Asymptotically Correct Defect Control Software for Boundary Value Ordinary Differential Equations

By

Adrian J. Ellis

A Thesis Submitted to Saint Mary's University, Halifax, Nova Scotia
in Partial Fulfillment of the Requirements for the
Degree of Master of Science in Applied Science.

Halifax, Nova Scotia
July 14, 2014,

Certification

Asymptotically Correct Defect Control Software for Boundary Value Ordinary Differential Equations

By

Adrian J. Ellis

A Thesis Submitted to Saint Mary's University, Halifax, Nova Scotia
in Partial Fulfillment of the Requirements for the
Degree of Master of Science in Applied Science.

July 14, 2014, Halifax, Nova Scotia

Examination Committee:

Approved: Dr. David Iron, External Examiner
Department of Mathematics and Statistics, Dalhousie University

Approved: Dr. Paul Muir, Senior Supervisor
Department of Mathematics and Computing Science

Approved: Dr. Walt Finden, Supervisory Committee Member
Department of Mathematics and Computing Science

Approved: Dr. Stavros Konstantinidis, Supervisory Committee Member
Department of Mathematics and Computing Science

Approved: Dr. Sean Kennedy, Chair of Thesis Defence
Faculty of Graduate Studies and Research

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to my supervisor Dr. Paul Muir for his valuable expertise, patient guidance, enthusiastic encouragement and constructive suggestions during this research work. His willingness to generously devote quality time to my program made this research a much more enjoyable experience. I would also like to thank the members of my supervisory committee, Dr. David Iron, Dr. Walt Finden and Dr. Stavros Konstantinidis for their advice, suggestions and encouragement.

Special thanks go to my colleague Jack Pew whose technical expertise was an invaluable resource throughout this research project.

Finally, I would like to thank my family for their patience and unwavering support throughout my studies.

Table of Contents

List of Tables

List of Figures

Asymptotically Correct Defect Control Software for Boundary Value Ordinary Differential Equations

By

Adrian J. Ellis

Abstract

BVP_SOLVER II [Boisvert, Muir, Spiteri, 2013] is an efficient software package for the numerical solution of systems of boundary value ordinary differential equations. It employs discrete mono-implicit Runge-Kutta (MIRK) schemes to transform the ODEs into nonlinear systems which are solved by modified Newton iterations. Continuous MIRK interpolants then augment the discrete solutions from the nonlinear system, to obtain a continuous solution approximation across the problem domain. The code monitors solution quality through defect analysis and employs an adaptive mesh refinement strategy as a means of controlling the defect, which is the amount by which the computed solution fails to satisfy the ODEs.

This thesis describes the development of new Hermite-Birkhoff interpolants and modifications to the BVP_SOLVER II software in order to implement a new defect estimation strategy called “Asymptotically Correct Maximum Defect Estimation”, based on the new interpolants. Numerical results which demonstrate the robustness and efficiency of the new strategy are presented.

July 14, 2014

Chapter 1

Introduction

1.1 Differential Equations

A differential equation is an equation that expresses a relationship between a function and its derivatives. This relationship is often used to describe how a quantity varies with time and space. Differential equations arose from early attempts by scientists at solving physical problems - a process which led to mathematical models involving an equation in which a function and its derivatives play important roles. Mathematical modeling presently provides widespread and essential insight into the analysis of many real world problems ranging from chemical reactions at the molecular level to motion of planetary bodies at the cosmic level.

The discipline of computational science represents the computer modeling of complex phenomena and plays an important role is all areas of science and engineering. Such computer models are usually based on complicated systems of differential equations. The complexity of these systems means that they often possess no analytic solution; in other words, they cannot be solved by analytical techniques. Therefore sophisticated and robust software packages are required for the computation of approximate numerical solutions to these systems.

1.1.1 Ordinary Differential Equations

An ordinary differential equation (ODE) involves a function of one independent variable and its derivatives. There are two main types: initial value ODES (IVODEs) and boundary value ODEs (BVODEs). An IVODE system consists of a set of differential equations with solution information specified at a single initial point. A simple IVODE system can be represented as:

$$\underline{y}'(t) = \underline{f}(t, \underline{y}(t)), \qquad t \geq a, \tag{1.1}$$

with initial condition(s)

$$\underline{y}(a) = \underline{\alpha}, \tag{1.2}$$

where $\underline{y}$ and $\underline{f}$ are vector functions, a is the initial point, and $\underline{\alpha}$ is a given constant vector.

BVODEs are systems of ordinary differential equations with boundary conditions imposed at two or more distinct points. The solution of the BVODE is then sought in the region between the boundary points. Two point BVODEs, (see for example [1]), have boundary conditions imposed at two distinct points and are usually represented as:

$$\underline{y}'(t) = \underline{f}(t, \underline{y}(t)), \quad t \in [a, b], \quad \underline{f} : \mathbb{R} \times \mathbb{R}^m \to \mathbb{R}^m, \quad \underline{y} \in \mathbb{R}^m, \tag{1.3}$$

where $\underline{y}$ and $\underline{f}$ are vector functions, $\underline{0}$ is a vector of zeros and a and b are known endpoints. Equation (1.3) is usually accompanied by a system consisting of either

non-separated boundary conditions

$$\underline{g}(\underline{y}(a), \underline{y}(b)) = \underline{0}, \tag{1.4}$$

or separated boundary conditions

$$[\underline{g}_0(\underline{y}(a)), \underline{g}_1(\underline{y}(b))]^T = \underline{0}, \tag{1.5}$$

where $\underline{g}$ is a vector function, $g_0 : \mathbb{R}^m \to \mathbb{R}^{m_o}$, $g_1 : \mathbb{R}^m \to \mathbb{R}^{m_1}$ and $m_0 + m_1 = m$. Equations (1.3) with boundary conditions (1.4), (1.5) is known as a first order system meaning that only first derivatives of the relevant quantities appear.

As mentioned earlier, systems of ordinary differential equations are used to model phenomena that vary with time or space and are employed in a variety of applications. These include, for example, modeling the human heart, predicting the extent of a viral outbreak, numerical simulations of fluid dynamics, studying the motion of celestial bodies, numerical weather forecasting, calculating the value of stocks options, simulating car crashes, and computing the trajectory of space crafts [1].

1.2 Thesis Structure

This thesis describes software development and modification associated with the algorithmic enhancement of the BVP_SOLVER II [45] software package. BVP_SOLVER II is a Fortran 90/95 based solver used in the numerical solution of systems of first order nonlinear, BVODEs, with separated boundary conditions. BVODEs are said to have separated boundary conditions if each of the components of $\underline{g}$ is given either at $t = a$ or at $t = b$, but none involves both ends simultaneously [5].

1.2.1 Summary of the BVP_SOLVER II Software

For a given mesh of points which partition the problem domain into subintervals, BVP_SOLVER II employs a combination of discrete mono-implicit Runge-Kutta (MIRK) formulas [17] and continuous mono-implicit Runge-Kutta (CMIRK) schemes [40], to provide approximate solutions to BVODE systems. The discrete MIRK schemes are used for the discretization of the ODEs leading to solution approximations at the mesh points whilst the CMIRK schemes augment the discrete solutions to produce a continuous solution approximation over the entire problem domain. The solver monitors the quality of the numerical solution through defect control. The defect of a numerical solution is the amount by which that solution fails to satisfy the ODE system. If, for instance, $u(t)$ is the continuous approximate solution to (1.3), (1.4), (1.5), the defect $\delta(t)$ of $u(t)$ is defined as follows:

$$\delta(t) = u'(t) - f(t, u(t)). \tag{1.6}$$

The defect is computed by substituting the approximate solution $u(t)$ in place of the exact solution $y(t)$ into the ODE system (1.3) and subtracting the right hand side of the equation from the left hand side to see how well $u(t)$ satisfies the ODE system.

BVP_SOLVER II attempts to compute a numerical solution for which the maximum defect on each subinterval is less than a user-defined tolerance. This requires the code to estimate the maximum defect in an efficient and robust manner. This is currently done in BVP_SOLVER II by sampling the defect at two points per subinterval.

1.2.2 High Level Thesis Statement

The overall robustness and efficiency of BVP_SOLVER II depends on an efficient and accurate estimate of the maximum defect on each subinterval. The two-point sampling currently employed is efficient but can lead to significant underestimation of the maximum defect.

The main goal of this thesis is (i) to develop and investigate the numerical performance of new Hermite-Birkhoff interpolants, based on the existing CMIRK schemes, that lead to an improved defect estimation technique known as asymptotically correct defect estimation [23], and (ii) to develop a new version of the BVP_SOLVER II software that implements this new approach for defect estimation.

This thesis work consists of three main phases. Phase one constitutes a review and verification of the results of [23], for the sixth order Hermite-Birkhoff interpolation scheme and the derivation of the fourth and second order Hermite-Birkhoff schemes. Phase two relates to the modification of the BVP_SOLVER II software package in order to incorporate the new Hermite-Birkhoff interpolants as well as an auxiliary process known as a validity check. The third phase is concerned with the development of other types of CMIRK schemes which yield asymptotically correct defect estimates.

1.3 Thesis Organization

The organization of this thesis is as follows. Chapter 2 gives a review of standard methods and related software for the numerical solution of BVODEs. Chapter 3 provides an in-depth review of Runge-Kutta based methods and software, predominantly those that implement defect control. Chapter 4 details the derivation of

Hermite-Birkhoff schemes using a boot-strapping algorithm that leads to the new asymptotically correct approach for estimating the maximum value of the defect on each subinterval. The suite of test problems which formed the basis of the numerical experiments are described in Chapter 5. Chapter 6 chronicles the software modifications to BVP_SOLVER II whilst Chapter 7 contains the description of the various numerical experiments conducted as well the relevant results. In Chapter 8 a special type of fourth order CMIRK scheme is developed that leads to an asymptotically correct estimate of the maximum defect. Chapter 9 contains the conclusion to the thesis and future work.

Chapter 2

Review of Numerical Solution of BVODES

2.1 Numerical Methods

2.1.1 Introduction

Numerical methods for the solution of BVODEs are generally categorized into initial value methods and global methods. The distinction between the two method classes depends on the computational approach as well as the manner in which the solution approximations are computed. The basic approach shared by all initial value methods is to compute a solution approximation by numerically integrating in a step-wise fashion from an initial starting point to a final terminal point on the problem interval. The global methods on the other hand, discretize the BVODE system using a given mesh which subdivides the problem interval. This produces a system of algebraic equations which are then solved to simultaneously produce an approximate solution over the problem interval.

2.1.2 Shooting / Multiple Shooting Methods

The simple shooting method is one of the most popular approaches employed in the numerical solution of BVODES. This intuitive method builds on the initial value ordinary differential equation (IVODE) approach and is a straightforward extension of initial value techniques. Starting with estimated initial conditions at the left end

point of the problem interval, a (since we are considering BVODEs, we do not have complete solution information at a and the missing information must be estimated), one essentially tries to hit known boundary values at the right end point, b, by integrating varying trajectories of the same ordinary differential equation over the problem domain.

Consider the BVODE system comprising of equations (1.3) and (1.4). We denote by $\underline{y}(t) \equiv \underline{y}(t;\underline{c})$ the vector solution of the ODE (1.3) which satisfies the initial (or left end point) condition $y(a;\underline{c}) = \underline{c}$. Note that since we are considering a BVODE system, not all components of $\underline{c}$ are known. Then we can write

$$\underline{h}(\underline{c}) \equiv \underline{g}(\underline{y}(a;\underline{c}), \underline{y}(b;\underline{c})) = \underline{g}(\underline{c}, \underline{y}(b;\underline{c})) = \underline{0}, \tag{2.1}$$

which gives a set of m nonlinear algebraic equations for the m unknown initial conditions $\underline{c}$.

The intuitive simplicity of this approach and availability of excellent, robust initial value numerical software [5] makes the simple shooting method an attractive computational approach. However a major difficulty associated with this method is its inherent instability which is due to the conditioning of each shooting step being dependent on the conditioning of the IVODE. This often leads to unbounded growth in the solution error [1]. The basic issue is that a well-posed, stable BVODE may have solution components that increase exponentially from left to right and it is difficult for an initial value solver, integrating from left to right, to compute accurate approximations to these components.

Multiple shooting [44, 36], is an improved variant of the simple shooting method. Essentially this approach employs a set of mesh points to partition the problem interval and on each subinterval a local initial value problem with estimated initial conditions specified at the left end point of each subinterval is set up and solved. The advantage of this method over simple shooting is that the local IVODE is solved over smaller problem intervals. However requiring the local initial value solutions to match at the internal mesh points and also satisfy the boundary conditions leads to a large system of nonlinear equations. The effectiveness of this approach is limited to less difficult classes of BVODES. In particular singularly perturbed problems expose the limitations of the method because the possible presence of rapidly increasing solution modes cannot be dealt with using an initial value solver even on smaller intervals.

2.1.3 Collocation Methods

The collocation methods are a popular type of global method for the numerical solution of BVODEs. The approximate solution is represented as a linear combination of known basis functions with unknown coefficients. Then the approximate solution is substituted into the system of ODEs with the requirement that the ODE system be satisfied exactly at a set of points distributed over the problem domain, called collocation points. The number of collocation points plus the number of boundary conditions must equal the number of unknown coefficients in the approximate solution. To achieve optimal accuracy careful consideration is placed on the choice of an appropriate basis as well as the positioning of the collocation points. These two criteria have been discussed in many papers; see, e.g., [7, 9]. A popular combination

employed is a linear space of piecewise polynomial functions, called splines, for the basis functions, and Gauss points [1] on each subinterval as the collocation points. The collocation and boundary conditions lead to a nonlinear system which must be solved iteratively. For each iteration, a structured linear system of equations must be solved.

2.1.4 Finite Difference Methods

The finite difference approach represents another type of global method for solving BVODEs. In this approach a mesh is defined on the problem interval [a,b] and the derivative in equation (1.1) is replaced by a finite difference approximation at each mesh point. The resulting difference equations plus the boundary conditions give a set of algebraic equations for the solution on the mesh. These equations are generally nonlinear. In [1] the basic steps for the use of a finite method for the solution of BVODES are outlined as follows:

1. For a given mesh π:

$$a = t_0 < t_1 < t_2 \cdots < t_{N-1} < t_N = b,$$

define approximate solution values, for $i = 0, \cdots, N$,

$$y_i \approx y(t_i).$$

2. Replace derivatives in the differential equations and boundary conditions with finite difference quotients

$$y'(t_i) \approx \frac{y_{i+1} - y_i}{t_{i+1} - t_i}.$$

This forms a set of algebraic nonlinear equations for the approximate solution at the mesh points.

3. Solve this set of equations together with the boundary conditions for the approximate solution values at the mesh points.

2.1.5 Runge-Kutta Methods

The Runge-Kutta (RK) schemes are another important class of global methods for BVODEs. Explicit Runge-Kutta (ERK) schemes were originally devised for the solution of IVODES by Runge with further development by Heun, Kutta, and Nystrom - see, e.g., [14]. Implicit Runge-Kutta (IRK) methods for IVODEs were first proposed by Butcher, see, e.g., [14], with a focus on methods based on Gaussian quadrature formulae. A remarkable feature of this latter class of methods is that they are all A-stable, making them especially suitable for stiff initial value differential problems [14].

The Runge-Kutta schemes represent higher order generalizations of finite difference methods and have also been used in the numerical solution of BVODEs. Assuming a given mesh and discrete approximate solution values as in section 2.1.4, the ODE

$$\underline{y'} = \underline{f}(t, \underline{y}),$$

is replaced on each subinterval by an IRK scheme which discretizes the ODE at t_i. The general form of an s stage IRK method is:

$$\frac{\underline{y}_{i+1} - \underline{y}_i}{h_i} = \sum_{r=1}^{s} b_r \underline{k}_r, \qquad i = 0, \cdots, N-1, \tag{2.2}$$

where

$$\underline{k}_r = \underline{f}(t_i + c_r h_i, \underline{y}_i + h_i \sum_{j=1}^{s} a_{rj} \underline{k}_j), \tag{2.3}$$

r=1,$\cdots$,s, are called the stages of the method, h_i=t_{i+1} - t_i, and $\{b_r\}_{r=1}^{s}$, $\{c_r\}_{r=1}^{s}$ and $\{a_{rj}\}_{j=1}^{s}$ are the coefficients of the Runge-Kutta scheme. These methods are often represented in a tableau containing their coefficients, which for the above method, will have the structure:

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \cdots & a_{1s} \\ c_2 & a_{21} & a_{22} & \cdots & a_{2s} \\ \vdots & \vdots & \vdots & & \vdots \\ c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\ \hline & b_1 & b_2 & \cdots & b_s \end{array} .$$

The equations (2.4) together with boundary conditions give a system of nonlinear equations for the solution approximations at the mesh points.

2.1.6 Comparisons

There is a direct correlation between the knowledge gained through extensive research into BVODE literature and the evolution of the different numerical approaches being implemented presently. Scientific research in computational disciplines leads to the inevitable discovery of new and more complex types of boundary value problems and so the search for efficient numerical methods remains a continuous process. As a

consequence, certain methods developed earlier for the numerical solution of BVODEs are now obsolete. This section of the chapter chronicles the evolution of numerical methods and approaches from an efficiency standpoint, highlighting the advantages and disadvantages of implementing various numerical schemes.

The intuitive simplicity of the shooting methods and relative ease of implementation makes this a popular numerical scheme. In fact, in an earlier survey of existing numerical methods, Keller [31] argued that for a first order system, the simple shooting method was superior to certain collocation methods. The main reasoning behind this assertion was the relatively low computational cost in implementing this approach especially in cases where the boundary conditions are separated. Simple shooting methods are generally very efficient in the numerical solution of easy BVODEs; however this class of BVODEs constitute only a small proportion of all BVODEs. The inherent instability of the shooting methods is a consequence of its inability in dealing with the dichotomy of solution modes commonly present in more complex BVODE systems. Essentially the shooting approach involves integrating an ODE system over a problem interval in the forward direction. However integrating ODEs that have solutions which grow exponentially from left to right, leads to unbounded growth in solution error. The multiple shooting method does limit this undesirable trait to some extent by reducing the problem interval; however it is ineffective for difficult BVODEs. Subsequent research to improve upon the multiple shooting approach by decoupling solution modes proved very costly to implement prompting researchers to explore other more efficient alternatives [1].

The limitations of the shooting method required a shift in direction of the research into numerical approaches. Rather than adapt existing initial value based approaches, more emphasis was placed on the derivation of methods possessing qualities suited to the numerical solution of BVODEs. This led to the use of finite difference schemes for BVODEs. The finite difference methods are based on an intuitively simple concept of discretizing ODE systems. However as a global method, they are more suited to handling the dichotomy of solution modes in BVODE systems and are therefore more efficient than the shooting methods. Keller observed in his survey of existing numerical methods [31], that the finite difference method called the centered Euler (or Box) scheme [32], was superior to the shooting methods as well as certain collocation methods. The main disadvantage of this scheme however was that it produced only second order solution approximations; that is, the error is $O(h^2)$, where h is the maximum subinterval size. More research into the derivation of higher order finite difference schemes culminated in the derivation of a computationally efficient and stable one-step finite difference method of order four by Cash and Moore [17] which required only about twice as much computational effort as the box scheme. The emergence of the Runge-Kutta methods as a viable approach in the numerical solution of BVODEs generalized the finite difference schemes, thereby making it easier to derive higher order one-step (i.e., one subinterval) formulas. More importantly, the implicit subclass of these one-step methods has further extended the range and capability of the finite difference methods in efficiently solving difficult classes of BVODEs. Implementation

of the finite difference and Runge-Kutta schemes produces discrete approximate solution values only at the mesh points; however, the derivation of interpolants which augment the discrete solutions and produce a continuous solution approximation has successfully overcome this limitation.

A specific type of IRK methods, the collocation methods, have proven to be very popular for BVODEs. There are two main advantages of implementing the collocation approach in software packages. Firstly, these schemes are suited to handling types of BVODE systems known as mixed order systems in which varying orders of the solution derivatives occur, and secondly, unlike the finite difference methods, they possess a natural implementation which produces a continuous solution approximation. It is well-known [48, 47] that for every collocation method there is an equivalent implicit Runge-Kutta method. They are equivalent in the sense that the discrete solution of the implicit Runge-Kutta method agrees exactly with the piecewise polynomial approximation generated by the collocation method, evaluated at the mesh points. Given a collocation method it is a reasonably straightforward procedure to obtain the corresponding implicit Runge-Kutta method. Weiss [47] also showed that the Lobatto quadrature points were the most efficient, giving an accuracy of $O(h^{2s-2})$ when s collocation points were used in each subinterval. The major disadvantage with the collocation approach is the necessity to solve a system of $m \times s$ nonlinear algebraic equations on each subinterval, which is a significant computational expense. (Recall that m is the number of ODEs).

2.2 Numerical Software

2.2.1 Introduction

The development of high quality general purpose software for the solution of BVODEs started with the initial value approach since the underlying mathematical theory for IVPs was much better understood and extensive research had already been undertaken to develop the associated software. BVODES were also originally regarded as specific types of IVODES and so software designed to solve IVODEs was adapted and applied to this problem class. However the inherent instability of the initial value method as well as the realization that BVODES were a separate and indeed more complicated class of problems led researchers to seek more efficient alternate strategies.

Numerical software packages for BVODEs implement various types of error control to measure solution quality. Common measures of solution quality are (i) local truncation error (LTE), which is the error incurred on each subinterval, (ii) Global Error (GE) - the difference between the exact and approximate solutions, and (iii) the defect, the amount by which the computed solution fails to satisfy the system of differential equations and the boundary conditions.

The remaining sections in this chapter chart the development of general purpose BVODE software based primarily on the first two error controls described above. A more detailed examination of defect control software is done in the next chapter.

2.2.2 Shooting Method Software

Shooting software generally consists of an IVODE solver working together with a nonlinear equation solver. One of the earliest shooting codes for BVODES was that of Riley et al.[44], called SUPORT. Shooting methods were also implemented by England, Nichols and Reid [21], and then in the software package BOUNDS developed by Bulirsch, Stoer and Deuflhard [12].

Multiple shooting has been implemented in many software packages, a number of which constitute components of standard libraries for numerical software. Keller popularized the method by developing both a simple shooting code (SSM) [31] and a multiple shooting variant (MSM) [32]. The MUSN package [36] developed by Mattheij and Staarink is based on a more recent version of the multiple shooting method designed for non-stiff nonlinear BVODEs whilst MUSL [44] is the multiple shooting variant designed for non-stiff linear BVODEs.

2.2.3 Deferred Correction Based Software

TWPBVP [35] is a deferred correction based software package which employs A-stable, symmetric, MIRK schemes. This software is designed for the numerical solution of first order systems of nonlinear BVODEs and implements a deferred correction method based on MIRK schemes of orders 4, 6, and 8. The first step of the deferred correction approach involves the computation of a 4th order solution approximation using the 4th order scheme. Then the 6th and 8th order methods are employed to generate two subsequent corrections of the approximate solution to 6th and 8th order respectively. The code controls a LTE estimate of the solution at the mesh points.

Deferred correction has also been the basis for a number of other codes; notably the PASVA3 solver developed by Lentini and Pereyra [35] which implements deferred correction based on the box scheme. An experimental solver, generalizing the approach employed in PASVA3 through the use of MIRK methods is discussed in [29]. MIRK methods and Lobatto collocation methods are implemented within a deferred correction framework in the BVODE solvers TWPBVP [35] and ACDC [19], and the related solver, TWPBVPL [15]. All of these solvers control estimates of the LTE and base mesh refinement on these estimates. Extensions that consider mesh refinement based on the LTE estimates and on estimates of the conditioning constant of the BVODE have led to new versions of TWPBVP and TWPBVPL, called TWPBVPC and TWPBVPLC [16].

2.2.4 Collocation Based Software

The collocation based numerical software package COLSYS (COLocation for SYStems) [2, 3] was one of the earliest BVODE solvers to implement GE control. Several modifications of this solver have been developed to improve its capabilities; examples include COLNEW[4, 8], COLDAE [6], and COLMOD [19]. These solvers are capable of handling mixed order nonlinear BVODE systems. The method of spline collocation at Gaussian points is implemented using a B-spline basis [2, 3] in COLSYS. COLNEW employs the same spline collocation approach but uses a monomial basis. The modifications in COLDAE significantly extend the range of the COLSYS/COLNEW solvers. Nonlinear systems of semi-explicit differential algebraic equations (DAEs) as well as some fully implicit boundary value DAE problems can be efficiently solved.

COLMOD, an extension to COLNEW employs an automatic continuation strategy, which is an approach where a sequence of progressively more difficult problems is solved by using information from one problem to solve for the next.

Error estimation in these solvers is implemented in two ways. The computed estimate of the discretization error is used for mesh refinement and a preliminary assessment of the acceptability of the numerical solution. This estimate may be unreliable for crude tolerances or high order so an estimate of the GE is computed using Richardson Extrapolation [10]. Only after this second estimate satisfies the user tolerance is the numerical solution accepted.

Chapter 3

Runge-Kutta Methods and Defect Control Software Packages

3.1 Runge-Kutta Methods

3.1.1 Explicit and Implicit Runge-Kutta Methods

The Runge-Kutta methods, as mentioned in the previous chapter, are generalizations of the finite difference schemes, commonly used for the numerical solution of the IVODE system (1.1), (1.2). Explicit Runge-Kutta (ERK) schemes have excellent efficiency properties because each stage is explicitly defined in terms of quantities that are already known. Therefore all the stages of the method can be computed without the necessity of solving linear or non-linear systems. The efficiency of the ERK schemes however is offset by inherent stability issues which make them generally unsuitable for use in the solution of difficult BVODEs. ERK methods can be represented as follows,

$$\underline{y}_{i+1} = \underline{y}_i + h_i \sum_{r=1}^{s} b_r \underline{k}_r \tag{3.1}$$

with stages,

$$\underline{k}_1 = \underline{f}(t_i, \underline{y}_i),$$

$$\underline{k}_r = \underline{f}(t_i + c_r h, \underline{y}_i + h_i \sum_{j=1}^{r-1} a_{r,j}\underline{k}_j) \qquad 2 \leq r \leq s, \tag{3.2}$$

where s is the number of stages, and $\{a_{r,j}\}_{j=1,r=1}^{r-1,s}$ and $\{b_r\}_{r=1}^{s}$ are the internal and external weights, respectively. The abscissa c_r are defined by $c_r = \sum_{j=1}^{r-1} a_{r,j}$ and the length of the subinterval is $h_i = t_{i+1} - t_i$. The coefficients of the ERK schemes are usually expressed in a Butcher tableau of the form:

$$\begin{array}{c|cccccc} c_1 & 0 & 0 & 0 & \cdots & 0 \\ c_2 & a_{21} & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \cdots & \cdots & \vdots \\ c_s & a_{s1} & a_{s2} & \cdots & a_{s,s-1} & 0 \\ \hline & b_1 & b_2 & \cdots & \cdots & b_s \end{array} .$$

The Implicit Runge-Kutta (IRK) methods described in section 2.1.5 constitute the other major half of the Runge-Kutta class of schemes. These methods were first presented by Butcher [14] for use in the numerical solution of initial value ODEs. These schemes, unlike their explicit counterparts, can possess excellent stability properties; however from (2.5), it is evident that each stage, $\underline{k_r}$, is defined implicitly in terms of itself and the other stages. Therefore in order to obtain approximate stage values, it is necessary to solve a system of $m \times s$ coupled nonlinear equations where m represents the number of differential equations and s represents the number of stages of the Runge-Kutta method. The most popular approach for solving this system of

nonlinear equations is to use some form of a modified Newton iteration which makes the stage calculations a somewhat computationally expensive process.

Consider the initial value ODE system consisting of equations (1.1) and (1.2). The Runge-Kutta method (3.1) with stages (2.3), computes a sequence of discrete approximation vectors $y_i \approx y(t_i)$ in a step-wise fashion, starting with $t_0 = a$ and $y_0 = y_a$. On a general step with y_i available, a step size h_i is chosen and the next approximation is computed at $t_{i+1} := t_i + h_i$.

In the two point BVODE system described by (1.3) and (1.4), given a mesh which subdivides the problem interval, the basic approach is to use the IRK formulas to form a discrete algebraic system consisting of the boundary conditions and m more equations per subinterval, which can then be solved with a Newton iteration to obtain a discrete solution Y having the form $Y = [y_0, y_1, \cdots, y_N]^T$, where N is the number of subintervals in the current mesh. When an IRK scheme is employed as the discretization scheme, the set of m equations associated with the ith subinterval has the form:

$$\underline{\phi}(y_{i+1}, y_i) = \underline{y}_{i+1} - \underline{y}_i - h_i \sum_{r=1}^{s} b_r \underline{k}_r = \underline{0}, \tag{3.3}$$

where

$$\underline{k}_r = \underline{f}(t_i + c_r h, \underline{y}_i + h_i \sum_{j=1}^{s} a_{rj} \underline{k}_j). \tag{3.4}$$

The boundary value ODEs to be solved are assumed to be expressible in the general form described by equations (1.3) and (1.4).

3.1.2 Mono Implicit Runge-Kutta Methods

Due to the considerable attention devoted to the IRK schemes in the research literature, a number of interesting subclasses have been identified and investigated. These methods attempt to trade-off the higher accuracy of the fully IRK methods for methods that can be implemented more efficiently. The *parameterized* implicit Runge-Kutta (PIRK) methods presented by Muir and Enright [42], is an alternate representation of the IRK schemes having the form

$$\underline{y}_{i+1} = \underline{y}_i + h_i \sum_{r=1}^{s} b_r \underline{k}_r. \tag{3.5}$$

with stages,

$$\underline{k}_r = \underline{f}(t_i + c_r h, (1 - v_r)\underline{y}_i + v_r \underline{y}_{i+1} + h_i \sum_{j=1}^{s} x_{r,j} \underline{k}_j). \tag{3.6}$$

The scheme is defined by the number of stages, s, the coefficients $\{v\}_{r=1}^{s}$ and $\{x_{r,j}\}_{j=1,r=1}^{s,s}$ and the weights $\{b_r\}_{r=1}^{s}$. The abscissa c_r are defined by $c_r = v_r + \sum_{j=1}^{s} x_{r,j}$ and the length of the step is $h_i = t_{i+1} - t_i$. The coefficients of the PIRK schemes are usually represented in a modified tableau of the form:

$$\begin{array}{c|c|cccc} c_1 & v_1 & x_{11} & x_{12} & \cdots & x_{1s} \\ c_2 & v_2 & x_{21} & x_{22} & \cdots & x_{2s} \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ c_s & v_s & x_{s1} & x_{s2} & \cdots & x_{ss} \\ \hline & & b_1 & b_2 & \cdots & b_s \end{array} .$$

The added restriction that the matrix X (whose $(i,j)th$ component is x_{ij}) in the PIRK schemes be strictly lower triangular results in the MIRK schemes which are a popular subclass of the IRK methods. The MIRK schemes are essentially a compromise between fully explicit RK methods and fully implicit RK methods, derived to achieve some of the computational efficiency of the former as well as some of the higher accuracy and stability characteristics of the latter. Cash and Singhal [18] and Van Bokhoven [11], discussed certain subclasses of IRK methods in the IVODE context.

In the IVP context, when the underlying discretization RK scheme is a MIRK scheme the set of m equations associated with the ith subinterval has the form (3.1), with stages,

$$\underline{k}_r = \underline{f}(t_i + c_r h, (1 - v_r)\underline{y}_i + v_r \underline{y}_{i+1} + h_i \sum_{j=1}^{r-1} x_{r,j}\underline{k}_j), \qquad r = 1, \cdots, s. \tag{3.7}$$

Note that the rth stage depends only on stages $1, \cdots, r-1$ and y_{i+1}. When a MIRK scheme is the underlying discretization scheme for a BVODE system, the set of equations associated with the ith subinterval has an identical form to (3.3):

$$\underline{\phi}(y_{i+1}, y_i) = \underline{y}_{i+1} - \underline{y}_i - h_i \sum_{r=1}^{s} b_r \underline{k}_r = \underline{0}, \tag{3.8}$$

where the stages k_r are identical to those in (3.7).

A modified Butcher tableau is used to represent the coefficients of the MIRK formulas and has the following structure:

$$\begin{array}{c|c|ccccc} c_1 & v_1 & 0 & 0 & 0 & \cdots & 0 \\ c_2 & v_2 & x_{21} & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \cdots & \cdots & \vdots \\ c_s & v_s & x_{s1} & x_{s2} & \cdots & x_{s,s-1} & 0 \\ \hline & & b_1 & b_2 & \cdots & \cdots & b_s \end{array} .$$

The introduction of the additional parameters v_r, $r = 1, \cdots, s$, allow for an explicit dependence on y_{i+1} in each stage. Gupta [29] discussed their use in the solution of BVODEs. Because the stages are defined explicitly in terms of y_i and y_{i+1}, in the BVODE context, these methods have approximately the same efficiency as the ERK methods. Burrage et al. [13], determine the maximum order for an s-stage MIRK scheme as well as a complete characterization of those methods having a number of stages $s \leq 5$. The derivation of optimal MIRK schemes from multi-parameter families is addressed by Muir [40] with several optimization criteria identified and applied in the derivation process.

A MIRK method is of p^{th} order if the numerical solution at the i^{th} mesh point obtained by solving (3.5) satisfies,

$$|y_i(t_{i+1}) - y_{i+1}| = O(h_i^{p+1}), \tag{3.9}$$

where $y_i(t)$ is the exact solution of the local IVODE,

$$\underline{y}(t) = \underline{f}(t, \underline{y}(t)), \qquad \underline{y}(t_i) = \underline{y}_i. \tag{3.10}$$

A family of MIRK schemes of a particular order p is derived by requiring its coefficient to satisfy a set of equations called *order conditions*, see e.g., Burrage et al. [13].

3.1.3 Continuous Runge-Kutta Methods

The use of the IRK methods as discretization schemes in the numerical solution of BVODEs produces discrete solution approximations at the mesh points. A continuous solution approximation can be very useful not just when the user requires solution information at off-mesh points, but also in certain processes within a BVODE code itself, for example, for error estimation, defect control, provision of initial estimates for Newton iterates, or mesh refinement and redistribution.

The idea of extending the discrete solution approximation to get a continuous solution first gained traction in the area of initial value ODE problems, with a number of authors (see, for example, [27]), having demonstrated the possibility of generating inexpensive interpolants for ERK formulas. A natural way to do this, which ties in with the one-step nature of the ERK method, is to construct a local solution approximation $u_i(t)$ on the step from t_i to t_{i+1}. A global approximation is obtained by joining these local continuous approximations in a piecewise fashion. The basic form of a continuous Runge-Kutta (CRK) scheme on the ith step, $[t_i,\ t_{i+1}]$, is a

polynomial in θ of the form:

$$\underline{u}_i(t_i + \theta h_i) = \underline{y}_i + h_i \sum_{r=1}^{s^*} b_r(\theta)\underline{k}_r, \qquad i = 0, \cdots, N-1, \tag{3.11}$$

where $0 \leq \theta \leq 1$ and $s^* \geq s$ is the total number of required stages. The stage values are defined in the same way as the IRK schemes in (3.5) above. If $s^* > s$, which is usually the case, then extra stages are computed in order to form the continuous extension. Observe that $u_i(t_i + \theta h_i)$ can be regarded as the result of step of length θh_i with an ERK scheme whose coefficients are $\{c_r/\theta\}_{r=1}^{s^*}$, $\{a_{rj}/\theta\}_{j=1,r=1}^{r-1,s^*}$ and $\{b_r(\theta)\}_{r=1}^{s^*}$. The conditions $b_r(1) = b_r$ for $r = 1, \cdots, s$, and $b_r(1) = 0$ for $r = s+1, \cdots, s^*$, ensure that (3.13) reduces to the basic formula (3.5) at $\theta = 1$.

3.1.4 Continuous Mono Implicit Runge-Kutta Methods

The CMIRK methods are a particular subclass of the CRK schemes. In [41], the class of CMIRK schemes is investigated. A summary of the authors work is done in Enright and Muir [26] who also discuss the application of CMIRK schemes to obtain continuous solution approximations to BVODE problems. The CMIRK interpolant is constructed by requiring it to satisfy the interpolatory conditions, $u(t_i) = y_i$, $u'(t_i) = f(t_i, y_i)$, and $u'(t_{i+1}) = f(t_{i+1}, y_{i+1})$ thus giving the scheme C^1 continuity over $[a, b]$. For the *ith* subinterval, the basic form of a CMIRK scheme is a polynomial in θ,

$$\underline{u}_i(t_i + \theta h_i) = \underline{y}_i + h_i \sum_{r=1}^{s^*} b_r(\theta)\underline{k}_r, \qquad 0 \leq \theta \leq 1, \qquad s^* \geq s, \tag{3.12}$$

with stages k_r, of the same form as those in (3.10). The remaining $s^* - s$ stages are defined by determining new coefficients v_r and $x_{r,j}$, $r = s+1, \cdots, s^*$, $j = 1, \cdots, r-1$.

The functions, $b_r(\theta)$, $r = 1, ..., s^*$, are weight polynomials of a certain degree related to the order of the CMIRK scheme. The coefficients are usually represented in a tableau having the structure:

$$\begin{array}{c|c|ccccc} c_1 & v_1 & 0 & 0 & 0 & \cdots & 0 \\ c_2 & v_2 & x_{21} & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \cdots & \vdots & \vdots \\ c_{s^*} & v_{s^*} & x_{s^*1} & x_{s^*2} & \cdots & x_{s^*,s^*-1} & 0 \\ \hline & & b_1(\theta) & b_2(\theta) & \cdots & b_{s^*-1}(\theta) & b_{s^*}(\theta) \end{array} .$$

A CMIRK scheme is of order p if we have

$$\max_{0 \le \theta \le 1} |\underline{y}_i(t_i + \theta h_i) - \underline{u}_i(t_i + \theta h_i)| = O(h_i^{p+1}), \tag{3.13}$$

where $\underline{y}_i(t)$ is the exact solution to the local initial value ODE,

$$\underline{y}'(t) = \underline{f}(t, \underline{y}(t)), \qquad \underline{y}(t_i) = \underline{y}_i.$$

To derive a p^{th} order CMIRK scheme, the stages and weight polynomials are required to satisfy continuous versions of the MIRK order conditions, as described in Muir and Owren [41]. The derivation of optimal CMIRK schemes from multi-parameter families was carried out by Muir [40], with the author identifying several optimization criteria which are then applied in the derivation process.

3.1.5 Brief Review of the BVP_SOLVER II Algorithm

This section briefly reviews the basic algorithm implemented in BVP_SOLVER II which employs MIRK and CMIRK schemes as the underlying discretization and interpolation schemes in the numerical solution of systems of BVODEs. The basic approach is to use the MIRK methods to determine a non-linear discrete algebraic system which can then be solved with a Newton iteration to obtain a discrete solution. Once this solution is obtained, a CMIRK scheme is employed to provide a continuous C^1 solution approximation over the problem interval for use in the computation of defect estimates, mesh redistribution, and initial guesses for subsequent Newton iterates. The boundary value ODEs to be solved are assumed to be expressible in the general form defined in section 2.1.1.

The standard approach involves a two level iteration scheme to describe the solution process [10]:

(0) Prior to beginning the two level iteration, a suitable initial mesh and associated initial guess for the discrete solution approximation is provided by the user.

(1) The first step of the upper level iteration is the setup and solution of a discrete system $\Phi(Y) = \underline{0}$ where Φ is the residual function (to be defined shortly) and $Y = [\underline{y}_0, \underline{y}_1, \cdots, \underline{y}_N].^T$ The residual function has $N+1$ components each of size m. There are N components, (3.8), associated with the N subintervals and one component corresponding to the boundary conditions. This discrete system is solved using a

modified Newton iteration which constitutes the lower level iteration.

(2) Upon convergence of the Newton iteration, we obtain the discrete solution $\{y_i\}_{i=0}^{N}$ which serves as a basis for the continuous solution approximations $\{u_i(t)\}_{i=0}^{N-1}$ based on a CMIRK scheme. The CMIRK scheme is used on each mesh subinterval to augment the discrete solution, leading to a C^1-continuous interpolant over the whole problem interval. The continuous solution approximation, $u(t)$, is then the piecewise polynomial defined by the collection of local continuous solution approximations $\{u_i(t)\}_{i=0}^{N-1}$ and has the same order of accuracy as the underlying discrete solution.

The defect on each subinterval is sampled in order to obtain an estimate of the maximum defect in a given mesh. The algorithm terminates if the estimate of the maximum defect on each subinterval is within a given user defined tolerance, TOL.

(3) If the above criterion isn't met, the algorithm determines a new mesh, with redistributed mesh points and possibly a different number of points, to attempt to ensure that the maximum defect estimates will be approximately the same on each subinterval and that they will each be less than the user tolerance.

(4) After a new mesh is determined, the continuous solution approximation is used to compute an initial iterate for the next Newton iteration.

3.2 Defect Control BVODE Software

3.2.1 MIRKDC

MIRKDC [26] is a FORTRAN 77 defect control code which implements MIRK and CMIRK schemes in the numerical solution of BVODEs. The basic algorithm employed in the software package uses MIRK formulas to discretize the ODE system, a process which together with the boundary conditions gives a nonlinear system for the solution approximations at the mesh points. Once this solution is obtained, a CMIRK scheme is used to provide a polynomial solution approximation over each subinterval. MIRKDC provides the option of second, fourth and sixth order, symmetric MIRK methods as discretization schemes. Symmetric methods are those which are invariant regardless of the sign of h. The stages of the MIRK schemes are embedded within the CMIRK scheme in the construction of the continuous solution. Reusing the stages of the MIRK scheme is computationally efficient. MIRKDC implements a hybrid damped Newton and fixed Jacobian iteration combination, with a switching scheme, to solve the nonlinear system obtained from the discretization process. The Jacobian matrices arising from the nonlinear system possess a special sparsity structure known as almost block diagonal [20] and specialized software COLROW [20], designed to handle these type of structures, is employed. Both the termination criterion for the overall computation and the mesh selection algorithm require an estimate of the maximum defect on each subinterval. In MIRKDC this is done by sampling the defect at two points within each subinterval. The estimate of the maximum defect is required to satisfy a user provided tolerance.

3.2.2 Defect Control MATLAB Software: bvp4c, bvp5c, bvp6c

The MATrix LABoratory or MATLAB numerical codes bvp4c by Kierzenka and Shampine [33], bvp5c by Kierzenka and Shampine [34], and bvp6c by Hale and Moore [30], are other examples of defect control solvers. The bvp4c and bvp6c codes are based on MIRK schemes and do not attempt to *directly* control the global error (GE), while bvp5c is based on a four point Lobatto collocation formula. All three codes control an estimate of the maximum defect in the computed solution as a measure of solution quality, although, as is shown in [33], bvp5c also simultaneously controls an estimate of the global error (GE) as well. The relationship between the defect and GE is considered in [33], where it is shown that a scaled norm of the defect asymptotically approaches the norm of the GE.

3.2.3 BVP_SOLVER

BVODE codes such as MIRKDC and other numerical software packages such as PASVA3/BVPFD [35], COLSYS/COLNEW [2, 3, 8], and TWPBVP [19] possess very complex user interfaces that often deter most potential users from investing the time needed to learn how to use them properly. In order to broaden their appeal to a larger audience, these interfaces (consisting mainly of argument lists and subroutine) through which the user communicates with the software, must be drastically simplified. Drawing upon their experience in writing user interfaces for ODE solvers in Matlab and Fortran 90/95, Shampine, Muir and Xu [45] developed a user-friendly Fortran 90/95 BVP solver based on an extensive modification of MIRKDC. This project was related to earlier work by Kierzenka and Shampine [33], which exploited

the capabilities of the MATLAB programming environment to obtain solvers with greatly simplified BVODE solver interfaces. The authors, in the course of developing a completely new user interface, also added significantly to the algorithmic capabilities of MIRKDC by taking advantage of certain properties of the Fortran 90/95 programming language. Their effort culminated in the production of the BVP_SOLVER software which features a substantial reduction in the number of user supplied subroutines, as well as a vastly simplified argument list. The latter was achieved by exploiting features of the Fortran 90/95 such as dynamically allocated arrays and modules which replaced static work arrays and common blocks in MIRKDC. In addition, all low level linear algebra subroutines were replaced with calls to intrinsic array functions thus improving greatly the maintainability of the code for future development. BVP_ SOLVER I also extends the class of BVPs solved by MIRKDC to problems with unknown parameters and singular coefficients. This extended problem class has the form,

$$\underline{y}'(t) = \frac{1}{t-a} S\underline{y}(t) + \underline{f}(t, \underline{y}(t), \underline{p}), \tag{3.14}$$

subject to general nonlinear separated boundary conditions,

$$\underline{g}_a(\underline{y}(a), \underline{p}) = \underline{0}, \qquad \underline{g}_b(\underline{y}(b), \underline{p}) = \underline{0}, \tag{3.15}$$

where S is an optional $m \times m$ matrix and $\underline{\mathrm{p}}$ is an optional vector of unknown parameters. It also uses improved MIRK and CMIRK formulas [40]. In particular, the sixth order case is an improvement on the corresponding formula employed in MIRKDC because it requires one less stage evaluation. The solver also provides a global error

estimate based on Richardson extrapolation and a conditioning constant estimate as well. The software package has the additional convenience of auxiliary routines which evaluate the solution and its derivative, save and retrieve solution information, and facilitate continuation in the length of the problem interval.

3.2.4 BVP_SOLVER II

The BVODE software package BVP_SOLVER II, by Boisvert, Muir, Spiteri [10], represents an expansion in the error controlling capabilities of BVP_SOLVER. It provides the user with the possibility of computing a defect controlled numerical solution as well as an option for controlling an estimate of the GE of the numerical solution. This software upgrade modifies the BVP_SOLVER I code to include hybrid defect control/GE control by introducing implementations of three GE estimation schemes as alternatives to Richardson extrapolation for the a posteriori estimate of the GE. These schemes are based on (i) the direct use of a higher order discretization formula, (ii) the use of a higher order discretization formula within a deferred correction framework, and (iii) the product of an estimate of the maximum defect and an estimate of the BVODE conditioning constant. The BVP_SOLVER II code also possesses an option for the estimation and control of the GE meaning that this new version provides options for GE control, defect control, as well as hybrid combinations of both of these measures of solution accuracy.

Chapter 4

Derivation and Analysis of Asymptotically Correct Defect Estimation Schemes

4.1 Detailed Description of Maximum Defect Estimation Process

The defect $\delta(t)$ described in the previous chapter is a continuous function over the problem interval and provides a measure of the quality of the computed solution. The central idea behind all defect control solvers is to adaptively choose a mesh which approximately equidistributes the defect across all subintervals so that, for the final accepted numerical solution, an estimate of the maximum defect over the entire problem domain is bounded by a user-provided tolerance. It is an essential requirement, therefore, for defect control based solvers to accurately and efficiently estimate the maximum defect on each subinterval. It is straightforward to compute $\delta(t)$ at any point in the domain; however the bigger challenge is to determine, in an efficient manner, the maximum value of the defect on each subinterval. When a standard CMIRK interpolant is employed for $u(t)$, the usual approach is to simply sample the defect at a small number of points on each subinterval with the hope that one of the points will be close enough to the location of the true maximum defect. In order for the estimation process to be reasonably efficient the number of sample estimates

must be kept reasonably small. Given that the maximum value of the defect can be located anywhere within a given subinterval, there is no particular justification that any of the sampling points selected will coincide with the location of the maximum defect. It was observed in Enright and Muir [23], that the true maximum defect in some cases exceeded the estimated maximum defect by an order of magnitude. The implication of this observation is that a defect control code, employing standard defect sampling for the estimation process, may accept a numerical solution for which the defect is in fact substantially larger than the user tolerance. This underestimate of the maximum defect can impact negatively on the performance of the rest of the computation because the mesh selection algorithm will not have access to a good profile of the defect over the subintervals of the mesh.

Since the continuous solution approximation, $u(t)$, is based on a continuous Runge-Kutta scheme (3.11), (3.4), it is possible to express the defect in terms of the coefficients of the scheme. Let $u_i(t)$ be an approximation to the exact solution, $z_i(t)$, of the local initial value problem (IVP)

$$z_i' = f(t, z_i), \qquad z_i(t_i) = y_i, \qquad t \in [t_i, t_{i+1}]. \tag{4.1}$$

The continuous error of $u_i(t)$ on the ith subinterval is (see equation (3.13))

$$u_i(t) - z_i(t) = O(h_i^{p+1}). \tag{4.2}$$

Similarly, the derivative of this numerical solution satisfies

$$u_i'(t) - z_i'(t) = O(h_i^p), \tag{4.3}$$

since the variables t and θ are related by the equations $t = t_i + \theta h$, $\theta = \frac{1}{h}(t - t_i)$ and $\frac{d\theta}{dt} = \frac{1}{h}$. Hence the right hand side of (4.9) is reduced by a factor of h.

Recall that the defect of the numerical solution, $u_i(t)$, on the ith subinterval is described by the equation

$$\delta_i(t) = u_i'(t) - f(t, u_i(t)). \tag{4.4}$$

Taking advantage of the fact that $z_i(t)$ is the exact solution of (4.1), (4.4) can be written as

$$\delta_i(t) = u_i'(t) - f(t, u_i(t)) + f(t, z_i(t)) - z_i'(t). \tag{4.5}$$

A slight rearranging of (4.5) gives

$$\delta_i(t) = u_i'(t) - z_i'(t) - (f(t, u_i(t)) - f(t, z_i(t))) . \tag{4.6}$$

Imposing a Lipschitz assumption [1] on f, the second term in (4.6) then can be seen to be of $O(h_i^{p+1})$ and the defect can hence be written as

$$\delta_i(t) = u_i'(t) - z_i'(t) + O(h_i^{p+1}). \tag{4.7}$$

The leading term in the defect (4.7) is thus $O(h_i^p)$ from (4.3). Furthermore the leading order term in the defect can be seen to be equal to the leading order term in the error for $u_i'(t)$. When $u_i(t)$ is based on a CMIRK scheme, the leading error term is known from the theory of Runge-Kutta methods [14].

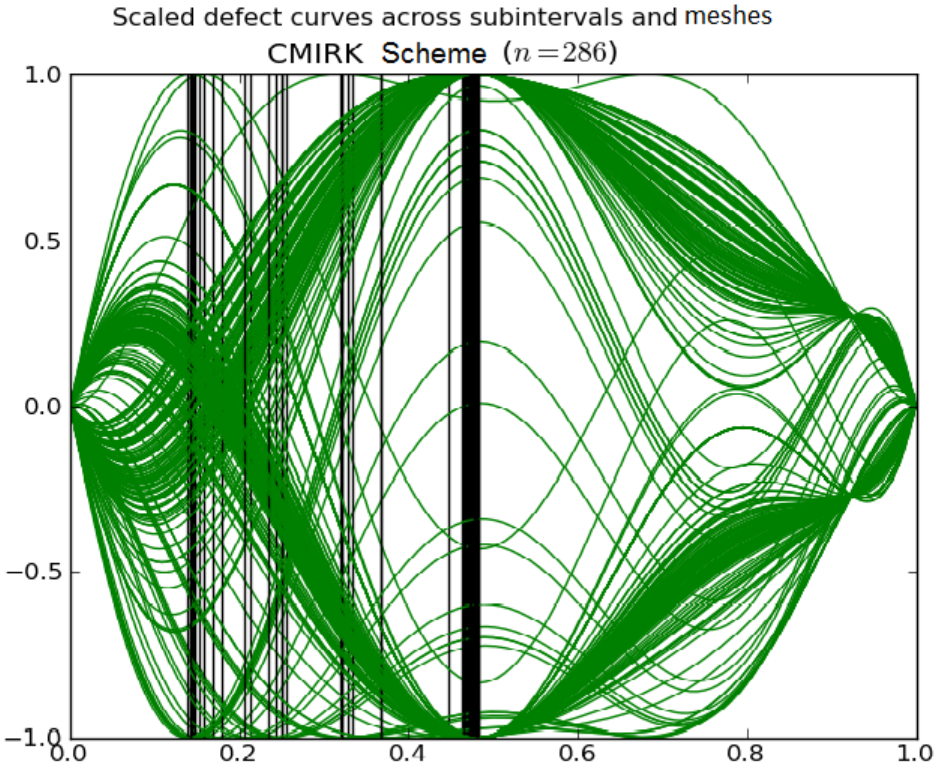
On the ith subinterval the defect, denoted by $\delta_i(t)$, can be expressed in an expansion that is related to the local error expansion of the appropriate solution. It has the form,

$$\delta_i(t) = \left(\sum_{j=0}^{\rho} q_j(\theta) F_j \right) h_i^p + O(h_i^{p+1}), \tag{4.8}$$

where p is the order of the Runge-Kutta scheme, the $q_j(\theta)$'s are polynomials of degree p dependent on the continuous Runge-Kutta (CRK) scheme but independent of the problem or h_i, the F_j's are elementary differentials [14] which depend only on the problem and $\rho + 1 > 1$ is the number of elementary differentials of $(p+1)st$ order. As $h_i \to 0$, it is evident from (4.1) that the value of the defect will approach a linear combination of the $q_j(\theta)$ values, where the coefficients of this linear combination depend on the values of the elementary differentials, F_j. Since the F_j values depend on the problem, it is impossible to predict on each subinterval which of these elementary differentials is the largest in value. The location of the maximum will hence vary from subinterval to subinterval depending on the problem. This means that on any given subinterval it is virtually impossible to make an *a priori* determination of where the maximum value of the leading term of the defect will occur. Figure 4.1 is a graphical illustration of the typical behavior of the defect in this case. In order to produce this plot, the defect estimates in each subinterval across all meshes considered in the computation of the numerical solution by BVP_SOLVER II were scaled (normalized) by dividing through by the maximum defect for that subinterval. This was done to ensure that the maximum defect peaks at 1 or -1. The curves of the normalized defect

for each subinterval (the total number of subintervals considered is denoted by n) are then superimposed on each other on the interval [0,1].

Figure 4.1: Plot of the results for test problem IV with $\epsilon = 10^{-2}$ using BVP_SOLVER II with fourth order schemes and $TOL = 10^{-7}$.



4.2 Hermite - Birkhoff Interpolants Derived Via Bootstrapping Process

The discussion at the end of the previous section demonstrates precisely why defect estimation using the standard CMIRK schemes isn't a good idea. Since the defect of the numerical solution is the measure of solution quality and control in the BVODE solvers MIRKDC, and BVP_SOLVER I, it was imperative therefore to derive new types of interpolants capable of providing inexpensive and accurate estimations of the maximum defect. A particularly promising line of research was the derivation

of interpolants with vastly simplified expressions for the defect expansion. The intuitiveness of this approach is based on the fact that since the continuous solution approximations are based on Runge-Kutta schemes (3.4),(3.5), the defect (4.1) on each subinterval can be expressed in an expanded form as:

$$\delta_i(t) = (q_0(\theta)F_0 + q_1(\theta)F_1 + \cdots + q_\rho(\theta)F_\rho)\, h_i^p + O(h_i^{p+1}). \tag{4.9}$$

Let the coefficient of h_i^p in the leading order term be

$$G(t) = q_0(\theta)F_0 + q_1(\theta)F_1 + \cdots + q_\rho(\theta)F_\rho. \tag{4.10}$$

A careful examination of (4.3) reveals a strong correlation between the number of terms and the ease with which the term contributing the maximum value to the leading term of the defect expansion can be identified. This relationship provided an avenue of research into ways by which the expression $G(t)$ can be simplified. The authors Enright and Muir [23] describe one approach in which an interpolant with a greatly simplified expression corresponding to (4.3) is derived. Starting with a standard CMIRK scheme, they employ a boot-strapping algorithm developed by Enright et al. [27] to derive a special type of interpolant expressed in the form of a Hermite-Birkhoff interpolant [27]. These special interpolants yield a defect for which the location of the maximum defect on each subinterval can be determined (at least asymptotically) in an *a priori* manner. The estimate of the maximum defect obtained in this case is said to be asymptotically correct.

The general form of a Hermite- Birkhoff scheme on the subinterval $[t_i, t_{i+1}]$, with $0 \leq \theta \leq 1$ is:

$$\tilde{u}_i(t_i + \theta h_i) \quad = \quad d_0(\theta) y_i + d_1(\theta) y_{i+1} + h_i \sum_{r=1}^{\tilde{s}^*} \tilde{b}_r(\theta) k_r, \tag{4.11}$$

where the k_r's have the same general form as (3.6), $d_0(\theta)$, $d_1(\theta)$, $\{\tilde{b}_r(\theta)\}_{r=1}^{\tilde{s}^*}$, are polynomials in θ and $\tilde{s}^*$ is the total number of required stages. The determination of the required stages and weight polynomials as described by Enright and Muir [23], is done by requiring the interpolant (4.4) and its derivative to satisfy certain interpolation conditions at a number of points within the i^{th} subinterval. This process is detailed in sections 4.2.1 and 4.2.3, during the derivation of fourth and sixth order Hermite-Birkhoff schemes.

The *asymptotically correct* quality possessed by the Hermite-Birkhoff interpolants is essentially a consequence of the vast simplification of the *coefficient function*, $G(t)$ (4.3). Guided by work done earlier in Enright and Hayes [28], Enright and Muir [23], derived a sixth order Hermite-Birkhoff interpolant (leading to an asymptotically correct maximum defect estimate) by employing a bootstrapping process. This led to an interpolant with only a single term contributing to the leading coefficient in the expansion of the associated defect. The problem of locating the maximum defect was now essentially one of locating the maximum of the polynomial in the leading term of the defect expansion which is relatively easy to compute. We discuss the process through which this special interpolant was obtained, in detail, in Section 4.2.1.

It is a relatively straightforward process to convert Hermite-Birkhoff form of $\tilde{u}(t)$ to its CMIRK equivalent. By substituting for $\underline{y}_{i+1}$ in the discrete formula

$$\underline{y}_{i+1} \quad = \quad \underline{y}_i + h_i \sum_{r=1}^{s} b_r \underline{k}_r, \tag{4.12}$$

in (4.4) and noting the interpolation condition $d_0(\theta) + d_1(\theta) = 1$, the CMIRK form of $\underline{\tilde{u}}(t)$ can be written as:

$$\underline{\tilde{u}}_i(t_i + \theta h_i) \quad = \quad \underline{y}_i + h_i \sum_{r=1}^{\tilde{s}^*} (b_r d_1(\theta) + \tilde{b}_r(\theta)) \underline{k}_r. \tag{4.13}$$

Given the relative ease of conversion, and also that the changes to BVP_SOLVER II will be minimal in comparison, the implementation of (4.6) as the primary interpolant in the BVP_SOLVER series seemed a forgone conclusion. However, it is pointed out in [23] that the lack of an explicit dependence on y_{i+1} in (4.6) means that $\underline{\tilde{u}}(t)$ may have discontinuities that are the size of the Newton tolerance (used to solve the nonlinear system for the $\{\underline{y}_i\}_{i=0}^{N}$) at the mesh points. Furthermore, it introduces an additional error of $O(h^{p+1})$ associated with the error for y_{i+1} from the discrete formula. On the other hand, since $\tilde{u}(t)$ in (4.4) has an explicit dependence on y_{i+1}, the interpolant and its first derivative will be continuous across each internal mesh point.

The remainder of this chapter provides a detailed account of the approach implemented by Enright and Muir [23] in the derivation of a sixth order Hermite-Birkhoff

scheme. This is then followed by the development of second and fourth order Hermite-Birkhoff schemes using the boot-strapping scheme, that lead to asymptotically correct estimates of the maximum defect on each subinterval.

4.2.1 Derivation of a Sixth Order Hermite-Birkhoff Interpolant

In the course of deriving $\tilde{u}_i(t)$, several considerations were taken into account by the authors [23].

1. The standard sixth order interpolant, $u_i(t)$, computed by BVP_ SOLVER II is of degree six and involves the two stages $k_1 = f(t_i, y_i)$, $k_2 = f(t_{i+1}, y_{i+1})$, the three stages computed for use with the MIRK scheme, k_3, k_4, k_5 and the three additional stages k_6, k_7, k_8 needed for the CMIRK method.

2. The new interpolant involves the same y_i, y_{i+1}, k_1, and k_2 values and the boot-strapping process is employed to define four new stages *based* on *evaluations* of *the standard interpolant*. The new stages are assigned the corresponding abscissa c_9, c_{10}, c_{11} and c_{12} and are constructed based on the evaluations of $u_i(t)$ at these abscissas. On the *ith* subinterval, they are of the form

$$k_{8+j} = f(t_i + c_{8+j}h_i, u_i(t_i + c_{8+j}h_i)), \tag{4.14}$$

   where $j = 1, 2, 3, 4$. The abscissa values, $\frac{7}{100}$, $\frac{14}{100}$,$\frac{86}{100}$,$\frac{93}{100}$ for c_9, c_{10}, c_{11}, c_{12}, are chosen so that the size of leading coefficient of the defect expansion is significantly larger than the coefficients in the next higher order term.

3. Next, eight interpolatory conditions are imposed in order to determine $\tilde{u}_i(t)$:

$\tilde{u}_i(t)$ is the unique polynomial of degree at most seven that satisfies $\tilde{u}_i(t_i) = y_i$, $\tilde{u}_i(t_{i+1}) = y_{i+1}$, $\tilde{u}_i'(t_i) = f(t_i, y_i) = k_1$, $\tilde{u}_i'(t_{i+1}) = f(t_{i+1}, y_{i+1}) = k_2$ and for $j = 1, 2, 3, 4$,

$$\tilde{u}_i'(t_i + c_{8+j}h_i) = f(t_i + c_{8+j}h_i, u_i(t_i + c_{8+j}h_i)) = k_{8+j}. \tag{4.15}$$

Then the Hermite-Birkhoff representation of this interpolant has the form

$$\begin{aligned} \tilde{u}_i(t_i + \theta h_i) &= d_0(\theta)y_i + d_1(\theta)y_{i+1} \\ &\quad + h_i \Big(\tilde{b}_1(\theta)k_1 + \tilde{b}_2(\theta)k_2 + \tilde{b}_9(\theta)k_9 + \tilde{b}_{10}(\theta)k_{10} \\ &\quad + \tilde{b}_{11}(\theta)k_{11} + \tilde{b}_{12}(\theta)k_{12} \Big), \end{aligned} \tag{4.16}$$

where $d_0(\theta)$, $d_1(\theta)$, $\tilde{b}_1(\theta)$, $\tilde{b}_2(\theta)$, $\tilde{b}_9(\theta)$, $\cdots$ $\tilde{b}_{12}(\theta)$ are weight polynomials of degree seven obtained in a straightforward fashion from the interpolation conditions.

4. Since $u_i(t)$ is a sixth order CMIRK scheme, each evaluation has an error that is $O(h_i^7)$ and with a Lipschitz assumption on f, the error in each of the stages k_2, $k_9, \cdots, k_{12}$ defined in (4.15) is $O(h_i^7)$ as well. Given that the y_i and k_1 terms are assumed to be exact and do not contribute to the local error, every other term in (4.16) with the exception of the $d_1(\theta)y_{i+1}$ term contributes an error of $O(h_i^8)$ to the Hermite-Birkhoff scheme, since all stages are multiplied by h_i.

5. We also note from standard interpolation theory that the interpolation error associated with $\tilde{u}_i$ is $O(h_i^8)$. Hence $d_1(\theta)y_{i+1}$ is the term contributing the largest

data error, of $O(h_i^7)$, to $\tilde{u}_i(t)$. Therefore, the continuous error of $\tilde{u}_i(t)$ is

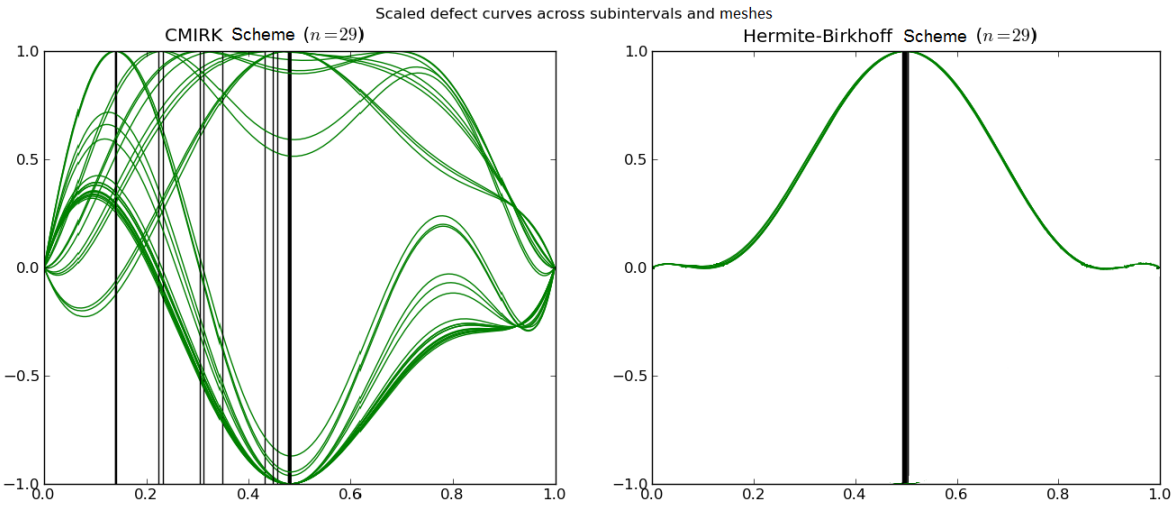
$$\tilde{u}_i(t) - z_i(t) = d_1(\theta) C_i {h_i}^7 + O(h_i^8), \tag{4.17}$$

where C_i is associated with the error in y_i. Thus from (4.7) the defect of $\tilde{u}_i(t)$ satisfies

$$\tilde{\delta}(t) = \tilde{u}_i'(t) - z_i'(t) = d_1'(\theta) C_i h_i^6 + O(h_i^7) = \tilde{q}_1(\theta) C_i h_i^6 + O(h_i^7), \tag{4.18}$$

where $\tilde{q}_1(\theta) = d_1'(\theta)$ is a polynomial of degree six. Hence, for this scheme (referring to (4.10)), we have $\tilde{G}(t) = \tilde{q}_1(\theta) C_i$. The simplification of $\tilde{G}(t)$ to a single term means that, assuming that $C_i \neq 0$, the defect for any $t \in [t_{i-1}, t_i]$ will be a multiple of the polynomial $d_1'(\theta)$ for sufficiently small h_i. This means that an asymptotically correct estimate of the maximum magnitude of the defect can be obtained on each subinterval at the extremum of $d_1'(\theta)$ for $\theta \in [0, 1]$. The implication of this is that as $h_i \to 0$, the maximum defect will occur at the same place within every subinterval for every problem. In the sixth order case, the local maximum of $d_1'(\theta)$ occurs at $\theta = \frac{1}{2}$. Figure 4.2 illustrates the typical behavior of the defect for both $u_i(t)$ and $\tilde{u}_i(t)$ when h_i is sufficiently small.

Figure 4.2: Plot of the normalized defect for problem V over all subintervals for sixth order CMIRK and Hermite-Birkhoff schemes.



4.2.2 Derivation of a Fourth Order Hermite-Birkhoff Scheme

As mentioned earlier, the methodology implemented in the derivation of the fourth order Hermite-Birkhoff scheme closely mirrors the approach employed by Enright and Muir [23] in the development of the sixth order scheme.

1. The fourth order Hermite-Birkhoff scheme is developed using the standard fourth order CMIRK interpolant as a basis. The latter scheme depends on y_i, y_{i+1}, the stages $k_1 = f(t_i, y_i)$ and $k_2 = f(t_{i+1}, y_{i+1})$, the third stage k_3, computed for use with the discrete MIRK scheme and one additional stage, k_4, for the CMIRK method.

2. The fourth order Hermite-Birkhoff scheme utilizes y_i, y_{i+1}, k_1, k_2, and two additional stages constructed using the boot-strapping algorithm described in

[27]. The extra stages associated with the abscissa values $c_5 = \frac{86}{100}$ and $c_6 = \frac{93}{100}$, are based on evaluations of the underlying CMIRK scheme and are of the form,

$$k_{4+j} = f(t_i + c_{4+j}h_i, u_i(t_i + c_{4+j}h_i)), \tag{4.19}$$

where $j = 1, 2$.

3. Imposing the appropriate interpolatory conditions makes $\tilde{u}_i(t)$ the unique polynomial of degree at most five which satisfies $\tilde{u}_i(t_i) = y_i$, $\tilde{u}_i(t_{i+1}) = y_{i+1}$, $\tilde{u}_i'(t_i) = f(t_i, y_i)$, $\tilde{u}_i'(t_{i+1}) = f(t_{i+1}, y_{i+1})$ and, for $j = 1, 2$,

$$\tilde{u}_i'(t_i + c_{4+j}h_i) = f(t_i + c_{4+j}h_i, u_i(t_i + c_{4+j}h_i)). \tag{4.20}$$

Note that the right hand side of (4.20) involves evaluations of the CMIRK scheme.

Then the Hermite-Birkhoff representation of this interpolant has the form,

$$\begin{aligned} \tilde{u}_i(t_i + \theta h_i) &= d_0(\theta)y_i + d_1(\theta)y_{i+1} \\ &\quad + h_i\left(\tilde{b}_1(\theta)k_1 + \tilde{b}_2(\theta)k_2 + \tilde{b}_5(\theta)k_5 + \tilde{b}_6(\theta)k_6\right), \end{aligned} \tag{4.21}$$

where $d_0(\theta)$, $d_1(\theta)$, $\tilde{b}_1(\theta)$, $\tilde{b}_2(\theta)$, $\tilde{b}_5(\theta)$, and $\tilde{b}_6(\theta)$ are weight polynomials of degree five, obtained from the interpolation conditions.

4. Since $u_i(t)$ is a fourth order CMIRK scheme, each evaluation of this scheme as well as the stages k_2, k_5, and k_6 (with a Lipschitz assumption on f) has an error

that is $O(h_i^5)$. Therefore the error contributions of the terms $h_i k_2$, $h_i k_5$, and $h_i k_6$ are $O(h_i^6)$ while the y_i and k_1 terms are considered exact and so contribute no data error to $\tilde{u}_i(t)$.

5. We note also that, from standard interpolation theory, the interpolation error associated with $\tilde{u}_i$ is $O(h_i^6)$. Thus the term $d_1(\theta)y_{i+1}$ contributes the largest data error of $O(h_i^5)$ to the new interpolant $\tilde{u}_i(t)$. The continuous local error is

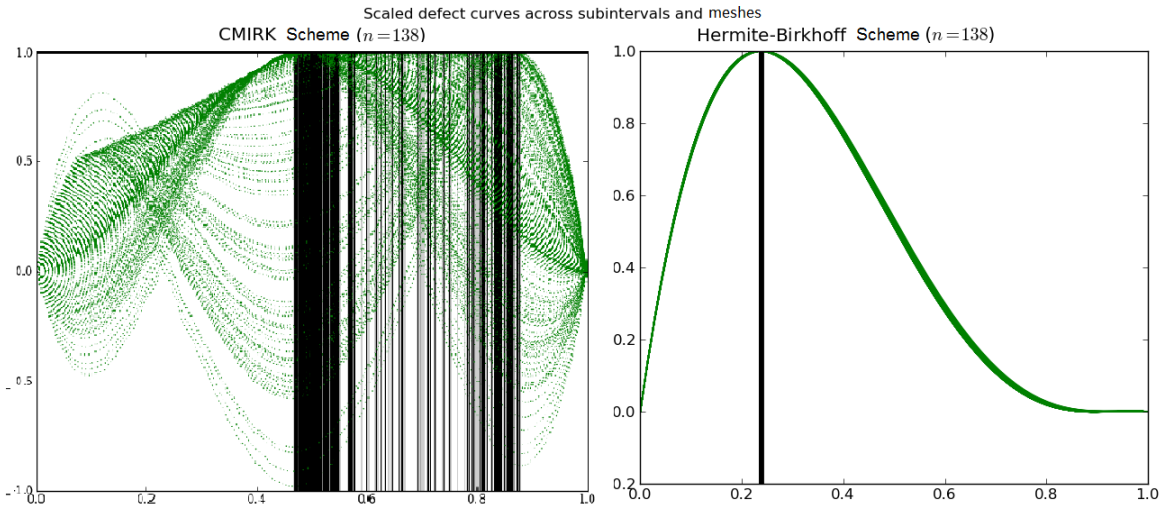
$$\tilde{u}_i(t) - z_i(t) = d_1(\theta) C_i {h_i}^5 + O(h_i^6), \tag{4.22}$$

where C_i is associated with the error for y_i, and from (4.7) the defect of $\tilde{u}_i(t)$ satisfies

$$\tilde{\delta}(t) = \tilde{u}_i'(t) - z_i'(t) = \tilde{q}_1(\theta) C_i h_i^4 + O(h_i^5), \tag{4.23}$$

where $\tilde{q}_1(\theta) = d_1'(\theta)$ is a polynomial of degree four. Hence $\tilde{G}(t) = \tilde{q}_1(\theta)C_i$. As in the sixth order case, an asymptotically correct estimate of the maximum magnitude of the defect will coincide on each subinterval with the extremum of $d_1'(\theta)$ for $\theta \in [0, 1]$. The implication of this is that as $h_i \to 0$, the maximum defect will occur at the same place within every subinterval for every problem. The local maximum for $d_1'(\theta)$ occurs at $\theta \approx 0.231$, which corresponds to the maximum of the polynomial $\tilde{q}_1(\theta) = d_1'(\theta)$. Figure 4.3 illustrates the typical behavior of the defect for $u_i(t)$ and $\tilde{u}_i(t)$ when h_i is sufficiently small.

Figure 4.3: Plot of the normalized defect for problem I over all subintervals for fourth order CMIRK and Hermite-Birkhoff schemes.



4.2.3 Second Order

The second order CMIRK scheme by default already possesses the characteristics which yield an asymptotically correct estimate of the maximum defect, hence the boot-strapping process isn't required in its derivation. However, it does not explicitly depend on y_i meaning that the interpolant has a discontinuity at right hand mesh point of each subinterval that will be of the order of the Newton tolerance applied in the computation of the discrete solution. Thus a Hermite-Birkhoff interpolant is preferred and it is derived directly as follows.

1. The second order Hermite-Birkhoff scheme utilizes the y_i, y_{i+1}, k_1 and k_2 and imposing the appropriate interpolatory conditions makes $\tilde{u}_i(t)$ the unique polynomial of at most degree three which satisfies the conditions, $\tilde{u}_i(t_i) = y_i$, $\tilde{u}_i(t_{i+1}) = y_{i+1}$, $\tilde{u}_i'(t_i) = f(t_i, y_i)$, and $\tilde{u}_i'(t_{i+1}) = f(t_{i+1}, y_{i+1})$.

$$\tilde{u}_i(t_i + \theta h_i) \;=\; d_0(\theta)y_i + d_1(\theta)y_{i+1} + h_i\left(\tilde{b}_1(\theta)k_1 + \tilde{b}_2(\theta)k_2\right) \qquad (4.24)$$

where $d_0(\theta)$, $d_1(\theta)$, $\tilde{b}_1(\theta)$, and $\tilde{b}_2(\theta)$ are weight polynomials of degree three obtained by requiring $\tilde{u}_i(t)$ to interpolate y_i, y_{i+1} and $\tilde{u}_i'(t)$ to interpolate k_1 and k_2.

2. The terms y_i and k_1 are considered to be exact for the local solution and so contribute no data error. The error contribution from the stage k_2 is also $O(h_i^3)$, hence the term $h_i k_2$ contributes an error of $O(h_i^4)$.

3. We note also that, from standard interpolation theory, the interpolation error associated with $\tilde{u}_i$ is $O(h_i^4)$. Therefore the largest contributor of data error to $\tilde{u}_i(t)$ is the $d_1(\theta)y_{i+1}$ term with an error of $O(h_i^3)$. The continuous local error is thus

$$\tilde{u}_i(t) - z_i(t) = d_1(\theta)C_i{h_i}^3 + O(h_i^4), \qquad (4.25)$$

where C_1 is associated with the error in y_i.

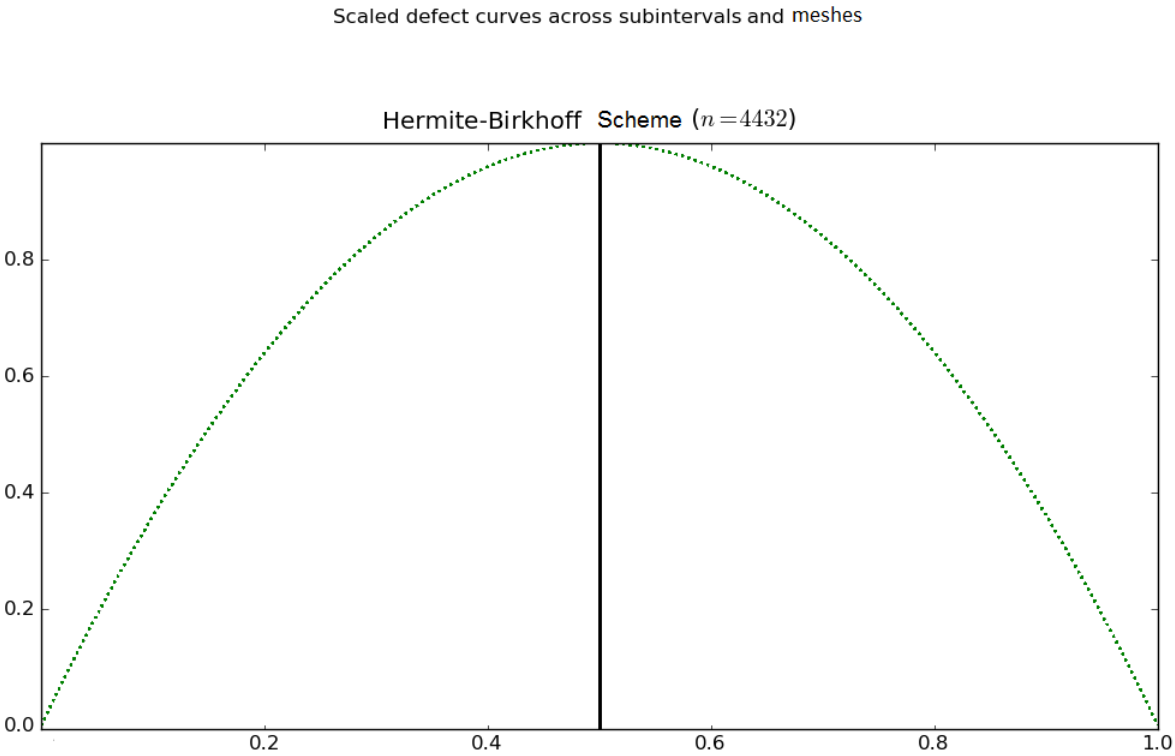
The defect of $\tilde{u}_i(t)$ satisfies

$$\tilde{\delta}(t) = \tilde{u}_i'(t) - z_i'(t) = d_1'(\theta)C_i h_i^2 + O(h_i^3). \qquad (4.26)$$

Therefore as h_i becomes sufficiently small, the location of the maximum defect on each subinterval for any problem will coincide with the extremum of the polynomial, $\tilde{q}_1(\theta)$

$= d_1'(\theta)$. The local maximum of this polynomial is at $\theta = \frac{1}{2}$. Figure 4.4 illustrates the typical defect behavior for $\tilde{u}_i(t)$ for sufficiently small h_i values.

Figure 4.4: Plot of the normalized defect for problem III over all subintervals for the second order Hermite-Birkhoff scheme.



4.2.4 Validity Check

In this subsection we discuss the implementation of the auxiliary process called a validity check. This process monitors the accuracy and robustness of the new defect sampling process by checking the value of the defect estimate at a point known as

validity check sampling point. The value of the defect estimate evaluated at this point should be half the value of the defect sampled at the asymptotically correct sample point where we expect the defect to have its maximum value, in the same subinterval. Put in other words, the defect of the interpolant $\tilde{u}_i(t)$ is also computed at a second predetermined spot within each subinterval. If the subinterval size is sufficiently small that we are within the asymptotic regime of the formula, then the value of the defect at this location should be half that of the value of the maximum defect for the same subinterval. The auxiliary validity check process was discussed in Enright and Muir [23], who observed that the successful defect estimation rate of the sixth order Hermite-Birkhoff for the final converged mesh was around 83% for a collection of test problems. Closer examination revealed that the subintervals where the estimation failed were relatively large and thus the associated computation wasn't within the asymptotic regime for the formula. Hence the error contribution from the higher order terms was significant enough to interfere with the dominance of the leading order term in the defect expansion. The validity check process implemented in our software (optionally) allows the user to check, for the final converged mesh, which subintervals satisfy the validity check and flag suspect subintervals. The validity check routine provides an additional layer of confidence for the defect sampling and control process.

4.3 Chapter Comments

In the course of constructing the Hermite-Birkhoff scheme only the left and right discrete solution values y_i and y_{i+1} together with the stages, $k_1 = f(t_i, y_i)$ and

$k_2 = f(t_{i+1}, y_{i+1})$ from the standard interpolant are utilized. Therefore the weight polynomials $b_r(\theta)$ corresponding to the discrete and continuous MIRK schemes are all zero in the Hermite-Birkhoff scheme. Only the Hermite-Birkhoff weight polynomials $\{d_j(\theta)\}_{j=1}^{2}$ and $\{\tilde{b}_r(\theta)\}_{r=s^*+1}^{\tilde{s}^*}$ are used to define the Hermite-Birkhoff scheme.

Chapter 5

Test Problems

This thesis utilizes a set of five test boundary value problems in the conduction of various numerical experiments on the software package. The five test problems have been chosen from a suite of test problems used consistently in testing earlier versions of BVP_SOLVER II. They are a typical mixture of scientific or engineering oriented problems and, unless otherwise stated, have no known closed form solution. Though each problem is described in its original form here, each was rewritten in the first order form required by BVP_SOLVER II.

5.1 Test Problem I

Test problem I is from Ascher et al. [1], Example 1.20. The problem considers the steady flow of a viscous incompressible axisymmetric (swirling flow) fluid between two rotating coaxial disks located at $x = 0$ and at $x = 1$. The BVODE is described by two equations and a total of six boundary conditions:

$$\epsilon f'''' + f f''' + g g' = 0, \qquad 0 < x < 1, \tag{5.1}$$

$$\epsilon g'' + f g' + f' g = 0, \qquad 0 < x < 1, \tag{5.2}$$

with boundary conditions

$$f(0) = f(1) = f'(0) = f'(1) = 0, \tag{5.3}$$

$$g(0) = \Omega_0, \qquad g(1) = \Omega_1, \tag{5.4}$$

where Ω_0 and Ω_1 are the angular velocities of the infinite disks, $|\Omega_0| + |\Omega_1| \neq 1$ and ϵ is a viscosity parameter, $0 < \epsilon << 1$.

5.2 Test Problem II

Test problem II, from Ascher et al. [1], Example 1.17, describes a shock wave in a one-dimensional nozzle flow. The BVODE is governed by a second order steady state Navier-Stokes equation:

$$\epsilon A(x)uu'' - \left[\frac{1+\gamma}{2} - \epsilon A'(x)\right]uu' + \frac{u'}{u} + \frac{A'(x)}{A(x)}\left(1 - \frac{\gamma-1}{2}u^2\right) = 0, \tag{5.5}$$

where $0 < x < 1$ is the normalized downstream distance from the throat of the nozzle, u is a normalized velocity, $A(x) = 1 + x^2$, is the area of the nozzle at x, $\gamma = 1.4$ and ϵ is a parameter, $0 < \epsilon << 1$. The boundary conditions are,

$$u(0) = 0.9129, \qquad u(1) = 0.375. \tag{5.6}$$

5.3 Test Problem III

The third test problem is Problem 20 from Jeff Cash's BVP Test Set [49]. This problem is described by a single second order differential equation and two boundary conditions. It possesses an exact closed form solution. The BVODE is outlined below:

$$\epsilon y'' + (y')^2 = 1, \tag{5.7}$$

$$y(0) = 1 + \epsilon \ln\left(\cosh\left(\frac{-0.745}{\epsilon}\right)\right), \tag{5.8}$$

$$y(1) = 1 + \epsilon \ln\left(\cosh\left(\frac{0.255}{\epsilon}\right)\right), \tag{5.9}$$

and ϵ is a problem parameter. The true solution is

$$y(x) = 1 + \epsilon \ln\left(\cosh\left(\frac{x - 0.745}{\epsilon}\right)\right). \tag{5.10}$$

5.4 Test Problem IV

Test problem IV is Problem 21 also from Jeff Cash's BVP Test Set [49]. It consists of a single second order differential equation with two boundary conditions. The problem has an exact closed form solution. The BVODE is outlined below:

$$\epsilon y'' = y + y^2 - e^{\left(\frac{-2x}{\sqrt{\epsilon}}\right)}. \tag{5.11}$$

with boundary conditions

$$y(0) = 1, \qquad y(1) = e^{\left(\frac{-1}{\sqrt{\epsilon}}\right)}, \tag{5.12}$$

where ϵ is a problem parameter. The analytic solution is

$$y(x) = e^{\left(\frac{-x}{\sqrt{\epsilon}}\right)}. \tag{5.13}$$

5.5 Thesis Test Problem V

The fifth test problem is Example 4.17 from Ascher et al. [1]. The problem is used to illustrate the use of the MUSN solver described in [1]. The BVODE consists of five first order ODEs and matching boundary conditions. These are

$$y_1'(t) = \alpha \frac{y_1(t)}{y_2(t)} \left(y_3(t) - y_1(t)\right), \qquad y_2'(t) = -\alpha(y_3(t) - y_1(t)), \tag{5.14}$$

$$y_3'(t) = \frac{1}{y_4(t)}(B - C(y_3(t) - y_5(t)) - \alpha y_3(t)(y_3(t) - y_1(t)), \tag{5.15}$$

$$y_4'(t) = \alpha(y_3(t) - y_1(t)), \qquad y_5'(t) = -\frac{C}{D}(y_5(t) - y_3(t)), \tag{5.16}$$

with boundary conditions,

$$y_1(0) = y_2(0) = y_3(0) = 1, \qquad y_4(0) = 10, \qquad y_3(1) = y_5(1), \tag{5.17}$$

where $B = 0.9$, $C = 1000$, $D = 10$ and $\alpha = 1.0$.

Chapter 6

Software Modifications - BVP_Solver III

6.1 Introduction

The introduction of the sixth, fourth, and second order Hermite-Birkhoff schemes into the BVP_SOLVER II software package to replace the current CMIRK interpolants required some modification to a number of components of the existing software. This thesis chapter focuses on the software engineering effort associated with the modification of BVP_SOLVER II to extend an existing component or add a new one whilst attentively ensuring that the interfaces, memory management, and software documentation are in proper order and remain so after the changes have been implemented.

This chapter chronicles the major changes to the BVP_SOLVER II software package. Each narrative begins with a header naming the particular component being modified and a simplified description of its original function. This is then followed by a detailed description of the modifications implemented and their impact.

The sampling points **TAU** and θ referred to in this chapter are the same and are used interchangeably in this chapter. TAU is the name given to the evaluation point in the BVP_SOLVER II code while it is denoted by θ in other parts of this thesis. In a similar manner, the asymptotically correct sampling point **C_TAU** is also denoted by C_τ elsewhere in this thesis.

6.2 Description of the Software Modifications

6.2.1 SUBROUTINE DEFECT_ESTIMATE

This routine contains an argument list of ten entities which are defined as follows:

NEQN: The sum of the number of differential equations and unknown parameters,

NSUB: The number of subintervals in the current mesh,

MESH: The current mesh,

Y: The discrete solution associated with the current mesh,

DEFECT: The maximum defect for the current solution approximation on each subinterval,

DEFECT_NORM: Estimated norm of the maximum defect,

INFO: Communication flag that monitors the status of the computation,

K_DISCRETE: Storage for the discrete Runge-Kutta stages,

K_INTERP: Storage for the continuous Runge-Kutta stages,

FSUB: User supplied routine which defines the right hand side of the ODEs.

This routine utilizes the discrete approximate solution, Y, together with the weight polynomials and the discrete and continuous Runge-Kutta stages associated with a MIRK and CMIRK scheme in order to construct a continuous approximate solution on the problem domain. The next step is the computation of an estimate of the defect of the CMIRK interpolant on each subinterval for each solution component. For

the symmetric relative sampling points TAU and (1-TAU) within each subinterval, the routine calls subroutine INTERP_WEIGHTS to evaluate the weight polynomials at the two sampling points. Then, for the *ith* subinterval a call to the routine SUM_STAGES gives the value of the CMIRK interpolant $u_i(t)$ at the aforementioned sampling points. The routine P_FSUB is next called to provide function evaluations of these interpolant values. Next, the defect of the solution is computed at the two sample points and whichever value is larger is taken to be the estimate of the maximum defect for the subinterval. These steps are repeated over all the subintervals of the current mesh. The maximum of these defect estimates is then computed and determines the suitability of the current solution.

Software Modification

Phase 1 : Since the extra stages of the Hermite-Birkhoff scheme are constructed through *evaluations* of the CMIRK scheme, the first major extension to subroutine DEFECT_ESTIMATE is designed to implement the boot-strapping mechanism in the computation of these new stages. The steps are outlined as follows:

1. The process starts by making $\tilde{s}^* - s^*$ calls to the routine INTERP_WEIGHTS. During each call to the latter routine, information, which includes the Hermite-Birkhoff abscissas, is transferred from the former. Subroutine INTERP_WEIGHT then evaluates the CMIRK weight polynomials $b_1(\theta) \cdots \cdots b_s(\theta)$ for θ equal to each Hermite-Birkhoff abscissa value. (The quantity $\tilde{s}^* - s^*$ represents the number of additional boot-strap stages required to construct the Hermite-Birkhoff scheme).

2. Next, for the *ith* subinterval, the subroutine DEFECT_ESTIMATE makes $\tilde{s}^* - s^*$ calls to the subroutine SUM_STAGES. This routine utilizes the previously computed discrete and continuous Runge-Kutta stages together with the weight polynomials evaluated above to compute the value of the CMIRK interpolant at each of the Hermite-Birkhoff abscissa.

3. The final stage of the boot-strapping process consists of $\tilde{s}^* - s^*$ calls to the subroutine P_FSUB to obtain function evaluations corresponding to the CMIRK interpolant evaluations at each Hermite-Birkhoff abscissa. These are the extra stages required for the Hermite-Birkhoff interpolant.

   Steps 2. and 3. are performed for each subinterval of the current mesh.

Having constructed the extra boot-strapped derived stages $\{k_r\}_{r=s^*+1}^{\tilde{s}^*}$, the Hermite-Birkhoff interpolant can now be assembled. On the *ith* subinterval it has the form:

$$\tilde{u}_i(t_{i-1} + \theta h_i) \quad = \quad d_0(\theta) y_{i-1} + d_1(\theta) y_i + h_i \sum_{r=1}^{\tilde{s}^*} \tilde{b}_r(\theta) k_r \tag{6.1}$$

where $d_0(\theta)$, $d_1(\theta)$, and $\tilde{b}_r(\theta)$ are known polynomials obtained from the interpolation conditions. Recall that since the Hermite-Birkhoff interpolant makes use only of the new Hermite-Birkhoff stages and the first two stages k_1 and k_2, we have $\tilde{b}_r = 0$ for $r = 3, \cdots, s^*$.

Phase 2 : The next step in the process computes the defect of the continuous solution on each subinterval by evaluating the Hermite-Birkhoff interpolant and its derivative

at the predetermined asymptotically correct sample point. The defect of the continuous solution is also evaluated at a second spot (the validity checking sampling point). The next set of modifications to the DEFECT_ESTIMATE routine are as follows.

1. The first step in this phase is the evaluation of weight polynomials of the Hermite-Birkhoff scheme at the location of the expected maximum defect ($\theta = C_\tau$). within each subinterval. This is accomplished by a call to the subroutine INTERP_HB_TAU (a new routine described later in this section) which evaluates the Hermite-Birkhoff weight polynomials for a given value of θ.

2. Next, for the *ith* subinterval, using the boot-strapped stages derived in step (3) of the previous section, together with the discrete solution at the left and right endpoints of the subintervals and their corresponding stages, and the weight polynomials evaluated in step (1) above, the Hermite-Birkhoff scheme and its first derivative are both constructed in three steps. The first step involves the multiplication of the stages $k_1 = f(t_{i-1}, y_{i-1})$, $k_2 = f(t_i, y_i)$, and the discrete end point solutions y_{i-1}, y_i by the appropriate weight coefficients $\tilde{b_1}(\theta = C_\tau)$, $\tilde{b_2}(\theta = C_\tau)$, $d_0(\theta = C_\tau)$ and $d_1(\theta = C_\tau)$ respectively. The second step takes care of the multiplication of the extra $\tilde{s}^* - s^*$ Hermite-Birkhoff stages and their corresponding weight polynomials, whilst the third stage assembles the various components of the interpolant together. The derivative is also assembled in the same manner.

3. The next step is a call to the subroutine P_FSUB which performs a function

evaluation corresponding to the Hermite-Birkhoff interpolant value at C_τ. The defect of the continuous solution is then computed using:

$$\tilde{\delta}_i(t) = \tilde{u}_i'(t) - f(t, \tilde{u}_i(t)), \tag{6.2}$$

where $t = t_{i-1} + C_\tau h_i$ represents the location of the expected maximum defect sampling point within each subinterval.

4. Steps (1-3) are repeated for the validity check sampling point in order to implement the *validity check* auxiliary process. The defect evaluation for each subinterval, obtained in Step 3., is compared with the defect evaluation obtained in this step to confirm that the former is about twice as large as the latter, in magnitude.

5. The final major change to the subroutine DEFECT_ESTIMATE is the addition of an extra parameter FLAGGED_SUBS to the argument list. This variable stores information about subintervals which do not satisfy the *validity check* criterion.

6.2.2 SUBROUTINE INTERP_TABLEAU

This subroutine defines the extra coefficients for the Runge-Kutta stages associated with the CMIRK scheme. It also defines the sample points for the defect associated with the CMIRK interpolant and its order. The changes to this routine are relatively minimal and include introducing: (1) the extra abscissas for the boot-strap stages,

(2) the maximum defect sample point, and (3) the validity check sampling point. The associated variable names are as follows:

1. C_C_TILDE_STAR: The Hermite-Birkhoff abscissa.
2. C_TAU: The maximum defect sampling point.
3. C_TAU_VALIDITY: The validity check sampling point.

The changes and extensions to the routine are listed in terms of the order of the particular interpolant being considered.

Second Order Hermite-Birkhoff scheme:

1. C_C_TILDE_STAR: None
2. C_TAU = 0.5
3. C_TAU_VALIDITY = 0.14645

Fourth Order Hermite-Birkhoff scheme:

1. C_C_TILDE_STAR (1) = $\frac{86}{100}$
2. C_C_TILDE_STAR (2) = $\frac{93}{100}$
3. C_TAU = 0.23133
4. C_TAU_VALIDITY = 0.49822

Sixth Order Hermite-Birkhoff scheme:

1. C_C_TILDE_STAR (1) = $\frac{7}{100}$
2. C_C_TILDE_STAR (2) = $\frac{14}{100}$
3. C_C_TILDE_STAR (3) = $\frac{86}{100}$
4. C_C_TILDE_STAR (4) = $\frac{93}{100}$
5. C_TAU = 0.5
6. C_TAU_VALIDITY = 0.31078

6.2.3 SUBROUTINE INTERP_HB_WEIGHTS

This is a new routine added to perform evaluations of the Hermite-Birkhoff interpolant weight polynomials ($d_0(\theta)$, $d_1(\theta)$ and $\{\tilde{b_r}(\theta)\}_{r=1}^{\tilde{s}^*}$) and, optionally, their first derivatives at the relative point θ within a subinterval. The subroutine has six arguments which are defined as follows:

1. S_TILDE_STAR : The number of stages used to construct the Hermite-Birkhoff scheme.
2. DD : An array which stores evaluations of the weight polynomials $d_0(\theta)$ and $d_1(\theta)$.
3. DDp: An array which contains evaluations of first derivatives of the weight polynomials $d_0(\theta)$ and $d_1(\theta)$.
4. B: Array which contains evaluations of the weight polynomials $\{\tilde{b_r}(\theta)\}_{r=1}^{\tilde{s}^*}$.

5. Bp: Array in which the corresponding evaluations of first derivatives of $\{\tilde{b_r}(\theta)\}_{r=1}^{\tilde{s}^*}$ are stored.

6. θ: The specific sampling point at which the weight polynomials are to be evaluated.

6.2.4 SUBROUTINE SOL_EVAL

This auxiliary routine has seven arguments and evaluates the interpolant at any given point T within the problem interval [a,b]. The entities in the argument list are as follows:

1. NODE : Number of ODEs.

2. NEQN : Sum of NODE and the number of unknown parameters.

3. IWORK : Array which contain relevant information about the number of stages and the order of the method that was used to compute the solution.

4. WORK: Array containing the Runge-Kutta and Hermite-Birkhoff stages, the mesh, and the corresponding discrete solution.

5. T: The evaluation point.

6. Z: Storage for the value of the interpolant at T.

7. Z_PRIME: Storage for the value of the first derivative of the interpolant at T.

The first major modification to this routine is that the call to the subroutine INTERP_WEIGHTS which evaluates the weight polynomials in the CMIRK scheme

is replaced by a call to the newly constructed subroutine INTERP_HB_WEIGHTS described earlier. Subroutine INTERP_HB_WEIGHTS then evaluates the weight polynomials for the Hermite-Birkhoff interpolant for a given value of TAU (corresponding to the input T). The subsequent call to subroutine SUM_STAGES which completes the evaluation process for the CMIRK scheme is eliminated. Instead, using the information stored in the arrays IWORK and WORK, the Hermite-Birkhoff interpolant is evaluated in a similar manner to what is done in the DEFECT_ESTIMATE subroutine.

Chapter 7

Numerical Experiments

7.1 Introduction

The modifications to the BVP_SOLVER II software package described in Chapter 6 led to the development of a new version of the code. The purpose of this chapter is to investigate the impact of these modifications on the overall computational performance of the code. This investigative process consists of the conduction of a series of machine dependent and independent numerical experiments based on the suite of test problems described in chapter five. These numerical experiments are designed to provide a direct comparison between the performances of the standard CMIRK schemes and their Hermite-Birkhoff counterparts. The results of the various tests will be presented across a variety of quality measures which include:

- Percentage success of each scheme in estimating the maximum defect.
- Normalized plots of the defect curves.
- Kernel density plots of the defect curves. (These plots illustrate the distribution of the location of the maximum defect.)
- Measurement of the work aggregate, $\sum_j N_j \times NI_j$. (N_j is the number of subintervals used in the *jth* mesh employed by BVP_SOLVER II to solve a given

problem; NI_j is the number of Newtons iteration required to obtain a solution to the nonlinear system constructed based on the *jth* mesh. See Section 3.15.

- Computational time.

Prior to performing the numerical experiments described here, the first suite of numerical tests conducted on the newest version of BVP_SOLVER, in the early experimentation phase, were specifically designed in order for the code to replicate the numerical results, produced by an experimental version of the software, which were published in [23]. The success of this first set of tests, in reproducing comparable numerical results for the sixth order Hermite-Birkhoff interpolant, paved the way for the experiments conducted in this chapter.

All non-graphical numerical results will be presented in a tabular form followed by a descriptive analysis and discussion of important points.

7.2 Maximum Defect Estimates

7.2.1 Experimental Setup

The numerical experiments conducted on each test problem are performed across a range of tolerances from 10^{-4} to 10^{-10}, for specified values of the problem dependent parameter(s) occurring in a given problem. As discussed in section 3.1.5, the determination of the numerical solution of each problem involves computations over a sequence of meshes. We compare the estimated maximum defect based on the standard interpolant with the estimated maximum defect based on the new interpolant for each subinterval of each mesh employed in the computation process. This comparison is expressed in terms of the number of subintervals ($\mathbf{N_{SI}}$) and percentage of

subintervals (**%SI**) in each mesh in which the maximum defect estimate is accurate to within 90%, 95%, and 99% of the true maximum defect. An estimate of the true maximum defect over each subinterval is determined by sampling the defect at a thousand (1000) uniformly distributed points within each subinterval and selecting the largest value to be the true maximum defect. **Ratio** is the estimated maximum defect over the true maximum defect. We also report how well the maximum defect is estimated through the values **min ratio** and **max ratio**. These represent, in ratio form, how accurate the worst and best estimates of the maximum defect are in comparison to the their respective true maximums, over all subintervals.

We also measure the computational costs in a machine independent fashion, incurred by BVP_SOLVER II during the construction and factorization of the Newton matrices which arise from the discretization of the ODEs on a given mesh. This is the most significant cost incurred by the solver and represents a good machine independent measure of the overall computational cost for a given problem. This is represented through the sum of terms $\sum_j N_j \times NI_j$, where j ranges over the meshes employed in the solution of a given problem. $\mathbf{N}_j$ is the number of subintervals of the mesh that is associated with the jth mesh and $\mathbf{NI}_j$ is the number of Newton iterations needed to solve the nonlinear system associated with this mesh.

We start with the presentation of some of the results based on numerous tests conducted to investigate the effectiveness of the asymptotically correct interpolants in providing a more robust defect estimation procedure. These results provide a numerical comparison of defect estimation based on the standard fourth and sixth

order CMIRK interpolants versus the fourth and sixth order asymptotically correct Hermite-Birkhoff interpolants. The criterion for success in these numerical experiments is that estimates of the maximum defect produced by the Hermite-Birkhoff and CMIRK schemes should **underestimate** the true maximum defect by less than 1%. In other words the accuracy of the maximum defect estimates yielded by a scheme should be greater than 99%. The inclusion of the two additional ranges (95% and 90%) provides a slightly different yet insightful perspective (in the Hermite-Birkhoff context) on the analysis of the auxiliary validity check process.

Table 7.1: Results using fourth order schemes for test problem IV with $\epsilon = 10^{-2}$ and $TOL = 10^{-7}$.

Hermite-Birkhoff Scheme						CMIRK Scheme				
Mesh	N	Ratio	N_{SI}	$\%SI$		Mesh	N	Ratio	N_{SI}	$\%SI$
1	9	Ratio>0.99	3	33%		1	9	Ratio>0.99	0	0%
		Ratio >0.95	6	67%				Ratio>0.95	0	0%
		Ratio >0.90	9	100%				Ratio>0.90	0	0%
min ratio = 0.9430						min ratio = 0.4904				
max ratio = 0.9926						max ratio = 0.6646				
Mesh	N	Ratio	N_{SI}	$\%SI$		Mesh	N	Ratio	N_{SI}	$\%SI$
2	36	Ratio >0.99	36	100%		2	37	Ratio>0.99	0	0%
		Ratio >0.95	36	100%				Ratio>0.95	0	0%
		Ratio >0.90	36	100%				Ratio>0.90	0	0%
min ratio = 0.9979						min ratio = 0.4900				
max ratio = 0.9990						max ratio = 0.6805				
Mesh	N	Ratio	N_{SI}	$\%SI$		Mesh	N	Ratio	N_{SI}	$\%SI$
3	109	Ratio >0.99	109	**100%**		3	108	Ratio>0.99	**0**	**0%**
		Ratio >0.95	109	100%				Ratio>0.95	0	0%
		Ratio >0.90	109	100%				Ratio>0.90	0	0%
min ratio = 0.9996						min ratio = 0.4898				
max ratio = 0.9999						max ratio = 0.6794				
$\sum_j N_j \times NI_j$=154						$\sum_j N_j \times NI_j$=152				

Table 7.2: Results using sixth order schemes for test problem III with $\epsilon = 10^{-2}$ and $TOL = 10^{-7}$.

Hermite-Birkhoff Scheme						CMIRK Scheme				
Mesh	N	Ratio	N_{SI}	$\%SI$		Mesh	N	Ratio	N_{SI}	$\%SI$
1	36	Ratio>0.99	6	17%		1	36	Ratio>0.99	0	0%
		Ratio >0.95	11	31%				Ratio>0.95	0	0%
		Ratio >0.90	14	39%				Ratio>0.90	6	17%
min ratio = 0.5438						min ratio = 0.0184				
max ratio = 0.9997						max ratio = 0.9468				
Mesh	N	Ratio	N_{SI}	$\%SI$		Mesh	N	Ratio	N_{SI}	$\%SI$
2	72	Ratio >0.99	14	19%		2	72	Ratio>0.99	4	6%
		Ratio >0.95	15	21%				Ratio>0.95	4	6%
		Ratio >0.90	19	26%				Ratio>0.90	14	19%
min ratio = 0.6157						min ratio = 0.2855				
max ratio = 0.9994						max ratio = 1.0000				
Mesh	N	Ratio	N_{SI}	$\%SI$		Mesh	N	Ratio	N_{SI}	$\%SI$
3	79	Ratio >0.99	69	**87%**		3	79	Ratio>0.99	3	**4%**
		Ratio >0.95	73	92%				Ratio>0.95	9	11%
		Ratio >0.90	74	94%				Ratio>0.90	63	80%
min ratio = 0.7384						min ratio = 0.2238				
max ratio = 0.9999						max ratio = 0.9941				
$\sum_j N_j \times NI_j$=2095						$\sum_j N_j \times NI_j$= 2095				

Table 7.3: Results using fourth order schemes for test problem V with $\epsilon = 1.0$ and $TOL = 10^{-8}$.

Hermite-Birkhoff Scheme						CMIRK Scheme				
Mesh	N	Ratio	N_{SI}	$\%SI$		Mesh	N	Ratio	N_{SI}	$\%SI$
1	9	Ratio>0.99	8	89%		1	9	Ratio>0.99	0	0%
		Ratio >0.95	9	100%				Ratio>0.95	0	0%
		Ratio >0.90	9	100%				Ratio>0.90	0	0%
min ratio = 0.9873						min ratio = 0.4582				
max ratio = 1.0000						max ratio = 0.5623				
Mesh	N	Ratio	N_{SI}	$\%SI$		Mesh	N	Ratio	N_{SI}	$\%SI$
2	36	Ratio >0.99	36	100%		2	36	Ratio>0.99	0	0%
		Ratio >0.95	36	100%				Ratio>0.95	0	0%
		Ratio >0.90	36	100%				Ratio>0.90	0	0%
min ratio = 0.9986						min ratio = 0.4634				
max ratio = 1.0000						max ratio = 0.5693				
Mesh	N	Ratio	N_{SI}	$\%SI$		Mesh	N	Ratio	N_{SI}	$\%SI$
3	91	Ratio >0.99	91	**100%**		3	91	Ratio>0.99	0	**0%**
		Ratio >0.95	91	100%				Ratio>0.95	0	0%
		Ratio >0.90	91	100%				Ratio>0.90	0	0%
min ratio = 0.9997						min ratio = 0.4710				
max ratio = 1.0000						max ratio = 0.5795				
$\sum_j N_j \times NI_j$=163						$\sum_j N_j \times NI_j$= 163				

Table 7.4: Results using sixth order schemes for test problem V with $\epsilon = 1.0$ and $TOL = 10^{-8}$.

Hermite-Birkhoff Scheme						CMIRK Scheme				
Mesh	N	Ratio	N_{SI}	$\%SI$		Mesh	N	Ratio	N_{SI}	$\%SI$
1	9	Ratio>0.99	8	89%		1	9	Ratio>0.99	0	0%
		Ratio >0.95	9	100%				Ratio>0.95	0	0%
		Ratio >0.90	9	100%				Ratio>0.90	6	67%
min ratio = 0.9973						min ratio = 0.7735				
max ratio = 1.0000						max ratio = 0.9297				
Mesh	N	Ratio	N_{SI}	$\%SI$		Mesh	N	Ratio	N_{SI}	$\%SI$
2	20	Ratio >0.99	20	**100%**		2	20	Ratio>0.99	1	**5%**
		Ratio >0.95	20	100%				Ratio>0.95	3	15%
		Ratio >0.90	20	100%				Ratio>0.90	14	70%
min ratio = 0.9944						min ratio = 0.1624				
max ratio = 1.0000						max ratio = 0.9914				
$\sum_j N_j \times NI_j$=56						$\sum_j N_j \times NI_j$=56				

Table 7.5: Results using sixth order schemes for test problem I with $\epsilon = 10^{-2}$ and $TOL = 10^{-9}$.

Hermite-Birkhoff Scheme						CMIRK Scheme				
Mesh	N	Ratio	N_{SI}	$\%SI$		Mesh	N	Ratio	N_{SI}	$\%SI$
1	9	Ratio>0.99	5	56%		1	9	Ratio>0.99	1	11%
		Ratio >0.95	9	100%				Ratio>0.95	5	56%
		Ratio >0.90	9	100%				Ratio>0.90	7	78%
min ratio = 0.9578						min ratio = 0.2660				
max ratio = 1.0000						max ratio = 0.9997				
Mesh	N	Ratio	N_{SI}	$\%SI$		Mesh	N	Ratio	N_{SI}	$\%SI$
2	36	Ratio >0.99	31	86%		2	36	Ratio>0.99	1	3%
		Ratio >0.95	36	100%				Ratio>0.95	5	14%
		Ratio >0.90	36	100%				Ratio>0.90	27	75%
min ratio = 0.9811						min ratio = 0.3249				
max ratio = 1.0000						max ratio = 0.9935				
Mesh	N	Ratio	N_{SI}	$\%SI$		Mesh	N	Ratio	N_{SI}	$\%SI$
3	67	Ratio >0.99	65	**97%**		3	67	Ratio>0.99	0	**0%**
		Ratio >0.95	67	100%				Ratio>0.95	3	4%
		Ratio >0.90	67	100%				Ratio>0.90	54	81%
min ratio = 0.9837						min ratio = 0.4334				
max ratio = 1.0000						max ratio = 0.9666				
$\sum_j N_j \times NI_j$=112						$\sum_j N_j \times NI_j$= 112				

Table 7.6: Results using fourth order schemes for test problem I with $\epsilon = 10^{-2}$ and $TOL = 10^{-9}$.

Hermite-Birkhoff Scheme						CMIRK Scheme				
Mesh	N	Ratio	N_{SI}	$\%SI$		Mesh	N	Ratio	N_{SI}	$\%SI$
1	9	Ratio>0.99	2	22%		1	9	Ratio>0.99	0	0%
		Ratio >0.95	6	67%				Ratio>0.95	2	22%
		Ratio >0.90	7	78%				Ratio>0.90	2	22%
min ratio = 0.5675						min ratio = 0.3453				
max ratio = 0.9984						max ratio = 0.9526				
Mesh	N	Ratio	N_{SI}	$\%SI$		Mesh	N	Ratio	N_{SI}	$\%SI$
2	36	Ratio >0.99	27	75%		2	36	Ratio>0.99	0	0%
		Ratio >0.95	33	92%				Ratio>0.95	0	0%
		Ratio >0.90	35	97%				Ratio>0.90	0	0%
min ratio = 0.8573						min ratio = 0.2750				
max ratio = 1.0000						max ratio = 0.6835				
Mesh	N	Ratio	N_{SI}	$\%SI$		Mesh	N	Ratio	N_{SI}	$\%SI$
3	144	Ratio >0.99	138	96%		3	144	Ratio>0.99	0	0%
		Ratio >0.95	144	100%				Ratio>0.95	0	0%
		Ratio >0.90	144	100%				Ratio>0.90	0	0%
min ratio = 0.9583						min ratio = 0.2614				
max ratio = 1.0000						max ratio = 0.8864				
Mesh	N	Ratio	N_{SI}	$\%SI$		Mesh	N	Ratio	N_{SI}	$\%SI$
4	410	Ratio >0.99	410	100%		4	410	Ratio>0.99	0	0%
		Ratio >0.95	410	100%				Ratio>0.95	0	0%
		Ratio >0.90	410	100%				Ratio>0.90	1	0%
min ratio = 0.9932						min ratio = 0.2878				
max ratio = 1.0000						max ratio = 0.9061				
Mesh	N	Ratio	N_{SI}	$\%SI$		Mesh	N	Ratio	N_{SI}	$\%SI$
5	506	Ratio >0.99	506	**100%**		5	506	Ratio>0.99	0	**0%**
		Ratio >0.95	506	100%				Ratio>0.95	0	0%
		Ratio >0.90	506	100%				Ratio>0.90	1	0%
min ratio = 0.9919						min ratio = 0.2882				
max ratio = 1.0000						max ratio = 0.9031				
$\sum_j N_j \times NI_j$=1105						$\sum_j N_j \times NI_j$= 1105				

Table 7.7: Results using fourth order schemes for test problem II with $\epsilon = 0.5$ and $TOL = 10^{-9}$.

Hermite-Birkhoff Scheme						CMIRK Scheme				
Mesh	N	Ratio	N_{SI}	$\%SI$		Mesh	N	Ratio	N_{SI}	$\%SI$
1	36	Ratio>0.99	7	19%		1	36	Ratio>0.99	1	3%
		Ratio >0.95	18	50%				Ratio>0.95	3	8%
		Ratio >0.90	27	75%				Ratio>0.90	3	8%
min ratio = 0.5525						min ratio = 0.0790				
max ratio = 0.9934						max ratio = 0.9993				
Mesh	N	Ratio	N_{SI}	$\%SI$		Mesh	N	Ratio	N_{SI}	$\%SI$
2	72	Ratio >0.99	35	49%		2	72	Ratio>0.99	2	3%
		Ratio >0.95	61	85%				Ratio>0.95	6	8%
		Ratio >0.90	70	97%				Ratio>0.90	11	15%
min ratio = 0.6388						min ratio = 0.0474				
max ratio = 0.9998						max ratio = 0.9978				
Mesh	N	Ratio	N_{SI}	$\%SI$		Mesh	N	Ratio	N_{SI}	$\%SI$
3	288	Ratio >0.99	260	90%		3	288	Ratio>0.99	8	3%
		Ratio >0.95	287	100%				Ratio>0.95	20	7%
		Ratio >0.90	287	100%				Ratio>0.90	30	10%
min ratio = 0.7563						min ratio = 0.2133				
max ratio = 1.0000						max ratio = 0.9999				
Mesh	N	Ratio	N_{SI}	$\%SI$		Mesh	N	Ratio	N_{SI}	$\%SI$
4	1065	Ratio >0.99	1061	100%		4	1065	Ratio>0.99	27	3%
		Ratio >0.95	1064	100%				Ratio>0.95	66	6%
		Ratio >0.90	1065	100%				Ratio>0.90	102	10%
min ratio = 0.9079						min ratio = 0.1772				
max ratio = 1.0000						max ratio = 1.0000				
Mesh	N	Ratio	N_{SI}	$\%SI$		Mesh	N	Ratio	N_{SI}	$\%SI$
5	1384	Ratio >0.99	1375	**99%**		5	1384	Ratio>0.99	35	**3%**
		Ratio >0.95	1382	100%				Ratio>0.95	85	6%
		Ratio >0.90	1383	100%				Ratio>0.90	131	9%
min ratio = 0.8978						min ratio = 0.1771				
max ratio = 1.0000						max ratio = 1.0000				
$\sum_j N_j \times NI_j$=4969						$\sum_j N_j \times NI_j$= 4969				

Table 7.8: Results using sixth order schemes for test problem II with $\epsilon = 0.5$ and $TOL = 10^{-9}$.

Hermite-Birkhoff Scheme						CMIRK Scheme				
Mesh	N	Ratio	N_{SI}	$\%SI$		Mesh	N	Ratio	N_{SI}	$\%SI$
1	36	Ratio>0.99	21	58%		1	36	Ratio>0.99	4	11%
		Ratio >0.95	28	78%				Ratio>0.95	14	39%
		Ratio >0.90	29	81%				Ratio>0.90	26	72%
min ratio = 0.6209						min ratio = 0.0043				
max ratio = 1.0000						max ratio = 0.9992				
Mesh	N	Ratio	N_{SI}	$\%SI$		Mesh	N	Ratio	N_{SI}	$\%SI$
2	144	Ratio >0.99	136	94%		2	144	Ratio>0.99	6	4%
		Ratio >0.95	140	97%				Ratio>0.95	24	17%
		Ratio >0.90	140	97%				Ratio>0.90	100	69%
min ratio = 0.8332						min ratio = 0.0452				
max ratio = 1.0000						max ratio = 0.9999				
Mesh	N	Ratio	N_{SI}	$\%SI$		Mesh	N	Ratio	N_{SI}	$\%SI$
3	204	Ratio >0.99	194	95%		3	204	Ratio>0.99	8	4%
		Ratio >0.95	202	99%				Ratio>0.95	34	17%
		Ratio >0.90	202	99%				Ratio>0.90	147	72%
min ratio = 0.8734						min ratio = 0.3325				
max ratio = 1.0000						max ratio = 0.9999				
Mesh	N	Ratio	N_{SI}	$\%SI$		Mesh	N	Ratio	N_{SI}	$\%SI$
4	224	Ratio >0.99	215	**96%**		4	224	Ratio>0.99	11	**5%**
		Ratio >0.95	220	98%				Ratio>0.95	35	16%
		Ratio >0.90	221	99%				Ratio>0.90	164	73%
min ratio = 0.9580						min ratio = 0.2397				
max ratio = 1.0000						max ratio = 1.0000				
$\sum_j N_j \times NI_j$=1400						$\sum_j N_j \times NI_j$= 1400				

Based on the results presented in Tables 7.1 to 7.8, a number of general observations can be made. Firstly, employing the standard CMIRK interpolants rarely leads to a successful estimate of the maximum defect. The numerous experiments conducted on all five test questions reveal that over all subintervals of all meshes treated, the estimated maximum defect was within 1% of the true maximum defect in only an

average 5% of all subintervals. In contrast the maximum defect estimates yielded by the Hermite-Birkhoff schemes are generally very close to the true maximum defects. This fact is evident in most of tables in which the number of subintervals in the terminal mesh yielding estimates of the maximum defect within 99% of the true maximum defect, is approaching a 100% success rate. The lowest success rate on a converged mesh was recorded at 87% in Table 7.3. Enright and Muir [23] point out that this lower success rate occurs in relatively larger sized subintervals where the leading term in the defect expansion doesn't dominate the higher order terms. Similar behavior is observed in numerical experiments conducted at lower tolerances of about 10^{-4} and 10^{-5}. This can be explained as follows: sharper tolerances say, 10^{-9}, require the code to make repeated mesh adaptations until the maximum defect estimate on each subinterval is less than the tolerance. By this time most of the subintervals are already small enough for the leading term to dominate in the asymptotic expansion of the defect. However in the case of coarser tolerances, say around 10^{-4}, the size of a significant number of subintervals may not be small enough to justify one point sampling.

7.3 Plots of the Normalized Defect

7.3.1 Experimental Setup

There are two graphical representations in which plots of the defect curves are presented. The first type of graphic represents plots of the normalized defect on [0,1]. For a given subinterval, the defect is normalized by dividing through (or scaling) it by

the maximum defect, which ensures that the maximum curve value will be 1, in magnitude. The defect curves on each subinterval over all meshes are then superimposed on [0,1]. The kernel density plots of the defect are the second visual representation in which certain numerical results are presented. These are obtained by determining the location of the maximum defect on each subinterval over all meshes. This type of graphic illustrates the frequency distribution of the location of the maximum defect throughout the computation process.

In this section we present some numerical results for the two types of graphical representations described above. Each of the plots presented here will provide a visual perspective on a particular experiment considered in the previous section. Our choice of presentation layout is a side by side comparison of the normalized defect plots for both the Hermite-Birkhoff and CMIRK schemes immediately followed by a similar comparison of their kernel density plots. The term **scaled defect** in the plots headings is an alternative phrase for normalized defect. The plots for the normalized defects closely mirror the information presented in tabular form in the previous section. Rather than only plotting the normalized defect curves for each subinterval in the final converged mesh, we plot the defect curves for every subinterval from the first mesh to the converged mesh. (The total number of subintervals considered is n). The defect plots for the Hermite-Birkhoff interpolant will be much cleaner if only the last converged mesh results are considered.

Figure 7.1: Plot of the results for test problem IV using fourth order schemes with $\epsilon = 10^{-2}$ and $TOL = 10^{-7}$.

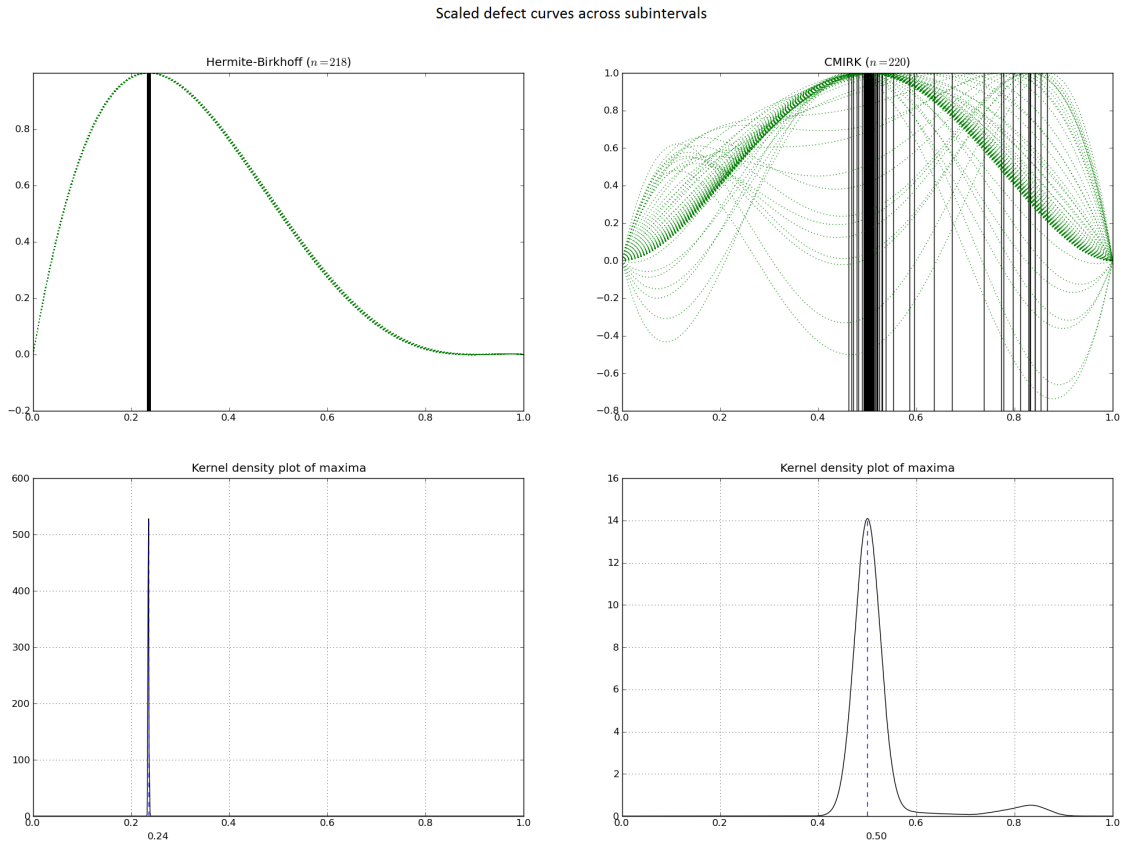


Figure 7.2: Plot of the results for test problem III using sixth order schemes with $\epsilon = 10^{-2}$ and $TOL = 10^{-7}$.

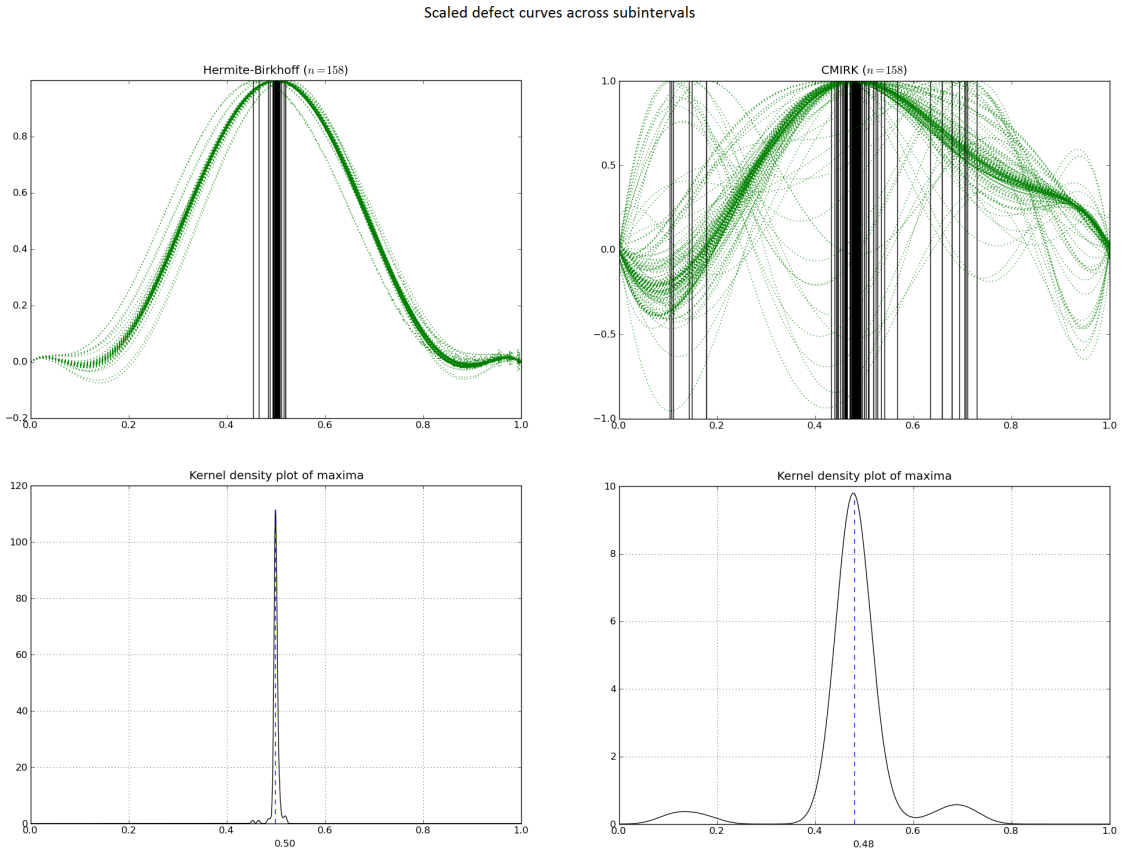


Figure 7.3: Plot of the results for test problem V using fourth order schemes with $\epsilon = 1.0$ and $TOL = 10^{-8}$.

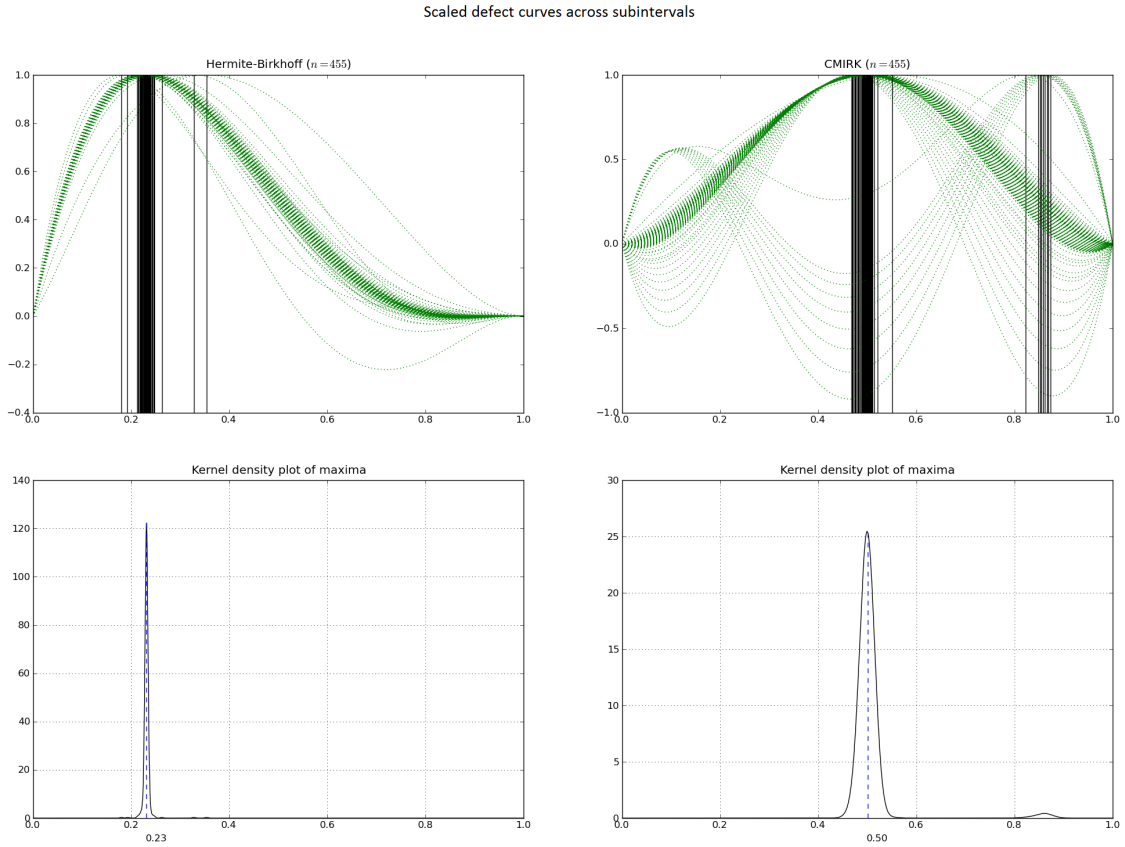


Figure 7.4: Plot of the results for test problem V using sixth order schemes with $\epsilon = 1.0$ and $TOL = 10^{-8}$.

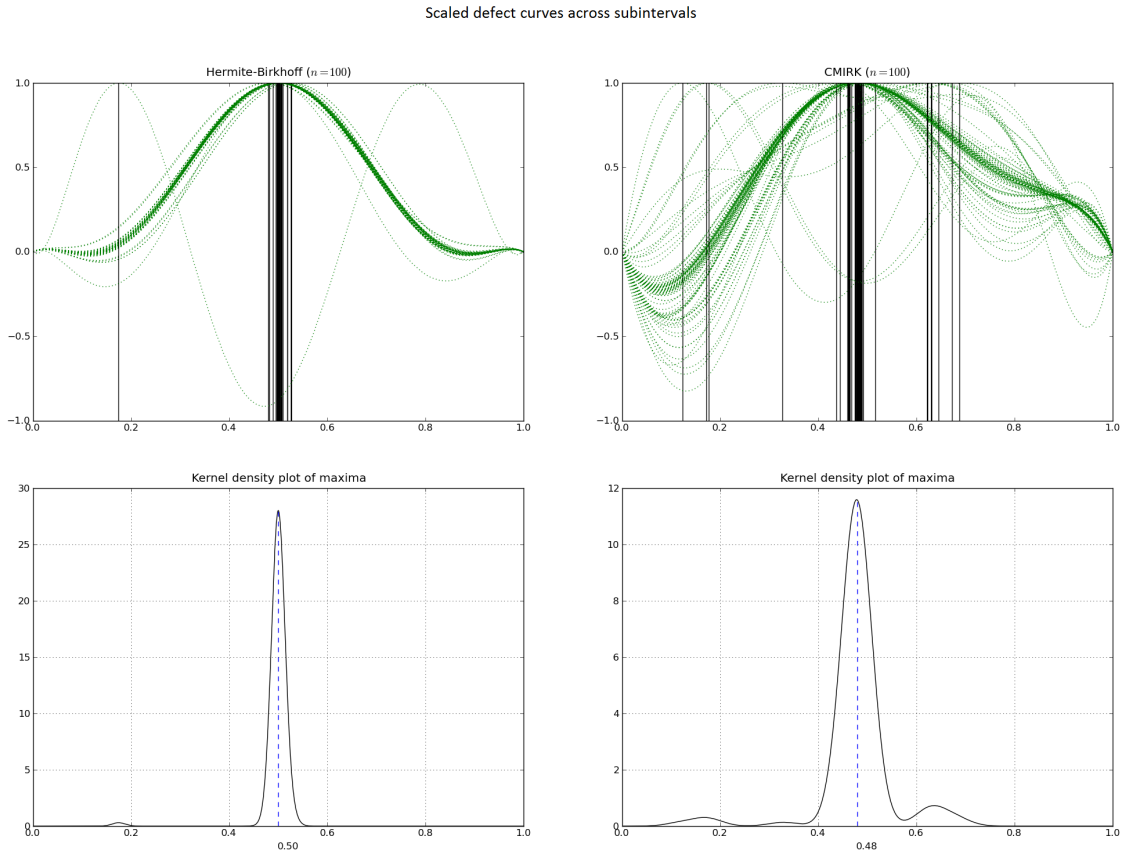


Figure 7.5: Plot of the results for test problem I using sixth order schemes with $\epsilon = 10^{-2}$ and $TOL = 10^{-9}$.

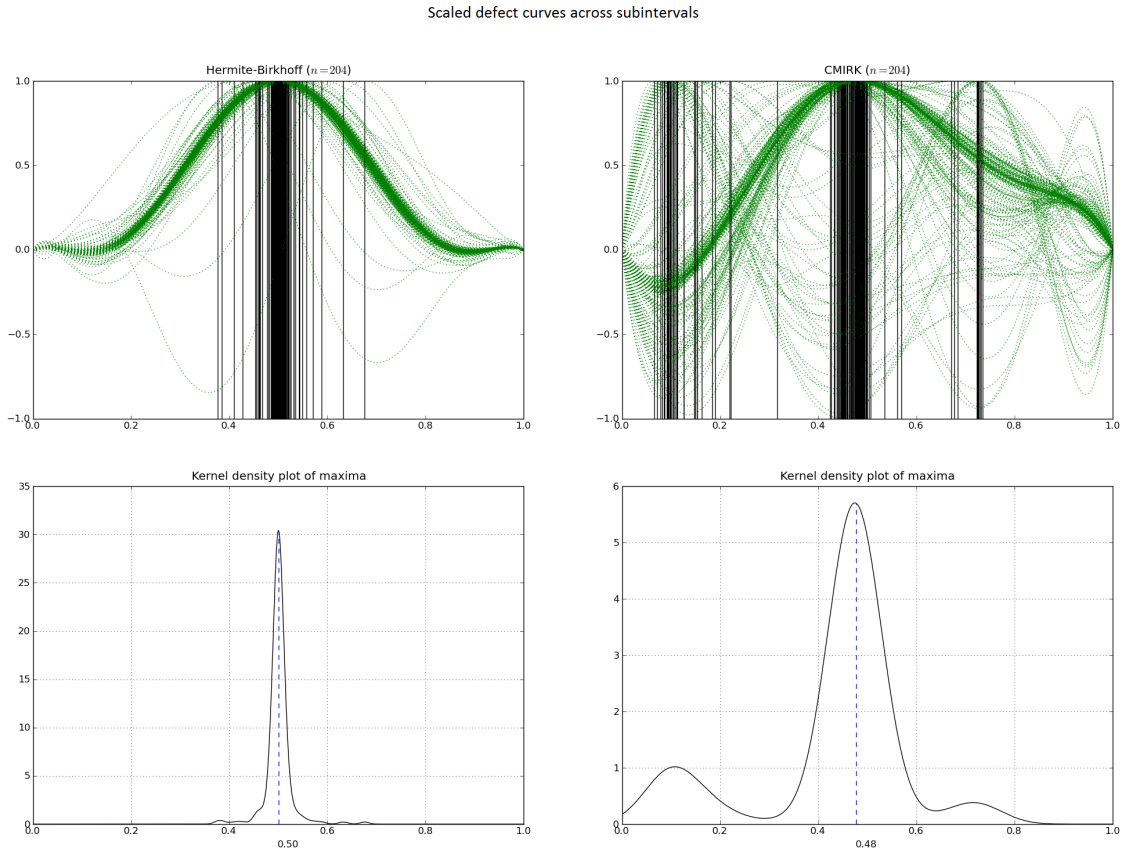


Figure 7.6: Plot of the results for test problem I using fourth order schemes with $\epsilon = 10^{-2}$ and $TOL = 10^{-9}$.

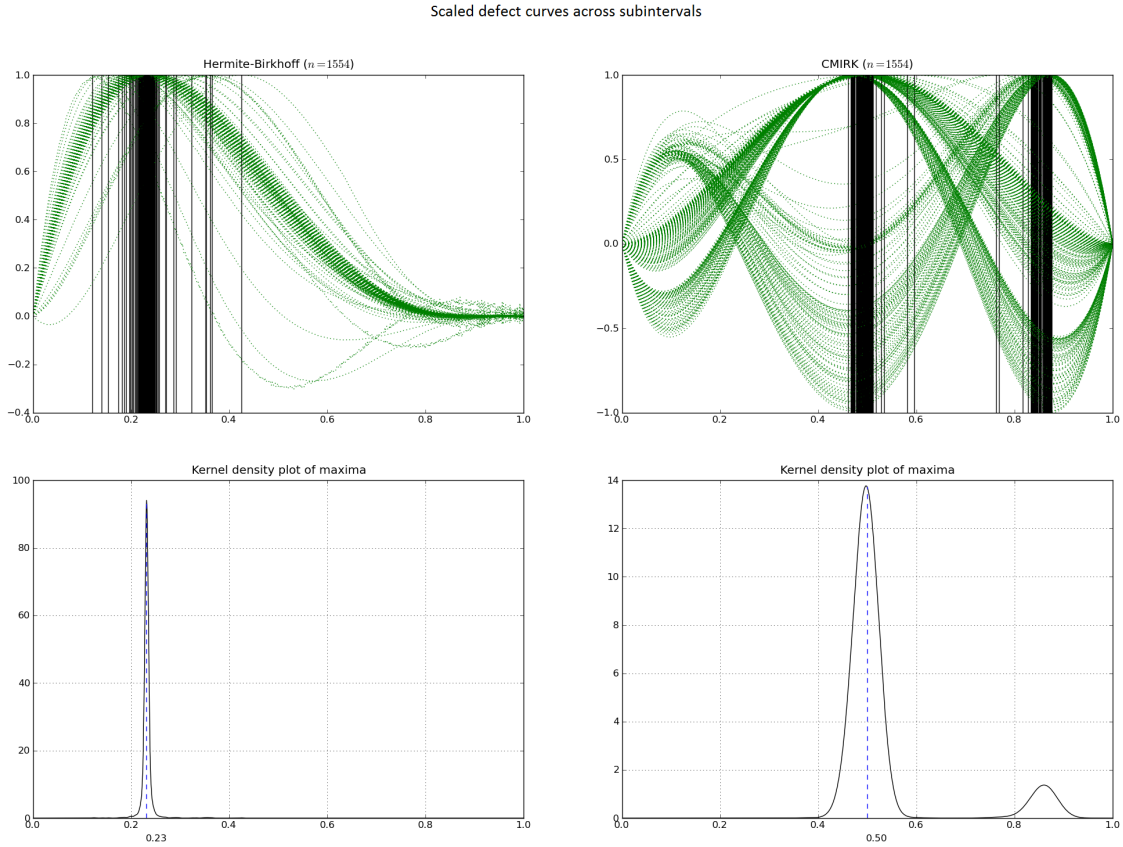


Figure 7.7: Plot of the results for test problem II using fourth order schemes with $\epsilon = 0.1$ and $TOL = 10^{-9}$.

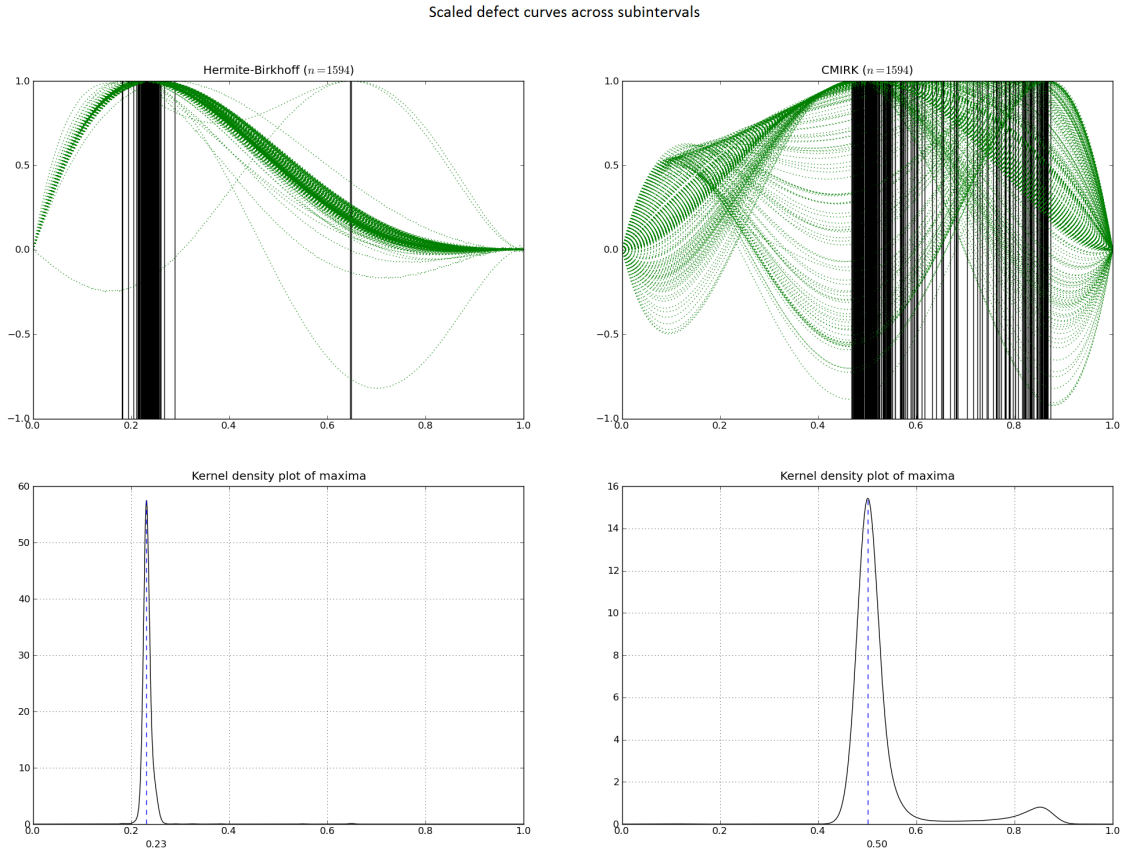
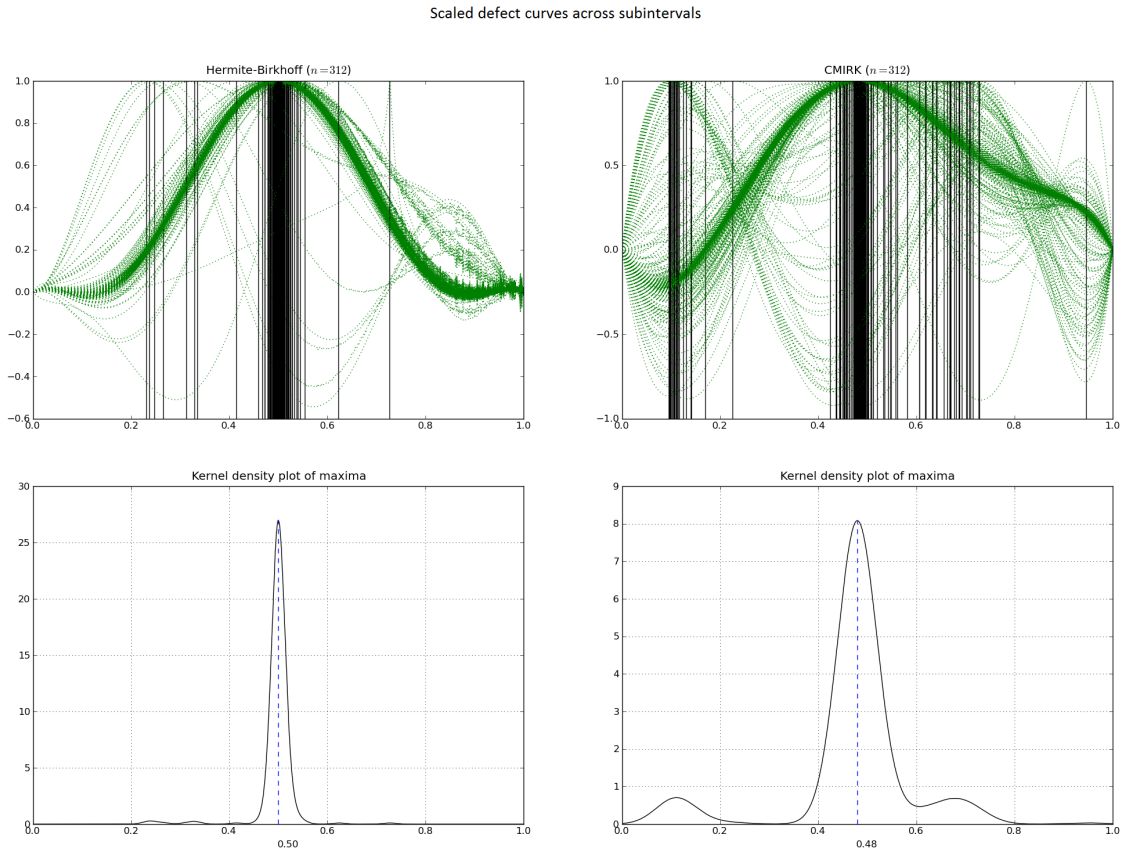


Figure 7.8: Plot of the results for test problem II using fourth order schemes with $\epsilon = 0.1$ and $TOL = 10^{-9}$.



7.3.2 Comments and Discussion

Figures 7.1-7.8 provide further visual corroboration of the tabulated results discussed earlier. In Figure 7.1 the fact that all the subintervals in the terminal mesh yield estimates of the defect which successfully meet our accuracy criterion is transmitted via a smooth normalized defect plot for the Hermite-Birkhoff scheme. The location of the true maximum ($\theta \approx 0.24$) in [0,1] is also accurately pinpointed by both the neat overlay of the vertical lines in the normalized defect plot as well as the spike in the

kernel density plot which represents its frequency distribution. The defect plot for the CMIRK interpolant on the other hand isn't as smooth and the parallel vertical lines depict multiple locations of the maximum defect. Whilst the kernel plot indicates an optimal frequency distribution of the maximum defect's location at ($\theta \approx 0.50$) on [0,1], the density mapping is substantially less than in the Hermite-Birkhoff case - a fact illustrated by a more rounded kernel plot. Although these observations have been made for Figure 7.1, they can be extended to the remaining plots in this section.

It was observed during the experimental phase, that there were some instances in which both Hermite-Birkhoff and CMIRK interpolants yielded poor estimates of the maximum defect. These situations occurred at lower tolerances of about 10^{-4}, when it is likely that many subintervals aren't small enough to justify the one point sampling criterion. Graphical representation of this situation is shown in Figures 7.9 - 7.10. The kernel density plot for the fourth order Hermite Birkhoff in the first graphic, pinpoints the location of the maximum defect within each subinterval at $\theta = 0.25$. In actuality the location of the maximum defect for fourth order asymptotically correct Hermite-Birkhoff schemes is at $\theta = 0.23$. The second set of plots provide an even better illustration of the situation. Here the actual location of the maximum defect for a majority of the curves is at $\theta = 1.00$ while the theoretical asymptotically correct location of the maximum defect for sixth order Hermite-Birkhoff schemes is at $\theta = 0.50$. This means that in both cases the leading term in the defect expansion isn't the dominant term.

The two sets of graphics also illustrate the importance of the kernel density plots

in indicating the location of the maximum defect for a particular experiment. Unlike in the previous graphics, where a neat overlay of vertical lines in the normalized plots indicated the location of the maximum defect, the situation here is quite different. Multiple parallel lines in the plots make it very difficult to identify exactly where the maximum defect is located.

We emphasize that this situation arises when the subinterval sizes are large and thus we are not in the asymptotic regime where the use of one-point sampling is justified. In such cases, the validity check flags the issue. See section 7.5.

Figure 7.9: Plot of the results for test problem III using fourth order schemes with $\epsilon = 10^{-2}$ and $TOL = 10^{-4}$.

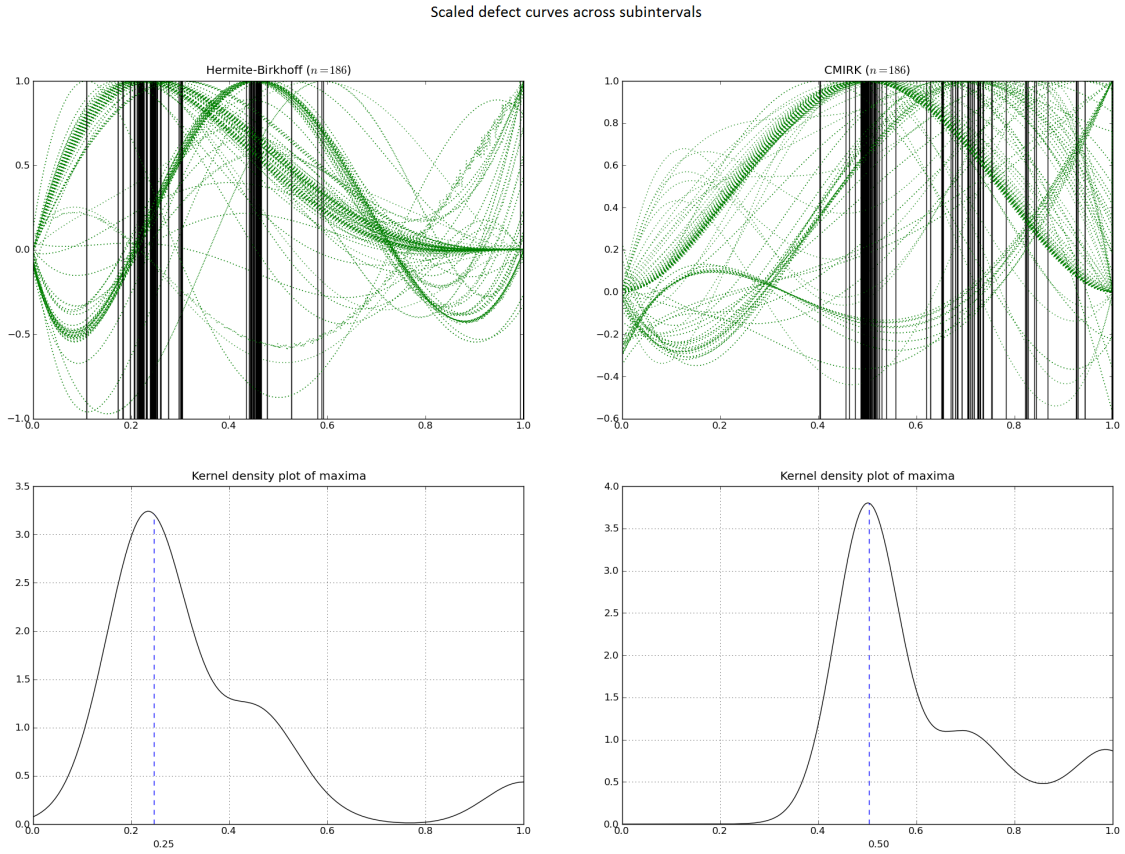
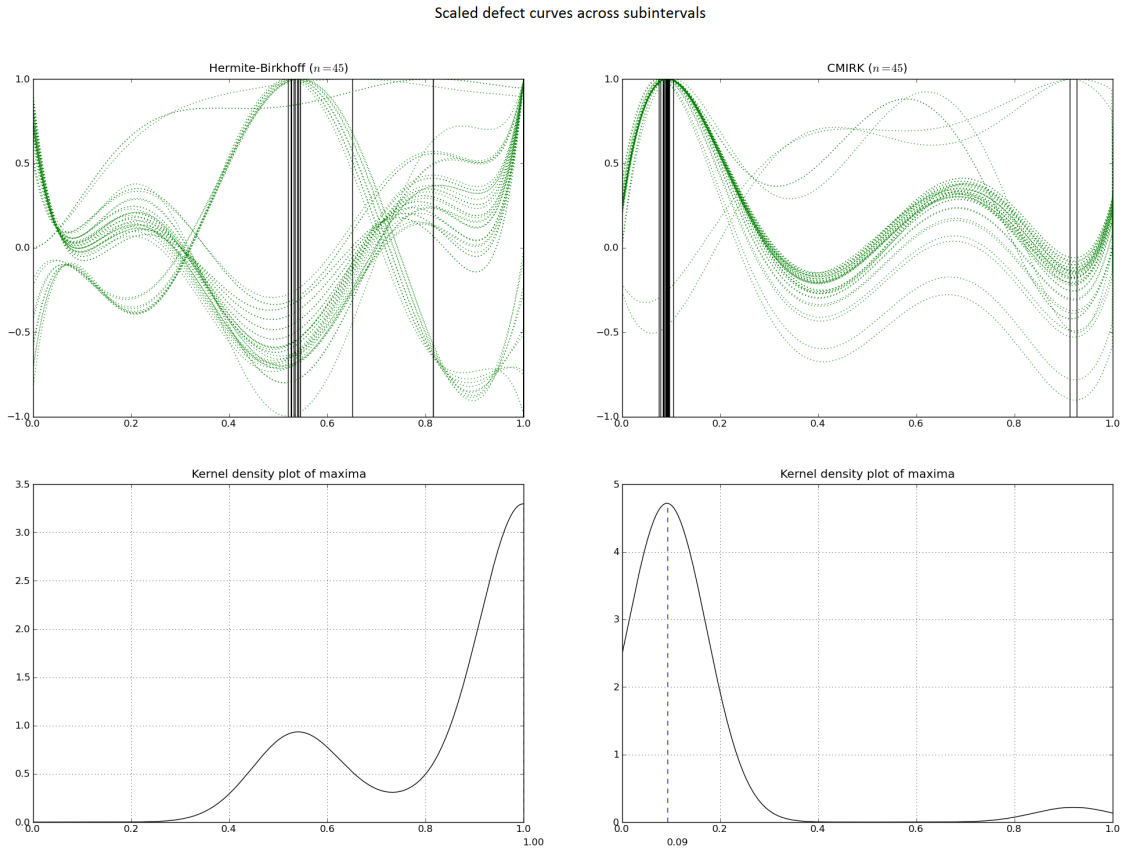


Figure 7.10: Plot of the results for test problem V using sixth order schemes with $\epsilon = 10^{-1}$ and $TOL = 10^{-4}$.



7.4 Machine Dependent Numerical Tests

The next set of numerical experiments conducted measure the time it takes the BVP_SOLVER code to compute a numerical solution to a particular problem. We specifically report the elapsed CPU time after the successful solution of a problem.

In this section we conduct a series of benchmarking comparisons between the version of the BVP_SOLVER software package which employs the standard CMIRK and

the new version which employs the Hermite-Birkhoff schemes. These tests specifically measure the time **(real time in micro seconds)** required by both codes to successfully compute a numerical solution to each of the five test problems. The FORTRAN90 intrinsic SYSTEM_CLOCK function which measures real time is the principal tool used to conduct the experiments.

7.4.1 Computational Time

The execution time was measured for both versions of the BVP_SOLVER code and the results recorded for each numerical solution computed by the pair.

Table 7.9: Execution time results for the two versions of the BVP_SOLVER code

Test Problem	Time μs		
	TOL	Hermite-Birkhoff Scheme	CMIRK Scheme
IV	$TOL = 10^{-7}$	1.0×10^{-3}	1.0×10^{-3}
III	$TOL = 10^{-9}$	1.38×10^{-1}	1.03×10^{-1}
V	$TOL = 10^{-9}$	1.7×10^{-2}	1.7×10^{-2}
I	$TOL = 10^{-8}$	1.0×10^{-3}	1.0×10^{-3}
II	$TOL = 10^{-7}$	1.0×10^{-3}	1.0×10^{-3}

Table 7.10: Both codes required identical execution timing in almost all the test problems with the exception of test problem three where the CMIRK code recorded a slightly faster execution time

7.5 Validity Checking

The numerical results for the auxiliary validity check process were recorded for the final converged mesh in the eight numerical experiments described in earlier sections. The term **Suspect subintervals** in the table below means subintervals in which the validity check failed. It was observed during the tests that the subintervals which

failed the validity check process corresponded to subintervals with poor defect estimates. Table 7.11 gives a summary of the results.

Table 7.11: Summary results for the auxiliary validity check process.

			Subintervals	
Test Problem	Method	TOL	Total Number	Suspect
IV ($\epsilon = 0.02$)	4	$TOL = 10^{-7}$	109	0
III ($\epsilon = 0.02$)	6	$TOL = 10^{-7}$	79	10
V ($\epsilon = 1.0$)	4	$TOL = 10^{-8}$	91	0
V ($\epsilon = 1.0$)	6	$TOL = 10^{-8}$	20	0
I ($\epsilon = 0.02$)	6	$TOL = 10^{-9}$	67	2
I ($\epsilon = 0.02$)	4	$TOL = 10^{-9}$	506	0
II ($\epsilon = 0.1$)	4	$TOL = 10^{-9}$	1375	9
II ($\epsilon = 0.1$)	6	$TOL = 10^{-9}$	224	9

Table 7.12: The highest percentage of suspect subintervals was 13% recorded for test problem III. In the other cases the percentages ranged between zero and four percent.

7.6 Overall Observations and Conclusions

The results presented in this chapter clearly demonstrate the superiority of the asymptotically correct Hermite-Birkhoff schemes in yielding high quality estimates of the maximum defect. We use these results to make a number of general observations.

- For the Hermite-Birkhoff interpolants, the lowest percentage of subintervals within 1% of the true maximum defect observed for tolerances of 10^{-7} and sharper was 87%. In the overwhelming majority of cases this percentage approaches a hundred percent for the converged mesh.

- The defect estimates produced by the Hermite-Birkhoff schemes are closer to the true maximum. This demands more from the code to compute an acceptable

numerical solution since smaller subinterval sizes (h_i) are normally required. This supports the findings made in the first observation.

- Despite the previous observation,the number of subintervals per mesh and the number of Newton iterations required by both versions of BVP_SOLVER are about the same. Hence the machine independent measure of computation cost, $\sum_j N_j \times NI_j$, and measurements of actual computer time, produced almost identical results.

The conclusion we can make having conducted numerous tests across a variety of platforms is that the asymptotically correct Hermite-Birkhoff schemes are vastly superior to their CMIRK counterparts and together with the validity check routine, provide the BVP_SOLVER III package with a more robust defect estimation process.

Chapter 8

Analysis of Directly Derived Asymptotically Correct Defect Control Schemes

8.1 Introduction

This chapter provides a detailed description of the derivation of a new fourth order CMIRK interpolant capable of yielding asymptotically correct estimates of the maximum defect. The CMIRK scheme is constructed through the approach of requiring the coefficients and weight polynomials of a standard fourth order CMIRK scheme to satisfy an additional order condition.

The more general approach for developing interpolants with the special *asymptotically correct defect* quality is via the boot-strapping approach implemented by Enright and Muir [23] in the derivation of a sixth order Hermite-Birkhoff scheme and considered in chapter four of this thesis. The boot-strap algorithm is intrinsically linked to interpolation theory. By using the relationship between the number of data points and degree of the corresponding unique interpolating polynomial, the contribution of some of the higher order terms to the error can be eliminated (in an asymptotic context). The end result is an interpolant leading to a defect expansion dominated by a single error term whose maximum value can be determined *a priori*,

at least asymptotically. However, the boot-strapping algorithm isn't the most optimal approach. This is because the extra sampling points within each subinterval generate extra stages which increase the cost of the overall computation, on a per subinterval basis.

The search for a more efficient approach (in terms of the number of stages required to obtain an interpolant leading to an asymptotically correct maximum defect estimate) has led to the investigation of interpolants developed via the direct approach alluded to at the beginning of this chapter. The next section of this chapter describes this process.

8.2 Directly Derived Fourth Order CMIRK Schemes

The groundwork for the development of the special fourth order CMIRK scheme is provided in Muir [40] who employs three optimization criteria, namely: (1) Minimization of the number of stages, (2) Maximization of the stage order of individual stages and (3) Minimization of the local error coefficient, in the derivation of optimal CMIRK schemes. This section first provides a brief background to the main concepts of that paper. The first criterion relates directly to the computational cost associated with the use of a CMIRK scheme and is dependent on the number of stages, which should be as small as possible. Maximizing the stage order is dependent on the availability of sufficient free parameters and leads to a simpler derivation and simpler expressions for the weight polynomials. Criterion three relates the accuracy of the scheme to the principal error coefficient of $\mathrm{O}(h^{p+1})$ in the local truncation error (assuming a method of order p). This coefficient depends on the parameters of the

Runge-Kutta scheme and is expressed in terms of the appropriately weighted unsatisfied conditions for order $p+1$. (In the fourth order case, this criterion is applied to the principal error coefficient of the $\mathrm{O}(h^5)$ term which has coefficients expressed in terms of the unsatisfied fifth order conditions). Assuming that the two-norm of the parameters in the principal error coefficient for order five is C_{p+1} and the two-norm of the corresponding parameters in the principal error coefficient for order six is C_{p+2}, then the requirement is that C_{p+1} is minimized subject to the condition that the ratio of C_{p+1} to C_{p+2} isn't too small. The idea is that a scheme with a smaller C_{p+1} value has a lower local error and may be more accurate than another scheme of the same order with a larger C_{p+1} value, but the C_{p+1} value should still be sufficiently large with respect to the C_{p+2} value so that the $p+1$ order term dominates.

A *pth* order MIRK scheme (see, e.g. [40]) has stage order q, $(q \leq p)$ if its coefficients satisfy the stage order conditions,

$$Xc^{j-1} + \frac{v}{j} = \frac{c^j}{j}, \qquad j = 1, \cdots, q, \tag{8.1}$$

where $c^j = (c_1^j, \cdots, c_s^j)^T$, where s is the number of stages of the method. We note that (8.1) is actually a set of s equations for each value of j and that there is one equation for each stage, for a given j. The maximum q for which (8.1) holds is the stage order of the method. However, it is possible for individual stages of a method to satisfy additional stage order conditions. The usual notation for recording the stage order conditions satisfied by each of the stages of a CMIRK scheme employs a stage order vector, SOV $= (q_1, q_2 \cdots, q_s)$, and this notation is adopted in this thesis chapter as

well. (Note that $\min_j\, q_j = q$).

To obtain the directly derived fourth order CMIRK scheme that leads to an asymptotically correct defect estimate, we start with the unique three stage, fourth order, stage order three, MIRK scheme with $c_1 = v_1 = 0$, $c_2 = v_2 = 1$, $x_{2,1} = 0$, $c_3 = v_3 = \frac{1}{2}$ and $x_{3,1} = -x_{3,2} = \frac{1}{8}$. This method is representable as a Butcher tableau of structure,

$$\begin{array}{c|c|ccc} 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{8} & -\frac{1}{8} & 0 \\ \hline & & \frac{1}{6} & \frac{1}{6} & \frac{2}{3} \end{array} .$$

The next step in the derivation process is to embed the MIRK scheme above into the family of five stage, fourth order, stage order three CMIRK schemes with stage order vector, SOV = (4, 4, 3, 3, 4). The resulting Butcher tableau is

$$\begin{array}{c|c|ccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{8} & -\frac{1}{8} & 0 & 0 & 0 \\ c_4 & v_4 & x_{41} & x_{42} & x_{43} & 0 & 0 \\ c_5 & v_5 & x_{51} & x_{52} & x_{53} & x_{54} & 0 \\ \hline & & b_1(\theta) & b_2(\theta) & b_3(\theta) & b_4(\theta) & b_5(\theta) \end{array} ,$$

where the weight polynomials $\{b_j(\theta)\}_{j=1}^{5}$ are required to satisfy the usual continuity

and order conditions, Muir [40], and the fourth and fifth order stages are required to satisfy the stage order three and stage order four respectively.

After imposing the appropriate stage order conditions on stages four and five (this gives x_{41}, x_{42}, and x_{43} in terms of c_4 and v_4, and x_{51}, x_{52}, and x_{53}, and x_{54} in terms of c_5 and v_5), we then require that the weight polynomials and remaining free coefficients, c_4, v_4, c_5, v_5, satisfy the standard fourth order continuous conditions: $b(\theta)^T e = \theta$, $b(\theta)^T c = \frac{1}{2}\theta^2$, $b(\theta)^T c^2 = \frac{1}{3}\theta^3$ and $b(\theta)^T c^3 = \frac{1}{4}\theta^4$. This is sufficient to guarantee that the CMIRK scheme will be of fourth order.

There are nine unsatisfied fifth order conditions associated with the principal error coefficient term of $\mathrm{O}(h^5)$ in the continuous local error expansion and these are expressed as nine unique polynomials. The imposition of the stage order three conditions effectively reduces the number of fifth order conditions from the nine unique polynomials to multiples of just two unique polynomials. The remaining free parameters are chosen to satisfy one or the other of the two fifth order conditions: $b(\theta)^T c^4 = \frac{1}{5}\theta^5$ and $b(\theta)^T(Xc^3 + \frac{v}{4}) = \frac{1}{20}\theta^5$. (To satisfy both would create a fifth order CMIRK scheme). Specific choices of the free parameters collapse the five multiples of one of the two polynomials to zero and four nonzero multiples of the one remaining polynomial. The Butcher tableau below, for the choices of $c_4 = v_4 = 1/4$ and $c_5 = v_5 = 7/8$, gives an example of such a method:

$$
\begin{array}{c|c|ccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{2} & \frac{1}{2} & \frac{1}{8} & -\frac{1}{8} & 0 & 0 & 0 \\
\frac{1}{4} & \frac{1}{4} & \frac{1}{8} & -\frac{1}{16} & -\frac{1}{16} & 0 & 0 \\
\frac{7}{8} & \frac{7}{8} & \frac{7}{2048} & -\frac{455}{6144} & \frac{7}{1024} & \frac{49}{768} & 0 \\
\hline
 & & b_1(\theta) & b_2(\theta) & b_3(\theta) & b_4(\theta) & b_5(\theta)
\end{array},
$$

where

$$\hat{b}_1(\theta) = \frac{1}{210}\theta(210 - 855\theta + 1540\theta^2 - 1260\theta^3 + 384\theta^4),$$

$$\hat{b}_2(\theta) = \frac{1}{90}\theta^2(-105 + 500\theta - 780\theta^2 + 384\theta^3),$$

$$\hat{b}_3(\theta) = \frac{2}{45}\theta^2(-105 + 430\theta - 510\theta^2 + 384\theta^3),$$

$$\hat{b}_4(\theta) = -\frac{16}{225}\theta^2(-105 + 290\theta - 285\theta^2 + 96\theta^3),$$

$$\hat{b}_5(\theta) = -\frac{256}{1575}\theta^2(-15 + 70\theta - 105\theta^2 + 48\theta^3).$$

The nine simplified polynomials appearing in the principal error coefficient for the fifth order are :

$$q_1(\theta) = q_3(\theta) = q_4(\theta) = q_7(\theta) = q_9(\theta) = 0, \tag{8.2}$$

$$q_2(\theta) = q_6(\theta) = q_8(\theta) = -\frac{1}{86400}\theta^2(-105 + 1690\theta - 2535\theta^2 + 1056\theta^3),$$

$$q_5(\theta) = -\frac{1}{28800}\theta^2(-105 + 1690\theta - 2535\theta^2 + 1056\theta^3). \tag{8.3}$$

It is obvious from the above expressions for $q_2(\theta), q_6(\theta), q_8(\theta)$, and $q_5(\theta)$, that the leading principal error coefficient in the local error expansion has contributions only from four multiples of a single polynomial. We could have chosen coefficients to satisfy the second order condition leading to a different asymptotically correct scheme but we haven't pursued that further here. Preliminary investigations along this line indicate a similar collapse to an identical polynomial to that shown above. In the defect control context, the polynomial of interest is $q_2'(\theta)$, which is associated with the single $O(h^4)$ term in the defect expansion. Therefore as $h_i \to 0$ the location of the maximum defect on each subinterval coincides with the maximum of the $q_2'(\theta)$ polynomial, which in this case occurs when $\theta \approx 0.47645$.

The basic form of the directly derived asymptotically correct CMIRK scheme on the ith subinterval, is a polynomial in θ of the form,

$$\hat{u}_i(t) = \hat{u}_i(t_i + \theta h_i) = y_i + h_i \sum_{r=1}^{5} \hat{b}_r(\theta) k_r, \qquad 0 \le \theta \le 1, \tag{8.4}$$

with stages k_r of the form given in (3.7). The parameters $\hat{b}_r(\theta)$, $r = 1, ..., \hat{s}$, are weight polynomials of degree five. This new CMIRK scheme can be used as the basis to implement defect control within BVP_SOLVER III.

However this method (like all CMIRK schemes) lacks an explicit dependence on y_{i+1}, and so has discontinuities in the defect, of the order of Newton tolerance at the

right hand end point of each subinterval. A standard approach which overcomes this limitation, is to convert the CMIRK scheme into its Hermite-Birkhoff form. However this involves introducing a polynomial (say $d_1(\theta)$) multiple of y_{i+1} into the expression for the interpolant and this in turn involves introducing an error term of the form $d_1(\theta)O(h_i^5)$ since the y_{i+1} value involves an error that is $O(h_i^5)$. Thus the error for the interpolant become $(d_1(\theta)C_1 + q_2(\theta)C_2)h^5$, where C_1 and C_2 are constants that depend on the error associated with y_{i+1} and the CMIRK scheme, respectively. The error term for this interpolant is therefore a linear combination of two polynomials and thus does not lead to a scheme which yields an asymptotically correct estimate of the defect.

Further investigation of this approach is required and is left for future work.

Chapter 9

Conclusion And Future Work

9.1 Conclusions

The thesis makes a number of contributions:

- Second and fourth order Hermite-Birkhoff interpolants leading to asymptotically correct maximum defect estimation schemes have been derived using the boot-strapping algorithm. The standard second order CMIRK scheme leads to an asymptotically correct estimate of the maximum defect, but this thesis describes how to obtain a smoother Hermite-Birkhoff interpolant that also leads to an asymptotically correct estimate of the maximum defect.

- Software modifications to the BVP_SOLVER II package were implemented to incorporate these schemes together with the sixth order case derived in Enright and Muir[23]. Numerical experiments conducted on both the standard schemes and the new interpolants demonstrate the latter's superiority and by extension the subsequent algorithmic enhancement of the software package.

- The software package was also modified to incorporate an auxiliary process known as validity checking. This optional routine provides an additional layer of confidence in the computed solution as well as in the defect control process

implemented by the solver.

- A fourth order CMIRK scheme leading to an asymptotically correct defect has also been developed using an alternative approach.

9.2 Future Work

The main direction of future work following on from this thesis is the development of alternative strategies for defect estimation in subintervals which are flagged during the validity check process. In such cases the sampling point occurs outside the formula's asymptotic regime meaning that the leading term in the expansion does not dominate the higher order terms, a necessary criterion for the one-point sampling process to be valid.

There are a number of possibilities presently under investigation in the development of such auxiliary computations that will improve the quality of the defect estimate. A simple approach under consideration is:

- Sampling the defect at several additional points on each subinterval and choosing the maximum of these as the estimate of the maximum defect.

More sophisticated approaches involve a closer examination of the leading terms in the defect expansion from a number of different perspectives which include:

- Identifying the dominant term in the defect expansion and locate the maximum point of its polynomial. The maximum defect is then sampled at this new location in all subintervals initially flagged in the validity check process. The viability of this idea is due to the fact that validity check fails when the sampling

point is outside the asymptotic regime of the formula and the leading term is no longer the dominant contributor to the defect expansion.

.

- Identifying the dominant term in the defect expansion and employ the boot-strapping process again in order to compute enough stages necessary to eliminate the error contributions due to the inherent polynomial interpolation error. In a similar manner to the discussions in chapter four of this thesis, the boot-strapping algorithm raises the error contributions of the error inherent in polynomial interpolation by an order leaving an interpolant dominated by data error. In this case the first two terms of the defect expansion will now be dominated by data error contribution from a single term each. The defect estimate process in the suspect subintervals now simplifies into locating the maximum of a single polynomial.

A second direction for future work involves further investigation of the direct, i.e. non-boot-strapping, approach for the determination of CMIRK schemes leading to asymptotically correct maximum defect estimates. A related investigation would consider how to obtain Hermite-Birkhoff forms for these schemes.

Bibliography

[1] U.M. Ascher, R.M.M. Mattheij, R.D. Russell, *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*, Classics in Applied Mathematics Series, SIAM, Philadelphia, 1995.

[2] Uri M. Ascher and Raymond J. Spiteri, *Collocation software for boundary value differential-algebraic equations*, SIAM J. Sci. Comput. 15 (1994), 938 - 952.

[3] U. Ascher, *Collocation for two-point boundary value problems revisited,* SIAM J. Numer. Anal. 23 (1986), 596 - 609.

[4] U. Ascher, J. Christiansen, R. D. Russell, *A collocation solver for mixed order systems of boundary value problems,* Math. Comp. 33 (1979), 659 - 679.

[5] U.M. Ascher and L. P. Petzold, *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations,* SIAM, Philadelphia, 1998.

[6] U. Ascher, J. Christiansen, R. D. Russell, *Collocation software for boundary value ordinary differential equations,* ACM Trans. Math. Software. 7 (1981), 209 - 222.

[7] U. Ascher, *Solving boundary-value problems with a spline-collocation code,* J. Comput. Phys. 34 (1980), 401 - 413.

[8] U.M. Ascher and G. Bader. *A new basis implementation for a mixed order boundary value ODE solver,* SIAM J. Sci. Statist. Comput. 8 (1987), 483 - 500.

[9] U. Ascher and R. D. Russell, *Evaluation of B-splines for solving systems of boundary value ordinary differential equations,* Tech. Rep., Dept. of Comp. Sci., University of British Columbia, 1977.

[10] J. Boisvert, P.H. Muir, R.J. Spiteri *A Runge-Kutta BVODE solver with global error and defect control,* ACM Trans. Math. 39 (2013), Art. 11.

[11] W.M.G. van Bokhoven, *Efficient higher order implicit one-step methods for integration of stiff differential equations,* BIT. 20 (1980), 34 - 43.

[12] R. Bulirsch, J. Stoer, P. Deuflhard, *Numerical Solution of Nonlinear Two-Point Boundary Value Problems,* Inst. of Math. Technische Universität München, 1977.

[13] K. Burrage, F. H. Chipman, P. H. Muir, *Order results for mono-implicit Runge-Kutta methods,* SIAM J. Numer. Anal. 31 (1994), 876 - 891.

[14] J.C. Butcher, *The Numerical Analysis of Ordinary Differential Equations,* Wiley, Chichester, 1987.

[15] S. Capper, J. Cash, F. Mazzia, *On the development of effective algorithms for the numerical solution of singularly perturbed two-point boundary value problems* Int. J. Comput. Sci. Math. 1 (2007), 42 - 57.

[16] J. R. Cash and F. Mazzia, *Hybrid mesh selection algorithms based on conditioning for two point boundary value problems,* J. Numer. Anal. Ind. Appl. Math. 1 (2006), 81 - 90.

[17] J. R. Cash and D. R. Moore, *A high order method for the numerical solution of two-point boundary value problems,* BIT. 20 (1980), 44 - 52.

[18] J. R. Cash and A. Singhal, *Mono-implicit Runge-Kutta formulae for the numerical integration of stiff differential systems,* IMA J. Numer. Anal. 2 (1982), 211 - 227.

[19] J. R. Cash, G. Moore, R.W. Wright, *An automatic continuation strategy for the solution of singularly perturbed linear two-point boundary value problems,* J. Comput. Phys. 122 (1995), 266 - 279.

[20] J.C. Diaz, G. Fairweather, P. Keast, *Algorithm 603. COLROW and ARCECO: FORTRAN packages for solving certain almost block diagonal linear systems by modified alternate row and column elimination,* ACM Trans. Math. Software. 9 (1983), 376 - 380.

[21] R. England, N. Nichols, J. Reid, *Subroutine D003AD*, Harwell subroutine library, Harwell, England, 1973.

[22] W.H. Enright. *Continuous numerical methods for odes with defect control,* J. Comput. Appl. Math. 125 (2000), 159 - 170.

[23] W.H. Enright and P.H. Muir, *New interpolants for asymptotically correct defect control of BVODES,* Numer. Alg. 53 (2010), 219 - 238.

[24] W. H. Enright, *The relative efficiency of alternative defect control schemes for high-order continuous runge-kutta formulas,* SIAM J. Numer. Anal. 30 (1993), 1419 - 1445.

[25] W. H. Enright and P. H. Muir, *Super convergent interpolants for the collocation solution of boundary value ordinary differential equations,* SIAM J. Sci. Comput. 21 (1999), 227 - 254.

[26] W. H. Enright and P. H. Muir, *Runge-Kutta software with defect control for boundary value ODEs,* SIAM J. Sci. Comput. 17 (1996), 479 - 497.

[27] W.H. Enright, K.R. Jackson, S.P. Norsett, *Interpolants for Runge-Kutta formulas,* ACM Trans. Math. Software. 12 (1986), 193 - 218.

[28] W.H. Enright and W.B. Hayes, *Robust and reliable defect control for Runge-Kutta Methods,* ACM Trans. Math. Software. 33 (2007), 1 - 19.

[29] S. Gupta, *An adaptive boundary value Runge-Kutta solver for first order boundary value problems,* SIAM J. Numer. Anal. 22 (1985), 114 - 126.

[30] N. Hale and D.R. Moore, *A sixth-order extension to the MATLAB package bvp4c of J. Kierzenka and L. F. Shampine,* Tech. Rep., Oxford University Computing Laboratory, Numerical Analysis Group, Oxford, 2004.

[31] H. B. Keller, *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations,* A. K. Aziz, ed., Academic Press, New York, (1975), 27 - 88.

[32] Herbert B. Keller, *Accurate difference methods for nonlinear two point boundary value problems,* SIAM, J. Numer. Anal. 11 (1974), 305 - 320.

[33] J. Kierzenka and L.F. Shampine, *A BVP Solver based on residual control and the MATLAB PSE,* ACM Trans. Math. Software. 27 (2001), 299 - 316.

[34] J. Kierzenka and L.F. Shampine, *A BVP Solver that controls residual and error,* JNAIAM J. Numer. Anal. Ind. Appl. Math. 3 (2008), 27 - 41.

[35] M. Lentini and V. Pereyra, *An adaptive finite difference solver for nonlinear two point boundary problems with mild boundary layers,* SIAM J. Numer. Anal. 14 (1977), 91 - 111.

[36] R.M.M. Mattheij and G.W.M. Staarink, *MUSN,* http://www.netlib. org/ode/, June 1992.

[37] R.M.M. Mattheij and G.W.M. Staarink, *An efficient algorithm for solving general linear two point bvp,* Report 8220, Math. Inst. Catholic University, Nijmegen, 1982.

[38] P.H. Muir, R.N. Pancer, K.R. Jackson, *PMIRKDC: a parallel mono-implicit Runge-Kutta code with defect control for boundary value ODEs,* Parallel Comput. 29 (2003), 711 - 741.

[39] P.H. Muir and M. Adams, *Mono-implicit Runge-Kutta-Nystrom methods with application to boundary value ordinary differential equations,* BIT 41 (2001), 776 - 799.

[40] P.H. Muir, *Optimal discrete and continuous mono-implicit Runge-Kutta schemes for BVODEs,* Adv. Comput. Math. 10 (1999), 135 - 167.

[41] Paul Muir and Brynjulf Owren, *Order barriers and characterizations for continuous mono-implicit Runge-Kutta schemes,* Math. Comp. 61 (1993), 675 - 699.

[42] Paul Muir and W.H. Enright, *Relationships among some classes of implicit Runge-Kutta methods and their stability functions,* BIT 27 (1987), 403 - 423.

[43] S. P. Norsett, E. Hairer, G. Wanner, *Solving Ordinary Differential Equations,* I. Springer-Verlag, Berlin, 1993.

[44] J. D. Riley, D. D. Morrison, J. F. Zancanaro, *Multiple shooting methods for two-point boundary value problems,* Comm. AMC. 1 (1962), 613 - 614.

[45] L. F. Shampine, P.H. Muir, and H. Xu, *A user-friendly Fortran BVP solver,* J. Numer. Anal. Ind. Appl. Math. 1 (2006), 201 - 217.

[46] L. F. Shampine and P.H. Muir, *Estimating conditioning of BVPs for ODEs,* Math. Comput. Modeling. 40 (2004), 1309 - 1321.

[47] R. Weiss, *The application of implicit Runge-Kutta and Collocation methods to boundary value problems,* Math. Comp. 28 (1974), 449 - 464.

[48] M.H. Wright and J.R. Cash, *A deferred correction method for nonlinear two-point boundary value problems: implementation and numerical evaluation,* SIAM J. Sci. Statist. Comput. 12 (1991), 971 - 989.

[49] J. Cash, http://www2.imperial.ac.uk/ jcash/BVP_software/PROBLEMS.PDF