

DOTTORATO DI RICERCA
IN
SCIENZE COMPUTAZIONALI E INFORMATICHE
Ciclo XVIII

Consorzio tra Università di Catania, Università di Napoli Federico II,
Seconda Università di Napoli, Università di Palermo, Università di Salerno

SEDE AMMINISTRATIVA:
UNIVERSITÀ DI NAPOLI FEDERICO II



IDA DEL PRETE

**EFFICIENT NUMERICAL METHODS
FOR VOLTERRA INTEGRAL EQUATIONS
OF HAMMERSTEIN TYPE**

TESI DI DOTTORATO DI RICERCA

To my son

Acknowledgments

First, I wish to sincerely thank Prof. Elvira Russo for introducing me to numerical analysis and for providing me this interesting Ph.D. subject. She helped me immensely in these years by providing me with the necessary work environment and giving me the possibility to participate in several conferences and schools. She always helped me with questions and problems related to numerical analysis and last but not least for cross-reading this thesis.

A special thank to Prof. Maria Rosaria Crisci, who left us soon. Some of the ideas of this thesis came from her. I miss our fruitful discussions and her passion for numerical analysis.

I wish to thank especially my colleague Dr. Dajana Conte. In the last two years we studied very hard and together we succeed to prove the main results of this thesis. It was a very good experience to work together and I hope that our collaboration will continue.

Dr. Giovanni Capobianco for his collaboration in this work and for cross-reading this thesis.

I would like to thank also all the members of my research group, in particular Prof. Beatrice Paternoster and Dr. Eleonora Messina.

I also would like to thank Prof. Christian Lubich for suggesting many helpful ideas and for inviting me and Dr. Dajana Conte to the University of Tuebingen.

In addition I wish to thank all my colleague of the Ph. D. course and the Ph. D. students of Salerno.

The last, but the most important thanks to my husband Raffaele who supported me with patience and love and to my parents for supporting my studies, helping me during difficult times and for providing moral support.

Contents

Introduction	V
1 Theoretical background	1
1.1 Introduction	2
1.2 Volterra Integral equations	3
1.3 Models of Volterra integral equations	5
1.4 Review of basic theory for VIEs	
with smooth kernel	11
1.4.1 Linear VIEs	11
1.4.2 Linear convolution equations	12
1.4.3 Nonlinear VIEs	13
1.5 Review of basic theory for VIEs	
with weakly singular kernel	15
1.5.1 Linear VIEs	15
1.5.2 Nonlinear VIEs	16
2 Numerical methods for Volterra integral equations	18
2.1 Introduction	19
2.2 Collocation methods	21
2.2.1 Convergence results for smooth kernel	24

2.2.2	Convergence results for weakly singular kernel	26
2.2.3	Discretized collocation methods and related convergence results	27
2.3	Volterra Runge-Kutta methods	31
2.3.1	Extended VRK methods of Pouzet type	31
2.3.2	Modified VRK methods of de Hoog and Weiss	33
2.3.3	Convergence results	34
3	The Laplace transform and its numerical inversion	36
3.1	Introduction	37
3.2	The Laplace transform	39
3.3	The inverse Laplace transform	42
3.4	Methods for numerical inversion	43
3.4.1	Talbot's method	43
3.4.2	Modified Talbot's method (Rizzardi)	46
3.4.3	Modified Talbot's method (Lubich-Schädle)	47
3.5	Fast convolution quadrature formulas	54
3.5.1	Approximation of definite integrals	54
3.5.2	The fast convolution algorithm	56
4	Fast Collocation methods for Volterra Integral equations of convolution type	63
4.1	Introduction	64
4.2	Fast collocation methods	66
4.2.1	Fast computation of the lag terms	67
4.2.2	Computation of the increment term	71
4.2.3	Determination of the approximate solution	72
4.3	Convergence analysis	75

4.4	Numerical results	79
4.4.1	Convergence	80
4.4.2	Computational cost	83
4.5	Concluding remarks	87
5	Fast Runge–Kutta methods for Volterra Integral equations of convolution type	88
5.1	Introduction	89
5.2	Fast VRK methods	91
5.2.1	Fast computation of the lag terms	92
5.2.2	Determination of the approximate solution	95
5.3	Computational cost	98
5.4	Convergence analysis	100
5.5	Stability analysis	103
5.5.1	Basic test equation	103
5.5.2	Convolution test equation	108
5.6	Numerical results	118
5.6.1	Convergence	119
5.6.2	Computational cost	121
5.7	Concluding remarks	123
6	On the asymptotic periodicity of the solutions of nonlinear discrete Volterra equations of convolution type	124
6.1	Introduction	125
6.2	Representation of analytical solution	126
6.3	DVEs with asymptotically periodic solution	129
6.4	Asymptotical periodicity for Hammerstein type DVEs	130
6.5	Applications	133

<i>Contents</i>	IV
6.6 A class of AP-stable methods	137
6.6.1 AP-stability of the exact collocation method	139
6.6.2 AP-stability of the discretized collocation method	140
6.7 Concluding remarks	142
Bibliography	143

Introduction

Volterra integral equations (VIEs) are the mathematical model of many evolutionary problems with memory arising from biology, chemistry, physics, engineering. For example they arise from population dynamics, epidemic diffusion, viscoelasticity, neurophysiology, feedback control theory, study of the behaviour of nuclear reactors and from the treatment of special hyperbolic differential equations. Section 1.3 contains an high number of references on these applications. It is known that the numerical treatment of VIEs has an high computational cost, due mainly to the computation of the “lag term” or “tail term” which contains the past history of the phenomenon. Since it depends on time t , the “lag term” has to be computed for each time step and its cost increases when time passes. Among the Volterra equations, the Hammerstein type ones, are particularly interesting for the applications and several authors investigated on the construction of stable, accurate and efficient numerical methods for these kind of equations, see for example [27], [52] and their references.

The aim of this thesis is the construction of numerical methods for VIEs of Hammerstein type which produce accurate solution at a low computational cost and “catch” the qualitative behaviour of the exact solution.

The study developed has been concerned at first with the construction and analysis of efficient methods for the numerical treatment of VIEs of Hammerstein type

$$y(t) = f(t) + \int_0^t k(t - \tau)g(y(\tau))d\tau \quad t \in [0, T], \quad (1)$$

where the Laplace transform of the kernel rather than the convolution kernel itself is a priori known. This is not an anomalous or restricting situation, as a matter of fact these kind of problems arise in chemical absorption kinetics [69], in the determination of non reflecting boundary conditions [56], [68], and in general in situations when Laplace transform technique are used to reduce systems of ordinary or partial differential equations in VIEs.

It is known that a classical numerical method for computing the numerical solution of (1) over N_t time steps requires $O(N_t^2)$ operations and $O(N_t)$ memory space (see [18]). In [52] and [68] fast algorithms to compute the solution of (1) have been proposed for the first time. More precisely in [52] a 4-stage Runge Kutta method for VIE of order $p = 4$ was developed by using Fast Fourier Transform technique in order to get a computational cost of $O(N_t(\log N_t)^2)$. Afterwards in [68] a quadrature formula was constructed for computing evolutionary integrals of convolution type; this formula can be used as a direct quadrature method for VIEs with a computational cost of $O(N_t \log N_t)$ operations and $O(\log N_t)$ memory requirement, however the order of accuracy comes out to be $p = 2$, and hence, quite low.

In this thesis we construct two classes of fast numerical methods for the equation (1) based on collocation and Runge–Kutta formulas respectively. These methods have an high order of accuracy and they have again a computational cost of $O(N_t \log N_t)$ operations and $O(\log N_t)$ memory requirement. In both cases the knowledge of the Laplace transform of the kernel and the

convolution nature of the kernel itself are exploited in order to obtain a fast computation of the lag term.

This is possible by using an opportune inverse Laplace transform approximation formula which is described in [68] for computing the kernel evaluations. In this thesis such formula has been inserted into “classical” collocation and Volterra Runge–Kutta (VRK) methods and, in the new resulting methods the same computational scheme proposed in [68] has been used for the lag-term. The inverse Laplace transform approximation formula results from applying the trapezoidal rule to a parametrization of the contour integral for the Riemann inverse Laplace transform formula, where the integration contour is an opportune Talbot curve ([82] [86]). The error of such formula decreases exponentially with the number M of the quadrature nodes, uniformly on $[0, T]$, and it depends on the distance of the singularities of the Laplace transform of the function that has to be inverted, to the Talbot contour.

The fast numerical methods constructed in this thesis tend to the corresponding classical methods when the inverse Laplace transform approximation formula is exact, that is when $M \rightarrow \infty$. The convergence analysis of the fast collocation and fast Runge–Kutta methods shows that their order of convergence depends on M , and it is proved that such order coincides with the order of the corresponding classical methods when M is suitably chosen.

We also analyse the stability properties of the fast Runge–Kutta methods with respect to the convolution test equation

$$y(t) = 1 + \int_0^t [\mu + \sigma(t - \tau)]y(\tau)d\tau \quad t \in [0, T], \quad \mu, \sigma \in \mathbb{R}^-. \quad (2)$$

This equation, that misses of course of some typical feature of VIEs, is generally used by several authors to test the stability properties of numerical methods for VIEs, see for example [6], [28]. In this thesis we prove that the stability regions depend on the number of the points M chosen for the approximation

of the inverse Laplace transform and that for $M \rightarrow \infty$ the stability properties of the classical VRK methods are obtained.

The numerical experiments on some significant problems taken from the “Test Set” collection project confirm the expected accuracy, computational cost and the stability properties of the constructed methods. From the experiments it comes out that, since the error of the inverse Laplace transform approximation formula decreases exponentially with M , relatively few points on the Talbot contour produce the expected order and stability region.

This research is carried out in collaboration with Dajana Conte (Ph.d student, University of Salerno).

The second part of the thesis is concerned with the numerical treatment of problems of *SIS* epidemic diffusion with periodic immigration flow [29]. The mathematical model of such problems is represented by an Hammerstein type VIE with convolution kernel of the form

$$y(t) = f(t) + q(t) + \int_0^t a(s)k(t-s)y(s)(1-y(s))ds. \quad (3)$$

We consider here problems characterized by the relapse of the epidemic which implies that the VIE (3) has an asymptotically periodic solution.

It is clear that an efficient numerical method has to reproduce the asymptotically periodic solution whenever applied to equation (3).

For this reason we analyse the discrete Volterra equation (DVE) corresponding to problem (3) and we prove a theorem which establishes the existence and the uniqueness of the asymptotically periodic solution of the DVE.

Moreover we consider SIS epidemic models with periodic immigration flow and constant contact rate ($a(s) = a = \text{const.}$). In this case we prove, for the DVE corresponding to the problem, the existence and the uniqueness of the asymptotically periodic solution when the DVE satisfies some significant

hypothesis depending only on its kernel and forcing term.

In order to analyse if the existing numerical methods satisfy these conditions, that is if they are AP-stable, we consider the class of θ -methods and we prove that they are AP-stable if the integration step satisfies an inequality depending only on some characteristics of the problem.

The thesis consists of six chapters.

In the first chapter we report the main theoretical results about the existence, uniqueness and smoothness of the solution of a VIE. This study is the basis for the numerical treatment of VIEs.

The second chapter is a brief survey of numerical methods for VIEs with particular attention to collocation methods and Volterra Runge-Kutta methods and it represents an introduction to the problems related to the numerical treatment of VIEs.

The third chapter is a short introduction to the Laplace transform and its inverse and to some existing numerical methods for the inverse Laplace transform approximation. Moreover it is illustrated how these methods have been used for the construction of a fast convolution quadrature formula for the computation of evolutionary integrals of convolution type in which the Laplace transform of the kernel is known a priori ([56], [68]). This fast quadrature formula represents the basic idea for the construction of the fast numerical methods for VIEs illustrated in Chapters 4 and 5

In the fourth chapter we give the detailed construction of the fast collocation methods. In particular we describe how we obtain the fast computation of the lag terms and we give a detailed analysis of the convergence and of the computational cost of the constructed methods.

The fifth chapter is concerned with the construction of the fast Runge-Kutta methods. We describe the computation of the lag terms and we give

a detailed analysis of the convergence, of the computational cost and of the stability analysis of the constructed methods.

For both classes of fast methods we propose several numerical experiments in order to validate the theoretical results proved.

In the sixth chapter we prove some theorems for the existence and the uniqueness of the asymptotically periodic solution of DVEs of Hammerstein type. Moreover we introduce the basic concepts of asymptotic periodicity for DVEs of Hammerstein type and we investigate on the AP-stability of the class of θ -methods.

The thesis was developed within two GNCS research projects: “Metodi numerici e software matematico per problemi di evoluzione” supervised by Prof. M. Zennaro; “Metodi innovativi per problemi evolutivi con memoria”, supervised by Prof. E. Russo and within one research project of Regione Campania: “Metodi numerici ad alte prestazioni per la risoluzione di problemi di diffusione di epidemie modellizzati da equazioni integrali di Volterra” supervised by Prof. E. Russo.

Chapter 1

Theoretical background

1.1 Introduction

An integral equation is a functional equation in which the unknown function appears under one or several integral signs. The integral equations are the mathematical model of many problems arising from biology, chemistry, physics and engineering (see Section 1.3 and the related bibliography).

In this thesis we focus our attention on Volterra Integral Equations (VIEs), characterized by a variable upper limit of integration.

Even if we are interested in the numerical solution of VIEs and in the problems related to their numerical treatment, as it will be well explained in the following chapters, we can not ignore the properties of the exact solution, since the numerical solution of a VIE must reproduce the behaviour of the exact solution.

Thus the aim of this chapter is to report the main theoretical results about the existence, uniqueness and smoothness of the solution of a VIE.

1.2 Volterra Integral equations

Let $I := [0, T]$ denote a given closed and bounded interval, with $T > 0$, and set $S := \{(t, s) | 0 \leq s \leq t \leq T\}$. The more general VIE (for the unknown function y) is of the form

$$\theta(t)y(t) = f(t) + \int_0^t k(t, s, y(s))ds \quad t \in I \quad (1.2.1)$$

$$\theta, f, y : t \in I \rightarrow \mathbb{R}, \quad k : S \times \mathbb{R} \rightarrow \mathbb{R}$$

where the function f is referred to as *forcing function* and k is called the *kernel* of the integral equation.

The function $\theta(t)$ determines the classification of VIEs in:

- **first kind VIEs.** If $\theta(t) = 0 \quad \forall t \in I$. The equation (1.2.1) becomes:

$$f(t) = - \int_0^t k(t, s, y(s))ds, \quad t \in I.$$

- **second kind VIEs.** If $\theta(t) \neq 0 \quad \forall t \in I$. The equation (1.2.1) can be led to the form:

$$y(t) = f(t) + \int_0^t k(t, s, y(s))ds \quad t \in I. \quad (1.2.2)$$

- **third kind VIEs.** If θ is a continuous function possessing a finite number of zeros in the interval I .

In this thesis we focus our attention on second kind VIEs.

A VIE is said to be *linear* if its kernel has the form

$$k(t, s, y) = \bar{k}(t, s)y \quad \forall t, s, y, \quad (1.2.3)$$

of *convolution* type if

$$k(t, s, y) = \bar{k}(t - s, y) \quad \forall t, s, y, \quad (1.2.4)$$

of *Hammerstein* type if

$$k(t, s, y) := \bar{k}(t, s)g(s, y(s)), \quad (1.2.5)$$

weakly singular (or of *Abel* type) if

$$k(t, s, y) = (t - s)^{-\alpha}\gamma(t, s, y), \quad 0 < \alpha < 1, \quad (1.2.6)$$

where γ is a smooth function in $S \times \mathbb{R}$.

1.3 Models of Volterra integral equations

Volterra integral equations arise in great many branches of science like physics, biology, chemistry and engineering. They are particularly suitable to describe evolutionary phenomena with memory and this feature makes the theoretical study and the numerical treatment complicated, as it will be showed in the following chapters.

We will not present a full description of VIEs arising in practical applications, but we refer to the literature and we will analyze in more detail only the problems which will be taken into consideration throughout the thesis.

The following books and survey papers contain sections with various applications of VIEs in the physical and biological sciences: Schmeidler (1950), Bellman and Cooke (1963), Anselone (1964), Miller (1971), Brunner (1982), Burton (1983), Webb(1985), Okrasinski (1989), Corduneanu (1991), Guy and Salès (1991), Prüss (1993), Agarwal and O'Regan (2000), Cordunenanu and Sandberg (2000), Zhao (2003). Most of these also include extensive lists of references.

As regards the specific applications of second kind VIEs, they are for example models of

- population dynamics and spread of epidemics: Brauer (1975, 1976), Diekmann (1978, 1979), Thieme (1977, 1979), Gripenberg (1981), Brauer and Castillo-Chávez (2001).
- renewal equations: Feller (1941), Karlin (1955), Bellman and Cooke (1963), Brauer (1976), Deligonoul and Bilgen (1984).
- reaction-diffusion in small cells: Dixon (1987).

- wave problems: Levinson (1960) (superfluidity), Gilding (1993) (traveling wave analysis in nonlinear reaction-convection-diffusion problems), Kabanichin and Lorenzi (1999) (identification problems for wave phenomena), Franco (1999) (nonlinear waves).
- water percolation: Okrański (1978).
- semi-conductor devices: Miller and Unterreiter (1992), Schmeiser, Unterreiter and Weiss (1993), Unterreiter (1996) (models for switching behaviour of PN-diodes).
- inverse problems related to wave propagation: Kabanichin and Lorenzi (1999).
- identification of memory kernels in viscoelasticity and heat conduction: Wolfersdorf (1994), Unger and Wolfersdorf (1995), Janno and Wolfersdorf (1997), Kiss (1999), Berrone (1995) (modelling of materials that may undergo a change of phase).
- Heat transfer problems: this is one of the major sources of VIEs with weakly singular kernels. See for example the papers by Mann and Wolf (1951), Roberts and Mann (1951), Keller and Olmstead (1972), Olmstead and Handelsman (1976), Norbury and Stuart (1987), Groetsch (1989, 1991), Jumarhon (1994), Jumarhon and McKee (1996), Ibrahim and Alnasr (1997).
- Gas absorption: Olmstead (1977).

Many of the evolutionary problems described above (e.g. population dynamics, epidemic diffusion, renewal equations) do not depend on the precise

instant t , but on the whole passed interval, thus the VIEs of *convolution* type (1.2.4) are very representative in the applications. Often the convolution kernel is of *Hammerstein* type (1.2.5) (see e.g. [16] and its references), and it may be continuous or weakly singular. Examples can be found in neurophysiology [2], epidemiology [54],[29], feedback control theory [25], study of the behaviour of nuclear reactors. VIEs of this kind also arise the treatment of special partial differential equations, see for example the problems regarding an elastic beam being hit by a sphere [75], the reflection of sound pulses by convex parabolic reflectors [45], the problem of determining the temperature in a nonlinearly radiating semi-infinite solid [64]. Infact for solving partial differential equations with several independent variables it is often preferable to recast the given problem as an integral equation with fewer independent variables.

Moreover, in many applications, as for example in chemical absorption kinetics [69], in the determination of non reflecting boundary conditions [56], [68], and in general in situations when Laplace transform technique are used to reduce systems of ordinary or partial differential equations in VIEs, the equation is of Hammerstein type

$$y(t) = f(t) + \int_0^t k(t - \tau)g(y(\tau))d\tau \quad t \in [0, T], \quad (1.3.1)$$

where the Laplace transform $K(s)$ of the kernel $k(t)$ rather than the convolution kernel itself is known a priori.

Example 1.1. CHEMICAL ABSORPTION KINETICS. The following example arises in chemical absorption kinetics (see [69]) and leads to an integral equation of the form (1.3.1), where only the Laplace transform of the kernel is known. The problem is to find $y(t)$ given by the coupled system of ordinary differential equation and the diffusion equation

$$\frac{dy}{dt}(t) = -\alpha \frac{\partial u}{\partial r}(1, t), \quad y(0) = y_0, \quad (1.3.2)$$

where $u(r, t)$ satisfies

$$\frac{\partial u}{\partial t} = \beta \left(\frac{\partial^2 u}{\partial r^2} + \frac{2}{r} \frac{\partial u}{\partial r} \right) \quad \text{in } 0 < r < 1, \quad t > 0, \quad (1.3.3)$$

with nonlinear boundary conditions (b smooth)

$$\begin{aligned} u(1, t) &= b(y(t)), \quad t > 0, \\ \frac{\partial u}{\partial r}(0, t) &= 0, \quad t > 0, \end{aligned} \quad (1.3.4)$$

and initial condition

$$u(r, 0) = 0, \quad 0 < r < 1. \quad (1.3.5)$$

Here $u(r, t)$ represents a concentration profile in a spherical absorbing particle, and $y(t)$ is the concentration in the surrounding aqueous solution.

The above system can be reduced to a single Volterra integral equation by the following (standard) arguments: taking Laplace transforms in (1.3.3)-(1.3.5) leads to the boundary value problem

$$sU = \beta \left(U'' + \frac{2}{r} U' \right), \quad U'(0, s) = 0, \quad U(1, s) = B(s) \quad \left(' = \frac{\partial}{\partial r} \right)$$

which can be solved analytically for U . In particular, one obtains

$$U'(1, s) = \frac{s}{\beta} K(s) B(s) \quad (1.3.6)$$

with

$$K(s) = \frac{1}{\sqrt{s/\beta} \tanh \sqrt{s/\beta}} - \frac{1}{s/\beta}. \quad (1.3.7)$$

By (1.3.2) the Laplace transform of $y(t)$ satisfies

$$sY(s) - y_0 = -\alpha U'(1, s).$$

Inserting (1.3.6) and applying the inverse Laplace transform gives the (weakly singular) Volterra integral equation

$$y(t) = y_0 - \frac{\alpha}{\beta} \int_0^t k(t - \tau) b(y(\tau)) d\tau, \quad t > 0,$$

where the laplace transform $K(s)$ of the kernel $k(t)$ is known from (1.3.7), rather than the kernel itself.

Example 1.2. AN SIS EPIDEMIC MODEL (CROMER)

This model was proposed by Cromer [29] and it is made of a scalar VIE. The disadvantage of this model is that it ignores the different responses to the diseases exhibited by males and females, thus a system of VIEs would have been more suitable.

SIS models are for diseases which confer no immunity and have a negligible incubation period. In such models there are only two classes of individuals: susceptibles, denoted as S and infectives, denoted as I . A susceptible moves from class S to class I upon being infected, and then to class S when cured.

$$S \longrightarrow I \longrightarrow S$$

To derive the model it is necessary to define the following functions:

$N(t)$ is the total population at time t .

$I(t)$ is the fraction of population infected at time t .

$M(t)$ is the fraction of infective immigrants.

$S(t)$ is the fraction of susceptibles at time t .

$P(t)$ is the probability that an individual not have recovering t units of time after contracting the disease.

$a(t)$ is the contact rate at time t .

$f(t)$ is the fraction of those initially infected who have not recovered by time t .

It is assumed that $a(t)S(t)$ is the number of effective contacts per infective per unit time, thus $N(t)I(t)a(t)S(t)$ is the total number of effective contacts at time t . At time t the total infective population is $N(t)I(t)$ for which we

have the equation

$$N(t)I(t) = N(t)f(t) + N(t)M(t) + \int_0^t a(s)N(s)I(s)S(s)P(t-s)ds.$$

The first term on the right is the total number of initially infected individuals who have not recovered. The second term is the total number of infective immigrants. The integral term gives the number of infectives produced in the population. If we divide through by $N(t)$ and use the fact that $I(t) + S(t) = 1$ we obtain the equation

$$I(t) = f(t) + M(t) + \int_0^t a(s)P(t-s)\frac{N(s)}{N(t)}I(s)(1-I(s))ds.$$

In his work Cromer assumes that $N(t) = N$ is a constant thus he studies the equation

$$I(t) = f(t) + M(t) + \int_0^t a(s)P(t-s)I(s)(1-I(s))ds. \quad (1.3.8)$$

Cromer shows that if the influx of infective immigrants is periodic and the contact rate $a(s)$ is constant the solution of the equation (1.3.8) is asymptotically periodic, the same hold if $a(s)$ is assumed periodic.

The Volterra integral equation SIS model studied by Cromer contains as special cases the SIS model studied by Hethcote [54] and by Cooke and Kaplan [22]. The study of causes of the oscillations in this model is very representative since this phenomenon is of interest also in models of population growth, harvesting and economics (see [13], [12], [22]).

1.4 Review of basic theory for VIEs with smooth kernel

It is the purpose of this section to present, without proofs, the basic quantitative theory of VIEs of the second kind with smooth kernel, distinguishing the cases of linear kernel, linear convolution kernel and nonlinear kernel. For a more detailed analysis see [15], [18].

1.4.1 Linear VIEs

Consider the linear VIE

$$y(t) = f(t) + \int_0^t k(t, s)y(s)ds \quad t \in I \quad (1.4.1)$$

and assume that the kernel k and the forcing function f are real-valued and continuous on S and I respectively.

The solution of the equation (1.4.1) is expressed in terms of the *resolvent kernel*, defined as follows:

Definition 1.1. Let $k \in C(S)$. Then the (unique) resolvent kernel $R = R(t, s)$ corresponding to the given kernel k in the linear VIE (1.4.1) is defined by either of the resolvent equations:

$$R(t, s) = k(t, s) + \int_s^t k(t, \tau)R(\tau, s)d\tau \quad (t, s) \in S \quad (1.4.2)$$

$$R(t, s) = k(t, s) + \int_s^t R(t, \tau)k(\tau, s)d\tau \quad (t, s) \in S. \quad (1.4.3)$$

It can be shown that the resolvent kernel R is the sum of the so called Neumann series:

$$R(t, s) = \sum_{n=1}^{\infty} k_n(t, s)$$

where k_n are the *iterated kernels* defined as

$$k_1(t, s) = k(t, s) \quad k_n(t, s) = \int_s^t k(t, \tau)k_{n-1}(\tau, s)d\tau \quad (t, s) \in S.$$

The following theorem, due to Vito Volterra in his Nota I [91], establishes the existence and uniqueness of solution to the linear VIE (1.4.1).

Theorem 1.1. *Let $k \in C(S)$, and let R denote the resolvent kernel associated with k . Then for any $f \in C(I)$ the second kind VIE (1.4.1) has a unique solution $y \in C(I)$, and this solution is given by*

$$y(t) = f(t) + \int_0^t R(t, s)f(s)ds, \quad t \in I. \quad (1.4.4)$$

Remark 1.1. The Theorem 1.1 can be straightforwardly extended to the case of a system of linear VIEs.

1.4.2 Linear convolution equations

A linear VIE with convolution kernel, often encountered in applications, reads:

$$y(t) = f(t) + \int_0^t k(t-s)y(s)ds \quad t \in I. \quad (1.4.5)$$

The resolvent kernel corresponding to (1.4.5) inherits the convolution structure of the kernel and assumes the form $R(t, s) =: \rho(t-s)$. The resolvent equations (1.4.2) and (1.4.3) become

$$\rho(z) = k(z) + \int_z^0 k(z-\tau)\rho(\tau)d\tau \quad z \in I \quad (1.4.6)$$

and

$$\rho(z) = k(z) + \int_z^0 \rho(z-\tau)k(\tau)d\tau \quad z \in I \quad (1.4.7)$$

respectively, with $z := t-s$, and Theorem 1.1 for (1.4.5) may be restated as

Theorem 1.2. *Let $k \in C(I)$. Then for any $f \in C(I)$ the convolution VIE (1.4.5) possesses a unique solution $y \in C(I)$ which is given by*

$$y(t) = f(t) + \int_0^t \rho(t-s)f(s)ds, \quad t \in I, \quad (1.4.8)$$

where the resolvent kernel ρ is defined by the equation (1.4.6) or (1.4.7).

1.4.3 Nonlinear VIEs

In case of the general second kind integral equation (1.2.2) we report two (global and local) existence and uniqueness theorems under the assumption that the kernel k satisfies a uniform or punctual Lipschitz condition.

Theorem 1.3. *Let $k(t, s, y)$ be continuous for all $(t, s) \in S$ and all y , and suppose that k satisfy the uniform Lipschitz condition,*

$$|k(t, s, y_1) - k(t, s, y_2)| \leq L|y_1 - y_2|, \quad \forall (t, s) \in S, \forall y_1, y_2 \in \mathbb{R},$$

with Lipschitz constant L being independent of y_1, y_2 . Then for each $f \in C(I)$ the nonlinear equation (1.2.2) possesses a unique solution $y \in C(I)$.

In many cases, however, the condition imposed on the kernel function $k(t, s, y)$ will not be satisfied for all $y \in \mathbb{R}$, but will hold only in some compact region. In this case the solution is not guaranteed on the whole interval $[0, T]$, as showed in the following result [73]:

Theorem 1.4. *Let $f \in C(I)$ and assume that $k(t, s, y)$ is continuous in the region*

$$\Omega := \{(t, s, y) | (t, s) \in S, \quad |y - f(t)| \leq B\}.$$

In addition let k satisfy a Lipschitz condition of the form

$$|k(t, s, y_1) - k(t, s, y_2)| \leq L|y_1 - y_2|, \quad \forall (t, s, y_1), (t, s, y_2) \in \Omega.$$

and set

$$M := \max\{|k(t, s, y)| \mid (t, s, y) \in \Omega\} \quad e \quad T_0 := \min\{T, B/M\},$$

Then the nonlinear equation (1.2.2) possesses a unique continuous solution y in $[0, T_0]$.

A result of the above type is called a *local* existence theorem since it guarantees the existence of a solution to (1.2.2) only on a sufficiently small interval $[0, T_0]$. However it is possible to construct a continuation of the solution beyond T_0 . How far the solution can be continued will depend on the constants B and M . Details concerning this question can be found in [19], [73].

As discussed in Section 1.3, often in the applications the nonlinear equation (1.2.2) is of *Hammerstein* type

$$y(t) = f(t) + \int_0^t k(t, s)g(s, y(s))ds \quad t \in I. \quad (1.4.9)$$

Here $g : I \times \mathbb{R} \rightarrow \mathbb{R}$ is smooth, while the kernel function k (often of convolution type) may be continuous (bounded) or weakly singular. General results on the existence of solutions to Hammerstein integral equations (1.4.9) can also be found in [14] and [41].

1.5 Review of basic theory for VIEs with weakly singular kernel

The more general weakly singular VIE or of *Abel* type reads

$$y(t) = \int_0^t (t-s)^{-\alpha} k(t, s, y(s)) ds + f(t), \quad t \in I \quad (1.5.1)$$

$$0 < \alpha < 1,$$

$$f, y : t \in I \rightarrow \mathbb{R}, \quad k : S \times \mathbb{R} \rightarrow \mathbb{R}, \quad S = \{(t, s) | 0 \leq s \leq t \leq T\},$$

where k is supposed to be continuous in its domain of definition. The results presented in this section may be found in [15], [18].

1.5.1 Linear VIEs

When the equation (1.5.1) is linear, that is of the form

$$y(t) = f(t) + \int_0^t (t-s)^{-\alpha} k(t, s) y(s) ds, \quad t \in I \quad (1.5.2)$$

it is possible to express the solution of (1.5.2) depending on the *resolvent* kernel, as for linear VIEs with smooth kernel, according to the following theorem

Theorem 1.5. *Assume that the function f and k are continuous on I and S , respectively and $\alpha \in (0, 1)$. Then the equation (1.5.2) has a unique solution $y \in C(I)$ given by*

$$y(t) = f(t) + \int_0^t R(t, s; \alpha) f(s) ds, \quad t \in I \quad (1.5.3)$$

where the resolvent kernel $R(t, s; \alpha)$ has the form

$$R(t, s; \alpha) = (t-s)^{-\alpha} Q(t, s; \alpha), \quad (t, s) \in S$$

with $Q(t, s; \alpha) \in C(S)$ for each $\alpha \in (0, 1)$. Furthermore Q satisfies the identities

$$\begin{aligned} Q(t, s; \alpha) &= (t-s)^\alpha \int_s^t (t-\tau)^{-\alpha} (\tau-s)^{-\alpha} k(t, \tau) Q(\tau, s; \alpha) d\tau + k(t, s) = \\ &= (t-s)^\alpha \int_s^t (t-\tau)^{-\alpha} (\tau-s)^{-\alpha} Q(t, \tau; \alpha) k(\tau, s) d\tau + k(t, s), \end{aligned}$$

for all $(t, s) \in S$.

1.5.2 Nonlinear VIEs

Since the singular term $(t-s)^{-\alpha}$ in (1.5.1) is integrable, it can be shown in a straightforward way that the existence and uniqueness result of Theorem 1.4 remains valid. However the number T_0 defining the existence interval now depends on α .

Now we report some relevant result on the smoothness of the exact solution of a VIE of Abel type since this information is important when analyzing the order of convergence of a numerical method.

Theorem 1.6. *Let m be a nonnegative integer, let α be a real number with $0 < \alpha < 1$ and suppose that the real-valued function f is analytic in a neighborhood of I . Let $h(t, y)$ be real-valued and analytic on an open set containing $I \times \mathbb{R}$. Then the solution of*

$$y(t) = f(t) + \int_0^t (t-s)^{m-\alpha} h(s, y(s)) ds, \quad t \in I$$

a) *is real analytic in a neighborhood of the left-open interval $(0, T]$;*

b) *if α is a rational number, written in lowest terms, $\alpha = p/q$, then $y(t^q)$ is analytic in a neighborhood of $t = 0$.*

Theorem 1.7. *Consider the equation*

$$y(t) = f_1(t) + t^{1/2}f_2(t) \int_0^t (t-s)^{-1/2}k(t,s,y(s))ds, \quad t \in I \quad (1.5.4)$$

and assume that

- (i) $f_i \in C^m([0,t]), i = 1, 2$;
- (ii) $k(t,s,y)$ is continuous with respect to $(t,s) \in S$ and satisfies a (uniform) Lipschitz condition with respect to y in \mathbb{R} ;
- (iii) $k(t,s,y)$ has continuous partial derivatives of order m with respect to t and s and of order $2m$ with respect to y , for all $(t,s) \in S$ and all y in some open neighborhood of $y(t)$.

Then the solution of (1.5.4) has the form

$$y(t) = v(t) + t^{1/2}w(t), \quad t \in I, \quad v, w \in C^m(I) \quad (1.5.5)$$

where (v,w) is the solution of the following system of VIEs:

$$\begin{aligned} v(t) &= f_1(t) + \int_0^t p_1(t,s)k_1(t,s,v(s),w(s))ds \\ w(t) &= f_2(t) + \int_0^t p_2(t,s)k_2(t,s,v(s),w(s))ds \end{aligned} \quad t \in I$$

with

$$\begin{aligned} p_1(t,s) &= (s/(t-s))^{1/2}, \quad p_2(t,s) = t^{-1/2}, \\ k_1(t,s,v,w) &= (k(t,s,v+s^{1/2}w) - k(t,s,v-s^{1/2}w))/(2s^{1/2}), \\ k_2(t,s,v,w) &= (k(t,s,v+s^{1/2}w) + k(t,s,v-s^{1/2}w))/2. \end{aligned}$$

An analogous result for arbitrary $\alpha \in (0,1)$ may be found in [70].

Chapter 2

Numerical methods for Volterra integral equations

2.1 Introduction

Volterra integral equations are particularly suitable to describe evolutionary phenomena that remember their past history thanks to the presence of a memory term in the integral operator. In fact the solution of a VIE at time t depends on the solution on the whole interval $[0, t]$. This important feature makes difficult the theoretical study of such equations and only in some special case it is possible to know the analytical solution of a VIE. In all other cases if one is interested in the knowledge of the qualitative or quantitative behaviour of the solution, a numerical method has to be used to approximate the analytical solution.

In this chapter, for seek of completeness, we illustrate collocation and Runge-Kutta methods for VIEs and their connections, since they represent the most important classes of numerical methods for VIEs. Moreover they represent the basis for the construction of the new methods developed throughout the thesis. Both classes of methods are based on a discretization of the given interval $I = [0, T]$ by a uniform mesh

$$I_h = \{t_n := nh, n = 0, \dots, N_t, h \geq 0, N_t h = T\}.$$

The integral equation (1.2.2) can be rewritten, by relating it to this mesh, as

$$y(t) = F_n(t) + \int_{t_n}^t k(t, s, y(s)) ds \quad t \in (t_n, t_{n+1}], \quad (2.1.1)$$

where

$$F_n(t) := f(t) + \int_0^{t_n} k(t, s, y(s)) ds \quad (2.1.2)$$

and

$$\Phi_n(t) = \int_{t_n}^t k(t, s, y(s)) ds \quad (2.1.3)$$

represent respectively the “lag term” and the “increment function”. The “lag term” or “tail term” contains the past history of the phenomenon. Since it depends on time t , it has to be computed for each time step and its cost increases when time passes. Thus the “lag term” computation makes the numerical integration of VIEs very expensive in terms of computational cost. This problem will be well analyzed in the following chapters since one of the aim of this thesis is to look for accurate and efficient numerical methods for solving VIEs.

2.2 Collocation methods

A collocation method is based on the idea of approximating the exact solution of a given integral equation with a suitable function belonging to a chosen finite dimensional space such that the approximated solution satisfies the integral equation on a certain subset of the interval on which the equation has to be solved (called the set of collocation points). Here we consider as the approximating space the polynomial spline space of degree $m - 1$ and continuity class $d = -1$ on the set I_h that is we look for an approximated solution u belonging to nonsmooth piecewise polynomial space

$$S_{m-1}^{(-1)}(I_h) = \{u : u(t)|_{t \in (t_n, t_{n+1}]} =: u_n(t) \in \pi_{m-1}, n = 0, \dots, N_t - 1\}, \quad (2.2.1)$$

where π_{m-1} denotes the space of polynomials of degree not exceeding $m - 1$. Let us fix m collocation parameters $0 \leq c_1 < \dots < c_m \leq 1$ and let $t_{n,i} := \{t_n + c_i h, n = 0, \dots, N_t - 1, i = 1, \dots, m\}$ be the collocation points. Setting $Y_{n,i} := u_n(t_{n,i})$, the restriction of u to each subinterval $(t_n, t_{n+1}]$ can be written as:

$$u_n(t_n + \nu h) = \sum_{j=1}^m L_j(\nu) Y_{n,j} \quad \nu \in (0, 1] \quad n = 0, \dots, N_t - 1 \quad (2.2.2)$$

where $L_j(\nu)$ is the j -th Lagrange fundamental polynomial with respect to the collocation parameters c_1, \dots, c_m . The m unknowns $Y_{n,j}$ are the solution of the linear system

$$Y_{n,i} = \bar{F}_n(t_{n,i}) + \int_{t_n}^{t_{n,i}} k(t_{n,i}, s, u_n(s)) ds \quad i = 1, \dots, m \quad (2.2.3)$$

obtained by imposing that u satisfies exactly the integral equation (2.1.1) at the collocation points $t_{n,i}$.

Here

$$\bar{F}_n(t) = f(t) + \int_0^{t_n} k(t, s, u(s)) ds \quad (2.2.4)$$

represents the approximation to the exact lag term (2.1.2).

By considering the change of variable $s = t_n + \nu h$ the formulas (2.2.3) and (2.2.4) can be equivalently written as

$$Y_{n,i} = \bar{F}_n(t_{n,i}) + h \int_0^{c_i} k(t_{n,i}, t_n + \nu h, u_n(t_n + \nu h)) d\nu \quad i = 1, \dots, m \quad (2.2.5)$$

and

$$\bar{F}_n(t) = f(t) + h \sum_{k=0}^{n-1} \int_0^1 k(t, t_k + \nu h, u_k(t_k + \nu h)) d\nu \quad (2.2.6)$$

When the equation is linear, i.e. of the form (1.4.1), the formulas (2.2.5) and (2.2.6), after employing the local representation (2.2.2), can be written as

$$Y_{n,i} = \bar{F}_n(t_{n,i}) + h \sum_{j=1}^m \left(\int_0^{c_i} k(t_{n,i}, t_n + \nu h) L_j(\nu) d\nu \right) Y_{n,j} \quad i = 1, \dots, m \quad (2.2.7)$$

and

$$\bar{F}_n(t_{n,i}) = f(t_{n,i}) + h \sum_{k=0}^{n-1} \sum_{j=1}^m \left(\int_0^1 k(t_{n,i}, t_k + \nu h) L_j(\nu) d\nu \right) Y_{k,j}. \quad (2.2.8)$$

Let $\bar{\mathbf{Y}}_{\mathbf{n}} := (Y_{n,1}, \dots, Y_{n,m})^{\mathbf{T}}$, $\bar{\mathbf{F}}_{\mathbf{n}} := (\bar{F}_n(t_{n,1}), \dots, \bar{F}_n(t_{n,m}))^{\mathbf{T}}$ and define the matrix

$$\mathbf{B}_n^{(k)} := \begin{pmatrix} \int_0^1 k(t_{n,i}, t_k + \nu h) L_j(\nu) d\nu & 0 \leq k < n \leq N_t - 1 \\ \int_0^{c_i} k(t_{n,i}, t_n + \nu h) L_j(\nu) d\nu & k = n \end{pmatrix}. \quad (2.2.9)$$

The system (2.2.7) then assumes the form

$$(\mathbf{I} - h\mathbf{B}_n^{(n)}) \bar{\mathbf{Y}}_{\mathbf{n}} = \bar{\mathbf{F}}_{\mathbf{n}} \quad n = 0, \dots, N_t - 1, \quad (2.2.10)$$

where \mathbf{I} denotes the identity matrix of dimension m and

$$\bar{\mathbf{F}}_{\mathbf{n}} = h \sum_{k=0}^{n-1} \mathbf{B}_n^{(k)} \bar{\mathbf{Y}}_{\mathbf{k}}. \quad (2.2.11)$$

Theorem 2.1. *Assume that f and k in (1.4.1) are continuous on their respective domains. Then there exists $\bar{h} > 0$ so that for any mesh I_h with stepsize*

$h \in (0, \bar{h})$ each of the linear algebraic system (2.2.10) has a unique solution \bar{Y}_n ($n = 0, \dots, N_t - 1$). Hence the equations (2.2.7) and (2.2.8) define a unique collocation solution $u \in S_{m-1}^{(-1)}(I_h)$ with local representation given by (2.2.2).

Remark 2.1. In the computational use of piecewise collocation methods the value of m does usually not exceed $m = 4$. Hence the obvious candidate for the local representation of the collocation solution on $(t_n, t_{n+1}]$ is the local Lagrange basis corresponding to the $\{c_i\}$.

Remark 2.2. Note that for $m \geq 2$ the choice $c_1 = 0$ and $c_m = 1$ leads to a continuous approximation

$$u \in S_{m-1}^{(-1)}(I_h) \cap C(I) = S_{m-1}^{(0)}(I_h) \quad (2.2.12)$$

Remark 2.3 (Computational cost). For each fixed time step t_n , the lag terms computation (2.2.6) or (2.2.8) requires a sum involving all the previous time steps, leading to a computational cost proportional to n operations. On the other hand the computation of the solution of the nonlinear system (2.2.5) or of the linear system (2.2.7) requires a number of operations depending only on m . As m is fixed and independent of N_t (generally in concrete applications it is $m \leq 4$, as observed in Remark 2.1), the total cost for computing the numerical solution over N_t time steps is of $O(N_t^2)$ operations and $O(N_t)$ memory.

Example 2.1. Approximation in $S_0^{(-1)}(I_h)$.

Here we have $m = 1$, $0 < c_1 =: \theta \leq 1$, and $u_n(t_n + \nu h) = Y_{n,1}$, for all $\nu \in (0, 1]$. Setting $y_{n+1} := Y_{n,1}$ the collocation solution is determined by the equation

$$\left(y_{n+1} - h \int_0^\theta k(t_{n,1}, t_n + \nu h, y_{n+1}) d\nu \right) = \bar{F}_n(t_{n,1})$$

with $n = 0, \dots, N_t - 1$, $t_{n,1} = t_n + \theta h$ and with the lag term given by

$$\bar{F}_n(t_{n,1}) = f(t_{n,1}) + h \sum_{k=0}^{n-1} \int_0^1 k(t_{n,1}, t_k + \nu h, y_{k+1}) d\nu.$$

Example 2.2. Approximation in $S_1^{(-1)}(I_h)$.

Here we have $m = 2$, $0 \leq c_1 < c_2 \leq 1$, and

$$u_n(t_n + \nu h) = \frac{1}{c_2 - c_1} [(c_2 - \nu)Y_{n,1} + (\nu - c_1)Y_{n,2}].$$

with $Y_{n,j} = u_n(t_{n,j})$ $j = 1, 2$. The system (2.2.3) becomes

$$\begin{aligned} Y_{n,1} &= \bar{F}_n(t_{n,1}) + h \int_0^{c_1} k(t_{n,1}, t_n + \nu h, u_n(t_n + \nu h)) d\nu \\ Y_{n,2} &= \bar{F}_n(t_{n,2}) + h \int_0^{c_2} k(t_{n,2}, t_n + \nu h, u_n(t_n + \nu h)) d\nu. \end{aligned} \quad (2.2.13)$$

where

$$\begin{aligned} \bar{F}_n(t_{n,1}) &= f(t_{n,1}) + h \sum_{k=0}^{n-1} \int_0^1 k(t_{n,1}, t_k + \nu h, u_k(t_k + \nu h)) d\nu \\ \bar{F}_n(t_{n,2}) &= f(t_{n,2}) + h \sum_{k=0}^{n-1} \int_0^1 k(t_{n,2}, t_k + \nu h, u_k(t_k + \nu h)) d\nu. \end{aligned} \quad (2.2.14)$$

In particular if $c_2 = 1$, then $u_n(t_{n+1}) = Y_{n,2}$. Furthermore for the particular choice of $c_1 = 0$ and $c_2 = 1$, we generate a continuous piecewise approximation such that $Y_{n,1} = u_n(t_n) = u_{n-1}(t_n)$.

2.2.1 Convergence results for smooth kernel

Let $e = y - u$, denote its restriction to the interval $(t_n, t_{n+1}]$ by e_n and

$$\|e\|_\infty := \sup \{|e_n(t)| : t \in (t_n, t_{n+1}], n = 0, \dots, N_t - 1\}.$$

The following theorem establishes the global convergence order of a collocation method.

Theorem 2.2. *Suppose that the function f and k are m times continuously differentiable on their domain of definition and $k_y(t, s, \cdot)$ is continuous and bounded. Then the error e satisfies, for every choice of the collocation parameters $0 \leq c_1 < \dots < c_m \leq 1$,*

$$\|e\|_\infty \leq Ch^m \quad (2.2.15)$$

where the finite constant C is independent of h but depending on the $\{c_i\}$ and $\|y^m\|_\infty$.

This theorem shows that if we choose a collocation approximation $u \in S_{m-1}^{(-1)}(I_h)$ the global order of convergence $p = m$ is the best attainable. It is possible to increase this order of convergence only in the mesh points with a particular choice of the collocation parameters according to the following local superconvergence theorem.

Theorem 2.3. *Let $u \in S_{m-1}^{(-1)}(I_h)$, $f, k \in C^{2m-v}$, with $v \in \{0, 1, 2\}$ and $m \geq \lfloor \nu/2 \rfloor + 1$.*

(a) *If the collocation parameters $\{c_i\}$ are the Radau II points for $(0, 1]$, then for $v = 1$,*

$$\max_{t_n \in I_h - \{0\}} |e(t_n)| = O(h^{2m-1})$$

(b) *If the collocation parameters $\{c_i\}$ are the Lobatto points for $[0, 1]$, then for $v = 2$,*

$$\max_{t_n \in I_h - \{0\}} |e(t_n)| = O(h^{2m-2}).$$

(c) *If the collocation parameters $\{c_i\}$ are the Gauss points for $(0, 1)$, then for $v = 0$*

$$\max_{t_n \in I_h - \{0\}} |e(t_n)| = O(h^m),$$

in other words Gauss collocation does not lead to superconvergence.

(d) If the collocation parameters $\{c_i\}$ are the $m - 1$ Gauss points for $(0, 1)$, $c_m = 1$, then for $v = 2$

$$\max_{t_n \in I_h - \{0\}} |e(t_n)| = O(h^{2m-2}).$$

Remark 2.4. Local order $O(h^{2m})$ can be attained on $I_h - \{0\}$ by using iterated collocation and employing Gauss points as collocation parameters (see [15], [18]).

2.2.2 Convergence results for weakly singular kernel

The Theorem 2.2 remains valid if the kernel $k(t, s, y)$ is of the form $(t - s)^{-\alpha} k(t, s) y(s)$ with $0 < \alpha < 1$ and $k \in C(S)$ and if the exact solution y is in $C^m(I)$; the nonhomogeneous term f need not to be smooth. However, as we have seen in Section 1.5, such a situation represents an exception rather than the rule. In general smooth f and k now yield to a solution whose first derivatives near $t = 0$ behaves like $y'(t) \sim t^{-\alpha}$. As a consequence if we employ a uniform mesh the global order of convergence of the collocation approximation drops to $p = 1 - \alpha$, regardless of how we choose its degree $m - 1$, as stated by the following result [18] for the equation (1.5.2).

Theorem 2.4. *Let the function f and k in (1.5.2) belong to $C^m(I)$ and $C^m(S)$, respectively, with $m \geq 1$, and assume that neither function vanishes identically. Then there exists an $\bar{h} > 0$ such that the error of the the collocation approximation satisfies, for every choice of the collocation parameters $\{c_i\}$*

$$\|e\|_{\infty} = O(h^{1-\alpha}).$$

In order to restore the optimal order of convergence the underlying mesh has to be graded so to replace the structure of $y(t)$ near the origin. If we

still want to employ a uniform mesh, then we must consider nonpolynomial spline collocation spaces. For more details on these two possible alternative approaches for dealing with this order reduction problem see [15], [18].

2.2.3 Discretized collocation methods and related convergence results

In the previous sections we assumed that the integrals occurring in (2.2.5) and (2.2.6) can be evaluated analitically. Since in the applications this is rarely possible it is important to establish how to approximate these integrals.

By choosing a suitable quadrature formula and by disregarding the quadrature error terms, the equations (2.2.5) and (2.2.6) become

$$\bar{Y}_{n,i} = \bar{F}_{n,i} + h \sum_{l=1}^{\mu_0} w_{i,l} k(t_{n,i}, t_n + d_{i,l}h, \bar{u}_n(t_n + d_{i,l}h)) \quad i = 1, \dots, m \quad (2.2.16)$$

$$\bar{F}_{n,i} = f(t_{n,i}) + h \sum_{k=0}^{n-1} \sum_{l=1}^{\mu_1} w_l k(t_{n,i}, t_k + d_l h, \bar{u}_k(t_k + d_l h)) \quad i = 1, \dots, m. \quad (2.2.17)$$

Here μ_0 and μ_1 are positive integers. Generally the quadrature formulas chosen are interpolatory ones, with the quadrature parameters $d_{i,l}$ and d_l satisfying $0 \leq d_{i,1} < \dots < d_{i,\mu_0} \leq c_i$ and $0 \leq d_1 < \dots < d_{\mu_1} \leq 1$. The quadrature weights are then given by

$$w_l = \int_0^1 \prod_{\substack{r=1 \\ r \neq l}}^{\mu_1} \frac{v - d_r}{d_l - d_r} dv, \quad l = 1, \dots, \mu_1 \quad (2.2.18)$$

and

$$w_{i,l} = \int_0^{c_i} \prod_{\substack{r=1 \\ r \neq l}}^{\mu_0} \frac{v - d_{i,r}}{d_{i,l} - d_{i,r}} dv, \quad l = 1, \dots, \mu_0 \quad i = 1, \dots, m. \quad (2.2.19)$$

The formulas (2.2.16) and (2.2.17) yield an approximation $\bar{u} \in S_{m-1}^{(-1)}(I_h)$ which is different from u and whose local representation is

$$\bar{u}_n(t_n + \nu h) = \sum_{j=1}^m L_j(\nu) \bar{Y}_{n,j} \quad \nu \in (0, 1] \quad n = 0, \dots, N_t - 1, \quad (2.2.20)$$

with $\bar{Y}_{n,j} := \bar{u}_n(t_{n,j})$. The error of this approximation is given by

$$|\bar{e}(t)| = |y(t) - \bar{u}(t)| \leq |y(t) - u(t)| + |u(t) - \bar{u}(t)|.$$

Setting $e(t) = y(t) - u(t)$ and $\epsilon(t) = u(t) - \bar{u}(t)$, by theorem 2.2 we have $|e(t)| \leq Ch^m$ uniformly on I . The order of $\epsilon(t)$ will depend on the choice the quadrature formulas according to the following theorem

Theorem 2.5. *Let f and k be at least m times continuously differentiable on I and S respectively, and $u \in S_{m-1}^{(-1)}(I_h)$ denote the approximation to the solution y defined by (2.2.3) and (2.2.4). Assume to discretize the integrals with quadrature formulas satisfying*

$$\int_0^1 \phi(t_j + \tau h) d\tau - \sum_{l=1}^{\mu_1} w_l \phi(t_j + d_l h) = O(h^{r_1}),$$

and for $i = 1, \dots, m$

$$\int_0^{c_i} \phi(t_n + \tau h) d\tau - \sum_{l=1}^{\mu_0} w_{i,l} \phi(t_n + d_{i,l} h) = O(h^{r_0}),$$

whenever the integrand is a sufficiently smooth function. If $\bar{u} \in S_{m-1}^{(-1)}(I_h)$ is the approximation given by the discrete collocation equations (2.2.16) and (2.2.17), then there exists a finite constant Q such that ϵ satisfies

$$\|\epsilon\|_{\infty} \leq Qh^r \quad r = \min\{r_0 + 1, r_1\}.$$

The proof of this theorem makes use of the following result concerning the discrete Gronwall-type inequalities, which we report because we will need it in later applications:

Theorem 2.6. *Let $z_n \geq 0$ for $n = 0, \dots, N$ and suppose that the sequence z_n obeys the inequality*

$$z_n \leq hC_1 \sum_{i=0}^{n-1} z_i + C_2, \quad n = k, \dots, N, \quad (2.2.21)$$

where $k > 0$, $C_i > 0$, $i = 1, 2$, and $h > 0$. Then the elements of this sequence can be bounded by

$$z_n \leq (hC_1 z + C_2)(1 + hC_1)^{n-k}, \quad n = k, \dots, N,$$

provided the starting values z_0, \dots, z_{k-1} satisfy $z_j \leq z/k$.

The following result is an immediate consequence of the Theorem 2.5.

Corollary 2.1. *Let the assumptions of Theorem 2.5 hold. If the quadrature formulas chosen to discretize the integrals are of interpolatory type with $\mu_0 = \mu_1 = m$, then the approximation $\bar{u} \in S_{m-1}^{(-1)}(I_h)$ defined by (2.2.16) and (2.2.17) leads to an error $\bar{e}(t)$ satisfying*

$$\|\bar{e}\|_\infty = O(h^m)$$

for every choice of collocation parameters $\{c_i\}$.

Now we will deal with the question of local superconvergence in approximation $\bar{u} \in S_{m-1}^{(-1)}(I_h)$.

Theorem 2.7. *Let $\bar{u} \in S_{m-1}^{(-1)}(I_h)$, $f, k \in C^{2m-v}$, with $v \in \{1, 2\}$, and with $m \geq \lfloor \nu/2 \rfloor + 1$.*

(a) *If the collocation parameters $\{c_i\}$ are the Radau II points for $(0, 1]$ and the quadrature formulas are of interpolatory type with $\mu_0 = \mu_1 = m$, $d_l = c_l$, $d_{i,l} = c_i c_l$ then for $v = 1$,*

$$\max_{t_n \in I_h - \{0\}} |\bar{e}(t_n)| = O(h^{2m-1}).$$

(b) If the collocation parameters $\{c_i\}$ are the Lobatto points for $[0, 1]$, if the quadrature formulas are in the form described in (a) then for $v = 2$,

$$\max_{t_n \in I_h - \{0\}} |\bar{e}(t_n)| = O(h^{2m-2}).$$

(c) Let the first $m - 1$ collocation parameters $\{c_i\}$ are the Gauss points for $(0, 1)$ and let $c_m = 1$. Suppose that the quadrature formulas are interpolatory $m - 1$ -point formulas with $\mu_0 = \mu_1 = m - 1$, $d_l = c_l$, $d_{i,l} = c_i c_l$ ($l = 1, \dots, m - 1$, $i = 1, \dots, m$) then for $v = 2$

$$\max_{t_n \in I_h - \{0\}} |\bar{e}_n| = O(h^{2m-2}).$$

The following illustration is the discrete counterpart of the exact collocation method described in Example 2.1.

Example 2.3. Approximation in $S_0^{(-1)}(I_h)$.

Here we have $m = 1$, $0 < c_1 =: \theta \leq 1$. Setting $\bar{y}_{n+1} := \bar{u}_n(t_n + \nu h) = \bar{Y}_{n,1}$ the collocation solution is determined by the equation

$$(\bar{y}_{n+1} - \theta h k(t_{n,1}, t_n + \theta^2 h, \bar{y}_{n+1})) = \bar{F}_{n,1}$$

with $n = 0, \dots, N_t - 1$, $t_{n,1} = t_n + \theta h$ and with the lag term given by

$$\bar{F}_{n,1} = f(t_{n,1}) + h \sum_{k=0}^{n-1} k(t_{n,1}, t_k + \theta h, \bar{y}_{k+1}).$$

2.3 Volterra Runge-Kutta methods

Runge-Kutta methods for the numerical solution of Volterra integral equations were introduced by Pouzet and Bel'tyukov about the mid-1960s.

A Volterra Runge-Kutta (VRK) method is based on an approximation scheme for the increment function (2.1) that will be called a Volterra Runge-Kutta (VRK) formula and denoted by $\bar{\Phi}_n(t)$, and on an approximation scheme, $\bar{F}_n(t)$, for the lag term (2.1.2) that will be called lag term formula.

The approximation of the equation in the mesh point t_{n+1} leads to the discrete method of the form:

$$y_{n+1} = \bar{F}_n(t_n + h) + \bar{\Phi}_n(t_n + h) \quad n = 0, \dots, N_t - 1. \quad (2.3.1)$$

As concern this thesis we will consider *extended* VRK methods of Pouzet type and the modified methods of *de Hoog and Weiss*.

2.3.1 Extended VRK methods of Pouzet type

An *extended* VRK method of Pouzet type (PVRK method) uses an m -stage Pouzet VRK formula (PVRK formula), which has the form

$$\bar{\Phi}_n(t) = h \sum_{i=1}^m b_i k(t, t_{n,i}, Y_{n,i}) \quad (2.3.2)$$

with

$$Y_{n,i} = \bar{F}_n(t_{n,i}) + h \sum_{s=1}^m a_{i,s} k(t_n + c_i h, t_n + c_s h, Y_{n,s}) \quad i = 1, \dots, m. \quad (2.3.3)$$

Here, the vectors $c = (c_i)$, $b = (b_i)$ and the square matrix $A = (a_{i,s})$ are completely determined by the "Butcher array" for ODEs

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array}$$

and $t_{n,i} := t_n + c_i h$. The lag term formula is given by

$$\bar{F}_n(t) = f(t) + h \sum_{r=0}^{n-1} \sum_{s=1}^m b_s k(t, t_{r,s}, Y_{r,s}). \quad (2.3.4)$$

The following steps describe how to compute the approximate solution y_{n+1} :

- evaluate the lag terms $\bar{F}_n(t_{n,i})$ $i = 1, \dots, m+1$, using the expression (2.3.4), having set $c_{m+1} = 1$;
- compute the increment term $\bar{\Phi}_n(t_n+h)$ after solving the nonlinear system (2.3.3) for $Y_{n,i}$ $i = 1, \dots, m$;
- determine the approximate solution y_{n+1} through the expression (2.3.1).

A PVRK formula is

- (i) *explicit* if $a_{i,s} = 0$ for $s \geq i$
- (ii) *diagonally implicit* if $a_{i,s} = 0$ for $s > i$
- (iii) *implicit* if it is neither explicit or diagonally implicit.

Remark 2.5. Observe that an extended m -stages PVRK method with Butcher

array $\begin{array}{c|c} c & A \\ \hline & b^T \end{array}$ can be seen as a discrete collocation method (2.2.16)-(2.2.17)

using $m+1$ collocation parameters with $\{c_i\}_{i=1}^m$ given by the Butcher array and $c_{m+1} = 1$. The quadrature formulas involved are not necessarily interpolatory ones, as the number of nodes is $\mu_0 = \mu_1 = m$, the nodes are $d_{il} = c_l$ for $i = 1, \dots, m+1$, $l = 1, \dots, m$, $d_l = c_l$ for $l = 1, \dots, m$, and the weights are $w_{il} = a_{il}$ for $i = 1, \dots, m$, $l = 1, \dots, m$, $w_{m+1,l} = b_l$ for $l = 1, \dots, m$, $w_l = b_l$ for $l = 1, \dots, m$.

2.3.2 Modified VRK methods of de Hoog and Weiss

An implicit PVRK formula generally employs kernel values $k(t, s, y)$ with $s > t$. In this case the domain of $k(t, s, y)$ must be extended to include the required points outside S , but in general a continuation of the kernel, even if it is smooth, is arbitrary and non related to the original equations. The problem can be avoided by a modification of the VRK part in an implicit Pouzet method due to *de Hoog and Weiss* [31] characterized by $0 \leq c_1 < \dots < c_m = 1$, $b_k = \int_0^1 L_k(s) ds$, where $L_k(t)$ are the Lagrange fundamental polynomials in $[0, 1]$ with respect to c_i . These methods do not employ kernel values $k(t - s)$ with $s > t$ and, while the lag term formula (2.3.4) is unchanged, the system (2.3.3) assumes the form

$$Y_{n,i} = \bar{F}_n(t_{n,i}) + hc_i \sum_{l=1}^m b_l k \left(t_{n,i}, t_n + c_l c_l h, \sum_{s=1}^m L_s(c_l c_l) Y_{n,s} \right) \quad i = 1, \dots, m. \quad (2.3.5)$$

and (2.3.1) becomes

$$y_{n+1} = Y_{n,m} \quad n = 0, \dots, N_t - 1. \quad (2.3.6)$$

In this case the steps for the computation of the approximate solution y_{n+1} are:

- evaluate the lag terms $\bar{F}_n(t_{n,i})$ $i = 1, \dots, m$, using the expression (2.3.4);
- solve the nonlinear system (2.3.5) for $Y_{n,i}$ $i = 1, \dots, m$;
- determine the approximate solution y_{n+1} through (2.3.6).

Remark 2.6 (Computational cost). As observed in Remark 2.3 for collocation methods, also VRK methods (either of Pouzet type or of de Hoog and Weiss)

require $O(N_t^2)$ operations and $O(N_t)$ memory for computing the numerical solution over N_t time steps.

Remark 2.7. Observe that a modified VRK method of *de Hoog and Weiss* can be seen as a discrete collocation method (2.2.16)-(2.2.17) using interpolatory quadrature formulas with $\mu_0 = \mu_1 = m$, nodes $d_l = c_l$, $d_{i,l} = c_i c_l$ and corresponding weights $w_l = b_l$, $w_{i,l} = c_i b_l$.

Remark 2.8. Let us consider a modification of a VRK method of *de Hoog and Weiss* consisting of the formulas (2.3.4) and (2.3.5) respectively for the lag terms computation and for the determination of the stages, and of the formulas (2.3.1)-(2.3.2) for the computation of the approximated solution. This method can be seen as a discrete collocation method (2.2.16)-(2.2.17) with $m+1$ collocation parameters, $c_{m+1} = 1$, using interpolatory quadrature formulas with $\mu_0 = \mu_1 = m$, nodes $d_l = c_l$, $d_{i,l} = c_i c_l$, and corresponding weights $w_l = b_l$, $w_{i,l} = c_i b_l$ $l = 1, \dots, m$, $i = 1, \dots, m+1$.

2.3.3 Convergence results

For an easy analysis of the convergence of a PVRK method it is convenient to introduce the discrete increment operator Φ_n associated with (2.3.2)-(2.3.4) and defined by $\Phi_n[\bar{F}_n](= \Phi_n[\bar{F}_n; t, h]) := \bar{\Phi}_n(t)$, $n = 0, \dots, N_t - 1$.

The following theorem holds

Theorem 2.8. *If the following assumptions hold*

1. $A = (a_{i,j})$, $b = (b_i)$ ($i, j = 1, \dots, m$) define an m -stage RK method of order p for a first order ODE,

2. the increment operator Φ_n satisfies the Lipschitz condition

$$|\Phi_n[f_n] - \Phi_n[g_n]| \leq L |f_n - g_n|, \quad t \in [t_n, T], \quad n = 0, \dots, N_t - 1,$$

for all real f_n, g_n with L independent of n and h .

Then the approximation y_n generated by the PVRK method (2.3.1)-(2.3.4) satisfies, for all sufficiently differentiable functions f and k ,

$$\max_{1 \leq n \leq N_t} |y(t_n) - y_n| = O(h^p) \quad (2.3.7)$$

As regards the convergence analysis of a modified VRK method of de Hoog and Weiss, from Remarks 2.7 and 2.8 and from Theorem 2.7, it follows the local superconvergence result

Corollary 2.2. *Let $f, k \in C^{2m-v}$, with $v \in \{0, 1, 2\}$ with $m \geq \lfloor v/2 \rfloor + 1$,*

(i) *If the nodes $\{c_i\}$ are the Radau II points for $(0, 1]$, then the VRK method of the Hoog and Weiss (2.3.4)-(2.3.6) satisfies, for $v = 1$,*

$$\max_{1 \leq n \leq N_t} |y(t_n) - y_n| = O(h^{2m-1})$$

(ii) *If the nodes $\{c_i\}$ are the Lobatto points for $[0, 1]$, then the VRK method of the Hoog and Weiss (2.3.4)-(2.3.6) satisfies, for $v = 2$,*

$$\max_{1 \leq n \leq N_t} |y(t_n) - y_n| = O(h^{2m-2})$$

(iii) *Let the nodes $\{c_i\}$ are the the m Gauss points for $(0, 1)$, $c_{m+1} = 1$. Furthermore, suppose to consider the modification of the VRK method of the Hoog and Weiss described in Remark 2.8. Then, for $v = 0$,*

$$\max_{1 \leq n \leq N_t} |y(t_n) - y_n| = O(h^{2m})$$

Chapter 3

The Laplace transform and its numerical inversion

3.1 Introduction

The numerical inversion of a Laplace Transform function arises in many applications of science and engineering, such as fluid mechanics, circuit theory, spectroscopy, meteorology, medical imaging, etc., in which data delivered by an instrumental system mathematically represent the Laplace transform of the unknown object, that is of the “cause” which determined such effects. The interpretation of the physical system requires the restoration of the unknown object, i.e. the inverse Laplace function, from the measured experimental data, i.e. its Laplace function.

On the other hand there are many other situations in which methods based on integral transforms are used as instruments for the resolution of mathematical models which describe the real phenomena. For example the Laplace transform is employed to solve ordinary and partial differential equations, integral equations of convolution type and so on. In the image domain of the Laplace transformation such kind of equations are usually considerably simpler than the original ones (they are algebraic equations) and their solution is often a function of quite simple structure. Then, finally, one has to go back to the original domain inverting one or more Laplace functions. The inverse Laplace transform is, therefore, the computational kernel of the overall solution procedure.

In this chapter we first give the definition of Laplace transform and its inverse, showing their main properties. Then we examine some existing numerical methods for the inverse Laplace transform approximation ([56], [68], [82], [86]), which will be useful in the construction of the fast methods treated in this thesis. Finally we describe how these inverse Laplace transform approximation methods have been used for the construction of a fast convolution

quadrature formula for the computation of evolutionary convolution integrals in which the Laplace transform of the kernel is known a priori ([56], [68]). This fast quadrature formula represents the basic idea for the construction of the fast numerical methods for VIEs illustrated in Chapters 4 and 5.

3.2 The Laplace transform

In this section we will use a notation taken from the specialized literature on the Laplace transform [39],[40],[46],[53].

Definition 3.1. Let $f : \mathbb{R} \rightarrow \mathbb{C}$ be a function such that $f(t) = 0$ for all $t < 0$.

Let $s \in \mathbb{C}$. The integral

$$\int_0^{\infty} e^{-st} f(t) dt \quad (3.2.1)$$

is said to be the *Laplace integral* of the function $f(t)$ at the point s .

Definition 3.2. Let $C(f)$ be the subset of \mathbb{C} where the integral (3.2.1) is convergent. Then the *Laplace transform* of $f(t)$ is defined to be the function $\mathcal{L}(f) : C(f) \rightarrow \mathbb{C}$ obtained by setting

$$\mathcal{L}(f)(s) := \int_0^{\infty} e^{-st} f(t) dt \quad s \in C(f). \quad (3.2.2)$$

The set $C(f) \subseteq \mathbb{C}$ is said to be the *region of convergence* of $\mathcal{L}(f)$.

In the following a lower-case letter will be used to denote the function being transformed and the corresponding capital letter will be employed to represent the Laplace transform of this function: $F := \mathcal{L}(f)$. We will say that a function f is *Laplace transformable* (*L-transformable*) if its Laplace transform exists for at least one point $s \in \mathbb{C}$.

The region of convergence $C(f)$ of an L-transformable function is the whole complex plane or it is a complex half-plane of the form:

$$C(f) = \{s \in \mathbb{C} : \operatorname{Re}(s) > s_0\},$$

where $s_0 \in \mathbb{R}$ is called the *abscissa of convergence* of F .

Theorem 3.1. *The Laplace transform of an L -transformable function f is holomorphic in the region of convergence of the Laplace integral. The derivatives can be computed by derivating under the integration sign*

$$F^{(n)}(s) = (-1)^n \int_0^{\infty} e^{-st} t^n f(t) dt.$$

As the Laplace transform is analytic in its half-plane of convergence, it is possible to give a characterization of the abscissa of convergence in terms of the singularities of the function \tilde{F} , obtained by extending the Laplace transform F to all the complex plane. Infact

$$s_0 = \sup \left\{ \operatorname{Re}(s) : s \in \mathbb{C}, s \text{ singularity of } \tilde{F} \right\}. \quad (3.2.3)$$

Definition 3.3. The function $f(t)$ is said to be of exponential order γ for $t \rightarrow \infty$ if there exist real numbers γ , $M > 0$, and t_0 such that

$$|f(t)| < M e^{\gamma t} \quad \forall t \geq t_0. \quad (3.2.4)$$

The following theorem establishes a sufficient condition for the existence of the Laplace transform.

Theorem 3.2. *Let $f(t)$ be a piecewise continuous function in every bounded interval in the range $0 \leq t \leq t_0$. If $f(t)$ is of exponential order γ for $t > t_0$, then the Laplace transform $F(s)$ of $f(t)$ exists for all $s > \gamma$.*

Now we report the main properties of the Laplace transform:

Property 3.1 (Linearity). *Let f and g be L -transformable functions and let α and β be real numbers. Then: $\mathcal{L}(\alpha f + \beta g) = \alpha \mathcal{L}(f) + \beta \mathcal{L}(g)$*

Property 3.2 (Differentiation). *Let the $(n-1)$ -th derivative of the function f be locally absolutely continuous in \mathbb{R}_0^+ and its n -th derivative be L -transformable for $\operatorname{Re}(s) > s_0$. Then the function f is L -transformable at least for $\operatorname{Re}(s) > \max(0, s_0)$ and*

$$\mathcal{L}(f^{(n)})(s) = s^n \mathcal{L}(f)(s) - s^{n-1} f(0) - s^{n-2} f'(0) - \dots - s f^{(n-2)}(0) - f^{(n-1)}(0)$$

Property 3.3 (Integration). *Let f be L -transformable for $\operatorname{Re}(s) > s_0$, Then the function:*

$$g : t \in \mathbb{R}_0^+ \rightarrow \int_0^t f(x)dx$$

is L -transformable at least for $\operatorname{Re}(s) > \max(0, s_0)$ and

$$\mathcal{L}(g)(s) = s^{-1} \mathcal{L}(f)(s)$$

Property 3.4 (Convolution). *Let f be L -transformable in the half-plane $\{\operatorname{Re}(s) > \sigma_0\}$ and g be L -transformable in the half-plane $\{\operatorname{Re}(s) > \sigma_1\}$. Let $(f * g)(t) = \int_0^t f(t - \tau)g(\tau)d\tau = \int_0^t f(\tau)g(t - \tau)d\tau$ be the convolution between f and g . Then $f * g$ is L -transformable at least in the half-plane $\{\operatorname{Re}(s) > \max(\sigma_0, \sigma_1)\}$ and*

$$\mathcal{L}(f * g) = \mathcal{L}(f) \cdot \mathcal{L}(g)$$

3.3 The inverse Laplace transform

The Laplace transform of an L-transformable function is obviously unique. On the other hand there are infinite functions f whose Laplace transform is equal to a fixed function F , and they only differ on a subset of measure 0. So we can speak of the invertibility of the operator \mathcal{L} , provided that we identify two functions which coincide almost everywhere.

The following theorem gives an inversion formula, on which most of the methods of numerical inversion are based.

Theorem 3.3. (*Inversion formula of Bromwich-Mellin or Riemann-Fourier*).
Let f be an L-transformable function with Laplace transform F and abscissa of convergence s_0 . For any real number $\alpha > s_0$ it results

$$f(t) = \mathcal{L}^{-1}(F)(t) = \frac{1}{2\pi i} \int_{\alpha-i\infty}^{\alpha+i\infty} e^{st} F(s) ds \quad (3.3.1)$$

for each $t > 0$ in which $f(t)$ is continuous.

If $f(t)$ has a discontinuity of the first kind in $t > 0$, then

$$\frac{1}{2} [f(t^+) + f(t^-)] = \frac{1}{2\pi i} \int_{\alpha-i\infty}^{\alpha+i\infty} e^{st} F(s) ds.$$

The integral in the second hand side of the formula (3.3.1) is an improper integral defined as follows:

$$\int_{\alpha-i\infty}^{\alpha+i\infty} = \lim_{\beta \rightarrow \infty} \int_{\alpha-i\beta}^{\alpha+i\beta}$$

The inversion formula (3.3.1) requires the integration of a complex function along the vertical line whose equation is $x = \alpha$. This line is called *Bromwich line*. Note that the formula does not depend on α , provided that $\alpha > s_0$.

3.4 Methods for numerical inversion

The Riemann inversion formula (3.3.1) represents a valid instrument for the construction of numerical inversion methods, especially thanks to the fact that the integration contour can be arbitrarily translated, on the only condition that it remains at the right of the singularities of the Laplace transform. This possibility leads to important consequences from the computational point of view. For example, as we will see, the approximation obtained through the numerical integration of Riemann inversion formula highly depends on the position of the integration contour.

3.4.1 Talbot's method

The main numerical difficulties in the direct integration of the Riemann inversion formula are caused by the oscillations of the exponential e^{st} when $Im(s)$ tends to infinity. In *Talbot's method* [86] this difficulty is overcome by replacing the vertical line $x = \alpha$ by a contour Γ , starting and ending in the left half-plane, so that $Im(s)$ keeps finite and $Re(s) \rightarrow -\infty$ at each end. In this way the exponentials decay rapidly along such contour.

This replacement is permissible, i.e. Γ is equivalent to the line $x = \alpha$ in the computation of the integral (3.3.1), if

- Γ is situated at the right of all singularities of $F(s)$.
- $|F(s)| \rightarrow 0$ uniformly in $Re(s) \leq s_0$ as $|s| \rightarrow \infty$.

An example of integration contour proposed by Talbot is shown in Figure 3.1, whose parametrical representation is of the kind

$$\begin{aligned} (-\pi, \pi) &\rightarrow \Gamma \\ \vartheta &\rightarrow \gamma(\vartheta) = \sigma + \mu(\vartheta \cot(\vartheta) + i\nu\vartheta). \end{aligned} \quad (3.4.1)$$

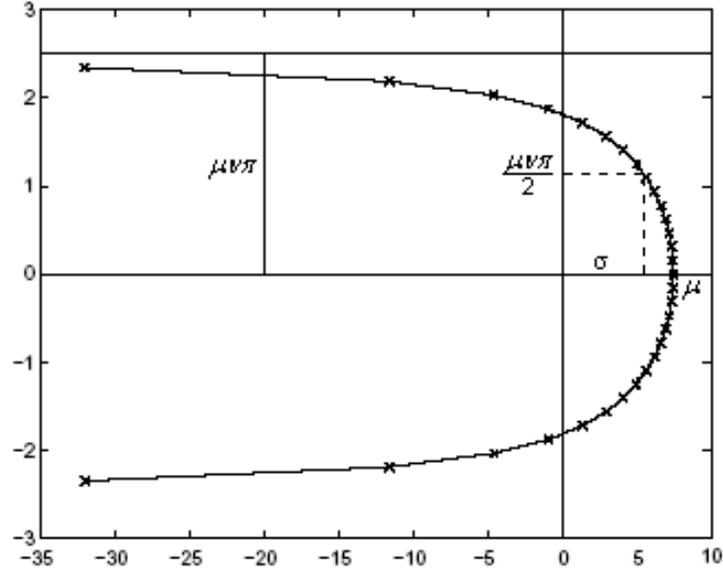


Figure 3.1: Talbot contour (3.4.1)

The real parameters σ , μ and ν have to be chosen such that the contour $\Gamma = \Gamma(\sigma, \mu, \nu)$ is

- as much as possible shifted to the left, in order to reduce the oscillations in the exponentials;
- not too near to the singularities of $F(s)$, in order to avoid oscillations in $F(s)$.

As shown in Figure 3.1, an appropriate choice of the parameters σ , μ and ν permits to adapt the integration contour to the geometrical arrangement of the singularities of $F(s)$: σ permits to shift the contour, μ permits to effect a radial dilatation of Γ , ν permits to effect a vertical dilatation.

After replacing the line $x = \alpha$ with the contour Γ defined by (3.4.1), the Riemann integral (3.3.1) becomes

$$f(t) = \frac{1}{2\pi i} \int_{-\pi}^{\pi} f(\gamma(\vartheta)) e^{t\gamma(\vartheta)} \gamma'(\vartheta) d\vartheta. \quad (3.4.2)$$

By applying the composite trapezoidal rule in $[-\pi, \pi]$ with stepsize $h = \frac{\pi}{N+1}$ and nodes $\{\vartheta_j = \frac{j\pi}{N+1}, j = -(N+1), \dots, N+1\}$, and by observing that the integrand function vanishes at $\theta = -\pi, \pi$, the (3.3.1) leads to the approximation

$$\tilde{f}(t) = \frac{1}{2\pi i} h \sum_{j=-N}^N f(\gamma(\vartheta_j)) e^{t\gamma(\vartheta_j)} \gamma'(\vartheta_j),$$

and hence

$$\tilde{f}(t) = \sum_{j=-N}^N \omega_j F(\lambda_j) e^{t\lambda_j}, \quad (3.4.3)$$

where $\lambda_j = \gamma(\vartheta_j)$ are the nodes on the Talbot contour and $\omega_j = -\frac{i}{2(N+1)} \gamma'(\vartheta_j)$ are the corresponding weights.

Let us denote $M = 2N+1$ the total number of points on the Talbot contour and

$$E(t, M) = \tilde{f}(t) - f(t)$$

the error of Talbot's method. In [82] [86] it has been shown that $\lim_{M \rightarrow \infty} E(t, M) = 0$ for all $t > 0$, and it decays exponentially with M . In fact

$$|E(t, M)| \leq CM^2 e^{\sigma t + c_1 \mu t - c_2 \sqrt{\mu t M}} \quad (3.4.4)$$

where the constant $C > 0$ depends on the geometrical parameters μ, σ, ν and on bounds on F , and the constants $c_1, c_2 > 0$ depend on the distance of the singularities of the analytic function F to the contour Γ . In [86] a deep analysis of the discretization and round off error has been carried out, leading to the determination of

- the optimal geometrical parameters $\mu = \mu(t)$, $\sigma = \sigma(t)$, $\nu = \nu(t)$, and hence the optimal contour $\Gamma = \Gamma(t)$ which minimizes the discretization error $E(t, M)$;

- the optimal value of $M = M(t, \epsilon)$ dependent not only on t but also on the accuracy ($O(10^{-\epsilon})$) required in the computation of $\tilde{f}(t)$ (for this reason M is also referred to as the accuracy parameter).

This method has been implemented in a FORTRAN subroutine TAPAR, which is available on the library of the Collected Algorithms of ACM TOMS [74].

3.4.2 Modified Talbot's method (Rizzardi)

Unfortunately a disadvantage in Talbot's method is that it needs to repeat all computation for each t , and this may become very inefficient when we have to compute the inverse Laplace transform in an interval $[t_{\min}, t_{\max}]$. A modification of Talbot's method has been proposed in [82]: the parameters μ , σ , ν , M are chosen at a point $t^* \in [t_{\min}, t_{\max}]$ and used for the computation of $f(t)$ on the whole interval. The error analysis has permitted to establish a further correction to produce on the parameter M and to choose the optimal $t^* = \frac{1}{2}(t_{\min} + t_{\max})$.

The computational strategy consists in the following algorithm:

- by a call to the subroutine TAPAR [74] the parameters μ , σ , ν , M according to Talbot's method are computed for $t = t^*$.
- a correction to the accuracy parameter M is computed by choosing $M^* = \max\{M_1, M_2\}$, with

$$\begin{aligned} M_1 &= M + \frac{1}{2}(t_{\min} - t_{\max}) \left(\mu \frac{\nu + 1}{2} + \frac{\sigma}{\rho} \right) \\ M_2 &= M + \frac{1}{2}(t_{\min} - t_{\max}) \left(\frac{\sigma + \mu}{\rho} - \mu \frac{\nu - 1}{2} \right), \end{aligned}$$

where ρ is obtained from the solution of the nonlinear equation

$$\rho > 0 : \rho \left(\frac{1}{e^\rho - 1} - \frac{\nu - 1}{2} \right) = \frac{1}{2} \left(1 - \frac{\sigma - s_0}{\mu} \right),$$

with s_0 abscissa of convergence of F (see (3.2.3)).

- use the parameters $\mu(t^*)$, $\sigma\mu(t^*)$, $\nu\mu(t^*)$ and M^* to compute the approximation (3.4.3) for all $t \in [t_{\min}, t_{\max}]$: a unique Talbot contour $\Gamma(t^*)$ is associated to the whole interval $[t_{\min}, t_{\max}]$.

3.4.3 Modified Talbot's method (Lubich-Schädle)

When the interval $[t_{\min}, t_{\max}]$ is large, the modified Talbot's method described in the previous subsection has the limitation that a uniform approximation of $f(t)$ on the whole interval would require a rather large number M of points on the Talbot contour (this because of t at the exponent in the error estimate (3.4.4)).

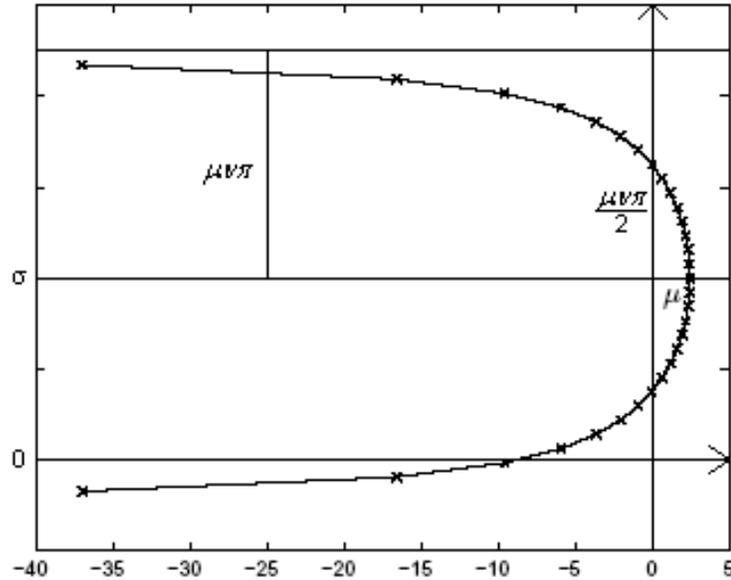


Figure 3.2: Talbot contour (3.4.6)

The idea proposed in [56], [68] has been that of splitting a large interval into a sequence of subintervals, and in each of them to use a suitably chosen Talbot contour.

More precisely let us consider an interval of the form $[0, T]$ and let $h > 0$. Then $f(t)$ is approximated *locally* on a sequence of fast growing intervals covering $[0, T]$

$$I_0 = [0, h], \quad I_l = [B^{l-1}h, (2B^l - 1)h], \quad (3.4.5)$$

where $B > 1$ is a fixed integer and $l = 1, \dots, L$ with $(2B^L - 1)h \geq T$. A Talbot contour Γ_l

$$(-\pi, \pi) \rightarrow \Gamma_l \quad (3.4.6)$$

$$\vartheta \rightarrow \gamma_l(\vartheta) = \sigma + \mu_l(\vartheta \cot(\vartheta) + i\nu\vartheta),$$

is associated to each subinterval I_l , and is obtained by opportunely choosing the geometrical parameters:

- the parameter $\mu_l := \mu_0/T_l$ depends on I_l via its end-point $T_l = (2B^l - 1)h$,
- the parameters ν and σ depend on the singularities of $F(s)$ and have to be chosen in a way such that all the singularities lie on the left of the contour and such that they are not too “close” to the contour. In particular σ is chosen to be purely imaginary (instead of being real), so that it permits to vertically shift the Talbot contour and ν is chosen of the form $\nu = \nu_0(1 + \omega/\beta)$, $\beta = \pi\mu_l\nu_0/2$ and permits to effect a vertical dilatation (see Figure 3.2).

The choice of the parameters μ_0 , ν_0 , ω and σ has been done experimentally in [56], [68] by minimizing the error of the inverse Laplace transform approximation. The curve Γ_l can either be composed by only one Talbot contour enclosing all the singularities of $F(s)$, or it can also be composed by more Talbot contours each enclosing a certain number of singularities. Namely, if the singularities are “sufficiently close” to each other then Γ_l is composed by only one Talbot contour, as shown in Figure 3.3. If they are “far” from each other

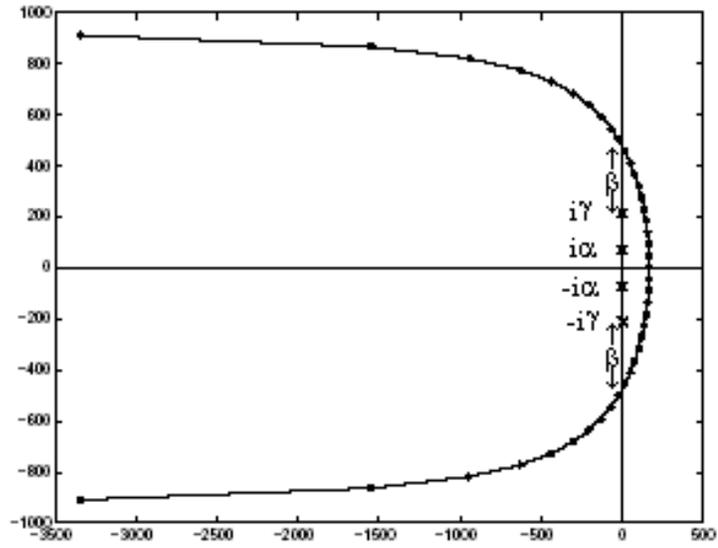


Figure 3.3: One contour enclosing all singularities.

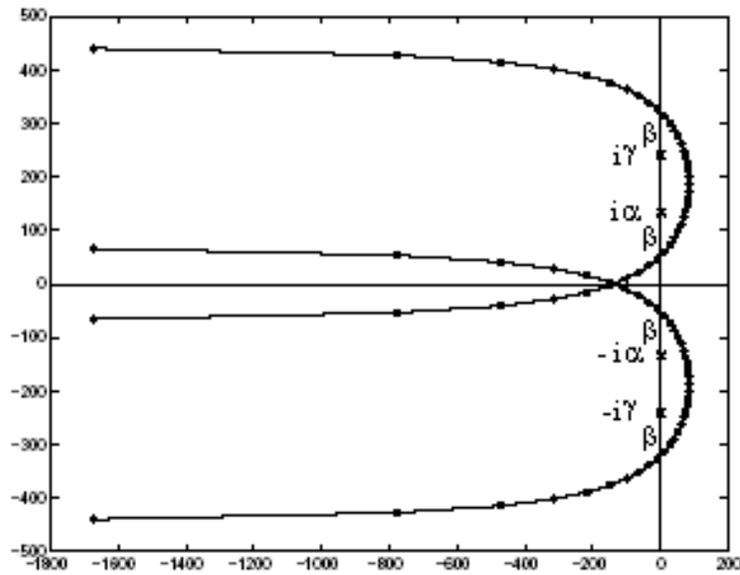


Figure 3.4: Two contours enclosing all singularities.

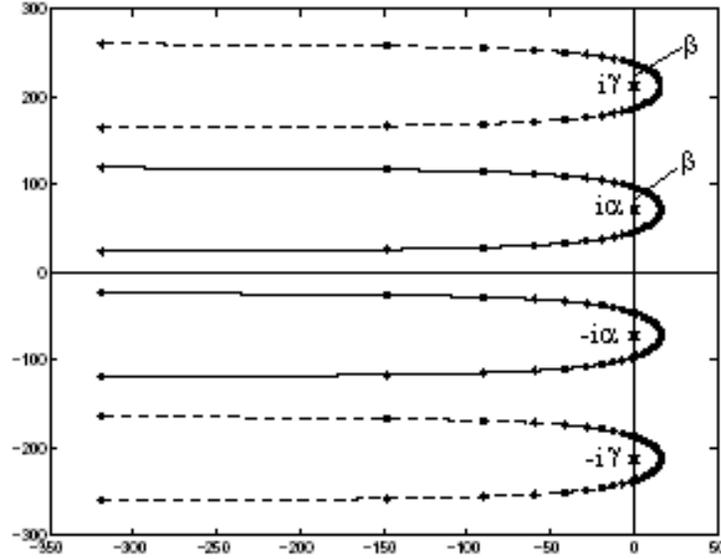


Figure 3.5: Four contours enclosing all singularities.

we may choose Γ_l composed by more Talbot contours, the number depending on the distance among the singularities, as shown in Figures 3.4-3.5. All the pictures show that the choice of σ and ω is done in a way such that the distance between the intersections of the Talbot contour with the imaginary axis and each singularity is at least equal β . This will be illustrated in the following examples, which will be useful in the subsequent chapters. In all of them the parameters μ_0 and ν_0 are chosen to be $\mu_0 = 8$, $\nu_0 = 0.6$ (see [56], [68]).

Example 3.1. Let us assume that $F(s)$ has two singularities at the points $s = \pm i\alpha$. Then if $\beta < \alpha$ (that is the two singularities are “sufficiently far” from each other) we choose two contours with $\sigma = \pm i\alpha$ and $\omega = 0$, as shown in Figure 3.6. If $\beta > \alpha$ we choose one contour with $\sigma = 0$ and $\omega = \alpha$, as shown in Figure 3.7.

Example 3.2. Let us assume that $F(s)$ has four singularities at the points $s = \pm i\alpha$, $s = \pm i\gamma$. Then we distinguish four cases:

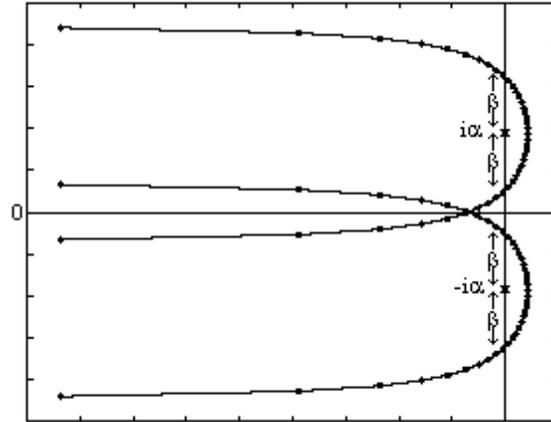


Figure 3.6: Two contours enclosing the singularities $s = \pm i\alpha$.

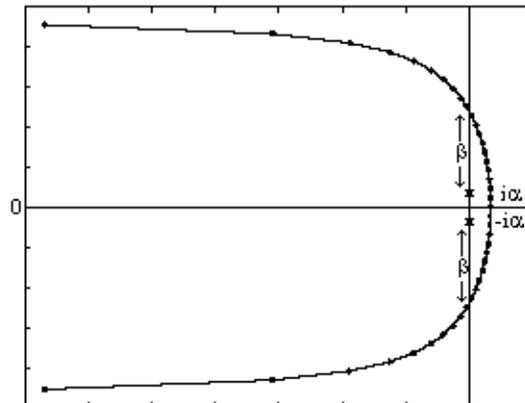


Figure 3.7: One contour enclosing the singularities $s = \pm i\alpha$.

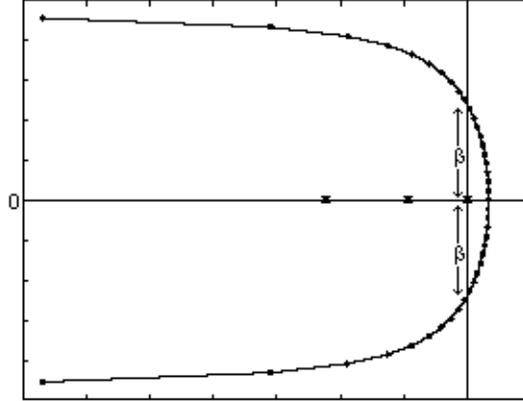


Figure 3.8: One contour enclosing singularities at $s = -\lambda$, $\lambda \geq 0$.

1. $\beta > \alpha$, $\beta \geq (\gamma - \alpha)/2$. In this case we choose one contour with $\sigma = 0$, $\omega = \gamma$ as shown in Figure 3.3.
2. $\beta \leq \alpha$, $\beta < (\gamma - \alpha)/2$. In this case we choose four contours with $\sigma = \pm i\alpha$, $\omega = 0$; $\sigma = \pm i\gamma$, $\omega = 0$, as shown in Figure 3.5.
3. $\alpha < \beta < (\gamma - \alpha)/2$. In this case we choose three contours with $\sigma = 0$, $\omega = \alpha$; $\sigma = \pm i\gamma$, $\omega = 0$.
4. $(\gamma - \alpha)/2 < \beta < \alpha$. In this case we choose two contours with $\sigma = \pm i(\gamma + \alpha)/2$, $\omega = (\gamma - \alpha)/2$, as shown in Figure 3.4.

Example 3.3. Let us assume that $F(s)$ has singularities on the real non positive semiaxe, i.e. $s = -\lambda$, $\lambda \geq 0$. Then we choose $\sigma = 0$, $\omega = 0$, as shown in figure 3.8.

The approximation of $f(t)$ on I_l results from (3.4.3) and reads

$$f(t) \approx \sum_{j=-N}^N \omega_j^{(l)} F(\lambda_j^{(l)}) e^{t\lambda_j^{(l)}} \quad t \in I_l \quad (3.4.7)$$

with $\omega_j^{(l)} = -\frac{i}{2(N+1)}\gamma_l'(\vartheta_j)$, $\lambda_j^{(l)} = \gamma_l(\vartheta_j)$, $\vartheta_j = \frac{j\pi}{N+1}$. The number of quadrature points $M = 2N + 1$ is chosen on Γ_l is independent of l and it is much smaller than it would be required for a uniform approximation on the whole interval $[0, T]$.

Taking in consideration the expression (3.4.4) for the error of Talbot's approximation method, exploiting the fact that $\mu = \mu_0/T_l$ and $\sigma \in i\mathbb{R}$, it follows that in this case:

$$|E(t, M)| \leq C_l M^2 e^{c_1 \mu_0 - c_2 \sqrt{\mu_0 M}} \quad t \in I_l$$

and thus, by putting $C = \max_{l=1, \dots, L} C_l e^{c_1 \mu_0}$, $c = -c_2 \sqrt{\mu_0}$, it follows that $|E(t, M)| \leq C M^2 e^{-c\sqrt{M}}$ for all $t \in [0, T]$, and then

$$\|E(t, M)\|_{t \in [0, T]} = O(e^{-c\sqrt{M}}) \quad (3.4.8)$$

for $M \rightarrow \infty$, uniformly on $[0, T]$.

3.5 Fast convolution quadrature formulas

The inverse Laplace transform approximation (3.4.7) described in section (3.4.3) has been used in [56], [68] for the construction of a fast algorithm for the computation of the temporal convolution

$$\int_0^t k(t-\tau)y(\tau)d\tau \quad t \in [0, T] \quad (3.5.1)$$

on the grid $I_h = \{t_n := nh, n = 0, \dots, N_t, h \geq 0, N_t h = T\}$ with stepsize h , in the assumption that the Laplace transform of the kernel $K(s)$ rather the kernel itself is known a priori. It has been taken in consideration the case in which the evaluation of $y(\tau)$ at $\tau = nh$ requires the knowledge of the values of the convolution up to $(n-1)h$, so that the required values of $y(\tau)$ cannot be computed in advance. This is the situation in Volterra-type convolution equations.

A naive implementation of a quadrature formula would require $O(N_t^2)$ operations and $O(N_t)$ memory for computing the temporal convolution over N_t time steps. The fast convolution algorithm, instead, takes $O(N_t \log N_t)$ operations and $O(\log N_t)$ memory.

3.5.1 Approximation of definite integrals

Let us split the interval $[0, T]$ according to (3.4.5).

For a definite integral $\int_a^b k(t-\tau)y(\tau)d\tau$, if $[t-a, t-b] \subseteq I_l$, using (3.4.7) we obtain:

$$\int_a^b k(t-\tau)y(\tau)d\tau \approx \sum_{j=-N}^N \omega_j^{(l)} K(\lambda_j^{(l)}) e^{(t-b)\lambda_j^{(l)}} \int_a^b e^{(b-\tau)\lambda_j^{(l)}} y(\tau)d\tau$$

and thus, setting $z(b, a, \lambda) = \int_a^b e^{(b-\tau)\lambda} y(\tau) d\tau$, we have

$$\int_a^b k(t-\tau)y(\tau)d\tau \approx \sum_{j=-N}^N \omega_j^{(l)} K(\lambda_j^{(l)}) e^{(t-b)\lambda_j^{(l)}} z(b, a, \lambda_j^{(l)}). \quad (3.5.2)$$

Note that $z(b, a, \lambda)$ can be recognized as the solution at time b of the scalar linear initial value problem

$$\begin{cases} z' = \lambda z + y \\ z(a) = 0. \end{cases} \quad (3.5.3)$$

Remark 3.1. Since $\int_a^b k(t-\tau)y(\tau)d\tau = \frac{1}{2\pi i} \int_{\Gamma_t} K(\lambda) e^{(t-b)\lambda} z(b, a, \lambda) d\lambda$, the formula (3.5.2) represents the approximation of the inverse Laplace transform of the function $K(\lambda)z(b, a, \lambda)$ with the formula (3.4.7). Thus, according to (3.4.8), the error of the quadrature formula (3.5.2) satisfies

$$\left| \int_a^b k(t-\tau)y(\tau)d\tau - \sum_{j=-N}^N \omega_j^{(l)} K(\lambda_j^{(l)}) e^{(t-b)\lambda_j^{(l)}} z(b, a, \lambda_j^{(l)}) \right| = O(e^{-\rho M}). \quad (3.5.4)$$

The $2N + 1$ differential equations (3.5.3) with $\lambda = \lambda_j^{(l)}$ are then solved approximately by replacing the function y with its piecewise linear approximation and then solving exactly.

More precisely, we split the interval $[a, b]$ in subintervals $[a + t_k, a + t_{k+1}]$, $k = 0, \dots, \bar{n} - 1$, $\bar{n} = (b - a)/h$. We denote with $z_k = z(a + t_k, a, \lambda)$ the exact solution at the point $a + t_k$. The differential problem (3.5.3), restricted to each of these subintervals, reads

$$\begin{cases} z'(t) = \lambda z(t) + y(t) \\ z(a + t_k) = z_k \end{cases} \quad t \in [a + t_k, a + t_{k+1}], \quad (3.5.5)$$

whose exact solution is given by the variation-of-constant formula:

$$z_{k+1} = e^{\lambda h} z_k + h \int_0^1 e^{(1-\theta)\lambda h} y(a + t_k + \theta h) d\theta \quad k = 0, \dots, \bar{n} - 1. \quad (3.5.6)$$

Denoting by $\tilde{y}_k = y(a + t_k)$ and replacing $y(t)$ by its linear approximation $u(t)$ in $[a + t_k, a + t_{k+1}]$:

$$u(a + t_k + \theta h) = \theta \tilde{y}_{k+1} + (1 - \theta) \tilde{y}_k \quad \theta \in (0, 1] \quad (3.5.7)$$

we obtain the following approximation $\bar{z}_k \approx z(a + t_k, a, \lambda)$:

$$\bar{z}_{k+1} = e^{\lambda h} \bar{z}_k + h \int_0^1 e^{(1-\theta)\lambda h} u(a + t_k + \theta h) d\theta \quad k = 0, \dots, \bar{n} - 1. \quad (3.5.8)$$

By exactly computing the integral in (3.5.8), we can obtain the approximation \bar{z}_k recursively via

$$\bar{z}_{k+1} = \bar{z}_k + \frac{e^{h\lambda} - 1}{h\lambda} \left(h\lambda \bar{z}_k + h\tilde{y}_k + h \frac{\tilde{y}_{k+1} - \tilde{y}_k}{h\lambda} \right) - h \frac{\tilde{y}_{k+1} - \tilde{y}_k}{h\lambda}. \quad (3.5.9)$$

3.5.2 The fast convolution algorithm

The integral (3.5.1) for $t = t_{n+1}$ is splitted as

$$\int_0^{t_{n+1}} k(t_{n+1} - \tau) y(\tau) d\tau = \int_{t_n}^{t_{n+1}} k(t_{n+1} - \tau) y(\tau) d\tau + \int_0^{t_n} k(t_{n+1} - \tau) y(\tau) d\tau. \quad (3.5.10)$$

1. The first integral in the second hand side of (3.5.10) can be computed by approximating $y(\tau)$ linearly:

$$\begin{aligned} \int_{t_n}^{t_{n+1}} k(t_{n+1} - \tau) y(\tau) d\tau &\approx y(t_n) \int_{t_n}^{t_{n+1}} k(t_{n+1} - \tau) d\tau + \\ &+ \frac{y(t_{n+1}) - y(t_n)}{h} \int_{t_n}^{t_{n+1}} k(t_{n+1} - \tau) \cdot (\tau - t_n) d\tau. \end{aligned}$$

The remaining integrals are approximated as the inverse Laplace transforms of $F(s)/s$ and $F(s)/s^2$ respectively:

$$\begin{aligned} \Phi_1 &= \int_{t_n}^{t_{n+1}} k(t_{n+1} - \tau) d\tau = \int_0^h k(h - \tau) d\tau \approx \sum_{j=-N}^N \omega_j \frac{K(\lambda_j)}{\lambda_j} e^{h\lambda_j} \\ \Phi_2 &= \int_{t_n}^{t_{n+1}} k(t_{n+1} - \tau) \cdot (\tau - t_n) d\tau = \int_0^h k(h - \tau) \tau d\tau \approx \sum_{j=-N}^N \omega_j \frac{K(\lambda_j)}{\lambda_j^2} e^{h\lambda_j}, \end{aligned}$$

where the weights ω_j and nodes λ_j are associated to the Talbot contour Γ_0 that corresponds to the interval I_0 of (3.4.5). Thus we obtain

$$\int_{t_n}^{t_{n+1}} k(t_{n+1} - \tau)y(\tau)d\tau \approx \Phi_1 y(t_n) + \Phi_2 \frac{y(t_{n+1}) - y(t_n)}{h} \quad (3.5.11)$$

2. Let us consider L as the smallest integer for which $t_{n+1} < 2B^L h$ and, for $l = 1, 2, \dots, L - 1$, we determine the integer $q_l \geq 1$ such that $\tau_l = q_l B^L h$ satisfies

$$t_{n+1} - \tau_l \in [B^l h, (2B^l - 1)h] \quad l = 1, \dots, L - 1 \quad (3.5.12)$$

and set $\tau_0 = t_n$, $\tau_L = 0$. Note that $t_n = \tau_0 > \tau_1 > \dots > \tau_{L-1} > \tau_L = 0$, so

$$[0, t_n] = \bigcup_{l=0}^L [\tau_l, \tau_{l-1}] \quad (3.5.13)$$

and this decomposition depends on n .

Then the second integral in the second hand side of (3.5.10) can be splitted as

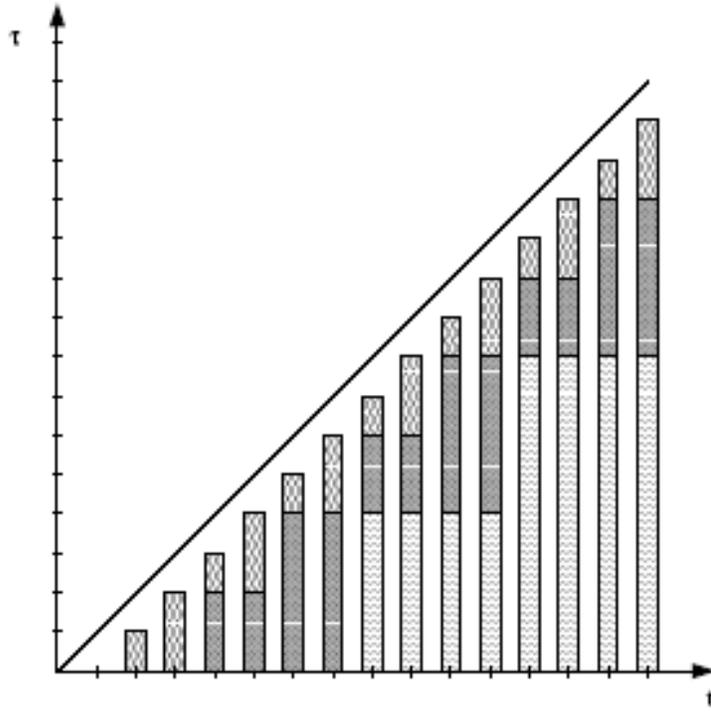
$$\int_0^{t_n} k(t_{n+1} - \tau)y(\tau)d\tau = \sum_{l=1}^L \int_{\tau_l}^{\tau_{l-1}} k(t_{n+1} - \tau)y(\tau)d\tau$$

It is easy to verify, thanks to (3.5.12), that $[t_{n+1} - \tau_{l-1}, t_{n+1} - \tau_l] \subseteq I_l$, thus it is possible to use the approximation (3.5.2) for each integral over $[\tau_l, \tau_{l-1}]$, obtaining

$$\int_0^{t_n} k(t_{n+1} - \tau)y(\tau)d\tau \approx \sum_{l=1}^L \sum_{j=-N}^N \omega_j^{(l)} K(\lambda_j^{(l)}) e^{(t_{n+1} - \tau_{l-1})\lambda_j^{(l)}} z(\tau_{l-1}, \tau_l, \lambda_j^{(l)}), \quad (3.5.14)$$

where $\omega_j^{(l)}$ and $\lambda_j^{(l)}$ are the weights and the quadrature points for the Talbot contour Γ_l that corresponds to the interval I_l of (3.4.5).

The fast convolution algorithm consists in an appropriate scheme in the computation of the approximation (3.5.11),(3.5.14) of (3.5.1). This algorithm

Figure 3.9: Decomposition for $B = 2$

is best explained by describing the first steps for $B = 2$ in (3.4.5), which leads to $I_0 = [0, h]$, $I_1 = [h, 3h]$, $I_2 = [2h, 7h]$, $I_3 = [4h, 15h]$,... and so on.

Figure 3.9 represents the decomposition of the triangle $\{(t, \tau) : 0 \leq \tau \leq t \leq T\}$ in L -shaped regions, according to the decomposition (3.5.13): along any vertical line, i.e. for any fixed $t = (n+1)h$, there are at most $\log_2 N_t$ such regions, and they are represented by different textures in Figure 3.9. Thus, the different textures correspond to different values of l , and so to different Talbot contours Γ_l , which in turn correspond to different approximation intervals I_l .

Example 3.4. For example, in correspondence of $t = t_8$, according to (3.5.12), we have $L = 3$, $\tau_0 = 7h$, $\tau_1 = 6h$, $\tau_2 = 4h$, $\tau_3 = 0$, as shown in Figure (3.10).

First step. At the first step we have $n = 0$, $t_{n+1} = h$, $L = 0$ and $\tau_0 = 0$.

Thus the integral $\int_0^h k(h - \tau)y(\tau)d\tau$ is approximated by (3.5.11).

Second step. At the second step we have $n = 1$, $t_{n+1} = 2h$, $L = 1$, $\tau_0 = h$ and $\tau_1 = 0$. Thus the integral $\int_h^{2h} k(2h - \tau)y(\tau)d\tau$ is approximated by

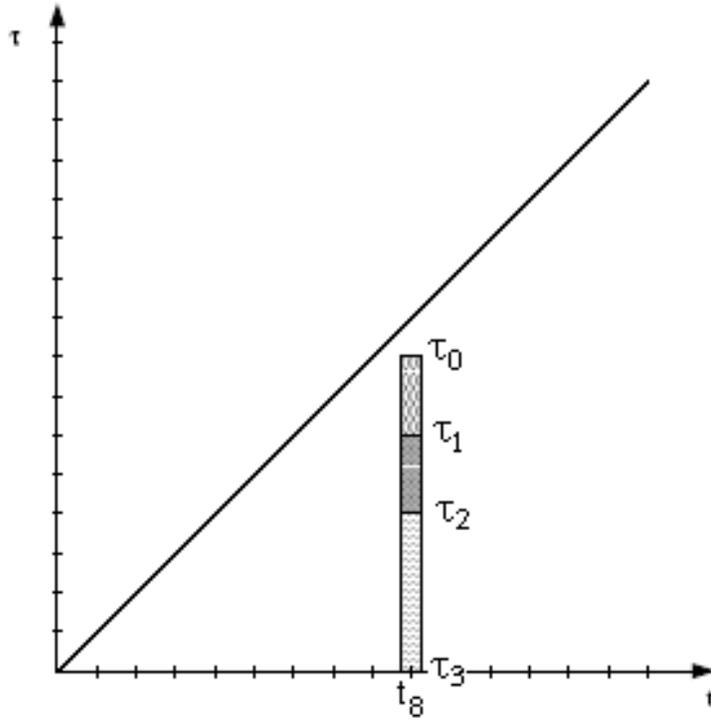


Figure 3.10: Decomposition for $B = 2$ in correspondence of $t = t_8$.

(3.5.11), while we approximate

$$\int_0^h k(2h - \tau)y(\tau)d\tau \approx \sum_{j=-N}^N \omega_j^{(1)} K(\lambda_j^{(1)}) e^{h\lambda_j^{(1)}} z(h, 0, \lambda_j^{(1)})$$

according to (3.5.14). The computation of $z(h, 0, \lambda_j^{(1)})$ requires solving M differential equations (3.5.3) with $\lambda = \lambda_j^{(1)} \in \Gamma_1$ by one step of (3.5.9), using the values of $y(0)$ and $y(h)$.

At this step we also compute $z(h, 0, \lambda_j^{(l)})$, by solving for all $l \geq 1$ the differential equations (3.5.3) with $\lambda = \lambda_j^{(l)} \in \Gamma_l$:

$$y(0), y(h) \xrightarrow{\text{one step of (3.5.9)}} z(h, 0, \lambda_j^{(l)}) \quad \forall l \geq 1. \quad (3.5.15)$$

This quantities will be used in the next time steps, without needing to keep in memory $y(0)$.

Third step. Now we have $n = 2$, $t_{n+1} = 3h$, $L = 1$, $\tau_0 = 2h$ and $\tau_1 = 0$.

As before, the integral $\int_{2h}^{3h} k(3h - \tau)y(\tau)d\tau$ is approximated by (3.5.11),

while (3.5.14) leads to

$$\int_0^{2h} k(3h - \tau)y(\tau)d\tau \approx \sum_{j=-N}^N \omega_j^{(1)} K(\lambda_j^{(1)}) e^{h\lambda_j^{(1)}} z(2h, 0, \lambda_j^{(1)}).$$

The computation of $z(2h, 0, \lambda_j^{(1)})$ requires advancing the solutions of the differential equations for Γ_1 from h to $2h$ by mean of one step of (3.5.9), using the values $z(h, 0, \lambda_j^{(1)})$ calculated at the previous step and the values $y(h)$ and $y(2h)$.

At this step we also advance the solution on all the other Talbot contours Γ_l :

$$\begin{array}{l} z(h, 0, \lambda_j^{(l)}) \\ y(h), y(2h) \end{array} \xrightarrow{\text{one step of (3.5.9)}} z(2h, 0, \lambda_j^{(l)}) \quad \forall l \geq 1. \quad (3.5.16)$$

This quantities will be used in the next time steps, without needing to keep in memory $y(h)$.

Fourth step. Now $n = 3$, $t_{n+1} = 4h$, $L = 2$, $\tau_0 = 3h$, $\tau_1 = 2h$ and $\tau_2 = 0$.

The integral $\int_{3h}^{4h} k(4h - \tau)y(\tau)d\tau$ is approximated by (3.5.11), while, according to (3.5.14), we obtain:

$$\begin{aligned} \int_{\tau_1}^{\tau_0} k(t_{n+1} - \tau)y(\tau)d\tau &= \int_{2h}^{3h} k(4h - \tau)y(\tau)d\tau \approx \\ &\approx \sum_{j=-N}^N \omega_j^{(1)} K(\lambda_j^{(1)}) e^{h\lambda_j^{(1)}} z(3h, 2h, \lambda_j^{(1)}) \quad (l = 1) \\ \int_{\tau_2}^{\tau_1} k(t_{n+1} - \tau)y(\tau)d\tau &= \int_0^{2h} k(4h - \tau)y(\tau)d\tau \approx \\ &\approx \sum_{j=-N}^N \omega_j^{(2)} K(\lambda_j^{(2)}) e^{2h\lambda_j^{(2)}} z(2h, 0, \lambda_j^{(2)}) \quad (l = 2) \end{aligned}$$

The computation of $z(2h, 0, \lambda_j^{(2)})$ has already been done at the previous time step by mean of (3.5.16), while the computation of $z(3h, 2h, \lambda_j^{(1)})$

requires the solution of the differential equations for Γ_1 with initial point $a = 2h$, by one step of (3.5.9):

$$y(2h), y(3h) \xrightarrow{\text{one step of (3.5.9)}} z(3h, 2h, \lambda_j^{(l)}) \quad l = 1. \quad (3.5.17)$$

At this step we also advance

$$\begin{array}{l} z(2h, 0, \lambda_j^{(l)}) \\ y(2h), y(3h) \end{array} \xrightarrow{\text{one step of (3.5.9)}} z(3h, 0, \lambda_j^{(l)}) \quad \forall l \geq 2 \quad (3.5.18)$$

according to Figure (3.9). These quantities will be used in the next time steps, without needing to keep in memory $y(2h)$. In this way, instead of proceeding step-wise from bottom up in the triangle, we proceed from left to right.

Fifth step. It is now clear that at the fifth step we need $z(4h, 2h, \lambda_j^{(1)})$ and again $z(2h, 0, \lambda_j^{(2)})$. Thus we evaluate:

$$\begin{array}{l} z(3h, 2h, \lambda_j^{(l)}) \\ y(3h), y(4h) \end{array} \xrightarrow{\text{one step of (3.5.9)}} z(4h, 2h, \lambda_j^{(l)}) \quad l = 1 \quad (3.5.19)$$

and

$$\begin{array}{l} z(3h, 0, \lambda_j^{(l)}) \\ y(3h), y(4h) \end{array} \xrightarrow{\text{one step of (3.5.9)}} z(4h, 0, \lambda_j^{(l)}) \quad \forall l \geq 2 \quad (3.5.20)$$

which we will use in the subsequent time steps.

As described in the first time steps, the differential equations (3.5.3) determining $z(\tau_{l-1}, \tau_l, \lambda_j^{(l)})$ are solved approximately by means of (3.5.9) and are advanced by one step of (3.5.9) for all required values $\lambda_j^{(l)}$, $j = -N, \dots, N$ on all Talbot contours in every time step $t_n \rightarrow t_{n+1}$. In this way the past values of $y(t)$ need not to be kept in memory. Thus the operations counts and memory

requirements are proportional to MN_tL_t and ML_t respectively, where N_t is the number of time steps, $L_t \leq \log_B N_t$ is the number of different contours, and $M = 2N + 1$ is the number of quadrature points on each contour. Thus the computational cost is of $O(N_t \log_B N_t)$ operations and memory requirements of $O(N_t)$.

Chapter 4

Fast Collocation methods for Volterra Integral equations of convolution type

4.1 Introduction

This Chapter concerns with the numerical solution of VIEs of Hammerstein type

$$y(t) = f(t) + \int_0^t k(t - \tau)g(y(\tau))d\tau \quad t \in I := [0, T], \quad (4.1.1)$$

where only the Laplace transform of the kernel $K(s)$ rather than the convolution kernel itself is known a priori. We have seen in Section 1.3 that this kind of equations are the mathematical model of several real problems. The functions f, g in (4.1.1) are assumed to be sufficiently smooth on I so that the solution $y(t)$ is smooth, too. As we previously observed, the numerical treatment of a VIE leads to an high computational cost, since, for each time step, we have to compute the “lag term”, which contains the past history of the phenomenon. In order to compute the numerical solution of (4.1.1) with a classical numerical method over N_t time steps it would be required $O(N_t^2)$ operations and $O(N_t)$ memory (see Chapter 2). In the special case of convolution type equations in [52] was proposed a fast algorithm with a computational cost of $O(N_t(\log N_t)^2)$.

In this chapter we construct fast discrete collocation methods for the equation (4.1.1), which directly involve the evaluations of the Laplace transform of the kernel, using the basic idea underlying the quadrature formulas illustrated in Section 3.5 and proposed in [68]. It will be shown that these methods, as in [68], can be implemented with a computational cost of $O(N_t \log N_t)$ operations and $O(\log N_t)$ memory. Moreover they keep the same order of accuracy of the corresponding classical collocation methods.

In Section 4.2 we give the detailed construction of the fast collocation methods and the calculation of the computational cost. The convergence analysis is given in Section 4.3.

Section 4.4 contains numerical results for some significant test examples taken from the literature, which confirm the expected accuracy and computational cost of the constructed methods. In Section 4.5 some concluding remarks are reported.

Some of the results of this chapter are reported in [21].

4.2 Fast collocation methods

We want to solve the equation (4.1.1) by a discrete m -point collocation method, constructed taking into account the peculiarity of the considered equation.

Only for ease of exposition we shall refer to the linearized form of (4.1.1)

$$y(t) = f(t) + \int_0^t k(t - \tau)y(\tau)d\tau \quad t \in I, \quad (4.2.1)$$

since it is possible to extend in a natural way the same results to the more general equation (4.1.1).

Let us fix m collocation parameters c_1, \dots, c_m and let us consider a uniform mesh of the temporal interval I denoted by

$$I_h = \{t_n := nh, n = 0, \dots, N_t, h \geq 0, N_t h = T\}.$$

An m -point collocation method applied to the equation (4.2.1), according to the formulas (2.2.3), (2.2.4), reads

$$Y_{n,i} = \bar{F}_n(t_{n,i}) + \int_{t_n}^{t_{n,i}} k(t_{n,i} - \tau)u_n(\tau)d\tau \quad i = 1, \dots, m \quad (4.2.2)$$

where

$$\bar{F}_n(t_{n,i}) = f(t_{n,i}) + \int_0^{t_n} k(t_{n,i} - \tau)u(\tau)d\tau. \quad (4.2.3)$$

and

$$u_n(t_n + \theta h) = \sum_{j=1}^m L_j(\theta)Y_{n,j} \quad \theta \in (0, 1] \quad n = 0, \dots, N_t - 1. \quad (4.2.4)$$

In order to obtain a discrete collocation method we have to choose suitable quadrature formulas to approximate the integrals in (4.2.2) and (4.2.3). For the peculiarity of the equation (4.1.1), the quadrature formulas have to involve the evaluations of the Laplace transform of the kernel. For the computation of (4.2.2) and (4.2.3) we will use a generalization of the quadrature formulas

described in Section 3.5, and then we will show that the constructed numerical methods have a computational cost of $O(N_t \log N_t)$ instead of $O(N_t^2)$ operations, their order of accuracy is the same as the corresponding exact collocation method.

4.2.1 Fast computation of the lag terms

First we will illustrate how to approximate the integral arising in the lag-terms (4.2.3).

As shown in Section 3.5.2, in order to split the integral over $[0, t_n]$ in (4.2.3), we consider L as the smallest integer for which $t_{n+1} < 2B^L h$ and, for $l = 1, 2, \dots, L - 1$, we determine the integer $q_l \geq 1$ such that $\tau_l = q_l B^l h$ satisfies $t_{n+1} - \tau_l \in [B^l h, (2B^l - 1)h]$, with $\tau_0 = t_n$ and $\tau_L = 0$. Thus the integral over $[0, t_n]$ in (4.2.3) is split into the following way

$$\bar{F}_n(t_{n,i}) = f(t_{n,i}) + \sum_{l=1}^L \int_{\tau_l}^{\tau_{l-1}} k(t_{n,i} - \tau) u(\tau) d\tau. \quad (4.2.5)$$

Since we do not know the evaluations of the kernel, but of its Laplace transform, we use the formula (3.5.2) for approximating the lag terms. In order to apply this formula it is necessary that the condition $[t - a, t - b] \subseteq I_l$ is verified, where I_l is an opportune chosen interval to which corresponds a Talbot contour Γ_l for the inverse Laplace transform approximation formula. In our case we can not use the same I_l of the formula (3.4.5) since $[t_{n,i} - \tau_{l-1}, t_{n,i} - \tau_l] \not\subseteq I_l = [B^{l-1}h, (2B^l - 1)h]$. Thus the intervals I_l then need to be modified according to $\tilde{I}_l = [B^{l-1}h + (c_1 - 1)h, (2B^l - 1)h + (c_m - 1)h]$.

Now it is easy to verify that $[t_{n,i} - \tau_{l-1}, t_{n,i} - \tau_l] \subseteq \tilde{I}_l$. As a matter of fact, being by construction $[t_{n+1} - \tau_{l-1}, t_{n+1} - \tau_l] \subseteq I_l$, it follows that

$$[t_{n,i} - \tau_{l-1}, t_{n,i} - \tau_l] = [t_{n+1} - \tau_{l-1} + (c_i - 1)h, t_{n+1} - \tau_l + (c_i - 1)h] \subseteq \tilde{I}_l. \quad (4.2.6)$$

Thus it is possible to use the formula (3.5.2), for approximating each integral over $[\tau_l, \tau_{l-1}]$, obtaining

$$\int_{\tau_l}^{\tau_{l-1}} k(t_{n,i} - \tau)u(\tau) \approx \sum_{j=-N}^N \omega_j^{(l)} K(\lambda_j^{(l)}) e^{(t_{n,i} - \tau_{l-1})\lambda_j^{(l)}} z(\tau_{l-1}, \tau_l, \lambda_j^{(l)}) \quad (4.2.7)$$

and thus $\bar{F}_n(t_{n,i}) \approx \bar{F}_{n,i}$

$$\bar{F}_{n,i} := f(t_{n,i}) + \sum_{l=1}^L \sum_{j=-N}^N \omega_j^{(l)} K(\lambda_j^{(l)}) e^{(t_{n,i} - \tau_{l-1})\lambda_j^{(l)}} z(\tau_{l-1}, \tau_l, \lambda_j^{(l)}) \quad i = 1, \dots, m \quad (4.2.8)$$

where $z(\tau_{l-1}, \tau_l, \lambda_j^{(l)}) = \int_{\tau_{l-1}}^{\tau_l} e^{(\tau_l - \tau)\lambda_j^{(l)}} u(\tau) d\tau$.

If we denote by L_t the total number of different Talbot contours, we can observe that by construction

$$L \leq L_t \leq \log_B(N_t).$$

A direct implementation of the formula (4.2.8) would still lead to a computational cost of $O(N_t^2)$. The idea proposed in [68] (and described in Section 3.5.2) to reduce the computational cost was based on a new organization in the computation of the function z at each time step, which could exploit its evaluations at the previous time steps. In order to reach the same goal we split the interval $[\tau_l, \tau_{l-1}]$ in subintervals $[\tau_l + t_k, \tau_l + t_{k+1}]$, $k = 0, \dots, \bar{n} - 1$, $\bar{n} = (\tau_{l-1} - \tau_l)/h$. We denote with $z_k := z(\tau_l + t_k, \tau_l, \lambda_j^{(l)})$ the integral over $[\tau_l, \tau_l + t_k]$, thus

$$\begin{aligned} z_{k+1} &= e^{h\lambda_j^{(l)}} z_k + \int_{\tau_l + t_k}^{\tau_l + t_{k+1}} e^{(\tau_l + t_{k+1} - \tau)h\lambda_j^{(l)}} u(\tau) d\tau \\ &= e^{h\lambda_j^{(l)}} z_k + h \int_0^1 e^{(1-\theta)h\lambda_j^{(l)}} u(\tau_l + t_k + \theta h) d\theta \quad k = 0, \dots, \bar{n} - 1. \end{aligned} \quad (4.2.9)$$

Denoting by $\tilde{Y}_{k,r} := u(\tau_l + t_{k,r})$, the local representation (4.2.4) leads to

$$u(\tau_l + t_k + \theta h) = \sum_{r=1}^m L_r(\theta) \tilde{Y}_{k,r} \quad \theta \in (0, 1]. \quad (4.2.10)$$

Now it is easy to prove the following property of the Lagrange polynomials which is useful for computing the integrals in (4.2.9).

Property 4.1. *The Lagrange polynomials may be expanded in powers of θ according to*

$$L_r(\theta) = \prod_{\substack{j=1 \\ j \neq r}}^m \frac{\theta - c_j}{c_r - c_j} = B_r \sum_{s=0}^{m-1} (-1)^{m-1-s} \sigma_{r,m-1-s} \theta^s \quad (4.2.11)$$

where

$$B_r = \prod_{\substack{s=1 \\ r \neq s}}^m \frac{1}{c_r - c_s} \quad (4.2.12)$$

and $\sigma_{r,i}$ represents the sum of all possible products of i distinct elements taken in the set $\{c_i\}_{i=1, i \neq r}^m$:

$$\sigma_{r,0} = 1, \quad \sigma_{r,i} = \sum_{\substack{n_1 < \dots < n_i = 1 \\ n_k \neq r}}^m c_{n_1} c_{n_2} \dots c_{n_i}. \quad (4.2.13)$$

By substituting (4.2.10)-(4.2.11) in (4.2.9), we obtain the following one step formula for the evaluation the function z in the mesh points from τ_l to τ_{l-1} :

$$\begin{cases} z_0 = 0, \\ z_{k+1} = e^{h\lambda_j^{(l)}} \left(z_k + h \sum_{s=0}^{m-1} I_s(h\lambda_j^{(l)}) \sum_{r=1}^m (-1)^{m-1-s} \sigma_{r,m-1-s} B_r \tilde{Y}_{k,r} \right) \\ k = 0, \dots, \bar{n} - 1, \end{cases} \quad (4.2.14)$$

where B_r and $\sigma_{r,i}$ are respectively given by (4.2.12) and (4.2.13), and the integrals $I_s(h\lambda) = \int_0^1 e^{-\theta h\lambda} \theta^s d\theta$ $s = 0, \dots, m-1$, can be computed exactly via the following recursive formula

$$\begin{cases} I_0(h\lambda) = \frac{1-e^{-h\lambda}}{h\lambda} \\ I_s(h\lambda) = -\frac{e^{-h\lambda}}{h\lambda} + \frac{s}{h\lambda} I_{s-1}(h\lambda) \quad s = 1, \dots, m-1. \end{cases} \quad (4.2.15)$$

Remark 4.1. The formula (4.2.9) differs from (3.5.8) because here $u(t)$ represents a piecewise polynomial interpolant of degree $m-1$ instead of the piecewise linear approximation. In the special case of $m = 2$ and collocation parameters $c_1 = 0$, $c_2 = 1$, the function $u(t)$ given by (4.2.10) reduces to the piecewise linear approximation (3.5.7) and the recursive formula (4.2.14) coincides with (3.5.9), provided that we set $a = \tau_l$, $\lambda = \lambda_j^{(l)}$.

Now we analyse the computational cost of the lag terms computation given by (4.2.8) and (4.2.14).

As in Section 3.5.2, we can advance the values (4.2.14) of z by one step for all required values $\lambda_j^{(l)}$ on all Talbot contours in every time step $t_n \rightarrow t_{n+1}$. Note that the function z in (4.2.8) does not depend on i , so we have to evaluate it only one time at each step $t_n \rightarrow t_{n+1}$ independently the number m of collocation points. Observing that the computation of z_{k+1} through (4.2.14) only requires the value z_k of z at the previous step and the values $\tilde{Y}_{k,r} = u(\tau_l + t_{k,r})$ of the polynomial approximation in the interval $[\tau_l + t_k, \tau_l + t_{k+1}]$, we do not need to keep in memory all the past values, thus we need a memory requirement of $O(\log_B N_t)$.

In this way we have obtained the same computational scheme of Section 3.5.2, whose computational cost is proportional to MN_tL_t , where N_t is the number of time steps, $L_t \leq \log_B N_t$ is the number of different contours and M is the number of quadrature points on each contour. As it will be specified in Remark 4.2 the method proposed in Section 3.5.2 can be recognized as a fast discrete collocation method with collocation parameters $c_1 = 0$ and $c_2 = 1$. On the contrary in our case we have m collocation parameters, hence the lag terms (4.2.8) have to be computed for $i = 1, \dots, m$ and the computation of (4.2.14) requires a double summation over m . Thus we obtain the same computational cost of the method proposed in Section 3.5.2 except for a multiplicative factor of

$m+m^2$. As m is fixed and independent of N_t (generally in concrete applications it is $m \leq 4$, as observed in Remark 2.1), the total cost of the lag terms computation is $O(N_t \log_B N_t)$.

4.2.2 Computation of the increment term

In this section we develop the formulas for the approximation of the increment term in (4.2.2). As

$$\int_{t_n}^{t_{n,i}} k(t_{n,i} - \tau)u_n(\tau)d\tau = \int_0^{c_i h} k(c_i h - \tau)u_n(t_n + \tau)d\tau \quad i = 1, \dots, m,$$

substituting the expression (4.2.4) for the piecewise polynomial $u(t)$ and expanding the Lagrange fundamental polynomials as powers of τ , we obtain

$$\begin{aligned} & \int_{t_n}^{t_{n,i}} k(t_{n,i} - \tau)u_n(\tau)d\tau = & (4.2.16) \\ & = \sum_{r=1}^m B_r Y_{n,r} \sum_{s=0}^{m-1} \frac{(-1)^{m-1-s} \sigma_{r,m-1-s}}{h^s} \int_0^{c_i h} k(c_i h - \tau) \cdot \tau^s d\tau \quad i = 1, \dots, m. \end{aligned}$$

where B_r and $\sigma_{r,n}$ are respectively given by (4.2.12) and (4.2.13).

Since

$$\int_0^{c_i h} k(c_i h - \tau) \cdot \tau^s d\tau = q(c_i h)$$

where $q(t)$ represents the inverse Laplace transform of $Q(\lambda) = K(\lambda) \frac{s!}{\lambda^{s+1}}$, the approximation (3.4.7) leads to

$$\int_0^{c_i h} k(c_i h - \tau) \cdot \tau^s d\tau \approx \sum_{j=-N}^N \omega_j \frac{s!}{\lambda_j^{s+1}} K(\lambda_j) e^{c_i h \lambda_j} =: \Phi_{i,s} \quad (4.2.17)$$

where the weights ω_j and nodes λ_j are associated to the Talbot contour Γ_0 that corresponds to the interval $\tilde{I}_0 = [0, h]$.

Finally, the increment term approximation becomes

$$\begin{aligned} & \int_{t_n}^{t_{n,i}} k(t_{n,i} - \tau) u_n(\tau) d\tau \approx \\ & \approx \sum_{r=1}^m \left(B_r \sum_{s=0}^{m-1} \frac{(-1)^{m-1-s} \sigma_{r,m-1-s} \Phi_{i,s}}{h^s} \right) Y_{n,r} \quad i = 1, \dots, m. \end{aligned} \quad (4.2.18)$$

4.2.3 Determination of the approximate solution

Previously we determined the formulas (4.2.8), (4.2.14) to approximate the lag term (4.2.3) and (4.2.17)-(4.2.18) to approximate the increment term in (4.2.2), thus the approximate solution of (4.2.1) is

$$\bar{u}_n(t_n + \theta h) = \sum_{j=1}^m L_j(\theta) \bar{Y}_{n,j} \quad \theta \in (0, 1] \quad n = 0, \dots, N_t - 1, \quad (4.2.19)$$

where $\bar{Y}_{n,i}$ are given by the solution of the linear system

$$\bar{Y}_{n,i} = \bar{F}_{n,i} + \sum_{r=1}^m d_{i,r} \bar{Y}_{n,r} \quad i = 1, \dots, m \quad (4.2.20)$$

with

$$d_{i,r} = B_r \sum_{s=0}^{m-1} \frac{(-1)^{m-1-s} \sigma_{r,m-1-s} \Phi_{i,s}}{h^s}. \quad (4.2.21)$$

Setting $\mathbf{D} = (d_{i,r})_{i,r=1,\dots,m}$, $\bar{\mathbf{Y}}_{\mathbf{n}} = (\bar{Y}_{n,1}, \dots, \bar{Y}_{n,m})^{\mathbf{T}}$, and $\bar{\mathbf{F}}_{\mathbf{n}} = (\bar{F}_{n,1}, \dots, \bar{F}_{n,m})^{\mathbf{T}}$, the linear system (4.2.20) can be written in a matrix form as

$$(\mathbf{I} - \mathbf{D}) \bar{\mathbf{Y}}_{\mathbf{n}} = \bar{\mathbf{F}}_{\mathbf{n}} \quad n = 0, \dots, N_t - 1, \quad (4.2.22)$$

where \mathbf{I} denotes the identity matrix of order m .

Obviously, the approximate solution of (4.2.1) at the mesh points is obtained by setting $\theta = 1$ in the expression (4.2.19).

Remark 4.2. A special case occurs when $c_1 = 0$, $c_m = 1$, because, setting $\bar{Y}_{n,1} = \bar{Y}_{n-1,m}$, the linear system (4.2.20) has to be solved for $\bar{Y}_{n,i}$, $i = 2, \dots, m$,

thus becoming of dimension $m - 1$. In particular, if $m = 2$, taking into account what observed in Remark 4.1, the (4.2.8), (4.2.19)-(4.2.21) reproduce the approximate solution of (4.2.1) obtained by applying a direct quadrature method which straightforwardly employs the fast quadrature formulas (3.5.11),(3.5.14), illustrated in Section 3.5.

The linear system (4.2.22) is characterized by a nonstructured full coefficient matrix $\mathbf{I} - \mathbf{D}$ of order m . What's more we can prove the following theorem

Theorem 4.1. *Assume that f and k in the VIE (4.2.1) are continuous. Then there exists an $\bar{h} > 0$ so that for any mesh I_h with $h \in (0, \bar{h})$ the linear system (4.2.22) has a unique solution.*

Proof. We have to prove that the inverse of the matrix $\mathbf{I} - \mathbf{D}$ exists for h sufficiently small. It is known that if $\|\mathbf{D}\| < 1$ then the inverse of the matrix $\mathbf{I} - \mathbf{D}$ exists and is bounded in norm (see [4], p. 492). Let us rewrite \mathbf{D} as $\mathbf{D} = h\mathbf{D}_1$, with

$$(\mathbf{D}_1)_{i,r} = B_r \sum_{s=0}^{m-1} \frac{(-1)^{m-1-s} \sigma_{r,m-1-s}}{h^{s+1}} \Phi_{i,s}.$$

We have only to prove that the elements of the matrix \mathbf{D}_1 are all bounded since it is sufficient to choose $h < \frac{1}{\|\mathbf{D}_1\|}$. The coefficients B_r and $\sigma_{r,m-1-s}$, given by (4.2.12) and (4.2.13), depend on $\{c_i\}$ which are bounded. Thus it is necessary to prove that $\frac{\Phi_{i,s}}{h^{s+1}}$ are bounded for some matrix norm for $i = 1, \dots, m$, and $s = 0, \dots, m - 1$. Since the kernel k is supposed to be continuous, the integrals $\int_0^{c_i} k(c_i h - \theta h) \theta^s d\theta$ are all bounded.

Moreover, with the change of variable $\tau = \theta h$ we obtain

$$\left| \frac{\Phi_{i,s}}{h^{s+1}} - \int_0^{c_i} k(c_i h - \theta h) \theta^s d\theta \right| = \left| \frac{\Phi_{i,s} - \int_0^{c_i h} k(c_i h - \tau) \tau^s d\tau}{h^{s+1}} \right| \leq \frac{c_1 e^{-c\sqrt{M}}}{h^{s+1}},$$

since, according to the formula (4.2.17), $\left| \Phi_{i,s} - \int_0^{c_i h} k(c_i h - \tau) \tau^s d\tau \right|$ represents the error of the inverse Laplace transform approximation formula (3.4.7). Now it is possible to choose the number of points M on the Talbot contour in order that $c_1 e^{-c\sqrt{M}} \leq c_2 h^m$ is verified. Since $h < 1$ we obtain

$$\left| \frac{\Phi_{i,s}}{h^{s+1}} \right| \leq c_2 + \left| \int_0^{c_i} k(c_i h - \theta h) \theta^s d\theta \right|$$

and thus the thesis holds. \square

As observed in Remark 2.1, in concrete applications the value of m does not usually exceed $m = 4$. This allows us to solve the linear system (4.2.22) with a direct method obtaining a computational cost over N_t time steps proportional to $m^3 N_t$, and thus of $O(N_t)$.

Remark 4.3. Observe that when the VIE is of the Hammerstein type (4.1.1), then (4.2.19) still holds where $\bar{Y}_{n,i}$ are now the solution of the nonlinear system

$$\bar{Y}_{n,i} = \bar{F}_{n,i} + \sum_{r=1}^m d_{i,r} g(\bar{Y}_{n,i}) \quad i = 1, \dots, m, \quad (4.2.23)$$

where $\bar{F}_{n,i}$ is given by (4.2.8), (4.2.14), by replacing $\tilde{Y}_{k,r}$ with $g(\tilde{Y}_{k,r})$. Such system can be solved by an iterative method with a computational cost over N_t time steps proportional to $m^2 k N_t$, where k represents the number of iterations required. Thus, as in the linear case, the cost for solving the nonlinear system is $O(N_t)$.

Remark 4.4 (Computational cost). As concerns the total computational cost of the fast collocation method, we observe that it consists of the cost of the lag terms computation and the cost of the increment term, which includes the resolution of the (non)linear system (4.2.22)-(4.2.23). From what we mentioned above and in Section 4.2.1, it follows that the total cost of the fast collocation method is $O(N_t \log N_t)$.

4.3 Convergence analysis

The fast collocation method described in Section 4.2 is a discretized collocation method based on the quadrature formula (4.2.8) for the lag term and (4.2.18) for the increment term. For the discussion of the order of convergence of the classical collocation methods refer to Theorems 2.2 and 2.3. In order to establish the order of convergence of the fast discrete collocation method, we need the following result:

Lemma 4.1. *Let*

$$E_{n,l}^{(i)}[u] = \int_{\tau_l}^{\tau_{l-1}} k(t_{n,i} - \tau)u(\tau) - \sum_{j=-N}^N \omega_j^{(l)} K(\lambda_j^{(l)}) e^{(t_{n,i} - \tau_{l-1})\lambda_j^{(l)}} z(\tau_{l-1}, \tau_l, \lambda_j^{(l)})$$

$$E_{n,n}^{(i)}[u] = \int_{t_n}^{t_{n,i}} k(t_{n,i} - \tau)u_n(\tau)d\tau - \sum_{r=1}^m \left(B_r \sum_{s=0}^{m-1} \frac{(-1)^{m-1-s} \sigma_{r,m-1-s} \Phi_{i,s}}{h^s} \right) Y_{n,r}$$

respectively denote the error of the quadrature formulas (4.2.7) and (4.2.18).

Then

$$\left| E_{n,l}^{(i)}[u] \right| = O(e^{-c\sqrt{M}}) \quad (4.3.1)$$

$$\left| E_{n,n}^{(i)}[u] \right| = O(e^{-c\sqrt{M}}). \quad (4.3.2)$$

Proof. It immediately follows from (3.5.4) that $\left| E_{n,l}^{(i)}[u] \right| = O(e^{-c\sqrt{M}})$. Moreover, observing that $\Phi_{i,s}$ are obtained by a direct application of (3.4.7) to the function $\frac{s!}{\lambda^{s+1}}K(\lambda)$, we obtain $\left| E_{n,n}^{(i)}[u] \right| = O(e^{-c\sqrt{M}})$. \square

We are now ready to prove the following theorem

Theorem 4.2. *Let $\bar{u}(t)$ be the approximate solution of (4.2.1) obtained through the discrete collocation method (4.2.8), (4.2.19)-(4.2.20). Then the error $\bar{e}(t) = y(t) - \bar{u}(t)$ satisfies*

$$\|\bar{e}\|_{\infty} = O(h^m) \quad (4.3.3)$$

for every choice of the collocation parameters $0 \leq c_1 < \dots < c_m \leq 1$ with sufficiently large number M of points on the Talbot contour.

Proof. Denoting by u the solution of the exact collocation method we have

$$|\bar{e}(t)| \leq |y(t) - u(t)| + |u(t) - \bar{u}(t)| = |e(t)| + |\epsilon(t)| \quad t \in [0, T].$$

For Theorem 2.2 we have that $|e(t)| \leq C_1 h^m$ uniformly on I .

Let $\epsilon_n(t)$ denote the restriction of $\epsilon(t)$ to the subinterval $(t_n, t_{n+1}]$, $n = 0, \dots, N_t - 1$. Subtracting (4.2.20) from (4.2.2), we obtain

$$\epsilon_n(t_{n,i}) = \sum_{r=1}^m d_{i,r} \epsilon_n(t_{n,r}) + h \sum_{k=0}^{n-1} \sum_{r=1}^m q_{i,r}^{(n,k)} \epsilon_k(t_{k,r}) + E_{n,n}^{(i)}[u] + \sum_{l=1}^L E_{n,l}^{(i)}[u] \quad i = 1, \dots, m \quad (4.3.4)$$

where

$$q_{i,r}^{(n,k)} = \sum_{j=-N}^N \omega_j^{(l)} K(\lambda_j^{(l)}) \int_0^1 e^{(n-k+c_i-\theta)h\lambda_j^{(l)}} L_r(\theta) d\theta, \quad k = 0, \dots, n-1, \quad i, r = 1, \dots, m.$$

By writing down equation (4.3.4) in vectorial form we have

$$(\mathbf{I} - \mathbf{D})\eta_n = h \sum_{k=0}^{n-1} \mathbf{Q}^{(n,k)} \eta_k + \mathbf{r}_{n,n} + \sum_{l=1}^L \mathbf{r}_{n,l} \quad (4.3.5)$$

where $\eta_k = (\epsilon_k(t_{k,1}), \dots, \epsilon_k(t_{k,m}))^T$, $\mathbf{r}_{n,l} = (E_{n,l}^{(1)}[u], \dots, E_{n,l}^{(m)}[u])^T$ and $\mathbf{Q}^{(n,k)}$ is a square matrix whose elements are $q_{i,r}^{(n,k)}$ for $i, r = 1, \dots, m$. It follows from Lemma 4.1 that $\|\mathbf{r}_{n,n}\|_1 \leq C_2 e^{-c\sqrt{M}}$ and $\|\mathbf{r}_{n,l}\|_1 \leq C_3 e^{-c\sqrt{M}}$. In Theorem 4.1 we proved that, for sufficiently small $h > 0$, the inverse $(\mathbf{I} - \mathbf{D})^{-1}$ exists and is bounded, thus there exists a constant D_0 such that $\|(\mathbf{I} - \mathbf{D})^{-1}\|_1 \leq D_0$. Analogously there exists a constant D_1 such that that $\|\mathbf{Q}^{(n,k)}\|_1 \leq D_1$. Thus (4.3.5) leads to

$$\|\eta_n\|_1 \leq D_0 \left\{ h D_1 \sum_{k=0}^{n-1} \|\eta_k\|_1 + C_2 e^{-c\sqrt{M}} + L C_3 e^{-c\sqrt{M}} \right\}. \quad (4.3.6)$$

This represents a discrete Gronwall inequality for $\|\eta_n\|_1$,

$$\|\eta_n\|_1 \leq h C_4 \sum_{k=0}^{n-1} \|\eta_k\|_1 + C_5 e^{-c\sqrt{M}}, \quad (4.3.7)$$

where $C_4 = D_0 D_1$, $C_5 = D_0(C_2 + LC_3)$. Thus, by Theorem 2.6, it follows that $\|\eta_n\|_1 \leq Q_1 e^{-c\sqrt{M}}$. To complete the proof if $\Lambda_m := \max \left\{ \sum_{i=1}^m |L_i(\nu)| : 0 \leq \nu \leq 1 \right\}$ denote the Lebesgue constant associated with the collocation parameters $\{c_i\}$, then we have

$$|\epsilon_n(t_n + \nu h)| \leq \sum_{i=1}^m |\epsilon_n(t_{n,i})| |L_i(\nu)| \leq \|\eta_n\|_1 \Lambda_m \leq Q e^{-c\sqrt{M}}$$

with $Q = Q_1 \Lambda_m$.

This is equivalent to $\|\epsilon\|_\infty \leq Q e^{-c\sqrt{M}}$ and hence, by choosing M such that $Q e^{-c\sqrt{M}} \leq C_1 h^m$, the assertion of Theorem 4.2 now follows. \square

Since the error due to the approximation of the inverse Laplace transform decreases exponentially with M it is sufficient to fix a not too high number of points on the Talbot contour in order to make this error negligible. Moreover if $M \rightarrow +\infty$, that is the formula for approximating the inverse Laplace transform is exact, we return to the exact collocation method.

Obviously it is possible to achieve local superconvergence at the mesh points by opportunely choosing the collocation parameters $\{c_i\}$, and sufficiently large M , in analogy to the Theorem 2.7

Corollary 4.1. *Let $f, k \in C^{2m-v}$, with $v \in \{0, 1, 2\}$ with $m \geq \lfloor v/2 \rfloor + 1$,*

(a) If the collocation parameters are the Radau II points for $(0, 1]$, then for $v = 1$

$$\max_{t_n \in I_h} |\bar{e}(t_n)| = O(h^{2m-1})$$

(b) If the collocation parameters are the Lobatto points for $[0, 1]$, then we obtain then for $v = 2$

$$\max_{t_n \in I_h} |\bar{e}(t_n)| = O(h^{2m-2})$$

(c) If the first $m - 1$ collocation parameters are the Gauss points for $(0, 1)$ and $c_m = 1$, then for $v = 2$

$$\max_{t_n \in I_h} |\bar{e}(t_n)| = O(h^{2m-2}).$$

4.4 Numerical results

In this section we illustrate the performances of the fast m -point collocation methods in order to validate the convergence results proved in Section 4.3 and the reduction of the computational cost to $\mathcal{O}(N_t \log_B N_t)$ (Remark 4.3). The fast collocation methods have been implemented in MATLAB and the numerical experiments have been performed on different test examples. Here we report the results obtained on the three test problems:

- the linear Volterra integral equation taken from [18]:

$$y(t) = e^t + \int_0^t 2 \cos(t - \tau) y(\tau) d\tau \quad t \in [0, 1], \quad (4.4.1)$$

with $K(s) = \frac{2s}{1+s^2}$ and exact solution $y(t) = e^t(1+t)^2$;

- the nonlinear equation given in [52], arising in the analysis of neural networks with post inhibitory rebound:

$$y(t) = 1 + \int_0^t (t - \tau)^3 (4 - t + \tau) e^{-t+\tau} \frac{y^4(\tau)}{1 + 2y^2(\tau) + 2y^4(\tau)} d\tau, \quad t \in [0, 10], \quad (4.4.2)$$

with $K(s) = \frac{24s}{(1+s)^5}$ and reference solution $y(10) = 1.25995582337233$;

- the nonlinear Abel equation given in [69] and described in Example 1.1, arising in chemical absorption kinetics:

$$y(t) = y_0 - \frac{\alpha}{\beta} \int_0^t k(t - \tau) \frac{y(\tau)}{1 + y(\tau)^{0.75}} d\tau, \quad t \in [0, 1], \quad (4.4.3)$$

with $y_0 = 10$, $\alpha = 1$, $\beta = 10^{-2}$, $K(s) = \frac{1}{\sqrt{s/\beta \tanh(\sqrt{s/\beta})}} - \frac{1}{s/\beta}$ and reference solution $y(1) = 1.65087150782378$.

The reference solutions for problems (4.4.2) and (4.4.3) have been obtained numerically by using different codes with very stringent tolerances.

Remark 4.5. We remind that, in order to apply the fast collocation methods, we have to choose a family of Talbot contours. This choice is done according to Section 3.4.3. Namely, in problem (4.4.1) $K(s)$ has singularities in $s = \pm i$, thus it corresponds to Example 3.1. In problems (4.4.2), (4.4.3) $K(s)$ has only one singularity respectively at $s = -1$ and $s = 0$, thus it corresponds to Example 3.3. The number $M = 2N + 1$ of quadrature points on each Talbot contour is chosen to be $M = 501$.

4.4.1 Convergence

The following fast collocation methods for the approximation of the solution of each equation have been used, where p denotes the order of the method, according to Corollary 4.1:

- $L2$: 2 points Lobatto collocation ($c_1 = 0, c_2 = 1$), $p = 2$
- $R2$: 2 points Radau collocation ($c_1 = 1/3, c_2 = 1$), $p = 3$
- $L3$: 3 points Lobatto collocation ($c_1 = 0, c_2 = 1/2, c_3 = 1$), $p = 4$
- $R3$: 3 points Radau collocation ($c_1 = \frac{4-\sqrt{6}}{10}, c_2 = \frac{4+\sqrt{6}}{10}, c_3 = 1$), $p = 5$
- $G4$: 3 points Gauss collocation + $c_4 = 1$
 $(c_1 = \frac{5-\sqrt{15}}{10}, c_2 = 1/2, c_3 = \frac{5+\sqrt{15}}{10})$, $p = 6$

The accuracy is defined by the number of correct significant digits cd at the end point, i.e. the value

$$cd := -\log_{10} (|y(T) - y_{N_t}|/|y(T)|)$$

The order of the method is estimated with the formula $p(h) = \frac{cd(h) - cd(2h)}{\log_{10} 2}$ for a fixed h . For each test problem we plot the number of cd versus N_t .

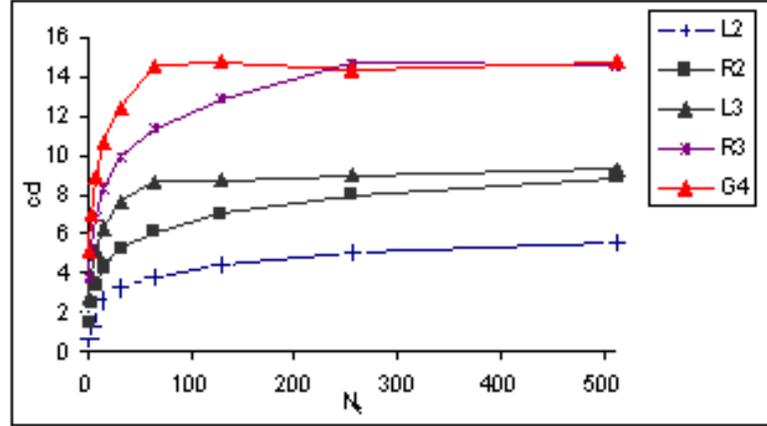


Figure 4.1: Number of correct significant digits for problem (4.4.1)

Table 4.1: Number of correct significant digits for problem (4.4.1) at $t=T=1$.

method	$N_t = 8$	$N_t = 16$	$N_t = 32$	$N_t = 64$	$p(h = \frac{1}{64})$
L2	2.01	2.61	3.22	3.82	1.99
R2	3.41	4.33	5.24	6.15	3.02
L3	5.11	6.33	7.65	8.66	3.36
R3	6.83	8.34	9.85	11.36	5.02
G4	8.81	10.63	12.44	14.51	6.88

Table 4.2: Number of correct significant digits for problem (4.4.2) at $t=T=10$.

method	$N_t = 32$	$N_t = 64$	$N_t = 128$	$N_t = 256$	$p(h = \frac{10}{256})$
L2	3.00	3.61	4.21	4.81	1.99
R2	5.11	6.01	6.92	7.83	3.02
L3	6.21	7.49	8.71	9.92	4.02
R3	7.06	8.54	10.03	11.54	5.02
G4	8.21	10.00	11.80	13.70	6.31

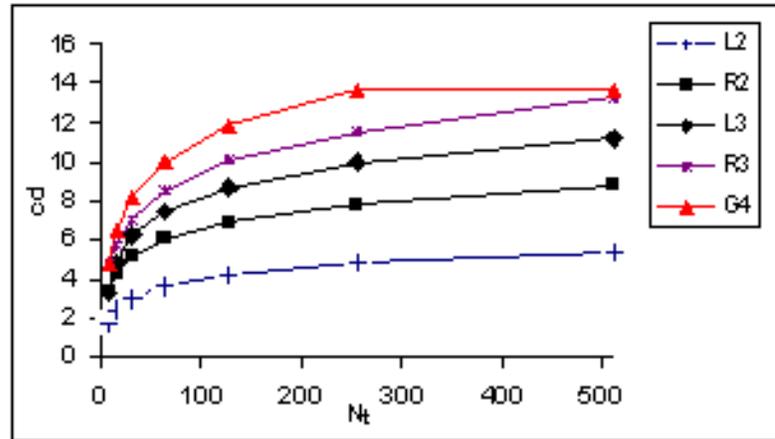


Figure 4.2: Number of correct significant digits for problem (4.4.2)

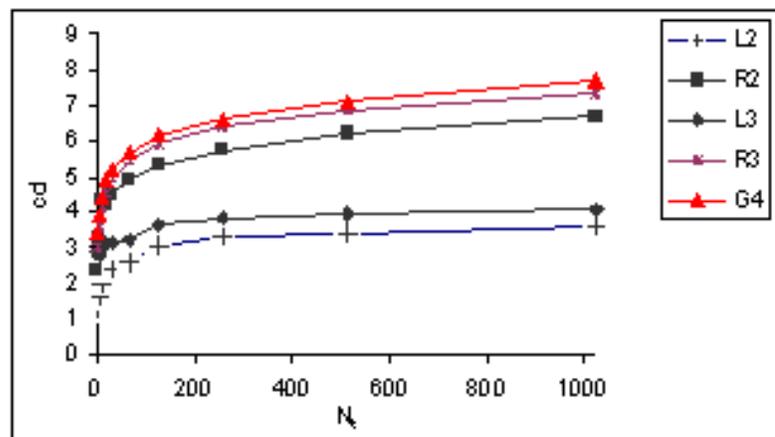


Figure 4.3: Number of correct significant digits for problem (4.4.3)

Table 4.3: *Number of correct significant digits for problem (4.4.3) at $t=T=1$.*

method	$N_t = 128$	$N_t = 256$	$N_t = 512$	$N_t = 1024$
<i>L2</i>	3.07	3.32	3.42	3.59
<i>R2</i>	5.33	5.77	6.21	6.67
<i>L3</i>	3.64	3.86	3.94	4.09
<i>R3</i>	5.91	6.37	6.84	7.36
<i>G4</i>	6.14	6.60	7.09	7.70

The Figures 4.1-4.2 and the results listed in Tables 4.1-4.2 clearly show that our methods produce the desired order according to the Corollary 4.1, since the functions involved are sufficiently regular. Only for equation (4.4.3), as Figure 4.3 and Table 4.3 show, the methods do not achieve the order of convergence of Corollary 4.1, since the inverse Laplace transform of $K(s)$ has a weak singularity of Abel type. This is not a surprising behaviour since we know from Section 2.2.2 that even a classical collocation method produces a drastic fall in its order of convergence when it is applied to Abel VIEs.

4.4.2 Computational cost

In order to verify that our methods have a computational cost of order $\mathcal{O}(N_t \log_B N_t)$, in Figure 4.4 we plot the cpu-time in seconds versus N_t obtained from an Intel Pentium 4/3,2 GHz. The experiments have been made on the test problem (4.4.1), enlarging the integration interval to $t \in [0, 200]$. The rhombuses correspond to the fast collocation method R3, and in the picture it clearly appears that the computational cost is between the cost of an algorithm of order $\mathcal{O}(N_t)$ and one of order $\mathcal{O}(N_t^2)$. More precisely, our algorithm perfectly follows the behaviour of the dashed line, which represents the function $C N_t \log_B N_t$.

In order to show the gain in efficiency of the fast collocation method versus

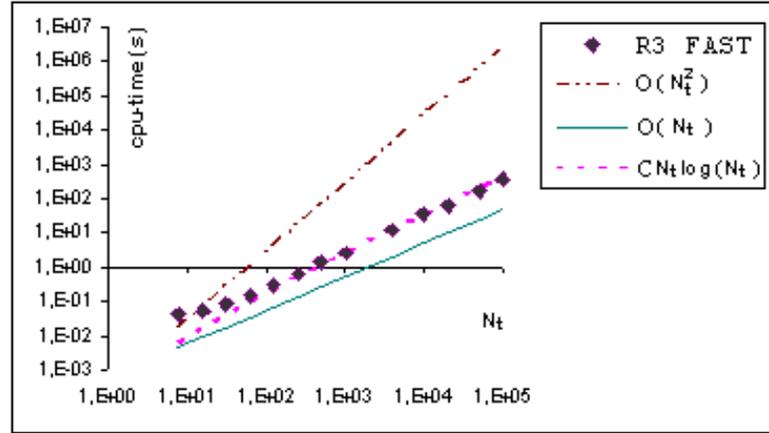


Figure 4.4: Computational cost

the classical one, in Table 4.4 we compare the cpu-time obtained by the fast collocation method with that of the existing solver COLVI2 (see [9],[10]), using the same collocation parameters of R3. The corresponding plot is reported in Figure 4.5. We choose COLVI2 because it is a FORTRAN implementation of an m -point classical collocation method and it is one of the most used among the few available software for VIEs. The picture shows that, as we expected, the cpu-time performed by COLVI2 is clearly proportional to N_t^2 . We can observe that at the beginning COLVI2 is more efficient, but when N_t increases (more precisely when $N_t = 4 \cdot 10^3$, corresponding to a stepsize $h = 1/20$), then the efficiency of the fast collocation method becomes superior. Similar results occur with different test problems. We would like to underline that the MATLAB code implementing the fast collocation method is only a prototypal one. We intend to implement it in a compilative programming language expecting more satisfactory results in terms of efficiency. As a matter

Table 4.4: *Cpu-time in seconds for (4.4.1) with $T=200$.*

N_t	R3 FAST	R3 COLVI2
128	0.29	0.02
256	0.59	0.06
512	1.46	0.23
1024	2.55	0.92
4.00E+03	11.75	14.04
1.00E+04	35.24	86.80
2.00E+04	63.03	348.50
5.00E+04	168.58	2159.44
1.00E+05	356.46	8716.53

Table 4.5: *Cpu-time in seconds for (4.4.1) with $T=200$.*

N_t	R3 FAST	R3 CLASSICAL
16	0.05	0.02
32	0.08	0.05
64	0.15	0.19
128	0.29	0.70
256	0.59	2.77
512	1.46	11.00
1024	2.55	43.71
4.00E+03	11.75	828.2910
1.00E+04	35.24	5918.27

of fact in Table 4.5 and in Figure 4.6 we compare the cpu-time of the fast collocation method R3 with that of a MATLAB implementation of the classical collocation method R3, when applied to test problem (4.4.1) with $T = 200$. In this case we can observe that the fast method is more efficient than the classical one already when $N_t = 64$.

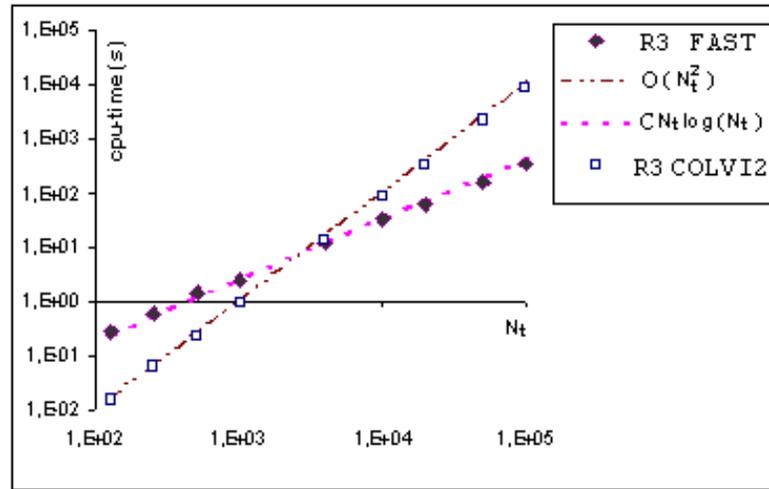


Figure 4.5: Fast collocation vs COLVI2

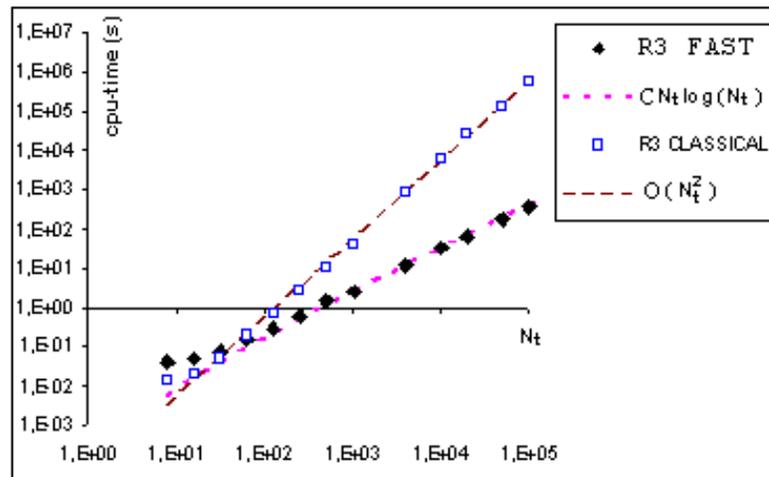


Figure 4.6: Fast collocation vs classical collocation

4.5 Concluding remarks

In this chapter we have developed fast discrete collocation methods (4.2.8), (4.2.14)–(4.2.15), (4.2.19)–(4.2.21) for the equations (4.1.1). Such methods taking into account the knowledge of the Laplace transform of the kernel instead of the kernel itself, differ from classical numerical methods since they involve the evaluations of $K(s)$. The quadrature formulas used to discretize the integrals appearing in the exact collocation method are a generalization of those introduced in [68] and illustrated in Section 3.5.

We proved that our methods can be implemented with the same scheme of such quadrature formulas which leads the computational cost to $O(N_t \log N_t)$, the memory requirements to $O(\log N_t)$ and have an order of accuracy which coincides with the order of the exact collocation methods with an opportune choice of the number of points M on the Talbot's contour.

The numerical experiments proposed clearly show that computational cost and the order of accuracy of our constructed methods are those expected.

The fast collocation methods are more efficient than classical collocation methods even for equations of which both kernel and its Laplace transform are known, thus the methods that we constructed are suitable for a wide class of equations.

The methods presented in this chapter are highly parallelizable and we think that they can be extended in a natural way to systems of Volterra integral equations.

In the near future we intend to study the stability properties of the fast collocation methods, and we expect that, for an opportune choice of M , they reflect the same properties of the corresponding classical ones.

Chapter 5

Fast Runge–Kutta methods for Volterra Integral equations of convolution type

5.1 Introduction

As in Chapter 4, this chapter concerns about the numerical solution of non-linear convolution VIEs of Hammerstein type

$$y(t) = f(t) + \int_0^t k(t - \tau)g(y(\tau))d\tau \quad t \in I := [0, T], \quad (5.1.1)$$

where the Laplace transform of the kernel rather than the convolution kernel itself is known a priori.

Now the functions f , g , k are assumed to be sufficiently smooth on I .

While in Chapter 4 we have introduced fast collocation methods for the equation (5.1.1), here we deal with the construction of fast VRK methods. We know that a naive implementation of a VRK method (see Section 2.3) would require $O(N_t^2)$ operations and $O(N_t)$ memory for the computation of the numerical solution over N_t time steps. In [52] it was constructed a VRK method for (5.1.1) of order 4 which reduced the computational effort to $O(N_t(\log N_t)^2)$ operations and kept memory requirements of $O(N_t)$. In this Chapter we will present fast VRK methods of generical order p for the equation (5.1.1), that requires only $O(N_t \log N_t)$ and $O(\log N_t)$ cost in time and space respectively.

In Section 5.2 we give the detailed construction of the fast VRK methods. The basic idea is inspired on the scheme described in Section 3.5.2 and also used in Chapter 4 for the fast collocation methods. Section 5.3 contains the calculation of the computational cost. The choice of a VRK method may be preferred to a collocation one when the VRK method is explicit, since it is possible to avoid the resolution of the non linear system appearing in the determination of the solution at each time step. The error analysis is given in Section 5.4.

In Section 5.5 the stability properties of the constructed methods are investigated with respect to the convolution test equation

$$y(t) = 1 + \int_0^t [\mu + \sigma(t - \tau)]y(\tau)d\tau \quad t \in [0, T], \quad \mu, \sigma \in \mathbb{R}^-. \quad (5.1.2)$$

This equation, that misses of course of some typical feature of VIEs, is typically used in the literature to test the VRK methods (see for example [6], [28]). We prove that the stability regions depend on the number of the points M chosen for the approximation of the Inverse Laplace Transform and if $M \rightarrow \infty$ then they tend to that of the classical VRK methods.

Section 5.6 contains numerical results that confirm the expected performances of the fast VRK methods in terms of accuracy, computational cost and stability properties. In Section 5.7 some concluding remarks are reported.

Some of the results of this chapter are reported in [20].

5.2 Fast VRK methods

In order to construct fast VRK (FVRK) methods for equation (5.1.1) we shall refer to explicit extended PVRK methods and implicit VRK methods of de Hoog and Weiss, described in Section 2.3, and we will opportunely modify them taking into account of the peculiarity of the considered equation. Namely, we will organize the computation of the lag terms both for involving the evaluations of the Laplace transform of the kernel and in order to reduce the computational cost, thus obtaining explicit extended fast PVRK (FPVRK) methods and implicit FVRK methods of de Hoog and Weiss.

Let us fix the vectors $c = (c_i)_{i=1}^m$, $b = (b_i)_{i=1}^m$ and the square matrix $A = (a_{i,s})_{i,s=1}^m$, determined by the "Butcher array" for ODEs

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array}$$

and let us fix $c_{m+1} = 1$.

Let us consider the uniform mesh $I_h = \{t_n := nh, n = 0, \dots, N_t, h \geq 0, N_t h = T\}$ and set $t_{n,i} := t_n + c_i h$, $i = 1, \dots, m + 1$.

An explicit m -stage PVRK method (2.3.1)-(2.3.4) applied to the Hammerstein equation (5.1.1) reads:

$$y_{n+1} = \bar{F}_n(t_{n,m+1}) + \bar{\Phi}_n(t_{n,m+1}) \quad n = 0, \dots, N_t - 1, \quad (5.2.1)$$

where the increment term is given by:

$$\bar{\Phi}_n(t_{n,m+1}) = h \sum_{i=1}^m b_i k((1 - c_i)h) g(Y_{n,i}). \quad (5.2.2)$$

The stages $Y_{n,i}$ are explicitly computed through

$$Y_{n,i} = \bar{F}_n(t_{n,i}) + h \sum_{s=1}^{i-1} a_{i,s} k((c_i - c_s)h) g(Y_{n,s}) \quad i = 1, \dots, m \quad (5.2.3)$$

and the lag term is given by

$$\bar{F}_n(t_{n,i}) = f(t_{n,i}) + h \sum_{r=0}^{n-1} \sum_{s=1}^m b_s k(t_{n,i} - t_{r,s}) g(Y_{r,s}) \quad i = 1, \dots, m+1. \quad (5.2.4)$$

An m -stage VRK method of de Hoog and Weiss (2.3.4)-(2.3.6), applied to equation (5.1.1) reads:

$$y_{n+1} = Y_{n,m} \quad n = 0, \dots, N_t - 1 \quad (5.2.5)$$

with $Y_{n,i}$ determined by the solution of the nonlinear system

$$Y_{n,i} = \bar{F}_n(t_{n,i}) + hc_i \sum_{l=1}^m b_l k(c_i(1 - c_l)h) \cdot g \left(\sum_{s=1}^m L_s(c_i c_l) Y_{n,s} \right) \quad i = 1, \dots, m \quad (5.2.6)$$

where the lag terms $\bar{F}_n(t_{n,i})$ are given by (5.2.4) for $i = 1, \dots, m$, and we remind that in this case is $c_m = 1$.

Remark 5.1. Note that the stages $Y_{n,i}$ can be regarded as an approximation of the exact solution in the point $t_{n,i}$, i. e. $Y_{n,i} \approx y(t_{n,i})$.

5.2.1 Fast computation of the lag terms

As regards the lag terms computation both PVRK methods and VRK methods of de Hoog and Weiss refer to formula (5.2.4). Since we only know the Laplace transform of the kernel k , we use the formula (3.4.7) for its approximation. To this aim let us define $\bar{I}_l = [B^{l-1}h + (c_1 - 1)h, (2B^l - 1)h]$, differing from $I_l = [B^{l-1}h, (2B^l - 1)h]$ defined in (3.4.5) for the dependence on c_1 . As in Section 3.5.2 let L be the smallest integer for which $t_{n+1} < 2B^L h$ and for $l = 1, 2, \dots, L - 1$ determine the integer $q_l \geq 1$ such that $\tau_l = q_l B^l h$ satisfies $t_{n+1} - \tau_l \in [B^l h, (2B^l - 1)h]$, and set $\tau_0 = t_n$ and $\tau_L = 0$. Similarly to (4.2.6) it is easy to verify that $[t_{n,i} - \tau_{l-1}, t_{n,i} - \tau_l] \subseteq \bar{I}_l$.

In order to use the formula (3.4.7) for the approximation of the kernel, we have to split the sum over r in (5.2.4) as

$$\bar{F}_n(t_{n,i}) = f(t_{n,i}) + h \sum_{l=1}^L \sum_{r=\frac{\tau_l}{h}}^{\frac{\tau_{l-1}}{h}-1} \sum_{s=1}^m b_s k(t_{n,i} - t_{r,s}) \cdot g(Y_{r,s}) \quad i = 1, \dots, m+1. \quad (5.2.7)$$

In this way for each fixed l the arguments $t_{n,i} - t_{r,s}$ of the kernel k , belong to the same \bar{I}_l for all $r = \tau_l/h, \dots, \tau_{l-1}/h - 1$. In fact

$$\begin{aligned} r \in \left[\frac{\tau_l}{h}, \frac{\tau_{l-1}}{h} - 1 \right] &\implies t_r \in [\tau_l, \tau_{l-1} - h] \implies t_{r,s} \in [\tau_l, \tau_{l-1}] \\ &\implies t_{n,i} - t_{r,s} \in [t_{n,i} - \tau_{l-1}, t_{n,i} - \tau_l] \subseteq \bar{I}_l. \end{aligned}$$

By inserting in (5.2.7) the formula (3.4.7) evaluated in $t = t_{n,i} - t_{r,s}$, we obtain the approximation $\bar{F}_{n,i}$ of $\bar{F}(t_{n,i})$

$$\bar{F}_{n,i} = f(t_{n,i}) + \sum_{l=1}^L \sum_{j=-N}^N \omega_j^{(l)} K(\lambda_j^{(l)}) e^{(t_{n,i} - \tau_{l-1})\lambda_j^{(l)}} z(\tau_{l-1}, \tau_l, \lambda_j^{(l)}) \quad i = 1, \dots, m+1, \quad (5.2.8)$$

where

$$z(\tau_{l-1}, \tau_l, \lambda_j^{(l)}) := h \sum_{r=\frac{\tau_l}{h}}^{\frac{\tau_{l-1}}{h}-1} \sum_{s=1}^m b_s e^{(\tau_{l-1} - t_{r,s})\lambda_j^{(l)}} \cdot g(Y_{r,s}). \quad (5.2.9)$$

Remark 5.2. The VRK methods of de Hoog and Weiss have $c_m = 1$, and thus the index i in (5.2.8) should arrive only up to $i = m$.

The implementation of the formula (5.2.8) by mean of the direct computation of (5.2.9) would still lead to a computational cost of $O(N_t^2)$. The idea proposed in [68] (and described in Section 3.5.2) to reduce the computational cost was based on a new organization in the computation of the function z at each time step, which could exploit its evaluations at the previous time steps. In order to reach the same goal we split the interval $[\tau_l, \tau_{l-1}]$ in subintervals

$[\tau_l + t_k, \tau_{l-1} + t_{k+1}]$ of length h and we denote with $z_k = z(\tau_l + t_k, \tau_l, \lambda_j^{(l)})$ and we prove that the following one step formula for the evaluation the function z in the mesh points from τ_l to τ_{l-1} holds.

Proposition 5.1. *Let $z(\tau_{l-1}, \tau_l, \lambda_j^{(l)})$ be given by (5.2.9), with $\tau_l = \bar{m}h$, $\tau_{l-1} = \tau_l + \bar{n}h$, then*

$$\begin{cases} z_{k+1} = e^{\lambda_j^{(l)} h} z_k + h \sum_{s=1}^m b_s e^{(1-c_s)h\lambda_j^{(l)}} \cdot g(\tilde{Y}_{k,s}) & k = 0, \dots, \bar{n} - 1 \\ z_0 = 0 \end{cases} \quad (5.2.10)$$

where $\tilde{Y}_{k,r} = Y_{\bar{m}+k,r}$.

Proof. From (5.2.10) we obtain :

$$\begin{aligned} z_1 &= h \sum_{s=1}^m b_s e^{(1-c_s)h\lambda_j^{(l)}} \cdot g(\tilde{Y}_{0,s}) \\ z_2 &= h \sum_{s=1}^m b_s e^{((1-c_s)+1)h\lambda_j^{(l)}} \cdot g(\tilde{Y}_{0,s}) + h \sum_{s=1}^m b_s e^{(1-c_s)h\lambda_j^{(l)}} \cdot g(\tilde{Y}_{1,s}) \\ z_3 &= h \sum_{s=1}^m b_s e^{((1-c_s)+2)h\lambda_j^{(l)}} \cdot g(\tilde{Y}_{0,s}) + h \sum_{s=1}^m b_s e^{((1-c_s)+1)h\lambda_j^{(l)}} \cdot g(\tilde{Y}_{1,s}) + \\ &\quad + h \sum_{s=1}^m b_s e^{(1-c_s)h\lambda_j^{(l)}} \cdot g(\tilde{Y}_{2,s}) \\ &\dots \\ z_{\bar{n}} &= h \sum_{s=1}^m b_s e^{((1-c_s)+(\bar{n}-1))h\lambda_j^{(l)}} \cdot g(\tilde{Y}_{0,s}) + h \sum_{s=1}^m b_s e^{((1-c_s)+(\bar{n}-2))h\lambda_j^{(l)}} \cdot g(\tilde{Y}_{1,s}) + \dots \\ &\dots + h \sum_{s=1}^m b_s e^{(1-c_s)h\lambda_j^{(l)}} \cdot g(\tilde{Y}_{\bar{n}-1,s}) = \\ &= h \sum_{k=\bar{m}}^{\bar{m}+\bar{n}-1} \sum_{s=1}^m b_s e^{(\tau_{l-1}-t_{k,s})\lambda_j^{(l)}} \cdot g(Y_{k,s}) = z(\tau_{l-1}, \tau_l, \lambda_j^{(l)}). \end{aligned}$$

Thus the thesis holds. \square

We can advance the values (5.2.10) of z by one step for all required values $\lambda_j^{(l)}$ on all Talbot contours in every time step $t_n \rightarrow t_{n+1}$, according to the scheme

illustrated in Section 3.5.2. So the computational cost of our algorithm is of $O(N_t \log_B N_t)$ operations (see Section 5.3 for the detailed calculation). Note that the function z in (5.2.8) does not depend on i , so we have to evaluate it only one time at each step $t_n \rightarrow t_{n+1}$ independently on the number stages m . Moreover the computation of z_{k+1} through (5.2.10) only requires the value z_k of z at the previous step and the values of the stages $\tilde{Y}_{k,r}$, which represent an approximation of the exact solution at the point $\tau_l + t_{k,r} \in [\tau_l + t_k, \tau_l + t_{k+1}]$ (as observed in Remark 5.1). So we do not need to keep in memory all the past values, thus leading to a memory requirement of $O(\log_B N_t)$.

5.2.2 Determination of the approximate solution

Once approximated the lag-terms in the points $t_{n,i}$, the next step to follow is to solve the nonlinear system (5.2.3) or (5.2.6), after inserting the inverse Laplace transform approximation (3.4.7).

Explicit extended FPVRK methods

Now we can use the approximations:

$$k((c_i - c_s)h) \approx \sum_{j=-N}^N \omega_j K(\lambda_j) e^{(c_i - c_s)h\lambda_j} =: \Psi_{is} \quad (5.2.11)$$

where the weights ω_j and the nodes λ_j correspond to the Talbot contour Γ_0 associated to the interval $\bar{I}_0 = [0, h]$. As concerns the increment term (5.2.2) the evaluations of k are approximated by

$$k((1 - c_i)h) \approx \sum_{j=-N}^N \omega_j K(\lambda_j) e^{(1 - c_i)h\lambda_j} =: \Psi_i.$$

We can explicitly compute the new stages $\bar{Y}_{n,i}$ through the formula

$$\bar{Y}_{n,i} = \bar{F}_n(t_{n,i}) + h \sum_{s=1}^{i-1} a_{i,s} \Psi_{is} g(Y_{n,s}) \quad i = 1, \dots, m \quad (5.2.12)$$

obtained by inserting the approximation (5.2.11) in the formula (2.3.3). The approximate solution of (5.1.1) in the mesh points I_h is thus obtained by

$$\bar{y}_{n+1} = \bar{F}_{n,m+1} + h \sum_{i=1}^m b_i \Psi_i \cdot g(\bar{Y}_{n,i}) \quad n = 0, \dots, N_t - 1. \quad (5.2.13)$$

The formula (5.2.12) can be written in vectorial form

$$\bar{\mathbf{Y}}_n = \bar{\mathbf{F}}_n + h \mathbf{D} \cdot g(\bar{\mathbf{Y}}_n) \quad (5.2.14)$$

where $\mathbf{D} = (d_{i,s})$ is a strictly lower triangular square matrix of dimension m whose elements are $d_{i,s} = a_{is} \Psi_{is}$, $\bar{\mathbf{F}}_n = (\bar{F}_{n,1}, \dots, \bar{F}_{n,m})^T$, $\bar{\mathbf{Y}}_n = (\bar{Y}_{n,1}, \dots, \bar{Y}_{n,m})^T$ and $g(\bar{\mathbf{Y}}_n) = (g(\bar{Y}_{n,1}), \dots, g(\bar{Y}_{n,m}))^T$.

Implicit FVRK methods of de Hoog and Weiss

As before we can use the approximations:

$$k(c_i(1 - c_l)h) \approx \sum_{j=-N}^N \omega_j K(\lambda_j) e^{c_i(1-c_l)h\lambda_j} =: \Psi_{il}$$

where the weights ω_j and the nodes λ_j correspond to the Talbot contour Γ_0 associated to the interval $\bar{I}_0 = [0, h]$.

Thus the nonlinear system (5.2.6) becomes

$$\bar{Y}_{n,i} = \bar{F}_{n,i} + hc_i \sum_{l=1}^m b_l \Psi_{il} g \left(\sum_{s=1}^m L_s(c_i c_l) \bar{Y}_{n,s} \right), \quad (5.2.15)$$

and the approximate solution of (5.1.1) in the mesh points I_h is obtained by

$$\bar{y}_{n+1} = \bar{Y}_{n,m} \quad n = 0, \dots, N_t - 1. \quad (5.2.16)$$

The nonlinear system (5.2.15) has the following vectorial form

$$\bar{\mathbf{Y}}_n = \bar{\mathbf{F}}_n + h \mathbf{H}(\bar{\mathbf{Y}}_n) \quad (5.2.17)$$

where \mathbf{H} is a vector valued function whose components are $H_i(\bar{Y}_n) = c_i \sum_{l=1}^m b_l \Psi_{il}$.
 $g(\sum_{s=1}^m L_s(c_i c_l) \bar{Y}_{n,s})$, $i = 1, \dots, m$ and $\bar{\mathbf{Y}}_n = (\bar{Y}_{n,1}, \dots, \bar{Y}_{n,m})^T$, $\bar{\mathbf{F}}_n = (\bar{F}_{n,1}, \dots, \bar{F}_{n,m})^T$.

Remark 5.3 (Linear case). In the case of linear VIEs the nonlinear systems (5.2.14) and (5.2.17) become linear, and they can be both written in the form

$$(\mathbf{I} - \mathbf{D})\bar{\mathbf{Y}}_n = \bar{\mathbf{F}}_n \quad (5.2.18)$$

where \mathbf{I} denotes the identity matrix of order m and $\mathbf{D} = (d_{i,s})$ is a square matrix of dimension m whose elements are

$$d_{i,s} = \begin{cases} a_{is} \Psi_{is} & \text{FPVRK methods} \\ c_i \sum_{l=1}^m b_l \Psi_{il} L_s(c_i c_l) & \text{FVRK of de Hoog and Weiss} \end{cases},$$

$$\Psi_{il} = \begin{cases} \sum_{j=-N}^N \omega_j K(\lambda_j) e^{(c_i - c_l) h \lambda_j} & \text{FPVRK methods} \\ \sum_{j=-N}^N \omega_j K(\lambda_j) e^{c_i (1 - c_l) h \lambda_j} & \text{FVRK of de Hoog and Weiss} \end{cases}.$$

5.3 Computational cost

In this section we will give the calculation of the computational cost of the FVRK methods in function of the number N_t of mesh points, proving that it is of $O(N_t \log(N_t))$ operations. We will only take into consideration the FVRK method of the Hoog and Weiss, since for the FPVRK methods the same result can be obtained in a similar way.

In the subsequent computations m will represent the number of stages, L_t the total number of different Talbot contours, $M = 2N + 1$ the number of points on each Talbot contour. The FVRK method of de Hoog and Weiss consists, for each time step t_n , $n = 0, \dots, N_t - 1$, in the following steps:

STEP 1 Evaluate the lag terms (5.2.8) for $i = 1, \dots, m + 1$, using all the values of the function z , already computed in the previous time steps.

We can observe that, by construction, the integer L in (5.2.8) satisfies

$$L \leq L_t \leq \log_B(N_t).$$

As the formula (5.2.8) involves, for each $i = 1, \dots, m$, a double sum for $l = 1, \dots, L$ and $j = -N, \dots, N$, the number of floating point operations $FLOP_{lag}$ for the lag terms computation is proportional to mML , where $M = 2N + 1$. Thus

$$FLOP_{lag} \leq C_1 m M \log_B(N_t).$$

STEP 2 Determine the approximate solution $\bar{y}_{n+1} = \bar{Y}_{n,m}$ by mean of (5.2.16).

Thus we need to solve the nonlinear system (5.2.15) for the stages $Y_{n,i}$, $i = 1, \dots, m$.

If we solve such system by an iterative method, then each iteration requires the computation, for each stage $Y_{n,i}$, $i = 1, \dots, m$, of a double sum

for $i = l, \dots, m$ and for $s = 1, \dots, m$. Thus the computational cost for the solution of the nonlinear system is of

$$FLOP_{sist} = C_2 k m^3$$

operations, where k is the number of iterations required by the iterative method.

STEP 3 Advance by one step the formula (5.2.10) for $l = 1, \dots, L_t$ (on all Talbot contours) and for $j = -N, \dots, N$. This values will be used for the computation of the lag terms in the subsequent time steps.

The formula (5.2.10) requires, for each l and for each j , the computation of a sum for $s = 1, \dots, m$, thus requiring a number of floating point operations $FLOP_{advance}$ proportional to mML_t . Thus

$$FLOP_{advance} \leq C_3 m M \log_B(N_t).$$

Thus the total number of floating point operations $FLOP_{tot}$ in function of the time steps N_t is given by

$$\begin{aligned} FLOP_{tot} &= N_t(FLOP_{lag} + FLOP_{sist} + FLOP_{advance}) \leq \\ &\leq (C_1 + C_3)mM \cdot N_t \log_B(N_t) + C_2 k m^3 \cdot N_t \end{aligned}$$

and then

$$FLOP_{tot} = O(N_t \log_B(N_t)).$$

5.4 Convergence analysis

Let us consider a classical explicit PVRK method or an implicit VRK method of De Hoog and Weiss of order p (for the discussion of the order of convergence of the classical VRK methods refer to Theorem 2.8 and Corollary 2.2). The following theorem establishes the order of convergence of the corresponding FVRK methods.

Theorem 5.1. *Let $\bar{e}_n = y(t_n) - \bar{y}_n$ be the error of the FVRK methods (explicit FPVRK or implicit FVRK methods of De Hoog and Weiss). If the function $g \in C^1(I)$ then*

$$\max_{1 \leq n \leq N_t} |\bar{e}_n| = O(h^p), \quad (5.4.1)$$

with sufficiently large number M of points on the Talbot contour.

Proof. Now we prove the thesis for the fast implicit De Hoog and Weiss methods, since in the case of explicit PVRK methods the proof is similar. Let $e_n = y(t_n) - y_n$ and $\bar{e}_n = y(t_n) - \bar{y}_n$, $n = 1, \dots, N_t$ respectively denote the error of the classical de Hoog and Weiss VRK method (2.3.4), (2.3.5), (2.3.6) and of the corresponding fast VRK method (5.2.8), (5.2.15), (5.2.16), and define $\epsilon_n = y_n - \bar{y}_n$.

According to our notation we have

$$\max_{1 \leq n \leq N_t} |\bar{e}_n| \leq \max_{1 \leq n \leq N_t} |e_n| + \max_{1 \leq n \leq N_t} |\epsilon_n| \quad n = 1, \dots, N_t$$

Since the classical methods we are considering are of order p , it follows that

$\max_{1 \leq n \leq N} |e_n| \leq C_1 h^p$. As regards ϵ_n , subtracting (5.2.15) from (5.2.6) and applying the Lagrange theorem to the function g we obtain

$$\eta_{n,i} = h \sum_{s=1}^m d_{i,s} \eta_{n,s} + h \sum_{k=0}^{n-1} \sum_{s=1}^m q_{i,s}^{(n,k)} \eta_{k,s} + h E_{n,n}^{(i)} + h \sum_{l=1}^L E_{n,l}^{(i)} \quad i = 1, \dots, m \quad (5.4.2)$$

where

$$\begin{aligned}\eta_{n,i} &= Y_{n,i} - \bar{Y}_{n,i}, \quad i = 1, \dots, m \\ q_{i,s}^{(n,k)} &= \sum_{j=-N}^N \omega_j^{(l)} K(\lambda_j^{(l)}) b_s e^{(t_{n,i}-t_{k,s})\lambda_j^{(l)}} g'(\xi_{k,s}) \quad k = 0, \dots, n-1, \quad i, s = 1, \dots, m \\ E_{n,l}^{(i)} &= \sum_{r=\frac{\tau_l}{h}}^{\frac{\tau_l-1}{h}-1} \sum_{s=1}^m b_s \left(k(t_{n,i} - t_{r,s}) - \sum_{j=-N}^N \omega_j^{(l)} K(\lambda_j^{(l)}) e^{(t_{n,i}-t_{r,s})\lambda_j^{(l)}} \right) g(Y_{r,s}) \\ E_{n,n}^{(i)} &= c_i \sum_{l=1}^m b_l (k(c_i(1-c_l)h) - \Psi_{il}) g \left(\sum_{s=1}^m L_s(c_i c_l) Y_{n,s} \right) \\ d_{i,s} &= c_i \sum_{l=1}^m b_l \Psi_{il} L_s(c_i c_l) g'(\xi_{i,l}^{(n)}).\end{aligned}$$

From this point on the proof is parallel to that of Theorem 4.2 and hence we shall provide a sketch only of the main steps. By writing down equation (5.4.2) in vectorial form, through (3.4.8) we obtain the discrete Gronwall inequality

$$\|\eta_n\|_1 \leq hC_3 \sum_{k=0}^{n-1} \|\eta_k\|_1 + C_2 e^{-c\sqrt{M}}, \quad n = 0, \dots, N_{t-1} \quad (5.4.3)$$

where $\eta_k = (\eta_{k,1}, \dots, \eta_{k,m})^T$. Applying the Gronwall Theorem 2.6 to (5.4.3), it follows that $\|\eta_n\|_1 \leq C_2 e^{-c\sqrt{M}}$ and hence, being $\epsilon_n = \eta_{n,m}$, we obtain the thesis by choosing M such that $C_2 e^{-c\sqrt{M}} \leq C_1 h^p$. \square

Since the error due to the approximation of the inverse Laplace transform decreases exponentially with M it is sufficient to fix a not too high number of points on the Talbot contour in order to make this error negligible. Moreover if $M \rightarrow +\infty$, that is the formula for approximating the inverse Laplace transform is exact, we return to the corresponding VRK method.

It immediately follows by Theorem 5.1, using Corollary 2.2, that it is possible to achieve local superconvergence at the mesh points by opportunely choosing the parameters of the VRK method:

Corollary 5.1. *Let $f, k \in C^{2m-v}$, with $v \in \{0, 1, 2\}$ with $m \geq \lfloor v/2 \rfloor + 1$,*

(i) If the nodes $\{c_i\}$ are the Radau II points for $(0, 1]$, then the fast VRK method of the Hoog and Weiss (5.2.8), (5.2.10), (5.2.15), (5.2.16) satisfies, for $v = 1$,

$$\max_{1 \leq n \leq N_t} |\bar{e}_n| = O(h^{2m-1})$$

(ii) If the nodes $\{c_i\}$ are the Lobatto points for $[0, 1]$, then the fast VRK method of the Hoog and Weiss (5.2.8), (5.2.10), (5.2.15), (5.2.16) satisfies, for $v = 2$,

$$\max_{1 \leq n \leq N_t} |\bar{e}_n| = O(h^{2m-2})$$

(iii) Let the nodes $\{c_i\}$ are the the m Gauss points for $(0, 1)$, $c_{m+1} = 1$. Furthermore, suppose to consider a modification of the fast VRK method of the Hoog and Weiss given by (5.2.8), (5.2.10), (5.2.15) for the computation of the stages, but with the approximate solution calculated by mean of (5.2.13), then, for $v = 0$,

$$\max_{1 \leq n \leq N_t} |\bar{e}_n| = O(h^{2m})$$

5.5 Stability analysis

In this Section we will study the stability properties of the FVRK methods with respect to test equations usually employed in literature for the stability analysis (see for example [6] [28] and their references), namely the basic test equation and the convolution test equation.

5.5.1 Basic test equation

The basic test equation assumes the form

$$y(t) = 1 + \mu \int_0^t y(\tau) d\tau \quad t \in [0, T], \mu \in \mathbb{C}. \quad (5.5.1)$$

Since the exact solution $y(t)$ of (5.5.1) tends to zero when t goes to $+\infty$ if and only if $\operatorname{Re}(\mu) < 0$, it is natural to require that the numerical solution y_n produced by the FVRK method when applied to the equation (5.5.1) with stepsize h , has the same behaviour. Thus we recall the following definition of numerical stability.

Definition 5.1. A numerical method is said to be stable for given $z := h\mu \in \mathbb{C}$ if the numerical solution y_n , resulting from applying the method to (5.5.1) with fixed stepsize h , tends to zero when $n \rightarrow +\infty$. The region of absolute stability of the method is the set of all values $z \in \mathbb{C}$ for which the method is stable. Furthermore the method is said *A*-stable if its region of absolute stability includes the negative complex half plane \mathbb{C}^- .

Let us first recall the main results in the stability analysis of classical VRK methods. We shall refer in particular to the two classes of methods:

1. The PVRK method of the form (2.3.1)-(2.3.4) with Butcher's array

$$\begin{array}{c|c} \mathbf{c} & \mathbf{A} \\ \hline & \mathbf{b}^T \end{array}. \quad (5.5.2)$$

2. The VRK method of de Hoog and Weiss (2.3.4)-(2.3.6) defined by the real parameters $0 \leq c_1 < \dots < c_m \leq 1$ and with

$$b_k = \int_0^1 L_k(\theta) d\theta \quad k = 1, \dots, m, \quad a_{is} = \int_0^{c_i} L_s(\theta) d\theta = c_i \sum_{j=1}^m b_j L_s(c_i c_j). \quad (5.5.3)$$

In [6] the following result was proved.

Theorem *The classical VRK method applied to the test equation (5.5.1) leads to the following two terms relation*

$$\mathbf{Y}_n = R(z) \mathbf{Y}_{n-1} \quad (5.5.4)$$

with

$$R(z) = 1 + z \mathbf{b}^T (\mathbf{I} - z \mathbf{A})^{-1} \mathbf{u}. \quad (5.5.5)$$

Here $\mathbf{Y}_k = (Y_{k,1}, \dots, Y_{k,m})^T$, $\mathbf{u} = (1, \dots, 1)^T$ and the matrices \mathbf{b}^T and \mathbf{A} are defined by (5.5.2) for PVRK methods and by (5.5.3) for VRK methods of de Hoog and Weiss.

Now we investigate the numerical stability of FVRK methods. To this aim we apply the FVRK method to the equation (5.5.1), and consequently the stages are determined by the solution of the linear system

$$\bar{Y}_{n,i} = \bar{F}_{n,i} + z \sum_{s=1}^m d_{i,s} \bar{Y}_{n,s} \quad i = 1, \dots, m \quad (5.5.6)$$

where

$$\bar{F}_{n,i} = 1 + z \sum_{l=1}^L \sum_{k=\frac{\tau_l}{h}}^{\frac{\tau_{l-1}}{h}-1} \sum_{j=-N}^N \frac{\omega_j^{(l)}}{\lambda_j^{(l)}} e^{(t_{n,i}-\tau_{l-1})\lambda_j^{(l)}} \sum_{s=1}^m b_s e^{(\tau_{l-1}-t_{k,s})\lambda_j^{(l)}} \bar{Y}_{k,s}, \quad i = 1, \dots, m, \quad (5.5.7)$$

$$d_{i,s} = \begin{cases} a_{is}\Psi_{is} & \text{FPVRK methods} \\ c_i \sum_{l=1}^m b_l \Psi_{il} L_s(c_i c_l) & \text{FVRK of de Hoog and Weiss} \end{cases}, \quad (5.5.8)$$

$$\Psi_{il} = \begin{cases} \sum_{j=-N}^N \frac{\omega_j}{\lambda_j} e^{(c_i - c_l)h\lambda_j} & \text{FPVRK methods} \\ \sum_{j=-N}^N \frac{\omega_j}{\lambda_j} e^{c_i(1-c_l)h\lambda_j} & \text{FVRK of de Hoog and Weiss} \end{cases}. \quad (5.5.9)$$

The formula (5.5.7) can be written in vectorial form as

$$\bar{\mathbf{F}}_n = \mathbf{u} + z \sum_{l=1}^L \sum_{k=\frac{\tau_l}{h}}^{\frac{\tau_{l-1}}{h}-1} \mathbf{Q}_{n,k}^{(l)} \bar{\mathbf{Y}}_k, \quad (5.5.10)$$

where $\bar{\mathbf{Y}}_k = (\bar{Y}_{k,1}, \dots, \bar{Y}_{k,m})^{\mathbf{T}}$, $\bar{\mathbf{F}}_n = (\bar{F}_{n,1}, \dots, \bar{F}_{n,m})^{\mathbf{T}}$, $\mathbf{u} = (1, \dots, 1)^{\mathbf{T}}$ and

$$\mathbf{Q}_{n,k}^{(l)} = \left(b_s \sum_{j=-N}^N \frac{\omega_j^{(l)}}{\lambda_j^{(l)}} e^{(t_{n,i} - t_{k,s})\lambda_j^{(l)}} \right)_{i,s=1,\dots,m} \quad (5.5.11)$$

is a square matrix of dimension m .

Consequently, assuming that $\det(\mathbf{I} - z\mathbf{D}) \neq 0$, (5.5.6) becomes

$$\bar{\mathbf{Y}}_n = (\mathbf{I} - z\mathbf{D})^{-1} (\mathbf{u} + z \sum_{l=1}^L \sum_{k=\frac{\tau_l}{h}}^{\frac{\tau_{l-1}}{h}-1} \mathbf{Q}_{n,k}^{(l)} \bar{\mathbf{Y}}_k), \quad (5.5.12)$$

where the elements of the matrix \mathbf{D} are defined by (5.5.8). The double sum in (5.5.12) can be written as one single sum over the index k :

$$\bar{\mathbf{Y}}_n = (\mathbf{I} - z\mathbf{D})^{-1} (\mathbf{u} + z \sum_{k=0}^{n-1} \mathbf{Q}_{n,k}^{(l)} \bar{\mathbf{Y}}_k), \quad (5.5.13)$$

provided that the index l of the Talbot contour is determined by n and k in such a way that $t_k \in [\tau_l, \tau_{l-1}]$. By subtracting the expressions of $\bar{\mathbf{Y}}_n$ and $\bar{\mathbf{Y}}_{n-1}$ given by (5.5.13) and by opportune manipulations we obtain, for $n \geq 1$,

$$\bar{\mathbf{Y}}_n = (\mathbf{I} + z(\mathbf{I} - z\mathbf{D})^{-1} \mathbf{Q}_1^{(1)}) \bar{\mathbf{Y}}_{n-1} + \sum_{k=0}^{n-2} z(\mathbf{I} - z\mathbf{D})^{-1} [\mathbf{Q}_{n,k}^{(l)} - \mathbf{Q}_{n-1,k}^{(l)}] \bar{\mathbf{Y}}_k \quad (5.5.14)$$

with

$$\bar{\mathbf{Y}}_0 = (\mathbf{I} - z\mathbf{D})^{-1}\mathbf{u}.$$

Now let $F(s) = \frac{b_i}{s}$ be the Laplace transform of the constant function $f(t) = b_j$, and let $\check{f}(t)$ be the inverse Laplace transform approximation of $F(s)$ obtained through the formula (3.4.7), from (5.5.11) it follows that $\left(\mathbf{Q}_{n,k}^{(l)}\right)_{i,j} = \check{f}(t_{n,i} - t_{k,j})$ and $\left(\mathbf{Q}_{n-1,k}^{(l)}\right)_{i,j} = \check{f}(t_{n-1,i} - t_{k,j})$. Now, at the same way as it is done for the local truncation error in the treatment of numerical stability for ODEs (see [66], p. 76), we can freeze the relative error of the inverse Laplace transform approximation $\check{f}(t)$ obtained through the formula (3.4.7) in the approximation interval, as this error is of order $O(e^{-c\sqrt{M}})$. As $f(t)$ is a constant function, this implies that $\check{f}(t)$ is a constant function, too. It follows that $\mathbf{Q}_{n,k}^{(l)} = \mathbf{Q}_{n-1,k}^{(l)}$ and so the the following theorem is proved.

In this way we have proved the following theorem:

Theorem 5.2. *A FVRK method applied to the test equation (5.5.1) leads to the following two terms relation:*

$$\bar{\mathbf{Y}}_n = \mathbf{R}(z)\bar{\mathbf{Y}}_{n-1} \tag{5.5.15}$$

where

$$\mathbf{R}(z) = (\mathbf{I} + z(\mathbf{I} - z\mathbf{D})^{-1}\mathbf{Q}_1^{(1)}) \tag{5.5.16}$$

is a square matrix of dimension m , with $\mathbf{Q}_1^{(1)} = \mathbf{Q}_{n,n-1}^{(1)}$ given by (5.5.11) and \mathbf{D} defined by (5.5.8).

The next result is an immediate consequence of Theorem 5.2 and of the Definition 5.1.

Corollary 5.2. *If the eigenvalues of $\mathbf{R}(z)$ are within the unit circle, then the FVRK method is stable. The region of absolute stability of the method is thus*

the set

$$\mathcal{S} = \{z \in \mathbb{C} : |\mathbf{R}(z)| < 1\},$$

where the stability matrix $\mathbf{R}(z)$ is defined by (5.5.16).

Remark 5.4. The stability regions of the FVRK methods tend, as $M \rightarrow \infty$, to the stability regions of the corresponding classical ones. Infact we can at first observe that, when $M \rightarrow \infty$,

$$\begin{aligned} \mathbf{Q}_1^{(1)} &\rightarrow \mathbf{u}\mathbf{b}^T \\ \mathbf{D} &\rightarrow \mathbf{A} \end{aligned}$$

where \mathbf{b}^T and \mathbf{A} are defined by (5.5.2) for FPVRK methods and by (5.5.3) for FVRK methods of de Hoog and Weiss. It immediately follows that

$$\mathbf{R}(z) \rightarrow \mathbf{I} + z(\mathbf{I} - z\mathbf{A})\mathbf{u}\mathbf{b}^T$$

This is an $m \times m$ matrix whose eigenvalues are $\lambda_1 = 1 + z\mathbf{b}^T(\mathbf{I} - z\mathbf{A})^{-1}\mathbf{u}$ with multiplicity 1 and $\lambda_2 = 1$ with multiplicity $m - 1$. As $\bar{\mathbf{Y}}_0 = (\mathbf{I} - z\mathbf{D})^{-1}\mathbf{u}$ is an eigenvector associated with the eigenvalue $1 + z\mathbf{b}^T(\mathbf{I} - z\mathbf{A})^{-1}\mathbf{u}$, it follows that the two terms recursion (5.5.15) of the FVRK method tends to that of the classical one (5.5.4).

Remark 5.5. For the the implicit Euler FVRK method, characterized by $m = 1$, $c_1 = 1$, $b_1 = 1$, $a_{11} = 1$, we have $R(z) = 1 + z(1 - zd_{11})^{-1}Q_1^{(1)}$, where $Q_1^{(1)} = \sum_{j=-N}^N \frac{\omega_j^{(1)}}{\lambda_j^{(1)}} e^{h\lambda_j^{(1)}} =: \beta$ and $d_{11} = c_1 b_1 \Psi_{11} L_1(c_1 c_1) = \Psi_{11} = \sum_{j=-N}^N \frac{\omega_j}{\lambda_j} =: \alpha$. Thus it follows that

$$R(z) = 1 + \frac{z\beta}{1 - z\alpha}$$

and an easy computation shows that $|R(z)| < 1$ if and only if z is outside the circle $\mathcal{C}_{\alpha,\beta}$ centered in $C = \left(\frac{1}{2\alpha-\beta}, 0\right)$ and with radius $r = \frac{1}{|2\alpha-\beta|}$. We can

observe that when $M \rightarrow \infty$ then $\alpha, \beta \rightarrow 1$ and the stability region tends to that of classical Euler method, that is the region outside the circle centered in $(1, 0)$ and with radius equal to 1. Moreover we would like to underline that the implicit Euler VRK method is A -stable for all values of M , being the circle $\mathcal{C}_{\alpha, \beta}$ all contained in the right complex plane.

5.5.2 Convolution test equation

Now we will study the stability properties of the FVRK methods with respect to the convolution test equation

$$y(t) = 1 + \int_0^t [\mu + \sigma(t - \tau)]y(\tau)d\tau \quad t \in [0, T], \quad \mu, \sigma \in R. \quad (5.5.17)$$

Since the exact solution of (5.5.17) $y(t)$ goes to zero when $t \rightarrow +\infty$ if and only if $\mu < 0$ and $\sigma \leq 0$, it is natural to require that the numerical solution y_n , produced by the FVRK methods when applied to (5.5.17) with stepsize h , has the same behaviour.

Thus we recall the following definition of numerical stability.

Definition 5.2. A numerical method is said to be stable for given $z := h\mu$, $w := h^2\sigma$ if it yields an approximate solution y_n which satisfies $y_n \rightarrow 0$ as $n \rightarrow \infty$ whenever it is applied with a fixed stepsize $h > 0$ to the test equation (5.5.17). The region of stability of the method is the set of all values (z, w) for which the method is stable.

The following theorem refers to classical VRK methods and can be derived from [6].

Theorem *The classical VRK method applied to the test equation (5.5.17) leads*

to the following three terms relation

$$\begin{aligned} \mathbf{N}\bar{\mathbf{Y}}_{n+2} &= (2\mathbf{N} + (z + w)\mathbf{u}\mathbf{b}^T + w\theta\mathbf{b}^T - w\mathbf{u}\mathbf{r}^T) \bar{\mathbf{Y}}_{n+1} \\ &- (\mathbf{N} + z\mathbf{u}\mathbf{b}^T + w\theta\mathbf{b}^T - w\mathbf{u}\mathbf{r}^T) \bar{\mathbf{Y}}_n, \end{aligned} \quad (5.5.18)$$

with

$$\mathbf{N} = \mathbf{I} - z\mathbf{A} - w\bar{\mathbf{A}}.$$

Here $\mathbf{Y}_k = (Y_{k,1}, \dots, Y_{k,m})^T$, $\mathbf{u} = (1, \dots, 1)^T$, the matrices \mathbf{b}^T and \mathbf{A} are defined by (5.5.2) for PVRK methods and by (5.5.3) for VRK methods of de Hoog and Weiss,

$$\begin{aligned} r &= [r_1, \dots, r_m]^T & r_i &= b_i c_i \\ \theta &= [\theta_1, \dots, \theta_m]^T & \theta_i &= c_i \\ \bar{\mathbf{A}} = (\bar{a}_{i,k})_{i,k=1\dots m} & \bar{a}_{i,k} = \begin{cases} a_{ik}(c_i - c_k) & \text{PVRK methods} \\ c_i \sum_{l=1}^m b_l c_l (1 - c_l) L_k(c_i c_l) & \text{VRK of de Hoog and Weiss} \end{cases} \end{aligned}$$

Now, in order to investigate the stability of FVRK methods, we apply the FVRK method to the test equation (5.5.17), obtaining

$$\bar{Y}_{n,i} = \bar{F}_{n,i} + z \sum_{s=1}^m d_{i,s} \bar{Y}_{n,s} + w \sum_{s=1}^m \bar{d}_{i,s} \bar{Y}_{n,s} \quad i = 1, \dots, m, \quad (5.5.19)$$

where

$$\begin{aligned} \bar{F}_{n,i} &= 1 + z \sum_{l=1}^L \sum_{r=\frac{\tau_l}{h}}^{\frac{\tau_{l-1}}{h}-1} \sum_{s=1}^m (\mathbf{Q}_{n,r}^{(l)})_{i,s} \bar{Y}_{r,s} + w \sum_{l=1}^L \sum_{r=\frac{\tau_l}{h}}^{\frac{\tau_{l-1}}{h}-1} \sum_{s=1}^m (n-r) (\bar{\mathbf{Q}}_{n,r}^{(l)})_{i,s} \bar{Y}_{r,s} \\ &+ w \sum_{l=1}^L \sum_{r=\frac{\tau_l}{h}}^{\frac{\tau_{l-1}}{h}-1} \sum_{s=1}^m (c_i - c_s) (\bar{\mathbf{Q}}_{n,r}^{(l)})_{i,s} \bar{Y}_{r,s} \quad i = 1, \dots, m, \end{aligned} \quad (5.5.20)$$

$d_{i,s}$ are given by (5.5.8)-(5.5.9),

$$\bar{d}_{i,s} = \begin{cases} a_{is} \bar{\Psi}_{is} & \text{FPVRK methods} \\ c_i \sum_{l=1}^m b_l \bar{\Psi}_{il} L_s(c_i c_l) & \text{FVRK of de Hoog and Weiss} \end{cases},$$

$$\bar{\Psi}_{il} = \begin{cases} \sum_{j=-N}^N \frac{\omega_j}{\lambda_j^2} \frac{e^{(c_i - c_l)h\lambda_j}}{h} & \text{FPVRK methods} \\ \sum_{j=-N}^N \frac{\omega_j}{\lambda_j^2} \frac{e^{c_i(1-c_l)h\lambda_j}}{h} & \text{FVRK of de Hoog and Weiss} \end{cases},$$

$\mathbf{Q}_{n,r}^{(l)}$ are given by (5.5.11), and

$$(\bar{\mathbf{Q}}_{n,r}^{(l)})_{i,s} = b_s \sum_{j=-N}^N \frac{\omega_j^{(l)}}{(\lambda_j^{(l)})^2} \frac{e^{(t_{n,i} - t_{r,s})\lambda_j^{(l)}}}{t_{n,i} - t_{r,s}}.$$

Assuming that $\det \mathbf{N} \neq 0$, the formulas (5.5.19), (5.5.20) can be written equivalently in vectorial form as

$$\bar{\mathbf{Y}}_{\mathbf{n}} = \mathbf{N}^{-1} \bar{\mathbf{F}}_{\mathbf{n}} \quad (5.5.21)$$

$$\bar{\mathbf{F}}_{\mathbf{n}} = \mathbf{u} + \sum_{l=1}^L \sum_{r=\frac{\tau_l}{h}}^{\frac{\tau_{l+1}-1}{h}-1} [z \mathbf{Q}_{n,r}^{(l)} + w(n-r) \bar{\mathbf{Q}}_{n,r}^{(l)} + w\bar{\theta} \bar{\mathbf{Q}}_{n,r}^{(l)} - w \mathbf{P}_{n,r}^{(l)}] \bar{\mathbf{Y}}_{\mathbf{r}}, \quad (5.5.22)$$

having set $\bar{\mathbf{Y}}_{\mathbf{r}} = (\bar{Y}_{r,1}, \dots, \bar{Y}_{r,m})^{\mathbf{T}}$, $\bar{\mathbf{F}}_{\mathbf{n}} = (\bar{F}_{n,1}, \dots, \bar{F}_{n,m})^{\mathbf{T}}$, $\mathbf{u} = (1, \dots, 1)^{\mathbf{T}}$. Here $\mathbf{N} = \mathbf{I} - z\mathbf{D} - w\bar{\mathbf{D}}$, $(\mathbf{P}_{n,r}^{(l)})_{i,s} = c_s (\bar{\mathbf{Q}}_{n,r}^{(l)})_{i,s}$, $\bar{\theta} = \text{diag}(c_1, \dots, c_s)$ are square matrices of dimension m .

From (5.5.21) and (5.5.22) it is possible to obtain the following relation

$$\begin{aligned}
\bar{\mathbf{Y}}_{n+2} &= \mathbf{N}^{-1} \left(2\mathbf{N} + z\mathbf{Q}_1^{(1)} + w\bar{\mathbf{Q}}_1^{(1)} + w\bar{\theta}\bar{\mathbf{Q}}_1^{(1)} - w\mathbf{P}_1^{(1)} \right) \bar{\mathbf{Y}}_{n+1} + \\
&\quad - \mathbf{N}^{-1} \left(\mathbf{N} + z\mathbf{Q}_1^{(1)} + w\bar{\theta}\bar{\mathbf{Q}}_1^{(1)} - w\mathbf{P}_1^{(1)} \right) \bar{\mathbf{Y}}_n + \\
&\quad + \mathbf{N}^{-1} \sum_{r=0}^n \left[z(\mathbf{Q}_{n+2,r}^{(l)} - \mathbf{Q}_{n+1,r}^{(l)}) + (n+2-r)w \left(\bar{\mathbf{Q}}_{n+2,r}^{(l)} - \bar{\mathbf{Q}}_{n+1,r}^{(l)} \right) + \right. \\
&\quad \left. + w\bar{\theta} \left(\bar{\mathbf{Q}}_{n+2,r}^{(l)} - \bar{\mathbf{Q}}_{n+1,r}^{(l)} \right) - w \left(\mathbf{P}_{n+2,r}^{(l)} - \mathbf{P}_{n+1,r}^{(l)} \right) \right] \bar{\mathbf{Y}}_r + \\
&\quad + \mathbf{N}^{-1} \sum_{r=0}^{n-1} \left[z(\mathbf{Q}_{n,r}^{(l)} - \mathbf{Q}_{n+1,r}^{(l)}) + (n-r)w \left(\bar{\mathbf{Q}}_{n,r}^{(l)} - \bar{\mathbf{Q}}_{n+1,r}^{(l)} \right) + \right. \\
&\quad \left. + w\bar{\theta} \left(\bar{\mathbf{Q}}_{n,r}^{(l)} - \bar{\mathbf{Q}}_{n+1,r}^{(l)} \right) - w \left(\mathbf{P}_{n,r}^{(l)} - \mathbf{P}_{n+1,r}^{(l)} \right) \right] \bar{\mathbf{Y}}_r. \tag{5.5.23}
\end{aligned}$$

In this formula the index l of the Talbot contour is determined by n and r in such a way that $t_r \in [\tau_l, \tau_{l-1}]$.

Now, at the same way as it is done for the local truncation error in the treatment of numerical stability for ODEs (see [66], p. 76), we can freeze the relative error of the inverse Laplace transform approximation $\check{f}(t)$ obtained through the formula (3.4.7) in the approximation interval, as this error is of order $O(e^{-c\sqrt{M}})$. As in Section 5.5.1 we have $\mathbf{Q}_{n,r}^{(l)} = \mathbf{Q}_{n-1,r}^{(l)}$. Similarly we obtain that $\bar{\mathbf{Q}}_{n,r}^{(l)} = \bar{\mathbf{Q}}_{n-1,r}^{(l)}$ and $\mathbf{P}_{n,r}^{(l)} = \mathbf{P}_{n-1,r}^{(l)}$. Thus the relation (5.5.23) becomes a difference equation of fixed order and we have proved the following theorem:

Theorem 5.3. *A FVRK method applied to the test equation (5.5.17) leads to the following recurrence relation*

$$\begin{aligned}
\bar{\mathbf{Y}}_{n+2} &= \mathbf{N}^{-1} \left(2\mathbf{N} + z\mathbf{Q}_1^{(1)} + w\bar{\mathbf{Q}}_1^{(1)} + w\bar{\theta}\bar{\mathbf{Q}}_1^{(1)} - w\mathbf{P}_1^{(1)} \right) \bar{\mathbf{Y}}_{n+1} \\
&\quad - \mathbf{N}^{-1} \left(\mathbf{N} + z\mathbf{Q}_1^{(1)} + w\bar{\theta}\bar{\mathbf{Q}}_1^{(1)} - w\mathbf{P}_1^{(1)} \right) \bar{\mathbf{Y}}_n. \tag{5.5.24}
\end{aligned}$$

where $\mathbf{Q}_1^{(1)} = \mathbf{Q}_{n+2,n+1}^{(1)}$, $\bar{\mathbf{Q}}_1^{(1)} = \bar{\mathbf{Q}}_{n+2,n+1}^{(1)}$ and $\mathbf{P}_1^{(1)} = \mathbf{P}_{n+2,n+1}^{(1)}$.

Let

$$\begin{aligned}\mathbf{E} &= \mathbf{N}^{-1} \left(2\mathbf{N} + z\mathbf{Q}_1^{(1)} + w\bar{\mathbf{Q}}_1^{(1)} + w\bar{\theta}\bar{\mathbf{Q}}_1^{(1)} - w\mathbf{P}_1^{(1)} \right), \\ \mathbf{F} &= -\mathbf{N}^{-1} \left(\mathbf{N} + z\mathbf{Q}_1^{(1)} + w\bar{\theta}\bar{\mathbf{Q}}_1^{(1)} - w\mathbf{P}_1^{(1)} \right),\end{aligned}$$

the relation (5.5.24) can be written in the form

$$\begin{bmatrix} \bar{\mathbf{Y}}_{n+1} \\ \bar{\mathbf{Y}}_n \end{bmatrix} = \mathbf{S} \begin{bmatrix} \bar{\mathbf{Y}}_n \\ \bar{\mathbf{Y}}_{n-1} \end{bmatrix} \quad (5.5.25)$$

for $n = 1, 2, \dots$, where

$$\mathbf{S} = \begin{bmatrix} \mathbf{E} & \mathbf{F} \\ \mathbf{I} & \mathbf{0} \end{bmatrix} \quad (5.5.26)$$

Now the next result immediately follows from relations (5.5.25)-(5.5.26) and from the Definition 5.2.

Corollary 5.3. *If the eigenvalues of \mathbf{S} are within the unit circle, then the FVRK method is stable. The stability region of the method is thus the set*

$$R = \{z, w \in \mathbb{R}^- : |\text{eig}(\mathbf{S})| < 1\}.$$

Remark 5.6. As observed in Remark 5.4 for basic test equation (5.5.1), also in the case of the convolution test equation (5.5.17) the stability regions of the FVRK methods tend, as $M \rightarrow \infty$, to the stability regions of the corresponding classical ones. Infact it is easy to verify that the three term recursion (5.5.24) tends to the three terms recursion (5.5.18).

Remark 5.7. Obviously when $w = 0$ we obtain again the region of stability associated to the basic test equation.

Remark 5.8. As in Remark 5.5 for basic test equation, let us consider the implicit Euler FVRK method characterized by $m = 1$, $c_1 = 1$, $b_1 = 1$, $a_{11} = 1$.

In this case we have

$$\begin{aligned} d_{11} &= c_1 b_1 \Psi_{11} L_1(c_1 c_1) = \Psi_{11} = \sum_{j=-N}^N \frac{\omega_j}{\lambda_j} =: \alpha \\ \bar{d}_{11} &= c_1 b_1 \bar{\Psi}_{11} L_1(c_1 c_1) = \bar{\Psi}_{11} = \sum_{j=-N}^N \frac{\omega_j}{h(\lambda_j)^2} =: \bar{\alpha} \\ Q_1^{(1)} &= \sum_{j=-N}^N \frac{\omega_j^{(1)}}{\lambda_j^{(1)}} e^{h\lambda_j^{(1)}} =: \beta \\ \bar{Q}_1^{(1)} &= \sum_{j=-N}^N \frac{\omega_j^{(1)}}{(\lambda_j^{(1)})^2} \frac{e^{h\lambda_j^{(1)}}}{h} =: \bar{\beta} \\ P_1^{(1)} &= \bar{Q}_1^{(1)} = \bar{\beta} \\ N &= 1 - \alpha z - \bar{\alpha} w, \end{aligned}$$

from which it follows that

$$\mathbf{S} = \begin{bmatrix} \frac{(\beta-2\alpha)z+(\bar{\beta}-2\bar{\alpha})w+2}{1-\alpha z-\bar{\alpha} w} & -\frac{(\beta-\alpha)z+\bar{\alpha} w+1}{1-\alpha z-\bar{\alpha} w} \\ 1 & 0 \end{bmatrix}.$$

An easy computation shows that $|eig(\mathbf{S})| < 1$ if and only if

$$w > \frac{4\alpha - 2\beta}{\bar{\beta} - 2\bar{\alpha}} z - \frac{4}{\bar{\beta} - 2\bar{\alpha}}$$

and the stability region is shown in Figure 5.2. We can observe that when $M \rightarrow \infty$ then $\alpha, \beta, \bar{\beta} \rightarrow 1$, $\bar{\alpha} \rightarrow 0$, and the stability region tends to that of classical Euler method, that is the region characterized by

$$w > 2z - 4$$

and represented in Figure 5.1.

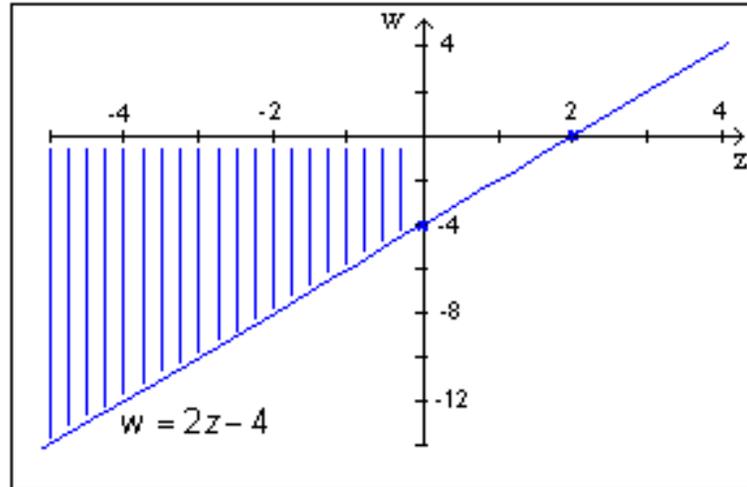


Figure 5.1: Stability region of the classical implicit Euler VRK method

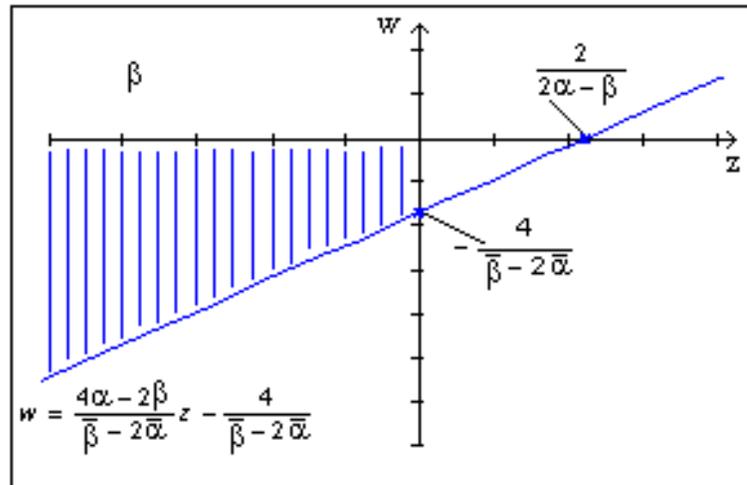


Figure 5.2: Stability region of the implicit Euler FVRK method

Remark 5.9. We present below the stability regions of the Radau II FVRK method with respect the convolution test equation (5.5.17). In Figure 5.3 we report the plot of the stability region of the 3-points classical Radau II VRK method, while in Figures 5.4-5.6 we report the plots of the stability regions of the corresponding Radau II FVRK method, with different values of the number M of points on the Talbot contour. The plots show as for M not very large (i.e. $M = 61$) the stability region of the FVRK method is very close to that of the corresponding classical method.

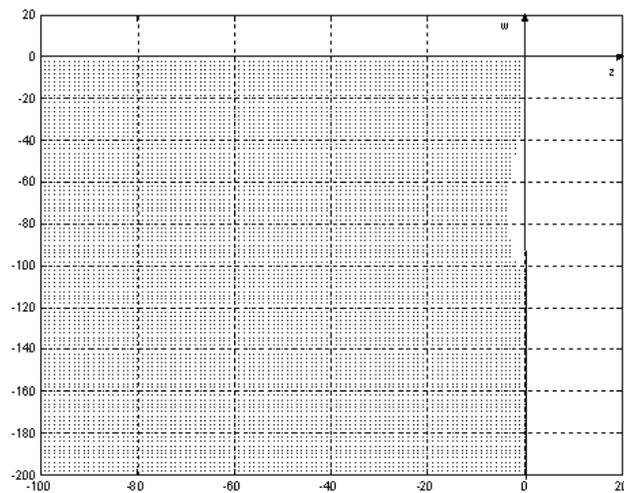


Figure 5.3: Stability region of the classical Radau II VRK method

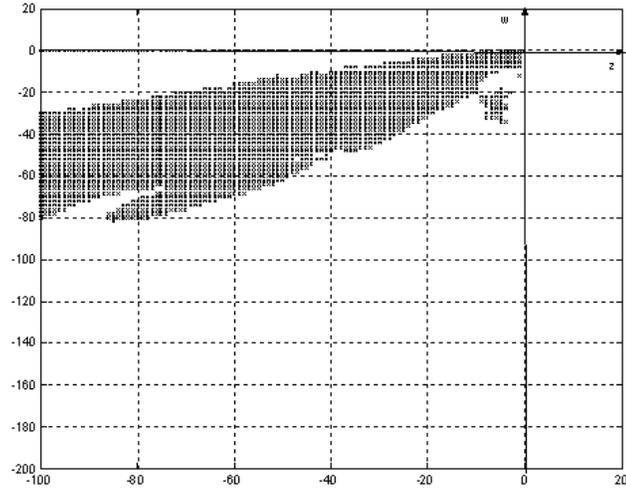


Figure 5.4: Stability region of the Radau II FVRK method ($M = 7$).

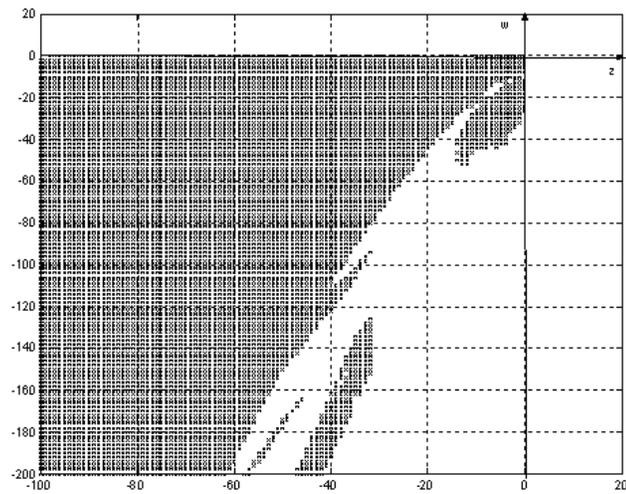


Figure 5.5: Stability region of the Radau II FVRK method ($M = 11$).

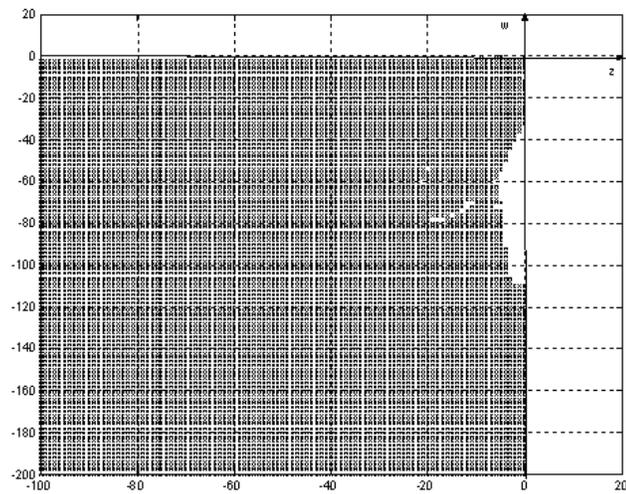


Figure 5.6: Stability region of the Radau II FVRK method ($M = 61$).

5.6 Numerical results

In the numerical experiments we tested the performances of the FVRK methods in terms of order of convergence and computational cost, in order to validate the theoretical results of Sections 5.3 and 5.4. The methods were implemented in MATLAB and the numerical experiment have been performed on different text examples. Here we report the results obtained on the two test problems:

- the linear Volterra integral equation of renewal theory taken from [18],[32]:

$$y(t) = 1 - e^{-\lambda t}(1 + \lambda t) + \int_0^t \lambda^2(t - \tau)e^{-\lambda(t-\tau)}y(\tau)d\tau \quad t \in [0, 10], \quad (5.6.1)$$

with $K(s) = \left(\frac{\lambda}{\lambda+s}\right)^2$, $\lambda = 1/2$, and exact solution $y(t) = \frac{1}{4}(2\lambda t - 1 + e^{-2\lambda t})$;

- the nonlinear equation given in [52], arising in the analysis of neural networks with post inhibitory rebound:

$$y(t) = 1 + \int_0^t (t - \tau)^3(4 - t + \tau)e^{-t+\tau} \frac{y^4(\tau)}{1 + 2y^2(\tau) + 2y^4(\tau)} d\tau, \quad t \in [0, 10], \quad (5.6.2)$$

with $K(s) = \frac{24s}{(1+s)^5}$ and reference solution $y(10) = 1.25995582337233$, obtained numerically by using different codes with very stringent tolerances.

Remark 5.10. We remind that the application of the FVRK methods requires the choice of a family of Talbot contours, and this choice is carried out as described in Section 3.4.3. We observe that the functions $K(s)$ of test examples (5.6.1) and (5.6.2) have only one singularity respectively at $s = -\lambda$ and $s = -1$. Thus, in both cases, the Talbot contours are chosen as in Example 3.3. The number $M = 2N + 1$ of quadrature points on each Talbot contour is chosen to be $M = 501$.

5.6.1 Convergence

The following FVRK methods have been used, where p denotes the order of the method, according to Theorem 5.1 and Corollary 5.1:

IMPLICIT METHODS:

$L2$: 2-points Lobatto ($c_1 = 0, c_2 = 1$), $p = 2$;

$R3$: 3-points Radau II ($c_1 = \frac{4-\sqrt{6}}{10}, c_2 = \frac{4+\sqrt{6}}{10}, c_3 = 1$), $p = 5$;

$G3$: 3-points Gauss ($c_1 = \frac{5-\sqrt{15}}{10}, c_2 = 1/2, c_3 = \frac{5+\sqrt{15}}{10}$), $p = 6$.

EXPLICIT METHODS:

$E3$: 3-points with Butcher array:

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ 2/3 & 2/3 & 0 & 0 \\ 2/3 & 5/12 & 1/4 & 0 \\ \hline & 1/4 & -1/4 & 1 \end{array}, \quad p = 3;$$

$E4$: classical 4-points with Butcher array:

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ \hline & 1/6 & 1/3 & 1/3 & 1/6 \end{array}, \quad p = 4.$$

The number of correct significant digits cd at the end point is defined to be

$$cd := -\log_{10} (|y(T) - y_{N_t}|/|y(T)|).$$

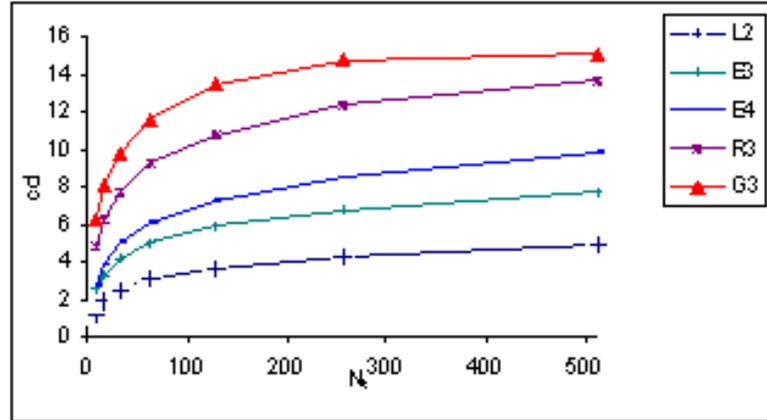


Figure 5.7: Number of correct significant digits for problem (5.6.1)

Table 5.1: Number of correct significant digits for problem (5.6.1) at $t=T=10$.

method	$N_t = 16$	$N_t = 32$	$N_t = 64$	$N_t = 128$	$p(h = \frac{10}{128})$
L2	1,82	2,41	3,01	3,61	1,99
E3	3,23	4,07	4,94	5,83	2,96
E4	3,78	4,93	6,10	7,29	3,95
R3	6,18	7,66	9,16	10,67	5,02
G3	7,98	9,76	11,56	13,35	5,95

Table 5.2: Number of correct significant digits for problem (5.6.2) at $t=T=10$.

method	$N_t = 16$	$N_t = 32$	$N_t = 64$	$N_t = 128$	$N_t = 256$	$p(h = \frac{10}{256})$
L2	2,82	3,27	3,84	4,44	5,04	1,99
E3	3,05	4,14	5,27	6,41	7,35	3,12
E4	3,38	4,96	6,51	7,20	8,26	3,52
R3	4,73	6,45	8,33	9,91	11,41	4,98
G3	6,35	8,62	9,26	11,53	13,17	5,45

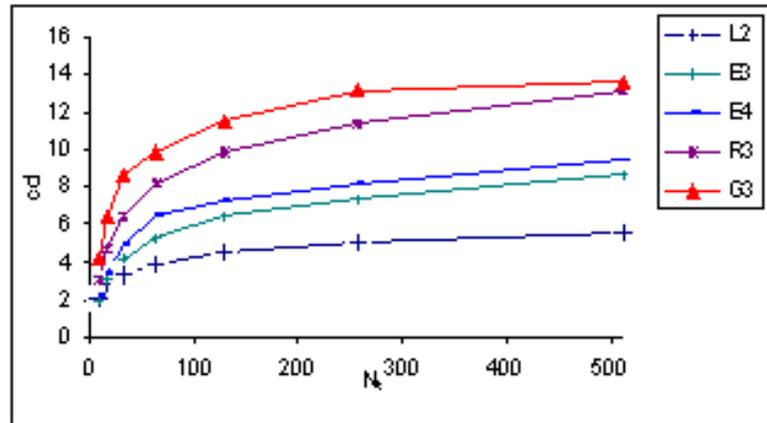


Figure 5.8: Number of correct significant digits for problem (5.6.2)

In Figures 5.7-5.8 we report the value of cd obtained by the application of each method respectively to the equations (5.6.1) and (5.6.2), with respect the number N_t of mesh points. In Tables 5.1-5.2 we report the numerical results and we compute a numerical estimation of the order of the method with the formula $p(h) = \frac{cd(h) - cd(2h)}{\log_{10} 2}$ for a fixed h , which shows that our methods produce the expected order.

5.6.2 Computational cost

In order to verify that our methods have a computational cost of order $\mathcal{O}(N_t \log_B N_t)$ in Figure 5.9 we plot for equation (5.6.1) the cpu-time in seconds versus N_t obtained from an Intel Pentium 4/3,2 GHz. The rhombuses correspond to R3, while the solid line correspond to algorithms of order $\mathcal{O}(N_t)$ and $\mathcal{O}(N_t^2)$. The dashed line represents the function $C N_t \log_B N_t$, and the picture shows that our

method perfectly follows its behaviour.

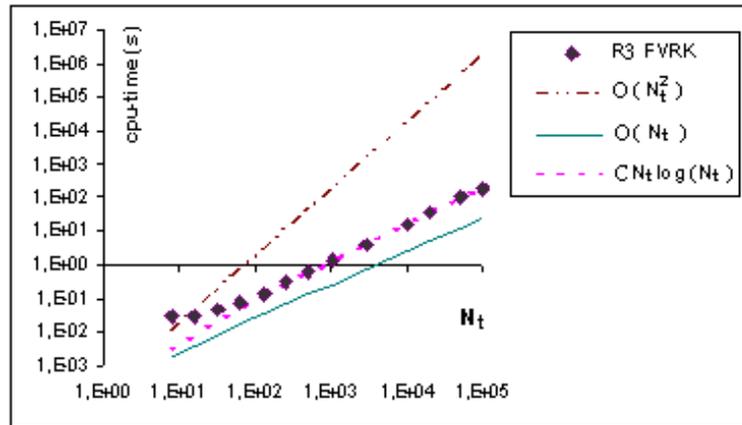


Figure 5.9: Computational cost

5.7 Concluding remarks

In this chapter we constructed fast VRK methods for the equation (5.1.1) of order p that can be implemented with a computational cost of $O(N_t \log N_t)$ operations and with a memory requirement of $O(\log N_t)$ to compute the numerical solution over N_t time steps. This was possible exploiting the knowledge of the Laplace transform of the kernel and that the equation is of convolution type.

We proved that the accuracy and stability properties depend on the number M of points chosen on the Talbot contour for the inverse Laplace transform approximation. In particular with a suitable choice of M , the order of convergence p is the same of the corresponding classical VRK methods, and the stability regions, for $M \rightarrow \infty$, tend to the classical ones.

The numerical experiments show that computational cost, the order of accuracy and stability regions of our constructed methods reflect the theoretical results.

The methods presented in this chapter are highly parallelizable and we think that they can be extended in a natural way to systems of Volterra integral equations.

Chapter 6

On the asymptotic periodicity
of the solutions of nonlinear
discrete Volterra equations of
convolution type

6.1 Introduction

Nonlinear Volterra integral equation of the form

$$y(t) = f(t) + q(t) + \int_0^t a(s)k(t-s)G(y(s))ds \quad t \in I := [0, T] \quad (6.1.1)$$

are of great interest in the applications. For example SIS epidemics with periodic immigration flow which confer no immunity and have a negligible incubation period have as mathematical model (see Example 1.2) a VIE with convolution kernel of the form

$$y(t) = f(t) + q(t) + \int_0^t a(s)k(t-s)y(s)(1-y(s))ds. \quad (6.1.2)$$

Such problems are characterized by the relapse of the epidemic which implies that the VIE (6.1.2) has an asymptotically periodic solution. In this chapter we are concerned with the approximate solution of (6.1.1) by means of efficient methods which “catch” the qualitative characteristics of the problem. In particular we are interested in numerical methods which generate an asymptotically periodic solution whenever the analytical solution shows this behaviour. For this reason we introduce discrete Volterra equations (DVEs) and we develop a theory equivalent to the one in [29] for difference equations of unbounded order. Section 6.2 contains the main results concerning the analytical solution of (6.1.1). In Section 6.3 we introduce the basic concepts of asymptotic periodicity for DVEs. In Section 6.4 we investigate the stability properties for DVEs of Hammerstein type and, in the case of constant contact rate, in Section 6.5, we give results on the asymptotical periodicity for problem (6.1.2). In Section 6.6 we investigate AP-stability of a class of numerical methods for (6.1.2) with constant contact rate.

6.2 Representation of analytical solution

In order to analyse the behaviour of numerical solutions of (6.1.1) we, of course, need information about the behaviour of its analytical solution. Since we are concerned with asymptotic periodicity we report here the existing analytical theory, mainly contained in [29], which represents the starting point of our discrete analysis. First we report the definition of asymptotically periodic function.

Definition 6.1. An asymptotically ω -periodic function $v(t)$ is a bounded continuous function for which there exists a continuous ω -periodic function $r(t)$ such that :

$$v(t + n\omega) - r(t) \rightarrow 0 \quad n \in \mathbb{N}$$

uniformly on $[0, \omega]$ as $n \rightarrow +\infty$.

Consider the limit equation associated to (6.1.1)

$$z(t) = q(t) + \int_{-\infty}^t a(s)k(t-s)G(z(s))ds, \quad t \in \mathbb{R} \quad (6.2.1)$$

The following theorem gives conditions for the solution of (6.1.1) to be asymptotically ω -periodic.

Theorem 6.1. *Assume that $k \in L_1(0, \infty)$, a , f , G and q are continuous function, a and q are ω -periodic and $f(t) \rightarrow 0$ as $t \rightarrow +\infty$. Suppose that (6.2.1) has an unique continuous solution satisfying $0 < \alpha \leq z(t) \leq \beta$ for some α, β with $\alpha < \beta$. Suppose also that (6.1.1) has a unique continuous solution satisfying $0 < \alpha \leq y(t) \leq \beta$ for all sufficiently large t . Then $z(t)$ is ω -periodic and $y(t + n\omega) - z(t) \rightarrow 0$ uniformly on $[0, \omega]$ as $n \rightarrow +\infty$ that is $y(t)$ is asymptotically ω -periodic.*

Remark 6.1. See [29] for a detailed proof.

Remark 6.2. Similar theorem can be found in Diekmann [38], Diekmann and Kaper [37]

As we have already observed, the situations when the function G assumes the form $G = y(1 - y)$ are often encountered in the applications. In [29] two interesting cases of equation (6.1.2) that arise in the applications are analyzed. One is the case of periodic contact rates and another is the case of periodic immigration flow and constant contact rates. In both cases it is possible to prove the existence and the uniqueness of the asymptotically periodic solution of (6.1.2).

a) *Periodic contact rates*

Consider the equation

$$y(t) = f(t) + \int_0^t a(s)k(t-s)y(s)(1-y(s))ds \quad t \in I := [0, T] \quad (6.2.2)$$

and assume that

1. a is a bounded, continuous, ω -periodic and positive function;
2. k is differentiable and $k(t) > 0$, $k(0) = 1$, $k'(t) \leq 0$ and $k(t) \in L^1(0, \infty)$;
3. $f(t)$ is positive, differentiable and $f(t) \rightarrow 0$ as $t \rightarrow +\infty$.

Assume also that $\|f\| + (\sup_{t \geq 0} \int_0^t a(s)k(t-s)ds)/4 \leq \frac{1}{2}$, $\bar{a} \leq a(s) \leq \bar{\bar{a}}$ and let $\bar{x} = 1 - (1/\int_0^\infty \bar{a}k(t)dt)$ and $\bar{\bar{x}} = 1 - (1/\int_0^\infty \bar{\bar{a}}k(t)dt)$.

With this assumptions it is possible to prove the following theorem

Theorem 6.2. *If $\bar{x} > \bar{\bar{x}}/2$ and $1 < \int_0^\infty \bar{a}k(t)dt \leq \int_0^\infty \bar{\bar{a}}k(t)dt < 2$ then the equation 6.2.2 has a nonzero asymptotically periodic solution. If $\int_0^\infty \bar{\bar{a}}k(t)dt \leq 1$ then the solution of 6.2.2 tends to 0 as $t \rightarrow \infty$.*

b) *Periodic Immigration*

Consider the equation

$$y(t) = f(t) + q(t) \int_0^t ak(t-s)y(s)(1-y(s))ds \quad t \in I := [0, T]. \quad (6.2.3)$$

Here the contact rate a is constant while the fraction of infective immigrants q is assumed ω -periodic.

Theorem 6.3. *Let $A = \int_0^{+\infty} ak(t)dt$. Assume that $A > 1$ and $k(t)$ is a differentiable function such that $k(t) = 0$, $k(0) = 1$, $k'(t) \leq 0$, $k(t) \in L^1(0, \infty)$. Also assume that $f(t) \rightarrow 0$ as $t \rightarrow \infty$; $q(t)$ is a positive ω -periodic function and $\|f + q\| + \frac{A}{4} \leq \frac{1}{2}$. Then (6.2.3) has a positive ω -periodic solution.*

Remark 6.3. See [29] for a detailed proof.

6.3 DVEs with asymptotically periodic solution

Now first of all we give the definition of asymptotically periodic sequence and then we make clear what we mean for discrete Volterra equations to have an asymptotically periodic solution.

Definition 6.2. A sequence v_n is *asymptotically M-periodic* if it is bounded and if there exists a M-periodic sequence r_n such that

$$v_{n+\nu M} - r_n \rightarrow 0 \quad n = 0, \dots, M \quad \text{as } \nu \rightarrow +\infty$$

For example

$$v_n = e^{-n\pi} + \alpha \sin\left(n\frac{2\pi}{M}\right).$$

is a bounded sequence and $v_{n+\nu M} \rightarrow r_n$ for $\nu \rightarrow +\infty$ where $r_n = +\alpha \sin\left(n\frac{2\pi}{M}\right)$ which is obviously an M-periodic sequence. Hence v_n is asymptotically M-periodic.

A general discrete Volterra equation reads

$$y_n = f_n + \sum_{j=0}^n w_{nj} k(t_n, t_j, y_j) \quad n \geq 0 \quad (6.3.1)$$

The motivation for studying DVEs stands in the fact that most of the numerical methods for VIEs fit into this class.

Now the solution of (6.3.1) is asymptotically periodic if y_n is an asymptotically periodic sequence.

In [26], [42] sufficient conditions for a DVE to have a unique asymptotically periodic solution are given for the linear case. In the next section we will see how to extend the theory to a special class of nonlinear DVEs.

6.4 Asymptotical periodicity for Hammerstein type DVEs

We consider the discrete analogue of (6.1.1)

$$y_n = f_n + \sum_{j=0}^n a_j k_{n-j} G(y_j) \quad n \geq 0 \quad (6.4.1)$$

with $f_n = q_n + g_n$, and we make the following assumptions: q_n and a_n are M -periodic sequences and $g_n \rightarrow 0$ $n \rightarrow +\infty$.

In order to find conditions for the solution of (6.4.1) to be asymptotically M -periodic we need the following lemma:

Lemma 6.1. *Suppose f_n and a_n are bounded, $\sum_{j=0}^{+\infty} k_j < +\infty$ and G is a bounded and uniformly continuous function on R^+ . Then for any sequence $m_k \rightarrow +\infty$ there exist three sequences z_n, q_n, a_j and a subsequence m_{k_p} such that*

$$\begin{aligned} y_{n+m_{k_p}} &\rightarrow z_n \\ f_{n+m_{k_p}} &\rightarrow q_n \\ a_{j+m_{k_p}} &\rightarrow a_j. \end{aligned} \quad (6.4.2)$$

Moreover z_n is the solution of

$$z_n = q_n + \sum_{j=-\infty}^n a_j k_{n-j} G(z_j). \quad (6.4.3)$$

Proof. The boundness of y_n easily follows from the hypothesis. Consider the sequence of the translated equations

$$y_{n+m_k} = f_{n+m_k} + \sum_{j=0}^{n+m_k} a_j k_{n+m_k-j} G(y_j) = f_{n+m_k} + \sum_{j=-m_k}^n a_{j+m_k} k_{n-j} G(y_{j+m_k}).$$

where m_k is such that $m_k \rightarrow +\infty$.

If $n + m_k > 0$ then of course y_{n+m_k} , f_{n+m_k} and a_{n+m_k} are bounded and thus (6.4.2) is proved.

Now we have to show that

$$\left| \sum_{j=-m_{k_p}}^n a_{j+m_{k_p}} k_{n-j} G(y_{j+m_{k_p}}) - \sum_{j=-\infty}^n a_j k_{n-j} G(z_j) \right| \rightarrow 0, \quad p \rightarrow +\infty.$$

Since a_n and the function G are bounded there exist two positive constants H and K such that

$$\begin{aligned} & \left| \sum_{j=-m_{k_p}}^n a_{j+m_{k_p}} k_{n-j} G(y_{j+m_{k_p}}) - \sum_{j=-\infty}^n a_j k_{n-j} G(z_j) \right| \leq \\ & HK \sum_{j=-m_{k_p}}^{-p} |k_{n-j}| + H \sum_{j=-p}^n |k_{n-j}| \left| G(y_{j+m_{k_p}}) - G(z_j) \right| - HK \sum_{j=-\infty}^{-p} |k_{n-j}| \leq \\ & \leq 2HK \sum_{j=n+p}^{m_{k_p}+n} |k_j| + H \sum_{j=-p}^n |k_{n-j}| \left| G(y_{j+m_{k_p}}) - G(z_j) \right| \end{aligned}$$

The last expression goes to zero when p tends to infinity. Hence z_n is the solution of (6.4.3). \square

Remark 6.4. From this proof it is clear that (6.4.3) corresponds to the limit equation of (6.4.1).

The following theorem is the discrete analogue of Theorem 6.1 and gives conditions under which (6.4.1) has an asymptotically M-periodic solution.

Theorem 6.4. *In the hypothesis of the previous lemma if (6.4.3) has a unique solution z_n satisfying $0 < \alpha \leq z_n \leq \beta$ for some α, β with $\alpha < \beta$ and (6.4.1) has a unique bounded solution y_n satisfying $0 < \alpha \leq y_n \leq \beta$ for all sufficiently large n , then z_n is M-periodic and y_n is asymptotically M-periodic.*

Proof. To see that z_n is M-periodic we look at z_{n+M}

$$z_{n+M} = q_{n+M} + \sum_{j=-\infty}^n a_{j+M} k_{n-j} G(z_{j+M})$$

Since a_n and q_n are M-periodic both z_{n+M} and z_n are solutions of (6.4.3) and by uniqueness

$$z_{n+M} = z_n.$$

Now we have to prove that

$$y_{n+\nu M} - z_n \rightarrow 0 \quad \text{as } \nu \rightarrow +\infty \quad \text{for } n = 0, \dots, M,$$

that is:

$$\forall \epsilon > 0 \quad \exists p_\epsilon : \quad |y_{n+\nu M} - z_n| < \epsilon \quad \forall \nu > p_\epsilon.$$

Suppose this is not true, so

$$\exists \epsilon > 0 \quad \exists \nu_k \quad \exists n_k \in \{0, \dots, M\} : \quad |y_{n_k+\nu_k M} - z_{n_k}| > \epsilon. \quad (6.4.4)$$

Since for all sufficiently large n , $0 < \alpha \leq y_n \leq \beta$ we may assume that $0 < \alpha \leq y_{n_k+\nu_k M} \leq \beta$ for all k . By Lemma 6.1 there exists a subsequence ν_{k_j} of ν_k such that $y_{n+\nu_{k_j} M} \rightarrow z_n$ with z_n satisfying (6.4.3) that is

$$\forall \epsilon > 0 \quad \exists p_\epsilon : \quad |y_{n+\nu_{k_j} M} - z_n| < \epsilon \quad \forall \nu_{k_j} > p_\epsilon.$$

This is in contradiction with (6.4.4), so $y_{n+\nu M} - z_n \rightarrow 0$. □

6.5 Applications

In this section we suppose that the function G in (6.4.1) has the form $G(y_j) = y_j(1 - y_j)$ and consider the particular case where the sequence a is constant then the equation (6.4.1) reads

$$y_n = f_n + q_n + \sum_{j=0}^n a k_{n-j} y_j (1 - y_j). \quad (6.5.1)$$

Observe that (6.5.1) is the discrete analogue of (6.2.3).

Here we suppose that

1. f_n is a bounded and nonnegative sequence, $f_n \rightarrow 0$ for $n \rightarrow +\infty$
2. q_n is a bounded, nonnegative and M -periodic sequence,
3. k_n is a positive and non increasing sequence such that $k_0 = 1$ and $\sum_{j=0}^{+\infty} k_j < +\infty$,
4. a is a positive constant.

Define $A = \sum_{j=0}^{+\infty} a k_j$, $\|f_n\| = \sup_n f_n$, $\hat{x} = 1 - \frac{1}{\sum_{j=0}^{+\infty} a k_j}$ and assume that $\|f_n + q_n\| + \frac{A}{4} \leq \frac{1}{2}$.

With these assumptions we are able to state the main result in this section.

Theorem 6.5. *The equations (6.5.1) has a positive asymptotically M -periodic solution.*

Remark 6.5. With this theorem it is possible to study the asymptotic periodicity of y_n without passing through the knowledge of the behaviour of the limit equations.

Proof. The theorem is proved using Theorem 6.4 whose hypothesis are established by the following lemmas. □

Lemma 6.2. *One of the solution \hat{y}_n of (6.5.1) is such that $0 \leq \hat{y}_n \leq \frac{1}{2} \quad \forall n \geq 0$.*

Proof. First we prove that (6.5.1) has a positive solution. We proceed by induction over n . For $n = 0$ we have

$$y_0 = f_0 + q_0 + ak_0y_0(1 - y_0).$$

It is easy to show that one of the two roots of this quadratic equation is positive. Let \hat{y}_0 be this positive root and assume that for $n = 1, \dots, k$ (6.5.1) has two roots one of which positive. Let \hat{y}_i be this positive root for $i = 1, \dots, k$. Now we look at $n = k + 1$

$$y_{k+1} = f_{k+1} + q_{k+1} + \sum_{j=0}^k ak_{k-j}\hat{y}_j(1 - \hat{y}_j) + ak_0y_{k+1}(1 - y_{k+1}) \quad (6.5.2)$$

Here $\Delta = (1 - ak_0)^2 + 4ak_0(f_{k+1} + q_{k+1} + \sum_{j=0}^k ak_{k-j}\hat{y}_j(1 - \hat{y}_j))$ and for our induction hypothesis $\hat{y}_n > 0, \quad \forall n$. Since

$$\begin{aligned} \hat{y}_n &\leq f_n + q_n + \sum_{j=0}^n ak_{n-j}\frac{1}{4} \leq \sup_n(f_n + q_n) + \sum_{j=0}^n ak_j\frac{1}{4} \leq \quad (6.5.3) \\ &\leq \sup_n(f_n + q_n) + \frac{\sum_{j=0}^{+\infty} ak_j}{4} \leq \frac{1}{2}. \end{aligned}$$

then for each $n = 1, 2, \dots$ $0 \leq \hat{y}_n \leq \frac{1}{2}$ and the assertion of the lemma is proved. \square

Lemma 6.3. *Let*

$$\begin{aligned} x_n &= v_n + \sum_{j=0}^n ak_{n-j}x_j(1 - x_j) \\ y_n &= g_n + \sum_{j=0}^n ak_{n-j}y_j(1 - y_j) \end{aligned}$$

with $v_n, g_n > 0$. If $v_n > g_n \quad \forall n \geq 0$, then $x_n > y_n \quad \forall n \geq 0$.

Proof. By induction over n . For $n = 0$ it is easy to prove that $x_0 > y_0$. Now we assume that $x_n > y_n$ $n = 0, \dots, k$. For $n = k + 1$ we have

$$x_{k+1} = \frac{-(1+a) + \sqrt{(1+a)^2 + 4a(v_{k+1} + \sum_{j=0}^k ak_{k-j}x_j(1-x_j))}}{2a}$$

$$y_{k+1} = \frac{-(1+a) + \sqrt{(1+a)^2 + 4a(g_{k+1} + \sum_{j=0}^k ak_{k-j}y_j(1-y_j))}}{2a}$$

And thus by our induction hypothesis $x_n > y_n$, $\forall n \geq 0$. \square

Lemma 6.4. *Let*

$$z_n = q_n + \sum_{j=-\infty}^n ak_{n-j}z_j(1-z_j) \quad (6.5.4)$$

be the limit equation associated to (6.5.1), then (6.5.4) has a unique solution satisfying $b\hat{x} \leq z_n \leq \frac{1}{2}$, where $\frac{1}{2} < b < 1$.

Proof. Let $BC(R)$ be a closed ball of radius R and $S = \{x_n \in BC(R) : b\hat{x} \leq x_n \leq \frac{1}{2}, \frac{1}{2} < b < 1\}$. Define the operator T as

$$Tx_n = q_n + \sum_{j=-\infty}^n ak_{n-j}x_j(1-x_j).$$

Consider $z_n \in S$ and apply T on z_n

$$\begin{aligned} Tz_n &= q_n + \sum_{j=-\infty}^n ak_{n-j}z_j(1-z_j) \geq \sum_{j=-\infty}^n ak_{n-j} \min(z_j(1-z_j)) \\ &= b\hat{x}(1-b\hat{x}) \sum_{j=0}^{+\infty} ak_j = b\hat{x}(1-b\hat{x}) \frac{1}{1-\hat{x}} \geq b\hat{x}, \end{aligned} \quad (6.5.5)$$

the last inequality coming from the fact that $\frac{1-b\hat{x}}{1-\hat{x}} \geq 1$. What's more

$$Tz_n \leq \sup_n q_n + \sum_{j=-\infty}^n ak_{n-j} \max(z_j(1-z_j)) \leq \frac{1}{4} \sum_{j=0}^{+\infty} ak_j + \frac{1}{2} - \sum_{j=0}^{+\infty} ak_j \leq \frac{1}{2},$$

so Tz_n belongs to S .

We also prove that T is a contraction over S . Let u, v in S

$$\begin{aligned} \|Tu_n - Tv_n\| &\leq \sup_n \sum_{j=-\infty}^n |ak_{n-j}(u_j - v_j)(1 - u_j - v_j)| \quad (6.5.6) \\ &\leq \sup_n \sum_{j=-\infty}^n |ak_{n-j}(u_j - v_j)(1 - 2b\hat{x})| \\ &\leq \frac{1 - 2b\hat{x}}{1 - \hat{x}} \|u_n - v_n\|. \end{aligned}$$

Thus $T : S \rightarrow S$ is a contraction since $\frac{1-2b\hat{x}}{1-\hat{x}} \leq 1$. So there exists a unique $z_n \in S$ such that $Tz_n = z_n$ with $b\hat{x} \leq z_n \leq \frac{1}{2}$. \square

Lemma 6.5. *For all sufficiently large n , the solution y_n of (6.5.1) is such that $b\hat{x} \leq y_n \leq \frac{1}{2}$.*

Proof. Let m be a positive number such that $0 < m < \min_n \{q_n\}$. Let r_n be a solution of

$$r_n = f_n + m + \sum_{j=0}^n ak_{n-j}r_j(1 - r_j)$$

By Lemma 6.3 $y_n > r_n$ for $n \geq 0$. Moreover $r_n \rightarrow \hat{r}$ where \hat{r} is the positive solution to $r = m + \sum_{j=0}^{+\infty} ak_j r(1 - r)$ and $\hat{r} > \hat{x}$ where \hat{x} is the positive solution to $x = \sum_{j=0}^{+\infty} ak_j x(1 - x)$. Thus for any b such that $\frac{1}{2} < b < 1$, $b\hat{x} \leq y_n$ if n is sufficiently large. \square

6.6 A class of AP-stable methods

In the previous section we prove sufficient conditions for the existence and the uniqueness of the asymptotically periodic solution of the DVE (6.5.1).

In this section we investigate the properties of a numerical method to preserve the asymptotic periodicity of the solution of (6.2.3). We consider the class of θ -methods applied to the equation (6.2.3) and we prove that they produce an asymptotically periodic solution if the integration step satisfies an inequality depending only on some parameters that are characteristic of the problem. First of all we need the following definitions.

Definition 6.3. A numerical method for (6.2.3) is AP-stable if it produces an asymptotically periodic solution whenever applied to a Volterra integral equation with asymptotically periodic solution.

Definition 6.4. A numerical method for (6.2.3) is conditionally AP-stable if it is AP-stable for all $h < h_0$ with $h_0 > 0$.

We focus our attention to the one-point collocation method described in Example 2.1 and we analyse the behaviour of the corresponding numerical solution when applied to the equation (6.2.3).

Let $I_h = \{t_n := nh, n = 0, \dots, N, h \geq 0\}$ be a uniform mesh on $[0, T]$, we approximate the unique solution of (6.2.3) by elements in piecewise polynomial space

$$S_0^{(-1)}(I_h) = \{u : u|_{(t_n, t_{n+1})} = \text{const}, n \geq 0\},$$

of dimension N . The approximation $u \in S_0^{(-1)}(I_h)$ will be required to satisfy the given integral equation (6.2.3) on the set of collocation points

$$\{t_{n,1} := t_n + \theta h, 0 \leq \theta \leq 1, n = 0, \dots, N - 1\}.$$

Let us denote $y_{n+1} = u_n(t_n + \nu h)$ $\nu \in [0, 1]$, then we are looking for the solution of

$$y_{n+1} = f(t_{n1}) + h \sum_{l=0}^{n-1} \omega_{n-l}(\theta)(y_{l+1}(1 - y_{l+1})) + h\omega_0(\theta)(y_{n+1}(1 - y_{n+1})) \quad (6.6.1)$$

where

$$\begin{cases} \omega_{n-l}(\theta) = \int_0^1 ak((n + \theta - l - s)h)ds & l < n \\ \omega_0(\theta) = \int_0^\theta ak((\theta - s)h)ds. \end{cases}$$

If the integrals in (6.6.1) cannot be found analytically, we have to resort to suitable quadrature formulas for their approximation. Here we choose the interpolatory one point quadrature formulas whose abscissas are, respectively, $t_l + \theta h$ if $l < n$, and $\theta^2 h$ (fully discretized) or θh (discretized) if $l = n$.

The discretized version of (6.6.1) is thus given by

$$\hat{y}_{n+1} = f(t_{n1}) + h \sum_{l=0}^{n-1} \hat{\omega}_{n-l} \hat{y}_{l+1}(1 - \hat{y}_{l+1}) + h\hat{\omega}_0(\theta)\hat{y}_{n+1}(1 - \hat{y}_{n+1}) \quad (6.6.2)$$

where we have defined

$$\begin{cases} \hat{\omega}_{n-l} = ak((n - l)h) & l < n \\ \hat{\omega}_0(\theta) = \theta ak(\theta(1 - \theta)h) & \text{fully discretized} \\ \hat{\omega}_0(\theta) = \theta ak(0) & \text{discretized.} \end{cases}$$

Notice that, since the integral equation (6.2.3) is of convolution type, $\hat{\omega}_{n-l}$ does not depend on θ . Now we derive some useful properties for the weights ω_j in (6.6.1).

Lemma 6.6. *If equation (6.2.3) satisfies the hypothesis of Theorem 6.3, then $0 < \omega_{n-l}(\theta) < a$ and $\omega_{n+1-l}(\theta) - \omega_{n-l}(\theta) \leq 0$ $l \leq n$.*

Proof. Since $a > 0$ and $k(t)$ is a nonnegative function then $\omega_{n-l}(\theta) > 0$. What's more since $k(t)$ is a nonincreasing function

$$\int_0^1 k((n + \theta - l - s)h)ds \leq \int_0^1 k(0)ds = 1$$

thus $0 < \omega_{n-l}(\theta) < a$. In order to prove the second part of the lemma we observe that

$$\int_0^1 ak((n+1+\theta-l-s)h)ds \leq \int_0^1 ak((n+\theta-l-s)h)ds$$

and hence

$$\omega_{n+1-l}(\theta) \leq \omega_{n-l}(\theta).$$

□

Remark 6.6. The same result holds for $\hat{\omega}_{n,l}(\theta)$.

6.6.1 AP-stability of the exact collocation method

Here we assume that the integrals in (6.6.1) can be found analytically and we suppose that equation (6.2.3) has an asymptotically periodic solution, that is it satisfies the hypothesis of Theorem 6.3. In this case we are able to state the following result:

Theorem 6.6. *The class of methods (6.6.1) is AP-stable.*

Proof. Let us rewrite the method (6.6.1) as

$$y_{n+1} = f(t_{n1}) + q(t_{n1}) + \sum_{l=0}^n Ak_{n-l}y_{l+1}(1 - y_{l+1}) \quad (6.6.3)$$

where

$$\begin{cases} Ak_{n-l} = \int_{t_l}^{t_{l+1}} ak(t_n + \theta h - s)ds \\ Ak_0 = \int_{t_n}^{t_n + \theta h} ak(t_n + \theta h - s)ds. \end{cases}$$

From the hypothesis of Theorem 6.3 for the equation (6.2.3) immediately follows that in (6.6.1) $f(t_{n1}) \rightarrow 0$ $t \rightarrow \infty$, $q(t_{n1})$ is positive and ω -periodic, k_n

is a positive and nonincreasing sequence such that $k_0 = 1$, $\sum_{j=0}^{+\infty} k_j < +\infty$. In order to apply Theorem 6.5 we only need to show that

$$\|f(t_{n1}) + q(t_{n1})\| + \frac{\sum_{l=0}^n Ak_{n-l}}{4} \leq \frac{1}{2} \quad (6.6.4)$$

Since $k(t)$ is a positive decreasing kernel we have

$$\sum_{l=0}^{n-1} \int_{t_l}^{t_{l+1}} ak(t_n + \theta h - s) ds < \sum_{l=0}^{n-1} \int_{t_l}^{t_{l+1}} ak(t_n - s) ds$$

and

$$\int_{t_n}^{t_n + \theta h} ak(t_n + \theta h - s) ds < \int_{t_n}^{t_n + \theta h} ak(t_n - s) ds.$$

Observing that

$$\sum_{l=0}^{n-1} \int_{t_l}^{t_{l+1}} ak(t_n - s) ds + \int_{t_n}^{t_n + \theta h} ak(t_n - s) ds \leq \int_0^T ak(t_n - s) ds < \int_{t_n}^{+\infty} ak(t) dt,$$

then

$$\|f(t_{n1}) + q(t_{n1})\| + \frac{\sum_{l=0}^n Ak_{n-l}}{4} \leq \|f + q\| + \frac{A}{4}$$

thus by Theorem 6.3, (6.6.4) holds. \square

6.6.2 AP-stability of the discretized collocation method

Now we analyse the AP-stability property of the discretized collocation method (6.6.2), when applied to the equation (6.2.3) satisfying the hypothesis of Theorem 6.3.

Theorem 6.7. *The discretized collocation method (6.6.2) is AP-stable for*

$$h \leq \frac{4}{a\theta} \left[\frac{1}{2} - \|f + q\| - \frac{\int_0^{+\infty} ak(t) dt}{4} \right],$$

that is (6.6.2) is conditionally AP-stable.

Proof. Equation (6.6.2) can be rewritten as

$$y_{n+1} = f(t_{n1}) + q(t_{n1}) + \sum_{l=0}^n Ak_{n-l}y_{l+1}(1 - y_{l+1}) \quad (6.6.5)$$

where

$$Ak_{n-l} = ahk((n-l)h) \quad l < n$$

$$\begin{cases} Ak_0 = a\theta hk(\theta(1-\theta)h) & \text{fully discretized} \\ Ak_0 = a\theta hk(0) & \text{discretized} \end{cases}$$

Like in Theorem 6.6 we only have to show that

$$\|f(t_{n1}) + q(t_{n1})\| + \frac{\sum_{l=0}^{n-1} Ak_{n-l} + Ak_0}{4} \leq \frac{1}{2} \quad (6.6.6)$$

For a positive decreasing kernel k

$$\sum_{l=0}^{n-1} Ak_{n-l} + Ak_0 \leq \int_0^t ak(t)dt + Ak_0 \leq \int_0^{+\infty} ak(t)dt + Ak_0.$$

Since $Ak_0 = a\theta h$ ($k(0) = 0$ for Theorem 6.3) for the discretized version and $Ak_0 \leq a\theta$ for the fully discretized one, it follows that (6.6.6) holds if

$$h \leq \frac{4}{a\theta} \left[\frac{1}{2} - \|f + q\| - \frac{\int_0^{+\infty} ak(t)dt}{4} \right].$$

Notice that the expression in the square brackets is certainly positive by Cromer's Theorem 6.3.

6.7 Concluding remarks

This chapter is concerned with the numerical treatment of problems of *SIS* epidemic diffusion with periodic immigration flow [29]. The mathematical model of such problems is represented by an Hammerstein type VIE with convolution kernel of the form

$$y(t) = f(t) + q(t) + \int_0^t a(s)k(t-s)y(s)(1-y(s))ds. \quad (6.7.7)$$

We consider here problems characterized by the relapse of the epidemic which implies that the VIE (6.7.7) has an asymptotically periodic solution.

We analyse the discrete Volterra equation (DVE) corresponding to problem (6.7.7) and we prove a theorem which establishes the existence and the uniqueness of the asymptotically periodic solution of the DVE.

Moreover we consider *SIS* epidemic models with periodic immigration flow and constant contact rate. Also in this case we prove, for the DVE corresponding to the problem, the existence and the uniqueness of the asymptotically periodic solution when the DVE satisfies some significant hypothesis depending only on its kernel and forcing term.

In order to analyse if the existing numerical methods satisfy these conditions, that is if they are AP-stable, we consider the class of θ -methods and we prove that they are AP-stable if the integration step satisfies an inequality depending only on some parameters that are characteristic of the problem.

Bibliography

- [1] R. P. Agarwal, D. O'Regan (eds.), *Integral and Integrodifferential Equations. Theory, Methods and Applications*, Ser. Math. Anal. App., 2, Amsterdam, Gordon and Breach (2000).
- [2] U. An Der Heiden, *Analysis of Neural Networks*, Lectures Notes in Biomathematics (1980).
- [3] P. M. Anselone (ed.), *Nonlinear Integral Equations (Madison 1963)*, Madison, University of Wisconsin Press (1964).
- [4] K. E. Atkinson, *Introduction to Numerical Analysis*, New York, Wiley (1989).
- [5] C.T.H.Baker, C.Philips, *The numerical solution of nonlinear problems*, Oxford Clarendon Press, 1981.
- [6] A. Bellen, Z. Jackiewicz, R. Vermiglio, M. Zennaro, *Stability Analysis of Runge-Kutta methods for Volterra Integral Equations of second kind*, IMA J. Num. Anal. 10, 103-118 (1990).
- [7] R. Bellman, K. L. Cooke, *Differential-Difference Equations*, New York, Academic Press (1983).

- [8] L. R. Berrone, *Local positivity of the solution of Volterra integral equations and heat conduction in materials that may undergo changes of phase*, Math. Notae 38, 79–93 (1995).
- [9] J. G. Blom, H. Brunner, *The Numerical Solution of nonlinear Volterra Integral Equations of the second kind by collocation and iterated collocation methods*, SIAM J. Sci. Statist. Comput. 8 no. 5, pp. 806-830 (1987).
- [10] J. G. Blom, H. Brunner, *Algorithm 689, Discretized collocation and iterated collocation for nonlinear Volterra integral equations of the second kind*, ACM Trans. Math. Softw. 17 no. 2, pp. 167-177 (1991).
- [11] F. Brauer, C. Castillo-Chávez, *Mathematical Models in Population Biology and Epidemiology*, New York, Springer-Verlag (2001).
- [12] F. Brauer, *Constant rate harvesting of population governed by Volterra integral equations*, J. Math. Anal. Appl. 56, 18–27 (1976).
- [13] F. Brauer, *On a nonlinear integral equation for population growth problems*, SIAM J. Math. Anal. 6, 312–317 (1975).
- [14] H. Brezis, F. E. Browder, *Existence theorems for nonlinear integral equations of Hammerstein type*, Bull. Amer. Math. Soc. 81, 73-78 (1975).
- [15] H. Brunner, *Collocation Methods for Volterra Integral and Related Functional Equations*, Cambridge University Press (2004).
- [16] H. Brunner, *On implicitly linear and iterated collocation methods for Hammerstein integral equations*, J. Integral Equations Appl. 3, 475–488 (1991).

- [17] H. Brunner, *A survey of recent advances in the numerical treatment of Volterra integral and integro-differential equations*, J. Comput. Appl. Math. 8, 213–229 (1982).
- [18] H. Brunner and P.J. van der Houwen, *The Numerical Solution of Volterra Equations*, CWI Monographs, Vol. 3, North-Holland, Amsterdam, (1986).
- [19] T. A. Burton, *Volterra Integral and Differential Equations*, Academic Press, New York, (1983).
- [20] G. Capobianco, D. Conte, I. Del Prete, E. Russo *Fast Runge-Kutta methods for nonlinear Volterra integral equations of convolution type*, Preprint n. 15/2005 del Dip. di Matematica ed Applicazioni dell'Università Napoli "Federico II".
- [21] D. Conte, I. Del Prete, *Fast collocation methods for Volterra integral equations of convolution type*, Preprint n. 12/2005 del Dip. di Matematica ed Applicazioni dell'Università Napoli "Federico II" (accepted for publication on J. Comput. Appl. Math.).
- [22] K. L. Cooke, J. L. Kaplan, *A periodicity threshold theorem for epidemics and population growth*, Math. Biosc. 31, 87–104 (1976).
- [23] C. Corduneanu, I. W. Sandberg (eds.), *Volterra Equations and Applications*, Stability Control, Theory, Methods Appl. 10, Amsterdam, Gordon and Breach (2000).
- [24] C. Corduneanu, *Integral Equations and Applications*, Cambridge University Press, Cambridge (1991).

- [25] C. Corduneanu, *Integral Equations and Stability of Feedback Systems*, Academic Press, New York (1973).
- [26] M.R. Crisci, E. Russo, A. Vecchio, *Periodic Solution of Whole Line Difference Equations*, J. Differ. Equations Appl. 10, 357–368 (2004).
- [27] M. R. Crisci, Vb. Kolmanovskii, E. Russo, A. Vecchio, *Stability of discrete Volterra equations of Hammerstein type*, J. Differ. Equations Appl. 6 no. 2, 127-145 (2000).
- [28] M.R.Crisci, E. Russo, A.Vecchio, *Stability analysis of the de Hoog and Weiss implicit Runge-Kutta methods for Volterra integro and integro-differential equations*, BIT 29, 258-269 (1989).
- [29] T.L. Cromer, *Asymptotically periodic solutions of Volterra integral equations in epidemic models*, J. Math. Anal. Appl. 110, 483–494 (1985).
- [30] E.Cuesta, Ch.Lubich, C.Palencia, *Convolution quadrature time discretization of fractional diffusion-wave equations*, submitted to Math. Comp.,2004.
- [31] F. de Hoog, R. Weiss, *Implicit Runge-Kutta methods for second kind Volterra integral equations*, Numer. Math. 23, 199-213 (1975)
- [32] Z. S. Deligonoul, S. Bilgen: *Solution of the Volterra equation of renewal theory with the Galerkin technique using cubic splines*, J. Statist. Comput. Simulation 20, 37–45 (1984).
- [33] A. Dixon, *A non linear singular Volterra integro–differential equation arising from a reaction–diffusion study of a small cell*, J. Comput. Appl. Math. (1987)

- [34] O. Diekmann, *Integral equations and population dynamics*, in: Colloquium Numerical Treatment of Integral Equations (H.J.J. te Riele, ed.), pp. 115–149, iMC Syllabus 41, Amsterdam, Mathematisch Centrum (1979).
- [35] O. Diekmann, *Thresholds and travelling waves for the geographical spread of infection*, J. Math. Bio., 6, 109–130 (1978).
- [36] O. Diekmann, *On a nonlinear integral equation arising in mathematical epidemiology*, Differential equations and applications, (1978).
- [37] O. Diekmann, H. G. Kaper, *On the bounded solutions of a nonlinear convolution equation*, Nonlinear Anal. 6, 721–737 (1978).
- [38] O. Diekmann, *Limiting behaviour in an epidemic model*, Nonlinear Anal. 5, 459–470 (1977).
- [39] G. Doetsch, *Guide to the applications of the Laplace Transform*, D. Van Nostrand Company, LTD (1961).
- [40] G. Doetsch, *Handbuch der Laplace transformation*, Verlag, Birkhauser Basel (1950).
- [41] C. L. Dolph, G. J. Minty, *On nonlinear integral equations of Hammerstein type*, in Anselone (1984), 99-154.
- [42] S. Elaydi, *Periodicity and stability of linear Volterra difference systems*, J. Math. Anal. Appl. 181, 483–492 (1994).
- [43] W. Feller, *On the integral equation of renewal theory*, Ann. Math. Statist. 12, 243–267 (1941).

- [44] N. B. Franco, *A Volterra integral equation arising from the propagation of nonlinear waves*, Rev. Mat. Estat. 17, 35–49 (1999).
- [45] F. G. Friedlander, *The reflexion of sound pulses by convex parabolic reflectors*, Proc. Cambr. Phil. Soc. 37, 134–149 (1941).
- [46] A. Ghizetti, A. Ossicini, *Trasformate di Laplace e calcolo simbolico*, UTET (1971).
- [47] B. H. Gilding, *A singular nonlinear Volterra integral equation*, J. Integral Equations Appl. 5, 465–502 (1993).
- [48] G. Gripenberg, *On some epidemic models*, Quart. Appl. Math. 39, 317–327 (1981).
- [49] C. W. Groetsch, *A simple numerical model for nonlinear warming of a slab*, J. Comput. Appl. Math. 38, 149–156 (1991).
- [50] C. W. Groetsch, *Numerical methods for a non-linear integral equation and an associated inverse problem*, in Integral Equations and Inverse Problems (Varna 1989) (V. Petkov and R. Lazarov, eds.), pp. 90–95, Pitman Res. Notes in Mathh. Ser. 235, Harlow, Longman (1989).
- [51] J. Guy, A. Salès, *Integral Equations in Everyday Practice*, Paris, Lavoisier-TEC&DOC (1991).
- [52] E. Hairer, Ch. Lubich, M. Schlichte, *Fast numerical solution of nonlinear Volterra convolution equations*, SIAM J. Sci. Statist. Comput. 6, 532–541 (1985).
- [53] P. Henrici, *Applied and computational complex analysis*, Jon Wiley & Sons (1977).

- [54] H. W. Hethcote, *Asymptotic behaviour in deterministic epidemic model*, Bull. Math. Biol. 35, 607–614 (1973).
- [55] H. W. Hethcote, D. W. Tudor, *Integral equation models for endemic infectious diseases*, J. Math. Biol. 9, 37–47 (1980).
- [56] R. Hiptmair, A. Schädle, *Non-reflecting boundary conditions for Maxwell's equations*, Computing 71(3), 265–292 (2003).
- [57] M. A. K. Ibrahim, M. H. Alnasr, *Product θ -methods with the regularization technique for the high thermal loss problem*, J. Egyptian Math. Soc. 5, 155–169 (1997).
- [58] J. Janno, L. v. Wolfersdorf, *Inverse problems for identification of memory kernels in viscoelasticity*, Math. Methods Appl. Sci. 20, 291–314 (1997).
- [59] J. Janno, L. v. Wolfersdorf, *Identification of weakly singular memory kernels in heat conduction*, Z. Angew. Math. Mech. 77, 243–257 (1997).
- [60] B. Jumarhon, S. McKee, *Product integration methods for solving a system of nonlinear Volterra integral equations*, J. Comput. Appl. Math. 69, 285–301 (1996).
- [61] B. Jumarhon, *The one-dimensional heat equation and its associated Volterra integral equations*, Ph.D. thesis, Dept. of Mathematics, University of Strathclyde, Glasgow (1994).
- [62] S. I. Kabanichin, A. Lorenzi, *Identification Problems of Wave Phenomena: Theory and Numerics*, Utrecht, VSP (1999).
- [63] S. Karlin, *On the renewal equation*, Pacific J. Math. 5, 229–257 (1955).

- [64] J. B. Keller, W. E. Olmstead, *Temperature of a nonlinearly radiating semi-infinite rod*, Quart. Appl. Math. 29, 559–556 (1972).
- [65] E. M. Kiss, *Ein Beitrag zur Regularisierung und Diskretisierung des inversen Problems der Identifikation des Memorykerns in der Viskoe-lastizität*, Dissertation, Fakultät Mathematik und Informatik, TU Bergakademie Freiberg (1999).
- [66] J. D. Lambert, *Numerical Methods for Ordinary Differential Systems: the initial value problem*, Wiley (1991).
- [67] N. Levinson, *A nonlinear Volterra equation arising in the theory of superfluidity*, J. Math. Anal. Appl. 1, 1–11 (1960).
- [68] Ch. Lubich, A. Schädle, *Fast convolution for non-reflecting boundary conditions*, Siam. J. Sci. Comput. 24, 161–182 (2002).
- [69] Ch. Lubich, *Convolution quadrature and discretized operational calculus II*, Numer. Math. 52, 413–425 (1988).
- [70] Ch. Lubich, *Runge-Kutta theory for Volterra and Abel Integral Equations of the second kind*, Math. Comp. 41, 87–102 (1983).
- [71] W. R. Mann, F. Wolf, *Heat transfer between solids and gases under nonlinear boundary conditions*, Quart. Appl. Math. 9, 163–184 (1951).
- [72] R. K. Miller, A. Unterreiter, *Switching behaviour of PN-diodes: Volterra integral equation models*, J. Integral Equations Appl. 4, 257–272 (1992).
- [73] R. K. Miller, *Nonlinear Volterra Integral Equations*, Benjamin, Menlo Park (Cal), (1971).

- [74] A.Murli, M.Rizzardi, *Algorithm 682: Talbot's method for the Laplace inversion problem*, ACM Trans. Math. Softw. 16,2
- [75] B. Noble, *The numerical solution of nonlinear integral equations and related topics*, in Anselone, 215–318 (1964).
- [76] J. Norbury, A. M. Stuart, *Volterra integral equations and a new Gronwall inequality. I. The linear case; II. The nonlinear case*, Proc. Roy. Soc. Edinburgh Sect. A 106, 361–373; 375–384 (1987).
- [77] W. E. Olmstead, *A nonlinear integral equation associated with gas absorption in a liquid*, Z. Angew. Math. Phys. 28, 513–523 (1977).
- [78] W.E. Olmstead, R. A. Handelsman, *Diffusion in a semi-infinite region with a nonlinear surface dissipation*, SIAM Rev. 2, pp. 275-291 (1976). pp. 275-291.
- [79] W. Okrasinski, *Nonlinear Volterra equations and physical applications*, Extracta Math. 4, 51–80 (1989).
- [80] W. Okrasinski, *On a non-linear convolution equation occurring in the theory of water percolation*, Ann. Polon. Math. 37, 223–229 (1978).
- [81] J. Prüss, *Evolutionary Integral Equations and Applications*, Basel and Boston, Birkhäuser-Verlag (1993).
- [82] M.Rizzardi, *A modification of Talbot's method for the simultaneous approximation of several values of the inverse Laplace Transform*, ACM Trans. Math. Software, 21(4), 347-371, (1995).
- [83] J. H. Roberts, W. R. Mann, *On a certain nonlinear integral equation of the Volterra type*, Pacific J. Math. 1, 431–445 (1951).

- [84] W. Schmeidler, *Integralgleichungen mit Anwendungen in Physik und technik*, Leipzig, Akad. Verlagsgesellschaft Geest und Portig (1950).
- [85] C. Schmeiseer, A. Unterreiter, R. Weiss, *The switching behaviour of a one-dimensional PN-diode in low injection*, Math. Models Methods Appl. Sci. 3, 125–144 (1993).
- [86] A. Talbot, *The accurate numerical inversion of Laplace Transforms*, J. Inst. Math. Appl., 23, 97–120 (1979).
- [87] H. R. Thieme, *Asymptotic estimates of the solutions of nonlinear integral equations and asymptotic speeds for the spread of populations*, J. Reine. Angew. Math. 306, 94–121 (1979).
- [88] H. R. Thieme, *A model for the spatial spread of an epidemic*, J. Math. Biol., 4, 337–351, (1977).
- [89] F. Unger, L. v. Wolfersdorf, *Inverse Probleme zur Identifikation von Memory-Kernen*, Freiburger Forschungsberichte Mathematik C458, Technische Universität Bergakademie Freiberg (1995).
- [90] A. Unterreiter, *Volterra integral equation models for semiconductor devices*, Math. Models Appl. Sci. 19, 425–450 (1996).
- [91] V. Volterra, *Sulla inversione degli integrali definiti*, Atti R. Accad. Sci. Torino 31, 311–323 (Nota I) (1896).
- [92] G. F. Webb, *Theory of Nonlinear Age-Dependent Population Dynamics*, New York, Marcel Dekker (1985).
- [93] L. v. Wolfersdorf, *On identification of memory kernels in linear theory of heat conduction*, Math. Models Methods Appl. Sci. 17, 919–932 (1994).

- [94] X. Q. Zhao, *Dynamical Systems in Population Biology*, CMS Books in Mathematics, New York, Springer–Verlag (2003).