulation (1.9) of an implicit RK method is

$$Y_i = y_{n-1} + h\sum_{j=1}^{s} a_{ij} f(Y_j) \ , \qquad i = 1, \dots, s$$
$$y_n = y_{n-1} + h\sum_{i=1}^{s} b_i f(Y_i) \ ,$$

where, for the method to be implicit, the matrix $A = [a_{ij}]$ is considered to not be lower triangular with null diagonal elements. As the method is implicit, in order to compute the new step y_n, we must solve an implicit system, so this is a particular case of the problem studied in (2.7), where we have one known value y_{n-1} and s unknown values $Y_1, \dots, Y_s$ related by a system of s equations $F = (F_1, \dots, F_s)$ as follows

$$F(y_{n-1}, Y_1, \dots, Y_s) = 0 \ , \tag{3.1}$$

where, for all $i = 1, \dots, s$, the functions F_i are given by

$$F_i(y_{n-1}, Y_1, \dots, Y_s) = Y_i - y_{n-1} + h\sum_{j=1}^{s} a_{ij} f(Y_j) \ . \tag{3.2}$$

In order to reduce round-off errors in the computation of the solution, it is customary to introduce the smaller quantities

$$z_i = Y_i - y_{n-1} \ , \qquad i = 1 \dots, s \ ,$$

and thus the functions of the system $F = (F_1, \dots, F_s)$ are rewritten for all $i = 1, \dots, s$ as

$$F_i(y_{n-1}, z_1, \dots, z_s) = z_i + h\sum_{j=1}^{s} a_{ij} f(y_{n-1} + z_j) \ . \tag{3.3}$$

Before proceeding with the discussion of the implementation when using jet transport, let us first study how to find a solution of (3.1). For a general system, it has to be approximated iteratively, so one possibility is to consider a simple fixed-point iteration, but as remarked in [HW96], "this transforms the algorithm into a an explicit method and destroys the good stability properties", thus it is not convenient for stiff problems. That being the case, we will use a practical modification of Newton's method.

3.3.1 Simplified Newton iterations

When applying the Newton method to find a solution

$$\mathbf{z} = (z_1, \dots, z_s),$$

of the system (3.1), we will require the Jacobian of F with respect to $\mathbf{z}$; that is,

$$D_{\mathbf{z}}F(y_{n-1},\mathbf{z}) = \begin{pmatrix} I - ha_{11}D_y f(y_{n-1}+z_1) & \dots & -ha_{1s}D_y f(y_{n-1}+z_s) \\ \vdots & & \vdots \\ -ha_{s1}D_y f(y_{n-1}+z_1) & \dots & I - ha_{ss}D_y f(y_{n-1}+z_s) \end{pmatrix}.$$

This matrix can be easily computed using jet transport as shown in Lemma 2.7. Nevertheless, it is important to observe that each step the values $\mathbf{z}$ change, and thus the matrix must be recomputed. That being the case, it is customary to consider the approximation

$$D_y f(y_{n-1}+z_i) \approx D_y f(y_{n-1}), \quad i = 1,\dots,s\ ,$$

and so we obtain an approximated Jacobian for the Newton method given by

$$D_{\mathbf{z}}F(y_{n-1},\mathbf{z}) \approx D_{\mathbf{z}}F(y_{n-1},\mathbf{0}) = I - hA \otimes D_y f(y_{n-1})\ ,$$

where $\mathbf{0} = (0,\dots,0) \in \mathbb{R}^s$. Using this simplified Newton method, we observe that the simplified Jacobian can be computed using jet transport following Lemma 2.7 and, being the same for all iterations, we only need to compute its LU-decomposition once and use it for all Newton iterations.

Starting value

For the Newton method we will require an starting value for $\mathbf{z}$. As the exact solution of (3.3) satisfies $z_i = \mathcal{O}(h)$, $i = 1,\dots,s$, it is customary to choose

$$\mathbf{z}_0 = \mathbf{0}.$$

However, as mentioned in [HW96], better choices for particular cases can be considered.

Stopping criterion

As the Jacobian used in the simplified Newton iterations is approximated, the convergence of the method is no longer quadratic. In fact, as mentioned in [HW96], it is linear. Nevertheless, this can be used to devise an appropriate stopping criterion. For that, let $\mathbf{z}_{k+1}$ denote the value obtained at the $(k+1)$-th iteration of the Newton method, and $\Delta\mathbf{z}_k$ given by the relation

$$\mathbf{z}_{k+1} = \mathbf{z}_k + \Delta\mathbf{z}_k\ .$$

Then, for a desired tolerance $NTOL$ of the Newton method, we should stop the iterations when

$$\eta_k \|\Delta\mathbf{z}_k\| \le \kappa \cdot NTOL$$

where it is recommended (see [HW96]) to choose the following values

$$\eta_0 = \max\{\eta_{\text{old}}, UROUND\}^{0.8}\ , \qquad \eta_k = \frac{\|\Delta\mathbf{z}_k\|/\|\Delta\mathbf{z}_{k-1}\|}{1 - \|\Delta\mathbf{z}_k\|/\|\Delta\mathbf{z}_{k-1}\|} \text{ for } k \ge 1\ , \qquad \kappa \in [10^{-1}, 10^{-2}]\ ,$$

where η_0 is chosen this way to ensure that we are able to stop after the first iteration (which is necessary for linear systems), where η_{old} is the value of the last η_k of the preceding integration step and $UROUND$ the rounding unit (e.g., 10^{-16} for double precision).

3.3.2 Obtaining high order coefficients of the jets

Once we are able to approximate the solution of the system (3.1), we are ready to discuss the case in which we are using jet transport of, say, order M with K symbols $s = (s_1, \dots, s_K)$. That is, following the same notation as in Section 2.3.1, we have a known jet $[y_{n-1}]_K^M$ and s unknown jets $[\mathbf{z}]_K^M = ([z_1]_K^M, \dots, [z_s]_K^M)$ related by the expression

$$F([y_{n-1}]_K^M, [\mathbf{z}]_K^M) = 0 \ . \tag{3.4}$$

As discussed in Section 2.3.1, before applying Theorem 2.6 for obtaining the coefficients of high order of the jets, we first require to know that of order 0, but as $F({y_{n-1}}^{[0]}, \mathbf{z}^{[0]}) = 0$ is exactly system (3.1), we can do so using the simplified Newton iterations presented above. That being the case, let us assume that the coefficients of order 0 are already computed and let us apply the formulas derived in Theorem 2.6 for this particular problem. In this setting, the coefficients of the jets $[\mathbf{z}]_K^M = ([z_1]_K^M, \dots, [z_s]_K^M)$ are determined by the expressions

$$D_{\mathbf{z}}F({y_{n-1}}^{[0]}, \mathbf{z}^{[0]}) \begin{bmatrix} {z_1}^{[k]} \\ \vdots \\ {z_s}^{[k]} \end{bmatrix} = -D_{y_{n-1}}F({y_{n-1}}^{[0]}, \mathbf{z}^{[0]}){y_{n-1}}^{[k]} \ , \qquad |k| = 1$$

$$D_{\mathbf{z}}F({y_{n-1}}^{[0]}, \mathbf{z}^{[0]}) \begin{bmatrix} {z_1}^{[k]} \\ \vdots \\ {z_s}^{[k]} \end{bmatrix} = -F([y_{n-1}]_K^{m-1}, [\mathbf{z}]_K^{m-1})^{[k]} - D_{y_{n-1}}F({y_{n-1}}^{[0]}, \mathbf{z}^{[0]}){y_{n-1}}^{[k]}, \ |k| = m,$$
$$, \ m = 2, \dots, M \ .$$

We observe that the matrices $D_{\mathbf{z}}F({y_{n-1}}^{[0]}, \mathbf{z}^{[0]})$, $D_{y_{n-1}}F({y_{n-1}}^{[0]}, \mathbf{z}^{[0]})$ only depend on the current integration step and not on m, so they only need to be computed once at the beginning of this procedure. Furthermore, as the matrix $D_{\mathbf{z}}F({y_{n-1}}^{[0]}, \mathbf{z}^{[0]})$ is the same for all the linear systems that must be solved, we only need to compute its LU-decomposition once, so the costs of the procedure are therefore lower. In fact, they can be reduced even more by noticing that for each m, all the terms on the right-hand side can be computed simultaneously. As they share the same matrix, we can use a package like `LAPACK` (Linear Algebra PACKage) that allows to solve linear system with several columns of right-hand side, thus saving some computations.

3.3.3 Automatic step size control

In order to adjust the step size h of the method, we require a practical tool to estimate the error. Based on the simple idea: "to repeat the computations with halved step sizes and to compare the results", we present the Richardson Extrapolation as detailed in [HWN93] and [But16].

Studying carefully the known behavior of the error of a RK method of order p as a function of h, we obtain that, starting from y_{n-1} and being $y_n^{(h/2)}$ the solution obtained after two consecutive steps of step size $h/2$ and $y_n^{(h)}$ that obtained after one long step of step size h, the local error (Definition 1.12) of $y_n^{(h)}$ can be extrapolated as

$$y(x_{n-1} + h) - y_n^{(h)} = \frac{y_n^{(h/2)} - y_n^{(h)}}{1 - (1/2)^p} + \mathcal{O}(h^{p+2}) \ ,$$

thus we can obtain the following approximation $\widehat{y}_n$ of order $p+1$ to $y(x_n + h)$

$$\widehat{y}_n = y_n^{(h)} + \frac{y_n^{(h/2)} - y_n^{(h)}}{1 - (1/2)^p} \ .$$

Although other tools can be used to obtain an approximation $\widehat{y}_n$, like embedded Runge-Kutta fomulas (they are not presented in this text, but are of great interest for step size control.

See [HWN93]), what follows should be applied in the same way. The main idea is to control componentwise the error by

$$|y_{n,i} - \widehat{y}_{n,i}| \leq TOL_i \ , \text{ where } TOL_i := ATOL_i + \max\{|y_{n-1,i}|, |y_{n,i}|\} \cdot RTOL_i \ ,$$

being $ATOL_i$ and $RTOL_i$ the desired tolerances for the absolute and relative errors respectively. For that, we consider the following measure of the error (recall that N is the order of the IVP)

$$ERR = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{y_{n,i} - \widehat{y}_{n,i}}{TOL_i}\right)^2} \ ,$$

and we compare it with 1 to construct the optimal step size h_{opt}, given by (recall that p is the order of the RK method)

$$h_{\text{opt}} = h \cdot (1/ERR)^{1/(p+1)} \ .$$

For good code, it is necessary to avoid abrupt changes in the step size, and thus the new step size h_{new} (derived from h_{opt}) is not allowed to increase of decrease too fast

$$h_{\text{new}} = h \cdot \min\left(facmax, \max\left(facmin, fac \cdot (1/ERR)^{1/(p+1)}\right)\right)$$

where some reasonable choices (see [HWN93]) for the constant values in the formula are

$$facmax \in [1.5, 5] \ , \qquad facmin \in [0.1, 0.5] \ , \qquad fac \in \{0.8, 0.9, 0.25^{1/(p+1)}, 0.38^{1/(p+1)}\} \ .$$

In the case that $ERR \leq 1$, the step is accepted and the solution is advanced with y_n, choosing h_{new} as the step size for the next step. Otherwise, the step is rejected and the computations are repeated with the new step size h_{new}. Also, it is recommended to put $facmax = 1$ for the step right after a rejection.

3.3.4 Implementation with `taylor` and some numerical results

We have already seen in Section 3.2 how to generate the necessary files to use jet transport with a desired arithmetic. The goal of this section is to discuss what is necessary for a successful integration of an implicit RK method using `taylor` and to present some numerical results.

For ease and comprehension, let us limit the discussion to the integration of the VDPOL problem using the Radau IIA 5 method, and using jet transport of 2 symbols and order 3.

1. First, we must create the file specifying the ODE and the arithmetic for the integration. Conveniently, in this case, the file should be `vdpol_2_3.eq` introduced in Section 3.2. Using the commands introduced in that same section we can generate the necessary files. For simplicity, we have named (`[name]=taylor`) them `taylor.c` and `taylor.h`.

2. Now, notice that the desired arithmetic for the integration is not the only that is needed, because as described in Lemma 2.7, to be able to compute the matrix $D_{\mathbf{z}}F(y_{n-1}{}^{[0]}, \mathbf{z}^{[0]})$ we will require an arithmetic of $s \cdot N$ symbols and order 1, and to compute $D_{y_{n-1}}F(y_{n-1}{}^{[0]}, \mathbf{z}^{[0]})$ an arithmetic of N symbols and order 1.

   In this case, the Radau IIA 5 method has $s = 3$ stages and the VDPOL problem is of dimension $N = 2$, so we will require two files similar to `vpol_2_3.eq` but, for the file with arithmetic of $s \cdot N = 6$ symbols, we should change the last line to

   ```
   jet x,y symbols 6 deg 1;
   ```

   and for the one with $N = 2$ symbols, we should change it to

   ```
   jet x,y symbols 2 deg 1;
   ```

In order to handle better different files, it is useful to follow a naming convention. In our case, we follow

`[ODE_name]_[symbols]_[degree].eq`

so these two files should be named `vdpol_6_1.eq` and `vdpol_2_1.eq` respectively. As before, we must also generate the corresponding files using the commands introduced in Section 3.2. We have named them `taylor_sigma.c`, `taylor_sigma.h` and `taylor_rho.c`, `taylor_rho.h` respectively.

3. Once the auxiliary files have been created, we can use them in our program. For that, we can crate three files, say, `main.c`, `sigma.c` and `rho.c` that include respectively `taylor.h`, `taylor_sigma.h` and `taylor_rho.h`. In each of them, we must include a function, say `stage_F()`, that allows to evaluate the system (3.4) with their corresponding arithmetic. As this also involves evaluating the function of the IVP, we can take advantage of the fact that it has been defined in the `*.eq` files and do so using `taylor` (see [GJZ23]). Now, we proceed as follows:

   - In the `main.c` file, we need to create a function that allows to compute the next step of the integration using the implicit RK method (in this example, the Radau IIA 5 method). For that, we must write all operations between jets using the macros of the `taylor.h` file that define the arithmetic. Also, we must implement all that has been commented in this section, like the simplified Newton iterations, the automatic step size control and how to obtain the higher order coefficients of the jets during the integration. For the latter, at each step, we will need to compute the necessary matrices $D_{\mathbf{z}}F(y_{n-1}{}^{[0]}, \mathbf{z}^{[0]})$, $D_{y_{n-1}}F(y_{n-1}{}^{[0]}, \mathbf{z}^{[0]})$.
   - To compute the necessary matrices, we should be able to call from our `main.c` file some functions defined in the respective files `sigma.c` and `rho.c` that receive the values $y_{n-1}{}^{[0]}$ and $\mathbf{z}^{[0]}$, and following Lemma 2.7 evaluate them in `stage_F()` together with the respective identity matrix of symbols (again, each using its own arithmetic). Using this, both matrices can be constructed and returned to the `main.c` file so that they can be used to compute the higher order coefficients.
   - We remark that the Jacobian $D_{\mathbf{z}}F(y_{n-1}, \mathbf{0})$ of the simplified Newton iteration can also be computed using the procedure above by simply changing the value $\mathbf{z}^{[0]}$ sent to `sigma.c` by $\mathbf{0}$.

3.4 Computing the power expansion of Poincaré maps

As detailed in [GJ20], "a Poincaré section is defined as a codimension 1 smooth manifold transversal to the flow. [...] They are a standard tool to study the dynamics in regions of the space where orbits have some kind of recurrence." In this section we will follow [Sim90] and [GJ20] to study how to compute the power expansion of Poincaré maps and periodic orbits using jet transport. We remark that this process is independent of the choice of the numerical integrator. Nevertheless, we will also apply the theory of this section do the VDPOL problem, so given its stiffness, it will be of interest to study its behavior using implicit RK methods.

Henceforth, $\Phi(x; x_0, y_0)$ denotes the solution at time x of an autonomous IVP (1.3) of dimension N with initial condition $y(x_0) = y_0$. We will denote Poincaré sections by Σ and the corresponding Poincaré map by P.

3.4.1 Computing the return time

First, we will follow [Sim90] to show how to compute the return time to the section. For that, instead of considering the first return map to a section Σ, we will present the more general setting where we have two sections transversal to the flow given by

$$\Sigma_1 = \{y \in \mathbb{R}^N \ ; \ g_1(y) = 0\} \ , \qquad \Sigma_2 = \{y \in \mathbb{R}^N \ ; \ g_2(y) = 0\} \ ,$$

and the initial and final points $y^{(0)}$ and $y^{(1)}$ are in Σ_1 and Σ_2 respectively. That being the case, we define

$$\begin{array}{rcl} P \ : \ \Sigma_1 & \longrightarrow & \Sigma_2 \\ y^{(0)} & \longmapsto & y^{(1)} = \Phi(T(y^{(0)}); 0, y^{(0)}) \end{array} \ .$$

where, so that $y^{(1)} \in \Sigma_2$, we must impose that

$$g_2\Big(\Phi(T(y^{(0)}); 0, y^{(0)})\Big) = 0 \ . \tag{3.5}$$

The value that we aim to approximate is the arrival time $T(y^{(0)})$ to Σ_2, which we stress that depends on the initial point $y^{(0)}$ in Σ_1. To compute it, we notice that it is enough to study condition (3.5); that is, we can deduce that we have crossed Σ_2 by studying the changes of sign of the following function

$$\psi(x) = g_2\Big(\Phi(x; 0, y^{(0)})\Big) \ .$$

Once we have located a point t_0 where ψ has changed sign for the first time (in the case that we are looking for the k-th sign change, everything is equivalent), to refine the approximation to $T(y^{(0)})$ we can proceed by applying a Newton method to ψ; thus the Newton iterations will be of the form

$$t_{k+1} = t_k - \frac{g_2\Big(\Phi(t_k; 0, y^{(0)})\Big)}{\Big\langle \nabla g_2\Big(\Phi(t_k; 0, y^{(0)})\Big), f\Big(\Phi(t_k; 0, y^{(0)})\Big)\Big\rangle} \ ,$$

and iterations can be stopped when $\psi(t_{k+1})$ is sufficiently close to 0. In that case, $t_{k+1} \approx T(y^{(0)})$.

One particular case that is of interest is that where the Poincaré section is of the form

$$\Sigma = \{y = (y_1, \ldots, y_N) \in \mathbb{R}^N \ ; \ y_\ell = C\} \ .$$

where $\ell \in \{1, \ldots, N\}$ is a certain index and C some constant value. In that case, the function ψ is given by

$$\psi(x) = \pi_\ell \Phi(x; 0, y^{(0)}) - C \ ,$$

where we denote by $\pi_\ell \Phi(x; 0, y^{(0)})$ the ℓ-th coordinate of $\Phi(x; 0, y^{(0)})$. Noticing that

$$\psi'(x) = \pi_\ell D_t \Phi(x; 0, y^{(0)}) = \pi_\ell f\Big(\Phi(x; 0, y^{(0)})\Big)$$

the Newton iterations take the form

$$t_{k+1} = t_k - \frac{\pi_\ell \Phi(t_k; 0, y^{(0)}) - C}{\pi_\ell f\Big(\Phi(t_k; 0, y^{(0)})\Big)} \ .$$

3.4.2 Power expansion of Poincaré maps

Even though other possibilities exist for Poincaré sections, we will consider only those based on spatial sections, known as 'spatial Poincaré sections'. For computing their power expansion we will follow [Gim19]. For that, let the Poincaré section Σ be a hyperplane in general position, $\vec{n}$ its normal vector and $y^{(0)} \in \Sigma$. Without loss of generality, we can choose linearly independent unitary vectors $v_1, \ldots, v_{N-1}$ such that

$$\Sigma = \{y \in \mathbb{R}^N \ ; \ y = y^{(0)} + v_1 s_1 + \cdots + v_{N-1} s_{N-1}\} \ .$$

In this setting, we assume that for after some time $\tau_0 := T(y^{(0)})$ (that depends on the initial point and can be computed using last section), the trajectory returns to the section; that is, $\Phi(\tau_0; 0, y^{(0)}) \in \Sigma$. The Poincaré map is then defined as

$$\begin{aligned} P \ : \ \Sigma &\longrightarrow \Sigma \\ y^{(0)} &\longmapsto \Phi(\tau_0; 0, y^{(0)}) \end{aligned}$$

We observe that by integrating the IVP starting from the point $y^{(0)} + \overline{s}$, where $\overline{s} = v_1 s_1 + \cdots + v_{N-1} s_{N-1}$ using jet transport of order M with $N-1$ symbols $s = (s_1, \ldots, s_{N-1})$ up to time τ_0, we obtain the power expansion up to order M of the flow at time τ_0 with respect to the $N-1$ variables that are coordinates on Σ. Nevertheless, we must notice that this is not the power expansion of the Poincaré map because it does not lay inside Σ; that is, using the notation introduced in Section 2.3.1, the jet $\Phi(\tau_0; 0, y^{(0)} + \overline{s})$ is such that

$$\begin{aligned} &\Phi(\tau_0; 0, y^{(0)} + \overline{s})^{[0]} = \Phi(\tau_0; 0, y^{(0)}) \in \Sigma \ , \\ &\Phi(\tau_0; 0, y^{(0)} + \overline{s})^{[k]} \notin \Sigma \ , \quad \text{for } |k| = m \ , \quad m = 1, \ldots, M. \end{aligned}$$

In order to ensure that the higher order coefficients also lay in Σ, let us recall that the return time to the Poincaré section depends on the initial point, thus it also depends on s; that is, we have to write is as a formal power series on s

$$T(y^{(0)} + \overline{s}) = T(y^{(0)}) + \sum_{|k|=1}^{M} \tau_k s^k =: \tau_0 + \sum_{|k|=1}^{M} \tau_k s^k =: [\tau]_{N-1}^{M}$$

where the coefficients $\tau_k := T(y^{(0)} + \overline{s})^{[k]}$ are real numbers that must be determined by the condition

$$\Phi(T(y^{(0)} + \overline{s}); 0, y^{(0)} + \overline{s}) \in \Sigma \ ,$$

what can be done recurrently degree by degree.

Theorem 3.1. The values τ_k for $|k| = m$ and $m = 1, \ldots, M$ that determine the power expansion of the return time to the Poincaré section so that the condition (3.4.2) is satisfied are given by

$$\tau_k = -\frac{\left\langle \Phi(\tau_0; 0, y^{(0)} + \overline{s})^{[k]}, \vec{n} \right\rangle}{\left\langle f\Big(\Phi(\tau_0; 0, y^{(0)})\Big), \vec{n} \right\rangle} \ , \qquad |k| = 1$$

$$\tau_k = -\frac{\left\langle \Phi\Big([\tau]_{N-1}^{m-1}; 0, y^{(0)} + \overline{s}\Big)^{[k]}, \vec{n} \right\rangle}{\left\langle f\Big(\Phi(\tau_0; 0, y^{(0)})\Big), \vec{n} \right\rangle} \ , \quad |k| = m \ , \quad m = 2, \ldots, M \ .$$

Proof. Starting from $m = 1$, let us consider the power expansion of the flow at time $\tau_0 + \sum_{|k|=1} \tau_k$ around τ_0, considering jets only up to order 1

$$\Phi\Big(\tau_0 + \sum_{|k|=1} \tau_k s^k; 0, y^{(0)} + \overline{s}\Big) =$$
$$= \Phi(\tau_0; 0, y^{(0)} + \overline{s}) + D_t\Phi(\tau_0; 0, y^{(0)} + \overline{s}) \sum_{|k|=1} \tau_k s^k$$

where we observe that the first term is also a jet with the symbols s, so let us write its expansion explicitly up to order 1. Also, notice that $D_t\Phi$ is multiplying a term of order 1, so we should only consider the coefficient of order 0; that is,

$$= \Phi(\tau_0; 0, y^{(0)} + \overline{s})^{[0]} + \sum_{|k|=1} \Phi(\tau_0; 0, y^{(0)} + \overline{s})^{[k]} s^k + D_t\Phi(\tau_0; 0, y^{(0)}) \sum_{|k|=1} \tau_k s^k$$

and as we have that $\Phi(\tau_0; 0, y^{(0)} + \overline{s})^{[0]} = \Phi(\tau_0; 0, y^{(0)})$ and, by definition of the flow of the IVP, $D_t\Phi(\tau_0; 0, y^{(0)}) = f\Big(\Phi(\tau_0; 0, y^{(0)})\Big)$, we obtain

$$= \Phi(\tau_0; 0, y^{(0)}) + \sum_{|k|=1} \Phi(\tau_0; 0, y^{(0)} + \overline{s})^{[k]} s^k + f\Big(\Phi(\tau_0; 0, y^{(0)})\Big) \sum_{|k|=1} \tau_k s^k \ . \quad (3.6)$$

Finally, to impose condition (3.4.2) up to order 1 (recall that it is already satisfied for order 0), we must ask that

$$\sum_{|k|=1} \Big\langle \Phi(\tau_0; 0, y^{(0)} + \overline{s})^{[k]} + \tau_k f\Big(\Phi(\tau_0; 0, y^{(0)})\Big), \vec{n} \Big\rangle = 0 \ ,$$

that is, the coefficients τ_k for $|k| = 1$ are given by

$$\tau_k = -\frac{\Big\langle \Phi(\tau_0; 0, y^{(0)} + \overline{s})^{[k]}, \vec{n} \Big\rangle}{\Big\langle f\Big(\Phi(\tau_0; 0, y^{(0)})\Big), \vec{n} \Big\rangle} \ .$$

Now, for the higher order coefficients, let us assume that we have computed the values τ_k for $|k| \leq m-1$; that is, all the coefficients of $\tau_0 + \sum_{|k|=1}^{m-1} \tau_k s^k = [\tau]_{N-1}^{m-1}$ are known and we want to determine the τ_k for $|k| = m$. Let us then proceed as before by considering the power expansion of the flow at time $[\tau]_{N-1}^{m} = [\tau]_{N-1}^{m-1} + \sum_{|k|=m} \tau_k s^k$ around $[\tau]_{N-1}^{m-1}$, considering jets only up to order m

$$\Phi\Big([\tau]_{N-1}^{m-1} + \sum_{|k|=m} \tau_k s^k; 0, y^{(0)} + \overline{s}\Big) =$$
$$= \Phi\Big([\tau]_{N-1}^{m-1}; 0, y^{(0)} + \overline{s}\Big) + D_t\Phi\Big([\tau]_{N-1}^{m-1}; 0, y^{(0)} + \overline{s}\Big) \sum_{|k|=m} \tau_k s^k$$

as before, we can write the explicit expansion of the first term as a jet up to order m, and, as $D_t\Phi$ is multiplying a term of order m, we should only consider the coefficient of order 0, which we have seen that is exactly $f\Big(\Phi(\tau_0; 0, y^{(0)}\Big)$, obtaining

$$= \sum_{|k|\leq m-1} \Phi\Big([\tau]_{N-1}^{m-1}; 0, y^{(0)} + \overline{s}\Big)^{[k]} s^k + \sum_{|k|=m} \Phi\Big([\tau]_{N-1}^{m-1}; 0, y^{(0)} + \overline{s}\Big)^{[k]} s^k + f\Big(\Phi(\tau_0; 0, y^{(0)})\Big) \sum_{|k|=m} \tau_k s^k$$

Analogously, to impose condition (3.4.2) up to order m, as it is already satisfied up to order $m-1$, we must ask that

$$\sum_{|k|=m} \Big\langle \Phi\Big([\tau]_{N-1}^{m-1}; 0, y^{(0)} + \overline{s}\Big)^{[k]} + \tau_k f\Big(\Phi(\tau_0; 0, y^{(0)})\Big), \vec{n} \Big\rangle = 0 \ ,$$

that is, the coefficients τ_k for $|k| = m$ are given by

$$\tau_k = -\frac{\left\langle \Phi\left([\tau]_{N-1}^{m-1}; 0, y^{(0)} + \overline{s}\right)^{[k]}, \vec{n} \right\rangle}{\left\langle f\left(\Phi(\tau_0; 0, y^{(0)})\right), \vec{n} \right\rangle} .$$

□

There are a few things to consider about the above. First, we notice that the case $M = 1$ is much simpler than $M > 1$, as for the first we don't have the need to consider the time variable as a jet on our numerical integrator; that is, for $M = 1$ we can simply compute the values τ_k for $|k| = 1$ using the expression in Theorem 3.1 and then, to obtain the flow at $\tau_0 + \sum_{|k|=1} \tau_k s^k$, we can apply the formula derived in (3.6), as all needed values are known. For the case that $M > 1$, we can proceed recursively on m for $m = 1, \ldots, M$ by computing τ_k for $|k| = m$ and then perform a step of step size

$$h = \sum_{|k|=m} \tau_k s^k \tag{3.7}$$

with our numerical integrator. Another thing that must be noted in this context is that, as explained in [Gim+23], for a numerical integrator with local error of the form $\boldsymbol{O}(h^{p+1})$, after a step of the form (3.7), the error contains terms of order $p + 1$ in s, so it is impossible to obtain a power expansion of the Poincaré map of order higher than p. Therefore, the numerical integration for this recursive process should be done using a method like the Taylor method, as it allows to easily obtain a high order truncation error.

3.4.3 Implementation with implicit RK methods and some numerical results

Let us discuss in this section how to proceed when computing, for example, the power expansion of the Poincaré map of the periodic orbit of the VDPOL problem (for a fixed μ) using an implicit RK method. For that, we should explain two aspects: first, how to obtain periodic orbits; and second, how to compute their power expansion.

For the first topic, it is fundamental the idea that, to find the periodic orbit of the VDPOL problem (for a fixed μ) we should find the fixed point of the Poincaré map. For that, we choose the Poincaré section

$$\Sigma = \{y = (y_1, y_2) \in \mathbb{R}^2 \ ; \ y_2 = 0\} ,$$

and the Poincaré map given by

$$\begin{aligned} P \ : \ \Sigma &\longrightarrow \Sigma \\ y &\longmapsto \Phi(T(y); 0, y) \end{aligned}$$

and, as explained in [Sim90], the periodic orbit will be determined by some $y^* \in \Sigma$ such that $P(y^*) = y^*$. To determine it, we proceed as follows:

1. Let us choose an initial point $y^{(0)} \in \Sigma$. As mentioned in [MM08], $y^* \approx (2, 0)$, so we could choose $y^{(0)} = (2, 0)$.

2. We compute an approximation to the second return time $T(y^{(0)})$ (i.e., given by the second sign change) using Section 3.4.1. We will do so integrating with an implicit RK method using jet transport of 1 symbol and order 1 as described in Section 3.3, with initial value

	(state var.)	s_1
y_1	$y_1^{(0)}$	1
y_2	0	0

3. From step 2, if we recall Section 3.4.2, by using jet transport of 1 symbol and order 1, we have also computed the first derivative of the flow at time $T(y^{(0)})$ with respect to y_1. Using Section 3.4.2 we can project this derivative to the section to obtain the derivative of the Poincaré map at $y^{(0)}$ with respect to y_1; that is, $D_{y_1}P(y^{(0)})$.

4. We apply one step of Newton's method to $\pi_1 P(y) - y_1$, using as initial value $y_1^{(0)}$ by doing

$$y_1^{(1)} = y_1^{(0)} - \frac{\pi_1 P(y^{(0)}) - y_1^{(0)}}{\pi_1 D_{y_1} P(y^{(0)}) - 1}$$

where by π_1 we denote the projection to the first coordinate.

5. In step 4 we have obtained a better approximation $y^{(1)} = (y_1^{(1)}, 0)$ of y^*. We check if $P(y^{(1)}) - y^{(1)}$ is (close to) 0. If it is not, we go back to step 2 substituting $y^{(0)}$ by $y^{(1)}$.

Using the algorithm above, we have obtained the results in Table 3.1 for y_1^* and $T(y^*)$ for different values of μ, integrating with the Radau IIA 5 method, choosing the tolerances $ATOL = RTOL = 10^{-12}$, $NTOL = 10^{-15}$ and initial step size $h = 10^{-10}$. We have compared the computed periods $T(y^*)$ with the reference values in [Amo22], underlining all the coincident digits.

μ	y_1^*	$T(y^*)$
1	2.008619860874817	6.663286859322704
10	2.014285360925673	19.078369566919214
100	2.001318681176584	162.837071092175393

Table 3.1: First coordinate and period of the fixed points of the periodic orbit of the VDPOL problem for different values of μ, computed with the Radau IIA 5 method.

In order to visualize this algorithm, we have plotted in Figure 3.1 two iterations of this method for the simple case $\mu = 1$, choosing the more illustrative initial value $y^{(0)} = (1, 0)$.

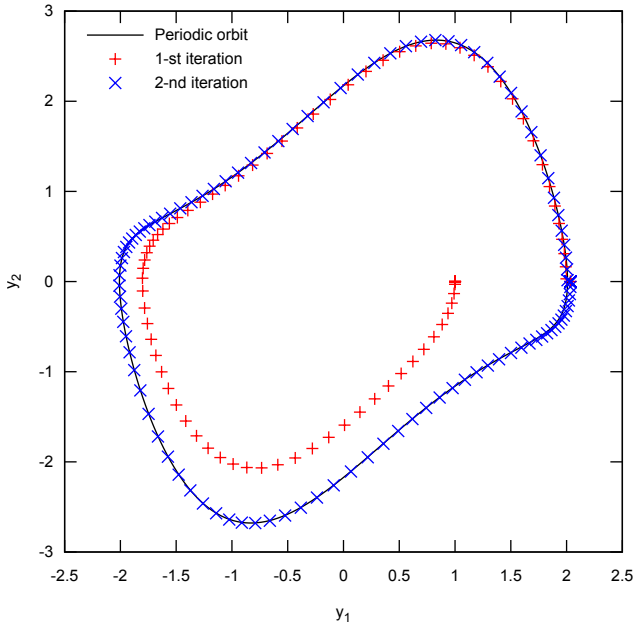

Figure 3.1: Two iterations of the method used to compute periodic orbits of the VDPOL problem as fixed points of the Poincaré map.

Once that the periodic orbit is determined (for a given μ), let us compute the power expansion of its Poincaré map with respect to μ. For that, as detailed in [Sim90], we can increase the dimensionality of the system by 1 introducing a new variable y_3 satisfying

$$y_3' = 0 \ , \qquad y_3(0) = \mu$$

and so being $y = (y_1, y_2, y_3) \in \mathbb{R}^3$, the VDPOL problem can be rewritten as

$$y' = f(y_1, y_2, y_3) = \begin{bmatrix} y_2 \\ \mu(1 - y_1^2)y_2 - y_1 \\ 0 \end{bmatrix} ,$$

We remark that this has to be reflected in `taylor`; that is, we must modify the `*.eq` file. For example, to compute the power expansion up to order $M = 3$, we could use the file `vdpol_1_3.eq`

`vdpol_1_3.eq`

```
x' = y;
y' = mu*(1-x^2)*y-x;
mu' = 0;

jet x,y,mu symbols 1 deg 3;
```

With this consideration, let us show how to compute the power expansion of the Poincaré map with respect to μ of order M of the periodic orbit.

1. Let us fix a value μ and its corresponding fixed point of the Poincaré map y^* and period $T(y^*)$. We integrate up to time $T(y^*)$ with an implicit RK method using jet transport of 1 symbol and order M as described in Section 3.3, using as initial value

	(state var.)	s_1	s_1^2	...	s_1^M
y_1	y_1^*	0	0		0
y_2	0	0	0		0
y_3	μ	1	0		0

2. From step 1, if we recall Section 3.4.2, by using jet transport of 1 symbol and order M, we have also computed the power expansion of the flow at time $T(y^*)$ with respect to y_3; that is, with respect to μ. Using Section 3.4.2 we can project this power expansion to the section to obtain the power expansion of the Poincaré map at y^* with respect to μ.

Conclusions

Throughout this project we have introduced two relevant topics: General Linear methods (GLM) and the numerical integration of variational equations (VE). In addition, we have taken a successful step towards the study on how GLM can be applied to solve VE.

In Chapter 1, we have first presented Linear multistep methods (LMM) and Runge-Kutta (RK) methods, studying their most fundamental properties and observing that they possess strengths and weaknesses in terms of both their numerical behavior and their theoretical development. In pursuit of a natural extension of these methods, we have introduced the multistep-multistage family of General Linear methods, studying their convergence, order and linear stability properties. As we have seen, their study is much harder than for the methods that preceded them. Nonetheless, this broadens the range of possibilities to more powerful methods, which could be introduced and developed in future work.

In Chapter 2, we have studied the technique of jet transport for the numerical integration of variational equations. It is in this section where we have presented our main contribution, showing that the numerical integration of an IVP using jet transport with a General Linear method is equivalent to the integration of its VE using the same method, also deriving the relation that the higher order coefficients of the jets must satisfy in order to be solutions of an implicit system defined by the stages of a GLM.

In Chapter 3, we have discussed the implementation of theory developed in the preceding sections. However, we have not done so with the most possible generality, leaving the implementation of jet transport with General Linear methods for future work. On the other hand, we have successfully applied this theory for the particular case of implicit Runge-Kutta methods, commenting a few results in the numerical integration of VE when applied to the computation of interesting mathematical objects in the theory of dynamical systems.

In conclusion, I personally believe that this project has allowed me to learn a lot about numerical integration and its use in dynamical systems, paving up the way for many potential future work options.

Bibliography

[Amo22] Paolo Amore. "Computing the solutions of the van der Pol equation to arbitrary precision". In: *Physica D: Nonlinear Phenomena* 435 (July 2022), p. 133279.

[But06] J. C. Butcher. "General linear methods". In: *Acta Numerica* 15 (May 2006), pp. 157–256.

[But16] J. C. Butcher. *Numerical methods for ordinary differential equations*. Third edition. Chichester, West Sussex, United Kingdom: Wiley, 2016. 513 pp.

[Dah56] Germund Dahlquist. "Convergence and stability in the numerical integration of ordinary differential equations". In: *MATHEMATICA SCANDINAVICA* 4 (Dec. 1, 1956), pp. 33–53.

[Gim19] Joan Gimeno. "Effective methods for recurrence solutions in delay differential equations". PhD thesis. Sept. 2019.

[GJ20] Joan Gimeno and Àngel Jorba. "Using automatic differentiation to compute periodic orbits of delay differential equations". In: *Discrete & Continuous Dynamical Systems - B* 25.12 (2020), pp. 4853–4867.

[GJZ23] Joan Gimeno, Àngel Jorba, and Maorong Zou. *taylor User's manual*. Version 2.1. May 2023.

[Gim+23] Joan Gimeno et al. "Numerical integration of high-order variational equations of ODEs". In: *Applied Mathematics and Computation* 442 (Apr. 2023), p. 127743.

[HW96] Ernst Hairer and Gerhard Wanner. *Solving Ordinary Differential Equations II*. Vol. 14. Springer Series in Computational Mathematics. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996.

[HWN93] Ernst Hairer, Gerhard Wanner, and Syvert P. Nørsett. *Solving Ordinary Differential Equations I*. Vol. 8. Springer Series in Computational Mathematics. Berlin, Heidelberg: Springer Berlin Heidelberg, 1993.

[Jac09] Zdzisław Jackiewicz. *General linear methods for ordinary differential equations*. Hoboken, N.J: Wiley, 2009. 482 pp.

[JZ05] Àngel Jorba and Maorong Zou. "A Software Package for the Numerical Integration of ODEs by Means of High-Order Taylor Methods". In: *Experimental Mathematics* 14.1 (Jan. 2005), pp. 99–117.

[MM08] Francesca Mazzia and Cecilia Magherini. *Test set for initial value problem solvers, release 2.4*. Technical Report 4. Department of Mathematics, University of Bari, Italy, Feb. 2008.

[Sim90] C. Simó. "On the analytical and numerical approximation of invariant manifolds." In: *Les Méthodes Modernes de la Mecánique Céleste*. Editions Frontières. Paris, Jan. 1990, pp. 285–329.