WAVEFORM RELAXATION METHODS FOR VOLTERRA INTEGRO-DIFFERENTIAL EQUATIONS

WADE WILLIAM PARSONS

Waveform Relaxation Methods for Volterra Integro-differential Equations

by

©Wade William Parsons

A thesis submitted to the
School of Graduate Studies
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Department of Mathematics and Statistics
Memorial University of Newfoundland

February 1999

St. John's Newfoundland Canada

Abstract

This thesis is concerned with the solution of systems Volterra integro-differential equations by the application of waveform relaxation methods. This is a timely topic since such methods can often be implemented efficiently on parallel architectures. It derives convergence results for both the regular kernel and the weakly singular kernel cases, and although our primary concern is with numerical methods, we consider both analytic and numerical solutions.

In Chapter 1 we study the history of waveform relaxation methods and try to bring the reader up to date with what is presently known about these methods. We emphasize their application to the solution of systems of ordinary differential equations and Volterra integral equations. In each case, we consider both the continuous-time and discrete-time methods and give convergence results for each. Therefore, this chapter will set the stage for the application of waveform relaxation techniques to the solution of Volterra integro-differential equations in Chapter 2.

Chapter 2 is the main chapter of the thesis and contains all of the original results from my research. It begins by giving the standard resolvent representation of the analytic solution of Volterra integro-differential equations, with both regular kernels and weakly singular kernels. It then considers continuous-time iteration waveform relaxation methods, in which we assume that the resulting equations can be solved exactly. We prove that these methods converge uniformly on all bounded intervals.

However, the main body of results in this chapter, concern the collocation solution of the iterates that result when waveform relaxation methods are applied to Volterra integro-differential equations. We will consider convergence, both as the

steplength tends to zero and as the number of iterations tend to infinity. We study the effect various iterative methods used to solve the resulting implicit nonlinear algebraic equations have on the convergence and complete the discretization by taking into account the use of quadrature to solve the integrals in the method. Throughout the chapter we include numerical examples which illustrate the various theorems, with tables of results and discussion placed in a section at the end of this chapter.

In Chapter 3, we point out that a major source of applications of Volterra integro-differential equations are the Volterra partial integro-differential equations. We also mention topics not considered in the main body of the thesis. These include numerical stability of Volterra integro-differential equations, and the use of graded meshes for the solution of Volterra integro-differential equations, with weakly singular kernels.

Acknowledgments

I begin by acknowledging my thesis advisor, Dr. Hermann Brunner. Clearly, this work would not have been possible without his guidance and patience.

Dr. Charron was always willing to discuss mathematics, and helped uncover a number of misunderstandings. Dr. Narayanaswami and Dr. Shawyer answered my LaTeX questions and were a constant source of encouragement. I wish to thank the Department of Mathematics and Statistics for making me welcome, and Dr. Mimi-nis, from the Computer Science Department, for his assistance with ©MATLAB.

Also, my employer, College of the North Atlantic, permitted me to begin the "Deferred Salary Plan" one year earlier than scheduled.

Finally, my wife, Roseanne, deserves special recognition. She proofread the thesis and graciously shared me with this project, for a very long time.

Contents

List of Tables

Chapter 1

Waveform Relaxation Methods: History and the State of the Art

1.1 Introduction

Waveform Relaxation (WR) algorithms involve the partitioning of a system of equations. Each of the resulting subsystems are integrated independently over a number of iterative steps with information from each subsystem being passed on only at the end of each pass. This is an old idea, embodied in the use of Picard-Lindelöf iterations in the proof of various existence theorems in differential equations. However, recently these methods have become of practical interest because of their ability to decouple the large systems of equations that often arise, either naturally from practical problems, or because of the necessity of using implicit methods. In addition, the availability of new parallel architectures have propelled these investigations.

This idea was originally introduced by Lelarasmee (1982) and Lelarasmee *et al.* (1982) for the analysis of large scale dynamical systems. In particular, these authors were modelling metal oxide semi-conductor digital circuits. Since the modelling of very large scale integrated (VLSI) circuits involves the solution of a very large system

of equations, the motivation for these methods is clear. See White *et al.* (1985) for a survey of these applications and the book by White and Sangiovanni-Vincentelli (1987).

Subsequently, these methods were developed for ordinary differential equations (ODEs), with Miekkala and Nevanlinna (1987a, 1987b) and Nevanlinna (1989a, 1989b, 1990) putting the convergence theory on a sound mathematical basis. See also Skeel (1989), Lie and Skålin (1992), Lubich (1992), Bellen and Zennaro (1993), Bellen *et al.*, (1993) and Bellen *et al.* (1994). We also refer the reader to the book Burrage (1995).

The computational problems for ODEs are even worse for Volterra integral equations, since one has to continually recompute the *lag-term* arising from the integral part. Crisci *et al.* (1993) and D'Alterio and Vecchio (1995a, 1995b) studied Parallel Iterated VRK methods, Crisci *et al.* (1995) and Paone and Vecchio (1993, 1994) studied Block Volterra Linear methods and Vecchio (1996) studied Parallel Transformed methods. More recently, Crisci *et al.* (1996a) considered continuous-time waveform relaxation methods, where we assume that the VIE can be solved exactly. Then Crisci *et al.* (1996b and 1997a) introduced discrete-time waveform relaxation Volterra Runge-Kutta methods to solve the numerical problem. See also Brunner *et al.* (1998) for waveform relaxation methods for VIEs with weakly singular kernels. Finally, Crisci *et al.* (1997b) have been looking at time-point relaxation methods for Volterra integro-differential equations with regular kernels.

Of course, iterative techniques and *relaxation* methods for the solution of systems of algebraic equations have a much longer history. The Jacobi method, sometimes called the "method of simultaneous displacements" and the Gauss-Seidel method

were well known. An early attempt to increase the rate of convergence led a British engineer, Richard Southwell to develop a *method of relaxation* scheme, which became the precursor to the *overrelaxation* acceleration technique. See Allen (1954) for the history of these ideas, and see Axelsson (1994) for a modern introduction.

1.2 Summary of the Thesis

In Chapter 1 we give a brief history of waveform relaxation and its special case time-point relaxation (TR) methods for the solution of ordinary differential equations and Volterra integral and integro-differential equations (VIEs and VIDEs, respectively) up to the time of writing of this thesis. Chapter 2 forms the main body of the thesis and contains all of the original results of the thesis research. In fact, Chapter 1 will introduce and lay the ground work for Chapter 2. In Chapter 3, we point out that a major source of applications of Volterra integro-differential equations is the partial integro-differential equations. The semi-discretization of such problems leads to a large system of Volterra integro-differential equations and would benefit from a time-point relaxation approach. Finally we look back over the thesis, draw some conclusions and point out areas where the analysis can be taken further. For example, different windowing and different splitting functions could be utilized, or higher order inner iteration schemes could be applied to take advantage of high order outer iteration methods. We also address topics not considered in the main body of the thesis. These include numerical stability of Volterra integro-differential equations, and the use of graded meshes for the solution of Volterra integro-differential equations, with weakly singular kernels.

We begin by listing the original results in the order in which they will be pre-

sented. See also Tables 2.2, 2.3 and 2.4 for a listing of the most important results.

- **Theorem 2.7** The continuous-time iteration WR method (2.53) for the solution of the VIDE with regular kernel (??), is uniformly convergent on all bounded intervals $[0, T]$.

- **Theorem 2.10** The continuous-time iteration WR method (2.87) for the solution of the VIDE with weakly singular kernel (2.21), is uniformly convergent on all bounded intervals $[0, T]$.

- **Theorem 2.11** The discrete-time iteration TR method, using Gauss-Jacobi or Gauss-Seidel iterations, is commutative.

- **Theorem 2.12** The discrete-time iteration TR method (2.128), for the solution of the VIDE with regular kernel (2.106), using q Gauss-Jacobi or Gauss-Seidel iterations, is convergent for $q \to \infty$, for sufficiently small steplength.

- **Corollary 2.2** The discrete-time iteration TR method (2.128), for the solution of the VIDE with weakly singular kernel (2.107), using q Gauss-Jacobi or Gauss-Seidel iterations, is convergent for $q \to \infty$, for sufficiently small steplength.

- **Theorem 2.15** The discrete-time iteration TR method (2.128), for the solution of the VIDE with regular kernel (2.106), using q Gauss-Jacobi or Gauss-Seidel iterations and r-point collocation at the Gauss-points, has nodal order $\min\{2r, q+1\}$.

- **Corollary 2.3** The discrete-time iteration TR method (2.128), for the solution of the VIDE with weakly singular kernel (2.107), using q Gauss-Jacobi or

Gauss-Seidel iterations and r-point collocation, has order of global convergence $1-\alpha$, for $0 < \alpha < 1$.

- **Theorem 2.16** The discrete-time iteration TR method (2.146), for the solution of the VIDE with regular kernel (2.106), using q Gauss-Jacobi or Gauss-Seidel iterations, s Picard fixed-point iterations and r-point collocation at the Gauss-points, has nodal order $\min\{2r, q+1, q+s\}$.

- **Corollary 2.4** The discrete-time iteration TR method (2.146), for the solution of the VIDE with weakly singular kernel (2.107), using q Gauss-Jacobi or Gauss-Seidel iterations, s Picard fixed-point iterations and r-point collocation, has order of global convergence $1-\alpha$, for $0 < \alpha < 1$.

- **Theorem 2.18** The discrete-time iteration TR method (2.128), for the solution of the VIDE with regular kernel (2.106), using q Gauss-Jacobi or Gauss-Seidel iterations, s Newton iterations and r-point collocation at the Gauss-points, has nodal order $\min\{2r, q+1, 2^s q\}$.

- **Theorem 2.20** The discrete-time iteration TR method (2.128), for the solution of the VIDE with regular kernel (2.106), using q Gauss-Jacobi or Gauss-Seidel iterations, s Modified Newton iterations and r-point collocation at the Gauss-points, has nodal order $\min\{2r, q+1, (s+1)q\}$.

- **Theorem 2.21** The discrete-time iteration TR method (2.168), for the solution of the VIDE with regular kernel (2.106), using q Gauss-Jacobi or Gauss-Seidel iterations, s Picard fixed-point iterations, r-point collocation and r-point interpolatory quadrature at the Gauss-points, has nodal order $\min\{2r, q+$

$1, q+s\}$.

- **Theorem 2.22** The discrete-time iteration TR method (2.168), for the solution of the VIDE with regular kernel (2.106), using q Gauss-Jacobi or Gauss-Seidel iterations, s Newton iterations, r-point collocation and r-point interpolatory quadrature at the Gauss-points, has nodal order $\min\{2r, q+1, 2^s q\}$.

- **Theorem 2.23** The discrete-time iteration TR method (2.168), for the solution of the VIDE with regular kernel (2.106), using q Gauss-Jacobi or Gauss-Seidel iterations, s Modified Newton iterations, r-point collocation and r-point interpolatory quadrature at the Gauss-points, has nodal order $\min\{2r, q+1, (s+1)q\}$.

Continuous-Time Iteration WR Methods

The main results for this section are given by Theorems 2.7 and 2.10 which show convergence of the methods; see Illustration 2.5. However, from equations (2.58) and (2.59) for Theorem 2.7 and equation (2.90) for Theorem 2.10 it is clear that the error can be quite large on $[0, T]$, for larger values of T. Therefore one may wish to employ the idea of windows and subdivide the interval $[0, T]$ so as to promote relatively fast convergence on each of these subintervals; see Section 1.4.3.

Of course, our main concern in this thesis is the discrete-time TR iteration methods and there we employ windows of length equal to the steplength; see Section 2.3.

Comparing Theorems 2.7, 2.8, 2.9 and 2.10, and Corollary 2.1 we will see that continuous-time iteration WR methods for ODEs and VIDEs are uniformly convergent on all bounded intervals $[0, T]$, whereas VIEs are only conditionally convergent.

Therefore, VIDEs behave more like ODEs than VIEs.

Discrete-Time Iteration TR Methods

The remaining results concern discrete-time iteration TR methods which form the basis for the computational solution of VIDEs. In Section 2.3.1 we show that TR methods decouple an m-dimensional VIDE into a *system* of m one-dimensional VIDEs which are then easier to solve and (depending on the iteration mode chosen) easier to implement on parallel architectures.

Theorem 2.12 and Corollary 2.2 show that the discrete-time iteration TR methods converge as the number of Gauss-Jacobi or Gauss-Seidel iterations $q \to \infty$, but we are not claiming that they supply the solution of the problems. We must first assume an underlying numerical method, and we choose a collocation method, where we take Gauss points for the regular kernel case. Theorem 2.15 essentially tells us that if we take r (Gauss) collocation points, we will require $q = 2r - 1$ iterations to attain the optimal nodal order of $2r$. It is not hard to see that this theorem can be extended to methods using other collocation points.

Corollary 2.3 points to a limitation of these methods when applied to VIDEs with weakly singular kernels, since the order of global convergence is limited to $1 - \alpha$, $0 < \alpha < 1$. This is clearly a direct result of a limitation of the underlying numerical method; see Theorem 2.14. The good news is that applying a TR iteration method to such an equation, in itself, is not adversely affected by the weak singularity in the kernel.

In any realistic problem, Theorem 2.15 and Corollary 2.3 will not suffice. We must progress to Theorem 2.21 for the regular kernel VIDE and similarly to Corollary

2.4 for the weakly singular kernel VIDE (also see Remark, in Section 2.4). This is because we usually cannot solve the resulting implicit algebraic equations and integrals in closed form. Concerning the use of interpolatory quadrature, for the regular kernel case, we assumed that this quadrature always employed the Gauss points. However, other points could be used, although there will be an accompanying decrease in optimal nodal order. Of course one can always use the same set of points for the quadrature as for the collocation method, without any additional loss of nodal order. See Brunner and van der Houwen (1986). Recall from Section 2.4 that we are using natural discretizations.

Theorems 2.18 and 2.20 consider the case where we use a method of higher order than simple Picard fixed-point to solve the resulting algebraic equations, namely, Newton and modified Newton, respectively; Theorems 2.22 and 2.23 are the corresponding discretized versions. However, all these results are disappointing, since our nodal order is restricted to $\min\{2r, q+1\}$, where we have r-point Gauss collocation, r-point interpolatory Gauss quadrature and q Gauss-Jacobi or Gauss-Seidel inner iterations. Clearly we attain the same nodal order by using just *one* outer iteration and it does not matter what method, Picard, Newton or modified Newton we use. Therefore the increased cost incurred with the Newton or modified Newton is not justified.

We conclude by giving the numerical results from various tests. We tested the results of Theorem 2.16 with a linear test problem (Test Problem 2.1), the results of Theorems 2.21, 2.22 and 2.23 with a nonlinear test problem (Test Problem 2.2) and the results of Corollary 2.4 with a linear test problem with various weakly singular kernels (Test Problem 2.3).

Future Work/Open Problems

Finally in Chapter 3, we conclude with a list of topics for future work that include some open problems. For example, to promote faster convergence, different windowing and different splitting functions could be utilized, for both the continuous-time and discrete-time WR methods. And in the case of discrete-time TR methods, a higher order inner iteration scheme, like the Newton method, could be applied to take full advantage of the higher order outer iteration schemes, like the Newton and modified Newton methods. Also, a variety of collocation points and interpolatory quadrature formulae could be employed in the underlying numerical method. The two main open problems coming from the thesis are numerical stability of VIDEs, and the poor convergence of VIDEs with weakly singular kernels.

1.3 Ordinary Differential Systems

We are first concerned with the solution of the initial value problem (IVP)

$$y'(t) = f(t,y), \qquad y(0) = y_0, \qquad t \in [0,T] \tag{1.1}$$

where y is a vector on $\Re^m$, $f : [0,T] \times \Re^m \to \Re^m$, $0 < T < \infty$, and we assume that the function f is continuous and satisfies a Lipschitz condition in y. The idea behind waveform relaxation methods is already contained in the historical approach taken by Picard. That is, a sequence of solutions $y^1(t), y^2(t), \ldots, y^q(t)$, is constructed, where the initial solution $y^0(t)$ is assumed given, and it is hoped that the sequence converges to the solution $y(t)$. This means that we solve a sequence of differential equations

$$\frac{d}{dt}y^q(t) \;=\; f(t,y^{q-1}), \quad t \in [0,T], \tag{1.2}$$

$$y^q(0) = y_0,$$

or equivalently,

$$y^q(t) = y_0 + \int_0^t f(s, y^{q-1}(s))ds,$$

$q = 1, 2, \ldots$. Clearly, we have decoupled the m-dimensional system into m independent quadrature problems. As pointed out by Burrage (1995), this approach is "embarrassingly parallel", with *updating* being the only interaction necessary between subsystems. Unfortunately, the convergence of this method is very slow, and we must search for more efficient algorithms. Thus we consider other iteration schemes which give more efficient methods.

We choose a function $G_f : [0, T] \times \Re^m \times \Re^m \to \Re^m$, such that

$$G_f(t, u, u) = f(t, u),$$

for all $t \in [0, T]$ and for all $u \in \Re^m$. Assume that the initial solution $y^0(t)$ is given and compute a sequence $y^1(t), y^2(t), \ldots, y^q(t)$, of solutions of the equations

$$\begin{aligned} \frac{d}{dt}y^q(t) &= G_f(t, y^{q-1}, y^q), \quad t \in [0, T], \\ y^q(0) &= y_0, \end{aligned} \tag{1.3}$$

or,

$$y^q(t) = y_0 + \int_0^t G_f(s, y^{q-1}(s), y^q(s))ds,$$

$q = 1, 2, \ldots$, which converges to the solution $y(t)$ of (1.1) as $q \to \infty$.

These solutions are called *waveforms* and the function G_f is called a *splitting* function, since it defines how the problem is to be split up into subsystems. The following iteration schemes are possible:

- **Picard Iteration**

$$G_f(t,u,v) = f(t,u).$$

- **Gauss-Jacobi Iteration**

$$^iG_f(t,u,v) = f_i(t,u_1,\ldots,u_{i-1},v_i,u_{i+1},\ldots,u_m),$$

for $i = 1,2,\ldots,m$.

- **Gauss-Seidel Iteration**

$$^iG_f(t,u,v) = f_i(t,v_1,\ldots,v_{i-1},v_i,u_{i+1},\ldots,u_m),$$

for $i = 1,2,\ldots,m$.

- **Newton Iteration**

$$G_f(t,u,v) = f(t,u) + \frac{\partial f}{\partial y}(t,y)|_{y=u}[v-u].$$

The methods employing Picard and Gauss-Jacobi iteration are fully parallel, but have the disadvantage that large quantities of past data must be stored. This can be a computational difficulty if the dimension of the system m is large. On the other hand, methods using Gauss-Seidel and Newton iteration schemes avoid this problem, but do not lend themselves to efficient implementation on parallel architectures. Concerning speed of convergence, we suggest that the reader consult Burrage (1995), Chapters 7 and 8.

1.3.1 Continuous-Time Iteration WR Methods

These methods given by (1.3) are called *continuous-time iteration* WR methods, since they do not involve *yet* the discretization of the time interval and the application of numerical methods to solve the resulting differential equations. There is

much that can be said about the convergence rates of each of these iteration schemes, especially, for particular problems (*e.g.* linear), and we refer the reader to Bellen and Zennaro (1991, 1993), Burrage (1995), Chapter 7, 't Hout (1995) and Burrage *et al.* (1996).

For example, assume the function f in (1.1) is continuously differentiable and satisfies a Lipschitz condition, in some norm $\|\cdot\|$, with constant L, i.e.

$$\|f(t,y_1) - f(t,y_2)\| \leq L\|y_1 - y_2\|,$$

for all $y_1, y_2 \in \Re^m$ and all $t \in [0,T]$. In addition assume that the splitting function $G_f(t,u,v)$ is continuously differentiable and satisfies the following Lipschiz condition in u and v, with Lipschitz constants L_1, L_2, respectively,

$$\left.\begin{array}{rcl} \|G_f(t,u_1,v) - G_f(t,u_2,v)\| & \leq & L_1\|u_1 - u_2\|, \\ \|G_f(t,u,v_1) - G_f(t,u,v_2)\| & \leq & L_2\|v_1 - v_2\|, \end{array}\right\} \tag{1.4}$$

for all u_1, u_2, v_1, v_2 in $\Re^m$ and t in $[0,T]$. Then the resulting WR method (1.3) converges uniformly in $[0,T]$, where

$$\|y^q - y\|_T \leq \exp(L_2 T)\frac{(L_1 T)^q}{q!}\|y^0 - y\|_T, \tag{1.5}$$

$q = 1, 2, \ldots$. Note that we are referring to the usual maximum norm from the Banach space of continuous vector valued functions, defined on [0,T],

$$\|y\|_T := \max_{t\in[0,T]}\|y(t)\|,$$

inherited from any vector norm of $\Re^m$. See Bellen and Zennaro (1991) and Burrage (1995), Theorem 7.9.2

One can also give convergence results, where the two-sided Lipschitz constant L_2 of the splitting function G_f with respect to the second argument is replaced by a one-sided Lipschitz condition, see Burrage (1995), Theorem 7.9.3.

1.3.2 Time Discretization

Since (1.3) can seldom be solved in closed form we must discretize the time interval $[0, T]$ and apply a numerical method to approximate the solution. Numerical methods generally fall into the two categories of linear multistep (LMS) and one-step methods (typically Runge-Kutta). Since this thesis concerns collocation methods, which are continuous implicit Runge-Kutta (IRK) methods, we give a few standard results. See Hairer *et al.* (1993) and Lambert (1991).

Consider a partition $\Pi_N : 0 = t_0 < t_1 < \cdots < t_N = T$, $t_{n+1} - t_n =: h$, for $n = 0, 1, \ldots, N-1$, where the points $\{t_n\}_{n=0}^N$ are called the *mesh points* or *nodes*. The approximation y_{n+1} to the exact solution $y(t_{n+1})$ of (1.1) is given by an r-stage Runge-Kutta method,

$$\begin{aligned} y_{n+1} &= y_n + h\sum_{i=1}^{r} b_i k_i \\ k_i &= f\left(t_n + c_i h, y_n + h\sum_{j=1}^{r} a_{ij}k_j\right), \quad i = 1, 2, \ldots, r, \end{aligned} \tag{1.6}$$

$n = 0, 1, \ldots, N-1$. Assume that the *row-sum* condition always holds:

$$c_i = \sum_{j=1}^{r} a_{ij}, \qquad i = 1, 2, \ldots, r.$$

The coefficients occurring in the method are often displayed in a *Butcher array*:

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \cdots & a_{1r} \\ c_2 & a_{21} & a_{22} & \cdots & a_{2r} \\ \vdots & \vdots & & & \vdots \\ c_r & a_{r1} & a_{r2} & \cdots & a_{rr} \\ \hline & b_1 & b_2 & \cdots & b_r \end{array} \qquad \text{or simply} \qquad \begin{array}{c|c} c & A \\ \hline & b^T \end{array}$$

upon defining the r-dimensional vectors c and b and the $r \times r$ matrix A:

$$c = [c_1, c_2, \ldots, c_r]^T, \quad b = [b_1, b_2, \ldots, b_r]^T, \quad A = [a_{ij}].$$

- We say that an RK method is explicit if:

$$a_{ij} = 0, \text{ if } j \geq i,\ j = 1, 2, \ldots, r \Leftrightarrow \text{A strictly lower triangular.}$$

- We say that an RK method is semi-implicit (or diagonally implicit) if:

$$a_{ij} = 0, \text{ if } j > i,\ j = 1, 2, \ldots, r \Leftrightarrow \text{A lower triangular.}$$

- We say that an RK method is implicit if:

$$a_{ij} \neq 0, \text{ for some } j > i, \Leftrightarrow \text{A not lower triangular.}$$

Runge-Kutta methods are *discrete*, in that they yield approximations only at the mesh points. However, if interpolation is added, can they be defined throughout the interval $[0, T]$, in which case we call them *continuous* methods. Collocation methods, on the other hand, are defined over the interval $[0, T]$ and therefore have a *natural* interpolation built into their definition. They are therefore examples of continuous IRK methods.

Collocation methods involve choosing a unique function (usually a piecewise polynomial) by the condition that it satisfies the given problem, (1.1), at a given set of points called the *collocation points*.

Note: These piecewise polynomials are called *polynomial splines* and a collocation method utilizing them is called a *polynomial spline collocation* method. Since

such methods are prevalent and since we exclusively deal with this type of collocation method in this thesis, we will refer to them simply as collocation methods. ◇

Recall the partition $\Pi_N : 0 = t_0 < t_1 < \cdots < t_N = T$, $t_{n+1} - t_n =: h$, for $n = 0, 1, \ldots, N-1$, and let $Z_N := \{t_n : n = 1, \ldots, N-1\}$ be the set of interior points. Also let $\bar{Z}_N := Z_N \cup T$, and define the intervals $\sigma_0 := [0, t_1]$ and $\sigma_n := (t_n, t_{n+1}]$, $n = 1, 2, \ldots, N-1$.

Given integers r and d satisfying $-1 \leq d \leq r-1$, and letting π_r denote the space of real polynomials of degree not exceeding r, we define the finite-dimensional space $S_r^{(d)}(Z_N)$ of (real) *polynomial splines of degree* r and *continuity class* d, with the *knots* Z_N:

$$S_r^{(d)}(Z_N) := \{\eta \in C^d([0,T]) : \eta|_{\sigma_n} := \eta_n \in \pi_r, n = 0, 1, \ldots, N-1\}. \tag{1.7}$$

Note that $S_r^{(d)}(Z_N)$ is a finite dimensional subspace of $C^d([0,T])$, and

$$dim\ S_r^{(d)}(Z_N) = N(r-d) + d + 1, \qquad -1 \leq d \leq r-1. \tag{1.8}$$

The two most useful cases are $d = -1$, in which the functions may have jump discontinuities at the knots Z_N, and $d = 0$, in which we have continuity over all of [0,T]. The latter case is most useful for ODEs (although $d = 1$ is also used), and the former for VIEs. See Brunner and van der Houwen (1986) and Brunner (1998).

Associated with this partition Π_N we consider the collocation points $T_N := \{t_n + c_j h; j = 1, 2, \ldots, r;\ n = 0, 1, \ldots, N-1\}$, where $0 \leq c_1 < c_2 < \cdots < c_r \leq 1$, and choose a continuous piecewise polynomial η of degree at most r which satisfies

(1.1) on the set T_N. More precisely, we wish to determine $\eta \in S_r^{(0)}(Z_N)$, so that

$$\eta'(t) = f(t, \eta), \qquad \eta(0) = y_0, \qquad \text{for} \quad t \in T_N. \tag{1.9}$$

Since this equation defines the collocation method, it is called the *collocation* equation. To show the connection between collocation methods and continuous IKR methods choose $\eta \in S_r^{(0)}(Z_N)$. Then $\eta(t_n + \tau h)$ represents the approximation to the exact solution $y(t_n + \tau h)$ of (1.1) given by this collocation method, where $\tau \in [0, 1]$.

Then the collocation equation becomes

$$\eta'(t_n + c_j h) = f(t_n + c_j h, \eta(t_n + c_j h)), \qquad \eta(0) = y_0, \tag{1.10}$$

$j = 1, 2, \ldots, r,\ n = 0, 1, \ldots, N-1.$

Note that there are Nr collocation conditions plus the given initial condition $\eta(0) = y_0$; this number equals the dimension of the space $S_r^{(0)}(Z_N)$; see (1.8). Therefore, since $\eta'(t_n + \tau h) \in \pi_{r-1}$, for $n = 0, 1, \ldots, N-1$, and $\tau \in (0, 1]$, we can use Lagrange polynomial interpolation to get

$$\eta'(t_n + \tau h) = \sum_{j=1}^{r} \eta'(t_n + c_j h) L_j(\tau),$$

where $L_j(\tau)$ are the fundamental Lagrange polynomials:

$$L_j(\tau) = \prod_{\substack{k=1 \\ k \neq j}}^{r} \frac{\tau - c_k}{c_j - c_k}. \tag{1.11}$$

Then

$$\eta(t_n + \tau h) = \eta(t_n) + h \int_0^{\tau} \eta'(t_n + vh) dv, \quad \tau \in [0, 1],$$

which becomes

$$\eta(t_n + \tau h) = \eta(t_n) + h \int_0^\tau \left(\sum_{j=1}^{r} \eta'(t_n + c_j h) L_j(v) \right) dv,$$

or,

$$\eta(t_n + \tau h) = \eta(t_n) + h \sum_{j=1}^{r} \alpha_j(\tau) Y_{n,j}, \tag{1.12}$$

where

$$\alpha_j(\tau) = \int_0^\tau L_j(v) dv,$$

and

$$Y_{n,j} = \eta'(t_n + c_j h) = f(t_n + c_j h, \eta(t_n + c_j h)), \tag{1.13}$$

$j = 1, 2, \ldots, r,\ n = 0, 1, \ldots, N-1.$

Using (1.12) we can rewrite (1.13) to get the following form of the collocation method for the solution of (1.1):

$$\begin{aligned} \eta(t_n + \tau h) &= \eta(t_n) + h \sum_{j=1}^{r} \alpha_j(\tau) Y_{n,j}, \quad \tau \in [0,1], \\ Y_{n,j} &= f\left(t_n + c_j h, \eta(t_n) + h \sum_{k=1}^{r} \alpha_k(c_j) Y_{n,k} \right), \quad j = 1, 2, \ldots, r, \end{aligned} \tag{1.14}$$

where $n = 0, 1, \ldots, N-1$.

It is clear that for $h > 0$ sufficiently small, the resulting collocation method (1.14) is defined throughout the interval $[0, T]$ and is therefore a *continuous* numerical method. Setting $\tau = 1$, (1.14) becomes

$$\begin{aligned} \eta(t_{n+1}) &= \eta(t_n) + h \sum_{j=1}^{r} \alpha_j(1) Y_{n,j}, \quad \tau \in [0,1], \\ Y_{n,j} &= f\left(t_n + c_j h, \eta(t_n) + h \sum_{k=1}^{r} \alpha_k(c_j) Y_{n,k} \right), j = 1, 2, \ldots, r, \end{aligned} \tag{1.15}$$

where $n = 0, 1, \ldots, N-1$. It is easy to see that (1.15) is equivalent to the r-stage implicit Runge-Kutta method (1.6) with coefficients given by:

$$a_{jk} = \alpha_k(c_j) = \int_0^{c_j} L_k(\tau)d\tau \quad \text{and} \quad b_j = \alpha_j(1) = \int_0^1 L_j(\tau)d\tau, \quad j,k = 1,2,\ldots,r.$$

For easy reference, we put this result in the following theorem. See also Hairer *et al.* (1993).

Theorem 1.1 *(Guillou and Soulé 1969, Wright 1970) Consider the initial value problem given by (1.1) and suppose it is solved by collocation based on the collocation points $0 \leq c_1 < c_2 < \cdots < c_r \leq 1$ in the polynomial spline space $S_r^{(0)}(Z_N)$, leading to the method given by (1.15). Then this method coincides on the nodes $\bar{Z}_N$ with the implicit Runge-Kutta method given by (1.6).*

In order to quantify the propagation of errors in a numerical method, we consider the concept of order.

Definition 1.1 *The Runge-Kutta method (1.6) has order p, if for all sufficiently smooth problems (1.1),*

$$\|y(h) - y_1\| \leq Ch^{p+1},$$

for some constant $0 < C < \infty$, where y is the exact solution and y_1 is the numerical solution generated at $t_1 = h$.

Note that this is equivalent to requiring that the Taylor series for the exact solution $y(h)$ and the Taylor series for the numerical solution y_1, expanded about y_0, are identical up to and including the h^p term. See Lambert (1991), Section 5.2, where he discusses the concept of the local truncation error. We call $e(h) := y(h) - y_1$

the *local error* at h, since it involves the *local* error committed at the first step in the iteration, see Hairer et al (1993). This definition can be extended to any step, if we impose the *localizing assumption*, which assumes that all "back values" are exact; see Lambert (1991).

Recall the asymptotic symbol $\mathcal{O}$.

Definition 1.2 *For functions $f(x)$ and $g(x)$*

$$f(x) = \mathcal{O}[g(x)], \quad as\ x \to a,$$

where a is a constant, if for some constant C

$$\frac{|f(x)|}{|g(x)|} \leq C,$$

as $x \to a$.

See Kahn (1990).

Using this notation, we now say that a method of order p, has a local error $\mathcal{O}(h^{p+1})$, where it is clear that this involves the limit as $h \to 0$. If no localizing assumption is made,

$$E_{n+1} := y(t_{n+1}) - y_{n+1}, \quad n = 1, 2, \ldots, N-1,$$

is the *global error* at t_{n+1} and includes the accumulation of errors after several steps. This always leads to a loss of one power of h in the order. If the local error is $\mathcal{O}(h^{p+1})$, then the global error will be $\mathcal{O}(h^p)$. See Lambert (1991), Section 3.5 for a nice discussion of these ideas through specific examples, or Hairer et al (1993), Section II.3 for a formal proof.

Definition 1.3 *If a Runge-Kutta method (1.6) defined by* $\begin{array}{c|c} c & A \\ \hline & b^T \end{array}$, *satisfies the condition*

$$\sum_{j=1}^{r} a_{ij} c_j^{\sigma-1} = c_i^{\sigma}/\sigma, \qquad \sigma = 1, 2, \ldots, r, \qquad i = 1, 2, \ldots, r,$$

then it is said to satisfy one of Butcher's simplifying assumptions, namely, $C(r)$.

We now state the following characterization of collocation methods; see Hairer *et al.* (1993).

Theorem 1.2 *(Hairer et al. 1993) A continuous implicit Runge-Kutta method with distinct* c_i *and order at least* r *is a collocation method iff* $C(r)$ *is true.*

Since a continuous method is defined throughout the interval $[0, T]$, we can consider the question of order at *any* or *all* point(s) in this interval. Therefore for collocation, we can talk about the order of global convergence throughout the interval and the order of local convergence at specific points; for example, at the nodes. Care should be taken not to confuse these two notions with the terms local error and global error, defined previously. Because of their importance in the thesis, we itemize these two definitions.

- If y is the exact solution and η is the numerical (collocation) solution (1.14), to the problem (1.1), then if

  $$\|e(t)\| := \|y(t) - \eta(t)\| = \mathcal{O}(h^p), \qquad t \in \bar{Z}_N,$$

  the numerical method has *nodal order* p.

- If y is the exact solution and η is the numerical (collocation) solution (1.14), to the problem (1.1), then if

$$\|e\|_T := \max_{t\in[0,T]}\{\|y(t) - \eta(t)\|\} = \mathcal{O}(h^p),$$

the numerical method has *order of global convergence* p (throughout $[0, T]$).

Theorem 1.3 *(Hairer et al. 1993) The r-point collocation method, given by (1.14), based on the space $S_r^{(0)}(Z_N)$ in general has order of global convergence r throughout $[0, T]$, when applied to the solution of (1.1), if f is r times continuously differentiable. However, if the $\{c_i\}$ satisfy the condition*

$$\int_0^1 \prod_{i=1}^{r}(s - c_i)ds = 0,$$

(i.e. if the degree of precision of the quadrature formula is $\geq r$) then the order is $r + 1$.

However, the nodal order may be greater than r; this is called local *superconvergence*. For example Gauss, Radau IIA, and Lobatto IIIA are collocation methods with nodal orders $2r$, $2r - 1$, and $2r - 2$, respectively.

See Lambert (1991), pp. 194-196, and Hairer *et al.* (1993), pp. 211-214, for more related details. Brunner (1998) gives a general survey of the application of collocation to the numerical solution of differential and Volterra integral equations.

1.3.3 Discrete-Time Iteration WR Methods

The name *discrete-time iteration* WR methods refers to waveform relaxation methods that involve discretization of the time interval and application of numerical

methods to solve the resulting differential equation (1.3). For example see Burrage (1995), Chapter 8, for the case where the underlying numerical method is linear multistep.

Consider a partition $\Pi_N : 0 = t_0 < t_1 < \cdots < t_N = T$, $t_{n+1} - t_n =: h$, for $n = 0, 1, \ldots, N-1$, and divide the integration interval $[0, T]$ into κ subintervals, called *windows*, each of length $\mathcal{N}h$, i.e.

$$[0, T] = \bigcup_{s=0}^{\kappa-1} [sh\mathcal{N}, (s+1)h\mathcal{N}],$$

where $\mathcal{N} = \frac{T}{\kappa h}$. Then the discrete-time WR method is applied, in turn, to each window.

Definition 1.4 *In a discrete-time WR method, if $\mathcal{N} = 1$, that is, the window length coincides with the step length, h, then the method is called a time-point relaxation (TR) method.*

Following Bellen *et al.* (1993), we study discrete-time iteration TR methods involving the Gauss-Jacobi and Gauss-Seidel iteration schemes and consider in this paper the use of collocation as the underlying numerical method. Bellen *et al.* (1993) employed continuous Runge Kutta (CRK) methods with interpolation given by *natural continuous extensions* (NCEs). As we stated in Section 1.3.2, collocation methods have a natural interpolation built into their definition.

Let $z(t)$ be any component $y_i(t)$ of the exact solution $y(t)$, of the problem above given by (1.1), $i = 1, 2, \ldots m$. In addition, we define

$$u(t) := [y_1(t), y_2(t), \ldots, y_{i-1}(t)], \qquad v(t) := [y_{i+1}(t), y_{i+2}(t), \ldots, y_m(t)].$$

The problem can then be written component-wise as:

$$z'(t) = f(t, u, z, v), \qquad z(0) = (y_0)_i,$$

where we have suppressed the subscript i, by letting it be understood that the i^{th}-component of f is used. We assume that the (continuous) approximation $\eta(t)$ to the solution $y(t)$ has already been computed for $t \in [0, t_n]$. To find the approximation $Z_n(t)$ of the corresponding i^{th}-component of $\eta(t)$ generated by the numerical method for $t \in [t_n, t_{n+1}]$ we consider first, the continuous-time iterations

$$\begin{aligned} \frac{d}{dt} z^q(t) &= f(t, u^{q-1}, z^q, v^{q-1}) \\ z^q(t_n) &= \eta_i(t_n), \end{aligned}$$

$q = 1, 2, \ldots$, $i = 1, 2, \ldots, m$. This requires that we define the initial guess for these components when $q = 0$, and although it is arbitrary, Bellen *et al.* (1993) take it to be the components of the constant function $\eta(t_n)$. That is, for all $t \in [t_n, t_{n+1}], n = 0, 1, \ldots, N-1$

$$\begin{aligned} u^0(t) &:= [y_1^0(t), y_2^0(t), \ldots, y_{i-1}^0(t)] = [\eta_1(t_n), \eta_2(t_n), \ldots, \eta_{i-1}(t_n)], \\ v^0(t) &:= [y_{i+1}^0(t), y_{i+2}^0(t), \ldots, y_m^0(t)] = [\eta_{i+1}(t_n), \eta_{i+2}(t_n), \ldots, \eta_m(t_n)], \end{aligned}$$

where $\eta(0) = y_0$. We will take a slightly different approach in this thesis and therefore get a better nodal order; see Remark, in Section 2.3.7.

Collocation is now applied to generate approximations for each iteration and for each of the components of y. This is continued for either a fixed number of iterations or until some measure of the difference between two successive approximations is less than a given tolerance.

We now look more closely at the form the collocation will take. Given the set of collocation points $T_N = \{t_n + c_j h; j = 1, 2, \ldots, r;\ n = 0, 1, \ldots, N-1\}$, where $0 \le c_1 < c_2 < \ldots < c_r \le 1$ the method becomes:

$$\begin{aligned} Z_n^q(t_n + \tau h) &= Z_n(t_n) + h \sum_{j=1}^{r} \alpha_j(\tau) Y_{n,j}^q, \quad \tau \in [0,1] \\ Y_{n,j}^q &= f\Big(t_n + c_j h, u_n^{q-1}(t_n + c_j h), Z_n(t_n) \\ &\quad + h \sum_{k=1}^{r} \alpha_k(c_j) Y_{n,k}^q, v_n^{q-1}(t_n + c_j h)\Big), \quad j = 1, 2, \ldots, r. \end{aligned} \tag{1.16}$$

As explained earlier, we have suppressed the subscript i, by letting it be understood that the i^{th}-component of both the function f and the $Y_{n,j}^q$ is used. Also, we have extended the definition of $u(t)$ and $v(t)$ to $u_n(t)$ and $v_n(t)$, respectively.

The iterations in the above method are of Gauss-Jacobi type; see Section 1.3. Note that we compute all the components of $\eta(t)$ for $t \in [t_n, t_{n+1}]$ before we "upgrade" to the new values. That is, we do not use the "new" $\eta_i(t)$ to compute the remaining components, even though we have calculated it. In the Gauss-Seidel iterations, we use the new components of $\eta(t)$ as soon as they are computed. As before, we first consider the continuous-time iterations

$$\begin{aligned} \frac{d}{dt} z^q(t) &= f(t, u^q, z^q, v^{q-1}) \\ z^q(t_n) &= \eta_i(t_n), \end{aligned}$$

$q = 1, 2, \ldots$, $i = 1, 2, \ldots, m$, and for $t \in [t_n, t_{n+1}], n = 0, 1, \ldots, N-1$, and let

$$\begin{aligned} u^0(t) &:= [y_1^0(t), y_2^0(t), \ldots, y_{i-1}^0(t)] = [\eta_1(t_n), \eta_2(t_n), \ldots, \eta_{i-1}(t_n)], \\ v^0(t) &:= [y_{i+1}^0(t), y_{i+2}^0(t), \ldots, y_m^0(t)] = [\eta_{i+1}(t_n), \eta_{i+2}(t_n), \ldots, \eta_m(t_n)], \end{aligned}$$

where $\eta(0) = y_0$. Again, collocation is now applied,

$$
\begin{aligned}
Z_n^q(t_n + \tau h) &= Z_n(t_n) + h \sum_{j=1}^{r} \alpha_j(\tau) Y_{n,j}^q, \quad \tau \in [0,1] \qquad (1.17)\\
Y_{n,j}^q &= f\left(t_n + c_j h, u_n^q(t_n + c_j h), Z_n(t_n)\right.\\
&+ \left. h \sum_{k=1}^{r} \alpha_k(c_j) Y_{n,k}^q, v_n^{q-1}(t_n + c_j h)\right), \quad j = 1, 2, \ldots, r.
\end{aligned}
$$

We refer the reader to the discussion at the end of Section 1.3 of the relative merits of these iteration schemes. In any case, a standard contraction principle argument shows that for sufficiently small h, these methods (1.16) and (1.17) are well defined and converge to the same limit as $q \to \infty$. As far as order conditions are concerned, Bellen *et al.* (1993) give the following theorem, which I have adapted for collocation methods.

Theorem 1.4 *(Bellen et al. 1993) Assuming the function f in (1.1) is sufficiently smooth and the underlying collocation method has order p^*, then the order of methods (1.16) and (1.17) is $p = min\{p^*, q\}$, for q iterations.*

Stopping Error

Since all collocation methods are *implicit* Runge-Kutta methods, they involve the solution of systems of implicit nonlinear algebraic equations. This clearly involves some iterative method, which can be a Picard fixed-point iteration or some variant of the Newton method. Picard iteration increases the order of accuracy by one at each iteration, Burrage (1995). This slow rate of convergence can be often troublesome, so we are led to consider Newton-type iterations, and in particular, the "stopping" error involved. Sugiura and Torii (1991) derived estimates of the Newton stopping error by using results originally due to Kantorovich and Akilov (1982). Later Spijker

(1994) considered the effect of these errors on the order of the underlying Runge-Kutta methods.

In Section 2.3.9, we consider the case where Newton's method is used to solve the system of implicit algebraic equations appearing in (1.16) and (1.17). We refer the reader to Theorem 2.18.

Evaluation of the Jacobian in Newton's method can be very expensive, especially if it must be continually updated. Therefore, one often uses the *modified* Newton method, in which the Jacobian is evaluated once, and this value is used in all subsequent calculations, Lambert (1991). We therefore get new stopping errors, and in general, a reduction in order. We refer the reader to Section 2.3.10 and Theorem 2.20.

However, as stated in Section 2.3.10, the use of high order methods to solve the implicit algebraic equations is not justified. See the discussion in that section and Table 2.1.

1.4 Volterra Integral Systems

After considering ODEs, it is quite natural to try to extend these methods to Volterra integral equations and integro-differential equations. In the papers by Crisci, *et al.* (1996a,1996b,1997a), the authors apply WR techniques to VIEs: they are concerned with the solution of the second kind VIE

$$y(t) = f(t) + \int_0^t k(t,s,y(s))ds, \qquad t \in [0,T], \tag{1.18}$$

where y is a vector on $\Re^m$, $f : [0,T] \to \Re^m$, $k : S \times \Re^m \to \Re^m$, where $0 < T < \infty$ and $S = \{(t,s) : 0 \leq s \leq t \leq T\}$. For its theory, see Miller (1971) and Gripenberg

et al. (1990).

In this thesis we will study Volterra *integro-differential* systems (VIDEs) which involve, in a very natural way, elements from both ODEs and VIEs. The non-linear Volterra integro-differential equation of interest is given by:

$$y'(t) = f(t,y) + \int_0^t k(t,s,y(s))ds, \quad y(0) = y_0, \quad t \in [0,T], \tag{1.19}$$

where y is a vector on $\Re^m$, $f : [0,T] \times \Re^m \to \Re^m$, $k : S \times \Re^m \to \Re^m$, where $0 < T < \infty$ and $S = \{(t,s) : 0 \leq s \leq t \leq T\}$. We assume that the function f is defined and continuous on the bounded interval $[0,T]$, the kernel k is defined and continuous on the triangle S and both f and k satisfy Lipschitz conditions in y.

Then, (1.19) possesses a unique solution $y \in C^1([0,T])$. Recently, Crisci *et al.* (1997b) have been looking at WR methods applied to VIDEs, with regular kernels.

A Volterra integro-differential equation can be considered a differential equation with a "memory" term, or lag term, that is represented by the integral. Therefore, unlike a differential equation, which depends only on *local* information, a VIDE depends on *global* information. That is, it considers its whole "past history", as does a VIE.

Again, the simplest iteration scheme is Picard or fixed-point iteration. That is, we construct a sequence of solutions $y^1(t), y^2(t), \ldots, y^q(t)$, by solving the equations:

$$y^q(t) = f(t) + \int_0^t k(t,s,y^{q-1}(s))ds, \quad t \in [0,T], \tag{1.20}$$

$q = 1, 2, \ldots$, and we take $y^0(t) = f(t)$, which converges to the solution $y(t)$ of (1.18), as $q \to \infty$.

In the case of the VIDE (1.19) we have

$$\begin{aligned} \frac{d}{dt}y^q(t) &= f(t, y^{q-1}) + \int_0^t k(t, s, y^{q-1}(s))ds, \quad t \in [0, T] \\ y^q(0) &= y_0, \end{aligned} \tag{1.21}$$

$q = 1, 2, \ldots$ and $y^0(t)$ is arbitrary, with $y^q(t)$ converging to the solution $y(t)$ of (1.19), as $q \to \infty$.

Unfortunately, as with ODEs (see Section 1.2) the convergence of this method is very slow and we must search for more efficient algorithms. Thus, we choose a splitting function $G_k : S \times \Re^m \times \Re^m \to \Re^m$, for the kernel $k(t, s, y)$ such that

$$G_k(t, s, u, u) = k(t, s, u),$$

for all $(t, s) \in S$ and for all $u \in \Re^m$. We consider the Volterra integral equation, (1.18).

Compute a sequence $y^1(t), y^2(t), \ldots, y^q(t)$, of solutions of the equations

$$\begin{aligned} y^q(t) &= f(t) + \int_0^t G_k(t, s, y^{q-1}(s), y^q(s))ds, \quad t \in [0, T] \\ y^0(t) &= f(t), \end{aligned} \tag{1.22}$$

$q = 1, 2, \ldots$, which converges to the solution $y(t)$ of (1.18) as $q \to \infty$. As before, we refer to these solutions as *waveforms*. Again, the following iteration schemes are popular (see Section 1.3).

- **Picard Iteration**

$$G_k(t, s, u, v) = k(t, s, u).$$

- **Gauss-Jacobi Iteration**

$${}^iG_k(t, s, u, v) = k_i(t, s, u_1, \ldots, u_{i-1}, v_i, u_{i+1}, \ldots, u_m),$$

for $i = 1, 2, \ldots, m$.

- **Gauss-Seidel Iteration**

$$^{i}G_k(t, s, u, v) = k_i(t, s, v_1, \ldots, v_{i-1}, v_i, u_{i+1}, \ldots, u_m),$$

for $i = 1, 2, \ldots, m$.

- **Newton Iteration**

$$G_k(t, s, u, v) = k(t, s, u) + \frac{\partial k}{\partial y}(t, s, y)|_{y=u}[v - u].$$

For the Volterra integro-differential equation (1.19) we must also choose a splitting function $G_f : [0, T] \times \Re^m \times \Re^m \to \Re^m$, where $0 < T < \infty$, for the function $f(t, y)$ such that

$$G_f(t, u, u) = f(t, u),$$

for all $t \in [0, T]$ and for all $u \in \Re^m$. See Section 1.3, for typical forms of G_f.

Then we compute a sequence $y^1(t), y^2(t), \ldots, y^q(t)$, of solutions of the equations

$$\begin{aligned} \frac{d}{dt}y^q(t) &= G_f(t, y^{q-1}, y^q) + \int_0^t G_k(t, s, y^{q-1}(s), y^q(s))ds, \quad t \in [0, T] \quad (1.23) \\ y^q(0) &= y_0, \end{aligned}$$

$q = 1, 2, \ldots$, and $y^0(t)$ is arbitrary, which converges to the solution $y(t)$ of (1.19) as $q \to \infty$.

We may use any of the iteration schemes mentioned above for G_k and any of the iteration schemes mentioned in Section 1.3 for G_f. See the discussion at the end of that section which compares the four iteration modes.

1.4.1 Continuous-Time Iteration WR Methods

Recall that these methods given by (1.22) and (1.23) are called continuous-time iteration WR methods, since they do not involve the discretization of the time interval and the application of numerical methods to solve the resulting Volterra equations.

In the case of the Volterra integral equation (1.18), see Crisci *et al.* (1996a) for convergence results for the continuous time WR method (1.22). For example, it is shown that if the splitting function $G_k(t,s,u,v)$ satisfies the following uniform Lipschiz condition in u and v, with Lipschitz constants L_1 and L_2, respectively,

$$\|G_k(t,s,u_1,v_1) - G_k(t,s,u_2,v_2)\| \leq L_1\|u_1 - u_2\| + L_2\|v_1 - v_2\|, \qquad (1.24)$$

for all u_1, u_2, v_1, v_2 in $\Re^m$ and for all $(t,s) \in S$, then the resulting WR method converges in $[0,T_1]$, where

$$T_1 < \frac{1}{L_1 + L_2}$$

(see Crisci *et al.* (1996a)). Note that we are referring to the usual maximum norm from the Banach space of continuous vector valued functions, defined on [0,T] inherited from any vector norm of $\Re^m$. (Note: I beleive the WR method converges, as in the VIDE case, on any bounded $[0,T]$. See Theorem 2.7).

These authors also show that convergence of (1.22) can be extended to any finite interval $[0,T]$, if a linear kernel is assumed in (1.18). In addition, they consider the following linear convolution kernel

$$k(t,s,y(s)) = [A + B(t-s)]y(s),$$

for (1.18) and show that convergence of (1.22) can be extended to the infinite interval $[0, \infty)$.

The case of Volterra integro-differential equations will be considered in Chapter 2 of this thesis. We will see that VIDEs have more in common with ODEs than VIEs, with regard to convergence results. For example, assume that, in line with the ODE case and the VIE case, the splitting functions for f and k satisfy conditions (1.4) and (1.24), respectively. Then the method converges uniformly for all finite $T > 0$, and a result similar to that given by (1.5) can be given, see Theorem 2.7.

1.4.2 Time Discretization

In analogy with the case for ODEs in Section 1.3.2, (1.22) and (1.23) can seldom be solved in closed form, so we must discretize the time interval $[0, T]$ and apply a numerical method to approximate the solution. Therefore, we consider the numerical solution of (1.18) and (1.19). We can use a linear method (VLM) or a Runge-Kutta method (VRK), see Brunner and van der Houwen (1986), Chapters 3 and 4, respectively.

Consider a partition $\Pi_N : 0 = t_0 < t_1 < \ldots < t_N = T$, $t_{n+1} - t_n = h$, for $n = 0, 1, \ldots, N-1$. We deal with the Volterra integral equation and the Volterra integro-differential equation separately.

Volterra integral equation:

Rewrite (1.18) in a form related to this partition:

$$y(t) = F_n(t) + h\Phi_n(t), \quad t \in [t_n, t_{n+1}], \quad n = 0, 1, \ldots, N-1, \tag{1.25}$$

where the expression F_n defined by

$$F_n(t) := f(t) + \int_0^{t_n} k(t, s, y(s))ds, \quad t \in [t_n, t_{n+1}], \quad n = 0, 1, \ldots, N-1,$$

is called the *lag* or *tail* term, and the function $\Phi_n(t)$, defined by

$$h\Phi_n(t) := \int_{t_n}^{t} k(t, s, y(s))ds, \quad t \in [t_n, t_{n+1}], \quad n = 0, 1, \ldots, N-1,$$

is called the (exact) *increment function* for the interval $[t_n, t_{n+1}]$. The numerical method must now approximate both the lag term and the increment function. In general, these are independent problems.

Let y_{n+1} represent the approximation to the exact solution $y(t_{n+1})$ of (1.25), and let us use an r-stage Runge-Kutta method to generate these approximations. This leads to the *Pouzet-Volterra-Runge-Kutta* method (PVRK):

$$\begin{aligned}
y_{n+1} &= \tilde{F}_n(t_n + h) + h\tilde{\Phi}_n(t_n + h), \qquad (1.26)\\
\tilde{\Phi}_n(t) &= \sum_{i=1}^{r} b_i k(t, t_n + c_i h, Y_{n,i}),\\
Y_{n,i} &= \tilde{F}_n(t_n + c_i h) + h\sum_{j=1}^{r} a_{ij} k(t_n + c_i h, t_n + c_j h, Y_{n,j}),\\
& \quad i = 1, 2, \ldots, r\\
\tilde{F}_n(t) &= f(t) + h\sum_{l=0}^{n-1}\sum_{s=1}^{r} b_s k(t, t_l + c_s h, Y_{l,s}),
\end{aligned}$$

$n = 0, 1, \ldots, N-1$.

Volterra integro-differential equation:

Rewrite (1.19):

$$y'(t) = f(t, y) + F_n(t) + h\Phi_n(t), \quad t \in [t_n, t_{n+1}], \; n = 0, 1, \ldots, N-1, \; (1.27)$$

$$y(0) = y_0,$$

where we call

$$F_n(t) := \int_0^{t_n} k(t,s,y(s))ds, \quad t \in [t_n, t_{n+1}], \quad n = 0,1,\ldots,N-1,$$

the *lag* or *tail* term; the function $\Phi_n(t)$, defined by

$$h\Phi_n(t) := \int_{t_n}^{t} k(t,s,y(s))ds, \quad t \in [t_n, t_{n+1}], \quad n = 0,1,\ldots,N-1,$$

is called the (exact) *increment function* for the interval $[t_n, t_{n+1}]$.

As in the case of VIEs, the numerical method must now deal with the two (independent) problems of approximating the lag term and the increment function. However, the VIDE has the additional ODE-type problem due to the term $f(t,y)$.

As in the case of VIEs, we use an r-stage Runge-Kutta method to generate the approximation y_{n+1} to the exact solution $y(t_{n+1})$ of (1.27). This leads to the *Pouzet-Volterra-integro-differential-Runge-Kutta* method (PVDRK):

$$\begin{aligned}
y_{n+1} &= y_n + h\sum_{i=1}^{r} b_i \left\{ f(t_n + c_i h, Y_{n,i}) + \tilde{F}_n(t_n + c_i h) + h\tilde{\Phi}_n(t_n + c_i h) \right\}, \\
&\quad n = 0,1,\ldots,N-1, \qquad (1.28) \\
Y_{n,i} &= y_n + h\sum_{j=1}^{r} a_{ij} \left\{ f(t_n + c_j h, Y_{n,j}) + \tilde{F}_n(t_n + c_j h) + h\tilde{\Phi}_n(t_n + c_j h) \right\}, \\
&\quad i = 1,2,\ldots,r, \\
\tilde{\Phi}_n(t_n + c_j h) &= \sum_{l=1}^{r} a_{jl} k(t_n + c_j h, t_n + c_l h, Y_{n,l}), \quad j = 1,2,\ldots,r, \\
\tilde{F}_n(t) &= h\sum_{l=0}^{n-1}\sum_{s=1}^{r} b_s k(t, t_l + c_s h, Y_{l,s}), \quad n = 0,1,\ldots,N-1. \qquad (1.29)
\end{aligned}$$

A VRK or VDRK method, whose lag term formula is as in (1.26) or (1.28), is called an *extended* PVRK or PVDRK method, respectively. Note that these methods are completely characterized by their *Butcher arrays*:

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

and the corresponding RK methods for ODEs is called the *associated* RK method. Recall the definition of order of RK methods (for ODEs) given by Definition 1.1, and subsequent discussion. Clearly these definitions apply to the case of Volterra equations. See Brunner and van der Houwen, (1986), Chapter 4, for the following result.

Theorem 1.5 *(Brunner et al. 1982, Lubich 1981 and 1982) If the associated Runge-Kutta method of a Pouzet-Volterra-Runge-Kutta method or a Pouzet-Volterra-integro-differential-Runge-Kutta method has order p, then these methods have order p.*

Recall that the *row-sum* condition always holds:

$$c_i = \sum_{j=1}^{r} a_{ij}, \qquad i = 1, 2, \ldots, r,$$

and in analogy to RK methods for ODEs, we can define explicit, semi-implicit and implicit PVRK and PVDRK methods; see Section 1.3.2.

In Section 1.3.2 we discussed collocation methods as examples of continuous implicit Runge-Kutta methods for the solution of ODEs. The same discussion is

relevant to Volterra equations. Our focus is on Volterra integro-differential equations; for Volterra integral equations many similar techniques apply. See Brunner and van der Houwen (1986).

We now extend collocation methods to Volterra integro-differential equations (see also Brunner and van der Houwen (1986)). Volterra integro-differential equations will employ continuous collocation approximations in the space $S_r^{(0)}(Z_N)$. Collocation for the solution of the Volterra integro-differential equation (1.19) can be derived in a completely analogous fashion to the derivation of collocation for the solution of the ODE (1.1). See Section 1.3.2.

Recall the partition $\Pi_N : 0 = t_0 < t_1 < \cdots < t_N = T$, $t_{n+1} - t_n =: h$, (for $n = 0, 1, \ldots, N-1$), the set $Z_N := \{t_n : n = 1, 2, \ldots, N-1\}$ of interior mesh points, and $\bar{Z}_N := Z_N \cup T$. Also, recall the set of collocation points $T_N = \{t_n + c_j h; j = 1, 2, \ldots, r;\ n = 0, 1, \ldots, N-1\}$, where $0 \leq c_1 < c_2 < \cdots < c_r \leq 1$.

Consider collocation on $S_r^{(0)}(Z_N)$ for the solution of (1.19), that is, choose $\eta \in S_r^{(0)}(Z_N)$ such that

$$\eta'(t) = f(t, \eta) + \int_0^t k(t, s, \eta(s))ds, \quad \eta(0) = y_0, \quad t \in T_N. \tag{1.30}$$

Note that there are Nr collocation conditions plus the given initial condition $\eta(0) = y_0$; this number equals the dimension of the space $S_r^{(0)}(Z_N)$; see (1.8). Therefore, since, $\eta'(t_n + \tau h) \in \pi_{r-1}$, for $n = 0, 1, \ldots, N-1$ and $\tau \in (0, 1]$, we can use Lagrange polynomial interpolation to get

$$\eta'(t_n + \tau h) = \sum_{j=1}^{r} \eta'(t_n + c_j h) L_j(\tau), \quad \tau \in [0, 1],$$

where $L_j(\tau)$ are the fundamental Lagrange polynomials,

$$L_j(\tau) = \prod_{\substack{k=1 \\ k \neq j}}^{r} \frac{\tau - c_k}{c_j - c_k}.$$

Then

$$\eta(t_n + \tau h) = \eta(t_n) + h \int_0^\tau \eta'(t_n + vh) dv,$$

which becomes

$$\begin{aligned} \eta(t_n + \tau h) &= \eta(t_n) + h \sum_{j=1}^{r} \alpha_j(\tau) Y_{n,j}, \quad \tau \in [0,1] \\ Y_{n,j} &= f(t_n + c_j h, \eta(t_n + c_j h)) + \int_0^{t_n} k(t_n + c_j h, v, \eta(v)) dv \\ &+ \int_{t_n}^{t_n + c_j h} k(t_n + c_j h, v, \eta(v)) dv, \end{aligned}$$

for $n = 0, 1, \ldots, N-1$, where

$$\alpha_j(\tau) = \int_0^\tau L_j(v) dv.$$

These equations can then be written as

$$\begin{aligned} \eta(t_n + \tau h) &= \eta(t_n) + h \sum_{j=1}^{r} \alpha_j(\tau) Y_{n,j}, \quad \tau \in [0,1] \\ Y_{n,j} &= f\left(t_n + c_j h, \eta(t_n) + h \sum_{k=1}^{r} \alpha_k(c_j) Y_{n,k}\right) \\ &+ h \sum_{l=0}^{n-1} \int_0^1 k\left(t_n + c_j h, t_l + \tau h, \eta(t_l + \tau h)\right) d\tau \\ &+ h \int_0^{c_j} k\left(t_n + c_j h, t_n + \tau h, \eta(t_n) + h \sum_{k=1}^{r} \alpha_k(\tau) Y_{n,k}\right) d\tau, \end{aligned} \tag{1.31}$$

for $n = 0, 1, \ldots, N-1$.

Observe that the collocation solution defined by (1.31) is not only defined throughout the interval of definition $[0, T]$, but in fact, is continuous there, as well. Such a method is called *exact* since the integrals appearing have not been approximated by suitable quadrature formulas. For most realistic problems, it will not be possible to evaluate these integrals in closed form, but they will have to be approximated. This leads to *fully discretized* methods. In light of the importance of these considerations, we give a few relevant results on numerical quadrature. See Brunner and van der Houwen (1986), Chapter 2.

Discretized Methods

Consider the integral

$$I(\phi) = \int_a^b \phi(t)dt,$$

where a and b are finite and $a < b$. Then the numerical approximation $I_n(\phi)$ to $I(\phi)$ is given by

$$I_n(\phi) = \sum_{j=0}^{n} c_{n,j}\phi(t_{n,j}), \tag{1.32}$$

where the points $t_{n,0} < t_{n,1} < \cdots < t_{n,n}$ are called the *abscissas* and do not necessarily lie in the interval $[a, b]$. We call the $\{c_{n,j}\}_{j=0}^{n}$ the *coefficients* or *quadrature weights* of the method. Also we call

$$E_n(\phi) = I(\phi) - I_n(\phi)$$

the *quadrature error* and $I_n(\phi)$ the $(n+1)$-point *quadrature formula.* We say that a quadrature formula is *exact* for a function ϕ if $E_n(\phi) = 0$.

Definition 1.5 *A quadrature formula has degree of precision q if it is exact for all polynomials ϕ of degree not exceeding q, but if $E_n(t^{q+1}) \neq 0$.*

Let us *interpolate* the function ϕ by Lagrange polynomial interpolation,

$$p_n[t;\phi] = \sum_{j=0}^{n} \phi(t_{n,j}) L_{n,j}(t),$$

where $p_n[t;\phi]$ is the polynomial interpolation of $\phi(t)$ at the $n+1$ points $t_{n,0} < t_{n,1} < \cdots < t_{n,n}$ not necessarily lying in the interval $[a,b]$ and the $L_{n,j}$ are given by (1.11). Then the quadrature weights in (1.32) are given by

$$c_{n,j} = \int_a^b L_{n,j}(t)dt. \tag{1.33}$$

Such a formula is called *interpolatory*, and since the coefficients are unique, we get the following result.

Theorem 1.6 *(Brunner and van der Houwen 1986) A quadrature formula based on $n+1$ (distinct) points is of interpolatory type if and only if its degree of precision q is greater than or equal to n.*

See Brunner and van der Houwen (1986), Section 2.1.2.

Therefore an $(n+1)$-point interpolatory quadrature formula has a minimum degree of precision equal to n. However, this degree of precision can be exceeded for particular choices of the abcissas $\{t_{n,j}\}$. The maximum degree of precision possible is $2n+1$ and occurs when the *Gauss* points are taken as the abcissas. We then get the *Gauss-Legendre* formulas. See Brunner and van der Houwen (1986), Section 2.3.

We now assume that interpolatory quadrature based on the abscissas $\{t_n + c_i h : i = 1,2,\ldots,r\}$ is used to approximate the integrals in (1.31). This will lead to the *fully discretized* method. Find $\eta \in S_r^{(0)}(Z_N)$ from:

$$\eta(t_n + \tau h) \;=\; \eta(t_n) + h\sum_{j=1}^{r} \alpha_j(\tau) Y_{n,j}, \quad \tau \in [0,1] \tag{1.34}$$

$$
\begin{aligned}
Y_{n,j} &= f\left(t_n + c_j h, \eta(t_n) + h\sum_{k=1}^{r}\alpha_k(c_j)Y_{n,k}\right) \\
&+ h\sum_{l=0}^{n-1}\sum_{k=1}^{r}\alpha_k(1)k\left(t_n + c_j h, t_l + c_k h, \eta(t_l + c_k h)\right) \\
&+ h\sum_{k=1}^{r}\alpha_k(c_j)k\left(t_n + c_j h, t_n + c_k h, \eta(t_n) + h\sum_{s=1}^{r}\alpha_s(c_k)Y_{n,s}\right), \\
&\quad j = 1, 2, \ldots, r,
\end{aligned}
$$

for $n = 0, 1, \ldots, N-1$.

Next, setting $\tau = 1$, (1.34) becomes

$$
\begin{aligned}
\eta(t_{n+1}) &= \eta(t_n) + h\sum_{j=1}^{r}\alpha_j(1)Y_{n,j}, \qquad (1.35) \\
Y_{n,j} &= f\left(t_n + c_j h, \eta(t_n) + h\sum_{k=1}^{r}\alpha_k(c_j)Y_{n,k}\right) \\
&+ h\sum_{l=0}^{n-1}\sum_{k=1}^{r}\alpha_k(1)k\left(t_n + c_j h, t_l + c_k h, \eta(t_l + c_k h)\right) \\
&+ h\sum_{k=1}^{r}\alpha_k(c_j)k\left(t_n + c_j h, t_n + c_k h, \eta(t_n) + h\sum_{s=1}^{r}\alpha_s(c_k)Y_{n,s}\right), \\
&\quad j = 1, 2, \ldots, r,
\end{aligned}
$$

for $n = 0, 1, \ldots, N-1$.

The coefficients are given by:

$$
a_{jk} = \alpha_k(c_j) = \int_0^{c_j} L_k(\tau)d\tau \quad \text{and} \quad b_j = \alpha_j(1) = \int_0^1 L_j(\tau)d\tau, \quad j, k = 1, 2, \ldots, r.
$$

Thus we have the following theorems:

Theorem 1.7 *(Brunner and van der Houwen 1986) Consider the Volterra integro-differential equation (1.19) and suppose it is solved by collocation using the collocation points T_N in the space $S_r^{(0)}(Z_N)$. Also assume that the integrals appearing*

in the resulting method are approximated by interpolatory quadrature, based on the abscissas $\{t_n + c_j h : n = 0, 1, \ldots, N-1, j = 1, 2, \ldots, r\}$, *leading to the discretized method given by (1.35), where we assume that the kernel can be smoothly extended to values outside the triangle* S. *Then this method coincides with the implicit extended Pouzet-Volterra-integro-differential-Runge-Kutta method given by (1.28).*

See Brunner and van der Houwen (1986), p. 291.

Theorem 1.8 *(Brunner and van der Houwen 1986) Consider the Volterra integro-differential equation (1.19), where* f *and* k *are* r *times continuously differentiable on* $[0, T]$ *and* S, *respectively, and suppose it is solved by collocation using the collocation points* T_N *in the polynomial spline space* $S_r^{(0)}(Z_N)$. *Also assume that the integrals appearing in the resulting method are approximated by interpolatory quadrature, based on the abscissas* $\{t_n + c_j h : n = 0, 1, \ldots, N-1, j = 1, 2, \ldots, r\}$, *leading to the discretized method given by (1.34). Then this method has order of global convergence* r *throughout* $[0, T]$, *regardless of the choice of the (distinct) collocation parameters,* $\{c_j\}$.

See Brunner and van der Houwen (1986), Corollary 5.4.1.

It is possible to attain superconvergence at the nodes t_n, $n = 1, 2, \ldots, N-1$. If the Radau I points for $[0, 1)$ or Radau II points for $(0, 1]$ are taken for the collocation parameters $\{c_i\}$, we get nodal order $2r-1$ or if the Lobatto points for $[0, 1]$ are used, we get nodal order $2r-2$. Fortunately, we also attain the optimal order $2r$ (as in ODEs) with the Gauss points. Clearly, in this way, a Volterra integro-differential equation is "closer" to an ordinary differential equation, than to a Volterra integral equation. See Brunner and van der Houwen (1986), Sections 5.4.2 and 5.4.3.

So far the methods we have considered for the numerical solution of Volterra equations have employed kernel values $k(t,s,y)$ with $s > t$ in the discretization of the increment function Φ_n which are outside the natural definition of the kernel. This can be rectified by a transformation and the resulting methods are called *modified methods of de Hoog and Weiss*; see Brunner and van der Houwen (1986), p. 263.

In particular, for the VIDE (1.19), we replace the abscissas, $\{t_n + c_k h : n = 0,1,\ldots,N-1, k = 1,2,\ldots,r\}$ in (1.34) by the abscissas, $\{t_n + c_k c_j h : n = 0,1,\ldots,N-1; j,k = 1,2,\ldots,r\}$. To better see how this works, consider the following integral from (1.31)

$$\begin{aligned} & \int_0^{c_j} k\left(t_n + c_j h, t_n + \tau h, \eta(t_n) + h\sum_{s=1}^{r}\alpha_s(\tau)Y_{n,s}\right)d\tau \\ = \; & c_j \int_0^1 k\left(t_n + c_j h, t_n + c_j\xi h, \eta(t_n) + h\sum_{s=1}^{r}\alpha_s(c_j\xi)Y_{n,s}\right)d\xi \qquad (1.36) \\ \approx \; & \sum_{k=1}^{r} c_j\alpha_k(1)k\left(t_n + c_j h, t_n + c_k c_j h, \eta(t_n) + h\sum_{s=1}^{r}\alpha_s(c_k c_j)Y_{n,s}\right), \end{aligned}$$

where we have used a transformation of variable and interpolatory quadrature with the weights given as in (1.33).

The method becomes,

$$\begin{aligned} \eta(t_n + \tau h) &= \eta(t_n) + h\sum_{j=1}^{r}\alpha_j(\tau)Y_{n,j}, \quad \tau \in [0,1] \qquad (1.37) \\ Y_{n,j} &= f\left(t_n + c_j h, \eta(t_n) + h\sum_{k=1}^{r}\alpha_k(c_j)Y_{n,k}\right) \\ &+ h\sum_{l=0}^{n-1}\sum_{k=1}^{r}\alpha_k(1)k\left(t_n + c_j h, t_l + c_k h, \eta(t_l + c_k h)\right) \\ &+ h\sum_{k=1}^{r} c_j\alpha_k(1)k\left(t_n + c_j h, t_n + c_k c_j h, \eta(t_n) + h\sum_{s=1}^{r}\alpha_s(c_k c_j)Y_{n,s}\right), \\ & j = 1,2,\ldots,r, \end{aligned}$$

for $n = 0, 1, \ldots, N-1$.

Since it is these methods that we use in the thesis and since we are primarily interested in methods of optimal order, we give the following theorem.

Theorem 1.9 *(Brunner 1984) Consider the Volterra integro-differential equation (1.19), where f and k are $2r$ times continuously differentiable on $[0,T]$ and S, respectively, and suppose it is solved by collocation using the collocation points T_N in the polynomial spline space $S_r^{(0)}(Z_N)$. Also assume that the integrals appearing in the resulting method are solved by interpolatory quadrature, based on the abscissas $\{t_l + c_k h : l = 0, 1, \ldots, n-1, k = 1, 2, \ldots, r\}$ and $\{t_n + c_k c_j h : n = 0, 1, \ldots, N-1; j, k = 1, 2, \ldots, r\}$, leading to the discretized method given by (1.37). Then this method has a nodal order of $2r$, iff the r collocation parameters, $\{c_j\}$, are taken to be the Gauss points in $(0,1)$.*

1.4.3 Discrete-Time Iteration WR Methods

In Section 1.3.3, we considered discrete-time iteration WR methods for the solution of ODEs. We now turn our attention to discrete-time iteration WR methods for Volterra equations. We recall that these are waveform relaxation methods that involve discretization of the time interval and application of numerical methods to solve the resulting VIE, given by (1.22), or the VIDE, given by (1.23).

Consider a partition $\Pi_N : 0 = t_0 < t_1 < \cdots < t_N = T$, $t_{n+1} - t_n =: h$, for $n = 0, 1, \ldots, N-1$, and divide the integration interval $[0,T]$ into κ subintervals, called *windows*, each of length $\mathcal{N}h$, i.e.

$$[0,T] = \bigcup_{s=0}^{\kappa-1} [sh\mathcal{N}, (s+1)h\mathcal{N}],$$

where $\mathcal{N} = \frac{T}{\kappa h}$. Then the discrete-time WR method is applied, in turn, to each window. In this thesis, we concentrate on a special case of the WR method, called the time-point relaxation (TR) method, which results when $\mathcal{N} = 1$. See Definition 1.4.

If the PVRK method (1.26) is applied to the numerical solution of (1.22), for each iteration q, the result is the *discrete-time time-point relaxation Volterra Runge-Kutta* method, or simply TRPVRK method, Crisci *et al.*, (1997a). We give a few details.

The approximation $\eta^q(t)$ to the exact solution $y(t)$ is evaluated a fixed number of times, say $q = 1, 2, \ldots, q_f$ on each window

$$[0, t_1], [t_1, t_2], \ldots [t_{(N-1)}, t_N],$$

using the PVRK method (1.26). The approximation on the interval $[t_n, t_{n+1}]$ is given by

$$\begin{aligned}
\eta^{q_f}(t_n + h) &= \tilde{F}_n^{q_f}(t_n + h) + h\tilde{\Phi}_n^{q_f}(t_n + h), \qquad (1.38)\\
\tilde{\Phi}_n^{q_f}(t) &= \sum_{i=1}^{r} b_i k(t, t_n + c_i h, Y_{n,i}^{q_f}),\\
Y_{n,i}^{q} &= \tilde{F}_n^{q_f}(t_n + c_i h) + h\sum_{j=1}^{r} a_{ij} G_k(t_n + c_i h, t_n + c_j h, Y_{n,j}^{q-1}, Y_{n,j}^{q}),\\
\tilde{F}_n^{q_f}(t) &= f(t) + h\sum_{l=0}^{n-1}\sum_{s=1}^{r} b_s k(t, t_l + c_s h, Y_{l,s}^{q_f}),\\
Y_{n,i}^{0} &= \tilde{F}_n^{q_f}(t_n + c_i h),
\end{aligned}$$

$i = 1, 2, \ldots, r$, $q = 1, 2, \ldots, q_f$ and $n = 0, 1, \ldots, N-1$, where G_k is the splitting function; see Section 1.4.

We refer to this method as the TRPVRK method.

If the splitting function G_k satisfies the Lipschitz condition (1.24), then the method (1.38) is convergent as $h \to 0$, for every q_f. In analogy with Theorem 1.4, we have

Theorem 1.10 *(Crisci et al. 1997a) Assume that the functions f and k in (1.18) are sufficiently smooth and the splitting function G_k in (1.22) satisfies the Lipschitz condition (1.24). Then if the underlying PVRK (1.26), has order p^*, the TRPVRK method (1.38) has order $p = min\{p^*, q+1\}$, for q iterations.*

See Crisci *et al.* (1997a).

Concerning the VIDE (1.19), we can apply a PVDRK method (1.28) to solve (1.23), for each iteration q. The resulting method is a *discrete-time time-point relaxation Pouzet-Volterra-integro-differential-Runge-Kutta method* (TRPVDRK). However we shall use the collocation method (1.37) in this thesis. In Chapter 2, we will consider this method in detail.

1.4.4 Weakly Singular Kernels

So far we have restricted our discussion to Volterra equations with regular kernels; see (1.18) and (1.19). We now drop this restriction and consider the class of Volterra equations with weakly singular kernels. That is, in both of these problems, we let

$$k(t, s, y(s)) := (t-s)^{-\alpha} K(t, s, y(s)), \tag{1.39}$$

where $0 < \alpha < 1$, $(t, s) \in S$, and K is a regular kernel.

The second-kind Volterra integral equation with weakly singular kernel is given by

$$y(t) = f(t) + \int_0^t (t-s)^{-\alpha} K(t,s,y(s))ds, \qquad t \in [0,T], \tag{1.40}$$

where $0 < \alpha < 1$, y is a vector on $\Re^m$, $f : [0,T] \to \Re^m$ and $K : S \times \Re^m \to \Re^m$, where $0 < T < \infty$. We assume that (1.40) possesses a unique solution $y \in C([0,T])$; see Brunner and van der Houwen (1986), Chapter 1.

The Volterra integro-differential equation with weakly singular kernel is given by:

$$y'(t) = f(t,y) + \int_0^t (t-s)^{-\alpha} K(t,s,y(s))ds, \quad y(0) = y_0, \quad t \in [0,T], \tag{1.41}$$

where $0 < \alpha < 1$, y is a vector on $\Re^m$, $f : [0,T] \times \Re^m \to \Re^m$ and $K : S \times \Re^m \to \Re^m$, where $0 < T < \infty$. We assume that (1.41) possesses a unique solution $y \in C^1([0,T])$; see Brunner and van der Houwen (1986), Chapter 1.

The results for Volterra equations with regular kernels, in the previous section, assumed polynomial spline collocation employing a uniform mesh $\Pi : 0 = t_0 < t_1 < \cdots < t_N$, $t_{n+1} - t_n = h$, for $n = 0, 1, \ldots, N-1$. Under such conditions, the global convergence error for the solution of (1.40) and (1.41) drops to $\mathcal{O}(h^{1-\alpha})$, where $0 < \alpha < 1$, regardless of how we choose the collocation parameters and regardless of how we choose the degree of the underlying polynomials. In fact, this is the best we can do in the sense that the exponent $1 - \alpha$ cannot be replaced by any number greater than $1 - \alpha$. Furthermore, superconvergence is lost, as well. To do better, one must consider suitable graded meshes for polynomial collocation, or nonpolynomial spline spaces for uniform meshes. See Brunner (1986a), Brunner and van der Houwen (1986), Chapter 1 and 6, Brunner (1998), Tang (1992 and 1993),

and Brunner *et al.* (1998). See also Chapter 3.

Chapter 2

Waveform Relaxation Methods for Volterra Integro-Differential Equations

2.1 The Analytic Solution

In this thesis we will study systems of *Volterra integro-differential equations* (VIDEs) which involve, in a very natural way, elements from systems of ordinary differential equations (ODEs) and systems of Volterra integral equations (VIEs). We begin by giving some standard results on the existence and smoothness of solutions for VIDEs. See Brunner and van der Houwen (1986), Chapter 1.

Theorem 2.1 *(Miller 1971) Given the initial value problem (1.19). If the function f is defined and continuous on the bounded interval $[0,T]$ and the kernel k is defined and continuous on the triangle S, and both satisfy the following Lipschitz conditions:*

$$\|f(t,y) - f(t,\tilde{y})\| \leq L_f \|y - \tilde{y}\|,$$

$$\|k(t,s,z) - k(t,s,\tilde{z})\| \leq L_k \|z - \tilde{z}\|,$$

where $(t,s) \in S$ *and* $y,\ \tilde{y},\ z,\ \tilde{z} \in \Re^m$, *then (1.19) possesses a unique solution that is continuous on* $[0,T]$, *and possesses a continuous derivative on* $[0,T]$.

Note: We have required that the functions f and k in the above theorem, satisfy Lipschitz conditions for $y \in \Re^m$. If however, these conditions apply only on some compact region of $\Re^m$, we can only give a *local* existence theorem. That is, we can only be sure of the existence of a (unique) continuous solution on some neighbourhood $(-\delta, \delta)$, for some $0 < \delta < T$. See Brunner and van der Houwen (1986). $\diamond$

If we assume a special linear equation, we can write down the analytic solution, and subsequently, give a more complete analysis of the smoothness of these solutions. Therefore, we consider the following example, where we take the one-dimensional case, $m = 1$; the analysis of the m-dimensional case is essentially the same.

Example 2.1

$$y'(t) = a(t)y(t) + g(t) + \int_0^t k(t,s)y(s)ds, \qquad y(0) = y_0, \qquad t \in [0,T], \tag{2.1}$$

where $a, g : [0,T] \to \Re$, $k : S \to \Re$, *with* $a, g \in C([0,T])$ *and* $k \in C(S)$.

Under these conditions (2.1) possesses a unique solution on $[0,T]$, which we can explicitly write down in terms of a special kernel, continuous on S; see Theorem 2.2. To motivate this discussion, we take the following simple example in which we can use ODE theory to write down this special kernel. As a bonus, we will use this example in the proof of Theorem 2.7 in Section 2.2.1.

Example 2.2

$$y'(t) = ay(t) + g(t) + \int_0^t by(s)ds, \qquad y(0) = y_0, \qquad t \in [0,T], \tag{2.2}$$

where a and b are non-zero real constants, $g : [0,T] \to \Re$ and $g \in C^1([0,T])$.

Clearly (2.2) is equivalent to the second-order linear constant coefficient ODE

$$y''(t) = ay'(t) + by(t) + g'(t), \quad y(0) = y_0, \quad y'(0) = ay_0 + g(0) \quad t \in [0,T], \quad (2.3)$$

which is equivalent to the first order system

$$\begin{pmatrix} y(t) \\ y'(t) \end{pmatrix}' = \begin{pmatrix} 0 & 1 \\ b & a \end{pmatrix} \begin{pmatrix} y(t) \\ y'(t) \end{pmatrix} + \begin{pmatrix} 0 \\ g'(t) \end{pmatrix}, \quad (2.4)$$

where $[y_0, ay_0 + g(0)]^T$ are known initial values.

The solution can then be found by applying the *variation of constants formula*; see Edwards and Penney (1994), Section 2.7 for a solution based on equation (2.3) and Hairer, *et al.* (1993) for a solution based on equation (2.4). We give details for the first case.

Consider the characteristic equation of (2.3),

$$\lambda^2 - a\lambda - b = 0,$$

and for simplicity, assume the roots are real and distinct, that is

$$\lambda^2 - a\lambda - b = (\lambda - \lambda_1)(\lambda - \lambda_2), \quad (2.5)$$

where $\lambda_1 \neq \lambda_2, \quad \lambda_i \in \Re, \ i = 1, 2$. By expanding the RHS of (2.5),

$$\lambda_1 + \lambda_2 = a. \quad (2.6)$$

Note, that λ_1, λ_2 are the eigenvalues of the matrix

$$\begin{pmatrix} 0 & 1 \\ b & a \end{pmatrix},$$

associated with (2.4).

Clearly the complementary solution of (2.3), y_c, is given by

$$y_c(t) = c_1 e^{\lambda_1 t} + c_2 e^{\lambda_2 t},$$

where c_1, c_2 are arbitrary constants. By the variation of constants formula, we can write the particular solution y_p of (2.3) in the form,

$$y_p(t) = \int_0^t G(t,s) g'(s) ds, \tag{2.7}$$

where

$$G(t,s) = \frac{e^{\lambda_1(t-s)} - e^{\lambda_2(t-s)}}{\lambda_1 - \lambda_2}, \tag{2.8}$$

$(t,s) \in S$, where G is called a *Green's function* for (2.3), see Edwards and Penney (1994). Applying integration by parts to (2.7), and using (2.8), where we note that $G(t,t) = 0, \quad t \in [0,T]$, we obtain

$$y_p(t) = -G(t,0)g(0) - \int_0^t \frac{\partial}{\partial s} G(t,s) g(s) ds,$$

in which case the solution to (2.3) is

$$y(t) = c_1 e^{\lambda_1 t} + c_2 e^{\lambda_2 t} - G(t,0)g(0) - \int_0^t \frac{\partial}{\partial s} G(t,s) g(s) ds. \tag{2.9}$$

Using Leibniz's Rule, see Zwillinger (1992), we now apply the initial conditions

$$\begin{aligned} y(0) &= y_0 \\ y'(0) &= a y_0 + g(0), \end{aligned}$$

in which case the constants c_1, c_2 become

$$c_1 = \frac{y_0(a - \lambda_2) + g(0)}{\lambda_1 - \lambda_2}, \quad c_2 = -\frac{y_0(a - \lambda_1) + g(0)}{\lambda_1 - \lambda_2}.$$

By (2.6) they can be written as

$$c_1 = \frac{y_0\lambda_1 + g(0)}{\lambda_1 - \lambda_2}, \quad c_2 = -\frac{y_0\lambda_2 + g(0)}{\lambda_1 - \lambda_2}. \tag{2.10}$$

Define the *resolvent kernel* $R(t,s)$ by setting:

$$R(t,s) = -\frac{\partial G(t,s)}{\partial s} = \frac{\lambda_1 e^{\lambda_1(t-s)} - \lambda_2 e^{\lambda_2(t-s)}}{\lambda_1 - \lambda_2}, \quad (t,s) \in S. \tag{2.11}$$

Therefore, using (2.11), (2.10), and (2.9), the solution to (2.3) and therefore (2.2) becomes

$$y(t) = R(t,0)y_0 + \int_0^t R(t,s)g(s)ds. \tag{2.12}$$

We note that, in this simple case of (2.2), its resolvent kernel, defined by (2.11), is of the form $R(t-s)$. Such kernels form an important class of kernels called *convolution* kernels.

Fortunately, this approach to the solution of (2.2), can be extended to the more general equation (2.1). We formally define the following function. See Grossman and Miller (1970) and Brunner (1986b).

Definition 2.1 *Let the function $R : S \to \Re$ be defined by*

$$\begin{aligned} \frac{\partial R(t,s)}{\partial s} &= -R(t,s)a(s) - \int_s^t R(t,\tau)k(\tau,s)d\tau, \quad (t,s) \in S, \\ R(t,t) &= 1, \quad t \geq s. \end{aligned} \tag{2.13}$$

The function R is called the *resolvent* kernel of (2.1), and the above equation (2.13) is called the *resolvent equation* associated with it. We leave it to the reader to check that $R(t,s)$ given by (2.11) satisfies the resolvent equation (2.13).

The following theorem uses this resolvent kernel to give a representation of the solution of (2.1), analogous to the representation of the solution of (2.2), given by (2.12).

Theorem 2.2 *(Grossman and Miller 1970) Let $a \in C([0,T])$, $k \in C(S)$, and assume that R satisfies (2.13). Then, for any $g \in C([0,T])$, the initial-value problem (2.1) has a unique solution $y \in C^1([0,T])$, given by*

$$y(t) = R(t,0)y_0 + \int_0^t R(t,s)g(s)ds, \qquad t \in [0,T]. \tag{2.14}$$

Proof:

For $y \in C^1[0,T]$, the following identity follows from integration by parts:

$$\int_0^t \left\{ R(t,s)y'(s) + \frac{\partial R(t,s)}{\partial s} y(s) \right\} ds = R(t,t)y(t) - R(t,0)y(0), \tag{2.15}$$

for $t \in [0,T]$. Since $R(t,t) = 1$, for all $t \geq s$, we establish

$$y(t) = R(t,0)y(0) + \int_0^t \left\{ R(t,s)y'(s) + \frac{\partial R(t,s)}{\partial s} y(s) \right\} ds, \tag{2.16}$$

for $t \in [0,T]$. If y is a solution to the VIDE (2.1), then substitution of the expression from (2.1) for $y'(s)$ into (2.16) yields

$$\begin{aligned} y(t) &= R(t,0)y_0 + \int_0^t R(t,s) \left\{ a(s)y(s) + g(s) + \int_0^s k(s,\tau)y(\tau)d\tau \right\} ds \\ &+ \int_0^t \frac{\partial R(t,s)}{\partial s} y(s)ds, \end{aligned} \tag{2.17}$$

for $t \in [0,T]$. We rewrite the double integral in (2.17), by interchanging the variables s and τ, that is

$$\int_0^t \int_0^s R(t,s)k(s,\tau)y(\tau)d\tau ds,$$

becomes

$$\int_0^t \int_0^\tau R(t,\tau)k(\tau,s)y(s)dsd\tau.$$

We now apply Dirichlet's formula, see Brunner and van der Houwen (1986), to obtain

$$\int_0^t \left(\int_s^t R(t,\tau)k(\tau,s)d\tau \right) y(s)ds. \tag{2.18}$$

Therefore, using (2.18) we rewrite (2.17) as

$$\begin{aligned} y(t) &= R(t,0)y_0 + \int_0^t R(t,s)g(s)ds \\ &+ \int_0^t \left\{ \frac{\partial R(t,s)}{\partial s} + R(t,s)a(s) + \int_s^t R(t,\tau)k(\tau,s)d\tau \right\} y(s)ds \end{aligned} \tag{2.19}$$

for $t \in [0,T]$. Since $R(t,s)$ satisfies the resolvent equation (2.13),

$$\frac{\partial R(t,s)}{\partial s} + R(t,s)a(s) + \int_s^t R(t,\tau)k(\tau,s)d\tau = 0, \quad t \in [0,T],$$

equation (2.19) becomes

$$y(t) = R(t,0)y_0 + \int_0^t R(t,s)g(s)ds,$$

and the result is proved. □

It is easy to see that the function R, given by (2.11) also satisfies the following equation (2.20), which is called the *adjoint resolvent equation* associated with (2.2).

Definition 2.2 *Let the function $R : S \to \Re$ be defined by*

$$\begin{aligned} \frac{\partial R(t,s)}{\partial t} &= R(t,s)a(t) + \int_s^t k(t,\tau)R(\tau,s)d\tau, \quad (t,s) \in S, \\ R(s,s) &= 1, \quad s \le t. \end{aligned} \tag{2.20}$$

Therefore, for (2.2), Theorem 2.2 is true, if R is given by (2.20) or equivalently by (2.13). Fortunately, this equivalence can be extended to the more general equation (2.1). We refer the reader to the paper by Grossman and Miller (1970), in which the equivalence of (2.13) and (2.20) was first proved. In this thesis, we supply the proof of the corresponding result for the case of a VIDE with a weakly singular kernel, which also follows from Grossman and Miller (1970). See Theorem 2.5.

Note that the resolvent kernel is completely described by the data of the homogeneous equation corresponding to (2.1). It should be pointed out that the representation given by (2.14) can also be derived from rewriting (2.1) as a second-kind VIE and applying the classical results for Volterra integral equations. See Brunner and van der Houwen (1986) and Brunner (1986b).

By Theorem 2.2, the smoothness of the solutions to (2.1) is strongly dependent on the smoothness of the resolvent kernel R. In light of this, we give the following lemma.

Lemma 2.1 *(Brunner 1986b) If $a \in C^n([0,T])$ and $k \in C^n(S)$, for some $n \in \mathcal{N}_0$, then the resolvent kernel R associated with the VIDE (2.1) and given by (2.13) or (equivalently) (2.20), is an element of the space $C^{n+1}(S)$.*

See Brunner (1986b). The following theorem is an immediate consequence of Theorem 2.2 and Lemma 2.1.

Theorem 2.3 *(Brunner 1986b) Let $a \in C^n([0,T])$, $k \in C^n(S)$, for some $n \in \mathcal{N}_0$, and assume that R satisfies (2.13), or (equivalently) (2.20). Then for any $g \in C^\nu([0,T])$, with $\nu \geq n$, the unique solution y of the initial-value problem (2.1) satisfies $y \in C^{n+1}([0,T])$.*

Proof:

If we differentiate (2.14) $n+1$ times, using Leibniz's rule, the highest order derivative of $g(t)$ that appears is of order n, while we get highest order derivatives of order $n+1$ for the resolvent kernel R. The result then follows from Lemma 2.1. □

Consider the following nonlinear VIDE, with a weakly singular kernel, given by

$$y'(t) = f(t,y) + \int_0^t (t-s)^{-\alpha} k(t,s,y(s))ds, \quad y(0) = y_0, \quad t \in [0,T], \tag{2.21}$$

where $0 < \alpha < 1$ and y is a vector in $\Re^m$. Assume f and k are given as in Theorem 2.1. We will assume that this equation posses a unique continuous solution on $[0,T]$, with a continuous first derivative.

However, in many cases we will resort to its linear counterpart; see Theorem 2.5.

We first consider an example analogous to Example 2.2.

Example 2.3

$$y'(t) = ay(t) + g(t) + b\int_0^t (t-s)^{-\alpha} y(s)ds, \quad y(0) = y_0, \quad t \in [0,T], \tag{2.22}$$

where $0 < \alpha < 1$, *a and b are real constants and* $g : [0,T] \to \Re$, *with* $g \in C([0,T])$.

Under these conditions (2.22) possesses a unique solution on $[0,T]$. However, we immediately notice a difference between this example and Example 2.2. Clearly, we cannot use Leibniz's Rule to differentiate (2.22), since the kernel $(t-s)^{-\alpha}$ is unbounded on the diagonal (t,t) where $t \in [0,T]$ and $0 < \alpha < 1$. We can however, rewrite (2.22) as the second-kind VIE

$$y(t) = q(t) + \int_0^t Q(t,s;\alpha)y(s)ds, \tag{2.23}$$

for $t \in [0,T]$, where

$$Q(t,s;\alpha) := a + b\int_s^t (\tau - s)^{-\alpha} d\tau = a + b\frac{(t-s)^{1-\alpha}}{1-\alpha} \tag{2.24}$$

for $(t,s) \in S$, $0 < \alpha < 1$ and

$$q(t) := y_0 + \int_0^t g(s)ds, \quad t \in [0,T].$$

Since the kernel Q, in (2.24) is continuous (hence bounded) on $[0,T]$, (2.23) can be solved by classical methods for Volterra integral equations. See Brunner and van der Houwen (1986) and Brunner (1986b).

If we assume that the nonhomogeneous term in (2.22) has a Laplace transform, we can use the convolution theorem for Laplace transforms to solve it.

Let us denote by $\mathcal{L}\{f(t)\} = F(s)$ the Laplace transform of a function f.

See Edwards and Penney (1994). The proof of the following theorem is omitted.

Theorem 2.4 *Consider Example 2.3. Then (2.22) has a unique solution* $y \in C^1([0,T])$ *given by*

$$y(t) = \Phi(t)y_0 + \int_0^t \Phi(t-s)g(s)ds, \tag{2.25}$$

where

$$\Phi(t) = \mathcal{L}^{-1}\left\{\frac{s^{1-\alpha}}{s^{2-\alpha} - as^{1-\alpha} - b\Gamma(1-\alpha)}\right\}, \quad t \in [0,T], \tag{2.26}$$

with $0 < \alpha < 1$.

Denote the resolvent (convolution) kernel R associated with (2.22) by,

$$R(t,s;\alpha) = \Phi(t-s), \tag{2.27}$$

for $0 < \alpha < 1$ and $(t,s) \in S$, where Φ is given by (2.26).

In analogy with (2.12), the solution of (2.22) can be written

$$y(t) = R(t,0;\alpha)y_0 + \int_0^t R(t,s;\alpha)g(s)ds, \tag{2.28}$$

for $0 < \alpha < 1$ and $t \in [0,T]$. Equation (2.22) reduces to (2.2), if $\alpha = 0$, in which case Example 2.3 reduces to Example 2.2. Therefore (2.26) becomes

$$\Phi(t) = \mathcal{L}^{-1}\left\{\frac{s}{s^2 - as - b}\right\}, \tag{2.29}$$

where we assume, again, for ease of exposition, as in Example 2.2, that

$$s^2 - as - b = (s - s_1)(s - s_2),$$

where $s_1 \neq s_2$, and $s_i \in \Re$, $i = 1,2$. See equation (2.5).

Then, by partial fraction expansion and by elementary properties of the Laplace transform, we find

$$\Phi(t) = \frac{s_1 e^{s_1 t} - s_2 e^{s_2 t}}{s_1 - s_2}, \tag{2.30}$$

in which case the resolvent kernel R given by (2.27) becomes the previously defined resolvent kernel R, given by (2.11), and the solution given by (2.28) becomes (2.12).

We now generalize these results, by considering an example analogous to Example 2.1, in which we can give a theorem analogous to Theorem 2.2.

Example 2.4

$$y'(t) = a(t)y(t) + g(t) + \int_0^t (t-s)^{-\alpha}k(t,s)y(s)ds, \quad y(0) = y_0, \quad t \in [0,T], \tag{2.31}$$

where $0 < \alpha < 1$, *and* a, g *and* k, *are as in Example 2.1.*

Under these conditions (2.31) possesses a unique solution $y \in C^1([0,T])$, which we can explicitly write down in terms of the resolvent kernel. As with Example (2.1), we can define (formally) this resolvent kernel by the solution of the resolvent equation, or the adjoint resolvent equation.

Definition 2.3 *For $0 < \alpha < 1$, let the function $R : S \to \Re$ be formally defined by*

$$\begin{aligned} \frac{\partial R(t,s;\alpha)}{\partial s} &= -R(t,s;\alpha)a(s) - \int_s^t (\tau - s)^{-\alpha} R(t,\tau;\alpha)k(\tau,s)d\tau, \quad (t,s) \in S, \\ R(t,t;\alpha) &= 1, \quad t \geq s. \end{aligned} \tag{2.32}$$

This function R, given by (2.32) is called the *resolvent* kernel of (2.31), and (2.32) is called the *resolvent equation* associated with (2.31). The resolvent kernel can equivalently be defined as follows.

Definition 2.4 *For $0 < \alpha < 1$, let the function $R : S \to \Re$ be formally defined by*

$$\begin{aligned} \frac{\partial R(t,s;\alpha)}{\partial t} &= R(t,s;\alpha)a(t) + \int_s^t (t - \tau)^{-\alpha} k(t,\tau)R(\tau,s;\alpha)d\tau, \quad (t,s) \in S, \\ R(s,s;\alpha) &= 1, \quad s \leq t. \end{aligned} \tag{2.33}$$

Equation (2.33) is called the *adjoint resolvent equation* associated with (2.31).

Note that if we set $\alpha = 0$ in (2.31), it reduces to (2.1). Therefore, setting $\alpha = 0$ in (2.32) and (2.33) gives (2.13) and (2.20), respectively.

We now give the following representation of the solution of (2.31) in terms of this resolvent kernel R, the proof of which will establish the equivalency of (2.32) and (2.33).

Theorem 2.5 *(Brunner 1983) Let $a \in C([0,T])$, $k \in C(S)$, and assume that $R(t,s;\alpha)$ satisfies (2.32) or (equivalently) (2.33). Then, for any $g \in C([0,T])$,*

the initial-value problem (2.31) has a unique solution $y \in C^1([0,T])$, *given by*

$$y(t) = R(t,0;\alpha)y_0 + \int_0^t R(t,s;\alpha)g(s)ds, \qquad t \in [0,T]. \tag{2.34}$$

Clearly, the proof of Theorem 2.2 can be adapted in a straightforward way to supply the following result. However, we choose a different proof, one that explicitly shows that (2.32) and (2.33) define the same function R. As in Example 2.3, we rewrite (2.31) as the integral equation (2.35), given below, where the kernel Q, given by (2.36) is continuous (hence bounded) on S, for all $0 < \alpha < 1$.

$$y(t) = q(t) + \int_0^t Q(t,s;\alpha)y(s)ds, \tag{2.35}$$

for $t \in [0,T]$, where

$$Q(t,s;\alpha) := a(s) + \int_s^t (\tau - s)^{-\alpha} k(\tau,s)d\tau, \tag{2.36}$$

for $(t,s) \in S$ and $0 < \alpha < 1$, and

$$q(t) := y_0 + \int_0^t g(s)ds, \quad t \in [0,T]. \tag{2.37}$$

We also need the classical results for second-kind VIEs with continuous kernels and we collect these results in the following Lemma 2.2. See Brunner and van der Houwen (1986) and Brunner (1986b).

Lemma 2.2 *(Brunner 1986b) Consider the Volterra integral equation,*

$$y(t) = g(t) + \int_0^t k(t,s)y(s)ds, \qquad t \in [0,T], \tag{2.38}$$

where $g : [0,T] \to \Re$, $k : S \to \Re$ *and* $k \in C(S)$.

- *If $g \in C([0,T])$, then (2.38) has a unique solution $y \in C([0,T])$ given by the expression*

$$y(t) = g(t) - \int_0^t R(t,s)g(s)ds, \qquad t \in [0,T], \tag{2.39}$$

where the resolvent kernel $R : S \to \Re$ satisfies (equivalently) the resolvent equation:

$$R(t,s) = -k(t,s) + \int_s^t R(t,\tau)k(\tau,s)d\tau, \qquad (t,s) \in S, \tag{2.40}$$

or the adjoint resolvent equation

$$R(t,s) = -k(t,s) + \int_s^t k(t,\tau)R(\tau,s)d\tau, \qquad (t,s) \in S. \tag{2.41}$$

Furthermore, for $n \in \mathcal{N}_0$, if $k \in C^n(S)$, then $R \in C^n(S)$.

- *If $g \in C^1([0,T])$, then (2.38) has a unique solution $y \in C([0,T])$ given by the expression*

$$y(t) = U(t,0)g(0) + \int_0^t U(t,s)g'(s)ds, \qquad t \in [0,T], \tag{2.42}$$

where $U(t,s)$ is the unique continuous solution of

$$U(t,s) = 1 + \int_s^t k(t,\tau)U(\tau,s)d\tau, \qquad (t,s) \in S, \tag{2.43}$$

and

$$\frac{\partial U(t,s)}{\partial s} = R(t,s), \qquad (t,s) \in S. \tag{2.44}$$

Proof of Theorem 2.5:

Using Lemma 2.2, let R_Q be the resolvent kernel associated with the kernel Q. Note that it is continuous, hence bounded on S for all $0 < \alpha < 1$, and satisfies the resolvent equation (2.40), which for us is given by

$$R_Q(t,s;\alpha) = -Q(t,s;\alpha) + \int_s^t R_Q(t,\tau;\alpha)Q(\tau,s;\alpha)d\tau \quad (t,s) \in S, \tag{2.45}$$

where $0 < \alpha < 1$. Also, the (unique) continuous solution of (2.35), and therefore of (2.31), can be written

$$y(t) = q(t) - \int_0^t R_Q(t,s;\alpha)q(s)ds, \quad t \in [0,T], \tag{2.46}$$

$0 < \alpha < 1$, see (2.39).

Define

$$R(t,s;\alpha) := 1 - \int_s^t R_Q(t,\tau;\alpha)d\tau, \quad (t,s) \in S, \tag{2.47}$$

for all $0 < \alpha < 1$. Then, by using (2.37) and (2.47), and Dirichlet's Formula, the solution given by (2.46) above becomes (2.34), as desired. Now, since R_Q is continuous on S for all $0 < \alpha < 1$, it follows that

$$R(t,t;\alpha) = 1, \quad t \geq s, \quad 0 < \alpha < 1.$$

To show that R satisfies (2.32), we first note, from Leibniz's Rule that

$$\frac{\partial R(t,s;\alpha)}{\partial s} = R_Q(t,s;\alpha), \quad (t,s) \in S \quad 0 < \alpha < 1.$$

Then using (2.36) and since R_Q satisfies (2.45), we find

$$\begin{aligned} \frac{\partial R(t,s;\alpha)}{\partial s} &= -Q(t,s;\alpha) + \int_s^t R_Q(t,\tau;\alpha)Q(\tau,s;\alpha)d\tau \quad (t,s) \in S \\ &= -\left\{1 - \int_s^t R_Q(t,\tau;\alpha)d\tau\right\} a(s) \\ &- \int_s^t \left\{1 - \int_\tau^t R_Q(t,\xi;\alpha)d\xi\right\} (\tau - s)^{-\alpha}k(\tau,s)d\tau, \end{aligned} \tag{2.48}$$

where $(t,s) \in S$ and $0 < \alpha < 1$. Hint: To get one of the terms, you will need to swap variables τ and ξ, and then perform an interchange of variables by applying Dirichlet's formula. If we apply (2.47) to (2.48), it becomes (2.32), as desired.

To show that (2.33) also defines the same function R, we note that from (2.37), $q \in C^1[0,T]$ and we can use the second part of Lemma 2.2. Using (2.42), the (unique) continuous solution of (2.35), and therefore (2.31) can be written

$$y(t) = U(t,0;\alpha)q(0) + \int_0^t U(t,s;\alpha)q'(s)ds, \quad t \in [0,T], \tag{2.49}$$

$0 < \alpha < 1$, where U is the unique continuous solution of

$$U(t,s;\alpha) := 1 + \int_s^t Q(t,\tau;\alpha)U(\tau,s;\alpha)d\tau, \quad (t,s) \in S, \tag{2.50}$$

for all $0 < \alpha < 1$. However, from (2.37), it is clear that

$$q(0) = y_0, \;\; q'(s) = g(s),$$

for $(t,s) \in S$. Therefore this function U gives the solution (2.34), as well. Since Q is continuous on S for all $0 < \alpha < 1$, it follows that

$$U(s,s;\alpha) = 1, \quad s \le t, \;\; 0 < \alpha < 1.$$

Apply Leibniz's Rule to (2.50), noting that

$$Q(t,t;\alpha) = a(t), \;\; \frac{\partial Q(t,\tau;\alpha)}{\partial t} = (t-\tau)^{-\alpha}k(t,\tau),$$

for $0 \le \tau < t \le T$. We determine

$$\begin{aligned} \frac{\partial U(t,s;\alpha)}{\partial t} &= U(t,s;\alpha)a(t) + \int_s^t (t-\tau)^{-\alpha}k(t,\tau)U(\tau,s;\alpha)d\tau, \quad (t,s) \in S, \\ U(s,s;\alpha) &= 1, \quad \text{for} \quad s \le t, \end{aligned}$$

and $0 < \alpha < 1$. The equivalence of (2.32) and (2.33), then follows from the uniqueness of the solution y to (2.31). □

There is a price to pay for the weak singularity in (2.31), as the following lemma and theorem shows.

Lemma 2.3 *(Brunner and van der Houwen 1986) If $a \in C^n([0,T])$ and $k \in C^n(S)$, for some $n \in \mathcal{N}_0$, and $0 < \alpha < 1$, with k not vanishing identically on S, then the resolvent kernel R associated with (2.31) and given by (2.32) or (equivalently) (2.33), is an element of the space $C^1(S) \cap C^{n+1}(\hat{S})$, where*

$$\hat{S} := \{(t,s) \in \Re^2 : 0 \leq s < t \leq T\}.$$

For more details, we refer the reader to Brunner (1985), where a complete characterization of the kernel R is given for a VIE with weakly singular kernel. As pointed out in the paper, this characterization can be easily extended to the VIDE with weakly singular kernel, given by (2.31), by rewriting it as an integral equation.

We now give the following theorem from Brunner and van der Houwen (1986); compare with Theorem 2.3.

Theorem 2.6 *(Brunner and van der Houwen 1986) Let $a \in C^n([0,T])$ and $k \in C^n(S)$, for some $n \in \mathcal{N}_0$, and $0 < \alpha < 1$, Then, provided $g \in C^\nu([0,T])$, with $\nu \geq n$, the unique solution y of the initial-value problem (2.31) satisfies:*

- $y \in C^1([0,T]) \cap C^{n+1}((0,T])$,

- $|y''(t)| \leq \Psi t^{-\alpha}$, *for some positive constant* Ψ, *"near"* $t = 0$.

Then although Theorem 2.5 guarantees that the problem given by (2.31) has a unique solution $y \in C^1([0,T])$, its second order derivative is unbounded at $t = 0$. See Brunner (1983).

2.2 Continuous-Time Iteration WR Methods

We consider the nonlinear VIDE given by (1.19), and recall the following definitions from Chapter 1, Sections 1.3 and 1.4.

Choose a function $G_f : [0,T] \times \Re^m \times \Re^m \to \Re^m$, for the function f such that

$$G_f(t,u,u) = f(t,u), \tag{2.51}$$

for all $t \in [0,T]$ and for all $u \in \Re^m$, and a function $G_k : S \times \Re^m \times \Re^m \to \Re^m$, for the kernel k, so that:

$$G_k(t,s,u,u) = k(t,s,u), \tag{2.52}$$

for all $(t,s) \in S$ and for all $u \in \Re^m$.

We compute a sequence $y^1(t), y^2(t), \ldots, y^q(t)$, of solutions of the equations

$$\begin{aligned} \frac{d}{dt}y^q(t) &= G_f(t, y^{q-1}, y^q) + \int_0^t G_k(t,s,y^{q-1}(s), y^q(s))ds \\ y^q(0) &= y_0, \end{aligned} \tag{2.53}$$

where $y^0(t)$ is arbitrary and $q = 1,2,\ldots$, $t \in [0,T]$, which converges to the solution $y(t)$ of (1.19) as $q \to \infty$.

It is typical to choose the initial solution to be the constant function

$$y^0(t) = y_0, \qquad t \in [0,T].$$

We recall that these solutions are called *waveforms*, and the functions G_f, G_k are called *splitting* functions. We may use any of the iteration schemes mentioned earlier (Sections 1.3 and 1.4, respectively) for G_f and G_k, independently. This is one of the major differences between WR methods applied to VIDEs and those applied to ODEs. In Section 2.3.3 we will give an illustration of this; see Example 2.8. Clearly, the methods employing Picard and Gauss-Jacobi iteration are fully parallel. The resulting methods given by (2.53) are called *continuous-time iteration* WR methods, since they do not involve *yet* the discretization of the time interval and the application of numerical methods to solve the resulting VIDE.

Remark:

In some applications, see Brunner (1989), VIDEs occur in the following *nonstandard* form:

$$y'(t) = f(t,y) + \int_0^t k(t,s,y(t),y(s))ds, \quad y(0) = y_0, \quad t \in [0,T], \tag{2.54}$$

with assumptions analogous to those for (1.19).

Clearly, the splitting function, G_f, for f is defined as before. Define a function $G_k : S \times \Re^m \times \Re^m \times \Re^m \times \Re^m \to \Re^m$, satisfying:

$$G_k(t,s,u,u,\hat{u},\hat{u}) = k(t,s,u,\hat{u}),$$

for all $(t,s) \in S$ and all $u, \hat{u} \in \Re^m$. For illustration, we give two iteration schemes:

- **Picard Iteration**

$$G_k(t,s,u,v,\hat{u},\hat{v}) = k(t,s,u,\hat{u}).$$

- **Gauss-Jacobi Iteration**

$$\begin{aligned} {}^{i}G_k(t,s,u,v,\hat{u},\hat{v}) &= k_i(t,s,u_1,\ldots,u_{i-1},v_i,u_{i+1},\ldots,u_m, \\ &\qquad \hat{u}_1,\ldots,\hat{u}_{i-1},\hat{v}_i,\hat{u}_{i+1},\ldots,\hat{u}_m), \end{aligned}$$

for $i = 1,2,\ldots,m$.

The analogue of equation (2.53) is,

$$\left.\begin{array}{lcl} \frac{d}{dt}y^q(t) & = & G_f(t,y^{q-1},y^q) + \int_0^t G_k(t,s,y^{q-1}(t),y^q(t),y^{q-1}(s),y^q(s))ds \\ y^q(0) & = & y_0, \end{array}\right\} \quad (2.55)$$

where $y^0(t)$ is arbitrary and $q = 1,2,\ldots$, $t \in [0,T]$. But in the following, we consider only VIDEs of the form (1.19). $\diamond$

2.2.1 VIDEs with Regular Kernels

Consider the continuous-time iteration WR method (2.53) for the solution of the integro-differential equation (1.19), with regular kernel; see Theorem 2.1. The following Theorem 2.7 gives convergence results for (2.53), but in order for us to state this theorem, we recall the usual maximum norm. Given a continuous vector valued function, y, defined on [0,T]

$$\|y\|_T := \max_{t\in[0,T]} \|y(t)\|,$$

for the vector norm $\|\cdot\|$ on $\Re^m$.

We assume that the norm $\|\cdot\|$ is generated by an inner product $\langle\cdot,\cdot\rangle$ on $\Re^m \times \Re^m$. In most cases, however, this inner product is $\langle u, v\rangle_2 := u^T v,\ u, v \in \Re^m$, so the norm generated is the usual l_2-norm defined by $\|u\|_2 = \sqrt{u^T u} = \sqrt{\sum_{i=1}^m u_i^2}$.

The proof of the theorem to follow was adapted from a similar result for ODEs given in Burrage (1995). See Theorem 2.8.

Theorem 2.7 *Consider the integro-differential equation (1.19) and assume that the continuous splitting functions G_f and G_k satisfy (2.51) and (2.52), respectively, and assume the following uniform Lipschitz conditions in u and v:*

$$\begin{array}{rcl} \|G_f(t,u,v) - G_f(t,\tilde{u},v)\| & \leq & L_{f,1}\|u-\tilde{u}\|, \\ \|G_f(t,u,v) - G_f(t,u,\tilde{v})\| & \leq & L_{f,2}\|v-\tilde{v}\|, \end{array} \tag{2.56}$$

for all $u, v, \tilde{u}, \tilde{v}$ in $\Re^m$ and t in $[0,T]$, and

$$\|G_k(t,s,u,v) - G_k(t,s,\tilde{u},\tilde{v})\| \leq L_{k,1}\|u-\tilde{u}\| + L_{k,2}\|v-\tilde{v}\|, \tag{2.57}$$

for all $u, v, \tilde{u}, \tilde{v}$ in $\Re^m$ and for all $(t,s) \in S$. Assume that the initial solution $y^0(t) \in C([0,T])$. Then the resulting WR method (2.53) converges uniformly in $[0,T]$, for all finite $T > 0$.

If $L_{f,2} = L_{k,2} = 0$, then for $q = 1, 2, \ldots,$

$$\|y^q - y\|_T \leq \frac{T^q}{q!}(L_{f,1} + L_{k,1}T)^q \cdot \|y^0 - y\|_T, \tag{2.58}$$

otherwise, if $L_{k,2}$ is not zero, then

$$\|y^q - y\|_T \leq \frac{T^q}{q!}(L_{f,1} + L_{k,1}T)^q \cdot (e^{\lambda_1 T} + 1)^q \|y^0 - y\|_T, \tag{2.59}$$

where

$$\lambda_1 = \frac{L_{f,2} + \sqrt{4L_{k,2} + L_{f,2}^2}}{2}. \tag{2.60}$$

Remark: The case $L_{k,2} = 0$ and $L_{f,2} \neq 0$ is the well known ODE case discussed, for example in Burrage (1995). $\diamond$

Proof:

We will need the following lemma.

Lemma 2.4 *Consider the linear Volterra integro-differential inequality,*

$$y'(t) \leq a(t)y(t) + g(t) + \int_0^t k(t,s)y(s)ds, \qquad t \in [0,T], \tag{2.61}$$

where $y \in C^1([0,T])$, $a, g \in C([0,T])$ *and* $k \in C(S)$. *Also,* a *and* k *are nonnegative in* $[0,T]$ *and* S, *respectively. Suppose that* r *is the solution of*

$$r'(t) = a(t)r(t) + g(t) + \int_0^t k(t,s)r(s)ds, \qquad r(0) = y(0), \qquad t \in [0,T]. \tag{2.62}$$

Then

$$y(t) \leq r(t), \qquad t \in [0,T].$$

Proof:

The integro-differential inequality (2.61) is equivalent to the integral inequality

$$y(t) \leq q(t) + \int_0^t Q(t,s)y(s)ds, \quad t \in [0,T],$$

where

$$Q(t,s) := a(s) + \int_s^t k(\tau,s)d\tau, \quad (t,s) \in S,$$

and

$$q(t) := y(0) + \int_0^t g(s)ds, \quad t \in [0,T].$$

Since y, q and Q are continuous on their respective domains, and Q is nonnegative, we can apply Theorem 2.1 in Bainov and Simeonov (1992) to conclude that

$$y(t) \leq r(t), \quad t \in [0,T],$$

where r is the solution of

$$r(t) = q(t) + \int_0^t Q(t,s)r(s)ds, \quad t \in [0,T].$$

By Theorem 2.2, the proof is complete. □

Proof of Theorem 2.7:

For $q = 0, 1, \ldots,$ and $t \in [0,T]$, define $e^q(t) := y^q(t) - y(t)$. Then for $q = 1, 2, \ldots,$ and $t \in [0,T]$ by the definition of the splitting functions G_f, G_k,

$$\begin{aligned} \frac{de^q(t)}{dt} &= \frac{dy^q(t)}{dt} - \frac{dy(t)}{dt} \\ &= G_f(t, y^{q-1}, y^q) - G_f(t,y,y) \\ &+ \int_0^t \left\{ G_k(t,s,y^{q-1}(s), y^q(s)) - G_k(t,s,y(s),y(s)) \right\} ds, \end{aligned}$$

using (1.19), (2.51), (2.52) and (2.53). By linearity of the inner product,

$$\begin{aligned} \langle \frac{de^q(t)}{dt}, e^q(t) \rangle &= \langle G_f(t, y^{q-1}, y^q) - G_f(t, y^{q-1}, y), e^q(t) \rangle \\ &+ \langle G_f(t, y^{q-1}, y) - G_f(t,y,y), e^q(t) \rangle \\ &+ \langle \int_0^t \left\{ G_k(t,s,y^{q-1}(s), y^q(s)) - G_k(t,s,y(s),y(s)) \right\} ds, e^q(t) \rangle. \end{aligned}$$

Using the Cauchy-Schwarz inequality, see Lancaster and Tismenetsky (1985), and the Lipschitz condition (2.56) for G_f (in both components) we obtain

$$\begin{aligned} \langle \frac{de^q(t)}{dt}, e^q(t) \rangle &\leq L_{f,2} \|e^q(t)\|^2 + L_{f,1} \|e^{q-1}(t)\| \cdot \|e^q(t)\| \\ &+ \| \int_0^t \left\{ G_k(t,s,y^{q-1}(s), y^q(s)) - G_k(t,s,y(s),y(s)) \right\} ds \| \cdot \|e^q(t)\|. \end{aligned}$$

Then the Lipschitz condition (2.57) for G_k leads to

$$\begin{aligned}\langle \frac{de^q(t)}{dt}, e^q(t)\rangle &\leq L_{f,2}\|e^q(t)\|^2 + L_{f,1}\|e^{q-1}(t)\| \cdot \|e^q(t)\| \\ &+ \int_0^t \left\{ L_{k,1}\|e^{q-1}(s)\| + L_{k,2}\|e^q(s)\| \right\} ds \cdot \|e^q(t)\|.\end{aligned}$$

When is $\|e^q(t)\|$ not differentiable? Only at points t where $e^q(t) = 0$, in which case (2.58) and (2.59) are satisfied. Therefore we assume that $\|e^q(t)\| \neq 0$.

From the product rule for derivatives for the (real) inner product,

$$\langle \frac{de^q(t)}{dt}, e^q(t)\rangle = \frac{1}{2}\frac{d}{dt}\|e^q(t)\|^2 = \|e^q(t)\|\frac{d}{dt}\|e^q(t)\|,$$

it follows that

$$\frac{d}{dt}\|e^q(t)\| \leq L_{f,2}\|e^q(t)\| + L_{f,1}\|e^{q-1}(t)\| + \int_0^t \left\{ L_{k,2}\|e^q(s)\| + L_{k,1}\|e^{q-1}(s)\| \right\} ds.$$

Next define the positive function $v^q(t) := \|e^q(t)\|$, and consider the solution of the integro-differential inequality

$$\frac{d}{dt}v^q(t) \leq L_{f,2}v^q(t) + \int_0^t L_{k,2}v^q(s)ds + \left\{ L_{f,1}v^{q-1}(t) + \int_0^t L_{k,1}v^{q-1}(s)ds \right\}, \quad (2.63)$$

subject to the initial condition $v^q(0) = 0$. This integro-differential inequality can be solved by considering the corresponding integro-differential equation,

$$\frac{d}{dt}v^q(t) = L_{f,2}v^q(t) + \int_0^t L_{k,2}v^q(s)ds + \left\{ L_{f,1}v^{q-1}(t) + \int_0^t L_{k,1}v^{q-1}(s)ds \right\}, \quad (2.64)$$

where $v^q(0) = 0$, see Lemma 2.4. Note that, for simplicity of notation, I use the same symbol v^q for the solution of (2.64). It is here that we need the continuity of $y^0(t)$ to guarantee the continuity of $v^0(t)$ needed in Lemma 2.4. Note that v^q is piecewise C^1 on $[0, T]$, for $q = 1, 2, \ldots$; see the comment at the top of this page.

Equation (2.64) is Example 2.1 which has a solution given by (2.14), where the resolvent kernel R satisfies (2.13).

However, we can write down this resolvent kernel R, explicitly; see equation (2.11).

Let λ_1, λ_2 be the solutions of the quadratic

$$\lambda^2 - L_{f,2}\lambda - L_{k,2} = 0,$$

that is

$$\lambda_i = \frac{L_{f,2} \pm \sqrt{4L_{k,2} + L_{f,2}^2}}{2}, \tag{2.65}$$

$i = 1, 2$, and note that these roots are always real and distinct, except for the case where $L_{f,2} = L_{k,2} = 0$.

Assume $L_{k,2}$ is not zero.

Then (2.64) can be solved to give

$$v^q(t) = \int_0^t R(t,s) \left[L_{f,1} v^{q-1}(s) + \int_0^s L_{k,1} v^{q-1}(\tau) d\tau \right] ds,$$

where the resolvent kernel $R(t,s)$ is given by

$$R(t,s) = \frac{\lambda_1 e^{\lambda_1(t-s)} - \lambda_2 e^{\lambda_2(t-s)}}{\lambda_1 - \lambda_2}, \quad (t,s) \in S. \tag{2.66}$$

Therefore, returning to the inequality (2.63),

$$v^q(t) \leq \int_0^t R(t,s) \left[L_{f,1} v^{q-1}(s) + \int_0^s L_{k,1} v^{q-1}(\tau) d\tau \right] ds. \tag{2.67}$$

Let us define

$$\|R\|_t := \max_{0 \leq s \leq t} |R(t,s)|,$$

and

$$\|v^0\|_t := \max_{0\le s\le t} v^0(s),$$

for $t \in [0,T]$. By inspection of (2.65), it is easy to see that λ_1 is always positive and λ_2 is always negative and by the triangle inequality, $\lambda_1 \ge |\lambda_2|$. Then,

$$|R(t,s)| \le \left|\frac{\lambda_1 e^{\lambda_1(t-s)}}{\lambda_1-\lambda_2}\right| + \left|\frac{\lambda_2 e^{\lambda_2(t-s)}}{\lambda_1-\lambda_2}\right|, \quad (t,s)\in S.$$

Since $\lambda_2 < 0$, and $|\lambda_2| \le \lambda_1$,

$$|\lambda_2 e^{\lambda_2(t-s)}| \le \lambda_1, \quad (t,s)\in S.$$

Then

$$|R(t,s)| \le \frac{\lambda_1}{\lambda_1-\lambda_2}\left(e^{\lambda_1(t-s)}+1\right), \quad (t,s)\in S.$$

However, since $L_{f,2} \le \sqrt{4L_{k,2}+L_{f,2}^2}$,

$$\frac{\lambda_1}{\lambda_1-\lambda_2} \le 1,$$

and it follows that

$$\|R\|_t \le \left(e^{\lambda_1 t}+1\right), \quad t\in[0,T].$$

Then for $q=1$ and using $\|R\|_t$ and $\|v^0\|_t$, (2.67) becomes

$$v^1(t) \le \int_0^t \left[L_{f,1} + \int_0^s L_{k,1}\,d\tau\right] ds \quad (\|R\|_t \cdot \|v^0\|_t), \tag{2.68}$$

which can be integrated in closed form

$$v^1(t) \le \left(L_{f,1}t + L_{k,1}t^2/2\right)\cdot(e^{\lambda_1 t}+1)\|v^0\|_t, \quad t\in[0,T]. \tag{2.69}$$

By induction,

$$v^q(t) \le \sum_{j=0}^{q}\left(\frac{q!}{j!(q-j)!}\right)\frac{L_{f,1}^{q-j}L_{k,1}^{j}t^{q+j}}{(q+j)!}\cdot(e^{\lambda_1 t}+1)^q\|v^0\|_t, \tag{2.70}$$

for all $t \in [0,T]$ and $q = 1,2,\ldots$, which we rewrite as

$$v^q(t) \le t^q \sum_{j=0}^{q} \left(\frac{q!}{j!(q-j)!} \right) \frac{L_{f,1}^{q-j}(L_{k,1}t)^j}{(q+j)!} \cdot (e^{\lambda_1 t}+1)^q \|v^0\|_t. \tag{2.71}$$

Since $\frac{1}{(q+j)!} \le \frac{1}{q!}$, for $j = 0,1,\ldots,q$,

$$v^q(t) \le \frac{t^q}{q!} \sum_{j=0}^{q} \left(\frac{q!}{j!(q-j)!} \right) L_{f,1}^{q-j}(L_{k,1}t)^j \cdot (e^{\lambda_1 t}+1)^q \|v^0\|_t,$$

and by using the binomial theorem

$$v^q(t) \le \frac{t^q}{q!} (L_{f,1} + L_{k,1}t)^q \cdot (e^{\lambda_1 t}+1)^q \|v^0\|_t, \quad \text{for all } t \in [0,T]. \tag{2.72}$$

It is clear that the iteration scheme converges uniformly, as $q \to \infty$, for all finite $T > 0$, and

$$\|y^q - y\|_T \le \frac{T^q}{q!} (L_{f,1} + L_{k,1}T)^q \cdot (e^{\lambda_1 T}+1)^q \|y^0 - y\|_T, \tag{2.73}$$

$q = 1,2,\ldots$, where $T > 0$ is finite.

Assume $L_{f,2} = L_{k,2} = 0$.

Then (2.64) becomes

$$\frac{d}{dt} v^q(t) = L_{f,1} v^{q-1}(t) + \int_0^t L_{k,1} v^{q-1}(s) ds, \tag{2.74}$$

with $v^q(0) = 0$. This can be solved by direct integration, to become

$$v^q(t) = L_{f,1} \int_0^t v^{q-1}(s) ds + L_{k,1} \int_0^t (t-\tau) v^{q-1}(\tau) d\tau.$$

Again, returning to the inequality (2.63),

$$v^q(t) \leq L_{f,1} \int_0^t v^{q-1}(s)ds + L_{k,1} \int_0^t (t-\tau)v^{q-1}(\tau)d\tau. \tag{2.75}$$

This equation is similar to (2.67), so similar analysis gives

$$v^q(t) \leq \sum_{j=0}^{q} \left(\frac{q!}{j!(q-j)!} \right) \frac{L_{f,1}^{q-j} L_{k,1}^{j} t^{q+j}}{(q+j)!} \cdot \|v^0\|_t, \tag{2.76}$$

for all $t \in [0,T]$ and $q = 1, 2, \ldots$; compare with (2.70). Continuing with the arguments that led to (2.72), we establish

$$v^q(t) \leq \frac{t^q}{q!} (L_{f,1} + L_{k,1}t)^q \cdot \|v^0\|_t, \quad t \in [0,T]. \tag{2.77}$$

Again, it is clear that the iteration scheme converges uniformly, as $q \to \infty$ for all finite $T > 0$, and

$$\|y^q - y\|_T \leq \frac{T^q}{q!} (L_{f,1} + L_{k,1}T)^q \cdot \|y^0 - y\|_T, \tag{2.78}$$

$q = 1, 2, \ldots$, where T is finite and $T > 0$. □

To illustrate this result we consider the following (nonhomogeneous) linear two-dimensional problem.

Example 2.5 *Consider the following Volterra integro-differential system*

$$\begin{pmatrix} y_1'(t) \\ y_2'(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} + \int_0^t \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} y_1(s) \\ y_2(s) \end{pmatrix} ds + \begin{pmatrix} 1 - e^t \\ 1 - e^t \end{pmatrix},$$

where $[y_{1,0}\ y_{2,0}]^T = [1\ 1]^T$ *are the given initial conditions. Then the solution is*

$$\begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} = \begin{pmatrix} e^t \\ e^t \end{pmatrix}.$$

We compute a sequence of solutions using (2.53), where we take the initial solution

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

for $t \in [0, T]$.

For the **Gauss-Jacobi** iteration scheme we obtain the following iterates:

$$\begin{aligned}
y_j^1(t) &= \frac{t^2}{2!} + 2t + 2 - e^t \\
&= \{1 + t\} - \frac{t^3}{3!} - \frac{t^4}{4!} - \cdots \\
y_j^2(t) &= \frac{t^4}{4!} + \frac{t^3}{2} + 2t^2 + 4t + 4 - 3e^t \\
&= \left\{1 + t + \frac{t^2}{2!}\right\} - \frac{2t^4}{4!} - \frac{3t^5}{5!} - \frac{3t^6}{6!} \cdots \\
y_j^3(t) &= \left\{1 + t + \frac{t^2}{2!} + \frac{t^3}{3!}\right\} - \frac{3t^5}{5!} - \frac{6t^6}{6!} - \frac{7t^7}{7!} \cdots \\
&\vdots \\
y_j^q(t) &= \left\{1 + t + \frac{t^2}{2!} + \cdots + \frac{t^q}{q!}\right\} - \frac{qt^{q+2}}{(q+2)!} - \cdots
\end{aligned} \tag{2.79}$$

$j = 1, 2; \quad q = 1, 2, \ldots.$

For the **Gauss-Seidel** iteration scheme we obtain the following iterates:

$$\begin{aligned}
y_1^1(t) &= \{1 + t\} - \frac{t^3}{3!} - \frac{t^4}{4!} - \cdots \\
y_2^1(t) &= \left\{1 + t + \frac{t^2}{2!}\right\} - \frac{2t^4}{4!} - \frac{3t^5}{5!} - \frac{3t^6}{6!} \cdots \\
y_1^2(t) &= \left\{1 + t + \frac{t^2}{2!} + \frac{t^3}{3!}\right\} - \frac{3t^5}{5!} - \frac{6t^6}{6!} - \frac{7t^7}{7!} \cdots
\end{aligned}$$

$$\vdots$$

$$y_1^q(t) = \left\{1 + t + \frac{t^2}{2!} + \cdots + \frac{t^{2q-1}}{(2q-1)!}\right\} - \frac{(2q-1)t^{2q+1}}{(2q+1)!} - \cdots \tag{2.80}$$

$$y_2^q(t) = \left\{1 + t + \frac{t^2}{2!} + \cdots + \frac{t^{2q}}{(2q)!}\right\} - \frac{2qt^{2q+2}}{(2q+2)!} - \cdots \tag{2.81}$$

$q = 1, 2, \ldots$.

Clearly, for both iteration schemes,

$$\lim_{q \to \infty} y_j^q(t) = y_j(t) = e^t,$$

for $j = 1, 2; \quad t \in [0, T]$.

Note that we increase the order of accuracy by one for each iteration q taken. For this particular test equation, the Gauss-Seidel iteration scheme converges "approximately" twice as fast as the Gauss-Jacobi iteration scheme; compare (2.80) and (2.81), to (2.79), respectively. In Burrage (1995), he comments that for certain linear systems of equations, Gauss-Seidel iteration will converge approximately twice as fast as Gauss-Jacobi iterations.

Let us compare Theorem 2.7 with the analogous theorems for ODEs and VIEs; recall Chapter 1, Sections 1.3.1 and 1.4.1. We consider ODEs first.

In Burrage (1995), we acquire the following result for a nonlinear autonomous ODE

$$y'(t) = f(y), \qquad y(0) = y_0, \tag{2.82}$$

for $t \in [0, T]$, $y \in \Re^m$ and $f : \Re^m \to \Re^m$.

Theorem 2.8 *(Burrage 1995) Consider the autonomous differential equation (2.82) and assume that the splitting function G_f given by*

$$G_f(u, u) = f(u),$$

for all $u \in \Re^m$, *and* $G_f : \Re^m \times \Re^m \to \Re^m$ *is continuous with respect to both components and satisfies uniform Lipschiz conditions in* u *and* v*:*

$$\begin{aligned} \|G_f(u,v) - G_f(\tilde{u},v)\| &\leq L_{f,1}\|u - \tilde{u}\|, \\ \|G_f(u,v) - G_f(u,\tilde{v})\| &\leq L_{f,2}\|v - \tilde{v}\|, \end{aligned}$$

for all $u, v, \tilde{u}, \tilde{v}$ *in* $\Re^m$. *Then the resulting WR method*

$$\frac{d}{dt}y^q(t) = G_f(y^{q-1}, y^q), \quad y^q(0) = y_0,$$

$q = 1, 2, \ldots$, $t \in [0, T]$, *converges uniformly in* $[0, T]$, *for all finite* $T > 0$, *and*

$$\|y^q - y\|_T \leq \frac{(L_{f,1}T)^q}{q!} e^{L_{f,2}T} \|y^0 - y\|_T. \tag{2.83}$$

See Jansen *et al.* (1994) for an illustration for Theorem 2.8, analogous to our Example 2.5 for Theorem 2.7.

We can use our Theorem 2.7 to develop a result for non-autonomous differential equations.

Corollary 2.1 *Consider the non-autonomous differential equation (1.1) and assume that the splitting function* G_f *given by (2.51) is continuous with respect to both components and satisfies uniform Lipschiz condition in* u *and* v*:*

$$\begin{aligned} \|G_f(t,u,v) - G_f(t,\tilde{u},v)\| &\leq L_{f,1}\|u - \tilde{u}\|, \\ \|G_f(t,u,v) - G_f(t,u,\tilde{v})\| &\leq L_{f,2}\|v - \tilde{v}\|, \end{aligned}$$

for all $u, v, \tilde{u}, \tilde{v}$ *in* $\Re^m$ *and* t *in* $[0, T]$. *Assume* $y^0(t) \in C([0, T])$. *Then the resulting WR method (1.3) converges uniformly in* $[0, T]$, *for all finite* $T > 0$. *If* $L_{f,2} = 0$, *then,*

$$\|y^q - y\|_T \leq \frac{(L_{f,1}T)^q}{q!} \cdot \|y^0 - y\|_T, \tag{2.84}$$

otherwise

$$\|y^q - y\|_T \leq \frac{(L_{f,1}T)^q}{q!} \cdot \left(e^{L_{f,2}T} + 1\right)^q \|y^0 - y\|_T. \tag{2.85}$$

Proof:

The proof follows by setting

$$L_{k,1} = L_{k,2} = 0,$$

in Theorem 2.7. □

Note that (2.84) is identical with (2.83) from Theorem 2.8 (upon setting $L_{f,2} = 0$). Equation (2.85) shows a significant departure from (2.83), but we still increase the order of accuracy by one for each iteration q.

For the sake of comparison, we briefly turn to VIEs where we have the following result.

Theorem 2.9 *(Crisci et al. 1996a) Consider (1.17) and assume that the splitting function G_k given by (2.52), satisfies a uniform Lipschiz condition in u and v:*

$$\|G_k(t,s,u,v) - G_k(t,s,\tilde{u},\tilde{v})\| \leq L_{k,1}\|u - \tilde{u}\| + L_{k,2}\|v - \tilde{v}\| \tag{2.86}$$

for all $u, v, \tilde{u}, \tilde{v}$ in $\Re^m$ and for all $(t,s) \in S$. Then the resulting WR method (1.21) converges in $[0, T_1]$, where

$$T_1 < \frac{1}{L_1 + L_2}.$$

For the proof, see Crisci *et al.* (1996a).

Note, that we do not get uniform convergence for all finite $T > 0$ for VIEs, as we do for VIDEs and ODEs, see Theorem 2.7 and Corollary 2.1, respectively. In this

sense, a Volterra integro-differential equation is more like a "perturbed" ordinary differential equation than a Volterra integral equation. However, if we assume a linear VIE we can extend this convergence to all finite $T > 0$, Crisci *et al.* (1996a). See Section 1.4.1.

2.2.2 VIDEs with Weakly Singular Kernels

To complete our analysis of continuous-time iteration WR methods, we return to the special nonlinear VIDE (2.21), with a weakly-singular kernel. We now give a theorem analogous to Theorem 2.7 for this problem. Our main concern in this theorem is to find out how the presence of the weak singularity affects the way the order changes with the number of iterations q taken. As before, we choose splitting functions G_f for the function f and G_k for k, given by (2.51) and (2.52), respectively. Again, the previous iteration schemes can be used.

We compute a sequence $y^1(t), y^2(t), \ldots, y^q(t)$, of solutions of the equations

$$\begin{aligned} \frac{d}{dt}y^q(t) &= G_f(t, y^{q-1}, y^q) + \int_0^t (t-s)^{-\alpha} G_k(t, s, y^{q-1}(s), y^q(s))ds \\ y^q(0) &= y_0, \end{aligned} \tag{2.87}$$

where $y^0(t)$ is arbitrary and $q = 1, 2, \ldots$, $t \in [0, T]$ and $0 < \alpha < 1$, which converges to the solution $y(t)$ of (2.21) as $q \to \infty$. It is typical to choose the initial solution to be the constant function equal to the given initial value, that is

$$y^0(t) = y_0, \qquad t \in [0, T].$$

We may use any of the iteration schemes mentioned earlier (see the beginning of Section 2.2) for G_f and any one for G_k, independently.

We recall that the resulting methods given by (2.87) are called *continuous-time iteration* WR methods, since they do not involve *yet* the discretization of the time interval and the application of numerical methods to solve the resulting VIDEs. Theorem 2.10 gives convergence results for these methods. Recall the definitions of the norm introduced in Section 2.2.1.

Theorem 2.10 *Consider the integro-differential equation with weakly singular kernel (2.21) and assume that it possesses a unique solution* $y \in C^1([0,T])$. *Assume that the splitting functions* G_f *and* G_k *given by (2.51) and (2.52), respectively, are continuous with respect to both components and satisfy uniform Lipschiz condition in* u *and* v*:*

$$\begin{array}{rcl} \|G_f(t,u,v) - G_f(t,\tilde{u},v)\| & \leq & L_{f,1}\|u-\tilde{u}\|, \\ \|G_f(t,u,v) - G_f(t,u,\tilde{v})\| & \leq & L_{f,2}\|v-\tilde{v}\|, \end{array} \tag{2.88}$$

for all $u, v, \tilde{u}, \tilde{v}$ *in* $\Re^m$ *and* $t \in [0,T]$, *and*

$$\|G_k(t,s,u,v) - G_k(t,s,\tilde{u},\tilde{v})\| \leq L_{k,1}\|u-\tilde{u}\| + L_{k,2}\|v-\tilde{v}\|, \tag{2.89}$$

for all $u, v, \tilde{u}, \tilde{v}$ *in* $\Re^m$ *and* $(t,s) \in S$. *Assume that the initial solution* $y^0 \in C([0,T])$. *Then the resulting WR method (2.87) converges uniformly in* $[0,T]$, *for all finite* $T > 0$.

For $q = 1, 2, \ldots,$

$$\|y^q - y\|_T \leq \frac{T^q}{q!}\left(L_{f,1} + L_{k,1}\Gamma(1-\alpha)T^{1-\alpha}\right)^q \|R\|_T^q \cdot \|y^0 - y\|_T, \tag{2.90}$$

where $0 \prec \alpha < 1$, *and*

$$\|R\|_T := \max_{0 \leq s \leq T} |R(T,s;\alpha)|,$$

where R *is given by (2.27) and (2.26).*

Proof:

The proof will be based on the following lemmas.

Lemma 2.5 *For $0 < \alpha < 1$ and $p > -1$,*

$$\int_0^t \int_0^s (s-\tau)^{-\alpha}\tau^p d\tau ds = \frac{\Gamma(1-\alpha)\Gamma(p+1)}{\Gamma(p-\alpha+3)} t^{p-\alpha+2}, \quad t > 0.$$

In addition, we will need to solve an integro-differential inequality in which the kernel has a weak singularity. Therefore, we give the following lemma which is a generalization of Lemma 2.4.

Lemma 2.6 *Consider the linear Volterra integro-differential inequality, with weakly singular kernel,*

$$y'(t) \leq a(t)y(t) + g(t) + \int_0^t (t-s)^{-\alpha} k(t,s)y(s)ds, \qquad t \in [0,T], \tag{2.91}$$

where $0 < \alpha < 1$, $y \in C^1([0,T])$, $a, g \in C([0,T])$ and $k \in C(S)$. Also, a and k are nonnegative in $[0,T]$ and S, respectively. Suppose that r is the solution of

$$r'(t) = a(t)r(t) + g(t) + \int_0^t (t-s)^{-\alpha} k(t,s)r(s)ds, \quad r(0) = y(0), \quad t \in [0,T]. \tag{2.92}$$

Then

$$y(t) \leq r(t), \qquad t \in [0,T].$$

Proof:

The integro-differential inequality (2.91) is equivalent to the integral inequality

$$y(t) \leq q(t) + \int_0^t Q(t,s;\alpha)y(s)ds, \quad t \in [0,T],$$

where

$$Q(t,s;\alpha) := a(s) + \int_s^t (\tau - s)^{-\alpha} k(\tau,s)d\tau, \quad (t,s) \in S,$$

$0 < \alpha < 1$, and

$$q(t) := y(0) + \int_0^t g(s)ds, \quad t \in [0,T].$$

Since y, q and Q are continuous on their respective domains, and Q is nonnegative, we can apply Theorem 2.1 in Bainov and Simeonov (1992) to conclude that

$$y(t) \leq r(t), \quad t \in [0,T],$$

where r is the solution of

$$r(t) = q(t) + \int_0^t Q(t,s;\alpha)r(s)ds, \quad t \in [0,T]. \tag{2.93}$$

By Theorem 2.5, the proof is complete. □

Proof of Theorem 2.10:

For $q = 0,1,\ldots,$ and $t \in [0,T]$, define $e^q(t) := y^q(t) - y(t)$. Then for $q = 1,2,\ldots,$ $0 < \alpha < 1$ and $t \in [0,T]$, by the definition of the splitting functions G_f, G_k,

$$\begin{aligned}
\frac{de^q(t)}{dt} &= \frac{dy^q(t)}{dt} - \frac{dy(t)}{dt} \\
&= G_f(t, y^{q-1}, y^q) - G_f(t,y,y) \\
&+ \int_0^t (t-s)^{-\alpha} \left\{ G_k(t,s,y^{q-1}(s), y^q(s)) - G_k(t,s,y(s),y(s)) \right\} ds,
\end{aligned}$$

using (2.21), (2.51), (2.52) and (2.87). By the linearity of the inner product,

$$\begin{aligned}\langle\frac{de^q(t)}{dt}, e^q(t)\rangle &= \langle G_f(t, y^{q-1}, y^q) - G_f(t, y^{q-1}, y), e^q(t)\rangle \\ &+ \langle G_f(t, y^{q-1}, y) - G_f(t, y, y), e^q(t)\rangle \\ &+ \langle \int_0^t (t-s)^{-\alpha} \\ &\quad \left\{G_k(t, s, y^{q-1}(s), y^q(s)) - G_k(t, s, y(s), y(s))\right\} ds, e^q(t)\rangle.\end{aligned}$$

Using the Cauchy-Schwarz inequality and the Lipschitz condition for G_f (in both components), given by (2.88),

$$\begin{aligned}\langle\frac{de^q(t)}{at}, e^q(t)\rangle &\leq L_{f,2}\|e^q(t)\|^2 + L_{f,1}\|e^{q-1}(t)\| \cdot \|e^q(t)\| \\ &+ \| \int_0^t (t-s)^{-\alpha} \left\{G_k(t, s, y^{q-1}(s), y^q(s)) - G_k(t, s, y(s), y(s))\right\} ds\| \\ &\cdot \|e^q(t)\|.\end{aligned}$$

Now using the Lipschitz condition (2.89) for G_k,

$$\begin{aligned}\langle\frac{de^q(t)}{dt}, e^q(t)\rangle &\leq L_{f,2}\|e^q(t)\|^2 + L_{f,1}\|e^{q-1}(t)\| \cdot \|e^q(t)\| \\ &+ \int_0^t (t-s)^{-\alpha} \left\{L_{k,1}\|e^{q-1}(s)\| + L_{k,2}\|e^q(s)\|\right\} ds \cdot \|e^q(t)\|.\end{aligned}$$

Without loss of generality, we assume $\|e^q(t)\| \neq 0$; see the comment on the top of page 70.

From the product rule for derivatives for the (real) inner product,

$$\langle\frac{de^q(t)}{dt}, e^q(t)\rangle = \frac{1}{2}\frac{d}{dt}\|e^q(t)\|^2 = \|e^q(t)\|\frac{d}{dt}\|e^q(t)\|.$$

$$\frac{d}{dt}\|e^q(t)\| \leq L_{f,2}\|e^q(t)\| + L_{f,1}\|e^{q-1}(t)\| + \int_0^t (t-s)^{-\alpha} \left\{L_{k,2}\|e^q(s)\| + L_{k,1}\|e^{q-1}(s)\|\right\} ds.$$

Next define the positive function $v^q(t) := \|e^q(t)\|$ and consider the solution of the integro-differential inequality

$$\begin{aligned} \frac{d}{dt}v^q(t) &\leq L_{f,2}v^q(t) + \int_0^t (t-s)^{-\alpha} L_{k,2}v^q(s)ds \\ &+ \left\{ L_{f,1}v^{q-1}(t) + \int_0^t (t-s)^{-\alpha} L_{k,1}v^{q-1}(s)ds \right\}. \end{aligned} \tag{2.94}$$

This integro-differential inequality can be solved by considering the corresponding integro-differential equation,

$$\begin{aligned} \frac{d}{dt}v^q(t) &= L_{f,2}v^q(t) + \int_0^t (t-s)^{-\alpha} L_{k,2}v^q(s)ds \\ &+ \left\{ L_{f,1}v^{q-1}(t) + \int_0^t (t-s)^{-\alpha} L_{k,1}v^{q-1}(s)ds \right\}, \end{aligned} \tag{2.95}$$

where $v^q(0) = 0$; see Lemma 2.6. Note that, for simplicity of notation, I use the same symbol v^q for the solution of (2.95). It is here that we need the continuity of $y^0(t)$ to guarantee the continuity of $v^0(t)$ needed in Lemma 2.6. Note that v^q is piecewise C^1 on $[0,T]$, for $q = 1, 2, \ldots$; see the comment on the top of page 70.

This is Example 2.3, whose solution is given by (2.28), where the resolvent kernel R is given by (2.27) and (2.26), where $(t,s) \in S$ and $0 < \alpha < 1$; see Theorem 2.4. Note that, by Lemma 2.3, this resolvent kernel is continuous, and therefore bounded on S.

Then (2.95) can be solved to give

$$v^q(t) = \int_0^t R(t,s;\alpha) \left[L_{f,1}v^{q-1}(s) + \int_0^s (s-\tau)^{-\alpha} L_{k,1}v^{q-1}(\tau)d\tau \right] ds, \tag{2.96}$$

where $0 < \alpha < 1$, $t \in [0,T]$ and $q = 1, 2, \ldots$.

Returning to the inequality (2.94)

$$v^q(t) \leq \int_0^t R(t,s;\alpha)\left[L_{f,1}v^{q-1}(s) + \int_0^s (s-\tau)^{-\alpha} L_{k,1} v^{q-1}(\tau)d\tau\right] ds, \qquad (2.97)$$

where $0 < \alpha < 1$, $t \in [0,T]$ and $q = 1, 2, \ldots$.

Let us define

$$\|R\|_t := \max_{0\leq s\leq t} |R(t,s;\alpha)|,$$

and

$$\|v^0\|_t := \max_{0\leq s\leq t} v^0(s),$$

for $t \in [0,T]$ and $0 < \alpha < 1$.

Then for $q = 1$ and using $\|R\|_t$ and $\|v^0\|_t$, (2.97) becomes

$$v^1(t) \leq \int_0^t \left[L_{f,1} + \int_0^s (s-\tau)^{-\alpha} L_{k,1} d\tau\right] ds \quad (\|R\|_t \cdot \|v^0\|_t), \qquad (2.98)$$

which can be integrated in closed form, using Lemma 2.5. Thus,

$$v^1(t) \leq \left(L_{f,1}t + \frac{L_{k,1}\Gamma(1-\alpha)}{\Gamma(3-\alpha)} t^{2-\alpha}\right) \cdot \quad (\|R\|_t \cdot \|v^0\|_t), \qquad (2.99)$$

for $t \in [0,T]$ and $0 < \alpha < 1$. Substitute this expression (2.99) into (2.97) with $q = 2$, using the recursion property of the Γ function to simplify your answer. We find

$$\begin{aligned} v^2(t) \quad \leq \quad & \left(\frac{L_{f,1}^2 t^2}{\Gamma(3)} + 2\frac{L_{f,1}L_{k,1}\Gamma(1-\alpha)t^{3-\alpha}}{\Gamma(4-\alpha)} + \frac{L_{k,1}^2\Gamma^2(1-\alpha)t^{4-2\alpha}}{\Gamma(5-2\alpha)}\right) \qquad (2.100) \\ & \cdot \quad (\|R\|_t^2 \cdot \|v^0\|_t), \end{aligned}$$

for $t \in [0,T]$ and $0 < \alpha < 1$. Similarly, for $q = 3$,

$$v^3(t) \leq \left(\frac{L_{f,1}^3 t^3}{\Gamma(4)} + 3\frac{L_{f,1}^2 L_{k,1}\Gamma(1-\alpha)t^{4-\alpha}}{\Gamma(5-\alpha)}\right. \tag{2.101}$$
$$\left. + \; 3\frac{L_{f,1}L_{k,1}^2\Gamma^2(1-\alpha)t^{5-2\alpha}}{\Gamma(6-2\alpha)} + \frac{L_{k,1}^3\Gamma^3(1-\alpha)t^{6-3\alpha}}{\Gamma(7-3\alpha)}\right) \cdot (\|R\|_t^3 \cdot \|v^0\|_t),$$

for $t \in [0,T]$ and $0 < \alpha < 1$. Note the binomial coefficient in front of each term.

By induction, we establish

$$v^q(t) \leq \sum_{j=0}^{q}\left(\frac{q!}{j!(q-j)!}\right)\frac{L_{f,1}^{q-j}L_{k,1}^j\Gamma^j(1-\alpha)t^{(q+j)-j\alpha}}{\Gamma((q+j+1)-j\alpha)}(\|R\|_t^q \cdot \|v^0\|_t), \tag{2.102}$$

for all $t \in [0,T]$ and $q = 1,2,\ldots$, where $0 < \alpha < 1$. Now, since $\Gamma(q+j(1-\alpha)+1) \geq \Gamma(q+1) = q!$, for $j = 0,1,\ldots,q$, we can rewrite (2.102), to obtain

$$v^q(t) \leq \frac{t^q}{q!}\sum_{j=0}^{q}\left(\frac{q!}{j!(q-j)!}\right)L_{f,1}^{q-j}\left(L_{k,1}\Gamma(1-\alpha)t^{1-\alpha}\right)^j\|R\|_t^q \cdot \|v^0\|_t, \tag{2.103}$$

which by the Binomial Theorem is

$$v^q(t) \leq \frac{t^q}{q!}\left(L_{f,1} + L_{k,1}\Gamma(1-\alpha)t^{1-\alpha}\right)^q\|R\|_t^q \cdot \|v^0\|_t. \tag{2.104}$$

Therefore, the iteration scheme converges uniformly on $[0,T]$, as $q \to \infty$, and

$$\|y^q - y\|_T \leq \frac{T^q}{q!}\left(L_{f,1} + L_{k,1}\Gamma(1-\alpha)T^{1-\alpha}\right)^q\|R\|_T^q \cdot \|y^0 - y\|_T, \tag{2.105}$$

where $0 < \alpha < 1$, $q = 1,2,\ldots$, and $T > 0$ is finite. □

We now compare Theorem 2.7 (see equation (2.59)) and Theorem 2.10 (see equation (2.90)), where for easy of comparison, we set $L_{f,1}$ equal to zero. Then for $q = 1,2,\ldots$, where C_1, C_2 are constants we have:

Regular Kernel ($\alpha = 0$):

$$\|y^q - y\|_T \leq C_1^q L_{k,1}^q \frac{T^{2q}}{q!} \cdot \|y^0 - y\|_T.$$

Weakly Singular Kernel ($0 < \alpha < 1$):

$$\|y^q - y\|_T \leq C_2^q L_{k,1}^q \Gamma^q(1-\alpha) \frac{T^{q(2-\alpha)}}{q!} \cdot \|y^0 - y\|_T.$$

Note that $\Gamma(1-\alpha) \to +\infty$ as $\alpha \to 1^-$.

2.3 Discrete-Time Iteration WR Methods

For most applied problems, we will not be able to solve analytically the equations arising in (2.53) or (2.87), but will have to use some numerical method. In this case, where we discretize the time interval and apply numerical methods to solve the VIDEs, we obtain *discrete-time iteration* WR methods, see Chapter 1, Sections 1.4.3 and 1.4.4. For the remainder of this chapter, we specialize our analysis in the following four ways:

- We consider a special nonlinear Volterra integro-differential equation. This is the form usually found in applications.
- With one exception (see Section 2.3.3) we use the same iteration modes for the splitting functions G_f and G_k.
- The time intervals over which the iterations take place coincide with the step length.
- We use collocation methods as our underlying numerical methods.

For the purposes of this thesis, we call these methods *time-point relaxation collocation methods*, or **TRCol**, for short. In addition, we consider the Gauss-Jacobi and Gauss-Seidel iteration modes, only.

2.3.1 The Problem

Since it arises often in applications, we are concerned with the numerical solution of the VIDE

$$y'(t) = f(t,y) + \int_0^t a(t-s)K(y(s))ds, \tag{2.106}$$

where y is a vector on $\Re^m$, $f : [0,T] \times \Re^m \to \Re^m$, $K : \Re^m \to \Re^m$, where $0 < T < \infty$, and a is a scalar function, possibly a $C^0[0,T]$ or $C^1[0,T]$ function and we have the initial condition $y(0) = y_0 \in \Re^m$. Also, we assume that the kernel K and the function f are continuous on S and $[0,T]$, respectively, and satisfy uniform Lipschitz conditions (in y) with Lipschitz constants L_K and L_f, respectively. Referring to Theorem 2.1, this equation has a unique solution $y \in C^1([0,T])$. Theorem 2.7 considered continuous-time iteration WR methods for the solution of the problem (1.19), which includes this problem as a special case.

By relaxing the requirement that the scalar function a be continuous on $[0,T]$, we determine the following "weakly singular" version of the problem (2.106),

$$y'(t) = f(t,y) + \int_0^t (t-s)^{-\alpha}K(y(s))ds, \tag{2.107}$$

where $0 < \alpha < 1$ and we assume it has a unique solution $y \in C^1([0,T])$, for each initial value y_0. Clearly, this is a special case of the problem given by (2.21). See Theorem 2.10.

Since the problem is taking place in $\Re^m$ we let $z(t)$ be some (fixed) component

$y_i(t)$ of the exact solution $y(t)$, $i = 1, 2, \ldots m$ and define

$$\begin{aligned} u(t) &:= [y_1(t), y_2(t), \ldots, y_{i-1}(t)], \\ v(t) &:= [y_{i+1}(t), y_{i+2}(t), \ldots, y_m(t)]. \end{aligned} \tag{2.108}$$

The problem can then be written componentwise as:

$$z'(t) = f(t, u, z, v) + \int_0^t a(t-s)K(u(s), z(s), v(s))ds, \tag{2.109}$$

$z(0) = (y_0)_i$, where we have suppressed the subscript i, by letting it be understood that the i^{th}-component of both f and K are being used.

Gauss-Jacobi iterations:

Consider a partition $\Pi_N : 0 = t_0 < t_1 < \cdots < t_N = T$, $t_{i+1} - t_i = h$, for $i = 0, 1, \ldots, N-1$ and assume that the (continuous) approximation $\eta(t)$ to the solution $y(t)$ has already been computed for $t \in [0, t_n]$. To find the approximation $Z_n(t)$ of the corresponding i^{th}-component of $\eta(t)$ generated by the numerical method for $t \in [t_n, t_{n+1}]$ we consider first, the continuous-time iterations

$$\begin{aligned} \frac{d}{dt}z^q(t) &= f(t, u^{q-1}, z^q, v^{q-1}) + \int_0^{t_n} a(t-s)K(y(s))ds \\ &+ \int_{t_n}^t a(t-s)K(u^{q-1}(s), z^q(s), v^{q-1}(s))ds \\ z^q(t_n) &= \eta_i(t_n), \end{aligned} \tag{2.110}$$

where $z^q(0) = (y_0)_i$, for $q = 1, 2, \ldots$ and $i = 1, 2, \ldots, m$. In analogy with (2.108) for $i = 1, 2, \ldots, m$ and $q = 0, 1, \ldots$, define

$$u^q(t) := [y_1^q(t), y_2^q(t), \ldots, y_{i-1}^q(t)],$$

(2.111)

$$v^q(t) \;:=\; [y^q_{i+1}(t), y^q_{i+2}(t), \ldots, y^q_m(t)].$$

We define the initial guess for these components when $q = 0$ and although it is arbitrary, it is usually taken to be the components of the computed solution on the preceding interval, except for the first interval, in which case it is the constant function equal to the corresponding component of the given initial value, y_0. That is, let

$$\begin{aligned}
u^0(t) &:= [y^0_1(t), y^0_2(t), \ldots, y^0_{i-1}(t)] = [\eta_1(t-h), \eta_2(t-h), \ldots, \eta_{i-1}(t-h)], \\
v^0(t) &:= [y^0_{i+1}(t), y^0_{i+2}(t), \ldots, y^0_m(t)] = [\eta_{i+1}(t-h), \eta_{i+2}(t-h), \ldots, \eta_m(t-h)], \\
&\quad t \in [t_n, t_{n+1}],\ n = 1, 2, \ldots, N-1 \qquad (2.112) \\
u^0(t) &:= [(y_0)_1, (y_0)_2, \ldots, (y_0)_{i-1}], \\
v^0(t) &:= [(y_0)_{i+1}, (y_0)_{i+2}, \ldots, (y_0)_m]. \\
&\quad t \in [0, t_1]
\end{aligned}$$

Collocation is the numerical method we use to generate these approximations for each iteration and for each of the components of y. This is continued for either a fixed number of iterations or until some norm of the difference between two successive approximations is less than a given tolerance. We now look more closely at the form this collocation will take. The reader may wish to return to Chapter 1, Section 1.4.2; we are using equation (1.31). Recall that this method is an exact method, in that the integrals appearing in the method are not approximated. In practical implementation, quadrature formulas will be needed.

Consider the set of collocation points $T_N = \{t_n + c_j h; j = 1, 2, \ldots, r;\ n =$

$0, 1, \ldots, N-1\}$, where $0 \leq c_1 < c_2 < \cdots < c_r \leq 1$ and recall that the parameters associated with a collocation method are $\alpha_j(\tau) = \int_0^\tau L_j(\xi)d\xi$, where the L_j are the fundamental Lagrange polynomials with respect to the set $\{c_j\}$. In analogy with (2.111), for $i = 1, 2, \ldots, m$, $n = 0, 1, \ldots, N-1$ and $q = 0, 1, \ldots$, define

$$u_n^q(t_n + \tau h) := [\eta_1^q(t_n + \tau h), \eta_2^q(t_n + \tau h), \ldots, \eta_{i-1}^q(t_n + \tau h)],$$

and

$$v_n^q(t_n + \tau h) := [\eta_{i+1}^q(t_n + \tau h), \eta_{i+2}^q(t_n + \tau h), \ldots, \eta_m^q(t_n + \tau h)],$$

$\tau \in [0, 1]$, where $\eta \in S_r^{(0)}(Z_N)$.

Clearly, the initial functions u_n^0, v_n^0 are defined as in (2.112), i.e.

$$\begin{aligned}
u_n^0(t) &:= [\eta_1^0(t), \eta_2^0(t), \ldots, \eta_{i-1}^0(t)] = [\eta_1(t-h), \eta_2(t-h), \ldots, \eta_{i-1}(t-h)], \\
v_n^0(t) &:= [\eta_{i+1}^0(t), \eta_{i+2}^0(t), \ldots, \eta_m^0(t)] = [\eta_{i+1}(t-h), \eta_{i+2}(t-h), \ldots, \eta_m(t-h)], \\
&\quad t \in [t_n, t_{n+1}],\ n = 1, 2, \ldots, N-1 \qquad (2.113) \\
u_0^0(t) &:= [(y_0)_1, (y_0)_2, \ldots, (y_0)_{i-1}], \\
v_0^0(t) &:= [(y_0)_{i+1}, (y_0)_{i+2}, \ldots, (y_0)_m]. \\
&\quad t \in [0, t_1]
\end{aligned}$$

The method becomes

$$\begin{aligned}
Z_n^q(t_n + \tau h) &= Z_n(t_n) + h\sum_{j=1}^{r} \alpha_j(\tau) Y_{n,j}^q, \quad \tau \in [0, 1] \qquad (2.114) \\
Y_{n,j}^q &= f\left(t_n + c_j h, u_n^{q-1}(t_n + c_j h), Z_n(t_n) + h\sum_{k=1}^{r} \alpha_k(c_j) Y_{n,k}^q, v_n^{q-1}(t_n + c_j h)\right) \\
&\quad + h\sum_{l=0}^{n-1} \int_0^1 a[(t_n + c_j h) - (t_l + \xi h)] K(\eta(t_l + \xi h)) d\xi
\end{aligned}$$

$$+ \quad h\int_0^{c_j} a[(c_j-\xi)h] \\ K\left(u_n^{q-1}(t_n+\xi h), Z_n(t_n)+h\sum_{k=1}^{r}\alpha_k(\xi)Y_{n,k}^q, v_n^{q-1}(t_n+\xi h)\right)d\xi,$$

where $q = 1, 2, \ldots, n = 0, 1, \ldots, N-1$ and u_n^0, v_n^0 are given by (2.113).

Clearly, the corresponding method for solution of the weakly singular problem (2.107) is

$$Z_n^q(t_n+\tau h) = Z_n(t_n)+h\sum_{j=1}^{r}\alpha_j(\tau)Y_{n,j}^q, \quad \tau\in[0,1] \tag{2.115}$$

$$Y_{n,j}^q = f\left(t_n+c_jh, u_n^{q-1}(t_n+c_jh), Z_n(t_n)+h\sum_{k=1}^{r}\alpha_k(c_j)Y_{n,k}^q, v_n^{q-1}(t_n+c_jh)\right) \\ + \quad h\sum_{l=0}^{n-1}\int_0^1[(t_n+c_jh)-(t_l+\xi h)]^{-\alpha}K(\eta(t_l+\xi h))d\xi \\ + \quad h\int_0^{c_j}[(c_j-\xi)h]^{-\alpha} \\ K\left(u_n^{q-1}(t_n+\xi h), Z_n(t_n)+h\sum_{k=1}^{r}\alpha_k(\xi)Y_{n,k}^q, v_n^{q-1}(t_n+\xi h)\right)d\xi,$$

where $0<\alpha<1, q = 1, 2, \ldots, n = 0, 1, \ldots, N-1$ and u_n^0, v_n^0 are given by (2.113).

As in the case of (2.109) and (2.110), we have suppressed the subscript i, by letting it be understood that the i^{th}-component of both the function f, the kernel K and the $Y_{n,j}^q$ are being used.

Since the iterations in the above method involve decoupling the system of equations, they are naturally suited to implementation on parallel architectures. See Burrage (1995) for the ODE situation. Note that we compute all the components of $\eta(t)$ for $t\in[t_n, t_{n+1}]$ before we "upgrade" to the new values. That is, we do not use the "new" $\eta_i(t)$ to compute the remaining components, even though we have calculated it. In the Gauss-Seidel iterations, we use the new components of $\eta(t)$ as soon as they are computed.

Gauss-Seidel iterations:

As before, we first consider the continuous-time iterations

$$\begin{aligned}\frac{d}{dt}z^q(t) &= f(t,u^q,z^q,v^{q-1}) + \int_0^{t_n} a(t-s)K(y(s))ds \\ &+ \int_{t_n}^{t} a(t-s)K(u^q(s),z^q(s),v^{q-1}(s))ds \\ z^q(t_n) &= \eta_i(t_n),\end{aligned}$$

$z^q(0) = (y_0)_i$, $q = 1,2,\ldots$, $i = 1,2,\ldots,m$, and for all $t \in [t_n, t_{n+1}]$, $n = 0,1,\ldots,N-1$ and u^0, v^0 are given by (2.112).

Again, collocation is now applied to a system of m one-dimensional VIDEs

$$\begin{aligned}Z_n^q(t_n+\tau h) &= Z_n(t_n) + h\sum_{j=1}^{r}\alpha_j(\tau)Y_{n,j}^q, \quad \tau \in [0,1] \qquad (2.116)\\ Y_{n,j}^q &= f\left(t_n + c_j h, u_n^q(t_n+c_jh), Z_n(t_n) + h\sum_{k=1}^{r}\alpha_k(c_j)Y_{n,k}^q, v_n^{q-1}(t_n+c_jh)\right)\\ &+ h\sum_{l=0}^{n-1}\int_0^1 a[(t_n+c_jh)-(t_l+\xi h)]K(\eta(t_l+\xi h))d\xi\\ &+ h\int_0^{c_j} a[(c_j-\xi)h]\\ &\quad K\left(u_n^q(t_n+\xi h), Z_n(t_n) + h\sum_{k=1}^{r}\alpha_k(\xi)Y_{n,k}^q, v_n^{q-1}(t_n+\xi h)\right)d\xi,\end{aligned}$$

where $q = 1,2,\ldots, n = 0,1,\ldots,N-1$ and u_n^0, v_n^0 are given by (2.113).

Clearly, the corresponding method for solution of the weakly singular problem (2.107) is

$$\begin{aligned}Z_n^q(t_n+\tau h) &= Z_n(t_n) + h\sum_{j=1}^{r}\alpha_j(\tau)Y_{n,j}^q, \quad \tau \in [0,1] \qquad (2.117)\\ Y_{n,j}^q &= f\left(t_n + c_j h, u_n^q(t_n+c_jh), Z_n(t_n) + h\sum_{k=1}^{r}\alpha_k(c_j)Y_{n,k}^q, v_n^{q-1}(t_n+c_jh)\right)\end{aligned}$$

$$+ \quad h\sum_{l=0}^{n-1}\int_0^1 [(t_n + c_j h) - (t_l + \xi h)]^{-\alpha} K(\eta(t_l + \xi h))d\xi$$

$$+ \quad h\int_0^{c_j} [(c_j - \xi)h]^{-\alpha}$$

$$K\left(u_n^q(t_n + \xi h), Z_n(t_n) + h\sum_{k=1}^{r} \alpha_k(\xi) Y_{n,k}^q, v_n^{q-1}(t_n + \xi h)\right) d\xi,$$

where $0 < \alpha < 1, q = 1, 2, \ldots, n = 0, 1, \ldots, N-1$ and u_n^0, v_n^0 are given by (2.113).

2.3.2 Commutativity

In the procedure we have described above, time-point relaxation is applied to the m-dimensional system to decouple it into a system of m one-dimensional VIDEs, each of which is solved by collocation for each iteration. A natural question to ask, then, is whether the application of these methods can be commuted? That is, if we first apply collocation, which gives us an implicit m-dimensional system of algebraic equations to solve, and then use time-point relaxation to solve this system iteratively, will we get the same approximation? That is, does the following diagram commute?

$$\begin{array}{ccc} y & \rightarrow & \eta \\ \downarrow & & \downarrow \\ \{z\}_{i=1}^m & \rightarrow & \{Z_n\}_{i=1}^m \end{array}$$

When time-point relaxation (using Gauss-Jacobi iterations) is applied first to the problem followed by the application of collocation we refer to the resulting methods as time-point relaxation Gauss-Jacobi collocation methods, abbreviated **TRGJCol**. If Gauss-Seidel iterations are used instead, we call the methods time-point relaxation Gauss-Seidel collocation methods, abbreviated **TRGSCol**. When a

collocation method is applied first to the problem followed by the application of time-point relaxation (using Gauss-Jacobi iterations) we refer to the resulting methods as collocation time-point relaxation Gauss-Jacobi methods, abbreviated **ColTRGJ**. If Gauss-Seidel iterations are used instead, we call the methods collocation time-point relaxation Gauss-Seidel methods, abbreviated **ColTRGS**.

Theorem 2.11 *If we take the same initial guesses for* $\eta^0(t_n + \tau h)$ *and* $y^0(t_n + \tau h)$, $n = 0, 1, \ldots, N-1$, $\tau \in [0, 1]$; *that is if*

$$\eta^0(t_n + \tau h) = y^0(t_n + \tau h) = g(t_n + \tau h),$$

then the time-point relaxation Gauss-Jacobi collocation method is equivalent to the collocation time-point relaxation Gauss-Jacobi method, which we can symbolize as:

$$TRGJCol \equiv ColTRGJ,$$

and similarly, the time-point relaxation Gauss-Seidel collocation method is equivalent to the collocation time-point relaxation Gauss-Seidel method, which we can symbolize as:

$$TRGSCol \equiv ColTRGS.$$

Proof:

Since our interest in this chapter concerns the problem given by (2.106), our proof will be developed using this problem. However, it is clear that the arguments to follow apply to the more general VIDE with regular kernel (1.19), and weakly singular kernel (2.21).

Assume $\eta(t)$ is known on $[0, t_n]$, $n = 1, \ldots, N$, where, for all cases, we let $\eta(t)$ represent the resulting numerical solution of the problem. Our job then, will be to generate solutions on $[t_n, t_{n+1}]$. We continue with the concise component-wise notation, letting $z(t)$ represent a typical i^{th}-component, $i = 1, 2, \ldots, m$ of the exact solution $y(t)$, and $Z_n(t)$ represent the corresponding component of the numerical solution $\eta(t)$. In addition, we suppress the subscript i whenever possible by letting it be understood that the appropriate component of the vectors being referred to are being used.

TRGJCol $\equiv$ ColTRGJ:

TRGJCol:

Time-point relaxation (using Gauss-Jacobi iterations) decouples the system of *integro-differential* equations, so the i^{th}-component of (2.106) becomes

$$\begin{aligned} \frac{d}{dt}z^q(t) &= f(t, u^{q-1}, z^q, v^{q-1}) + \int_0^{t_n} a(t-s)K(y(s))ds \\ &+ \int_{t_n}^{t} a(t-s)K(u^{q-1}(s), z^q(s), v^{q-1}(s))ds \\ z^q(t_n) &= \eta_i(t_n), \end{aligned}$$

$z^q(0) = (y_0)_i$, $q = 1, 2, \ldots$, $i = 1, 2, \ldots, m$, and for all $t \in [t_n, t_{n+1}], n = 0, 1, \ldots, N-1$. From the assumption of the theorem, we take the initial guess as:

$$y^0(t_n + \tau h) = g(t_n + \tau h),$$

for $n = 0, 1, \ldots, N-1$ and $\tau \in [0, 1]$; that is

$$\begin{aligned} u^0(t) &:= [g_1(t), g_2(t), \ldots, g_{i-1}(t)], \\ v^0(t) &:= [g_{i+1}(t), g_{i+2}(t), \ldots, g_m(t)] \end{aligned}$$

where $i = 1, 2, \ldots, m$.

Collocation is now applied to generate the approximations which for each iteration is given by (2.114).

ColTRGJ:

Apply collocation to (2.106) to get the following continuous m-dimensionl approximation:

$$\begin{aligned}
\eta(t_n + \tau h) &= \eta(t_n) + h \sum_{j=1}^{r} \alpha_j(\tau) Y_{n,j}, \quad \tau \in [0,1], \\
Y_{n,j} &= f\left(t_n + c_j h, \eta(t_n) + h \sum_{k=1}^{r} \alpha_k(c_j) Y_{n,k}\right) \\
&+ h \sum_{l=0}^{n-1} \int_0^1 a[(t_n + c_j h) - (t_l + \xi h)] K(\eta(t_l + \xi h)) d\xi \\
&+ h \int_0^{c_j} a[(c_j - \xi)h] K\left(\eta(t_n) + h \sum_{k=1}^{r} \alpha_k(\xi) Y_{n,k}\right) d\xi.
\end{aligned}$$

Time-point relaxation (using Gauss-Jacobi iterations) can now be applied to decouple this system of *algebraic* equations. To evoke the conditions of the theorem, we let the initial guess for $\eta(t_n + \tau h)$ satisfy the equation

$$\eta^0(t_n + \tau h) = g(t_n + \tau h),$$

$n = 0, 1, \ldots, N-1$ and $\tau \in [0,1]$; that is

$$\begin{aligned}
u_n^0(t) &:= [g_1(t), g_2(t), \ldots, g_{i-1}(t)], \\
v_n^0(t) &:= [g_{i+1}(t), g_{i+2}(t), \ldots, g_m(t)]
\end{aligned}$$

where $i = 1, 2, \ldots, m$.

The i^{th}-component of $\eta(t)$ becomes (2.114).

This establishes the first equivalence and similar arguments establish the second equivalence, for Gauss-Seidel iterations. □

Let us illustrate these ideas by the special case of $\Re^2$.

2.3.3 Three Examples in $\Re^2$

We now give the following examples in $\Re^2$, that are all special cases of (2.106), with the same smoothness assumptions. This will require us to refer to specific components of the vectors $Y_{n,j}^q$ and rather than clutter the symbol with yet another subscript to represent this component number, we will use a superscript to the immediate left of the symbol. That is ${}^iY_{n,j}^q$, will be used to represent the i^{th}-component of $Y_{n,j}^q$.

Example 2.6 *Consider the case $m = 2$ of problem (2.106).*

Assume $[\eta_1(t), \eta_2(t)]^T$ is known on $[0, t_n]$, $n = 0, 1, \ldots N-1$, where we let η represent the resulting numerical solution of the problem. Our job, then, will be to generate solutions on $[t_n, t_{n+1}]$, using time-point relaxation (two-stage) collocation. We assume that

$$\eta^0(t_n + \tau h) = \eta(t_{n-1} + \tau h), \quad \text{for } n = 1, 2, \ldots, N-1, \ \tau \in [0, 1],$$

and

$$\eta^0(\tau h) = [y_{1,0}, y_{2,0}]^T \quad \text{for} \quad \tau \in [0, 1].$$

TRGJCol:

$$\eta_1^q(t_n + \tau h) \ = \ \eta_1(t_n) + h\{\alpha_1(\tau)\ {}^1Y_{n,1}^q + \alpha_2(\tau)\ {}^1Y_{n,2}^q\},$$

$$\eta_2^q(t_n+\tau h) \;=\; \eta_2(t_n)+h\{\alpha_1(\tau)\ {}^2Y_{n,1}^q+\alpha_2(\tau)\ {}^2Y_{n,2}^q\},$$

$\tau\in[0,1]$, where

$$\begin{aligned}
{}^1Y_{n,j}^q \;=\;& f_1\left(t_n+c_jh,\eta_1(t_n)+h\{\alpha_1(c_j)\ {}^1Y_{n,1}^q+\alpha_2(c_j)\ {}^1Y_{n,2}^q\},\eta_2^{q-1}(t_n+c_jh)\right)\\
+\;& h\sum_{l=0}^{n-1}\int_0^1 a[(t_n+c_jh)-(t_l+\xi h)]K_1(\eta_1(t_l+\xi h),\eta_2(t_l+\xi h))d\xi\\
+\;& h\int_0^{c_j} a[(c_j-\xi)h]\\
& K_1\left(\eta_1(t_n)+h\{\alpha_1(\xi)\ {}^1Y_{n,1}^q+\alpha_2(\xi)\ {}^1Y_{n,2}^q\},\eta_2^{q-1}(t_n+\xi h)\right)d\xi,
\end{aligned}$$

and

$$\begin{aligned}
{}^2Y_{n,j}^q \;=\;& f_2\left(t_n+c_jh,\eta_1^{q-1}(t_n+c_jh),\eta_2(t_n)+h\{\alpha_1(c_j)\ {}^2Y_{n,1}^q+\alpha_2(c_j)\ {}^2Y_{n,2}^q\}\right)\\
+\;& h\sum_{l=0}^{n-1}\int_0^1 a[(t_n+c_jh)-(t_l+\xi h)]K_2(\eta_1(t_l+\xi h),\eta_2(t_l+\xi h))d\xi\\
+\;& h\int_0^{c_j} a[(c_j-\xi)h]\\
& K_2\left(\eta_1^{q-1}(t_n+\xi h),\eta_2(t_n)+h\{\alpha_1(\xi)\ {}^2Y_{n,1}^q+\alpha_2(\xi)\ {}^2Y_{n,2}^q\}\right)d\xi,
\end{aligned}$$

for $j=1,2$, and $q=1,2,\ldots$.

TRGSCol:

$$\begin{aligned}
\eta_1^q(t_n+\tau h) \;&=\; \eta_1(t_n)+h\{\alpha_1(\tau)\ {}^1Y_{n,1}^q+\alpha_2(\tau)\ {}^1Y_{n,2}^q\},\\
\eta_2^q(t_n+\tau h) \;&=\; \eta_2(t_n)+h\{\alpha_1(\tau)\ {}^2Y_{n,1}^q+\alpha_2(\tau)\ {}^2Y_{n,2}^q\},
\end{aligned}$$

$\tau\in[0,1]$, where

$${}^1Y_{n,j}^q \;=\; f_1\left(t_n+c_jh,\eta_1(t_n)+h\{\alpha_1(c_j)\ {}^1Y_{n,1}^q+\alpha_2(c_j)\ {}^1Y_{n,2}^q\},\eta_2^{q-1}(t_n+c_jh)\right)$$

$$
\begin{aligned}
&+ \quad h\sum_{l=0}^{n-1}\int_0^1 a[(t_n+c_jh)-(t_l+\xi h)]K_1(\eta_1(t_l+\xi h),\eta_2(t_l+\xi h))d\xi \\
&+ \quad h\int_0^{c_j} a[(c_j-\xi)h] \\
&\qquad K_1\left(\eta_1(t_n)+h\{\alpha_1(\xi)\ {}^1Y^q_{n,1}+\alpha_2(\xi)\ {}^1Y^q_{n,2}\},\eta_2^{q-1}(t_n+\xi h)\right)d\xi,
\end{aligned}
$$

and

$$
\begin{aligned}
{}^2Y^q_{n,j} &= f_2\left(t_n+c_jh,\eta_1^q(t_n+c_jh),\eta_2(t_n)+h\{\alpha_1(c_j)\ {}^2Y^q_{n,1}+\alpha_2(c_j)\ {}^2Y^q_{n,2}\}\right) \\
&+ \quad h\sum_{l=0}^{n-1}\int_0^1 a[(t_n+c_jh)-(t_l+\xi h)]K_2(\eta_1(t_l+\xi h),\eta_2(t_l+\xi h))d\xi \\
&+ \quad h\int_0^{c_j} a[(c_j-\xi)h] \\
&\qquad K_2\left(\eta_1^q(t_n+\xi h),\eta_2(t_n)+h\{\alpha_1(\xi)\ {}^2Y^q_{n,1}+\alpha_2(\xi)\ {}^2Y^q_{n,2}\}\right)d\xi,
\end{aligned}
$$

for $j=1,2$, and $q=1,2,\ldots$.

Example 2.7 *Consider the following Volterra integro-differential system*

$$
\begin{pmatrix} y_1'(t) \\ y_2'(t) \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}\begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} + \begin{pmatrix} \int_0^t a(t-s)K_1(y_1(s),y_2(s))ds \\ \int_0^t a(t-s)K_2(y_1(s),y_2(s))ds \end{pmatrix},
$$

where the a_{ij} are real constants, $[y_{1,0},y_{2,0}]^T$ are given initial conditions and we use one-stage collocation.

Assume $[\eta_1(t),\eta_2(t)]^T$ is known on $[0,t_n]$, $n=0,1,\ldots N-1$, where we let η represent the resulting numerical solution of the problem. Our job then, will be to generate solutions on $[t_n,t_{n+1}]$, using time-point relaxation (one-stage) collocation.

We assume that

$$\eta^0(t_n+\tau h)=\eta(t_{n-1}+\tau h)$$

for $n=1,2,\ldots,N-1$, and

$$\eta^0(\tau h)=[y_{1,0},y_{2,0}]^T,$$

for $\tau \in [0,1]$.

TRGJCol:

$$\begin{aligned}
\eta_1^q(t_n + \tau h) &= \eta_1(t_n) + h\tau\ {}^1Y_n^q, \\
\eta_2^q(t_n + \tau h) &= \eta_2(t_n) + h\tau\ {}^2Y_n^q,
\end{aligned}$$

$\tau \in [0,1]$, where

$$\begin{aligned}
{}^1Y_n^q &= a_{11}\{\eta_1(t_n) + hc\ {}^1Y_n^q\} + a_{12}\eta_2^{q-1}(t_n + ch) \\
&+ h\sum_{l=0}^{n-1}\int_0^1 a[(t_n + ch) - (t_l + \xi h)]K_1(\eta_1(t_l + \xi h), \eta_2(t_l + \xi h))d\xi \\
&+ h\int_0^c a[(c - \xi)h]K_1(\eta_1(t_n) + \xi h\ {}^1Y_n^q, \eta_2^{q-1}(t_n + \xi h))d\xi,
\end{aligned}$$

and

$$\begin{aligned}
{}^2Y_n^q &= a_{21}\eta_1^{q-1}(t_n + ch) + a_{22}\{\eta_2(t_n) + hc\ {}^2Y_n^q\} + \\
&+ h\sum_{l=0}^{n-1}\int_0^1 a[(t_n + ch) - (t_l + \xi h)]K_2(\eta_1(t_l + \xi h), \eta_2(t_l + \xi h))d\xi \\
&+ h\int_0^c a[(c - \xi)h]K_2(\eta_1^{q-1}(t_n + \xi h), \eta_2(t_n) + \xi h\ {}^2Y_n^q)d\xi,
\end{aligned}$$

$n = 0, 1, \ldots, N-1$ and $q = 1, 2, \ldots$, where we call the single collocation point $c_1 =: c$ and we have used the fact that

$$\alpha(\tau) = \tau.$$

TRGSCol:

$$\begin{aligned}\eta_1^q(t_n+\tau h) &= \eta_1(t_n)+h\tau\ {}^1Y_n^q,\\ \eta_2^q(t_n+\tau h) &= \eta_2(t_n)+h\tau\ {}^2Y_n^q,\quad \tau\in[0,1],\end{aligned}$$

where

$$\begin{aligned}{}^1Y_n^q &= a_{11}\{\eta_1(t_n)+hc\ {}^1Y_n^q\}+a_{12}\eta_2^{q-1}(t_n+ch)\\ &+ h\sum_{l=0}^{n-1}\int_0^1 a[(t_n+ch)-(t_l+\xi h)]K_1(\eta_1(t_l+\xi h),\eta_2(t_l+\xi h))d\xi\\ &+ h\int_0^c a[(c-\xi)h]K_1(\eta_1(t_n)+\xi h\ {}^1Y_n^q,\eta_2^{q-1}(t_n+\xi h))d\xi,\end{aligned}$$

and

$$\begin{aligned}{}^2Y_n^q &= a_{21}\eta_1^q(t_n+ch)+a_{22}\{\eta_2(t_n)+hc\ {}^2Y_n^q\}+\\ &+ h\sum_{l=0}^{n-1}\int_0^1 a[(t_n+ch)-(t_l+\xi h)]K_2(\eta_1(t_l+\xi h),\eta_2(t_l+\xi h))d\xi\\ &+ h\int_0^c a[(c-\xi)h]K_2(\eta_1^q(t_n+\xi h),\eta_2(t_n)+\xi h\ {}^2Y_n^q)d\xi,\end{aligned}$$

$n=0,1,\ldots,N-1$ and $q=1,2,\ldots$.

As promised in Section 2.3, we now give an example where we use *different* iteration schemes for the splitting functions G_f and G_k.

Example 2.8 *Consider Example 2.7 and again use one-stage collocation and Gauss-Jacobi and Gauss-Seidel modes, respectively, for G_f. However, assume $K_i\in C^1(S)$, $i=1,2$, and use Newton iteration for G_k.*

Assume $[\eta_1(t), \eta_2(t)]^T$ is known on $[0, t_n]$, $n = 0, 1, \ldots N-1$, where we let η represent the resulting numerical solution of the problem. Our job, then, will be to generate solutions on $[t_n, t_{n+1}]$, using time-point relaxation (one-stage) collocation.

We assume that

$$\eta^0(t_n + \tau h) = \eta(t_{n-1} + \tau h)$$

for $n = 1, 2, \ldots, N-1$, and

$$\eta^0(\tau h) = [y_{1,0}, y_{2,0}]^T, \quad \tau \in [0, 1].$$

In both methods to follow, we assume that Newton iteration is used for G_k. To simplify the resulting equations, we write:

$$J_{ij}(\xi) = \frac{\partial K_i(y_1, y_2)}{\partial y_j},$$

where this partial derivative is evaluated at

$$\begin{aligned} y_1 &= \eta_1^{q-1}(t_n + \xi h) \\ y_2 &= \eta_2^{q-1}(t_n + \xi h), \end{aligned}$$

for $i, j = 1, 2$, $n = 0, 1, \ldots, N-1, q = 1, 2 \ldots$ and $\xi \in [0, 1]$.

Gauss-Jacobi iterations for G_f:

$$\begin{aligned} \eta_1^q(t_n + \tau h) &= \eta_1(t_n) + h\tau\ {}^1Y_n^q, \\ \eta_2^q(t_n + \tau h) &= \eta_2(t_n) + h\tau\ {}^2Y_n^q, \quad \tau \in [0, 1], \end{aligned}$$

where

$${}^1Y_n^q = a_{11}\{\eta_1(t_n) + hc\ {}^1Y_n^q\} + a_{12}\eta_2^{q-1}(t_n + ch)$$

$$
\begin{aligned}
&+ \quad h\sum_{l=0}^{n-1}\int_0^1 a[(t_n+ch)-(t_l+\xi h)]K_1(\eta_1(t_l+\xi h),\eta_2(t_l+\xi h))d\xi \\
&+ \quad h\int_0^c a[(c-\xi)h]\Big(K_1(\eta_1(t_n)+h\xi\ {}^1Y_n^{q-1},\eta_2(t_n)+h\xi\ {}^2Y_n^{q-1}) \\
&+ \quad h\xi\{J_{11}(\xi)[{}^1Y_n^q-{}^1\,Y_n^{q-1}]+J_{12}(\xi)[{}^2Y_n^q-{}^2\,Y_n^{q-1}]\}\Big)\,d\xi,
\end{aligned}
$$

and

$$
\begin{aligned}
{}^2Y_n^q &= a_{21}\eta_1^{q-1}(t_n+ch)+a_{22}\{\eta_2(t_n)+hc\ {}^2Y_n^q\}+ \\
&+ \quad h\sum_{l=0}^{n-1}\int_0^1 a[(t_n+ch)-(t_l+\xi h)]K_2(\eta_1(t_l+\xi h),\eta_2(t_l+\xi h))d\xi \\
&+ \quad h\int_0^c a[(c-\xi)h]\Big(K_2(\eta_1(t_n)+h\xi\ {}^1Y_n^{q-1},\eta_2(t_n)+h\xi\ {}^2Y_n^{q-1}) \\
&+ \quad h\xi\{J_{21}(\xi)[{}^1Y_n^q-{}^1\,Y_n^{q-1}]+J_{22}(\xi)[{}^2Y_n^q-{}^2\,Y_n^{q-1}]\}\Big)\,d\xi,
\end{aligned}
$$

$n=0,1,\ldots,N-1$ and $q=1,2,\ldots$, where we call the single collocation point $c_1=:c$ and we have used the fact that

$$\alpha(\tau)=\tau.$$

Gauss-Seidel iterations for G_f:

$$
\begin{aligned}
\eta_1^q(t_n+\tau h) &= \eta_1(t_n)+h\tau\ {}^1Y_n^q, \\
\eta_2^q(t_n+\tau h) &= \eta_2(t_n)+h\tau\ {}^2Y_n^q, \quad \tau\in[0,1],
\end{aligned}
$$

where

$$
\begin{aligned}
{}^1Y_n^q &= a_{11}\{\eta_1(t_n)+hc\ {}^1Y_n^q\}+a_{12}\eta_2^{q-1}(t_n+ch) \\
&+ \quad h\sum_{l=0}^{n-1}\int_0^1 a[(t_n+ch)-(t_l+\xi h)]K_1(\eta_1(t_l+\xi h),\eta_2(t_l+\xi h))d\xi
\end{aligned}
$$

$$+ \quad h\int_0^c a[(c-\xi)h]\left(K_1(\eta_1(t_n)+h\xi\ {}^1Y_n^{q-1},\eta_2(t_n)+h\xi\ {}^2Y_n^{q-1})\right.$$
$$+ \quad \left. h\xi\{J_{11}(\xi)[{}^1Y_n^q - {}^1\ Y_n^{q-1}] + J_{12}(\xi)[{}^2Y_n^q - {}^2\ Y_n^{q-1}]\}\right) d\xi,$$

and

$$\begin{aligned}
{}^2Y_n^q &= a_{21}\eta_1^q(t_n+ch) + a_{22}\{\eta_2(t_n)+hc\ {}^2Y_n^q\} + \\
&+ h\sum_{l=0}^{n-1}\int_0^1 a[(t_n+ch)-(t_l+\xi h)]K_2(\eta_1(t_l+\xi h),\eta_2(t_l+\xi h))d\xi \\
&+ h\int_0^c a[(c-\xi)h]\left(K_2(\eta_1(t_n)+h\xi\ {}^1Y_n^{q-1},\eta_2(t_n)+h\xi\ {}^2Y_n^{q-1})\right. \\
&+ \left. h\xi\{J_{21}(\xi)[{}^1Y_n^q - {}^1\ Y_n^{q-1}] + J_{22}(\xi)[{}^2Y_n^q - {}^2\ Y_n^{q-1}]\}\right) d\xi,
\end{aligned}$$

$n = 0,1,\ldots,N-1$ and $q = 1,2,\ldots$.

2.3.4 Convergence of the Iterations

In Section 2.2 we considered the question of convergence of the continuous-time iteration waveform relaxation methods, see Theorems 2.7 and 2.10. We now ask the same question for the discrete-time methods. That is, are these methods (2.114), (2.115), (2.116) and (2.117) well defined as $q \to \infty$? Using standard contraction principle arguments, we now show that these limits do exist for sufficiently small h.

It should be pointed out that we are not concerned in this section, with the function the methods converge to, just that they *do* converge. We leave this other question to later sections.

In both the Gauss-Jacobi (2.114) and the Gauss-Seidel (2.116) methods, collocation on $[t_n, t_{n+1}]$ gives

$$Z_n^q(t_n+\tau h) = Z_n(t_n) + h\sum_{j=1}^{r}\alpha_j(\tau)Y_{n,j}^q, \tag{2.118}$$

$q = 1, 2, \ldots$ and $\tau \in [0,1]$, where the initial guess, $Z_n^0(t_n + \tau h) = Z_n(t_{n-1} + \tau h)$, for $n = 1, 2, \ldots, N-1$, and $Z_0^0(\tau h) = (y_0)_i$, for $i = 1, 2, \ldots, m$ and $\tau \in [0,1]$. If we can show that $Y_{n,j}^q \to Y_{n,j}$, as $q \to \infty$, then we find immediately

$$
\begin{aligned}
Z_n(t_n + \tau h) &= Z_n(t_n) + h \sum_{j=1}^{r} \alpha_j(\tau) Y_{n,j}, \quad \tau \in [0,1], \\
Y_{n,j} &= f\left(t_n + c_j h, u_n(t_n + c_j h), Z_n(t_n) + h \sum_{k=1}^{r} \alpha_k(c_j) Y_{n,k}, v_n(t_n + c_j h)\right) \\
&+ h \sum_{l=0}^{n-1} \int_0^1 a[(t_n + c_j h) - (t_l + \xi h)] K(\eta(t_l + \xi h)) d\xi \\
&+ h \int_0^{c_j} a[(c_j - \xi)h] \\
&\quad K\left(u_n(t_n + \xi h), Z_n(t_n) + h \sum_{k=1}^{r} \alpha_k(\xi) Y_{n,k}, v_n(t_n + \xi h)\right) d\xi,
\end{aligned}
$$

where u_n^0, v_n^0 are given by (2.113) and we indicate limits (as $q \to \infty$) by dropping the subscript q.

This requires a standard contraction principle type argument. That is, given

$$x = \varphi(x), \quad \varphi : \Re^n \to \Re^n, \tag{2.119}$$

then the sequence $\{x^q\}$ defined by

$$x^q = \varphi(x^{q-1}), \quad q = 1, 2, \ldots, \tag{2.120}$$

where x^0 is arbitrary, converges to the unique solution of (2.119), if the conditions of the following theorem are satisfied.

Lemma 2.7 *Let $\varphi(x)$ satisfy a Lipschitz condition*

$$\|\varphi(x) - \varphi(\tilde{x})\| \leq L_c \|x - \tilde{x}\|$$

for all $x, \tilde{x}$ with Lipschitz constant L_c satisfying $0 \leq L_c < 1$. Then there exists a unique solution $\bar{x}$ of (2.119), and if $\{x^q\}$ is defined by (2.120), then $x^q \to \bar{x}$ as $q \to \infty$.

See Lambert (1991).

First we have to rewrite the methods (2.114) and (2.116), in vector form that will allow us to apply this lemma. To facilitate this, we introduce the following vectors and matrices:

$\underline{Y_n^q \in \Re^{rm}}$:

$$Y_n^q := \left([{}^1Y_{n,1}^q, {}^1Y_{n,2}^q, \ldots, {}^1Y_{n,r}^q]^T, [{}^2Y_{n,1}^q, {}^2Y_{n,2}^q, \ldots, {}^2Y_{n,r}^q]^T, \ldots, \right. \\ \left. [{}^mY_{n,1}^q, {}^mY_{n,2}^q, \ldots, {}^mY_{n,r}^q]^T\right)^T, \tag{2.121}$$

$q = 1, 2, \ldots$.

$\underline{M_i \in \Re^{rm \times m}}$, $i = 1, 2, \ldots rm$:

$$M_i = (m_{uv}) = \begin{cases} 1 & u = i \text{ and } v = \lceil i/r \rceil \\ 0 & \text{otherwise} \end{cases},$$

where we define, for any real number a, $\lceil a \rceil$ to be the smallest integer greater than or equal to a. That is, M_i has all zeros, except for a single 1 in row i, column $\lceil i/r \rceil$:

$$M_i = \begin{pmatrix} 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \end{pmatrix}. \tag{2.122}$$

$\underline{A_i(\tau) \in \Re^{m \times rm}}$, $i = 1, 2, \ldots m, \tau \in [0, 1]$:

$$A_i(\tau) = (a_{uv}) = \begin{cases} \alpha_k(\tau) & u = i \text{ and } v = r(i-1)+k, k = 1,2,\ldots,r \\ 0 & \text{otherwise} \end{cases}.$$

That is, $A_i(\tau)$ has all zeros, except for the row elements, $\alpha_1(\tau), \alpha_2(\tau), \ldots, \alpha_r(\tau)$, *beginning* at row i, column $r(i-1)+1$:

$$A_i(\tau) = \begin{pmatrix} 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & \alpha_1(\tau) & \alpha_2(\tau) & \cdots & \alpha_r(\tau) & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \end{pmatrix}. \tag{2.123}$$

We first consider **Gauss-Jacobi** iterations. To do this we write the expression for ${}^iY^q_{n,j}$, explicitly showing all components.

$$\begin{aligned} {}^iY^q_{n,j} &= f_i\Bigg(t_n + c_j h, \eta_1(t_n) + h\sum_{k=1}^{r} \alpha_k(c_j)\ {}^1Y^{q-1}_{n,k}, \eta_2(t_n) + h\sum_{k=1}^{r} \alpha_k(c_j)\ {}^2Y^{q-1}_{n,k}, \ldots, \\ &\quad \eta_i(t_n) + h\sum_{k=1}^{r} \alpha_k(c_j)\ {}^iY^q_{n,k}, \eta_{i+1}(t_n) + h\sum_{k=1}^{r} \alpha_k(c_j)\ {}^{i+1}Y^{q-1}_{n,k}, \ldots, \\ &\quad \eta_m(t_n) + h\sum_{k=1}^{r} \alpha_k(c_j)\ {}^mY^{q-1}_{n,k}\Bigg) \\ &+ h\sum_{l=0}^{n-1}\int_0^1 a[(t_n + c_j h) - (t_l + \xi h)] K_i(\eta(t_l + \xi h))d\xi \\ &+ h\int_0^{c_j} a[(c_j - \xi)h] \\ &\quad K_i\Bigg(\eta_1(t_n) + h\sum_{k=1}^{r} \alpha_k(\xi)\ {}^1Y^{q-1}_{n,k}, \eta_2(t_n) + h\sum_{k=1}^{r} \alpha_k(\xi)\ {}^2Y^{q-1}_{n,k}, \ldots, \\ &\quad \eta_i(t_n) + h\sum_{k=1}^{r} \alpha_k(\xi)\ {}^iY^q_{n,k}, \eta_{i+1}(t_n) + h\sum_{k=1}^{r} \alpha_k(\xi)\ {}^{i+1}Y^{q-1}_{n,k}, \ldots, \\ &\quad \eta_m(t_n) + h\sum_{k=1}^{r} \alpha_k(\xi)^m Y^{q-1}_{n,k}\Bigg)\, d\xi, \quad \tau \in [0,1], \end{aligned} \tag{2.124}$$

where $i = 1,2,\ldots m,\ j = 1,2,\ldots r$ and $q = 1,2,\ldots$.

Note: In light of the initial guess $Z_n^0(t_n+\tau h) = Z_n(t_{n-1}+\tau h)$, for $n = 1,2,\ldots,N-1$, and $Z_0^0(\tau h) = (y_0)_i$, for $i = 1,2,\ldots,m,\ \tau \in [0,1]$, it is clear that we take

$$\begin{aligned} {}^iY_{n,j}^0 &= {}^iY_{n-1,j}, \quad \text{for } n = 1,2,\ldots,N-1, \\ {}^iY_{0,j}^0 &= 0, \end{aligned} \tag{2.125}$$

for $i = 1,2,\ldots m,\ j = 1,2,\ldots r$, and we recall that we indicate limits (as $q \to \infty$) by dropping the subscript q, that is

$${}^iY_{n-1,j} := \lim_{q\to\infty} {}^iY_{n-1,j}^q.$$

◇

Clearly (2.124) gives a single component of the rm-dimensional vector Y_n^q defined by (2.121). Now using the matrices (2.122) and (2.123), we obtain

$$\begin{aligned} Y_n^q &= \sum_{i=1}^{m}\sum_{j=1}^{r} M_{(i-1)r+j}\left[f\left(t_n + c_j h, \eta(t_n) + h\{A_i(c_j)Y_n^q + \sum_{\substack{k=1\\k\neq i}}^{m} A_k(c_j)Y_n^{q-1}\}\right)\right. \\ &+ \left. h\int_0^{c_j} a[(c_j-\tau)h]K\left(\eta(t_n) + h\{A_i(\tau)Y_n^q + \sum_{\substack{k=1\\k\neq i}}^{m} A_k(\tau)Y_n^{q-1}\}\right)d\tau\right] \\ &+ \sum_{i=1}^{m}\sum_{j=1}^{r} M_{(i-1)r+j} h \sum_{l=0}^{n-1}\int_0^1 a[(t_n+c_jh)-(t_l+\tau h)]K(\eta(t_l+\tau h))d\tau, \end{aligned} \tag{2.126}$$

$q = 1,2,\ldots$ and Y_n^0 given by (2.125).

The corresponding version for **Gauss-Seidel** iterations is

$$
\begin{aligned}
Y_n^q &= \sum_{i=1}^{m}\sum_{j=1}^{r} M_{(i-1)r+j}\left[f\left(t_n + c_j h, \eta(t_n) + h\{\sum_{k=1}^{i} A_k(c_j)Y_n^q + \sum_{k=i+1}^{m} A_k(c_j)Y_n^{q-1}\}\right)\right. \\
&+ \left. h\int_0^{c_j} a[(c_j-\tau)h]K\left(\eta(t_n) + h\{\sum_{k=1}^{i} A_k(\tau)Y_n^q + \sum_{k=i+1}^{m} A_k(\tau)Y_n^{q-1}\}\right)d\tau\right] \\
&+ \sum_{i=1}^{m}\sum_{j=1}^{r} M_{(i-1)r+j} h \sum_{l=0}^{n-1}\int_0^1 a[(t_n + c_j h) - (t_l + \tau h)]K(\eta(t_l + \tau h))d\tau, \qquad (2.127)
\end{aligned}
$$

$q = 1, 2, \ldots$ and Y_n^0 given by (2.125).

To complete the vector formulation, we define the following vectors:

$\underline{\alpha(\tau) \in \Re^r}$:

$$\alpha(\tau) := [\alpha_1(\tau), \alpha_2(\tau), \ldots, \alpha_r(\tau)]^T, \quad \tau \in [0,1].$$

$I_m \in \Re^{m\times m}$ is the identity matrix, and recall the definition of the direct product $\otimes$, sometimes called Kronecker or tensor product of matrices. See Lancaster and Tismenetsky (1985). Then $I_m \otimes \alpha^T(\tau) \in \Re^{m\times rm}$ is given by

$$
I_m \otimes \alpha^T(\tau) = \begin{pmatrix}
\alpha^T(\tau) & \cdots & 0 & \cdots & 0 \\
\vdots & & \vdots & & \vdots \\
0 & \cdots & \alpha^T(\tau) & \cdots & 0 \\
\vdots & & \vdots & & \vdots \\
0 & \cdots & 0 & \cdots & \alpha^T(\tau)
\end{pmatrix}, \quad \tau \in [0,1].
$$

Then, in both cases we can write the method in vector form. Equation (2.118) becomes

$$\eta^q(t_n + \tau h) = \eta(t_n) + h[I_m \otimes \alpha^T(\tau)] \cdot Y_n^q, \qquad (2.128)$$

where Y_n^0 is given by (2.125), for $\tau \in [0,1]$, $q = 1, 2, \ldots$ and $n = 0, 1, \ldots, N-1$. The only difference between iteration modes is the expression for Y_n^q, which is given by

(2.126) or (2.127). Observe that

$$\sum_{k=1}^{m} A_k(\tau) = I_m \otimes \alpha^T(\tau).$$

Then, as we now show in the following theorem, in the limit as $q \to \infty$, this method becomes

$$\eta(t_n + \tau h) = \eta(t_n) + h[I_m \otimes \alpha^T(\tau)] \cdot Y_n, \quad \tau \in [0,1], \tag{2.129}$$

where for both modes of iteration,

$$\begin{aligned} Y_n &= \sum_{i=1}^{m}\sum_{j=1}^{r} M_{(i-1)r+j}[f(t_n + c_j h, \eta(t_n) + h[I_m \otimes \alpha^T(c_j)] \cdot Y_n) \\ &+ h\int_0^{c_j} a[(c_j - \tau)h]K(\eta(t_n) + h[I_m \otimes \alpha^T(\tau)] \cdot Y_n)d\tau] \\ &+ \sum_{i=1}^{m}\sum_{j=1}^{r} M_{(i-1)r+j} h \sum_{l=0}^{n-1}\int_0^1 a[(t_n + c_j h) - (t_l + \tau h)]K(\eta(t_l + \tau h))d\tau, \end{aligned} \tag{2.130}$$

$n = 0, 1, \ldots, N-1$, where we drop the subscript q on Y_n to indicate that the limit as $q \to \infty$ has been taken. Now, using the vector formulation we can apply Lemma 2.7 to prove the following theorem.

Theorem 2.12 *Consider the Volterra integro-differential equation (2.106), where K and f satisfy uniform Lipschitz conditions (in y) with Lipschitz constants L_K and L_f, respectively, and the scalar function a is a $C^0[0,T]$ function, where $0 < T < \infty$. Then the time-point relaxation collocation method (2.128), where the iteration mode is Gauss-Jacobi or Gauss-Seidel, in which case the expression for Y_n^q is given by (2.126) or (2.127), respectively, converges as $q \to \infty$ for sufficiently small $h > 0$. In this case, the two methods are identical and we have:*

$$Y_n = \sum_{i=1}^{m}\sum_{j=1}^{r} M_{(i-1)r+j}[f(t_n + c_j h, \eta(t_n) + h[I_m \otimes \alpha^T(c_j)] \cdot Y_n)$$

$$+ \quad h\int_0^{c_j} a[(c_j-\tau)h]K(\eta(t_n)+h[I_m\otimes\alpha^T(\tau)]\cdot Y_n)d\tau] \qquad (2.131)$$

$$+ \quad \sum_{i=1}^{m}\sum_{j=1}^{r} M_{(i-1)r+j}h\sum_{l=0}^{n-1}\int_0^1 a[(t_n+c_jh)-(t_l+\tau h)]K(\eta(t_l+\tau h))d\tau,$$

$n=0,1,\ldots,N-1.$

Proof:

Let $\tau\in[0,1]$ and $n\in\{0,1,\ldots,N-1\}$ fixed, and consider iteration over $q=1,2,\ldots$. We can show that $\eta^q(t_n+\tau h)$ converges as $q\to\infty$, if we can show that Y_n^q converges as $q\to\infty$, where Y_n^q is given by (2.126) or (2.127), and Y_n^0 is given by (2.125).

Clearly, Y_n^q is given *implicitly* by (2.126) and (2.127), and we apply Lemma 2.7 to these expressions.

Let us consider Y_n^q and $\tilde{Y}_n^q$ and calculate their difference. We get, for Gauss-Jacobi iterations,

$$\begin{aligned}
\|Y_n^q-\tilde{Y}_n^q\| \;=\; & \|\sum_{i=1}^{m}\sum_{j=1}^{r} M_{(i-1)r+j} \\
& \left[f(t_n+c_jh,\eta(t_n)+h\{A_i(c_j)Y_n^q+\sum_{\substack{k=1\\k\neq i}}^{m} A_k(c_j)Y_n^{q-1}\}) \right. \\
& \left. + \; h\int_0^{c_j} a[(c_j-\tau)h]K(\eta(t_n)+h\{A_i(\tau)Y_n^q+\sum_{\substack{k=1\\k\neq i}}^{m} A_k(\tau)Y_n^{q-1}\})d\tau \right] \\
& - \sum_{i=1}^{m}\sum_{j=1}^{r} M_{(i-1)r+j} \\
& \left[f(t_n+c_jh,\eta(t_n)+h\{A_i(c_j)\tilde{Y}_n^{\,q}+\sum_{\substack{k=1\\k\neq i}}^{m} A_k(c_j)\tilde{Y}_n^{\,q-1}\}) \right.
\end{aligned}$$

$$+ \quad h\int_0^{c_j} a[(c_j-\tau)h]K(\eta(t_n)+h\{A_i(\tau)\tilde{Y}_n^{\,q}+\sum_{\substack{k=1\\k\neq i}}^{m} A_k(\tau)\tilde{Y}_n^{\,q-1}\})d\tau\Bigg]\Bigg\|$$

$q = 1, 2, \ldots$.

Define the non-negative real number $\triangle$,

$$\triangle := \max\{\|M_{(i-1)r+j}\|, \|A_i(\tau)\| : i = 1, 2, \ldots, m,\ j = 1, 2, \ldots, r,\ \tau \in [0,1]\}.$$

Note that we have implicitly assumed that the above matrix norm is *compatible* with our vector norm $\|\cdot\|$. That is, for any vector v and matrix Q, where the product Qv is defined

$$\|Qv\| \leq \|Q\| \cdot \|v\|.$$

See Lancaster and Tismenetsky (1985).

We now apply the generalized triangle inequality and recall that both f and K satisfy Lipschitz conditions with Lipschitz constants L_f and L_K, respectively. Therefore, the above simplifies to

$$\begin{aligned}\|Y_n^q - \tilde{Y}_n^q\| \quad = \quad & \|\sum_{i=1}^{m}\sum_{j=1}^{r} M_{(i-1)r+j} \\ & \Bigg[\Bigg(f(t_n + c_j h, \eta(t_n) + h\{A_i(c_j)Y_n^q + \sum_{\substack{k=1\\k\neq i}}^{m} A_k(c_j)Y_n^{q-1}\}) \\ & - \quad f(t_n + c_j h, \eta(t_n) + h\{A_i(c_j)\tilde{Y}_n^{\,q} + \sum_{\substack{k=1\\k\neq i}}^{m} A_k(c_j)\tilde{Y}_n^{\,q-1}\})\Bigg)\end{aligned}$$

$$
\begin{aligned}
&+ \quad h\int_0^{c_j} a[(c_j-\tau)h]\left(K(\eta(t_n)+h\{A_i(\tau)Y_n^q+\sum_{\substack{k=1\\k\neq i}}^{m}A_k(\tau)Y_n^{q-1}\}\right.\\
&- \quad \left.K(\eta(t_n)+h\{A_i(\tau)\tilde{Y}_n{}^q+\sum_{\substack{k=1\\k\neq i}}^{m}A_k(\tau)\tilde{Y}_n{}^{q-1}\})d\tau\right)\Bigg]\Bigg\|\\
&\leq \quad \Delta\sum_{i=1}^{m}\sum_{j=1}^{r}\Bigg\|\Bigg[\Bigg(f(t_n+c_jh,\eta(t_n)+h\{A_i(c_j)Y_n^q+\sum_{\substack{k=1\\k\neq i}}^{m}A_k(c_j)Y_n^{q-1}\})\\
&- \quad f(t_n+c_jh,\eta(t_n)+h\{A_i(c_j)\tilde{Y}_n{}^q+\sum_{\substack{k=1\\k\neq i}}^{m}A_k(c_j)\tilde{Y}_n{}^{q-1}\})\Bigg)\\
&+ \quad h\int_0^{c_j} a[(c_j-\tau)h]\left(K(\eta(t_n)+h\{A_i(\tau)Y_n^q+\sum_{\substack{k=1\\k\neq i}}^{m}A_k(\tau)Y_n^{q-1}\})\right.\\
&- \quad \left.K(\eta(t_n)+h\{A_i(\tau)\tilde{Y}_n{}^q+\sum_{\substack{k=1\\k\neq i}}^{m}A_k(\tau)\tilde{Y}_n{}^{q-1}\})d\tau\right)\Bigg]\Bigg\|\\
&\leq \quad \Delta\sum_{i=1}^{m}\sum_{j=1}^{r}\Bigg\{hL_f\|A_i(c_j)(Y_n^q-\tilde{Y}_n{}^q)+\sum_{\substack{k=1\\k\neq i}}^{m}A_k(c_j)(Y_n^{q-1}-\tilde{Y}_n{}^{q-1})\|\\
&+ \quad h\int_0^{c_j}|a[(c_j-\tau)h]|\\
&\cdot \quad hL_K\|A_i(\tau)(Y_n^q-\tilde{Y}_n{}^q)+\sum_{\substack{k=1\\k\neq i}}^{m}A_k(\tau)(Y_n^{q-1}-\tilde{Y}_n{}^{q-1})\|d\tau\Bigg\}\\
&\leq \quad \Delta\sum_{i=1}^{m}\sum_{j=1}^{r}\left\{hL_f\{\Delta\|Y_n^q-\tilde{Y}_n{}^q\|+(m-1)\Delta\|Y_n^{q-1}-\tilde{Y}_n{}^{q-1}\|\right\}
\end{aligned}
$$

$$
\begin{aligned}
&+ \quad h\int_0^{c_j} |a[(c_j-\tau)h]| \\
&\cdot \quad hL_K\{\Delta\|Y_n^q - \tilde{Y}_n{}^q\| + (m-1)\Delta\|Y_n^{q-1} - \tilde{Y}_n{}^{q-1}\|\}\, d\tau \\
&\leq \quad \Delta\sum_{i=1}^{m}\sum_{j=1}^{r} h\left\{\{\Delta\|Y_n^q - \tilde{Y}_n{}^q\| + (m-1)\Delta\|Y_n^{q-1} - \tilde{Y}_n{}^{q-1}\|\}(L_f\right. \\
&+ \quad \left. L_K h\int_0^{c_j} |a[(c_j-\tau)h]|d\tau)\right\}
\end{aligned}
$$

$q = 1, 2, \ldots.$

Since $a(t)$ is bounded on [0,T]

$$\int_0^{c_j} |a[(c_j-\tau)h]|d\tau \leq c_j a_{max} \leq a_{max}, \tag{2.132}$$

where $a_{max} := \max\{|a(t)| : 0 \leq t \leq T\}$, and we recall that $c_j \in [0,1], \forall j$. Then

$$\|Y_n^q - \tilde{Y}_n^q\| \leq rm\Delta^2 h(L_f + L_K h a_{max})(\|Y_n^q - \tilde{Y}_n{}^q\| + (m-1)\|Y_n^{q-1} - \tilde{Y}_n{}^{q-1}\|), \tag{2.133}$$

and solving for $\|Y_n^q - \tilde{Y}_n^q\|$, we get

$$
\left.
\begin{aligned}
\|Y_n^q - \tilde{Y}_n^q\| &\leq \frac{rm(m-1)\Delta^2 h(L_f + L_K h a_{max})\|Y_n^{q-1} - \tilde{Y}_n^{q-1}\|}{1 - rm\Delta^2 h(L_f + L_K h a_{max})} \\
&\leq hL_{GJ}\|Y_n^{q-1} - \tilde{Y}_n^{q-1}\|,
\end{aligned}
\right\} \tag{2.134}
$$

$q = 1, 2, \ldots$, and for sufficiently small h, so that $0 \leq hL_{GJ} < 1$.

In the case of Gauss-Seidel iterations, analogous calculations give

$$\|Y_n^q - \tilde{Y}_n^q\| \leq r\Delta^2 h(L_f + L_K h a_{max})\sum_{i=1}^{m}(i\|Y_n^q - \tilde{Y}_n{}^q\| + (m-i)\|Y_n^{q-1} - \tilde{Y}_n{}^{q-1}\|),$$

which becomes

$$
\begin{aligned}
\|Y_n^q - \tilde{Y}_n^q\| \quad \leq \quad & r\Delta^2 h(L_f + L_K h a_{max}) \\
& \left(\frac{m(m+1)}{2}\|Y_n^q - \tilde{Y}_n{}^q\| + \frac{(m-1)m}{2}\|Y_n^{q-1} - \tilde{Y}_n{}^{q-1}\|\right).
\end{aligned} \tag{2.135}
$$

Hence, solving for $\|Y_n^q - \tilde{Y}_n^q\|$, we obtain

$$\left.\begin{array}{rcl} \|Y_n^q - \tilde{Y}_n^q\| & \leq & \dfrac{\frac{r(m-1)m}{2}\triangle^2 h(L_f + L_K h a_{max})\|Y_n^{q-1} - \tilde{Y}_n^{q-1}\|}{1 - \frac{rm(m+1)}{2}\triangle^2 h(L_f + L_K h a_{max})} \\ & \leq & hL_{GS}\|Y_n^{q-1} - \tilde{Y}_n^{q-1}\| \end{array}\right\}, \qquad (2.136)$$

$q = 1, 2, \ldots$, and again, for sufficiently small h, so that $0 \leq hL_{GS} < 1$. Now, using the existence and uniqueness conclusions of Lemma 2.7, the proof is complete. □

Of course, in the case where we iterate a fixed number of times, or where we iterate until some norm of the difference between two successive approximations is less than some given tolerance, we define the limit as the last value calculated.

For the weakly singular case given by (2.107), the method is again given by (2.128), where for **Gauss-Jacobi** iterations

$$\begin{aligned} Y_n^q &= \sum_{i=1}^{m}\sum_{j=1}^{r} M_{(i-1)r+j}\left[f\left(t_n + c_j h, \eta(t_n) + h\{A_i(c_j)Y_n^q + \sum_{\substack{k=1 \\ k\neq i}}^{m} A_k(c_j)Y_n^{q-1}\}\right)\right. \\ &+ \left. h\int_0^{c_j}[(c_j-\tau)h]^{-\alpha}K\left(\eta(t_n) + h\{A_i(\tau)Y_n^q + \sum_{\substack{k=1 \\ k\neq i}}^{m} A_k(\tau)Y_n^{q-1}\}\right)d\tau\right] \\ &+ \sum_{i=1}^{m}\sum_{j=1}^{r} M_{(i-1)r+j} h \sum_{l=0}^{n-1}\int_0^1 [(t_n + c_j h) - (t_l + \tau h)]^{-\alpha} K(\eta(t_l + \tau h))d\tau, \quad (2.137) \end{aligned}$$

and for **Gauss-Seidel** iterations

$$\begin{aligned} Y_n^q &= \sum_{i=1}^{m}\sum_{j=1}^{r} M_{(i-1)r+j}\left[f\left(t_n + c_j h, \eta(t_n) + h\{\sum_{k=1}^{i} A_k(c_j)Y_n^q + \sum_{k=i+1}^{m} A_k(c_j)Y_n^{q-1}\}\right)\right. \\ &+ \left. h\int_0^{c_j}[(c_j-\tau)h]^{-\alpha}K\left(\eta(t_n) + h\{\sum_{k=1}^{i} A_k(\tau)Y_n^q + \sum_{k=i+1}^{m} A_k(\tau)Y_n^{q-1}\}\right)d\tau\right] \end{aligned}$$

$$+ \sum_{i=1}^{m}\sum_{j=1}^{r} M_{(i-1)r+j} h \sum_{l=0}^{n-1} \int_0^1 [(t_n + c_j h) - (t_l + \tau h)]^{-\alpha} K(\eta(t_l + \tau h)) d\tau, \quad (2.138)$$

$q = 1, 2, \ldots,$ $0 < \alpha < 1$ and Y_n^0 is given by (2.125).

The following corollary is immediate.

Corollary 2.2 *Consider the Volterra integro-differential equation with weakly singular kernel given by (2.107), where K and f satisfy uniform Lipschitz conditions (in y) with Lipschitz constants L_K and L_f, respectively, and $0 < \alpha < 1$. Then the time-point relaxation collocation method (2.128), where the iteration mode is Gauss-Jacobi or Gauss-Seidel, in which case the expression for Y_n^q is given by (2.137) or (2.138), respectively, converges to Y_n as $q \to \infty$ for sufficiently small $h > 0$. In this case, the two methods coincide and we obtain:*

$$\begin{aligned} Y_n &= \sum_{i=1}^{m}\sum_{j=1}^{r} M_{(i-1)r+j} [f(t_n + c_j h, \eta(t_n) + h[I_m \otimes \alpha^T(c_j)] \cdot Y_n) \\ &+ h \int_0^{c_j} [(c_j - \tau)h]^{-\alpha} K(\eta(t_n) + h[I_m \otimes \alpha^T(\tau)] \cdot Y_n) d\tau] \quad (2.139) \\ &+ \sum_{i=1}^{m}\sum_{j=1}^{r} M_{(i-1)r+j} h \sum_{l=0}^{n-1} \int_0^1 [(t_n + c_j h) - (t_l + \tau h)]^{-\alpha} K(\eta(t_l + \tau h)) d\tau, \end{aligned}$$

$n = 0, 1, \ldots, N-1$.

Proof:

The proof is the same as the proof of Theorem 2.12 except for the calculation in (2.132), which becomes

$$\int_0^{c_j} |a[(c_j - \tau)h]| d\tau = \int_0^{c_j} [(c_j - \tau)h]^{-\alpha} d\tau = \frac{h^{-\alpha} c_j^{1-\alpha}}{1-\alpha} \leq \frac{h^{-\alpha}}{1-\alpha},$$

where $0 < \alpha < 1$ and $j = 1, 2, \ldots, r$.

Then for Gauss-Jacobi iterations (2.134) becomes

$$\begin{aligned}\|Y_n^q - \tilde{Y}_n^q\| &\leq \frac{rm(m-1)\triangle^2 h\left(L_f + \frac{L_K h^{1-\alpha}}{1-\alpha}\right)\|Y_n^{q-1} - \tilde{Y}_n^{q-1}\|}{1 - rm\triangle^2 h\left(L_f + \frac{L_K h^{1-\alpha}}{1-\alpha}\right)} \\ &\leq hL_{GJ}\|Y_n^{q-1} - \tilde{Y}_n^{q-1}\|,\end{aligned}$$

$0 < \alpha < 1,\ q = 1, 2, \ldots$ and for $0 \leq hL_{GJ} < 1$.

Similarly, for Gauss-Seidel iterations (2.136) becomes:

$$\begin{aligned}\|Y_n^q - \tilde{Y}_n^q\| &\leq \frac{\frac{r(m-1)m}{2}\triangle^2 h\left(L_f + \frac{L_K h^{1-\alpha}}{1-\alpha}\right)\|Y_n^{q-1} - \tilde{Y}_n^{q-1}\|}{1 - \frac{rm(m+1)}{2}\triangle^2 h\left(L_f + \frac{L_K h^{1-\alpha}}{1-\alpha}\right)} \\ &\leq hL_{GS}\|Y_n^{q-1} - \tilde{Y}_n^{q-1}\|,\end{aligned}$$

$0 < \alpha < 1,\ q = 1, 2, \ldots$ and again, for sufficiently small h, so that $0 \leq hL_{GS} < 1$. Now, using the existence and uniqueness conclusions of Lemma 2.7, the proof is complete. □

2.3.5 Optimal order conditions

In the previous section, we showed that for sufficiently small h, the time-point relaxation collocation method (2.128), for each mode of iteration, (2.126), or (2.127) is well defined and converges to the same *unique* method given by (2.131), as $q \to \infty$. However, we do not know if the solution given by this method has any relation to the exact solution y of (2.106). We now tackle the question of convergence of these methods as $h \to 0$.

In particular, we will be concerned with the conditions necessary for optimal order of convergence at the nodes. See Section 1.4.2, where we discuss the idea of superconvergence at the nodes.

Recall that an r-stage collocation method has an order of global convergence of r, but if the collocation parameters $\{c_i\}$ are chosen in special ways, we can attain higher order at the nodes, provided the analytic solution is sufficiently regular. In particular, if the Gauss points are taken we attain the optimal nodal order $2r$. See Theorems 1.8 and 1.9, respectively. See Hairer at al (1993) for similar results for ODEs. In the next sections we consider the conditions necessary to attain this optimal nodal order, $2r$. Clearly, this analysis can be repeated for lower order methods, like the Radau II and the Lobatto methods.

We start out by considering the simplest case. Let us list all the assumptions:

- The iteration q is taken to ∞.
- The implicit algebraic equations in (2.126) and (2.127) can be solved exactly.
- The integrals in equations (2.126) and (2.127) can be found analytically.

In the coming sections we eliminate, in order, each of these assumptions. The final method will be the most realistic, and we will refer to it as the fully discretized time-point relaxation collocation method.

Alongside with these results we will also consider the VIDE with weakly singular kernel (2.107), whose time-point relaxation collocation method is given by (2.128), where the iteration mode is Gauss-Jacobi or Gauss-Seidel ((2.137) or (2.138), respectively). However, for polynomial spline collocation employing a uniform mesh, the global convergence is of order $1-\alpha$, where $0 < \alpha < 1$. See Section 1.4.4.

2.3.6 The Ideal Case

We take this simplest case and assume that the integrals and implicit algebraic equations in (2.126) and (2.127) can be solved exactly, and we take the limit as the number of iterations q goes to infinity. The idea in this section is to show that, if we take an r-stage collocation as the underlying method, we can attain an optimal nodal order of $2r$ (if the Gauss-points are taken as the collocation points). The following theorem and proof are given in Brunner (1984).

Theorem 2.13 *(Brunner 1984) Consider the Volterra integro-differential equation given by (2.106), and the regions* $D_f := [0,T] \times \Re^m$ *and* $D_K := S \times \Re^m$, *where* $0 < T < \infty$ *and* $i = 1, 2, \ldots, m$. *Assume the following regularity conditions:*

- $f \in C^{2r}(D_f)$,
- $K \in C^{2r}(D_K)$,
- $a \in C^{2r-1}([0,T])$.

Also, consider the time-point relaxation collocation method (2.128), where the iteration mode is Gauss-Jacobi or Gauss-Seidel, in which case the expressions for Y_n^q *are (2.126) and (2.127), respectively. Assume that the integrals and implicit algebraic equations in (2.126) and (2.127) can be solved exactly, and take the limit as the number of iterations q goes to infinity. Then the optimal nodal order is given by 2r, if the r collocation parameters* $\{c_i\}$ *are taken to be the Gauss points on (0,1).*

PROOF:

We begin by pointing out that this proof does not differ from the classical proof, since the TR method (2.128) simply becomes the classical method (2.129) in the limit as the number of iterations $q \to \infty$.

Define $e(t) := y(t) - \eta(t)$, where y refers to the exact solution of the problem, and η refers to the (collocation) approximation in the limit as $q \to \infty$. That is,

$$\begin{aligned}
\eta(t_n + \tau h) &= \eta(t_n) + h[I_m \otimes \alpha^T(\tau)] \cdot Y_n, \\
Y_n &= \sum_{i=1}^{m} \sum_{j=1}^{r} M_{(i-1)r+j} [f(t_n + c_j h, \eta(t_n) + h[I_m \otimes \alpha^T(c_j)] \cdot Y_n) \\
&+ h \int_0^{c_j} a[(c_j - \tau)h] K(\eta(t_n) + h[I_m \otimes \alpha^T(\tau)] \cdot Y_n) d\tau] \\
&+ \sum_{i=1}^{m} \sum_{j=1}^{r} M_{(i-1)r+j} h \sum_{l=0}^{n-1} \int_0^1 a[(t_n + c_j h) - (t_l + \tau h)] K(\eta(t_l + \tau h)) d\tau,
\end{aligned}$$

$\tau \in [0,1]$ and $n = 0, 1, \ldots, N-1$, where the subscript q on Y_n has been dropped to indicate that the limit as $q \to \infty$ has been taken. Recall that $\eta \in S_r^{(0)}(Z_N)$. Since η satisfies the problem (2.106) at the collocation points $T_N = \{t_n + c_j h; j = 1, 2, \ldots, r;\ n = 0, 1, \ldots, N-1\}$,

$$\eta'(t) = f(t, \eta) + \int_0^t a(t-s) K(\eta(s)) ds - \delta(t), \quad t \in [0, T], \tag{2.140}$$

with $\delta(0) = 0$. In fact, the *defect* δ vanishes on the set T_N of collocation points. Subtraction of (2.106) from (2.140) gives,

$$e'(t) = f(t, y) - f(t, \eta) + \int_0^t a(t-s)[K(y(s)) - K(\eta(s))] ds + \delta(t),$$

where $e(0) = 0$, since $\eta(0) = y(0) = y_0$. Since r is at least one, both f and K are differentiable with respect to y, (i.e., at least, $f \in C^2(D_f)$ and $K \in C^2(D_K)$), so we

can apply the mean value theorem, to convert this non-linear equation to the linear equation

$$e'(t) = A(t)e(t) + \int_0^t a(t-s)k(s)e(s)ds + \delta(t), \tag{2.141}$$

$e(0) = 0$. Here $A(t)$ is the $m \times m$ Jacobian matrix of f with respect to y, where each row is evaluated at a different mean value; that is

$$A(t) = \left(\frac{\partial f_i(t, \zeta_i)}{\partial y_j}\right),$$

where for each $t \in [0, T]$, the mean value $\zeta_i, i = 1, 2, \ldots, m$ is an internal point of the line segment in $\Re^m$ joining $y(t)$ to $\eta(t)$.

Similarly,

$$k(s) = \left(\frac{\partial K_i(\kappa_i)}{\partial y_j}\right),$$

where for each $s \in S$, the mean value $\kappa_i, i = 1, 2, \ldots, m$ is an internal point of the line segment in $\Re^m$ joining $y(s)$ to $\eta(s)$. See Lambert (1991).

The solution of the equation is given by Theorem 2.2, using the resolvent kernel R in (2.13),

$$e(t) = R(t, 0) \cdot e(0) + \int_0^t R(t, s) \cdot \delta(s)ds, \quad t \in [0, T],$$

and we recall that $e(0) = 0$.

At the nodal points

$$e(t_n) = h \sum_{k=0}^{n-1} \int_0^1 R(t_n, t_k + \tau h) \cdot \delta(t_k + \tau h)d\tau. \tag{2.142}$$

To complete the calculation, we evaluate the integrals using an interpolatory quadrature formula based on abscissas $\{t_k + d_j h\}$ where $0 \leq d_1 < \cdots < d_r \leq 1$, and weights

$\{w_j\}, (j = 1, 2, \ldots r)$. Then,

$$e(t_n) = h \sum_{k=0}^{n-1} \left(\sum_{j=1}^{r} w_j R(t_n, t_k + d_j h) \cdot \delta(t_k + d_j h) + E_{n,k} \right),$$

where $E_{n,k}$ denotes the error term associated with the quadrature formula. Choosing $d_j = c_j$ (the collocation parameters), we get $\delta(t_k + c_j h) = 0$ for $j = 1, 2, \ldots, r$, $k = 0, 1, \ldots n-1$, and the above equation becomes

$$e(t_n) = h \cdot \sum_{k=0}^{n-1} E_{n,k}.$$

Since this sum contains at most N terms, where $Nh = T < \infty$, the order of the nodal error $e(t_n)$ equals the order of the quadrature error. Choosing these collocation parameters equal to the Gauss points in (0,1), we establish optimal quadrature order $2r$.

Note, that it is here where we require the regularity conditions itemized in the assumptions of this theorem. In order to guarantee the optimal quadrature order $2r$, we assume that the integrand is sufficiently smooth (i.e., an element of the space $C^{2r}(\cdot)$), since the error formula contains derivatives of the integrand of this same order; see Brunner (1984) and Brunner and van der Houwen (1986). Therefore, let us look more closely at the integrand in (2.142), $R(t_n, t_k + \tau h) \cdot \delta(t_k + \tau h)$, where $k = 0, 1, \ldots n-1$, $n = 0, 1, \ldots, N-1$ and $\tau \in [0, 1]$, and show that it has smoothness of order $C^{2r}(\cdot)$.

Firstly, $\delta \in C^{2r}[(t_k, t_{k+1})]$ since by (2.140) it inherits the smoothness of f, a and K, and on (t_k, t_{k+1}), η is a polynomial. Observe that the resolvent kernel is derived from the linear equation (2.141). Since $A \in C^{2r-1}([0, T])$ and $a, k \in C^{2r-1}(S)$, by Lemma 2.1 this resolvent kernel $R \in C^{2r}(S)$. $\square$.

We now consider the VIDE with weakly singular kernel given by (2.107) and give the following theorem from Brunner (1986a).

Theorem 2.14 *(Brunner 1986a) Consider the Volterra integro-differential equation with weakly singular kernel given by (2.107), and the regions* $D_f := [0,T] \times \Re^m$ *and* $D_K := S \times \Re^m$, *where* $0 < T < \infty$ *and* $i = 1,2,\ldots,m$. *Assume the following regularity conditions:*

- $f \in C^r(D_f)$,
- $K \in C^r(D_K)$,

and assume that K *does not vanish identically. Also, consider the time-point relaxation collocation method given by (2.128), where the iteration mode is Gauss-Jacobi or Gauss-Seidel, in which case the expression for* Y_n^q *is given by (2.137) or (2.138), respectively. Assume that the integrals and implicit algebraic equations in (2.137) and (2.138) can be solved exactly, and we take the limit as the number of iterations* q *goes to infinity. Then the global convergence is of order* $1-\alpha$, *where* $0 < \alpha < 1$ *regardless of how the collocation parameters* $\{c_j : 0 \leq c_1 < c_2 < \cdots < c_r \leq 1\}$ *are chosen. That is, the error* $e(t) = y(t) - \eta(t)$ *satisfies*

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

2.3.7 The Effect of Iteration

Again, we assume that the integrals and the implicit algebraic equations in (2.126) and (2.127) can be treated exactly, but the number of iterations is finite. We can then consider the resulting effect on the underlying order of the method, which for

us is optimal at $2r$. In particular, we will be interested in establishing the minimum number of iterations necessary to regain the optimal order $2r$.

Let us define $e^q(t) := y(t) - \eta^q(t),\ q = 1, 2, \ldots$ and note that

$$\begin{aligned} e^q(t) &= (y(t) - \eta(t)) + (\eta(t) - \eta^q(t)) \\ &= e(t) + \epsilon^q(t), \end{aligned}$$

where we define $\epsilon^q(t) := \eta(t) - \eta^q(t)$. By the triangle inequality at the nodal points t_n

$$\|e^q(t_n)\| \leq \|e(t_n)\| + \|\epsilon^q(t_n)\|. \tag{2.143}$$

The previous theorem gives us the order of $e(t_n)$, so we need only consider the order of $\epsilon^q(t_n)$, since the order of $e^q(t_n)$ will be the minimum of these two orders. Fortunately, the same calculations that showed convergence of the iterations in Section 2.3.4 can be repeated to give the order of convergence of $\epsilon^q(t_n)$.

Theorem 2.15 *Consider the Volterra integro-differential equation (2.106), and the regions $D_f := [0, T] \times \Re^m$ and $D_K := S \times \Re^m$, where $0 < T < \infty$ and $i = 1, 2, \ldots, m$. Assume the following regularity conditions:*

- $f \in C^{2r}(D_f)$,
- $K \in C^{2r}(D_K)$,
- $a \in C^{2r-1}([0, T])$.

Also, the kernel K and the function f satisfy uniform Lipschitz conditions (in y) with Lipschitz constants L_K and L_f, respectively. Also, consider the time-point relaxation

collocation method (2.128), where the iteration mode is Gauss-Jacobi (2.126) or Gauss-Seidel (2.127), and assume that the integrals and implicit algebraic equations in (2.126) and (2.127) can be solved exactly. Then the optimal nodal order is given by $\nu = min\{2r, q+1\}$, *provided the* r *collocation parameters* $\{c_i\}$ *are taken to be the Gauss points in (0,1).*

PROOF:

In light of (2.143), we need only show that $\|\epsilon^q(t_n)\| = \mathcal{O}(h^{q+1})$. Therefore, subtracting (2.128) from (2.129),

$$\begin{aligned} \epsilon^q(t_n + \tau h) &= \eta(t_n + \tau h) - \eta^q(t_n + \tau h) \\ &= h[I_m \otimes \alpha^T(\tau)] \cdot [Y_n - Y_n^q], \end{aligned}$$

where $q = 1, 2, \ldots, \tau \in [0, 1]$, $n = 0, 1, \ldots, N-1$ and Y_n^0 given by (2.125), we now look more closely at the difference $Y_n - Y_n^q$.

We have already calculated this difference, see Theorem 2.12, and in particular, (2.134) and (2.136). Thus, for Gauss-Jacobi iterations,

$$\begin{aligned} \|Y_n - Y_n^q\| &\leq \frac{rm(m-1)\triangle^2 h(L_f + L_K h a_{max})\|Y_n - Y_n^{q-1}\|}{1 - rm\triangle^2 h(L_f + L_K h a_{max})} \\ &\leq hL_{GJ}\|Y_n - Y_n^{q-1}\|, \end{aligned}$$

$q = 1, 2, \ldots$ and Y_n^0 is given by (2.125). In the case of Gauss-Seidel iterations,

$$\begin{aligned} \|Y_n - Y_n^q\| &\leq \frac{\frac{r(m-1)m}{2}\triangle^2 h(L_f + L_K h a_{max})\|Y_n - Y_n^{q-1}\|}{1 - \frac{rm(m+1)}{2}\triangle^2 h(L_f + L_K h a_{max})} \\ &\leq hL_{GS}\|Y_n - Y_n^{q-1}\|, \end{aligned}$$

$q = 1, 2, \ldots$ and Y_n^0 is given by (2.125).

Applying these results recursively, for Gauss-Jacobi iterations

$$\|Y_n - Y_n^q\| \leq h^q L_{GJ}^q \|Y_n - Y_n^0\|, \tag{2.144}$$

$q = 1, 2, \ldots$ and Y_n^0 is given by (2.125).

For Gauss-Seidel iterations, the result is

$$\|Y_n - Y_n^q\| \leq h^q L_{GS}^q \|Y_n - Y_n^0\|. \tag{2.145}$$

So we obtain, for Gauss-Jacobi iterations

$$\begin{aligned} \|\epsilon^q(t_n + \tau h)\| &= \|h[I_m \otimes \alpha^T(\tau)] \cdot [Y_n - Y_n^q]\| \\ &\leq h\|I_m \otimes \alpha^T(\tau)\| \cdot [h^q L_{GJ}^q \|Y_n - Y_n^0\|] \leq \text{const} \cdot h^{q+1}, \end{aligned}$$

$q = 1, 2, \ldots,\ \tau \in [0, 1],\ n = 0, 1, \ldots, N-1$ and Y_n^0 is given by (2.125).

For Gauss-Seidel iterations,

$$\begin{aligned} \|\epsilon^q(t_n + \tau h)\| &= \|h[I_m \otimes \alpha^T(\tau)] \cdot [Y_n - Y_n^q]\| \\ &\leq h\|I_m \otimes \alpha^T(\tau)\| \cdot [h^q L_{GS}^q \|Y_n - Y_n^0\|] \leq \text{const} \cdot h^{q+1}, \end{aligned}$$

$q = 1, 2, \ldots,\ \tau \in [0, 1],\ n = 0, 1, \ldots, N-1$ and Y_n^0 is given by (2.125).

Hence,

$$\|\epsilon^q(t_n)\| = \mathcal{O}(h^{q+1}),$$

as $h \to 0^+$, and so

$$e^q(t_n) = \mathcal{O}(h^{\min\{2r, q+1\}}),$$

as $h \to 0^+$, if the collocation points are taken to be the Gauss points in (0,1). Therefore the nodal order is $\nu = min\{2r, q+1\}$. □

Remark:

Compare Theorem 2.15 for VIDEs to Theorem 1.10 for VIEs. Both results show that we obtain the order p^* of the underlying numerical method in $q = p^* - 1$ iterations. However, this depends on the initial solution (2.125) taken. In performing the numerical testing of Test Problem 2.2, for the Gauss-Jacobi case (only), I noticed that if I took Y_n^0 to be the null vector, I would need $q = p^*$ iterations to regain the order p^* of the underlying numerical method. Therefore, Theorem 1.4 for ODEs *is* compatible with Theorem 2.15. ⋄

To test and illustrate this result we consider the following simple linear two-dimensional test problem:

Test Problem 2.1 *Consider the following Volterra integro-differential system*

$$\begin{pmatrix} y_1'(t) \\ y_2'(t) \end{pmatrix} = \begin{pmatrix} -3 & 1 \\ 1 & -3 \end{pmatrix} \begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} + \int_0^t \begin{pmatrix} -2 & 2 \\ 2 & -2 \end{pmatrix} \begin{pmatrix} y_1(s) \\ y_2(s) \end{pmatrix} ds,$$

where $[y_{1,0}, y_{2,0}]^T$ *are given initial conditions.*

If we define

$$Y(t) := \begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix},$$

and differentiate, this linear VIDE becomes the following first order ODE

$$\begin{pmatrix} Y(t) \\ Y'(t) \end{pmatrix}' = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -2 & 2 & -3 & 1 \\ 2 & -2 & 1 & -3 \end{pmatrix} \begin{pmatrix} Y(t) \\ Y'(t) \end{pmatrix},$$

where

$$Y'(0) = \begin{pmatrix} -3 & 1 \\ 1 & -3 \end{pmatrix} Y(0).$$

Choosing the initial value

$$Y(0) = \begin{pmatrix} 2 \\ 3 \end{pmatrix},$$

we get the solution

$$Y(t) = \begin{pmatrix} e^{-2t}(2+t) \\ e^{-2t}(3-t) \end{pmatrix}.$$

See Edwards and Penney (1994).

I wrote a **C++** program to solve such a two dimensional linear VIDE using both one-point and two-point collocation (Gauss points).

In Section 2.5, see Tables 2.5, 2.6, 2.7 and 2.8 which summarize the results of these tests.

We now consider the VIDE with weakly singular kernel given by (2.107).

Corollary 2.3 *Consider the Volterra integro-differential equation with weakly singular kernel given by (2.107), and the regions* $D_f := [0,T] \times \Re^m$ *and* $D_K := S \times \Re^m$, *where* $0 < T < \infty$ *and* $i = 1, 2, \ldots, m$. *Assume the following regularity conditions:*

- $f \in C^r(D_f)$,
- $K \in C^r(D_K)$,

and assume that K *does not vanish identically. Also, consider the time-point relaxation collocation method given by (2.128), where the iteration mode is Gauss-Jacobi (2.137) or Gauss-Seidel (2.138). Assume that the integrals and implicit algebraic equations in (2.137) and (2.138) can be solved exactly. Then the order of global convergence is given by* $1 - \alpha$, *where* $0 < \alpha < 1$ *and we use* r*-point collocation.*

Proof:

This follows from Theorem 2.14 by applying the proof used in Theorem 2.15 and using the results given in the proof of Corollary 2.2. □

2.3.8 Stopping Error

Since the underlying method is a collocation method, which is an *implicit* continuous Volterra Runge-Kutta method, one must solve a system of implicit nonlinear algebraic equations to proceed with the method. That is, to find the Y_n^q using (2.126) or (2.127), we must again use an iterative method, which can be a Picard-type fixed-point iteration or some variant of the Newton method. Our job in this section will be to give the error associated with "stopping" this final iteration after a finite number of iterations. In particular, we will establish the minimum number of iterations necessary to regain the optimal order $2r$ (for the regular kernel case).

We assume that the integrals in (2.126) and (2.127) can be found analytically, leaving a later section to deal with the fully discretized cases, where we also approximate these integrals. We shall see that this will not result in a reduction of order if we use interpolatory quadrature, where the collocation Gauss points are taken as the quadrature abscissas.

As before we assume that the approximation η is already computed on $[0, t_n]$ and the approximation on the interval $[t_n, t_{n+1}]$ is given by

$$\eta^{q,s}(t_n + \tau h) = \eta(t_n) + h[I_m \otimes \alpha^T(\tau)] \cdot (Y_n^q)^s, \tag{2.146}$$

$q = 1, 2, \ldots$, $s = 1, 2, \ldots$, $\tau \in [0, 1]$ and $n = 0, 1, \ldots, N-1$.

Also, $\eta^{q,0}(t_n + \tau h) = \eta^{q-1}(t_n + \tau h)$ and $\eta^0(t_n + \tau h) = \eta(t_{n-1} + \tau h)$, for $n = 1, 2, \ldots, N-1$, and $\eta^0(\tau h) = y_0$, for $\tau \in [0, 1]$. The expressions for $(Y_n^q)^s$ depend on

the types of iteration modes used over both q and s, and we refer to these as the "inner" and "outer" iterations, respectively. Recall that Y_n^q is given by (2.126) or (2.127). Assume $q = 1, 2, \ldots$ is fixed and consider Picard fixed-point iteration over $s = 1, 2, \ldots$ to evaluate the Y_n^q. Let

$$\lim_{s \to \infty} (Y_n^q)^s = Y_n^q,$$

where $(Y_n^q)^0 = Y_n^{q-1}$ and Y_n^0 is given by (2.125).

Then for **Gauss-Jacobi** iterations we obtain

$$\begin{aligned}(Y_n^q)^s &= \sum_{i=1}^{m} \sum_{j=1}^{r} M_{(i-1)r+j} \\ &\left[f\left(t_n + c_j h, \eta(t_n) + h\{A_i(c_j)(Y_n^q)^{s-1} + \sum_{\substack{k=1 \\ k \neq i}}^{m} A_k(c_j) Y_n^{q-1}\} \right) \right. \\ &+ \left. h \int_0^{c_j} a[(c_j - \tau)h] K\left(\eta(t_n) + h\{A_i(\tau)(Y_n^q)^{s-1} + \sum_{\substack{k=1 \\ k \neq i}}^{m} A_k(\tau) Y_n^{q-1}\} \right) d\tau \right] \\ &+ \sum_{i=1}^{m} \sum_{j=1}^{r} M_{(i-1)r+j} h \sum_{l=0}^{n-1} \int_0^1 a[(t_n + c_j h) - (t_l + \tau h)] K(\eta(t_l + \tau h)) d\tau,\end{aligned} \tag{2.147}$$

$q = 1, 2, \ldots$ and $s = 1, 2, \ldots$.

The corresponding version for **Gauss-Seidel** iterations is

$$\begin{aligned}(Y_n^q)^s &= \sum_{i=1}^{m} \sum_{j=1}^{r} M_{(i-1)r+j} \\ &\left[f\left(t_n + c_j h, \eta(t_n) + h\{\sum_{k=1}^{i} A_k(c_j)(Y_n^q)^{s-1} + \sum_{k=i+1}^{m} A_k(c_j) Y_n^{q-1}\} \right) \right. \\ &+ \left. h \int_0^{c_j} a[(c_j - \tau)h] \right. \\ &\left. K\left(\eta(t_n) + h\{\sum_{k=1}^{i} A_k(\tau)(Y_n^q)^{s-1} + \sum_{k=i+1}^{m} A_k(\tau) Y_n^{q-1}\} \right) d\tau \right]\end{aligned} \tag{2.148}$$

$$+ \sum_{i=1}^{m}\sum_{j=1}^{r} M_{(i-1)r+j}h \sum_{l=0}^{n-1}\int_0^1 a[(t_n + c_j h) - (t_l + \tau h)]K(\eta(t_l + \tau h))d\tau,$$

$q = 1, 2, \ldots$ and $s = 1, 2, \ldots$.

To find the stopping error for iteration over s, we must look at $\|Y_n^q - (Y_n^q)^s\|$. Clearly the results of Section 2.3.4, in particular (2.133) and (2.135), can be applied. Therefore, for Gauss-Jacobi iteration mode, we find, for $q \in \{1, 2, \ldots\}$ fixed and $s = 1, 2, \ldots$,

$$\|Y_n^q - (Y_n^q)^s\| \leq rm\triangle^2 h(L_f + L_K h a_{max})\|Y_n^q - (Y_n^q)^{s-1}\|,$$

and for Gauss-Seidel iteration mode

$$\|Y_n^q - (Y_n^q)^s\| \leq r\triangle^2 h(L_f + L_K h a_{max})\left(\frac{m(m+1)}{2}\|Y_n^q - (Y_n^q)^{s-1}\|\right),$$

where we recall that

$$\triangle := \max\{\|M_{(i-1)r+j}\|, \|A_i(\tau)\| : i = 1, 2, \ldots, m;\ j = 1, 2, \ldots, r;\ \tau \in [0, 1]\},$$

and

$$a_{max} := \max\{|a(t)| : 0 \leq t \leq T\}.$$

Then applying this recursively, for both iteration modes, we obtain, for q fixed and $s = 1, 2, \ldots$,

$$\|Y_n^q - (Y_n^q)^s\| \leq C \cdot h^s \|Y_n^q - (Y_n^q)^0\|.$$

But $(Y_n^q)^0 = Y_n^{q-1}$, and from the results of Section 2.3.7

$$\begin{aligned}
\|Y_n^q - Y_n^{q-1}\| &= \|(Y_n^q - Y_n) + (Y_n - Y_n^{q-1})\| \\
&\leq \|Y_n^q - Y_n\| + \|Y_n - Y_n^{q-1}\| \\
&= 0(h^q) + 0(h^{q-1}) \\
&= 0(h^{q-1}).
\end{aligned}$$

Putting these two results together, we find

$$\|Y_n^q - (Y_n^q)^s\| = 0(h^{s+q-1}). \tag{2.149}$$

These results are now collected in the following theorem, which applies to both the Gauss-Jacobi and Gauss-Seidel iteration modes.

Theorem 2.16 *Consider the Volterra integro-differential equation given by (2.106), and the regions* $D_f := [0,T] \times \Re^m$ *and* $D_K := S \times \Re^m$, *where* $0 < T < \infty$ *and* $i = 1, 2, \ldots, m$. *Assume the following regularity conditions:*

- $f \in C^{2r}(D_f)$,
- $K \in C^{2r}(D_K)$,
- $a \in C^{2r-1}([0,T])$.

Also, the kernel K *and the function* f *satisfy uniform Lipschitz conditions (in* y*) with Lipschitz constants* L_K *and* L_f, *respectively. Also, consider the time-point relaxation collocation method (2.146), where the inner iteration mode is Gauss-Jacobi or Gauss-Seidel and the outer one is Picard fixed-point, in which case the expression for* $(Y_n^q)^s$ *is given by (2.147) or (2.148), respectively, and assume that the integrals in these expressions can be found exactly. Then the optimal nodal order is given by* $\nu = min\{2r, q+1, q+s\}$, *provided the* r *collocation parameters* $\{c_i\}$ *are taken to be the Gauss points in (0,1).*

Proof:

Let us define $e^{q,s}(t) := y(t) - \eta^{q,s}(t)$, $q = 1, 2, \ldots, s = 1, 2, \ldots$, and note that

$$\begin{aligned} e^{q,s}(t) &= (y(t) - \eta(t)) + (\eta(t) - \eta^q(t)) + (\eta^q(t) - \eta^{q,s}(t)) \\ &= e(t) + \epsilon^q(t) + \epsilon^{q,s}(t), \end{aligned}$$

where we define $\epsilon^{q,s}(t) := \eta^q(t) - \eta^{q,s}(t)$. By setting $t = t_n$ and using the triangle inequality, we establish

$$\|e^{q,s}(t_n)\| \leq \|e(t_n)\| + \|\epsilon^q(t_n)\| + \|\epsilon^{q,s}(t_n)\|. \tag{2.150}$$

The previous two theorems, Theorem 2.13 and Theorem 2.15, give us the order of $e(t_n)$ and $\epsilon^q(t_n)$, respectively, so we only need consider the order of $\epsilon^{q,s}(t_n)$, since the order of $e^{q,s}(t_n)$, will be the minimum of these three orders.

Subtracting (2.146) from (2.128), we find

$$\begin{aligned} \|\epsilon^{q,s}(t_n + \tau h)\| &= \|h[I_m \otimes \alpha^T(\tau)] \cdot [Y_n^q - (Y_n^q)^s]\| \\ &\leq h\|I_m \otimes \alpha^T(\tau)\| \cdot \|Y_n^q - (Y_n^q)^s\|, \end{aligned}$$

$q = 1, 2, \ldots$, $s = 1, 2, \ldots$, $\tau \in [0, 1]$ and $n = 0, 1, \ldots, N-1$. Recalling equation (2.149),

$$\|Y_n^q - (Y_n^q)^s\| = 0(h^{s+q-1}),$$

so

$$\epsilon^{q,s}(t_n) = \mathcal{O}(h^{s+q}),$$

and therefore

$$e^{q,s}(t_n) = \mathcal{O}(h^{\min\{2r, q+1, q+s\}}),$$

as $h \to 0^+$, if the collocation points are taken to be the Gauss points in (0,1). □

To test and illustrate this result we consider the following simple nonlinear two-dimensional test problem:

Test Problem 2.2 *Consider the following Volterra integro-differential system*

$$\begin{pmatrix} y_1'(t) \\ y_2'(t) \end{pmatrix} = \begin{pmatrix} y_1(t) - y_1(t)y_2(t) \\ 2y_1(t)y_2(t) - 2y_2(t) \end{pmatrix} + \int_0^t (t-s) \begin{pmatrix} y_1(s)y_2(s) \\ y_1(s)y_2(s) \end{pmatrix} ds + \begin{pmatrix} g_1(t) \\ g_2(t) \end{pmatrix},$$

where $[y_{1,0}, y_{2,0}]^T$ *are given initial conditions.*

The non-homogeneous terms are

$$\begin{aligned} g_1(t) &= e^{-t}(1-2t) + \frac{2e^{-3t}(12t-1)}{27} - \frac{3t-2}{27}, \\ g_2(t) &= \frac{-e^{-3t}(57t+2)}{27} - \frac{3t-2}{27}. \end{aligned}$$

The solution corresponding to the initial values,

$$\begin{aligned} y_{1,0} &= 0, \\ y_{2,0} &= 1, \end{aligned}$$

is given by

$$\begin{aligned} y_1(t) &= te^{-t}, \\ y_2(t) &= e^{-2t}. \end{aligned}$$

I wrote a C++ program to solve this two-dimensional nonlinear VIDE using both one-point and two-point collocation (Gauss points). I applied Picard iteration to solve the implicit algebraic equations and I used one and two point Gauss quadrature, respectively, to evaluate the integrals. Although we postpone the discussion of the approximation of the integrals that occur in these methods until Section 2.4, we can say that it does not result in a reduction of the order.

In Section 2.5, Tables 2.9, 2.10 and 2.11 summarize the results of these tests, which verify Theorem 2.16.

Weakly Singular Case

Consider the VIDE with weakly singular kernel given by (2.107). Then the time-point relaxation collocation method is given by (2.146), where for **Gauss-Jacobi** iterations we establish

$$
\begin{aligned}
(Y_n^q)^s &= \sum_{i=1}^{m}\sum_{j=1}^{r} M_{(i-1)r+j} \qquad (2.151)\\
&\left[f\left(t_n + c_j h, \eta(t_n) + h\{A_i(c_j)(Y_n^q)^{s-1} + \sum_{\substack{k=1\\k\neq i}}^{m} A_k(c_j) Y_n^{q-1}\} \right) \right.\\
&+ \left. h\int_0^{c_j} [(c_j-\tau)h]^{-\alpha} K\left(\eta(t_n) + h\{A_i(\tau)(Y_n^q)^{s-1} + \sum_{\substack{k=1\\k\neq i}}^{m} A_k(\tau) Y_n^{q-1}\} \right) d\tau \right]\\
&+ \sum_{i=1}^{m}\sum_{j=1}^{r} M_{(i-1)r+j} h \sum_{l=0}^{n-1} \int_0^1 [(t_n + c_j h) - (t_l + \tau h)]^{-\alpha} K(\eta(t_l + \tau h)) d\tau,
\end{aligned}
$$

$0 < \alpha < 1$, $q = 1, 2, \ldots$ and $s = 1, 2, \ldots$.

The corresponding version for **Gauss-Seidel** iterations is

$$
(Y_n^q)^s = \sum_{i=1}^{m}\sum_{j=1}^{r} M_{(i-1)r+j} \qquad (2.152)
$$

$$
\begin{aligned}
&\left[f\left(t_n + c_j h, \eta(t_n) + h\{\sum_{k=1}^{i} A_k(c_j)(Y_n^q)^{s-1} + \sum_{k=i+1}^{m} A_k(c_j) Y_n^{q-1}\} \right) \right. \\
+ \; & h \int_0^{c_j} [(c_j - \tau)h]^{-\alpha} \\
& \left. K\left(\eta(t_n) + h\{\sum_{k=1}^{i} A_k(\tau)(Y_n^q)^{s-1} + \sum_{k=i+1}^{m} A_k(\tau) Y_n^{q-1}\} \right) d\tau \right] \\
+ \; & \sum_{i=1}^{m} \sum_{j=1}^{r} M_{(i-1)r+j} h \sum_{l=0}^{n-1} \int_0^1 [(t_n + c_j h) - (t_l + \tau h)]^{-\alpha} K(\eta(t_l + \tau h)) d\tau,
\end{aligned}
$$

$0 < \alpha < 1$, $q = 1, 2, \ldots$ and $s = 1, 2, \ldots$.

Corollary 2.4 *Consider the Volterra integro-differential equation with weakly singular kernel given by (2.107), and the regions* $D_f := [0, T] \times \Re^m$ *and* $D_K := S \times \Re^m$, *where* $0 < T < \infty$ *and* $i = 1, 2, \ldots, m$. *Assume the following regularity conditions:*

- $f \in C^r(D_f)$,
- $K \in C^r(D_K)$,

and assume that K *does not vanish identically. Also, consider the time-point relaxation collocation method given by (2.146), where the inner iteration mode is Gauss-Jacobi or Gauss-Seidel, and the outer is Picard fixed-point, in which case the expression for* $(Y_n^q)^s$ *is given by (2.151) or (2.152), respectively and the integrals in these expressions can be solved exactly. Then the order of global convergence is given by* $1 - \alpha$, *where* $0 < \alpha < 1$ *and we use* r*-point collocation.*

Proof:

This follows from Theorem 2.14 by applying the proof used in Theorem 2.16 and using the results given in the proof of Corollary 2.2. □.

To test and illustrate this result we consider the following simple linear two-dimensional test problem with weakly singular kernel:

Test Problem 2.3 *Consider the following Volterra integro-differential system*

$$\begin{pmatrix} y_1'(t) \\ y_2'(t) \end{pmatrix} = \begin{pmatrix} -1 & -1 \\ -1 & -2 \end{pmatrix} \begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} + \int_0^t (t-s)^{-\alpha} \begin{pmatrix} -1 & -1 \\ -1 & -2 \end{pmatrix} \begin{pmatrix} y_1(s) \\ y_2(s) \end{pmatrix} ds + \begin{pmatrix} g_1(t) \\ g_2(t) \end{pmatrix},$$

$0 < \alpha < 1$, *where* $[y_{1,0}, y_{2,0}]^T$ *are given initial conditions.*

We assume the solution:

$$\begin{aligned} y_1(t) &= t^{4-\alpha}, \\ y_2(t) &= 2t^{4-\alpha}, \end{aligned} \tag{2.153}$$

subject to the initial conditions

$$\begin{aligned} y_{1,0} &= 0, \\ y_{2,0} &= 0, \end{aligned}$$

where $0 < \alpha < 1$. The non-homogeneous terms are then given by

$$\begin{aligned} g_1(t) &= (4-\alpha)t^{3-\alpha} + 3t^{4-\alpha} + \frac{3\Gamma(1-\alpha)\Gamma(5-\alpha)}{\Gamma(6-2\alpha)} t^{5-2\alpha}, \\ g_2(t) &= 2(4-\alpha)t^{3-\alpha} + 5t^{4-\alpha} + \frac{5\Gamma(1-\alpha)\Gamma(5-\alpha)}{\Gamma(6-2\alpha)} t^{5-2\alpha}, \end{aligned}$$

for $0 < \alpha < 1$.

Note that this problem is a two-dimensional version of the problem given in Example 2.3. However, by Theorem 2.6, the solution (2.153) is not a typical solution. For comparison, let us assume the following solution to Test Problem 2.3

$$\begin{aligned} y_1(t) &= t^{2-\alpha}, \\ y_2(t) &= 2t^{2-\alpha}, \end{aligned} \tag{2.154}$$

subject to the same initial conditions as (2.153),

$$\begin{aligned} y_{1,0} &= 0, \\ y_{2,0} &= 0, \end{aligned}$$

where $0 < \alpha < 1$. The non-homogeneous terms are now given by

$$\begin{aligned} g_1(t) &= (2-\alpha)t^{1-\alpha} + 3t^{2-\alpha} + \frac{3\Gamma(1-\alpha)\Gamma(3-\alpha)}{\Gamma(4-2\alpha)}t^{3-2\alpha}, \\ g_2(t) &= 2(2-\alpha)t^{1-\alpha} + 5t^{2-\alpha} + \frac{5\Gamma(1-\alpha)\Gamma(3-\alpha)}{\Gamma(4-2\alpha)}t^{3-2\alpha}, \end{aligned}$$

for $0 < \alpha < 1$.

I wrote ©MATLAB software to solve this test problem, using both of these solutions. We used two-point collocation (Gauss-points), Picard iteration (one iteration) to solve the implicit algebraic equations and two point Gauss quadrature to solve the integrals. The inner iteration was Gauss-Jacobi with $q = 3$ iterations, so for both solutions (2.153) and (2.154), we expect an order of global convergence of

$1-\alpha$, where $0 < \alpha < 1$. See Corollary 2.4 and the "Remark" at the end of Section 2.4.

The results of these tests are collected in Tables 2.12 and 2.13, Section 2.5.

2.3.9 Newton Iteration

In addition to the simple Picard iteration, we consider the Newton type methods to solve the implicit algebraic equations in the method. To apply the Newton type iteration formulas, we let (2.126) and (2.127) be written $Y_n^q = F(Y_n^q, Y_n^{q-1})$, and define

$$P(Y_n^q, Y_n^{q-1}) = Y_n^q - F(Y_n^q, Y_n^{q-1}) = 0, \tag{2.155}$$

$q \in \{1, 2, \ldots\}$ (fixed). The form of F and therefore P is clear and depends on which inner iteration scheme, Gauss-Jacobi or Gauss-Seidel we are considering. We now apply Newton's methods to the solution of (2.155), which in turn will provide a solution to (2.126) or (2.127).

The Newton method for (2.155) with initial value $(Y_n^q)^0 = (Y_n^{q-1})$, where $q = 1, 2, \ldots$ (fixed) and Y_n^0 is given by (2.125), is given by

$$(Y_n^q)^s = (Y_n^q)^{s-1} - \left\{P'((Y_n^q)^{s-1}, Y_n^{q-1})\right\}^{-1} P((Y_n^q)^{s-1}, Y_n^{q-1}), \tag{2.156}$$

where $s = 1, 2, \ldots$ and P' is the Jacobian of P. Let us look more closely at the form of P'. Firstly,

$$P'(Y_n^q, Y_n^{q-1}) = I_{rm} - \frac{\partial F(Y_n^q, Y_n^{q-1})}{\partial Y_n^q},$$

where I_{rm} is the rm-dimensional identity matrix. We subsequently must look at the form of $\dfrac{\partial F(Y_n^q, Y_n^{q-1})}{\partial Y_n^q}$. In order to justify taking these derivatives and moving the

derivative "inside" the integral sign (for the integral component) we must assume sufficient regularity conditions for f and K. These are given in Theorem 2.18.

By the linearity property of derivatives, we determine for the Gauss-Jacobi mode of iteration

$$\frac{\partial F(Y_n^q, Y_n^{q-1})}{\partial Y_n^q} = h \cdot diag(\{J(f_i, K_i; \eta_i)(Y_n^q, Y_n^{q-1})\}_{i=1}^m),$$

where $diag(\{U_i\}_{i=1}^m)$ is the rm-dimensional square block matrix with diagonal blocks U_i, $i = 1, 2, \ldots, m$, that is

$$diag(\{U_i\}_{i=1}^m) = \begin{pmatrix} U_1 & \cdots & 0 & \cdots & 0 \\ \vdots & & \vdots & & \vdots \\ 0 & \cdots & U_i & \cdots & 0 \\ \vdots & & \vdots & & \vdots \\ 0 & \cdots & 0 & \cdots & U_m \end{pmatrix},$$

where $U_i = J(f_i, K_i; \eta_i)(Y_n^q, Y_n^{q-1})$ is the matrix:

$$J(f_i, K_i; \eta_i)(Y_n^q, Y_n^{q-1}) = (j_{uv}),$$

where each element j_{uv} is given by

$$\begin{aligned} j_{uv} &= \alpha_v(c_u)\frac{\partial f_i}{\partial \eta_i}(t_n + c_u h, \cdot) + h\int_0^{c_u} a[(c_u - \tau)h]\alpha_v(\tau)\frac{\partial K_i}{\partial \eta_i}(\cdot)d\tau \\ &:= \alpha_v(c_u)f_{ii}(c_u) + hI_{ii}^v(c_u), \end{aligned}$$

$i = 1, 2, \ldots, m$, $u = 1, 2, \ldots, r$, $v = 1, 2, \ldots, r$ and $\tau \in [0, 1]$. To get a picture of these block matrices, let us write out the components for the case $r = 2$. For $i = 1, 2, \ldots, m$,

$$U_i = \begin{pmatrix} \alpha_1(c_1)f_{ii}(c_1) + hI_{ii}^1(c_1) & \alpha_2(c_1)f_{ii}(c_1) + hI_{ii}^2(c_1) \\ \alpha_1(c_2)f_{ii}(c_2) + hI_{ii}^1(c_2) & \alpha_2(c_2)f_{ii}(c_2) + hI_{ii}^2(c_2) \end{pmatrix}.$$

Therefore

$$P'((Y_n^q)^{s-1}, Y_n^{q-1}) = I_{rm} - h \cdot diag(\{J(f_i, K_i; \eta_i)((Y_n^q)^{s-1}, Y_n^{q-1})\}_{i=1}^m), \qquad (2.157)$$

where both partials of f and K are evaluated at $(Y_n^q)^{s-1}$ and Y_n^{q-1}.

For the Gauss-Seidel mode of iteration, $\dfrac{\partial F(Y_n^q, Y_n^{q-1})}{\partial Y_n^q}$ becomes the following lower triangular block matrix of partial derivatives,

$$\frac{\partial F(Y_n^q, Y_n^{q-1})}{\partial Y_n^q} = h \cdot L(\{J(f_i, K_i; \eta_j)(Y_n^q, Y_n^{q-1})\}_{j \le i=1}^m),$$

where $L(\{U_{ij}\}_{j \le i=1}^m)$ is the rm-dimensional square block matrix with blocks $U_{i,j}$, $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, i$, that is

$$L(\{U_{i,j}\}_{j \le i=1}^m) = \begin{pmatrix} U_{11} & \cdots & 0 & \cdots & 0 \\ \vdots & & \vdots & & \vdots \\ U_{i1} & \cdots & U_{i,i} & \cdots & 0 \\ \vdots & & \vdots & & \vdots \\ U_{m1} & \cdots & U_{mi} & \cdots & U_{mm} \end{pmatrix},$$

where $U_{i,j} = J(f_i, K_i; \eta_j)(Y_n^q, Y_n^{q-1})$ is the matrix

$$J(f_i, K_i; \eta_j)(Y_n^q, Y_n^{q-1}) = (j_{uv}),$$

where each element j_{uv} is given by

$$\begin{aligned} j_{uv} &= \alpha_v(c_u)\frac{\partial f_i}{\partial \eta_j}(t_n + c_u h, \cdot) + h\int_0^{c_u} a[(c_u - \tau)h]\alpha_v(\tau)\frac{\partial K_i}{\partial \eta_j}(\cdot)d\tau, \\ &:= \alpha_v(c_u)f_{ij}(c_u) + hI_{ij}^v(c_u), \end{aligned}$$

$i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, i$, $u = 1, 2, \ldots, r$, $v = 1, 2, \ldots, r$ and $\tau \in [0, 1]$.

Again we write out the components for the case $r = 2$. For $j \le i = 1, 2, \ldots, m$,

$$U_{i,j} = \begin{pmatrix} \alpha_1(c_1)f_{ij}(c_1) + hI_{ij}^1(c_1) & \alpha_2(c_1)f_{ij}(c_1) + hI_{ij}^2(c_1) \\ \alpha_1(c_2)f_{ij}(c_2) + hI_{ij}^1(c_2) & \alpha_2(c_2)f_{ij}(c_2) + hI_{ij}^2(c_2) \end{pmatrix}.$$

Therefore,

$$P'((Y_n^q)^{s-1}, Y_n^{q-1}) = I_{rm} - h \cdot L(\{J(f_i, K_i; \eta_j)((Y_n^q)^{s-1}, Y_n^{q-1})\}_{j \le i=1}^{m}), \qquad (2.158)$$

where both partial derivatives of f and K are evaluated at $(Y_n^q)^{s-1}$ and Y_n^{q-1}.

The stopping error for Newton's method is now derived by a straightforward extension of the results given in Sugiura and Torii (1991), which were derived by using results originally due to Kantorovich and Akilov (1982). We now give this theorem, adapting it for our purposes. We refer the reader to Theorem 2.18 for the assumed regularity conditions.

Theorem 2.17 *(Kantorovich and Akilov 1982) Let* $(Y_n^q)^0 \in \Re^{rm}$ *and* $B(e) := \{Y_n^q : \|Y_n^q - (Y_n^q)^0\| \le e\}$ *be the closed ball around* $(Y_n^q)^0$. *Suppose:*

$$\Gamma = \left[P'((Y_n^q)^0)\right]^{-1} \text{ exists}, \qquad (2.159)$$

$$\|\Gamma P((Y_n^q)^0)\| \le \xi, \qquad (2.160)$$

$$\|\Gamma P''(Y_n^q)\| \le \Upsilon, \quad Y_n^q \in B(e). \qquad (2.161)$$

If

$$\rho = \Upsilon\xi < \frac{1}{2}, \quad \frac{1-\sqrt{1-2\rho}}{\rho}\xi = e_0 \le e \le e_1 = \frac{1+\sqrt{1-2\rho}}{\rho}\xi, \qquad (2.162)$$

then (2.155) has a unique solution Y_n^q *in* $B(e)$ *and*

$$\|(Y_n^q)^s - Y_n^q\| \le \frac{(2\rho)^{2^s}\xi}{2^s\rho}, \quad s = 0, 1, \ldots. \qquad (2.163)$$

Applying this theorem to our problem, for Gauss-Jacobi iteration,

$$\begin{aligned}
[P'((Y_n^q)^0, Y_n^{q-1})]^{-1} &= [P'(Y_n^{q-1}, Y_n^{q-1})]^{-1} \\
&= [I_{rm} - h \cdot diag(\{J(f_i, K_i; \eta_i)(Y_n^{q-1}, Y_n^{q-1})\}_{i=1}^m)]^{-1} \\
&\to I_{rm},
\end{aligned}$$

as $h \to 0$, which means that the first condition of the theorem, given by (2.159), is fulfilled, since Γ will exist for sufficiently small h. Now,

$$\begin{aligned}
\|\Gamma P''(Y_n^q, Y_n^{q-1})\| &\leq h\|\Gamma\|\| \left(diag(\{J(f_i, K_i; \eta_i)(Y_n^q, Y_n^{q-1})\}_{i=1}^m)\right)' \| \\
&\leq \Upsilon,
\end{aligned}$$

and since both f and K are at least twice continuously differentiable, (see Theorem 2.18), hence bounded, this establishes the third condition, given by (2.161), and allows us the conclude that $\Upsilon = \mathcal{O}(h)$.

Similarly for Gauss-Seidel iteration,

$$\begin{aligned}
[P'((Y_n^q)^0, Y_n^{q-1})]^{-1} &= [P'(Y_n^{q-1}, Y_n^{q-1})]^{-1} \\
&= [I_{rm} - h \cdot L(\{J(f_i, K_i; \eta_j)(Y_n^{q-1}, Y_n^{q-1})\}_{j \leq i=1}^m)]^{-1} \\
&\to I_{rm},
\end{aligned}$$

as $h \to 0$, which means that the first condition of the theorem, given by (2.159), is fulfilled, since Γ will exist for sufficiently small h.

$$\begin{aligned}
\|\Gamma P''(Y_n^q, Y_n^{q-1})\| &\leq h\|\Gamma\|\| \left(L(\{J(f_i, K_i; \eta_j)(Y_n^q, Y_n^{q-1})\}_{j \leq i=1}^m)\right)' \| \\
&\leq \Upsilon.
\end{aligned}$$

This establishes the third condition, given by (2.161), and allows us the conclude that $\Upsilon = \mathcal{O}(h)$. This theorem can now be applied directly to our problem for *both* iteration schemes.

Theorem 2.18 *Consider the Volterra integro-differential equation given by (2.106), and the regions* $D_f := [0,T] \times \Re^m$ *and* $D_K := S \times \Re^m$, *where* $0 < T < \infty$ *and* $i = 1, 2, \ldots, m$. *Assume the following regularity conditions:*

- $f \in C^{2r}(D_f)$,
- $K \in C^{2r}(D_K)$,
- $a \in C^{2r-1}([0,T])$.

Also, the kernel K *and the function* f *satisfy uniform Lipschitz conditions (in* y*) with Lipschitz constants* L_K *and* L_f, *respectively. Also, consider the time-point relaxation collocation method (2.128), where the inner iteration mode is Gauss-Jacobi or Gauss-Seidel and the outer one is given by the Newton method (2.156), in which case the expression for* $P(Y_n^q, Y_n^{q-1})$ *is given by (2.155), the expression for* $F(Y_n^q, Y_n^{q-1})$ *is given by (2.126) or (2.127), and the expression for* $P'(Y_n^q, Y_n^{q-1})$ *is given by (2.157) or (2.158), respectively. Also assume that the integrals in these expressions can be found exactly. Then the optimal nodal order is given by* $\nu = min\{2r, q+1, 2^s q\}$, *provided the* r *collocation parameters* $\{c_i\}$ *are the* r *Gauss points in (0,1).*

Proof:

For $q \in \{1, 2, \ldots\}$ fixed and $n = 0, 1, \ldots, N-1$,

$$P((Y_n^q)^0, Y_n^{q-1}) = P(Y_n^{q-1}, Y_n^{q-1}) = Y_n^{q-1} - F(Y_n^{q-1}, Y_n^{q-1}),$$

where the expressions for F is given by (2.126) or (2.127). We consider three cases:

- Case 1: For $q = 1$ and $n = 0$,

$$Y_0^0 - F(Y_0^0, Y_0^0) = F(0,0),$$

using (2.125), where the expression for F is given by (2.126) or (2.127). By the assumed regularity of the problem, $\|F(0,0)\|$ is bounded.

- Case 2: For $q = 1$ and $n = 1, 2, \ldots, N-1$,

$$Y_n^0 - F(Y_n^0, Y_n^0) = Y_{n-1} - F(Y_{n-1}, Y_{n-1}) = 0,$$

using (2.125) and Theorem 2.12, in which case $Y_{n-1} = F(Y_{n-1}, Y_{n-1})$; see equation (2.131). Note that this assumes limits as $q \to \infty$.

- Case 3: For $q = 2, 3, \ldots$ and $n = 0, 1, \ldots, N-1$,

$$Y_n^{q-1} - F(Y_n^{q-1}, Y_n^{q-1}) = F(Y_n^{q-1}, Y_n^{q-2}) - F(Y_n^{q-1}, Y_n^{q-1}).$$

Similar calculations that led to (2.133) and (2.135), give us

$$\begin{aligned} \|F(Y_n^{q-1}, Y_n^{q-2}) - F(Y_n^{q-1}, Y_n^{q-1})\| &\leq rm(m-1)\triangle^2 h(L_f + L_K h a_{max}) \\ &\cdot \|Y_n^{q-2} - Y_n^{q-1}\|, \end{aligned}$$

for Gauss-Jacobi iterations and

$$\begin{aligned} \|F(Y_n^{q-1}, Y_n^{q-2}) - F(Y_n^{q-1}, Y_n^{q-1})\| &\leq r\frac{m(m-1)}{2}\triangle^2 h(L_f + L_K h a_{max}) \\ &\cdot \|Y_n^{q-2} - Y_n^{q-1}\|, \end{aligned}$$

for Gauss-Seidel iterations.

Since, in both cases,

$$\|Y_n^{q-2} - Y_n^{q-1}\| \leq \|Y_n^{q-2} - Y_n\| + \|Y_n - Y_n^{q-1}\| = \mathcal{O}(h^{q-2}),$$

by (2.144) and (2.145), we conclude that

$$\|F(Y_n^{q-1}, Y_n^{q-2}) - F(Y_n^{q-1}, Y_n^{q-1})\| = \mathcal{O}(h^{q-1}).$$

Therefore in cases 1 and 2, for $q = 1,\ n = 0, 1, \ldots, N-1$ we have $P((Y_n^q)^0, Y_n^{q-1}) = \mathcal{O}(h^0)$. See Section 1.3.2, Definition 1.2. Combining with Case three we conclude that for $q = 1, 2, \ldots,\ n = 0, 1, \ldots, N-1$, the starting error satisfies

$$P((Y_n^q)^0, Y_n^{q-1}) = \mathcal{O}(h^{q-1}).$$

This together with the fact that $\Gamma \longrightarrow I_{rm}$, as $h \to 0$, implies $\|\Gamma P((Y_n^q)^0, Y_n^{q-1})\| = \mathcal{O}(h^{q-1})$, so we can take $\xi = \mathcal{O}(h^{q-1})$ in Theorem 2.17. Also, since above we showed that $\Upsilon = \mathcal{O}(h)$, we get $\rho = \Upsilon\xi = \mathcal{O}(h^q)$ in Theorem 2.17.

Consider the expressions given in (2.162). For sufficiently small h, $\rho < \frac{1}{2}$, and hence

$$\frac{1 - \sqrt{1 - 2\rho}}{\rho} \to 1,$$

as $h \to 0$. Thus e_0 is bounded ($q = 1$) or $e_0 \to 0$ ($q > 1$) as $h \to 0$. Also,

$$\left|\frac{1 + \sqrt{1 - 2\rho}}{\rho}\xi\right| \leq 2\frac{\xi}{\rho},$$

and so $e_1 \to \infty$, as $h \to 0$.

Therefore, by Theorem 2.17, we conclude that (2.155) has a unique solution Y_n^q in $B(e)$, for each $q = 1, 2, \ldots$, and

$$\|(Y_n^q)^s - Y_n^q\| \leq \frac{(2\rho)^{2^s}\xi}{2^s\rho} = \mathcal{O}(h^{2^s q - 1}), \quad s = 0, 1, \ldots.$$

The proof then follows by combining these results with those of Section 2.3.8, in particular the proof of Theorem 2.16, since

$$\begin{aligned} \|\epsilon^{q,s}(t_n + \tau h)\| &= \|h[I_m \otimes \alpha^T(\tau)] \cdot [(Y_n^q)^s - Y_n^q]\| \\ &\leq h\|I_m \otimes \alpha^T(\tau)\| \cdot \|(Y_n^q)^s - Y_n^q\|, \end{aligned}$$

$q = 1, 2, \ldots$, $s = 1, 2, \ldots$, $\tau \in [0, 1]$, $(Y_n^q)^0 = Y_n^{q-1}$ for $n = 0, 1, \ldots, N-1$, and Y_n^0 is given by (2.125). □

2.3.10 Modified Newton Method

Evaluation of the Jacobian P' can be very expensive, especially if it must be continually updated. Therefore, one often uses the *modified* Newton method, in which the Jacobian is evaluated once, and this value is used in all subsequent calculations. We therefore establish new stopping errors and in general, a reduction in order.

The modified Newton method for (2.155) with initial value $(Y_n^q)^0 = Y_n^{q-1}$, where $q \in \{1, 2, \ldots\}$ (fixed) and Y_n^0 is given by (2.125), is given by

$$(Y_n^q)^s = (Y_n^q)^{s-1} - \left\{P'((Y_n^q)^0, Y_n^{q-1})\right\}^{-1} P((Y_n^q)^{s-1}, Y_n^{q-1}), \tag{2.164}$$

where $s = 1, 2, \ldots$, and

$$P'((Y_n^q)^0, Y_n^{q-1}) = I_{rm} - h \cdot diag(\{J(f_i, K_i; \eta_i)(Y_n^{q-1}, Y_n^{q-1})\}_{i=1}^m), \tag{2.165}$$

for Gauss-Jacobi iteration and

$$P'((Y_n^q)^0, Y_n^{q-1}) = I_{rm} - h \cdot L(\{J(f_i, K_i; \eta_j)(Y_n^{q-1}, Y_n^{q-1})\}_{j \leq i=1}^m), \tag{2.166}$$

for Gauss-Seidel iteration. See Section 2.3.9.

The following theorem given by Kantorovich and Akilov (1982), can be used to give order conditions for the modified Newton method.

Theorem 2.19 *(Kantorovich and Akilov 1982) Let* $(Y_n^q)^0 \in \Re^{rm}$ *and* $B(e) := \{Y_n^q : \|Y_n^q - (Y_n^q)^0\| \leq e\}$ *be the closed ball around* $(Y_n^q)^0$. *Suppose:*

$$\Gamma = \left[P'((Y_n^q)^0)\right]^{-1} \text{ exists,}$$

$$\|\Gamma P((Y_n^q)^0)\| \leq \xi,$$

$$\|\Gamma P''(Y_n^q)\| \leq \Upsilon, \quad Y_n^q \in B(e).$$

If

$$\rho = \Upsilon\xi < \frac{1}{2}, \quad \frac{1-\sqrt{1-2\rho}}{\rho}\xi = e_0 \leq e \leq e_1 = \frac{1+\sqrt{1-2\rho}}{\rho}\xi,$$

then (2.164) has a unique solution Y_n^q *in* $B(e)$ *and*

$$\|(Y_n^q)^s - Y_n^q\| \leq \frac{\xi}{\rho}\left(1-\sqrt{1-2\rho}\right)^{s+1}, \quad s = 0, 1, \ldots.$$

The following theorem is immediate.

Theorem 2.20 *Consider the Volterra integro-differential equation given by (2.106), and the regions* $D_f := [0,T] \times \Re^m$ *and* $D_K := S \times \Re^m$, *where* $0 < T < \infty$ *and* $i = 1, 2, \ldots, m$. *Assume the following regularity conditions:*

- $f \in C^{2r}(D_f)$,
- $K \in C^{2r}(D_K)$,

- $a \in C^{2r-1}([0,T])$.

Also, the kernel K and the function f satisfy uniform Lipschitz conditions (in y) with Lipschitz constants L_K and L_f, respectively. Also, consider the time-point relaxation collocation method (2.128), where the inner iteration mode is Gauss-Jacobi or Gauss-Seidel and the outer is given by the modified Newton method (2.164), in which case the expression for $P(Y_n^q, Y_n^{q-1})$ is given by (2.155), the expressions for $F(Y_n^q, Y_n^{q-1})$ is given by (2.126) or (2.127), and the expression for $P'((Y_n^q)^0, Y_n^{q-1})$ is given by (2.165) or (2.166), respectively. Assume that the integrals in these expressions can be found exactly. Then the optimal nodal order is given by $\nu = min\{2r, q+1, (s+1)q\}$, provided the r collocation parameters $\{c_i\}$ are the r Gauss points in (0,1).

Proof:

Since all the conditions were verified for Newton's method, we need only apply the final conditions of the theorem. Using Theorem 2.19, for each $q = 1, 2, \ldots$, we find

$$\|(Y_n^q)^s - Y_n^q\| \leq \frac{\xi}{\rho}\left(1 - \sqrt{1-2\rho}\right)^{s+1}, \quad s = 0, 1, \ldots.$$

Expand $1 - \sqrt{1-2\rho}$ in a Taylor Series about the origin for $\rho < \frac{1}{2}$, and use the fact from Section 2.3.9 that $\rho = \mathcal{O}(h^q)$ and $\xi = \mathcal{O}(h^{q-1})$. This implies that

$$\|(Y_n^q)^s - Y_n^q\| = \mathcal{O}(h^{(s+1)q-1}), \quad s = 0, 1, \ldots.$$

The result then follows by applying the results of Section 2.3.8, since

$$\begin{aligned} \|\epsilon^{q,s}(t_n + \tau h)\| &= \|h[I_m \otimes \alpha^T(\tau)] \cdot [(Y_n^q)^s - Y_n^q]\| \\ &\leq h\|I_m \otimes \alpha^T(\tau)\| \cdot \|(Y_n^q)^s - Y_n^q\|, \end{aligned}$$

$q = 1, 2, \ldots$, $s = 1, 2, \ldots$, $\tau \in [0,1]$, $(Y_n^q)^0 = Y_n^{q-1}$ for $n = 0, 1, \ldots, N-1$, and Y_n^0 is given by (2.125). □

Let us compare Theorems 2.16, 2.18 and 2.20 in Table 2.1.

Table 2.1: Comparisons of Picard and Newton Methods

	Order = $min\{2r, q+1, \text{and...}\}$		
s	Picard	Newton	Modified Newton
1	$q+1$	$2q$	$2q$
2	$q+2$	$4q$	$3q$
3	$q+3$	$8q$	$4q$
4	$q+4$	$16q$	$5q$

Note:

The inner iteration mode sets the limit $q+1$ on the order and this order can be attained easily with Picard iteration $s = 1$. Both the Newton and modified Newton methods reach this limit also with $s = 1$. Therefore, if we are to take full advantage of the higher orders possible with the Newton methods we must use a higher order method (possibly a Newton method) for the inner iteration mode. ◇

To illustrate we solved Test Problem 2.2 using both the Newton and modified Newton methods to solve the implicit equations. I wrote a ©MATLAB program using Gauss-Jacobi inner iteration, two-point collocation (Gauss points) for the underlying numerical method and two point Gauss quadrature to evaluate the integrals. In practice it is not efficient to invert the matrix given in (2.156). Therefore we multiply this equation by $P'((Y_n^q)^{s-1}, Y_n^{q-1})$ and solve the following algebraic system by

LU decomposition, see Lambert (1991),

$$P'((Y_n^q)^{s-1}, Y_n^{q-1}) \cdot \tilde{\Delta}(Y_n^q)^{s-1} = -(Y_n^q)^{s-1} + F((Y_n^q)^{s-1}, Y_n^{q-1}), \tag{2.167}$$

where $s = 1, 2, \ldots$, and

$$\tilde{\Delta}(Y_n^q)^{s-1}) = (Y_n^q)^s - (Y_n^q)^{s-1},$$

is the increment that must be added to $(Y_n^q)^{s-1}$ to get $(Y_n^q)^s$.

We recall that P' is given by (2.157) and note that for this test problem we can calculate these partial derivatives in closed form.

Similar remarks apply to the modified Newton method. In Section 2.5, Tables 2.14 and 2.15 summarize the results of these tests.

2.4 The Discretized Case

All the results of the previous sections assume that the integrals appearing in the methods can be found analytically. However, in general, we are interested in the solution of a non-linear VIDE, whose kernel is sufficiently complicated to require that the integrals be approximated by suitable quadrature formulas. In other words, we have the so called *discretized cases*, sometimes called *fully discretized cases*. Therefore, we must compute an approximation which is a *perturbation* of the previous results. In order to illustrate how this can be accomplished, let us consider the case where the outer iteration mode is Picard fixed-point. The extension to the Newton cases is clear.

We consider an approximation $\hat{\eta}^{q,s} \in S_r^{(0)}(Z_N)$, which satisfies the equation

$$\hat{\eta}^{q,s}(t_n + \tau h) = \hat{\eta}(t_n) + h[I_m \otimes \alpha^T(\tau)] \cdot (\hat{Y}_n^q)^s, \tag{2.168}$$

$q = 1, 2, \ldots$, $s = 1, 2, \ldots$, $\tau \in [0, 1]$ and $n = 0, 1, \ldots, N-1$. Also, $\hat{\eta}^{q,0}(t_n + \tau h) = \hat{\eta}^{q-1}(t_n + \tau h)$ and $\hat{\eta}^0(t_n + \tau h) = \hat{\eta}(t_{n-1} + \tau h)$, for $n = 1, 2, \ldots, N-1$, and $\hat{\eta}^0(\tau h) = y_0$, for $\tau \in [0, 1]$.

The expressions for $(\hat{Y}_n^q)^s$ depend on the types of inner and outer iteration modes used over both q and s; see (2.147) or (2.148), respectively. Clearly, many quadrature formulas can be used to approximate these integrals; however, we choose interpolatory quadrature. In particular, we choose Gauss quadrature with abscissas $\{t_l + c_p h\}$ and $\{t_n + c_j c_p h\}$, with corresponding weights $\{w_p\} = \{\alpha_p(1)\}$ and $\{w_{jp}\} = \{c_j \alpha_p(1)\}$, respectively, where $l = 0, 1, \ldots, n-1$, $n = 0, 1, \ldots, N-1$ and $j, p = 1, 2, \ldots, r$. See Section 1.4.2, equation (1.37).

For **Gauss-Jacobi** iterations $(\hat{Y}_n^q)^s$ becomes

$$
\begin{aligned}
(\hat{Y}_n^q)^s \;=\; & \sum_{i=1}^{m} \sum_{j=1}^{r} M_{(i-1)r+j} \\
& \Bigg[f(t_n + c_j h, \hat{\eta}(t_n) + h\{A_i(c_j)(\hat{Y}_n^q)^{s-1} + \sum_{\substack{k=1 \\ k \neq i}}^{m} A_k(c_j) \hat{Y}_n^{q-1}\}) \\
+ \; & h \sum_{p=1}^{r} c_j \alpha_p(1) \cdot a[(c_j - c_j c_p)h] K(\hat{\eta}(t_n) + h\{A_i(c_j c_p)(\hat{Y}_n^q)^{s-1} \\
+ \; & \sum_{\substack{k=1 \\ k \neq i}}^{m} A_k(c_j c_p) \hat{Y}_n^{q-1}\}) \Bigg] \\
+ \; & \sum_{i=1}^{m} \sum_{j=1}^{r} M_{(i-1)r+j} \\
& h \sum_{l=0}^{n-1} \sum_{p=1}^{r} \alpha_p(1) \cdot a[(t_n + c_j h) - (t_l + c_p h)] K(\hat{\eta}(t_l + c_p h)),
\end{aligned}
\tag{2.169}
$$

$q = 1, 2, \ldots,\ n = 0, 1, \ldots, N-1$, where $(\hat{Y}_n^q)^0 = \hat{Y}_n^{q-1}$ and $\hat{Y}_n^0$ is given by (2.125).

For **Gauss-Seidel** iterations $(\hat{Y}_n^q)^s$ is

$$
\begin{aligned}
(\hat{Y}_n^q)^s \;=\; & \sum_{i=1}^{m}\sum_{j=1}^{r} M_{(i-1)r+j} \qquad (2.170)\\
& \Bigg[f(t_n + c_j h, \hat{\eta}(t_n) + h\{\sum_{k=1}^{i} A_k(c_j)(\hat{Y}_n^q)^{s-1} + \sum_{k=i+1}^{m} A_k(c_j)\hat{Y}_n^{q-1}\}) \\
+\; & h\sum_{p=1}^{r} c_j\alpha_p(1)\cdot a[(c_j - c_j c_p)h]K(\hat{\eta}(t_n) + h\{\sum_{k=1}^{i} A_k(c_j c_p)(\hat{Y}_n^q)^{s-1} \\
+\; & \sum_{k=i+1}^{m} A_k(c_j c_p)\hat{Y}_n^{q-1}\})\Bigg] \\
+\; & \sum_{i=1}^{m}\sum_{j=1}^{r} M_{(i-1)r+j} \\
& h\sum_{l=0}^{n-1}\sum_{p=1}^{r} \alpha_p(1)\cdot a[(t_n + c_j h) - (t_l + c_p h)]K(\hat{\eta}(t_l + c_p h)),
\end{aligned}
$$

$q = 1, 2, \ldots,\ n = 0, 1, \ldots, N-1$, where $(\hat{Y}_n^q)^0 = \hat{Y}_n^{q-1}$ and $\hat{Y}_n^0$ is given by (2.125).

Clearly, these expressions are a perturbation of (2.147) and (2.148), respectively.

Remark: In the integrand of our problem (2.106), the function a is restricted to the triangle $S = \{(t,s) : 0 \le s \le t \le T\}$, where $T < \infty$. Therefore, we take the *natural* discretizations using abcissas $t_n + c_j c_p h;\ n = 0, 1, \ldots, N-1,\ j = 1, 2, \ldots r, p = 1, 2, \ldots r$ to avoid including values outside S in the domain of a. In special cases, where we can extend the definition of the kernel to include values of s greater than values of t, we obtain additional methods of optimal order; see Brunner (1984). Also see Section 1.4.2. $\diamond$

Theorem 2.21 *Consider the Volterra integro-differential equation given by (2.106), and the regions* $D_f := [0,T] \times \Re^m$ *and* $D_K := S \times \Re^m$, *where* $0 < T < \infty$ *and*

$i = 1, 2, \ldots, m$. *Assume the following regularity conditions:*

- $f \in C^{2r}(D_f)$,
- $K \in C^{2r}(D_K)$,
- $a \in C^{2r-1}([0,T])$.

Also, the kernel K and the function f satisfy uniform Lipschitz conditions (in y) with Lipschitz constants L_K and L_f, respectively. Also, consider the time-point relaxation collocation method (2.168), where the inner iteration mode is Gauss-Jacobi or Gauss-Seidel and the outer one is Picard fixed-point. Interpolatory quadrature, corresponding to the abscissas $\{t_l + c_p h\}$ and $\{t_n + c_j c_p h\}$ where $l = 0, 1, \ldots, n-1$, $n = 0, 1, \ldots, N-1$ and $j, p = 1, 2, \ldots, r$, are used to solve the integrals in which case the expression for $(\hat{Y}_n^q)^s$ is given by (2.169) or (2.170), respectively.

Then the optimal nodal order is given by $\nu = min\{2r, q+1, q+s\}$, provided the r collocation parameters $\{c_i\}$ are the r Gauss points in (0,1).

Proof:

We define $\hat{e}^{q,s}(t) := y(t) - \hat{\eta}^{q,s}(t)$, $q, s = 1, 2, \ldots$, then

$$\begin{aligned} \hat{e}^{q,s}(t) &= (y(t) - \eta(t)) + (\eta(t) - \hat{\eta}(t)) + (\hat{\eta}(t) - \hat{\eta}^q(t)) + (\hat{\eta}^q(t) - \hat{\eta}^{q,s}(t)) \\ &= e(t) + \hat{\varepsilon}(t) + \hat{\epsilon}^q(t) + \hat{\epsilon}^{q,s}(t), \end{aligned}$$

where we have the following definitions:

- $e(t) := y(t) - \eta(t)$,

- $\hat{\varepsilon}(t) := (\eta(t) - \hat{\eta}(t))$,
- $\hat{\epsilon}^q(t) := \hat{\eta}(t) - \hat{\eta}^q(t)$,
- $\hat{\epsilon}^{q,s}(t) := \hat{\eta}^q(t) - \hat{\eta}^{q,s}(t)$,

for $q, s = 1, 2, \ldots$.

Setting $t = t_n$ and using the triangle inequality, we have

$$\|\hat{e}^{q,s}(t_n)\| \leq \|e(t_n)\| + \|\hat{\varepsilon}(t_n)\| + \|\hat{\epsilon}^q(t_n)\| + \|\hat{\epsilon}^{q,s}(t_n)\|,$$

$n = 0, 1, \ldots, N-1$.

Theorem 2.16 gives us the order of $e(t_n), \epsilon^q(t_n)$ and $\epsilon^{q,s}(t_n)$, where we recall that:

- $\epsilon^q(t_n) := \eta(t_n) - \eta^q(t_n)$,
- $\epsilon^{q,s}(t_n) := \eta^q(t_n) - \eta^{q,s}(t_n)$,

for $q, s = 1, 2, \ldots$.

It is easy to see that the results of Theorem 2.16 do not change if discretized versions of equations (2.146), (2.147) and (2.148) are assumed. This allows us to use Theorem 2.16 to give the order of $e(t_n), \hat{\epsilon}^q(t_n)$ and $\hat{\epsilon}^{q,s}(t_n)$.

Therefore we need only consider $\hat{\varepsilon}(t_n)$.

From Brunner (1984), we get

$$\hat{\varepsilon}(t_n) = \mathcal{O}(h^{2r}),$$

and therefore

$$\hat{e}^{q,s}(t_n) = \mathcal{O}(h^{\min\{2r, q+1, q+s\}}),$$

as $h \to 0^+$, if the collocation points are taken to be the r Gauss points in (0,1). □

For Newton outer iteration:

Theorem 2.22 *Consider the Volterra integro-differential equation given by (2.106), and the regions* $D_f := [0,T] \times \Re^m$ *and* $D_K := S \times \Re^m$, *where* $0 < T < \infty$ *and* $i = 1,2,\ldots,m$. *Assume the following regularity conditions:*

- $f \in C^{2r}(D_f)$,
- $K \in C^{2r}(D_K)$,
- $a \in C^{2r-1}([0,T])$.

Also, the kernel K *and the function* f *satisfy uniform Lipschitz conditions (in* y*) with Lipschitz constants* L_K *and* L_f, *respectively. Also, consider the time-point relaxation collocation method (2.168), where the inner iteration mode is Gauss-Jacobi or Gauss-Seidel and the outer is given by the Newton method (2.156), in which case the expression for* $P(Y_n^q, Y_n^{q-1})$ *is given by (2.155), the expression for* $F(Y_n^q, Y_n^{q-1})$ *is given by (2.126) or (2.127) and the expression for* $P'(Y_n^q, Y_n^{q-1})$ *is given by (2.157) or (2.158), respectively. Assume that the integrals in these expressions are approximated by interpolatory quadrature with abscissas* $\{t_l + c_p h\}$ *and* $\{t_n + c_j c_p h\}$, *where* $l = 0,1,\ldots,n-1,\ n = 0,1,\ldots,N-1$ *and* $j = 1,2,\ldots,r$. *Then the optimal nodal order is given by* $\nu = min\{2r, q+1, 2^s q\}$, *provided the* r *collocation parameters* $\{c_i\}$ *are the* r *Gauss points in* $(0,1)$.

And for modified Newton outer iteration:

Theorem 2.23 *Consider the Volterra integro-differential equation given by (2.106), and the regions* $D_f := [0,T] \times \Re^m$ *and* $D_K := S \times \Re^m$, *where* $0 < T < \infty$ *and* $i = 1,2,\ldots,m$. *Assume the following regularity conditions:*

- $f \in C^{2r}(D_f)$,
- $K \in C^{2r}(D_K)$,
- $a \in C^{2r-1}([0,T])$.

Also, the kernel K and the function f satisfy uniform Lipschitz conditions (in y) with Lipschitz constants L_K and L_f, respectively. Also, consider the time-point relaxation collocation method (2.168), where the inner iteration mode is Gauss-Jacobi or Gauss-Seidel and the outer is given by the modified Newton method (2.164), in which case the expression for $P(Y_n^q, Y_n^{q-1})$ is given by (2.155), the expressions for $F(Y_n^q, Y_n^{q-1})$ is given by (2.126) or (2.127) and the expression for $P'((Y_n^q)^0)$ is given by (2.165) or (2.166), respectively. Assume that the integrals in these expressions are approximated by interpolatory quadrature with abscissas $\{t_l + c_p h\}$ and $\{t_n + c_j c_p h\}$, where $l = 0, 1, \ldots, n-1$, $n = 0, 1, \ldots, N-1$ and $j = 1, 2, \ldots, r$. Then the optimal nodal order is given by $\nu = min\{2r, q+1, (s+1)q\}$, provided the r collocation parameters $\{c_i\}$ are the r Gauss points in (0,1).

Remark:

When considering the VIDE with weakly singular kernel given by (2.107), the discretized method (2.168) requires discretized versions of equations (2.151) and (2.152). However, special case must be taken to avoid the kernel from becoming unbounded at the abcissas in the quadrature formulas. In Test Problem 2.3, I used (two-point) Gauss quadrature at the abcissas $\{t_l + c_p h\}$, $\{t_n + c_j c_p h\}$ where

$l = 0, 1, \ldots, n-1$, $n = 0, 1, \ldots, N-1$, $j = 1, 2$ and $p = 1, 2$ in (2.151) and (2.152). Note that since $c_1 \neq 0$ and $c_2 \neq 1$ the kernel is finite at these abcissas. We refer the reader to Brunner and van der Houwen (1986) where they discuss "product integration formulas". $\diamond$

Summary

We summarize the main theorems of this thesis in Tables 2.2, 2.3 and 2.4.

Table 2.2: Continuous-Time Iteration WR Methods

Kernel Type	Convergence and Theorem
Regular	equations (2.58) and (2.59) Theorem 2.7
Weakly Singular	equation (2.90) Theorem 2.10

Table 2.3: Discrete-Time Iteration TR Methods - Regular Kernel

	Order and Theorem		
Type	Picard	Newton	Modified Newton
Exact	$min\{2r, q+1, q+s\}$ Theorem 2.16	$min\{2r, q+1, 2^s q\}$ Theorem 2.18	$min\{2r, q+1, (s+1)q\}$ Theorem 2.20
Discretized	$min\{2r, q+1, q+s\}$ Theorem 2.21	$min\{2r, q+1, 2^s q\}$ Theorem 2.22	$min\{2r, q+1, (s+1)q\}$ Theorem 2.23

Table 2.4: Discrete-Time Iteration TR Methods - Weakly Singular Kernel

	Order and Theorem
Type	Picard
Exact	$1-\alpha$, $0<\alpha<1$ for any $q \geq 1$ Corollary 2.4
Discretized	$1-\alpha$, $0<\alpha<1$ for any $q \geq 1$

Note: In Table 2.4, the poor convergence for discrete-time iteration TR methods for VIDEs with weakly singular kernel, is due to the fact that we are using *uniform* meshes. See Chapter 3. $\diamond$

2.5 Numerical Results

This section contains the results of various numerical tests, with a discussion of each.

Linear Test Problem

Consider Tables 2.5, 2.6, 2.7 and 2.8.

These are the results for the two-dimensional linear Test Problem 2.1, using both one-point and two-point collocation (Gauss points). The error is defined by

$$E_n^q := \|e^q(t_n)\|_2 = \|y(t_n) - \eta^q(t_n)\|_2,$$

$t_n = nh, \quad n = 0, 1, \ldots, N$, where $T = Nh$ and $q = 1, 2, \ldots$.

- **One-Point Collocation**

  First, consider Tables 2.5 and 2.6. By Theorem 2.15, we expect the order to be $\nu = min\{2, q+1\}$, since $r = 1$ and in each table $q \geq 1$. Not only do we observe this, but we confirm that this is the *best* we can achieve, since taking

$q > 1$ results in no increase in order. However, the error E_n^1 is generally larger that E_n^2 which is almost identical to E_n^3. We conclude by noting that there was no discernable difference in the performance, when using either Gauss-Jacobi iterations or Gauss-Seidel iterations.

- **Two-Point Collocation**

  In Tables 2.7 and 2.8, we have the results for the case where $r = 2$, so the order is $\nu = min\{4, q+1\}$, and we expect the error to decrease by a factor of $\frac{1}{2^\nu}$ over consecutive columns, for fixed q, in each table. For Gauss-Jacobi iterations, E_n^2 showed an order of 3, while E_n^3 and E_n^4 both showed an order of 4, as predicted by Theorem 2.15. However, for Gauss-Seidel iterations we attained an order of four even for $q = 2$, which is better than that predicted by Theorem 2.15. Clearly, for such a simple linear problem as Test Problem 2.1, the results of Theorem 2.15 can be exceeded.

Note: Tables 2.7 and 2.8 represent the only cases in these tests (in all of Section 2.5) where the order changes across the any of the columns. $\diamond$

Table 2.5: One-Point Collocation - Gauss-Jacobi Iterations (linear)

	$E_n^1 \times 10^{-4}$			$E_n^2 \times 10^{-4}$			$E_n^3 \times 10^{-4}$		
t_n	$h = 0.1$	0.05	0.025	$h = 0.1$	0.05	0.025	$h = 0.1$	0.05	0.025
0.2	184	37.4	8.42	42.0	11.3	2.92	48.5	12.1	3.01
0.4	23.8	3.71	0.924	56.8	14.3	3.58	57.7	14.4	3.58
0.6	54.5	15.0	3.87	53.4	13.2	3.26	52.1	13.0	3.24
0.8	80.6	20.2	5.05	44.7	10.9	2.69	42.6	10.6	2.65
1.0	83.1	20.3	5.03	35.7	8.63	2.12	33.5	8.34	2.08

Table 2.6: One-Point Collocation - Gauss-Seidel Iterations (linear)

	$E_n^1 \times 10^{-4}$			$E_n^2 \times 10^{-4}$			$E_n^3 \times 10^{-4}$		
t_n	$h = 0.1$	0.05	0.025	$h = 0.1$	0.05	0.025	$h = 0.1$	0.05	0.025
0.2	154	30.8	6.86	48.6	12.1	3.01	48.7	12.1	3.01
0.4	18.3	5.48	1.56	57.7	14.4	3.58	57.8	14.4	3.58
0.6	62.2	16.1	4.09	52.1	13.0	3.24	52.1	13.0	3.24
0.8	77.1	19.0	4.71	42.5	10.6	2.65	42.5	10.6	2.65
1.0	72.7	17.6	4.35	33.4	8.33	2.08	33.8	8.33	2.08

Table 2.7: Two-Point Collocation - Gauss-Jacobi Iterations (linear)

	$E_n^2 \times 10^{-7}$			$E_n^3 \times 10^{-7}$			$E_n^4 \times 10^{-7}$		
t_n	$h = 0.1$	0.05	0.025	$h = 0.1$	0.05	0.025	$h = 0.1$	0.05	0.025
0.2	7012	722.0	81.97	195.2	10.53	0.6141	64.77	3.955	0.2454
0.4	2111	210.7	23.61	117.6	6.911	0.4201	74.13	4.608	0.2878
0.6	1245	154.0	19.19	91.62	5.465	0.3339	64.28	4.020	0.2516
0.8	1726	208.4	25.60	70.32	4.132	0.2506	49.38	3.097	0.1941
1.0	1834	217.5	26.48	51.25	2.953	0.1773	35.28	2.222	0.1394

Nonlinear Test Problem

See Tables 2.9, 2.10 and 2.11.

These are the results for the two-dimensional nonlinear Test Problem 2.2, where we used both one-point and two-point collocation (Gauss points), Picard iteration to solve the implicit algebraic equations, and one and two point Gauss quadrature, respectively, to evaluate the integrals.

The error is defined by

$$E_n^{q,s} := \|e^{q,s}(t_n)\|_2 = \|y(t_n) - \eta^{q,s}(t_n)\|_2,$$

$t_n = nh$, $n = 0, 1, \ldots, N$, where $T = Nh$ and $q, s = 1, 2, \ldots$.

Table 2.8: Two-Point Collocation - Gauss-Seidel Iterations (linear)

	$E_n^2 \times 10^{-7}$			$E_n^3 \times 10^{-7}$			$E_n^4 \times 10^{-7}$		
t_n	$h = 0.1$	0.05	0.025	$h = 0.1$	0.05	0.025	$h = 0.1$	0.05	0.025
0.2	141.3	7.300	0.4212	62.90	3.915	0.2441	62.91	3.915	0.2441
0.4	105.4	6.596	0.4120	74.09	4.610	0.2878	74.09	4.610	0.2878
0.6	108.5	6.570	0.4031	64.89	4.038	0.2521	64.89	4.038	0.2521
0.8	90.23	5.296	0.3207	50.10	3.118	0.1947	50.10	3.118	0.1947
1.0	64.77	3.710	0.2221	36.02	2.242	0.1400	35.02	2.242	0.1400

For the two-point collocation cases, the results for Gauss-Jacobi iterations were found to be almost identical to the results for Gauss-Seidel iterations, so we do not include them.

- **One-Point Collocation**

  First, consider Tables 2.9 and 2.10. From Theorem 2.16, since $r = 1$, we expect the order to be $\nu = min\{2, q+1, q+s\}$. Not only do we observe this, but we confirm that this is the *best* we can achieve; that is, taking $q, s > 1$ results in no increase in order. We conclude by noting that there was no discernable difference in the the order, when using either Gauss-Jacobi iterations or Gauss-Seidel iterations, although the error is generally a little larger for the Gauss-Jacobi case than the Gauss-Seidel case. As the number of iterations increases this differences diminishes.

- **Two-Point Collocation**

  In Table 2.11 we have the results for the case where $r = 2$, so the order is $\nu = min\{4, q+1, q+s\}$, and we expect the error to decrease by a factor of $\frac{1}{2^\nu}$ over consecutive columns in each table, for fixed q and s. In fact, as a direct

consequence of Theorem 2.16,

$$E_n^{q,s} = \mathcal{O}(h^{\tilde{\nu}}) = E_n^{q,1},$$

where $\tilde{\nu} = min\{2r, q+1\}$, and $t_n = nh, \quad n = 0, 1, \ldots, N$, where $T = Nh$ and $q, s = 1, 2, \ldots$. This was observed since $E_n^{3,1}$ and $E_n^{3,2}$ both showed an order of four, as did $E_n^{4,1}$ and $E_n^{4,2}$. Therefore, from an order point of view, it never pays to iterate more than once when using Picard iteration to solve the implicit equations in the method. However, the actual error may become smaller if more iterates are taken.

Table 2.9: One-Point Collocation - Gauss-Jacobi Iterations (nonlinear)

	$E_n^{1,1} \times 10^{-4}$			$E_n^{1,2} \times 10^{-4}$			$E_n^{2,1} \times 10^{-4}$		
t_n	$h = 0.1$	0.05	0.025	$h = 0.1$	0.05	0.025	$h = 0.1$	0.05	0.025
0.2	169.4	40.37	9.685	53.41	14.34	3.666	4.848	1.788	0.6865
0.4	90.02	20.57	4.884	36.44	9.497	2.399	12.00	4.039	1.128
0.6	51.94	11.83	2.822	26.45	6.766	1.697	18.19	5.040	1.311
0.8	45.19	10.82	2.660	20.59	5.260	1.323	21.91	5.639	1.421
1.0	53.82	13.17	3.273	13.38	4.542	1.161	25.08	6.231	1.544

Weakly Singular Test Problem

See Tables 2.12 and 2.13.

These are the results for the two-dimensional linear weakly singular Test Problem 2.3, using two-point collocation (Gauss-points), Picard iteration (one iteration) to solve the implicit algebraic equations and two point Gauss quadrature to evaluate the integrals. We use Gauss-Jacobi inner iteration mode only. The error is defined

Table 2.10: One-Point Collocation - Gauss-Seidel Iterations (nonlinear)

	$E_n^{1,1} \times 10^{-4}$			$E_n^{1,2} \times 10^{-4}$			$E_n^{2,1} \times 10^{-4}$		
t_n	$h = 0.1$	0.05	0.025	$h = 0.1$	0.05	0.025	$h = 0.1$	0.05	0.025
0.2	114.0	27.52	6.647	6.771	2.481	0.7479	4.311	2.429	0.7665
0.4	60.79	14.28	3.437	14.14	3.944	1.033	15.09	4.422	1.174
0.6	36.18	8.727	2.140	16.00	4.081	1.031	20.05	5.254	1.336
0.8	35.87	9.053	2.276	15.26	3.765	0.9430	22.74	5.722	1.430
1.0	45.64	11.57	2.916	13.68	3.378	0.8540	25.05	6.205	1.539

Table 2.11: Two-Point Collocation - Gauss-Seidel Iterations (nonlinear)

	$E_n^{3,1} \times 10^{-7}$		$E_n^{3,2} \times 10^{-7}$		$E_n^{4,1} \times 10^{-7}$		$E_n^{4,2} \times 10^{-7}$	
t_n	$h = 0.1$	0.05	$h = 0.1$	0.05	$h = 0.1$	0.05	$h = 0.1$	0.05
0.2	95.91	5.920	4.690	0.3008	6.124	0.3191	4.777	0.3021
0.4	52.22	3.286	7.547	0.4814	7.865	0.4801	7.626	0.4834
0.6	34.47	2.324	9.578	0.6106	9.631	0.6044	9.678	0.6143
0.8	31.99	2.315	11.45	0.7297	11.35	0.7192	11.58	0.7354
1.0	36.76	2.717	13.44	0.8569	13.23	0.8422	13.61	0.8648

by

$$\|e^{3,1}\|_T := \max_{t\in[0,T]}\{\|y(t) - \eta^{3,1}(t)\|\} \approx E^{3,1} := \max_{n=0,1,\ldots,N-1}\{\|y(t_{n+1}) - \eta^{3,1}(t_{n+1})\|_\infty\}.$$

Comparing tables 2.12 and 2.13, we note that the results in both of these tables verify Corollary 2.4 and the "Remark" at the end of Section 2.4. Although solution (2.153) has greater regularity than solution (2.154), we obtain the same order, namely $1 - \alpha$. The only difference is that the higher smoothness in the solution gives rise to slightly smaller errors $E^{3,1}$.

Newton Test Problem

See Tables 2.14 and 2.15.

Table 2.12: Weakly Singular Solution (2.153)

	$E^{3,1} \times 10^{-3}$			
α	$h = 0.05$	$h = 0.025$	$h = 0.0125$	$h = 0.00625$
$\frac{1}{3}$	6.7811	4.1701	2.5951	1.6247
$\frac{1}{2}$	33.285	22.993	16.054	11.272
$\frac{2}{3}$	152.16	117.14	91.164	71.407

Table 2.13: Weakly Singular Solution (2.154)

	$E^{3,1} \times 10^{-3}$			
α	$h = 0.05$	$h = 0.025$	$h = 0.0125$	$h = 0.00625$
$\frac{1}{3}$	8.5943	5.3187	3.3201	2.0817
$\frac{1}{2}$	40.396	27.987	19.565	13.744
$\frac{2}{3}$	172.78	132.95	103.37	80.889

We solved the two-dimensional nonlinear Test Problem 2.2 using both the Newton and modified Newton methods to solve the implicit equations. The inner iteration mode was Gauss-Jacobi, the underlying numerical method was two-point collocation (Gauss points) and two point Gauss quadrature was used to evaluate the integrals. Also, the system given by (2.167) was solved by LU decomposition.

These results verify Theorems 2.18 and 2.20 which are summarized in Table 2.1. Since the Newton method and modified Newton method are identical for $s = 1$ and the results were found to be almost identical for $s = 2$, we only include the results for the Newton method.

In Table 2.14, we have the results for the case where $q = 2$, $s \geq 1$, and we expect

an order of 3. In Table 2.15, we have the results for the case where $q > 2$, $s \geq 1$, and we expect an order of 4. Clearly, all these claims are verified by these results.

Note that in Tables 2.14 and 2.15 the errors corresponding to $s = 2$ are larger than the errors corresponding to $s = 1$. This is likely due to the choice of initial approximation since the Newton method will eventually converge (in general, it does not converge monotonically).

Table 2.14: Newton Method

	$E_n^{2,1} \times 10^{-9}$			$E_n^{2,2} \times 10^{-9}$		
t_n	$h = 0.025$	0.0125	0.00625	$h = 0.025$	0.0125	0.00625
0.2	122.98	21.045	3.0067	1251.4	157.28	19.712
0.4	375.86	51.504	6.7330	856.63	108.20	13.593
0.6	516.35	69.438	9.0029	657.20	82.930	10.413
0.8	613.82	83.117	10.810	553.65	69.123	8.6331
1.0	720.70	98.591	12.878	542.98	66.498	8.2232

Table 2.15: Newton Method

	$E_n^{3,1} \times 10^{-8}$		$E_n^{3,2} \times 10^{-8}$		$E_n^{4,1} \times 10^{-8}$		$E_n^{4,2} \times 10^{-8}$	
t_n	$h = 0.1$	0.05	$h = 0.1$	0.05	$h = 0.1$	0.05	$h = 0.1$	0.05
0.2	57.308	2.8105	446.80	25.377	47.729	3.0224	49.903	3.0166
0.4	63.070	4.1797	246.14	14.247	76.360	4.8389	75.930	4.8036
0.6	72.311	4.8101	194.83	11.917	96.961	6.1508	96.130	6.1086
0.8	81.213	5.3407	229.21	14.327	116.04	7.3640	114.87	7.3101
1.0	93.242	6.1042	288.32	18.005	136.46	8.6610	134.90	8.5921

Chapter 3

Outlook

3.1 Applications

The major source of VIDEs is the nonlinear partial integro-differential equations, PIDEs, where the spatial derivatives are discretized by the *Method of Lines* (MOL); see Schiesser (1991) for partial differential equations, and for PIDEs see Kauthen (1992), and Chen and Shih (1998), also for additional references. This semi-discretization gives rise to a (large) system of VIDEs, and can therefore take advantage of WR methods, especially the fully parallel methods.

Yanik and Fairweather (1988) state in their introduction, that the need to include "memory" effects in a system which is a function of space and time is often described by partial integro-differential equations. These arise in various fields of engineering and physics and include heat transfer, nuclear reactor dynamics and thermoelasticity.

A particular problem of interest is the reaction diffusion models, with hereditary effects from population dynamics. These models give rise to PIDEs which combine spatial diffusion with hereditary interaction of species. Fife (1979) and Britton

(1986) are good references for the diffusion part and Cushing (1977) is the standard reference for the heredity part.

The *Journal of Integral Equations*, Vol. **10**, Numbers **1-3**, Supplement (1985) published a special conference issue on integro-differential evolution equations and applications. See also, Chapter 1 in Kolmanovskii and Myshkis (1992).

3.2 Future Work and Open Problems

No body of work stands in isolation; it begins where others have left off and hopefully, forms a basis for future work. Certainly this thesis began with and utilized the results of a wide variety of authors. We collected these results in Chapter 1. And now we finish the thesis, by itemizing some topics for future consideration. Some of these will be straightforward extensions of the theorems we listed in Section 1.2 and others will lead to unanswered questions, some of which may point to extensive and difficult areas of research.

3.2.1 WR Methods

Consider WR methods with different windowing.

- For the continuous-time iteration WR methods (on $[0,T]$), we commented that the error became large for large T. Therefore we could attempt to extend Theorems 2.7 and 2.10 to a partition of $[0,T]$, thereby promoting faster convergence.

- For the discrete-time iteration WR methods, recall that all employ windows of length equal to the steplength, which is why they are called time-point (TR)

methods. Clearly, for some problems, it could be advantageous to employ different "labour" to different subintervals of the integration interval $[0, T]$. And this could be reflected in the windows chosen.

3.2.2 TR Methods for Regular Kernels

- In the discrete-time iteration TR methods, we assumed the nonlinear problem (2.106). Clearly none of the results for these methods would change if we assumed the more general nonlinear problem (??).

- We could extend the results for the discrete-time iteration TR methods to the nonstandard VIDE (2.54); see the "Remark" in Section 2.2.

- In the discrete-time iteration TR methods one could experiment with considering different splitting functions G_f and G_k for f and k, respectively. See Example 2.8.

- The main theorems of Chapter 2 could be extended to collocation methods, employing a variety of collocation points and interpolatory quadrature formulae. See Brunner and van der Houwen (1986).

- Use higher order inner iteration schemes, for example, Newton iteration; see Section 2.2. This then, we would allow us to take advantage of high order outer iteration methods like the Newton and modified Newton. See Theorems 2.22 and 2.23, respectively.

3.2.3 TR Methods for Weakly Singular Kernels

In the discrete-time iteration TR methods applied to VIDEs with weakly singular kernels, we are limited by Theorem 2.14, since the underlying numerical method was polynomial spline collocation on a uniform mesh; see Table 2.4. In Section 1.4.4 we alluded to the fact that the use of nonpolynomial spline collocation methods or suitably graded meshes is more suitable for these problems; see Brunner (1986a) and Brunner and van der Houwen (1986). For example, Tang (1992 and 1993) showed that on suitably graded meshes, superconvergence properties are possible and therefore the restriction of Theorem 2.14 can be removed. Therefore an interesting proposal would be to consider discrete-time iteration TR methods utilizing such underlying numerical methods.

3.2.4 Stability

Note that we have two main concerns.

- Stability of the underlying classical numerical method for the solution of VIDEs.
- Stability of TR methods in general.

The question of numerical stability of discrete-time iteration TR methods for the solution of VIDEs requires addressing *both* of these concerns.

3.3 Conclusion

This thesis began by studying continuous-time iteration WR methods for VIDEs with regular and weakly singular kernels. We showed that by placing very general

assumptions on the splitting functions we could guarantee uniform convergence of the iterates on all bounded intervals, although maybe at the expense of having a very small window. Many questions remain unanswered. For example, how do we extend this analysis to unbounded intervals and how would different splitting functions affect the convergence rate?

The majority of the thesis dealt with the discrete-time iteration TR methods for VIDEs with regular and weakly singular kernels. Using collocation methods as their underlying numerical method, we developed detailed order conditions which considered separately, the effects on the order caused by the iteration in the TR method, the iteration in the method used to solve the implicit algebraic equations and the quadrature formula employed. The resulting method was therefore fully discretized. The results were encouraging for the regular kernel case, but disappointing for the weakly singular kernel case. However this low order is likely due entirely to the use of uniform meshes, and we hope that this can be remedied by employing suitably graded meshes.

Again, many problems remain open. They include, what restrictions must be placed on the steplength to guarantee convergence of the two types of iterations employed in the methods?

Although, these unanswered questions are important, I feel that this thesis sets the stage for their attack.

Bibliography

[1] Allen, D. N. (1954) *Relaxation Methods*, McGraw-Hill, New York.

[2] Axelsson, O. (1994) *Iterative Solution Methods*, Cambridge University Press, Cambridge.

[3] Bainov, D., Simeonov, P. (1992) *Integral Inequalities and Applications*, Kluwer, Dordrecht.

[4] Bellen, A. and Zennaro, M. (1993) The use of Runge-Kutta formulae in waveform relaxation methods, *Appl. Numer. Math.*, **11**, pp. 95-114.

[5] Bellen, A., Jackiewicz, Z. and Zennaro, M. (1993) Time-point relaxation Runge-Kutta methods for ordinary differential equations *J. Comput. Appl. Math.*, **45**, pp. 121-137.

[6] Bellen, A., Jackiewicz, Z. and Zennaro, M. (1994) Contractivity of waveform relaxation Runge-Kutta iterations and related limit methods for dissipative systems in the maximum norm, *SIAM J. Numer. Anal.*, **31**, pp. 499-523.

[7] Britton, N.F. (1986) *Reaction-Diffusion Equations and Their Applications to Biology*, Academic Press, London.

[8] Brunner, H. (1983) Nonpolynomial spline collocation for Volterra equations with wearly singular kernels, *SIAM J. Numer. Anal*, **20**, pp. 1106-1119.

[9] Brunner, H. (1984) Implicit Runge-Kutta methods of optimal order for Volterra integro-differential equations, *Math. Comput.*, **42**, pp. 95-109.

[10] Brunner, H. (1985) The approximate solution of Volterra equations with nonsmooth solutions, *Utilitas Mathematica*, **27**, pp. 57-95.

[11] Brunner, H. (1986a) Polynomial spline collocation methods for Volterra integro-differential equations with weakly singular kernels, *IMA J. Numer. Anal.*, **6**, pp.221-239.

[12] Brunner, H. (1986b) Resolvent equations for Volterra integral and integro-differential equations, *Course Notes*, University of Fribourg.

[13] Brunner, H. (1989) The numerical treatment of Volterra integro-differential equations with unbounded delay, *J. Comput. Appl. Math.*, **28**, pp. 5-23.

[14] Brunner, H. (1998) The use of splines in the numerical solution of differential and Volterra integral equations in: *Spline Functions and the Theory of Wavelets*, **I, II**, (Montréal, 1996) (Dubuc, S. (ed.)), CRM Proceedings and Lecture Notes, American Mathematical Society, Providence, RI, (to appear).

[15] Brunner, H., Crisci, M.R., Russo, E. and Vecchio, A. (1998) Waveform relaxation methods for weakly singular Volterra integral equations of the second kind, *IMA J. Numer. Anal.*, (to appear).

[16] Brunner, H., Hairer, E. and Nørsett, S.P. (1982), Runge-Kutta theory for Volterra integral equations of the second kind, *Math. Comp.*, **39**, pp. 147-163.

[17] Brunner, H. and van der Houwen, P.J. (1986) *The Numerical Solution of Volterra Equations*, North-Holland, Amsterdam.

[18] Burrage, K. (1995) *Parallel and Sequential Methods for Ordinary Differential Equations*, Clarendon Press, Oxford.

[19] Burrage, K., Jackiewicz, Z., Nørsett, S. P., and Renaut, R. A. (1996) Preconditioning waveform relaxation iterations for differential systems, *BIT*, **36**, pp. 54-76.

[20] Crisci, M.R., van der Houwen, P.J., Russo, E., and Vecchio, A. (1993) Stability of parallel Volterra Runge-Kutta methods, *J. Comput. Appl. Math.* , **45**, pp. 169-180.

[21] Crisci, M.R., Russo, E. and Vecchio, A. (1995) Block methods for second kind Volterra equations *Ricerche Mat. V.*, **XLIV**, pp. 507-524.

[22] Crisci, M.R., Ferraro, N. and Russo, E. (1996a) Convergence results for continuous-time waveform methods for Volterra integral equations *J. Comput. Appl. Math.*, **71**, pp. 33-45.

[23] Crisci, M.R., Russo, E. and Vecchio, A. (1996b) Discrete-time waveform relaxation Volterra-Runge-Kutta methods - Part II: stability analysis, *IAM - Rap. Tech.* **129/96**, C.N.R., Naples.

[24] Crisci, M.R., Russo, E. and Vecchio, A. (1997a) Discrete-time waveform relaxation Volterra-Runge-Kutta methods: Convergence analysis *J. Comput. Appl. Math.*, **86**, pp. 359-374.

[25] Crisci, M.R., Russo, E. and Vecchio, A. (1997b) Time point relaxation methods for Volterra integro-differential equations, *IAM-Rap. Tech.*, **158/97**, C.N.R., Naples.

[26] Cushing, J.M. (1977) *Integrodifferential Equations and Delay Models in Population Dynamics*, Lecture Notes in Biomathematics, **20**, Springer-Verlag, Berlin.

[27] D'Alterio, A. and Vecchio, A. (1995a) A variable step fifth order parallel algorithm for second kind Volterra integral equations: the code PIVRK3. *IAM - Rap. Tecn.*, **115/95**, C.N.R., Naples.

[28] D'Alterio, A. and Vecchio, A. (1995b) A variable step fifth order parallel algorithm for second kind Volterra integral equations: performance evaluation. *IAM - Rap. Tecn.*, **116/95**, C.N.R., Naples.

[29] Edwards, C.H. and Penney, D. E. (1994) *Elementary Differential Equations with Applications*, Third Edition, Prentice-Hall, Englewood Cliffs, New Jersey.

[30] Fife, Paul C. (1979) *Mathematical Aspects of Reacting and Diffusing Systems*, Lecture Notes in Biomath., **28**, Springer-Verlag, Berlin.

[31] Grossman, S.I., and Miller, R.K. (1970) Perturbation theory for Volterra integro-differential systems, *J. Differential Equations*, **8**, pp. 457-474

[32] Guillou, A. and Soulé, J.L. (1969) La résolution numérique des problèmes différentiels aux conditions initiales par des méthodes de collocation, *R.I.R.O*, No. **R-3**, pp. 17-44.

[33] Hairer, E., Nørsett, S.P. and Wanner, G. (1993) *Solving Ordinary Differential Equations I: Nonstiff Problems*, Second Revised Edition, SCM **8**, Springer-Verlag, Berlin.

[34] t' Hout, K.J. in (1995) On the convergence of waveform relaxation methods for stiff nonlinear ordinary differential equations, *Appl. Numer. Math.*, **18**, pp. 175-190.

[35] Jansen, J.K.M., Mattheij, R.M.M., Penders, M.T.M., and Schilders, W.H.A. (1994) Stability and efficiency of waveform relaxation methods *Comput. Math. Appl.*, **28**, pp. 153-166.

[36] Kahn, P.B. (1990) *Mathematical Methods For Scientistics & Engineers - Linear & Nonlinear Systems*, John Wiley & Sons, New York.

[37] Kantorovich, L.V., and Akilov, G.P. (1982) *Functional Analysis*, Second Edition, Pergamon Press, Oxford.

[38] Kauthen, J.P. (1992) The method of lines for parabolic partial integro-differential equations, *J. Integral Equations Appl.*, **4**, pp. 69-81.

[39] Kolmanovskii, V. and Myshkis, A. (1992) *Applied Theory of Functional Differential Equations*, Mathematics and Its Applications, **85**, Kluwer, Dordrecht.

[40] Lambert, J.D. (1991) *Numerical Methods for Ordinary Differential Systems: The Initial Value Problem*, Wiley, Chichester.

[41] Lancaster, P., and Tismenetsky, M. (1985) *The Theory of Matrices with Applications*, Second Edition, Computer Science and Applied Mathematics, Academic Press, Orlando.

[42] Lelarasmee, E. (1982) The waveform relaxation method for the time domain analysis of large scale nonlinear dynamical systems, Ph.D. Thesis, Univ. California, Berkley.

[43] Lelarasmee, E., Ruehli, A. and Sangiovanni-Vincentelli, A. (1982) The waveform relaxation method for time domain analysis of large scale integrated circuits, *IEEE Trans. Computer-Aided Design of Integrated Circuits and System* **1**, pp. 131-145.

[44] Lie, I. and Skålin, R. (1992) Relaxation based integration by Runge-Kutta methods and its applications to the moving finite element method. Preprint.

[45] Lubich, Ch. (1981) *Numerische Behandlung Volterra'scher Integro differentialgleichunger*, Diplomarbeit, Universität Innsbruck.

[46] Lubich, Ch. (1982) Kunge-Kutta theory for Volterra integrodifferential equations, *Numer. Math.*, **40**, pp. 119-135.

[47] Lubich, Ch. (1992) Chebyshev acceleration of Picard-Lindelöf iteration, *BIT* **32**, pp. 535-538.

[48] Miekkala, U. and Nevanlinna, O. (1987a) Convergence of dynamic iteration methods for initial value problems, *SIAM J. Sci. Statist. Comput.*, **8**, pp. 459-482.

[49] Miekkala, U. and Nevanlinna, O. (1987b) Sets of convergence and stability regions, *BIT*, **27**, pp. 557-584.

[50] Miller, R. K. (1971), *Nonlinear Volterra Integral Equations*, Benjamin, Menlo Park (Cal.).

[51] Morton, K.W., and Mayers, D.F. (1994) *Numerical Solution of Partial Differential Equations: An Introduction*, Cambridge University Press, Cambridge.

[52] Nevanlinna, O. (1989a) Remarks on Picard-Lindelöf iteration, Part I, *BIT*, **29**, pp. 328-346.

[53] Nevanlinna, O. (1989b) Remarks on Picard-Lindelöf iteration, Part II, *BIT*, **29**, pp. 535-562.

[54] Nevanlinna, O. (1990) Linear acceleration of Picard-Lindelöf iteration, *Numer. Math.*, **57**, pp. 147-156.

[55] Paone, A. and Vecchio, A. (1993) BVLM: a prototypal parallel code for solving second kind Volterra integral equations, *IAM - Rap. Tecn.* **101/93**, C.N.R., Naples.

[56] Paone, A. and Vecchio, A. (1994) Performance evaluation of a parallel algorithm for VIEs on local memory multiprocessor, *IAM - Rap. Tecn.* **105/94**, C.N.R, Naples.

[57] Schiesser, W.E. (1991) *The Numerical Method of Lines - Integration of Partial Differential Equations*, Academic Press, San Diego.

[58] Skeel R. D. (1989) Waveform iteration and the shifted Picard splitting, *SIAM J. Sci. Statist. Comput.*, **10**, pp. 756-776.

[59] Spijker, M.N. (1994) On the error committed by stopping the Newton iteration in implicit Runge-Kutta methods, *Ann. Numer. Math.* **1**, pp. 199-212.

[60] Sugiura, H. and Torii, T. (1991) A method for constructing generalized Runge-Kutta methods, *J. Comput. Appl. Math.*, **38**, pp. 399-410.

[61] Tang, T., (1992) Superconvergence of numerical solutions to weakly singular Volterra integro-differential equations, *Numer. Math.*, **61**, pp. 373-382.

[62] Tang, T., (1993) A note on collocation methods for Volterra integro-differential equations with weakly singular kernels, *IMA J. Numer. Anal.*, **13**, pp. 93-99.

[63] Vecchio, A. (1996) Highly stable parallel Volterra Runge-Kutta methods, *J. Comput. Appl. Math.* , **71**, pp. 225-236.

[64] White, J.K., Sangiovanni-Vincentelli, A., Odeh, F. and Ruehli, A. (1985) Waveform relaxation: Theory and practice, *Trans. Soc. Computer Simulation* **2**, pp. 95-133.

[65] White, J.K. and Sangiovanni-Vincentelli, A. (1987) *Relaxation Techniques for the Simulation of VLSI Circuits*, Kluwer Academic Publishers, Boston.

[66] Wright, K. (1970) Some relationships between implicit Runge-Kutta collocation and Lanczos τ methods, and their stability properties *BIT*, Vol. **10**, pp. 217-227.

[67] Yanik, E.G., Fairweather, G. (1988) Finite element methods for parabolic and hyperbolic partial integro-differential equations, *Nonlinear Anal.*, **12**, pp. 785-809.

[68] Zwillinger, D. (1992) *Handbook of Integration*, Jones and Bartlett Publishers, Boston.