uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Hemza Yagoub

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M.Sc. (Mathematics)

GRADE / DEGREE

Department of Mathematics and Statistics

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Variable-step Variable-order 3-stage Hermite-Birkhoff ODE/DDE Solver of Order 5 to 15

TITRE DE LA THÈSE / TITLE OF THESIS

Dr. R. Vaillancourt

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

Dr. T. Giordano

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

Dr. T. Yeap

Dr. D. Amundsen

Gary W. Slater

Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

Variable-step Variable-order 3-stage Hermite–Birkhoff ODE/DDE Solver of Order 5 to 15

Hemza Yagoub

Thesis submitted to the Faculty of Graduate and Postdoctoral Studies
in partial fulfilment of the requirements for the degree of Master of Science in
Mathematics [1]

Department of Mathematics and Statistics
Faculty of Science
University of Ottawa

[1]The M.Sc. program is a joint program with Carleton University, administered by the Ottawa-Carleton Institute of Mathematics and Statistics

Your file Votre référence
ISBN: 978-0-494-51663-8
Our file Notre référence
ISBN: 978-0-494-51663-8

Abstract

This thesis presents the variable-step variable-order 3-stage Hermite–Birkhoff numerical solver HB515DDE of order 5 to 15. This method can solve ordinary and delay differential equations (ODEs/DDEs) with state-dependent, non-vanishing, small, vanishing and asymptotically vanishing delays. Delayed values are computed using Hermite interpolation and small delays are dealt with using extrapolation. Discontinuities in DDEs are located by a bisection method. HB515DDE was tested and compared with other solvers. The results are given along with the convergence theory which supports the experimentation.

Résumé

Cette thèse introduit le solveur numérique Hermite–Birkhoff HB515DDE à 3 étages et à pas et ordres variables d'ordre 5 à 15. Cette méthode est construite pour résoudre des équations différentielles ordinaires ou avec retard. Les valeurs antérieures sont calculées avec un interpolant de type Hermite et l'extrapolation est utilisées pour le cas des petits retards. Les discontinuités introduites par le retard sont localisées avec la méthode de la bisection. Des tests numériques ont été effectués sur HB515DDE et les résultats sont exposés dans cette thèse en plus de la théorie de convergence qui appuie la partie expérimentale.

Acknowledgements

I would like to thank my family for their patience and continuous support. I would also like to thank my supervisor, Professor Rémi Vaillancourt, and his research associate, Dr. Truong Nguyen-Ba, for their enormous help and support.

Special thanks to my co-supervisor, professor Giordano, for his time and his suggestions.

I gratefully acknowledge the National Sciences and Engineering Research Council of Canada and the University of Ottawa for providing me with financial support. I also thank the Department of Mathematics and Statistics for providing me with all the facilities for a great research experience throughout my Master's program.

Dedication

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The most general delay differential equation (DDE) we consider is of the form

$$\begin{cases} y'(t) = f\left(t, y(t), y\left(\alpha_1(t, y(t))\right), \ldots, y\left(\alpha_s(t, y(t))\right)\right), & t_0 \le t \le t_f, \\ y(t) = \phi(t), \quad r \le t \le t_0, \end{cases} \tag{1.0.1}$$

where

$$y : [t_0, t_f] \to \mathbb{R}^d, \qquad f : [t_0, t_f] \times \mathbb{R}^d \times \cdots \times \mathbb{R}^d \to \mathbb{R}^d,$$

$$\phi : [r, t_0] \to \mathbb{R}^d \quad \text{and} \quad \alpha_i : [t_0, t_f] \times \mathbb{R}^d \to [r, t_f]$$

with

$$r < t_0 \quad \text{and} \quad i = 1, 2, \ldots, s.$$

These functions satisfy some smoothness requirements that we will introduce later.

Delay differential equations have many applications in sciences and engineering including population dynamics, infectious diseases, physiological and chemical kinetics, circuit theory... The delay can appear in the form of a latent component which only produces an effect after a certain period of time and its impact can depend on many parameters. Examples of delayed phenomena include the gestation time in a predator-prey model, the incubation period for bacteria, the capacitor response in an electrical circuit, the feedback components in an optical device, iterations and repetitive signals in control of parallel problems...

Hence, we want to construct a numerical code which can efficiently solve as many different types of DDEs as possible. Since many available ODE methods can solve very hard problems, it is tempting to use the ODE technology to solve DDEs because both types of differential equations seem, at first sight, very similar. However, it must be taken into account that the $\alpha_i(t, y(t))$ arguments have a great effect on the behavior of the differential equation and give the delay differential equation properties that are sometimes very different from those of the same equation without delay.

Therefore, having in mind the different properties of both types of differential equations, our goal is to transform the variable-step variable-order 3-stage Hermite–Birkhoff ODE solver of order 5 to 15 (HB(5-15)3 or HB515 for short) constructed in [22] into a DDE solver that we call HB515DDE.

In Chapter 2, we introduce basic properties of DDEs. This will help us understand the step-by-step transformation, given in Chapter 3, of the ODE solver HB515 into the DDE method HB515DDE. In Chapter 4, we prove the convergence of our two solvers. Then, we display in Chapter 5 the results of numerical tests where HB515DDE was compared with other known DDE solvers. Finally, in Chapter 6, we summarize our work and mention directions for future investigations.

Note that the work on HB515DDE presented in this thesis paper is based on the numerical article [30] and the theoretical article [28]. We mention that a similar transformation was done on the variable-step constant-order 7-stage Hermite–Birkhoff–Taylor ODE solver HBT(8)7 of order 8 which was transformed into the DDE solver HBT8DDE presented in [31] and on the variable-step variable-order 3-stage

Hermite–Birkhoff–Obrechkoff ODE solver HBO(4-14)3 with order varying between 4 and 14 constructed in [23] and the resulting DDE solver HBO414DDE was exhibited in [29].

1.1 Contributions

As it can be seen in the numerical chapter of the thesis, many experts in the numerical solutions of differential equations base their DDE solvers on a constant-order Runge–Kutta structure. However, the solver HB515DDE combines the following:

1) a Runge–Kutta structure which gives the solver a better overview of the behavior of the function at hand on the integration interval;

2) a multistep structure which allows the method to raise the order from 5 to 15 while always using three function evaluations. This low usage of function evaluations is an obvious advantage over high order Runge–Kutta methods at stringent tolerances as can be seen in Chapter 5;

3) a variable-order which enables the method to adapt itself to the different behaviors of the DDEs. Hence, lower order can be used when the DDE can be easily integrated using big stepsizes and to lower the number of backsteps when a discontinuity is close to the integration point. On the other hand, when a large number of reliable and discontinuity-free backsteps is available, these values can be used to gain a high order integration with no extra function evaluations.

Due to the advantages above, the solver HB515DDE gave impressive results offering the best number of function evaluations over maximum relative error ratio for all seven test problems. Hence, the easy-to-use variable-step variable-order solver HB515DDE is a very promising method for the increasingly complex DDE problems that we encounter because it offers both stability and efficiency.

Chapter 2

Elementary Concepts of DDEs

Since our goal is to transform an ODE solver into a DDE solver, we must study the properties of DDEs and compare them with those of an ODE.

Consider the following DDE

$$\begin{cases} y'(t) = f\left(t, y(t), y\left(\alpha_1(t, y(t))\right), \ldots, y\left(\alpha_s(t, y(t))\right)\right), & t_0 \leq t \leq t_f, \\ y(t) = \phi(t), \quad r \leq t \leq t_0, \end{cases} \tag{2.0.1}$$

where

$$y : [t_0, t_f] \to \mathbb{R}^d, \qquad f : [t_0, t_f] \times \mathbb{R}^d \times \cdots \times \mathbb{R}^d \to \mathbb{R}^d,$$

$$\phi : [r, t_0] \to \mathbb{R}^d \quad \text{and} \quad \alpha_i : [t_0, t_f] \times \mathbb{R}^d \to [r, t_f]$$

with

$$r < t_0 \quad \text{and} \quad i = 1, 2, \ldots, s.$$

The main difference that one sees between (2.0.1) and an ODE is the appearance of the arguments $\alpha_i(t, y(t))$. These functions deeply change the behavior and the properties of the differential equation. In this chapter, we will only tackle notions which must be taken into account when solving DDEs using an ODE method. We refer to [2, Chap. 1] for a more detailed view of the different properties which distinguish a DDE from an ODE.

First, we introduce the following definitions.

Definition 2.0.1 *Let* $i \in \{1, \ldots, s\}$.

1. *The function* $\alpha_i(t, y)$ *is called the delay,* $y(\alpha_i(t, y))$ *is called the delayed argument, and* $t - \alpha_i(t, y)$*, the lag denoted by* $\tau_i(t, y)$.

2. *If the delay* $\alpha_i(t, y)$ *depends on* $y(t)$*, it is said to be state dependent (and hence, the corresponding lag is also state dependent). Else, they are said to be state independent.*

3. *A state independent lag* $\tau_i(t)$ *is said to be variable if it depends on* t*. Else, it is said to be constant.*

The delayed argument is then an evaluation of the function y at some time t with $t \in [r, t_f]$. Since the concept of delay does not appear in ODEs, an interpolant should be added to a discrete solver to compute delayed values. One can use known interpolation schemes such as Newton, Hermite or Hermite–Birkhoff depending on the available information or construct a special interpolant that fits the needs of the method as Enright and Hu did in [7].

Moreover, it can be seen that the usual initial condition $y_0 = y(t_0)$ for ODEs is not sufficient for DDEs as soon as some delay satisfies $\alpha_i(t, y(t)) < t_0$. When this happens, we are forced to provide some initial function $\phi(t)$, called *history*, whose domain of definition usually includes all possible values $\alpha_i(t, y(t))$ that can be encountered during the integration.

2.1 Special Delays

It is very useful to study the different forms that a delay can take. Indeed, this tells us which delays may introduce numerical difficulties. On the other hand, some delays give the DDE special properties which can be used to make the integration more efficient.

Definition 2.1.1 *Let $i \in \{1, \ldots, s\}$.*

1. *The lag τ_i is said to be nonvanishing if there exists $\varepsilon > 0$ such that for all $t \geq t_0$ we have $\tau_i(t, y) \geq \varepsilon$ (equivalently $\alpha_i(t, y) \leq t - \varepsilon$).*
2. *The lag τ_i is said to be vanishing at t_* if $\tau_i(t, y) \to 0$ as $t \to t_*$.*
3. *The lag τ_i is said to be asymptotically vanishing if $\lim_{t\to\infty} \tau_i(t, y) = 0$.*
3. *When integrating over the step $[t_n, t_{n+1}]$, if $\alpha_i(t, y) > t_n$ for some $t \in [t_n, t_{n+1}]$, we say that we have a small delay.*

The first main difficulty in solving (2.0.1) comes from small delays. Note that small delays are naturally handled by implicit solvers but are a nuisance for explicit methods because they ask for a value of y which is not yet available. Strategies to avoid implicit equations for explicit solvers include reducing the stepsize until $\alpha_i(t, y) \leq t_n$, $\forall t \in [t_n, t_{n+1}]$ and $\forall i \in \{1, \ldots, s\}$. Else, extrapolation can be used.

2.2 The Appearance of Discontinuities

The second main problem in solving (2.0.1) is the appearance and propagation of discontinuities. A detailed theory on this phenomenon can be found in [2, Chap. 2]. Let us study this important property through the DDE test Problem 1 of Chapter 5.

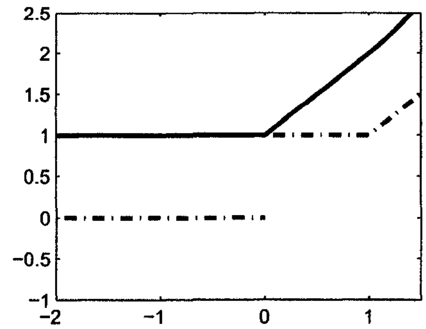


Figure 2.1: Independent variable t (horizontal axis) versus dependent variables y (solid line) and y' (dash-dot line) for Problem 1.

We have

$$\begin{cases} y'(t) = y(t-1), & t \in [0, 15], \\ \phi(t) = 1, & t \in [-1, 0], \end{cases}$$

with exact solution

$$y(t) = \sum_{i=0}^{\lfloor t \rfloor + 1} \frac{(t-i+1)^i}{i!}, \quad t \in [0, 15].$$

Comparing the exact solution with the history, it is seen that y is continuous at 0 but y' is not because $y'(0^-) = \phi'(0^-) = 0$ and $y'(0^+) = \phi(-1) = 1$ giving us

$$y'(0^-) \neq y'(0^+).$$

Hence, the junction between the history and the function y is not smooth (see Figure 2.1). This discontinuity, which is very common in DDEs, does not appear in ODEs because the history is defined at a single point.

The appearance of this discontinuity at t_0 is already a major obstacle that a normal ODE solver cannot overcome. Still, this is only the tip of the iceberg because the discontinuity at 0 propagates into an infinite set of other discontinuities. Indeed, using the fact that $y^{(2)}(t) = y'(t-1)$, we get that

$$y^{(2)}(1^-) = y'(0^-) \neq y'(0^+) = y^{(2)}(1^+)$$

which implies the presence of a discontinuity in $y^{(2)}$ at $t = 1$. Using a similar argument and noticing that $y^{(i)}(t) = y'(t-i+1)$, we realize that the discontinuity in y' at $\xi_1 = 0$ propagates into discontinuities which appear at $\xi_2 = 1, \xi_3 = 2, \ldots$ with $\xi_i = i-1$ being a discontinuity in the ith derivative of y, $i \geq 1$.

When the order of the derivative of y at which the discontinuity appears increases during the propagation, we say that the discontinuity smooths out. This smoothing phenomenon does not always take place and some DDEs, which are not part of the class of problems we consider, may have many propagated discontinuities in the first derivative of y. Hence, one must be careful when programming a DDE code which strongly relies on y' by making sure that one always saves both left and right values of y' at all discontinuities which affect the first derivative.

Moreover, when some derivative of ϕ, f or α_i (for some i) is not continuous then the resulting discontinuity can also propagate into other discontinuities.

Definition 2.2.1

1. *The discontinuity at t_0 due to a non-smooth junction between y and ϕ and its propagations are called primary discontinuities.*

2. *If ϕ, f or α_i (for some i) have some discontinuity ξ with respect to t in some of their derivatives, then ξ and its propagations are called secondary discontinuities.*

Remark. *When constructing our DDE solver, we do not consider secondary discontinuities and we suppose that no discontinuity in y' appears at $t > t_0$. We also* *assume that primary discontinuities are sufficiently spaced to avoid clustering of low order discontinuities (see [2] for more details).*

Finally, the number of propagated discontinuities may grow exponentially in the case of DDEs with multiple delays. Indeed, if we have n delays, then each discontinuity ξ may propagate into up to n discontinuities and each propagation of ξ may also propagate into up to n other discontinuities and so on.

To illustrate this, suppose that we have three constant lags $\tau_1 = 1, \tau_2 = 1.5$ and $\tau_3 = 1.75$ and a primary discontinuity at $\xi_1 = 0$. Then, $\xi_1 = 0$ can propagate into $\xi_2 = 1, \xi_3 = 1.5$ and $\xi_4 = 1.75$ and each of ξ_2, ξ_3 and ξ_4 can also propagate following the three lags and so on.

Thus, the set of primary discontinuities which are the propagation of $\xi_1 = 0$ can include all possible combinations

$$a\tau_1 + b\tau_2 + c\tau_3$$

with $a, b, c \in \mathbb{N}$ which would imply that

$$\xi_1 = 0 < \xi_2 = 1 < \xi_3 = 1.5 < \xi_4 = 1.75 < \xi_5 = 2 < \xi_6 = 2.5 < \xi_7 = 2.75 \ldots$$

Locating the discontinuities which are harmful for the solver is very important but the exponential growth of the propagated discontinuities tells us that we should be selective and not pay attention to the harmless ones; else we would have our hands full. This issue and solutions to it will be discussed in Chapter 3.

Chapter 3

From an ODE to a DDE Solver

In this chapter, we walk through the steps of transforming the ODE solver HB515 into the DDE solver HB515DDE. First, in order to understand what kind of solver we are working with, we briefly introduce the ODE method HB515 (see [22] for more details).

3.1 The ODE Solver HB(5-15)3

3.1.1 Hermite–Birkhoff Solvers

Many recent hybrid ODE solvers which combine the Runge–Kutta and the multistep structures were introduced. Some of them use Hermite–Birkhoff type (HB type) interpolation polynomials in the main integration formulae. They include the basic Hermite–Birkhoff (HB) solver HB515 which only uses the first derivative of y in its integration formula, the Hermite–Birkhoff–Obrechkoff (HBO) solvers (see [24], [23], [20], [21], [19] and [18]) which use both y' and y'' and the Hermite–Birkhoff–Taylor (HBT) solvers ([13], [15], [17], [16] and [14]) which use many derivatives of y.

Depending on how they are constructed, solvers have different strength points.

Many DDE codes adapted from ODE codes are based on constant-order Runge–Kutta structure because they provide a lot of freedom (self-starting, etc.) but multistep methods have their own advantages. Indeed, astronomers consider that multistep methods are good for long simulations and the use of backsteps lowers the number of function evaluations.

Since ODEs have different behaviors, different solvers are needed. Still, a variable order is generally preferable to constant order and a large number of stages usually stabilizes the method. For HB type methods, HBT solvers are more precise than HB or HBO because of the large number of derivatives they use but HB and HBO can handle low order discontinuities better than HBT. Indeed, the more derivatives HBT uses, the higher the discontinuity order must be before HBT can handle it.

3.1.2 The Solver HB(5-15)3

Let us concentrate now on the variable-step variable-order 3-stage Hermite–Birkhoff ODE solver HB515 of order 5 to 15 which was introduced in [22]. When it is started, the solver calls the embedded Runge–Kutta method Dormand–Prince DP(4,5)7M (or DP45 for short) to compute at least three initial steps after which HB515 takes over. The idea behind HB515 is to force an expansion of the numerical solution to agree with a Taylor expansion of the true solution. This leads to multistep- and Runge–Kutta-type order conditions which are reorganized into linear confluent Vandermonde-type systems. Fast algorithms are developed for solving these systems of order p in $O(p^2)$ operations to obtain HB interpolation polynomials in terms of generalized Lagrange basis functions.

The resulting method is as follows. Suppose we are solving the ODE

$$\begin{cases} y'(t) = f(t, y(t)), & t_0 \leq t \leq t_f, \\ y(t_0) = y_0, \end{cases} \tag{3.1.1}$$

where $y : [t_0, t_f] \to \mathbb{R}^d$ and $f : [t_0, t_f] \times \mathbb{R}^d \to \mathbb{R}^d$ and suppose that we are integrating (3.1.1) over the step $[t_n, t_{n+1}]$ with stepsize h_{n+1} and integration order p.

As explained in details in [22], our predictor-corrector scheme uses variable coefficients formulae. Hence, HB515 first solves the linear systems $M^l u^l = r^l$, $l = 1, \ldots, 4$, to get the coefficients $\alpha_{ij}, \beta_{ij}, a_{ij}$ and b_{ij} appearing in the formulae below. Then, the method computes two predicted values Y_{n+c_2} (by Predictor P_2) at $t_n + c_2 h_{n+1}$ and Y_{n+c_3} (by Predictor P_3) at $t_n + c_3 h_{n+1}$, both of local order $p-2$, where $c_2 = 2/3$ and $c_3 = 1$. The two predictors P_2 and P_3 are given by

$$Y_{n+c_2} = \alpha_{20} y_n + \alpha_{21} y_{n-1} + h_{n+1}\left(a_{21} f_n + \sum_{i=1}^{p-4} \beta_{2j} f_{n-j} \right)$$

and

$$Y_{n+c_3} = \alpha_{30} y_n + \alpha_{31} y_{n-1} + h_{n+1}\left(a_{31} f_n + a_{32} F_{n+c_2} + \sum_{i=1}^{p-4} \beta_{3j} f_{n-j} \right)$$

where $f_{n-j} = f(t_{n-j}, y_{n-j})$ with $j = 0, \ldots, p-4$ and $F_{n+c_2} = f(t_n + c_2 h_{n+1}, Y_{n+c_2})$.

Then, we compute the corrected value y_{n+1} to order p at t_{n+1} using the integration formula (IF)

$$y_{n+1} = \alpha_{10} y_n + \alpha_{11} y_{n-1} + h_{n+1}\left(b_{11} f_n + b_{12} F_{n+c_2} + b_{13} F_{n+c_3} + \sum_{j=1}^{p-4} \beta_{1j} f_{n-j} \right)$$

where $F_{n+c_3} = f(t_n + c_3 h_{n+1}, Y_{n+c_3})$.

Once this is done, the method evaluates $f_{n+1} = f(t_{n+1}, y_{n+1})$ and uses the predictor P_4 to get $\tilde{y}_{n+1}$ at t_{n+1} to order $p-2$ using the formula

$$\tilde{y}_{n+1} = y_{n+1} + h_{n+1}\left(a_{41}f_n + a_{42}f_{n+1} + \sum_{i=1}^{p-4}\beta_{4j}f_{n-j}\right).$$

Predictor P_4 is used to decide whether the step should be accepted or not.

When HB515 begins, it integrates with order 5 and advances using backsteps computed by DP45. While the acceptance of a step only depends on the predictor P_4, three extra error estimators are computed after a step is accepted to decide whether the order should be decreased (if it is greater than 5), increased (if it is lower than 15) or kept unchanged. These three error estimators do not use extra function evaluations.

The upper part of the unscaled regions of absolute stability, R, of HB(5-15)3 are shown in grey in Fig. 3.1. The region R is symmetric with respect to the real axis. The good shape of the stability regions is remarkable.

The main procedures for the stepsize selection and the order variation were adapted from the work of Shampine and Gordon in [26] and are discussed in more details in Chapter 4. When coded in C++, the ODE solver HB515 was most efficient at stringent tolerances for problems where $y' = f(t, y)$ was expensive to evaluate as in the cubic wave problem, outperforming the Dormand–Prince DP(8,7)13M (see [22]).

3.2 Locating Discontinuities

ODE solvers are known to behave poorly when confronted to low order discontinuities. Hence, crossing a low order discontinuity usually produces a large estimated error giving us a powerful tool for detecting those discontinuities. However, the ODE method cannot handle the discontinuity by itself and hence the solver cannot be used

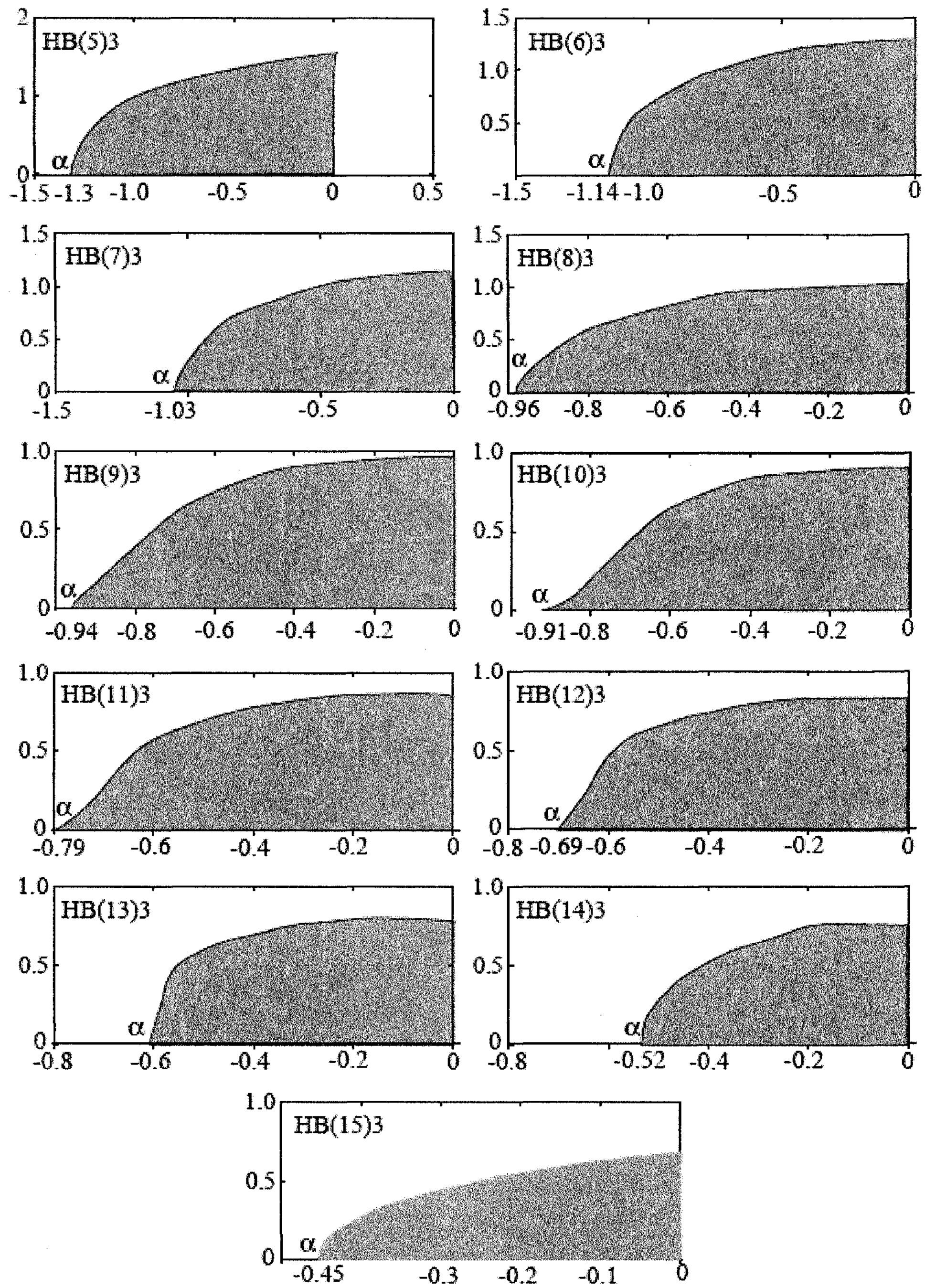


Figure 3.1: Unscaled regions of absolute stability, R, of HB(5-15)3.

to integrate DDEs directly. Therefore, it should be called on intervals of time where no discontinuity of low order exists. For this, the position of the low order discontinuity should be well approximated so that it can be added to the mesh points which constitute the endpoints of the intervals on which the solver is called.

Some DDE solvers locate the discontinuities as they advance. After a suitable stepsize was chosen for accuracy, Karoui and Vaillancourt in [11] called a bisection method before each step to check whether a discontinuity was hidden inside the interval they were trying to integrate on. In the presence of a discontinuity, the stepsize was lowered so that the new integration interval was discontinuity-free. From our point of view, this approach has the disadvantage of taking into account discontinuities that can be of no nuisance to the solver (when their order is high enough or when the size of the jump is very small).

On the other hand, Enright and Hayashi monitored in [6] the size of the defect. The defect is given by

$$\tilde{y}'(t) - f(t, \tilde{y}(t), \tilde{y}(t - \tau(t, \tilde{y}(t))))$$

where $\tilde{y}(t)$ is the continuous approximation to the solution that is constructed along the integration. Only after a step was rejected did they suspect the presence of a discontinuity. They then decreased the stepsize and, after a step was finally accepted, they called a bisection method to locate the possible discontinuity.

Willé and Baker explained in [27] how it was possible to use network theory to track the propagation of discontinuities up to some order. After the tracking algorithm was done, the solver was called on the intervals delimited by the found discontinuities. The tracking was easy in the case of state independent delays but nontrivial in the state dependent case because it would ask for a reliable approximation to $y(t)$ at various $t > t_0$ before the integration even started.

Another disadvantage of tracking is experienced when using multistep methods or when the Runge–Kutta values k_i are not saved. Indeed, many steps may be lost because, between any two discontinuities, the solver needs to place enough mesh points for an eventual high order interpolation (when computing delayed values). Thus, any numerically negligible discontinuity can cause the waste of many steps.

Consequently, because the number of propagated discontinuities can grow exponentially, we chose to call the bisection algorithm for the location of a discontinuity only after a step was rejected to avoid dealing with many harmless discontinuities.

Finally, the theory shows that accurately locating the discontinuities is very important to get a DDE solver with high convergence order. Hence, Karoui and Vaillancourt located the discontinuities with a precision of 1E-16 and Enright and Hayashi chose to locate the discontinuity with a precision of $\texttt{TOL}/|\Delta|$ where TOL is the given tolerance and $|\Delta|$ is the size of the defect at a sample point.

It seems to us that the first option is too stringent for the case of state dependent delays where the y-values are only known to a precision of TOL. Indeed, in the bisection algorithm, the y-values are used to evaluate the state dependent delay $\alpha(t, y(t))$ at many sample points in order to locate the discontinuity. Hence, when TOL is relaxed, the y-values have low precision and this can have a direct effect on the accuracy of the bisection method.

On the other hand, a "non-careful" implementation of the second option would give an inaccurate approximation of the discontinuity when $|\Delta| < 1$ which implies that the requested bound would be greater than TOL. Hence, we chose to ask for a precision of 1E-16 for state independent delays and TOL for state dependent delays.

3.3 Handling a Discontinuity

As explained above, we gave up systematic tracking of discontinuities and chose to deal with numerically significant discontinuities only. To achieve this, we mainly followed the idea of Enright *et al.* that was presented, justified and detailed in [5], [6] and [10] for the location of discontinuities.

As we explained above, when integrating on $[t_n, t_n + H]$, we do not suspect the presence of a discontinuity unless the step is rejected. Then, if the step is rejected, we keep on lowering the stepsize until one of two following scenarios happens.

If by lowering the rejected stepsize we hit the minimum stepsize $h_{\min}$ without having an accepted stepsize, we must run some special code. Hence, we call the bisection routine to check if a discontinuity is behind these rejections. If it is the case, we start the discontinuity location process to get a good approximation of the discontinuity and we add it to the mesh points. Else, we suspect that the rejections are due to a quick change in the function behavior. Thus, we use extrapolation to get over that "high turbulence" region. This process gave very satisfactory results for the seven test problems described in Chapter 5.

On the other hand, if a stepsize $h \geq h_{\min}$ finally passes the error control test, then we do not accept it directly. Indeed, we first suspect that a discontinuity lies in $I = [t_n + h, t_n + H]$ so we call a bisection algorithm on I. If no discontinuity is found, we accept the step $[t_n, t_n + h]$, i.e. we put $h_{n+1} = h$ and we continue the integration. Else, the bisection method is asked to bound the discontinuity $t_d \in I$ by some values t_L and t_H, $t_L \leq t_d \leq t_H$, at the required precision ($t_H - t_L \leq$ TOL or $t_H - t_L \leq$ 1E-16.

See above).

Once the discontinuity is located precisely, our goal is to jump over it using extrapolation. Indeed, we want to extrapolate the value of y at t_H giving us our next grid point after which we restart the solver. To do the extrapolation, we need to make sure that $t_n + h$ is not too far from t_H. So we define

$$\tilde{h} = t_H - t_n \quad \text{and} \quad \bar{h} = \min\{\tilde{h}/1.126, t_L - t_n\}$$

where the 1.126 makes sure that the quotient of successive stepsizes agrees with the theory presented in Chapter 4. We then apply the following algorithm adapted from [6, Fig. 5] (the comments are in italic style).

if $\bar{h} > 1.1h$ **then** *//(C*)*

  attempt the step $[t_n, t_n + \bar{h}]$. *//because $t_n + h$ is too far from t_H*

  if the step is accepted **then**

    put $h_{n+1} = \bar{h}$ and extrapolate y at t_H using values up to $t_n + \bar{h}$. Hence, $h_{n+2} = \tilde{h} - \bar{h}$.

  else

    put $h_{n+1} = h$ and accept the step $[t_n, t_n + h_{n+1}]$.

  end if

else *//extrapolation can be done directly because $t_n + h$ and t_H are close*

  extrapolate y at t_H using values up to $t_n + h$.

end if

We note that Enright and Hayashi's condition (C*) is

$$\tilde{h}/h_n > 1.2 \quad \text{and} \quad (\bar{h} - h_n)/h_n > 0.1,$$

where h_n in their notation corresponds to our h. The second part is obviously the same as our $\bar{h} > 1.1h$ condition but the first part is superfluous because it follows from the first condition and from the fact that they defined $\bar{h} = \min\{\tilde{h}/1.1, t_L - t_n\}$. Indeed,

$$1.1h < \bar{h} \Rightarrow 1.2h \leq (1.2/1.1)\bar{h} < 1.1\bar{h} \leq \tilde{h}.$$

This explains why we only used the second part of their condition (C*).

3.4 Computing Delayed Values

Suppose for this section that we are integrating over the step $[t_n, t_{n+1}]$ and consider for simplicity the case of one delay $\alpha(t, y)$. In the case of a state independent delay, finding the value of $\alpha(t)$ for $t \in [t_n, t_{n+1}]$ is done by a direct evaluation of the delay function. However, in the case of a state dependent delay, the task of computing the value $\alpha(t, y)$ for $t \in [t_n, t_{n+1}]$ is more complex. If the solver provides a sufficiently accurate approximation to $y(t)$ before each function evaluation (i.e. evaluation of $f(t, y(t), y(\alpha(t, y)))$ where the value $\alpha(t, y)$ is needed) then this approximation is used. Else, the solver extrapolates existing y-values to get a good approximation to $y(t)$ to be used in finding $\alpha(t, y)$.

Next, having $\alpha(t, y)$, we want to compute $y(\alpha(t, y))$. Now, if there exists a mesh point $t_i \leq t_n$ such that $\alpha(t, y) = t_i$ then $y(\alpha(t, y))$ is taken to be y_i. This happens rarely and when it does not happen, we have to look into the following two cases:

When $\alpha(t, y) < t_n$, the solver must choose appropriate points $t_k, \ldots, t_{k+m-1} \leq t_n$ with $\alpha(t, y) \in]t_k, t_{k+m-1}[$ to approximate $y(\alpha(t, y))$ by Hermite interpolation. Since HB515 is a variable order solver of order $p \in \{5, 6, \ldots, 15\}$, the interpolation order will depend on p.

Else, $\alpha(t, y) > t_n$ and the explicit solver has to cope with an implicit equation. As explained before, this difficulty can be overcome by extrapolating y as in

Karoui [12] or by special interpolants as Enright did in [7] to avoid far extrapolations. The current version of HB515DDE only uses extrapolation.

3.4.1 Special Technique for Asymptotically Vanishing Delays

For asymptotically vanishing delays, Karoui and Vaillancourt provided in [12] two different methods. The first, called "Method 1", used the usual extrapolation to solve any asymptotically vanishing delay problem. In "Method 2", they suggested to integrate the DDE problem (scalar y case)

$$\begin{cases} y'(t) = f\left(t, y(t), y(\alpha(t,y))\right), \quad t \geq t_0, \\ y(t) = \phi(t), \quad t \leq t_0, \end{cases}$$

up to some t_ε after which they would switch to the ODE approximation of the differential equation (see [12] for the details)

$$\begin{cases} y'(t) = \dfrac{f\left(t, y(t), y(t)\right)}{1 + [t - \alpha(t,y)] f_z(t, y(t), y(t))}, \quad t \geq t_0, \\ y(t) = \phi(t), \quad t \leq t_0, \end{cases}$$

where $f_z(t, y(t), y(t))$ is the partial derivative of f with respect to the third argument computed at the point $(t, y(t), y(t))$.

The choice of t_ε after which the DDE is switched to the ODE must be suitable to ensure that the approximations are accurate enough. This very original idea was then tested by the two authors on two DDE problems with one state independent asymptotically vanishing delay in each problem and gave encouraging improvements over "Method 1". However, it must be noted that the second method, as it was implemented by the authors, had the following disadvantages:

(1) The user had to know in advance which delays would vanish asymptotically in order to provide the appropriate ODE function that would replace the DDE.

Difficulty: This is simple for state independent delays but very difficult for state dependent delays because bounds for $y(t)$ are needed in order to have some knowledge of the asymptotic behavior of $\alpha(t, y)$.

(2) The user was asked to provide an explicit formula for the partial derivatives of f with respect to all delayed terms $y(\alpha_i(t, y))$ which vanish asymptotically. *Difficulty:* A software could be used to give an explicit formula for the partial derivative for simple DDEs but it is clear that this cannot be done for some complex equations.

(3) The user was not given any explicit formula to compute t_ε. He/she was asked to enter a number manually. *Difficulty:* The theoretic conditions for the choice of t_ε cannot be used because they require knowledge of the behavior and require bounds for $y(t)$. It is clear that one does not want to guess a t_ε when one wants to satisfy stringent tolerances.

After discussing the matter with Professor Karoui and looking into his useful suggestions, we decided to implement both of his methods. Even though many challenges arise in an effective implementation of the second method, many DDEs that are used to model natural phenomena are simple and hence "Method 2" could be used to solve them.

We tried many approaches to make "Method 2" more effective and we finally came up with the following algorithm. The user has the choice of providing a t_ε (Option 1) after which the ODE is switched to without any interference from the solver (to check for accuracy) as in Karoui's implementation. The other option (Option 2) is for the user to give a t_* from which the solver begins to check for a suitable t_ε. Note that putting $t_* < t_0$ makes the solver check for a t_ε from the beginning of the integration and this removes the third disadvantage stated above.

When "Option 2" is chosen, the solver compares, at about every ten steps, then the y computed using the DDE with the one given by the ODE. If the approximation is accurate and the error is found to satisfy the tolerance bound for five *consecutive* steps, the ODE is finally switched to for the rest of the integration. Checking the accuracy of the ODE integration for five consecutive steps avoids (or tries to avoid) using this technique on vanishing delays which are not asymptotic.

Finally, because "Method 2" requires additional input from the user and is difficult to use on complex DDEs, it is only provided as an optional alternative to "Method 1" which is used by default.

3.5 Efficiency Matters

For efficiency, the user may optionally fill an array called `depDev` where he/she indicates to the solver which coordinate of y each delayed argument α_j is related to, i.e. for which y_i do we need to compute the value of $y_i(\alpha_j(t, y))$. This is mainly useful for large systems of DDEs or when dealing with a large number of delayed arguments because, instead of calculating the value of all $y_i(\alpha_j(t, y))$ for each α_j, we can limit ourselves to the needed ones only. This lowers the number of interpolations/extrapolations necessary to find those values.

Also, HB515DDE requires the error to satisfy both the absolute and relative tolerances. At first thought, this constraint seemed too restrictive but it actually gave us very good results in past experiences so we kept it. Note that the requirement of satisfying the relative tolerance is sometimes waived when many steps are rejected successively and the solver is only asked to satisfy the absolute tolerance.

Chapter 4

Convergence Theory

In [22], we constructed the ODE solver HB515 but we did not prove its convergence. Hence, we first prove that the ODE solver HB515 is convergent of order 5 and then we show that the DDE solver HB515DDE is convergent of order 3 under some assumptions.

4.1 Convergence of HB515

Following the notation of [2], we want to prove the convergence of the ODE solver HB515 when applied to the initial value problem

$$\begin{cases} y'(t) = f(t, y(t)), & t_0 \leq t \leq t_f, \\ y(t_0) = y_0, \end{cases} \tag{4.1.1}$$

where $f(t, y) \in C^0([t_0, t_f] \times \mathbb{R}^d, \mathbb{R}^d)$ is globally Lipschitz continuous with respect to y in a given norm $\|\cdot\|$ of $\mathbb{R}^d$, i.e. there exists some $L > 0$ such that

$$\|f(t, y_1) - f(t, y_2)\| \leq L\|y_1 - y_2\| \quad \forall t \in [t_0, t_f] \quad \text{and} \quad \forall y_1, y_2 \in \mathbb{R}^d.$$

Let $\Delta = \{t_0, \ldots, t_N = t_f\}$ be a mesh and let the stepsize h_{n+1} be given by

$h_{n+1} = t_{n+1} - t_n$ for $n = 0, \dots, N-1$. Next, we define the general k-step method

$$y_{n+1} = \alpha_{n,1} y_n + \cdots + \alpha_{n,k} y_{n-(k-1)} + h_{n+1} \Phi(y_n, \dots, y_{n-(k-1)}; f, \Delta_n), \tag{4.1.2}$$

with $n \geq k-1$, $\Delta_n = \{t_{n-(k-1)}, \dots, t_n, t_{n+1}\}$ and where the increment function Φ satisfies a global Lipschitz condition with respect to the y arguments.

Convergence theory for the general method (4.1.2) used as an ODE solver or a DDE solver was developed in [2] so we will first make sure that we satisfy the conditions of the theorems in [2] before we prove the convergence of our method. The first step is to rewrite, for $p \in \{5, \dots, 15\}$, the $(p-3)$-step HBp method in the form of equation (4.1.2). Hence, let $p \in \{5, \dots, 15\}$ and $n \geq p-4$. We get

$$y_{n+1} = \alpha_{n,1} y_n + \alpha_{n,2} y_{n-1} + \cdots + \alpha_{n,p-3} y_{n-(p-4)} + h_{n+1} \Phi(y_n, \dots, y_{n-(p-4)}; f, \Delta_n) \tag{4.1.3}$$

where $n \geq p-4$ and $\Delta_n = \{t_{n-(p-4)}, \dots, t_n, t_{n+1}\}$. In the notation of [22], $\alpha_{n,i} = 0$ for all $i \in \{3, \dots, p-3\}$, $\alpha_{n,1}$ and $\alpha_{n,2}$ correspond to α_{10} and α_{11}, respectively, and Φ is given by

$$\Phi(y_n, \dots, y_{n-(p-4)}; f, \Delta_n) = b_{11} f_n + b_{12} F_{n+c_2} + b_{13} F_{n+c_3} + \sum_{j=1}^{p-4} \beta_{1j} f_{n-j} \tag{4.1.4}$$

where

$$F_{n+c_2} = f\left(\alpha_{20} y_n + \alpha_{21} y_{n-1} + h_{n+1}\left(a_{21} f_n + \sum_{i=1}^{p-4} \beta_{2j} f_{n-j}\right)\right)$$

and

$$F_{n+c_3} = f\left(\alpha_{30} y_n + \alpha_{31} y_{n-1} + h_{n+1}\left(a_{31} f_n + a_{32} F_{n+c_2} + \sum_{i=1}^{p-4} \beta_{3j} f_{n-j}\right)\right)$$

with $f_i = f(t_i, y_i)$ for all $i = n-(p-4), \dots, n$.

Secondly, the 1-step 7-stage DP45 can also be written in the form of equation (4.1.2) as follows. Let $n \geq 0$ be a step index then

$$y_{n+1} = y_n + h_{n+1} \Phi'(y_n; f, h_{n+1}) \tag{4.1.5}$$

where Φ' is the increment function for DP45 for which the coefficients are given in [4] (in the cited reference, DP45 is called RK5(4)7).

4.1.1 Constant-Step and Variable-Order

We will now deeply analyze the convergence of HB515 in the constant-step and variable-order case (CSVO). Firstly, we need to prove that the increment function Φ of (4.1.3) satisfied a Lipschitz condition with respect to the y argument. To do that, we must bound the coefficients of our integration method.

Boundedness of the Coefficients (HB515 and DP45)

Since we are considering CSVO, it is possible to find explicit values for the coefficients of the predictors P_2, P_3, the integration formula IF and the step control predictor P_4. Indeed, these coefficients are independent of n (in CSVO) but depend on the integration order.

Hence, we solved, for each order, the systems $M^1u^1 = r^1$, $M^2u^2 = r^2$, $M^3u^3 = r^3$ and $M^4u^4 = r^4$ defined in [22] (note that since we are considering constant stepsize, we have $h_{n+1}/h_n = 1$ for all n and hence $\eta_i = 1 - i$ for $i \in \{2, \ldots, 12\}$. See [22]). Next, we computed the maximum of the absolute value of each coefficient over all orders $p \in \{5, \ldots, 15\}$ and these bounds are given in Tables 4.1 and 4.2. Hence, it is trivial to find a uniform bound for all those coefficients and such a bound is $K_c = 185$.

For DP45, it is easy to see that 15 is an upper bound for all the coefficients appearing in its Butcher Tableau. Now, it is possible to deduce the required Lipschitz condition on the increment function Φ.

Pred. P_2	Max Value
α_{20}	7.373789131192e+00
α_{21}	1.267725442851e+01
a_{21}	5.354154046659e+00
β_{21}	1.229323234583e+01
β_{22}	1.010198025735e+01
β_{23}	1.304142129637e+01
β_{24}	1.455088215848e+01
β_{25}	1.300209462918e+01
β_{26}	9.020822150849e+00
β_{27}	7.619237643929e+00
β_{28}	8.665078399532e+00
β_{29}	9.691747891211e+00
β_{210}	1.070152716252e+01
β_{211}	1.169621195316e+01

Pred. P_4	Max Value
a_{41}	1.497907779087e+00
a_{43}	1.407796044440e+00
β_{41}	2.529842718423e+00
β_{42}	5.148749023996e+00
β_{43}	8.353585022761e+00
β_{44}	1.045591752561e+01
β_{45}	1.001944562465e+01
β_{46}	7.287053301238e+00
β_{47}	3.955385239997e+00
β_{48}	1.553646672142e+00
β_{49}	4.175722252213e-01
β_{410}	6.876437545209e-02
β_{411}	8.333333333333e-02
-	-

Table 4.1: Coefficients of P_2 and P_4

IF	Max Value
α_{10}	8.573388203018e-01
α_{11}	8.604881900714e-01
b_{11}	1.217909396358e+00
b_{12}	3.703703703704e-01
b_{13}	5.715592135345e-02
β_{11}	4.888761867360e-01
β_{12}	1.331755570059e-01
β_{13}	9.019177097007e-02
β_{14}	6.209491672050e-02
β_{15}	3.766439117974e-02
β_{16}	1.890113756238e-02
β_{17}	7.506321537931e-03
β_{18}	2.250885074152e-03
β_{19}	4.769913115603e-04
β_{110}	6.352277030229e-05
β_{111}	3.992914194077e-06

Pred. P_3	Max Value
α_{30}	8.362133720521e+01
α_{31}	1.360367556699e+02
a_{31}	5.001681866079e+01
a_{32}	1.407796044440e+00
β_{31}	1.398225448007e+02
β_{32}	1.230691444360e+02
β_{33}	1.627609277082e+02
β_{34}	1.840209095766e+02
β_{35}	1.658232452298e+02
β_{36}	1.157201905028e+02
β_{37}	6.547435896226e+01
β_{38}	7.863567181544e+01
β_{39}	9.231737401470e+01
β_{310}	1.064691762642e+02
β_{311}	1.210528308429e+02
-	-

Table 4.2: Coefficients of IF and P_3

Lipschitz Condition with Respect to y (HB515 and DP45)

Let h be a stepsize. In the following arguments we will use h_{n+1} and h interchangeably to designate the stepsize of the integration interval $[t_n, t_{n+1}]$ because the argument will generalize to the variable-step case that we will discuss later. Now, let $\Delta = \{t_0, \ldots, t_N = t_f\}$ be a mesh and let $y_i, \check{y}_i \in \mathbb{R}^d$ be approximations to $y(t_i)$ for $i = n-p+4, \ldots, n$. Put $\check{f}_i = f(t_i, \check{y}_i)$, $i = n-p+4, \ldots, n$. In the same way, $\check{F}_{n+c_2}$, $\check{F}_{n+c_3}$ and $\check{f}_{n+1}$ are given by the same formulas as F_{n+c_2}, F_{n+c_3} and f_{n+1}, respectively, with y_i being replaced by $\check{y}_i$, $i = n-p+4, \ldots, n$. Finally, let $\|\check{y} - y\| = \max\{\|\check{y}_n - y_n\|, \ldots, \|\check{y}_{n-p+4} - y_{n-p+4}\|\}$ where $\|\cdot\|$ is the norm given in (4.1.1). Hence,

$$\begin{aligned}
\|\check{F}_{n+c_2} - F_{n+c_2}\| \leq & \left\| f\left(t_n + c_2 h_{n+1}, \alpha_{20}\check{y}_n + \alpha_{21}\check{y}_{n-1} + h_{n+1}\left(a_{21}\check{f}_n + \sum_{j=1}^{p-4} \beta_{2j}\check{f}_{n-j}\right)\right)\right. \\
& \left. - f\left(t_n + c_2 h_{n+1}, \alpha_{20} y_n + \alpha_{21} y_{n-1} + h_{n+1}\left(a_{21} f_n + \sum_{j=1}^{p-4} \beta_{2j} f_{n-j}\right)\right)\right\| \\
& \leq L\left((|\alpha_{20}| + |\alpha_{21}|)\|\check{y} - y\| + h_{n+1}\left\| a_{21}(\check{f}_n - f_n) + \sum_{j=1}^{p-4} \beta_{2j}(\check{f}_{n-j} - f_{n-j})\right\|\right) \\
& \leq L(2K_c + h_{n+1}(p-3)K_c L)\|\check{y} - y\| \\
& \leq LK_c(2 + (t_f - t_0)(p-3)L)\|\check{y} - y\| \equiv \Psi_2\|\check{y} - y\|
\end{aligned} \tag{4.1.6}$$

where $\Psi_2 = LK_c(2 + (t_f - t_0)(p-3)L) > 0$. Also, using (4.1.6) we have

$$
\begin{aligned}
\|\check{F}_{n+c_3} - F_{n+c_3}\| \le & \Bigg\| f\Bigg(t_n + h_{n+1}, \alpha_{30}\check{y}_n + \alpha_{31}\check{y}_{n-1} + h_{n+1}\Bigg(a_{31}\check{f}_n + a_{32}\check{F}_{n+c_2} + \sum_{j=1}^{p-4}\beta_{3j}\check{f}_{n-j}\Bigg)\Bigg) \\
& - f\Bigg(t_n + h_{n+1}, \alpha_{30}y_n + \alpha_{31}y_{n-1} + h_{n+1}\Bigg(a_{31}f_n + a_{32}F_{n+c_2} + \sum_{j=1}^{p-4}\beta_{3j}f_{n-j}\Bigg)\Bigg)\Bigg\| \\
\le & L\Bigg(2K_c\|\check{y} - y\| + h_{n+1}\Bigg\|a_{31}(\check{f}_n - f_n) + a_{32}(\check{F}_{n+c_2} - F_{n+c_2}) \\
& + \sum_{j=1}^{p-4}\beta_{3j}(\check{f}_{n-j} - f_{n-j})\Bigg\|\Bigg) \\
\le & L(2K_c + h_{n+1}K_c[(p-3)L + \Psi_2])\|\check{y} - y\| \\
\le & L(2K_c + (t_f - t_0)K_c[(p-3)L + \Psi_2])\|\check{y} - y\| \equiv \Psi_3\|\check{y} - y\|
\end{aligned}
\tag{4.1.7}
$$

where $\Psi_3 = L(2K_c + (t_f - t_0)K_c[(p-3)L + \Psi_2]) > 0$.

Hence, from (4.1.6) and (4.1.7), we see that Φ satisfies

$$
\begin{aligned}
&\|\Phi(\check{y}_n, \check{y}_{n-1}, \dots, \check{y}_{n-p+4}; f, \Delta_n) - \Phi(y_n, y_{n-1}, \dots, y_{n-p+4}; f, \Delta_n)\| \\
&\qquad \le K_c((p-3)L + \Psi_2 + \Psi_3)\|\check{y} - y\| \equiv \Psi_1\|\check{y} - y\|
\end{aligned}
$$

where $\Psi_1 = K_c((p-3)L + \Psi_2 + \Psi_3) > 0$ and we are done for the Hermite–Birkhoff solver.

Next, DP45 being a 7-stage Runge–Kutta method and taking 15 as the uniform bound for the coefficients, we get

$$
\|\Phi'(y_n; f, h_{n+1}) - \Phi'(\check{y}_n; f, h_{n+1})\| \le \sum_{i=1}^{7} 15^i i h_{n+1}^i \|y_n - \check{y}_n\| \le 7\frac{15^8(t_f - t_0)^8 - 1}{15(t_f - t_0) - 1}\|y_n - \check{y}_n\|.
\tag{4.1.8}
$$

Putting

$$
L' = 7\frac{15^8(t_f - t_0)^8 - 1}{15(t_f - t_0) - 1}
$$

as the Lipschitz constant ends the Lipschitz condition for DP45.

Now, convergence will follow from the two important concepts of consistency and 0-stability.

Consistency (HB515 and DP45)

This section also applies to the variable-step case so we will keep our notation of h_{n+1} which designates the stepsize on the integration interval $[t_n, t_{n+1}]$.

Definition 4.1.1 *We say that the ODE method (4.1.2) is consistent of order (or, equivalently, has order) p if $p \geq 1$ is the largest integer such that, for all C^p-continuous functions f in (4.1.1) and for all mesh points, we have that*

$$\|y(t_{n+1}) - \hat{y}_{n+1}\| = O(h_{n+1}^{p+1})$$

uniformly with respect to $n = 0, 1, \ldots, N-1$, where $y(t)$ is the exact solution to (4.1.1) and

$$\hat{y}_{n+1} = \alpha_{n,1}y(t_n) + \cdots + \alpha_{n,k}y(t_{n-k+1}) + h_{n+1}\Phi(y(t_n), \ldots, y(t_{n-k+1}); f, \Delta_n).$$

To avoid confusion, we point out that there are two different notions of order used in this thesis: consistency and convergence orders (convergence order will be introduced below). As emphasized in the definition and unless otherwise specified, the term *order* refers to the consistency order.

In [22], we constructed the 3-stage variable-step variable-order Hermite-Birkhoff ODE solver HB(5-15)3 to satisfy the consistency order conditions for orders 5 to 15. Hence, we do not need to prove it here. On the other hand, DP45 was also constructed to satisfy the Runge–Kutta order conditions for order 5 and hence is consistent of order 5.

0-Stability (HB515 and DP45)

Let $n \in \{0, \ldots, N\}$ be the step index and $p \in \{5, \ldots, 15\}$ be the integration (consistency) order at step n. Define the first characteristic polynomial $p_n(x)$ as follows

$$p_n(x) = x^{p-3} - \sum_{i=1}^{p-3} \alpha_{n,i} x^{p-3-i}$$

where $\alpha_{n,i}$ are the coefficients appearing in (4.1.3). Since we are analyzing constant stepsize, then for a given order, all $p_n(x)$ are the same polynomial which we call $p(x)$. Since $\alpha_{n,i} = 0$ for all $i \geq 3$ and $\alpha_{n,2} + \alpha_{n,1} = 1$ (see [22]), a simple factorization gives

$$p(x) = x^{p-5}(x-1)(x + \alpha_{n,2}).$$

Now, we want to prove that $p(x)$ satisfies the well-known root condition.

Definition 4.1.2 *The polynomial $p(x)$ satisfies the root condition if*

(1) all roots r of $p(x) = 0$ lie inside the unit disk, i.e. $|r| \leq 1$ and

(2) if r is a root of $p(x) = 0$ of absolute value 1, then r is simple.

We use the fact that $\alpha_{n,2}$ is α_{11} in the notation of [22] and get from Table 4.2 that $\alpha_{n,2}$ is always smaller than 1. Hence, all the roots of $p(x) = 0$ being of absolute value smaller or equal to 1 and 1 being a simple root, we get that $p(x)$ satisfies the root condition.

Next, define the matrix

$$C_n = \begin{bmatrix} \alpha_{n,1} & \alpha_{n,2} & \alpha_{n,3} & \cdots & \alpha_{n,p-4} & \alpha_{n,p-3} \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix} = \begin{bmatrix} \alpha_{n,1} & \alpha_{n,2} & 0 & \cdots & 0 & 0 \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \tag{4.1.9}$$

as the companion matrix of the polynomial

$$p_n(x) = x^{p-3} - \sum_{i=1}^{p-3} \alpha_{n,i} x^{p-3-i}.$$

We now introduce the following definition adapted from [2] .

Definition 4.1.3 *The method HB515 in (4.1.3) satisfies the 0-stability condition if there exists a norm $\|\cdot\|$ on $\mathbb{R}^{p-3}$ independent of both n and Δ such that the operator norm on the matrix C_n satisfies*

$$\|C_n\| \leq 1.$$

Using the same argument as for the independence of $p(x)$ with respect to n, we realize that for a fixed order all C_n are equal to the same matrix C which is the companion matrix of $p(x)$. Solving our systems for the different orders, we found that $0 < \alpha_{n,2} < 1$ is always true and hence $0 < \alpha_{n,1} < 1$ because $\alpha_{n,1} + \alpha_{n,2} = 1$. Thus, it is easy to see that if we choose the infinity norm, we get

$$\|C\|_\infty = \sup_{\|x\|_\infty = 1} \|Cx\|_\infty \leq 1$$

which implies that our method is 0-stable.

More generally, Bellen and Zennaro remark in [2] that proving the first characteristic polynomial of method (4.1.2) satisfies the root condition is equivalent, for the constant-step case, to proving the 0-stability described in Definition 4.1.3.

On the other hand, we usually do not consider 0-stability for Runge–Kutta methods but because we use general theorems to prove the convergence of both HB515 and DP45, we need to say a word about it. Hence, we remark that the 0-stability is trivial for 1-step methods because the companion matrix of the first characteristic polynomial is a the trivial 1 by 1 identity matrix and thus the infinity norm can also be used to get that DP45 is 0-stable.

Convergence of DP45-HB515

Using the above information, we can state the convergence Theorem 4.1.4 which is adapted from a general theorem in [2].

Theorem 4.1.4 *Let the constant k-step ODE method (4.1.2) be consistent of order* $p = 5$, *0-stable and Lipschitz in* y. *Then, the following two conditions:*

1) the function $f(t, y)$ *in (4.1.1) is* C^5*-continuous;*

2) the set of starting values $y_0, \ldots, y_{k-1}$ *approximate the exact solution to order 5*

imply that the ODE method is convergent of order 5 on $[t_0, t_f]$, *that is*

$$\max_{1 \leq n \leq N} \|y(t_n) - y_n\| = O(h^5)$$

where h *is the integration stepsize.*

Proof: See the proof for Theorem 4.1.6 which is strictly more general than the one for the theorem above. ∎

Thus, we get convergence of the discrete constant-order ODE solvers HB5 and DP45 with convergence order 5. Moreover, using the fact that HB515 is consistent of order $p \in \{5, \ldots, 15\}$, it is easy to see that the convergence order cannot decrease because the local error is smaller when the order is increased. We also remark that the convergence order cannot increase because, unlike the consistency which is a local property, convergence considers the maximum error over all integration steps. Hence, the smallest order is the dominant one. Thus, the ODE solver DP45-HB515 is convergent of (convergence) order 5.

4.1.2 Variable-Step and Variable-Order

At first sight, the only difference between the variable-step variable-order (VSVO) case and the CSVO case is that the stepsize is not constant. Still, this small change makes the whole convergence theory a lot more difficult to put in place.

We recall that the stepsize control formulae used for our method are based on the well-known work of Shampine and Gordon in [26]. The two authors argued that close to constant stepsize was obtained when using this type of formulae. Hence, if we suppose that we have almost constant stepsize, the theory of constant stepsize could be generalized heuristically to the variable stepsize. However, we will try to be as rigorous as possible and avoid heuristics even though this will cost us some efficiency. We note that all the preliminary theory for DP45 still applies for the VSVO case because its coefficients are constant and hence no effect is noticed when we generalize from the CSVO to the VSVO case.

Boundedness of the Coefficients (HB515)

When we look at the systems used to compute the coefficients of method (4.1.3) as they appear in [22], we see that they are functions of the ratios of preceding stepsizes. Hence, we need to study the stepsize change rates in order to have an idea on the size of the coefficients. First, when the stepsize h_n is accepted for some $n \geq 0$, we choose h_{n+1} using a formula of the type

$$h_{n+1} = \min\left\{h_n \cdot \text{stfac} \cdot \left(\frac{\mathtt{TOL}}{\text{EST}}\right)^{1/(\kappa-1)}, c_1 h_n, h_{\max}\right\} \tag{4.1.10}$$

where stfac $= 0.81$ is a safety factor, EST is the local error estimate at step n, κ is the chosen order for the next integration step, $h_{\max}$ is the maximum allowed stepsize and $c_1 > 1$ is a constant. On the other hand, when, at the step n, a stepsize is rejected (call it $h_{n+1}^{(0)}$), we lower the stepsize using a formula of the type

$$h_{n+1}^{(i)} = \max\left\{h_{n+1}^{(i-1)} \cdot 0.7 \cdot \text{stfac} \cdot \left(\frac{\mathtt{TOL}}{\text{EST}}\right)^{1/\kappa}, 0.5 h_{n+1}^{(i-1)}, h_{\min}\right\} \tag{4.1.11}$$

where $i \geq 1$, κ is the actual integration order and $h_{\min} = c_2 h_n$ (h_n being the last accepted step, $c_2 \in]0, 1[$ is a constant and $h_{\min} \equiv \mathtt{TOL}$ when $n = 0$). Hence, we see from (4.1.10) and (4.1.11) that $c_2 \leq h_{n+1}/h_n \leq c_1$. After long experimental tests, we chose $c_1 = 1.9$ and $c_2 = 0.5^3$ giving us, for all $n \geq 0$,

$$0.5^3 \leq h_{n+1}/h_n \leq 1.9. \tag{4.1.12}$$

The choice of $c_2 = 0.5^3$ comes from the fact that we allow the solver to halve the stepsize at most three times. On the other hand, the systems involved in finding the coefficients were very ill conditioned when using $c_1 \approx 2$ so we took 1.9 to be on the safe side.

Since we only need some constant uniform bound on the coefficients of the method, we will not explicitly use the values c_1 and c_2 to bound our coefficients. Indeed, it remains a very difficult task to get an absolute bound for all the coefficients by analyzing the worst possible cases when solving the systems. Instead, we did some experimentations using the available information and we chose a large constant uniform bound $K = 2\text{E}25$ on the coefficients and we "forced" the solver to only accept steps for which the coefficients were smaller than K. Note that this bound is extremely big from a numerical point of view but to get the Lipschitz condition, we merely need to provide some constant upper bound for the coefficients. Hence, we found no reason to constrain our solver with a smaller K.

Next, we mention that proving that the method is consistent and satisfies a Lipschitz condition with respect to the y argument is the same as in the CSVO case so we will not repeat the procedure. The only difference is in the value of the uniform bound K which replaces the K_c.

0-Stability (HB515)

We now get to one of the major difficulties that we encountered in reconciling a rigorous theory with the experimentation. First, it is easy to see that the 0-stability (as it is given in Definition 4.1.3) is satisfied under the infinity norm if we impose

$$\alpha_{n,2} \in]0, 1[. \tag{4.1.13}$$

Now, we must clarify that (4.1.13) is a sufficient but definitely not a necessary condition for 0-stability. Indeed, we had pleasing results when we only asked for $|\alpha_{n,2}| \leq 1$ (see [22]). Actually, we already obtained good results when no constraint was imposed on $\alpha_{n,2}$. Still, it is very difficult to prove that big $\alpha_{n,2}$ values are compensated by smaller ones and by corrections taking place along the integration.

Another way to see how difficult it is to get minimal conditions for 0-stability is to look at the notion from the theory of Hairer *et al.* in [9]. Indeed, they link the 0-stability to finding a bound for finite products of successive matrices C_n defined in (4.1.9). Hence, it is now obvious that one cannot predict the norm of those products when no *simple* explicit formulae are available for the coefficients $\alpha_{n,1}$ and $\alpha_{n,2}$ let alone when no explicit formulae are available in the first place. Hence, imposing (4.1.13) was a very difficult choice to make and a major constraint to impose but we chose to abide by it in order to be certain that we satisfy the theoretical 0-stability condition.

Convergence of DP45-HB515

For sake of completeness, we restate and prove the convergence theorem for the general variable-step case which is adapted from [2]. Before that, we state and prove a preliminary lemma.

Lemma 4.1.5 *Let C be a real $k \times k$ matrix and $\|\cdot\|_*$ be a vector norm on $\mathbb{R}^k$. Let $d \geq 1$ be an integer. We define a vector norm on $\mathbb{R}^{dk}$ as follows. Suppose $B = [B^{(1)}, \dots, B^{(k)}]^T$ is a stacking of k vectors $B^{(i)}$ each of them belonging to $\mathbb{R}^d$. Then define*

$$|||B||| = \max_{1 \leq i \leq d} \left\| \left[B_i^{(1)}, \dots, B_i^{(k)} \right]^T \right\|_* . \tag{4.1.14}$$

For both norms, the induced matrix norm will be denoted in the same way as the vector norm to avoid confusion. Now, if $\|C\|_* \leq 1$, then

$$|||C \otimes I_d||| \leq 1$$

where $\otimes$ is the Kronecker product (see [8, §4.5.5]) and I_d is the $d \times d$ identity matrix.

Proof: We have $\|C\|_* \leq 1$. Then,

$$\sup_{z \neq 0} \frac{\|Cz\|_*}{\|z\|_*} \leq 1.$$

Now, we want to prove

$$|||C \otimes I_d||| = \sup_{x \neq 0} \frac{|||(C \otimes I_d)x|||}{|||x|||} \leq 1.$$

We have the following property on Kronecker products (see [8, p.180]):

$$y = (C \otimes I_d)x \iff Y = I_d X C^T = X C^T$$

where $y, x \in \mathbb{R}^{dk}$ are stackings of k vectors $y^{(1)}, \ldots, y^{(k)}$ and $x^{(1)}, \ldots, x^{(k)}$, respectively, with $y^{(i)}, x^{(i)} \in \mathbb{R}^d$, and $Y, X \in \mathbb{R}^{d\times k}$ are matrices whose columns are $y^{(1)}, \ldots, y^{(k)}$ and $x^{(1)}, \ldots, x^{(k)}$, respectively. Define on $\mathbb{R}^{d\times k}$ the norm

$$|||A|||' = \max_{1\le i\le d} \|A(i,:)^T\|_*.$$

Hence,

$$\sup_{x\neq 0} \frac{|||(C\otimes I_d)x|||}{|||x|||} \le 1 \iff \sup_{X\neq 0} \frac{|||XC^T|||'}{|||X|||'} \le 1.$$

Let $0 \neq X \in \mathbb{R}^{d\times k}$. We have

$$\frac{|||XC^T|||'}{|||X|||'} \le 1 \iff \frac{\max_{1\le i\le d} \|(XC^T)(i,:)\|_*}{\max_{1\le i\le d} \|(X(i,:))^T\|_*} \le 1 \iff \frac{\max_{1\le i\le d} \|C(X(i,:))^T\|_*}{\max_{1\le i\le d} \|(X(i,:))^T\|_*} \le 1.$$

Next, let k be such that $\max_{1\le i\le d} \|(XC^T)(i,:)\|_* = \|(XC^T)(k,:)\|_*$ and set $z = (X(k,:))^T$ (z cannot be the 0-vector since it is the row of X with biggest norm and if it were 0 then X would be the 0 matrix which is not the case by the choice of X). Hence,

$$\frac{\max_{1\le i\le d} \|C(X(i,:))^T\|_*}{\max_{1\le i\le d} \|(X(i,:))^T\|_*} = \frac{\|Cz\|_*}{\max_{1\le i\le d} \|(X(i,:))^T\|_*} \le \frac{\|Cz\|_*}{\|z\|_*} \le 1$$

by hypothesis. Therefore,

$$|||C\otimes I_d||| = \sup_{x\neq 0} \frac{|||(C\otimes I_d)x|||}{|||x|||} \le 1$$

which proves the statement. ∎

Theorem 4.1.6 *Let the variable k-step ODE method (4.1.2) be consistent of order 5, 0-stable and Lipschitz in y. Then, the following two conditions:*

1) the function $f(t, y)$ in (4.1.1) is C^5-continuous;

2) the set of starting values $y_0, \ldots, y_{k-1}$ approximate the exact solution to order 5;

imply that the ODE method is convergent of order 5 on $[t_0, t_f]$*, that is*

$$\max_{1 \leq n \leq N} \|y(t_n) - y_n\| = O(h^5)$$

where $h = \max_{1 \leq n \leq N}\{h_n\}$.

Proof: To avoid a conflict of notation, we will use Φ as the increment function for the k-step method at hand which will apply to both the increment functions of HB515 and DP45. We use the fact that the method is consistent of order 5 and get

$$y_{n+1} = \alpha_{n,1} y(t_n) + \cdots + \alpha_{n,k} y(t_{n-k+1}) + h_{n+1}\Phi(y(t_n), \ldots, y(t_{n-k+1}); f, \Delta_n) + \epsilon_{n+1}, \tag{4.1.15}$$

with

$$\|\epsilon_{n+1}\| \leq c h_{n+1}^6, \tag{4.1.16}$$

for some constant $c > 0$ for all $n = k-1, \ldots, N-1$.

We define the $\boldsymbol{y}_n = [y_n, y_{n-1}, \ldots, y_{n-k+1}]^T$ and $\boldsymbol{y}(t_n) = [y(t_n), y(t_{n-1}), \ldots, y(t_{n-k+1})]^T$ both of dimensions $dk \times 1$. Then, subtracting (4.1.2) from (4.1.15) gives us

$$\boldsymbol{y}(t_{n+1}) - \boldsymbol{y}_{n+1} = \zeta_n(\boldsymbol{y}(t_n) - \boldsymbol{y}_n) + h_{n+1}\Gamma_n + E_{n+1}, \tag{4.1.17}$$

$n = k-1, \ldots, N-1$, where $\zeta_n = C_n \otimes I_d$,

$$\Gamma_n = [\Phi(\boldsymbol{y}(t_n); f, \Delta_n) - \Phi(\boldsymbol{y}_n; f, \Delta_n), 0, \ldots, 0]^T$$

and $E_{n+1} = [\epsilon_{n+1}, 0, \ldots, 0]^T$, with 0 is the zero vector of $\mathbb{R}^d$.

Let $|||\cdot|||$ be the norm on $\mathbb{R}^{kd}$ defined in Lemma 4.1.5 (C and $\|\cdot\|_*$ in the lemma are C_n and $\|\cdot\|_\infty$, respectively, in this proof) and get that $|||\zeta_n||| \leq 1$.

Therefore, by (4.1.17) we get

$$|||\boldsymbol{y}(t_{n+1}) - \boldsymbol{y}_{n+1}||| \leq |||\boldsymbol{y}(t_n) - \boldsymbol{y}_n||| + h_{n+1}|||\Gamma_n||| + |||E_{n+1}|||,$$

$n = k-1, \ldots, N-1$. Next, since Φ is Lipschitz with respect to the y argument and by the equivalence of the norms in finite dimensional spaces, there exists a constant $Q > 0$ such that

$$|||\Gamma_n||| \leq Q|||\boldsymbol{y}(t_n) - \boldsymbol{y}_n|||.$$

Again, using the equivalence of the norms in finite dimensional spaces, there exists $c' > 0$ such that (4.1.16) becomes

$$|||\boldsymbol{y}(t_{n+1}) - \boldsymbol{y}_{n+1}||| \leq (1 + h_{n+1}Q)|||\boldsymbol{y}(t_n) - \boldsymbol{y}_n||| + c'h_{n+1}h^5,$$

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

We next define

$$\hat{c} = \max_{k \leq i \leq n+1} \frac{||\epsilon_i||}{h_i^6}$$

and get

$$\begin{aligned}
|||\boldsymbol{y}(t_{n+1}) - \boldsymbol{y}_{n+1}||| &\leq \left[\prod_{i=k}^{n+1}(1 + h_i Q)\right] |||\boldsymbol{y}(t_{k-1}) - \boldsymbol{y}_{k-1}||| + \left(\sum_{i=k}^{n+1}\left[\prod_{j=i+1}^{n+1}(1 + h_j Q)\right] h_i\right) \hat{c}h^5 \\
&\leq \left[\prod_{i=k}^{n+1} e^{h_i Q}\right] |||\boldsymbol{y}(t_{k-1}) - \boldsymbol{y}_{k-1}||| + \left(\sum_{i=k}^{n+1}\left[\prod_{j=i+1}^{n+1} e^{h_j Q}\right] h_i\right) \hat{c}h^5 \\
&\leq e^{Q(t_f - t_0)}|||\boldsymbol{y}(t_{k-1}) - \boldsymbol{y}_{k-1}||| + e^{Qt_f}\left(\sum_{i=k}^{n+1} e^{-Qt_i}h_i\right) \hat{c}h^5 \\
&\leq e^{Q(t_f - t_0)}|||\boldsymbol{y}(t_{k-1}) - \boldsymbol{y}_{k-1}||| + e^{Qt_f}\left(\int_{t_0}^{t_f} e^{-Qt}dt\right) \hat{c}h^5 \\
&= e^{Q(t_f - t_0)}|||\boldsymbol{y}(t_{k-1}) - \boldsymbol{y}_{k-1}||| + \frac{e^{Q(t_f - t_0)} - 1}{Q}\hat{c}h^5,
\end{aligned}$$

$n = k-1, \ldots, N-1$. Thus, again by equivalence of the norms, since the starting values approximate the solution $y(t)$ up to order 5, the proof is complete. ∎

The above theorem gives us convergence of order 5 for the variable-step constant-order HB5. Hence, we can deduce the convergence of order 5 for the variable-order

HB515 because the local error does not increase when the order is raised. Finally, assuming the consistency of DP45, we can adapt the theorem to get that DP45 is convergent of order 5. Thus, the variable-step variable-order ODE solver DP45-HB515 is convergent of convergence order 5.

4.2 Convergence of HB515DDE

In this section, we prove, under some assumptions, the convergence of the general VSVO DDE solver HB515DDE for the problem

$$\begin{cases} y'(t) = f\left(t, y(t), y\left(t - \tau(t, y(t))\right)\right), & t_0 \le t \le t_f, \\ y(t) = \phi(t), \quad t \le t_0, \end{cases} \tag{4.2.1}$$

where $f(t,u,v) \in C^0([t_0,t_f] \times \mathbb{R}^d \times \mathbb{R}^d, \mathbb{R}^d)$ is globally Lipschitz continuous with respect to u and v in a given norm $\|\cdot\|$ of $\mathbb{R}^d$ with Lipschitz constants $L > 0$ and $M > 0$ respectively.

4.2.1 Effect of Discontinuities

The appearance and propagation of discontinuities is a major threat for the convergence of an ODE method. Hence, a lot of attention is required when modifying an ODE code to solve DDEs. Indeed, knowing which discontinuities the solver can detect enables us to determine the smoothness of the integrated function and the actual order of convergence.

Recall that the error estimate (EST) is given by

$$\mathrm{EST} = \|y_{n+1} - \tilde{y}_{n+1}\|_2$$

where y_{n+1} and $\tilde{y}_{n+1}$ are given by the integration formula IF and the corrector P_4, respectively. Then a special routine for detecting a discontinuity at $\xi \in (t_n, t_{n+1}]$ is called only if a step is rejected by the error test EST $\leq$ TOL. Hence, we must study the effect of crossing a discontinuity on the error estimate. This includes knowing how the size and the order of the jump discontinuity affect the error estimate and which coefficients of the method are involved in signaling the appearance of the discontinuity.

We first give the following definition adapted from [2].

Definition 4.2.1 *A discontinuity point ξ is said to be of order q if $y^{(v)}(\xi)$ exists for $v = 0, \ldots, q$ and $y^{(q)}$ is Lipschitz continuous at ξ.*

To illustrate the effect of discontinuities, suppose, for simplicity, that we have the following scalar DDE:

$$\begin{cases} y'(t) = f\left(t, y(t), y\left(\alpha(t)\right)\right), & t_0 \leq t \leq t_f, \\ y(t) = \phi(t), \quad t \leq t_0, \end{cases}$$

with α being a strictly increasing state independent delay. Suppose also that we are integrating at order $p \in \{5, \ldots, 15\}$ on $[t_n, t_{n+1}]$ and that (only) one jump discontinuity ξ lies on the integration interval. Let the order of the discontinuity be $q \geq 0$ (i.e. the discontinuity is in $y^{(q+1)}$) and the size of the jump be K_q.

Suppose for now that $t_n + c_2h < \xi < t_n + c_3h$. Hence, ξ does not affect Y_{n+c_2} which approximates the exact solution to order $p-2$. Also, Y_{n+c_3} is not *a priori* affected by ξ because it only uses values of y and f at $t < \xi$. However, because the extrapolation does not take ξ into account, Y_{n+c_3} will lie on some smooth continuation of y and not on y itself. This will lead to an inaccurate y_{n+1} because it uses $F_{n+c_3} = f(t_n + c_3h, Y_{n+c_3}, y(\alpha(t_n + c_3h)))$. Finally, $\tilde{y}_{n+1}$ will also get a share of inaccuracy because it uses the computed y_{n+1}. This lack of precision will appear as a huge error estimation whose size depends on q and K_q. Following a similar argument, if the discontinuity happens to be in the interval $]t_n, t_n + c_2h[$ then we expect

Min vals	$\|b_{12}\|$	$\|b_{13}\|$	$\|a_{43}\|$
P1	0.51030	0.002342	0.20934
P2	0.52876	0.037879	0.21398
P3	0.52876	0.039220	0.21213
P4	0.52885	0.004365	0.19592
P5	0.52885	0.038814	0.19837
P6	0.52885	0.001481	0.19592
P7	0.52892	0.023461	0.20785
Min	0.51030	0.001481	0.19592

Max vals	$\|b_{12}\|$	$\|b_{13}\|$	$\|a_{43}\|$
P1	0.74647	0.09918	0.49719
P2	0.68486	0.09605	0.46667
P3	0.68254	0.09605	0.46525
P4	0.74336	0.09603	0.49668
P5	0.68301	0.09603	0.46568
P6	0.74777	0.09603	0.49835
P7	0.71225	0.09602	0.47260
Max	0.74777	0.09919	0.49835

Table 4.3: Bounds for b_{12}, b_{13} and a_{43}.

that both Y_{n+c_2} and Y_{n+c_3} will be affected by it and, again, a large EST should appear.

Taking a close look at the terms where F_{n+c_2} and F_{n+c_3} appear in y_{n+1} and $\tilde{y}_{n+1}$ (which compose EST), we see that the main coefficients linked to the effect of the discontinuity are b_{12}, b_{13} and a_{43}. Since the routine for locating a discontinuity is called only after a step rejection, it is useful to make sure that b_{12}, b_{13} and a_{43} are neither too small in which case they would absorb the effect of the discontinuity, nor to large in which case they would amplify a small K_q which would have been of no nuisance to the solver. Hence, the size of the three coefficients has been monitored for the seven test problems given in [30] and lower and upper bounds for each coefficient are displayed in Table 4.3. These results show that b_{12}, b_{13} and a_{43} are close to constant and neither too small nor too big, as desired.

As pointed out by Calvo *et al.* in [3], a general concept for an adaptive code like ours is that a discontinuity of order q usually produces an error of $O(h^q)$ (i.e. of order $q-1$) at the error estimation. This can be deduced from the following adaptation of Taylor's theorem.

Theorem 4.2.2 *Let $n \geq 0$ be a step index and t_n and $h_{n+1} > 0$ be the corresponding* *mesh point and stepsize, respectively. Let $t_{n+1} = t_n + h_{n+1}$ and $r \geq 0$ be an integer. If $y \in C^r([t_n, t_{n+1}])$ and $y \in C^{r+1}(]t_n, t_{n+1}[)$ then*

$$y(t_{n+1}) = y(t_n) + \frac{y'(t_n)}{1!} h_{n+1} + \cdots + \frac{y^{(r)}(t_n)}{n!} h_{n+1}^r + O(h_{n+1}^{r+1}).$$

Because y_{n+1} is of order p and $\tilde{y}_{n+1}$ is of order $p-2$, then our error estimator is of order $p-2$. Hence, for y sufficiently smooth, EST can achieve an $O(h^{p-1})$ accuracy. However, if a discontinuity $\xi \in]t_n, t_{n+1}[$ of order q appears in $y^{(q+1)}$ then the error term is $O(h^q)$ and the local order becomes $q-1$. Therefore, if $q-1 \geq p-2$, then the method will remain of order $p-2$ and we may not even notice the discontinuity, whereas for $q \leq p-2$ the solver usually produces an EST which is larger than the prescribed tolerance for the current step.

Following a similar argument, the local error estimator (EST') of DP45 being of order 4, a discontinuity of order q may not be noticed by the local error estimator if $q-1 \geq 4$ whereas for $q \leq 4$ the solver usually produces a large EST' that we can use to signal the presence of the discontinuity. Hence, we make the following optional assumption.

Assumption 1: (Optional) *The error estimates of DP45 and HB515DDE detect all numerically significant discontinuities of order smaller than or equal to 3.*

Note that this assumption follows from the fact that the error estimates of DP45 and HBp, $p = 5, \ldots, 15$, are locally of order greater than or equal to 3. Moreover, Assumption 1 is justified since it is impossible to determine an exact order q such that the solver will always detect all discontinuities of order smaller than or equal to q because for any q, there exists, a jump discontinuity with a K_q small enough such that the discontinuity will not be seen by the solver.

Hence, there are always cases where the estimate does not detect the presence of a discontinuity of order smaller than or equal to the one of the local error estimator. In that case we assume that, because the error term is small, the theoretical discontinuity has no major consequence on the overall integration.

Giving exact conditions for the detection of some discontinuity in the general case is complex and not useful to the user of the solver. Indeed, the criteria for the detection of a discontinuity would involve computing data (giving sharp bounds on partial derivatives of f and on the deviated arguments, etc.) that the user would only have if he/she already knows $y(t)$ and has a deep analysis of its properties at hand. This would obviously go against the goal of solving the DDE numerically.

4.2.2 Hermite Interpolant $\eta(t)$

Now, we dedicate some time for the analysis of the Hermite interpolant. As a reminder from Chapter 3, we use a Hermite interpolating polynomial $\eta(t)$ to find the value of y at a delayed time t^* by interpolating/extrapolating past accepted y_j's. The number of points m used for the interpolation/extrapolation is given by the formula

$$m(p) = \lceil p/2 \rceil \tag{4.2.2}$$

where $p \in \{5, \ldots, 15\}$ is the actual integration order, $\lceil p/2 \rceil$ is the smallest integer not less than $p/2$. Hence,

$$3 \leq m(p) \leq 8.$$

Suppose we are integrating at order p. Let $s \geq 0$ and $t_s, \ldots, t_{s+m-1}$ be m distinct mesh points where m is given by equation (4.2.2). Put $u = s + m - 1 \leq N - 1$. If the solution y of (4.2.1) is $C^1([t_s, t_u])$, then the unique polynomial $\eta(t)$ of least degree with

1) $\eta(t_i) = y(t_i)$,

2) $\eta'(t_i) = f(t_i, y(t_i), z(t_i))$,

for all $i = s, \ldots, u$ is the Hermite polynomial of degree at most $2m-1$ given by

$$\eta(t) = \sum_{j=s}^{u} H_{m-1,j}(t)y(t_j) + \Psi(y(t_u), \ldots, y(t_s); t, f, \Delta') \tag{4.2.3}$$

where the increment is given by

$$\Psi(y(t_u), \ldots, y(t_s); t, f, \Delta') = \sum_{j=s}^{u} \widehat{H}_{m-1,j}(t) f(t_j, y(t_j), z(t_j))$$

with $\Delta' = \{t_s, \ldots, t_u\}$ and the coefficients are as follows

$$H_{m-1,j}(t) = [1 - 2(t-t_j)L'_{m-1,j}(t_j)]L^2_{m-1,j}(t) \quad \text{and} \quad \widehat{H}_{m-1,j}(t) = (t-t_j)L^2_{m-1,j}(t)$$

with

$$L_{m-1,j}(t) = \prod_{i=s, i\neq j}^{u} \frac{t-t_i}{t_j-t_i}$$

being the jth Lagrange polynomial of degree $m-1$.

First, as it was the case with the Hermite–Birkhoff method, we need to have a uniform bound on the various coefficients of the interpolant to be able to prove the convergence of our DDE solver.

Boundedness of the Coefficients (Interpolant)

As we did before with method (4.1.3), it is possible to bound the coefficients $H_{m-1,j}(t)$ and $\widehat{H}_{m-1,j}(t)$ using the bounds on the ratio of consecutive stepsizes h_{n+1}/h_n given in (4.1.12). For the Hermite–Birkohff method, we chose to study the behavior of the solver under the given bounds and then force the solver to lower the stepsize until the coefficients were bounded by the chosen constant. However, for the interpolant, we do not want to force step rejections and hence we will provide theoretical bounds for the most extreme case. Hence, using c_1 and c_2 defined in 4.1.2, we put $\bar{c}_2 = c_2/2^6 = 2^{-9}$ and the following holds

$$\bar{c}_2 \leq h_{n+1}/h_n \leq c_1$$

for all $n \geq 0$. Note that $\bar{c}_2$ gives us the freedom of halving the minimum stepsize that the solver produces at most six times. This will be useful when steps have to be inserted between two close discontinuities for high order interpolation (see Subsection 4.2.3).

Let $p \in \{5, \ldots, 15\}$ and m be the number of mesh points used for interpolation when integrating at order p. Put

$$h^* = \min_{s \leq l \leq u-1} \{t_{l+1} - t_l\}$$

where $u = s + m - 1$ and $s \geq 0$. Then, for $t \in [t_s, t_{u+1}]$ we have

$$|L_{m-1,j}(t)| \leq \prod_{i=s,i\neq j}^{u} \frac{|t-t_i|}{h^*} \leq \prod_{i=s,i\neq j}^{u} \frac{\sum_{l=0}^{m} h^* c_1^l}{h^*} = \left(\frac{c_1^{m+1}-1}{c_1-1}\right)^{m-1} \leq \left(\frac{c_1^{9}-1}{c_1-1}\right)^{7} \equiv \varrho.$$

Therefore,

$$|\widehat{H}_{m-1,j}(t)| = |(t-t_j)L^2_{m-1,j}(t)| \leq (t_f - t_0)\varrho^2.$$

On the other hand,

$$\begin{aligned}
|L'_{m-1,j}(t_j)| &= \left|\left(\prod_{i=s,i\neq j}^{u} \frac{1}{t_j - t_i}\right)\left(\sum_{i=s,i\neq j}^{u} \prod_{l=s,l\neq j,i}^{u} (t_j - t_l)\right)\right| \\
&\leq \frac{1}{(h^*)^{m-1}}\left(\sum_{i=s,i\neq j}^{u} \prod_{r=s,r\neq j,i}^{u} \left[\sum_{l=0}^{m-1} h^* c_1^l\right]\right) \\
&= \frac{m-1}{h^*}\left(\frac{c_1^m - 1}{c_1 - 1}\right)^{m-2} \leq \frac{7}{h^*}\left(\frac{c_1^8 - 1}{c_1 - 1}\right)^6 \equiv \frac{\tilde{\varrho}}{h^*}.
\end{aligned}$$

Hence,

$$\begin{aligned}|H_{2m-1,j}(t)| &= |1-2(t-t_j)L'_{m-1,j}(t_j)|L^2_{m-1,j}(t) \\ &\le (1+2|t-t_j||L'_{m-1,j}(t_j)|)L^2_{m-1,j}(t) \\ &\le \left[1+2h^*\left(\frac{c_1^{m+1}-1}{c_1-1}\right)\frac{\tilde{\varrho}}{h^*}\right]\varrho^2 \\ &\le \left[1+2\tilde{\varrho}\left(\frac{c_1^9-1}{c_1-1}\right)\right]\varrho^2.\end{aligned}$$

Now, it is easy to see that

$$K_{\text{int}} = \max\left\{(t_f-t_0)\varrho^2, \left[1+2\tilde{\varrho}\left(\frac{c_1^9-1}{c_1-1}\right)\right]\varrho^2\right\}$$

is a mesh-independent uniform bound on all the coefficients of the Hermite interpolant.

Lipschitz Condition with Respect to y (Interpolant)

We can see from (4.2.3) that the increment function Ψ of $\eta(t)$ satisfies a Lipschitz condition with respect to the y argument because

1) Ψ is linear in f;
2) f is Lipschitz with respect to y;
3) the coefficients of $\eta(t)$ are bounded by K_{int}.

Consistency (Interpolant)

In the case of a smooth y, we use the localizing assumption to get that the local error term of the Hermite interpolation/extrapolation at $t^* \in [t_s, t_{u+1}]$ is

$$\frac{y^{(2m)}(c)}{(2m)!}\prod_{i=s}^{u}(t^*-t_i)^2 = O\left(h^{2m}\right)$$

where $c \in [t_s, t_{u+1}]$ and $h = \max_{s \leq i \leq u} t_{i+1} - t_i$.

The above formula tells us that the local error is of order $2m - 1$ which is greater or equal to $p - 1$ by equation (4.2.2). Having the order of the local error of the interpolant greater or equal to $p - 1$ is required for convergence of order p. However, we will only be able to prove a theoretical convergence of order $p - 2$ but we try to get the best convergence order numerically.

Next, when y is not smooth, Assumption 1 tells us that HB515 detects all numerically significant discontinuities of order smaller or equal to 3. Upon its detection, HB515DDE includes the discontinuity in the mesh. It then calls the interpolant on intervals $[t_s, t_u]$ such that no tracked discontinuity lies in their interior. Hence, the local error of the Hermite interpolant cannot be greater than $O(h^3)$ giving us a consistency order of at least 2. Finally, since the error is $O(h^3)$ at any point $t \in [t_s, t_{u+1}]$, then the Hermite interpolant has uniform order 2 on $[t_s, t_{u+1}]$.

4.2.3 Restarting the Method.

As it is known in the literature, it is very important to restart a linear multistep method after crossing a discontinuity ξ_i. Indeed, the effect of the discontinuity can be disastrous for the reliability of the method when it uses values from both sides of the discontinuity. Hence, after a discontinuity ξ is precisely located (see [30]) and added to the mesh, the solver is restarted and the self-starting DP45 is called on the new integration interval $[\xi, t_f]$. Indeed, DP45 does not use backsteps for its integration so it is less affected than the Hermite–Birkhoff part by a "turbulence" in the behavior of the DDE. Then, DP45 computes enough backsteps so that HB515DDE can take over without feeling the effect of the discontinuity.

Hence, the intervals on which the method is called are usually bounded by two consecutive discontinuities. Therefore, it is worth making sure that there are enough mesh points in each interval $[\xi_{i-1}, \xi_i]$ so that high-order interpolation can be performed within the interval when a delayed value is needed at some $t \in]\xi_{i-1}, \xi_i[$. Since the maximum number of interpolation points is 8, then the solver is asked to insert enough steps between every two consecutive discontinuities until the number of mesh points in each interval $[\xi_{i-1}, \xi_i]$ is at least 8.

4.2.4 Convergence

First, let us prove the following two lemmas needed in the proof of Theorem 4.2.6.

Lemma 4.2.3 (HB515) *Let $p \in \{5, \ldots, 15\}$ and let $f = f(t, y(t))$ be defined as in (4.1.1) along with the given norm $\|\cdot\|$ of $\mathbb{R}^d$. Next, let $n \in \{p-4, \ldots, N-1\}$ and $y_{n-p+4}, \ldots, y_n$ be nodal values associated with a mesh Δ_n. Then, there exists $\gamma_f > 0$ such that for all $\hat{f} \in C^0([t_{n-p+4}, t_{n+1}] \times \mathbb{R}^d, \mathbb{R}^d)$ we have*

$$\|\Phi(\boldsymbol{y}_n; f, \Delta_n) - \Phi(\boldsymbol{y}_n; \hat{f}, \Delta_n)\| \leq \gamma_f \sup_{t_{n-p+4} \leq t \leq t_{n+1},\ y \in \mathbb{R}^d} \|f(t, y(t)) - \hat{f}(t, y(t))\|$$

where $\boldsymbol{y}_n = [y_n, \ldots, y_{n-p+4}]^T$ and Φ is the increment function defined in (4.1.4).

Proof: Put $I = [t_{n-p+4}, t_{n+1}]$. Then,

$$\begin{aligned} \|\Phi(\mathbf{y}_n; f, \Delta_n) - \Phi(\mathbf{y}_n; \hat{f}, \Delta_n)\| &\leq \left| b_{11} + b_{12} + b_{13} + \sum_{j=1}^{p-4} \beta_{1j} \right| \sup_{t \in I,\ y \in \mathbb{R}^d} \|f(t, y(t)) - \hat{f}(t, y(t))\| \\ &\leq (p-1)K \sup_{t \in I,\ y \in \mathbb{R}^d} \|f(t, y(t)) - \hat{f}(t, y(t))\| \end{aligned}$$

where K is the chosen uniform bound for the coefficients of the Hermite–Birkhoff method (4.1.3). Hence, taking $\gamma_f = 14K$ gives the desired bound and ends the proof. ∎

Lemma 4.2.4 (Hermite interpolant) *Let $p \in \{5, \ldots, 15\}$ and $m = m(p)$ be the number of interpolation/extrapolation points for the Hermite interpolant associated with the order p. Let $f = f(t, y(t))$ be defined as in (4.1.1) along with the given norm $\|\cdot\|$ of $\mathbb{R}^d$. Next, for $s \geq 0$ let $y_s, \ldots, y_{s+m-1}$ be nodal values associated with some mesh. Then, there exists $\gamma_{f,int} > 0$ such that for all $\hat{f} \in C^0([t_s, t_{s+m}] \times \mathbb{R}^d, \mathbb{R}^d)$ we have*

$$\begin{aligned} &\|\Psi(y_s, \ldots, y_{s+m-1}; t, f, \Delta') - \Psi(y_s, \ldots, y_{s+m-1}; t, \hat{f}, \Delta')\| \\ &\qquad\qquad \leq \gamma_{f,int} \sup_{t_s \leq t \leq t_{s+m},\ y \in \mathbb{R}^d} \|f(t, y(t)) - \hat{f}(t, y(t))\| \end{aligned} \tag{4.2.4}$$

where $\Delta' = \{t_s, \ldots, t_{s+m-1}\}$ and Ψ is the increment function of the Hermite interpolant defined in (4.2.3).

Proof: This can be proved using a similar argument as the one of Lemma 4.2.3 by noticing the linearity of Ψ in f and using the fact that all coefficients in Ψ are bounded by K_{int}. Thus, taking $\gamma_{f,\text{int}} = 8K_{\text{int}}$ ends the proof because $m \leq 8$. ∎

It must be noted that, in Lemma 4.2.4, $m + s - 1$ is always smaller than the index n of the last integrated step because we do not use interpolation points which are not yet computed. Also, the supremum was taken over $t \in [t_s, t_{s+m}]$ instead of $t \in [t_s, t_{s+m-1}]$ to include the extrapolation case which is done within $(t_{s+m-1}, t_{s+m}]$.

Because our solver is constructed to deal efficiently with many types of delays, many special codes were added to make the solver stable and able to adapt itself to the different behaviors of the DDE functions. Hence, for sake of rigor, we make the following assumptions:

1) When integrating on $[t_n, t_n + h]$, we ask for $\alpha(t, y(t)) \leq t_{n+1}$ for all $t \in [t_n, t_{n+1}]$. This constraint was also used by Baker *et al.* as mentioned in [1]. Then, to

avoid far extrapolations, we truncate to t_{n+1} any $\alpha(t, y(t))$ which exceeds it. We suppose that these truncations do not affect the order of convergence of the method.

2) Suppose there is a discontinuity at t_i, $i \geq 0$, and that an approximation to y is needed at $t > t_i$ when t_{i+1} is not available. Then, the solver is allowed to extrapolate over t_i i.e. the solver is allowed to use the m points $t_{i-m+1}, \ldots, t_i$ to extrapolate at t. We suppose that these extrapolations do not affect the convergence order of the method.

3) Suppose we are integrating over $[t_n, t_{n+1}]$. When a value at $t \in [t_i, t_{i+1}]$ is needed, $i \in \{0, \ldots, n\}$, the m-point interpolant does not always use the mesh points $t_{i-m+1}, \ldots, t_i$. Actually, these interpolating points are only chosen when $i = n$ or when there is a tracked discontinuity at t_i. Else, the solver picks, depending on their availability, the same number of interpolating points on both sides of t. This choice can only increase the accuracy of the approximation of y at t. To avoid a long case-by-case analysis of the different ways interpolation/extrapolation is done, we suppose that the solver always picks the interpolating points $t_{i-m+1}, \ldots, t_i$ for any interpolation within $[t_i, t_{i+1}]$ with a local error of at most $O(h^3)$.

Let us now prove the convergence of the 2-step HB5DDE together with the 3-point Hermite interpolant. Assuming the well-posedness of the problem along with Assumption 1, we state and prove Lemma 4.2.5 and Theorem 4.2.6 adapted from a general convergence theorem in [2].

Lemma 4.2.5 *Let $\zeta(t)$ and $\eta(t)$ be the numerical solutions of the initial value prob-*

lems

$$\begin{cases} z'_{n+1} = f\left(t, z_{n+1}, u(t-\tau(t, z_{n+1}(t)))\right), & t_n \le t \le t_{n+1}, \\ z_{n+1}(t_n) = z_n, \end{cases}$$

and

$$\begin{cases} w'_{n+1} = f\left(t, w_{n+1}, v(t-\tau(t, w_{n+1}(t)))\right), & t_n \le t \le t_{n+1}, \\ w_{n+1}(t_n) = w_n, \end{cases}$$

respectively, obtained by method (4.1.3) with starting values $\boldsymbol{z}_n = [z_n, z_{n-1}]^T$ *and* $\boldsymbol{w}_n = [w_n, w_{n-1}]^T$*, and by interpolant (4.2.3) with nodal values* $\tilde{\boldsymbol{z}}_n = [z_{n-2}, z_{n-1}, z_n]^T$ *and* $\tilde{\boldsymbol{w}}_n = [w_{n-2}, w_{n-1}, w_n]^T$*.*

Then there exist constants $P, Q, R, S, T > 0$*, independent of the mesh* Δ*, such that*

$$|||\boldsymbol{z}_{n+1} - \boldsymbol{w}_{n+1}||| \le (1 + h_{n+1}Q)|||\boldsymbol{z}_n - \boldsymbol{w}_n||| + h_{n+1}P \max_{t \le t_{n+1}} \|u(t) - v(t)\|, \tag{4.2.5}$$

where $\boldsymbol{z}_{n+1} = [\zeta(t_{n+1}), z_n]^T$*,* $\boldsymbol{w}_{n+1} = [\eta(t_{n+1}), w_n]^T$ *and* $|||\cdot|||$ *is the norm on* $\mathbb{R}^{2d}$ *defined by (4.1.14) with* $\|\cdot\|_* = \|\cdot\|_\infty$*, and*

$$\max_{t_n \le t \le t_{n+1}} \|\zeta(t) - \eta(t)\| \le (T + h_{n+1}S) \max_{n-2 \le s \le n} \|z_s - w_s\| + h_{n+1}R \max_{t \le t_{n+1}} \|u(t) - v(t)\| \tag{4.2.6}$$

Proof: By subtracting the two formulae (4.1.3) for $z_{n+1} = \zeta(t_{n+1})$ and $w_{n+1} = \eta(t_{n+1})$ we get

$$\begin{aligned} \boldsymbol{z}_{n+1} - \boldsymbol{w}_{n+1} &= \begin{bmatrix} \zeta(t_{n+1}) - \eta(t_{n+1}) \\ z_n - w_n \end{bmatrix} \\ &= \begin{bmatrix} \alpha_{n,1}(z_n - w_n) + \alpha_{n,2}(z_{n-1} - w_{n-1}) + h_{n+1}(\Phi(\boldsymbol{z}_n; f_u, \Delta_n) - \Phi(\boldsymbol{w}_n; f_v, \Delta_n)) \\ z_n - w_n \end{bmatrix} \\ &= (C_n \otimes I_d)(\boldsymbol{z}_n - \boldsymbol{w}_n) + h_{n+1}\Gamma_n \end{aligned}$$

where $\Gamma_n = [\Phi(\boldsymbol{z}_n; f_u, \Delta_n) - \Phi(\boldsymbol{w}_n; f_v, \Delta_n), 0]^T$ with $f_u(t, y) = f\left(t, y, u(t - \tau(t, y(t)))\right)$ and $f_v(t, y) = f\left(t, y, v(t - \tau(t, y(t)))\right)$.

Next, using the fact that $|||C_n \otimes I_d||| \leq 1$, we get

$$|||\boldsymbol{z}_{n+1} - \boldsymbol{w}_{n+1}||| \leq |||\boldsymbol{z}_n - \boldsymbol{w}_n||| + h_{n+1}|||\Gamma_n|||.$$

Hence, by equivalence of the norms on finite dimensional spaces, there exists $C > 0$ such that

$$\begin{aligned}|||\Gamma_n||| &\leq C\|\Phi(\boldsymbol{z}_n; f_u, \Delta_n) - \Phi(\boldsymbol{w}_n; f_v, \Delta_n)\| \\ &\leq C\left(\|\Phi(\boldsymbol{z}_n; f_u, \Delta_n) - \Phi(\boldsymbol{w}_n; f_u, \Delta_n)\| + \|\Phi(\boldsymbol{w}_n; f_u, \Delta_n) - \Phi(\boldsymbol{w}_n; f_v, \Delta_n)\|\right).\end{aligned}$$

Then, using the fact that Φ is Lipschitz with respect to the y argument, Lemma 4.2.3 and the fact that f is Lipschitz with respect to its third argument, then there exist $Q > 0$, $\gamma_f > 0$ and $P > 0$ such that

$$\begin{aligned}|||\Gamma_n||| &\leq Q\|\boldsymbol{z}_n - \boldsymbol{w}_n\| + C\gamma_f \sup_{t_{n-1} \leq t \leq t_{n+1},\ y \in \mathbb{R}^d} \|f_u(t, y) - f_v(t, y)\| \\ &\leq Q\|\boldsymbol{z}_n - \boldsymbol{w}_n\| + P \sup_{t_{n-1} \leq t \leq t_{n+1},\ y \in \mathbb{R}^d} \|u(t - \tau(t, y(t))) - v(t - \tau(t, y))\|.\end{aligned}$$

Now, when integrating on $[t_n, t_{n+1}]$, we constrain the solver to give $t - \tau(t, y(t)) \leq t_{n+1}$ for all $t \in [t_n, t_{n+1}]$. Then, we get (4.2.5).

As for the continuous extensions, if we put $\Delta'_n = \{t_{n-2}, t_{n-1}, t_n\}$ then we have, for all $t = t_n + \theta h_{n+1}$ with $\theta \in [0, 1]$,

$$\zeta(t) - \eta(t) = \sum_{j=n-2}^{n} H_{2,j}(t)(z_j - w_j) + h_{n+1}\left(\Psi(\tilde{\boldsymbol{z}}_n; \theta, f_u, \Delta'_n) - \Psi(\tilde{\boldsymbol{w}}_n; \theta, f_v, \Delta'_n)\right).$$

Therefore, putting

$$D_n = \max_{n-2 \leq s \leq n} \|z_s - w_s\|$$

gives

$$\begin{aligned}\max_{t_n \leq t \leq t_{n+1}} \|\zeta(t) - \eta(t)\| &\leq 3K_{\text{int}} D_n + h_{n+1} \max_{t_n \leq t \leq t_{n+1}} \|\Psi(\tilde{\boldsymbol{z}}_n; \theta, f_u, \Delta'_n) - \Psi(\tilde{\boldsymbol{w}}_n; \theta, f_v, \Delta'_n)\| \\ &\leq 3K_{\text{int}} D_n + h_{n+1} \max_{t_n \leq t \leq t_{n+1}} \left(\|\Psi(\tilde{\boldsymbol{z}}_n; \theta, f_u, \Delta'_n) - \Psi(\tilde{\boldsymbol{w}}_n; \theta, f_u, \Delta'_n)\|\right. \\ &\quad \left. + \|\Psi(\tilde{\boldsymbol{w}}_n; \theta, f_u, \Delta'_n) - \Psi(\tilde{\boldsymbol{w}}_n; \theta, f_v, \Delta'_n)\|\right).\end{aligned}$$

Now, using the fact that Ψ is Lipschitz with respect to the y and f arguments and that f is Lipschitz with respect to its third argument, there exist constants $R > 0$ and $S > 0$ such that

$$\begin{aligned}\max_{t_n \le t \le t_{n+1}} \|\zeta(t) - \eta(t)\| \le & (3K_{\text{int}} + h_{n+1}S)D_n \\ & + h_{n+1}R \max_{t_n \le t \le t_{n+1}} \|u(t - \tau(t, w_{n+1}(t))) - v(t - \tau(t, w_{n+1}(t)))\|\end{aligned}$$

which gives (4.2.6) after we put $T = 3K_{\text{int}}$ and use the fact that $t - \tau(t, y(t)) \le t_{n+1}$ for all $t \in [t_n, t_{n+1}]$. ∎

Theorem 4.2.6 *Consider the state dependent DDE*

$$\begin{cases} y' = f\left(t, y(t), y\left(t - \tau(t, y(t))\right)\right), & t_0 \le t \le t_f, \\ y(t) = \phi(t), \quad r \le t \le t_0, \end{cases} \tag{4.2.7}$$

where $f \in C^3([t_i, t_{i+1}] \times \mathbb{R}^d \times \mathbb{R}^d)$ *for all* $i = 0, \ldots, N-1$ *with the mesh being* $\Delta = \{t_0, t_1, \ldots, t_n, \ldots, t_N = t_f\}$, *the delay* $\tau \in C^3([t_0, t_f] \times \mathbb{R}^d)$ *and* $\phi \in C^3([r, t_0])$. *Moreover, we have that*

(h_1) *The mesh* Δ *includes all discontinuity points* $\xi_1 < \xi_2 < \cdots < \xi_s$ *of order smaller or equal to 3.*

(h_2) *The ODE method HB5DDE is restarted after each discontinuity point* ξ_i, $i = 0, 1, \ldots, s$, *by a method of order greater or equal to 2.*

(h_3) *The 3-point Hermite interpolant is consistent of uniform order 2.*

(h_4) *For each* n, *the interval where the interpolation takes place is included in* $[\xi_i, \xi_{i+1}]$ *for some* $i \in \{0, \ldots, s\}$.

Then the resulting DDE method has discrete global order and uniform global order 3, that is

$$\max_{1 \le n \le N} \|y(t_n) - y_n\| = O(h^3)$$

and

$$\max_{t_0 \le t \le t_f} \|y(t) - \eta(t)\| = O(h^3),$$

where $h = \max_{1 \le n \le N} h_n$.

Proof: In connection to the DDE (4.2.7), for each $n = 1, \ldots, N-1$, consider the local ODE

$$\begin{cases} z'_{n+1}(t) = f\left(t, z_{n+1}(t), y(t - \tau(t, z_{n+1}(t)))\right), & t_n \le t \le t_{n+1}, \\ z_{n+1}(t_n) = y(t_n), \\ y(t) = \phi(t), \quad t \le t_0, \end{cases} \tag{4.2.8}$$

whose solution evidently is $z_{n+1} = y(t)$, i.e. the solution of (4.2.7). Moreover, consider the auxiliary problem

$$\begin{cases} w'_{n+1}(t) = f\left(t, w_{n+1}(t), \eta(t - \tau(t, w_{n+1}(t)))\right), & t_n \le t \le t_{n+1}, \\ z_{n+1}(t_n) = \eta(t_n), \\ \eta(t) = \phi(t), \quad t \le t_0, \end{cases} \tag{4.2.9}$$

where, for $s \le t_{n+1}$, $\eta(s)$ is the continuous numerical solution given by the DDE method itself. Then define $\mathbf{y}_n = [y(t_n), y(t_{n-1})]^T$ and $\boldsymbol{\eta}_n = [\eta(t_n), \eta(t_{n-1})]^T$. By Lemma 4.2.5 with $u(x) = y(x)$, $v(x) = \eta(x)$, $\mathbf{z}_n = \mathbf{y}_n$ and $\mathbf{w}_n = \boldsymbol{\eta}_n$, the numerical solutions $\zeta(t)$ and $\eta(t)$ of (4.2.8) and (4.2.9), respectively, satisfy the inequality

$$|||\mathbf{z}_{n+1} - \boldsymbol{\eta}_{n+1}||| \le (1 + h_{n+1}Q)|||\mathbf{y}_n - \boldsymbol{\eta}_n||| + h_{n+1}P \max_{t \le t_{n+1}} \|y(t) - \eta(t)\|, \tag{4.2.10}$$

where $\mathbf{z}_{n+1} = [\zeta(t_{n+1}), y(t_n)]^T$, and

$$\max_{t_n \le t \le t_{n+1}} \|\zeta(t) - \eta(t)\| \le (T + h_{n+1}S) \max_{n-2 \le s \le n} \|y(t_s) - \eta(t_s)\| + h_{n+1}R \max_{t \le t_{n+1}} \|y(t) - \eta(t)\|. \tag{4.2.11}$$

Now consider the inequality

$$|||\mathbf{y}_{n+1} - \boldsymbol{\eta}_{n+1}||| \le |||\mathbf{y}_{n+1} - \mathbf{z}_{n+1}||| + |||\mathbf{z}_{n+1} - \boldsymbol{\eta}_{n+1}|||. \tag{4.2.12}$$

Owing to hypothesis (h_1), the solution $z_{n+1}(t)$ of (4.2.8) is at least 4-times differentiable on $]t_n, t_{n+1}[$. Therefore, by hypothesis (h_2), (4.2.10) yields

$$|||\mathbf{y}_{n+1} - \boldsymbol{\eta}_{n+1}||| \leq M_1 h_{n+1}^4 + (1 + h_{n+1}Q)|||\mathbf{y}_n - \boldsymbol{\eta}_n||| + h_{n+1}P \max_{t \leq t_{n+1}} \|y(t) - \eta(t)\|.$$

Therefore, with

$$e_n = \max_{1 \leq i \leq n} |||\mathbf{y}_i - \boldsymbol{\eta}_i|||$$

and

$$E_n = \max_{t \leq t_{n+1}} \|y(t) - \eta(t)\|$$

for $n = 1, \ldots, N$, we obtain

$$e_{n+1} \leq M_1 h_{n+1}^4 + (1 + h_{n+1}Q)e_n + h_{n+1}PE_{n+1} \tag{4.2.13}$$

for $n = 1, \ldots, N-1$. Similarly, for the interpolant consider the inequality

$$\max_{t_n \leq t \leq t_{n+1}} \|y(t) - \eta(t)\| \leq \max_{t_n \leq t \leq t_{n+1}} \|y(t) - \zeta(t)\| + \max_{t_n \leq t \leq t_{n+1}} \|\zeta(t) - \eta(t)\|.$$

By the smoothness of $y(t)$, hypotheses (h_3) and (h_4), and (4.2.11), we get

$$\max_{t_n \leq t \leq t_{n+1}} \|y(t) - \eta(t)\| \leq M_2 h_{n+1}^3 + (T + h_{n+1}S) \max_{n-2 \leq s \leq n} \|y(t_s) - \eta(t_s)\|$$
$$+ h_{n+1}R \max_{t \leq t_{n+1}} \|y(t) - \eta(t)\|.$$

Thus, since there exists a constant $K > 0$ such that

$$\|y(t_s) - \eta(t_s)\| \leq K|||\mathbf{y}_s - \boldsymbol{\eta}_s|||, \quad \text{for all } s,$$

we obtain

$$\max_{t_n \leq t \leq t_{n+1}} \|y(t) - \eta(t)\| \leq M_2 h_{n+1}^3 + (T + h_{n+1}S)Ke_n + h_{n+1}RE_{n+1} \tag{4.2.14}$$

for $n = 1, \ldots, N-1$. With $L = \max\{M_1, M_2, P, Q, R, TK, SK\}$ the inequalities (4.2.13) and (4.2.14) yield

$$e_{n+1} \leq (1 + h_{n+1}L)e_n + h_{n+1}LE_{n+1} + h_{n+1}Lh^3 \tag{4.2.15}$$

and

$$\max_{t_n \le t \le t_{n+1}} \|y(t) - \eta(t)\| \le (1+h)Le_n + hLE_{n+1} + Lh^3 \tag{4.2.16}$$

for $n = 1, \ldots, N-1$. Since both e_n and E_n are monotone increasing, (4.2.16) implies

$$E_{n+1} \le (1+h)Le_n + hLE_{n+1} + Lh^3 \tag{4.2.17}$$

and hence, for $h < 1/L$, we have

$$E_{n+1} \le \frac{(1+h)L}{1-hL} e_n + \frac{L}{1-hL} h^3, \tag{4.2.18}$$

for $n = 1, \ldots, N-1$. Now assume, without any restriction, that $h \le \min\{1, 1/(2L)\}$, and define $\Lambda = 2(L + L^2)$. By substituting (4.2.18) into (4.2.15), we get

$$\begin{aligned} e_{n+1} &\le \left[1 + h_{n+1}\left(L + \frac{(1+h)L^2}{1-hL}\right)\right] e_n + h_{n+1}\left(L + \frac{L^2}{1-hL}\right) h^3 \\ &\le (1 + \Lambda h_{n+1}) e_n + h_{n+1}\Lambda h^3 \\ &\le e^{\Lambda h_{n+1}} e_n + h_{n+1}\Lambda h^3 \end{aligned} \tag{4.2.19}$$

for $n = 1, \ldots, N-1$. Now we have

$$\begin{aligned} e_n &\le e^{\Lambda(t_n - t_1)} e_1 + \left(\sum_{i=2}^{n} e^{\Lambda(t_n - t_i)} h_i\right) \Lambda h^3 \\ &\le e^{\Lambda(t_f - t_1)} e_1 + e^{\Lambda t_f}\left(\int_{t_1}^{t_f} e^{-\Lambda t} dt\right) \Lambda h^3, \\ &\le e^{\Lambda(t_f - t_1)} e_1 + \left(e^{\Lambda(t_f - t_1)} - 1\right) h^3, \end{aligned}$$

and hence, since hypotheses (h_2), (h_3) and (h_4) imply $e_1 = O(h^3)$, the proof is complete. ∎

Next, it is possible to adapt Theorem 4.2.6 to prove that DP45 together with a 3-point Hermite interpolant is convergent of global, discrete and uniform, (convergence) orders 3. Hence, the DP45-HB5DDE solver together with a 3-point Hermite interpolant is convergent of global, discrete and uniform, (convergence) orders 3. Again, note that variable-order does not lower the convergence order because the local error of any order p greater than 5 can only be smaller than the one associated with order 5.

Hence, under the given assumptions, the combined DP45-HB515DDE with an m-point Hermite interpolant is convergent of global, discrete and uniform, (convergence) orders 3. Moreover, it is easy to see that we cannot get a higher convergence order because the convergence considers the error over all integration steps and hence the smallest order is the dominant one.

If we want to disregard Assumption 1 (and the location of the discontinuities as a whole), then it is possible to prove the following theorem adapted from [2] which says that we can get an approximate solution accurate to *at least* order 2.

Theorem 4.2.7 *If equation (4.2.1) has a smooth solution apart from a finite number of discontinuities of order 1, then the ODE method DP45-HB515 along with the m-point Hermite interpolant furnishes an approximate solution of uniform global order at least 2 for any choice of the mesh.*

Chapter 5

Numerical Results for HB515DDE

5.1 Comparison Tests

HB515DDE was tested on seven problems and results were compared with other available DDE solvers.

Here is the list of the solvers used for the test problems:

1. HB515DDE: the solver described in this thesis.
2. SYSDEL (Karoui and Vaillancourt [11] and [12]): code based on the Runge–Kutta–Verner (5,6) formula pair. The results for SYSDEL were taken from the two cited articles.
3. DDVSS6 (Sharp and Smart): code based on a sixth order Runge–Kutta method from the Verner class. The results for DDVSS6 were taken from [7].
4. DDRK6N (Enright and Hu [7]): code based on a sixth order Runge–Kutta method. The results for DDRK6N were taken from the cited article.
5. DDVERK (Hayashi and Enright): code using the same discrete and continuous coefficients as DDVSS6. The results for DDVERK were taken from [7].

6. MATLAB's `ddesd` (Shampine [25]): code based on the classical fourth order Runge–Kutta method. The results for `ddesd` were obtained using Matlab.

The tables contain, for each asked tolerance, the results given by the solvers. Note that `MAXRE` is the maximum relative error over the mesh points and `NFE` is the number of function evaluations. Note that the speed of our solver is linear with respect to the NFE. When no result was available from a solver for a given tolerance, a "-" was left in the cell corresponding to that tolerance. Also, note that `ddesd` was sometimes compared with HB515DDE in a different graph than the other solvers because the former used too many function evaluations. Putting it in the same graph with the other ones would have changed the scale drastically and would have been inappropriate for comparison.

Problem 1: One constant delay (over $[0, 15]$):

$$\begin{cases} y'(t) = y(t-1), & t \in [0, 15], \\ y(t) = 1, & t \in [-1, 0], \end{cases}$$

with exact solution

$$y(t) = \sum_{i=0}^{\lfloor t \rfloor + 1} \frac{(t-i+1)^i}{i!}, \quad t \in [0, 15].$$

Results for this problem are given in Table 5.1 and graphed in Fig. 5.1.

Problem 2: One state dependent delay (over $[1, e^2]$):

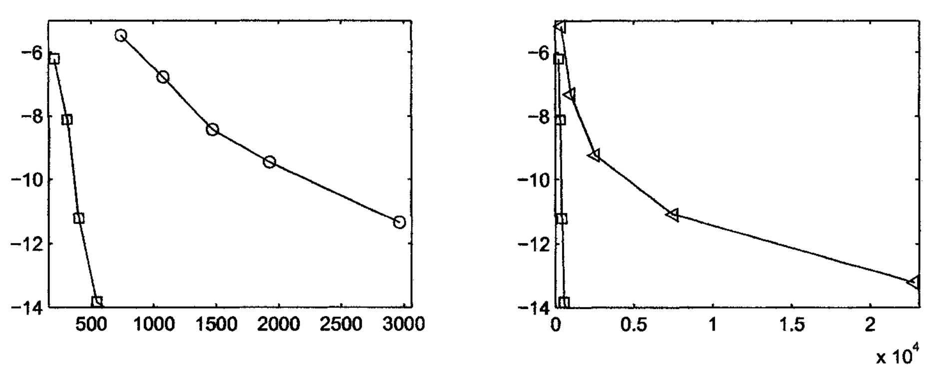
$$\begin{cases} y'(t) = \frac{1}{t} y(t) y(\ln(y(t))), & t \in [1, e^2], \\ y(t) = 1, & t \in [0, 1], \end{cases}$$

with exact solution

$$y(t) = \begin{cases} t, & t \in [1, e], \\ e^{t/e}, & t \in [e, e^2]. \end{cases}$$

TOL		HB515DDE	SYSDEL	ddesd
1E-04	NFE	205	741	372
	MAXRE	6.17E-07	3.51E-06	6.66E-06
1E-06	NFE	308	1079	961
	MAXRE	7.63E-09	1.64E-07	4.82E-08
1E-08	NFE	399	1471	2515
	MAXRE	6.22E-12	3.81E-09	5.91E-10
1E-10	NFE	544	1926	7521
	MAXRE	1.48E-14	3.51E-10	8.22E-12
1E-12	NFE	815	1962	22799
	MAXRE	2.83E-15	4.69E-12	6.17E-14

Table 5.1: Numerical results for Problem 1



HB515DDE □, SYSDEL ○ and ddesd ◁

Figure 5.1: NFE (horizontal axis) versus $\log_{10}$(MAXRE) (vertical axis) for Problem 1.

TOL		HB515DDE	SYSDEL	ddesd
1E-04	NFE	121	190	144
	MAXRE	9.61E-07	3.15E-06	4.97E-05
1E-06	NFE	186	232	317
	MAXRE	1.05E-08	1.47E-07	7.72E-07
1E-08	NFE	201	316	751
	MAXRE	8.60E-12	6.22E-09	1.03E-08
1E-10	NFE	207	596	2009
	MAXRE	1.02E-11	1.15E-10	1.32E-10
1E-12	NFE	372	1191	5913
	MAXRE	1.36E-13	3.15E-13	1.17E-12

Table 5.2: Numerical results for Problem 2

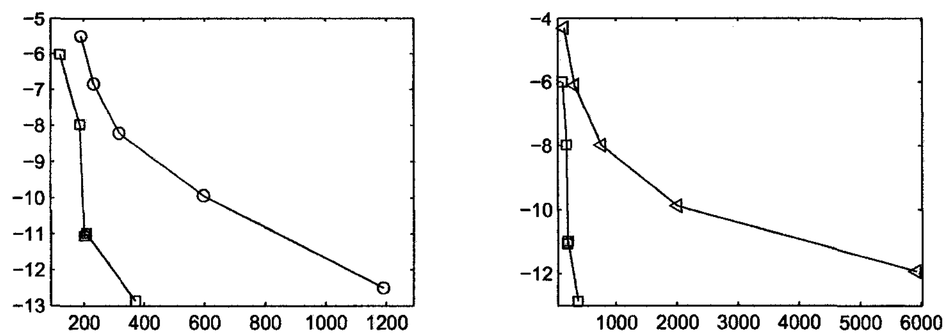
Results for this problem are given in Table 5.2 and graphed in Fig. 5.2.

Problem 3: Two constant delays (over $[0, 1]$):

$$\begin{aligned}
y_1'(t) &= y_5(t-1) + y_3(t-1), \\
y_2'(t) &= y_1(t-1) + y_2(t-0.5), \\
y_3'(t) &= y_3(t-1) + y_1(t-0.5), \qquad t \in [0,1], \\
y_4'(t) &= y_5(t-1)y_4(t-1), \\
y_5'(t) &= y_1(t-1),
\end{aligned}$$

with history

$$\begin{aligned}
y_1(t) &= e^{t+1}, \\
y_2(t) &= e^{t+0.5}, \\
y_3(t) &= \sin(t+1), \qquad t \in [-1,0]. \\
y_4(t) &= e^{t+1}, \\
y_5(t) &= e^{t+1},
\end{aligned}$$



HB515DDE □, SYSDEL ○ and **ddesd** ◁

Figure 5.2: NFE (horizontal axis) versus $\log_{10}$(MAXRE) (vertical axis) for Problem 2.

The exact solution is

$$
\begin{aligned}
&y_1(t) = e^t - \cos t + e, && t \in [0,1],\\
&y_2(t) = \begin{cases} 2e^t + e^{0.5} - 2, & t \in [0,0.5],\\ e^t + 2e^{t-0.5} + te^{0.5} - 2t + 1.5e^{0.5} - 3, & t \in [0.5,1],\end{cases}\\
&y_3(t) = \begin{cases} e^{t+0.5} - \cos t + 1 - e^{0.5} + \sin 1, & t \in [0,0.5],\\ -\cos t + e^{t-0.5} - \sin(t-0.5) + (t+0.5)e - e^{0.5} + \sin 1, & t \in [0.5,1],\end{cases}\\
&y_4(t) = 0.5e^{2t} - 0.5 + e, && t \in [0,1],\\
&y_5(t) = e^t + e - 1, && t \in [0,1].
\end{aligned}
$$

Results for this problem are given in Table 5.3 and graphed in Fig. 5.3.

Problem 4: One asymptotically vanishing delay (over $[2,30]$):

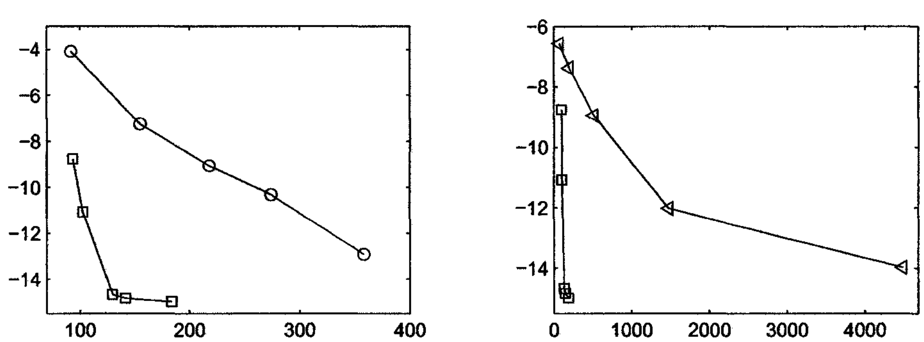
$$
\begin{cases} y'(t) = \dfrac{t^4 - 3}{(t^5 + t)\ln\left(t - t^{-3} + [t - t^{-3}]^{-3}\right)}\, y(t - t^{-3}), & t \in [2,30],\\ y(t) = \ln(t + t^{-3}), & t \in [1.5,2],\end{cases}
$$

with exact solution

$$y(t) = \ln(t + t^{-3}), \quad t \in [2,30].$$

TOL		HB515DDE	SYSDEL	ddesd
1E-04	NFE	94	92	67
	MAXRE	1.74E-09	8.43E-05	2.67E-07
1E-06	NFE	103	155	192
	MAXRE	8.30E-12	5.66E-08	4.24E-08
1E-08	NFE	130	218	510
	MAXRE	2.11E-15	8.90E-10	1.13E-09
1E-10	NFE	142	274	1480
	MAXRE	1.46E-15	4.72E-11	9.48E-13
1E-12	NFE	184	358	4483
	MAXRE	1.02E-15	1.19E-13	1.06E-14

Table 5.3: Numerical results for Problem 3



HB515DDE □, SYSDEL ○ and **ddesd** ◁

Figure 5.3: NFE (horizontal axis) versus $\log_{10}$(MAXRE) (vertical axis) for Problem 3.

TOL		HB515DDE M1	HB515DDE M2*	SYSDEL M1	ddesd
1E-04	NFE	134	158	-	251
	MAXRE	7.11E-04	7.11E-04	-	1.76E-06
1E-06	NFE	218	191	483	645
	MAXRE	6.57E-06	1.80E-06	1.20E-04	2.55E-08
1E-08	NFE	338	317	588	1845
	MAXRE	1.14E-07	8.22E-08	9.39E-07	4.00E-10
1E-10	NFE	488	452	1099	5611
	MAXRE	1.51E-09	1.15E-09	8.23E-09	4.72E-12
1E-12	NFE	704	698	2233	16561
	MAXRE	1.33E-11	1.03E-11	7.96E-11	5.25E-14

Table 5.4: Numerical results for Problem 4

Results for this problem are given in Tables 5.4 and 5.5 and graphed in Fig. 5.4 (M1: Karoui's first method for asymptotically vanishing delays, M2: second method with $t_\varepsilon = 8.00$ given manually, M2*: second method where the code had to choose an appropriate t_ε by running extra tests. When a good t_ε was not found, a "-" was left in the cell).

Problem 5: One asymptotically vanishing delay (over $[1.5, 50]$):

$$\begin{cases} y'(t) = \dfrac{y(t - t^{-2})}{(1+t^2)\arctan(t - t^{-2})}, & t \in [1.5, 50], \\ y(t) = \arctan t, & t \in [0, 1.5], \end{cases}$$

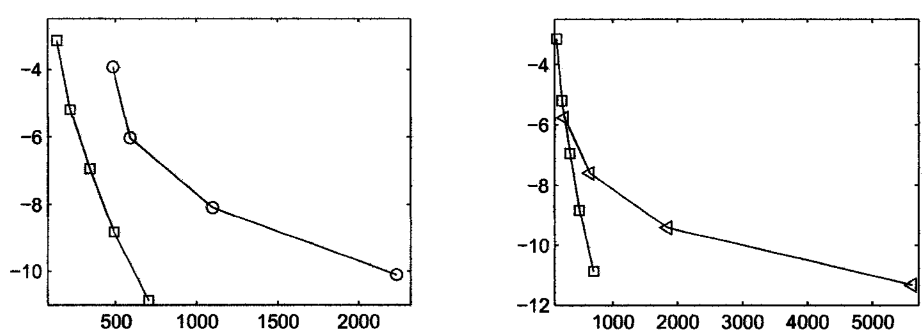
with exact solution

$$y(t) = \arctan t, \quad t \in [1.5, 50].$$

Results for this problem are given in Tables 5.6 and 5.7 and graphed in Fig. 5.5 (M1: Karoui's first method for asymptotically vanishing delays, M2: second method with $t_\varepsilon = 15.00$ given manually, M2*: second method where the code had to choose an

TOL		HB515DDE M2*	HB515DDE M2	SYSDEL M2
1E-04	NFE	158	119	-
	MAXRE	7.11E-04	3.26E-04	-
	t_ε	-	8.00	-
1E-06	NFE	191	197	329
	MAXRE	1.80E-06	3.11E-06	1.11E-04
	t_ε	4.96	8.00	8.00
1E-08	NFE	317	263	413
	MAXRE	8.22E-08	7.83E-08	9.10E-07
	t_ε	9.16	8.00	8.00
1E-10	NFE	452	374	553
	MAXRE	1.15E-09	8.99E-10	7.47E-09
	t_ε	11.78	8.00	8.00
1E-12	NFE	698	510	931
	MAXRE	1.03E-11	6.60E-10	6.93E-10
	t_ε	17.48	8.00	8.00

Table 5.5: Numerical results for Problem 4 - "Method 2"



HB515DDE M1 □, SYSDEL M1 ○ and **ddesd** ◁

Figure 5.4: NFE (horizontal axis) versus $\log_{10}$(MAXRE) (vertical axis) for Problem 4.

TOL		HB515DDE M1	HB515DDE M2*	SYSDEL M1	ddesd
1E-04	NFE	98	116	-	263
	MAXRE	9.49E-05	5.85E-05	-	3.22E-06
1E-06	NFE	167	197	770	657
	MAXRE	3.57E-06	2.98E-06	1.08E-09	5.19E-08
1E-08	NFE	260	254	833	1893
	MAXRE	3.02E-08	1.37E-08	7.36E-10	6.04E-10
1E-10	NFE	344	359	1680	5563
	MAXRE	4.88E-10	3.98E-10	8.51E-12	6.42E-12
1E-12	NFE	491	524	3528	16573
	MAXRE	5.72E-12	6.83E-12	4.50E-14	5.56E-14

Table 5.6: Numerical results for Problem 5

appropriate t_ϵ by running extra tests).

Problem 6: One vanishing delay at $t = 1$ (over $[0.1, 10]$):

$$\begin{cases} y'(t) = 1 - y\left(e^{1-1/t}\right), & t \in [0.1, 10], \\ y(t) = \ln t, & t \in [0, 0.1], \end{cases}$$

with exact solution

$$y(t) = \ln t, \quad t \in [0.1, 10].$$

Results for this problem are given in Table 5.8 and graphed in Fig. 5.6.

Problem 7: One vanishing delay at $t = 1$ (over $[0.1, 5]$):

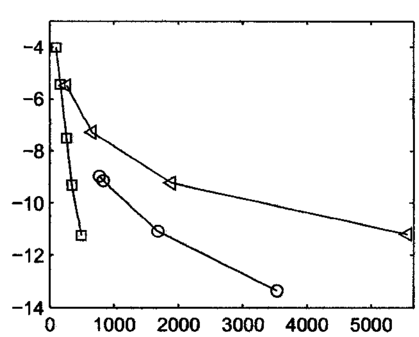
$$\begin{cases} y_1'(t) = y_2(t), \quad y_2'(t) = -y_2\left(e^{1-y_2(t)}\right) y_2^2(t) e^{1-y_2(t)}, & t \in [0.1, 5], \\ y_1(t) = \ln t, \quad y_2(t) = 1/t, & t \in [0, 0.1], \end{cases}$$

with exact solution

$$y_1(t) = \ln t, \quad y_2(t) = 1/t, \quad t \in [0.1, 5].$$

TOL		HB515DDE M2*	HB515DDE M2	SYSDEL M2
1E-04	NFE	116	95	-
	MAXRE	5.85E-05	7.67E-05	-
	t_ε	11.08	15.00	-
1E-06	NFE	197	167	525
	MAXRE	2.98E-06	2.60E-06	1.14E-09
	t_ε	18.84	15.00	15.00
1E-08	NFE	254	251	553
	MAXRE	1.37E-08	2.30E-08	7.75E-10
	t_ε	8.31	15.00	15.00
1E-10	NFE	359	326	805
	MAXRE	3.98E-10	3.77E-10	2.87E-11
	t_ε	12.99	15.00	15.00
1E-12	NFE	524	482	1344
	MAXRE	6.83E-12	2.40E-11	2.05E-11
	t_ε	22.02	15.00	15.00

Table 5.7: Numerical results for Problem 5 - "Method 2"

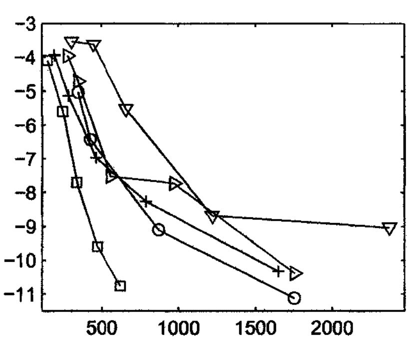


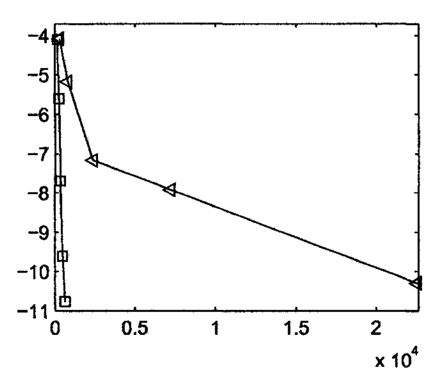
HB515DDE M1 □, SYSDEL M1 ○ and **ddesd** ◁

Figure 5.5: NFE (horizontal axis) versus $\log_{10}$(MAXRE) (vertical axis) for Problem 5.

TOL		HB515DDE	SYSDEL	DDVSS6	DDRK6N	DDVERK	ddesd
1E-04	NFE	146	-	193	276	302	285
	MAXRE	7.90E-05	-	1.13E-04	1.09E-04	2.96E-04	8.33E-05
1E-06	NFE	245	343	282	344	446	793
	MAXRE	2.49E-06	9.48E-06	7.44E-06	1.95E-05	2.40E-04	2.42E-06
1E-08	NFE	335	420	458	557	657	2359
	MAXRE	1.98E-08	3.77E-07	1.07E-07	2.99E-08	3.02E-06	6.77E-08
1E-10	NFE	470	868	789	971	1219	7229
	MAXRE	2.47E-10	7.98E-10	5.38E-09	1.87E-08	2.06E-09	1.18E-08
1E-12	NFE	620	1750	1648	1755	2369	22527
	MAXRE	1.72E-11	7.55E-12	4.56E-11	4.06E-11	9.43E-10	5.13E-11

Table 5.8: Numerical results for Problem 6





HB515DDE □, SYSDEL ○, DDVSS6 +, DDRK6N ▷, DDVERK ▽ and ddesd ◁

Figure 5.6: NFE (horizontal axis) versus $\log_{10}$(MAXRE) (vertical axis) for Problem 6.

TOL		HB515DDE	SYSDEL	DDVSS6	DDRK6N	DDVERK	ddesd
1E-04	NFE	185	-	251	327	286	353
	MAXRE	3.65E-04	-	4.59E-05	3.10E-05	2.32E-05	1.69E-05
1E-06	NFE	287	553	310	548	499	951
	MAXRE	3.11E-06	1.85E-06	1.36E-06	5.98E-07	1.82E-06	2.08E-07
1E-08	NFE	347	959	517	1035	974	2833
	MAXRE	4.54E-08	5.05E-09	2.11E-08	1.65E-08	6.85E-08	2.44E-09
1E-10	NFE	455	1946	934	2036	1750	8633
	MAXRE	1.12E-10	2.85E-11	2.80E-09	4.33E-10	4.89E-09	2.34E-11
1E-12	NFE	638	4214	2077	4160	-	26831
	MAXRE	7.90E-12	1.11E-13	4.56E-11	2.75E-11	-	3.24E-13

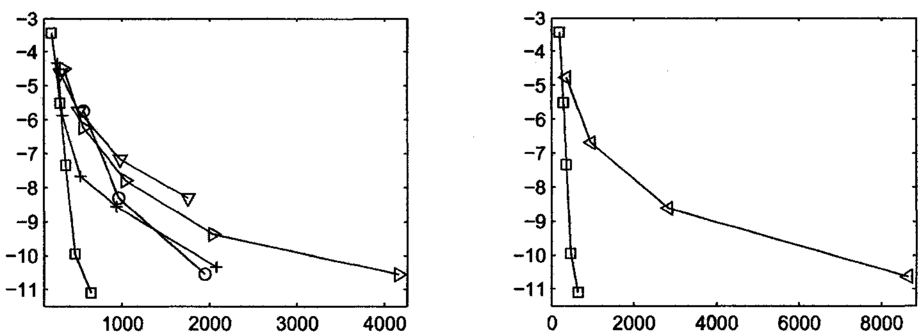
Table 5.9: Numerical results for Problem 7

Results for this problem are given in Table 5.9 and graphed in Fig. 5.7.

5.2 Interpretation of the Results

Firstly, we can see from the above results that for the same accuracy, the solver HB515DDE uses less function evaluations than the other solvers most of the time. This could be explained by the special structure of the solver. Indeed, most of the other solvers cited above use a Runge–Kutta structure but HB515DDE makes use of the multistep advantages which include reducing the `NFE`. Also, variable order gives HB515DDE a big advantage over constant order solvers because it can adapt itself to the different environment it encounters. Indeed, lower order is preferable when y is smooth because it uses larger stepsizes and thus, advances faster. On the other side, higher order gives better precision and is favored when there is turbulence in y.

Secondly, we note that, for asymptotically vanishing delays, Method 2* is less



HB515DDE □, SYSDEL ∘, DDVSS6 +, DDRK6N ▷, DDVERK ▽ and ddesd ◁

Figure 5.7: NFE (horizontal axis) versus $\log_{10}$(MAXRE) (vertical axis) for Problem 7.

effective than Method 2. This is normal because the former is less dependent on the user and hence receives less input. Indeed, the solver wastes some function evaluations to find a satisfactory t_ϵ which should not be guessed by the user. Rather, the values 8 and 15 given by Karoui for problems 4 and 5, respectively, could have been chosen after knowledge of the exact solution. Hence, we do not recommend standard users to use Method 2 but we provide it as an option for advanced users. Inspite of all this, HB515DDE's Method 2* gives better results than SYSDEL's Method 2.

We mention that we do not put much emphasis on the results of our Method 2* because they did not significantly improve the results given by Method 1 for the above problems. We note that all DDE functions above are very simple. This could explain the small difference between solving the problem as a DDE or an ODE. We think that long simulation problems with delay functions that are costly to evaluate would make Method 2* a lot better than Method 1.

Chapter 6

Conclusion

This thesis describes the main ideas behind the adaptation of the ODE solver HB(5-15)3 to solve DDEs. This nontrivial transformation makes sure that the ODE solver is able to deal with the special properties that delays introduce into differential equations. This includes adding a Hermite interpolant to calculate delayed values, choosing the right way to locate discontinuities and doing all that precisely and efficiently.

The DDE solver HB515DDE was tested along with other known codes on seven problems. The obtained results show, for each test problem, that HB515DDE behaves better than the other methods in the following way: for the same number of function evaluations, HB515DDE gave the most precise maximum relative error most of the time. These results could be explained by the fact that HB515DDE uses both multistep and Runge–Kutta structures and that it is a variable order code.

6.1 Directions for Future Work

Ideas for future work include testing all the above solvers on problems for which the evaluation of $y' = f$ is costly. There, we think that HB515DDE should be the fastest because the above results show that it uses fewer function evaluations than the other methods most of the time. Also, we could add to HB515DDE the option of solving initial value DDEs, NDDEs (or even delay differential algebraic equations (DDAEs)) and some code that will take care of secondary discontinuities.

Moreover, Enright and Hu's idea in [7] of developing special interpolants to deal with small delays is attractive and could be adapted for DP45 used by HB515DDE to avoid extrapolation when the stepsize is big. Finally, we could try to adapt the HBT methods to solve DDEs because HBT methods are more precise than HB methods. Also, they are one step methods and hence are less affected by turbulence in the backsteps. Still, it must be taken into account that they might have more trouble in dealing with low order discontinuities because they use very high order derivatives of y in their integration.

Bibliography

[1] C. T. H. Baker, C. A. H. Paul, and D. R. Willé. Issues in the Numerical Solution of Evolutionary Delay Differential Equations. Technical Report 248, Manchester, England, 1994.

[2] A. Bellen and M. Zennaro. *Numerical Methods for Delay Differential Equations.* Oxford University Press, New York, NY, USA, 2003.

[3] M. Calvo, J. I. Montijano, and L. Rández. On the solution of discontinuous IVPs by adaptive Runge–Kutta codes. *Numerical Algorithms*, 33:163–182, 2003.

[4] J. R. Dormand and P. J. Prince. A family of embedded Runge–Kutta formulae. *J. of Computational and Applied Mathematics*, 6(1):19–26, 1980.

[5] W. H. Enright and al. Effective solution of discontinuous IVPs using a Runge–Kutta formula pair with interpolants. *Applied Mathematics and Computation*, 27:313–335, 1988.

[6] W. H. Enright and H. Hayashi. A delay differential equation solver based on a continuous Runge–Kutta method with defect control. *Numerical Algorithms*, 16:349–364, 1997.

[7] W. H. Enright and M. Hu. Interpolating Runge–Kutta methods for vanishing delay differential equations. *Computing*, 55:223–236, 1995.

[8] G. H. Golub and C. F. Van Loan. *Matrix Computations (3rd ed.).* Johns Hopkins University Press, Baltimore, MD, USA, 1996.

[9] E. Hairer, S.P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems.* Springer-Verlag, Berlin, Germany, 1987.

[10] H. Hayashi. *Numerical Solution of Retarded and Neutral Delay Differential Equations Using Continuous Runge–Kutta Methods.* PhD thesis, University of Toronto, 1996.

[11] A. Karoui and R. Vaillancourt. Computer solutions of state-dependent delay differential equations. *Comput. Math. Applic.*, 27(4):37–51, 1994.

[12] A. Karoui and R. Vaillancourt. A numerical method for vanishing-lag delay differential equations. *Applied Numerical Mathematics*, 17:383–395, 1995.

[13] T. Nguyen-Ba, V. Bozic, E. Kengne, and R. Vaillancourt. One-step 9-stage Hermite–Birkhoff–Taylor ODE solver of order 10. *J. Appl. Math. and Computing.* In press.

[14] T. Nguyen-Ba, V. Bozic, E. Kengne, and R. Vaillancourt. One-step 4-stage Hermite–Birkhoff–Taylor ODE solver of order 14. *Scientific Proceedings of Riga Technical University*, 33:26–41, 2007.

[15] T. Nguyen-Ba, V. Bozic, A. Przybylo, and R. Vaillancourt. One-step 9-stage Hermite–Birkhoff–Taylor ODE solver of order 11. *University Scientific J., Telecommunications and Electronics Series, University of Technology and Life Sciences (UTP), Bydgoszcz, Poland*, 11:33–52, 2008.

[16] T. Nguyen-Ba, V. Bozic, and R. Vaillancourt. One-step 7-stage Hermite–Birkhoff–Taylor ODE solver of order 13. *International J. Pure Appl. Math.*, 43(4):569–592, 2008.

[17] T. Nguyen-Ba, H. Hao, H. Yagoub, and R. Vaillancourt. One-step 5-stage Hermite–Birkhoff–Taylor ODE solver of order 12. *Appl. Math. and Computation.* In press.

[18] T. Nguyen-Ba, P. W. Sharp, and R. Vaillancourt. Hermite–Birkhoff–Obrechkoff 4-stage 4-step ODE solver of order 14 with quantized stepsize. *J. of Computational and App. Math.*, 222(2):608–621, 2008.

[19] T. Nguyen-Ba, P. W. Sharp, H. Yagoub, and R. Vaillancourt. Hermite–Birkhoff–Obrechkoff 3-stage 4-step ODE solver of order 14 with quantized stepsize. *Can. Appl. Math. Quarterly*, 15(2):181–201, 2007.

[20] T. Nguyen-Ba and R. Vaillancourt. Hermite–Birkhoff–Obrechkoff 3-stage 6-step ODE solver of order 14. *Can. Appl. Math. Quarterly*, 13(2):151–181, 2005.

[21] T. Nguyen-Ba, H. Yagoub, S. J. Desjardins, and R. Vaillancourt. Variable-step variable-order 4-stage Hermite–Birkhoff–Obrechkoff ODE solver of order 5 to 14. *Scientific Proceedings of Riga Technical University*, 29:53–80, 2006.

[22] T. Nguyen-Ba, H. Yagoub, Y. Li, and R. Vaillancourt. Variable-step variable-order 3-stage Hermite–Birkhoff ODE solver of order 5 to 15. *Can. Appl. Math. Quarterly*, 14(1):43–69, 2006.

[23] T. Nguyen-Ba, H. Yagoub, Y. Zhang, and R. Vaillancourt. Variable-step variable-order 3-stage Hermite–Birkhoff–Obrechkoff ODE solver of order 4 to 14. *Can. Appl. Math. Quarterly*, 14(4):413–437, 2006.

[24] T. Nguyen-Ba, H. Yagoub, Y. Zhang, and R. Vaillancourt. Variable-step variable-order 2-stage Hermite–Birkhoff–Obrechkoff ODE solver of order 3 to 14. *Scientific Proceedings of Riga Technical University*, 37(50), 2008. In press.

[25] L. F. Shampine. Solving ODEs and DDEs with Residual Control. Technical report, Amsterdam, The Netherlands.

[26] L. F. Shampine and M. K. Gordon. *Computer Solution of Ordinary Differential Equations: The Initial Value Problem.* W. H. Freeman and Company, San Francisco, 1975.

[27] D. R. Willé and C. T. H. Baker. The tracking of derivative discontinuities in systems of delay differential equations. *Appl. Num. Math.*, 9:209–222, 1992.

[28] H. Yagoub, T. Giordano, T. Nguyen-Ba, and R. Vaillancourt. Convergence of the variable-step variable-order 3-stage Hermite–Birkhoff ODE/DDE solver of order 5 to 15. *J. Comput. Appl. Math.* Submitted on 2009.03.10.

[29] H. Yagoub, T. Nguyen-Ba, and R. Vaillancourt. Variable-step variable-order 3-stage Hermite–Birkhoff–Obrechkoff DDE solver of order 4 to 14. *Appl. Math. Computation.* Submitted on 2008.11.14. Revised on 2009.01.24.

[30] H. Yagoub, T. Nguyen-Ba, and R. Vaillancourt. Variable-step variable-order 3-stage Hermite–Birkhoff NDDE solver of order 5 to 15. *Appl. Num. Math.* Submitted on 2009.04.09.

[31] H. Yagoub, T. Nguyen-Ba, and R. Vaillancourt. Variable-step 7-stage Hermite–Birkhoff–Taylor DDE solver of order 8. *Scientific Proceedings of Riga Technical University*, 37(50), 2008. In press.