Numerical study of time integration methods based on perturbation techniques : application to mechanical problems

Cynthia Tayeh

DOCTORAT BRETAGNE SCIENCES LOIRE POUR L'INGENIEUR

Université Bretagne Sud

THÈSE DE DOCTORAT DE

L'UNIVERSITÉ DE BRETAGNE SUD

ÉCOLE DOCTORALE N° 602
Sciences pour l'Ingénieur
Spécialité : *Mécanique des milieux fluides*

Par
Cynthia TAYEH

Étude numérique de méthodes d'intégration temporelle basées sur des techniques de perturbation. Application à des problèmes de mécanique.

Thèse présentée et soutenue à l'Université Bretagne Sud, le 26 Janvier 2022
Unité de recherche : Institut de recherche Dupuy de Lôme

Composition du Jury :

Président :	M. Aziz HAMDOUNI	Professeur des Universités, LaSIE, Université de la Rochelle
Rapporteurs :	M. Bruno COCHELIN	Professeur des Universités, LMA, Centrale Marseille
	M. Mejdi AZAIEZ	Professeur des Universités, ENSCBP Bordeaux INP
Examinateur :	Mme Marianne BERINGHIER	Maître de conférences, ENSMA, Poitiers
Directeur de thèse :	M. Jean-Marc CADOU	Maître de conférences, Université Bretagne Sud, Lorient
Co-encadrant de thèse :	M. Yann GUEVEL	Ingénieur d'études, Université Bretagne Sud, Lorient

ACKNOWLEDGEMENT

Tout d'abord, je remercie le bon Dieu, tout puissant, de m'avoir donné la force pour survivre, ainsi que l'audace pour dépasser toutes les difficultés et mener à bien ce travail. Ce travail n'aurait jamais vu le jour sans l'appui d'un certain nombre de personnes que j'aimerais remercier ici.

Je tiens en tout premier lieu à remercier mon directeur de thèse Monsieur Jean-Marc Cadou et mon co-encadrant Monsieur Yann Guevel pour la confiance qu'ils m'ont accordée en me proposant ce sujet. Même s'il y a eu de nombreux moments difficiles durant les trois ans, je n'aurais jamais pu y aller jusqu'au bout, sans leur soutien, leur appui scientifique, leurs outils pédagogiques, et surtout leur côté humain. Je n'oublierai jamais le bénéfice qu'ils ont apporté sur le développement et l'amélioration de ma réflexion en mathématiques pures vers le monde numérique. Je ne trouve pas les mots pour définir à quel point je leur suis reconnaissante. Pendant ces trois années, j'ai vraiment eu l'impression de faire partie non seulement d'une équipe, mais aussi d'une famille. Je les remercie de m'avoir permis de réaliser cette thèse dans des conditions optimales.

Je remercie particulièrement Monsieur Aziz Hamdouni qui m'a fait l'honneur de présider mon jury de thèse. J'apprécie ses remarques et ses conseils sur la méthode de resommation. Je remercie également Monsieur Bruno Cochelin, Monsieur Mejdi Azaiez pour leur rôle de rapporteurs et Madame Marianne Beringhier pour son rôle d'examinatrice. Je les remercie pour l'intérêt qu'ils ont porté à mon travail de thèse et pour leurs remarques constructives qui ont été bien prises en compte. Je suis fière d'avoir partagé mon travail avec eux.
Je remercie également Monsieur Nicolas Jacques et Monsieur Cyrille Allery d'avoir accepté d'être les membres du CSI (comité de suivi individuel) pour ces trois années de thèse.

Je tiens à remercier tous les autres membres du groupe, en particulier Laetitia, Grégory et Khaoula. Je n'oublierai pas les moments partagés au bureau avec Nejia, Diane, Oudayfa, Léopold, Thibault, Bernardette et surtout Stéphanie, et Mathias. J'ai partagé avec Mathias mes galères de thèse ainsi que des moments de fous rires qui ont soudé notre amitié. Je n'oublierai jamais les cours de Libanais tous les jours avec lui, les soirées au bureau jusqu'à 22 h pour déboguer les codes, et les sandwichs à la Mozzarella à 23 h. Je remercie Stéphanie pour les différentes soirées passées ensemble, son départ de Lorient

n'a pas été facile pour moi. Je remercie également les autres doctorants : Xavier, Adrien, Alemeh, Nora, Malik, Delphine, Ghita, Thomas, Miguel, Maxime et Julien pour tous les bons moments partagés durant ces trois années.
Je n'oublierai pas Amine, mon partenaire de déjeuner tous les jours au RU. Je le remercie pour toutes les pauses-café, les différents cadeaux, les chocolats, les belles surprises, et les différents conseils administratifs et moraux. Je remercie également Asmahan, avec elle, nous nous sommes soutenus dans les bons et dans les mauvais moments (notamment le COVID) et je suis immensément fière du chemin que nous avons parcouru ensemble.

Mes pensées vont également à Bertrand qui a sauvé mes fichiers et mon ordinateur plusieurs fois. Merci également à Béatrice, Virginie, Noluenn, et Roseline pour leurs aides. Je remercie aussi Ahmad pour nos différents échanges scientifiques à la rochelle sur les méthodes de resommation. Je tiens à remercier mes autres amis libanais en France et à l'étranger qui m'ont permis de vivre des moments de bonheur durant ces 3 années : Freddy, Marwan, Hassan, Marianne, Grace, Joe, Ghewa, Zeinab, Jihane, Maria, Abbas, Patrick, Marie-Noël, Jad, Machhour, Marinelle, Elie, Joseph, Mohamad, Tatiana et Romaine. J'ai une pensée particulière pour Marie-Noël, avec elle, j'ai partagé des jours d'aventure en Bretagne que je n'oublierai jamais. Je tiens à remercier Kassem pour les différentes soirées à la résidence qui m'ont donné l'impression d'avoir une seconde famille en Bretagne. Je sais que cela a été dur pour lui de me supporter dans mes moments stressants et de m'accompagner durant ces années. Je n'ai pas de mots pour lui dire combien je suis reconnaissante, 1000 merci Kassem. Merci aussi pour le chef Hamza pour tous les plats et les délicieux desserts qu'il a préparé dans la résidence. Merci également à Assi pour toutes les promenades en voiture, les dîners ensemble, l'aide apportée pour faire les courses, et pour avoir été là quand je me sentais seule.

Et enfin, mes remerciements les plus sincères vont à mes parents qui m'ont toujours encouragé à poursuivre mes études à l'étranger. Je leur remercie pour leur compréhension et leur soutien pendant les moments difficiles à l'étranger. Je ne serai jamais assez reconnaissante pour tout ce que vous avez fait pour moi depuis le début. Sans vous, je n'en serais pas là ! Je n'oublierai jamais à remercier ma sœur Nancy et mon frère Elie pour les appels et les visios chaque jour pour me soutenir. Je réalise vraiment ce que signifie avoir un frère et une sœur malgré la distance de 3155 km qui nous sépare.

À mon père et ma mère,

Table of Contents

INTRODUCTION

This thesis is related to the numerical solution of time-dependent problems. In this context, consider an ordinary differential or a partial derivative equation in the following form:

$$\frac{du}{dt}(t,x) = F(t,u) \quad u, x \in \mathbb{R}^n \tag{1}$$

where u is the unknown and F is a nonlinear operator.
The possible resolution of such a problem can be analytical or numerical. The analytical solution does not exist in some cases, and it is difficult to find in others. This is especially true for problems with multiple degrees of freedom. As a result, a numerical method is employed. The usual numerical schemes for evolution problems are based on time discretization and, if the problem is modeled by a partial differential equation, a spatial discretization is then used. Established approaches for spatial discretization include the finite element method (FEM), the finite difference method (FDM), and the finite volume method (FVM). These latter aim to find approximate solutions of partial differential equations, in a set of points rather than in the whole domain. Concerning time discretization, the first numerical method for approximating the first-order derivative in time, which is $\left(\frac{du}{dt}(t,x)\right)$, was introduced in 1768 by Leonhard Euler who treated it in his book [1]. Since then, various techniques have been developed for different orders, which are based on the temporal discretization of the problem to be solved, by advancing the model by successive discrete-time steps denoted by Δt. Depending on the type of formula used to approximate the solutions, a distinction is made between single and multi-step schemes, and between explicit and implicit schemes. Explicit methods calculate the state of a system at a later time from the current state, whereas implicit methods find a solution by solving an equation involving both the current and later states of the system. For example, Eq.(1) can be discretized using the simplest explicit and implicit methods, which are the forward and backward Euler methods, in the following forms:

$$u(t+\Delta t) = u(t) + \Delta t\ F(u(t)) \qquad \text{Forward Euler scheme} \tag{2}$$

$$u(t+\Delta t) = u(t) + \Delta t\ F\left(u(t+\Delta t)\right) \qquad \text{Backward Euler scheme} \tag{3}$$

The explicit method is simpler to program and takes less time to compute. However, it has a low range of stability. Therefore, a small enough step size is required to avoid divergence. On the other hand, the implicit method has high stability and converges when the appro-

priate parameters are set. They are used because many problems encountered in practice are stiff, requiring impractically small time steps to keep the error bounded. However, the main drawback of implicit time integrators is their high computational cost compared to explicit methods. This is due to the linear system resolution. Nevertheless, it is important to note that explicit integrators are more suitable for computing unsteady phenomena such as shocks and acoustic waves. One may cite the common time integration methods used in the literature which are Euler, Crank-Nicholson, Newmark, Runge-Kutta, Adams-Bashforth, Adams-Moulton, Backward differentiation formulas, Heun, etc... A review of several time integrators of different type can be found in [2]. The appropriate use of a numerical scheme for a given equation is evaluated using three criteria proposed in numerical analysis. The first criterion is the convergence which guarantees that the approximated solution is close to the exact solution. The second one is the order which quantifies the quality of the approximation. The last one is stability, which shows the behavior of the error. In many cases, when solving evolving problems, these criteria or one of them are no longer respected. For example, the classical integration schemes might be unstable and the third criterion is no longer satisfied. Thus, the choice of their parameters is an essential and delicate issue. For explicit schemes, beyond the critical time step related to the Courant-Friedrichs-Lewy (CFL) condition [3], solving the problem by classical discretization techniques becomes imprecise. This leads to a destruction in the invariance properties of the physical phenomena, thus the conservation of energy and momentum is no longer respected. With a time step smaller than the critical one, it may be possible to respect the physical phenomenon, but on the other hand, the number of iterations and the computation time may increase.
Another example to highlight the limitations of classical time integration schemes is the numerical resolution of coupled problems, such as those of Fluid-Structure Interaction (FSI). In these cases, both the structure and fluid domains require the use of a distinct critical time step [4]. It is then necessary to consider the smaller of these. Furthermore, depending on the numerical techniques for tracking the free surface, managing the break-up of the boundary, or activating two-phase zones in the fluid domain, many critical time steps must be considered before running a simulation.

It is admitted that it is essential to use numerical tools that are both robust in terms of numerical accuracy and efficient in terms of computation time. Thus, the present study is dedicated to developing alternatives schemes to these methods by presenting numerical schemes that and are not a priori subject to a restrictive stability condition. The latter may have the potential to overcome the previously listed constraints of classical time integration schemes and are able to make the calculations more efficient with a reduced computation time. These chosen alternative methods are based on time pertur-

bation approach. We note that several attempts based on perturbation theory and series development of the solution exist in the literature. Perturbation methods have been used to solve nonlinear equations for many years. However, the studies based on time perturbation theory to overcome the critical time step and thus to obtain continuous solutions in time, are not very frequent. Some researchers who have performed some problems using the temporal perturbation are cited. Hamada and Aboanber [5], [6] developed the Power Series Solution denoted in their work by "PWS" to obtain approximate transient solutions to point kinetics equations with feedback. Lozi et al. [7], [8] described the method of construction of approximate chaotic solutions to dynamical model equations with quadratic nonlinearities, using an accurate numerical method based on time series. Fafard et al. [9], Berrahma-Chekroun et al. [10], [11] applied a new approach called the Dynamic Asymptotic method ("DAM") for the solution of transient dynamic problems. Braikat et al. [12] performed some tests also on transient nonlinear mechanical problems. Fellah and Fafard [13] also applied an approach based on time expansion for linear structural dynamic problems.

The perturbation theory used in the last five cited works is related to a developing area of research that deals with nonlinear problems called the Asymptotic Numerical Method and is denoted by "ANM". Thus, the application of ANM to temporal problems was initiated by the last five researchers mentioned previously, and then applied by D. Razafindralandy [14]. ANM has just been used as a temporal perturbation method in [9]–[14]. Note that the latter has been extensively studied and successfully applied, as high order perturbation method during the past 30 years to many problems as solid mechanics in [15]–[17], as well as fluid mechanics in [18]–[22]. It is used, in particular, for bifurcation analysis both in the field of structural design and fluid mechanics.

In this work, ANM based on the time perturbation theory is used. Since the objective is to solve evolution problems and develop alternative methods to time integration schemes, it is very reasonable to take the time as the perturbation parameter for ANM. The main idea is to represent the unknowns of the problem given in Eq.(1), in the form of a polynomial by using a perturbation parameter denoted by $\hat{t}$:

$$u(\hat{t}) = u_0 + u_1\hat{t} + u_2\hat{t}^2 + \cdots + u_N\hat{t}^N + \ldots \tag{4}$$

when N denotes a truncation order. Then, Eq.(4) is injected into the initial problem given by Eq.(1). When the operator F is a differential polynomial in u, a recurrence relation of

the following form is obtained:

$$u_{k+1} = F_{k+1}(u_0, \ldots, u_k), \quad k \in \mathbb{N} \tag{5}$$

where F_{k+1} is another (differential) polynomial operator in u. Therefore, an analytical approximation of the solution is then continuous and differentiable on a temporal validity domain to be determined.
The most interesting property of this time perturbation approach is that the computed solutions obtained are continuous in time. However, one must be very careful when manipulating time series, since there exist some series in physics that are divergent [23]–[26]. In such a case, the temporal validity domain of ANM becomes extremely small which leads to difficulties to advance the solution and reach the final time. For instance, the time series solutions of the heat equation, the Burgers equation, and the Navier-Stokes equations are generally divergent [23]–[26]. This does not imply that we should neglect the divergent series since the latter can lead to very satisfactory results if they are properly handled. For this reason, several methods of summing divergent series exist in the literature. One may cite: Borel summation, Ramanujan's irregular summation, Weierstrass summation, Abel summation, Lindelöf summation, etc... [27]. The most popular one is the Borel-Laplace summation. Towards the end of the 19th century, E. Borel [28] extended the interesting role of divergent series in analysis by calculating the "sum" of a divergent series. It converts the coefficients of a divergent model into a table of approximate values that approach the Borel sum of a divergent series. This method extended from the Cezaro sum is particularly used by mathematicians to compute the sum of the divergent series to study the solutions of differential equations within the theoretical framework. In 1991, it has been proved that the Borel summation (or Borel-Laplace summation) is the most suitable numerically [29]. By using the well-known Padé approximant [30], [31] combined with the latter, Borel-Padé-Laplace has become the most popular approach for summing divergent series, owing to its algorithmic simplicity.

For this reason, recently Hamdouni and Razafindralandy [32] have extended this algorithm numerically on applications to real time-dependent problems in mechanics, limited nevertheless to problems of small size in an exploratory framework. The proposed numerical algorithm based on the time series perturbation ANM, followed by the Borel summation technique of divergent series, is considered as a time integration scheme. It is named Borel-Padé-Laplace and is denoted by "BPL".

We are interested in this new class of temporal integrators (BPL) since it is promising to make the numerical simulations more efficient in terms of capture of the fast dynamic phenomenon and numerical stability over a large number of cycles and computation time.

It has been successfully applied for the resolution of differential equations as well as some partial differential equations [14], [32]–[36]. In addition, numerical results have shown that the BPL algorithm reduces the number of iterations compared to the classical schemes, and is able to respect the properties of the Hamiltonian systems [36].
However, many disadvantages of BPL need to be alleviated. One may cite the appearance of some numerical poles that can arise with the use of the Padé approximant, which sometimes block the continuation technique. To avoid these problems, an orientation towards another technique of resummation of divergent series and considered as a time integration scheme has been proposed in [37]. It is the Inverse factorial series denoted by "IFS" [37]. A first study on the effectiveness of the Inverse factorial series scheme, used as temporal integrator has been initiated just on some academic examples in [33], [37].

Finally, Mera et al. [38] proposed in 2018 a new method for summing divergent series, called "Meijer-G approximant" and denoted by "MG". The resulting Meijer-G approximations are easily parameterized using an hypergeometric ansatz. They applied the latter to different examples in quantum mechanics and quantum field theory [38]. For these examples, results have shown that Meijer-G is more powerful than Borel summation. These approximants are very well suited for problems where the Borel plane has a branch cut as mentioned in [38]. The exploitation of the possibility to use this method for time-dependent problems will be initiated in this manuscript.

Goals

Since the development of numerical methods for the resolution of nonlinear evolution problems is currently a growing field of research, the main objective of this thesis is to address several needs regarding the development of the recent time perturbation and numerical resummation techniques used as time integration schemes.

- The first goal is to understand more the behavior, develop the feasibility of such integration as well as consolidate the results obtained previously. The properties and qualities of the Inverse factorial series used as a time integration scheme should be studied in detail, since the latter has just been applied as a first prospective study on some simple examples. The potential of the temporal perturbation approaches compared to the classical schemes should be highlighted more on ordinary differential equations, by providing examples where classical schemes suffer from many weaknesses. The stiff or highly non-linear problems should be studied by time perturbation methods.

- The second goal is to advance their use on more complex models represented by PDEs and finally test their relevance and their robustness. The aim is to see how these alternative numerical methods behave for models involving a high number of degrees of freedom. Therefore, we must determine whether they are simultaneously efficient over a large interval time and with an acceptable computation cost. Hence, the efficiency of the temporal methods is not the only objective. The comparison of CPU times between classical and time perturbation methods is an important point to be discussed.

- The third goal of this thesis is to explore the potential of using the recent proposed Meijer-G for summing divergent series, in the context of time perturbation method. The numerical difficulties of the method, as well as its capacity, for transient problems should be presented. The development of certain aspects necessary for the adaptation of MG to the time-dependent problem should be introduced.

- The final goal of this thesis is to compare these different time perturbation and resummation methods with each other and be able to conclude the capabilities and weaknesses of each of them, in terms of efficiency and computation time, for different test cases.

To accomplish the goals listed above, the manuscript is organized around three parts in separate four chapters.

Organization of the manuscript

This document is divided into three main parts:

- **Part I**:
  1. Preliminary study of some theoretical foundations of the time perturbation and summation methods and their numerical algorithms, used as time integration schemes, that exist in the literature.
  2. Presentation of some developments and propositions that we have established for BPL and IFS.
  3. Exploitation of the use of the Meijer-G approximant for temporal test cases, by developing several points required for this adaptation.
- **Part II**:
  Application of time perturbation and resummation methods to ordinary differential

equations (ODEs).

- **Part III**:
  Application of time perturbation and resummation methods to partial differential equations (PDEs).

In Chapter 1, first, we introduce some necessary theoretical notions on time perturbation and resummation techniques of divergent series. Then, we describe the numerical algorithm of these methods used as time integration schemes, as well as some developments that we have established. Finally, we introduce the recent "Meijer-G approximant" divergent series summation method as an algorithm for solving time-dependent problems, by developing many points necessary for this exploitation. These methods are applied to examples of nonlinear ODEs in Chapter 2 to highlight their advantages of use over classical schemes in some cases. The following examples are handled: Euler equation, simple harmonic oscillator, free and forced Van der Pol oscillators, Elastic hardening spring problem, combustion equation, Lorenz system, nonlinear single pendulum problem,...
The interesting properties and difficulties of each time perturbation approach will also be presented.

To be able to apply, for future works, time perturbation methods to simulate Fluid-Structure Interaction (FSI) problems where equation governing the fluid motion (Navier-Stokes equation) are coupled with the equations governing the evolution of the free surfaces (for example level set equation), the new temporal approaches are applied to solve the Navier-Stokes equations in Chapter 3, and the popular level set evolution equations in Chapter 4. The progression difficulties will be done in stages.
First of all in Chapter 3, we start to study the efficiency of the methods in terms of feasibility and computation time on the heat equation, the Burgers equation and finally on the incompressible Navier-Stokes equations. We compare the results obtained with those of the literature. For instance, an application to the Lid-Driven cavity problem was chosen in order to discuss the behavior and the efficiency of these solvers, as it is a common benchmark problem in fluid dynamics. In this manuscript, we present the spatial discretization of the numerical resolution of PDEs by the finite element method and some finite difference schemes. Then, in Chapter 4, benchmarks level set problems, are studied by time perturbation and resummation methods. A discussion of a popular technique for level set called the reinitialization procedure, in the context of continuous transient solutions, is presented. Finally, the conclusion will briefly recall the most important results obtained and propose some orientations for future developments.

Chapter 1

Time series development and resummation of divergent series

This chapter consists of the description of the time perturbation methods and resummation of divergent series that we will use, develop and apply in this manuscript. First, a reminder of some theoretical foundations of the time perturbation and divergent series resummation techniques is presented. Then, the numerical algorithm of these methods used as time integration schemes is also described. ANM with time series development, Borel-Laplace summation, Inverse factorial series, and Meijer-G approximants are presented respectively in subsections 1.1, 1.2, 1.3 and 1.4. It is recalled that as the Meijer-G approximant is used in this manuscript for the first time as a time integration scheme, we will present the necessary adaptation of this transformation in this chapter, before discussing the numerical application. A new version of Meijer-G applied for a series of vectors will be also proposed.

1.1 Asymptotic Numerical Method (ANM)

In the framework of this thesis, we are interested to study ANM using the time as its perturbation parameter to solve evolution problems.

1.1.1 Numerical algorithm ANM used as a time integration scheme

Consider the following ordinary differential problem:

$$M\ddot{u} + C\dot{u} + Ku + \mathcal{N}_l(u) = 0 \tag{1.1}$$

where $\dot{u} = \dfrac{du}{dt}$ and $\ddot{u} = \dfrac{d^2u}{dt^2}$, with the following initials conditions at a given time t_0

$$\begin{cases} u(t_0) &= u_0 \\ \dot{u}(t_0) &= u_1 \end{cases} \tag{1.2}$$

where $\mathcal{N}_l(u)$ denotes a nonlinear term, M, C and K are prescribed scalars. In the case of multiple degrees of freedom, those quantities are matrices and u, $\dot{u}$, $\ddot{u}$ are vectors.
For simplicity, we give the following expression to the nonlinear term $\mathcal{N}_l$ that is classical in a solid mechanics framework:

$$\mathcal{N}_l(u) = u^3 \tag{1.3}$$

Perturbation: Unknowns of the problem (and time-dependant data) are sought as truncated power series. In this work, time is the perturbation parameter. Global time t is then defined using a given initial time t_0 and a perturbation time denoted by $\hat{t}$ as

$$t = t_0 + \hat{t} \tag{1.4}$$

Continuous approximation writes as follow:

$$u(\hat{t}) = u_0 + \sum_{n=1}^{N} u_n \hat{t}^n \tag{1.5}$$

with a known $u_0 = u(t_0)$. $\forall n = \{1, \ldots, N\}, u_n$ are called the series terms and N denotes the truncation order. It is then possible to differentiate this expression as

$$\dot{u}(\hat{t}) = \sum_{n=1}^{N} n u_n \hat{t}^{n-1} \tag{1.6}$$

$$\ddot{u}(\hat{t}) = \sum_{n=2}^{N} n(n-1) u_n \hat{t}^{n-2} \tag{1.7}$$

When we replace these expansions into Eq.(1.1), and identify according to the power of $\hat{t}$, we notice that it is not easy to find the recurrence formula that allows to calculate the series term. We note that to incorporate polynomial approximations, it is preferable to write all the nonlinear equations to be solved in a quadratic form by introducing new variables [39]. For this reason, let the following change of variable:

$$v = u^2 \tag{1.8}$$

Therefore Eq.(1.3) becomes:

$$\mathcal{N}_l(u, v) = uv \tag{1.9}$$

This yields to the new quadratic equation

$$M\ddot{u} + C\dot{u} + Ku + uv = 0 \tag{1.10}$$

Writing $v(\hat{t})$ as time power series

$$v(\hat{t}) = v_0 + \sum_{n=1}^{N} v_n \hat{t}^n \tag{1.11}$$

The developments of (Eq.(1.5), Eq.(1.6), Eq.(1.7) and Eq.(1.11)) are injected into Eq.(1.10) and by identifying the terms according to the powers of $\hat{t}$, a cascade of linear problems is then obtained:

$$\hat{t}^0 \quad : \quad \begin{cases} 2Mu_2 + Cu_1 + Ku_0 + u_0 v_0 = 0 \\ v_0 = u_0^2 \end{cases} \tag{1.12}$$

$$\hat{t}^1 \quad : \quad \begin{cases} 6Mu_3 + 2Cu_2 + Ku_1 + u_1 v_0 + u_0 v_1 = 0 \\ v_1 = 2u_1 u_0 \end{cases} \tag{1.13}$$

$$\hat{t}^2 \quad : \quad \begin{cases} 12Mu_4 + 3Cu_3 + Ku_2 + u_2 v_0 + u_0 v_2 + u_1 v_1 = 0 \\ v_2 = 2u_2 u_0 + u_1^2 \end{cases} \tag{1.14}$$

$$\vdots$$

$$\hat{t}^{N-2} \quad : \quad \begin{cases} MN(N-1)u_N + C(N-1)u_{N-1} + Ku_{N-2} + \sum_{i=0}^{N-2} (N-1-i)u_{N-1-i} v_i = 0 \\ v_{N-2} = \sum_{j=0}^{N-2} u_j u_{N-2-j} \end{cases} \tag{1.15}$$

Series terms $u_n, \forall n = 1, \cdots, N$ are obtained by the numerical resolutions of those systems. Once these series terms are calculated, the approximate solution Eq.(1.5) is evaluated. The latter is valid, not at discrete time steps, but in a whole interval, corresponding to the numerical radius of convergence of the series.

Radius of convergence R: It represents the limit between convergence and divergence of the series and can be estimated with different techniques. The quality of the solutions improves with the truncation order N for $\hat{t} < R$ but degrades for $\hat{t} > R$. After some practice tests for different examples with $\hat{t}$ values close to R, it has been shown that it is more interesting to introduce the notion of the domain of validity (or called also a range of validity) for a truncated series.

Range or domain of validity: Solutions of truncated series are valid in the sense of a desired precision, only on a specific range or domain of validity $[0, t_{max}]$. By a numerical observation that two successive polynomial approximations move away briskly beyond a certain domain, Cochelin [40] proposed an explicit expression of t_{max} based on the idea that the difference between two solutions evaluated at two consecutive truncation orders is less than a small parameter δ called the tolerance or the precision/accuracy parameter

(see Fig.(1.1)):

$$\frac{||u_{\hat{t},N} - u_{\hat{t},N-1}||_2}{||u_{\hat{t},N}||_2} = \frac{\hat{t}^N ||u_N||_2}{||u_1\hat{t} + \cdots + u_N\hat{t}^N||_2} < \delta \tag{1.16}$$

where $u_{\hat{t},N} = \sum_{n=0}^{N} u_n \hat{t}^n$ (series truncated at order N), and $u_{\hat{t},N-1} = \sum_{n=0}^{N-1} u_n \hat{t}^n$ (series truncated at order $N-1$).

Approximating the denominator by $\hat{t}||u_1||_2$, we get the explicit expression of t_{max} proposed in [40]:

$$t_{max} = \left(\delta \frac{||u_1||_2}{||u_N||_2}\right)^{\frac{1}{N-1}} \tag{1.17}$$

where δ denotes a given tolerance and $\|.\|_2$ is the Euclidean norm.

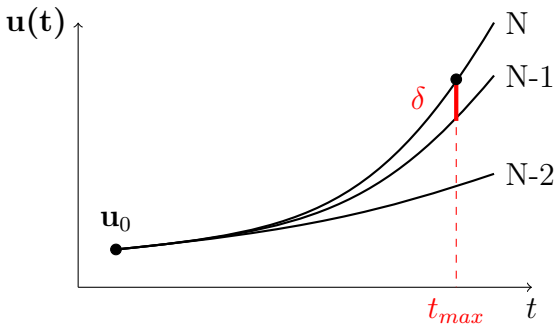


Figure 1.1 – Illustration of the upper bound of the domain of validity t_{max}.

Eq.(1.17) depends on two parameters, the truncation order N and the accuracy parameter or tolerance δ. The order N is usually chosen a priori to minimize the computation time. This criterion requires only an insignificant computation time for the algorithm resolution.

Note that another way to evaluate the domain of validity exists in [40], [41]. As solutions are continuous in time, they can be evaluated at any time $\hat{t} \in [0, t_{max}]$.

As can be seen in Fig.(1.2), we note that for this example the more terms added in the representation, the wider is the range of validity. $u(\hat{t})$ is not able to lead the exact solution for different truncation orders N, it may not be possible to reach $u(\hat{t})$ for a large value of $\hat{t}$ until a final time t. Therefore, a continuation procedure is proposed to reach a specific given time t.

Remark 1.1.1 *In all the figures of the manuscript, we note that the time is in seconds.*

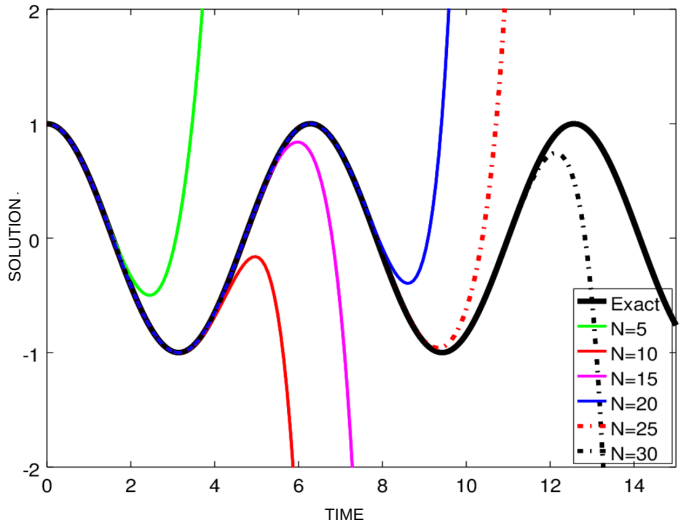


Figure 1.2 – Example of ANM series solutions using different truncation orders N compared to the exact solution. (Case of a differential equation representing the motion of a simple pendulum).

Continuation: To reach high values of t, a continuation procedure is performed, that is, the problem is solved again with the last valid solution. The continuation procedure is easy, it is sufficient to determine a new starting point u_0 after using the perturbation technique, and then re-apply the latter technique from this point. The new starting point, denoted by u_0^{j+1} on the piece^j of the solution, is calculated using its asymptotic series for the value of t_{max} at each calculation step (see Fig.(1.3)). It can be summarized as follows:

Algorithm 1 Continuation technique for ANM with time as perturbation parameter

Inputs:

1. The initial conditions u_0 and u_1.
2. The truncation order N.
3. The tolerance or the accuracy parameter δ.
4. The final time t_{end} or the domain in which we want to obtain the curve, for example: $[0, t_{end}]$.

Step 1: Compute series terms until a given order N.
Step 2: Evaluate the upper bound of the validity domain t_{max} using Eq.(1.17).
Step 3: Evaluate a new starting point u_0 at a new time $t_0 = t_0 + t_{max}$. The new point $u_0 = u(t_0 + t_{max})$ is denoted by u_0^{j+1} on the piece^j of the solution at time t_{j+1}.
Step 4: Go to Step 1 and repeat the algorithm until reaching the final time t_{end}.

The number of times to a continuation iteration will be called the number of continuation steps. This time series development allows obtaining a numerical approximation of the solutions continuous in time, which is different from a discrete approach. The performance of the algorithm is therefore based on the domain of validity. If the latter is too

small, a high number of continuous steps becomes necessary.

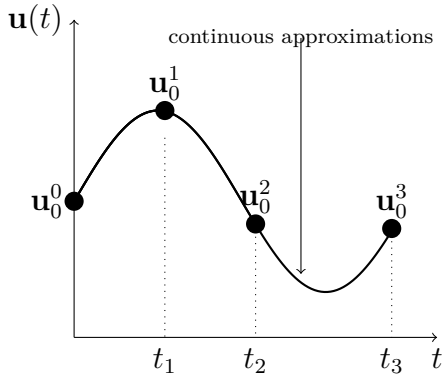


Figure 1.3 – Illustration of the continuation technique.

1.1.2 Time series and Padé approximants

It is possible to enlarge the domain of validity of the series: $[0, t_{max}]$ by replacing the polynomial with a rational fraction (Padé approximant in this work) which is asymptotically equivalent to it. This procedure has proven to be efficient in many cases [42]–[45]. The polynomial approximation can be replaced by a rational one, using Padé approximants.

The latter are very frequently used in computer sciences and are well known for their efficiency even to extend a series out of its convergence disk. This new representation is written in the following form:

$$u_{Padé,N}(\hat{t}) = u_0 + \sum_{i=1}^{N-1} \frac{R_{N-1-i}(\hat{t})}{Q_{N-1}(\hat{t})}\, \hat{t}^i u_i \tag{1.18}$$

where R_i and Q_i are polynomials of degree i. For more details about the popular Padé, readers are refered to [30], [31].

According to several test calculations carried out, rational approximations for different truncation orders give very close results when the parameter $\hat{t}$ is within a certain domain, but beyond this domain, they diverge strongly. Thus, the simple way to define the range of validity of this rational representation is to extend the criterion used in Eq.(1.17). In this case, it is sufficient to require that the divergence between two rational solutions at two consecutive orders remains small at the end of the step. Therefore, the range of validity of the rational representation is sought by dichotomy via a tolerance criterion δ_P, as in [44]:

$$\delta_P = \frac{\left\| u_{Padé,N}(\hat{t}) - u_{Padé,N-1}(\hat{t}) \right\|_2}{\left\| u_{Padé,N}(\hat{t}) \right\|_2} \tag{1.19}$$

with $\hat{t}$ in $[0, t_{max}]$. The range of validity of the rational representation is usually larger than the polynomial one (i.e t_{max} of Padé $> t_{max}$ of ANM). Consequently, fewer steps of continuation are necessary.

With Eq.(1.19), t_{max} of Padé approximants is computed and a new starting point for the asymptotic expansions Eq.(1.5) can be defined by introducing the value of t_{max} in the expression Eq.(1.18). Thus, a continuation method is defined.

Remark 1.1.2 *If one needs to approximate numerically the derivatives of the Padé solution (i.e the derivative of Eq.(1.18)), one can proceed as follows:*
First, consider for example a truncation order $N = 5$, therefore we have:

$$u(t) = \sum_{n=0}^{N} u_n t^n = u_0 + u_1 t + u_2 t^2 + u_3 t^3 + u_4 t^4 + u_5 t^5 \tag{1.20}$$

Therefore the derivative of $u(t)$ is

$$\dot{u}(t) = \sum_{n=1}^{N} n u_n t^{n-1} = u_1 + 2u_2 t + 3u_3 t^2 + 4u_4 t^3 + 5u_5 t^4 \tag{1.21}$$

On the other hand, $\dot{u}(t)$ can be written in the following form:

$$\dot{u}(t) = \dot{u}_0 + \dot{u}_1 t + \dot{u}_2 t^2 + \dot{u}_3 t^3 + \dot{u}_4 t^4 \tag{1.22}$$

where

$$\dot{u}_0 = u_1, \quad \dot{u}_1 = 2u_2, \quad \dot{u}_2 = 3u_3, \quad \dot{u}_3 = 4u_4 \quad \dot{u}_4 = 5u_5 \tag{1.23}$$

To calculate the derivatives of the Padé solution (i.e the derivative of Eq.(1.18)), one can evaluate the Padé solution of Eq.(1.22) with the new series terms $\dot{u}_0, \dot{u}_1, \dot{u}_2, \dot{u}_3, \dot{u}_4$.
In the same way, one can evaluate the second derivative of the Padé solution of $u(t)$ by writing $\ddot{u}(t)$ in the same way as Eq.(1.22).

Remark 1.1.3 *We will not present the ANM with the time series followed by the Padé approximants in all of the examples in this document because we have found numerical instability in the majority of ODEs cases when using the ANM+Padé approximant for some temporal large problem. Note that the concept of Padé approximant has been defined here for its later use in Borel-Padé Laplace subsection 1.2.2.*

1.1.3 Divergent series

The drawback of the ANM approach using the time as its perturbation parameter is that the time series might be divergent. In such a case, the range of validity of this

representation in the temporal domain becomes extremely small. Therefore, classical continuation is no longer efficient. An example of divergent time series will be presented [32], [33]. The behavior of ANM for different truncation orders N, without any continuation procedure is presented in Fig.(1.4). It is observed that when the truncation order N increases, the computed solutions tend to diverge earlier, contrary to what has been observed in Fig.(1.2).

Example 1.1.4 *Consider the Euler equation*

$$t^2\dot{u} + u = t \tag{1.24}$$

where $\dot{u} = \dfrac{du}{dt}$, *with the following initial condition*

$$u(t_0 = 0) = 0 \tag{1.25}$$

The formal power series solution of this equation is

$$u(t) = \sum_{k\geq 0}(-1)^k k! t^{k+1} \tag{1.26}$$

This series is divergent for any $t > 0$. *ANM would not be able to provide an analytical solution.*

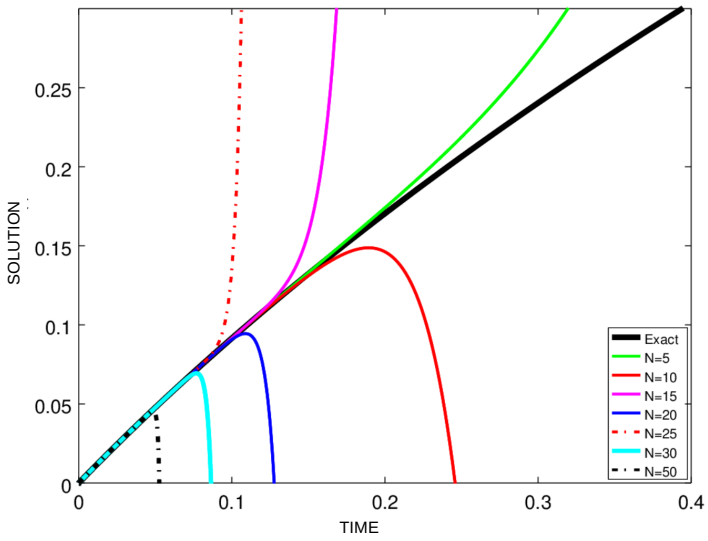


Figure 1.4 – ANM series solutions for Euler equation using different truncation order N (without continuation), compared to the exact solution.

In the absence of theoretical information on the type of time series, we cannot be certain whether the infinite numerical series is convergent or divergent because the solution $u(t)$ cannot be obtained numerically when N tends towards infinity for any computer

memory.

How can we study divergent series ?
As there are physical equations that lead to divergent asymptotic series, these cases should not be ignored. In the study of equations, whether they are ordinary differential equations (ODEs) or partial differential equations (PDEs), the aim is to recover information as much as possible regarding the solution, by summing up the divergent asymptotic series. To obtain an accurate solution, one must be interested in the summation methods of divergent series that give the highest information about the associated function, even if the latter is truncated. Therefore, ANM is unusable when we have a very small or zero radius of convergence, hence the need to apply the summation techniques of divergent series. As mentioned in the introduction, one of the most popular summation methods is the Borel-Laplace summation that will be exposed in the next section.

1.2 Borel-Laplace (BL)

First, a theoretical necessary reminder of the summation of Borel-Laplace (or simply Borel summation) denoted by BL is presented. The reader is referred to [26], [28], [46], [47] for more details concerning the definition of Borel summation method. Then the numerical algorithm of Borel-Padé-Laplace (BPL) [32] is exposed. We start with some useful necessary definitions and theorems.

1.2.1 Theoretical reminder

First, we note that Watson [48] classified all divergent series according to a so-called Gevrey index, after noticing that the general term of the most frequently identified divergent series always behaved as $CA^n(n!)^{1/k}$ for certain constants C, A, and $k > 0$ when n is large enough, which yields to the following definition:

Definition 1.2.1 *A formal power series* $u(t) = \sum_{n=0}^{\infty} u_n t^n$ *is a Gevrey series of index or order* k*, if there exists two constants* $C > 0$ *and* $A > 0$ *such that*

$$|u_n| \leq CA^n(n!)^{1/k}, \quad \forall n \in \mathbb{N} \tag{1.27}$$

In particular, the Gevrey series of order 0 is the convergent series. This definition will be an interesting condition for the Borel transform, that we will introduce in the following, in order to have a non-zero radius of convergence. Note that, it is well-known that most of the series arising in engineering problems are k-Gevrey series for some positive rational

number k. In what follows, we will present the Borel-Laplace resummation only for the case of $k = 1$.

Note that Maillet [49] has proven that formal series that are solutions of an algebraic ODE are Gevrey series. Malgrange [50] has extended this observation to the analytic nonlinear ODEs. The theory for PDEs is not quite complete. Some equations in fluid mechanics have been investigated, such as the heat equation and its generalizations and the Burgers equation [25], [26], [51]–[53]. For the Navier-Stokes equations, summability conditions to the formal power series are provided recently by Costin et al. [54].

Remark 1.2.2 *Note that in the case of a formal series solution of a linear equation with polynomial coefficients, the Gevrey index k can be determined from the Newton polygon of the equation. We will not present the details for this remark since most of the cases that will be considered in the following chapters are nonlinear. The interested reader is invited to consult the following references [55], [56].*

We recall that we will present the definitions required for Borel-Laplace resummation only for the case of the Gevrey series of order $k = 1$.

The approximate obtained solution will be an analytical function in a sector and has the series $u(t) = \sum_{n=0}^{\infty} u_n t^n$ as the Gevrey asymptotic, as defined below:

Definition 1.2.3 *An analytic function $u(t)$ in an open sector S, is Gevrey-asymptotic of order I to a series $u(t) = \sum_{n\geq 0} u_n t^n$ at the origin in S, if for any compact subsector $T \subset S$, there exists two constants C and A such that*

$$\forall n \in \mathbb{N}, \forall t \in T, \qquad \left| u(t) - \sum_{k=0}^{N-1} u_k t^k \right| \leq C A^n |t|^n (n!)^I \tag{1.28}$$

Definition 1.2.4 *The Borel transform of the time power series $u(t) = \sum_{n=0}^{\infty} u_n t^n$ is*

$$\mathbb{B}u(\xi) = \sum_{n=0}^{\infty} \frac{u_{n+1}}{n!} \xi^n \tag{1.29}$$

The Borel transform $\mathbb{B}u(\xi)$ has a non-zero convergence radius if u is a Gevrey series analytical in the neighborhood of 0.

To incorporate the notion of Laplace transformation, one must first establish the concept of growth at most exponential to infinity.

Definition 1.2.5 *Let $P(\xi)$ be an analytic function in a domain containing a semi-line starting from the origin denoted by d. We say that $P(\xi)$ has an exponential decay at the infinity in the direction d if there exists two constants A and C such that*

$$|P(\xi)| \leq Ae^{C|\xi|} \tag{1.30}$$

as $|\xi|$ tends to infinity along d.

We can now establish the definition of the Laplace transform.

Theorem 1.2.6 *Let P be an analytic function in an infinite sector S containing the direction d. Assume that $P(\xi)$ has an exponential decay at the infinity in the direction d, $\forall \xi \in S$, with ξ large enough (Definition 1.2.5). Then the Laplace transform of P denoted by $\mathcal{L}P(t)$ in the d direction, is the following function*

$$\mathcal{L}P(t) = \int_d P(\xi)e^{-\xi/t}d\xi. \tag{1.31}$$

Definition 1.2.7 *A power series $u(t)$ is summable (or 1-summable) in a direction d if*

- *$u(t)$ is a Gevrey series of index 1. (Definition 1.2.1)*
- *$\mathbb{B}u(\xi)$ can be analytically prolonged into a function $P(\xi)$ in a domain containing d.*
- *$P(\xi)$ has an exponential decay at the infinity in the direction d. (Definition 1.2.5)*

Therefore, the Borel sum in a direction d denoted by $S_d u(t)$ is

$$\begin{aligned} S_d u(t) &= u_0 + \mathcal{L}P(t) \\ &= u_0 + \int_d P(\xi)e^{-\xi/t}d\xi. \end{aligned} \tag{1.32}$$

Remark 1.2.8 *The Laplace transform can be considered as the inverse of the Borel transform when it is applied formally to a power series.*

After the required definitions and theorems, we can establish the following three steps of Borel-Laplace to calculate the Borel sum $S_d u(t)$ in a direction d:

(i) Compute the Borel transform $\mathbb{B}u(\xi)$.

(ii) Prolongate naturally $\mathbb{B}u(\xi)$ into an analytical function $P(\xi)$.

(iii) Compute the Laplace transform of P along a direction d which permits to go back to the originally space.

These three steps constitute the Borel-Laplace summation method. It is simple to understand: the coefficients of a series with a zero radius of convergence typically grow factorially to high orders. This factorial growth is eliminated in the first step, and the

transformed Borel series is easy to handle because it has a non-zero finite radius of convergence. By summing the Borel-transformed series one finds the function $P(\xi)$, and then the Laplace transform can be computed to find finally the Borel sum.

Remark 1.2.9 *When $d = \mathbb{R}^+$, we will note the Borel sum $S_d u(t)$ by $Su(t)$.*

Remark 1.2.10 *The generalization of all the above definitions for Gevrey series of index k is possible. For more information about the resummation of Borel-Laplace for k-Gevrey series, the reader is invited to consult the following references [46], [47].*

1.2.2 Numerical algorithm BPL used as a time integration scheme

The previous three steps are translated into the following numerical algorithm BPL. It is organized around the following three stages: Borel transform ($\mathbb{B}$) + Approximation of Padé (P) + Laplace transform ($\mathcal{L}$) (BPL). We present this numerical algorithm for Gevrey series of index $k = 1$ in a direction $d = \mathbb{R}^+$. Since the singular directions of the Borel transform are not known a priori, therefore this direction is set as the real positive axis in this work as in [32].

- **Fist step: Borel transform**
  Numerically, for the first stage, series terms are computed up to an order N

$$u(\hat{t}) = \sum_{n=0}^{N} u_n \hat{t}^n \tag{1.33}$$

  where $\hat{t}$ is the perturbation time.
  This yields that numerically Borel transform $\mathbb{B}(\xi)$ is of order $N-1$. It is given by:

$$\mathbb{B}u(\xi) = \sum_{n=0}^{N-1} \frac{u_{n+1}}{n!} \xi^n \tag{1.34}$$

- **Second step: Prolongation via Padé approximant**
  For the second stage, an extension of the Borel transform is obtained by establishing an approximation of Padé:

$$P(\xi) = \frac{A_0 + A_1\xi + A_2\xi^2 + \cdots + A_L\xi^L}{1 + B_1\xi + \cdots + B_M\xi^M} \tag{1.35}$$

  where L and M are two integers such that $L + M = N - 1$ (because the Borel transform is of order $N-1$), $A_0, A_1, \ldots, A_L$ and $B_1, \ldots, B_M$ are the coefficients of Padé approximants. In the cases of large size problems (for example when solving a PDE), it is recommended to use a vector version of Padé [57]. Note that prolongation of Borel transform can be done with different methods other than Padé.

The interested reader is referred to [58] for furthers details, but these methods are designed for linear equations.

- **Third step: Laplace transform**
  For the last stage, the Laplace transform is applied to the Padé approximants to return to the temporal space, with a Gauss-Laguerre quadrature [59]. Note that the Gauss-Laguerre method has been chosen for the evaluation of the Laplace transform because it is one of the most well-known quadrature methods applied to semi-infinite integrals. But definitely, other methods can be considered [60]–[62]. The Borel sum is

$$Su(\hat{t}) = u_0 + \int_0^\infty P(\xi) e^{-\xi/\hat{t}} d\xi \tag{1.36}$$

  then applying Gauss-Laguerre quadrature to approximate the integral contained in the Borel sum (the Laplace transform).
  Before elaborating how to process the Borel sum numerically, a brief review of Gauss-Laguerre quadrature is presented.

Recall 1.2.11 *In numerical analysis, Gauss–Laguerre quadrature is an extension of the Gaussian quadrature method for approximating the value of integrals of the following kind:*

$$\int_0^\infty f(x) e^{-x} dx \tag{1.37}$$

by the following sum

$$\sum_{i=1}^{N_G} w_i f(x_i) \tag{1.38}$$

where x_i points (or nodes) are real, distinct, unique and are the roots of N_G Laguerre polynomials denoted by $L_n(x)$ (N_G number of Gauss points), and the weight w_i is given by

$$w_i = \frac{x_i}{(n+1)^2 [L_{n+1}(x_i^2)]} \tag{1.39}$$

Using the following relation:

$$P(\xi) e^{-\xi/\hat{t}} = \left(P(\xi) e^{-\xi/\hat{t}} e^{\xi} \right) e^{-\xi} \tag{1.40}$$

then applying the Gauss-Laguerre quadrature to Eq.(1.40) yield to

$$\int_0^\infty P(\xi) e^{-\xi/\hat{t}} d\xi \approx \sum_{i=1}^{N_G} P(\xi_i) e^{-\xi(\frac{1}{\hat{t}}-1)} w_i \tag{1.41}$$

where ξ_i are the roots of the N_G-th Gauss-Laguerre polynomial and w_i are the

weights.

Eq.(1.41) is not valid for $\hat{t}$ too small. For this reason, consider the following change of variable $\xi = \xi/\hat{t}$. This yields to

$$\int_0^\infty P(\xi)e^{-\xi/\hat{t}}d\xi \approx \hat{t}\sum_{i=1}^{N_G} P(\xi_i\hat{t})w_i \tag{1.42}$$

By replacing Eq.(1.42) in Eq.(1.36), one can obtain finally the Borel sum given by

$$Su(\hat{t}) = u_0 + \hat{t}\sum_{i=1}^{N_G} P(\xi_i\hat{t})w_i \tag{1.43}$$

Note that changing the order of this scheme is very simple. We just have to replace the value of N with another one.

This method can be considered as a method to extend the range of validity of the series when the latter is convergent and plays its task as a resummation method when the series diverges [32].

Since the initial series Eq.(1.33) is truncated up to an order N, and because of the numerical effects, numerical resummation methods are not able to lead the exact solution, i.e by applying the resummation only once, it may not be possible to reach $Su^N(\hat{t})$ for a large value of $\hat{t}$ until a final time t. Therefore, Borel sum Eq.(1.43) allows approximating analytically the exact solution only in a certain time interval. Moreover, the definition of the range of validity $[0, t_{max}]$ of BPL is still not evaluated explicitly like that of ANM (see Eq.(1.17)). Therefore, a continuation procedure is proposed to reach specific given time t. The continuation of BPL algorithm might be carried out using a criterion based on the residual since it is the most useful criterion in engineering science. The residual $Res(t)$ of the differential problem (Eq.(1.1)) is defined by

$$Res(t) = M\ddot{S}(t) + C\dot{S}(t) + KS(t) + \mathcal{N}_l(S(t)) \tag{1.44}$$

where

— $S(t) = Su(t)$ is the Borel sum presented in Eq.(1.43).

— $\dot{S}(t) = \dfrac{dSu(t)}{dt}$ is the first derivative of Borel sum.

— $\ddot{S}(t) = \dfrac{d^2Su(t)}{dt^2}$ is the second derivative of Borel sum.

replaced in the initial equation of the problem (Eq.(1.1)), M, C and K are defined before. Those latter should be determined in order to evaluate the residual. Here it should be mentioned that to calculate the derivatives there are (as for the Padé approximant) at

least two ways of doing it:

1. Calculate the derivatives of the series and establish an identification between the terms. (see remark 1.1.2).
2. Establish an analytical expression of the derivatives of the "Borel expression" which is an original contribution of this thesis. Those latters are established in Appendix A.

We will evaluate numerically some examples of the two ways of calculating the derivatives for high simulation times in the next chapter.
The analytical approximations are examined as long as

$$\|Res(t)\|_2 < \epsilon \tag{1.45}$$

where ϵ is a small positive parameter called the residual tolerance or the accuracy parameter and $\|.\|_2$ denotes the Euclidean norm. For the next step, we redefine a new starting point $u(t_1)$ where $t_1 > t_0$, which corresponds to the end of the step where relation Eq.(1.45) holds.
The continuation procedure of BPL is summarized by the algorithm below:

Algorithm 2 Continuation technique for BPL

Inputs:
1. The initial conditions u_0 and u_1.
2. The truncation order N.
3. The residual tolerance or the accuracy parameter ϵ.
4. The final time t_{end} or the domain in which we want to obtain the curve, for example: $[0, t_{end}]$.

Step 1: Compute series terms until a given order N.
Step 2: Apply the three steps of BPL (Borel transform, prolongation via Padé approximant, Laplace transform) to calculate the Borel sum $Su(t)$.
Step 3: Apply the first and second derivatives approximations of BPL to calculate $\dot{S}(t)$ and $\ddot{S}(t)$. (See Appendix A).
Step 4: Evaluate the residual $Res(t)$ given in Eq.(1.45).
Step 5: Evaluate the validity domain t_{max} such that t_{max} is the last value of t for which $\|Res(t)\|_2 < \epsilon$.
Step 6: Evaluate a new starting point u_0 at a new time $t_0 = t_0 + t_{max}$.
Step 7: Go to Step 1 and repeat the algorithm until reaching the final time t_{end}.

This approach with the residual criterion is quite a time expensive, but we will see in the numerical applications if it is fast enough to outperform other numerical schemes. An evaluation of the range of validity of BPL (t_{max} of BPL) without using the residual computation is preferable but is not yet available.

This type of solver is promising to make the numerical simulations more efficient in terms of capture of the fast dynamic phenomenon and numerical stability over a large number of cycles and computation time. This algorithm has been successfully applied on academic examples and some examples of Hamiltonian systems in [14], [32]–[36].

1.2.3 Strengths and weaknesses of BPL

Numerical results in [32], [33], [63] have shown that the BPL algorithm presents some interesting properties. To illustrate the BPL algorithm, we use one example that have already been discussed in [32], [33], [63].

Example 1.2.12 *Consider again the Euler equation*

$$t^2\dot{u} + u = t \tag{1.46}$$

where $\dot{u} = \dfrac{du}{dt}$, *with the following initial condition*

$$u(t_0 = 0) = 0 \tag{1.47}$$

The formal power series solution of this equation is

$$u(t) = \sum_{n\geq 0}(-1)^n n! t^{n+1} \tag{1.48}$$

which is a divergent series.
The Borel transform of Eq.(1.48) *is*

$$\mathbb{B}u(\xi) = \sum_{n=1}^{\infty}(-1)^n \xi^n \tag{1.49}$$

This series can be naturally extended into the analytic function

$$P(\xi) = \frac{1}{1+\xi} \tag{1.50}$$

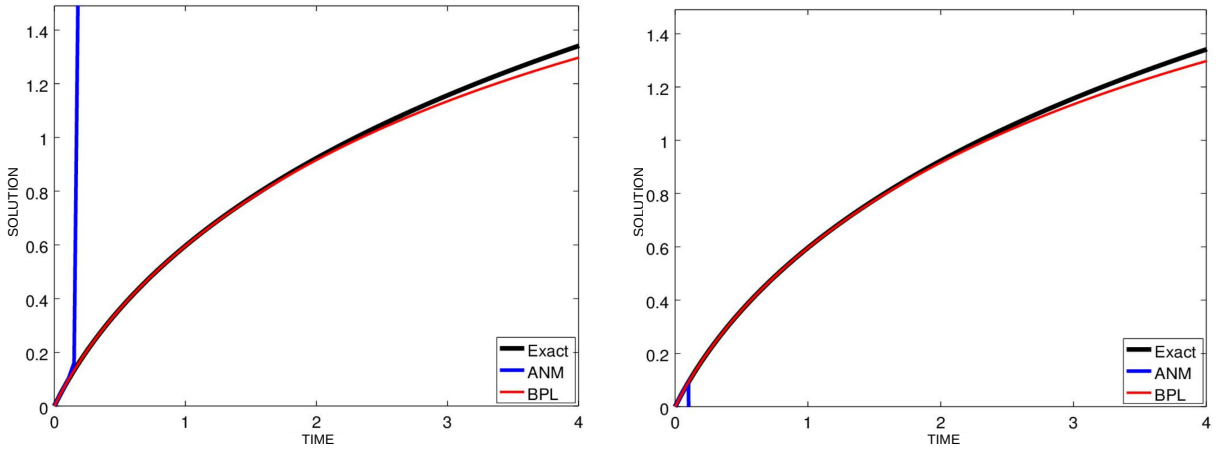
Therefore, the Laplace transform in the direction $d = \mathbb{R}^+$ *is*

$$\mathcal{L}P(t) = \int_0^{\infty} P(\xi)e^{-\xi/t}d\xi = \int_0^{\infty} \frac{e^{-\xi/t}}{1+\xi}d\xi \tag{1.51}$$

this yield to the Borel sum

$$Su(t) = u_0 + \int_0^{\infty} \frac{e^{-\xi/t}}{1+\xi}d\xi = \int_0^{\infty} \frac{e^{-\xi/t}}{1+\xi}d\xi. \tag{1.52}$$

Numerically, Fig.(1.5) shows a comparison between the reference solution, the solution approximated by ANM (without continuation, only one step), and by the BPL algorithm (without continuation). As the Euler series is divergent, it can be seen in Fig.(1.5) that the time power series alone (ANM) deviates very rapidly from the reference solution for two different truncation orders. Whereas the BPL algorithm gives a satisfactory solution up to a time much greater than ANM.



(a) N=15 (b) N=20

Figure 1.5 – Approximated solutions for Euler equation using ANM and BPL with two truncation orders $N = 15$ and $N = 20$ (without continuation).

Many advantages can be obtained by using BPL [14], [32]–[36]. One may cite:
(*i*) Ability to solve the problem over a large time interval.
(*ii*) BPL holds the properties of Hamiltonian systems (symplecticity, iso spectrality).
(*iii*) BPL needs less steps than the classical scheme to reach the same final time.

Padé approximants constitute simultaneously the strength as well as the weakness of the BPL algorithm presented. The latter makes it possible to identify the possible poles of the function to be approximated, which thus enables to choose the direction of the integration of the Laplace transformation. But when the Padé table is not normal, the linear problem resulting from the Padé algorithm becomes non-invertible. And if it is not the case, the linear system is ill-conditioned in general for many examples, which can lead to the appearance of spurious null pole pairs [64]. To avoid the problems induced by the Padé approximants, several techniques exist in the literature. For example, one can cite the use of the "Froissart doublet" [64]. One can also mention the use of the Padé approximant where the denominator (and the poles) are prescribed. Another technique that can be also listed is the use of conformal mapping to extend the Borel transform series, but the latter requires the knowledge of the nearest pole of $\mathbb{B}u$.

In [37], [63], the weaknesses of the BPL algorithm for the numerical computation of the Borel sum have been discussed through different examples introducing some cases where:

(*i*) There is no uniqueness of Padé approximant.

(*ii*) The Pole in Padé approximant on $\mathbb{R}^+$ might appear. (See an example in Fig.(1.6)).

(*iii*) There is some discretization errors due to the numerical integration of Gauss-Laguerre.

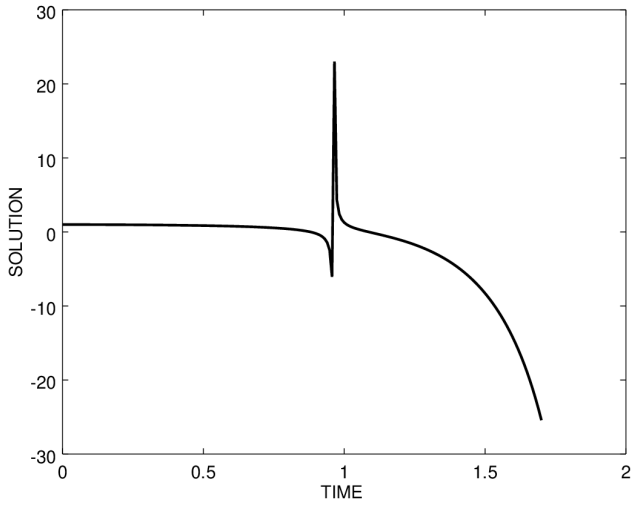


Figure 1.6 – Example of the existence of a pole in Padé approximant on $\mathbb{R}^+$ of the BPL solution.

To avoid these problems by Padé approximants, another numerical resummation method namely Inverse factorial series and denoted by "IFS" is proposed as a time integration scheme in [37] and is presented in the following subsection.

1.3 Inverse factorial series (IFS)

The Borel sum can be represented by a generalized factorial series which is computed directly from the terms of the series, without passing through the Borel space. The key point of this approach is that it avoids the three-step calculation and there is no need to evaluate $P(\xi)$.

The introduction of the inverse factorial series as an asymptotic of a power series was initiated at the beginning of the last century: [48], [65], [66]. More recently, the effective summation by the inverse factorial series has been numerically investigated in [29], [67], [68], and very recently used as a time integration scheme in [37], [63] for which the details will be presented in the following.

First, we introduce some useful definitions and theoretical results. Then, we present the numerical algorithm of IFS used as a time integration scheme.
Note that the link between Borel sum and Inverse factorial series is shown in [37] and will be recalled here since it is important to illustrate it.
First, we consider again equation Eq.(1.32):

$$Su(t) - u_0 = \int_0^\infty P(\xi) e^{-\xi/t} d\xi \tag{1.53}$$

Let $\xi = -\ln \mu$, therefore

$$\int_0^\infty P(\xi) e^{-\xi/t} d\xi = \int_0^1 P(-\ln \mu) \mu^{\frac{1}{t}-1} d\mu = \sum_{n\geq 0}^{\infty} b_n \int_0^1 (1-\mu)^n \mu^{\frac{1}{t}-1} d\mu \tag{1.54}$$

where b_n are the coefficients of the Taylor expansion of $P(-\ln \mu)$ at $\mu = 1$ given by:

$$b_n = \frac{1}{n!} \sum_{k=1}^{n+1} |\mathbb{S}(n, k-1)| u_k \tag{1.55}$$

where $\mathbb{S}(n, k)$ denotes the Stirling numbers of the first kind. Therefore, Eq.(1.53) becomes:

$$Su(t) - u_0 = \sum_{n\geq 0}^{\infty} b_n \int_0^1 (1-\mu)^n \mu^{\frac{1}{t}-1} d\mu \tag{1.56}$$

After integration by part, we get

$$Su(t) - u_0 = \sum_{n\geq 0}^{\infty} b_n \int_0^1 (1-\mu)^n \mu^{\frac{1}{t}-1} d\mu \tag{1.57}$$

$$= \sum_{n\geq 0}^{\infty} \frac{b_n n!}{\frac{1}{t}(\frac{1}{t}+1)\dots(\frac{1}{t}+n)} \tag{1.58}$$

$$= \sum_{n\geq 0}^{\infty} \frac{b_n n! t^{n+1}}{(1+t)\dots(1+nt)} \tag{1.59}$$

Finally, we have

$$Su(t) = u_0 + \sum_{n\geq 0}^{\infty} \frac{b_n n! t^{n+1}}{(1+t)\dots(1+nt)} \tag{1.60}$$

which represents the Inverse factorial series.
And in this way, we have recalled how to proceed from the Borel sum to the Inverse factorial series.

Definition 1.3.1 *We call Inverse factorial series $I(t)$ the following*

$$I(t) = u_0 + \sum_{n=0}^{\infty} \frac{b_n n! t^{n+1}}{(1+t)\dots(1+nt)} \tag{1.61}$$

with

$$b_n = \frac{1}{n!}\sum_{k=1}^{n+1} |\mathbb{S}_s(n,k-1)|u_k \tag{1.62}$$

where $\mathbb{S}_s(n,k)$ *denotes the Stirling numbers of the first kind.*

$I(t)$ is unique and absolutely convergent for sufficiently high values of the real part of $(1/t)$ [66], [69], [70] and converges towards $S\hat{u}(t)$.
Note that Eq.(1.61) is valid for the 1-summable series in a direction $d = \mathbb{R}^+$.

1.3.1 Numerical algorithm IFS used as a time integration scheme

Numerically, $I(t)$ is truncated up to an order N and is denoted by $I^N(\hat{t})$ with $\hat{t}$ the perturbation time:

$$I^N(\hat{t}) = u_0 + \sum_{n=0}^{N} \frac{b_n n! \hat{t}^{n+1}}{(1+\hat{t})\dots(1+n\hat{t})} \tag{1.63}$$

Theorem 1.3.2 *Let* $u(\hat{t})$ *be a 1-summable series,* A *and* C *two constants implicated by the exponential decay of the prolongation* $P(\xi)$ *of its Borel transformation and* $T = \dfrac{1}{C}$. *Therefore, the Inverse factorial series summation* $I(\hat{t})$ *of* $S\hat{u}(\hat{t})$ *satisfies*

$$|Su^N(\hat{t}) - I^N(\hat{t})| \leq C_N \left| \frac{\hat{t}^{N+1} N!}{(1+\hat{t})\cdots(1+N\hat{t})(Re\frac{1}{\hat{t}} - \frac{1}{T})} \right| \tag{1.64}$$

where $Su^N(\hat{t})$ *is the Borel sum truncated at an order* N, *and*

$$C_N = CT^{\frac{1}{T}} \frac{(N+1+\frac{1}{T})^{N+1+\frac{1}{T}}}{(N+1)^N} \tag{1.65}$$

Remark 1.3.3 *When the truncation order* N *is large, Eq.*(1.62) *can not be used because of the numerical explosion of the Stirling numbers* $\mathbb{S}(n,k)$ *[33].*

Hence, an equivalent method to evaluate $I^{N-1}(\hat{t})$, proposed in [69], is described:

Suppose that $u(\hat{t})$ is 1-summable in a direction d, with an angle θ with the positive half-axis.

Let $\tau_l = s_l e^{i\theta}$ with $l = (1, 2, \dots, N)$, where $\{s_1, s_2, \dots, s_N\}$ is any sequence of complex numbers. Let

$$z = \frac{1}{\hat{t}}, \qquad y = ze^{i\theta} \tag{1.66}$$

$$a_1 = \frac{u_0}{z}, a_2 = \frac{u_1}{z^2}, \dots, a_m = \frac{u_{m-1}}{z^m}, \quad m = 1, \dots, N \tag{1.67}$$

Note that if the series are summable in the $\mathbb{R}^+$ direction i.e $d = \mathbb{R}^+$, we have $y = z$ and we will choose $s_l = l$ for $l \geq 1$.

We call the Inverse factorial series the following

$$I^{N-1}(\hat{t}) = \frac{1}{\hat{t}} \sum_{n=0}^{N-1} v_{n+1} = z \sum_{n=0}^{N-1} v_{n+1} = z(v_1 + v_2 + \cdots + v_N) \tag{1.68}$$

where v_n is the nth term of IFS which are calculated using the following recursive algorithm based on

$$v_{n+1}^{(j)} = \frac{\tau_{n-1} v_n^{(j)} + y v_n^{(j+1)}}{y + \tau_n} \qquad n \geq 1, j \geq 1, \tag{1.69}$$

with

$$v_1^{(1)} = a_1 \qquad v_1^{(2)} = a_2 \qquad \ldots \qquad v_1^{(N)} = a_N \tag{1.70}$$

The terms $v_1^{(1)}, v_2^{(1)}, \ldots, v_n^{(1)}, v_{n+1}^{(1)}$ are the terms of the IFS denoted by: $v_1, v_2, \ldots, v_n, v_{n+1}$.

Remark 1.3.4 *Note that the evaluation of the IFS sum with this approach should start from $t_0 \neq 0$ (because of Eq.(1.66)). For this reason, for all the examples that we will present in this manuscript, we will fix numerically the initial time t_0 to a very small value close to 0.*

The IFS algorithm can overcome the problems encountered by BPL, therefore it is a good alternative to BPL. Note that only one step of this algorithm has been considered in [63] i.e. no continuation has been applied, and only one example with continuation procedure in [33].
Below is an example extracted from [33] where there are poles in Padé approximant using BPL, showing that IFS (denoted by Factorial in Fig.(1.7)) can surmount the problem of poles in Padé approximants.

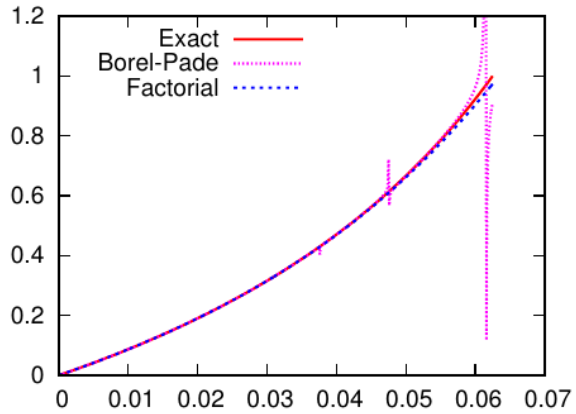


Figure 1.7 – Figure extracted from [33] where there is a pole in Padé approximant.

We propose a continuation of IFS algorithm using the criterion of the residual. First, note that relation Eq.(1.64) can be used to evaluate the error useful to apply a continuation as mentioned in [63].
The residual $Res(t)$ is defined by:

$$Res(t) = M\ddot{I}(t) + C\dot{I}(t) + KI(t) + \mathcal{N}_l(I(t)) \tag{1.71}$$

where $\dot{I}(t)$ the first derivative of IFS and $\ddot{I}(t)$ is the second derivative of IFS. Therefore the approximations of derivatives of the IFS algorithm are required for the residual evaluation replaced in the initial equation of the problem (Eq.(1.1)). Here it should be mentioned that to calculate the derivatives there are (as for the Padé approximant) at least two ways of doing it:

1. Calculate the derivatives of the series and establish an identification between the terms. (see remark 1.1.2).
2. Establish an analytical expression of the derivatives of the IFS sum which is an original contribution of this thesis. Those latter are established in Appendix B.

We will evaluate numerically some examples of the two ways of calculating the derivatives of IFS for high simulation times in the next chapter.
Therefore, the analytical approximations are examined as long as

$$\|Res(t)\|_2 < \epsilon \tag{1.72}$$

where ϵ is a small positive parameter called the tolerance and $\|.\|_2$ denotes the Euclidean norm. For the next step, we redefine a new starting point $u(t_1)$ where $t_1 > t_0$, which corresponds to the end of the step where relation Eq.(1.72) holds.
Therefore, the continuation procedure of IFS is summarized by the Algorithm 3.

Remark 1.3.5 *With the new approximations of the first and second derivatives of BPL and IFS in Appendices A, B, one can evaluate the residual and compare not only the IFS and BPL solutions to the exact or classical ones but also the velocity and acceleration (especially for nonlinear dynamics problems).*

Remark 1.3.6 *In this manuscript, we have chosen a criterion based on the residual for the following two reasons:*

1. *To ensure a good quality of the solution.*
2. *To have the same criterion for all the methods and to compare the solutions obtained by the following schemes ANM, BPL, IFS with the same tolerance value of the residue.*

It should be noted that the continuation of ANM by the residual criterion is based on the same concept described for BPL and IFS.

Algorithm 3 Continuation technique for IFS

Inputs:

1. The initial conditions u_0 and u_1.
2. The truncation order N.
3. The residual tolerance or the accuracy parameter ϵ.
4. The final time t_{end} or the domain in which we want to obtain the curve, for example: $[0, t_{end}]$.

Step 1: Compute series terms until a given order N.
Step 2: Calculate the sum of IFS denoted by $I(t)$.
Step 3: Apply the first and second derivatives approximations of IFS to calculate $\dot{I}(t)$ and $\ddot{I}(t)$. (See Appendix B).
Step 4: Evaluate the residual $Res(t)$ given in Eq.(1.71).
Step 5: Evaluate the validity domain t_{max} such that t_{max} is the last value of t for which $\|Res(t)\|_2 < \epsilon$.
Step 6: Evaluate a new starting point u_0 at a new time $t_0 = t_0 + t_{max}$.
Step 7: Go to Step 1 and repeat the algorithm until reaching the final time t_{end}.

1.4 Meijer-G approximant (MG)

According to the recent research [38], the more powerful method of summing divergent series than Borel-Laplace is the new Meijer-G summation with Hypergeometric approximants denoted by "MG". It is a simple algorithm proposed in 2018 by Mera et al. [38] that allows a fast and accurate low-order Borel summation. It replaces the Padé approximants in the Borel plane with more general and flexible hypergeometric functions that are, in principle, of an arbitrarily high order. MG is a generalized Borel-Hypergeometric approach where the hypergeometric ansatz is used to transform the coefficients of a divergent series into an array of hypergeometric approximations in the Borel plane [38].

This technique has been successfully applied to various examples in quantum mechanics and quantum field theory (QFT) (for example: zero-dimensional ϕ_4 theory, the quartic anharmonic oscillator, and the three-dimensional self-avoiding walk case in the N vector model,...) used as reference models for summing divergent series in these fields [38], [71]. It has also been applied for test cases where the perturbed expansion is not Borel summable along the positive real axis due to some obstacles (for example: summation of transseries/multi-instanton expansions). In all of these cases, the Meijer-G summation approach works effectively, provides a fast way to calculate the sum of divergent series, and outperforms Borel-Padé-Laplace technique [38].

In this manuscript, we will transform MG resummation to a time perturbation approach. According to the literature [38], this resummation method is particularly suitable

for series with a branch cut. However, the encouraging results presented in [38] motivated us to adapt this technique to sum divergent time series and investigate it further. Since MG is a recent summation method, we will not limit the analysis in this manuscript only for BPL and IFS summations, MG will be also considered as a new time integration scheme.
First of all, we review the Meijer-G approximant algorithm as it was proposed initially in [38]. Then, we will transform it into a time integration scheme by adding the necessary explanations on the subtle points of this transformation. Finally, we propose a new method for applying Meijer-G to a vector series, which has not been addressed in any of the previously listed studies of MG resummation.

1.4.1 Numerical algorithm of Meijer-G (MG)

The Meijer-G algorithm is described here [38]. First, we assume in the description of this algorithm that the truncation order N is an odd number (because we will use $(N-1)/2$ in the following). The generalization for even orders will be presented at the end of this subsection.
Let

$$u(t) = \sum_{n=0}^{N} u_n t^n \tag{1.73}$$

with normalized coefficients, $u_0 = 1$.
This algorithm consists of the following four steps:

- **Fist step: Borel transform**
  The first stage is to compute Borel-transformed coefficients denoted by b_n and given by

  $$b_n = \frac{u_n}{n!} \tag{1.74}$$

  and the ratios of the consecutive Borel-transformed coefficients denoted by $r_N(n)$ and given by:

  $$r_N(n) = \frac{b_{n+1}}{b_n} \tag{1.75}$$

- **Second step: Hypergeometric ansatz**
  The second stage is to make the ansatz that $r_N(n)$ is a rational function of n defined as:

  $$r_N(n) = \frac{\sum_{m=0}^{l} p_m n^m}{1 + \sum_{m=1}^{l} q_m n^m} \tag{1.76}$$

  where n is from 0 to $N-1$, $l = (N-1)/2$ and for this reason, the truncation order

N is assumed to be an odd number. The N unknowns p_m and q_m are determined from N equations in Eq.(1.76). A system of N linear equations with N unknowns is obtained and solved numerically.

- **Third step: Hypergeometric approximants in the Borel plane**
  After calculating the N unknowns p_m and q_m in the previous step, the third step is to find the two vectors $(x_1, \ldots, x_l)$ and $(y_1, \ldots, y_l)$ solutions of the following two equations:

$$\sum_{m=0}^{l} p_m x^m = 0 \tag{1.77}$$

$$1 + \sum_{m=1}^{l} q_m y^m = 0 \tag{1.78}$$

  We call those latter hypergeometric vectors. This is derived from the definition of hypergeometric functions. The hypergeometric vectors thus found identify in a unique way the hypergeometric function ${}_{l+1}F_l$ [72]:

$$\mathbb{B}_N(\xi) \equiv {}_{l+1}F_l\left(x, y, \frac{p_l}{q_l}\xi\right) \tag{1.79}$$

  where $x = (1, -x_1, \ldots, -x_l)$, $y = (-y_1, \ldots, -y_l)$ and $\mathbb{B}_N(\xi)$ is the hypergeometric approximant in the Borel plane.

- **Fourth step: Meijer-G approximants**
  Finally, one must return to the original series in the variable t and eliminate the $1/n!$ by the Laplace transform of Eq.(1.79). This gives the sum of Meijer-G denoted by $S_{MG}(t)$ represented as:

$$S_{MG}(t) = \int_0^\infty e^{-\xi} \mathbb{B}_N(t\xi) d\xi \tag{1.80}$$

  The Laplace transform of any hypergeometric function can be calculated analytically using Meijer-G functions. This yields to the following sum of Meijer-G approximant:

$$S_{MG}(t) = \frac{\prod_{i=1}^{l} \Gamma(-y_i)}{\prod_{i=1}^{l} \Gamma(-x_i)} G^{l+2,1}_{l+1,l+2}\left(\begin{matrix} 1, -y_1, \ldots, -y_l \\ 1, 1, -x_1, \ldots, -x_l \end{matrix} \middle| -\frac{q_l}{p_l t}\right) \tag{1.81}$$

where $\Gamma(x)$ is the Euler Gamma function and $G^{p,q}_{m,n}\left(\begin{array}{c} c_1,\ldots,c_p \\ d_1,\ldots,d_q \end{array}\middle| z\right)$ is the Meijer-G function [73] with $c_1,\ldots,c_p$ and $d_1,\ldots,d_q$ are the parameters of the function.

The hypergeometric ansatz and the resulting hypergeometric vectors are all that are required to parametrize these extremely complex functions. After determining the latter, the Meijer-G approximants can be parameterized for arbitrarily large truncation orders.

Recall 1.4.1 *The Meijer-G function has the following integral representation [74]:*

$$G^{p,q}_{m,n}\left(\begin{array}{c} c_1,\ldots,c_p \\ d_1,\ldots,d_q \end{array}\middle| z\right) = \frac{1}{2\pi i}\int_L \frac{\prod_{k=1}^{n}\Gamma(s-c_k+1)\prod_{k=1}^{m}\Gamma(d_k-s)}{\prod_{k=n+1}^{p}\Gamma(-s+c_k)\prod_{k=m+1}^{q}\Gamma(s-d_k+1)} z^s ds \tag{1.82}$$

where $z\neq 0$ and m,n,p,q are integer numbers such that $0\leq m\leq q$ and $0\leq n\leq p$. L is the path of integration that can be from $-i\infty$ to $+i\infty$ [74]. Note that Meijer-G function is an analytic function of z. This integral is called the Mellin-Barnes integral.

Remark 1.4.2 *The idea of generalizing the MG algorithm to an even truncation order N can be found in [38], [71]. It is straightforward to generalize to an even number of N. Simply subtract the constant term from the original series, factor the linear component, and then apply the technique for odd N as described previously. Finally, the result must be re-multiplied with the linear term and the re-added constant. For example, suppose that the truncation order N is an even number, for example consider $N=4$, therefore:*

$$u(t) = u_0 + u_1 t + u_2 t^2 + u_3 t^3 + u_4 t^4 \tag{1.83}$$

$$= u_0 + u_1 t\left(1 + \frac{u_2}{u_1}t + \frac{u_3}{u_1}t^2 + \frac{u_4}{u_1}t^3\right) \tag{1.84}$$

We apply the algorithm previously described for odd-N on $\left(1 + \frac{u_2}{u_1}t + \frac{u_3}{u_1}t^2 + \frac{u_4}{u_1}t^3\right)$, then we multiply the result by $u_1 t$ and finally we add u_0 to the final result.

Finally, it should be noted that a recent work in 2020 proposed a resummation algorithm based also on Meijer-G but without using the Borel or Padé techniques. The suggested algorithm in [75], [76] uses hypergeometric functions that have a zero radius of convergence by summarizing them using the equivalent integral representation of the Meijer-G function as an approximation of the given perturbation series.

1.4.2 Advantages/Disadvantages of MG

The advantages and disadvantages of MG that will be listed here are deduced from the examples that have been discussed in [38] in the field of quantum mechanics. Thus, we recall that these advantages led us to treat the Meijer-G scheme for temporal cases. We will check whether we would benefit from the same advantages for temporal cases.

Many advantages were obtained by using the Meijer-G approximants in QFT fields. A brief review of the advantages is recalled here [38]:

(*i*) The parameterization of Meijer-G approximants is simple. The hypergeometric ansatz provides a fast parameterization of Meijer-G approximants. The ratios between consecutive Borel transform coefficients are approximated by a rational function. This then allows defining of a hypergeometric function in the Borel plane. The Laplace transform of any hypergeometric function can be calculated analytically using Meijer-G functions.

(*ii*) The Meijer-G approximants are more precise than the Borel-Padé. The examples presented in [38] have shown that the Meijer-G approximant converges to the exact Borel sum in various QFT examples.

(*iii*) Meijer-G functions also provide an approximation to the generalized Borel sums. Generalizations of the Borel sum can be easily constructed using Meijer-G functions.

Although the use of the Meijer-G approximant has obvious advantages, it also has some disadvantages that should be taken into consideration. One may cite the following:

(*i*) Since Meijer-G approximants are new, their convergence properties are not yet developed and generalized for all truncation orders while the properties of the Padé approximants are well known.

(*ii*) For large truncation orders, Meijer-G approximants might be difficult to evaluate numerically.

(*iii*) There are difficulties with the numerical access to the Meijer-G function which does not exist in all the programming languages.

We now address the new part, compared to what has been done in the previously listed references, for the Meijer-G approximant:

1. Meijer-G for temporal test cases.
2. Meijer-G for vector series.
3. Meijer-G associated with the finite difference and finite element methods.

We start with the first point. The third point will be explored in the following chapters in the numerical applications.

1.4.3 Numerical algorithm MG used as a time integration scheme

In this section, we transform the numerical algorithm Meijer-G described previously into a time integration scheme. Therefore, all steps remain the same, except the last step where the perturbation parameter becomes $\hat{t}$. Therefore, the sum of Meijer-G approximant becomes:

$$S_{MG}(\hat{t}) = \frac{\prod_{i=1}^{l}\Gamma(-y_i)}{\prod_{i=1}^{l}\Gamma(-x_i)} G^{l+2,1}_{l+1,l+2}\left(\begin{array}{c} 1,-y_1,\ldots,-y_l \\ 1,1,-x_1,\ldots,-x_l \end{array}\middle| -\frac{q_l}{p_l\hat{t}}\right) \tag{1.85}$$

As already mentioned for the other methods, by applying the Meijer-G sum Eq.(1.85) just once, it may not be possible to reach $S_{MG}(\hat{t})$ for a large value of $\hat{t}$ up to a final time t. Hence the definition of the notion of the validity domain $[0, t_{max}]$ for the Meijer-G sum. Note that t_{max} of Meijer-G is not yet evaluated explicitly. Therefore, a continuation procedure is proposed to reach a given specific time t. We propose the continuation of the MG algorithm by using a criterion based on the residual.

The residual $Res(t)$ by using MG, is defined by:

$$Res(t) = M\ddot{S}_{MG}(t) + C\dot{S}_{MG}(t) + KS_{MG}(t) + \mathcal{N}(S_{MG}(t)) \tag{1.86}$$

where $\dot{S}_{MG}(t)$ the first derivative of Meijer-G sum and $\ddot{S}_{MG}(t)$ is the second derivative of Meijer-G sum. Therefore the approximations of derivatives of the MG algorithm are required for the residual evaluation replaced in the initial equation of the problem (Eq.(1.1)). Here it should be mentioned that to calculate the derivatives, there are (as for the Padé approximant) at least two ways of doing it:

1. Calculate the derivatives of the series and establish an identification between the terms. (see remark 1.1.2).
2. Establish an analytical expression of the derivatives of the Meijer-G sum. The latter are established in Appendix C.

We will see in the numerical example which one is the the most appropriate.

The continuation procedure of the MG algorithm used as a time integration scheme is summarized by:

Algorithm 4 Continuation technique for MG

Inputs:

1. The initial conditions u_0 and u_1.
2. The truncation order N.
3. The residual tolerance or the accuracy parameter ϵ.
4. The final time t_{end} or the domain in which we want to obtain the curve, for example: $[0, t_{end}]$.

Step 1: Compute series terms until a given order N.
Step 2: Apply the four steps of MG to calculate the Meijer-G sum which is $S_{MG}(t)$.
Step 3: Apply the first and second derivatives approximations of MG to calculate $\dot{S}_{MG}(t)$ and $\ddot{S}_{MG}(t)$. (See Appendix C).
Step 4: Evaluate the residual $Res(t)$ given in Eq.(1.86).
Step 5: Evaluate the validity domain t_{max} such that t_{max} is the last value of t for which $\|Res(t)\|_2 < \epsilon$.
Step 6: Evaluate a new starting point u_0 at a new time $t_0 = t_0 + t_{max}$.
Step 7: Go to Step 1 and repeat the algorithm until reaching the final time t_{end}.

For the Step 5 of the Algorithm 4 of the continuation technique of Meijer-G, it can be noted that the new starting point u_0 is $u(t_{max})$. To be able to apply Meijer-G algorithm with a normalized coefficient u_0 as described in subsection 1.4.1, one can proceed as follows:
Firstly, we have:

$$u(t) = u_0 + u_1 t + u_2 t^2 + \cdots + u_N t^n \tag{1.87}$$

Secondly, we multiply and divide the right hand side of Eq.(1.87) by u_0 to obtain the first normalized coefficient. Therefore,

$$u(t) = \left(\frac{u_0}{u_0} + \frac{u_1}{u_0}t + \frac{u_2}{u_0}t^2 + \cdots + \frac{u_N}{u_0}t^n\right) u_0 \tag{1.88}$$

$$= \left(1 + \frac{u_1}{u_0}t + \frac{u_2}{u_0}t^2 + \cdots + \frac{u_N}{u_0}t^n\right) u_0 \tag{1.89}$$

Thirdly, we apply the Meijer-G algorithm previously described on $\left(1 + \frac{u_1}{u_0}t + \frac{u_2}{u_0}t^2 + \cdots + \frac{u_N}{u_0}t^n\right)$, and finally we multiply the obtained result by u_0.
This is also valid for the case where u_0 is not 1.

1.4.4 New version of MG approximant for a series of vector

This subsection is dedicated to the new part that we propose, regards introducing the Meijer-G approximant for a series of vectors.
Suppose now that

$$\mathbf{u}(t) = \sum_{n=0}^{N} \mathbf{u}_n t^n \tag{1.90}$$

where $\mathbf{u}_n$ is either a vector of $\mathbb{R}^n$ or a continuous field. We will take inspiration from what has already been done for the Padé approximants for a vector series [44] and we will make the necessary adaptations to apply successfully Meijer-G to a vector series.

Example: First, we restrict the details of the development to an expansion of order $N = 5$. The idea is to orthogonalize the vectors $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4, \mathbf{u}_5$ by constructing an orthogonal basis using the classical Gram-Schmidt procedure in the following form:

$$\mathbf{u}_1 = \alpha_{11}\mathbf{u}_1^*, \tag{1.91}$$
$$\mathbf{u}_2 = \alpha_{21}\mathbf{u}_1^* + \alpha_{22}\mathbf{u}_2^*, \tag{1.92}$$
$$\mathbf{u}_3 = \alpha_{31}\mathbf{u}_1^* + \alpha_{32}\mathbf{u}_2^* + \alpha_{33}\mathbf{u}_3^*, \tag{1.93}$$
$$\mathbf{u}_4 = \alpha_{41}\mathbf{u}_1^* + \alpha_{42}\mathbf{u}_2^* + \alpha_{43}\mathbf{u}_3^* + \alpha_{44}\mathbf{u}_4^*, \tag{1.94}$$
$$\mathbf{u}_5 = \alpha_{51}\mathbf{u}_1^* + \alpha_{52}\mathbf{u}_2^* + \alpha_{53}\mathbf{u}_3^* + \alpha_{54}\mathbf{u}_4^* + \alpha_{55}\mathbf{u}_5^*. \tag{1.95}$$

This allows $\mathbf{u}(t)$ to be re-written in the new bases $\mathbf{u}_1^*, \mathbf{u}_2^*, \mathbf{u}_3^*, \mathbf{u}_4^*, \mathbf{u}_5^*$ by replacing these expansions in the polynomial form $\mathbf{u}(t)$ in Eq.(1.90):

$$\begin{aligned}\mathbf{u}(t) - \mathbf{u}_0 = (\alpha_{11} + \alpha_{21}t + \alpha_{31}t^2 + \alpha_{41}t^3 + \alpha_{51}t^4)t\mathbf{u}_1^* + (\alpha_{22} + \alpha_{32}t + \alpha_{42}t^2 + \alpha_{52}t^3)t^2\mathbf{u}_2^* \\ + (\alpha_{33} + \alpha_{43}t + \alpha_{53}t^2)t^3\mathbf{u}_3^* + (\alpha_{44} + \alpha_{54}t)t^4\mathbf{u}_4^* + \alpha_{55}t^5\mathbf{u}_5^*\end{aligned} \tag{1.96}$$

Polynomials of degree 1, 2, 3, and 4 thus appear. The main point is to replace these polynomials by the sum of Meijer-G approximants in the following way:

$$\mathbf{u}_{MG}(t) - \mathbf{u}_0 = S^1_{MG}(t)t\mathbf{u}_1^* + S^2_{MG}(t)t^2\mathbf{u}_2^* + S^3_{MG}(t)t^3\mathbf{u}_3^* + S^4_{MG}(t)t^4\mathbf{u}_4^* + \alpha_{55}t^5\mathbf{u}_5^* \tag{1.97}$$

where:

— $S^1_{MG}(t)$ is the Meijer-G sum of $\left(\alpha_{11} + \alpha_{21}t + \alpha_{31}t^2 + \alpha_{41}t^3 + \alpha_{51}t^4\right)$ *(i.e by applying the even Meijer-G algorithm to* $\alpha_{11} + \alpha_{21}t + \alpha_{31}t^2 + \alpha_{41}t^3 + \alpha_{51}t^4$*).*

— $S^2_{MG}(t)$ is the Meijer-G sum of $\left(\alpha_{22} + \alpha_{32}t + \alpha_{42}t^2 + \alpha_{52}t^3\right)$ *(i.e by applying the odd Meijer-G algorithm to* $\alpha_{22} + \alpha_{32}t + \alpha_{42}t^2 + \alpha_{52}t^3$*).*

— $S^3_{MG}(t)$ is the Meijer-G sum of $\left(\alpha_{33} + \alpha_{43}t + \alpha_{53}t^2\right)$ *(i.e by applying the even Meijer-G algorithm to* $\alpha_{33} + \alpha_{43}t + \alpha_{53}t^2$*).*

— $S^4_{MG}(t)$ is the Meijer-G sum of $(\alpha_{44} + \alpha_{54}t)$ *(i.e by applying the odd Meijer-G algorithm to* $\alpha_{44} + \alpha_{54}t$*).*

The two approximations of the curve are asymptotically equivalent in the following

sense:

$$\mathbf{u}_{MG}(t) - \mathbf{u}(t) \approx \alpha_{55} t^5 \mathbf{u}_5^* + O(t^6) \tag{1.98}$$

Generalization: We will generalize what we have just introduced for an expansion of order $N = 5$ to a given order N by providing the following two steps:

- The Gram-Schmidt method is used to define a set of orthonormal vectors:

$$\begin{aligned} \mathbf{u}_1 &= \alpha_{11} \mathbf{u}_1^* \\ \mathbf{u}_i &= \sum_{j=1}^{i} \alpha_{ij} \mathbf{u}_j^*, \quad \forall i \geq 2 \end{aligned} \tag{1.99}$$

  The Gram-Schmidt coefficients $\alpha_{ij}, (1 \leq j \leq i)$ are qualified by the orthonormality condition:

$$\langle \mathbf{u}_i^*, \mathbf{u}_j^* \rangle = \delta_{ij} \tag{1.100}$$

  where δ_{ij} is the symbol of Kronecker.
- By replacing Eq.(1.99) in Eq.(1.90), one can obtain:

$$\mathbf{u}_{MG}(t) - \mathbf{u}_0 = \sum_{i=1}^{N} t^i f^i(t) \mathbf{u}_i^* \tag{1.101}$$

  where the $f^i(t)$ are polynomials of degree $N - i$ calculated from the Gram-Schmidt coefficients:

$$f^i(t) = \sum_{k=0}^{N-i} t^k \alpha_{i+k,i} \tag{1.102}$$

  We replace each of these polynomials by their corresponding Meijer-G sum denoted by $S_{MG}^i(t)$. Therefore, the truncated series Eq.(1.101) has been replaced by a new Meijer-G representation which is of the following form

$$\mathbf{u}_{MG}(t) - \mathbf{u}_0 = \sum_{i=1}^{N} t^i S_{MG}^i(t) \mathbf{u}_i^* \tag{1.103}$$

Remark 1.4.3 *We would like to make it clear that in the rest of the manuscript, we will not apply Meijer-G resummation to all the examples we deal with in this manuscript, because we have already implemented some numerical tests in languages where the numerical access of the Meijer-G function $G_{m,n}^{p,q}$ is not available (e.g. the C language). For example, the Meijer-G resummation will not be applied to the level set function in Chapter 4 because the latter was written in the C programming language.*

Having outlined the methods, developments and our proposals, we now proceed to their numerical applications to different examples of ODEs and PDEs in the following chapters.

Chapter 2

Application on nonlinear ODEs

After exposing the time perturbation and summation approaches as time integration schemes in Chapter 1, we seek to validate the latter on various nonlinear ODE examples. This chapter aims to understand the behavior and the feasibility of these integrations, as well as to validate the first and second derivatives of BPL and IFS that we have established, and finally to consolidate the previous results. This leads us to highlight the potential of time perturbation approaches compared to the classical schemes by providing examples where the latter suffers from many issues. The properties and qualities of IFS will be studied in detail since the latter has already been applied as a first prospective study on some simple examples. The Meijer-G (MG) algorithm used in temporal cases will be studied on two ODE examples in this chapter. We recall that MG will not be applied to all the examples because of an access difficulty of the Meijer-G function in some programming languages.

In this chapter, different examples of nonlinear ODEs will be presented. First, we will consider again the popular Euler equation in order to validate and study the behavior of MG on this example. Then, the simple harmonic oscillator will also be studied by MG to validate the proposal of studying MG in temporal cases. BPL, IFS, and ANM will be used for further examples. The details will be performed only for a chosen solid mechanics problem which is the free and forced Van der Pol oscillators. Then, the proposed techniques are applied to the nonlinear combustion equation, the hardening spring problem, and the nonlinear simple pendulum problem. These three examples aim to investigate the behavior of these methods for problems presenting large nonlinearity where classical explicit schemes suffer from different constraints. Finally, the efficiency of these methods is also established for the most famous chaotic system which is the Lorenz system (three degrees of freedom DOF example). The goal of this example is to check whether the temporal approaches can respect the different phase portraits obtained by the Lorenz system and to validate their solving ability for a 3 DOF example before dealing with problems involving several DOFs in the next chapter.

Since we have a priori no information on the convergence type of the obtained numerical series, the BPL and IFS summation techniques are also applied. Therefore, even if the obtained series are convergent, we will use resummation techniques to observe their

interest on a convergent series and study their accuracy and robustness.

For a fair comparison: the accurate first, third and fourth-order Runge-Kutta (RK) methods using fixed or adaptive time steps schemes are considered in this chapter.

2.1 Application to the Euler equation

Consider again the Euler equation

$$t^2\dot{u} + u = t \tag{2.1}$$

with the following initial condition

$$u(t_0 = 0) = 0 \tag{2.2}$$

The formal power series solution of this equation is

$$u(t) = \sum_{n=0}^{\infty}(-1)^n n! t^{n+1} \tag{2.3}$$

which is a divergent series.
As the resulting series is divergent, this example is important to be tested by the summation algorithms. Since BPL has been applied to this example in Chapter 1 (subsection 1.1.3) and in [32], [33], the goal of this example is to present a first validation of the Meijer-G algorithm used in the temporal case. To be able to apply the Meijer-G approximants, the initial conditions u_0 cannot be zero. The series are calculated up to a truncation order $N = 5$. We compare the solutions obtained by the MG, ANM, BPL, and IFS without any continuation in Fig.(2.1). The solution obtained by the Meijer-G approximant is in good agreement with the exact solution and presents a better approximation of the solution than ANM, BPL, and IFS.
Now, we maintain the truncation order $N = 5$ for Meijer-G (MG) and increase it to $N = 10$ for ANM, BPL, and IFS. Even with a higher truncation order than the Meijer-G approximant, MG provides a better solution than the other methods as can be seen in Fig.(2.2) using a small truncation order.
We recall that the general definition of the Meijer-G function is given by a line integral in the complex plane, hence the possibility to obtain a complex result. (See the definition of the Meijer-G function in Chapter 1). We note that we are interested in the real part of the solution. The obtained imaginary part of the Meijer-G solution of the Euler equation is zero.

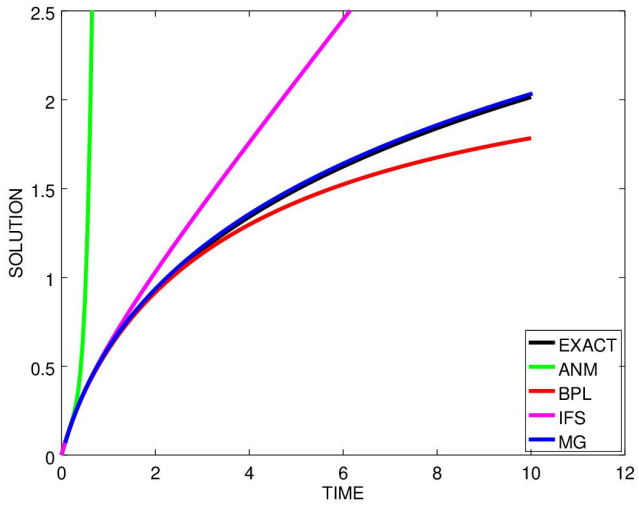


Figure 2.1 – Approximated solution using time perturbation method ANM, time perturbation-resummation methods BPL, IFS and MG for $N = 5$ compared to the exact solution.

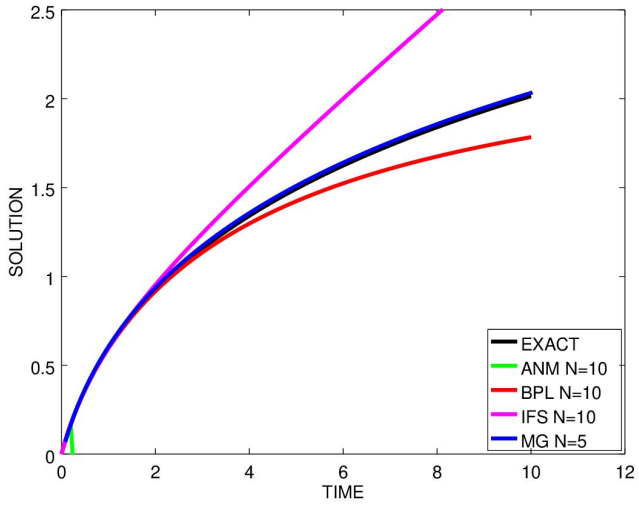
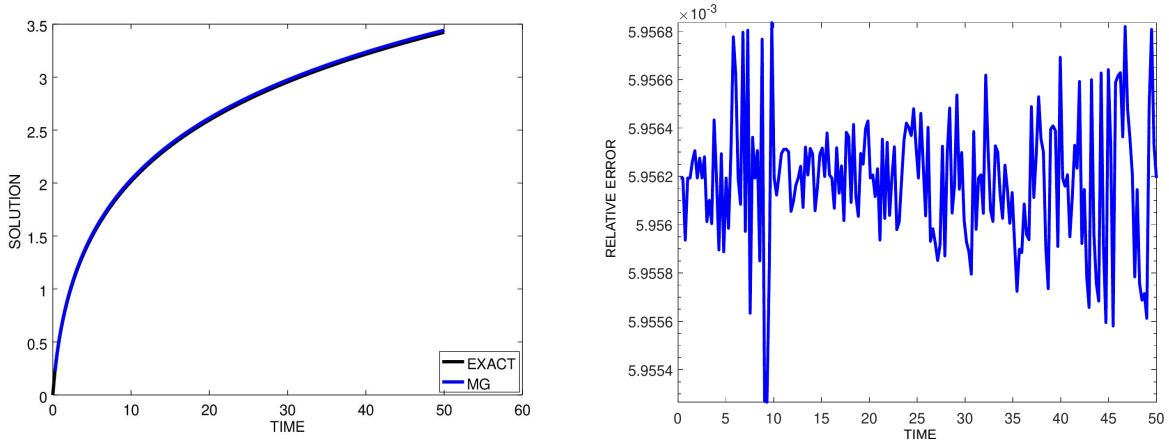


Figure 2.2 – Approximated solution using time perturbation method ANM, time perturbation-resummation methods BPL and IFS for $N = 10$ and time perturbation-resummation method MG for $N = 5$, compared to the exact solution.

The simulation of Meijer-G approximant is computing until a final time equal to $t = 50$s without continuation in Fig.(2.3a). The error between the exact (or called the reference solution) and the one obtained by the Meijer-G approximant is evaluated in Fig.(2.3b). The error is of order $5.9 \cdot 10^{-3}$. This test shows a good behavior of Meijer-G on this particular example of divergent series without the need to perform a continuation to reach the final time $t = 50$s.



(a) (b)

Figure 2.3 – (a): Approximated solution using time perturbation-resummation method MG for $N = 5$ compared to the exact solution for a final time $t = 50s$.
(b): Relative error between the MG solution for $N = 5$ and the exact solution: $\frac{|\text{exact solution} - \text{Meijer-G solution}|}{|\text{exact solution}|}$.

2.2 Application to the simple harmonic oscillator

Consider the following form of the simple harmonic oscillator given by

$$\ddot{u} + u = 0 \tag{2.4}$$

where $\dot{u} = \frac{du}{dt}$ and $\ddot{u} = \frac{d^2u}{dt^2}$, with the following initials conditions

$$\begin{cases} u(0) &= 1 \\ \dot{u}(0) &= 2 \end{cases} \tag{2.5}$$

which has an exact solution for $t_0 = 0$ on the form:

$$u(t) = u(0)\cos(t) + \dot{u}(0)\sin(t) \tag{2.6}$$

Series terms u_k, for $k \geq 2$ are given by the following recurrence formula

$$k(k-1)u_k = -u_{k-2}$$

The Meijer-G approximant is applied to this example to test the effectiveness of the proposal way to use it in a temporal case. The series are calculated up to a truncation order $N = 5$. As can be seen in Fig.(2.4), the imaginary part of the solution obtained by Meijer-G solution is zero until $t = 1$s. We note that in some cases, it is quite simple to give

a physical meaning to the imaginary part. In this case, it is not so obvious and the physical interpretation is not our main objective. Our goal is to obtain a precise approximation of the solution. In the following, we just plot the real part of the Meijer-G solution that interests us.

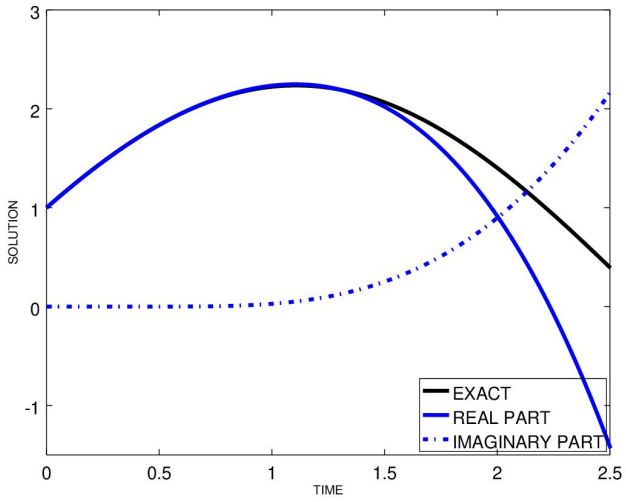
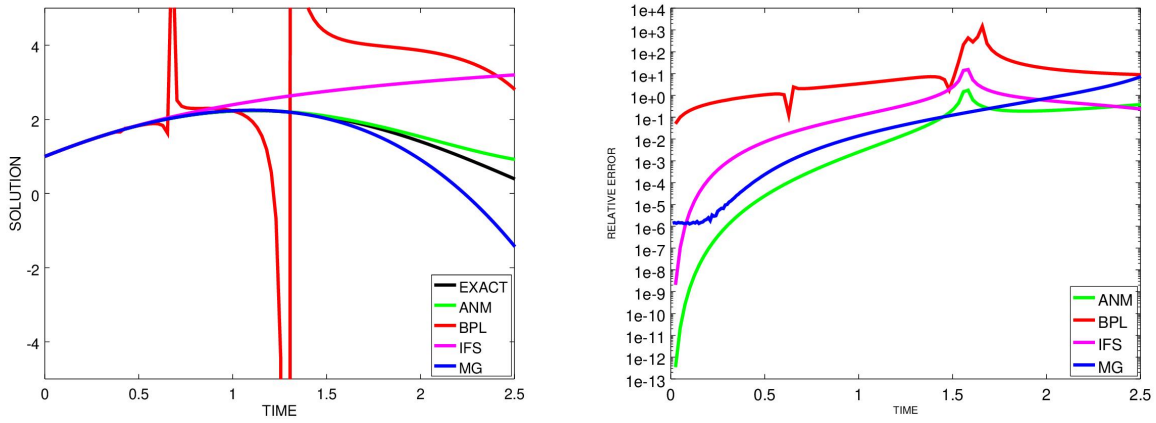


Figure 2.4 – Imaginary and real parts of the Meijer-G solution for $N = 5$, for a simple harmonic oscillator example.

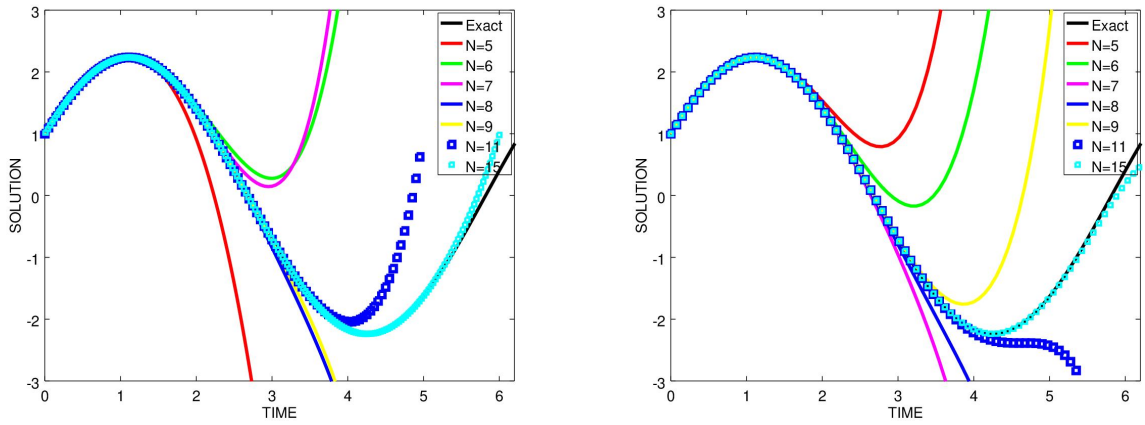
In Fig.(2.5a), we compare the solutions obtained by the Meijer-G approximant MG, ANM, BPL, and IFS without any continuation for $N = 5$ with the exact solution. The relative error between each of the solutions obtained with these methods and the exact solution is shown in Fig.(2.5b). The goal of this example is to verify that Meijer-G is also able to avoid the problem of the Padé poles that BPL encounters in some temporal test cases for some truncation orders. The results show that for this case and for $N = 5$, BPL suffers from the appearance of poles differently in contrast to the other methods.

The behavior of the solutions obtained by MG and ANM for different truncation orders N without any continuation is shown in Fig.(2.6). First, one can see in Fig.(2.6a) that the MG algorithm for even and odd truncation orders are validated (see subsection 1.4.1 and remark 1.4.2 in Chapter 1). Then, one can also observe that by increasing the truncation order for Meijer-G, the obtained solution of MG is more closer to the exact one. Finally, the behavior of Meijer-G for different truncation orders is similar to that of ANM presented in Fig.(2.6b) for this example.



(a) (b)

Figure 2.5 – (a): Approximated solution using time perturbation method ANM, time perturbation-resummation methods BPL, IFS, and MG for $N = 5$ compared to the exact solution.
(b): Relative error between the solution obtained using time perturbation method ANM, time perturbation-resummation methods BPL, IFS, and MG for $N = 5$ and the exact solution: $\frac{|\text{reference solution} - \text{numerical solution}|}{|\text{reference solution}|}$.



(a) MG solution (b) ANM solution

Figure 2.6 – Approximated solution using time perturbation method ANM and time perturbation-resummation method MG for different truncation orders compared to the exact solution.

2.3 Application to the free Van der Pol oscillator

In this section, we are interested to study the effectiveness of time perturbation and resummation methods on the free Van der Pol oscillator problem. The details of the

study will be presented only for this example. The Van der Pol oscillator has attracted considerable interest from many researchers for a long time since it models many problems in different areas [77]. It is a dynamic system described by a variable $x(t)$ satisfying the following nonlinear differential equation of the form

$$\ddot{x} - \mu(1 - x^2)\dot{x} + x = 0 \tag{2.7}$$

where $\dot{x} = \dfrac{dx}{dt}$ and $\ddot{x} = \dfrac{d^2x}{dt^2}$, with the following initials conditions

$$\begin{cases} x(0) &= a \\ \dot{x}(0) &= b \end{cases} \tag{2.8}$$

where a and b are two parameters, and $\mu \geq 0$ denotes the control parameter or called also the coefficient of nonlinearity. Therefore, the nonlinear behavior of the oscillations depends on μ. Therefore, according to its value, the trajectories have precise characteristics.

For $\mu = 0$, equation Eq.(2.7) will be the simple harmonic oscillator addressed in section 2.2. For $\mu \neq 0$, equation Eq.(2.7) has no exact solution, therefore it was first integrated by the classical time discretization schemes, to compare the efficiency of time perturbation methods to the classical ones.

2.3.1 Classical discretization schemes

Equation Eq.(2.7) is written as a first order system of differential equation:

$$\begin{cases} \dot{x} &= y \\ \dot{y} &= \mu(1 - x^2)\dot{x} - x \end{cases} \tag{2.9}$$

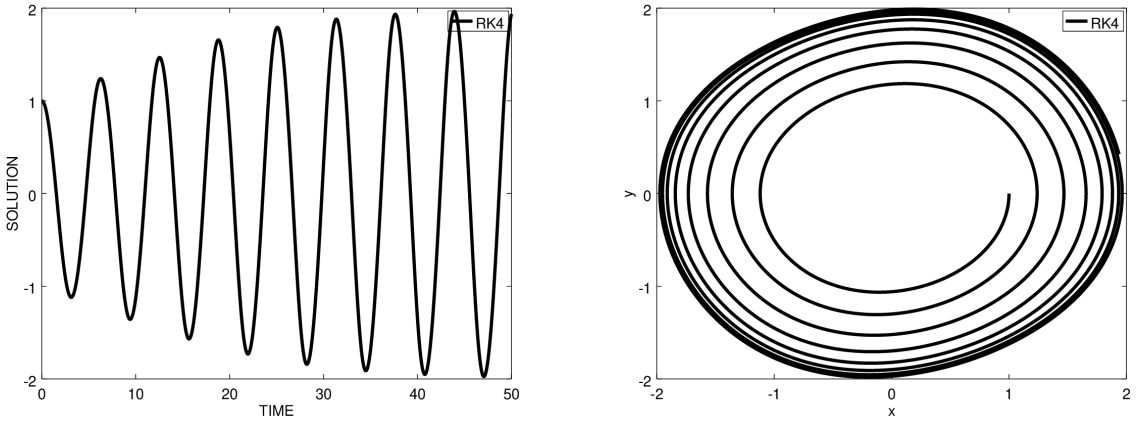
A convergence study for different time steps Δt between Runge-Kutta of 1-st (RK1) and 4th order (RK4) has been carried out. For sake of simplicity, only results obtained with RK4 are presented for this example. The goal of this comparison is to get a good approximation of the solution since it will be the reference solution.

Firstly, constant time step $\Delta t = 10^{-3}s$ is chosen for RK4. Secondly, a type of adaptive RK4 has been used as presented in [78]. It computes a local error between two solutions obtained after performing two consecutive time steps using step size Δt and another using only a one-time step but with a larger step size $2\Delta t$. The tolerance parameter on local error denoted Δ_0, makes it possible to adjust the step size.

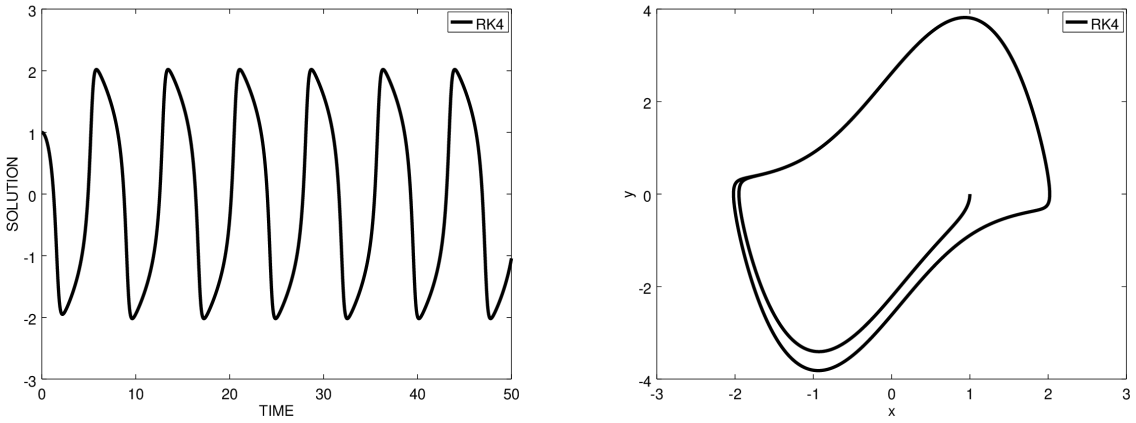
Initial conditions at a given time $t_0 = 0$ are arbitrarily chosen as $x(0) = 1$ and $\dot{x}(0) = 0$ and the coefficient of nonlinearity as $\mu = 2$.

It should be noticed that for every choice of initial conditions, except $\{x(0) = 0, \dot{x}(0) = 0\}$, a unique periodic motion is presented.

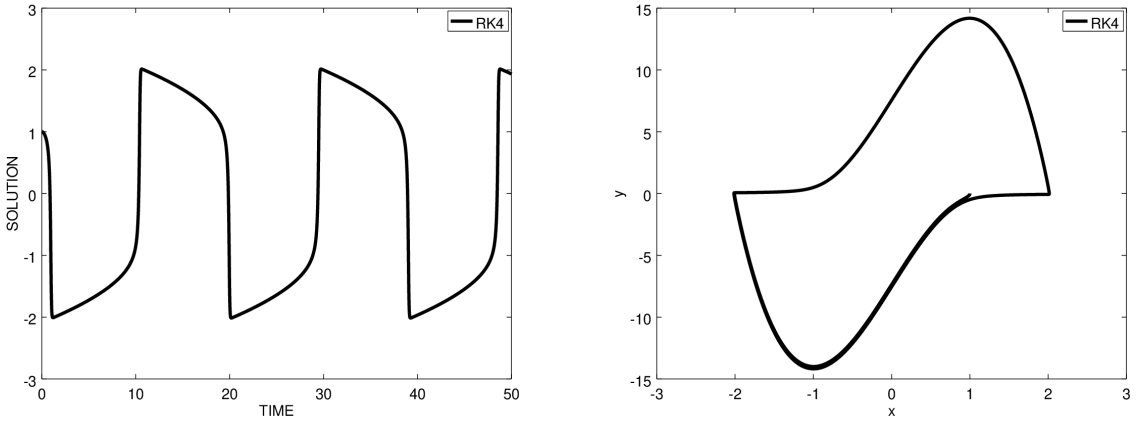
The results of numerical integration by Runge-Kutta methods show that the characteristic of the trajectory depends on the value of the coefficient of nonlinearity μ are presented in Fig.(2.7). For small μ, the limit cycle is nearly circular. For large μ, the limit cycle is no longer nearly circular and is representative of bounded periodic oscillatory behavior.



(a) $\mu = 0.1$



(b) $\mu = 2$



(c) $\mu = 10$

Figure 2.7 – Solutions (left) and phases portrait (right) for the free Van der Pol oscillator using RK4 with a discrete-time step $\Delta t = 10^{-3}s$, for different values of μ.

2.3.2 Time perturbation series recurrence formula

In this subsection, time perturbation and resummation methods are applied to the free Van der Pol oscillator. We seek to demonstrate the accuracy of these methods by comparing their results to RK4, check their ability on a problem involving significant temporal variations and limit cycles, and consolidate some results previously achieved. Consider again the Van der Pol equation

$$\ddot{x} - \mu(1 - x^2)\dot{x} + x = 0 \tag{2.10}$$

Unknowns of this equation $x(t)$, $\dot{x}(t)$ and $\ddot{x}(t)$ are sought as time power series

$$\begin{aligned} x(\hat{t}) &= x_0 + \sum_{i=1}^{N} x_i \hat{t}^i \\ \dot{x}(\hat{t}) &= \sum_{i=1}^{N} i x_i \hat{t}^{i-1} \\ \ddot{x}(\hat{t}) &= \sum_{i=2}^{N} i(i-1) x_i \hat{t}^{i-2} \end{aligned} \tag{2.11}$$

where N denotes the truncation order of the series. When we replace the expansion Eq.(2.11) into equation Eq.(2.10), and identify according to the power of $\hat{t}$, we notice that it is not easy to find the recurrence formula that allows to calculate the series term. For this reason, let the following change of variable

$$z = x^2 \tag{2.12}$$

This yields to the new quadratic equation

$$\ddot{x} - \mu(1 - z)\dot{x} + x = 0 \tag{2.13}$$

Writing $z(\hat{t})$ as time power series

$$z(\hat{t}) = z_0 + \sum_{i=1}^{N} z_i \hat{t}^i \tag{2.14}$$

This yields to the following sequence problems

$$\begin{cases} \hat{t}^0 & : & 2x_2 - \mu(x_1 - z_0 x_1) + x_0 = 0 \\ \hat{t}^1 & : & 6x_3 - \mu(2x_2 - 2x_2 z_0 - x_1 z_1) + x_1 = 0 \\ \hat{t}^2 & : & 12x_4 - \mu(3x_3 - 3x_3 z_0 - 2z_1 x_2 - z_2 x_1) + x_2 = 0 \\ & \vdots & \\ \hat{t}^N & : & (N+2)(N+1)x_{N+2} - \mu\left((N+1)x_{N+1} - \sum_{i=0}^{N}(N+1-i)x_{N+1-i}z_i\right) + x_N = 0 \end{cases}$$

Therefore, series terms x_k for $k \geq 2$ are given by the following recurrence formula

$$k(k-1)x_k = \mu\left((k-1)x_{k-1} - \sum_{i=0}^{k-2}(k-1-i)x_{k-1-i}z_i\right) - x_{k-2}$$

where

$$z_i = \sum_{j=0}^{i} x_j x_{i-j} \tag{2.15}$$

Once series terms x_k are numerically evaluated, range of validity t_{max} of the perturbed representation Eq.(2.11) is sought. To do so, two techniques are used in this study. The first uses Eq.(1.17), when using this criterion the method will be noted ANM(C). For the sake of comparison with other perturbation methods, the residual criterion will be used to evaluate this t_{max} (see remark 1.3.6). This method is noted by ANM(R) or simply ANM.

2.3.3 Numerical study of the two possible ways to evaluate the derivatives of the IFS sum

We recall that in Chapter 1, we proposed two possible ways to evaluate the derivatives of the Borel sum, the IFS sum and the Meijer-G sum which allow to calculate the residual and thus to apply the continuation based on the latter.

First possibility: It is based on calculating the initial series terms, the first and second derivative series terms, and finally applying the resummation algorithm for each of these terms. Therefore, this possibility is related to the three following steps:

- Calculate the series terms denoted by $x_k, \forall k = 1, \cdots, N$ corresponding to each problem, then apply the resummation algorithm to these terms for a truncation order equal to N.
- Calculate the first derivative series terms denoted by $\dot{x}_k, \forall k = 1, \cdots, N-1$ that are linked to the initial series terms x_k by the following relation:

$$\dot{x}_k = (k+1)x_{k+1} \quad \forall k = 1, \cdots, N-1 \tag{2.16}$$

then apply the resummation algorithm to these new terms $\dot{x}_k$, for a truncation order equal to $N-1$.
For a more detailed explanation of how Eq.(2.16) is obtained, see the remark 1.1.2 in Chapter 1.

- Calculate the second derivative series terms denoted by $\ddot{x}_k, \forall k = 1, \cdots, N-2$ that are linked to the initial series terms x_k by the following relation:

$$\ddot{x}_k = (k+1)(k+2)x_{k+2} \quad \forall k = 1, \cdots, N-2 \tag{2.17}$$

then apply the resummation algorithm to these new terms $\ddot{x}_k$, for a truncation order equal to $N-2$.
For a more detailed explanation of how Eq.(2.17) is obtained, see the remark 1.1.2 in Chapter 1.

As a result, this way requires three serial term calculations (x_k, $\dot{x}_k$, and $\ddot{x}_k$) and three resummation algorithms applied for each of these.
Since this possibility is based on a series derivation, it will be called **"Series.D"** in what follows in this section.

Second possibility: It consists of calculating the initial series terms once, then applying the resummation algorithm to these terms, and finally establishing the expression of the first and second derivatives to the sum formulas obtained through the resummation algorithms. For more details, this possibility is based on the following three steps:

- Calculate the series terms corresponding to each problem $x_k, \forall k = 1, \cdots, N$, then apply the resummation algorithm to these terms.
- Establish an analytical expression for the first derivatives of the BPL, IFS, and MG sum determined respectively in Appendices A, B, and C.
- Establish an analytical expression for the second derivatives of the BPL, IFS, and MG sum determined respectively in Appendices A, B, and C.

As a result, this way requires one calculation of series terms, one resummation algorithm, and two approaches of first and second derivation applied to the summation formulas of BPL, IFS, and MG.
Since this possibility is based on a resummation derivation, it will be called **"Resummation.D"** in what follows in this section.

Therefore, it is quite natural to wonder about the difference between **"Series.D"** and **"Resummation.D"**. The questions that can arise are:

- Which one is more appropriate to be applied?
- Which one is more suitable for long-time simulation?

– Which one is faster in terms of CPU times and requires fewer time steps to reach the same final time?

Hence, we will indicate the answers through some numerical tests applied to the example of Van der Pol by the Inverse factorial series (IFS) with continuation. Therefore, for "Resummation.D", Appendix B is used (first and second derivatives approximations of the exact sum of IFS).

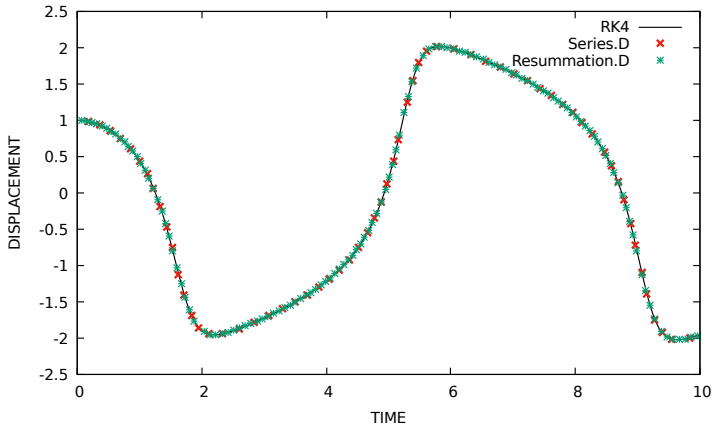
Test 1: IFS with continuation to a final time $t = 10$s using a small value of the coefficient of nonlinearity $\mu = 2$

The first test is a continuation of IFS for a final time $t = 10$s using a small value of the coefficient of nonlinearity which is fixed to $\mu = 2$. The two ways previously mentioned to calculate the derivatives and then the residual, are used for the continuation technique. The truncation order is fixed to $N = 15$ and the residual tolerance to $\epsilon = 10^{-4}$ for the both ways.
The reference solution is RK4 with a discrete-time step $\Delta t = 10^{-3}$s. The solutions of displacement ($x(t)$), velocity ($\dot{x}(t)$ or the first derivative of the displacement), and acceleration ($\ddot{x}(t)$ or the second derivative of the displacement) obtained by IFS are compared to the reference solutions in Fig.(2.8). A good agreement of the IFS solutions (using both ways) with the reference solutions can be seen in the Fig.(2.8), which shows the effectiveness of "Series.D" and "Resummation.D" for this test.

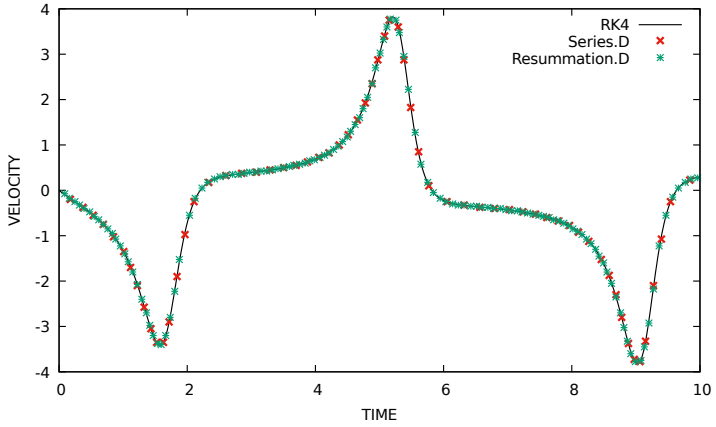
After validating the accuracy of "Series.D" and "Resummation.D", we are now interested in comparing the number of time steps required by both ways to reach the same final time $t = 10$s. As can be observed in Tab.(2.1), "Series.D" requires less steps than "Resummation.D" to reach $t = 10$s. Another test is presented afterwards to check whether these results hold true.

Residual tolerance ϵ	IFS with "Series.D"	IFS with "Resummation.D"
10^{-3}	47	94
10^{-4}	64	128
10^{-5}	86	168

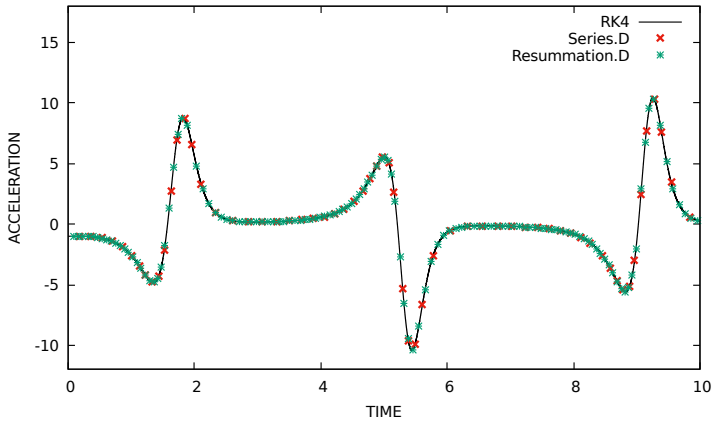
Table 2.1 – Number of steps required to reach $t = 10$s for the free Van der Pol oscillator for $\mu = 2$, using time perturbation-resummation method IFS with $N = 15$ and different values of the residual tolerance ϵ with "Series.D" and "Resummation.D".



(a) displacement $x(t)$



(b) velocity $\dot{x}(t)$



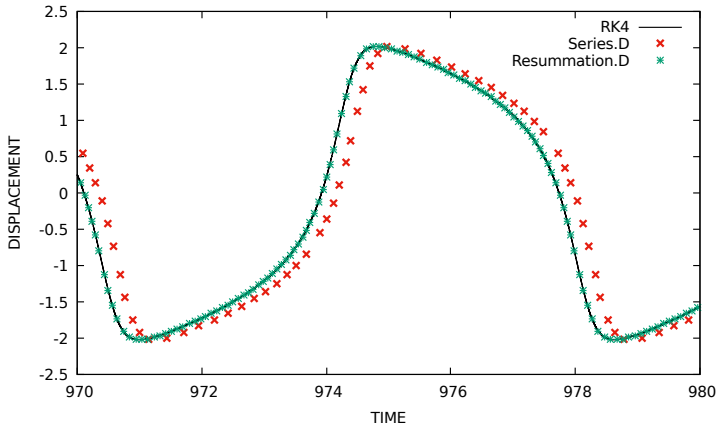
(c) acceleration $\ddot{x}(t)$

Figure 2.8 – Approximated solutions (displacement, velocity, acceleration) using time perturbation-resummation method IFS for $\epsilon = 10^{-4}$, $N = 15$, for the free Van der Pol oscillator for $\mu = 2$. In black the reference RK4 solution using $\Delta t = 10^{-3}s$.

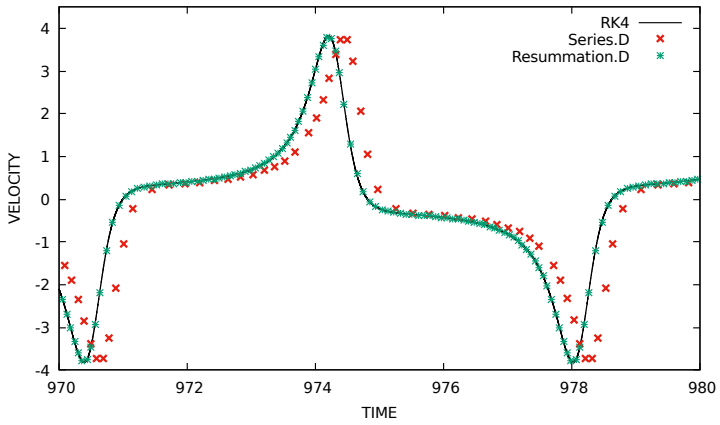
Test 2: IFS with continuation to a final time $t = 1000$s using a small value of the coefficient of nonlinearity $\mu = 2$

The second test is a continuation of IFS for a final time $t = 1000$s using a small value of the coefficient of nonlinearity $\mu = 2$. The truncation order and the residual tolerance are maintained respectively to $N = 15$ and $\epsilon = 10^{-4}$.

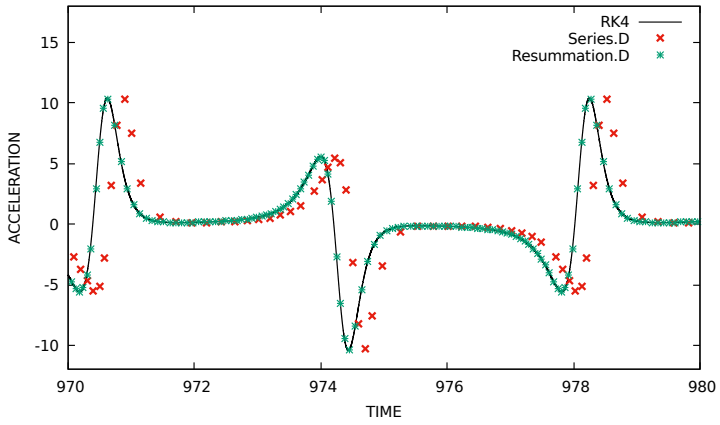
It can be seen in Fig.(2.9) that the solutions of displacement, velocity, and acceleration obtained by IFS using "Series.D" show a drift with the reference solution, whereas the solutions obtained using "Resummation.D" are in good agreement with the reference solution. We recall that for the sake of comparison, the same numerical parameters are used for the both methods. This test indicates that the application of "Series.D" over a long simulation period time may present a risk of phase shift with respect to the reference or the exact solution. This may create other significant phase differences at longer intervals.



(a) displacement $x(t)$



(b) velocity $\dot{x}(t)$

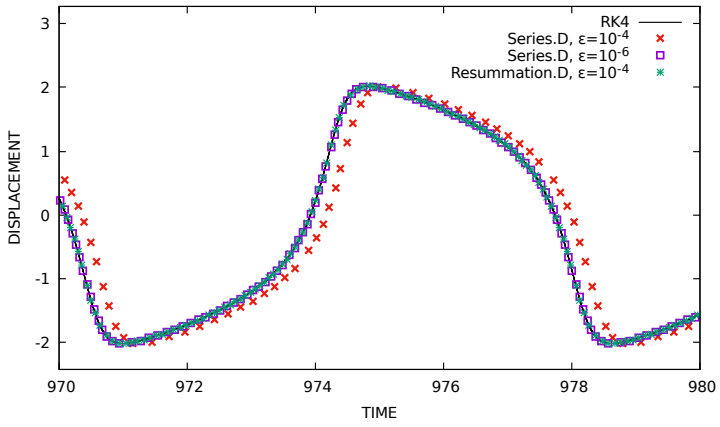


(c) acceleration $\ddot{x}(t)$

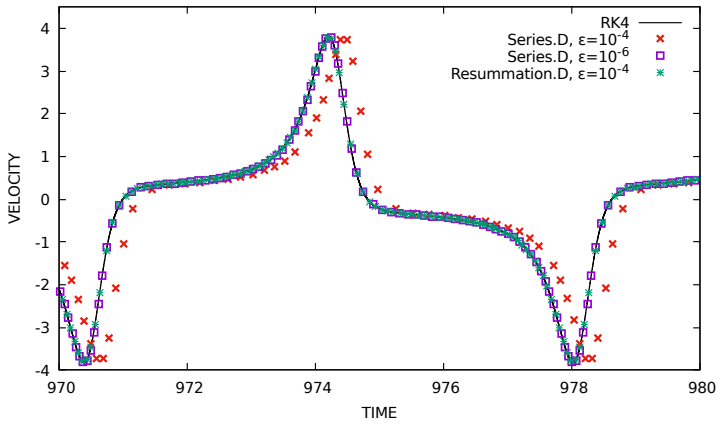
Figure 2.9 – Approximated solutions (displacement, velocity, acceleration) using time perturbation-resummation method IFS for $N = 15$ and different values of the tolerance, for the free Van der Pol oscillator for $\mu = 2$. In black the reference RK4 solution using $\Delta t = 10^{-3}s$.

The residual tolerance is now decreased from $\epsilon = 10^{-4}$ to $\epsilon = 10^{-5}$ for the "Series.D" to check if it allows to obtain a faithful solution as "Resummation.D" and the reference

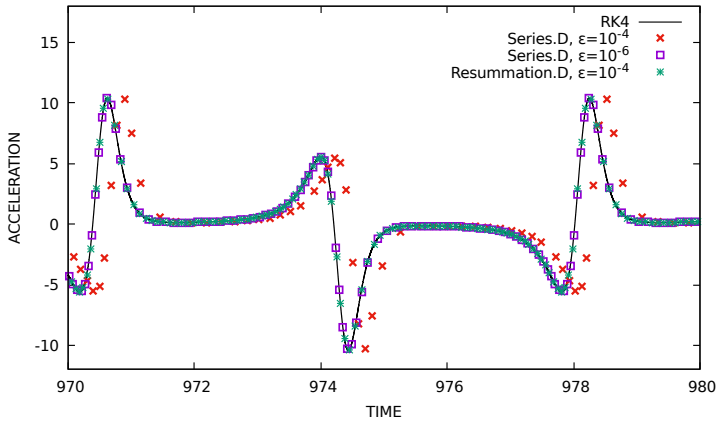
solution, for a long final time. After various studies on the range of tolerances, it was found that "Series.D" with a residual tolerance $\epsilon = 10^{-6}$ provides an efficient solution compared to the reference solution as can be appreciated in Fig.(2.10).



(a) displacement $x(t)$



(b) velocity $\dot{x}(t)$



(c) acceleration $\ddot{x}(t)$

Figure 2.10 – Approximated solutions (displacement, velocity, acceleration) using time perturbation-resummation method IFS for $\epsilon = 10^{-4}$, $N = 15$, for the free Van der Pol oscillator for $\mu = 2$. In black the reference RK4 solution using $\Delta t = 10^{-3}s$.

Regarding the number of steps, IFS using "Series.D" and a residual tolerance $\epsilon = 10^{-6}$ is compared to the IFS using "Resummation.D" with a residual tolerance $\epsilon = 5 \times 10^{-4}$

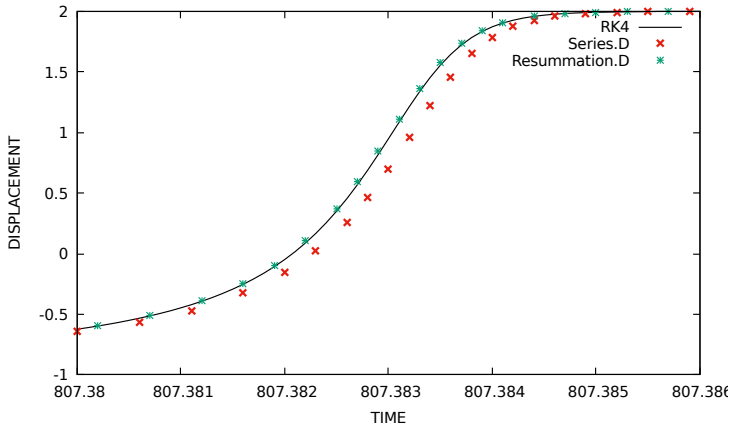
(since these two optimal choices of the residual tolerance provide an efficient solutions for each way, until a final time $t = 1000s$). As can be seen in Tab.(2.2), the number of steps of "Resummation.D" is in the same order of magnitude than that of "Series.D" for this test.

IFS with "Series.D" and $\epsilon = 10^{-6}$	IFS with "Resummation.D" and $\epsilon = 5 \times 10^{-4}$
10617	10223

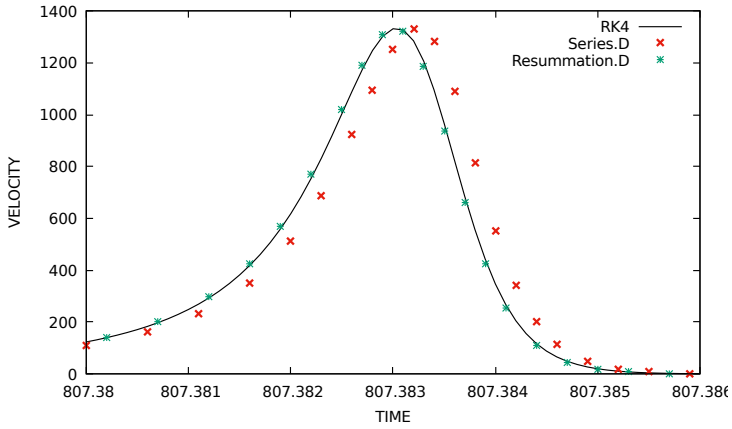
Table 2.2 – Number of steps required to reach $t = 1000$s for the free Van der Pol oscillator for $\mu = 2$, using time perturbation-resummation method IFS with $N = 15$ with different values of the residual tolerance ϵ.

Test 3: IFS with continuation to a final time $t = 1000$s using a large value of the coefficient of non linearity $\mu = 1000$

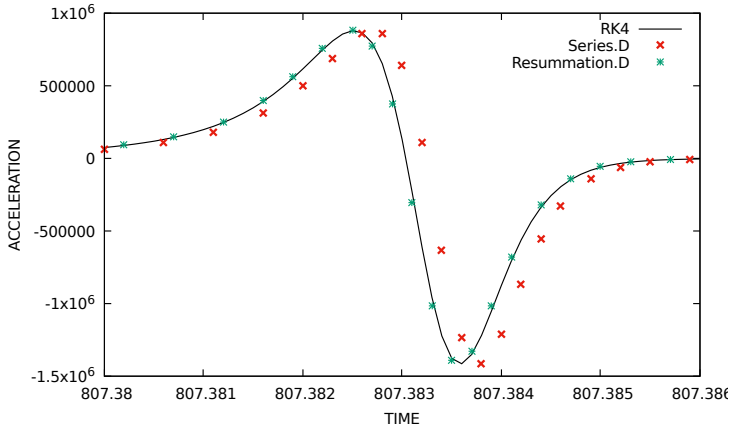
We have already addressed the case of a small nonlinearity coefficient for a final time $t = 10$s in Test 1, and $t = 1000$s in Test 2. In this third test case, we increase the value of the nonlinearity coefficient to $\mu = 1000$. The goal of this test is to study the behavior of using a continuation of IFS via the derivatives of "Series.D" and "Resummation.D" for a problem presenting large nonlinearity.
The simulation is carried out until a final time $t = 1000$s. We will only display in the figures the part where there is a jump in the solution, which is equivalent to the time interval $[807.38, 807.385]$s. As can be seen in Fig.(2.11), there is a shift between the reference solution that is either the RK4 with $\Delta t = 10^{-5}$s or the ANM solution (this part will be described in detail in subsection 2.3.5) and the IFS with "Series.D" with a residual tolerance $\epsilon = 10^{-4}$. With the same choice of ϵ, "Resummation.D" is able to provide an efficient solution.



(a) displacement $x(t)$



(b) velocity $\dot{x}(t)$



(c) acceleration $\ddot{x}(t)$

Figure 2.11 – Approximated solution (displacement, velocity, acceleration) using time perturbation-resummation method IFS for $\epsilon = 10^{-4}$, $N = 15$, for the free Van der Pol oscillator for $\mu = 1000$. In black the reference RK4 solution using $\Delta t = 10^{-3}s$.

Indication of this study: After these different test cases, one can in principle state that the application of the IFS continuation using the exact approximations of the deriva-

tives in Appendix A (named here by "Resummation.D") guarantees a good quality of the solution for different nonlinearity parameters of the problem. Moreover, the solution obtained by the latter is also efficient for long-time intervals. Calculate the derivatives of the series and establish an identification between the terms (named here by "Series.D") may also provide an efficient solution by using fewer time steps than "Resummation.D" to reach the same final time in some cases. However, one must be careful because "Series.D" may cause some shifts in the solution obtained for simulations over long time intervals. As a result, the residual tolerance must be reduced, which increases the number of steps required to reach the final time and, as a result, the CPU time too. This may also lead to the possibility of a large shift in the solution over time.
It is expected that the same results occur with the other resummation methods BPL and MG using "Series.D" and "Resummation.D" based on Appendix A for Borel sum and Appendix C for Meijer-G sum. In addition, it should be noted that similar numerical tests were carried out by using BPL. The results revealed that numerical difficulties could appear in particular in the numerical choice of the truncation orders of the 3 series (the initial series, the first derived series, and the second derived series) to avoid the appearance of poles of the Padé approximations. Therefore, after this simple study, we note that for the rest of the manuscript we will only apply the second possibility ("Resummation.D") to calculate the residual of BPL, IFS, and MG and finally to perform an efficient continuation. Therefore, Appendices A, B and C will be used.

2.3.4 Numerical results for time perturbation methods

Initial conditions are set to $x(0) = 1$ and $\dot{x}(0) = 0$, and the coefficient of nonlinearity is fixed in this subsection to $\mu = 2$.

Numerical results of ANM

Firstly, ANM behavior is presented without any continuation procedure. Effects of the truncation orders N are depicted in Fig.(2.12). Continuous ANM solutions are compared to the discrete solutions obtained using RK4. It is observed that ANM can reproduce correct solutions for a large range of time when compared to RK4. Then the continuation technique is used. The truncation order is fixed to $N = 15$ and the tolerance of ANM(C) is set to $\delta = 10^{-6}$.

Remark 2.3.1 *We note that we have not represented RK4 with the cross symbols as it will produce many points due to the solution for each discrete-time step which can make the figure less clear. It should be noted that in between each dot on the plot, solutions are continuous. This is one of the main features of such an approach.*

Continuation is depicted in Fig.(2.13). It also highlights the accuracy of the continuation of the time series, since it presents the same numerical result as RK4.

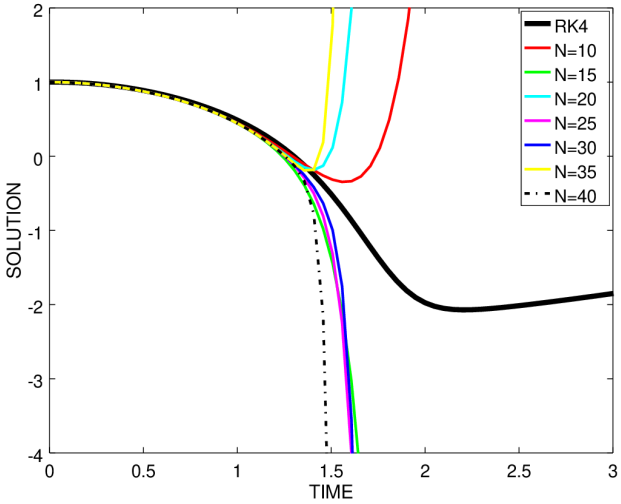


Figure 2.12 – Approximated solution using time perturbation method ANM (series) using different truncation order without continuation, for the free Van der Pol oscillator for $\mu = 2$. In black the reference RK4 solution using $\Delta t = 10^{-3}s$.

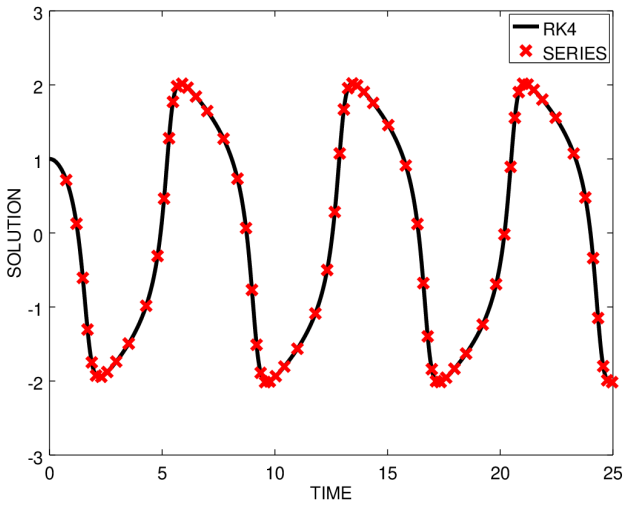


Figure 2.13 – Approximated solution using time perturbation method ANM (series) for $N = 15$ and $\delta = 10^{-6}$, for the free Van der Pol oscillator for $\mu = 2$. In black the reference RK4 solution using $\Delta t = 10^{-3}s$.

Numerical results of BPL

Numerical results of the resummation technique BPL, with the continuation procedure are presented. The Laplace transform is computed using $N_G = 8$ as the number of Gauss

points. The real positive axis is the direction of the Laplace transform $d = \mathbb{R}^+$. Scalar Padé in Eq.(1.35) applied on Borel's terms are chosen such that: M = integer part $((N-1)/2)$ and $L = (N-1) - M$. The residual (Eq.(1.44)) is evaluated using the first and second derivative of the Borel sum presented in Appendix A (strategy 1). The continuation of BPL based on the residual is carried up to $t = 25$s, with a residual tolerance $\epsilon = 10^{-4}$.

It can be observed in Fig.(2.14a) that BPL can reproduce the solutions as RK4. This is further supported by Fig.(2.14b), which shows the ability to reproduce the limit cycle.

Remark 2.3.2 *It is worth noting that the part of the solution shown in Fig.(2.14b) that is outside the limit cycle is not a deviation of the solution, but is just the starting part of the solution before reaching the limit cycle.*

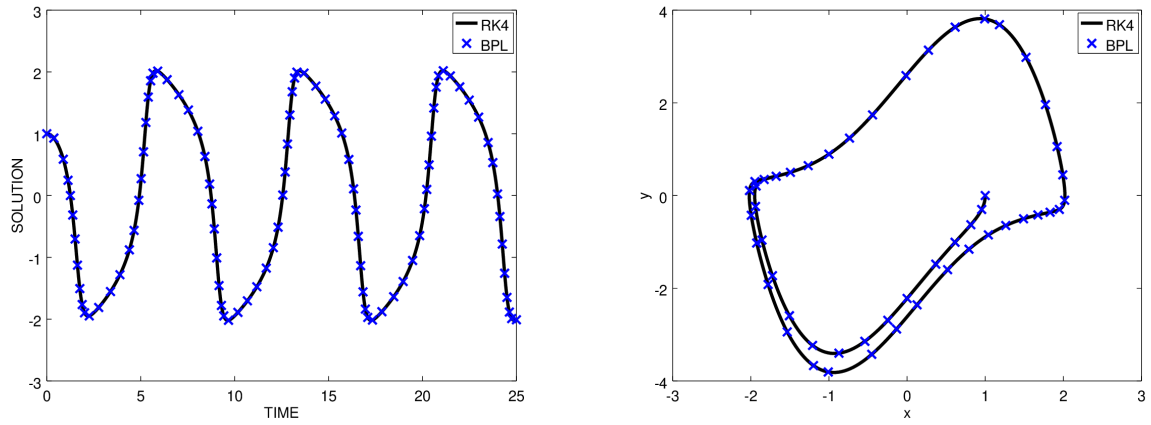
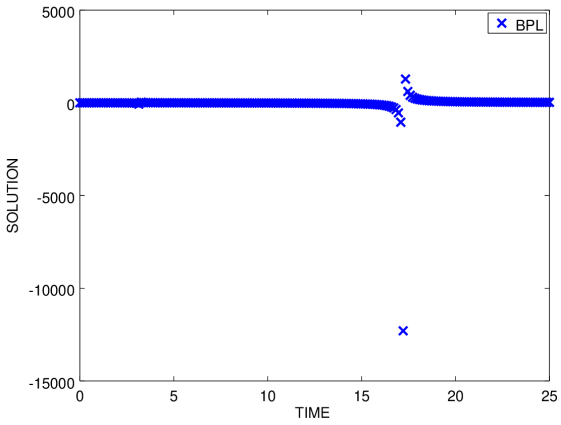
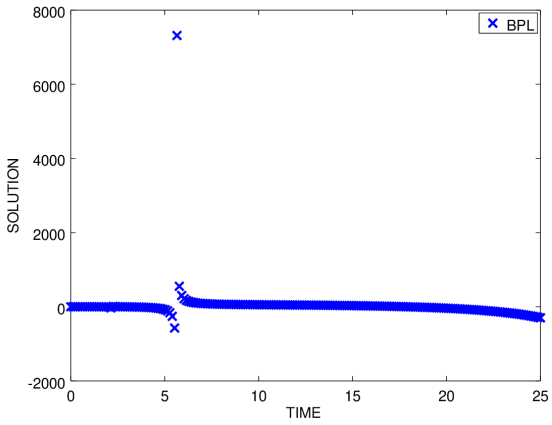


Figure 2.14 – Approximated solution (left) and phase portrait (right) using time perturbation-resummation method BPL for $\epsilon = 10^{-4}$, $N = 15$ for the free Van der Pol oscillator for $\mu = 2$. In black the reference RK4 solution using $\Delta t = 10^{-3}s$.

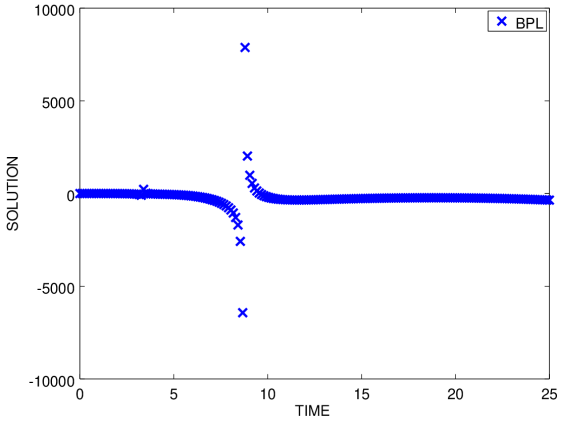
However, in some cases BPL lakes of robustness. For example, the apparition of poles resulting from Padé approximants with different examples of parameter changes are depicted in Fig.(2.15). This might disable efficient continuation procedure.
As already indicated in section 1.3 in Chapter 1 and the previously example, a way to overcome this problem is to use Inverse factorial series resummation or Meijer-G algorithms.



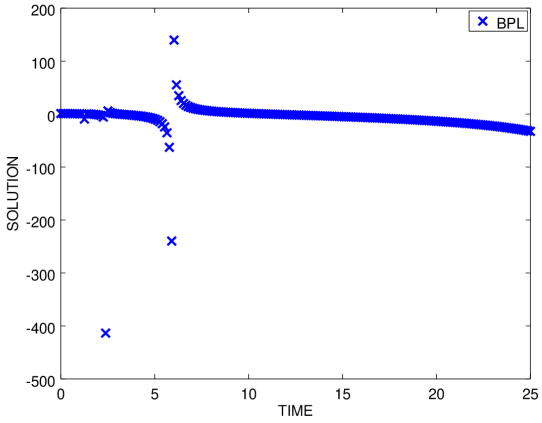
(a) $N = 5, \mu = 2$



(b) $N = 10, \mu = 2$



(c) $N = 35, \mu = 2$



(d) $N = 8, \mu = 2$

Figure 2.15 – Different examples of apparition of poles resulting from Padé approximant of BPL for the free Van der Pol oscillator for $\mu = 2$.

Numerical results of IFS

It is proposed to use IFS in the case of $N = 8, \mu = 2$. Poles appear for BPL method using those parameters (see Fig.(2.15d)).
A comparison of the solution obtained by the perturbation method ANM, resummation techniques BPL and IFS for $N = 8$ without continuation, is displayed in Fig.(2.16). As expected and per construction, no numerical poles are observed for IFS. A continuation procedure using IFS is performed using the residual criterion. The residual (Eq.(1.71)) is based on the first and second derivative of IFS presented in Appendix B. The continuation of IFS based on the residual is carried up to a final time $t = 15$s, using the residual tolerance $\epsilon = 10^{-4}$ and $N = 8$.

Evolution of the IFS solutions are plotted in Fig.(2.17). It can be observed that IFS presents the same numerical solution as RK4. Oscillations with IFS reach also a limit

cycle as it can be appreciated in Fig.(2.17b).

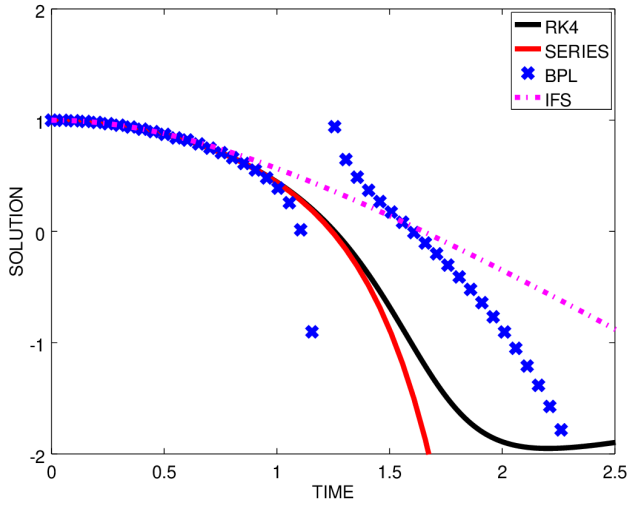


Figure 2.16 – Approximated solution using time perturbation method ANM (series), time perturbation-resummation BPL, and IFS for $N = 8$ without continuation, for the free Van der Pol oscillator for $\mu = 2$. In black the reference RK4 solution using $\Delta t = 10^{-3}s$.

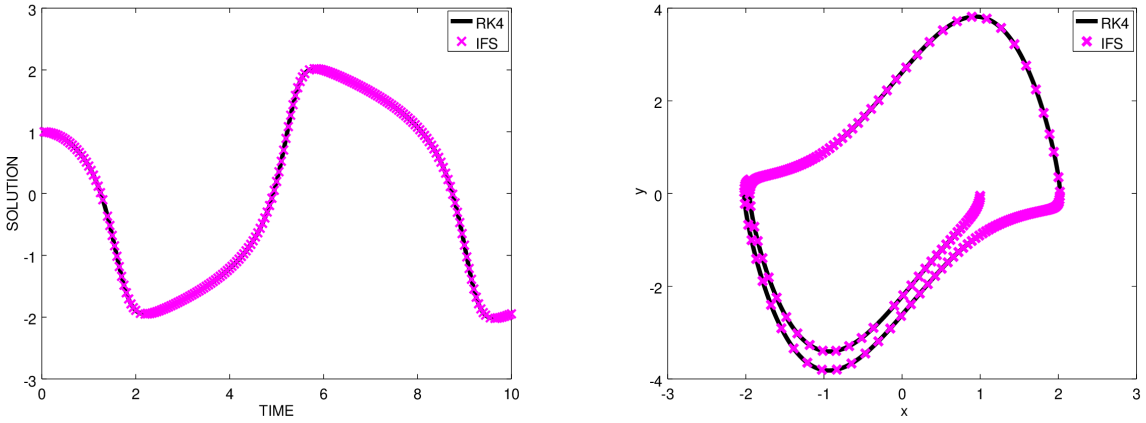


Figure 2.17 – Approximated solution (left) and phase portrait (right) using time perturbation-resummation method IFS for $\epsilon = 10^{-4}$, $N = 8$ for the free Van der Pol oscillator. In black the reference RK4 solution using $\Delta t = 10^{-3}s$.

Simulation over a large time

Finally, the simulations are executed by ANM, BPL, and IFS up to a final time $t = 10000$s. Numerical results show the ability of time perturbation and resummation methods to reproduce the dynamics of the system over a large time. The limit cycle is respected by these three methods even for a large final time. The effiency of the system

dynamic obtained by BPL is presented in Fig.(2.18) for the interval time $[9900; 10000]$s. It should be noted that the long time behavior of BPL is investigated also through numerical experiments on the Hamiltonian and non-Hamiltonian systems in [36]. A second fully detailed article from the same research team is in progress on the stability of the BPL scheme (see https://arxiv.org/pdf/2007.00390.pdf). As IFS used as a time integration scheme has never been applied for long time. This example can be used to illustrate the capability of IFS for long time simulations.

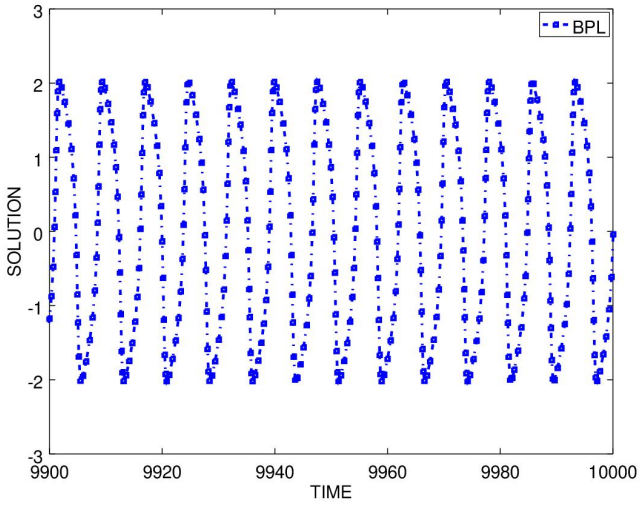


Figure 2.18 – Approximated solution using time perturbation-resummation method BPL in the interval time $[9900; 10000]$s, for the free Van der Pol oscillator for $\mu = 2$.

ANM, BPL and IFS: t_{max} comparisons

We now propose a comparison of ranges of validity t_{max}. For the sake of comparison, ranges of validity are evaluated using the residual criterion.

Evolution of t_{max} obtained using ANM, BPL and IFS is proposed in Fig.(2.19). It is observed an automatic adaptation of the range of validity $[0, t_{max}]$ accordingly to the cyclic solution. Ranges of validity of ANM and BPL oscillate around $0.4s$ and stay bounded in $[0.1, 0.78]s$ and is larger than that of IFS.

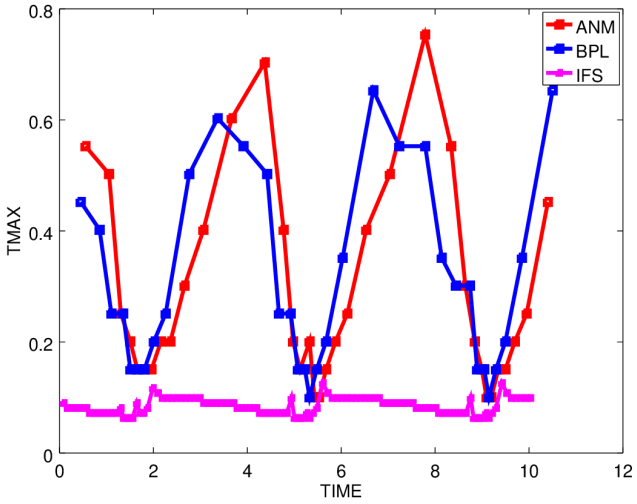


Figure 2.19 – t_{max} evolution for time perturbation method ANM, time perturbation-resummation methods BPL and IFS, for $N = 15$, and $\epsilon = 10^{-4}$ for the free Van der Pol oscillator for $\mu = 2$.

ANM, BPL, and IFS: Number of continuation steps comparisons

We now propose a comparison of the number of continuation steps needed to reach a specific final time. We recall here that for each continuation step, continuous approximated solutions are obtained using time perturbation methods. For the sake of comparison, ranges of validity are evaluated using the residual criterion. As a reminder, RK4 needs 10000 steps to reach a final time of $t = 10s$ using a discrete-constant time step of $\Delta t = 10^{-3}s$.

A small study on number of step is presented in Tab.(2.3) for adaptive RK4 [78]. We recall that adaptive RK4 computes a local error between two solutions obtained after performing two consecutive time steps using step size Δt and another using only a one-time step but with a larger step size $2\Delta t$. The tolerance parameter on local error denoted Δ_0, adjusts the step size.

Δ_0	Steps
10^{-3}	177
10^{-4}	575
10^{-5}	1922
10^{-6}	6026

Table 2.3 – Number of steps required to reach $t = 10$s for the free Van der Pol oscillator for $\mu = 2$, using RK4 adaptive step size with different values of the desire accuracy parameter Δ_0.

The number of steps for ANM, BPL, and IFS with each other is given in Tab.(2.4). It

is noted that ANM and BPL reach the final time using approximately the same number of steps. BPL is slightly better than ANM in this example. IFS performs two to four times more steps than ANM and BPL. Interestingly, these 4 methods need fewer steps than the classic and the adaptive RK4 in Tab.(2.3), while ensuring a high quality of the solution.

Residual tolerance ϵ	ANM	BPL	IFS
10^{-2}	24	21	55
10^{-3}	30	26	79
10^{-4}	36	33	99

Table 2.4 – Number of steps required to reach $t = 10$s for the free Van der Pol oscillator for $\mu = 2$, using time perturbation and resummation methods with $N = 15$ and different values of the residual tolerance ϵ.

Computational efforts

To the best of our knowledge and as already mentioned in Chapter 1, no explicit formula of temporal range of validity (t_{max}) of BPL and IFS has been established yet. Therefore, it is proposed to evaluate it via a residual criterion as in [32]. The latter guarantees a very good quality of the solution which is continuous but sometimes time-consuming because of a large number of numerical residue evaluations.

Some global information on the computational effort is given in this subsection. For this test case, the comparison is only an indication since the Van der Pol oscillator is considered a problem of small size.

We compare the performance of the numerical methods in terms of their computational efficiency (as measured both by computational load i.e the CPU time using here the GNU Octave) and in terms of numerical accuracy (as quantified by evaluating the residual of the computed numerical solution) for a final time of $t = 100s$.

As a reference, RK4 using a constant time step $\Delta t = 10^{-3}s$ needs 7.8709s of CPU time to reach a final time of $t = 100s$. When the RK4 adaptive time step is used with $\Delta_0 = 10^{-6}$, the CPU time needed is 10.258s.

Remark 2.3.3 *We just compare the CPU time given for $\Delta_0 = 10^{-6}$ since this choice of Δ_0 allows the residual of RK4 to stay in the same residual range as those of the perturbation methods.*

CPU times for temporal perturbation approaches are given in Tab.(2.5). ANM(C) with $\delta = 10^{-5}$, is in the same residual ranges as other methods. Complete computation is performed in 0.29s of CPU time. It should be noted that ANM(R) (ANM using residual

norm to evaluate t_{max}) and BPL require more CPU time than ANM(C). It is noted that IFS is relatively less efficient in terms of computational time as can be seen in Tab.(2.5). Interestingly, ANM(C), ANM(R), BPL, and IFS are faster than RK4 adaptive and RK4 with constant step size. Therefore, the results show that ANM(C) with $\delta = 10^{-5}$ is the fastest in CPU time while the residual stays in the same range as others methods. ANM(C) is 4 time faster than ANM(R), however, ANM(R), BPL, and IFS ensure a prescribed residual tolerance. Moreover, most of the CPU time is spent in the evaluation of the range of validity.

Method	CPU time (s)
ANM(C)	0.29951
ANM(R)	1.3056
BPL(R)	3.9281
IFS(R)	7.0321
Adaptive RK4	10.258

Table 2.5 – CPU time needed to reach $t = 100$s using the RK4 adaptive time step for a local error tolerance $\Delta_0 = 10^{-6}$, time perturbation method ANM, time perturbation-resummation methods BPL and IFS for a truncation order $N = 8$. (R) stands for ranges of validity evaluated using a residual tolerance criterion $\epsilon = 10^{-3}$, while (C) stands for the criterion proposed by Cochelin [40] using $\delta = 10^{-5}$.

Finally as conclusion regarding computational efforts and the number of steps for adaptive RK4 and time perturbation methods, it is possible to say that wider step sizes are performed, in a slightly better amount of time and for a better-prescribed accuracy of the solution.

Remark 2.3.4 *We note that only one of the methods will be displayed on the figures for the remaining test cases of this chapter (for clarity). However, we emphasize that we have investigated the efficiency of all methods: ANM, BPL, and IFS.*

2.3.5 Behavior for a higher value of μ

In this subsection, we consider the same equation of the free Van der Pol oscillator by increasing the non-linearity coefficient μ to 1000. In this particular case, the solution changes radically and presents oscillations on a much longer time scale.
Goal of this example: We are interested to see how the polynomial approaches can deal with large values of the nonlinearity coefficient where the equation becomes stiff and some very fast dynamic in the system is expected.

Fig.(2.20) displays the results of the solution obtained by time perturbation method ANM and RK4 for a discrete-time step $\Delta t = 10^{-4}$s, for a final time $t = 6000$s. One can

see that the solutions are superposed. A zoom of the plot close to the area of fast changes of the solution is presented in Fig.(2.21). It is observed that the ANM solution is different than the one obtained by RK4 with a discrete-time step $\Delta t = 10^{-4}$s. For this reason, we decrease the step size to $\Delta t = 10^{-5}$s. With this latter, RK4 solution is now in a good agreement with ANM solution as can be appreciated in Fig.(2.21).

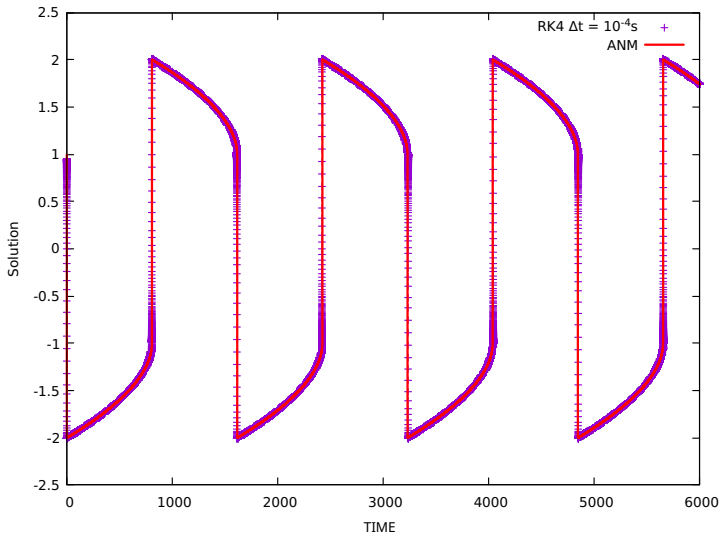


Figure 2.20 – Approximated solution using time perturbation method ANM for a truncation order $N = 15$ and RK4 with a discrete-time step $\Delta t = 10^{-4}s$, for the free Van der Pol oscillator with $\mu = 1000$.

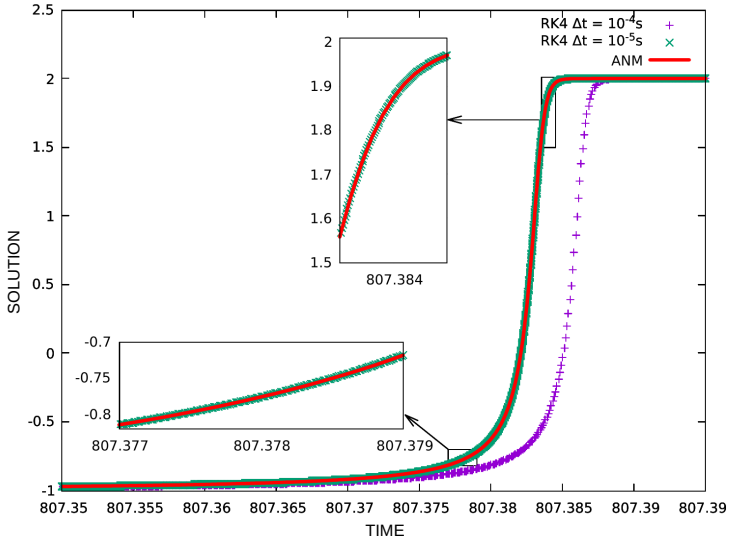


Figure 2.21 – Enlarged view of Fig.(2.20) of the approximated solution using time perturbation method ANM and RK4 with two discrete-time steps $\Delta t = 10^{-4}s$ and $\Delta t = 10^{-5}s$, for the free Van der Pol oscillator with $\mu = 1000$.

An additional plot of the velocity in that vicinity of the "jumps" that can give an additional insight in the quality of the approximation is presented in Fig.(2.22).

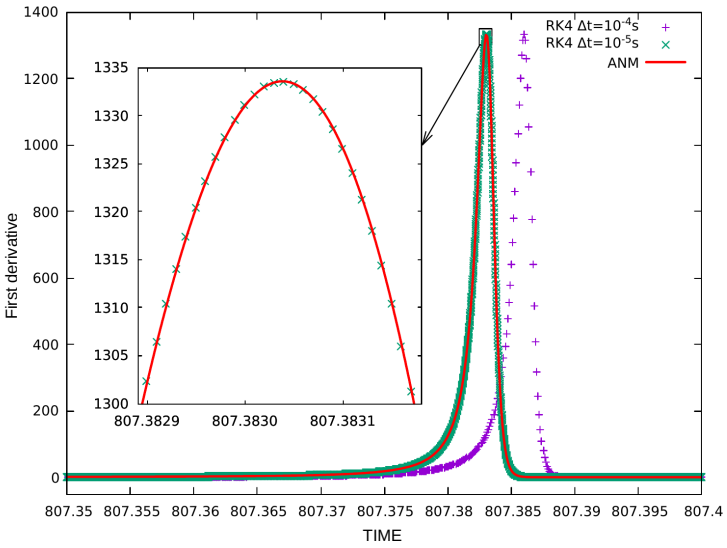
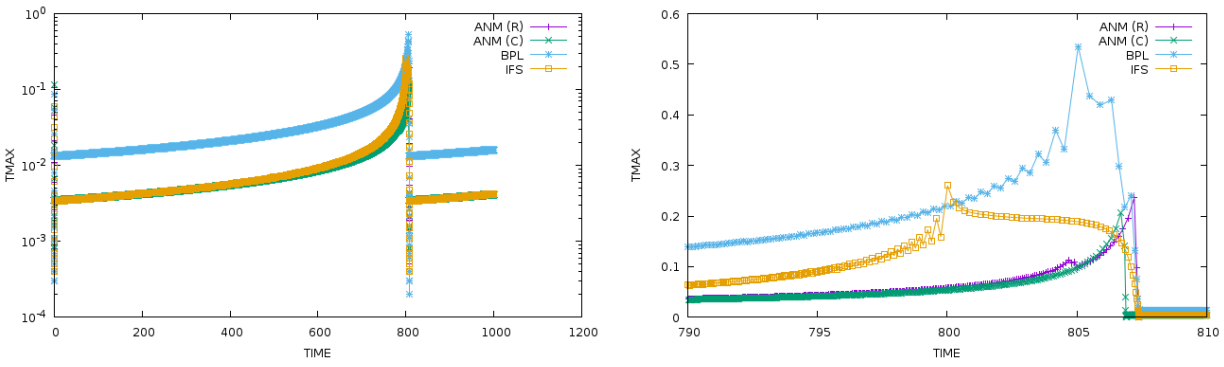


Figure 2.22 – Zoom plot of the approximated velocity using time perturbation method ANM and RK4 with a discrete-time step $\Delta t = 10^{-5}s$, for the free Van der Pol oscillator with $\mu = 1000$.

The same results are obtained with the other time perturbation-resummation methods BPL and IFS presented in this work.

We are now interested in comparing the range of validity of these time perturbation and resummation methods. The comparison is carried out in Fig.(2.23) for a truncation order $N = 25$.



(a) (b)

Figure 2.23 – t_{max} evolution for ANM(R), ANM(C), BPL and IFS for $N = 25$, for the free Van der Pol oscillator with $\mu = 1000$. (R) stands for ranges of validity evaluated using a residual tolerance criterion $\epsilon = 10^{-4}$, while (C) stands for the criterion proposed by Cochelin [40] using $\delta = 10^{-4}$. (b) is a zoom plot of (a).

It is observed that the use of the resummation methods BPL and IFS tend to increase the range of validity of the ANM series in this case. It can also be seen that the validity

domain of the four methods is larger than the discrete-time step of RK4 which is 10^{-5}. Regarding the adaptive time step of RK4, and as mentioned in [79], if we integrate a stiff problem using adaptive Runge-Kutta, the initial step size chosen generally leads to a large local error estimate. This then causes the routine to reduce the step size until the main local error is reduced to its specified limit. The routine sometimes works well to integrate the problem, but uses a much larger number of steps than seems logical. As a result, rounding error and computation time are constraints when using classical techniques to integrate such problems.
These constraints are surmounted with the time perturbation and resummation methods. This example is a very good illustration to realize that these methods seems to be appropriate for stiff problems.

2.4 Application to the forced Van der Pol oscillator

An external force can be simulated as acting on the system by adding a new periodic force to the second member of the Van der Pol equation [80]. This forced equation is studied in some papers [81]–[86], and is in the following form:

$$\ddot{x} - \mu(1 - x^2)\dot{x} + x = F(t) \tag{2.18}$$

This equation is called forced Van der Pol.
The external action will be assumed sinusoidal periodic force, therefore, the second member $F(t)$ is chosen as $A\cos(\omega t)$, where A denotes the forcing amplitude and ω the forcing frequency.
Goal of this example: While we use time perturbation for the unknowns, the right-hand side (RHS) $F(t)$ also needs to be represented by a power series. To obtain the best numerical solution of the complete forced equation Eq.(2.18), it is necessary to have the best approximation of this RHS. Therefore, a time perturbation study of this RHS is realized by comparing the Taylor method and time perturbation methods using specific change of variables [39] based on quadratic recast.
It is interesting to note that there could be differences between these two methods in the sense of the range of validity due to the truncation order. One has to find out which of these two methods has a larger range of validity to reach the final time.

2.4.1 Time perturbation of the forcing term

The chosen time-dependent forcing term is $F(t) = A\cos(\omega t)$. It is necessary to look for a polynomial approximation of it, as:

$$F(\hat{t}) = F_0 + \sum_{i=1}^{N} F_i \hat{t}^i \tag{2.19}$$

with $F_0 = F(t_0)$. Two specific procedures are compared to evaluate the F_i series terms: Variable changes and Taylor decomposition.

Variable change:

The following change of variables $U = \cos(\omega t)$ and $V = \sin(\omega t)$ proposed in [39] leads to

$$\left\{ \begin{array}{l} U = \cos(\omega t) \\ V = \sin(\omega t) \end{array} \right. \Longrightarrow \left\{ \begin{array}{rcl} dU & = & -\omega V \ dt \\ dV & = & \omega U \ dt \end{array} \right. \tag{2.20}$$

Writing U and V as series: $U = U_0 + \sum_{i=1}^{N} U_i \hat{t}^i$ and $V = V_0 + \sum_{i=1}^{N} V_i \hat{t}^i$, where $U_0 = \cos(\omega t_0)$ and $V_0 = \sin(\omega t_0)$ are the initial terms with t_0 a given initial time.
Injecting the latter into Eq.(2.20), and then equating terms with identical powers of $\hat{t}$, we get for $i \geq 1$:

$$\left\{ \begin{array}{rcl} U_i & = & -\frac{\omega}{i} V_{i-1} \\ V_i & = & \frac{\omega}{i} U_{i-1} \end{array} \right. \tag{2.21}$$

Thus, the recurrence formula for the RHS, $\forall i \geq 0$, is:

$$F_i = AU_i \tag{2.22}$$

Taylor decomposition:

The general formula of Taylor series for $\cos(\omega t)$ also makes it possible to determine the F_i series terms:

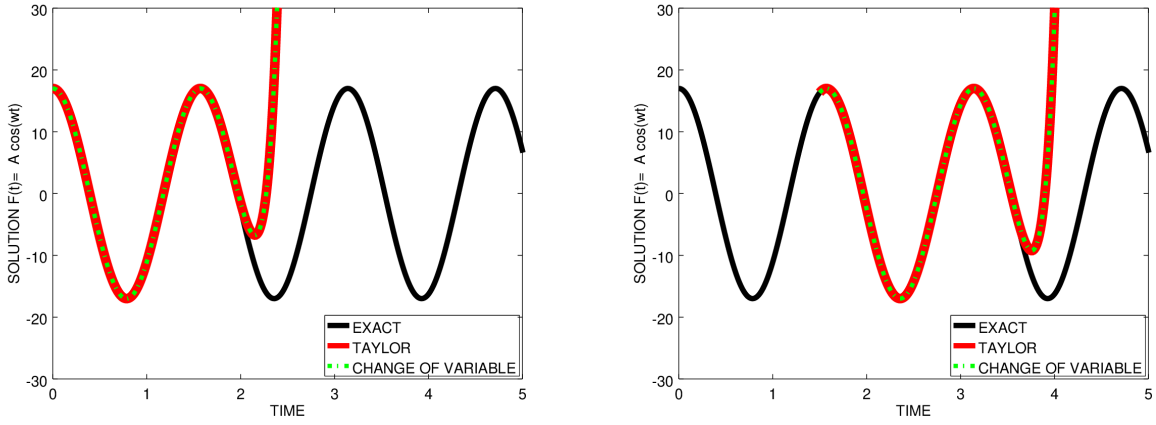
$$\cos(\omega t) = \sum_{i=0}^{N} \frac{(\cos(\omega t_0))^{(i)}}{i!} (t - t_0)^i \tag{2.23}$$

where $(\cos(\omega t_0))^{(i)}$ denotes the ith derivative of the cosine function evaluated at the point (ωt_0). This formula has been directly used for numerical validation. The recurrence formula for the F_i terms is not provided here as it was not necessary.

2.4.2 Numerical validation of the RHS series representation

The exact solution is compared to the solution given by the Taylor method and the change of variable method, for $A = 17, \omega = 4$. It is done without a continuation procedure to fully understand the behavior of the obtained series. Firstly, influence of the initial time t_0 is depicted in Fig.(2.24) using truncation order $N = 20$. No difference can be seen between both methods, nor influence of the initial time. Secondly, truncation order influence is presented in Fig.(2.25). It can be seen that the domain of validity t_{max} of the RHS, $\approx 1.5s$, is wider than the previously computed branch of solutions.

Numerical results without continuation procedure show that the two methods: the Taylor formula and the method based on change of variable, have the same behavior, the same range of validity, and the same error with the exact solution. Since the proposed method leads to a much simpler recurrence formula Eq.(2.22) for evaluating the F_i terms, we decide to use this method for the RHS.



(a) Initial time $t_0 = 0$ (b) Initial time $t_0 = 1.5$

Figure 2.24 – Influence of the initial time t_0 for the time perturbed RHS: $F(t) = A\cos(\omega t)$ without continuation procedure. Comparisons with the exact solution (plain black curve) are performed for $A = 17$, $\omega = 4$ and a truncation order of $N = 20$.

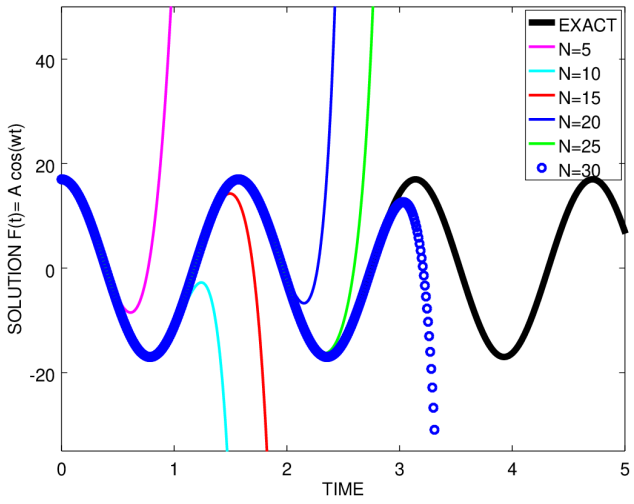


Figure 2.25 – Influence of the truncation order for the time perturbed RHS $F(t) = A\cos(\omega t)$ without continuation procedure using the introduction of the new variables. Comparisons with the exact solution (plain black curve) are performed for different truncation orders N, for $A = 17$, $\omega = 4$ and $t_0 = 0$.

2.4.3 Numerical study of the forced Van der Pol equation

Classical discretization schemes

The forced Van der Pol equation Eq.(2.18) is solved by converting it into a first order system of differential equation:

$$\begin{cases} \dot{x} &= y \\ \dot{y} &= \mu(1-x^2)\dot{x} - x + F(t) \end{cases} \tag{2.24}$$

A convergence study for different time steps Δt between RK4 and RK1 schemes has been also carried out in this case. The corresponding time step for $x(0) = 1, \dot{x}(0) = 0$ and $\mu = 2$ is $\Delta t = 10^{-3}s$.

Time perturbation representation

To solve the forced Van der Pol equation using time perturbation, we proceed with the same procedure as in the free case. Power series representation of the unknowns Eq.(2.11), using time as the perturbation parameter, is injected in the equation to solve Eq.(2.18). Moreover, series terms F_i of the RHS are evaluated using Eq.(2.22). Therefore, by identifying the terms according to the powers of $\hat{t}$, the following recurrence formula is

obtained:

$$k(k-1)x_k = \mu\left((k-1)x_{k-1} - \sum_{i=0}^{k-2}(k-1-i)\,x_{k-1-i}\;z_i\right) - x_{k-2} + F_{k-2} \tag{2.25}$$

where

$$z_i = \sum_{j=0}^{i} x_j x_{i-j} \tag{2.26}$$

Numerical results

For all the computations, initials conditions are set to $x(0) = 1, \dot{x}(0) = 0$, the amplitude to $A = 17$ and the frequency to $\omega = 4$.

To understand the behavior of the ANM, solutions are plotted without continuation procedure in Fig.(2.26). It can be seen as a standard evolution in the perturbation method. After some time, continuous solutions diverge. Moreover, the more terms added in the representation, the wider is the range of validity.

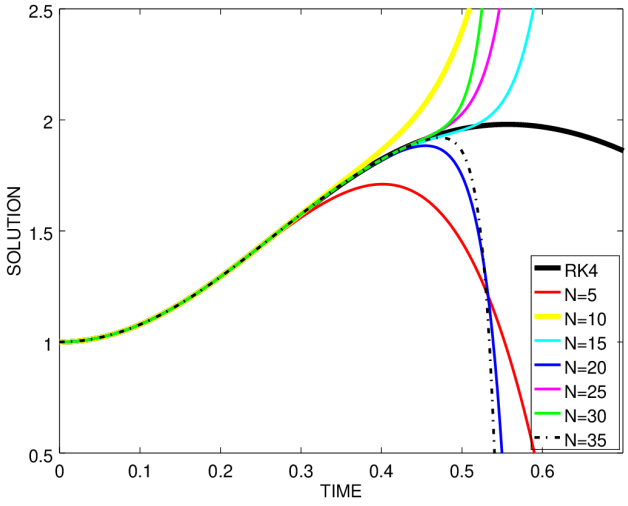
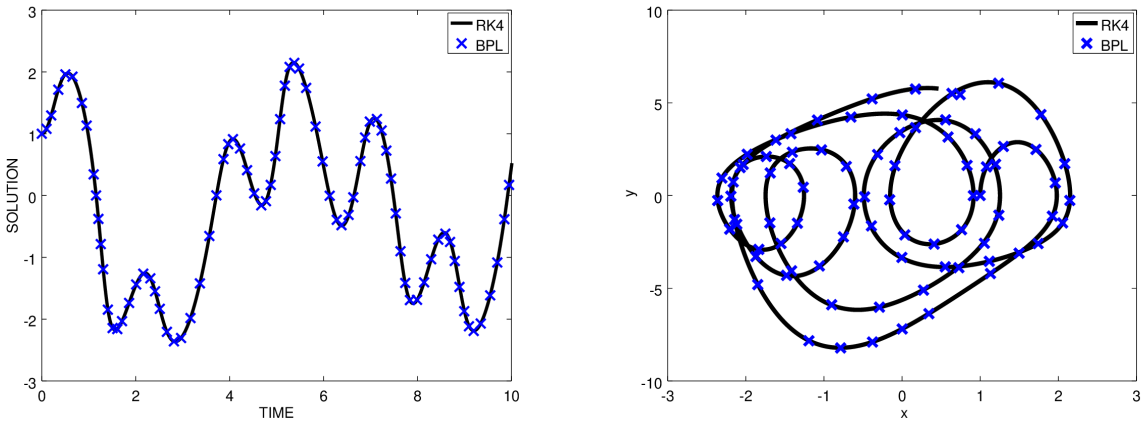


Figure 2.26 – Approximated solution using time perturbation method ANM (series) for different truncation orders N, in the case of the forced Van der Pol oscillator for $\mu = 2$, $\omega = 4$ and $A = 17$.

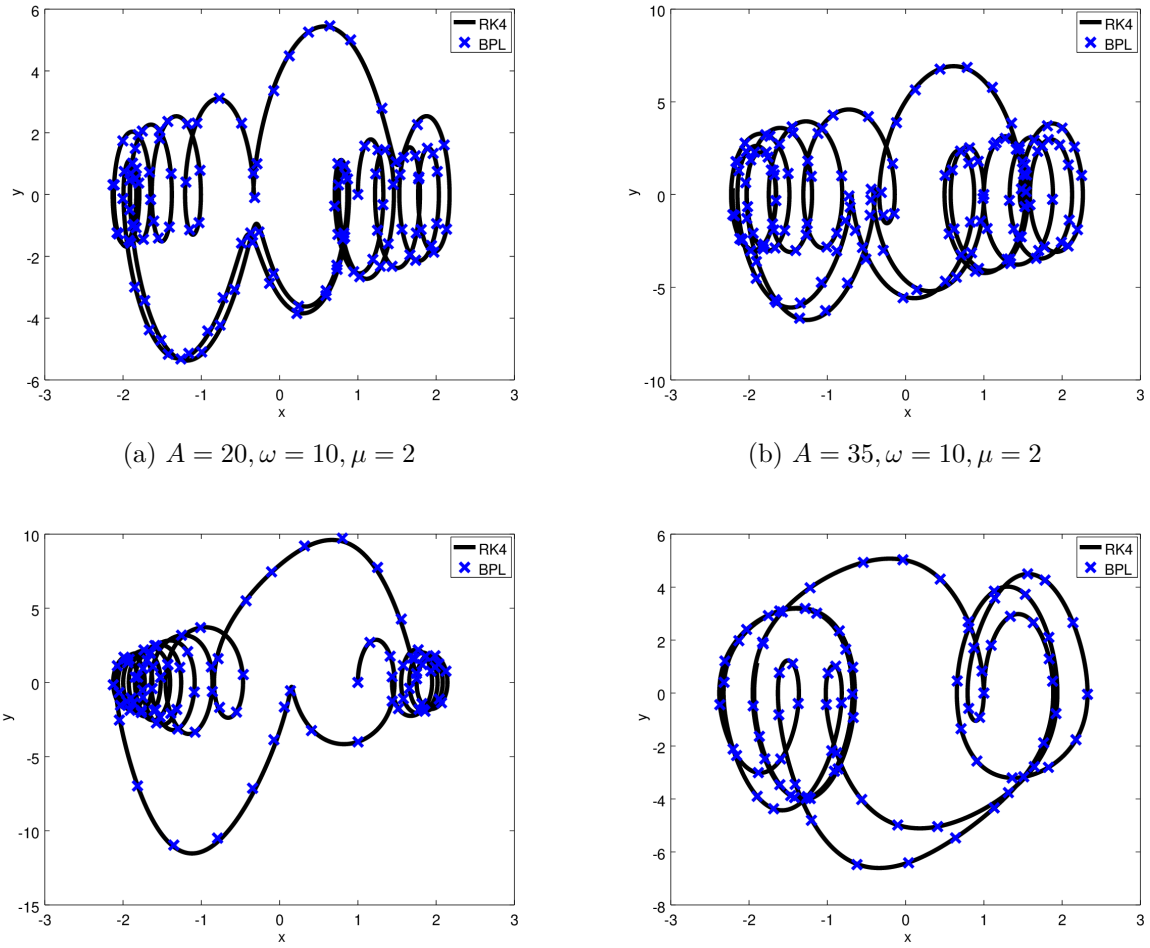
Solutions and phase portrait are proposed in Fig.(2.27). The continuation procedure is the same as in the free case. Residual norm tolerance is set to an ϵ value, this makes it possible to evaluate the range of validity of the time perturbation representation t_{max}. The same results are obtained using IFS and ANM. They are not plotted here for sake of conciseness. Hence, it is confirmed that ANM, BPL, and IFS are able to reproduce solutions of the forced Van der Pol equation.



(a) (b)

Figure 2.27 – Approximated solution (left) and phase portrait (right) using time perturbation resummation method BPL for $\epsilon = 10^{-4}$, $N = 15$ in the case of the forced Van der Pol oscillator for $\mu = 2$, $\omega = 4$ and $A = 17$. In black the reference RK4 solution using $\Delta t = 10^{-3}s$.

A small robustness study is proposed for the time perturbation methods. Phase portraits are plotted in Fig.(2.28) for BPL only (the same result is obtained by ANM and IFS methods). It is done for others values of the forced Van der Pol equation parameters A, ω, and μ. Perfect agreement with the RK4 solutions is observed. This confirms the ability of the proposed methods to reproduce the numerical solutions of the forced Van der Pol oscillator.



(a) $A = 20, \omega = 10, \mu = 2$

(b) $A = 35, \omega = 10, \mu = 2$

(c) $A = 35, \omega = 10, \mu = 6$

(d) $A = 20, \omega = 5, \mu = 2$

Figure 2.28 – Phase portrait using time perturbation resummation method BPL for $\epsilon = 10^{-4}$, $N = 15$, in the case of the forced Van der Pol oscillator, for different values of A, ω and μ. In black the reference RK4 solution using $\Delta t = 10^{-3}s$.

Remark 2.4.1 *The proposed time perturbation methods have also been successfully applied to the forced Van der Pol oscillator with two degrees of freedom. The example considered concerns a triode generator with an additional circuit which is the best-known example of a self-oscillatory system with two degrees of freedom. As it does not provide any new results, we made the choice not to add extra redundant information in the manuscript. Another model with three degrees of freedom has been studied in section 2.7 and the more next chapters of the manuscript.*

2.5 Application to the nonlinear combustion equation

This example is studied by Larry Shampine, one of the authors of MATLAB's suite of ordinary differential equations [87]. This problem is characterized by its transitory aspect in the middle of the integration interval. The solution switches from non-stiff to stiff, and then it becomes non-stiff again.

Goal of this example: As numerical drift is known to occur in such cases, we are interested in studying the performance of time perturbation and resummation methods in this example.

We consider a model of flame propagation. If a match is lit, the fireball spreads rapidly until it reaches a critical size. It then preserves this size because the amount of oxygen absorbed by the combustion inside the ball balances the amount accessible through the surface of the fireball. To illustrate this further, let us consider the following nonlinear combustion equation [88]:

$$\dot{y} = y^2(1 - y) \tag{2.27}$$

where $\dot{y} = \dfrac{dy}{dt}$, with the initial condition

$$y(0) = R \tag{2.28}$$

The scalar variable $y(t)$ represents the radius of the ball. The terms y^2 and y^3 refer to surface area and volume. The critical parameter R corresponds to the initial radius which is small. The solution is sought over a length of time such that $t \in [0, 2/R]$.

The exact solution of this differential equation is [89]:

$$y(t) = \frac{1}{W(a\exp(a - t)) + 1} \tag{2.29}$$

where $a = 1/R - 1$ and $W(t)$ is the Lambert W function (which is the inverse of the function: $t\exp(t)$).

2.5.1 Time perturbation series recurrence formula

Series terms of time perturbation method for equation Eq.(2.27) are given by the following recurrence formulas, $\forall k = 1, \cdots, N$ we have:

$$ky_k = -\sum_{i=0}^{k-1} z_i y_{k-1-i} \tag{2.30}$$

$$z_i = \sum_{j=0}^{i} y_j y_{i-j} \tag{2.31}$$

2.5.2 Numerical results

For the numerical test, we take $R = 0.0001$. We provide the solution of the combustion equation by time perturbation methods and MATLAB ode solvers.
We start by ode45 (based on an explicit Runge–Kutta (4, 5) formula and the Dormand-Prince pair [90]) since in general, it is the best solver to apply as a first try for the majority of problems.

Fig.(2.29) shows the exact solution compared to the solution obtained by the time perturbation method and the MATLAB solver ode45 with the zoom detail, for a final time equal to $2/R$, which is $t = 20000$s. One can see that the solution obtained by the solver ode45 gives oscillatory motion, while the ANM solution is in very good agreement with the exact one. The same result as ANM is obtained by BPL and IFS.

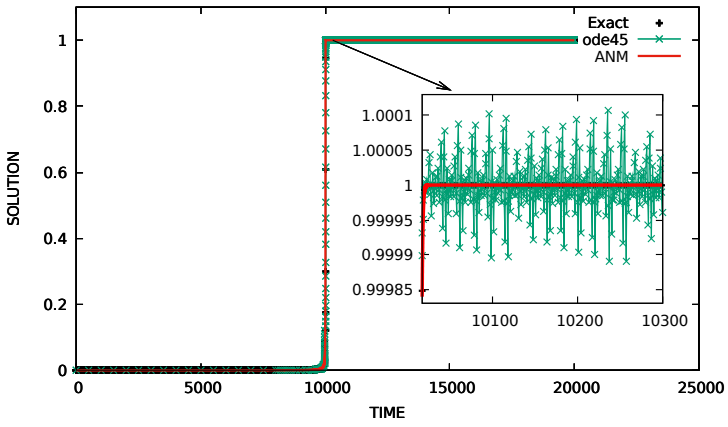


Figure 2.29 – Approximated solution of Eq.(2.27) obtained by time perturbation method ANM and ode45 for $\delta = 0.0001$ compared to the exact solution.

Note that good results can be obtained with MATLAB solver ode23s (based on a modified Rosenbrock formula of order 2 [87]). However, it only solves some kinds of stiff

problems and none of all ode solvers is exempt from the step size restriction. Moreover, as temporal perturbation methods are explicit methods and explicit schemes are known to present some difficulties when solving stiff problems, we have shown that time perturbation-resummation methods solves well this kind of problem.
Thus, the qualitative characteristics of the solution obtained by the time perturbation method are shown as well as the good performance of the solution up to a large final time ($t = 20000$s). This was proven by evaluating the error between the exact solution and the one obtained by the time perturbation method over time which is of order 10^{-5}.

2.6 Application to the Elastic hardening spring problem

Consider the nonlinear elastic hardening spring problem [91], [92] in the following form:

$$\ddot{u} + s_1(1 + s_2u^2)u = 0 \tag{2.32}$$

where $\dot{u} = \dfrac{du}{dt}$ and $\ddot{u} = \dfrac{d^2u}{dt^2}$, with the following initials conditions

$$\begin{cases} u(0) &= u_0 \\ \dot{u}(0) &= v_0 \end{cases} \tag{2.33}$$

where $s_1 > 0$ and $s_2 > 0$.
Goal of this example: This problem is well-known for being a great example for evaluating the performance of new temporal integration techniques as mentioned in [92]. It is an example of conservative systems which maintain constant total energy in their exact solutions. The solution to this problem presents periodic oscillations.

The exact period and solution can be found analytically in [93].

2.6.1 Time perturbation series recurrence formula

Series terms of time perturbation method for Eq.(2.32) are given by the following recurrence formulas, $\forall k = 1, \cdots, N$ we have:

$$k(k-1)u_k = -s_1\left(\sum_{i=0}^{k-2}(1 + s_2y_i)u_{k-2-i}\right) \tag{2.34}$$

$$y_i = \sum_{j=0}^{i} u_j u_{i-j} \tag{2.35}$$

2.6.2 Numerical results

For numerical results, we take $u_0 = 1.5$, $v_0 = 0.0$, $s_1 = 100$, $s_2 = 10$ as presented in [92]. The nonlinear period of the problem is $T = 0.15153$. Classic time integration schemes use a discrete-time step $\Delta T = T/32$ as in [92].
Solutions and phase portraits of the reference solution and the one obtained by the time perturbation method (ANM) are given in Fig.(2.30) and Fig.(2.31) (the same result is obtained by BPL and IFS). It is seen that good agreement is obtained. However, results obtained with classical discrete-time step Δt show the inaccuracy of the numerical solution. In conservative systems, the exact displacement-velocity portrait should be a closed cycle which is not satisfied by the classical integration schemes Runge-Kutta of third (RK3) and fourth-order (RK4) as can be deduced from Fig.(2.31). Note that similar results are obtained by using the following classical schemes: Newmark, the Houbolt method, α-method, the average acceleration method, the central difference method, and the Wilson θ-method. This has been confirmed by the deviation of the total energy obtained by these schemes from its initial value in [91].

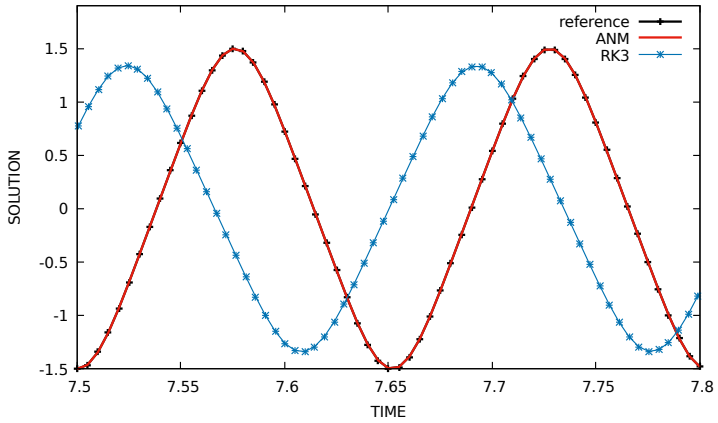
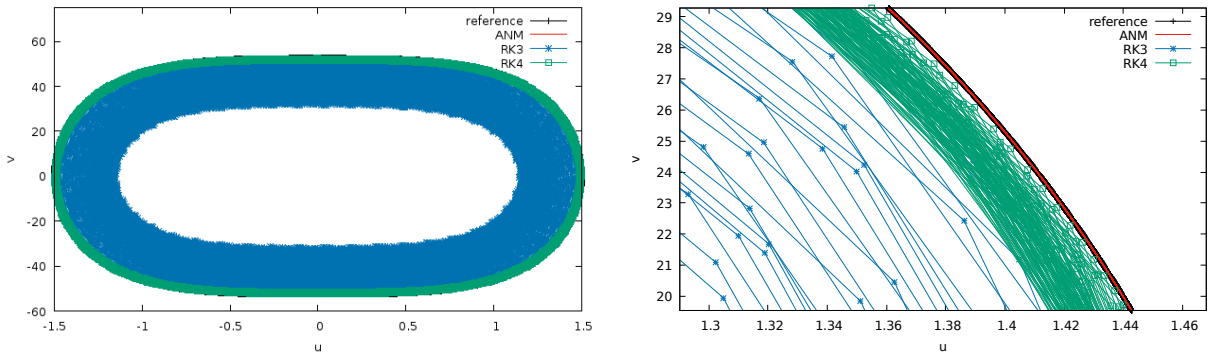


Figure 2.30 – Approximated solutions of the hardening spring problem for a final time equal to $50T$, by time perturbation method ANM and the third and fourth-order Runge-Kutta method (RK3/RK4) with a discrete-time step $\Delta t = T/32$. Reference solution is presented in [93] and is in black.



(a) (b)

Figure 2.31 – Phase portraits of the hardening spring problem for a final time equal to $50T$, by time perturbation method ANM and the third and fourth-order Runge-Kutta method (RK3/RK4) with a discrete-time, step $\Delta t = T/32$. The reference solution is presented in [93] and is in black. (b) is a zoomed plot of (a).

2.7 Application to the three degrees of freedom (3 DOF) example: Lorenz system

Goal of this example: A three-degree-of-freedom nonlinear ODE problem (Lorenz system) is used to verify the performance of the proposed methods for multi-degree-of-freedom problems. The ability of time perturbation techniques to follow the chaotic and non chaotic behavior of the Lorenz system is examined

One of the most famous chaotic systems was developed by Edward Lorenz [94] who was interested in the fluid flow patterns of the Earth's atmosphere and noticed an unexpected chaotic behavior.
Depending on the choice of the parameters that exist in the equations, the Lorenz system can exhibit chaotic and nonchaotic behavior. The Lorenz equations govern, in a lower order, the dynamics of convection in a heated fluid layer and present particular challenges due to its high sensitivity to small variations in initial conditions, from continuous convection to low turbulence (chaos). The Lorenz equations are:

$$\dot{x} = \sigma(y - x) \tag{2.36}$$

$$\dot{y} = x(\rho - z) - y \tag{2.37}$$

$$\dot{z} = xy - \beta z \tag{2.38}$$

where x, y, and z are dynamical variables, with the following initials conditions

$$\begin{cases} x(0) &= x_0 \\ y(0) &= y_0 \\ z(0) &= z_0 \end{cases} \tag{2.39}$$

The first component of the solution x is related to the convection in the atmospheric flow, while the other two components y and z are related to horizontal and vertical temperature variation. The parameters σ, ρ, and β are the related constants.
Note that this system of differential equations is not linear and it is very complicated to obtain exact solutions with the presence of two nonlinear terms.
Bifurcation studies show that with the following parameters $\sigma = 10$ and $\beta = 8/3$, the critical chaos parameter is $\rho = \rho_{cr} = 27.74$ [95]. The behavior of the solution is non-chaotic when $\rho < \rho_{cr}$ and chaotic otherwise.
Solutions are completely dependent on initial parameters and conditions and it is very difficult to predict their behavior. For some parameter values, the orbit of the solution in the three-dimensional space is a strange attractor. For other values of the parameters, the solution tends either to converge towards a fixed point or diverge to infinity or follow a periodic oscillation motion.

When resolving this system numerically, the numerical methods provide the solutions only at a discrete point in time. To ensure a good convergence of the solution, these methods require a very small discrete-time step. However, even by reducing the integration step Δt, the result can not be improved. This is because the integration error presents an extremum as a function of Δt as mentioned in [8]. The importance of handling the Lorenz system with a time perturbation study is outlined also in [8].

2.7.1 Time perturbation series recurrence formula

Series terms of time perturbation method for this system are given by the following recurrence formulas, $\forall k = 1, \cdots, N$ we have:

$$x_k = \frac{1}{k}\sigma\left(y_{k-1} - x_{k-1}\right) \tag{2.40}$$

$$y_k = \frac{1}{k}\left(\rho x_{k-1} - \sum_{i=0}^{k-1} x_i z_{k-1-i} - y_{k-1}\right) \tag{2.41}$$

$$z_k = \frac{1}{k}\left(\sum_{i=0}^{k-1} x_i y_{k-1-i} - \beta z_{k-1}\right) \tag{2.42}$$

2.7.2 Numerical results

For the numerical results, the following initials conditions are set first to $x_0 = 5, y_0 = 5, z_0 = 400$ and the constant ρ is set to $\rho = 350$. Fig.(2.32) illustrates the phase portrait of Lorenz system and shows the efficiency of time perturbation method.
Now, we consider $x_0 = 5, y_0 = 5, z_0 = 4$ and $\rho = 27$, Fig.(2.33) shows the ability of time perturbation methods to reproduce the two well-positioned attractors.

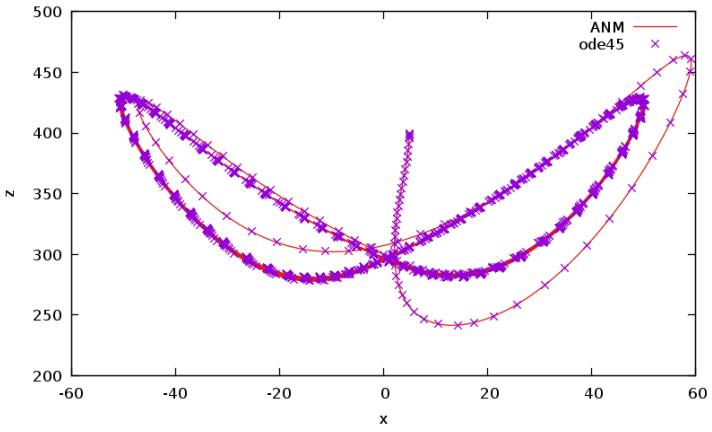


Figure 2.32 – Phase portraits of Lorenz system with $x_0 = 5, y_0 = 5, z_0 = 400$ and $\rho = 350$ obtained by time perturbation method ANM and ode45.

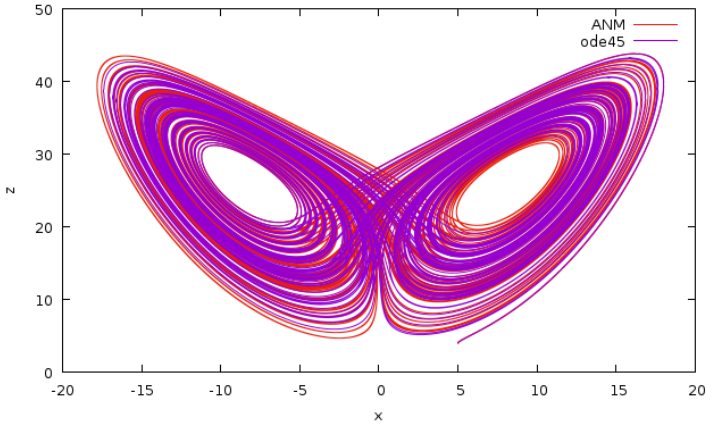


Figure 2.33 – Phase portraits of Lorenz system with $x_0 = 5, y_0 = 5, z_0 = 4$ and $\rho = 27$ obtained by time perturbation method ANM and ode45.

.

2.8 Application to the nonlinear single pendulum problem

It is well-known that time integration methods can perform very different results in nonlinear cases. For this reason, the temporal perturbation and numerical resummation methods considered as time integration schemes are applied on the nonlinear oscillating pendulum.

Goal of this example: Since different information about the motions of the pendulum can be precisely evaluated using given initial conditions, it is well recognized that this problem is an appropriate example for testing the performance of temporal integration schemes.

The equation of motion modeling the free, undamped simple pendulum is:

$$\ddot{\theta} + {\omega_0}^2 \sin(\theta) = 0 \tag{2.43}$$

where θ is the angle between the rigid rod and the vertical line at time t, and ω_0 is defined as

$$\omega_0 = \sqrt{\frac{g}{l}} \tag{2.44}$$

where l denotes the length of the pendulum and g the acceleration due to the gravity. Note that equation (2.43) is nonlinear, due to the presence of $\sin(\theta)$.

The oscillations of the pendulum are subjected to the initial conditions:

$$\begin{cases} \theta(0) &= \theta_0 \\ \dot{\theta}(0) &= \dot{\theta}_0 \end{cases} \tag{2.45}$$

where θ_0 denotes the amplitude of oscillations.
The energy in the system is conservative and can be expressed as:

$$\frac{1}{2}\dot{\theta}^2 - \omega^2 \cos(\theta) = constant \tag{2.46}$$

Fig.(2.34) describes the nonlinear single pendulum problem for two cases of initials conditions to illustrate two different motions of the single pendulum.

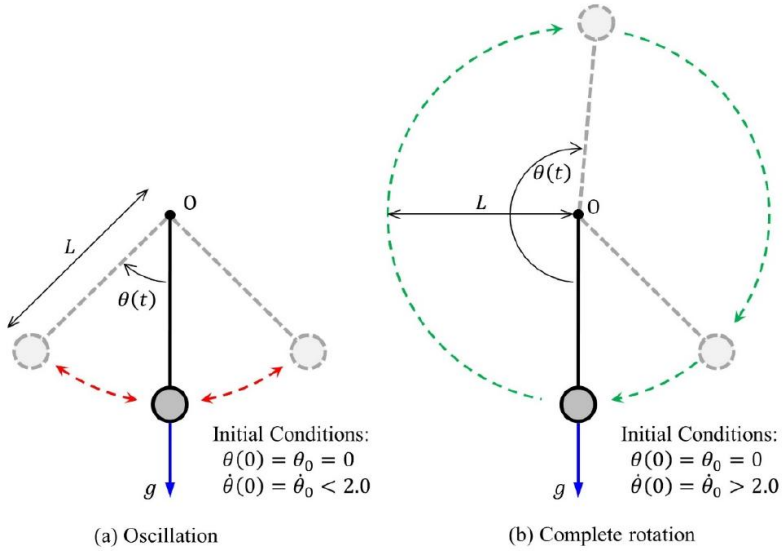


Figure 2.34 – Figure extracted from [92] describing the nonlinear single pendulum problem.

2.8.1 Time perturbation series recurrence formula

When we write $\theta(t)$ as time power series, replace the obtained expansions in the initial equation Eq.(2.43) and identify according to the power of $\hat{t}$, we notice that it is hard to find the recurrence formula that allows to calculate the series term $\theta_k, \forall k = 1, \cdots, N$. For this reason, let the following change of variables

$$U = \cos(\theta) \tag{2.47}$$

$$V = \sin(\theta) \tag{2.48}$$

$$F = {\omega_0}^2 \sin(\theta) \tag{2.49}$$

that lead to the following recurrence formulas of series terms. Therefore, $\forall k = 1, \cdots, N$ we have:

$$U_k = \frac{1}{k} \sum_{j=0}^{k-1} V_j F_{k-1-j} \tag{2.50}$$

$$V_k = \frac{1}{k} \sum_{j=0}^{i-1} U_j F_{k-1-j} \tag{2.51}$$

$$F_k = -\frac{w_0^2}{k} U_{k-1} \tag{2.52}$$

$$\theta_k = \frac{1}{k} F_{k-1} \tag{2.53}$$

$$\tag{2.54}$$

2.8.2 Numerical results

For the numerical test, we consider $g = 1$ and $l = 1$. We will present two cases of initials conditions to illustrate two different motions of the single pendulum as presented in [96] (see Fig.(2.34)).

First choice of initials conditions

Firstly, we consider $\theta(0) = 0.0$ and $\dot{\theta}(0) = 1.999999238456499$ as given in [96] to synthesize a highly nonlinear case, where the pendulum performs an oscillating motion. The period T is 33.72102056485366 [96]. For further information about the oscillating pendulum, the nonlinear period, and the maximum angle, the reader is invited to consult the following references [96], [97].

If $\theta_0 = 0$

$$t = \frac{2}{|\Omega_0|} Ei\left(\sin(\theta/2), \kappa\right) \tag{2.55}$$

where $\kappa = 2\omega/\Omega_0$ and $Ei(z, \kappa)$ is the elliptical integral of the first kind and Ω_0 is the initial angular velocity.
As can be appreciated in Fig.(2.35), BPL solution is superposed to the reference one, while classic schemes present very large error for different step size Δt. Even with small time step, the classical schemes RK are not able to provide a valid solution, whereas with a residue tolerance $\epsilon = 10^{-4}$, BPL reproduces the exact solution very easily. The same result is obtained with ANM and IFS.

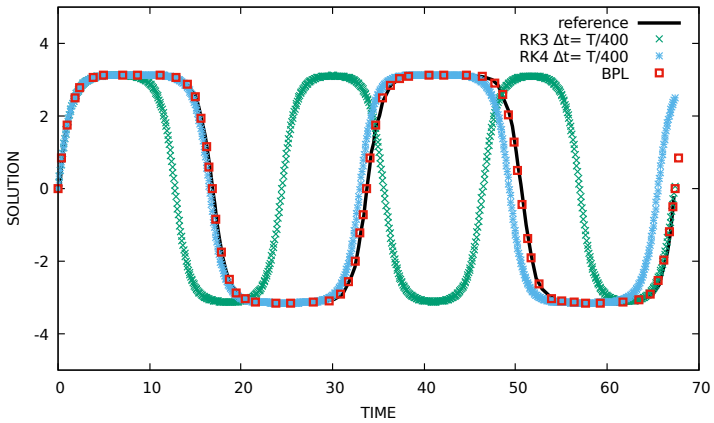


Figure 2.35 – Angles $\theta(t)$ of the oscillating nonlinear single pendulum problem for the following initials conditions $\theta(0) = 0.0$ and $\dot{\theta}(0) = 1.999999238456499$, by time perturbation resummation method BPL and the third and fourth-order Runge-Kutta method (RK3/RK4) with different discrete-time step Δt. The reference solution is in black [96], [97].

Second choice of initials conditions

Secondly, we consider $\theta(0) = 0.0$ and $\dot{\theta}(0) = 2.000000761543501$ as given in [92] to synthesize a highly nonlinear case where the pendulum performs a complete rotation instead of oscillation. The period is $T = 33.7210$. Also for the second set of initial conditions, BPL is superposed to the exact solution as shown in Fig.(2.36). However, the third-order RK (RK3) method shows a huge period error and provides a completely inaccurate solution. As can be seen in Fig.(2.36), the pendulum with RK3 oscillates between the two highest points instead of performing full rotations. RK4 method with $\Delta t = T/400$ shows a noticeable period error while BPL is in very good agreement with the exact solution.

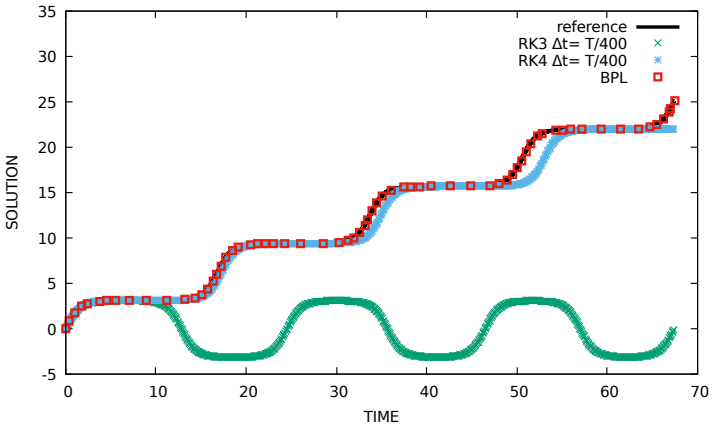


Figure 2.36 – Angles $\theta(t)$ of the rotating nonlinear single pendulum problem for the following initial conditions $\theta(0) = 0.0$ and $\dot{\theta}(0) = 2.000000761543501$, by time perturbation resummation method BPL and the third and fourth-order Runge-Kutta method (RK3/RK4) with different discrete-time step Δt. The reference solution is in black [96], [97].

2.9 Conclusion

Nonlinear differential equations have been successfully solved using the time perturbation and summation methods as time integrators. The first and second-order derivatives that we have developed explicitly for BPL and IFS are validated. The properties of these schemes have been well illustrated, in particular for the IFS scheme as it has been applied as a first prospective study on simple examples.

The recent resummation algorithm Meijer-G (MG) has been applied on the popular Euler equation and the simple harmonic oscillator. The proposal of studying MG in temporal cases has been validated. The conclusions that can be drawn about Meijer-G so far from the two examples are: In the case of divergent series, we noticed that Meijer-G provides an approximation of the solution using a truncation order less than BPL, ANM,

and IFS. It also overcomes the pole of Padé in BPL. The continuation of Meijer-G that allows to obtain a complete continuous solution in time and the vector aspect of Meijer-G will be handled in the next chapter.
Numerical examples have shown also that the time perturbation and resummation methods (ANM, BPL, and IFS) are well suited to simulation over long times (free Van der Pol oscillator study for a final time $t = 10000$s, nonlinear combustion equation for a final time $t = 20000$s). As we have seen, classical schemes often introduce a small time lag at each period, whereas time perturbation and summation methods reproduce the limit cycle more faithfully (Elastic hardening spring and nonlinear simple pendulum problems). Moreover, these methods are also appropriate for stiff problems and can deal with cases that exhibit very fast dynamics in the system (Van der Pol with $\mu = 1000$, nonlinear combustion equation). The efficiency of these methods has also been validated for a 3 DOF example (Lorenz system). The application to models with a large number of degrees of freedom, more accurate computational efforts, and comparisons of the proposed time perturbation integrators will be studied in the next chapters. These encouraging results lead us to apply these schemes to partial differential equations (PDEs) in Chapters 3 and 4, in particular those solving fluid problems.

Chapter 3

Towards applications to the Navier-Stokes equations

After the validation of the time perturbation and resummation algorithms on ODEs in Chapter 2, we will discuss interesting examples of time-dependent PDEs using these methods towards applying these approaches to the Navier-Stokes equation in this chapter. We consider first the heat equation and then the Burgers equation that are more suitable forms of the Navier-Stokes equations having the same convective or diffusion terms as the incompressible Navier-Stokes equations. We recall that validating the time perturbation and resummation methods on the Navier-Stokes equation, is a study that we are interested in.

We note that Hamdouni and Razafindralandy have already performed a first study on the heat equation and the Burgers equation by BPL in one dimension in [32]. The extension to the two-dimensional case was treated by Deeb in [33]. Concerning the application of the time perturbation resummation methods to the Navier-Stokes equations, a reduced model of the Navier-Stokes equation by the Proper Orthogonal Decomposition (POD) method is presented in [34] using BPL. In addition, a simple example of the complete Navier-Stokes equation (without applying the reduction methods), for which the exact solution is known, has been treated in [33] by BPL. Another non-reduced Navier-stokes problem is presented in [35] by BPL.
In this chapter, we will first review some existing applications of time perturbation methods on some examples in the literature by developing some points, and then discuss new ones. We further note that the IFS used as a time integration scheme has not been applied to PDEs in the previously cited work. For this reason, its properties on PDEs (efficiency in terms of feasibility and computation time) will also be discussed in this chapter.
The first goal of this chapter is to advance the use of temporal perturbation methods on models represented by PDEs. These methods have not been widely applied to an extensive number of PDEs. The second goal is to see how these different numerical methods perform for models evolving a large number of degrees of freedom. Therefore, we need to determine if they are both efficient over a large time interval and with an acceptable

computational cost. Therefore, the efficiency of time perturbation-resummation methods for time-dependent PDEs is not the only objective. A comparison of CPU times between classical schemes and time perturbation-resummation methods is essential. Therefore, we will present time studies for different parameters of the probem. The third goal is to show the possibility of associating these temporal perturbation methods to different spatial discretization methods and then determine whether there are any issues with this association. In this chapter, we will use the two well-known spatial discretization methods which are the finite element methods and the MAC method [98] (a popular finite difference method). The last goal is to explore the weaknesses and strengths of each of these methods applied on PDEs.
The Meijer-G approximant will also be tested on examples programmed in Python language, where access to the Meijer-G function is available. The advantages and difficulties of such a use will be presented.

For all the test problems treated in this chapter, we note that if the spatial discretization method used is the finite element method, we will use a popular open-source platform called FEniCS. The latter was founded in 2003 for solving PDEs by the finite element method and is currently being developed in collaboration with researchers from various universities and research institutes. For more information about FEniCS, as well as the most recent updates to the FEniCS software and tutorials, the interested reader is invited to visit the FEniCS web page [99]. Note that FEniCS can be programmed in C++ and Python. However, in this chapter, only Python is used, to be able to test Meijer-G for some examples (since the Meijer-G function is available on Python).

Before proceeding to the numerical applications, it should be noted that to the best of our knowledge, there is currently no unique result concerning the summability of the formal series obtained for all the PDEs (conditions to apply BPL and IFS algorithms). That is, it is not yet known whether any set of formal solutions of a PDE is a Gevrey set and, if so, the Gevrey index is also unknown [100]–[102]. On the other hand, the formal solution of the equations that we have already mentioned (heat equation, Burgers equation, and Navier-Stokes equation) is known to be summable in most cases. The interested reader is invited to consult the following references [24]–[26], [53], [54]. We recall that the Meijer-G approximant can be applied to the series that are not Borel summable [38].

3.1 Application to the heat equation 1D

Consider the following time-dependent heat equation in a one dimensional domain Ω:

$$\frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial x^2} \tag{3.1}$$

with the following initial condition

$$u(x, t_0) = u_0(x) \tag{3.2}$$

associated to the Dirichlet boundary conditions. The parameter ν denotes the thermal diffusivity coefficient. Note that this coefficient is often not included in mathematical studies of the heat equation, although its value can be very important in engineering sciences.

3.1.1 Time perturbation series recurrence formula

We note that the temporal perturbation method is performed first and then followed by the spatial discretization for each of the series terms.
The unknown of the problem u is written as time power series as

$$u(x, \hat{t}) = u_0 + \sum_{k=1}^{N} u_k(x)\hat{t}^k \tag{3.3}$$

where u_0 is the initial known solution. Series terms of time perturbation method for Eq.(3.1) are given by the following recurrence formula, $\forall k = 1, \cdots, N$ we have:

$$(k+1)u_{k+1} = \nu \frac{d^2 u_k}{dx^2} \tag{3.4}$$

that is

$$u_k = \nu^k \frac{u_0^{(2k)}}{k!} \tag{3.5}$$

such that $u_0^{(2k)}$ is the $(2k)$th derivative of u_0 with respect to x.

Remark 3.1.1 *In general cases, the formal series of Eq.(3.1) is divergent but it is convergent if u_0 is an integer function growing at most exponential of order 2 [24], i.e if:*

$$|u_0(x)| \leq Ce^{A|x|^2}, \quad x \in \mathbb{C} \tag{3.6}$$

3.1.2 Spatial discretization

We will use the central finite difference for the spatial discretization. Therefore, Ω is discretized into an uniform grid of size denoted by n_x, and the recurrence formula for obtaining series terms (Eq.(3.4)) becomes after discretization, as follows:

$$(k+1)u_{k+1}^{i} = \nu\left(\frac{u_k^{i-1} - 2u_k^i + u_k^{i+1}}{\Delta x^2}\right), \qquad i = 1, \cdots, n_x \quad \text{and} \quad k = 1, \cdots, N \tag{3.7}$$

where u^i are the values at the nodes.
As x belongs to Ω and the latter is discretized into a uniform grid, a vector noted by $\mathbf{x}$ is obtained. Therefore $u(\mathbf{x}, t)$ is also a vector.

3.1.3 Numerical example

For the numerical example, we take $\nu = 1$, $n_x = 16$, and $\Omega = [0, \pi]$. The exact solution is [32]:

$$u(x,t) = e^{-t}\sin x \tag{3.8}$$

Therefore, the initial condition is

$$u_0(x) = \sin x \tag{3.9}$$

Goals of this example: This example has different purposes. Firstly, it is a first application of Meijer-G to a time-dependent PDE. Secondly, it is also a first validation of Meijer-G approximant to a vector series, the notion that we have proposed in Chapter 1, based on the Gram-Schmidt procedure (see subsection 1.4.4 of Chapter 1). Thirdly, it consists in associating Meijer-G approximant to a spatial discretization technique (for example here is the finite difference method) also for the first time in the literature to the best of our knowledge. Finally, it allows validating the notion of the continuation technique for Meijer-G to obtain the complete solution, instead of only a part of the solution (the classical aspect in previous articles employing Meijer-G [38]). To satisfy these objectives and since this test case has been realized by BPL in [32], we will just present here the result by Meijer-G. Therefore, we have to check first if the new version of the Meijer-G approximant for a vector series is efficient in terms of feasibility, then we will present some comparisons.

Numerically: The series is truncated up to a truncation order $N = 5$. The Gram-Schmidt procedure is used to evaluate Meijer-G for a series of vectors. First, the solution is evaluated without applying any continuation. As can be appreciated in Fig.(3.1), the

solution obtained by Meijer-G approximant without any continuation is in a good agreement with the exact one until $t = 2.1$s visually and through error calculation which is in the range of $[10^{-3}; 10^{-2}]$.

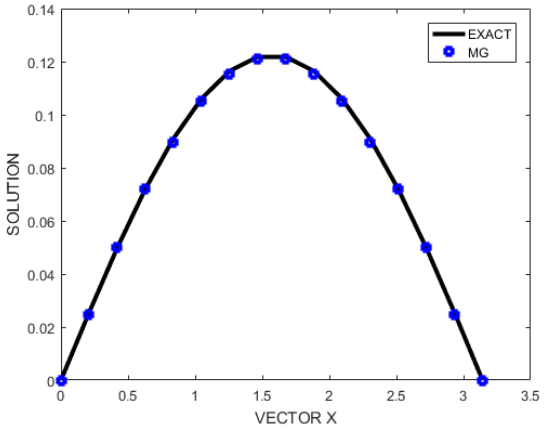
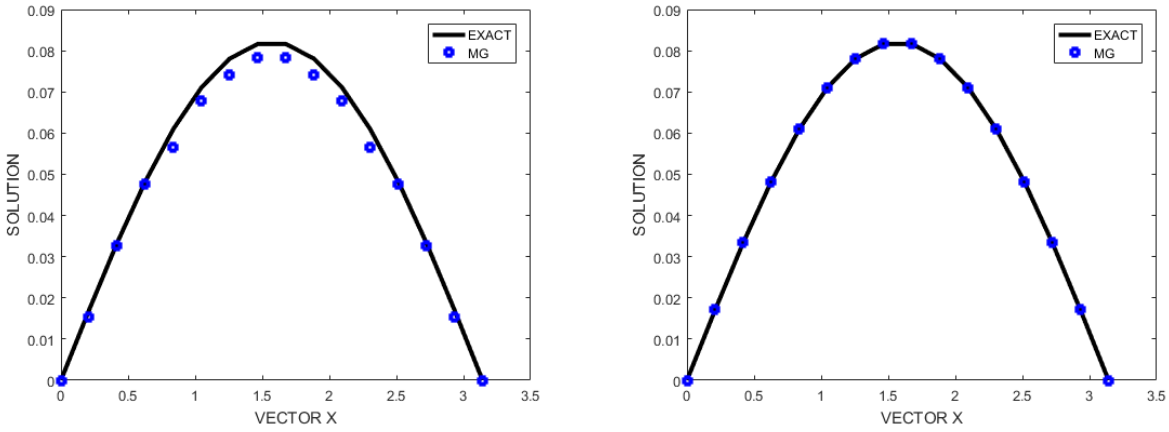


Figure 3.1 – Approximated solution for the heat equation using time perturbation-resummation method Meijer-G (MG) for $N = 5$, without continuation at $t = 2.1$s. The exact solution is in black.

After this time, one needs to apply the continuation procedure to obtain the solution for more great values of t. Fig.(3.2) illustrates the solution obtained by Meijer-G with and without the continuation procedure at $t = 2.5$s. The Meijer-G solution without continuation appears to deviate from the exact solution as can be seen in Fig.(3.2a), whereas applying the continuation technique to Meijer-G allows a good approximation of the solution, as can be appreciated in Fig.(3.2b).



(a) Meijer-G without continuation at $t = 2.5$s

(b) Meijer-G with continuation

Figure 3.2 – Approximated solution for the heat equation using time perturbation-resummation method Meijer-G (MG) for $N = 5$, with and without continuation at $t = 2.5$s. The exact solution is in black.

One can deduce from this part of study that the first use of Meijer-G for time-dependent PDE associated with the finite central difference method, is well validated. The use of Meijer-G for a series of vectors using the Gram-Schmidt procedure, as introduced in the subsection 1.4.4 in Chapter 1, is also well validated.

After the approval of Meijer-G on this example, we proceed now to a simple comparison between BPL and Meijer-G for $N = 5$, with continuation based on the residual criterion to reach a final time of $t = 3$s. With a residual tolerance fixed to $\epsilon = 3 \cdot 10^{-2}$, BPL needs 82 steps and Meijer-G needs 51 steps. No CPU time studies are presented for this example since it is not the main objective and since we are still studying the efficiency of Meijer-G summation for temporal problems.

The conclusions that can be drawn from this example are the following. To the best of our knowledge, for the first time in the literature, we have a:

- Validated application of Meijer-G for time-dependent problems.
- Robust application of Meijer-G for a PDE.
- Association of Meijer-G to a spatial discretization (here is the finite difference method).
- Validated application of Meijer-G for a vector series using the Gram-Schmidt procedure.
- Efficient continuation of Meijer-G which allows obtaining a complete solution instead of only a part of the solution.
- Validated residual and first derivative calculation of Meijer-G (see Appendix C).

3.2 Application to the heat equation 2D

Now, we consider the heat equation in a two dimensional domain Ω with a source term f in the following form:

$$\frac{\partial u}{\partial t} = \nu \Delta u + f \tag{3.10}$$

with Dirichlet boundary conditions

$$u_{|\partial\Omega} = g \tag{3.11}$$

and with the following initial condition

$$u(x, y, t_0) = u_0(x, y) \tag{3.12}$$

where Δ is the Laplace operator. The unknown u is now a function of two spatial variables x and y defined over a two-dimensional domain Ω.

3.2.1 Classical numerical resolution

For the classical resolution of this problem, the RK1 scheme is used for the temporal discretization and the finite element method for the spatial discretization using FEniCS. Therefore, first and foremost, the details of the variational formulation will be given only for this example.

Variational formulation: The basic point of the finite element method is to express a PDE in a variational form [103]. To convert a PDE to a variational problem, three steps are required. First, we multiply the equation by a test function v, then we integrate both members of the resulting equation on the domain Ω, and finally, we perform integration by parts for the terms with second-order derivatives. The function v that multiplies the PDE is called the test function. The unknown function u that should be approximated is called the trial function. The trial and test functions belong to a so-called function space denoted by V, that defines the properties of the functions. In what follows, V is the well-known Sobolev function $H_0^1(\Omega)$ and the definition of the integral related to the scalar product, is used.
The variational problem of Eq.(3.10) is as follows: find $u \in V$ such that

$$(\frac{\partial u}{\partial t}, v) = (\nu \Delta u, v) + (f, v), \quad \forall v \in V \tag{3.13}$$

We note that the first derivative $\dfrac{\partial u}{\partial t}$ is approximated by the classical time integration scheme RK1.
After some mathematical operations by using the Gauss theorem and by considering that the test function v vanishes on the Dirichlet boundary, Eq.(3.13) becomes

$$(\frac{\partial u}{\partial t}, v) + \nu(\nabla u, \nabla v) = (f, v), \quad \forall v \in V \tag{3.14}$$

This equation is the variational form of the original problem. It is a continuous problem that defines the solution u in the infinite dimensional function space V. The finite element method finds an approximate solution of the variational problem Eq.(3.14) by replacing the infinite space of functions with discrete or called finite-dimensional spaces $V_h \subset V$ where h denotes the characteristic size of an element of the mesh. Therefore, the discrete variational problem is as follows: find $u_h \in V_h \subset V$ such that

$$(\frac{\partial u^h}{\partial t}, v^h) + \nu(\nabla u^h, \nabla v^h) = (f^h, v^h), \quad \forall v^h \in V^h. \tag{3.15}$$

Following various steps of assemblage of elementary matrices, these equations are transformed into a matrix equation. Therefore, one can obtain the following semi-discrete equation in the following form:

$$\mathrm{M}\frac{du^h}{dt} = \mathrm{K}u^h + f^h \quad \text{in } \Omega^h \tag{3.16}$$

where M is the assembled mass matrix, K is the rigidity matrix and is associated with the operator Δ, whereas u^h is the vector containing the values of the nodes, and f^h is the discretized source term.

Details about the mesh: For this example, we use a triangular mesh of the 2D rectangle $(-2, 2) \times (2, 2)$ with n_x cells in each direction. Therefore the total number of triangles used is $2 \times n_x \times n_x$ and the total number of vertices is $(n_x + 1) \times (n_x + 1)$, (and n_x will be fixed in the following).

Concerning the type of elements, the standard P1 linear Lagrange element, which is a triangle with nodes at the three vertices of degree 1, is used.

3.2.2 Time perturbation series recurrence formula

After presenting the classical resolution of the initial problem in the previous section using RK1 for the temporal discretization and the finite element method for the spatial discretization, we now write the initial problem in the form of a temporal perturbation and then discretize the recurrence formula that allows obtaining the series terms using the finite element method.

The unknown of the problem u is written as time power series as

$$u(x, y, \hat{t}) = u_0(x, y) + \sum_{k=1}^{N} u_k(x, y)\hat{t}^n \tag{3.17}$$

where u_0 is the known initial data. The time-dependent source term f and g (if it is time-dependent) are also written as time power series as

$$f(x, y, \hat{t}) = f_0(x, y) + \sum_{n=1}^{N} f_n(x, y)\hat{t}^n \tag{3.18}$$

$$g(x, y, \hat{t}) = g_0(x, y) + \sum_{n=1}^{N} g_n(x, y)\hat{t}^n \tag{3.19}$$

where f_0 and g_0 are respectively the initial source term and the initial boundary data.

Continuous formulation in time: The following equations Eq.(3.17), Eq.(3.18), and Eq.(3.19) are injected in the initial problem Eq.(3.10). By identifying the terms according to the power of $\hat{t}$, series terms of time perturbation method for Eq.(3.10) are given by the

following recurrence formula, $\forall k = 1, \cdots, N$ we have:

$$ku_k + \nu\Delta u_{k-1} = f_{k-1} \tag{3.20}$$

Semi-discrete formulation in Ω: For the spatial discretization by the finite element method, we multiply Eq.(3.20) by the test function v which yields to the following equation:

$$k(u_k, v) + \nu(\nabla u_{k-1}, \nabla v) = (f_{k-1}, v), \quad \forall v \in V \tag{3.21}$$

The approximation of the series terms at the nodes of the mesh denoted by u_k^h is given by the following semi-discrete formulation:

$$u_k^h = \frac{1}{k}\mathrm{M}^{-1}(\mathrm{K}u_{k-1}^h + f_{k-1}^h) \quad \text{in } \Omega^h \tag{3.22}$$

where M is the mass matrix, and K is the stiffness matrix. We note that only one M-matrix inversion is required for the whole simulation.

3.2.3 Numerical example

For the numerical example, an interesting problem called the "Diffusion of a Gaussian function" is chosen. The initial data is as follows

$$u_0(x, y) = e^{-ax^2 - ay^2} \tag{3.23}$$

Homogeneous Dirichlet boundary conditions are used ($g = 0$). We take $a = 5$ and the domain $\Omega = [-2, 2] \times [2, 2]$ on the following grid $n_x \times n_x$ with $n_x = 30$. The number of degrees of freedom denoted by ndofs, corresponding to this grid is ndofs = 961.
The series terms are first evaluated and the Euclidean norm of each term of the series at the zeroth step (without continuation procedure) is presented in Fig.(3.3). The latter reports a steady increase as truncation order increases. Note that high values of the Euclidean norm of series terms are not a classical behavior in the ANM with perturbation parameters others than time.

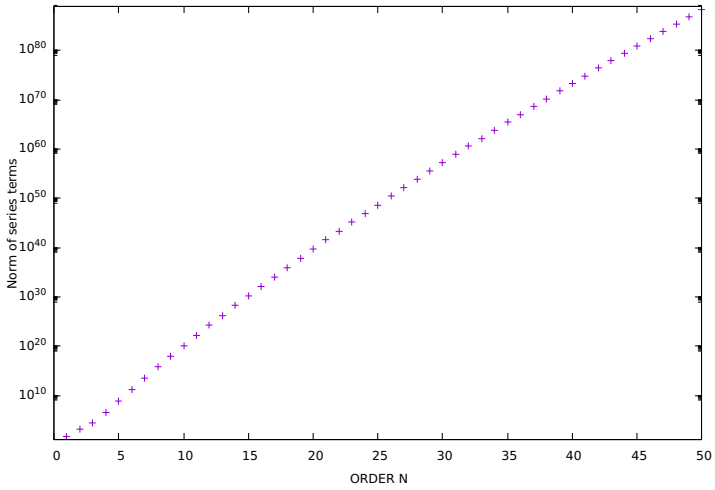
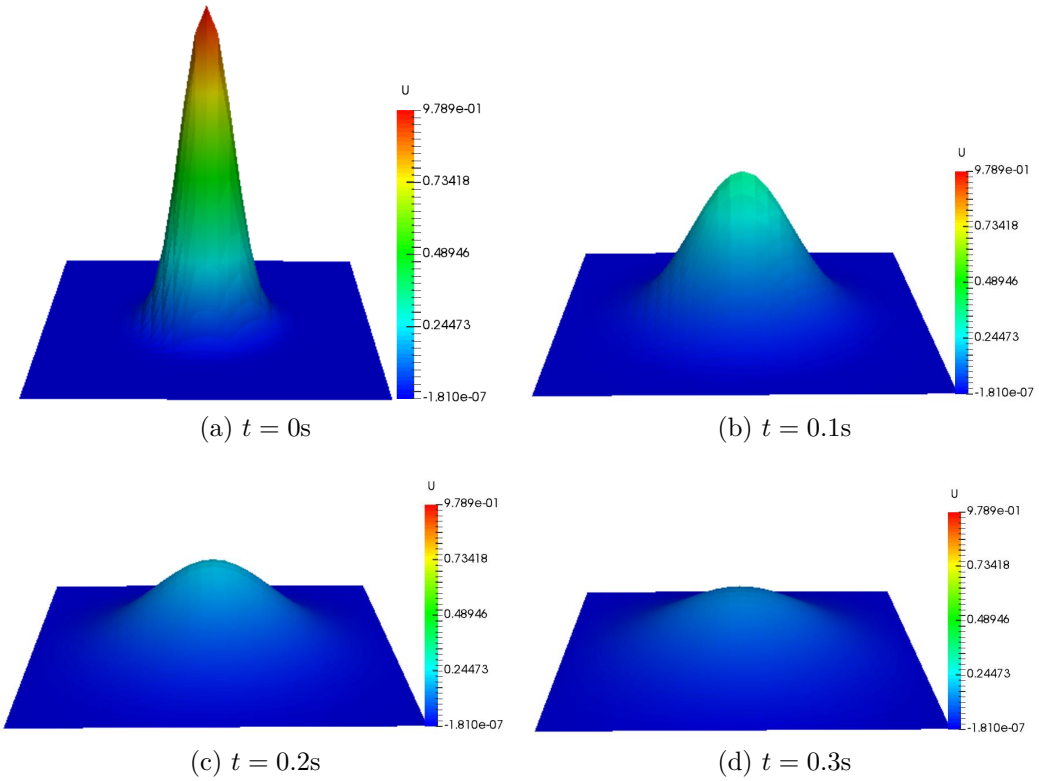


Figure 3.3 – Evolution of the norm of the series terms with the truncation order N at zeroth step (without continuation procedure), for the heat equation 2D. The grid used is 30×30.

Fig.(3.4) displays a sequence of snapshots of the solution of the Gaussian hill problem by the time perturbation method ANM, from $t = 0$s to $t = 0.3$s. The evolution of the solution over time for a chosen probe-line is presented in Fig.(3.5). The behavior of the solution obtained by ANM, BLP, and IFS is the same as the classical scheme RK1, for $t = 1$s in Fig.(3.6). The result shows the good agreement of these latter.

We also tested the Meijer-G resummation for this test case for $N = 5$. However, we were unable to compute the Meijer-G sum on some points on the domain. This could be due to the small value of t_{max} in the Meijer-G relation sum, or to the fact that the implementation of the Meijer-G function does not always give the right answers in some cases of specific parameters [104]; or the Euler Gamma function is infty for some parameters. Therefore, it would be interesting to study the cause.



(a) $t = 0$s (b) $t = 0.1$s

(c) $t = 0.2$s (d) $t = 0.3$s

Figure 3.4 – Sequence of snapshots of the solution of the Diffusion of a Gaussian function obtained by time perturbation method ANM from $t_0 = 0$s to $t = 0.3$s.

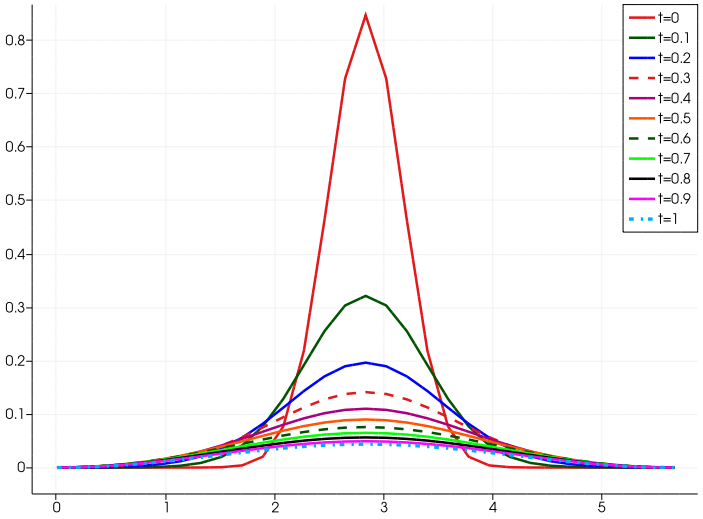


Figure 3.5 – Evolution of the solution of the Gaussian function for a probe-line from $t_0 = 0$s to $t = 1$s.

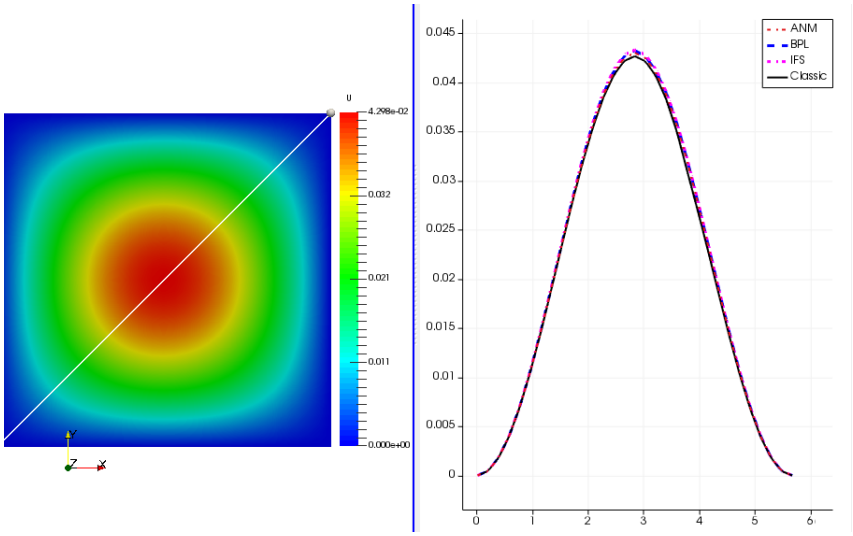


Figure 3.6 – Approximated solutions using time perturbation method ANM probed along a line (presented in white color in the figure). The classic RK1 solution is in black.

Comparisons:

After testing the effectiveness of the methods in this example, we want to compare them to the classical scheme and then to each other. The final time is fixed to $t = 1s$. For sake of comparisons, the CPU number is fixed to 1 (as FEniCS would use as many CPUs as possible by default), the Sparse Linear Equation Solver (SuperLU) is used for the resolution [105]. The residual criterion is used for continuation, and t_{max} is sought using dichotomy. The residual norm of the classical scheme is calculated and will be used as the prescribed residual norm tolerance ϵ of ANM, BPL, and IFS.
All computations have been performed on an Intel Xeon Gold 5120 @ 3.2 GHz, using only one thread.

For the rest of the manuscript, we note by:

- Total (s): The total CPU time spent in all the continuation procedures to reach the final time.
- Series (s): The CPU time spent to calculate the series term during all the continuation procedures, to reach the final time.
- Residual Series (s): The CPU time spent for evaluating the polynomial residual (dichotomy algorithm to perform the continuation technique and evaluate the range of validity via the residual criterion).
- Residual BPL (s): The CPU time spent for evaluating the BPL residual.
- Residual IFS (s): The CPU time spent for evaluating the IFS residual.
- #Residual: The number of times the residual is calculated during the continuation.

- $< t_{max} >$: The average of the upper bound of the validity domain $[0, t_{max}]$ to reach the final time.
- Steps: The number of the continuation steps needed to reach the final time.

The CPU time is all measured in seconds. The software used is FEniCS where the mass matrix M assembles once for all.

Remark 3.2.1 *We note that since the obtained system-matrix to solve is always the same for each time step, it is not necessary to factorize the mass matrix at each increment. It should be possible on FEniCS to store the mass matrix one time, and then use it every time, reducing "solver" to a linear algebra multiplication. This avoids assembly and LU factorization in every solve, which can save CPU time. This is done by defining a "LUSolver" object and asking for reusing the matrix factorization and then saving the factorization between time steps. The part of the code for direct solvers on FEniCS is considered here:*

```
solver = LUSolver()
solver.set_operator(M)
solver.parameters["reuse_factorization"] = True
```

However, as far as we look into the documentation of FEniCS, the option has been removed. "Remove `reuse_factorization`*. This parameter is now deprecated, there are some demo which uses it, it must be removed" as mentioned in [106].*
Therefore, the CPU time for series computation in FEniCS is not optimal, since factorizing the mass matrix is performed for every truncation order and every step. But, it is possible to estimate an optimal CPU time by considering some assumptions for the computation of the time perturbation approaches and the classical scheme. Therefore, for the rest of the manuscript, we consider that the factorization time is 90% of the solving time [107] in FEniCS.
The total CPU time evaluated in all the following tables of comparisons is computed via:

$$\begin{aligned} \textit{CPU Total (s)} = & \textit{CPU Total measured in Fenics (s)} - \ (\textit{Number of steps } \times N \times 0.9 \times \\ & \textit{CPU time of the factorization and Solve (s)}) + 0.9 \times \textit{CPU time of the} \\ & \textit{factorization and Solve(s)} \end{aligned} \tag{3.24}$$

The "CPU Series(s)" are computed in the same way.

1. Classical scheme:

The classical scheme RK1 with a fixed discrete-time $\Delta t = 10^{-3} s$ needs 1000 steps to reach

the final time $t = 1$s with a total CPU time equal to 110.05s. The residual norm of RK1 is $||Res||_2 \approx 2.2 \cdot 10^{-3}$ at the final time $t = 1$s. It was not possible to get a better quality of the solutions with RK1. This time step is then the only one where it is possible to perform comparisons. We will use this value of the residual norm, as the residual tolerance ϵ, to perform comparisons of perturbation methods. Therefore, for the following comparisons, ϵ is fixed to $\epsilon = 2.2 \cdot 10^{-3}$.

2. ANM:

For time perturbation method ANM, we present in Tab.(3.1) some statistics concerning the CPU times spent to reach the final time (Total (s)), the time dedicated to calculate the series terms (Series (s)) during all the continuation techniques and the residual computation time (Residual Series(s)) (dichotomy loop-algorithm). The number of steps required to reach the final time and the average of t_{max} are also presented in this table. As already mentioned, the residual tolerance is fixed to $\epsilon = 2.2 \cdot 10^{-3}$ and the study is carried out with different truncation orders N. It can be observed that the CPU time for evaluating the series terms is between 13s and 23s for different truncation orders N, varying from $N = 5$ to $N = 25$. The total CPU time taken by the ANM is approximately 25s for different truncation orders, while the one taken by the classical scheme is 110.05s for the same residual norm (that is fixed as the residual tolerance for ANM, BPL, and IFS). Therefore, the ANM method is about 5 times faster than the classical scheme for this example. It can be observed also that the CPU time seems to decrease and the t_{max} increase by increasing the truncation orders.

N	Steps	$< t_{max} >$	Total (s)	Series (s)	Residual Series (s)	#RES
5	417	$2.398 \cdot 10^{-3}$	23.69	4.11	15.67	9956
10	285	$3.509 \cdot 10^{-3}$	20.05	5.92	11.36	6840
15	208	$6.950 \cdot 10^{-3}$	16.27	5.68	8.54	4992
20	164	$6.098 \cdot 10^{-3}$	14.28	5.59	7.06	3936
25	136	$7.352 \cdot 10^{-3}$	13.47	5.63	6.19	3264

Table 3.1 – Statistics to reach the final time $t = 1$s for the heat equation 2D, using ANM with different truncation orders N, for a fixed residual tolerance $\epsilon = 2.2 \cdot 10^{-3}$ and a viscosity $\nu = 1$. The grid used is 30×30.

3. BPL:

The same statistic is presented for BPL in Tab.(3.2). For BPL, one can notice that it is fast in terms of CPU time for both the evaluation of the series terms and the calculation of t_{max} using the residual criterion. The t_{max} values are higher than those of the previous table (Tab.(3.1)). We also notice that for $N = 20$, BPL seems to be faster and more optimal than the other orders, requiring only a total CPU time of 3.68s to reach the final

time.

N	Steps	$< t_{max} >$	Total (s)	Series (s)	Residual BPL (s)	#RES
5	266	$3.760 \cdot 10^{-3}$	21.49	2.66	16.22	6384
10	79	$1.266 \cdot 10^{-2}$	8.75	1.47	6.54	2291
15	79	$1.266 \cdot 10^{-2}$	9.94	2.00	7.21	2271
20	25	$4.000 \cdot 10^{-2}$	3.68	0.86	4.87	729
25	70	$1.429 \cdot 10^{-2}$	11.87	2.95	8.08	2030

Table 3.2 – Statistics required to reach the final time $t = 1$s for the heat equation 2D, using BPL, with different truncation orders N for a fixed residual tolerance $\epsilon = 2.2 \cdot 10^{-3}$ and a viscosity $\nu = 1$. The grid used is 30×30. The number of Gauss used is $N_G = 6$.

4. IFS:

For IFS, one can see in Tab.(3.3) that most of the total CPU time is spent for the residual loop. This time grows significantly with the truncation order. It can be observed also that IFS is time-consuming especially for high truncation orders N.

N	Steps	$< t_{max} >$	Total (s)	Series (s)	Residual IFS(s)	#RES
5	512	$1.953 \cdot 10^{-3}$	146.87	5.21	136.64	12288
10	293	$3.413 \cdot 10^{-3}$	307.70	5.30	299.53	7032
15	196	$5.102 \cdot 10^{-3}$	482.92	5.36	475.65	4704
20	137	$7.299 \cdot 10^{-3}$	729.27	7.71	719.54	3288
25	92	$1.086 \cdot 10^{-2}$	704.04	3.96	689.68	2668

Table 3.3 – Statistics required to reach the final time $t = 1$s for the heat equation 2D, using IFS, with different truncation orders N for a fixed residual tolerance $\epsilon = 2.2 \cdot 10^{-3}$ and a viscosity $\nu = 1$. The grid used is 30×30.

5. Comparison between RK1, ANM, BPL, and IFS:

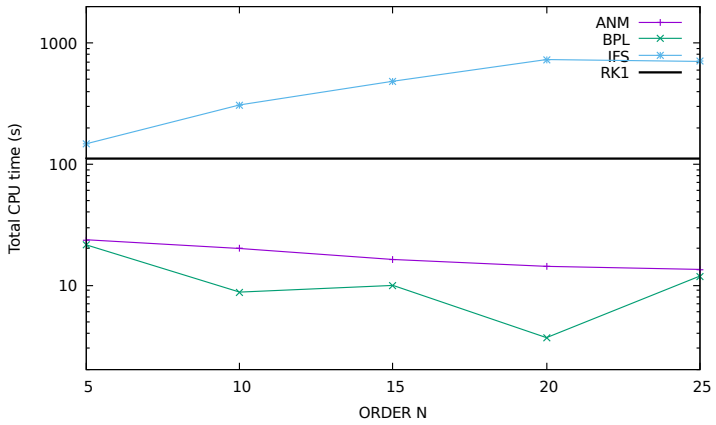
Finally, a brief summary of the methods comparisons is derived from the tables Tab.(3.1), Tab.(3.2) and Tab.(3.3) and is depicted in Fig.(3.7). We recall that the classical method RK1 requires 110.05s of CPU time for the same set of parameters (final time, residual norm). BPL requires only 3.68s for an optimized truncation order $N = 20$ and an approximate CPU time of 15s for the other truncation orders.

Based on Tab.(3.1), Tab.(3.2), Tab.(3.3) and Fig.(3.7), one can conclude that for the same fixed residual tolerance and same truncation order, BPL is the fastest in CPU time, requires the fewest steps, and thus has the largest range of validity to reach the final fixed time. We have seen that increasing the truncation orders increases the CPU time of IFS significantly, but decreases the CPU time of ANM and BPL for some orders.
Finally, ANM and BPL are much faster than the classical scheme for this fixed residual

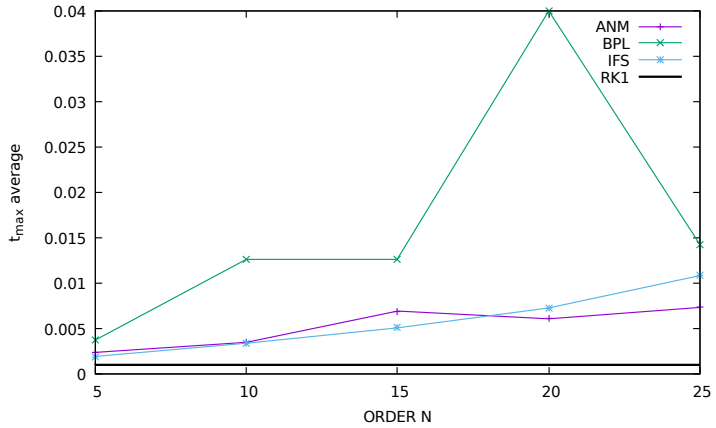
tolerance $\epsilon = 2.2 \cdot 10^{-3}$, but the latter is faster than IFS in terms of computing time. We would like to point out that the three methods are efficient in terms of feasibility.

Total (s) RK1	Total (s) ANM	Total (s) BPL	Total (s) IFS
110.05	14.28	3.68	729.27

Table 3.4 – CPU times required to reach a final time $t = 1$s for the heat equation 2D, using RK1 and ANM, BPL, IFS with $N = 20$ for a fixed residual tolerance $\epsilon = 2.2 \cdot 10^{-3}$ and a viscosity $\nu = 1$. The grid used is 30×30.



(a) Evolution of the CPU time with the truncation order.



(b) Evolution of t_{max} average with the truncation order.

Figure 3.7 – Comparison statistics between ANM, BPL, IFS and RK1 to reach the final time $t = 1$s for the heat equation 2D, for a fixed residual tolerance $\epsilon = 2.2 \cdot 10^{-3}$ and a viscosity $\nu = 1$. The grid used is 30×30.

Study for different mesh size

We are now interested in investigating the sensitivity of these methods to the mesh size. We perform the study for a fixed truncation order of $N = 20$ and three different mesh

size $n_x \times n_x$. The residuals of the classical schemes are calculated at the final time. These residual norms were used as residual tolerance for the temporal perturbation resummation methods computation. The following grids are considered:

- For $n_x = 30$: The mesh size is 30×30. The number of degrees of freedom denoted by ndofs, correspoding to this grid is ndofs $= 961$. The residual tolerance set here is $\epsilon = 2.2 \cdot 10^{-3}$.

- For $n_x = 60$: The mesh size is 60×60. The number of degrees of freedom is ndofs$= 3721$. The residual tolerance set here is $\epsilon = 1.9 \cdot 10^{-3}$.

- For $n_x = 120$: The mesh size is 120×120. The number of degrees of freedom is ndofs $= 14641$. The residual tolerance set here is $\epsilon = 3.7 \cdot 10^{-2}$.

The evolution of the CPU time and the t_{max} with the grid size, are presented respectively in Tab.(3.5) and Fig.(3.8). By increasing the grid size, one can see that the total CPU time of time perturbation-resummation has increased significantly. For this example and this residual tolerance, one can see that RK1 has a lower sensitivity to the grid size than the others approaches. For $n_x = 60$, we notice that BPL is always the fastest among RK1, ANM, and IFS. However for $n_x = 120$, RK1 becomes the fastest. We will see if this behavior remains valid for all the following test cases of this chapter. Finally, increasing the grid size from 30×30 ($n_x = 30$) to 120×120 ($n_x = 120$) decreases the range of validity t_{max} of the three methods, as can be seen in Fig.(3.8).

n_x	ndofs	$\|Res\|_2$	RK1	ANM	BPL	IFS
30	961	$2.2 \cdot 10^{-3}$	110.65	14.28	3.68	729.27
60	3721	$1.9 \cdot 10^{-3}$	136.81	125.54	72.11	2886.32
120	14641	$3.7 \cdot 10^{-2}$	238.13	1277.80	683.24	13010.75

Table 3.5 – Comparison of the total CPU time for several mesh sizes $n_x \times n_x$, for ANM, BPL, IFS, and RK1 to reach the final time $t = 1$s for the heat equation 2D, for a fixed truncation order $N = 20$. Three different mesh size are considered: 30×30, 60×60, and 120×120 ($n_x = 30$, $n_x = 60$ and $n_x = 120$).

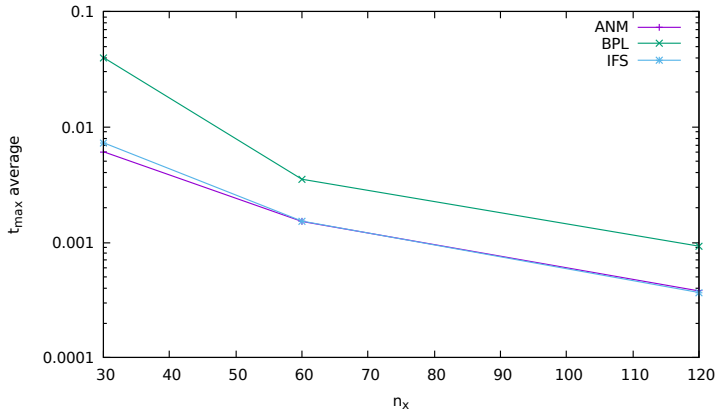


Figure 3.8 – Comparison of $< t_{max} >$ for several mesh sizes $n_x \times n_x$, for ANM, BPL and IFS to reach the final time $t = 1$s for the heat equation 2D, for a fixed truncation order $N = 20$. Three different mesh size are considered: 30×30, 60×60, and 120×120 ($n_x = 30$, $n_x = 60$ and $n_x = 120$).

3.3 Application to the Burgers equation

The two-dimensional coupled viscous Burgers equation is a more suitable form of the Navier-Stokes equations having the same convective and diffusion form as the incompressible Navier-Stokes equations. It is a simple model for understanding various flows and physical problems such as hydrodynamic turbulence, shock wave theory, wave processes in thermoelastic media, and vorticity trajectories.
Consider the two dimensional Burgers equations of the following form:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = \nu \Delta \mathbf{u} \tag{3.25}$$

with Dirichlet boundary conditions

$$\mathbf{u}_{|\partial\Omega} = \mathbf{g} \tag{3.26}$$

and with the following initials conditions

$$\mathbf{u}(\mathbf{x}, t_0) = \mathbf{u}_0(\mathbf{x}) = \mathbf{u}_0(x, y) \tag{3.27}$$

Note that the velocity $\mathbf{u} = (u, v)$ where u and v are the components of $\mathbf{u}$ in the x- and y-directions respectively and $\mathbf{x} = (x, y)$. The function $\mathbf{g}$ is a given Dirichlet boundary data that can be time-dependent.

3.3.1 Classical numerical resolution

We also use the finite element method for the spatial discretization using FEniCS and RK1 for the temporal discretization. After different operations, one can finally obtain the following semi-discrete equation of Eq.(3.25) in the following form:

$$\mathrm{M}\frac{d\mathbf{u}^h}{dt} + \mathrm{N}(\mathbf{u}^h, \mathbf{u}^h) = \nu \mathrm{K}\mathbf{u}^h, \quad \text{in } \Omega^h \tag{3.28}$$

where M is the assembled mass matrix, K is the rigidity matrix associated with the operator Δ, N is the nonlinear discretized operator corresponding to the convective term, whereas $\mathbf{u}^h$ is the vector containing the values of the nodes.

Details about the mesh: We use a triangular mesh of the 2D rectangle $(0, 0.5) \times (0, 0.5)$ with n_x cells in each direction (n_x will be chosen later). Therefore the total number of triangles used is $2 \times n_x \times n_x$ and the total number of vertices is $(n_x + 1) \times (n_x + 1)$ with a crossed diagonal directions. Concerning the type of elements, the standard P1 linear Lagrange element is used.

3.3.2 Time perturbation series recurrence formula

Having presented the classical solution of the initial problem in the previous section using RK1 for the temporal discretization and the finite element method for the spatial discretization, we now write the initial problem in the form of a temporal perturbation and then discretize the recurrence formula that permits to obtain the series terms using the finite element method.

The unknown of the problem $\mathbf{u}$ and $\mathbf{g}$ (if it is time-dependent) are written as time power series

$$\mathbf{u}(x, y, \hat{t}) = \mathbf{u}_0(x, y) + \sum_{k=1}^{N} \mathbf{u}_k(x, y)\hat{t}^k \tag{3.29}$$

$$\mathbf{g}(x, y, \hat{t}) = \mathbf{g}_0(x, y) + \sum_{k=1}^{N} \mathbf{g}_k(x, y)\hat{t}^k \tag{3.30}$$

where $\mathbf{u}_0$ and $\mathbf{g}_0$ are respectively the initial solution and the initial boundary data.

Remark 3.3.1 *In the one-dimensional case, Lysik showed that the formal solution of the Burgers equation is generally divergent, and is Gevrey of index 1 [25]. In this section, the two-dimensional case is studied. One can assume that the intervening series diverge no faster than a Gevrey series of index 1 as in [33] and that we can apply the resummation to this example.*

Continuous formulation in time: Injecting Eq.(3.29) and Eq.(3.30) in Eq.(3.25) and identifying the terms according to the power of $\hat{t}$, one can obtain the following recurrence formula that permits to evaluate the series terms, $\forall k = 1, \cdots, N$:

$$(k+1)\mathbf{u}_{k+1} + \sum_{n=0}^{k} \mathbf{u}_{k-n} \nabla \mathbf{u}_n = \nu \Delta \mathbf{u}_k \tag{3.31}$$

Semi-discrete formulation in Ω: For the spatial discretization by the finite element method, we multiply Eq.(3.31) by the test function v which yields to the following equation:

$$(k+1)(\mathbf{u}_{k+1}, v) + (\sum_{n=0}^{k} \mathbf{u}_{k-n} \nabla \mathbf{u}_n, v) = \nu(\Delta \mathbf{u}_k, v), \quad \forall v \in V \tag{3.32}$$

After some operations, the approximation of the series terms at the nodes of the mesh denoted by $\mathbf{u}_{k+1}^h$ is given by the following semi-discrete formulation:

$$(k+1)\mathrm{M}\mathbf{u}_{k+1}^h + \sum_{n=0}^{k} \mathrm{N}(\mathbf{u}_{k-n}^h, \mathbf{u}_n^h) = \nu \mathrm{K}\mathbf{u}_k^h, \quad \text{in } \Omega^h \tag{3.33}$$

where M is the mass matrix, K is the stiffness matrix, and N is the nonlinear discretized operator corresponding to the convective term. We note that only one M-matrix inversion is required for the whole simulation (as time perturbation-resummation methods are explicit schemes).

3.3.3 Numerical example

The computational two-dimensional domain is $\Omega = [0 \times 0.5] \times [0 \times 0.5]$ with the initial conditions

$$\begin{cases} u(x, y, 0) &= \sin(\pi x) + \cos(\pi y) \\ v(x, y, 0) &= x + y \end{cases} \tag{3.34}$$

and the following boundary conditions:
$\forall\ t \geq 0$ and $0 \leq y \leq 0.5$, we have:

$$\begin{cases} u(0, y, t) = \cos(\pi y) \\ u(0.5, y, t) = 1 + \cos(\pi y) \\ v(0, y, t) = y \\ v(0.5, y, t) = 0.5 + y \end{cases} \tag{3.35}$$

$\forall\ t \geq 0$ and $0 \leq x \leq 0.5$, we have:

$$\begin{cases} u(x,0,t) = 1 + \sin(\pi x) \\ u(x,0.5,t) = \sin(\pi x) \\ v(x,0,t) = x \\ v(x,0.5,t) = x + 0.5 \end{cases} \tag{3.36}$$

The Euclidean norm of each term of the series at zeroth step (without continuation procedure) is plotted in Fig.(3.9). The same behavior of the norms of the series terms as the previous case (heat equation) is depicted here as well. However, the norm of the series terms are much larger than that of the heat equation.

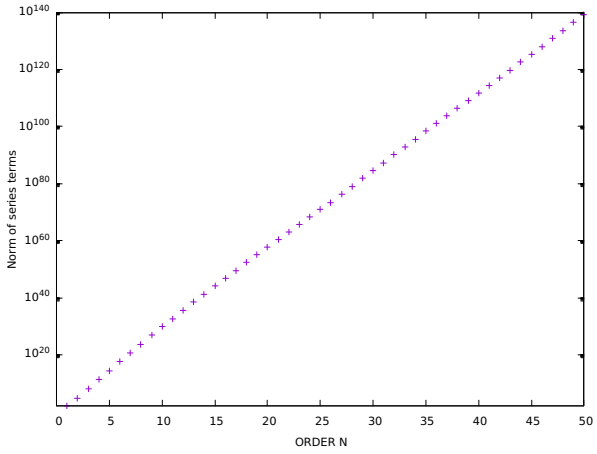


Figure 3.9 – Evolution of the norm of the series terms with the truncation order N at zeroth step (without continuation procedure), for the Burgers equation 2D. The grid used is 20×20.

Efficiency and feasibility study:

This example has been discussed in several papers by many authors, one may cite Jain and Holla [108], Bahadir [109], Srivastava et al. [110], [111], and Tamsir et al. [112]. We will compare the effectiveness of the feasibility of the current work with the result of the previously cited work. For this reason, we use the same parameters as those presented in these papers. The grid is fixed to 20×20 ($n_x = 20$), the viscosity to $\nu = 0.02$, and the final time to $t = 0.625$s as in [108]–[112].
Tab.(3.6) and Tab.(3.7) show the comparisons of the velocity numerical results at some typical mesh points obtained using time perturbation and resummations schemes at $t = 0.625$s with the methods of Jain and Holla [108] (based on cubic spline function technique), Bahadir [109], Srivastava et al. [110], [111] (denoted by I-LFDM and Expo-FDM in their works) and Tamsir et al. [112] (denoted by MCB-DQM, based on a the modified cubic

B-spline differential quadrature method).
Since the evaluated error between ANM, BPL and IFS is very small (approximately $\geq 10^{-6}$ depending on the tolerance), we note by the "present work" the results of the three methods in Tab.(3.6) and Tab.(3.7).
From both tables, it can be noticed that our present work achieves similar results as those of the literature. Efficient numerical results indicate that the time perturbation-resummation schemes are reliable and robust techniques for the numerical solutions of the Burgers equation. The results obtained by these methods are consistent with the literature and references that have treated this example.

(x, y)	Ref. [110]	Ref. [111]	Ref. [109]	Ref. [108]	Ref. [112]	present work
$(0.1, 0.1)$	0.97146	0.97146	0.96688	0.97258	0.97056	0.96955
$(0.3, 0.1)$	1.15280	1.15280	1.14827	1.16214	1.15152	1.14966
$(0.2, 0.2)$	0.86308	0.86308	0.85911	0.86281	0.86244	0.86202
$(0.4, 0.2)$	0.97985	0.97985	0.97637	0.96483	0.98078	0.97901
$(0.1, 0.3)$	0.66316	0.66316	0.66019	0.66318	0.66336	0.66345
$(0.2, 0.4)$	0.58181	0.58181	0.57966	0.58070	0.58273	0.58256
$(0.4, 0.4)$	0.75862	0.75862	0.75678	0.74435	0.76179	0.76048

Table 3.6 – Comparison of the computed values of u obtained by ANM, BPL, and IFS (denoted by the "present work") for $\nu = 0.02$ at $t = 0.625$s, with different references in the literature. The grid size is 20×20.

(x, y)	Ref. [110]	Ref. [111]	Ref. [109]	Ref. [108]	Ref. [112]	present work
$(0.1, 0.1)$	0.09869	0.09869	0.09824	0.09773	0.09842	0.09811
$(0.3, 0.1)$	0.14158	0.14158	0.14112	0.14039	0.14107	0.14035
$(0.2, 0.2)$	0.16754	0.16754	0.16681	0.16660	0.16732	0.16717
$(0.4, 0.2)$	0.17111	0.17111	0.17065	0.17397	0.17223	0.17133
$(0.1, 0.3)$	0.26378	0.26378	0.26261	0.26294	0.26380	0.26370
$(0.2, 0.4)$	0.32851	0.32851	0.32745	0.32402	0.32935	0.32871
$(0.4, 0.4)$	0.32502	0.32502	0.32441	0.31822	0.32884	0.32713

Table 3.7 – Comparison of the computed values of v obtained by ANM, BPL, and IFS (denoted by the "present work") for $\nu = 0.02$ at $t = 0.625$s, with different references in the literature. The grid size 20×20.

Comparisons:

After checking the efficiency of the methods for the Burgers equation, we are interested in carrying out the comparisons first between the classical methods and the time

perturbation-resummation methods, and then between the time perturbation-resummation methods, for an arbitrary final time of $t = 0.17$s. Like the previous example, for sake of comparisons, the CPU number is fixed to 1 (as FEniCS would use as many CPUs as possible by default), the Sparse Linear Equation Solver (SuperLU) is used for the resolution [105]. The residual criterion is used for continuation, and t_{max} is sought using dichotomy. The residual norm of the classical scheme is calculated and will be used as the prescribed residual norm tolerance ϵ of ANM, BPL, and IFS.
All computations have been performed on an Intel Xeon Gold 5120 @ 3.2 GHz, using only one thread.

1. Classical scheme:
The classical scheme RK1 with a fixed time step $\Delta t = 10^{-3}$s needs only 170 steps to reach the final time, with a total CPU time equal to 9.53s. The residual norm is $||Res||_2 \approx 7.3 \cdot 10^{-3}$ at the final time. To have the same residual range for all methods, we will use this value of residual norm to perform comparisons of perturbation methods. Therefore, for the following comparisons, the residual tolerance is fixed to $\epsilon = 2.2 \cdot 10^{-3}$.

2. ANM:
For time perturbation method ANM, we present in Tab.(3.8) some statistics concerning the CPU times spent to reach the final time (Total (s)), the time dedicated to calculate the series terms (Series (s)) during all the continuation technique, and the residual computation time (Residual Series(s)). The number of steps required to reach the final time and the average of t_{max} are also presented in this table. As already mentioned, the residual tolerance is fixed to $\epsilon = 7.3 \cdot 10^{-3}$ and the study is carried out with different truncation orders N. It can be observed that most of the total CPU time, is spent by evaluating the series terms at every step of the continuation. We also notice that the CPU time increases by increasing the truncation order.

N	Steps	$< t_{max} >$	Total (s)	Series (s)	Residual Series (s)	#RES
5	528	$3.220 \cdot 10^{-4}$	64.27	24.51	33.33	10560
10	336	$5.070 \cdot 10^{-4}$	60.05	34.31	21.87	6720
15	245	$6.950 \cdot 10^{-4}$	63.57	44.44	16.41	4900
20	192	$8.855 \cdot 10^{-4}$	70.01	54.35	13.36	3844
25	171	$9.987 \cdot 10^{-4}$	86.36	67.47	12.87	3660

Table 3.8 – Statistics to reach $t = 0.17$s for the Burgers equations 2D, using ANM with different truncation orders N, for a fixed residual tolerance $\epsilon = 7.3 \cdot 10^{-3}$ and a viscosity $\nu = 0.02$. The grid used is 20×20.

3. BPL:

The same statistic is presented for BPL in Tab.(3.9). For BPL, one can notice that it is fast in terms of CPU time for both the evaluation of the series terms and the calculation of t_{max} using the residual criterion. The t_{max} values are higher than those of ANM (see Tab.(3.8)).

N	Steps	$<t_{max}>$	Total (s)	Series (s)	Residual BPL (s)	#RES
5	177	$9.604 \cdot 10^{-4}$	27.51	7.13	17.66	3624
10	95	$1.789 \cdot 10^{-3}$	23.01	9.71	12.24	2285
15	79	$2.152 \cdot 10^{-3}$	26.43	14.23	11.31	1906
20	85	$2.000 \cdot 10^{-3}$	38.14	23.89	13.29	2045
25	48	$3.542 \cdot 10^{-3}$	28.97	19.13	8.12	1157

Table 3.9 – Statistics to reach $t = 0.17$s for the Burgers equations 2D, using BPL with different truncation orders N, for a fixed residual tolerance $\epsilon = 7.3 \cdot 10^{-3}$ and a viscosity $\nu = 0.02$. The grid used is 20×20. The number of Gauss used is $N_G = 6$.

Remark 3.3.2 *We also investigated the CPU time required to compute one-BPL solution for various truncation orders N, for this example. The results have shown that calculating one BPL solution is not time-consuming. We found that the CPU time required to compute one-BPL solution for $N = 10$ is approximately $1.3 \cdot 10^{-2}$s and increases with the truncation order. The CPU time required to compute the Borel transforms, the Padé coefficient, the Gauss Laguerre quadrature step, and the BPL solution were measured. These four steps were proven to be extremely fast. The BPL derivative is also quickly evaluated. And, if one wants to know which step takes the most CPU time, one can say it is the step related to the evaluation of the Padé coefficients.*

4. IFS:

For IFS, one can see in Tab.(3.10) that most of the total CPU time is spent for the residual loop. This time grows significantly with the truncation order. It can be observed also that IFS is time-consuming especially for high truncation orders N.

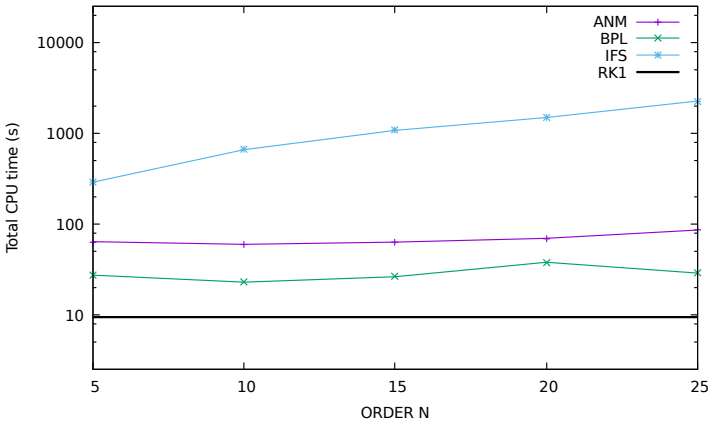
N	Steps	$< t_{max} >$	Total (s)	Series (s)	Residual IFS(s)	#RES
5	609	$2.791 \cdot 10^{-4}$	290.64	27.16	256.18	12180
10	360	$4.722 \cdot 10^{-4}$	662.79	36.86	621.49	7200
15	255	$6.667 \cdot 10^{-4}$	1080.80	45.27	1032.37	5100
20	196	$8.673 \cdot 10^{-4}$	1498.97	53.05	1443.00	3924
25	169	$1.006 \cdot 10^{-3}$	2272.73	71.86	2194.70	3752

Table 3.10 – Statistics to reach $t = 0.17$s for the Burgers equations 2D, using IFS with different truncation orders N, for a fixed residual tolerance $\epsilon = 7.3 \cdot 10^{-3}$ and a viscosity $\nu = 0.02$. The grid used is 20×20.

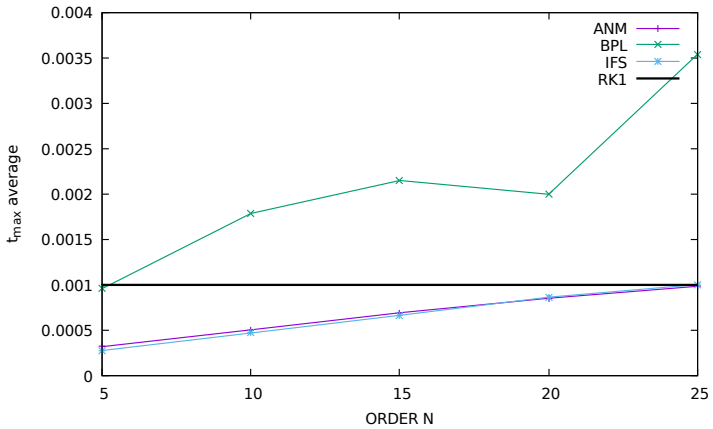
Remark 3.3.3 *We also investigated the CPU time required to compute one-IFS solution for various truncation orders N. It was found that evaluating one-IFS solution alone is time-consuming as well as its derivative.*

5. Comparison between RK1, ANM, BPL, and IFS:

Finally, a brief summary of the methods comparisons is derived from the tables Tab.(3.8), Tab.(3.9) and Tab.(3.10) and is depicted in Fig.(3.10). One can conclude that, for this example, BPL is the fastest among AMN and IFS. It has the highest range of validity and requires the fewest steps to reach the same final time. IFS is the most time-consuming. The classical scheme was slightly faster than BPL for this example because the calculation of the series terms for the Burgers equation is considered to be time-consuming. However, the t_{max} of BPL is larger than that of RK1.



(a) Evolution of the CPU time with the truncation order



(b) Evolution of t_{max} average with the truncation order

Figure 3.10 – Comparison statistics between ANM, BPL, IFS and RK1 to reach $t = 0.17$s for the Burgers equation 2D, for a fixed residual tolerance $\epsilon = 7.3 \cdot 10^{-3}$ and a viscosity $\nu = 0.02$. The grid used is 20×20.

Study for different mesh sizes

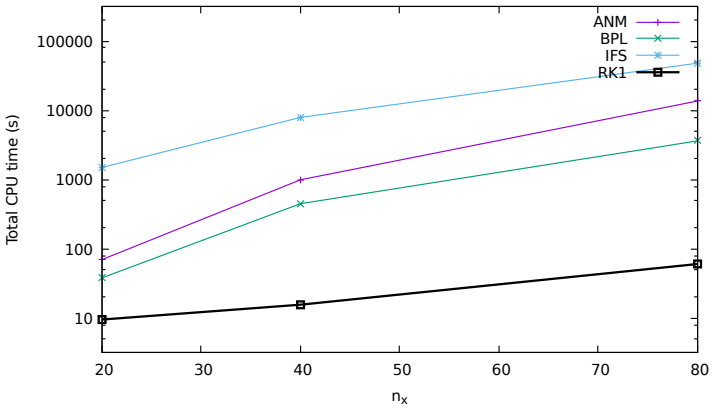
We are now interested in investigating the sensitivity of these methods to the mesh size. We perform the study with a fixed truncation order of $N = 20$ and three different mesh size $n_x \times n_x$. The residuals of the classical scheme are calculated at the final time. These residual norms are used as residual tolerances in the temporal perturbation methods computation. The following grids are considered:

- For $n_x = 20$: The mesh size is 20×20. The number of degrees of freedom denoted by ndofs, corresponding to this size is ndofs = 6562. The residual tolerance set here is $\epsilon = 7.3 \cdot 10^{-3}$.

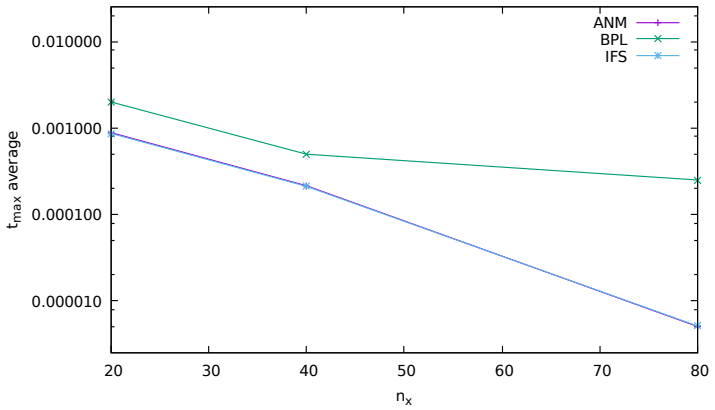
- For $n_x = 40$: The mesh size is 40×40. The number of degrees of freedom is ndofs = 25922. The residual tolerance set here is $\epsilon = 4.0 \cdot 10^{-3}$.

- For $n_x = 80$: The mesh size is 80×80. The number of degrees of freedom is ndofs = 103042. The residual tolerance set here is $\epsilon = 4.9 \cdot 10^{-1}$.

The evolution of the CPU time and the t_{max} with the grid size, are presented in Fig.(3.11). By increasing the mesh size, one can see that the t_{max} decreases, and the CPU time increases, for the three methods. We note that the t_{max} of IFS and ANM decreases much more than that of BPL when n_x varies from 40 to 80.



(a) Evolution of the total CPU time with the grid size



(b) Evolution of t_{max} with the grid size

Figure 3.11 – Comparison statistics for several mesh sizes $n_x \times n_x$, for ANM, BPL and IFS to reach $t = 0.17$s for the Burgers equation 2D, for a fixed truncation order $N = 20$. Three different mesh size are considered: 20×20, 40×40, and 80×80 ($n_x = 20$, $n_x = 40$ and $n_x = 80$).

3.4 Application to Navier-Stokes equations

After studying the time perturbation-resummation methods on the heat equation and the Burgers equation, we now proceed to their application to the Navier-Stokes equations that describe the motion of a viscous fluid flow. The Navier-Stokes equations contain the conservation laws of mass and momentum and describe the spatial and temporal change of the velocity field of the fluid. The numerical solution of the Navier-Stokes equations can be split into two main categories: compressible and incompressible flows. We have chosen to consider the assumptions of the incompressibility of the fluid in this manuscript.
Therefore, consider the time-dependent incompressible Navier-Stokes equations (NSE) in

a connected domain $\Omega \in \mathbb{R}^2$ with a piece-wise smooth boundary $\partial\Omega$, in the following form:

$$\rho\frac{\partial \mathbf{u}}{\partial t} - \mu\Delta\mathbf{u} + \rho\mathbf{u}\cdot\nabla\mathbf{u} + \nabla p = \mathbf{f} \tag{3.37}$$

$$\nabla\cdot\mathbf{u} = 0 \tag{3.38}$$

with Dirichlet boundary conditions

$$\mathbf{u}(\mathbf{x},t) = \mathbf{g}(\mathbf{x},t) \quad \text{in } \partial\Omega \tag{3.39}$$

with the following initial conditions

$$\mathbf{u}(\mathbf{x},0) = \mathbf{u}_0(\mathbf{x}) \tag{3.40}$$

Eq.(3.37) derives from the conservation of momentum, and Eq.(3.38) derives from the conservation of mass and is the incompressibility condition. Therefore, we note that for this problem we have two residues to consider (the momentum residual and the continuity residual linked respectively to Eq.(3.37) and Eq.(3.38)).
The unknowns of the problem are the velocity $\mathbf{u} = (u, v)$ and the pressure p. Note that u and v are the components of the fluid velocities in the x- and y-directions respectively. μ denotes the dynamic viscosity, $\mathbf{f}$ the given source term and ρ is the density of the fluid. We define the kinematic viscosity as $\nu = \frac{\mu}{\rho}$.

A description of each term in the Navier-Stokes equations is presented in the following:

- $\frac{\partial \mathbf{u}}{\partial t}$: The derivative of the velocity with respect to time.
  During each time step of the simulation, this term will be calculated at all grid points containing fluid.
- $\rho(\mathbf{u}\cdot\nabla\mathbf{u})$: The convection term.
  This term is derived from the conservation of momentum.
- ∇p: The pressure term.
  This term refers to the forces produced by pressure differences within a fluid. As the fluid is assumed to be incompressible, we will combine the pressure term with Eq.(3.38) to ensure that the flow remains incompressible.
- $\mu\Delta\mathbf{u}$: The viscosity term.
  In thick fluids, friction forces cause the velocity of the fluid to move toward the neighborhood average. This term captures this relationship using the Δ operator.
- $\mathbf{f}$: The source term.
  This term can include all external forces, such as gravity or forces of contact with objects.

One can note that the major difficulty in finding a solution to the Navier-Stokes equations is the coupling between velocity and pressure through the incompressibility condition which is Eq.(3.38). Different methods for solving the Navier-Stokes equations are proposed in the literature. One can mention artificial compressibility methods, projection methods, spectral methods, or penalty methods such as the augmented Lagrangian method. Projection methods are the most successful ones because of their practical simplicity of implementation and flexibility of use. They have been widely used and developed. The splitting or projection method proposed by Chorin in 1968 [113] and Temam in 1969 [114], also known as the Chorin method, is one of the oldest methods. It is also known in the literature as the "non-incremental pressure-correction scheme" or the "pressure-Poisson approach". We have chosen to use this original version of the projection algorithm since it is based on a fully explicit formulation like the time perturbation-resummation approaches. The Chorin projection method is described in the following subsection.

3.4.1 Chorin projection method and classical resolution

In the first place, an intermediate velocity field is determined before it is projected to zero divergences by a Hodge decomposition. Given the approximate discrete solution at time t^n, we look for the solution at time t^{n+1} following a time step Δt such that $t^{n+1} = t^n + \Delta t$. The Chorin projection method is based on the following three steps:

- Prediction: The first step consists in ignoring the pressure gradient term of the momentum equation and then calculating an intermediate velocity $\mathbf{u}^*$ by

$$\begin{cases} \rho\dfrac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} - \mu\Delta\mathbf{u}^n + \rho\mathbf{u}^n \cdot \nabla\mathbf{u}^n = \mathbf{f} \\ \mathbf{u}^* = \mathbf{g} \qquad \text{in } \partial\Omega \end{cases} \tag{3.41}$$

  where $\mathbf{u}^n$ is the velocity at n^{th} time step. Note that the first derivative $\dfrac{\partial u}{\partial t}$ was approximated by RK1.
- Pressure solution: The second step consists in correcting the intermediate velocity to obtain the final time step solution.

$$\begin{cases} \Delta p^{n+1} = \dfrac{\rho}{\Delta t}\nabla \cdot \mathbf{u}^* \\ \dfrac{\partial p^{n+1}}{\partial \mathbf{n}} = 0 \quad \text{in } \partial\Omega \end{cases} \tag{3.42}$$

  where $\mathbf{n}$ is the unit normal vector at $\partial\Omega$.
- Correction of the velocity field: The final step consists in correcting the predicted

velocity in order to satisfy the continuity equation.

$$\begin{cases} \mathbf{u}^{n+1} = \mathbf{u}^* - \dfrac{\Delta t}{\rho}\nabla p^{n+1} \\ \nabla \cdot \mathbf{u}^{n+1} = 0 \qquad \text{in } \partial\Omega \end{cases} \tag{3.43}$$

After presenting the classical resolution of the problem, we are interested in solving Eq.(3.37) and Eq.(3.38) with the time perturbation-resummation methods in the next subsection.

3.4.2 Time perturbation series recurrence formula

We first introduce the development in temporal series in detail, in order to see how the problem will be decomposed and then the obtained equations will be discretized.
The unknowns of the problem defined by Eq.(3.37) and Eq.(3.38), which are $\mathbf{u}$ and p are written as time power series:

$$\mathbf{u}(\mathbf{x}, \hat{t}) = \mathbf{u}_0(x, y) + \sum_{k=1}^{N} \mathbf{u}_k(x, y)\hat{t}^k \tag{3.44}$$

$$p(\mathbf{x}, \hat{t}) = \sum_{k=0}^{N} p_k(x, y)\hat{t}^k \tag{3.45}$$

where $\mathbf{u}_0$ is the initial known solution. We note that p_0 can be unknown in many cases and is calculated from $\mathbf{u}_0^*$. The external forces $\mathbf{f}$ and the Dirichlet boundary data $\mathbf{g}$ (if it is time-dependent) are also written as time power series:

$$\mathbf{f}(\mathbf{x}, \hat{t}) = \mathbf{f}_0(x, y) + \sum_{k=1}^{N} \mathbf{f}_k(x, y)\hat{t}^k \tag{3.46}$$

$$\mathbf{g}(\mathbf{x}, \hat{t}) = \mathbf{g}_0(x, y) + \sum_{k=1}^{N} \mathbf{g}_k(x, y)\hat{t}^k \tag{3.47}$$

where $\mathbf{f}_0$ and $\mathbf{g}_0$ are respectively the initial source term and the initial Dirichlet boundary data.
Then, Eq.(3.44), Eq.(3.45), Eq.(3.46) and Eq.(3.47) are injected in the initial problem Eq.(3.37) and Eq.(3.38). Finally, by equating the terms according to the power of $\hat{t}$, the

following nonlinear systems are obtained:

$$\hat{t}^0 : \begin{cases} \rho\mathbf{u}_1 &= \mu\Delta\mathbf{u}_0 - \rho\mathbf{u}_0\cdot\nabla\mathbf{u}_0 - \nabla p_0 + \mathbf{f}_0 \\ \nabla\cdot\mathbf{u}_1 &= 0 \\ \mathbf{u}_1 &= \mathbf{g}_1 \qquad \text{in } \partial\Omega \end{cases} \tag{3.48}$$

$$\hat{t}^1 : \begin{cases} 2\rho\mathbf{u}_2 &= \mu\Delta\mathbf{u}_1 - \rho\mathbf{u}_1\cdot\nabla\mathbf{u}_0 - \rho\mathbf{u}_0\cdot\nabla\mathbf{u}_1 - \nabla p_1 + \mathbf{f}_1 \\ \nabla\cdot\mathbf{u}_2 &= 0 \\ \mathbf{u}_2 &= \mathbf{g}_2 \qquad \text{in } \partial\Omega \end{cases} \tag{3.49}$$

$$\vdots$$

$$\hat{t}^k, k\in[2,N] : \begin{cases} \rho(k+1)\mathbf{u}_{k+1} &= \mu\Delta\mathbf{u}_k - \rho\sum_{r=0}^{k}\mathbf{u}_r\cdot\nabla\mathbf{u}_{k-r} - \nabla p_k + \mathbf{f}_k \\ \nabla\cdot\mathbf{u}_{k+1} &= 0 \\ \mathbf{u}_{k+1} &= \mathbf{g}_{k+1} \qquad \text{in } \partial\Omega \end{cases} \tag{3.50}$$

To resolve this system, a prediction-correction algorithm is used. Therefore, the concept of the Chorin projection method is applied to every order.

Calculation of $\mathbf{u}_1$ and p_0:

A detail of the resolution steps of the first system of $\hat{t}^0$ (Eq.(3.48)) that allows to calculate $\mathbf{u}_1$ and p_0 is presented.
First, consider again the first equation of the system presented in Eq.(3.48):

$$\rho\mathbf{u}_1 = \mu\Delta\mathbf{u}_0 - \rho\mathbf{u}_0\cdot\nabla\mathbf{u}_0 - \nabla p_0 + \mathbf{f}_0 \tag{3.51}$$

By applying a divergence to the latter on both sides, one can obtain:

$$\nabla\cdot(\rho\mathbf{u}_1) = \nabla\cdot(\mu\Delta\mathbf{u}_0 - \rho\mathbf{u}_0\cdot\nabla\mathbf{u}_0 - \nabla p_0 + \mathbf{f}_0) \tag{3.52}$$

Since $\nabla\cdot\mathbf{u}_1 = 0$ (continuity equation, see Eq.(3.38)), the left hand side of Eq.(3.52) vanishes. Therefore, one can get:

$$\nabla\cdot(\mu\Delta\mathbf{u}_0 - \rho\mathbf{u}_0\cdot\nabla\mathbf{u}_0 - \nabla p_0 + \mathbf{f}_0) = 0 \tag{3.53}$$

Thus, the development of the gradient in Eq.(3.53) yields to

$$-\Delta p_0 + \nabla\cdot(\mu\Delta\mathbf{u}_0 - \rho\mathbf{u}_0\cdot\nabla\mathbf{u}_0 + \mathbf{f}_0) = 0 \tag{3.54}$$

An auxiliary variable denoted by $\mathbf{u}_1^*$ is introduced [33] and defined by

$$\rho\mathbf{u}_1^* = \mu\Delta\mathbf{u}_0 - \rho\mathbf{u}_0 \cdot \nabla\mathbf{u}_0 + \mathbf{f}_0 \tag{3.55}$$

Therefore, Eq.(3.54) becomes

$$-\Delta p_0 + \nabla \cdot (\rho\mathbf{u}_1^*) = 0 \tag{3.56}$$

which is equivalent to

$$\Delta p_0 = \rho\ \nabla \cdot \mathbf{u}_1^* \tag{3.57}$$

associated to the following Neumann boundary condition

$$\nabla p_0 \cdot \mathbf{n} = 0 \tag{3.58}$$

where $\mathbf{n}$ denotes the normal to $\partial\Omega$.
Then, Eq.(3.51) becomes

$$\rho\mathbf{u}_1 = \mu\Delta\mathbf{u}_0 - \rho\mathbf{u}_0 \cdot \nabla\mathbf{u}_0 - \nabla p_0 + \mathbf{f}_0 \tag{3.59}$$

$$= \rho\mathbf{u}_1^* - \nabla p_0 \tag{3.60}$$

Therefore,

$$\mathbf{u}_1 = \mathbf{u}_1^* - \frac{1}{\rho}\nabla p_0 \tag{3.61}$$

Finally as summary, the steps that should be resolved to obtain $\mathbf{u}_1$ an p_0 are as follows:

$$\begin{cases} \rho\mathbf{u}_1^* &= \mu\Delta\mathbf{u}_0 - \rho\mathbf{u}_0 \cdot \nabla\mathbf{u}_0 + \mathbf{f}_0 \\ \Delta p_0 &= \rho\ \nabla \cdot \mathbf{u}_1^* \\ \mathbf{u}_1 &= \mathbf{u}_1^* - \dfrac{1}{\rho}\nabla p_0 \end{cases} \tag{3.62}$$

associated to the following Dirichlet boundary condition

$$\mathbf{u}_1 = \mathbf{g}_1 \quad \text{in } \partial\Omega \tag{3.63}$$

and to the Neumann boundary condition

$$\nabla p_0 \cdot \mathbf{n} = 0 \tag{3.64}$$

where $\mathbf{n}$ denotes the normal to $\partial\Omega$.

Calculation of $\mathbf{u}_k$ and p_{k-1} for $k \in [2, N]$:

The same procedure is applied for $\mathbf{u}_2, \mathbf{u}_3, \cdots, \mathbf{u}_N$. Therefore, the steps that are resolved to obtain $\mathbf{u}_k$ and p_{k-1} for $k \in [2, N]$ are as follows:

$$\mathrm{t}^0 : \begin{cases} \rho \mathbf{u}_1^* &= \mu \Delta \mathbf{u}_0 - \rho \mathbf{u}_0 \cdot \nabla \mathbf{u}_0 + \mathbf{f}_0 \\ \Delta p_0 &= \rho \ \nabla \cdot \mathbf{u}_1^* \\ \mathbf{u}_1 &= \mathbf{u}_1^* - \dfrac{1}{\rho} \nabla p_0 \end{cases} \tag{3.65}$$

$$\mathrm{t}^1 : \begin{cases} 2\rho \mathbf{u}_2^* &= \mu \Delta \mathbf{u}_1 - \rho \mathbf{u}_0 \cdot \nabla \mathbf{u}_1 - \rho \mathbf{u}_1 \cdot \nabla \mathbf{u}_0 + \mathbf{f}_1 \\ \Delta p_1 &= 2 \ \rho \ \nabla \cdot \mathbf{u}_2^* \\ \mathbf{u}_2 &= \mathbf{u}_2^* - \dfrac{1}{2 \ \rho} \nabla p_1 \end{cases} \tag{3.66}$$

$$\vdots \tag{3.67}$$

$$\mathrm{t}^k, k \in [2, N] : \begin{cases} \rho \ (k+1) \mathbf{u}_{k+1}^* &= \mu \Delta \mathbf{u}_k - \rho \sum_{r=0}^{k} \mathbf{u}_r \cdot \nabla \mathbf{u}_{k-r} + \mathbf{f}_k \\ \Delta p_k &= \rho \ (k+1) \ \nabla \cdot \mathbf{u}_{k+1}^* \\ \mathbf{u}_{k+1} &= \mathbf{u}_{k+1}^* - \dfrac{1}{\rho \ (k+1)} \nabla p_k \end{cases} \tag{3.68}$$

with the following Dirichlet boundary condition

$$\mathbf{u}_{k+1} = \mathbf{g}_{k+1} \quad \text{in } \partial\Omega \tag{3.69}$$

and the Neumann boundary condition

$$\nabla p_k \cdot \mathbf{n} = 0 \quad \text{in } \partial\Omega \tag{3.70}$$

Remark 3.4.1 *Costin has shown that the formal solution of the Navier-Stokes equation is in general divergent, Gevrey of index 1 and Borel summable [54].*

3.4.3 Spatial discretization

The obtained equations Eq.(3.68), Eq.(3.69) and Eq.(3.70) that permit to obtain series terms are discretized. All popular methods, such as the finite difference method, the finite element method, the spectral method, or the spectral element method, can be used for the Navier-Stokes equations. In this section, we have chosen to use the finite difference MAC method for the Lid-driven cavity problem, and the finite element method for the Navier-Stokes in an L-shaped domain, and the Channel flow examples using FEniCS.

3.4.4 Some useful definitions

Before we present the various test cases, we introduce some definitions that will be used for the rest of the document.

- Stream function:
  The flow field of some problem is visualized through streamline obtained from stream function denoted by ψ. Stream function is defined from the velocity components u and v. The formulations that relate the stream function ψ to velocity components u and v for a 2D flow are given by:

$$u = \frac{\partial \psi}{\partial y} \quad \text{and} \quad v = -\frac{\partial \psi}{\partial x} \tag{3.71}$$

  Streamlines are the lines which are tangent to the velocity vector at each point of the domain:

$$\frac{u}{v} = \frac{dx}{dy} \tag{3.72}$$

  Combining Eq.(3.72) with Eq.(3.71) leads to:

$$\frac{\partial \psi}{\partial x} dx + \frac{\partial \psi}{\partial y} dy = 0 \tag{3.73}$$

- Reynolds number:
  The Reynolds number characterizes a flow by classifying the latter from laminar (at low Reynolds numbers) to turbulent (at high Reynolds numbers). It is defined as:

$$Re = \frac{\rho U L}{\mu} = \frac{U L}{\nu} \tag{3.74}$$

  where U and L are respectively a speed and a reference length characteristic of the application. L denotes the characteristic length of the flow. The viscosities ν and μ are defined previously. This number can range from zero to millions or even tens of millions. This wide range corresponds to very different physical situations.

3.4.5 Numerical examples

First, a problem with a known analytical solution called the Channel or Poiseuille flow is presented by these methods. To validate the terms of the computed series, the error between the analytical solution and the solution obtained by the time perturbation methods is evaluated. As this example is a priori simple and is performed as a first validation of the time perturbation methods on the Navier-Stokes equations, the comparison and computation time studies will not be presented for the Channel flow example.
Then, an application of the methods to a flow in an L-shaped domain is presented. Finally,

their application to a reference problem in fluid mechanics, which is the LID-driven cavity for different Reynolds numbers, is provided. For the last two examples, the comparison between the time perturbation and resummation methods with each other and with the classical methods is performed.

3.4.6 Test problem 1: Channel flow

Presentation of the problem

A steady, laminar flow of a viscous fluid between two horizontal parallel plates separated by a distance, H, is defined as the Channel or Poiseuille flow. The flow is induced by a pressure gradient across the length of the plates, L, and is characterized by a symmetric 2D parabolic velocity profile about the horizontal mid-plane, as shown in Fig.(3.12). Note that there are no internal forces.

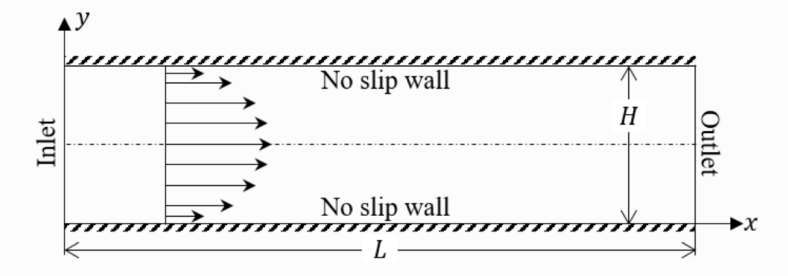


Figure 3.12 – Schematic of a 2D Poiseuille flow between two plates [115].

This problem has a well-known analytical solution in the steady-state.
First, the boundary conditions are imposed by the symmetry assumptions of the axisymmetric configuration. The pressure is fixed to $p = 8$ at the inlet $x = 0$, and $p = 0$ at the outlet $x = 1$. Then, the velocity is fixed to $\mathbf{u} = (0, 0)$ on the walls of the channel, that is at $y = 0$ and $y = 1$. Finally, we note that without further specifications, this defines the pressure drop and should result in unit maximum velocity at the inlet and outlet, as well as a parabolic velocity profile.
Using the boundary conditions, we end up with the following Navier-Stokes equations:

$$-\frac{1}{\rho}\frac{\partial p}{\partial x} + \mu\frac{\partial^2 u}{\partial y^2} = 0 \tag{3.75}$$

$$-\frac{1}{\rho}\frac{\partial p}{\partial y} = 0 \tag{3.76}$$

The Poiseuille equation gives the velocity of the fluid as a function of the radius at which it is placed in the pipe. It reflects the parabolic velocity profile through the channel,

symmetric about the central axis, in the following form:

$$u(y) = -6U_0 \left(\frac{y^2}{H^2} - \frac{y}{H} \right) \tag{3.77}$$

where U_0 is the characteristic inlet velocity. The pressure gradient along x is also analytical and is in the following form:

$$\frac{\partial p}{\partial x} = \frac{-12U_0\mu}{h^2} \tag{3.78}$$

Discretization

The finite element method is chosen to be used as a spatial discretization for this problem. Some finite element spaces are well known to be unstable for the Navier-Stokes equations. The use of continuous piecewise polynomials of the first degree for both velocity and pressure is an example of an unstable pair of finite element spaces. Using a pair of unstable spaces usually results in a pressure solution with spurious oscillations. Therefore, we use piecewise continuous quadratic elements for viscosity and piecewise linear elements for pressure. These elements combine to form the Taylor-Hood element. We use an uniform finite element triangular mesh over the 2D unit square $[0, 1] \times [0, 1]$. The mesh consists of cells, which in 2D are triangles with straight sides. The square is divided into $n_x \times n_x$ rectangles, each divided into a pair of triangles. The total number of triangles will be $2 \times n_x \times n_x$ and the total number of vertices will be $(n_x + 1) \times (n_x + 1)$. The size grid n_x is fixed to 16.

Simulation, implementation and results

The viscosity, the density, and the width of the channel are fixed to 1 ($\nu = \mu = 1$ and $\rho = 1$). To apply time perturbation and resummation methods, we calculate the series terms already evaluated. Then, we apply the algorithms of ANM, BPL, and IFS with continuation. The truncation order is fixed to $N = 6$ and the tolerance of ANM is fixed to $\delta = 10^{-4}$. We calculate the L2-norm error at the nodes to check the effectiveness of the time perturbation method and verify that our implementation is correct. The latter is presented in Fig.(3.14). The error decreases over time to zero and is approximately 10^{-3} at time $t = 0.5$s. The velocity profile at $t = 5$s via a probe-line is presented in Fig.(3.13). It is consistent with the analytical solution. The parabolic profile is also obtained with BPL and IFS. We do not present CPU time studies nor comparisons between methods for this test case presented just as the first validation of time perturbation resummation methods to the simple form of Navier-Stokes equations.

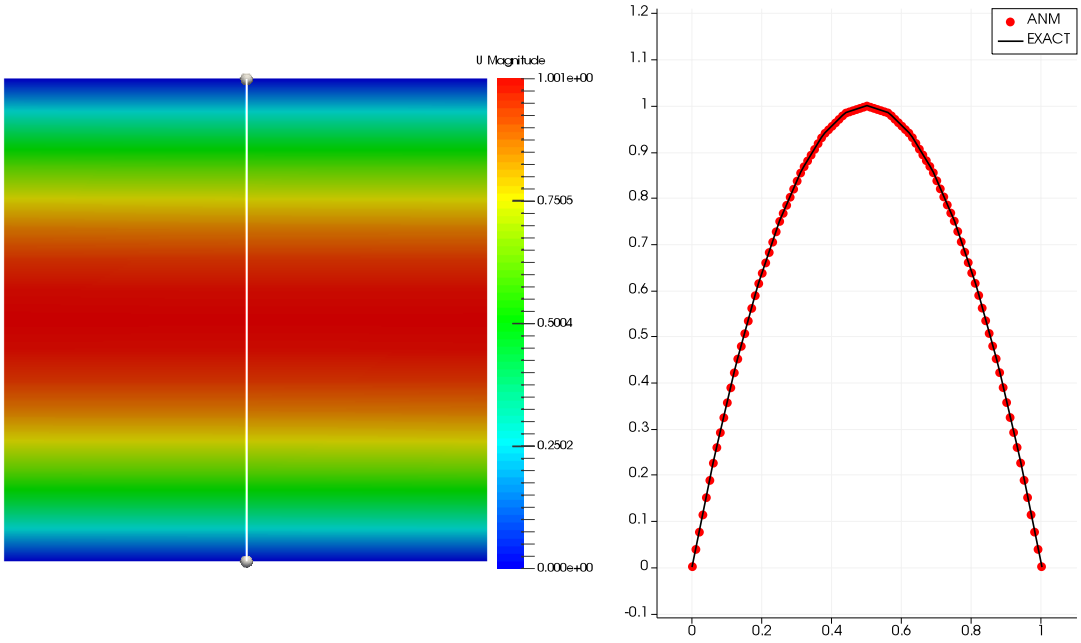


Figure 3.13 – Approximated solutions (the x-component velocity u and the y-component velocity v) using time perturbation method ANM for $N = 6$ and $\delta = 10^{-4}$ probed along a line (presented in white color in the figure). The exact solution is in black.

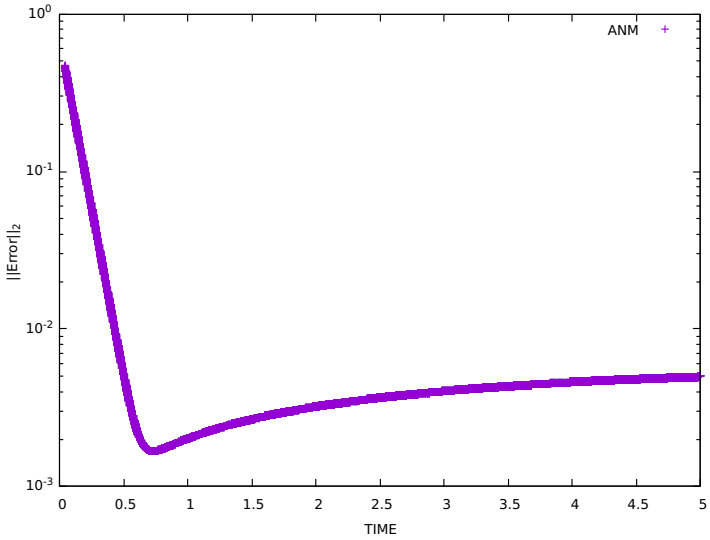


Figure 3.14 – L2-norm of the relative error between the solutions (the x-component velocity u, the y-component velocity v) obtained using time perturbation method ANM for $N = 6$ and the exact solution.

3.4.7 Test problem 2: Navier-Stokes equations on an L-shaped domain

Consider the Navier-Stokes equations governing the flow of an incompressible viscous fluid on an L-shaped domain. The latter is known as the L-shaped domain since it consists of a subset of the unit square while neglecting the upper right quadrant. This interesting shape of the domain has been used in a variety of articles and theses. One may cite [116]–[119].

Goal of this example

This example aims to study the efficiency of time perturbation and resummation methods for Navier-Stokes on an L-shaped domain geometry that has a corner singularity. The capabilities of these methods in capturing the behavior of 2D flows inside the L-shape geometry, for different values of the viscosity (thus for different Reynolds number), is tested.

Presentation of the problem

The flow is directed by an oscillating pressure $p_{in}(t) = \sin(3t)$ at the inlet $y = 1$ and a constant pressure at the outlet $x = 1$ denoted by $p_{out} = cte$ [99]. The flow boundary condition is set as free for the velocity at the inlet and outlet, and no-slip for the rest of the boundary. We note that the flow topologies also depend on the specific cavity shape, wall motions, and the Reynolds number Re. The geometry of the L-shaped cavity considered in the present example is shown in Fig.(3.15) and the length of the cavity L is fixed to 1.

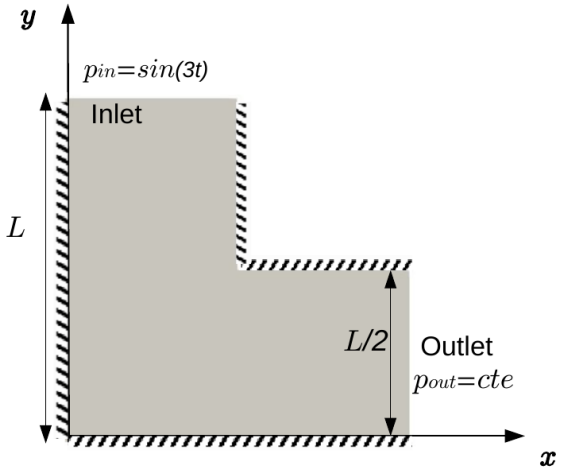


Figure 3.15 – Schematic of the L-shaped cavity and the problem considered in the current work.

Spatial discretization

For spatial discretization, the finite element method is employed using FEniCS. We use piecewise continuous quadratic elements for velocity and piecewise continuous linear elements for pressure. A mesh designed for the L-shape geometry, that can be found on the FEniCS website [99], is used here.

Simulation, implementation and results

For more simplicity the density of the fluid is fixed to $\rho = 1$, the kinematic viscosity is fixed to $\nu = 0.01$, the time step for the classic time integration RK1 scheme is $\Delta t = 10^{-3}$s. Before proceeding with the continuation of the temporal perturbation schemes, we are interested to see the behavior of the norms of the velocity and pressure series terms in Fig.(3.16) at time $t = 0.0002$s. It can be observed that the norms have extremely large values, reaching 10^{100} for $N = 30$.

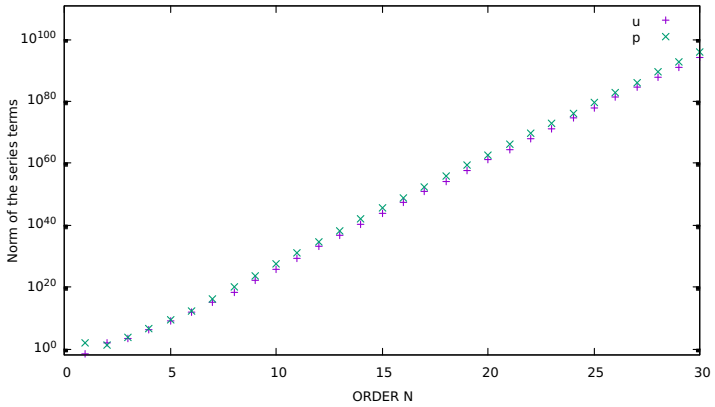
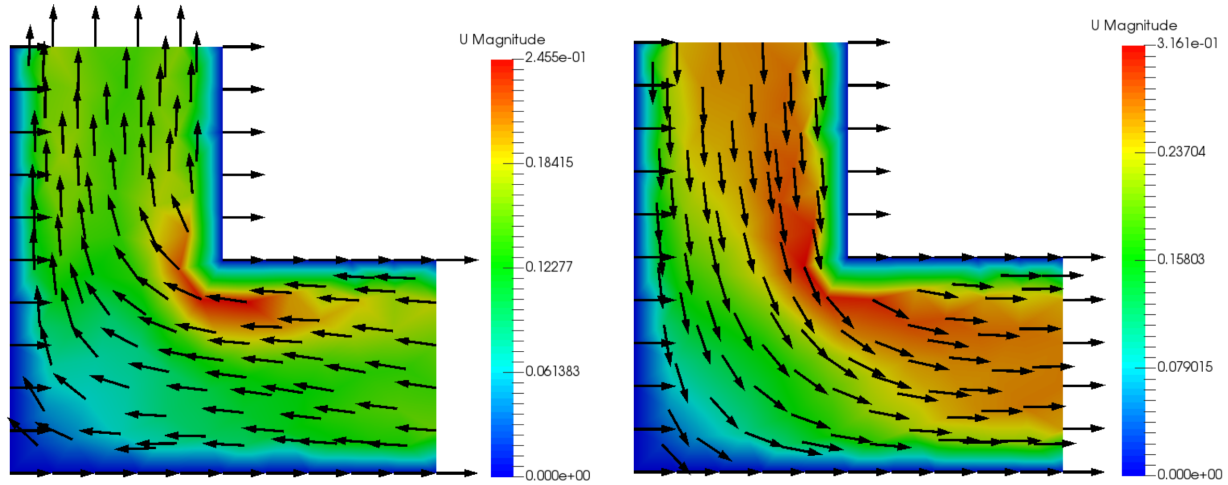


Figure 3.16 – Evolution of the norm of the series terms (velocity and pressure) with the truncation order N at $t = 0.0002$s for the Navier-Stokes equations on an L-shaped domain. The number of degrees of freedom is ndofs = 1147.

The simulation is computed up to a final time $t = 5$s, the truncation order is fixed to $N = 5$. The number of degrees of freedom for the used mesh is ndofs = 1147. We apply ANM, BPL, and IFS for this example with continuation. The momentum and the continuity residuals tolerances denoted respectively by ϵ_M and ϵ_c, are fixed to $\epsilon_M = 10^{-4}$ and $\epsilon_c = 9.898 \cdot 10^{-2}$.
The appropriate fluid behavior and velocity vectors are well respected using these methods as shown in Fig.(3.17) for $t = 2$s and $t = 3$s. For more details, we show the comparison of the solution velocity between ANM and RK1 for $t = 2$s and $\nu = 0.01$ in Fig.(3.18). Fig.(3.19) shows the velocity stream function for different values of the viscosity. Similar figures are obtained by the classical integration scheme. The same behavior verifies the

ability of these schemes to reproduce the solutions for different viscosities in the L-shaped domain.



(a) $t = 2$s (b) $t = 5$s

Figure 3.17 – Behavior of the flow in the L-shaped domain obtained by time perturbation method ANM for $N = 5$, and $\nu = 0.01$. The same figure is obtained by BPL and IFS. The black arrows design the velocity vector in the domain.

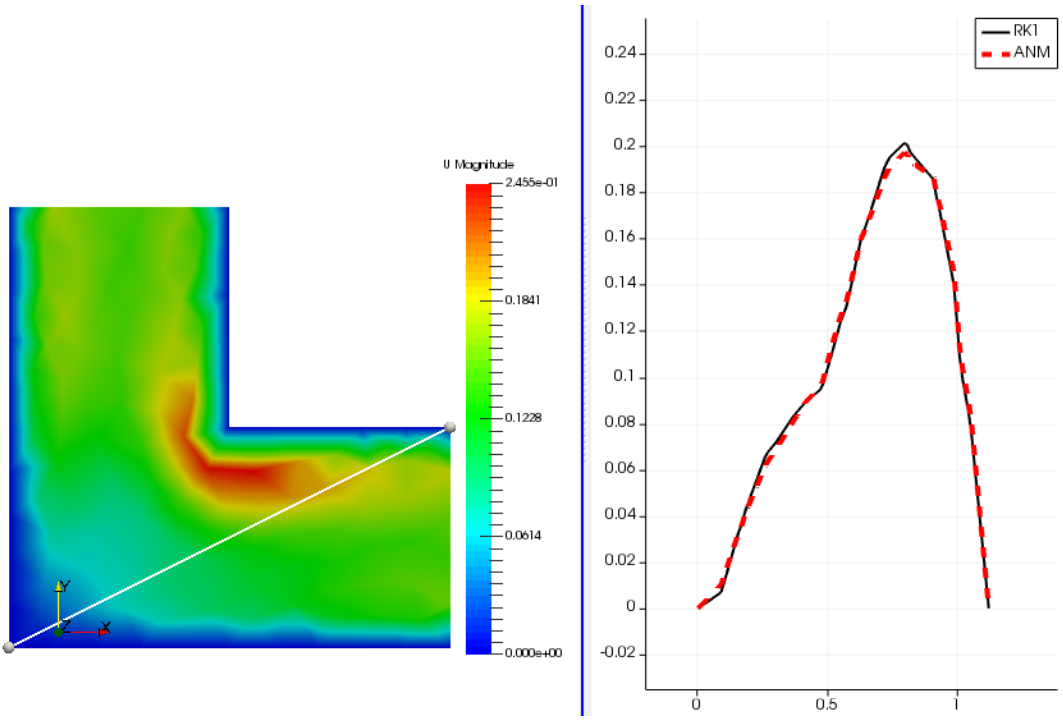
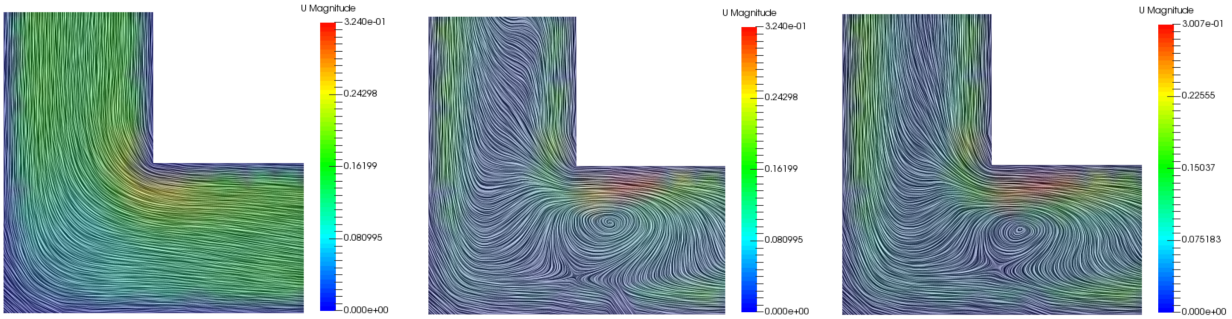


Figure 3.18 – Approximated solutions (velocity $\mathbf{u}$) using time perturbation method ANM for $N = 5$ probed along a line (presented in white color in the figure) for $t = 2$s, and $\nu = 0.01$. The same figure is obtained by BPL and IFS. The RK1 solution is in black.



(a) $\nu = 0.01$ (b) $\nu = 0.002$ (c) $\nu = 0.003$

Figure 3.19 – Velocity streamlines for different values of the viscosity ν, obtained by time perturbation method ANM. Same results are obtained with time perturbation-resummation methods BPL and IFS for $t = 2$s.

Comparisons

We are interested in comparing the efficiency of the methods for this test case first between the classical methods and the time perturbation-resummation methods, and then between these latter for an arbitrary final time of $t = 2$s and a viscosity $\nu = 0.01$. For the sake of comparison, the CPU number is set to 1 in FEniCS and the generalized minimal residual method (GMRES) [120] which is an iterative method for the numerical solution of an indefinite nonsymmetric system of linear equations, is used for the resolution. The residual criterion is used for continuation, and dichotomy is used to find t_{max}. The classical schemes residual norms are calculated and will be used as the prescribed residual norm tolerances.

1. Classical scheme:

The classical scheme RK1 with a fixed discrete-time step $\Delta t = 10^{-3}$s needs 2000 steps to reach the final time with a total CPU time equal to 668.82s. The number of degrees of freedom for the used mesh is ndofs = 1147. The residual momentum norm for RK1 during the calculation is $||Res_M||_2 \approx 10^{-4}$ and the continuity residual norm is $||Res_C||_2 \approx 9.898 \cdot 10^{-2}$. These values of the residual norm are used as the momentum and continuity residuals tolerances denoted respectively by ϵ_M and ϵ_c, to perform comparisons of perturbation methods. Therefore, for the following comparisons, ϵ_M and ϵ_c are fixed to $\epsilon_M = 10^{-4}$ and $\epsilon_c = 9.898 \cdot 10^{-2}$.

2. Time perturbation-resummation methods:

Statistics to reach $t = 2$s for the Navier-Stokes equations on an L-shaped domain, using ANM, BPL and IFS with different truncation orders N, are presented in Tab.(3.11), Tab.(3.12) and Tab.(3.13). The viscosity is $\nu = 0.01$. The number of degrees of freedom is

ndofs = 1147. The number of Gauss used for BPL is $N_G = 6$. For this example, one can see from Tab.(3.11), Tab.(3.12) and Tab.(3.13) that the t_{max} is almost constant when the truncation order varies for each of these methods. BPL has the largest t_{max}, the fewest number of steps, and the smallest CPU time. One can as well notice that the computation of series terms for the Navier-Stokes equation is considered time-consuming. Concerning IFS, we only made the statistic for $N = 5$ because the code diverges for large truncation orders.

N	Steps	$< t_{max} >$	Total (s)	Series (s)	Residual Pol (s)	#RES
5	4084	$4.897 \cdot 10^{-4}$	406.01	123.85	271.59	133772
10	4125	$4.848 \cdot 10^{-4}$	574.25	301.17	261.79	119620
15	4132	$4.484 \cdot 10^{-4}$	840.57	543.44	285.22	119803
20	4134	$4.837 \cdot 10^{-4}$	1091.62	801.37	278.50	119786
25	4134	$4.837 \cdot 10^{-4}$	1474.29	1170.54	292.23	119786

Table 3.11 – Statistics to reach $t = 2$s for the Navier-Stokes equations on an L-shaped domain, using ANM with different truncation orders N, for a fixed momentum residual tolerance $\epsilon_M = 10^{-4}$ and a continuity residual tolerance $\epsilon_c = 9.898 \cdot 10^{-2}$. The viscosity is fixed to $\nu = 0.02$.

N	Steps	$< t_{max} >$	Total (s)	Series (s)	Residual BPL (s)	#RES
5	1604	$1.24 \cdot 10^{-3}$	162.57	47.47	110.77	32186
10	1604	$1.24 \cdot 10^{-3}$	253.15	115.45	133.40	32186
15	1604	$1.24 \cdot 10^{-3}$	358.53	201.10	154.10	32186
20	1604	$1.24 \cdot 10^{-3}$	495.32	313.53	177.46	32186
25	1604	$1.24 \cdot 10^{-3}$	625.85	420.84	200.01	32186

Table 3.12 – Statistics to reach $t = 2$s for the Navier-Stokes equations on an L-shaped domain, using BPL with different truncation orders N, for a fixed momentum residual tolerance $\epsilon_M = 10^{-4}$ and a continuity residual tolerance $\epsilon_c = 9.898 \cdot 10^{-2}$. The viscosity is fixed to $\nu = 0.02$.

N	Steps	$< t_{max} >$	Total (s)	Series (s)	Residual IFS (s)	#RES
5	5233	$3.82 \cdot 10^{-4}$	2760.86	348.22	2406.64	104840

Table 3.13 – Statistics to reach $t = 2$s for the Navier-Stokes equations on an L-shaped domain, using IFS with $N = 5$, for a fixed momentum residual tolerance $\epsilon_M = 10^{-4}$ and a continuity residual tolerance $\epsilon_c = 9.898 \cdot 10^{-2}$. The viscosity is fixed to $\nu = 0.02$.

3. Comparisons between RK1, ANM, BPL, and IFS:

If we compare the results of ANM, BPL, and IFS in Tab.(3.11), Tab.(3.12) with the classical scheme RK1 that requires 2000 steps and a total CPU time of 668.82s to reach the final time, one can deduce:

- For the different truncation orders tested, BPL requires fewer steps than the classical scheme RK1, whereas ANM and IFS require significantly more steps than RK1 (2000 steps).
- The CPU time of BPL for different truncation orders is less than that of the classical scheme (which is 668.82s). For example, for $N = 5$, BPL needs 162.57s to reach the final time, whereas the classical scheme RK1 takes 668.82s for the same set of parameters.
- For $N < 10$, ANM requires less time than the classical scheme, while for $N > 10$, the classical scheme becomes faster.
- IFS is slower than the classical scheme RK1 and requires many more steps for even a small truncation order.

Study for different mesh size

We are now interested in investigating the sensitivity of these methods to the mesh size. We perform the study for a fixed truncation order of $N = 5$ and three different refining mesh sizes, evolving three different degrees of freedom ndofs. The residuals of the classical scheme are calculated and used here also as residual tolerance for the temporal perturbation resummation methods computation.

Tab.(3.14) shows the typical effect of mesh refinement on the total CPU time of the methods. This table shows that IFS is the most sensitive to the mesh change. BPL appears to be slightly more sensitive to the mesh change than ANM and RK1, but it is still the fastest in terms of CPU time among the other methods.

ndofs	Total(s) RK1	Total(s)ANM	Total(s) BPL	Total(s) IFS
1147	668.82	406.01	162.57	2708.39
4379	701.36	498.41	293.49	4389.90

Table 3.14 – The total CPU time required to reach $t = 2$s for the Navier-Stokes equations on an L-shaped domain, using RK1 with discrete-time step $\Delta t = 10^{-3}$s, ANM, BPL and IFS for $N = 5$. Different number of degrees of freedom ndofs are considered. The viscosity used is $\nu = 0.01$.

3.4.8 Test problem 3: Lid-driven cavity by MAC method

The Lid-driven cavity example is a primordial problem in fluid dynamics that has attracted the attention of the scientific community for a long time. It was first suggested by Ghia et al. [121]. Over the years, the problem has given rise to a large number of papers, the majority of the latter have been devoted to the development of computational

algorithms for solving the problem, focusing on the two-dimensional rectangular or three-dimensional cubic form of the problem. For the sake of brevity, we will limit our analysis in this example to calculations in two-dimensional rectangular.
This problem serves as a classical reference model for evaluating the effectiveness of numerical methods as well as for studying fundamental aspects of incompressible flows and the pure physics of two-dimensional flows. It is a suitable problem to be tested for several reasons. First, there is a wealth of literature to compare with. Second, the laminar solution is steady. Third, the boundary conditions are very easy. In addition, the fully developed flow exhibits almost all fluid mechanics phenomena, with increasingly complex aspects emerging as the Reynolds number increases. One can cite AbdelMigid et al. [122], who provides a recent and comprehensive review of the literature on this subject, presenting and discussing the work of several authors. Note that the good data set for comparison is that of Ghia et al. [121], as it includes tabular results for different Reynolds numbers. These simulation results are obtained by the non-primitive variable approach.

Goal of this example

The Lid-driven cavity is chosen to test the effectiveness of time perturbation and resummation methods, as it is a common benchmark problem in fluid dynamics and has been long used as a validation test for new numerical methods. The goal of this example is to test the capabilities of time perturbation and resummation methods in capturing the behavior of 2D flows inside confined volumes for different Reynolds numbers. We note that this example does not aim to present an optimized numerical model neither for CPU time nor for computational memory consumption. An indication of the comparison between the perturbation methods will be presented.

Presentation of the problem

The Lid-driven cavity problem is modeled by the Navier-Stokes equation presented in Eq.(3.37) and Eq.(3.38). It has no-slip boundary conditions on all walls. On the bottom wall and the side walls, $\mathbf{u} = (0, 0)$, while on the top wall (the lid), $\mathbf{u} = (U = 1, 0)$, where the driving velocity U in x-direction is one of the input parameters of the simulation that is chosen to be $1m/s$. It consists of a cubic cavity, whose lid (the top layer) has a uniform velocity, and whose interior is filled with a fluid undergoing incompressible flow. Here, the cavity of size $[0, 1]$ is considered. Note that, we simplify the problem by not considering the gravitational field force. The schematic of Lid-driven cavity flow is presented in Fig.(3.20).

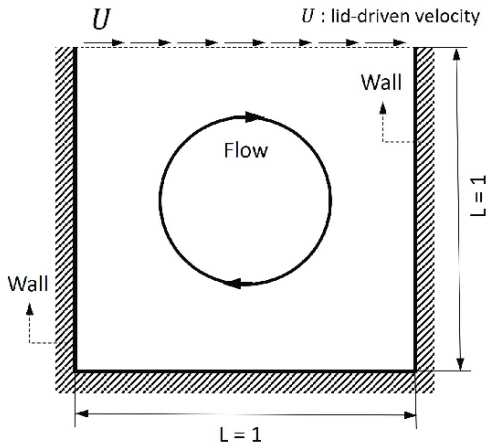


Figure 3.20 – Schematic of the Lid-driven cavity flow [123].

Spatial discretization

Regarding spatial discretization, a wide variety of different numerical methods have already been applied to the Lid-driven cavity problem. Here, we use the most popular finite difference method centered on a shifted grid (also known as the MAC mesh or MAC method) for the approximation of the Navier-Stokes equations. A brief background on the MAC method is presented in the following.

Background-MAC method: The MAC method is well-known for its simplicity, its efficiency, and its remarkable mathematical properties. It is a finite difference technique based on a staggered grid. It was developed by Harlow and Welch [98] in 1965 specifically for free surface flows. They used a particle marker to mark the cell containing the fluids and used interpolation to track the movement of the surface. One of the most notable features is the use of virtual Lagrangian particles, whose coordinates are saved and which move from one cell to the next based on the latest evaluated speed field.
Some of the notable characteristics of the MAC method include the superposition of mesh rectangular cells on the flow domain and the use of virtual particles to represent the existing fluid. The particles move from one cell to another as a function of flow velocities, keeping track of the fluid location. If a cell contains a certain number of particles, it is considered to contain fluid. Thus, in the MAC method, the flow domain is divided into a certain number of cells that are labeled as empty, full, surface, or border cells, and the labels must be updated as time evolves.
The idea of MAC, Marker, and Cell, is to place the unknowns $(\mathbf{u}, p) = (u, v, p)$ in different locations. Fig.(3.21) shows an illustration of the MAC mesh near the border [124]. Specifically the pressure p is located in the center of each cell and evaluated at square points $(i \mp 1/2, j \mp 1/2)$, the x-component velocity u on the middle points of vertical edges at

the triangle points $(i, j \mp 1/2)$ and the y-component velocity v on the middle points of horizontal edges at the circle points $(i \mp 1/2, j)$.

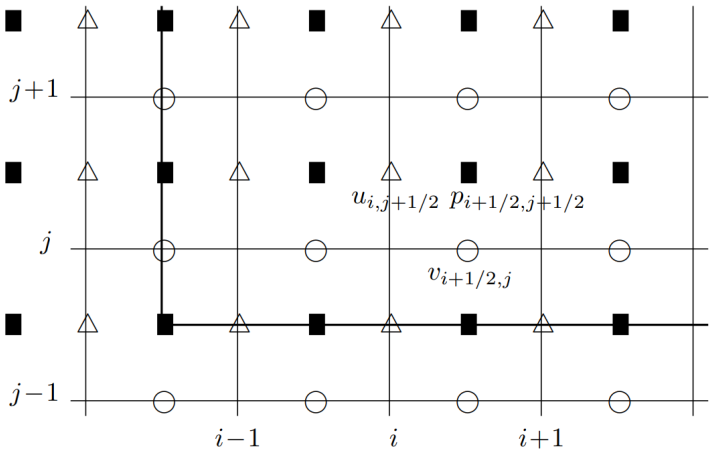


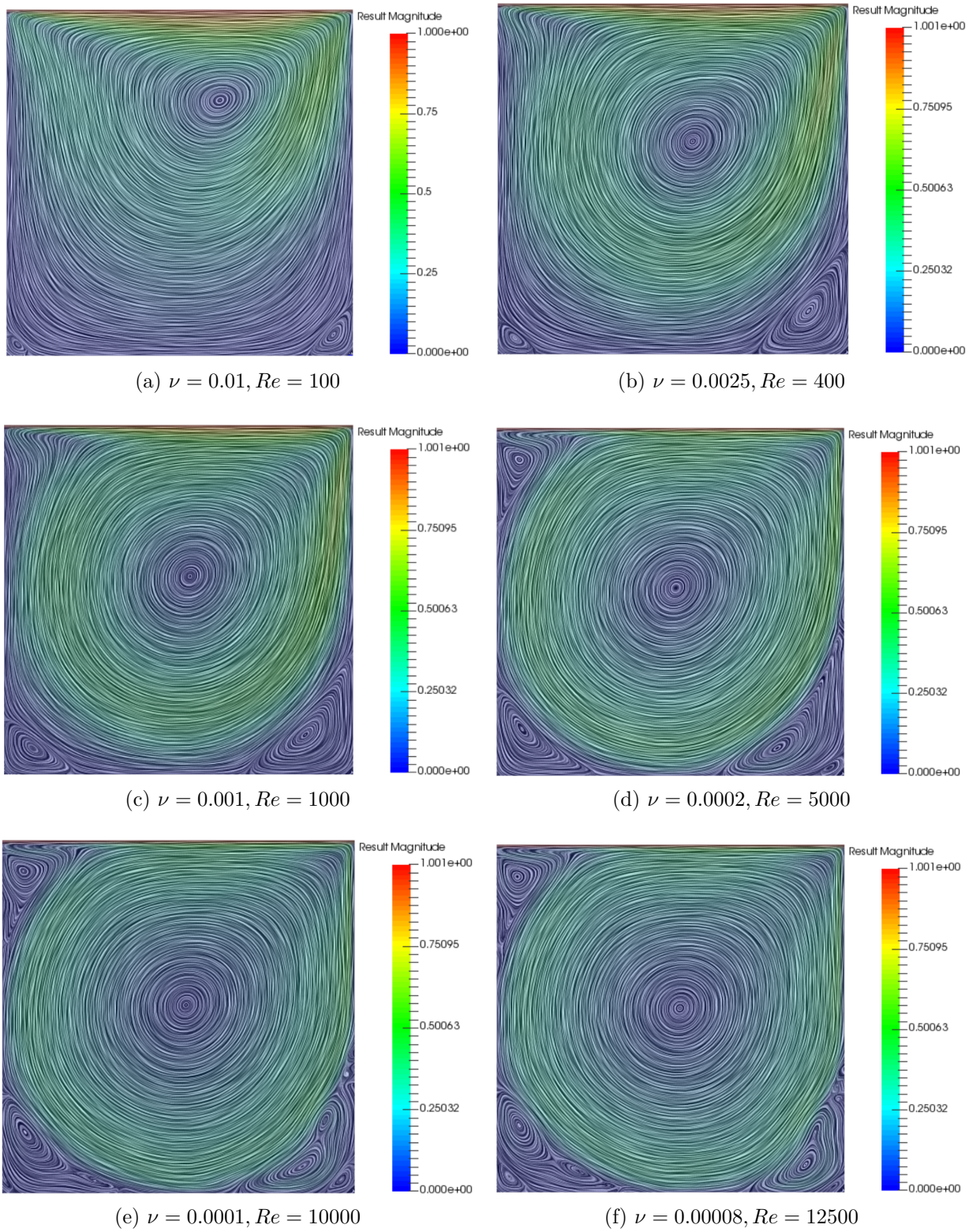
Figure 3.21 – The MAC mesh [124].

Simulation, implementation and results

Here, the time integration scheme Euler (RK1) with a discrete time-step $\Delta t = 10^{-4}$s is used for this problem.
The method has been numerically implemented with a solver coded in C++ language. We note that Meijer-G has not been tested for this problem (Meijer-G function is not available on C).
As already mentioned, the driving speed and the fluid density ρ are fixed to 1 in this problem. The following grid size 64×64 is used. Concerning the viscosity of the fluid, various parameters leading to different Reynolds numbers have been tested. The obtained results will be analyzed in the following.

The truncation order is fixed to $N = 10$. The simulation is running until a final time $t = 760$s by time perturbation method ANM, BPL, and IFS with continuation. We will first present the velocity stream function for different Reynolds numbers (Re) to show the ability of time perturbation and resummation methods to track the correct behavior of the fluid. Then, we will perform a detailed study of these methods, for this example for a Reynolds number fixed to $Re = 12500$.
The velocity stream function is illustrated in Fig.(3.22) for various Reynolds numbers. These velocity curves match the results obtained by Wahba [125] and Arumuga Perumal and Anoop Dass [126] for very similar Reynolds numbers. The flows correspond to those found in the literature [121]. Therefore, as can be seen, ANM, BPL, and IFS allow to determine the solution up to very high Reynolds numbers and show the same trend and

same flow behavior. A sequence of vortices appears as the Reynolds number increases. As can be observed in Fig.(3.22), the primary vortex forms near the cavity geometric center, while secondary vortices form near the cavity upper left and lower right corners. Higher Reynolds numbers lead to a much greater dispersion of the result, in particular for larger node arrangements. The flow is steady up to a Reynolds number close to 8000 and then becomes periodic in time.



(a) $\nu = 0.01, Re = 100$ (b) $\nu = 0.0025, Re = 400$

(c) $\nu = 0.001, Re = 1000$ (d) $\nu = 0.0002, Re = 5000$

(e) $\nu = 0.0001, Re = 10000$ (f) $\nu = 0.00008, Re = 12500$

Figure 3.22 – Velocity stream function plots for different Reynolds numbers Re, obtained by time perturbation method ANM. Same streamlines are obtained by BPL and IFS and are not presented in the figure.

Detailed numerical study for $Re = 12500$:

A detailed numerical study for $\nu = 0.00008$, which lead to $Re = 12500$, is presented.

1. Classical scheme:

Evolution of the v component velocity and the phase portrait for a probe point P1, located in the lower left corner, are presented in Fig.(3.23) obtained by the classic RK1 scheme. As can be seen in this figure, the limit cycle is reached.

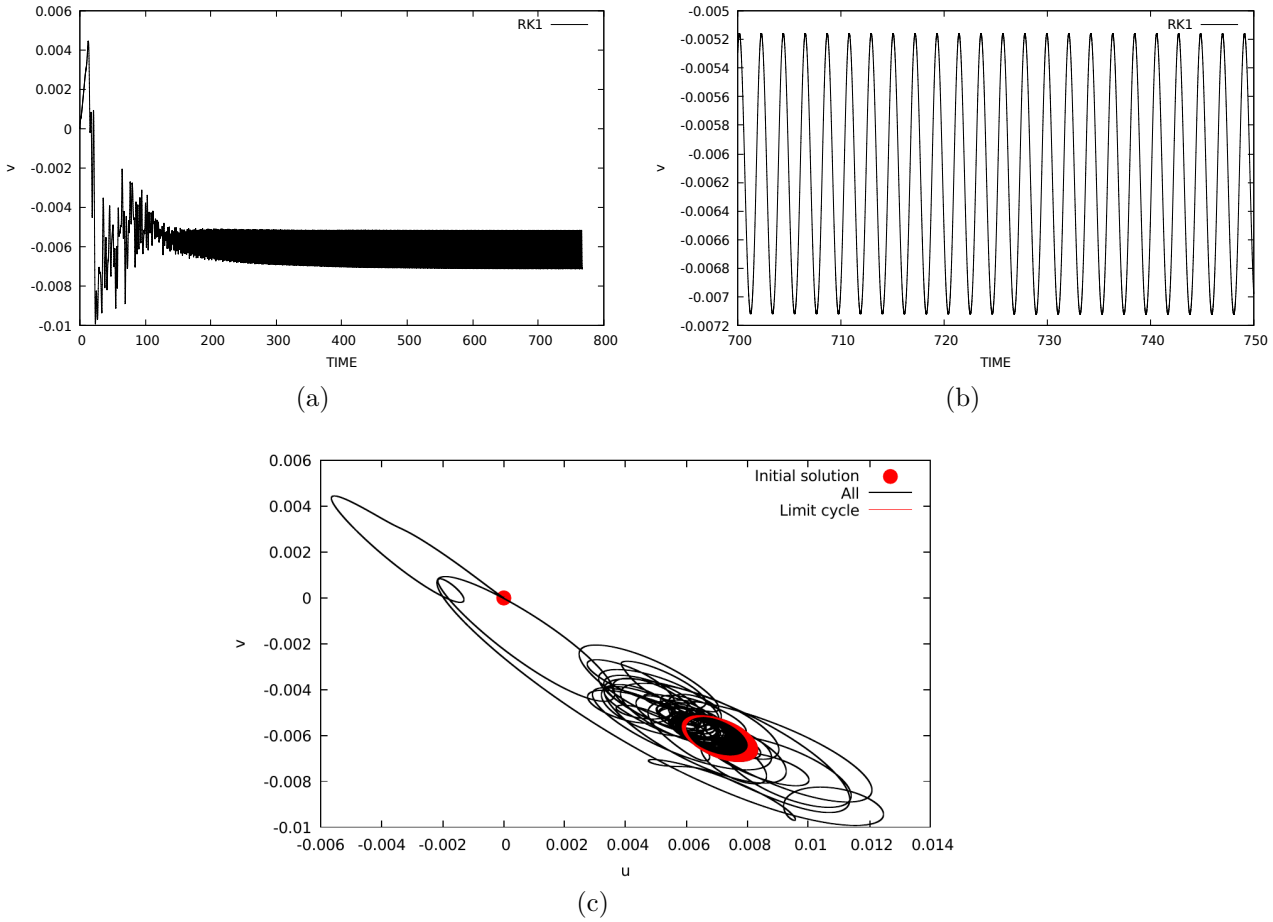
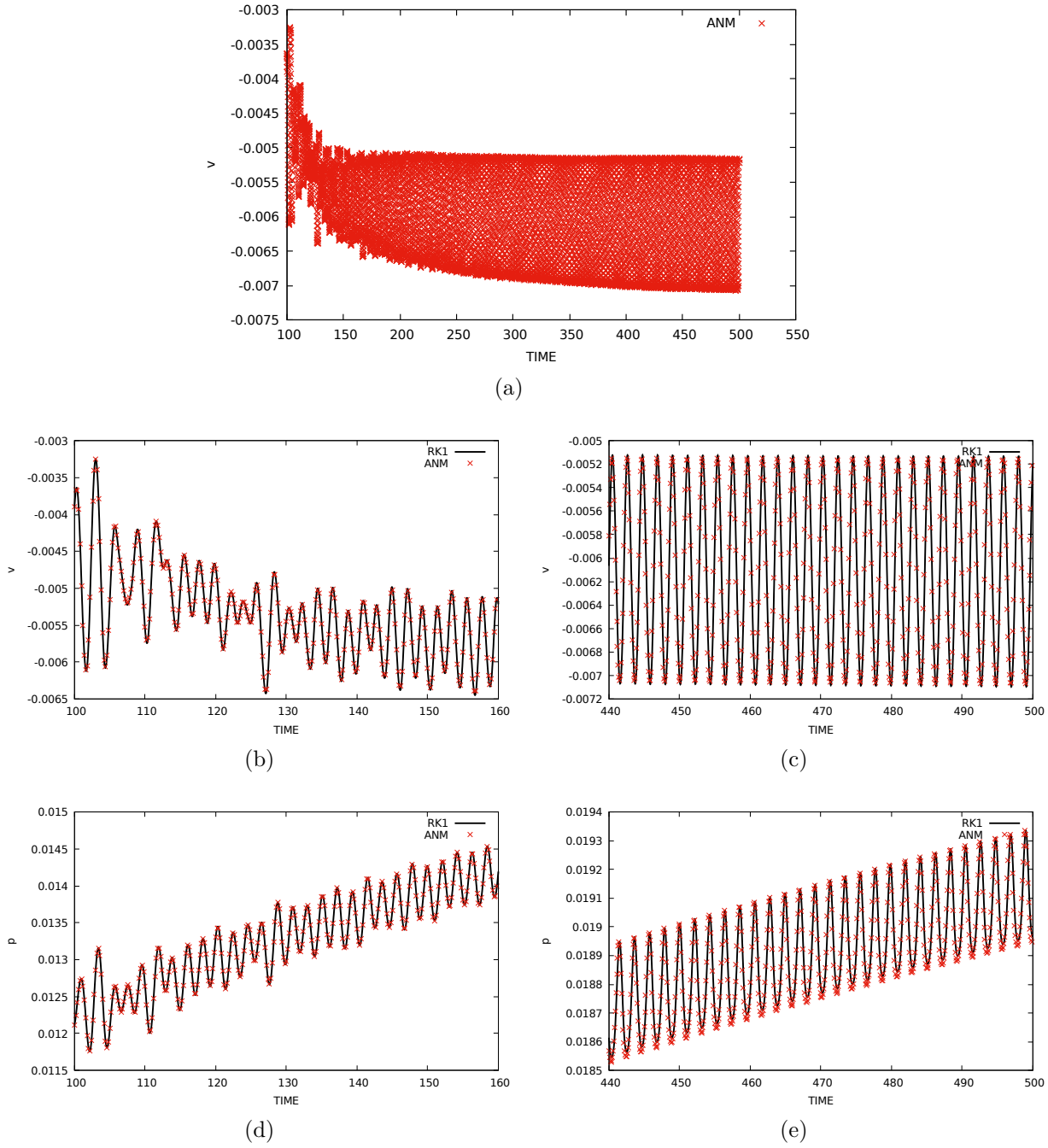


Figure 3.23 – Evolution of the v velocity and phase portrait for the probe point P1$(x(i = 5), y(j = 5))$ obtained by the classic scheme RK1 with a discrete-time step $\Delta t = 10^{-4}$s for a fixed Reynolds number $Re = 12500$.
(b) is a zoom plot of (a).

Therefore, one might wonder whether time perturbation-resummation methods are able to reproduce the solution with such a large number of Reynolds, for a large final time. We have verified that the fluid behavior is well respected using these methods, by the velocity stream function plots. Two tests are carried out to respond to the question. The first one is for an initial time $t_0 = 100$s, i.e. a starting point that is not located yet on the limit cycle. The second one is for an initial time $t_0 = 400$s i.e a starting point that

is located on the limit cycle.

2. Time perturbation and resummation methods for an initial time $t_0 = 100$s and $Re = 12500$:
The truncation order is fixed to $N = 10$. The simulation is running from an initial time $t_0 = 100$s until a final time $t = 700$s by time perturbation method ANM, BPL, and IFS with continuation.
The velocity solution v obtained by ANM based on the continuation proposed by Cochelin [40], with a tolerance $\delta < 10^{-32}$ (the choice of δ will be presented in details in the next subsection), is displayed in Fig.(3.24a) for a time interval $[100\,;500]$s. The comparison between the classical solutions (velocity and pressure) and the ANM solution for t in $[100;160]$s and $[440\,;500]$s are presented in Fig.(3.24b), Fig.(3.24c), Fig.(3.24d), Fig.(3.24e). These figures show the ability of ANM to reproduce the solution for an initial time $t_0 = 100$s and for $Re = 12500$. Regards BPL and IFS, their continuation is based on the residual criterion. The residual evaluated here has a wide range of values, and a corrector must be set up to correct the inappropriate residual values. As a result, we have improved these methods by specifying a suitable t_{max} during the calculation. The same results of Fig.(3.24) are obtained by BPL and IFS.



(a)

(b)

(c)

(d)

(e)

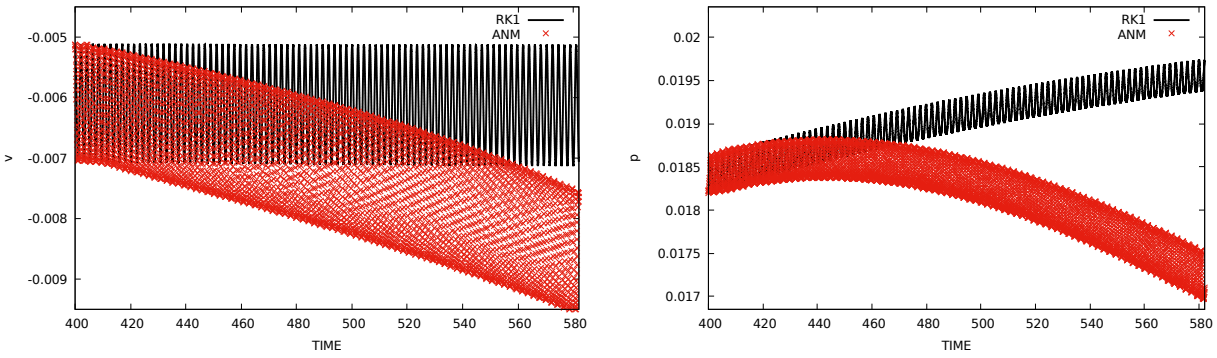
Figure 3.24 – Approximated solutions (velocity v and pressure p) using time perturbation method ANM for $N = 10$, $\nu = 0.00008$ ($Re = 12500$) compared to the classic RK1 solution for different time intervals.

3. Time perturbation and resummation methods for an initial time $t_0 = 400$s and $Re = 12500$:

For an initial point $t_0 = 400$s that is already located on the limit cycle, we are interested in determining if the methods maintain the solution on the limit cycle for a large time, for $Re = 12500$.

Without continuation: Before starting the simulation of the continuation methods, we show the behavior of the velocity series terms $\mathbf{u}$ for an initial time $t_0 = 400$s, a Reynolds number fixed to $Re = 12500$ and a truncation order that varies from $N = 0$ to $N = 35$ in Fig.(D.1), Fig.(D.2) and Fig.(D.3) presented in Appendix D. One can deduce that the norm of the velocity series terms increases with the truncation order and reaches a magnitude equal to 10^{38} for $N = 35$.

With continuation: ANM with continuation is now applied from $t_0 = 400$s using a truncation order of $N = 10$ and a tolerance $\delta = 10^{-32}$. As can be seen in Fig.(3.25), the ANM solution is shifted from the RK1 solution. The limit cycle is no longer respected. Note that the evolution of the validity domain t_{max} of ANM, corresponding to this choice of $\delta = 10^{-32}$ is presented in Fig.(3.26). It oscillates between 0.00009s and 0.00011s. Therefore, to get a smaller t_{max} than 0.00011s, one need to decrease the tolerance δ more.



(a) (b)

Figure 3.25 – Approximated solutions (velocity v and pressure p) using time perturbation method ANM for $N = 10$, $\delta = 10^{-32}$ compared to the classic RK1 solution for a time interval $[400; 500]$s.

Now, the tolerance δ of ANM is decreased to $\delta = 10^{-50}$. The result figure obtained is presented in Fig.(3.27). With the choice of a very small tolerances, one can confirm that the solutions become in a good agreement with the classic RK1 scheme and the limit cycle is well respected. The solution in the interval time $[570; 630]$s, presented in Fig.(3.27) performs well.

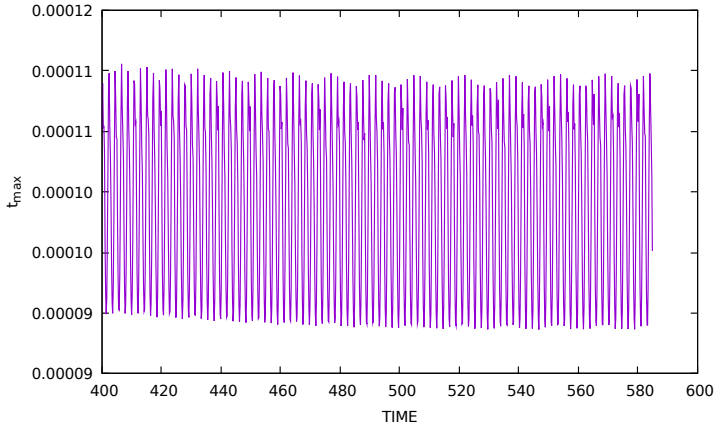
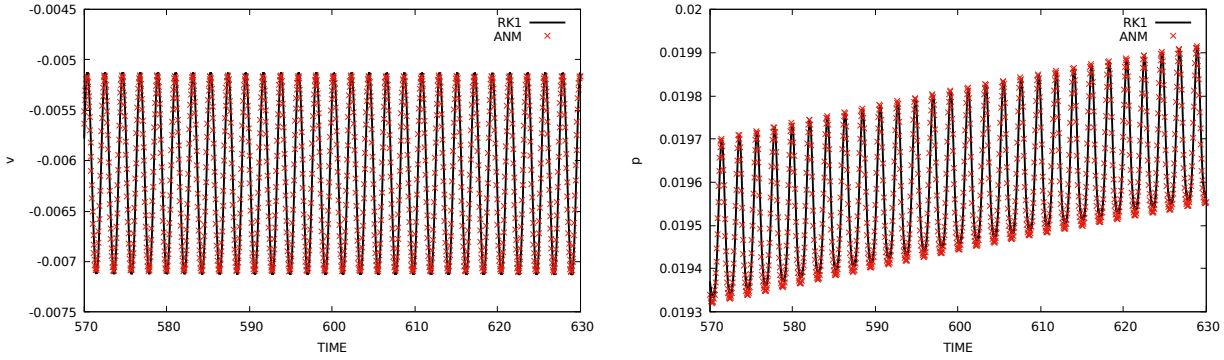


Figure 3.26 – t_{max} evolution for time perturbation method ANM, for $N = 10$, and $\delta = 10^{-32}$ for $t \in [400; 500]$s.



(a) (b)

Figure 3.27 – Approximated solutions (velocity v and pressure p) using time perturbation method ANM for $N = 10$, $\delta = 10^{-50}$ compared to the classic RK1 solution for a time interval $[570; 630]$s.

We are interested to see if time perturbation resummation methods improve the ANM solution.

Indication of the comparison between RK1, ANM, BPL, and IFS

We have noticed that the residual of the temporal perturbation methods has very large values overtime for this test case. Therefore, we forced the upper right bound of the validity domain t_{max}. The goal is to provide an indication of comparison between the methods and to advance the solution with the continuation technique. Due to the inappropriate residual value, t_{max} was not calculated from the residual criterion, as we have done for all the test cases in this manuscript. The forced t_{max} was chosen to be the most appropriate so that the solution does not deviate from the limit cycle. Therefore, it should be noted that the following comparisons are only for a forced t_{max}.

The selected time interval is $[400\,;405]$s, this is the interval to obtain 2 periods.
From Tab.(3.15), one can indicate that for this test case, ANM and IFS are much more time consuming than BPL. RK1 seems faster than BPL but we note that this study is just an indication of comparison.
We recall that the three methods are efficient in terms of feasibility to reach the final time $t = 750$s, with a forced corresponding t_{max} for different Reynolds number.

Total(s) RK1	Total(s) ANM	Total(s) BPL	Total(s) IFS
32.942	612.271	179.430	890.484

Table 3.15 – The total CPU time required for the selected time interval $[400\,;405]$s, for the LID driven cavity problem, using ANM, BPL and IFS, for $N = 6$. The Reynolds number is $Re = 12500$. The grid used is 64×64. The discrete-time step of RK1 used is $\Delta t = 10^{-4}$s. The computation study of ANM, BPL and IFS are only for a forced t_{max} and are just presented as an indication.

Conclusion of the Lid-driven cavity example

This study was made to test the efficiency and accuracy of the time perturbation and resummation methods. The latter is applied to the Lid-driven cavity problem with a Reynolds number that varies in the low and high range, from 100 to 12500. The calculated results are in good agreement with the literature. A numerical study of two-dimensional (2D) incompressible steady flow in the Lid-driven cavity for $Re = 12500$ was carried out in detail in the present study. Time perturbation and resummation methods have shown great efficiency in terms of stability, robustness, and accuracy for steady solutions at

high Reynolds numbers. The various computations at high Reynolds numbers show the robustness and efficiency of the approximation and the computational method. For the comparative studies, we only indicated the CPU time between the methods, for a forced t_{max} rather than a more real robust study: We forced the upper bound of the validity domain t_{max} to advance the continuation of the solution instead of using the residual criterion. This is due to the large residual value of the time perturbation methods over time. Since the starting residual is not optimal, we need to develop a method to have a corrector, and then have an appropriate value for the residual for every step.

3.5 Conclusion

In this chapter, a study was conducted on the performance of temporal perturbation and summation methods (ANM, BPL, IFS, and MG) to numerically solve the heat equation, Burges equation, and Navier-Stokes equations. Various tests have been carried out in two-dimensional geometry. We used the classical central finite difference method, the MAC method, and the finite element method for spatial discretization.
For the Navier-Stokes equations, we started with simple configurations where the analytical solution is known so that we could compare the solutions and evaluate errors (e.g. The Poiseuille problem). Then, we performed the simulations on more common configurations such as the flow around a domain of shape L and the flow of the cavity driven for different Reynolds numbers.

In all of the numerical tests conducted in this chapter, it can be proven that ANM, BPL, and IFS have demonstrated their ability to provide appropriate solutions for PDEs even for a large final time, for high numbers of degrees of freedom and when changing the parameters of the problem (the grid size, the viscosity, the Reynolds number). The obtained results were in good agreement with the literature. We have shown also that these methods can be associated with various spatial discretization techniques.
In our time computation studies, we found that when we compare the time perturbation and resummation schemes for the same comparison criterion and parameters, we revealed that: IFS is the most expensive in terms of time, BPL was faster than ANM for all of the examples presented in this chapter with a larger t_{max} requiring fewer steps to reach the final time. When compared to the classic RK1 scheme, RK1 was significantly faster than IFS in most cases. For the processed heat equation and the Navier-Stokes case on an L-shaped domain, RK1 was much slower than ANM and BPL. In the Burgers equation case, RK1 was slightly faster than BPL but much faster than ANM. Because the residual had incorrect values in the LID-driven cavity case for $Re = 12500$, only a comparison

indication was presented rather than a true comparison based on the residual criterion. According to the studies, calculating the series terms is time-consuming for some cases of the Navier-Stokes problems. As a result, we must devise a method to accelerate the calculation of series terms for time perturbation and summation methods of Navier-Stokes equations.
In terms of changing the grid and the mesh size, we observed that IFS is the most sensitive to the mesh. Mesh sensitivity is about the same for ANM and BPL. For some test cases with a very fine mesh, RK1 becomes faster than the other schemes.

Regarding the Meijer-G approximant, we wanted to explore the latter for temporal PDE problems. The problem of the 1D heat equation has been carried out by Meijer-G with the finite difference method. For this example, we have shown the efficiency of the Meijer-G for a series vector using the Gram-Schmidt procedure. Other examples, on the other hand, revealed some issues for specific domain areas and truncation orders. The Meijer-G sum was discovered to behave erratically in some areas. The cause is still being investigated. This could be due to the small value of t_{max} in the Meijer-G sum, or to the fact that the Meijer-G function implementation does not always provide correct answers in some specific parameter cases. This is also mentioned in Viktor T. Toth's technical notes (for Maple implementation) [104]. The obtained results open a perspective of the application of Meijer-G for temporal problems.

Chapter 4

Towards application to Fluid-Structure Interaction problems

We recall that our long term future goal is to establish the necessary adaptations of this new class of time perturbation and resummation schemes to solve full Fluid-Structure Interaction (FSI) problems with free surfaces. The Navier-Stokes equations and capture interface methods constitute the basis of such problems. Since it is known that the time series for Navier-Stokes equations is divergent, resummation techniques of divergent series are required to obtain accurate numerical solutions for such models. Hence, FSI models based on the coupling of Navier-Stokes equations, followed by the most popular technique for capturing the interface which is the level set, are chosen. To serve our future purpose, progression difficulties sould be done in stages. In Chapter 3, the applications of the time perturbation appraches have been applied on some examples of the Navier-Stokes equations. In this chapter, we limit their study on the level set equation alone because there are many points to consider, test, and validate just for the separate analysis of the level set alone by time perturbation methods before its coupling with the Navier-Stokes equations. Therefore, this chapter aims to focus on time perturbation and resummation methods applied to the level set evolution equation subjected to a prescribed velocity field.
We will first present a brief review of interface capture methods to clarify the motivation for choosing level set methods to be handled by the temporal perturbation techniques. We recall that the Meijer-G resummation will not be applied to the level set evolution equations since the numerical resolution of the level set has been written in the C programming language where the numerical access of the Meijer-G function $G^{p,q}_{m,n}$ is not available.

Review and motivation

Two main approaches are broadly used in the literature to model the free surfaces namely, Lagrangian methods (e.g. front tracking [127]) and Eulerian methods [128]–[131].

In Lagrangian methods, interface location is explicitly known and therefore the interfacial boundary conditions can be easily implemented. However, a primary drawback of this method is that once the interface topology starts changing, "surgical" procedures are needed to smooth and distribute the interfacial elements used to represent the interface [130]. Procedures of the Lagrangian methods are problematic when interfaces merge or split [132], specifically in three dimensions (3D), [132]. On the other hand, in Eulerian methods, interface location is implicitly represented by a scalar function in general on an Eulerian grid, which acts as an indicator and, as such, allows interfaces to merge or break with relative ease. A fixed numerical mesh is used to evolve the interface by solving a transport equation involving this indicator function (for example, the level set function or the VOF function). The level set method, introduced in 1988 [129] is a popular Eulerian numerical approach to implicitly evolve the interface using a smooth (Lipschitz continuous) function (denoted by ϕ) defined over the entire physical domain. The initial interface is represented as the zero level set function (ϕ_0) and it propagates naturally with the evolution of this zero level set. Among the advantages of this method over others, one may cite its ability to handle topological changes such as breakup and merging, as well as its effective parallelization, the simple generalization to three dimensions without any complications in numerical processing, and its relatively easy implementation. It has been successfully used for various fields of applications [132], [133].

The first goal of this study is to analyze the behavior and efficiency of time perturbation and numerical resummation methods using techniques that can handle all the topological changes of the interface. Therefore, in this chapter, benchmarks level set problems, are studied by time perturbation and resummation methods. As during the resolution, the level set function rapidly loses its property, several methods to reinitialize this function are proposed in the literature [134]. Therefore, the second goal of this chapter is related to a discussion of the reinitialization procedure in the context of solutions that are continuous in time.

4.1 Level set method

In this section, we introduce the principle of the level set method, the reinitialization technique, and the classical numerical resolution of those latter.

4.1.1 Principle of method

The level set method represents an interface Γ as the zero contours of a smooth sign distance function, $\phi(\mathbf{x}, t)$, called the level set function. The latter is defined over a domain

Ω (for example a two-dimensional domain 2D) in such a way that:

$$\begin{cases} \phi(\mathbf{x},t) & < 0 \quad \forall\, \mathbf{x} \in \Omega_1 \\ \phi(\mathbf{x},t) & > 0 \quad \forall\, \mathbf{x} \in \Omega_2 \\ \phi(\mathbf{x},t) & = 0 \quad \forall\, \mathbf{x} \in \Gamma \end{cases} \tag{4.1}$$

as shown in Fig.(4.1). Note that $\mathbf{x} = (x, y)$ since Ω is a two-dimensional computational domain.

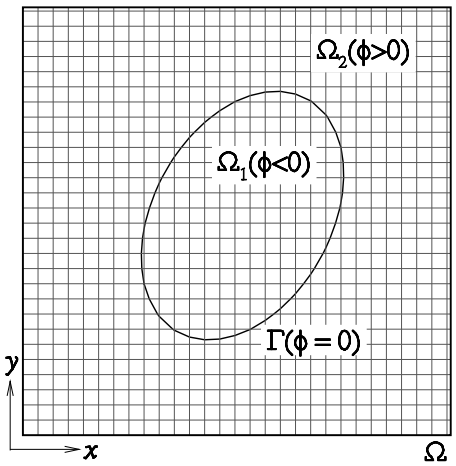


Figure 4.1 – Illustration of a two-dimensional computational domain Ω, showing the evolution of a level set function, $\phi(\mathbf{x}, t)$ with a zero contour representing the interface, Γ.

One of the most obvious observations is that the signed distance function is a regular function that allows to know immediately in which medium the point $\mathbf{x} = (x, y)$ is located.

The interface evolution in a given spatial time-dependant velocity field $\mathbf{u}(\mathbf{x}, t)$ where $\mathbf{u} = (u, v)$, is controlled by solving the following transport equation:

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0. \tag{4.2}$$

Properties

The properties of this method are as follows:

- The location of the interface is directly known since the sign of the function $\phi(\mathbf{x}, t)$ makes it possible to know immediately in which medium the point $\mathbf{x} = (x, y)$ is located, as shown in Fig. 4.1.
- The level set method permits the direct access to the local geometrical characteristics of the interface, namely the normal $\mathbf{n}$ and the curvature κ

$$\mathbf{n} = \frac{\nabla \phi}{|\nabla \phi|} \qquad \text{and} \qquad \kappa = \nabla \mathbf{n} \tag{4.3}$$

Thanks to the level set function, the normal $\mathbf{n}$ and the curvature of the interface κ are calculated in the whole domain. The normal vector $\mathbf{n}$ is oriented in the positive direction of the contours and its calculation is simplified using the distance property of the level set function.

- The level set function $\phi(\mathbf{x}, t)$ is a regular function:

$$\forall \mathbf{x} \in \Omega, \quad |\nabla \phi(\mathbf{x}, t)| = 1 \tag{4.4}$$

this is also the property of the signed distance function. When transporting a scalar field, this property is very interesting from a numerical point of view. It is always tricky to transport a discontinuous scalar for reasons of accuracy, stability, and diffusion of the numerical schemes. Therefore, when the level set function has a strong local curvature, the regularity property Eq.(4.4) may be lost.

4.1.2 Reinitialization technique

In this subsection, we present one of the major drawbacks of the level set method which is the loss of its distance property, and we describe a technique to overcome this problem.

The challenge of using the level set method is that the numerical solution of Eq.(4.2) rapidly loses its signed distance property during larger time computations. This means that the value of ϕ does not remain the shortest distance between the point and the interface: Eq.(4.4) will be no longer valid.
To overcome this problem and maintain ϕ as a signed distance function, the latter is reinitialized after a certain time, after each iteration for example.

Remark 4.1.1 *The interface may move slightly during the reinitialization of the level set function. This is not desirable, but it is almost unavoidable. Because the interface moves slightly every time we use the reinitialization technique, we conclude that reinitialization should only be used when absolutely necessary. Too much reinitialization of the interface movement will have a significant impact on the final results, and too few reinitialization will ruin the good properties of the level-set function. It should also be noted that it is not necessary for all level sets to be signed distance functions, but the zero level set and a small area next to it are enough, as mentioned in [135].*

Several ways of reinitialization are proposed in the literature. One may cite the fast marching method, the fast sweeping method, the algebraic Newton method, the hyperbolic PDE, and also a "brute force" approach. The interested reader is referred to [134] where the strength and weaknesses of each of these reinitialization methods are compared.

The most popular reinitialization technique is the PDE method proposed in [136] and is used in this study. It works as follow: a correction is introduced solving the so-called reinitialization equation [136] for few steps in a fictitious time τ (not a real physical time):

$$\frac{\partial \phi}{\partial \tau} + S(\phi_0)\left(|\nabla \phi| - 1\right) = 0 \tag{4.5}$$

where

$$S(\phi_0) = \frac{\phi_0}{\sqrt{\phi_0^2 + h}} \tag{4.6}$$

is a smoothed sign function, $\phi_0 = \phi(\tau = 0)$ is the initial zero level set function and h is proportional to the grid size or smoothing distance. Classically, for each step size Δt, several iterations of the reinitialization algorithm are needed to ensure the correct distribution of the level sets.

4.1.3 Classical numerical resolution

Various numerical experiences have shown that the discretization schemes of the transport equation Eq.(4.2) and the reinitialization equation Eq.(4.5) must have a sufficiently high order of convergence as well as a high precision to obtain suitable results [137].

Spatial discretization: Since there is a significant sensitivity of level set methods to the approximation of convective terms ($\nabla \phi$), a great deal of attention has been paid to spatial discretization. The literature review has shown that the majority of authors use the fifth-order of HJ-WENO scheme ("Hamilton-Jacobi Weighted Essentially Non-Oscillatory") [138], [139] to achieve better results.

Temporal discretization: Concerning temporal discretization, it presents difficulties but less than those of spatial discretization. S. Tanguy [137] has shown that even schemes of order 2 such as the Bell, Collela, and Glaz [140] or Adams-Bashford schemes are inadequate. As mentioned in [141], [142], the temporal discretization is performed either with the third or fourth order of the Total Variation Diminishing Runge Kutta (TVDRK) scheme. These schemes, proposed in [143], guarantee that no additional oscillations are created when the order of the temporal discretization is increased. In this chapter, we will use the fourth-order accuracy of TVDRK denoted by TVDRK4.
Solutions obtained by TVDRK4 and the fifth-order of HJ-WENO, are considered as reference solutions for comparisons in the present work.

4.2 Solution with time perturbation and resummation methods

After presenting the classical resolution of the initial problem in the previous section using TVDRK4 for the temporal discretization and the finite difference scheme HJ-WENO for the spatial discretization, this section is dedicated to the development of time series solution of the level set evolution equation. For more clarification and details, we will present the recurrence formulas to obtain series terms for the the two following cases separately:

- Time-independent velocity fields $\mathbf{u}(\mathbf{x})$.
- Time-dependent velocity fields $\mathbf{u}(\mathbf{x}, t)$.

In both cases, the velocity depends on space.
The obtained reccurence formulas will be then discretized using the finite difference scheme HJ-WENO which also permit to validate the association of time perturbation-resummation scheme to this finite difference scheme.
A discussion about the use of the reinitialization technique in the context of continuous solution time is presented at the end of this section.

4.2.1 Recurrence formulas for time-independent velocity field

First, we consider the case where the velocity field $\mathbf{u} = (u, v)$ depends on space but not on time. Therefore, only the level set function is represented in terms of power series using time as perturbation parameter:

$$\phi(\mathbf{x}, \hat{t}) = \phi_0(\mathbf{x}) + \sum_{i=1}^{N} \phi_i(\mathbf{x})\hat{t}^i \tag{4.7}$$

where N denotes the truncation order, and ϕ_0 is a known regular solution at an initial time t_0.

Injecting Eq.(4.7) into the transport equation Eq.(4.2) and equating the respective power of $\hat{t}$, yield to:

$$\begin{cases} \hat{t}^0 & : & \phi_1 + \mathbf{u} \cdot \nabla\phi_0 = 0 \\ \hat{t}^1 & : & 2\phi_2 + \mathbf{u} \cdot \nabla\phi_1 = 0 \\ \hat{t}^2 & : & 3\phi_3 + \mathbf{u} \cdot \nabla\phi_2 = 0 \\ & \vdots & \\ \hat{t}^{N-1} & : & N\phi_N + \mathbf{u} \cdot \nabla\phi_{N-1} = 0 \end{cases}$$

We recall that $\mathbf{u}$ and $\phi_i, \forall i \geq 1$ are space dependent. However, for the sake of simplicity of writing, we did not use $\phi_i(\mathbf{x}, t)$ and $\mathbf{u}(\mathbf{x})$ in this system of equations.

The recursive relation that permits to obtain the series terms ϕ_i is obtained:

$$\phi_i = -\frac{1}{i}\mathbf{u}\cdot\nabla\phi_{i-1} \qquad \forall i \geq 1. \tag{4.8}$$

4.2.2 Recurrence formulas for time-dependent velocity field

Now, we consider the case where the velocity field $\mathbf{u}(x,t)$ is spatial time-dependent, as result it is necessary to look for a time polynomial approximation also for it. Therefore, consider the following two power series:

$$\phi(\mathbf{x},\hat{t}) = \phi_0(\mathbf{x}) + \sum_{i=1}^{N}\phi_i(\mathbf{x})\hat{t}^i \tag{4.9}$$

$$\mathbf{u}(\mathbf{x},\hat{t}) = \mathbf{u}_0(\mathbf{x}) + \sum_{i=1}^{N}\mathbf{u}_i(\mathbf{x})\hat{t}^i \tag{4.10}$$

By substituting these power series into the transport equation Eq.(4.2) and equating the respective power of $\hat{t}$, one can obtain the following recurrence formula that permits to calculate ϕ_i series terms:

$$\begin{cases} \hat{t}^0 & : \quad \phi_1 + \mathbf{u}_0\cdot\nabla\phi_0 = 0 \\ \hat{t}^1 & : \quad 2\phi_2 + \mathbf{u}_0\cdot\nabla\phi_1 + \mathbf{u}_1\cdot\nabla\phi_0 = 0 \\ \hat{t}^2 & : \quad 3\phi_3 + \mathbf{u}_0\cdot\nabla\phi_2 + \mathbf{u}_1\cdot\nabla\phi_1 + \mathbf{u}_2\cdot\nabla\phi_0 = 0 \\ & \vdots \\ \hat{t}^{N-1} & : \quad N\phi_N + \displaystyle\sum_{k=0}^{N-1}\mathbf{u}_k\cdot\nabla\phi_{N-1-k} = 0 \end{cases}$$

Therefore, series terms are given by the following recurrence formula:

$$\phi_i = -\frac{1}{i}\sum_{k=0}^{i-1}\mathbf{u}_k\cdot\nabla\phi_{i-1-k} \qquad \forall i \geq 1. \tag{4.11}$$

We recall that $\mathbf{u}_i$ and $\phi_i, \forall i \geq 1$ are space dependent. However, for the sake of simplicity of writing, we did not use $\phi_i(\mathbf{x},t)$ and $\mathbf{u}_i(\mathbf{x})$ in this system of equations.
The terms $\mathbf{u}_i$ and ϕ_i are the unknowns of this recurrence formula. The velocity coefficients $\mathbf{u}_i = (u_i, v_i), \forall i \geq 1$ should be calculated accordingly to the velocity data of each problem, in order to determine ϕ_i terms (more details in the numerical example).

4.2.3 Reinitialization technique in the context of transients continuous solutions

Since our work is based on time perturbation methods, we will test three ways of using reinitialization within solution that are continuous in time:

1. The classical PDE reinitialization Eq.(4.5) only after the *last* step of the continuation procedure of ANM i.e. at the final physical time t,
2. The classical PDE reinitialization Eq.(4.5) at the *end* of each continuation step of ANM, i.e end of each t_{max},
3. *Time perturbation* of reinitialization procedure without using a modified version of Eq.(4.5).

The time perturbation reinitialization procedure is introduced here. We recall that the concept of a reinitialization technique is to maintain the level set function ϕ as a signed distance function, (i.e. $|\nabla\phi| = 1$ must be respected during the resolution).
We note that we have not solved the classical PDE reinitialization equation (Eq.(4.5)) by time perturbation approach because it contains a pseudo time τ. For this reason, our idea to test the reinitialization procedure by temporal perturbation is based on solving the following system (transport equation + property of the level set signed distance function $|\nabla\phi| = 1$):

$$\frac{\partial\phi}{\partial t} + \mathbf{u}\cdot\nabla\phi = 0 \tag{4.12}$$

$$|\nabla\phi| = 1 \tag{4.13}$$

by the temporal perturbation method. The following notation $|\nabla\phi| = 1$ stands for a modulus at each discretization point $(\forall \mathbf{x} \in \Omega, |\nabla\phi(\mathbf{x}, t)| = 1)$. As we want to use perturbation methods, it is proposed to use the following global information:

$$\nabla\phi\cdot\nabla\phi = 1 \tag{4.14}$$

Writing ϕ as time power series and replace it in the system (Eq.(4.12) and Eq.(4.13)), one can obtain the following recurrence formulas to obtain ϕ_i series terms $\forall i = 1, \cdots, N$:
For the case of time-independent velocity field:

$$\phi_i = -\frac{1}{i}\left(\mathbf{u}\cdot\nabla\phi_{i-1} + \sum_{k=0}^{i-1}\nabla\phi_k\cdot\nabla\phi_{i-1-k}\right) \tag{4.15}$$

For the case of time-dependent velocity field:

$$\phi_i = -\frac{1}{i}\sum_{k=0}^{i-1}(\mathbf{u}_k + \nabla\phi_k)\cdot\nabla\phi_{i-1-k} \tag{4.16}$$

Finally, note that if ϕ_0 is a signed distance function, then each series term ϕ_i for $i \geq 1$ may not remain so. This property is tracked by evaluating the value of $|\nabla\phi_i|, \forall i \geq 1$, which is usually moves away from unity.

4.3 Numerical validation for time-independent velocity field

To study the behavior of time perturbation techniques and numerical resummation methods for the solutions of level set evolution equations, various academic test problems proposed in [137], [144]–[146] are studied in this section. We deal with two problems in detail where there is a large deformation of the interface. We impose a velocity field that depends on space but not on time. The so-called "Vortex flow" and "deformation field" simulations are the most frequently encountered in the literature, so we have logically carried them out. A test case called "Compression/decompression of a circle" is also studied.

4.3.1 Vortex flow

The "Vortex flow" test case, introduced by Bell, Collela, and Glaz [140], quantifies the performance of an interface tracking method when the interface stretches into a long ligament due to shear flow. This test case is rigorous because, as we will see, the ligament stretches indefinitely over time. The numerical schemes used must be robust enough to ensure that the code does not diverge. Therefore the latter should be very precise to capture the ligament as much as possible when its width approaches that of the mesh, preserving the initial surface. It is very interesting to test the time perturbation and numerical summation methods for the interface tracking method, as it is almost inevitable when simulating two-phase shear flows.

Consider a square domain $\Omega = [0,1]^2$ containing a circle of radius $r_0 = 0.15$ and origin $(x_0, y_0) = (0.5, 0.75)$. The initial interface is given by:

$$\phi_0(x,y) = \sqrt{(x-x_0)^2 + (y-y_0)^2} - r_0 \tag{4.17}$$

A velocity field $\mathbf{u} = (u, v)$ is defined by the following potential ψ:

$$\psi = \frac{1}{\pi}\sin^2(\pi x)\sin^2(\pi y) \tag{4.18}$$

The analytical expressions for the components of $\mathbf{u} = (u, v)$, are given by:

$$u(x, y) = -\frac{\partial \psi}{\partial y} = -2\sin^2(\pi x)\sin(\pi y)\cos(\pi y) \tag{4.19}$$

$$v(x, y) = \frac{\partial \psi}{\partial x} = 2\sin(\pi x)\cos(\pi x)\sin^2(\pi y) \tag{4.20}$$

Numerical results of classical schemes

The two classical schemes (TVDRK4 + HJ-WENO) are applied here (see subsection 4.1.3). In this study, studies are conducted with the grid size 128×128 and for a discrete-time step, $\Delta t = 10^{-3}$s for the TVDRK4 scheme to respect the CFL condition.

Fig.(4.2) illustrates the evolution of the interface shapes at different times t computed by solving equation Eq.(4.2) with homogeneous Neumann boundary conditions without reinitialization and with TVDRK4. These numerical results clearly show the large deformation of the initial interface.

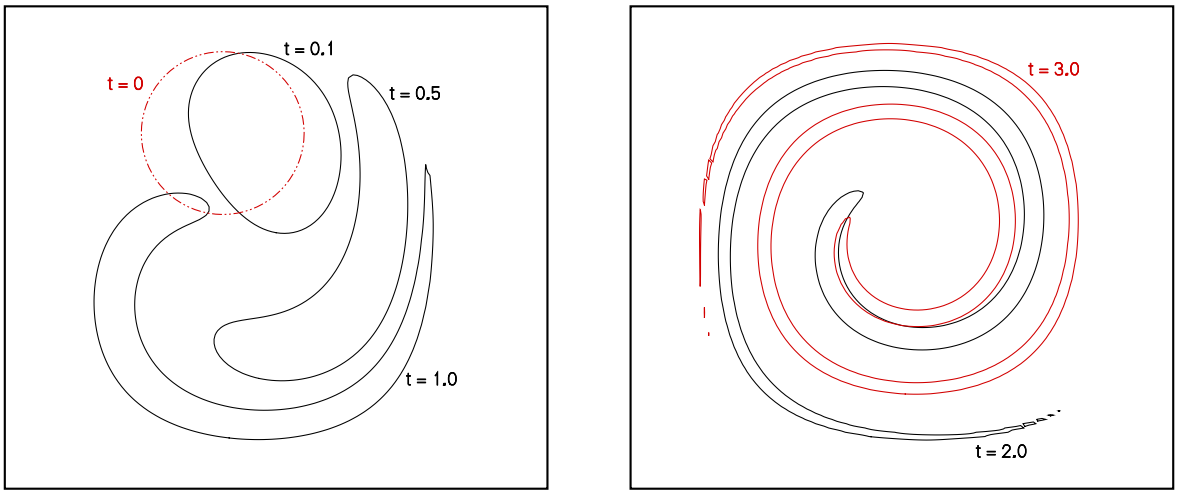


Figure 4.2 – Evolution of level set interfaces shapes using TVDRK4 without reinitialization, from an initial time $t = 0$s to the final time $t = 3$s. The grid size is 128×128. The discrete time step is $\Delta t = 10^{-3}s$.

Numerical results of time perturbation and resummation methods

In this subsection, the effectiveness of the time perturbation and resummation methods is studied on the level set equation. After determining the final recursive relation

that permits to calculate ϕ_i series terms (Eq.(4.8)), the ability of ANM, BPL, and IFS to provide the same approximate solutions as the classical method is examined. These schemes are used to evaluate the solution of level set transport equation (Eq.(4.2)) until a final time $t = 3s$, for the following grid size: 128×128. First, the Euclidean norm of series terms at zeroth step (without continuation procedure) is plotted in Fig.(4.3). It can be observed that the norm increases with the truncation order.

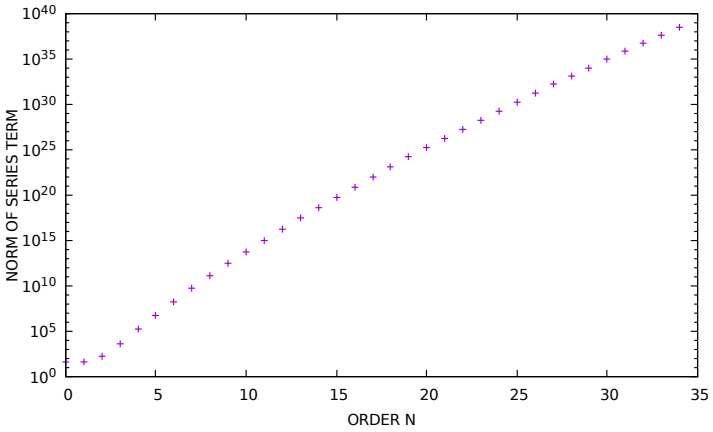


Figure 4.3 – Evolution of the norm of the series terms with the truncation order N at zeroth step (without continuation procedure), for the level set evolution equation. The grid used is 128×128.

Then, the continuation is carried up to $t = 3s$, using a residual tolerance $\epsilon = 10^{-4}$. The series is computed up to a truncation order $N = 20$. First, the Euclidean norm of the level set function $\|\phi(x, y, t)\|_2$ obtained by ANM, BPL and IFS are compared to the one obtained by TVDRK4 at time $t = 3$s in Fig.(4.4). No visible difference can be seen from this plot which is the first indicator of the global correctness of numerical time perturbation and resummation integration schemes.

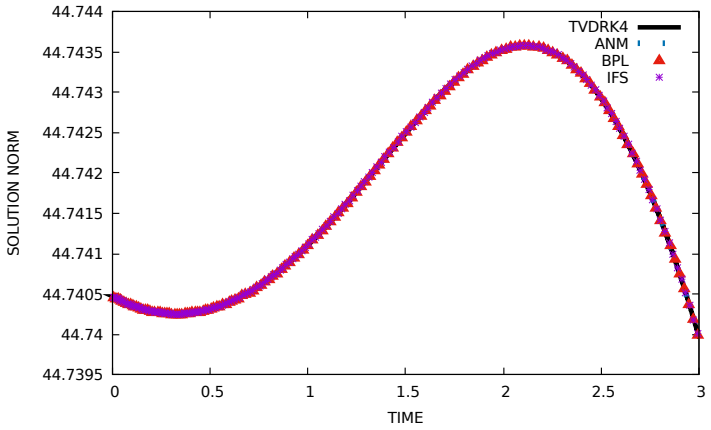
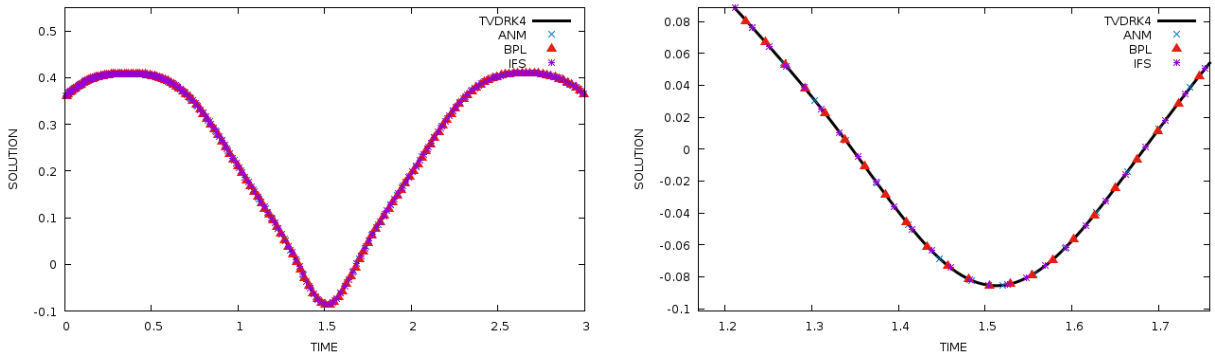


Figure 4.4 – Solution norm $||\phi(x,y,t)||_2$ computed by time perturbation-resummation methods ANM, BPL and IFS, and the classic scheme TVDRK4 with a discrete-time step $\Delta t = 10^{-3}$s. The grid size is 128×128.

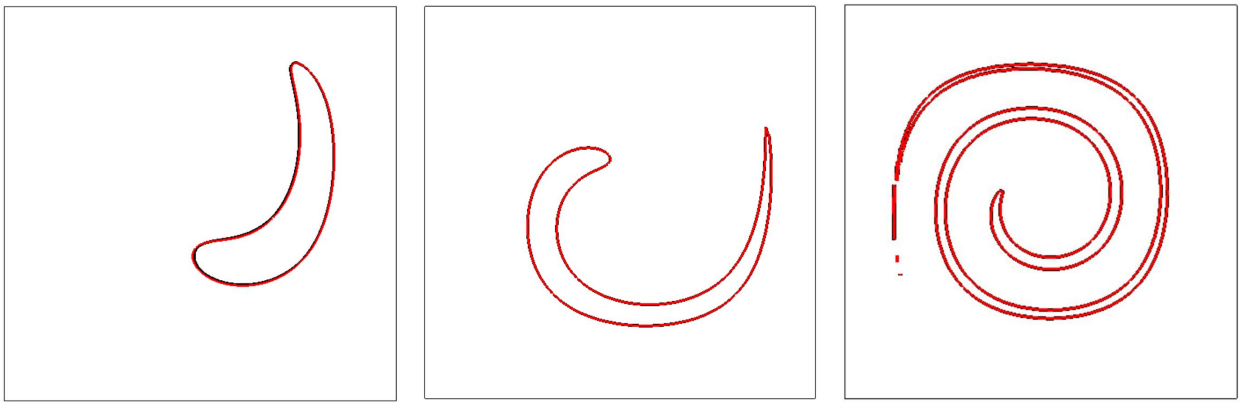
It is observed, in Fig.(4.5), that ANM, BPL, and IFS present the same numerical solution at a probe point of the domain, as the classic scheme TVDRK4. This is further supported by no visible distinction for the interface Γ (zero contour i.e $\{\mathbf{x} = (x,y); \phi(x,y,t) = 0\}$) between BPL (in red) and TVDKR4 (in black) in Fig.(4.6), for $t = 0.5$s, $t = 1$s and $t = 3$s. Same interfaces are obtained with IFS and ANM and are not presented in the figure because of the visualization difficulties among the 4 methods. This was verified by a computation of the relative error between the temporal perturbation methods which is of order 10^{-15}.

One can deduce that time perturbation methods alone and then followed by resummation methods can respect the evolution of the level set and represent well the important spatial variations in the evolution of the interface. The numerical result has shown the ability of the analyticity of the approximation to reproduce the large deformations of the initial circle.



(a) (b)

Figure 4.5 – Solution for the probe point P1($x(i = 30), y(j = 40)$) = $(0.2362205, 0.3149606)$ (i.e $\phi(0.2362205, 0.3149606, t)$) by time perturbation-resummation methods ANM, BPL and IFS, and the classic scheme TVDRK4 with a discrete-time step $\Delta t = 10^{-3}$s. (b) is a zoom plot of (a). The solution obtained by ANM, BPL, and IFS is continuous between the points of continuation indicated by symbols in Figure (b).



(a) t=0.5 s (b) t=1 s (c) t=3 s

Figure 4.6 – Level set interface by time perturbation-resummation method BPL (in red) and the classic scheme TVDRK4 (in black) with a discrete-time step $\Delta t = 10^{-3}s$. No visible distinction between the level set contours.

Comparisons

Now we are interested to compare the range of validity (t_{max}) of these temporal approaches. The comparison is performed in Fig.(4.7) with the residual criteria for a tolerance $\epsilon = 10^{-4}$. It can be seen that BPL has the largest t_{max} among the other methods and is significantly greater than the time step, $\Delta t = 10^{-3}$s, of TVDRK4. One can deduce that the range of validity of time perturbation and resummation schemes is much larger than the time step of TVDRK4. Therefore, they require relatively lesser steps than the classic scheme to reach the same final time.

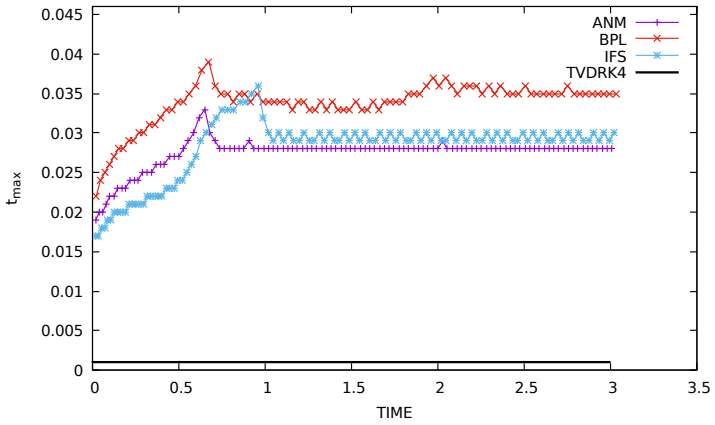


Figure 4.7 – Comparison of the range of validity (t_{max}) of ANM, BPL (for a fixed Gauss number $N_G = 10$) and IFS, for $N = 15$ and a residual tolerance $\epsilon = 10^{-4}$. The grid size is 128×128.

The classical time integration scheme TVDRK4 with $\Delta t = 10^{-3}s$ needs 3000 steps to reach a fixed final time of $t = 3s$. The number of steps performed with time perturbation-resummation schemes to reach $t = 3s$, for a truncation order of $N = 15$, is presented in Tab.(4.1). It is done for different choices of residual tolerance parameters ϵ using the same continuation criteria based on the residual. This table shows that the time perturbation method alone (ANM) and time perturbation-resummation methods (BPL, IFS) need much fewer steps than the classical scheme TVDRK4. Further, it is observed that IFS needs more steps than BPL.

Residual tolerance	ANM	BPL	IFS
10^{-2}	108	81	103
10^{-3}	109	86	105
10^{-4}	110	90	109
10^{-5}	114	94	116

Table 4.1 – Number of steps needed to reach $t = 3$s using ANM, BPL, and IFS for a truncation order $N = 15$, for different values of the residual tolerance ϵ. The grid used is 128×128. TVDRK4 with $\Delta t = 10^{-3}$s needs 3000 steps to reach the same final time.

CPU time is now presented in Tab.(4.2). Those computation have been performed on an Intel(R) Xeon(R) CPU W3565 3.20GHz, using gcc compiler with optimization flag "-03". The reference CPU time is 6.3s for the classical time integration scheme.

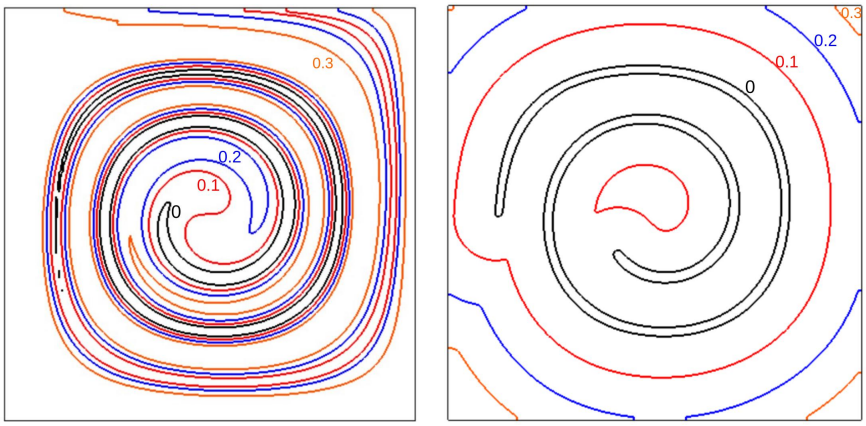
Total(s) TVDRK4	Total(s)ANM	Total(s) BPL	Total(s) IFS
6.3	2.5	5.6	10

Table 4.2 – The total CPU time required to reach $t = 3$s for the level set evolution equation using TVDRK4 with a discrete-time step $\Delta t = 10^{-3}$s and ANM, BPL, and IFS with a truncation order $N = 15$. The grid used is 128×128.

Computations using reinitialization technique

As mentioned in sections 4.1.2 and 4.2.3, the level set function rapidly loses its signed distance property over time. To avoid this, a reinitialization technique is applied. The example of Vortex flow is an ideal test case to study the efficiency of the reinitialization procedure.
Note that the transport equation of the level set has been constructed such that each iso-contour (or called also level lines) in addition to the interface itself, is advected by the local velocity field. Such a methodology implies that the iso-contours do not remain locally at the same distance from each other when the tangential component of the gradient of the normal velocity to the iso-contours is non-zero. The principle is to correct the position of the iso-contours from the only valid iso-contour which is the interface. The purpose of this correction based on the solution of the transport equation is to reset the level set values from the calculated position of the interface, to restore the property of the signed distance function. For this reason, we will illustrate the iso-contours to validate the efficiency of the reinitialization algorithm. First, we start solving the classical PDE reinitialization Eq.(4.5) by TVDRK4.

TVDRK4 with the classical PDE reinitialization: Firstly, the level lines of the solution obtained by TVDRK4, before and after using PDE reinitialization technique (see Eq.(4.5)) with $\Delta t = 10^{-3}s$, are presented in Fig.(4.8). It can be observed that the first plot in Fig.(4.8a) shows disturbances of the level sets while those latter are correctly positioned on both sides of the interface in Fig.(4.8b) (according to the literature [146]). Therefore, the result is significantly improved with the application of the reinitialization algorithm.



(a) Without reinitialization (b) With reinitialization

Figure 4.8 – Different level lines {0, 0.1, 0.2, 0.3} for solutions obtained by TVDRK4 at a final time $t = 3s$, with and without the reinitialization technique. The discrete-time step is $\Delta t = 10^{-3}$s. The grid size is 128×128.
Indication on the colors of the level lines {0, 0.1, 0.2, 0.3}: {0} in black, {0.1} in red, {0.2} in blue, {0.3} in orange.

Now, we seek to test numerically the three ways of reinitialization, in the context of solutions that are continuous in time. The latter are proposed in subsection 4.2.3 and recalled here:

1. The classical PDE reinitialization Eq.(4.5) only after the *last* step of the continuation procedure of ANM i.e. at the final physical time t,
2. The classical PDE reinitialization Eq.(4.5) at the *end* of each continuation step of ANM, i.e end of each t_{max},
3. *Time perturbation* of the reinitialization procedure without using a modified version of Eq.(4.5).

We are looking for a way of these three proposals that yields to well-positioned iso-contours like those illustrated in Fig.(4.8b).

1. ANM with the classical PDE reinitialization at the end of the last continuation step: Here, the classical PDE reinitialization (see Eq.(4.5)) only after the final step of the continuation procedure of ANM (i.e end of the final time $t = 3$s) is tested. As can be seen in Fig.(4.9), this does not lead to proper level lines as those presented in Fig.(4.8b). Thus, this test of reinitialization is not suitable.

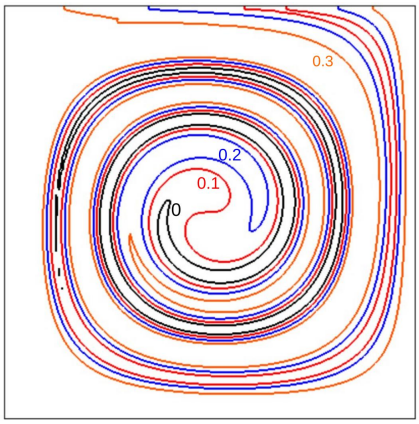
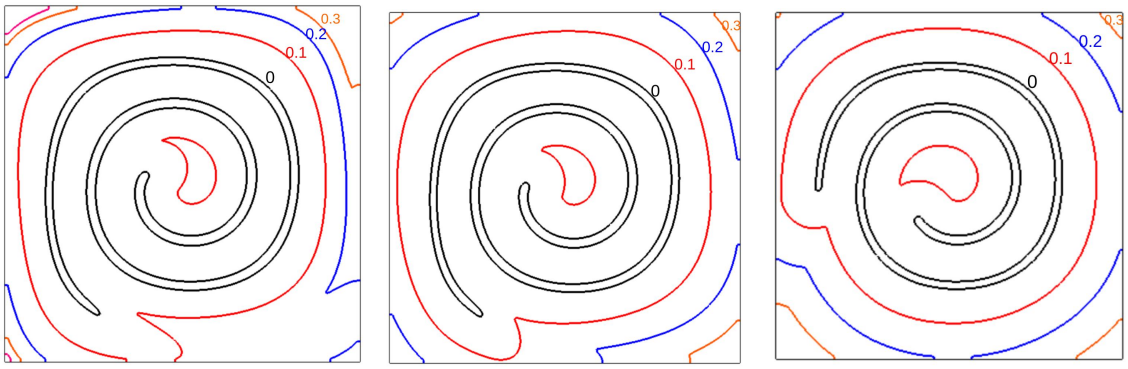


Figure 4.9 – Different level lines {0, 0.1, 0.2, 0.3} for solutions obtained by ANM with the classical PDE reinitialization at the end of the last continuation step at a final time $t = 3s$. ANM truncation order is $N = 20$. The grid size is 128×128.
Indication on the colors of the level lines {0, 0.1, 0.2, 0.3}: {0} in black, {0.1} in red, {0.2} in blue, {0.3} in orange.

2. ANM with the classical PDE reinitialization performed at the end of each continuation step: The classical PDE reinitialization (see Eq.(4.5)) performed at the end of each ANM continuation step (i.e end of every t_{max}) is tested and illustrated in Fig.(4.10). One can see that the level lines located in a region close to the zeroth contour are now better positioned. Using a small ANM tolerance of $\delta = 10^{-15}$ leads to a smoother result in Fig.(4.10b) than in Fig.(4.10a). The result with a smaller tolerance is presented in Fig.(4.10c). We will see that the zeroth contour of Fig.(4.10b) is better than that of Fig.(4.10c) when it is compared to the reference solution in the following. The well-positionned level lines confirm that this way of a reinitialization of continuous solutions in time, is well appropriate.



(a) with a tolerance $\delta = 10^{-6}$. (b) with a tolerance $\delta = 10^{-15}$ (c) with a tolerance $\delta < 10^{-15}$

Figure 4.10 – Different level lines {0, 0.1, 0.2, 0.3} for solutions obtained by ANM with PDE reinitialization at each continuation step, with different ANM tolerances δ, at a final time $t = 3s$. ANM truncation order is $N = 20$. The grid size is 128×128.

3. Time perturbation of reinitialization technique: For this test, we apply time perturbation of the reinitialization procedure (see Eq.(4.15)). We noticed in Fig.(4.11) that just one iso-contour (in red) next to the interface (in black) is well-positioned but only if a very small ANM tolerance δ is used. One can always observe the bad behavior of others iso-contours in blue which may be due to the application of the reinitialization technique during the entire continuation process.

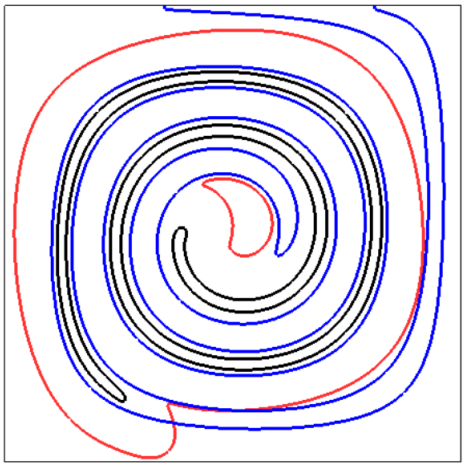


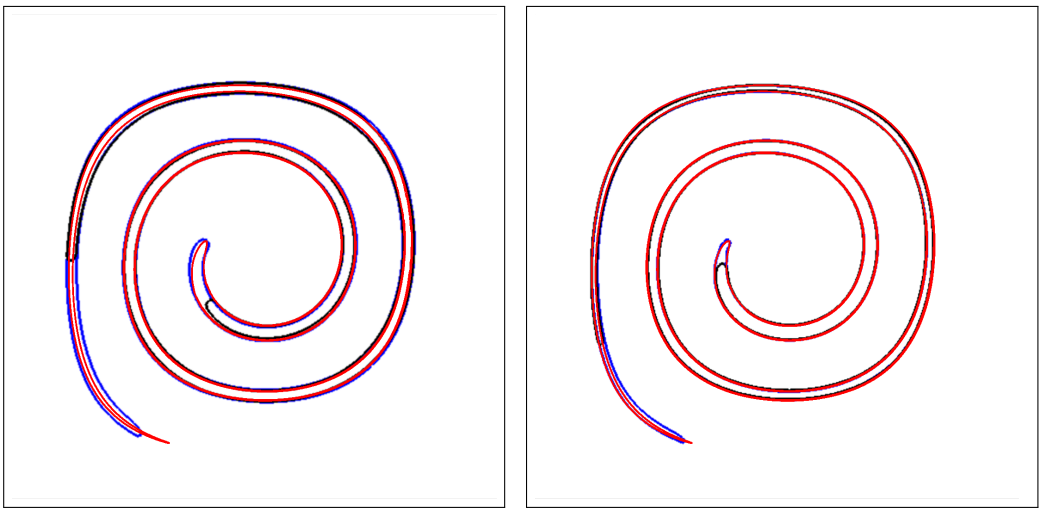
Figure 4.11 – Different level lines $\{0, 0.1, 0.2\}$ for solutions obtained by time perturbation of the reinitialization technique with a tolerance $\delta = 10^{-30}$, at a final time $t = 3s$. ANM truncation order is $N = 20$. The grid size is 128×128.

As a result of this study, one can conclude that adapting ANM with the classic PDE reinitialization at the end of each continuation step of the level set evolution equation is the relatively better choice to obtain a good result. Therefore, for the following, only this choice will be considered.

After the comparison of the different behavior of the level lines between different techniques, now a comparison only for the zeroth contour (interface) between the following methods:

- TVDRK4 with PDE reinitialization,
- The classical PDE reinitialization Eq.(4.5) at the *end* of each continuation step of ANM, i.e end of each t_{max},
- The "reference" solution given by Rider and Kothe [144] by using the analytical particle method. (It is not an exact mathematical solution, since it is obtained numerically. Nevertheless, it is considered as an approximation of the solution as fine as possible.)

is proposed in Fig.(4.12).



(a) grid size: 128×128 (b) grid size: 256×256

Figure 4.12 – Level-set interfaces at a final time $t = 3$s. In blue: ANM + PDE reinitialization end of every step of continuation, $N = 20$, $\delta = 10^{-6}$. In red: the "reference" solution presented by Rider and Kothe [144]. In black: TVDRK4 with reinitialization with a discrete-time step $\Delta t = 10^{-3}s$.

It can be seen that the solution given by ANM with PDE reinitialization end of each continuation step, is more accurate than the classic scheme TVDRK4 with reinitialization using a discrete-time step $\Delta t = 10^{-3}s$.
The length of the coil obtained by ANM with PDE reinitialization end of each continuation step (in blue) is nearly the same as that of the "reference" solution (in red) than that of TVDRK4 (in black) in Fig.(4.12). However, the thickness of the plume is slightly larger than that of the "reference" solution. This is due to the problem of the conservation of the mass of the classical reinitialization (and not a numerical error of ANM).

Finally, as a conclusion of this study, one can deduce that adapting the PDE reinitialization end of every step of continuation of ANM is the better choice in terms of the interface (or zeroth contour) and the other iso-contours (level lines).

4.3.2 Compression-Decompression of a circle

The goal of this example [145] is to validate the choice of adapting the PDE reinitialization end of every step of continuation on another example than the "Vortex flow". This test case does not present any deformation of the initial interface. We will not present a full study for this example. Only the validation of the reinitialization end of each step of ANM continuation will be performed.
The initial data given in Eq.(4.17) is used also for this problem. We consider the same

domain as the first example $\Omega = [0,1]^2$, containing a circle of radius $r_0 = 0.4$ and origin $(x_0, y_0) = (0.5, 0.5)$ successively compressed and decompressed by the velocity field $\mathbf{u} = (u, v)$, then by its opposite defined by:

$$u(x,y) = x_0 - x \tag{4.21}$$

$$v(x,y) = y_0 - y \tag{4.22}$$

The analytical solution after compression for an instant t_1 denoted by ϕ_c [145], is given by:

$$\phi_c(x,y,t) = \sqrt{(x-x_0)^2 + (y-y_0)^2} - (r_0 - t_1) \tag{4.23}$$

The analytical solution after decompression for an instant t_2 denoted by ϕ_d [145], is given by:

$$\phi_d(x,y,t) = \sqrt{(x-x_0)^2 + (y-y_0)^2} - (r_0 - t_1 + t_2) \tag{4.24}$$

Fig.(4.13) illustrates the decompression of the interface shapes during the decompression, at different times t computed by solving equation Eq.(4.2) without reinitialization and with TVDRK4.

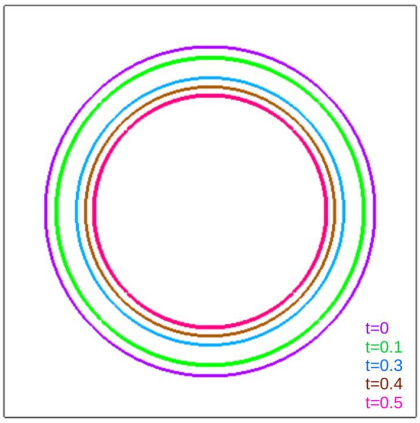
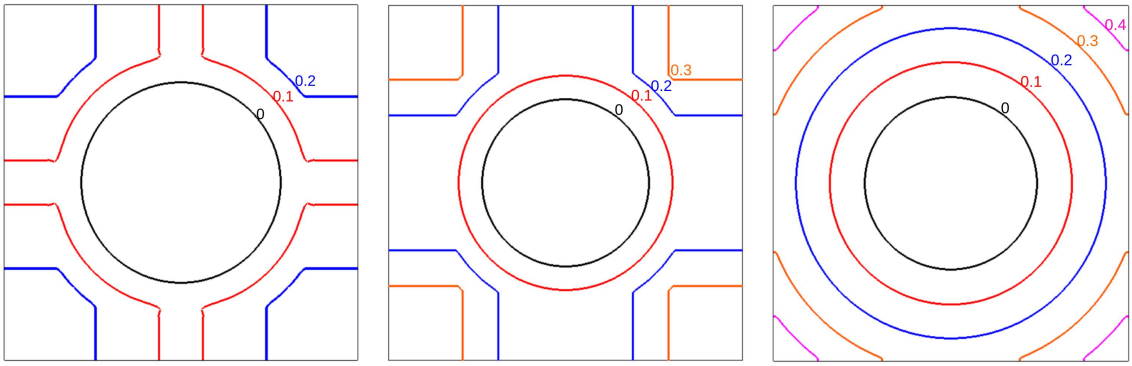


Figure 4.13 – Evolution of the level set interfaces shapes using TVDRK4 without reinitialization, from an initial time $t = 0$s to the final time $t = 0.5$s. The grid size is 128×128. The discrete time step is $\Delta t = 10^{-3}s$.

The classical PDE reinitialization (see Eq.(4.5)) performed at the end of each ANM continuation step is tested and illustrated in Fig.(4.14b) and Fig.(4.14c). One can see that only one level line $\{0.1\}$ located in a region close to the zeroth contour is now better positioned. Using a small ANM tolerance (for example $\delta = 10^{-20}$ which give a range of validity of order $t_{max} = 10^{-3}$) leads to a smoother result in Fig.(4.14c) than in Fig.(4.14b). Fig.(4.14c) is similar to the one obtained by the analytical solution.
This second test case confirms that this way of reinitialization of continuous solutions in time is well appropriate.



(a) (b) (c)

Figure 4.14 – Different level lines {0, 0.1, 0.2, 0.3, 0.4} for solutions obtained, at time $t = 0.5s$ with a truncation order $N = 20$ and a grid size 128×128, by
(a): ANM without reinitialization,
(b): ANM with PDE reinitialization at each continuation step, with $\delta = 10^{-6}$,
(c): ANM with PDE reinitialization at each continuation step, with $\delta = 10^{-20}$, (or a t_{max} of order 10^{-3}). Same figure is obtained by the analytical solution.
Indication on the colors of the level lines {0, 0.1, 0.2, 0.3}: {0} in black, {0.1} in red, {0.2} in blue, {0.3} in orange, {0.4} in purple.

4.3.3 Deformation field

The third test case that we will deal with is the deformation field, that is also processed in [144] by using the analytical particle method. In this test case, we will also detect a large deformation of the initial interface and the topology. The same initial assumptions of the domain, and radius of the circle are the same as section 4.3.1. The initial data given in Eq.(4.17) is used also for this problem, but the complex velocity field $\mathbf{u} = (u, v)$ is defined by the following potential ψ:

$$\psi(x, y) = \frac{1}{4\pi} \sin\left(4\pi(x + \frac{1}{2})\right) \cos\left(4\pi(y + \frac{1}{2})\right) \tag{4.25}$$

The analytical expressions for the components of the velocity field $\mathbf{u} = (u, v)$ are given by

$$u(x, y) = -\frac{\partial \psi}{\partial y} = \sin\left(4\pi(x + \frac{1}{2})\right) \sin\left(4\pi(y + \frac{1}{2})\right) \tag{4.26}$$

$$v(x, y) = \frac{\partial \psi}{\partial x} = \cos\left(4\pi(x + \frac{1}{2})\right) \cos\left(4\pi(y + \frac{1}{2})\right) \tag{4.27}$$

The fluid elements are deformed by the flow field. Fig.(4.15) plots the level set interface using TVDRK4 for $\Delta t = 10^{-3}s$ for the grid size 128×128. The deformed contours of the level set function appear stretched and ruptured. Furthermore, Fig.(4.16) shows the evolution of interface topology inside the whole domain during the time.

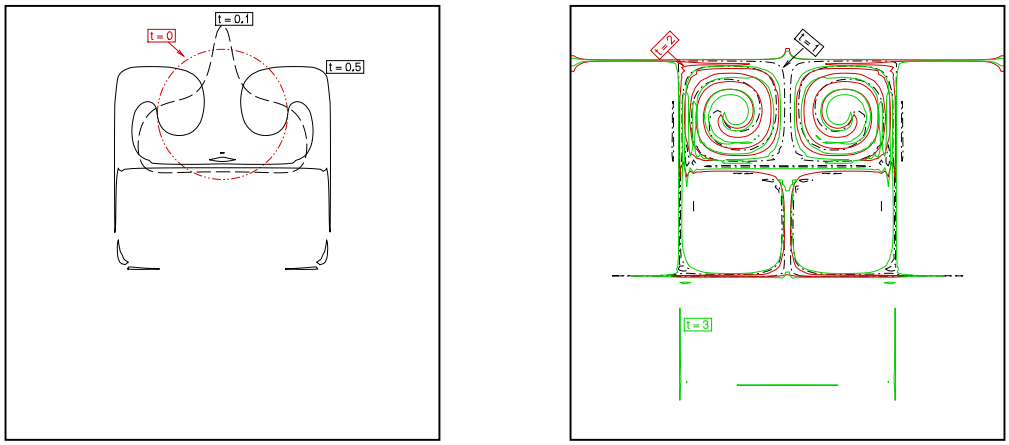
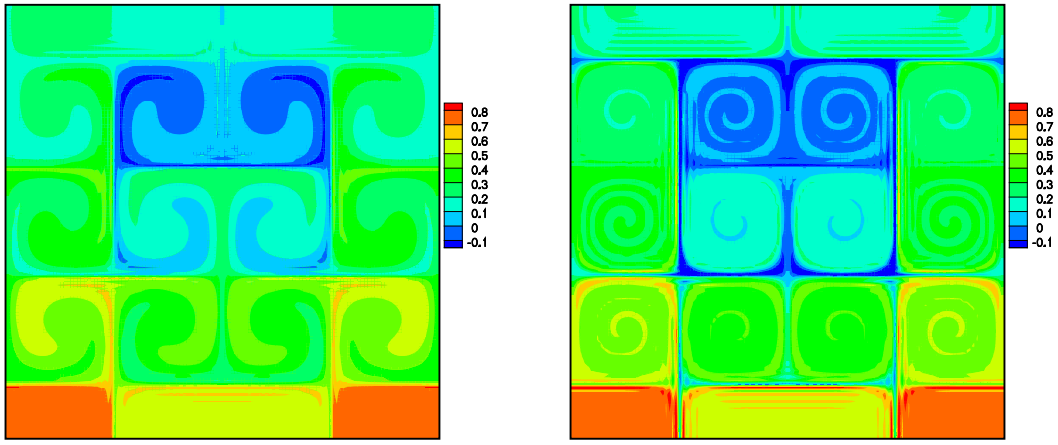


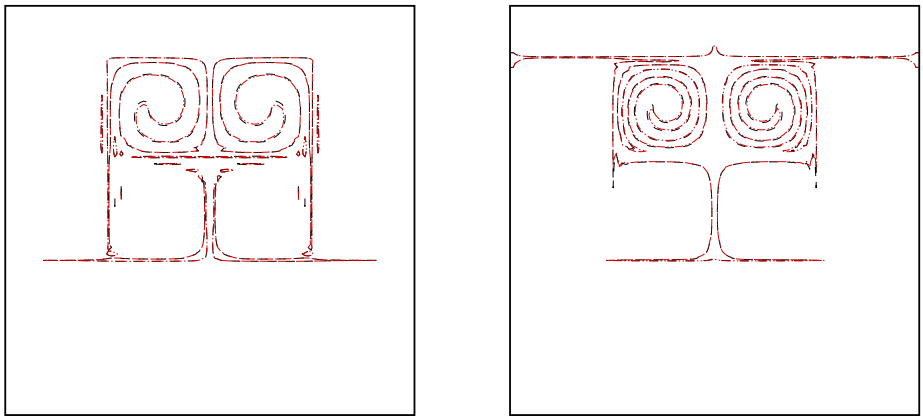
Figure 4.15 – Evolution of the level set interfaces shapes using TVDRK4 without reinitialization using a discrete time step $\Delta t = 10^{-3}s$, from initial time $t = 0$s to final time $t = 3$s. The grid size is 128×128.



(a) $t = 0.5s$ (b) $t = 3s$

Figure 4.16 – Interface topology in the whole domain using TVDRK4 without reinitialization using a discrete time step $\Delta t = 10^{-3}s$, at two different times t. The grid size is 128×128.

Level-set interface solutions for $t = 1s$ and $t = 2s$ obtained by BPL resummation procedure and TVDRK4 are in good agreement in Fig.(4.17). Note that BPL also reproduces the evolution of the interface topology as in Fig.(4.16). The same figures are obtained with ANM and IFS. This proves that these methods can follow this deformation and reproduce the solutions accurately.



(a) $t = 1$s (b) $t = 2$s

Figure 4.17 – Level set interface computed by time perturbation resummation method BPL (in red) using $N_G = 10$, $N = 15$, $\epsilon = 10^{-4}$ and the classic scheme TVDRK4 (in black) with a discrete-time step $\Delta t = 10^{-3}$s. The grid size is 128×128.

Finally as a conclusion of this study, one can deduce from these examples that the level set evolution equation with a time-independent velocity field has been successfully handled. Thus, the recurrence formula Eq.(4.8) is validated.

4.4 Numerical validation for time-dependent velocity field: "Time-Reversed Flow fields"

The goal of this test case is to validate the recurrence formula Eq.(4.11) in the case of time-dependent velocity field. This test has been also handled in [144].
The same initial assumptions as the previous sections are used for this problem, but the velocity field $\mathbf{u}(\mathbf{x}, t) = \mathbf{u} = (u, v)$ is multiplied by $\cos(\omega t)$, therefore:

$$u(x, y, t) = -2\sin^2(\pi x)\sin(\pi y)\cos(\pi y)\cos(\omega t) \tag{4.28}$$

$$v(x, y, t) = 2\sin(\pi x)\cos(\pi x)\sin^2(\pi y)\cos(\omega t) \tag{4.29}$$

where ω denotes the angular frequency.

It should be noticed that any velocity field can be multiplied by $\cos\left(\frac{\pi t}{T}\right)$ that made it periodic in time T, where T is the time period when the flow returns to its initial state [147]. The flow is time-reversed when the period T is equal to 2.

4.4.1 Calculation of velocity field series terms

A time perturbation of the prescribed time-dependent velocity field is required. The velocity coefficients $\mathbf{u}_i = (u_i, v_i)$, $\forall i \geq 1$ in the recurrence formula Eq.(4.11) should be calculated in order to determine ϕ_i

For the sake of simplicity, let:

$$A = -2\sin^2(\pi x)\sin(\pi y)\cos(\pi y) \tag{4.30}$$

Therefore, the first component of the velocity can be written in the following form:

$$u(x, y, t) = A\cos(\omega t) \tag{4.31}$$

Let the change of variable $D = \cos(\omega t)$ and $E = \sin(\omega t)$ as proposed in [39]. This leads to:

$$\begin{cases} D = \cos(\omega t) \\ E = \sin(\omega t) \end{cases} \Longrightarrow \begin{cases} dD &= -\omega E\ dt \\ dE &= \omega D\ dt \end{cases} \tag{4.32}$$

After writing D and E as a series: $D = D_0 + \sum_{n=1}^{N} D_n \hat{t}^n$ and $E = E_0 + \sum_{n=1}^{N} E_n \hat{t}^n$, then injecting them into the second part of Eq.(4.32), and finally equating terms with identical powers of $\hat{t}$: we get for $n \geq 1$:

$$\begin{cases} D_n &= \frac{-\omega}{n} E_{n-1} \\ E_n &= \frac{\omega}{n} D_{n-1} \end{cases} \tag{4.33}$$

The initial terms are $D_0 = \cos(\omega t_0)$ and $E_0 = \sin(\omega t_0)$ where t_0 denotes the initial time. Therefore,

$$u_n = AD_n \qquad \forall n = 1, \cdots, N \tag{4.34}$$

The second component of the velocity v_n is calculated in the same way. Finally, after identifying $\mathbf{u}_n = (u_n, v_n)$, series terms ϕ_n can now be computed by Eq.(4.11).

We want to verify the ability of these methods to reproduce the solution continuously also for the case of the time-dependent velocity field.

4.4.2 Numerical results

The truncation order is set to $N = 10$, the frequency to $\omega = 0.125\pi$ and the residual tolerance to $\epsilon = 10^{-5}$. The interface given by the classic scheme TVDRK4 with $\Delta t = 10^{-3}s$, and ANM, BPL and IFS is indistinguishably visually in Fig.(4.18) for $t = 3s$.

Local errors between the different couples of these methods is evaluated and some of them are displayed in Fig.(4.19), Fig.(4.20), and Fig.(4.21) respectively for $t = 1s$, $t = 2s$ and $t = 3s$. The error between ANM, BPL, and IFS is approximately of 10^{-14} order. The error between ANM, BPL, IFS and the classical scheme TVDRK4 is around 10^{-4}. Evaluating these errors clearly shows the efficiency of time perturbation-resummation methods also in the case of a time-dependent velocity field.

Finally as a conclusion of this study, one can deduce that the level set evolution equation with a time-dependent velocity field has been successfully handled. Thus, the recurrence formula Eq.(4.11) is well validated.

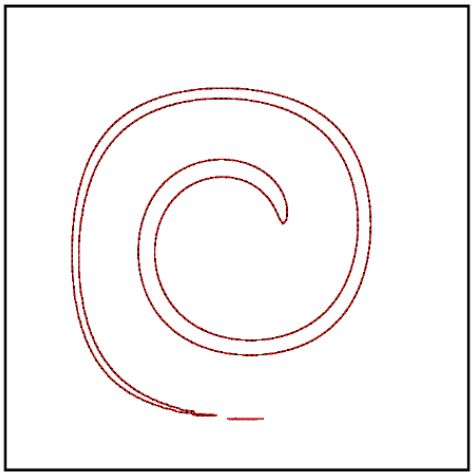
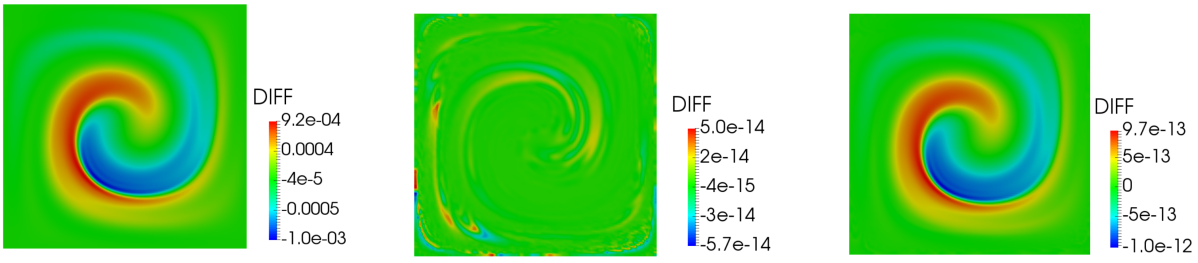
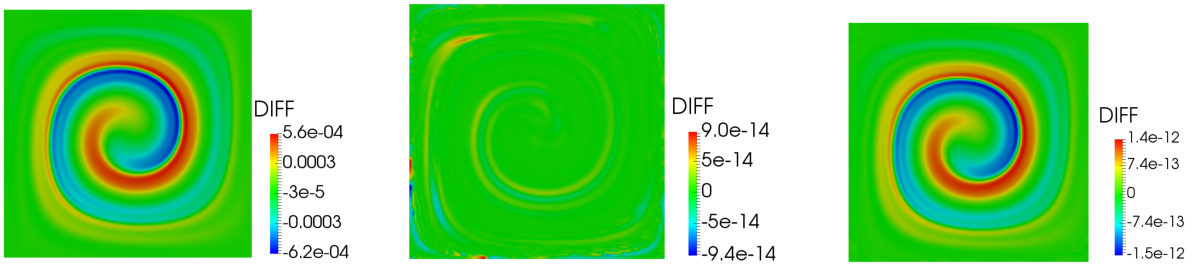


Figure 4.18 – Level-set interface at final time $t = 3$s. In blue: ANM, in red: BPL, in green: IFS, in black: TVDRK4 with a discrete-time step $\Delta t = 10^{-3}s$. No visible distinction between the level set contours.



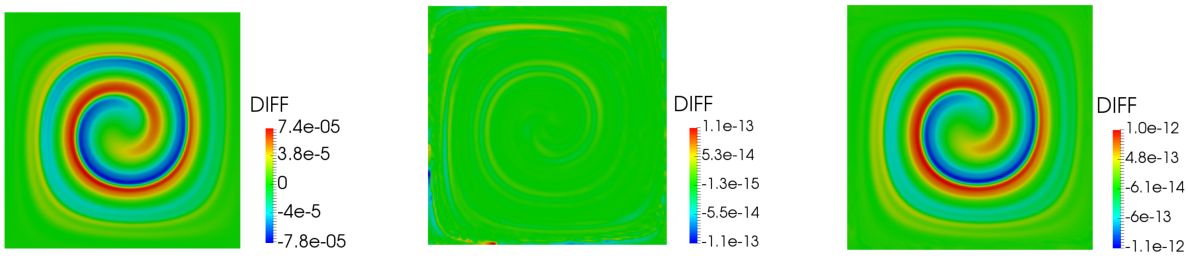
(a) Error between TVDRK4 and ANM (b) Error between ANM and BPL (c) Error between BPL and IFS

Figure 4.19 – Errors between the temporal methods for $t = 1s$.



(a) Error between TVDRK4 and ANM (b) Error between ANM and BPL (c) Error between BPL and IFS

Figure 4.20 – Errors between the temporal methods for $t = 2s$.



(a) Error between TVDRK4 and ANM (b) Error between ANM and BPL (c) Error between BPL and IFS

Figure 4.21 – Errors between the temporal methods for $t = 3s$.

4.5 Conclusion

The numerical time perturbation and resummation methods ANM, BPL, and IFS are tested and compared to the classic integration scheme for solving level set evolution equations in terms of efficiency and robustness. As previously stated, the numerical schemes used for the level set must be robust and precise enough to prevent the resolution code from diverging. We recall that the Meijer-G resummation was not applied to the level set evolution equation because the numerical resolution of the level set was written in the C programming language, which does not support numerical access to the Meijer-G function.

It is found that the proposed methods capture very accurately the evolution of the interface locations for a given time-dependent and independent velocity fields (but depending on space). It has been well shown that BPL and IFS tend to improve the validity domain of the ANM. The validity domain of these three methods is much larger than the time step of the classical TVDRK4 scheme.

Regarding the reinitialization technique in the context of solutions that are continuous

in time, we have to be careful how to perform it. Following various tests, we determine that the best choice is to adopt a classical reinitialization at the end of each ANM continuation step. This choice is effective and has the same advantages as the classical techniques. Regards the CPU time analysis, it is heartening to note that we have revealed that the CPU time required by ANM is approximately 3 times less than that of TVDRK4 for the same set of simulation parameters. Therefore, for this approach, the temporal perturbation methods seems to be more interesting than the classical schemes.

The testing of different reinitialization trials on level set evolution equations and the validation of numerical time perturbation and resummation techniques and in this chapter and on Navier-Stokes equations in Chapter 3, is an important step that will contribute to the process of their implementation to solve full Fluid-Structure Interaction (FSI) problems, or other coupled nonlinear problems.

CONCLUSIONS AND PERSPECTIVES

Conclusions

This thesis was related to the numerical resolution of time-dependent problems. Since the improvement of numerical methods for solving nonlinear evolutionary problems is currently a growing research field, the goal of this thesis was dedicated to the development of the numerical time perturbation and resummations methods for solving time-dependent problems. The main purpose was to study these schemes, as alternatives to time integration methods. The main idea of time perturbation approach is to represent the unknowns of the problem in the form of a polynomial by taking the time as the perturbation parameter. Therefore, an analytical approximation of the solution is then continuous and differentiable on a temporal validity domain to be determined. The most interesting property of this time perturbation approach is that the computed solutions and their derivatives are obtained continuously in time. However, one must be very careful when manipulating time series, since they can suffer from divergence. For this reason, resummation techniques of divergent series are used.

In a brief introduction, we provided an overview of the topic covered in this thesis, followed by a detailed description of the research study objectives. The work has been divided into four chapters.

The first chapter focused on time perturbation methods ANM (Asymptotic Numerical Method) and summation algorithms of divergent series BPL (Borel-Padé Laplace), IFS (Inverse Factorial Series), and Meijer-G approximant (MG) for solving time-dependent problems. This chapter provided the necessary background of these methods, as well as some theoretical concepts and literature results. It also addressed some developments that we have proposed. The developments include a first use of the recent Meijer-G approximant for time-dependent problems, since the latter has been applied for summing divergent series in quantum mechanical problems. According to the literature [38], MG is particularly designed for series with a branch cut. However, the encouraging results presented in [38] motivated us to explore and develop the use of this technique to the sum of divergent time series. Thus, with this new summation method, we did not limit the analysis in this manuscript to BPL and IFS only. For this reason, we developed some points of this exploration. We introduced the MG algorithm to solve temporal problems, for even

and odd truncation orders. We proposed a continuation technique based on the residual criterion, that allows us to obtain an entire solution rather than just one branch of the solution. As a result, the expressions of the derivatives, followed by the residual of MG, have been detailed. We also introduced a new Meijer-G version for the vector series. We also established the expressions of the first and second-order derivatives for BPL and IFS in an explicit way. Finally, we proposed to compare such exact approximations derivatives use with others suggestions in the numerical examples, in order to approximate better the derivatives and obtain the correct solutions.
The second chapter focused on the numerical applications of these methods to nonlinear ODEs in solid mechanics. The third and fourth chapters are devoted to their applications on time-dependent PDEs in fluid mechanics.
The conclusions that can be drawn from the numerical tests in Chapters 2, 3, and 4 will be divided into two main categories: in terms of efficiency feasibility, and computation effort.

We note first that increasing or decreasing the order of time perturbation-resummation schemes is very simple, one can just need to change the value of a parameter in the code. Concerning the calculation of the series terms using recurrence relations, previous researchs on perturbation methods has shown that rewriting nonlinear equations in quadratic form is recommendable. In this study, we then used this concept to propose simple recurrences relations.

In terms of efficiency and feasibility: After different numerical test cases, one can in principle state that the application of the continuation technique using the exact approximations of the derivatives of BPL, IFS, and MG sums, guarantees a good quality of the solution for different nonlinearity parameters of the problem. Moreover, the solution obtained by the latter is also efficient for long-time intervals. Another proposal of approximating the derivatives and calculating the residuals, than those formulas, has been investigated. The results, however, indicate that the latter may cause some shifts in the solution obtained for simulations over long time intervals. For this reason, we evaluated the residuals using the derivatives formulas that we have developed in Appendices.
We start with the conclusion on ANM, BPL, and IFS before moving on to Meijer-G, which has not been applied to all of the test cases in the manuscript (Meijer-G function is not available in all scientific languages).
ANM, BPL, and IFS have been successfully applied on ordinary differential equations, using examples of stiff and highly nonlinear problems where classical schemes suffer from many weaknesses. We have chosen interesting examples from the literature that are known to be excellent examples to evaluate the performance of new time integration techniques. Then, we showed that ANM, BPL, and IFS are well suited to long-term simulation. For

some high nonlinear cases, classical schemes frequently introduce a small time lag at each period, whereas the latter the numerical methods based on perturbation reproduce more faithfully the limit cycle. Furthermore, these methods are suitable for stiff problems and can deal with cases where the system exhibits extremely fast dynamics. These approaches have also been shown to be effective when the considered problem exhibits for a wide range of degrees of freedom. We demonstrated that they are a viable alternative for solving ODEs.

The encouraging results on ODEs, have motivated us to apply the latter also on the time-dependent PDEs in particular for fluid mechanics problems. These latter have demonstrated their ability to provide appropriate solutions for other time-dependent PDEs such as the Burgers equation, the Heat equation, the Navier-Stokes equation and the level set evolution equation. They have proven their high levels of efficiency in terms of stability, robustness, and accuracy for solutions at high Reynolds numbers. The obtained results are consistent with the literature. The simulation is validated for large final times and for different parameters changes such as the Reynolds numbers, the viscosity, and the grid size. We also showed that these methods can be associated to different spatial discretization techniques (finite element method, finite difference method: HJ-WENO scheme, MAC method). A numerical study of two-dimensional (2D) incompressible flow in the Lid-driven cavity for $Re = 12500$ was carried out in detail in the present study. We have verified that the fluid behavior is well respected using these methods, by the velocity stream function plots.
We recall that our long-term goal is to use time perturbation-resummations methods to simulate FSI problems where equations governing the fluid motion are coupled to equations governing the evolution of the free surface. The Navier-Stokes equations and the level set methods are the central elements of such problems.
It is known that the resolution of the level set equation necessitates the use of high-order numerical schemes in order to capture the interface movements and, as a result, respect the mass conservation because this property is unfortunately not inherent in the method. However, it has been demonstrated through the execution of some case studies that combining the temporal perturbation-resummation schemes ANM, BPL, and IFS with the spatial integration schemes of nonlinear terms (HJ-WENO), provides satisfactory results. The perturbation and temporal resummation methods used for the level set have proven to be robust enough to ensure that the solution does not diverge. These latter capture very accurately the evolution of the interface for both time-dependent and time-independent velocity fields. We also emphasized the need to redistribute the level set in order to ensure an accurate calculation of the interface curvature using the reinitialization procedure. Regarding the popular reinitialization technique in the context of time-continuous solutions,

we have seen that one has to be careful how to perform it. Following various numerical tests, we have determined that the best choice is to adopt a classical reinitialization at the end of each ANM continuation step. This choice has the same advantages as the classical techniques. If a difficulty is encountered, it is better to decrease the tolerance accuracy of the chosen method.

Regarding the Meijer-G approximant, we wanted to explore the latter for some time-dependent ODEs and PDEs. For some cases, we found that Meijer-G provides a better approximation of the solution than BPL, ANM, and IFS by using a lower truncation order. It overcomes the fact that poles appears in Padé's coefficient in BPL and leads to numerical issues for the continuation process in some cases. The Meijer-G continuation is validated, allowing to obtain a complete continuous solution in time on all of the requested intervals, rather than just only one power series representation of the solution that is valid in a specific temporal range of the solution. The proposed continuation is based on the residual criterion. As a result, we presented and validated the Meijer-G sum derivative exact approximations. The solution obtained by MG with continuation, agreed with the exact solution and provided the correct limit cycle. For the heat equation, we were able to associate this method with a finite difference spatial discretization method. In addition, we have shown the efficiency of the Meijer-G approximant for a series vector using the Gram-Schmidt procedure. However, other examples revealed some issues for specific areas of the domain and truncation orders. It was found that the Meijer-G sum did not behave smoothly in some areas. The cause is still being investigated. This could be due to the small value of t_{max} in the Meijer-G sum, or to incompatible parameters in the Meijer-G sum, or to the fact that the implementation of the Meijer-G function does not always give the right answers in some specific parameter cases. This is also mentioned in the technical notes of Viktor T. Toth (for Maple implementation) [104]. In all the document, we have plotted only the real part of the Meijer-G solution that interests us. We have noted that there are some cases where the physical interpretation of the obtained imaginary part is not so obvious and it is not our main goal. Our goal is to obtain an accurate approximation of the solution.

In terms of computation effort: We performed CPU time studies for one example of ODEs (Van der Pol oscillator) and all the studied examples of PDEs using ANM, BPL, and IFS. However, we did not perform CPU time analysis for MG because the study of its effectiveness and feasibility for time-dependent problems, is still in progress.
It should be noted that for both small and large problems, IFS is less efficient in terms of computation time than ANM, BPL, and the classical scheme for certain large test cases.

Furthermore, it is extremely sensitive to the mesh size since the CPU time increases significantly when we refine the mesh. Finally, one can say that it becomes slower with higher truncation orders.
For small size problems, we have noticed that when only a moderate accuracy is required, the classical time integration scheme can outperform ANM and BPL in some cases. However, when high accuracy is required, BPL becomes the most attractive. This is especially true when the system is large. When we compared ANM and BPL, we found that BPL is similar to ANM or slightly faster for small size problems.
However, for large problems with high number of degrees of freedom, the BPL algorithm improves ANM and is more than twice as fast. When simulating a high-dimensional problem, such as the Navier-Stokes equations, where the computation of serial terms takes a considerable amount of time, BPL can significantly reduce the computational cost. We found that when comparing the time perturbation and resummation schemes for the same comparison criterion and parameters, we revealed that: IFS is the most expensive in terms of time, BPL was faster than ANM for all examples with a larger t_{max} requiring fewer steps to reach the final time. For solving the level set evolution equation, ANM and BPL was faster than the traditional TVDRK4 scheme. Compared to the classical RK1 scheme, RK1 was significantly faster than IFS in most cases. For the processed heat equation and the Navier-Stokes equation on an L-shaped domain, RK1 was significantly slower than ANM and BPL. For the Burgers equation, RK1 was slightly faster than BPL but much faster than ANM. Because the residual had incorrect values in the case of the LID-driven cavity for $Re = 12500$, only a comparison indication was presented rather than a true comparison based on the residual criterion. According to the studies, the calculation of serial terms is time consuming for some cases of Navier-Stokes problems. Therefore, we need to design a method to speed up the computation of the series terms for the time perturbation and summation methods of Navier-Stokes equations. With respect to grid and mesh size modification, we observed that IFS is the most sensitive to the mesh size. The mesh sensitivity is about the same for ANM and BPL. For some test cases with very fine mesh, RK1 becomes faster than the other schemes.

Perspectives

We end this manuscript by indicating some potential future research directions that may emerge from the work presented here, building on the results summarized in the previous chapters. The perspectives that can be drawn are divided into two ways of work: theoretical and numerical parts.

Theoretically

As we have already seen, the theoretical notion of the methods was skipped in this manuscript and only the numerical aspects of the temporal schemes are discussed. However, optimizations to the algorithm may be possible by considering some theoretical improvements.

- We recall that one of the conditions for the series being 1-summable is that is a Gevrey series of index one (condition for BPL algorithm presented in Chapter 1). Note that Maillet [49] has proven that formal series that are solutions of an algebraic ODE are Gevrey series. Malgrange [50] has extended this observation to the analytic nonlinear ODEs. The theory for PDEs is not quite complete. Only some equations in fluid mechanics have been investigated. For this reason, an addition theoretical examination of the studied equation can be useful to determine the actual Gevrey index and evaluate it numerically using the series coefficients. For some numerical cases treated in this manuscript, the theoretical proof that the Gevrey index is 1 already existed. However, we did not present the Gevrey index studies for remaining test cases.
- We recall that the third step of BPL algorithm is to evaluate the Laplace transform of the Padé approximant in a direction linking the origin to the infinity in the direction θ (where θ is an angle), denoted by $d_\theta = [0, e^{i\theta}\infty[$, to return to the temporal space. Since the singular directions of the Borel transform are not known a priori, we have chosen $\theta = 0$, therefore this direction is set as the real positive axis in this work; $d_0 = \mathbb{R}^+$. We could make a more detailed study of the equations to which we have applied the resummation. This would allow us to find possible singularities in the Borel space and to have more precision on the best direction of integration during the Laplace transformation.
- It is possible to investigate whether there is a relation between the appearance of the imaginary part of the solution obtained by the Meijer-G approximant (see Fig.(2.4) in Chapter 2) and the determination of the upper bound of the validity domain t_{max}. If so, in what circumstances? Otherwise, determining the physical meaning of the imaginary part obtained by MG, related to the problem would be interesting.
- Since MG resummation is recent, the convergence properties are unknown. A convergence study of the algorithm at order $N = 3$ versus order $N = 4$ is presented in [71]. If the theoretical convergence properties are more developed, it may be help to find an explicit expression of t_{max} of MG or other error relation without going through the time-consuming residue criterion.

Numerically

From a numerical point of view, some directions that can arise from our work can be listed in the following:

- The computation time for evaluating the series terms of the Navier-Stokes equations is thought to be long. This is due to the numerical difficulties in dealing with big numbers in the series terms. Some optimization is required to accelerate this evaluation.
- The continuation technique of the time perturbation and numerical resummation schemes should be optimized. In some cases, the time step is quite expensive, in particular by considering the number of residual evaluations, especially for IFS. A less expensive method of quality evaluation should be developed. This should increase the speed of the scheme. Therefore, an efficient continuation criterion that is not based on the residual is requested. By optimizing the algorithm, the efficiency can be increased considerably. The calculation of the residual consumes a significant amount of CPU time in the solution (IFS algorithm). Another technique must be used to evaluate the error.
- Another potential research direction arising from our work in Chapters 3 and 4 is the validation of these perturbation-resummation methods on the coupling of the level set with the Navier-Stokes equations. Many fundamental aspects of this coupling must be investigated. This new class of time integration schemes must be adapted to fluid-structure interaction problems with free surface and real test cases.

  Moreover, another well-known method for tracking the interface than level set is the volume of fluid method (VOF). It has the same advection equation as the level set function, it is much more difficult to solve it recisely due to the abrupt change in the interface transition region. However, it has a significant advantage over the level set method. It is the preservation of the mass conservation. Coupled techniques are expected to combine the advantages of the both methods, namely good interface representation and mass conservation. One can also see how to study these methods using time perturbation methods.
- In some cases, the residuals of the temporal perturbation and resummation methods had inappropriate values during the time, as we saw in the previous chapter for the LID-driven cavity example. It will be interesting to develop a residual corrector that is suitable for the temporal perturbation aspect (in case the solution quality deteriorates). As a result, coupling the time perturbation methods with a nonlinear corrector is preferable.
- Because the Meijer-G approximant was only used as the first application for time-

dependent problems, no CPU time study for MG was presented in this manuscript. As a result, once the effectiveness study of MG is completed, evaluating the CPU time required by this method will be necessary. We notice that for a residual criterion, applying or calling a Meijer-G function $G_{m,n}^{p,q}$ for each time can be time-consuming.

- Modern computer algebra systems, such as Maple, include a Meijer-G function implementation. This implementation can be strictly symbolic, or it can include a numerical code. This function implementation is not yet available for all languages, such as C or Fortran, and a connection via Matlab, Python, Maple, or Mathematica is required. This link can be time-consuming sometimes.
- As we have seen, for some test cases and areas of domain, Meijer-G did not behave smoothly. This could be due to the small value of t_{max} in the Meijer-G relation sum, or to the fact that the implementation of the Meijer-G function does not always give the right answers in some cases of specific parameters [104]; or that these approximants are affected by spurious poles; or the obtained hypergeometrics vectors are not positive to be able to apply the Euler Gamma function. Therefore, it would be interesting to study the cause.
- In the following article that is in progress (https://arxiv.org/pdf/2007.00390.pdf), the linear stability of the BPL integrator has been introduced. It has been shown that BPL is not A-stable as most of the explicit method, but more efficient in solving the stiff problems than classical schemes. Therefore, what about the A-stability study of Meijer-G and the size of the linear stability domain?

Appendix A

Derivatives formulas for Borel-Padé-Laplace

The Borel sum denoted by $Su(t)$ in a direction $d = \mathbb{R}^+$ is given by:

$$Su(t) = u_0 + \int_0^\infty P(\xi)e^{-\xi/t}d\xi \tag{A.1}$$

As already mentioned in Chapter 1, we will note:

— $S(t) = Su(t)$ the Borel sum.

— $\dot{S}(t) = \dfrac{dSu(t)}{dt}$ the first derivative of Borel sum.

— $\ddot{S}(t) = \dfrac{d^2Su(t)}{dt^2}$ the second derivative of Borel sum.

The first derivative of Borel sum is given by:

$$\dot{S}(t) = \frac{dSu(t)}{dt} = \frac{d}{dt}\left(u_0 + \int_0^\infty P(\xi)e^{-\xi/t}d\xi\right) \tag{A.2}$$

$$= \frac{du_0}{dt} + \int_0^\infty P(\xi)\frac{d}{dt}(e^{-\xi/t})d\xi \tag{A.3}$$

$$= 0 + \int_0^\infty P(\xi)(\frac{\xi}{t^2})e^{-\xi/t}d\xi \tag{A.4}$$

$$= \frac{1}{t^2}\int_0^\infty \xi P(\xi)e^{-\xi/t}d\xi \tag{A.5}$$

The change of variable $\xi = rt$, yields to:

$$\dot{S}(t) = \frac{dSu(t)}{dt} = \frac{1}{t^2}\int_0^\infty rtP(rt)e^{-r}d(rt) \tag{A.6}$$

$$= \frac{t^2}{t^2}\int_0^\infty rP(rt)e^{-r}dr \tag{A.7}$$

$$= \int_0^\infty rP(rt)e^{-r}dr \tag{A.8}$$

Finally, the first derivative of $Su(t)$ is

$$\dot{S}(t) = \frac{dSu(t)}{dt} = \int_0^\infty rP(rt)e^{-r}dr \tag{A.9}$$

The second derivative of Borel sum is given by:

$$\ddot{S}(t) = \frac{d^2 Su(t)}{dt^2} = \frac{d}{dt}\left(\frac{1}{t^2}\int_0^\infty \xi P(\xi) e^{-\xi/t} d\xi\right) \tag{A.10}$$

$$= I_1 + I_2 \tag{A.11}$$

with

$$I_1 = \frac{d}{dt}\left(\frac{1}{t^2}\right)\int_0^\infty \xi P(\xi) e^{-\xi/t} d\xi \tag{A.12}$$

and

$$I_2 = \frac{1}{t^2}\frac{d}{dt}\int_0^\infty \xi P(\xi) e^{-\xi/t} d\xi \tag{A.13}$$

For I_1:

$$I_1 = \frac{d}{dt}\left(\frac{1}{t^2}\right)\int_0^\infty \xi P(\xi) e^{-\xi/t} d\xi \tag{A.14}$$

$$= \frac{-2}{t^3}\int_0^\infty \xi P(\xi) e^{-\xi/t} d\xi \tag{A.15}$$

Let the change of variable $\xi = rt$

$$I_1 = \frac{-2}{t^3}\int_0^\infty rtP(rt) e^{-r} d(rt) \tag{A.16}$$

$$= \frac{-2}{t}\int_0^\infty rP(rt) e^{-r} dr \tag{A.17}$$

$$= \frac{-2}{t}\frac{dSu(t)}{dt} \tag{A.18}$$

Therefore

$$I_1 = \frac{-2}{t}\frac{dSu(t)}{dt} \tag{A.19}$$

For I_2:

$$I_2 = \frac{1}{t^2}\frac{d}{dt}\int_0^\infty \xi P(\xi) e^{-\xi/t} d\xi \tag{A.20}$$

$$= \frac{1}{t^2}\int_0^\infty \xi P(\xi)\frac{\xi}{t^2} e^{-\xi/t} d\xi \tag{A.21}$$

$$= \frac{1}{t^4}\int_0^\infty \xi^2 P(\xi) e^{-\xi/t} d\xi \tag{A.22}$$

Let the change of variable $\xi = rt$

$$I_2 = \frac{1}{t^4}\int_0^\infty r^2 t^2 P(rt) e^{-r} d(rt) \tag{A.23}$$

$$= \frac{1}{t}\int_0^\infty r^2 P(rt) e^{-r} dr \tag{A.24}$$

Therefore, the second derivative of Borel sum is

$$\ddot{S}(t) = \frac{d^2 Su(t)}{dt^2} = \frac{-2}{t}\frac{dSu(t)}{dt} + \frac{1}{t}\int_0^\infty r^2 P(rt)e^{-r}dr \tag{A.25}$$

Remark A.0.1 *Note that the evaluation of the derivatives of the Borel sum must start from $t_0 \neq 0$. For this reason, for all the examples we have treated in this manuscript, we have numerically fixed the initial time with a small value very close to 0 (for example $t_0 = 10^{-7}$).*

Therefore, as a summary:

$$Su(t) = u_0 + t\int_0^\infty P(rt)e^{-r}dr \tag{A.26}$$

$$\dot{S}(t) = \frac{dSu(t)}{dt} = \int_0^\infty rP(rt)e^{-r}dr \tag{A.27}$$

$$\ddot{S}(t) = \frac{d^2 Su(t)}{dt^2} = \frac{-2}{t}\frac{dSu(t)}{dt} + \frac{1}{t}\int_0^\infty r^2 P(rt)e^{-r}dr \tag{A.28}$$

A brief recall of Gauss-Laguerre quadrature is presented:
In numerical analysis Gauss–Laguerre quadrature is an extension of the Gaussian quadrature method for approximating the value of integrals of the following kind:

$$\int_0^\infty f(x)e^{-x}dx \tag{A.29}$$

by the following sum

$$\sum_{i=1}^{N_G} w_i f(x_i) \tag{A.30}$$

with w_i are the quadrature coefficients (or weights). The x_i points, or nodes are real, distinct, unique and are the roots of N_G-Laguerre polynomials.
Generally, one can also consider the integrals that have a singularity of the law of power x^α known to $x = 0$, for $\alpha > -1$ with α a real number, that lead to integrals of forms:

$$\int_0^\infty x^\alpha f(x)e^{-x}dx = \sum_{i=1}^{N_G} w_i f(x_i) \tag{A.31}$$

This allows to estimate such integrals accurately for $f(x)$ smooth or polynomial even in the case where α is not an integer.

It should be noticed that the nodes x_i and the weighs w_i depend on α and N_G. Here are two different strategies for calculating first and second derivatives of BPL.

Strategy 1

Strategy 1 is an application of Gauss-Laguerre formula or the generalized formula with $\alpha = 0$.
Borel sum is given by:

$$\begin{aligned} Su(t) &= u_0 + t\int_0^\infty \underbrace{P(rt)}_{H(r)} e^{-r} dr \\ &= u_0 + t\int_0^\infty H(r) e^{-r} dr \\ &= u_0 + t\sum_{i=1}^{N_G} w_i H(r_i) \\ &= u_0 + t\sum_{i=1}^{N_G} w_i P(r_i t) \end{aligned} \tag{A.32}$$

The first derivative of Borel sum is given by:

$$\begin{aligned} \dot{S}(t) = \frac{dS(t)}{dt} &= \int_0^\infty \underbrace{rP(rt)}_{G(r)} e^{-r} dr \\ &= \int_0^\infty G(r) e^{-r} dr = \sum_{i=1}^{N_G} w_i G(r_i) = \sum_{i=1}^{N_G} w_i r_i P(r_i t) \end{aligned} \tag{A.33}$$

The second derivative of Borel sum is given by:

$$\begin{aligned} \ddot{S}(t) = \frac{d^2 Su(t)}{dt^2} &= \frac{-2}{t}\frac{dSu(t)}{dt} + \frac{1}{t}\int_0^\infty \underbrace{r^2 P(rt)}_{Q(r,t)} e^{-r} dr \\ &= \frac{-2}{t}\frac{dSu(t)}{dt} + \frac{1}{t}\sum_{i=1}^{N_G} w_i Q(r_i, t) \\ &= \frac{-2}{t}\frac{dSu(t)}{dt} + \frac{1}{t}\sum_{i=1}^{N_G} w_i {r_i}^2 P(r_i t) \end{aligned} \tag{A.34}$$

Therefore, with $\alpha = 0$, we have

$$Su(t) = u_0 + t\sum_{i=1}^{N_G} w_i P(r_i t) \tag{A.35}$$

$$\dot{S}(t) = \frac{dSu(t)}{dt} = \sum_{i=1}^{N_G} w_i r_i P(r_i t) \tag{A.36}$$

$$\ddot{S}(t) = \frac{d^2 Su(t)}{dt^2} = \frac{-2}{t}\frac{dSu(t)}{dt} + \frac{1}{t}\sum_{i=1}^{N_G} w_i {r_i}^2 P(r_i t) \tag{A.37}$$

For this first way, the Padé solution $P(r_i t)$ is calculated only once for the estimation of the Borel solution, the first and second derivatives. Note that we used this strategy in all the test cases in the manuscript.

Strategy 2

Strategy 2 is the second way is based on generalized Gauss-Laguerre.
For $\alpha = 0$, we have:

$$\begin{aligned} Su(t) &= u_0 + t \int_0^\infty \underbrace{P(rt)}_{H(r)} e^{-r} dr \\ &= u_0 + t \int_0^\infty H(r) e^{-r} dr \\ &= u_0 + t \sum_{i=1}^{N_G} w_i H(r_i) = u_0 + t \sum_{i=1}^{N_G} w_i P(r_i t) \end{aligned} \tag{A.38}$$

For $\alpha = 1$, we have:

$$\begin{aligned} \dot{S}(t) = \frac{dSu(t)}{dt} &= \int_0^\infty r \underbrace{P(rt)}_{G(r)} e^{-r} dr \\ &= \int_0^\infty r G(r) e^{-r} dr \\ &= \sum_{i=1}^{N_G} w_i G(r_i) = \sum_{i=1}^{N_G} w_i P(r_i t) \end{aligned} \tag{A.39}$$

For $\alpha = 2$, we have:

$$\begin{aligned} \ddot{S}(t) = \frac{d^2 Su(t)}{dt^2} &= \frac{-2}{t} \frac{dSu(t)}{dt} + \frac{1}{t} \int_0^\infty r^2 \underbrace{P(rt)}_{Q(r,t)} e^{-r} dr \\ &= \frac{-2}{t} \frac{dSu(t)}{dt} + \frac{1}{t} \int_0^\infty r^2 Q(r,t) e^{-r} dr \\ &= \frac{-2}{t} \frac{dSu(t)}{dt} + \frac{1}{t} \sum_{i=1}^{N_G} w_i P(r_i t) \end{aligned} \tag{A.40}$$

Therefore,

$$Su(t) = u_0 + t\sum_{i=1}^{N_G} w_i P(r_i t) \qquad \alpha = 0 \tag{A.41}$$

$$\dot{S}(t) = \frac{dSu(t)}{dt} = \sum_{i=1}^{N_G} w_i P(r_i t) \qquad \alpha = 1 \tag{A.42}$$

$$\ddot{S}(t) = \frac{d^2 Su(t)}{dt^2} = \frac{-2}{t}\frac{dSu(t)}{dt} + \frac{1}{t}\sum_{i=1}^{N_G} w_i P(r_i t) \qquad \alpha = 2 \tag{A.43}$$

For this strategy, the development of Padé is not the same for Borel sum and its derivative since the roots and the weight depend on α. As a result, this strategy is more time-consuming than the first one. However, it has been presented for the integrals that have a singularity of the law of power x^α known to $x = 0$, for $\alpha > -1$. It was not used in any of the test cases presented.

Appendix B

Derivatives formulas for Inverse factorial series

In this Appendix, we develop the first and second derivatives of Inverse factorial series $I(t)$ of the second way of the subsection 1.3 (large order N) in Chapter 1. We recall some definition of this subsection before starting to evaluate the derivatives.
Suppose that $u(t)$ is 1-summable in a direction d, with an angle θ with the positive half-axis. Let $\tau_l = s_l e^{i\theta}$ with $l = (1, 2, \ldots, N)$, where $\{s_1, s_2, \ldots, s_N\}$ is any sequence of complex numbers. Let

$$z = \frac{1}{t}, \qquad y = ze^{i\theta} \tag{B.1}$$

$$a_1 = \frac{u_0}{z}, a_2 = \frac{u_1}{z^2}, \ldots, a_m = \frac{u_{m-1}}{z^m}, \quad m = 1, \ldots N \tag{B.2}$$

Note that if the series are summable in $\mathbb{R}^+$ direction, so $y = z$ and we will choose $s_l = l$ for $l \geq 1$.

We call Inverse factorial series the following :

$$I(t) = \frac{1}{t}\sum_{n=0}^{N-1} v_{n+1} = z\sum_{n=0}^{N-1} v_{n+1} = z(v_1 + v_2 + \cdots + v_N) \tag{B.3}$$

where v_n is the n-th term of IFS which is calculated using the following recursive algorithm based on

$$v_{n+1}^{(j)} = \frac{\tau_{n-1} v_n^{(j)} + y v_n^{(j+1)}}{y + \tau_n} \qquad n \geq 1, j \geq 1, \tag{B.4}$$

with

$$v_1^{(1)} = a_1 \qquad v_1^{(2)} = a_2 \qquad \ldots \qquad v_1^{(N)} = a_N \tag{B.5}$$

The terms $v_1^{(1)}, v_2^{(1)}, \ldots, v_n^{(1)}, v_{n+1}^{(1)}$ are the terms of the IFS denoted by: $v_1, v_2, \ldots, v_n, v_{n+1}$.

First derivative

The first derivative of Inverse factorial series is given by:

$$\dot{I}(t) = \frac{dI(t)}{dt} = \dot{z}\,(v_1 + v_2 + \cdots + v_N) + z(\dot{v_1} + \dot{v_2} + \cdots + \dot{v}_N) \tag{B.6}$$

We have

$$\dot{z} = \frac{-1}{t^2} \tag{B.7}$$

We will build a recursive algorithm of derivative to evaluate $\dot{I}(t)$.
We have

$$a_m = \frac{u_{m-1}}{z^m} \implies \dot{a}_m = \frac{-mu_{m-1}\dot{z}}{z^{m+1}} \tag{B.8}$$

This algorithm for $n \geq 1$ is based on

$$\frac{dv_{n+1}^{(j)}}{dt} = \frac{\left(\tau_{n-1}\dfrac{dv_n^{(j)}}{dt} + \dfrac{dy}{dt}v_n^{(j+1)} + y\dfrac{dv_n^{(j+1)}}{dt}\right)(y+\tau_n) - \dfrac{dy}{dt}\left(\tau_{n-1}v_n^{(j)} + yv_n^{(j+1)}\right)}{(y+\tau_n)^2} \tag{B.9}$$

with $\tau_0 = 0$ and

$$\frac{dv_1^{(i)}}{dt} = (\dot{v_1^{(i)}}) = \dot{a}_i \quad i \geq 1 \tag{B.10}$$

and

$$\frac{dy}{dt} = \dot{y} = \dot{z}e^{i\theta} \tag{B.11}$$

If the series are summable in the $\mathbb{R}^+$ direction, so

$$\dot{y} = \dot{z} \tag{B.12}$$

The terms $\dfrac{dv_1^{(1)}}{dt}, \dfrac{dv_2^{(1)}}{dt}, \ldots, \dfrac{dv_n^{(1)}}{dt}, \dfrac{dv_{n+1}^{(1)}}{dt} \ldots$ are the terms of the D_{IFS}, (first derivative terms of IFS) denoted by $\dot{v}_1, \dot{v}_2, \ldots, \dot{v}_n, \dot{v}_{n+1}$.

Second derivative

The second derivative of Inverse factorial series is given by:

$$\begin{aligned}\ddot{I}(t) = \frac{d^2I(t)}{dt^2} =& \ddot{z}(v_1 + v_2 + \cdots + v_{N+1}) + 2\dot{z}(\dot{v_1} + \dot{v_2} + \cdots + \dot{v}_{N+1}) \\ &+ z(\ddot{v_1} + \ddot{v_2} + \cdots + \ddot{v}_{N+1})\end{aligned} \tag{B.13}$$

We have

$$\ddot{z} = \frac{2}{t^3} \tag{B.14}$$

and

$$\dot{a}_m = \frac{-mu_{m-1}\dot{z}}{z^{m+1}} \implies \ddot{a}_m = \frac{-mu_{m-1}\left(\ddot{z}z^{m+1} - (m+1)z^m\dot{z}^2\right)}{z^{2(m+1)}} \tag{B.15}$$

We will build a recursive algorithm of second derivative. Since the second derivative is complicated to write it directly, we will evaluate it using A, B, C such that

$$A = \left(\tau_{n-1}\frac{dv_n^{(j)}}{dt} + \frac{dy}{dt}v_n^{(j+1)} + y\frac{dv_n^{(j+1)}}{dt}\right)(y+\tau_n) \tag{B.16}$$

and

$$B = -\dot{y}\left(\tau_{n-1}v_n^{(j)} + yv_n^{(j+1)}\right) \tag{B.17}$$

and

$$C = (y+\tau_n)^2 \tag{B.18}$$

The derivative of Eq.(B.9) is given by

$$\frac{d^2v_{n+1}^{(j)}}{dt^2} = \frac{(\dot{A}+\dot{B})C - \dot{C}(A+B)}{C^2} \tag{B.19}$$

with

$$\begin{aligned}\dot{A} = &\left(\tau_{n-1}\frac{d^2v_n^{(j)}}{dt^2} + \ddot{y}v_n^{(j+1)} + \dot{y}\frac{dv_n^{(j+1)}}{dt} + \dot{y}\frac{dv_n^{(j+1)}}{dt} + y\frac{dv_n^{(j+1)}}{dt}\right)(y+\tau_n)\\ &+ \dot{y}\left(\tau_{n-1}\frac{dv_n^{(j)}}{dt} + \dot{y}v_n^{(j+1)} + y\frac{dv_n^{(j+1)}}{dt}\right)\end{aligned} \tag{B.20}$$

and

$$\dot{B} = -\ddot{y}\left(\tau_{n-1}v_n^{(j)} + yv_n^{(j+1)}\right) - \dot{y}\left(\tau_{n-1}\frac{dv_n^{(j)}}{dt} + \dot{y}v_n^{(j+1)} + y\frac{dv_n^{(j+1)}}{dt}\right) \tag{B.21}$$

and

$$\dot{C} = 2(y+\tau_n)\dot{y} \tag{B.22}$$

with

$$\frac{d^2v_1^{(i)}}{dt^2} = \ddot{a}_i \quad i \geq 1 \tag{B.23}$$

with

$$\ddot{y} = \ddot{z}e^{i\theta} \tag{B.24}$$

If the series are summable in the $\mathbb{R}^+$ direction, therefore

$$\ddot{y} = \ddot{z} \tag{B.25}$$

The terms $\frac{d^2v_1^{(1)}}{dt^2}, \frac{d^2v_2^{(1)}}{dt^2}, \ldots, \frac{d^2v_n^{(1)}}{dt^2}, \frac{d^2v_{n+1}^{(1)}}{dt^2}$ are the terms of the D^2_{IFS}, (second derivative terms of IFS) denoted by $\ddot{v}_1, \ddot{v}_2, \ldots, \ddot{v}_n, \ddot{v}_{n+1}$.

Appendix C

Derivatives formulas for Meijer-G sum

The Meijer-G sum denoted by $S_{MG}(t)$ is given by:

$$S_{MG}(t) = \frac{\prod_{i=1}^{l} \Gamma(-y_i)}{\prod_{i=1}^{l} \Gamma(-x_i)} G^{l+2,1}_{l+1,l+2}\left(\begin{array}{c} 1, -y_1, \ldots, -y_l \\ 1, 1, -x_1, \ldots, -x_l \end{array} \middle| -\frac{q_l}{p_l t} \right) \tag{C.1}$$

Concerning the derivatives of arbitrary order h of the Meijer-G sum, first one needs to recall the following relationship for Meijer-G function.
We have $\forall z \neq 0$:

$$z^h \frac{d^h}{dz^h} G^{m,n}_{p,q}\left(\begin{array}{c} a_1, \ldots, a_p \\ b_1, \ldots, b_q \end{array} \middle| z^{-1} \right) = (-1)^h G^{m,n+1}_{p+1,q+1}\left(\begin{array}{c} 1-h, a_1, \ldots, a_q \\ b_1, \ldots, b_q \end{array} \middle| z^{-1} \right) \tag{C.2}$$

Therefore

$$\frac{d^h}{dz^h} G^{m,n}_{p,q}\left(\begin{array}{c} a_1, \ldots, a_p \\ b_1, \ldots, b_q \end{array} \middle| z^{-1} \right) = \frac{(-1)^h}{z^h} G^{m,n+1}_{p+1,q+1}\left(\begin{array}{c} 1-h, a_1, \ldots, a_q \\ b_1, \ldots, b_q \end{array} \middle| z^{-1} \right) \tag{C.3}$$

Therefore

$$\frac{d^h}{dz^h} G^{m,n}_{p,q}\left(\begin{array}{c} a_1, \ldots, a_p \\ b_1, \ldots, b_q \end{array} \middle| w z^{-1} \right) = \frac{(-1)^h}{z^h} G^{m,n+1}_{p+1,q+1}\left(\begin{array}{c} 1-h, a_1, \ldots, a_q \\ b_1, \ldots, b_q \end{array} \middle| w z^{-1} \right) \tag{C.4}$$

where w is a constant.

First derivative

Thus, the first derivative (order of derivative $h = 1$) of Meijer-G sum is:

$$\dot{S}_{MG}(t) = \frac{dS_{MG}(t)}{dt} = \frac{\prod_{i=1}^{l} \Gamma(-y_i)}{\prod_{i=1}^{l} \Gamma(-x_i)} \frac{-1}{t} G_{l+2,l+3}^{l+2,2} \left(\begin{array}{c} 0, 1, -y_1, \ldots, -y_l \\ 1, 1, -x_1, \ldots, -x_l, 1 \end{array} \middle| -\frac{q_l}{p_l t} \right) \tag{C.5}$$

Second derivative

The second derivative (order of derivative $h = 2$) of Meijer-G sum is:

$$\ddot{S}_{MG}(t) = \frac{d^2 S_{MG}(t)}{dt^2} = \frac{\prod_{i=1}^{l} \Gamma(-y_i)}{\prod_{i=1}^{l} \Gamma(-x_i)} \frac{1}{t^2} G_{l+2,l+3}^{l+2,2} \left(\begin{array}{c} -1, 1, -y_1, \ldots, -y_l \\ 1, 1, -x_1, \ldots, -x_l, 1 \end{array} \middle| -\frac{q_l}{p_l t} \right) \tag{C.6}$$

Derivative of order h

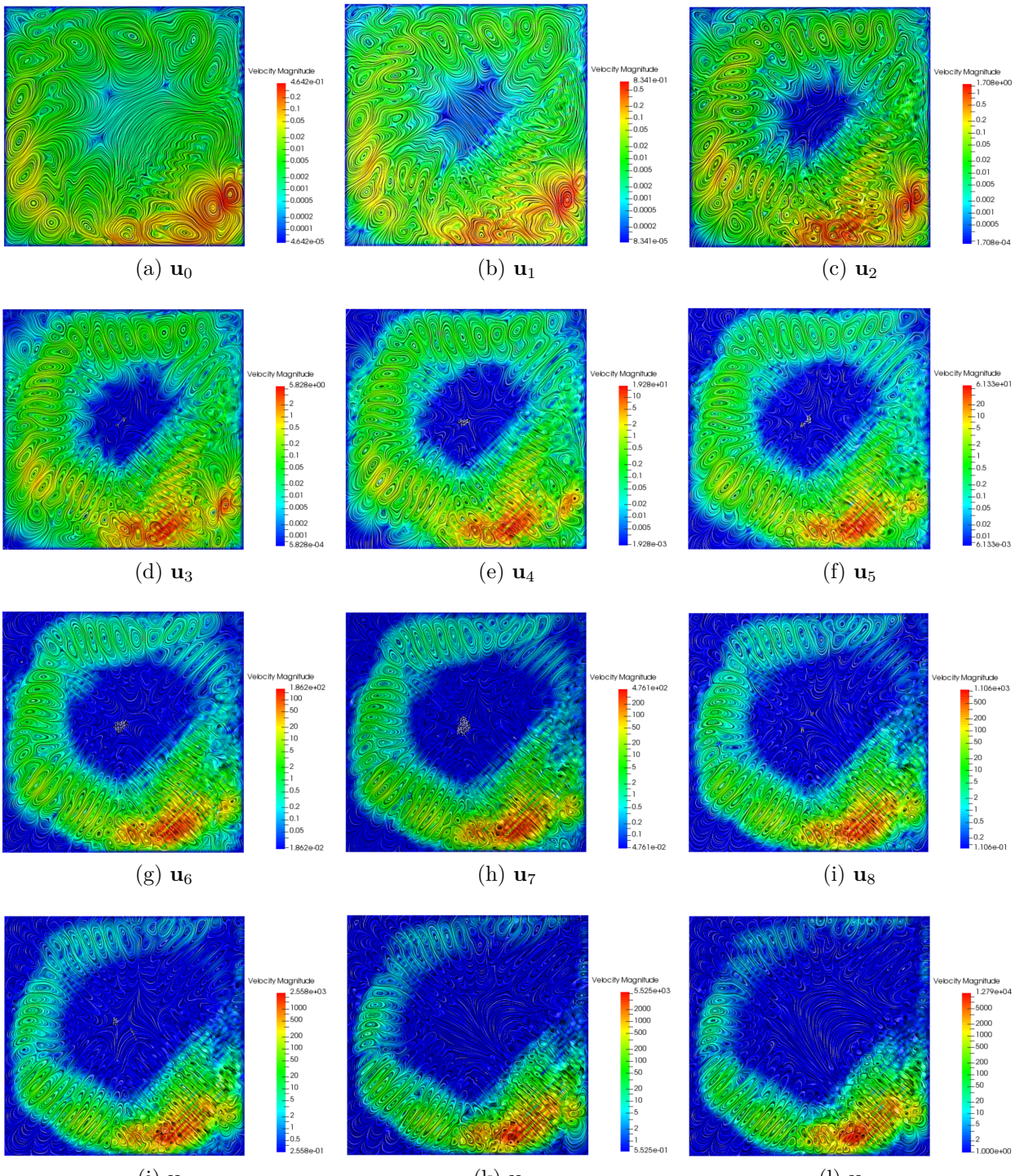
The derivative of order h of Meijer-G sum is:

$$S_{MG}^{(h)}(t) = \frac{d^h S_{MG}(t)}{dt^h} = \frac{\prod_{i=1}^{l} \Gamma(-y_i)}{\prod_{i=1}^{l} \Gamma(-x_i)} \frac{(-1)^h}{t^h} G_{l+2,l+3}^{l+2,2} \left(\begin{array}{c} 1-h, 1, -y_1, \ldots, -y_l \\ 1, 1, -x_1, \ldots, -x_l, 1 \end{array} \middle| -\frac{q_l}{p_l t} \right) \tag{C.7}$$

Appendix D

Velocity series terms for the Lid-driven cavity example at $t_0 = 400$s for a Reynolds number fixed to $Re = 12500$

The behavior of the velocity series terms $\mathbf{u}$ for the Lid-driven cavity example, for an initial time $t_0 = 400$s, a Reynolds number fixed to $Re = 12500$ and a truncation order that varies from $N = 0$ to $N = 35$ are shown in Fig.(D.1), Fig.(D.2) and Fig.(D.3).



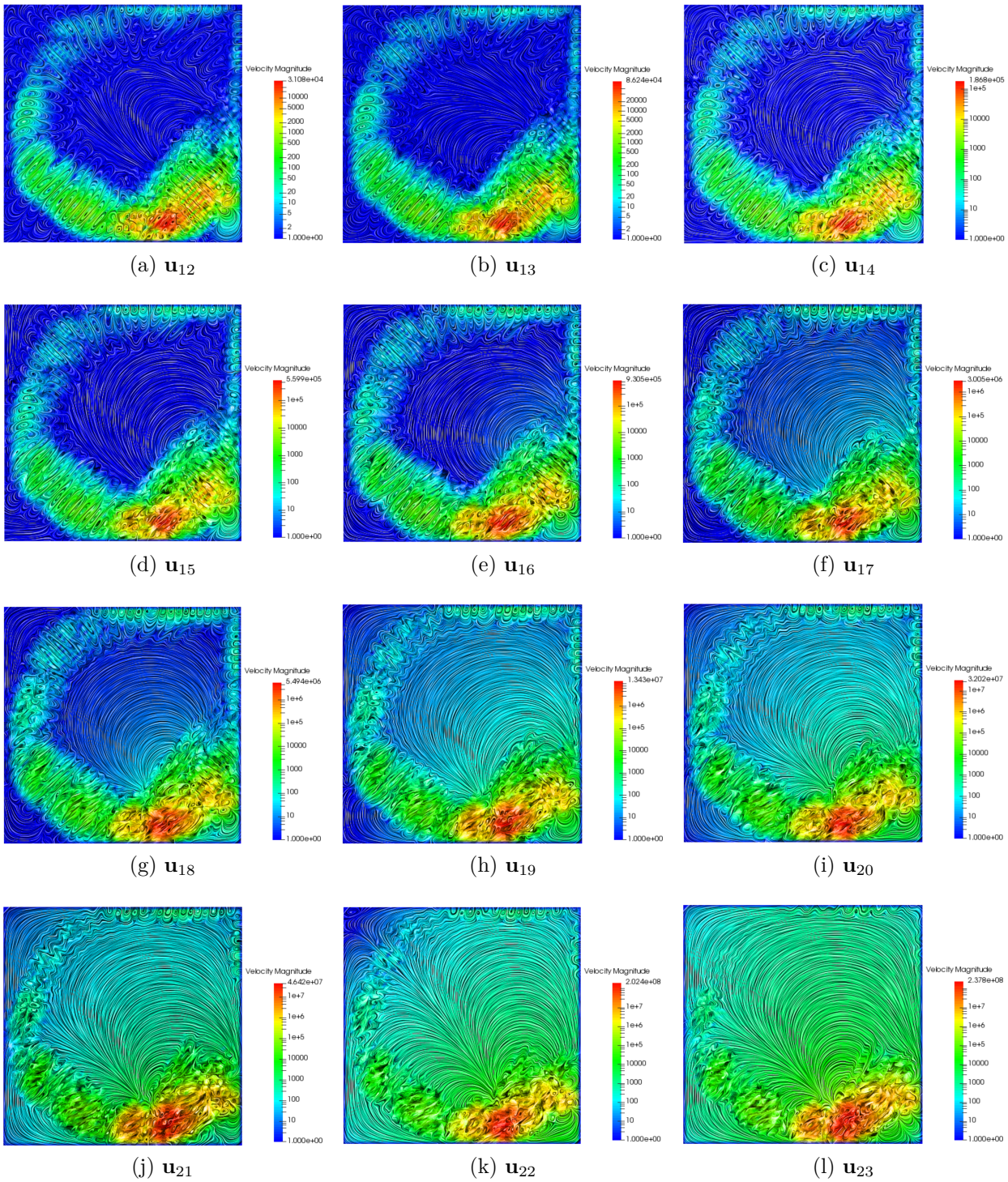
(a) $\mathbf{u}_0$ (b) $\mathbf{u}_1$ (c) $\mathbf{u}_2$

(d) $\mathbf{u}_3$ (e) $\mathbf{u}_4$ (f) $\mathbf{u}_5$

(g) $\mathbf{u}_6$ (h) $\mathbf{u}_7$ (i) $\mathbf{u}_8$

(j) $\mathbf{u}_9$ (k) $\mathbf{u}_{10}$ (l) $\mathbf{u}_{11}$

Figure D.1 – Velocity series terms $\mathbf{u}_0$ to $\mathbf{u}_{11}$ for an initial time $t_0 = 400$s and a Reynolds number $Re = 12500$.



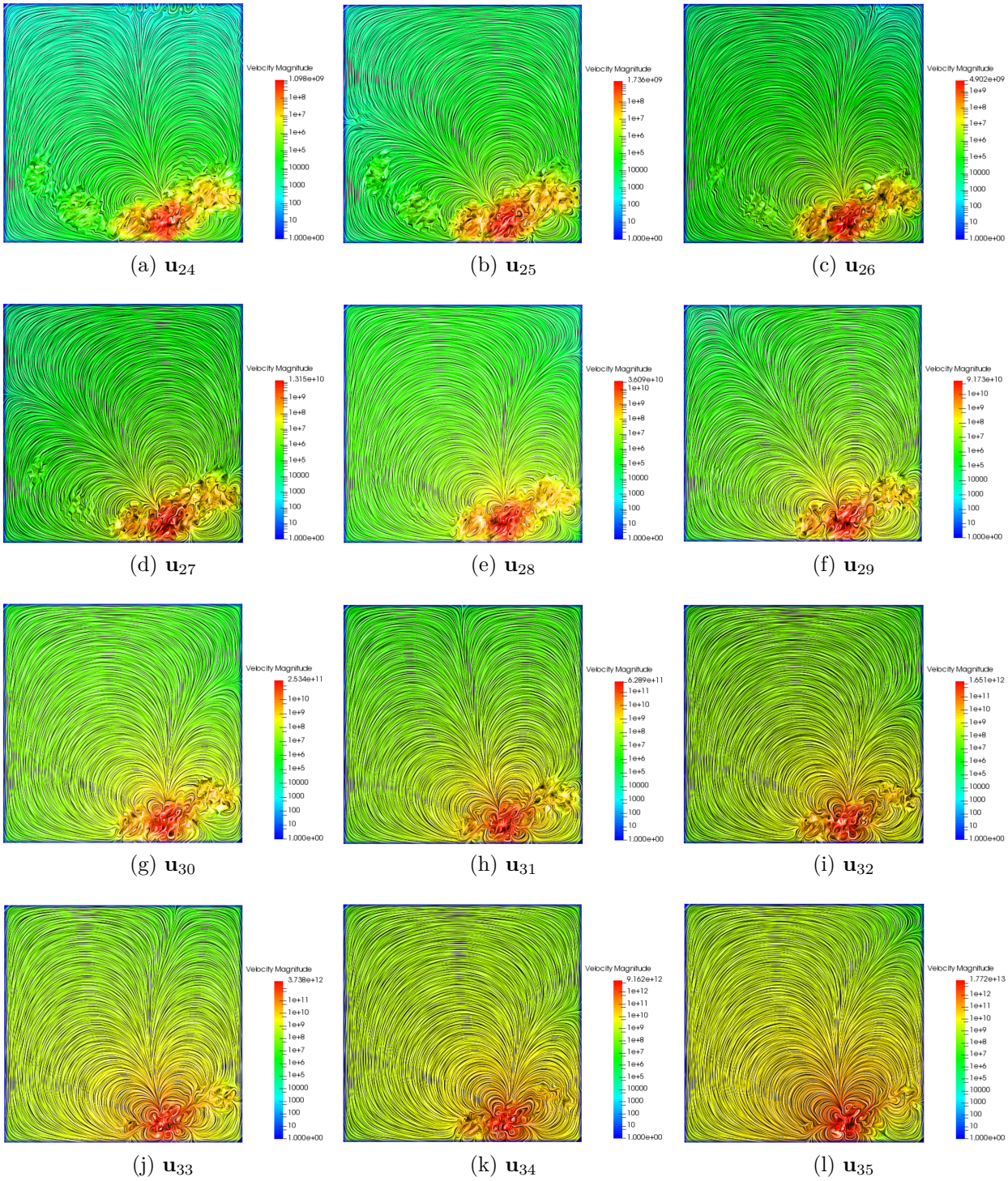
(a) $\mathbf{u}_{12}$ (b) $\mathbf{u}_{13}$ (c) $\mathbf{u}_{14}$

(d) $\mathbf{u}_{15}$ (e) $\mathbf{u}_{16}$ (f) $\mathbf{u}_{17}$

(g) $\mathbf{u}_{18}$ (h) $\mathbf{u}_{19}$ (i) $\mathbf{u}_{20}$

(j) $\mathbf{u}_{21}$ (k) $\mathbf{u}_{22}$ (l) $\mathbf{u}_{23}$

Figure D.2 – Velocity series terms $\mathbf{u}_{12}$ to $\mathbf{u}_{23}$ for an initial time $t_0 = 400$s and a Reynolds number $Re = 12500$.



(a) $\mathbf{u}_{24}$ (b) $\mathbf{u}_{25}$ (c) $\mathbf{u}_{26}$

(d) $\mathbf{u}_{27}$ (e) $\mathbf{u}_{28}$ (f) $\mathbf{u}_{29}$

(g) $\mathbf{u}_{30}$ (h) $\mathbf{u}_{31}$ (i) $\mathbf{u}_{32}$

(j) $\mathbf{u}_{33}$ (k) $\mathbf{u}_{34}$ (l) $\mathbf{u}_{35}$

Figure D.3 – Velocity series terms $\mathbf{u}_{24}$ to $\mathbf{u}_{35}$ for an initial time $t_0 = 400$s and a Reynolds number $Re = 12500$.

BIBLIOGRAPHY

[1] L. Euler, *Institutionum Calculi Integralis.*: Petropoli: Impensis Academia Imperialis Scientiarum, 1768, vol. 3.

[2] L. Muscat, « Coupling of time integration schemes for compressible unsteady flows », PhD thesis, Université de Toulouse, 2019.

[3] R. Courant, K. Friedrichs, and H. Lewy, « Über die partiellen Differenzengleichungen der mathematischen Physik », *Mathematische Annalen*, vol. 100, pp. 32–74, Jan. 1928. DOI: 10.1007/BF01448839.

[4] A. H. van Zuijlen and H. Bijl, « Implicit and explicit higher order time integration schemes for structural dynamics and fluid-structure interaction computations », *Computers & Structures*, vol. 83, no. *2*, pp. 93–105, 2005, Advances in Analysis of Fluid Structure Interaction, ISSN: 0045-7949. DOI: https://doi.org/10.1016/j.compstruc.2004.06.003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045794904002962.

[5] Y. M. Hamada, « Confirmation of accuracy of generalized power series method for the solution of point kinetics equations with feedback », *Annals of Nuclear Energy*, vol. 55, pp. 184–193, 2013, ISSN: 0306-4549. DOI: https://doi.org/10.1016/j.anucene.2012.12.013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306454912004938.

[6] A. E. Aboanber and Y. M. Hamada, « Power series solution (PWS) of nuclear reactor dynamics with newtonian temperature feedback », *Annals of Nuclear Energy*, vol. 30, no. *10*, pp. 1111–1122, 2003, ISSN: 0306-4549. DOI: https://doi.org/10.1016/S0306-4549(03)00033-1. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306454903000331.

[7] R. Lozi and A. N. Pchelintsev, « A New Reliable Numerical Method for Computing Chaotic Solutions of Dynamical Systems: The Chen Attractor Case », *International Journal of Bifurcation and Chaos*, vol. 25, p. 1 550 187, Dec. 2015. DOI: 10.1142/S0218127415501874.

[8] R. Lozi, V. A. Pogonin, and A. N. Pchelintsev, « A new accurate numerical method of approximation of chaotic solutions of dynamical model equations with quadratic nonlinearities », *Chaos, Solitons & Fractals*, vol. 91, pp. 108–114, 2016, ISSN: 0960-0779. DOI: https://doi.org/10.1016/j.chaos.2016.05.010.

[Online]. Available: http://www.sciencedirect.com/science/article/pii/S0960077916301850.

[9] M. Fafard, K. Henchi, G. Gendron, and S. Ammar, « Application of an asymptotic method to transient dynamic problems », *Journal of Sound and Vibration*, vol. 208, no. *1*, pp. 73–99, 1997, ISSN: 0022-460X. DOI: https://doi.org/10.1006/jsvi.1997.1169. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0022460X9791169X.

[10] N. Berrahma-Chekroun, M. Fafard, and J. Gervais, « Regular Article », *Journal of Sound and Vibration*, vol. 243, no. *3*, pp. 475–501, 2001. DOI: 10.1006/jsvi.2000.3423.

[11] N. Berrahma-Chekroun and M. Fafard, « Resolution of the transient dynamic problem with arbitrary loading using the asymptotic method », *Journal of Sound and Vibration*, vol. 243(3), pp. 475–501, 2001.

[12] B. Braikat, M. Jamal, and N. Damil, « A High Order Continuation Based On Time Power Series Expansion And Time Rational Representation For Solving Nonlinear Structural Dynamic Problems », *International Journal of Research in Engineering and Science*, vol. 1, no. *6*, pp.29–34, 2013, ISSN: 2320-9356.

[13] S. Fellah, B. Braikat, J. Mohammad, and N. Damil, « An approach based on time expansion, Padé approximants and Shanks transformation for linear structural dynamic problems », *Journal Maghrébin de Physique*, vol. 2, pp. 9–18, Jan. 2002.

[14] D. Razafindralandy, « Contribution à l'étude mathématique et numérique de la simulation des grandes échelles », PhD thesis, Université de la Rochelle, 2005.

[15] E. Daya and M. Potier-Ferry, « A numerical method for nonlinear eigenvalue problems application to vibrations of viscoelastic structures », *Computers & Structures*, vol. 79, no. *5*, pp. 533–541, 2001, ISSN: 0045-7949.

[16] E. H. Boutyour, H. Zahrouni, M. Potier-Ferry, and M. Boudi, « Asymptotic-numerical method for buckling analysis of shell structures with large rotations », *Journal of Computational and Applied Mathematics*, vol. 168, no. *1*, pp. 77–85, 2004, Selected Papers from the Second International Conference on Advanced Computational Methods in Engineering (ACOMEN 2002), ISSN: 0377-0427. DOI: https://doi.org/10.1016/j.cam.2003.05.010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377042703009658.

[17] F. Boumediene, A. Miloudi, J. M. Cadou, L. Duigou, and E. H. Boutyour, « Non-linear forced vibration of damped plates by an asymptotic numerical method », *Computers & Structures*, vol. 87, no. *23*, pp. 1508–1515, 2009, ISSN: 0045-7949. DOI: https://doi.org/10.1016/j.compstruc.2009.07.005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045794909001990.

[18] J. M. Cadou, M. Potier-Ferry, B. Cochelin, and N. Damil, « ANM for stationary Navier–Stokes equations and with Petrov–Galerkin formulation », *International Journal for Numerical Methods in Engineering*, vol. 50, no. *4*, pp. 825–845, 2001. DOI: 10.1002/1097-0207(20010210)50:4<825::AID-NME53>3.0.CO;2-0. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/1097-0207%2820010210%2950%3A4%3C825%3A%3AAID-NME53%3E3.0.CO%3B2-0. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/1097-0207%2820010210%2950%3A4%3C825%3A%3AAID-NME53%3E3.0.CO%3B2-0.

[19] J. M. Cadou, M. Potier-Ferry, and B. Cochelin, « A numerical method for the computation of bifurcation points in fluid mechanics », *European Journal of Mechanics - B/Fluids*, vol. 25, no. *2*, pp. 234–254, 2006, ISSN: 0997-7546. DOI: https://doi.org/10.1016/j.euromechflu.2005.07.002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0997754605000701.

[20] Y. Guevel, E. H. Boutyour, and J. M. Cadou, « Automatic detection and branch switching methods for steady bifurcation in fluid mechanics », *Journal of Computational Physics*, vol. 230, no. *9*, pp. 3614–3629, 2011, ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2011.02.004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999111000866.

[21] Y. Guevel, G. Girault, and J. M. Cadou, « Numerical comparisons of high-order nonlinear solvers for the transient Navier–Stokes equations based on homotopy and perturbation techniques », *Journal of Computational and Applied Mathematics*, vol. 289, pp. 356–370, 2015, Sixth International Conference on Advanced Computational Methods in Engineering (ACOMEN 2014), ISSN: 0377-0427. DOI: http://dx.doi.org/10.1016/j.cam.2014.12.008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377042714005500.

[22] M. Medale and B. Cochelin, « High performance computations of steady-state bifurcations in 3D incompressible fluid flows by Asymptotic Numerical Method », *Journal of Computational Physics*, vol. 299, pp. 581–596, 2015, ISSN: 0021-9991. DOI: http://dx.doi.org/10.1016/j.jcp.2015.07.021. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999115004660.

[23] O. Costin, « On Borel summation and Stokes phenomena for rank one nonlinear systems of ODE's », *Duke Mathematical Journal*, vol. 93, p. 289, 1998.

[24] D. Lutz, M. Miyake, and R. Schäfke, « On the Borel Summability of Divergent Solutions of the Heat Equation », *Nagoya Mathematical Journal*, vol. 154, p. 1, 1999.

[25] G. Lysik, « Borel summable solutions of the Burgers equation », *Annales Polonici Mathematici*, vol. 95, pp. 187–197, 2009.

[26] S. Michalik, « Summability of formal solutions to the n-dimensional inhomogeneous heat equation », *Journal of Mathematical Analysis and Applications*, vol. 347, p. 323, 2008.

[27] *Divergent Series Redux*, http://www.numericana.com/answer/sums.htm.

[28] E. Borel, « Mémoire sur les séries divergentes », *Annales scientifiques de l'E.N.S. 3ème série*, vol. 16, p. 9, 1899.

[29] J. Thomann, « Procédés formels et numériques de sommation de séries solutions d'équations différentielles », *Journées X-UPS 1991*, vol. 91, pp. 101–114, 1991.

[30] C. Brezinski, « Rationnal approximation to formal power serie », *Journal of Approximation Theory*, p. 295, 1979.

[31] P. Graves-Morris and D. Roberts, « Problems and progress in vector Padé approximation », *Journal of Computational and Applied Mathematics*, vol. 77, no. *1*, pp. 173–200, 1997, ISSN: 0377-0427. DOI: https://doi.org/10.1016/S0377-0427(96)00127-6. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377042796001276.

[32] D. Razafindralandy and A. Hamdouni, « Time integration algorithm based on divergent series resummation, for ordinary and partial differential equations », *Journal of Computational Physics*, vol. 236, pp. 56–73, 2013, ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2012.10.022. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999112006237.

[33] A. Deeb, « Intégrateurs temporels basés sur la resommation des séries divergentes. Application en mécanique », PhD thesis, Université de la Rochelle, 2015.

[34] D. Razafindralandy, A. Hamdouni, and C. Allery, « Numerical divergent series resummation in fluid flow simulation », *European Journal of Computational Mechanics*, vol. 17, no. *4*, pp. 431–451, 2008. DOI: 10.3166/remn.17.431-451. eprint: https://doi.org/10.3166/remn.17.431-451. [Online]. Available: https://doi.org/10.3166/remn.17.431-451.

[35] A. Deeb, A. Hamdouni, E. Liberge, and D. Razafindralandy, « Borel-Laplace summation method used as time integration scheme », *ESAIM: Procedings and Surveys*, vol. 45, p. 318, 2014.

[36] D. Razafindralandy, V. Salnikov, A. Hamdouni, and A. Deeb, « Some robust integrators for large time dynamics », *Advanced Modeling and Simulation in Engineering Sciences*, vol. 6, no. *1*, Mar. 2019. DOI: 10.1186/s40323-019-0130-2. [Online]. Available: https://doi.org/10.1186%2Fs40323-019-0130-2.

[37] D. Razafindralandy, A. Hamdouni, and A. Deeb, « Considering inverse factorial series as time integration method », *AIP Conference Proceedings*, vol. 1798, no. *1*, p. 020 129, 2017. DOI: 10.1063/1.4972721. [Online]. Available: https://aip.scitation.org/doi/abs/10.1063/1.4972721.

[38] H. Mera, T. G. Pedersen, and B. K. Nikolić, « Fast summation of divergent series and resurgent transseries from Meijer-G approximants », *Phys. Rev. D*, vol. 97, p. 105 027, 10 2018. DOI: 10.1103/PhysRevD.97.105027. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevD.97.105027.

[39] L. Guillot, B. Cochelin, and C. Vergez, « A Taylor series-based continuation method for solutions of dynamical systems », *Nonlinear Dynamics*, vol. 98, pp. 2827–2845, May 2019. DOI: 10.1007/s11071-019-04989-5.

[40] B. Cochelin, « A path-following technique via an asymptotic-numerical method », *Computers & Structures*, vol. 53, no. *5*, pp. 1181–1192, 1994, ISSN: 0045-7949. DOI: https://doi.org/10.1016/0045-7949(94)90165-1. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0045794994901651.

[41] S. Baguet, « Stability of thin-shell structures and imperfection sensitivity analysis with the Asymptotic Numerical Method », PhD thesis, Université de la Méditerranée - Aix-Marseille II, 2001.

[42] A. Najah, B. Cochelin, N. Damil, and M. Potier-Ferry, « A critical review of asymptotic numerical methods », *Archives of Computational Methods in Engineering*, vol. 5, no. *1*, pp. 31–50, 1998, ISSN: 1886-1784. DOI: 10.1007/BF02736748. [Online]. Available: https://doi.org/10.1007/BF02736748.

[43] N. Damil, M. Potier-Ferry, A. Najah, R. Chari, and H. Lahmam, « An iterative method based upon Padé approximants », *Communications in Numerical Methods in Engineering*, vol. 15, no. *10*, pp. 701–708, 1999. DOI: 10.1002/(SICI)1099-0887(199910)15:10<701::AID-CNM283>3.0.CO;2-L. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/%28SICI%291099-0887%28199910%2915%3A10%3C701%3A%3AAID-CNM283%3E3.0.CO%3B2-L. [Online].

Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%291099-0887%28199910%2915%3A10%3C701%3A%3AAID-CNM283%3E3.0.CO%3B2-L.

[44] A. Elhage-Hussein, M. Potier-Ferry, and N. Damil, « A numerical continuation method based on Padé approximants », *International Journal of Solids and Structures*, vol. 37, no. *46*, pp. 6981–7001, 2000, ISSN: 0020-7683. DOI: https://doi.org/10.1016/S0020-7683(99)00323-6. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020768399003236.

[45] E. Mallil, H. Lahmam, N. Damil, and M. Potier-Ferry, « An iterative process based on homotopy and perturbation techniques », *Computer Methods in Applied Mechanics and Engineering*, vol. 190, no. *13*, pp. 1845–1858, 2000, ISSN: 0045-7825. DOI: https://doi.org/10.1016/S0045-7825(00)00198-5. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045782500001985.

[46] B. Malgrange and J.-P. Ramis, « Fonctions multisommables », *Annales de l'Institut Fourier*, vol. 42, p. 353, 1992.

[47] J.-P. Ramis, « Séries divergentes et théories asymptotiques », *Journées X-UPS 1991*, p. 7, 1991.

[48] G. Watson, « The transformation of an asymptotic series into a convergent series of inverse factorials [Memoir crowned by the Danish Royal Academy of Science] », *Rendiconti del Circolo Matematico di Palermo (1884-1940)*, vol. 34, pp. 41–88, 1912.

[49] E. Maillet, « Sur les séries divergentes et les équations différentielles », *Annales scientifiques de l'E.N.S. 3ème série*, vol. 20, p. 487, 1903.

[50] B. Malgrange, « Sur le théorème de Maillet », *Asymptotic Analysis*, vol. 2, pp. 1–4, 1989.

[51] G. Łysik and S. Michalik, « Formal solutions of semilinear heat equations », *Journal of Mathematical Analysis and Applications*, vol. 341, no. *1*, pp. 372–385, 2008, ISSN: 0022-247X. DOI: https://doi.org/10.1016/j.jmaa.2007.10.005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0022247X0701236X.

[52] W. Balser, « Multisummability of formal power series solutions of partial differential equations with constant coefficients », *Journal of Differential Equations*, vol. 201, no. *1*, pp. 63–74, 2004, ISSN: 0022-0396. DOI: https://doi.org/10.1016/j.jde.2004.02.002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0022039604000579.

[53] W. Balser, « Divergent solutions of the heat equation: On an article of Lutz, Miyake and Schäfke », *Pacific Journal of Mathematics*, vol. 188, pp. 53–63, 1999.

[54] O. Costin, G. Luo, and S. Tanveer, « Divergent expansion, Borel summability and three-dimensional Navier-Stokes equation », *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 36, pp. 2775–2778, 2008.

[55] M. MIYAKE, « Newton polygons and formal Gevrey indices in the Cauchy-Goursat-Fuchs type equations », *Journal of the Mathematical Society of Japan*, vol. 43, no. *2*, pp. 305–330, 1991. DOI: 10.2969/jmsj/04320305. [Online]. Available: https://doi.org/10.2969/jmsj/04320305.

[56] Y. Hashimoto and M. Miyake, « Newton polygons and Gevrey indices for linear partial differential operators », *Nagoya Mathematical Journal*, vol. 128, no. *none*, pp. 15–47, 1992. DOI: nmj/1118783301. [Online]. Available: https://doi.org/.

[57] G.-l. Xu and A. Bultheel, « Matrix Padé approximation: definitions and properties », *Linear Algebra and its Applications*, vol. 137-138, pp. 67–136, 1990, ISSN: 0024-3795. DOI: https://doi.org/10.1016/0024-3795(90)90127-X. [Online]. Available: http://www.sciencedirect.com/science/article/pii/002437959090127X.

[58] J. Thomann, « Procédés formels et numériques de sommation de séries solutions d'équations différentielles », *Journées X-UPS 1991*, vol. 91, pp. 101–114, 1991.

[59] M. Kzaz, « Convergence acceleration of the Gauss-Laguerre quadrature formula », *Applied Numerical Mathematics*, vol. 29, no. *2*, pp. 201–220, 1999, ISSN: 0168-9274. DOI: https://doi.org/10.1016/S0168-9274(98)00075-0. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0168927498000750.

[60] J. V. Deun, A. Bultheel, and P. González-Vera, « On computing rational Gauss-Chebyshev quadrature formulas », *Math. Comput.*, vol. 75, pp. 307–326, 2006.

[61] G. Monegato and L. Scuderi, « Quadrature Rules for Unbounded Intervals and Their Application to Integral Equations », *in Approximation and Computation: In Honor of Gradimir V. Milovanović*, W. Gautschi, G. Mastroianni, and T. M. Rassias, Eds. New York, NY: Springer New York, 2011, pp. 185–208, ISBN: 978-1-4419-6594-3. DOI: 10.1007/978-1-4419-6594-3_13. [Online]. Available: https://doi.org/10.1007/978-1-4419-6594-3_13.

[62] R. Kumar and M. K. Jain, « Quadrature Formulas for Semi-Infinite Integrals », *Mathematics of Computation*, vol. 28, no. *126*, pp. 499–503, 1974, ISSN: 00255718, 10886842. [Online]. Available: http://www.jstor.org/stable/2005922.

[63] A. Deeb, A. Hamdouni, and D. Razafindralandy, « Comparison between Borel-Padé summation and factorial series, as time integration methods », *Discrete and Continuous Dynamical Systems - Series S*, vol. 9, no. *2*, pp. 393–408, Apr. 2016. DOI: 10.3934/dcdss.2016003. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02086772.

[64] H. Stahl, « Spurious poles in Padé approximation », *Journal of Computational and Applied Mathematics*, vol. 99, no. *1*, pp. 511–527, 1998, Proceeding of the VIIIth Symposium on Orthogonal Polynomials and Thier Application, ISSN: 0377-0427. DOI: https://doi.org/10.1016/S0377-0427(98)00180-0. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377042798001800.

[65] N. Nielsen, « Sur les séries factorielles et la fonction gamma (extrait d'une lettre adressée à M. N. de Sonin, à Saint-Pétersbourg) », fr, *Annales scientifiques de l'École Normale Supérieure*, vol. 3e série, 23, pp. 145–168, 1906. DOI: 10.24033/asens.564. [Online]. Available: http://www.numdam.org/item/ASENS_1906_3_23__145_0/.

[66] N. Nielsen, « Les séries de factorielles et les opérations fondamentales », *Mathematische Annalen*, vol. 59, p. 355, 1904.

[67] E. Delabaere and J.-M. Rasoamanana, « Sommation effective d'une somme de Borel par séries de factorielles », *Annales de l'Institut Fourier*, vol. 57, no. *2*, pp. 421–456, 2007. DOI: 10.5802/aif.2263. [Online]. Available: http://www.numdam.org/item/AIF_2007__57_2_421_0/.

[68] E.-J. Weniger, « Summation of divergent power series by means of factorial series », *Applied Numerical Mathematics*, vol. 60, no. *12*, pp. 1429–1441, 2010, Approximation and extrapolation of convergent and divergent sequences and series (CIRM, Luminy - France, 2009), ISSN: 0168-9274. DOI: https://doi.org/10.1016/j.apnum.2010.04.003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S016892741000067X.

[69] J. Thomann, « Resommation des series formelles. Solutions d'équations différentielles linéaires ordinaires du second ordre dans le champ complexe au voisinage de singularités irrégulières. », fre, *Numerische Mathematik*, vol. 58, pp. 503–536, 1990. [Online]. Available: http://eudml.org/doc/133514.

[70] N. Nielsen, « Recherches sur les séries de factorielles », *Annales Scientifiques de l'E.N.S. 3è série*, vol. 19, p. 409, 1902.

[71] O. Antipin, A. Maiezza, and J. C. Vasquez, « Resummation in QFT with Meijer G-functions », *Nuclear Physics B*, vol. 941, pp. 72–90, 2019, ISSN: 0550-3213. DOI: https://doi.org/10.1016/j.nuclphysb.2019.02.014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0550321319300422.

[72] *NIST Digital Library of Mathematical Functions*, https://dlmf.nist.gov..

[73] R. Beals and J. Szmigielski, « Meijer G-functions: a gentle introduction », *Notices of the American Mathematical Society*, vol. 60, pp. 866–873, 2013.

[74] H. Bateman, *Higher transcendental functions*. New York Toronto London: McGRAW-HILL BOOK COMPANY, INC, 1953.

[75] A. M. Shalaby, « Precise critical exponents of the $O(N)$-symmetric quantum field model using hypergeometric-Meijer resummation », *Phys. Rev. D*, vol. 101, p. 105 006, 10 2020. DOI: 10.1103/PhysRevD.101.105006. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevD.101.105006.

[76] A. M. Shalaby, « Weak-coupling, strong-coupling and large-order parametrization of the hypergeometric-Meijer approximants », *Results in Physics*, vol. 19, p. 103 376, 2020, ISSN: 2211-3797. DOI: https://doi.org/10.1016/j.rinp.2020.103376. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S221137972031843X.

[77] T. Marios, « Theoretical and Numerical Study of the Van der Pol equation », PhD thesis, Aristotle University of Thessaloniki School of Sciences, Department of Physics, 2006.

[78] W. H. Press and S. A. Teukolsky, « Adaptive Stepsize Runge-Kutta Integration », *Computers in Physics*, vol. 6, no. *2*, p. 188, 1992.

[79] J. Cartwright and O. Piro, « The dynamics of Runge-Kutta methods », *International Journal of Bifurcation and Chaos*, vol. 02, pp. 427–449, 1992.

[80] U. Parlitz and W. Lauterborn, « Period-doubling cascades and devil's staircases of the driven Van der Pol oscillator », *Phys. Rev. A*, vol. 36, pp. 1428–1434, 3 1987. DOI: 10.1103/PhysRevA.36.1428. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.36.1428.

[81] L. A. Hinvi, A. V. Monwanou, and J. B. C. Orou, « A Study of the Forced Van der Pol Generalized Oscillator with the Renormalization Group Method », *Applied Physics Research*, vol. 5, no. *6*, 2013.

[82] R. Zaripov, « Self-organization in the Van der Pol generator for nonextensive systems », *Technical Physics*, vol. 54, pp. 165–169, Feb. 2009. DOI: 10.1134/S1063784209020029.

[83] T. H. Fay, « The forced Van der Pol equation », *International Journal of Mathematical Education in Science and Technology*, vol. 40, no. *5*, pp. 669–677, 2009.

[84] M. Dadfar and J. Geer, « Resonances and Power Series Solutions of the Forced Van der Pol Oscillator », *SIAM Journal on Applied Mathematics*, vol. 50, no. *5*, pp. 1496–1506, 1990. DOI: 10.1137/0150087. eprint: https://doi.org/10.1137/0150087. [Online]. Available: https://doi.org/10.1137/0150087.

[85] J. Guckenheimer, K. Hoffman, and W. Weckesser, « The Forced Van der Pol Equation I: The Slow Flow and Its Bifurcations », *Society for Industrial and Applied Mathematics*, vol. 2, pp. 1–35, Feb. 2003. DOI: 10.1137/S1111111102404738.

[86] M. Cooper, P. Heidlauf, and T. Sands, « Controlling Chaos—Forced Van der Pol Equation », *Mathematics*, vol. 5, p. 70, Nov. 2017. DOI: 10.3390/math5040070.

[87] L. Shampine and M. Reichelt, « The Matlab ODE suite. », *SIAM Journal on Scientific Computing, Society for Industrial and Applied Mathematics*, vol. 18, pp. 1–22, 1997.

[88] S. Abelman and K. Patidar, « Comparison of some recent numerical methods for initial-value problems for stiff ordinary differential equations », *Computers & Mathematics with Applications*, vol. 55, pp. 733–744, 2008. DOI: 10.1016/j.camwa.2007.05.012.

[89] *The Math Works*, http://www.mathworks.com..

[90] J. Dormand and P. Prince, « A family of embedded Runge-Kutta formulae », *Journal of Computational and Applied Mathematics*, vol. 6, no. *1*, pp. 19–26, 1980, ISSN: 0377-0427. DOI: https://doi.org/10.1016/0771-050X(80)90013-3. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0771050X80900133.

[91] J. Liu and X. Wang, « An assessment of the differential quadrature time integration scheme for nonlinear dynamic equations », *Journal of Sound and Vibration*, vol. 314, no. *1*, pp. 246–253, 2008, ISSN: 0022-460X. DOI: https://doi.org/10.1016/j.jsv.2008.01.004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0022460X08000254.

[92] W. Kim, « Higher-order explicit time integration methods for numerical analyses of structural dynamics. », *Latin American Journal of Solids and Structures*, vol. 16, pp. 733–744, Jul. 2019. DOI: https://doi.org/10.1590/1679-78255609.

[93] W. L. Wood and M. E. Oduor, « Stability properties of some algorithms for the solution of nonlinear dynamic vibration equations », *Communications in Applied Numerical Methods*, vol. 4, no. *2*, pp. 205–212, 1988. DOI: 10.1002/cnm.1630040211. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cnm.1630040211. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cnm.1630040211.

[94] E. N. Lorenz, « Deterministic Nonperiodic Flow », *Journal of the Atmospheric Sciences*, vol. 20, no. *2*, pp. 130–141, Mar. 1963, ISSN: 0022-4928. DOI: 10.1175/1520-0469(1963)020<0130:DNF>2.0.CO;2.

[95] E. Belokolos, V. Kharchenko, and D. Kharchenko, « Chaos in a generalized Lorenz system », *Chaos, Solitons & Fractals*, vol. 41, no. *5*, pp. 2595–2605, 2009, ISSN: 0960-0779. DOI: https://doi.org/10.1016/j.chaos.2008.09.049. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0960077908004682.

[96] W. Kim and J. Lee, « An improved explicit time integration method for linear and nonlinear structural dynamics », *Computers & Structures*, vol. 206, Jun. 2018. DOI: 10.1016/j.compstruc.2018.06.005.

[97] T. Fung, « Solving initial value problems by differential quadrature method—part 2: second- and higher-order equations », *Int. J. Numer. Meth. Engng*, vol. 50, pp. 1429–1454, DOI: 10.1002/1097-0207(20010228)50:6<1429::AID-NME79>3.0.CO;2-A.

[98] F. Harlow and J. E. Welch, « Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface », *Physics of Fluids*, vol. 8, pp. 2182–2189, 1965.

[99] *Fenics project*, https://fenicsproject.org.

[100] By, R. Gérard, and Hidetoshi, « Formal Power Series Solutions of Nonlinear First Order Partial Differential Equations », 2005.

[101] A. Shirai, « Maillet type theorem for nonlinear partial differential equations and Newton polygons », *Journal of the Mathematical Society of Japan*, vol. 53, pp. 565–587, Jan. 2001. DOI: 10.2969/jmsj/1213023724.

[102] M. Miyake and Y. Hashimoto, « Newton polygons and gevrey indices for linear partial differential operators », *Nagoya Mathematical Journal*, vol. 128, pp. 15–47, 1992. DOI: 10.1017/S0027763000004207.

[103] A. Mejdi, *Eléments finis pour les fluides incompressibles [Texte imprimé] / Mejdi Azaïez, Michel Deville, Ernest Mund*, fre. Lausanne: Presses polytechniques et universitaires romandes, 2011, ISBN: 978-2-88074-894-4.

[104] https://www.vttoth.com/CMS/technical-notes/67-maple-and-meijers-g-function-a-numerical-instability-and-a-cur.

[105] X. S. Li, « An Overview of SuperLU: Algorithms, Implementation, and User Interface », vol. 31, no. *3*, pp. 302–325, Sep. 2005.

[106] https://gitlab.enpc.fr/jeremy.bleyer/comet-fenics/-/issues/5.

[107] A. Zadeh, H. Zeynal, and K. Nor, « Performance evaluation of SuperLU and PARADISO in power system load flow calculations », *Przeglad Elektrotechniczny*, vol. 87, pp. 290–294, Nov. 2011.

[108] P. C. Jain and D. N. Holla, « Numerical solution of coupled Burgers' equations », *Int. J. Numer. Meth. Eng.*, vol. 12, pp. 213–222, 1978.

[109] A. Bahadır, « A fully implicit finite-difference scheme for two-dimensional Burgers' equations », *Applied Mathematics and Computation*, vol. 137, no. *1*, pp. 131–137, 2003, ISSN: 0096-3003. DOI: https://doi.org/10.1016/S0096-3003(02)00091-7. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0096300302000917.

[110] V. K. Srivastava, M. K. Awasthi, and S. Singh, « An implicit logarithmic finite-difference technique for two dimensional coupled viscous Burgers' equation », *AIP Advances*, vol. 3, no. *12*, p. 122 105, Dec. 2013. DOI: 10.1063/1.4842595.

[111] S. S. V. K. Srivastava and M. K. Awasthi, « Numerical solutions of coupled Burgers' equations by an implicit finite-difference scheme », *AIP Advances*, vol. 3, p. 082 131, 2013.

[112] M. Tamsir, V. Srivastava, and J. Kumar, « Numerical Solution of two dimensional coupled viscous Burgers Equation using the Modified Cubic B Spline Differential Quadrature Method », *AIP Advances*, vol. 4, pp. 117 134–10, Sep. 2014. DOI: 10.1063/1.4902507.

[113] A. J. Chorin., « Numerical solution of the Navier-Stokes equations », *Math. Comp.*, vol. 22, pp. 745–762, 1968.

[114] R. Temam, « Sur l'approximation de la solution des équations de Navier-Stokes », *Arch. Rational Mech. Anal.*, vol. 32, pp. 377–385, 1969.

[115] https://mfix.netl.doe.gov/doc/vvuq-manual/main/html/fluid/fld-01.html.

[116] R. Reyes, « Stabilized reduced order models in low speed flows », PhD thesis, Escola Tècnica Superior d'Enginyers de Camins, Universitat Politècnica de Catalunya, 2020.

[117] W. Hu, N. Trask, X. Hu, and W. Pan, « A spatially adaptive high-order meshless method for fluid–structure interactions », *Computer Methods in Applied Mechanics and Engineering*, vol. 355, pp. 67–93, 2019, ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2019.06.009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045782519303494.

[118] S. R. Bhopalam and D. A. Perumal, « Numerical analysis of fluid flows in L-Shaped cavities using Lattice Boltzmann method », *Applications in Engineering Science*, vol. 3, p. 100 016, 2020, ISSN: 2666-4968. DOI: https://doi.org/10.1016/j.apples.2020.100016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666496820300169.

[119] F.-Y. Hsu, M. Jolly, and J. Campbell, « The design of L-shaped runners for gravity casting », Jan. 2007.

[120] Y. Saad and M. H. Schultz, « GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems », *Siam Journal on Scientific and Statistical Computing*, vol. 7, pp. 856–869, 1986.

[121] U. Ghia, K. Ghia, and C. Shin, « High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method », *Journal of Computational Physics*, vol. 48, pp. 387–411, 1982.

[122] T. A. AbdelMigid, K. M. Saqr, M. A. Kotb, and A. A. Aboelfarag, « Revisiting the lid-driven cavity flow problem: Review and new steady state benchmarking results using GPU accelerated code », *Alexandria Engineering Journal*, vol. 56, no. *1*, pp. 123–135, 2017, ISSN: 1110-0168. DOI: https://doi.org/10.1016/j.aej.2016.09.013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1110016816302800.

[123] T. Huang and H. Lim, « Simulation of Lid-Driven Cavity Flow with Internal Circular Obstacles », *Applied Sciences*, vol. 10, p. 4583, Jul. 2020. DOI: 10.3390/app10134583.

[124] R. Samelson, R. Temam, C. Wang, and S. Wang, « Surface Pressure Poisson Equation Formulation of the Primitive Equations: Numerical Schemes », *SIAM J. Numerical Analysis*, vol. 41, pp. 1163–1194, Jan. 2003. DOI: 10.1137/S0036142901396284.

[125] E. Wahba, « Multiplicity of states for two-sided and four-sided lid driven cavity flows », *Computers & Fluids*, vol. 38, no. *2*, pp. 247–253, 2009, ISSN: 0045-7930. DOI: https://doi.org/10.1016/j.compfluid.2008.02.001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045793008000303.

[126] D. Arumuga Perumal and A. K. Dass, « Multiplicity of steady solutions in two-dimensional lid-driven cavity flows by Lattice Boltzmann Method », *Computers & Mathematics with Applications*, vol. 61, no. *12*, pp. 3711–3721, 2011, Mesoscopic Methods for Engineering and Science — Proceedings of ICMMES-09, ISSN: 0898-1221. DOI: https://doi.org/10.1016/j.camwa.2010.03.053. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0898122110002427.

[127] S. O. Unverdi and G. Tryggvason, « A front-tracking method for viscous, incompressible, multi-fluid flows », *Journal of Computational Physics*, vol. 100, no. *1*, pp. 25–37, 1992, ISSN: 0021-9991. DOI: https://doi.org/10.1016/0021-9991(92)90307-K. [Online]. Available: http://www.sciencedirect.com/science/article/pii/002199919290307K.

[128] Y. Chang, T. Hou, B. Merriman, and S. Osher, « A Level Set Formulation of Eulerian Interface Capturing Methods for Incompressible Fluid Flows », *Journal of Computational Physics*, vol. 124, no. *2*, pp. 449–464, 1996, ISSN: 0021-9991. DOI: https://doi.org/10.1006/jcph.1996.0072. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999196900728.

[129] S. Osher and J. A. Sethian, « Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations », *Journal of Computational Physics*, vol. 79, no. *1*, pp. 12–49, 1988, ISSN: 0021-9991. DOI: https://doi.org/10.1016/0021-9991(88)90002-2. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0021999188900022.

[130] M. Raessi, J. Mostaghimi, and M. Bussmann, « A volume-of-fluid interfacial flow solver with advected normals », *Computers & Fluids*, vol. 39, no. *8*, pp. 1401–1410, 2010, ISSN: 0045-7930. DOI: https://doi.org/10.1016/j.compfluid.2010.04.010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045793010000861.

[131] R. Scardovelli and S. Zaleski, « Direct numerical simulation of free-surface and interfacial flow », *Annual Review of Fluid Mechanics*, vol. 31, no. *1*, pp. 567–603, 1999. DOI: 10.1146/annurev.fluid.31.1.567.

[132] S. Osher and R. Fedkiw, *Level set methods and dynamic implicit surfaces*. New York: Springer-Verlag, 2003, vol. 153, ISBN: 0-387-95482-1.

[133] J. A. Sethian, *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*. Cambridge University Press, 1999, vol. 3.

[134] S.-R. Hysing and S. Turek, « The Eikonal equation: Numerical efficiency vs. algorithmic complexity on quadrilateral grids », *Proceedings of Algoritmy*, vol. 35, pp. 22–31, Jan. 2005.

[135] N. Johansson, « Implementation of a standard level set method for incompressible two-phase flow simulations », Master's thesis, Uppsala University, Division of Scientific Computing, 2011, p. 30.

[136] M. Sussman, P. Smereka, and S. Osher, « A Level Set approach for Computing Solutions to Incompressible Two-Phase Flow », *Journal of Computational Physics*, vol. 114, no. *1*, pp. 146–159, 1994, ISSN: 0021-9991. DOI: https://doi.org/10.1006/jcph.1994.1155. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999184711557.

[137] S. Tanguy, « Développement d'une méthode de suivi d'interface. Applications aux écoulements diphasiques », PhD thesis, Université de Rouen, CORIA, 2004.

[138] X.-D. Liu, S. Osher, and T. Chan, « Weighted Essentially Non-oscillatory Schemes », *Journal of Computational Physics*, vol. 115, no. *1*, pp. 200–212, 1994, ISSN: 0021-9991. DOI: https://doi.org/10.1006/jcph.1994.1187. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999184711879.

[139] G.-S. Jiang and C.-W. Shu, « Efficient Implementation of Weighted ENO Schemes », *Journal of Computational Physics*, vol. 126, no. *1*, pp. 202–228, 1996, ISSN: 0021-9991. DOI: https://doi.org/10.1006/jcph.1996.0130. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999196901308.

[140] J. B. Bell, P. Colella, and H. M. Glaz, « A second-order projection method for the incompressible navier-stokes equations », *Journal of Computational Physics*, vol. 85, no. *2*, pp. 257–283, 1989, ISSN: 0021-9991. DOI: https://doi.org/10.1016/0021-9991(89)90151-4. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0021999189901514.

[141] C.-W. Shu and S. Osher, « Efficient implementation of essentially non-oscillatory shock-capturing schemes », *Journal of Computational Physics*, vol. 77, no. *2*, pp. 439–471, 1988, ISSN: 0021-9991. DOI: https://doi.org/10.1016/0021-9991(88)90177-5. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0021999188901775.

[142] M. Yousefzadeh and I. Battiato, « A Level-Set Immersed Boundary Method for Reactive Transport in Complex Topologies with Moving Interfaces. », *arXiv: Fluid Dynamics*, 2020.

[143] S. Gottlieb and C.-W. Shu, « Total Variation Diminishing Runge-Kutta Schemes », *Mathematics of Computation*, vol. 67, Aug. 1996. DOI: 10.1090/S0025-5718-98-00913-2.

[144] W. Rider and D. Kothe, « Stretching and Tearing Interface Tracking Methods », *12th Computational Fluid Dynamics Conference*, vol. 95-17171, Jun. 1995.

[145] J. Lakhlili, « Modélisation et simulation numériques de l'érosion par méthode DDFV », PhD thesis, Université de Toulon, 2015.

[146] F. Couderc, « Développement d'un code de calcul pour la simulation d'écoulements de fluides non miscibles. Application à la désintégration assistée d'un jet liquide par un courant gazeux », PhD thesis, ENSAE, Toulouse, 2007.

[147] R. J. LeVeque, « High-resolution conservative algorithms for advection in incompressible flow », *SIAM Journal on Numerical Analysis*, vol. 33, pp. 627–665, 1996.

DOCTORAT BRETAGNE SCIENCES LOIRE POUR L'INGENIEUR

Université Bretagne Sud

Titre : Étude numérique de méthodes d'intégration temporelle basées sur des techniques de perturbation. Application à des problèmes de mécanique.

Mot clés : Perturbation temporelle, Séries divergentes, Borel Padé Laplace, Séries factorielles inversées, Meijer-G approximant, Méthode asymptotique numérique, Équations de Navier-Stokes, Level-set.

Résumé : Le développement des méthodes numériques pour la résolution des problèmes d'évolution non linéaires est actuellement un domaine de recherche en pleine expansion. Par conséquent, l'objectif principal de cette thèse est d'étudier et développer les méthodes de perturbation temporelle et de sommation numérique de séries divergentes utilisées comme schémas d'intégration temporelle. Nous nous intéressons à la méthode asymptotique numérique, à la sommation de Borel Padé Laplace, des séries factorielles inversées et des approximants de Meijer-G. La propriété la plus intéressante de ces approches est que les solutions obtenues sont continues en temps. Ces méthodes sont appliquées aux équations différentielles ordinaires et aux équations aux dérivées partielles de la mécanique. Celles-ci comprennent l'équation de la chaleur, de Burgers, l'équation de suivi d'interface et les équations de Navier-Stokes. Ces approches se sont avérées efficaces sur de longs intervalles de temps et pour de nombreux problèmes. Ainsi, les tests numériques montrent qu'il est possible de réduire le temps de calcul des méthodes classiques pour certains problèmes tout en éliminant les contraintes d'intégration et les faiblesses liées aux schémas classiques. Pour d'autres problèmes tels que les équations de Navier-Stokes, nous avons remarqué que dans certains cas une optimisation doit être proposée pour améliorer ces approches. De plus, un nouvel éclairage est apporté sur l'utilisation des approximants de Meijer-G sur des problèmes dépendant du temps. Les différents développements liés à cette méthode ont été validés et laissent entrevoir des perspectives intéressantes.

Title: Numerical study of time integration methods based on perturbation techniques. Application to mechanical problems.

Keywords: Time perturbation, Divergent series, Borel-Padé Laplace, Inverse factorial series, Meijer-G approximants, Asymptotic numerical method, Navier-Stokes equations, Level-set.

Abstract: The development of numerical methods for solving nonlinear evolution problems is currently a growing research field. Therefore, the main goal of this thesis is to address several needs regarding the development of time perturbation methods and numerical summation of divergent series used as time integration schemes. We are interested in the Asymptotic Numerical Method, the Borel Padé Laplace summation, the Inverse factorial series, and the Meijer-G approximant. The most interesting property of these approaches is that the obtained solutions are continuous in time. These methods are applied to the ordinary and partial differential equations in mechanics. These include the heat equation, Burgers, the interface tracking equation and the Navier-Stokes equations. These approaches have shown their efficiency over long time intervals on a variety of problems. Thus, numerical tests show that it is possible to reduce the computation time of classical methods for some problems while eliminating the constraints related to the classical scheme. For other problems such as the Navier-Stokes equations, we have noticed that in some cases an optimization must be proposed to improve these approaches. Moreover, a new insight is provided on the use of the Meijer-G approximants on time-dependent problems. Different developments related to this method have been validated and show interesting perspectives.