Global Error Evaluation Strategies in Multistep Methods Applied to Ordinary Differential Equations and Index 1 Differential Algebraic Systems

Andriamihaja Ramanantoanina

School of Computational and Applied Mathematics.

University of the Witwatersrand

A dissertation submitted to the faculty of Science
in fulfillment of the requirements for the degree of Master of Science

Declaration

I declare that this dissertation is my own, unaided work. It is being submitted for the Degree of Master of Science to the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination to any other University.

____

(Signature)

____

(Date)

Abstract

Differential Equations (DEs) are among the most widely used mathematical tools in different area of sciences. Solving DEs, either analytically or numerically, has become a centre of interest for many mathematicians and a large variety of methods are nowadays available to solve DEs numerically.

When solving a mathematical problem numerically, evaluating the error is of high importance in practice. Most of the methods already available for solving DEs are implemented with a mechanism to perform a local error control.

However, in the real realm, it is common to require the numerical solution to approximate the exact solution with accuracy to a certain number of decimal places or significant figures. To satisfy this condition, we require the global error to be bounded by a specifically determined tolerance. In this case, a local error control is not longer efficient. On one hand, controlling the local error only cannot ensure that the required accuracy will be achieved. On the other hand, the use of such approach requires the user to do some preliminary studies on the problem, and have deep understanding of the method. Thus, we need a mechanism to control the global error in order to compute the numerical solution for a user-supplied accuracy requirement in automatic mode.

The global error estimate calculated in the course of such a control can also be applied to improve the numerical solution obtained. It is straight forward since, if the error estimate is found with sufficiently high accuracy, we can just add it to the numerical solution to get a better approximation to the exact value.

Thus, accurate evaluation of the the global error is crucial for the purpose mentioned above.

Several techniques are already developed to compute the global error of the numerical solution. The most common algorithms include the Richardson extrapolation, Zadunaisky's technique, Solving for the correction, and Using two different methods.

These methods use two integrations to evaluate the global error, and the provided error estimate is valid if the global error admits an expansion in powers of the step size. Another approach, known as solving the linearised discrete variational equation, can also be used. This last differs from the others by the use of a truncated Taylor expansion of the defect of the method to estimate the global error; and solving the problem and estimating the error is roughly the same as one step of the underlying method.

In this research, we will investigate numerically and compare the efficiency of different techniques for global error evaluation applied to multistep methods for solving ordinary differential equations (ODEs) and differential algebraic equations (DAEs). We will first study the global error evaluation techniques in multistep formulas for solving ODEs on uniform grids. In the case of nonuniform grids, both multistep methods with variable coefficients and interpolation-type multistep methods will be considered. Then, we will extend our study to multistep methods for solving DAEs.

Theoretical background will accompany numerical works. The accuracy and reliability of the global error evaluation strategies will be discussed and compared for different types of multistep methods for solving ODEs and DAEs. We will analyse the efficiency in terms of accuracy obtained and CPU time spent. For that, a series of numerical experiments is conducted on a set of test problems with known solutions.

Ho an'i Neny, nodimandry teo ampamitako ity asa ity, sy ho an'i Dada mitozo hatrany hahatafita anay amin'ny lafiny rehetra : fa ny vavaka, ny fitiavana ary ny anatranatrareo no nahatoy izao ahy.

Ho anareo zokiko sy zandriko, fa sarobidy amiko ianareo sy ny nataonareo.

Acknowledgments

First of all, I would like to express my gratitude to my supervisor Dr G. Yu. Kulikov for agreeing to supervise me in this work and for his understanding. I am also grateful to Dr S. K. Shindin for the knowledge of C++ that he freely shared with me. I would not have accomplished this dissertation without their patient care.

I wish also to thank everyone at the School of Computational and Applied Mathematics (CAM). Special thanks to the head of school, Professor D. Sherwell, who helped me to integrate the department and, most importantly, providing his expertise in the dissertation and helping me out on corrections. I have been very fortunate to meet the staff members and experience their kindness. I will not also forget all the time shared with my officemates and fellow postgraduate students. They provided a joyful environment and helped me to sort out the Linux and LaTeX difficulties.

I express my warmest appreciation to my family for their continuous and unlimited encouragement. I wish also to thank all my friends, I specially mention the AIMS2004@Wits group and Sanda, from the bottom of my heart. I am very grateful to them for all their wonderful support throughout the realization of this work.

Finally, in terms of financial assistance, the scholarships from the African Institute for Mathematical Sciences (AIMS) and the School of Computation and Applied Mathematics (CAM), and the Postgraduate Merit Award from the University of the Witwatersrand have fully supported my stay in Johannesburg and I wish to recognise their importance in this work.

Contents

1 Introduction **1**

1.1 Ordinary Differential Equations 2

1.1.1 Samples of Ordinary Differential Equations 4

1.2 Index 1 Semi-explicit Differential Algebraic Equations 7

1.2.1 Samples of Index-1 Differential Algebraic Equations 8

2 Multistep Methods and Error Evaluations **11**

2.1 Multistep Methods 11

2.1.1 Formulation of Multistep Methods 11

2.1.2 Consistency - Stability - Convergence 12

2.1.3 Local Error and Order of Convergence 14

2.2 Global Error Expansion 15

2.3 Stability of Multistep Methods 15

2.4 Variable Stepsize Multistep Methods 17

2.4.1 Variable Stepsize Multistep Methods with Variable Coefficients 17

2.4.2 Interpolation Type Multistep Methods 19

2.5 Multistep Methods for Semi-Explicit Index 1 DAE 21

2.6 Global Error Evaluation Techniques 22

2.6.1 Richardson Extrapolation 23
2.6.2 Using Two Different Methods 24
2.6.3 Zadunaisky Technique 25
2.6.4 Solving for the Correction 26
2.6.5 Solving the Linearised Discrete Variational Equation 28

3 Numerical Results for ODEs 30

3.1 Numerical Result for Adams Methods 30
3.1.1 Implementation on Uniform Grids 30
3.1.2 Implementation On Non-uniform Grids 38
3.2 Numerical Result for BDF formulae 44
3.2.1 Implementation on Uniform Grids 44
3.2.2 Implementation on non-uniform Grids 51
3.3 Numerical Results for Nyström Methods 58
3.3.1 Implementation on Uniform Grids 58
3.3.2 Implementation on Non-uniform Grids 66

4 Numerical Results for Index 1 DAEs 74

4.1 Numerical Results for Adams methods 74
4.1.1 Implementation on the Uniform Grids 74
4.1.2 Implementation on the Non-uniform Grids 75
4.2 Numerical Results for BDF formulae 78
4.2.1 Implementation on the Uniform Grids 78
4.2.2 Implementation on the Non-uniform Grids 83
4.3 Numerical Results for Nystöm methods 85

4.3.1 Implementation on the Non-uniform Grids 85

5 Conclusion **91**

Bibliography **92**

Chapter 1

Introduction

Mathematicians, specially Numerical Analysts, started to work on numerical methods for DEs since the work of Euler in 1810. Different methods have been developed and improved to provide good approximations to the solutions when there are any. Most importantly, to ensure the reliability of mathematical models, numerical methods should be accompanied by a procedure to monitor any drastic changes in the error.

Almost all numerical methods for solving DEs developed so far use a stepsize selection based on the local error control to obtain the numerical solution. However, this technique intended to keep the local error less than or equal to a prescribed tolerance has some drawbacks. Considering the principal term of the local error as its estimate does not guarantee that the local error itself will be small, unless the grid has sufficiently small diameter. Moreover, in one step of the integration, the local error does not remember the error introduced in all the previous steps. Thus, keeping the local error relatively small does not automatically produce a reasonably small global error which is more important in practice.

Numerical analysts started to work on a more indispensable feature, which is the global error evaluation, in early 1970. Several methods have been developed. A good survey of such techniques can be found in [20]. Methods presented in [20] are not only aimed to estimate the global error in numerical ODEs, they can also be applied for other problems, such as DAEs and PDEs. For numerical ODEs and DAEs, an additional approach termed as solving the linearised discrete variational equation (SLDVE) was introduced in [10] and developed in detail in [12] and [14].

In this dissertation, we focus on the behaviour of global error evaluation strategies when applied to multistep methods for ODEs and semi-explicit index 1 DAEs. The algorithms include:

1. Richardson extrapolation,
2. Using two different methods,
3. Zadunaisky's technique,
4. Solving for the correction,
5. Solving the linearised discrete variational equation.

We aim to compare the methods implemented in multistep formulas including both weakly and strongly stable ones. Their performance will be investigated for uniform and non-uniform grids and we will use the same set of test problems with exact solution graphs for all methods and grids. A similar comparison was presented by Aid and Levacher in [1] for ODEs.

We organise the remainder of this dissertation as follow: in this introductory part, ODEs and index 1 DAEs are presented with exact solutions graph. In the next chapter, we recall basic concepts of multistep methods, and outline briefly the global error expansion theory. Notions of weak and strong stability are introduced and different implementations of multistep methods with variable stepsize are presented. We give also a survey of global error evaluation strategies. In the third chapter, we conduct numerical experiments and discuss numerical result obtained for ODEs. In the fourth chapter we deal with the numerical data for index 1 DAE. We summarise the results and draw a conclusion in the last chapter.

1.1 Ordinary Differential Equations

The ODEs that we are interested in have the form

$$x'(t) = f(t, x(t)), \tag{1.1}$$

where t is called the *independent variable*, and $x(t)$, known as the *dependent variable*, is the solution. If x is an N dimensional vector valued function, the domain and the range of f and x are given by

$$x : D \to \mathbb{R}^N,$$
$$f : [t_0,\ T] \times D \to \mathbb{R}^N$$

where $[t_0,\ T] \subset \mathbb{R}$ and $D \subset \mathbb{R}^N$.

DEs are usually broken into two classes according to the additional conditions provided to solve them. If such conditions are given at several values of t, the problem is called a *boundary value problem* (BVP); and when the conditions are provided at a certain value of t, the problem is called *initial value problem* (IVP). In this work, we deal with IVP, i.e with a problem of the form

$$x'(t) = f(t, x(t)),\ \ t \in [t_0, T], \tag{1.2a}$$

$$x(t_0) = x_0. \tag{1.2b}$$

For practical reasons in scientific modelling, it is important to study whether an ODE admits solutions, and if it does whether it is unique. For this purpose, we recall the definition of a *Lipschitz condition*.

Definition 1.1. *[4] The function $f : [t_0, T] \times \mathbb{R}^N \to \mathbb{R}^N$ is said to satisfy a "Lipschitz condition" in its second variable if there exists a constant L such that for any $t \in [t_0, T]$ and $y, z \in \mathbb{R}^N$*

$$||f(t,y) - f(t,z)|| \leq L\,||y - z||\,.$$

L is known as the "Lipschitz constant".

The following theorem, proved in [4], ensures the existence and uniqueness of the solution to IVP (1.2).

Theorem 1.2. *[4] Consider an IVP (1.2) where $f : [t_0, T] \times \mathbb{R}^N \to \mathbb{R}^N$ is continuous in its first variable and satisfies the Lipschitz condition in its second variable. Then there exists a unique solution to this problem.*

An equation of the form (1.1) is said to be *non-autonomous* and represents the natural form of many problems which arise in mathematical modelling tasks. However, it is more practical, specially when dealing with numerical methods, to use the following representation the problem.

$$x'(t) = f(x(t)). \tag{1.3}$$

The latter is termed an *autonomous* equation. Any non-autonomous ODE can be written in an equivalent autonomous form by introducing a new independent variable that is always equal to t. This prototype will be used when we discuss numerical methods for ODEs.

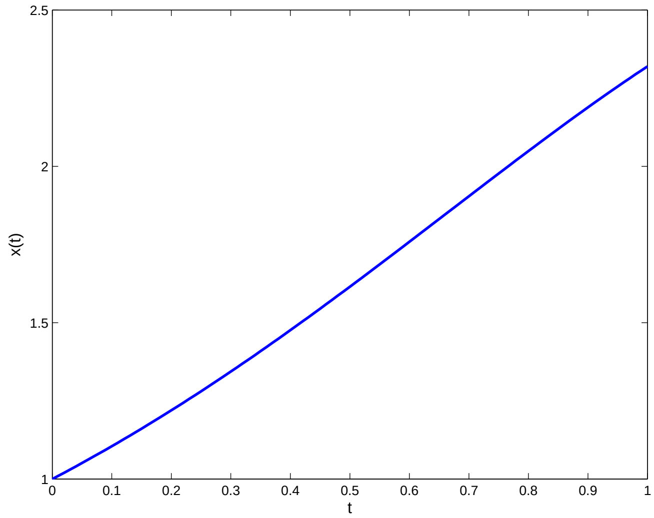

Figure 1.1: Graph of the exact solution to Problem ODE1

1.1.1 Samples of Ordinary Differential Equations

Problem ODE1: An oscillatory problem

We consider the ODE described by

$$x'(t) = x(t)\cos(t), \tag{1.4}$$

with the initial condition $x(0) = 1$ for $t \in [0,1]$. The exact solution to this problem is

$$x(t) = e^{\sin(x(t))}.$$

and is plotted in Figure 1.1

Problem ODE2: A non-linear stable ODE

The following equations represent a non-linear system of ODEs

$$\begin{aligned} x_1'(t) &= -x_3(t)x_1(t) + x_2(t), \\ x_2'(t) &= -x_1(t) - x_3(t)x_2(t), \\ x_3'(t) &= x_4(t), \;\; x_4'(t) = -x_3(t) \end{aligned} \tag{1.5}$$

for $t \in [0,1]$ and with the initial condition $x(0) = (1,1,1,1)^T$.

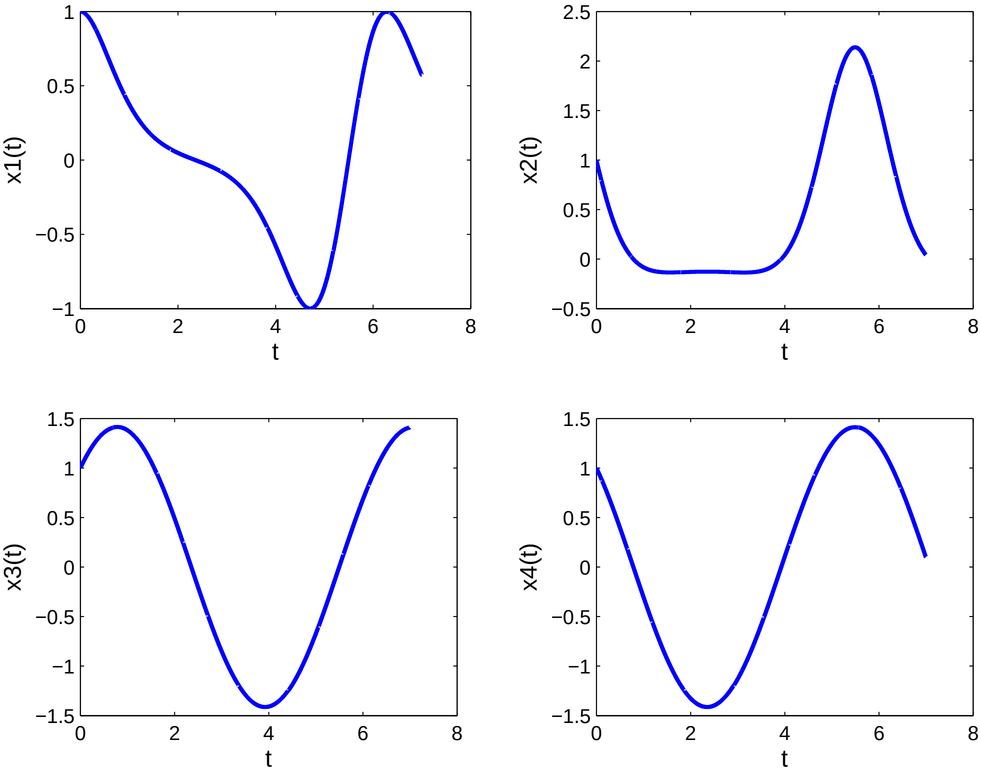

Figure 1.2: Graph of the exact solution to Problem ODE2

The exact solution to this problem is:

$$x_1(t) = (\cos t + \sin t)e^{-1+\cos t - \sin t},$$
$$x_2(t) = (\cos t - \sin t)e^{-1+\cos t - \sin t},$$
$$x_3(t) = \cos t + \sin t,$$
$$x_4(t) = \cos t - \sin t.$$

The behaviour of the exact solution is shown in Figure 1.2

Problem ODE3: A simple ODE. It is given by

$$\begin{aligned} x_1'(t) &= 2tx_2^2(t)x_4(t), \\ x_2'(t) &= 10te^{5(x_3(t)-1)x_4(t)}, \\ x_3'(t) &= 2tx_4(t), \\ x_4'(t) &= -2t\ln(x_1(t)) \end{aligned} \tag{1.6}$$

with the initial condition $x(0) = (1,1,1,1)^T$ and for $t \in [0,1]$. The exact solution to this

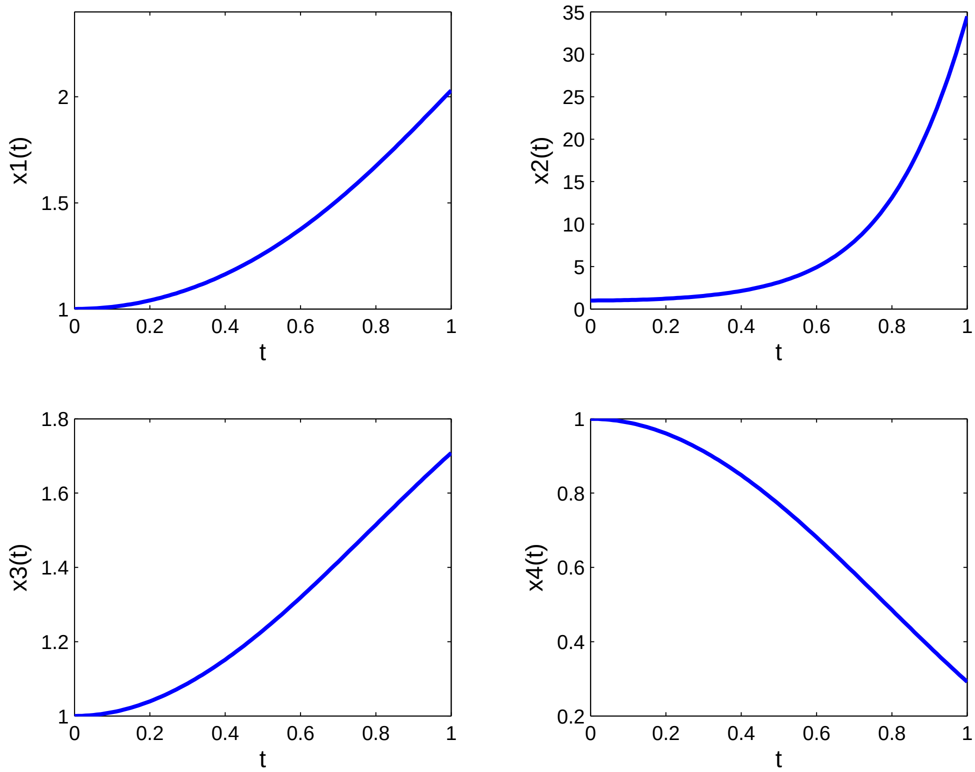

Figure 1.3: Exact Solution of Problem ODE3

problem is

$$x_1(t) = e^{\sin(t^2)},$$
$$x_2(t) = e^{5\sin(t^2)},$$
$$x_3(t) = \sin(t^2) + 1,$$
$$x_4(t) = \cos(t^2),$$

and shown graphically in Figure 1.3

Problem ODE4: A Stiff ODE. As a sample of stiff ODE we take the following problem

$$x'(t) = \lambda(x(t) - \sin(\mu t)) + \mu\cos(\mu t) \tag{1.7}$$

with the initial condition $x(0) = 1$ when $t \in [0,1]$ The exact solution to problem (1.7) is given by

$$x(t) = \sin(\mu t) + e^{\lambda t}.$$

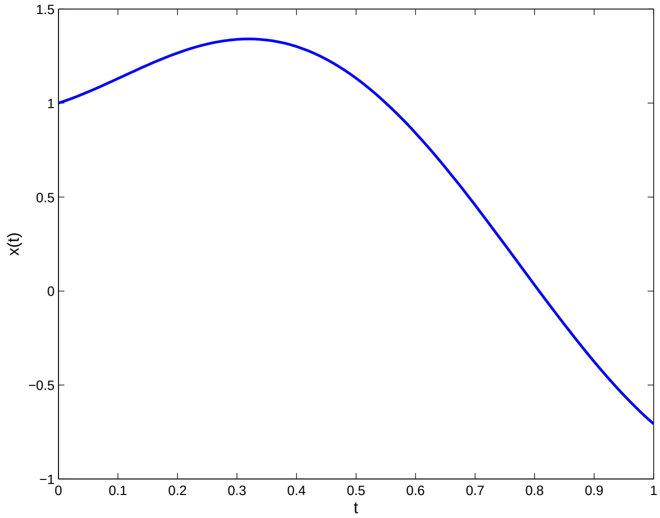

Figure 1.4: Graph of the exact solution to Problem ODE4

In this work, λ and μ take the values -3 and 4 respectively. The behaviour of the corresponding exact solution is shown in Figure 1.4.

1.2 Index 1 Semi-explicit Differential Algebraic Equations

Equation (1.1) represents the explicit form of an ODE. A general ODE can have the form

$$F(t,x(t),x'(t)) = 0. \tag{1.8}$$

Equation (1.8) is known as the implicit form of an ODE. When it is possible to solve this equation for x' (as a function of t and x), we will get the prototype (1.1).

Another form of DEs, known as Semi-Explicit Differential Algebraic Equations can also arise from equation (1.8). It is given by the system of differential and algebraic equations

$$\begin{aligned} x'(t) &= f(t,x(t),y(t), \\ 0 &= g(t,x(t),y(t)) \end{aligned}$$

or, equivalently,

$$x'(t) = f(t,x(t),y(t)), \tag{1.9a}$$

$$y(t) = g(t,x(t),y(t)). \tag{1.9b}$$

ODE (1.9a) depends on the additional algebraic variable y and the solution $(x,y)^T$ has to satisfy the algebraic constraint given in the form of equation (1.9b).

Semi-explicit DAEs are also broken into two classes: IVP and BVP. However, unlike explicit ODEs for which the initial or boundary values have a certain freedom, for DAE, they have to be *consistent*, that is to satisfy the algebraic constraint (1.9b). Thus, an initial value DAE has the form

$$x'(t) = f(t,x(t),y(t)), \tag{1.10a}$$

$$y(t) = g(t,x(t),y(t)), \tag{1.10b}$$

$$\begin{aligned} &x(t_0) = x_0, y(t_0) = y_0, \\ &y_0 = g(t_0,x_0,y_0). \end{aligned} \tag{1.10c}$$

In this dissertation, we foccus on *semi-explicit Index 1 DAE*, that is the case where $I_N - \partial_y g(x,y)$ is non-singular for any (x^T,y^T). Here and in what follows, I_N is the identity matrix in $\mathbb{R}^N$ and $\partial_y g(x,y)$ denotes the partial derivative of g with respect to y evaluated at the point (x,y).

Existence and uniqueness of the solution to the system (1.9) is not straightforward like that of (1.1). In addition to the condition under which ODE (1.9a) admits a unique solution, one needs also to examine the case for the algebraic restriction (1.9b). The uniqueness of the solution to the equation (1.9a) depends on the smoothness of f respect to the variable. Concerning equation (1.9b), the typical way to deal with a non-linear problem is the implicit function theorem.

If x and y are vector valued functions with dimension N and M respectively, D is a compact subset of $\mathbb{R}^{N+M}$ and $G = (f^T,g^T)^T$, problem (1.9) admits a unique solution $(x^T(t),y^T(t))$ if the following conditions are fulfilled:

I - *Smoothness condition*: The mapping $G : D \to \mathbb{R}^{N+M}$ is sufficiently differentiable.

II - *Non-singularity condition*: The matrix $I_N - \partial_y g(x,y)$ is non-singular for any (x^T,y^T).

III - *Inclusion condition*: There exist a convex set D_0 such that $(x_0^T,y_0^T)^T \in D_0$ and $D_0 \subset D$. Here $\subset$ denotes the inclusion with some neighbourhood.

1.2.1 Samples of Index-1 Differential Algebraic Equations

Problem DAE1: The first index 1 semi-explicit DAE problem is:

$$x_1'(t) = 10t\exp\left(5(y_2(t)-1)\right)x_2(t), \tag{1.11a}$$

$$x_2'(t) = -2t\ln\left(y_1(t)\right), \tag{1.11b}$$

$$y_1(t) = x_1(t)^{\frac{1}{5}}, \tag{1.11c}$$

$$y_2(t) = \left(x_2(t)^2 + y_2(t)^2\right)/2. \tag{1.11d}$$

We consider $t \in [1.0708712, 1.4123836]$ and the initial condition is assumed to be

$$(x_1(1.0708712), x_2(1.0708712), y_1(1.0708712), y_2(1.0708712))^T$$

where

$$\begin{aligned}
x_1(t) &= \exp\left(5\sin(t^2)\right),\\
x_2(t) &= \cos(t^2),\\
y_1(t) &= \exp\left(\sin(t^2)\right),\\
y_2(t) &= \sin(t^2) + 1.
\end{aligned}$$

The last formulae constitute the exact solution to problem (1.11) (See for example [11]).

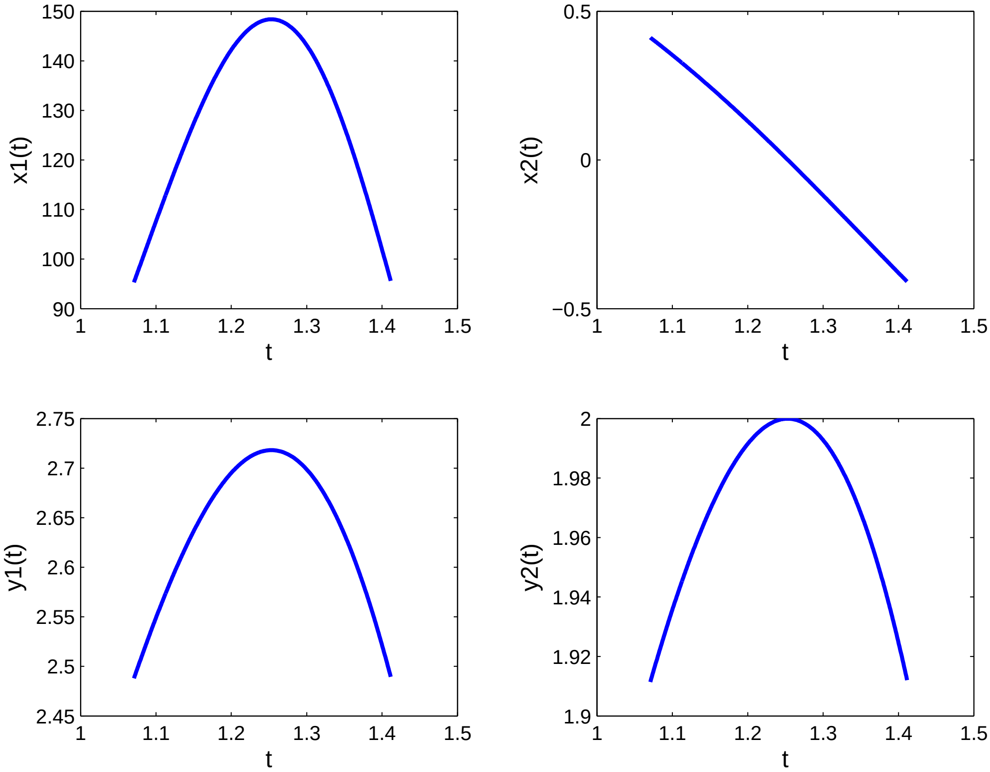

Figure 1.5: Graph of the exact solution to Problem DAE1

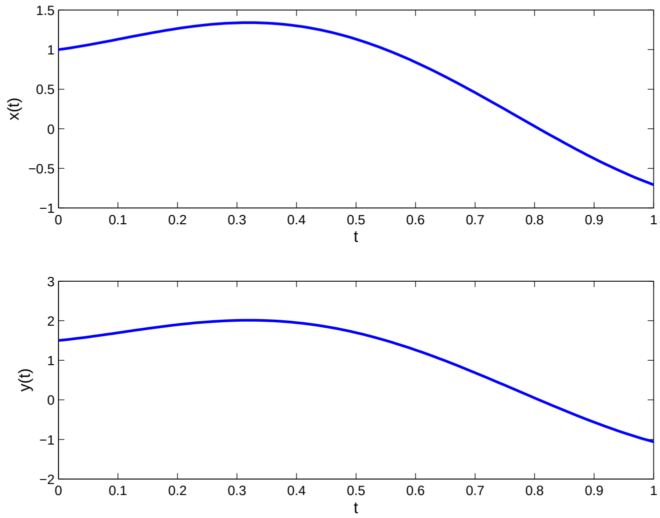

Figure 1.6: Graph of the exact solution to Problem DAE3

Problem DAE2: Middly Stiff DAE

As a sample of stiff DAE, we take the following problem:

$$x'(t) = \lambda \left(\frac{\lambda}{1+\lambda} x(t) - \sin(\mu t) \right) + y(t) + \mu \cos(\mu t) \tag{1.12a}$$

$$y(t) = \lambda (x(t) - y(t)), \tag{1.12b}$$

$t \in [0,1]$. The initial values are

$$x(0) = 1, \; y(0) = \frac{\lambda}{1+\lambda}.$$

The exact solution to this problem is well known (see [14]) and given by the formulas

$$x(t) = e^{\lambda t} + \sin(\mu t) \tag{1.13a}$$

$$y(t) = \frac{\lambda}{1+\lambda} x(t). \tag{1.13b}$$

We will examine the above-mentioned global error estimation strategies on test problem (1.12) when $\lambda = -3$ and $\mu = 4$. The graphs of the exact solution (1.13) are given in Figure 1.6.

Chapter 2

Multistep Methods and Error Evaluations

Multistep methods were developed as an extension of the Euler methods. Such methods are also referred to as *Methods with memory* by Shampine [17] because of the use of previously computed approximate solution to perform one integration step.

In this chapter, we recall the basic properties of multistep methods such as order, stability, convergence and global error expansion. Then, we introduce different strategies to evaluate the error for multistep method.

2.1 Multistep Methods

2.1.1 Formulation of Multistep Methods

Consider the uniform grid

$$w = \{t_0 < t_1 < \ldots < t_K = T,\ t_k = t_{k-1} + h \text{ for } k = 1, 2, \ldots K \text{ and } h \in \mathbb{R}\}. \tag{2.1}$$

At a point t_k of mesh (2.1), a multistep method for ODEs makes use of previously computed solution values to update the solution. If x_{k-i}, $i = 1, 2, \ldots l$ for some $l \in \mathbb{N}$, and the corresponding derivatives are used to compute the new value x_k, the method is an $l-$step linear method. Such a method has the following general form:

$$\sum_{i=0}^{l} \alpha_i x_{k-i} = h \sum_{i=0}^{l} \beta_i f_{k-i} \text{ for } k = l, l+1, \ldots K \tag{2.2}$$

where x_{k-i} stands for the approximation of $x(t_{k-i})$ and $f_{k-i} = f(x_{k-i})$.

The first multistep methods, known as the *Adams-Bashforth methods* where published in 1883 by Adams and Bashforth [8]. The l-step AB method has the form

$$x_k = x_{k-1} + h\sum_{i=1}^{l}\beta_i f_{k-i}. \tag{2.3}$$

Later, Moulton worked on the AB methods and came up with methods that have the general form

$$x_k = x_{k-1} + h\sum_{i=0}^{l}\beta_i f_{k-i} \tag{2.4}$$

and possess better properties than those of Adams and Bashforth.

In the AB methods (2.3), notice that $\beta_0 = 0$. The method is said to be *explicit*. Otherwise, that is if $\beta_0 \neq 0$, the method is *implicit*.

Another range of multistep methods, known as the *Backward Difference Formulae* (BDF) were introduced by Curtiss and Hirschfelder in 1952. These methods use several x values per step, but only one evaluation of f. BDF methods have the general formula

$$\sum_{i=0}^{l}\alpha_i x_{k-i} = hf_k. \tag{2.5}$$

Although the first multistep methods for ODE were developed in 1883, the fundamental theory of these methods was first established only in 1956 by Dahlquist [8]. Basic properties of numerical methods include consistency, stability and convergence.

2.1.2 Consistency - Stability - Convergence

The consistency of a method is defined by its ability to solve the test problems

$$x'(t) = 0, \quad \text{with } x(t_0) = 1 \tag{2.6}$$

and

$$x'(t) = 1, \quad \text{with } x(t_0) = 0 \tag{2.7}$$

correctly.

It is shown that a multistep method for ODE is consistent if the parameters α_i and β_i, $i = 0, 1, \ldots l$ satisfy

$$\begin{aligned} &\alpha_0 + \alpha_1 + \ldots + \alpha_l = 0, \\ &\alpha_1 + 2\alpha_2 \ldots + l\alpha_l = \beta_0 + \beta_1 + \ldots + \beta_l. \end{aligned} \tag{2.8}$$

In modern literature it is also termed as consistency of order 1.

The stability of a method is concerned with the boundedness of the numerical solution to

$$x'(0) = 0, \tag{2.9}$$

as the stepsize h tends to 0. The difference equation obtained when applying a multistep method to this problem has the form

$$\alpha_0 x_k + \alpha_1 x_{k-1} + \ldots + \alpha_l x_{k-l} = 0. \tag{2.10}$$

Thus, the method is *zero-stable* if all solutions to the difference equation (2.10) are bounded as $k \to \infty$. Using the properties of difference equations [4], a multistep method for ODE is zero-stable if its characteristic polynomial

$$\sum_{i=0}^{l} \alpha_i t^i = 0 \tag{2.11}$$

satisfies the *root condition*, that is the roots of (2.11) lie in the unit disk, and there is no repeated root on the boundary.

The stability property of a multistep method is defined by only the root condition. However, it is shown in practice that there is a difference in the stability of the methods. In fact, the root condition suggests that there is no repeated root of (2.11) on the unit circle, and the consistency of method (2.2) implies that 1 is a simple root. There may be or may not be other simple solutions of modulo 1. The presence of such other roots is referred to as *weak stability* and the method is described as *weakly stable* [8]. Otherwise, the method is *strongly stable*.

As example, the first Dahlquist barrier affirms that the order of a l-step method does not exceed $l+2$ if l is even and $l+1$ if l is odd [8], and it is stated in [4] that for methods with maximal order, all the roots of (2.11) lie on the unit circle. That is the methods are weakly stable.

The convergence of a multistep method is defined as follows

Definition 2.1. *[8] The linear multistep method (2.2) is called convergent if for all initial value problem (1.2) satisfying the conditions in Theorem 1.2,*

$$x(t) - x_h(t) \to 0 \textit{ for } h \to 0, t \in [t_0, T]$$

whenever

$$x(t_0 + kh) - x_h(t_0 + kh) \to 0 \quad \textit{for } h \to 0, k = 0, 1, \ldots, l-1$$

where

$$x_h(t) = x_k \text{ if } t = t_0 + kh.$$

The consistency, stability and convergence of a multistep method are related by the following theorem.

Theorem 2.2. *[4, 8] A linear multistep method is convergent if and only if it is stable and consistent.*

2.1.3 Local Error and Order of Convergence

The growing needs of highly accurate methods and the fast development in computer technology show that the convergence of a multistep method, as defined in Definition 2.1 is no longer sufficient. One needs stronger property of the multistep method to ensure that the error in the approximation is relatively small and the convergence to the exact solution can be achieved faster. This property is referred to as the *order of convergence* of the method. By analogy with Definition 2.1, we define the convergence of order p of a multistep method as follow:

Definition 2.3. *Method (2.2) is convergent of order p if for any sufficiently smooth right hand side f in (1.2),*

$$||x(t) - x_h(t)|| = O(h^p), \quad h \to 0, \tag{2.12}$$

wherever the starting values satisfy

$$||x(t_0) - x_k|| = O(h^p), \quad h \to 0, \quad k = 0, 1, \ldots l-1. \tag{2.13}$$

Define the defect of a multistep method by

$$L(t_k, h, x(t)) = \sum_{i=0}^{l} \alpha_i x(t_{k-i}) - h \sum_{i=0}^{l} \beta_i f(x(t_{k-i})). \tag{2.14}$$

The method is said to be *consistent of order p* if the defect satisfies

$$L(t_k, h, x(t)) = O(h^{p+1})$$

for any sufficiently regular ODE. It is proved also, that the method has order p if the defect vanishes for any polynomial of degree less than or equal to p [8].

It is also stated in [8] that a multistep method is convergent of order p if and only if it is consistent of order p and stable.

2.2 Global Error Expansion

In global error expansion, we seek for the global error expansion in powers of the stepsize h. To deal with the existence of such an expansion for multistep methods, Hairer and Lubich [7] considered the formulation of a multistep method as a one-step method in a space of higher dimension. This formulation, first introduced by Butcher in 1966 [3] and Skeel in 1976 [19], consists of:

- an initial procedure to compute the initial values

$$u_0 = \Phi(h), \tag{2.15}$$

- a forward step procedure to update the solution

$$u_{k+1} = Su_k + h\Phi_k(t_k, u_k, h) \tag{2.16}$$

  where S is a square matrix and the Φ_k are sufficiently differentiable, and

- a sufficiently smooth correct value function $z(t,h)$.

The vectors u_k and $z(t,h)$ are given by

$$u_k = (x_{k-l+1}, \ldots, x_k)^T \text{ and} \tag{2.17a}$$

$$z(t,h) = (x(t-(l-1)h), \ldots, x(t))^T. \tag{2.17b}$$

Having established this reduction to one-step method, one can now apply result of the global error expansion theory to the one-step method obtained and prove that if method (2.2) is convergent of order p, then the global error has an expansion of the form

$$z(t,h) - u_n = e_p(t)h^p + e_{p+1}(t)h^{p+1} + \ldots + e_N(t)h^N + E(t,h)h^{N+1} \tag{2.18}$$

where $t = a + nh$ [7]. The existence of expansion (2.18) is proved for strongly stable methods. The coefficients $e_j(t)$ of the above mentioned expansion are given in [7].

2.3 Stability of Multistep Methods

In addition to the zero-stability discussed in section 2.1.2, the notion of *A-stability* is also important for numerical methods. Basically, it determines whether a numerical method is suitable for stiff problems or not.

Consider the test problem

$$x'(t) = \lambda x(t) \tag{2.19}$$

where λ is a complex number. To be able to solve this problem using a multistep method, the difference equation

$$\sum_{i=0}^{l} \alpha_i x_{n-i} = h \sum_{i=0}^{l} \beta_i f_{n-i}$$

or, equivalently,

$$\sum_{i=0}^{l} (\alpha_i - z\beta_i) x_{n-i} = 0, \tag{2.20}$$

where $z = \lambda h$, must be bounded as $n \to \infty$. For the solution to (2.20) to be bounded, the roots of the characteristic equation given by

$$\sum_{i=0}^{l} (\alpha_i - z\beta_i) \omega^{l-i} = 0 \tag{2.21}$$

must lie in the open unit disk.

After rearranging (2.21), we have

$$z = \frac{\alpha(\omega)}{\beta(\omega)} \tag{2.22}$$

where

$$\alpha(\omega) = \sum_{i=0}^{l} \alpha_i \omega^i$$

and

$$\beta(\omega) = \sum_{i=0}^{l} \beta_i \omega^i$$

and we are interested in the values of z corresponding to $|\omega| < 1$. This part of the plan is called *stability region* of the method.

Equation (2.22) maps the unit circle on a closed curve known as the *boundary locus curve* in the complex plane. The stability region is the portion of the plan enclosed by the boundary locus curve. A method is said to be *A-stable* if its stability region covers the half plane the with negative real part.

The A-stability of a multistep method is restricted by the second Dahlquist barrier which states that an A-stable linear multistep method has order of convergence of at most 2.

2.4 Variable Stepsize Multistep Methods

Fixed stepsize methods have the advantage that they are easy to implement. However, on one hand, one may want to increase the stepsize to achieve the integration faster when the approximate solutions are reasonably accurate. On the other hand, one may want to reduce it to improve the accuracy of the computed values. Thus, methods with variable stepsize are practically more efficient.

So far, two classes of implementation of variable stepsize multistep methods were developed. The first class consists of recomputing the method coefficients at each step of the integration. The second class lies on the interpolation of the previously computed solutions and apply a fixed step method on a uniform grid within the step.

We further consider the non-uniform grid

$$w = \{t_0 < t_1 < \ldots < t_K = T,\ t_k = t_{k-1} + h_{k-1} \text{ for } k = 1,2,\ldots K\}. \tag{2.23}$$

2.4.1 Variable Stepsize Multistep Methods with Variable Coefficients

Consider the autonomous ODE

$$x' = f(x(t)) \tag{2.24}$$

where $f(x(t))$ and $x(t) \in \mathbb{R}^N$ and $t \in [t_0, T]$, with the initial condition $x(t_0)$. A variable stepsize multistep method with variable coefficients update the solution to the equation (2.24) using the formula

$$\sum_{i=0}^{l} \alpha_{i,k} x_{k-i} = h_k \sum_{i=0}^{l} \beta_{i,k} f_{k-i} \text{ for } k = l, l+1, \ldots K \tag{2.25}$$

where h_k is the stepsize at the step number k, and the coefficients $\alpha_{i,k}$ and $\beta_{i,k}$ depend on the stepsize ratios $\omega_i = \frac{h_i}{h_{i-1}},\ \ i = k-l, \ldots, k$.

Recall the definition of the local truncation error and the order of a multistep method introduced for a fixed-stepsize multistep method. Method (2.25) is *consistent of order p* if the local truncation error

$$L(t_k, h, x(t)) = \sum_{i=0}^{l} \alpha_{i,k} x(t_{k-i}) - h \sum_{i=0}^{l} \beta_{i,k} f(x(t_{k-i})). \tag{2.26}$$

vanishes for any polynomials of degree less than or equal to p [8].

To study the stability of method (2.25), we consider again the test problem $x'(t) = 0$ for $t \in [a,b]$. When method (2.25) is applied to this equation, we get

$$\sum_{i=0}^{l} \alpha_{i,k} x_{k+i} = 0.$$

Consider the vector $X_k = (x_{k+l-1}^T, \ldots, x_k^T)^T$. It is easy to see that the stability of method (2.25) is equivalent to the boundedness of each component of the vector X_k for all k. The vectors X_k are related by

$$X_{k+1} = A_k X_k$$

where

$$A_j = \begin{pmatrix} -\alpha_{l-1,j} & \ldots & -\alpha_{1,j} & -\alpha_{0,n} \\ 1 & 0 & \ldots & 0 \\ 0 & \ddots & & \vdots \\ 0 & \ldots & 1 & 0 \end{pmatrix} \tag{2.27}$$

for $j = 0, 1, \ldots k$ and $X_0 = (x_{l-1}^T, x_{l-2}^T, \ldots, x_0^T)^T$. Clearly, we have $X_{k+1} = A_k A_{k-1} \ldots A_0 X_0$. Thus, method (2.25) is *zero-stable* if the matrix $A_k A_{k-1} \ldots A_{k-l}$ is bounded for all $k, l \geq 0$ [8].

The following theorem, established in 1984 by Crouzeix and Lisbona [5], relates the stability of method (2.25) to the stepsize ratio and the coefficients α and β.

Theorem 2.4. *Assume that [8]:*

(i) $\sum_{i=0}^{l} \alpha_{i,k} = 0$

(ii) *The coefficients* $\alpha_{i,k} = \alpha_i(\omega_{k+l-1}, \ldots, \omega_{k+1}, \omega_k)$ *are continuous functions in a neighbourhood of* $(1, 1, \ldots, 1)$

(iii) *The roots of* $\sum_{i=0}^{l} \alpha_i(1, 1, \ldots, 1)t^j = 0$*, with the exception of* 1*, lie within the open unit disk* $|t| < 1$

Then there exist real numbers ω *and* Ω *such that the method is stable if* $\omega \leq \dfrac{h_k}{h_{k-1}} \leq \Omega$ *for all* k*.*

The definition of the convergence and convergence of order p of method (2.25) are the same as for methods with fixed stepsize methods given by Definition 2.1 and Definition 2.3 respectively, and the classical result given in Theorem 2.2 holds for variable stepsize methods with variable coefficients. The following theorem is proved in [8]:

Theorem 2.5. *[8] Assume that*

(a) *The method (2.25) is stable, consistent of order p and has bounded coefficients $\alpha_{i,n}, \beta_{i,n}$,*

(b) *the starting vector values $x_0, x_1, \ldots x_k$ are accurate up to $O(h^p)$ and*

(c) *the stepsize ratios $\dfrac{h_n}{h_{n-1}}$ are bounded by some Ω for all n.*

Then it is convergent of order p.

2.4.2 Interpolation Type Multistep Methods

The rigorous formulation and study of the second class of variable stepsize multistep method can be found in [13]. Such methods can be described as combinations of polynomial interpolation with a fixed stepsize multistep method. It works as follows:

At the $(k+1)$'st step of the integration, two additional uniform grids defined by

$$w_k = \{t_{k-i}^k = t_k - ih_{k-1},\ \ i = 0,\ 1,\ \ldots,\ l\} \tag{2.28a}$$

and

$$w_{k+1} = \{t_{k+1-i}^{k+1} = t_k - (i-1)h_k,\ \ i = 0,\ 1,\ \ldots,\ l\} \tag{2.28b}$$

are introduced. Using the grid points t_{k-i}^k with the corresponding solution values x_{k-1}^k, for $i = 0,\ 1,\ \ldots,\ l$, we compute the Hermite polynomial interpolation at the points x_{k+1-i}^{k+1}, that is

$$x_{k+1-i}^{k+1} = H_{l+1}^p(t_{k+1-i}^{k+1}),\ \ i = 1,\ 2,\ \ldots,\ l. \tag{2.29a}$$

Now, we can apply a fixed step multistep method on the grid w_{k+1} to get

$$\alpha_0 x_{k+1} + \sum_{j=1}^{l} \alpha_j x_{k+1-j}^{k+1} = \beta_0 f(x_{k+1}) + h_k \sum_{j=1}^{l} \beta_j f(x_{k+1-j}^{k+1}), \tag{2.29b}$$

$$t_{k+1}^{k+1} = t_{k+1},\ \ x_{k+1}^{k+1} = x_{k+1},\ \ k = l,\ l+1,\ \ldots,\ K-1 \tag{2.29c}$$

where $f(x_{k-i}^k),\ \ i = 0,\ 1,\ p-l-1$, with $p \leq 2l+1$ and $x_{l-i}^l,\ \ i = 0,\ 1,\ \ldots,\ l$ are given.

The defect of the interpolation-type multistep method is defined by

$$\begin{aligned} L(t_{k+1}, x(t), h_k) = {} & \alpha_0 x(t_{k+1}) - h_k \beta_0 f(x(t_{k+1})) \\ & \sum_{j=1}^{l} \alpha_j \tilde{H}_{l+1}^p(t_{k+1-j}^{k+1}) - h_k \sum_{j=1}^{l} \beta_i f(\tilde{H}_{l+1}^p(t_{k+1-j}^{k+1})), \\ & k = l-1, \ldots, K-1, \end{aligned} \tag{2.30}$$

where $\tilde{H}^{p}_{l+1}$ is the Hermite interpolating polynomial fitted to the points $x(t^k_{k-i}),\ \ i = 0,\ 1,\ \ldots,\ l$, and $f(x(t^k_{k-i})),\ \ i = 0,\ 1,\ p-l-1$.

The order of the interpolation LM method (2.29) is given by Lemma 1 in [13], which says that if the underlying multistep method is of order s, the non-uniform grid w in (2.23) has sufficiently small diameter and the stepsize ratios $\dfrac{h_k}{h_k-1}$ for $k = 1,2,\ldots K-1$ are bounded (in total) then the interpolation LM method (2.28) will be order $min(s,p)$.

The stability of method (2.29) does not result directly from that of the underlying LM method. To study the stability of method (2.29), Kulikov and Shindin considered a reduction of the multistep method to a one-step method in a space of higher dimension [13].

Given the vector

$$X^k_k = \left((x^k_k)^T, (x^k_{k-1})^T, \ldots, (x^k_{k-l})^T\right)^T \tag{2.31a}$$

and

$$F(X^k_k) = \left((f(x^k_k))^T, (f(x^k_{k-1}))^T, \ldots, (f(x^k_{k-l}))^T\right)^T, \tag{2.31b}$$

the interpolation LM method (2.29) is equivalent to the following one-step method :

$$\begin{aligned} X^{k+1}_{k+1} = &(\bar{U}_1 \otimes I_N)\left((H_1(k)\otimes I_N)X^k_k + h_k(H_2(k)\otimes I_N)F(X^k_k\right)\\ &+ h_k(\bar{U}_2\otimes I_N)F\left((H_1(k)\otimes I_N)X^k_k + h_k(H_2(k)\otimes I_N)G(X^k_k)\right)\\ &+ h_k(\bar{U}_3\otimes I_N)F(X^{k+1}_{k+1}). \end{aligned} \tag{2.32}$$

where I_N is the identity matrix of order N and $\otimes$ denotes the direct product of two matrices. The coefficient matrices are given in [10] and the interpolation matrices $H_1(k)$ and $H_2(k)$ are introduced in [13].

Define the set $\mathbb{W}^{\infty}_{\omega_1,\omega_2}(t_0,T)$ of grids on the interval $[t_0,T]$ satisfying the following conditions:

$$0 < \omega_1 \le h_k/h_{k1} \le \omega_2 < \infty, k = 1,2,...,K1, \tag{2.33a}$$

$$h/h_k < \infty, k = 0,1,...,K1 \tag{2.33b}$$

where h is the diameter of the grid.

The stability of the interpolation type multistep method is defined as follow.

Definition 2.6. *The interpolation type multistep method is said to be stable on the set* $\mathbb{W}^{\infty}_{\omega_1,\omega_2}(t_0,T)$ *if, for a finite constant R, we have*

$$\| \prod_{j=0}^{m} \bar{U}_1 H_1(k-j) \| \le R,\ \ m = 0,1,\ldots,k-l+1,\ \ k = l-1,l,\ldots,K-1 \tag{2.34}$$

for any grid $w \in \mathbb{W}^{\infty}_{\omega_1,\omega_2}(t_0,T)$.

It is also shown in [13] that the interpolation multistep method is convergent if and only if it is stable and consistent.

2.5 Multistep Methods for Semi-Explicit Index 1 DAE

Different methods have also been developed to solve a semi-explicit index 1 DAE. Examples can be found in [2, 9]. In this work, we will consider the *state space form* method to solve the system of equation (1.10).

We further assume that the problem has a unique solution. Given the initial conditions (1.10c) and the grid in (2.23), the state space form approach consist of solving (1.10a) using method a multistep method for ODE and require the solution $z^T = (x^T, y^T)^T$ to satisfy the algebraic constraint (1.10b). Thus, a state space form multistep method with variable coefficients for solving (1.10) has the form

$$\sum_{j=0}^{l} \alpha_{j,k} x_{k+1-j} = h_k \sum_{j=0}^{l} \beta_{j,k} f(x_{k+1-j}, y_{k+1-j}), \tag{2.35a}$$

$$y_{k+1} = g(x_{k+1}, y_{k+1}) \tag{2.35b}$$

for $k = l, l+1, \dots K$ where all coefficients are the same as for method (2.25) and $z_0, z_1, \dots z_{l-1}$ are given.

For interpolation multistep methods, we consider again the additional grids given in (2.28) and solve (1.10) using the procedure described in (2.29). Thus, an interpolation type LM method to solve (1.10) has the form

$$z^{k+1}_{k+1-i} = H^p_{l+1}(t^{k+1}_{k+1-i}), \;\; i = 1, 2, \dots, l, \tag{2.36a}$$

$$\alpha_0 x_{k+1} + \sum_{j=1}^{l} \alpha_j x^{k+1}_{k+1-j} = \beta_0 f(x_{k+1}, y_{k+1}) + h_k \sum_{j=1}^{l} \beta_j f(x^{k+1}_{k+1-j}, y^{k+1}_{k+1-j}), \tag{2.36b}$$

$$y_{k+1} = g(x_{k+1}, y_{k+1}), \tag{2.36c}$$

$$t^{k+1}_{k+1} = t_{k+1}, \; z^{k+1}_{k+1} = z_{k+1}, \;\; k = l, l+1, \dots, K-1 \tag{2.36d}$$

where H^p_{l+1} is the interpolating polynomial based on the points $z(t^k_{k-i}), i = 0, 1, \dots, l$, and

$$F\left(x(t^k_{k-i}), y(t^k_{k-i})\right), i = 0, 1, p-l-1.$$

The $x-$component of the defect of methods (2.35) and (2.36) are similar to (2.26) and (2.30) respectively, where the algebraic component y must be added as an argument of L and the right hand side function f. The $y-$ component of the defect is alway equal to zero as a direct consequence of (2.35b) and (2.36c).

2.6 Global Error Evaluation Techniques

When using numerical methods to solve any mathematical problems, evaluation of the error in the approximate solution is of extreme importance. The growing need for high accuracy computation has always obliged mathematicians to develop and improve routines to keep the error smaller than a given tolerance. Error evaluations, in one hand, tell us how accurate the approximate solutions are. On the other hand, they allow us to improve the approximation accordingly.

A review and classification of global error evaluation techniques were presented by Skeel in [20]. Most of the methods presented in [20] can be applied for various numerical methods for solving different mathematical problems such as ODEs, index 1 DAEs, PDEs. Concerning the estimation of the error propagated in numerical solutions of ODEs, a comparison of some methods was established by Aïd and Levacher in [1].

In this work, we focus on some methods presented in the Skeel's review [20] and the error evaluation introduced by Kulikov and Shindin in [10] for multistep methods. Namely, the algorithms include:

Richardson extrapolation,

Using two different methods,

Zadunaisky's technique,

Solving for the correction,

Solving the linearised discrete variational equation.

In this section, we give a brief theoretical overview of the above-mentioned techniques. For this purpose, we will further consider an ODE given by the prototype (1.8) and DAE defined in (1.10). Furthermore, $z^T = (x^T, y^T)^T$ will denote the solution to (1.10). A brief description of the implementation will follow the theoretical aspect of each method.

2.6.1 Richardson Extrapolation

The Richardson extrapolation is a widely used technique to evaluate the global error. It was first introduced in 1910 for partial differential equations [8]. For multistep methods, the validity of this method lies on the existence of global error expansion of the form (2.18) (See section 2.2).

In parallel to the integration of the ODE (1.8) with stepsize h, one integrates the same problem using the same multistep method, but with a smaller stepsize $\frac{h}{2}$. The existence of the expansion of the global error allows us to write

$$x(t_k) - x_k^h = h^s e_s(t_k) + O(h^{s+1}) \tag{2.37a}$$

and

$$x(t_k) - x_{2k}^{h/2} = \left(\frac{h}{2}\right)^s e_s(t_k) + O(h^{s+1}). \tag{2.37b}$$

where s is the order of the method, and x_k^h and $x_k^{h/2}$ denote the numerical solution obtained at the point t_k with stepsize h and $\frac{h}{2}$ respectively. It follows from equations (2.37a) and (2.37b) that the leading term of the global error in x_k^h is given by

$$h^s e_s(t_k) = \frac{x_k^h - x_{2k}^{h/2}}{2^{-s} - 1}. \tag{2.38}$$

This term provides an estimate of the global error accurate to of $O(h^{s+1})$.

The same procedure and arguments hold for the index 1 semi-explicit DAE given by (1.10); that is the global error of the numerical solution z_k is given by

$$z(t_k) - z_k = \frac{z_k^h - z_{2k}^{h/2}}{2^{-s} - 1} + O(h^{s+1}).$$

For the implementation, we will use the following procedure.

- **For** $k = 0, 1, \ldots, s-1$

  As the real solutions to the test problems are known, they will be used as initial values for the integrations. That is :

  $$x_k^h = x(t_k),\ k = 0, 1, \ldots s-1.$$

  Thus, the real error at t_k is also known to be equal to 0 for $k = 0, 1, \ldots, s-1$, and it will be used as its own estimate. The exact values of the solution will also be attributed as initial value for the integration using the stepsize $\frac{h}{2}$, that is

  $$x_j^{h/2} = x(t_j),\ j = 0, 1, \ldots 2(s-1).$$

- **For** $k \geq s$

  Step 1 Compute the numerical solution x_k^h with stepsize h using formula (2.2)

  Step 2 Compute the numerical solution $x_{2k-1}^{h/2}$ with stepsize $\dfrac{h}{2}$ using formula (2.2)

  Step 3 Compute the numerical solution $x_{2k}^{h/2}$ with stepsize $\dfrac{h}{2}$ using formula (2.2)

  Step 4 Compute the estimation of the error in x_k^h using formula (2.38).

Modified Newton methods with 3 iterations are used to solve the non-linear equation that results from (2.2) in the case of implicit methods. The initial guess for the iteration was computed using a polynomial interpolation based on the previously computed numerical solutions. This implementation is used for all the methods that we consider in this dissertation.

The same procedure will be used in the case of index 1 DAE where x_k^h, $x_{2k-1}^{h/2}$ and $x_{2k}^{h/2}$ will be replaced by z_k^h, $z_{2k-1}^{h/2}$ and $z_{2k}^{h/2}$ respectively.

Eventhough the Richardson extrapolation is one of the earliest method to estimate the global error, its validity is still restricted to uniform grids. It also has the drawback that the numerical solutions needed in **Step 2** and **Step 3** increase the cost of the integration considerably.

2.6.2 Using Two Different Methods

In this technique, two integrations of the original problem (1.10) are also carried out in parallel. The fundamental requirements are that the two methods have different orders, and the problem is integrated on the same grid by these two methods. Its validity lies on the existence of the asymptotic expansion of the global error as well.

Assume that the chosen methods have orders s_1 and s_2 such that $s_1 < s_2$. If the numerical solutions computed at the point t_k are denoted by x_k^1 and x_k^2, respectively, we have

$$x(t_k) - x_k^1 = x(t_k) - x_k^2 + x_k^2 - x_k^1.$$

Thus, we easily conclude

$$x(t_k) - x_k^1 = x_k^2 - x_k^1 + O(h^{s_2}). \tag{2.39}$$

Equation (2.39) says that the difference $x_k^2 - x_k^1$ provides an estimate for the error of the less accurate solution x_k^1. This estimate is accurate to $O(h^{s_2})$.

Similarly, for equation (1.10), we obtain

$$z(t_k) - z_k^1 = z_k^2 - z_k^1 + O(h^{s_2}). \tag{2.40}$$

For the implementation, the exact solutions at the points $t_k, k = 0, 1, \ldots, s_1 - 1, s_2 - 1$ will be used as initial values for each multistep methods, and the corresponding error estimate are set to be 0. In this work, we chose multistep methods of order $s_1 = 4$ to solve the initial problems and the methods of order $s_2 = 5$ from the same classes are used as higher order method to compute the second numerical solution and the error estimate for $k \geq s_1$.

2.6.3 Zadunaisky Technique

The Zadunaisky's technique first appeared in 1966 in [21]. The method is classified as a differential correction by Skeel in [20] and a survey of the method for differential equations is provided in [1].

The idea of the Zadunaisky's technique lies on the fact that if a problem, "close" to the original one, is given with its exact solution, then one can expect that the error produced in its numerical integration provides an approximation of the error in the numerical solution to the original problem. The method works as follow.

Using the approximate solutions of the original problem, one constructs a continuous approximation of the exact solution. This continuous approximation is usually given by the piecewise polynomial

$$P_h(t) = P_j(t),\ t \in [t_{(j-1)m},\ \ldots,\ t_{jm}],\ j = 1,\ 2, \ldots, \tag{2.41}$$

where P_j is a polynomial interpolation based on the points t_k and x_k, $k = (j-1)m,\ \ldots,\ jm$ for some integer m [21]. Then, we consider the neighbouring problem defined by the system

$$\tilde{x}'(t) = f(\tilde{x}(t)) - f(P_h(t)) + P_h'(t), \tag{2.42}$$

with the initial conditions of the original problem. It is easy to see that P_h satisfies the equation (2.42). We solve the equation (2.42) numerically using the same method and the same grid as for solving the original problem (1.8). If $\tilde{x}_k$ is the solution of the (2.42) at the point t_k then,

$$E_k = \tilde{x}_k - x_k \tag{2.43}$$

is expected to be an estimation of the error in x_k.

For index 1 DAE, P_h is the polynomial interpolation based on t_k and z_k, $k = (j-1)m, \ldots, jm$ and the neighbouring problem is given by

$$\tilde{x}'(t) = f(\tilde{x}(t), \tilde{y}(t)) - f(P_{h,x}, P_{h,y}) + \partial_x P_h, \tag{2.44a}$$

$$\tilde{y}(t) = g(\tilde{x}(t), \tilde{y}(t)) - g(P_{h,x}, P_{h,y}) + P_{h,y} \tag{2.44b}$$

Here and in what follows, $P_{h,x}$ and $P_{h,y}$ are the x and y-component of the polynomial P_h respectively.

For the implementation, exact solutions will also be used as initial values for the initial and the neighbouring problems. That is $x_k = x(t_k)$ and $\tilde{x}_k = x(t_k)$ for $k = 0, 1, \ldots, s-1$. The corresponding global error estimate are set to 0.

For $k \geq s$, we use the following procedure to solve the equation and estimate the global error.

step 1 To start, we set $j = 1$

Step 2 Compute the numerical solution x_k of the original probelm at the point t_k for $k = j(m-1), \ldots, jm$

Step 3 Using the numerical solutions computed in Step 1, construct the polynomial interpolation P_h using formula (2.41)

Step 4 Compute the numerical solution $\tilde{x}_k$ of the neighbouring problem (2.42) at the point t_k for $k = j(m-1), \ldots, jm$

Step 5 The global error estimate is $\tilde{x}_k - x_k$ for $k = j(m-1), \ldots, jm$

Step 6 Increase j by 1 and repeat the process from Step 2.

The same procedure is applied for the implementation of the method for index 1 DAE. The Zadunaiky's technique theoretically provides an approximation to the error with higher order, depending on the degree of the polynomial interpolation [1]. In this work, we chose the value $m = s + 1$.

2.6.4 Solving for the Correction

The solving for the correction uses the interpolation process introduced for Zadunaiky's technique in equation (2.41). With the same definition of P_h, the global error at the grid points $t_i, i = l, l+$

$1, \ldots, K$ is given by the equation

$$E(t) = x(t) - P_h(t) \tag{2.45}$$

One can easily verify that E satisfies to the ODE

$$E'(t) = f(P_h(t) + E(t)) - P_h'(t) \tag{2.46}$$

with the initial condition $E(0) = 0$. The idea is then to solve the equation (2.46) using the same method and on the same grid as for solving the original equation (1.8). The approximate solution E_k of this problem at the point t_k is an estimation of the error in x_k.

For index 1 DAE, the equation of the error consists also of an index 1 DAE described as follow.

$$\begin{aligned} E_x'(t) &= f(P_h(t) + E(t)) - P_{h,x}'(t) \\ E_y(t) &= g(P_h(t) + E(t)) - P_{h,y}(t) \end{aligned} \tag{2.47}$$

where E_x is the x-component of E and E_y denotes its y-component.

For the implementation, exact solutions will also be used as initial values for the initial problem, that is $x_k = x(t_k)$. For the equation (2.46) for the error, the initial values are 0.

For $k \geq s$, we use a procedure similar to the one used for Zadunaisky's techniques:

step 1 To start, we set $j = 1$

Step 2 Compute the numerical solution x_k of the original probelm at the point t_k for $k = j(m-1), \ldots, jm$

Step 3 Using the numerical solutions computed in Step 1, construct the polynomial interpolation P_h using formula (2.41)

Step 4 Compute the numerical solution E_k of the error equation (2.46) at the point t_k for $k = j(m-1), \ldots, jm$

Step 5 Increase j by 1 and repeat the process from Step 2.

The solving for the correction theoretically provides an estimation with the same accuracy of the Zadunaisky's technique [1, 20]. For the numerical tests, we also used $m = s+1$.

2.6.5 Solving the Linearised Discrete Variational Equation

The solving the linearised discrete variational equation (SLDVE) was first introduce in [10]. In the case of multistep methods for ODEs, the global error is given by the relation

$$\begin{aligned} a_{0,k}(x(t_{k+1}-x_{k+1}) = &-\sum_{i=1}^{s} a_{i,k}(x(t_{k+1-i}-x_{k+1-i}) \\ &+ h_k\sum_{i=0}^{k} b_{i,k}(f(x(t_{k+1-i}))-f(x_{k+1-i}))+L(t_k,x(t),h_k) \end{aligned} \tag{2.48}$$

The smoothness of the right hand side function f allows us to use the Taylor expansion and get

$$\begin{aligned} a_{0,k}(x(t_{k+1})-x_{k+1}) = &-\sum_{i=1}^{s} a_{i,k}(x(t_{k+1-i})-x_{k+1-i}) \\ &+ h_k\sum_{i=0}^{k} b_{i,k}J_f(x_{k+1-i})(x(t_{k+1-i})-x_{k+1-i}) \\ &+ L(t_k,x(t),h_k) \end{aligned} \tag{2.49}$$

for $k = s-1, l, \ldots, K-1$ where J_f designs the Jacobian of the function f.

If the errors at the initial points $t_{k+1-i},\ \ i = 1,2,\ldots,l$ are known, the global error is given by

$$\begin{aligned} x(t_{k+1})-x_{k+1} = &\left(a_{0,k}I_N - h_k b_{0,k}J_f(x_{k+1})\right)^{-1} \times \chi_k \\ &+ h_k\sum_{i=0}^{s} O\left(x(t_{k+1-i})-x_{k+1-i}\right)^2 \end{aligned} \tag{2.50}$$

where

$$\chi_k = \sum_{i=1}^{s}(h_k b_{i,k}J_f(x_{k+1-i})-a_{i,k}I_N)(x(t_{k+1-i})-x_{k+1-i})+L(t_k,x(t),h_k).$$

If we set $x(t_{k+1-i})-x_{k+1-i}=0$ for $i=1,2,\ldots,s$, we get from equation (2.50) the following approximation for the local error

$$\Delta\tilde{x}_{k+1} = \left(a_{0,k}I_N - h_k b_{0,k}J_f(x_{k+1})\right)^{-1} L(t_k,x(t),h_k). \tag{2.51}$$

Expression in equation (2.51) provides an accurate estimation of the local error. However, the formula cannot be used for real computation as it depends on the exact solution of the problem. To overcome this problem, Kulikov and Shindin [12] make use of the Taylor expansion of $x(t)$ and

$x'(t)$ at the point t_{k+1} and the fact that the method has order s to get an approximation $\tilde{L}(t_k, x(t), h_k)$ to $L(t_k, x(t), h_k)$ given by the formula

$$\begin{aligned}\tilde{L}(t_k, x(t), h_k) = & \frac{(-1)^{s+1}}{(s+1)!} x_{k+1}^{s+1} \\ & \times \sum_{i=1}^{s} \left(a_{i,k} \sum_{j=0}^{i-1} h_{k-j} + (s+1) h_k b_{i,k} \right) \left(\sum_{i=0}^{i-1} h_{k-1} \right)^{s} .\end{aligned} \tag{2.52}$$

This formula will also be used to compute the global error by the mean of equation (2.50).

Chapter 3

Numerical Results for ODEs

We introduced the theoretical background of different global error evaluation strategies in Chapter 2. In this chapter, we aim to compare these methods when they are implemented in multistep formulae for ODEs. To perform the test, we consider as test problems the ODEs with known solution described in section 1.1.1.

We present the result obtained for each global error evaluation technique on the test problems. Then at the end of each test, we will draw a conclusion according to accuracy and running time of the methods under discussion.

3.1 Numerical Result for Adams Methods

In this section, we discuss the numerical results obtained when the global error evaluation strategies are implemented on Adams methods. The coefficients of the methods, as well as their stability properties are discussed in [8, 9] for different order formulae. Adams methods are know to be strongly stable.

3.1.1 Implementation on Uniform Grids

To find the real error and its estimate, we integrate the test problems on the interval $[0, 1]$ using uniform grid and construct the global error evaluation techniques for the Adams methods of order 4. Three different stepsizes are used for each test problem and global error evaluation techniques.

	$h = 10^{-2}$	$h = 10^{-3}$	$h = 10^{-4}$
Richardson	2.668e-12	1.310e-16	2.180e-16
Using 2 methods	1.100e-12	8.760e-17	8.055e-17
Zadunaisky	1.458e-02	1.458e-03	1.458e-04
SC	9.612e-01	9.961e-01	9.996e-01
SLDVE	9.851e-12	1.906e-16	9.825e-18

Table 3.1: Accuracy of the global error evaluation techniques applied to ODE1 on the uniform grids.

	$h = 10^{-2}$	$h = 10^{-3}$	$h = 10^{-4}$
Richardson	0.032	0.311	3.049
Using 2 methods	0.026	0.249	2.631
Zadunaisky	0.082	0.781	7.875
SC	0.078	0.779	7.717
SLDVE	0.018	0.159	1.508

Table 3.2: CPU time (in sec) of the global error evaluation applied to ODE1 on the uniform grids

Here, we use test problems with known solution, thus the starting values which correspond to the exact solution at the starting points are given with sufficient accuracy.

We consider three uniform grids w_1, w_2 and w_3 in the interval $[0, 1]$ and with stepsize $h = 10^{-2}$, $h = 10^{-3}$ and $h = 10^{-4}$ respectively.

1. **Problem ODE1:** Numerical results for ODE1 on uniform grids are presented in Figures 3.1 and Table 4.19. The graphs for the Zadunaisky's technique and the solving for the corrections are not presented in Figure 3.1 because the error estimate computed using these methods differ significantly from the real error. The accuracy of the Zadunaisky's technique and solving for the correction, as shown in Table 4.19, are low. As for the using two methods, Richardson extrapolation and SLDVE, Figure 3.1 shows that the error estimate provided by these methods coincide with the real error for $h = 10^{-2}$ and $h = 10^{-3}$. For $h = 10^{-4}$ however, the Figure 3.1 and Table 4.19 exhibit the advantage of SLDVE as using two methods and the Richardson extrapolation lose some accuracy.

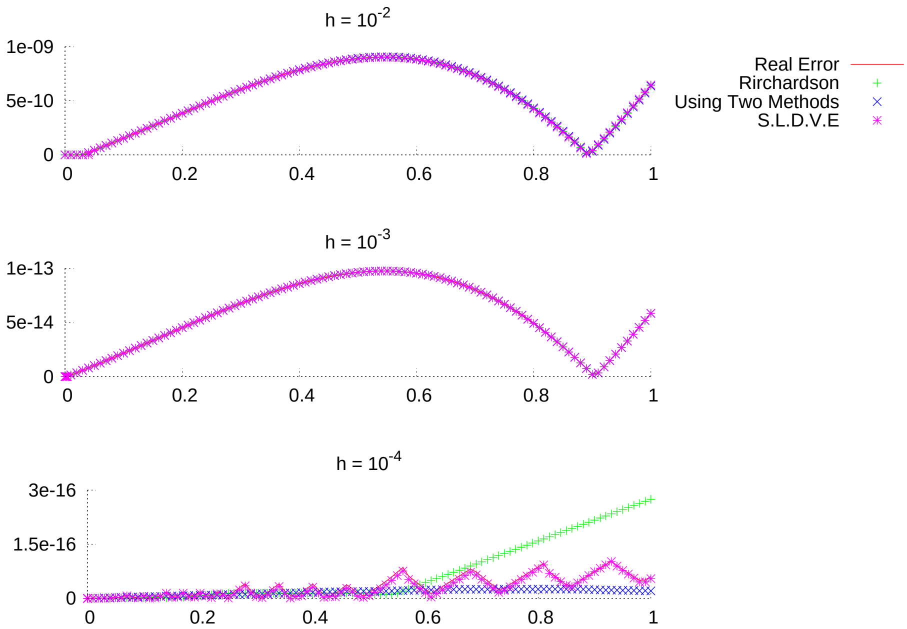

Figure 3.1: The true error and its estimates obtained for Adams methods applied to ODE1 on the uniform grids

In terms of CPU time, Table 3.2 shows that SLDVE is less expensive than the Richardson extrapolation and the using two different methods. Namely, SLDVE runs 1.5 times faster than using two methods and 2 times faster than the Richardson extrapolation when they are used to estimate the true error in Adams methods on the uniform grids.

2. **Problem ODE2:** Numerical results for ODE2 on uniform grids are given in Figure 3.2 and Table 3.3. In this case also, the graphs for the Zadunaisky's technique and the solving for the correction were omitted in Figure 3.2 as the error estimates computed using these strategies do not agree with the real error (See Table 3.3). For $h = 10^{-2}$ and $h = 10^{-3}$, Figure 3.2 shows that the behaviour of the error estimate computed using Richardson extrapolation, using two different methods and SLDVE are very similar to the real error. The global error evaluation strategies provided an error estimate with accuracy up to h^5. However, for $h = 10^{-4}$, only SLDVE gives the same accuracy. The order of error estimate is reduced for the Richardson extrapolation and using two different methods (See Table 3.3).

   In terms of CPU time, the SLDVE is characterised by its low running time, followed by the using two methods and Richardson extrapolation. The CPU time for the Zadunaisky's tech-

	$h = 10^{-2}$	$h = 10^{-3}$	$h = 10^{-4}$
Richardson	1.855e-11	3.372e-16	2.140e-16
Using 2 methods	3.515e-12	1.789e-16	9.156e-17
Zadunaisky	1.960e-02	1.990e-03	1.989e-04
SC	9.6125e-01	9.961e-01	9.996e-01
SLDVE	2.8407e-10	2.7998e-15	2.7618e-20

Table 3.3: Accuracy of the global error evaluation techniques applied to ODE2 on the uniform grids.

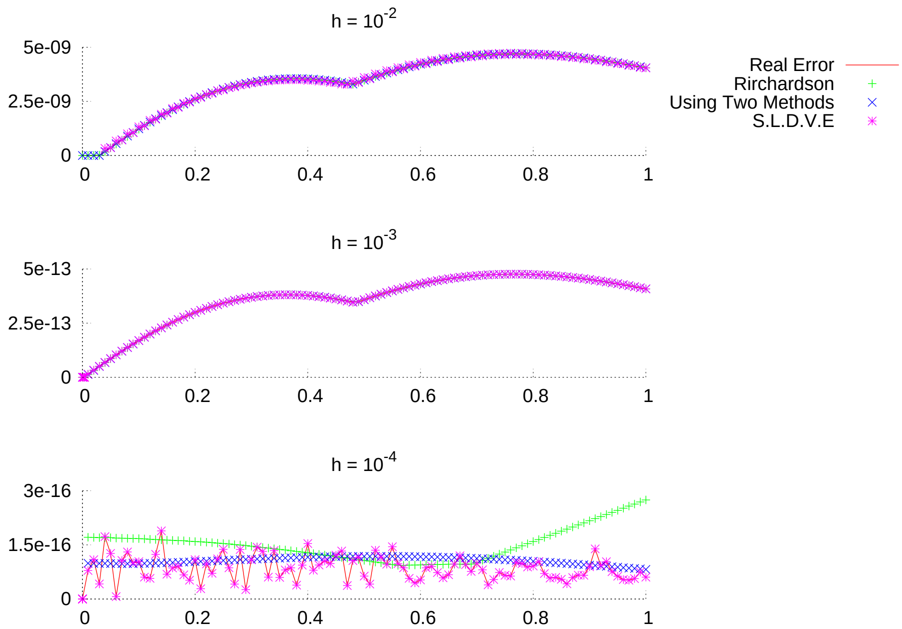

Figure 3.2: The true error and its estimates obtained for Adams methods applied to ODE2 on the uniform grids

	$h = 10^{-2}$	$h = 10^{-3}$	$h = 10^{-4}$
Richardson	0.054	0.539	5.296
Using 2 methods	0.046	0.419	4.471
Zadunaisky	0.119	1.186	11.952
SC	0.114	1.143	11.550
SLDVE	0.027	0.291	2.472

Table 3.4: CPU time (in sec) of the global error evaluation applied to ODE2 on the uniform grids

	$h = 10^{-2}$	$h = 10^{-3}$	$h = 10^{-4}$
Richardson	9.091e-08	9.077e-13	2.232e-14
Using 2 methods	1.146e-07	1.223e-13	1.497e-14
Zadunaisky	3.568e+00	3.623e-01	3.629e-02
SC	4.188e-05	4.423e-09	4.525e-13
SLDVE	3.101e-07	3.094e-12	1.815e-14

Table 3.5: Accuracy of the global error evaluation techniques applied to ODE3 on the uniform grids.

nique and the solving for the correction show that the methods are computationally expensive (See Table 3.4).

3. **Problem ODE3:** Numerical results for ODE3 on uniform grids are given in Figure 3.3 and Table 3.5. In Figure 3.3, only the error estimate given by Richardson extrapolation, using two methods and SLDVE are presented with the true error because of the lack of accuracy in the error estimate when Zadunaisky's technique and solving for the correction are used (See Table 3.5). Figure 3.3 exhibits the ability of the Richardson extrapolation, using two methods and SLDVE to estimate the true error. For the three stepsizes, the behaviour of the true error and its estimates are very similar.

   We notice that SLDVE runs faster than the other methods that we consider in this dissertation. The Zadunaisky's technique and solving for the correction are shown to be computationally expensive (See Table 3.6).

4. **Problem ODE4:** The accuracy of different error estimation strategies applied to the $4th$ order Adams method for solving ODE4 on the uniform grid are given in Table 3.7. Richard-

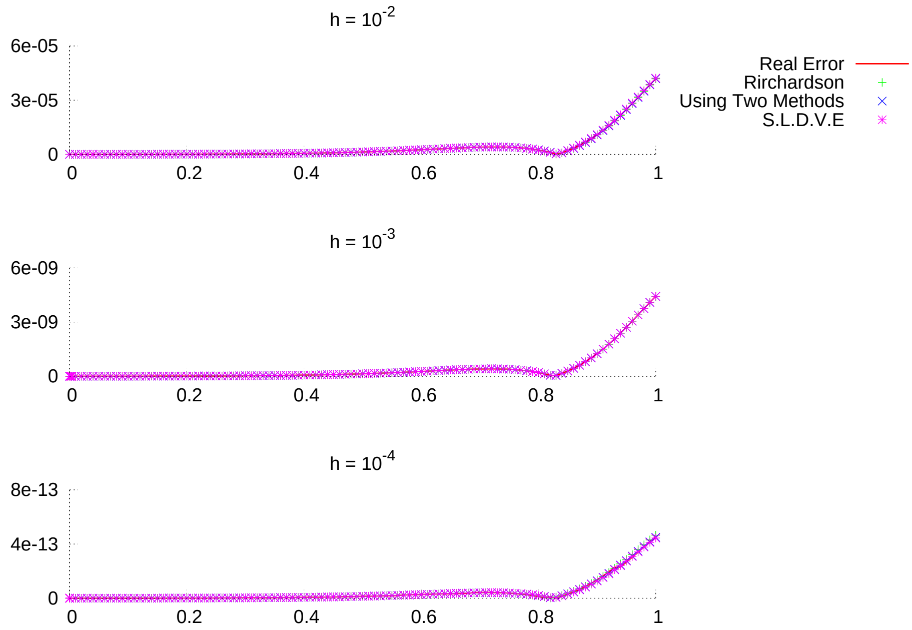

Figure 3.3: The true error and its estimates obtained for Adams methods applied to ODE3 on the uniform grids

	$h = 10^{-2}$	$h = 10^{-3}$	$h = 10^{-4}$
Richardson	0.061	0.583	5.852
Using 2 methods	0.048	0.458	4.839
Zadunaisky	0.128	1.217	12.242
SC	0.160	1.627	16.431
SLDVE	0.030	0.307	2.736

Table 3.6: CPU time (in sec) of the global error evaluation applied to ODE3 on the uniform grids

	$h = 10^{-2}$	$h = 10^{-3}$	$h = 10^{-4}$
Richardson	2.042e-10	2.019e-15	3.571e-16
Using 2 methods	4.664e-11	2.991e-16	3.052e-16
Zadunaisky	4.289e-02	4.290e-03	4.290e-04
SC	9.639e-01	9.961e-01	9.996e-01
SLDVE	2.872e-10	3.166e-15	3.388e-16

Table 3.7: Accuracy of the global error evaluation techniques applied to ODE4 on the uniform grids.

	$h = 10^{-2}$	$h = 10^{-3}$	$h = 10^{-4}$
Richardson	0.036	0.315	3.108
Using 2 methods	0.028	0.253	2.647
Zadunaisky	0.085	0.793	7.889
SC	0.160	1.627	16.431
SLDVE	0.020	0.157	1.571

Table 3.8: CPU time (in sec) of the global error evaluation applied to ODE4 on the uniform grids

son extrapolation, using two different methods and SLDVE work correctly and provide the same accuracy for different stepsizes.. Figure 3.4 shows that the true error and its estimates agree very well for $h = 10^{-2}$ and $h = 10^{-3}$, but the error evaluation technique present some difficulties to estimate the error for $h = 10^{-4}$. The graphs for Zadunaisky's technique and the solving for the correction are not show in Figure 3.4 because of the lack of accuracy in the error estimate. The error in the global error estimate raised up to 10^{-1} and 10^{-4} when $h = 10^{-4}$. In addition to their poor accuracy, these methods are computationally expensive (See Table 3.8). For the stiff problem ODE4, we notice that the SLDVE, like for non-stiff problems, solve the equation and evaluate the global error faster than the other methods under consideration here (See Table 3.8).

5. **Comparison** We stress that we used the Adams methods of order 4 in the experiments.

   For the oscillatory problem (**ODE1**) given by equation 1.4, table 4.19 and figure 3.1 show that Richardson extrapolation, Using Two Different Methods and SLDVE provide error estimate with the same accuracy. Concernening the Zadunaisky's technique and Solving for

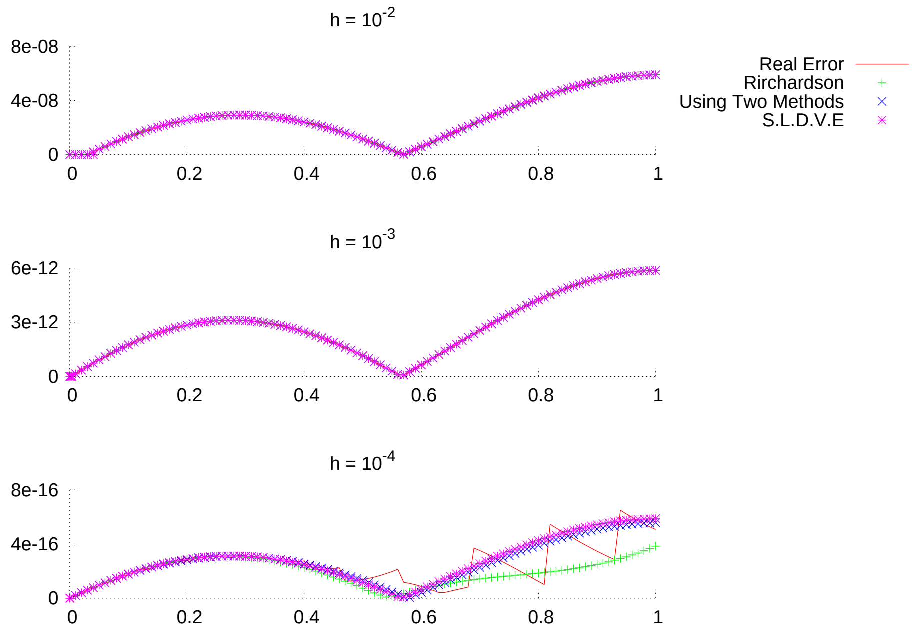

Figure 3.4: The true error and its estimates obtained for Adams methods applied to ODE4 on the uniform grids

the Correction, the accuracy of these error estimations are unreasonably low. In other words, we think that those two methods are not accurate to estimate the error generated by Adams methods. It correlates with the opinion of Aïd and Levacher [1]. In terms of CPU time (See table 3.2), the SLDVE runs 1.5 times faster than Using two different methods and 2 times faster than the Richardson extrapolation. The same results are observed when we run the test using (**ODE2**) described by the system of equations (1.5) and (**ODE3**) given by system (1.6), which are also non-stiff problems.

For ODE4, which is a stiff ODE, for $h = 10^{-2}$, Richardson extrapolation, Using two methods and SLDVE produced an error estimation with an accuracy of $O(h^{-5})$. The same results were obtained when we reduced the step size to $h = 10^{-3}$ and $h = 10^{-4}$. The errors in the error estimation provided the Zadunaisky's technique and the Solving for the correction raised up to 10^{-1}.

In terms of CPU time, SLDVE runs also 1.5 times faster than using two methods and 2 times faster than the Richardson extrapolation (See table 3.8).

	$\tau = 10^{-2}$	$\tau = 10^{-3}$	$\tau = 10^{-4}$
Richardson	–	–	–
Using 2 methods	1.434e-05	9.618e-07	1.629e-09
Zadunaisky	9.996e-01	9.965e-01	9.996e-01
SC	1.766e-04	6.035e-07	6.089e-09
SLDVE	2.862e-09	2.239e-14	2.098e-19

Table 3.9: Accuracy of the global error evaluation techniques for Adams methods applied to ODE1 on the non-uniform grids

3.1.2 Implementation On Non-uniform Grids

In this section, we present the numerical results for Adams methods of order 4 on the non-uniform grids.

For the non-uniform grid, we use the following scheme:

$$t_{i+1} = \tau\theta + t_i \quad i = 0, \cdots, K \tag{3.1}$$

where θ takes the values $\frac{4}{5}$ and $\frac{5}{4}$ consecutively and τ take the values 10^{-2}, 10^{-3} and 10^{-4}.

We first note that the Richardson extrapolation are only applicable on uniform grid. Thus, – (dash) will be used in the tables to indicate that no results are to be presented for the Richardson extrapolation.

1. **Problem ODE1:** Numerical results for ODE1 on non-uniform grids are given in Figure 3.5, and Table 3.9. The graph for the Zadunaisky's technique is not plotted in Figure 3.5 because of the lack of accuracy in the error estimate computed using this strategy (See Table 3.9). Figure 3.5 shows that only SLDVE appears to provide an accurate estimation of the global error. Although using two different methods and solving for the correction provided error estimates better than the Zadunaisky's technique, Figure 3.5 and Table 3.9 show that the accuracy of the error estimate are also low.

   In addition to its ability to estimate the global error correctly, the SLDVE presents also the smallest CPU time among the methods under consideration here (See Table 3.10).

2. **Problem ODE2:** Numerical results for ODE2 on non-uniform grids are given in Figures 3.6 and Table 3.11.

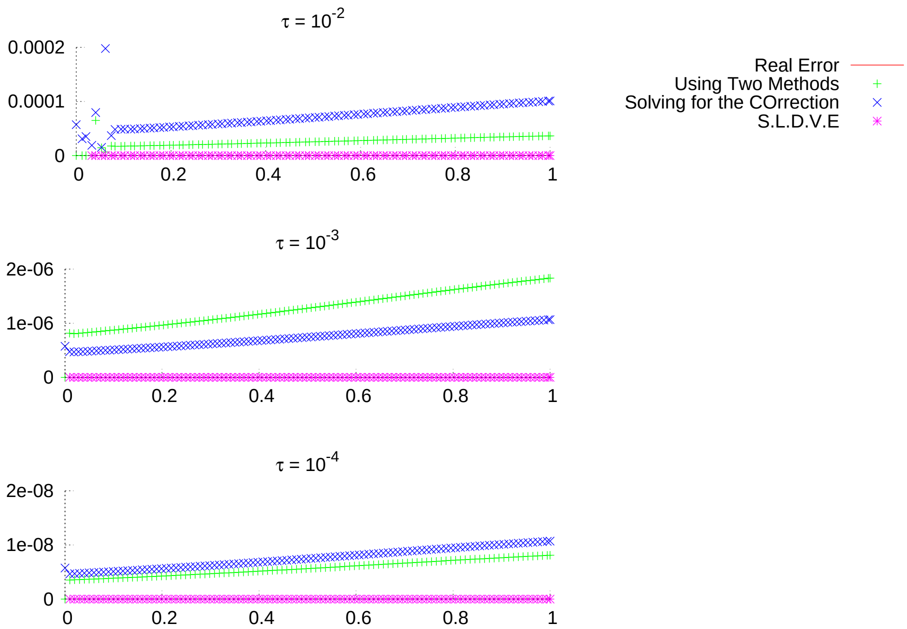

Figure 3.5: The true error and its estimates obtained for Adams methods applied to ODE1 on the non-uniform grids

	$\tau = 10^{-2}$	$\tau = 10^{-3}$	$\tau = 10^{-4}$
Richardson	–	–	–
Using 2 methods	0.028	0.253	2.647
Zadunaisky	0.085	0.793	7.889
SC	0.160	1.627	16.431
SLDVE	0.020	0.212	2.272

Table 3.10: CPU time (in sec) of the global error evaluation applied to ODE1 on the non-uniform grids

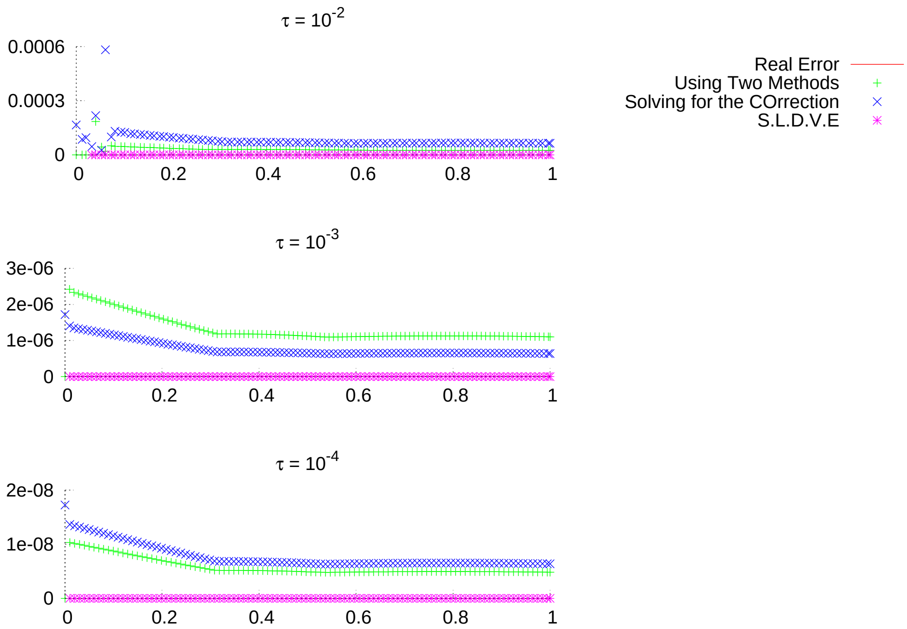

Figure 3.6: The true error and its estimates obtained for Adams methods applied to ODE2 on the non-uniform grids

For problem ODE2, Table 3.11 and Figure 3.6 tell us that only the SLDVE was able to estimate the global error correctly when the Adams method of order 4 is used to solve the problem on the non-uniform grids. In addition to the accuracy of the error estimate, the SLDVE has low CPU time when compared to the other methods that we consider here (See Table 3.12.

3. **Problem ODE3:** Numerical results for ODE3 on non-uniform grids are given in Figures 3.7 and Table 3.13. The same results as for ODE1 and ODE2 are obtained for ODE3.

4. **Problem ODE4:** Numerical results for ODE4 on non-uniform grids are given in Figures 3.8 and Table 3.15. Recall ODE4 was chosen as a sample of stiff ODE. Table 3.15 and figure 3.8 confirm that the result obtained for the non-stiff ODEs are also obtained for the stiff ODE when the different error evaluation strategies are applied to the Adams methods of order 4 on the non-uniform grids.

5. **Comparison** For the Adams methods of order 4, the tables and figures show the ability of the SLDVE to evaluate the true error for non-stiff and stiff problems. This method is also leading in terms of CPU time. Following the SLDVE is the using two different methods. Despite the

	$\tau = 10^{-2}$	$\tau = 10^{-3}$	$\tau = 10^{-4}$
Richardson	–	–	–
Using 2 methods	3.151e-05	1.307e-06	2.084e-09
Zadunaisky	9.655e-01	9.965e-01	9.996e-01
SC	5.307e-04	1.718e-06	1.724e-08
SLDVE	5.893e-09	5.922e-13	2.909e-16

Table 3.11: Accuracy of the global error evaluation techniques for Adams methods applied to ODE2 on the uniform grids

	$\tau = 10^{-2}$	$\tau = 10^{-3}$	$\tau = 10^{-4}$
Richardson	–	–	–
Using 2 methods	0.028	0.253	2.647
Zadunaisky	0.085	0.793	7.889
SC	0.160	1.627	16.431
SLDVE	0.040	0.332	3.580

Table 3.12: CPU time (in sec) of the global error evaluation applied to ODE2 on the non-uniform grids

	$\tau = 10^{-2}$	$\tau = 10^{-3}$	$\tau = 10^{-4}$
Richardson	–	–	–
Using 2 methods	8.957e-04	1.524e-04	2.550e-07
Zadunaisky	1.200+05	1.094e+05	1.081e+05
SC	7.218e-03	7.229e-05	7.229e-07
SLDVE	5.144e-06	5.322e-11	5.350e-16

Table 3.13: Accuracy of the global error evaluation techniques for Adams methods applied to ODE3 on the non-uniform grids

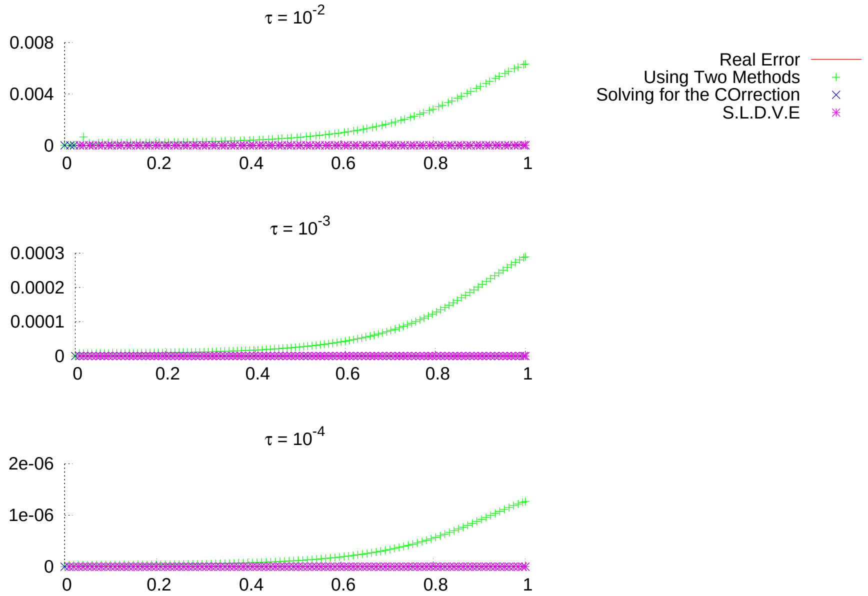

Figure 3.7: The true error and its estimates obtained for Adams methods applied to ODE3 on the non-uniform grids

	$\tau = 10^{-2}$	$\tau = 10^{-3}$	$\tau = 10^{-4}$
Richardson	–	–	–
Using 2 methods	0.028	0.253	2.647
Zadunaisky	0.085	0.793	7.889
SC	0.160	1.627	16.431
SLDVE	0.040	0.344	3.832

Table 3.14: CPU time (in sec) of the global error evaluation for Adams methods applied to ODE3 on the non-uniform grids

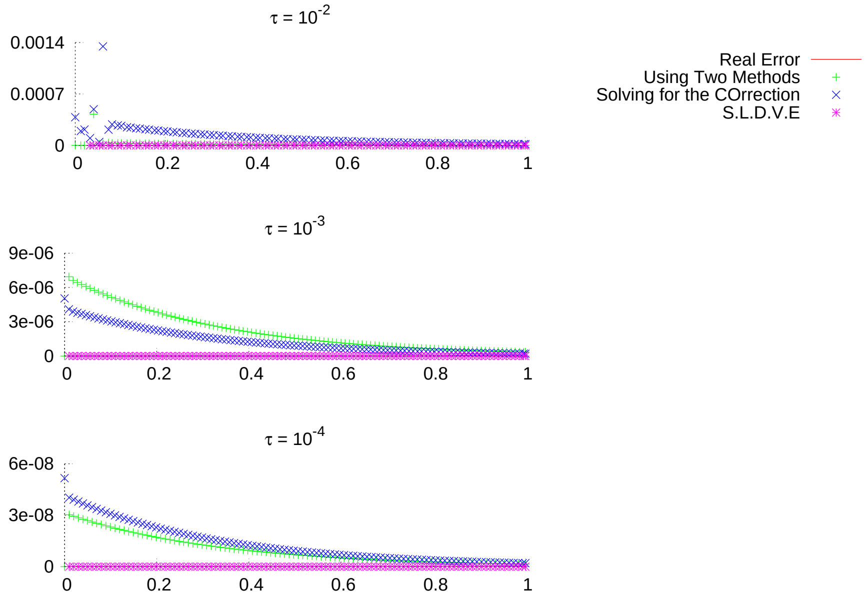

Figure 3.8: The true error and its estimates obtained for Adams methods applied to ODE4 on the non-uniform grids

	$\tau = 10^{-2}$	$\tau = 10^{-3}$	$\tau = 10^{-4}$
Richardson	–	–	–
Using 2 methods	9.143e-05	3.773e-06	6.340e-09
Zadunaisky	1.018e+00	1.000e+00	9.998e-01
SC	1.233e-03	5.039e-06	5.161e-08
SLDVE	1.452e-08	1.393e-13	1.383e-18

Table 3.15: Accuracy of the global error evaluation techniques for Adams methods applied to ODE4 on the non-uniform grids

	$\tau = 10^{-2}$	$\tau = 10^{-3}$	$\tau = 10^{-4}$
Richardson	–	–	–
Using 2 methods	0.028	0.253	2.647
Zadunaisky	0.085	0.793	7.889
SC	0.160	1.627	16.431
SLDVE	0.028	0.220	2.308

Table 3.16: CPU time (in sec) of the global error evaluation for Adams methods applied to ODE4 on the non-uniform grids

reasonably low CPU time for the using two different methods, the error estimates produced by this method are not very satisfactory for non-stiff and stiff problems. Concerning the Zadunaisky's technique and solving for the correction, the estimation of the global error are high, and the CPU time are not competitive when compared to the SLDVE and the using two different methods.

3.2 Numerical Result for BDF formulae

3.2.1 Implementation on Uniform Grids

In this section, we present the numerical results obtained for different global error evaluation techniques when they are applied to BDF formulae of order 4 to solve the test problems on the uniform grids.

1. **Problem ODE1:** Numerical results for ODE1 on uniform grids are presented in Figures 3.9 and Table 3.17. We did not plot the graph of the error estimate computed using the Zadunaisky's technique and solving for the correction in Figure 3.9 because the error estimates differ a lot from the real error. As shown in Table 3.17, the difference of between the real error and its estimate raised up to $5.104e-04$ for the Zadunaisky's technique and $9.997e-01$ for solving for the correction for $h = 10^{-4}$. The true error and the error estimates produced by the Richardson extrapolation, using two methods and SLDVE are drawn in Figure 3.9. The figure shows that the error estimates and the true error agree very well for $h = 10^{-2}$ and $h = 10^{-3}$. For $h = 10^{-4}$ the behaviour of the true error differs significantly

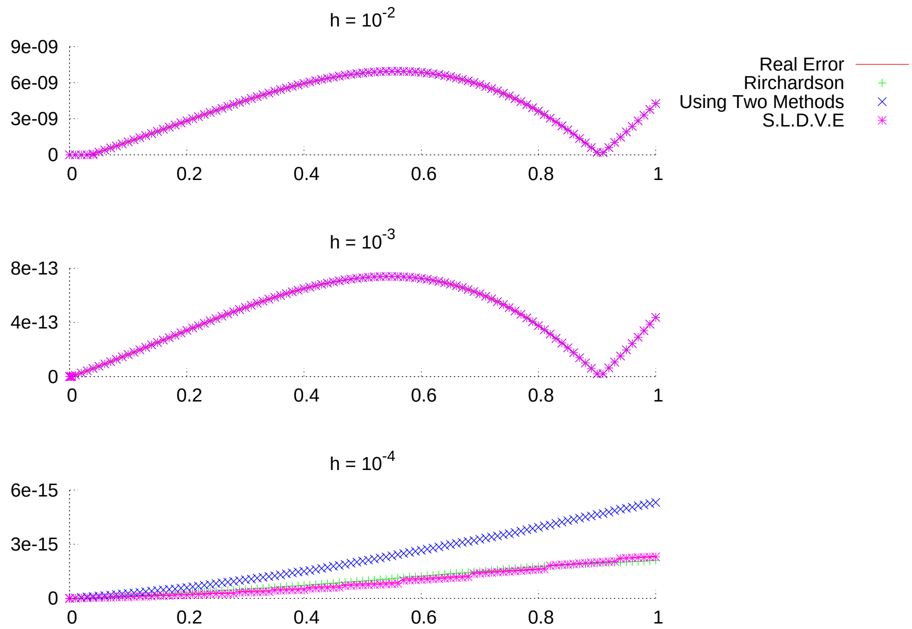

Figure 3.9: The true error and the estimates obtained for BDF formulae on the uniform grids, ODE1

from its estimates. However, the numerical data given Table 3.17 shows that, although the order of accuracy the estimations are reduced for $h =^{-4}$ for the Richardson extrapolation, using two methods and SLDVE, these methods are still very competitive when compared to the Zadunaisky's technique and the solving for the correction.

In terms of CPU time, the SLDVE represents the least expensive among the methods under consideration in this dissertation. The using two different methods and Richardson extrapolation are also cheap compared to the Zadunaisky's technique and the solving for the correction (See Table 3.18).

2. **Problem ODE2:** Numerical results for ODE2 on uniform grids are given in Figures 3.10 and Table 3.19. For the same reason as for ODE1, the graph for the Zadunaisky's technique and solving for the correction are not plotted in Figure 3.10.

   Table 3.19 shows the accuracy of the global error evaluation technique for BDF formulae applied to ODE2. The error estimation provided by the Richardson extrapolation and the using two different methods are the most accurate for $h = 10^{-2}$ and $h = 10^{-3}$. However for smaller stepsize, $h = 10^{-4}$, the SLDVE produced the best estimation of the error. In addition,

	$h = 10^{-2}$	$h = 10^{-3}$	$h = 10^{-4}$
Richardson	2.186e-11	5.590e-16	2.428e-16
Using 2 methods	1.106e-11	2.684e-16	2.973e-15
Zadunaisky	1.030e+00	5.104e-03	5.104e-04
SC	9.750e-01	9.975e-01	9.997e-01
SLDVE	7.143e-12	2.976e-16	2.297e-15

Table 3.17: Accuracy of the global error evaluation techniques for BDF formulae applied to ODE1 on the uniform grids.

	$h = 10^{-2}$	$h = 10^{-3}$	$h = 10^{-4}$
Richardson	0.040	0.384	3.785
Using 2 methods	0.032	0.309	3.045
Zadunaisky	0.065	0.522	5.109
SC	0.068	0.523	5.077
SLDVE	0.024	0.199	1.694

Table 3.18: CPU time (in sec) of the global error evaluation for BDF formulae applied to ODE1 on the uniform grids

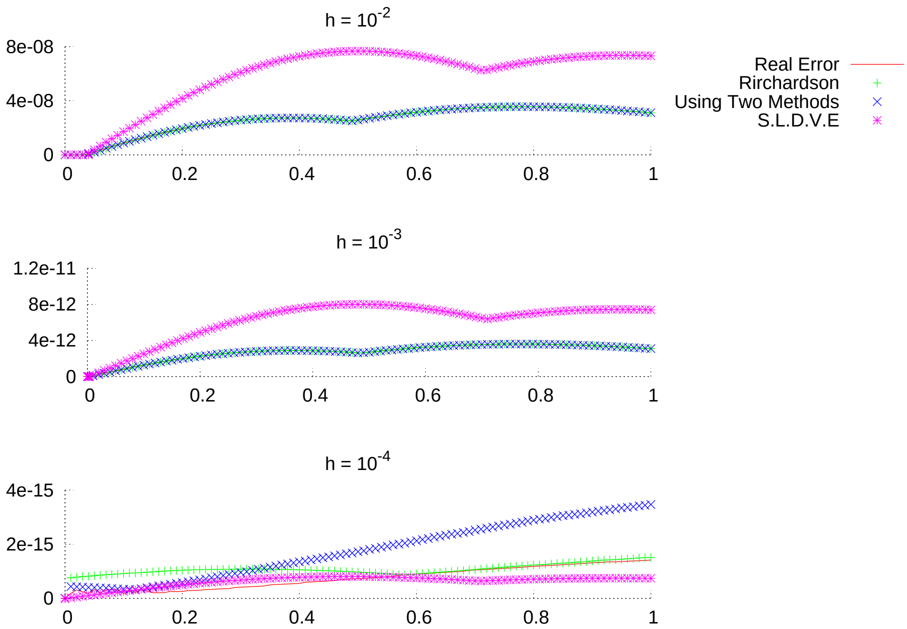

Figure 3.10: The true error and the estimates obtained for BDF formulae on the uniform grids, ODE2

SLDVE presents the advantage in terms of CPU time (See Table 3.20).

3. **Problem ODE3:** Numerical results for ODE3 on uniform grids are given in Figures 3.11 and Table 3.21.

   The graphs of the true error and its estimate provided by the Richardson extrapolation, using two methods and SLDVE are drawn in Figure 3.11. The figure shows that the behaviour of the true error and its estimate agree very well for $h = 10^{-2}$ and $h = 10^{-3}$. However, Table 3.21 show that the SLDVE evaluates the true error with more precision than the Richardson extrapolation and using two methods do. In addition to the accuracy of the error estimate, SLDVE represents the least time consuming among the methods that we consider here. The Zadunaisky's technique is characterised by the lack of ability to estimate the true error and high CPU time.

4. **Problem ODE4:** Numerical results for ODE4 on uniform grids are given in Figures 3.12 and Table 3.23.

   Similar results are obtained for the stiff ODE ODE4 in terms of accuracy and CPU time.

	$h = 10^{-2}$	$h = 10^{-3}$	$h = 10^{-4}$
Richardson	1.704e-10	1.910e-15	7.979e-16
Using 2 methods	3.534e-11	3.896e-16	2.049e-15
Zadunaisky	1.029e+00	6.979e-03	6.964e-04
SC	9.750e-01	9.975e-01	9.997e-01
SLDVE	5.150e-08	5.376e-12	6.790e-16

Table 3.19: Accuracy of the global error evaluation techniques for BDF formulae applied to ODE2 on the uniform grids.

	$h = 10^{-2}$	$h = 10^{-3}$	$h = 10^{-4}$
Richardson	0.069	0.683	6.887
Using 2 methods	0.052	0.532	5.302
Zadunaisky	0.097	0.832	8.328
SC	0.095	0.837	8.298
SLDVE	0.038	0.309	2.812

Table 3.20: CPU time (in sec) of the global error evaluation applied to BDF formulae for solving ODE2 on the uniform grids

	$h = 10^{-2}$	$h = 10^{-3}$	$h = 10^{-4}$
Richardson	1.450e-06	1.439e-11	1.229e-12
Using 2 methods	1.144e-06	1.184e-12	6.842e-13
Zadunaisky	1.193e+01	1.263e+00	1.269e-01
SC	2.949e-04	3.330e-08	2.779e-12
SLDVE	9.877e-08	1.969e-13	5.879e-13

Table 3.21: Accuracy of the global error evaluation techniques for BDF formulae applied to ODE3 on the uniform grids.

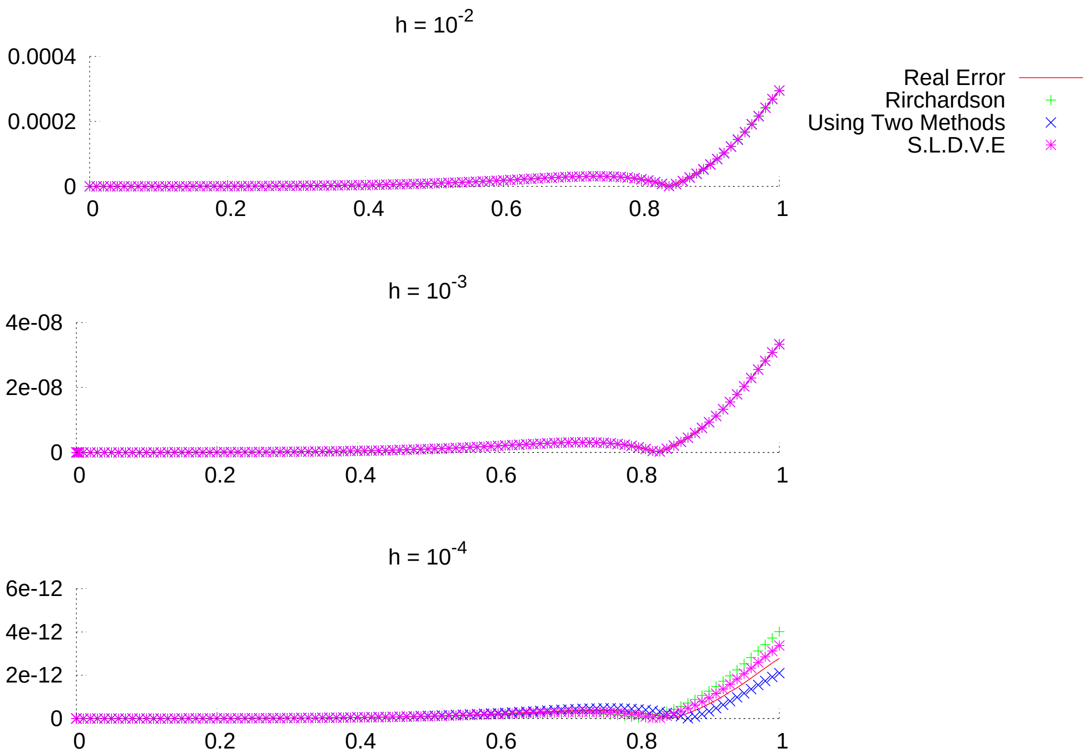

Figure 3.11: The true error and the estimates obtained for BDF formulae on the uniform grids, ODE3

	$h = 10^{-2}$	$h = 10^{-3}$	$h = 10^{-4}$
Richardson	0.080	0.786	7.868
Using 2 methods	0.054	0.582	5.831
Zadunaisky	0.097	0.872	8.851
SC	0.165	1.521	15.235
SLDVE	0.034	0.343	3.038

Table 3.22: CPU time (in sec) of the global error evaluation applied to BDF formulae for solving ODE3 on the uniform grids

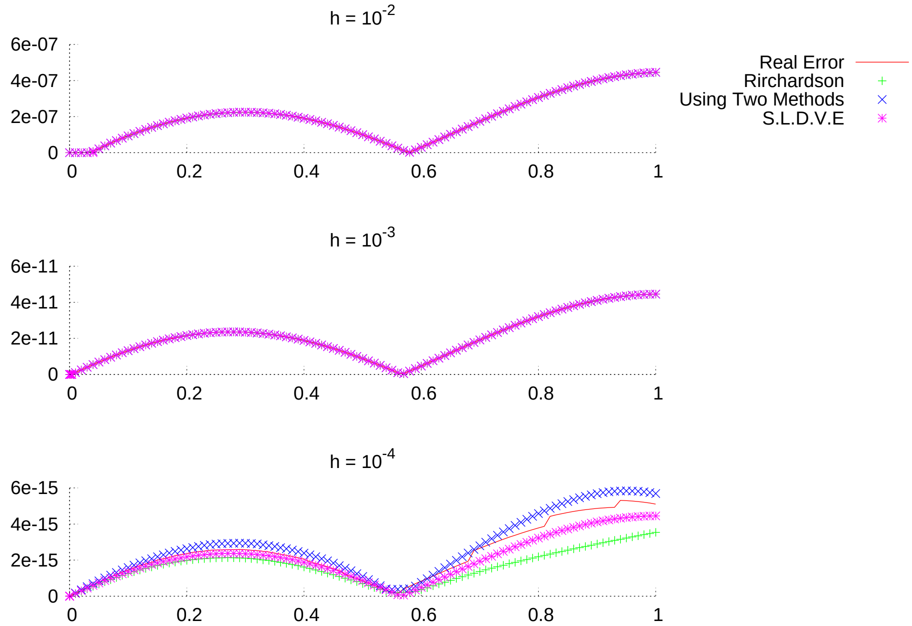

Figure 3.12: The true error and the estimates obtained for BDF formulae on the uniform grids, ODE4

	$h = 10^{-2}$	$h = 10^{-3}$	$h = 10^{-4}$
Richardson	1.822e-09	1.794e-14	2.154e-15
Using 2 methods	4.638e-10	6.346e-16	8.876e-16
Zadunaisky	1.033e+00	1.501e-02	1.501e-03
SC	9.999e-01	9.998e-01	9.999e-01
SLDVE	1.227e-09	1.267e-14	9.925e-16

Table 3.23: Accuracy of the global error evaluation techniques for BDF formulae applied to ODE4 on the uniform grids.

	$h = 10^{-2}$	$h = 10^{-3}$	$h = 10^{-4}$
Richardson	0.041	0.402	3.947
Using 2 methods	0.031	0.312	3.104
Zadunaisky	0.069	0.519	5.102
SC	0.066	0.522	5.057
SLDVE	0.019	0.188	1.728

Table 3.24: CPU time (in sec) of the global error evaluation applied to BDF formulae for solving ODE4 on the uniform grids

5. **Comparison** BDF formulae are known to have strong stability properties. Recall that to do the test, we chose to use the methods of order 4.

   The numerical results obtained for the error evaluation strategies applied to the BDF formulae are very similar to the results when the error evaluation techniques are applied to the Adams methods. SLDVE, Richardson extrapolation and Using tow different methods are competitive in terms of accuracy. However, SLDVE shows more interest in terms of CPU time. The opinion of Aïd and Levacher [1] about the Zadunaisky's technique and the solving for the correction agrees also with our results for BDF formulae on the uniform grids. That is the Zadunaisky's technique and the solving for the correction failed to estimate the true error correctly. In terms of CPU time, SLDVE has the advantage over all the other methods that we are interested in in this work.

3.2.2 Implementation on non-uniform Grids

In this section, we use the non-uniform grid introduced in Section 3.1.2. The numerical results for the global error evaluation strategies applied to the BDF formulae of order 4 are arranged as follows:

1. **Problem ODE1:** The numerical results for BDF methods for solving ODE1 on non-uniform grids are presented in Table 3.25 and Figure 3.13. In Figure 3.13, the graphs for the Zadunaisky's technique and solving for the correction are not plotted for low accuracy reason. As for SLDVE and using two methods, the figure illustrates that the error estimate computed using SLDVE agree very well with the real error and the error estimate given by using two method

	$\tau = 10^{-2}$	$\tau = 10^{-3}$	$\tau = 10^{-4}$
Richardson	–	–	–
Using 2 methods	1.147e-11	1.142e-16	4.423e-16
Zadunaisky	4.112e-03	4.047e-04	4.040e-05
SC	4.658e-03	7.907e-13	1.518e-15
SLDVE	7.397e-12	5.906e-17	5.753e-22

Table 3.25: Accuracy of the global error evaluation techniques for BDF formulae applied to ODE1 on the non-uniform grids.

	$\tau = 10^{-2}$	$\tau = 10^{-3}$	$\tau = 10^{-4}$
Richardson	–	–	–
Using 2 methods	0.024	0.659	4.389
Zadunaisky	0.140	1.369	15.195
SC	0.446	1.402	13.976
SLDVE	0.017	0.144	1.379

Table 3.26: CPU time (in sec) of the global error evaluation for BDF formulae applied to ODE1 on the non-uniform grids

differs slightly from the real error. Table 3.25 summarise the accuracy of the global error estimation strategies when they are applied to BDF on non-uniform grids. The table confirms the ability of SLDVE to estimate the error with high accuracy. This advantage of SLDVE over the other global error evaluation strategies considered here is also noticed in terms of CPU time (See Table 3.26).

2. **Problem ODE2:**

The numerical results for the global error evaluation strategies applied to BDF for solving ODE2 on the non-uniform grids are given in Table 3.27 and Figure 3.14. Figure 3.14 illustrates that the behaviour of the error estimate computed using SLDVE and using two methods are similar to the real error. Using two methods however is show to be less accurate than SLDVE. The graphs for Zadunaisky's technique and solving for the correction were not plotted for low accuracy reason. The accuracy of the methods are presented in Table 3.27, where for $\tau = 10^{-2}$ and $\tau = 10^{-3}$, SLDVE and using two methods have the same accuracy

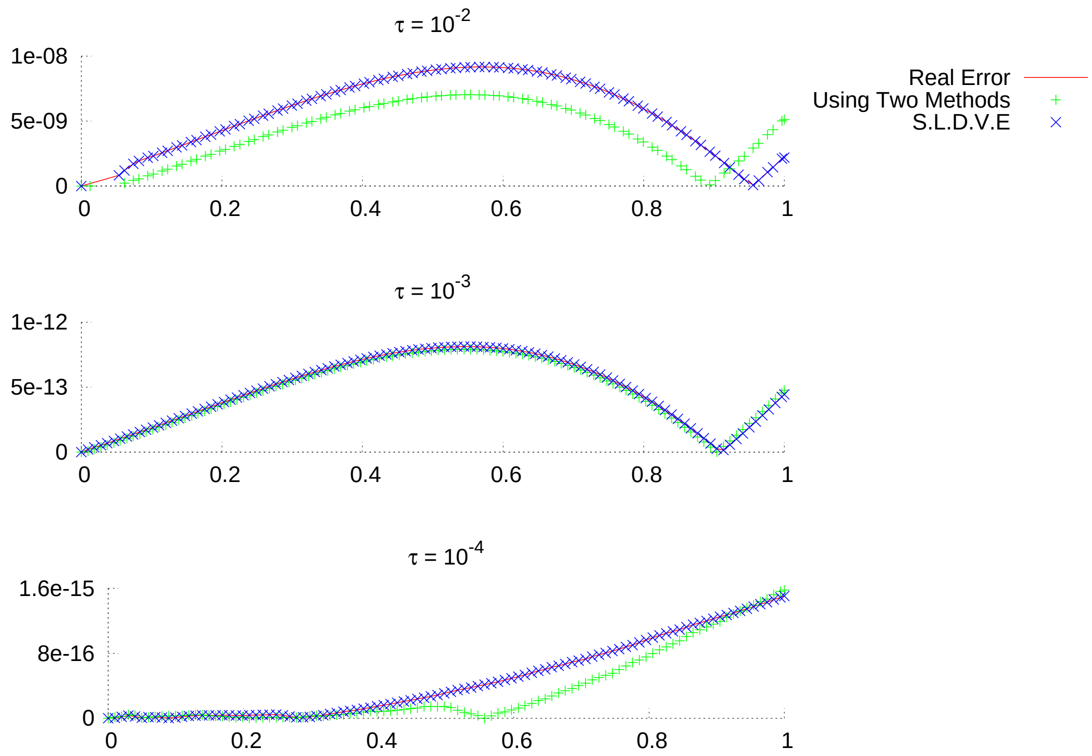

Figure 3.13: The true error and the estimates obtained for BDF formulae on the non-uniform grids, ODE1

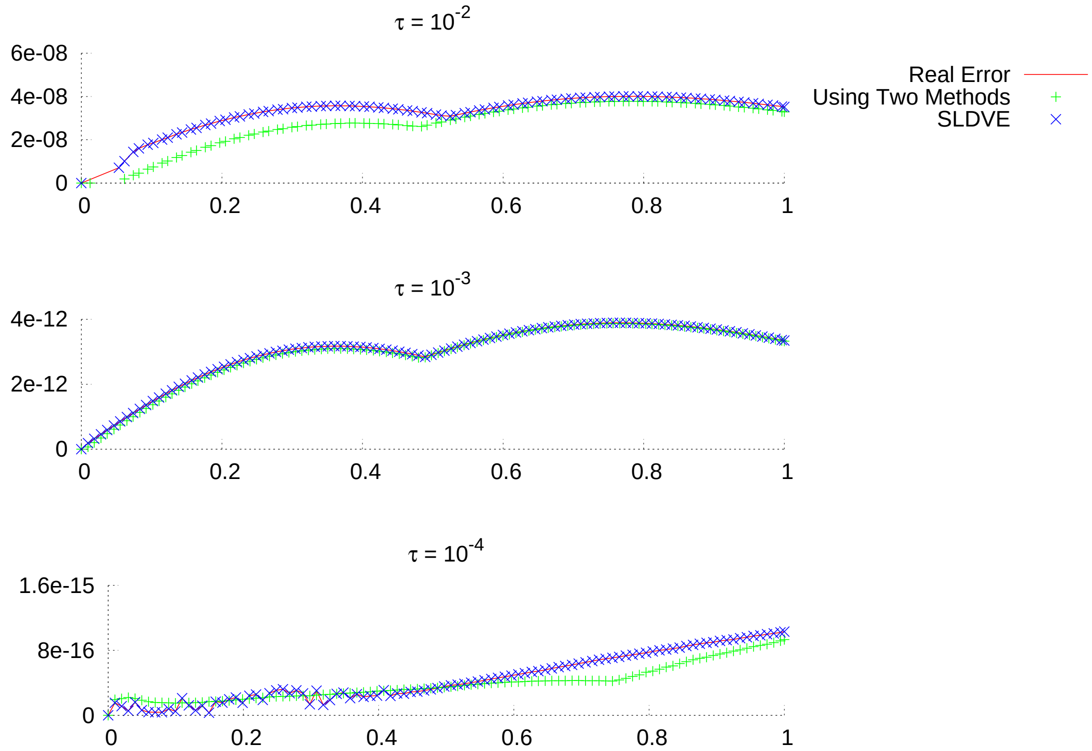

Figure 3.14: The true error and the estimates obtained for BDF formulae on the non-uniform grids, ODE2

but for $\tau = 10^{-4}$, SLDVE is more accurate. The efficiency of SLDVE is noticed also in terms of CPU time (See Table 3.28).

3. **Problem ODE3:**

   The numerical results for BDF for solving ODE3 on the non-uniform grids are given in Table 3.29 and Figure 3.15. In Figure 3.15, only the graphs for SLDVE and using two methods are plotted because of the lack of accuracy in the error estimate generated using Zadunaisky's technique and solving for the correction (See Table 3.29). Figure 3.15 illustrates that the error estimate computed using SLDVE and using two methods behave very similarly to the real error for $\tau = 10^{-2}, 10^{-3}$ and 10^{-4}. The accuracy of the methods given in Table 3.29 however shows the difference in the accuracy of the methods for $\tau = 10^{-4}$.

4. **Problem ODE4:** For the global error evaluation strategies applied to BDF for solving ODE4 on the non-uniform grids, the results are presented in Table 3.31 and Figure 3.16. The graphs

	$\tau = 10^{-2}$	$\tau = 10^{-3}$	$\tau = 10^{-4}$
Richardson	–	–	–
Using 2 methods	3.632e-11	3.520e-16	2.744e-16
Zadunaisky	6.140e-03	5.540e-04	5.510e-05
SC	8.889e-03	3.871e-12	1.002e-15
SLDVE	3.604e-11	3.350e-16	3.326e-21

Table 3.27: Accuracy of the global error evaluation techniques for BDF formulae applied to ODE2 on the non-uniform grids.

	$\tau = 10^{-2}$	$\tau = 10^{-3}$	$\tau = 10^{-4}$
Richardson	0.040	0.384	3.785
Using 2 methods	0.032	0.309	3.045
Zadunaisky	0.065	0.522	5.109
SC	0.068	0.523	5.077
SLDVE	0.024	0.199	1.694

Table 3.28: CPU time (in sec) of the global error evaluation for BDF formulae applied to ODE2 on the non-uniform grids

	$\tau = 10^{-2}$	$\tau = 10^{-3}$	$\tau = 10^{-4}$
Richardson	–	–	–
Using 2 methods	1.257e-06	1.259e-12	5.286e-14
Zadunaisky	1.024e+00	1.007e-01	1.005e-02
SC	1.461e-02	1.461e-02	1.461e-02
SLDVE	2.1776e-07	2.5542e-12	2.5908e-17

Table 3.29: Accuracy of the global error evaluation techniques for BDF formulae applied to ODE3 on the non-uniform grids.

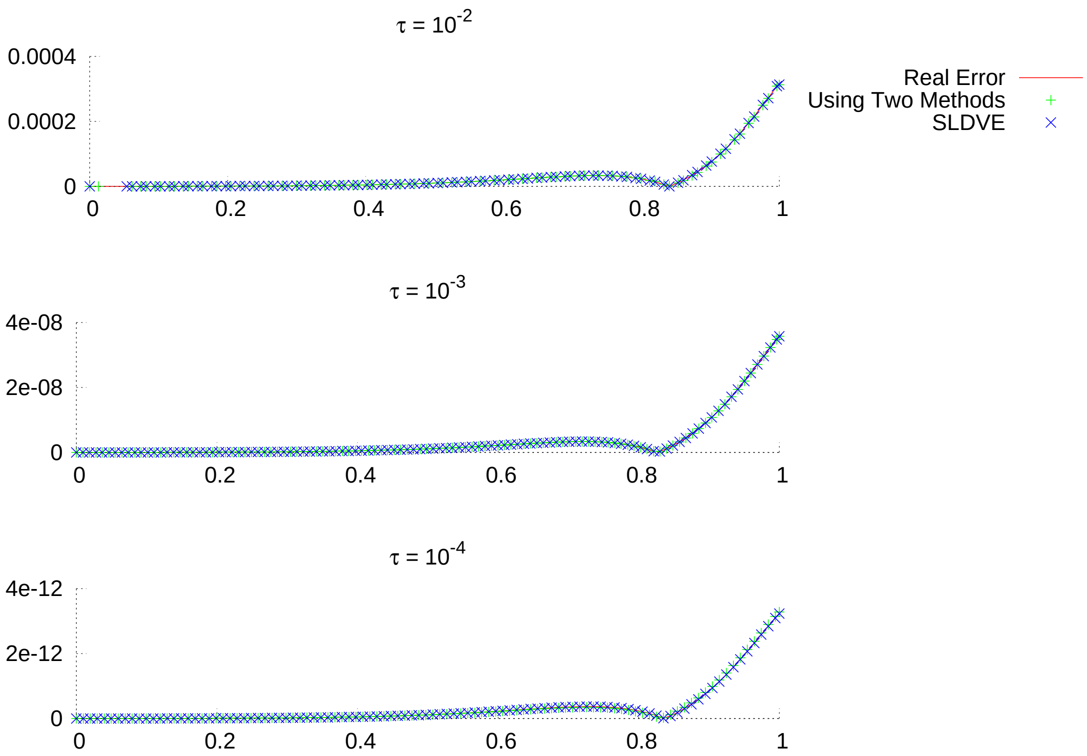

Figure 3.15: The true error and the estimates obtained for BDF formulae on the non-uniform grids, ODE3

	$\tau = 10^{-2}$	$\tau = 10^{-3}$	$\tau = 10^{-4}$
Richardson	–	–	–
Using 2 methods	0.032	0.309	3.045
Zadunaisky	0.065	0.522	5.109
SC	0.068	0.523	5.077
SLDVE	0.024	0.199	1.694

Table 3.30: CPU time (in sec) of the global error evaluation for BDF formulae applied to ODE3 on the non-uniform grids

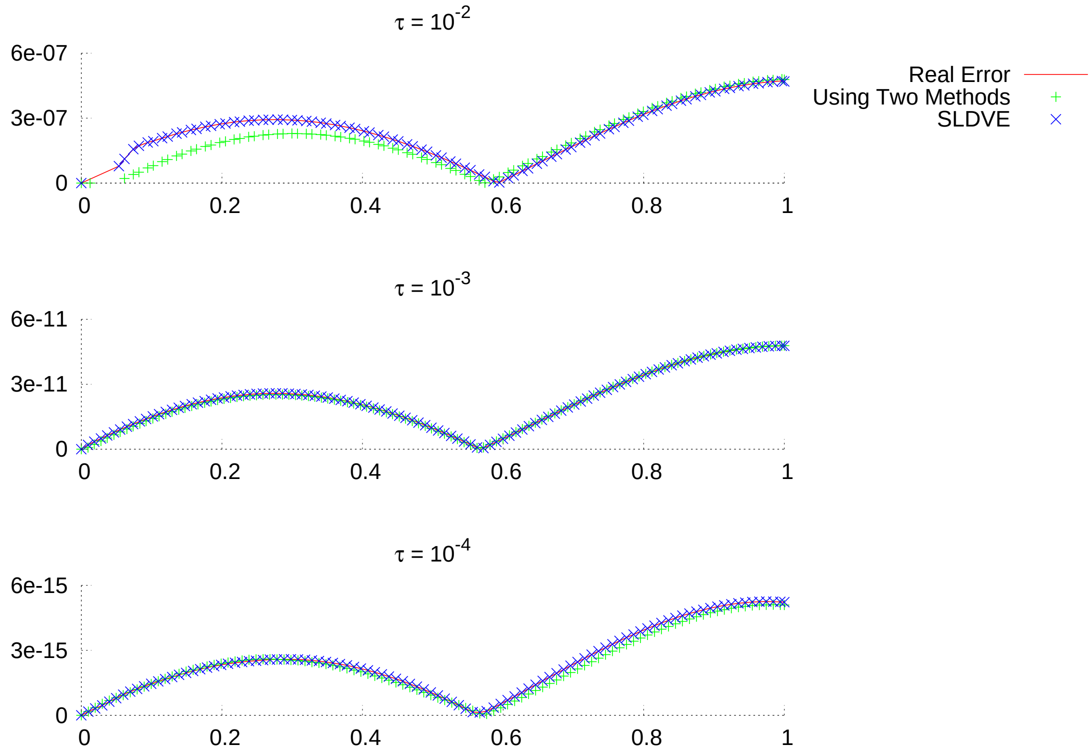

Figure 3.16: The true error and the estimates obtained for BDF formulae on the non-uniform grids, ODE4

for Zadunaisky's technique and solving for the correction were also omitted in this case. Figure 3.16 show that SLDVE estimates the global error correctly. Using two methods can also be considered as a good strategy to estimate the real error in this case for $\tau = 10^{-2}$ and $\tau = 10^{-3}$. For $\tau = 10^{-4}$, Table 3.31 illustrates the advantage of SLDVE.

5. **Comparison:**

   According to the tests conducted on the sample ODEs, SLDVE is a good strategy to evaluate the global error in BDF for solving ODEs on non-uniform grids. The efficiency of the technique is not only shown in its accuracy but also in terms of CPU time. Using two different methods can also be considered as a good technique for global error evaluation in this case. Although this strategy is more expensive than SLDVE in terms of running time, the accuracy of the error estimate were shown to be competitive specially for $\tau = 10^{-2}$ and $\tau = 10^{-3}$. Zadunaisky's technique and solving for the correction provided error estimate with low accuracy.

	$\tau = 10^{-2}$	$\tau = 10^{-3}$	$\tau = 10^{-4}$
Richardson	–	–	–
Using 2 methods	5.150e-10	4.758e-16	3.022e-16
Zadunaisky	1.183e-02	1.187e-03	1.188e-04
SC	1.183e-02	1.187e-03	1.188e-04
SLDVE	3.3471e-10	3.4178e-15	3.4247e-20

Table 3.31: Accuracy of the global error evaluation techniques for BDF formulae applied to ODE4 on the non-uniform grids.

	$\tau = 10^{-2}$	$\tau = 10^{-3}$	$\tau = 10^{-4}$
Richardson	0.040	0.384	3.785
Using 2 methods	0.032	0.309	3.045
Zadunaisky	0.065	0.522	5.109
SC	0.068	0.523	5.077
SLDVE	0.024	0.199	1.694

Table 3.32: CPU time (in sec) of the global error evaluation for BDF formulae applied to ODE4 on the non-uniform grids

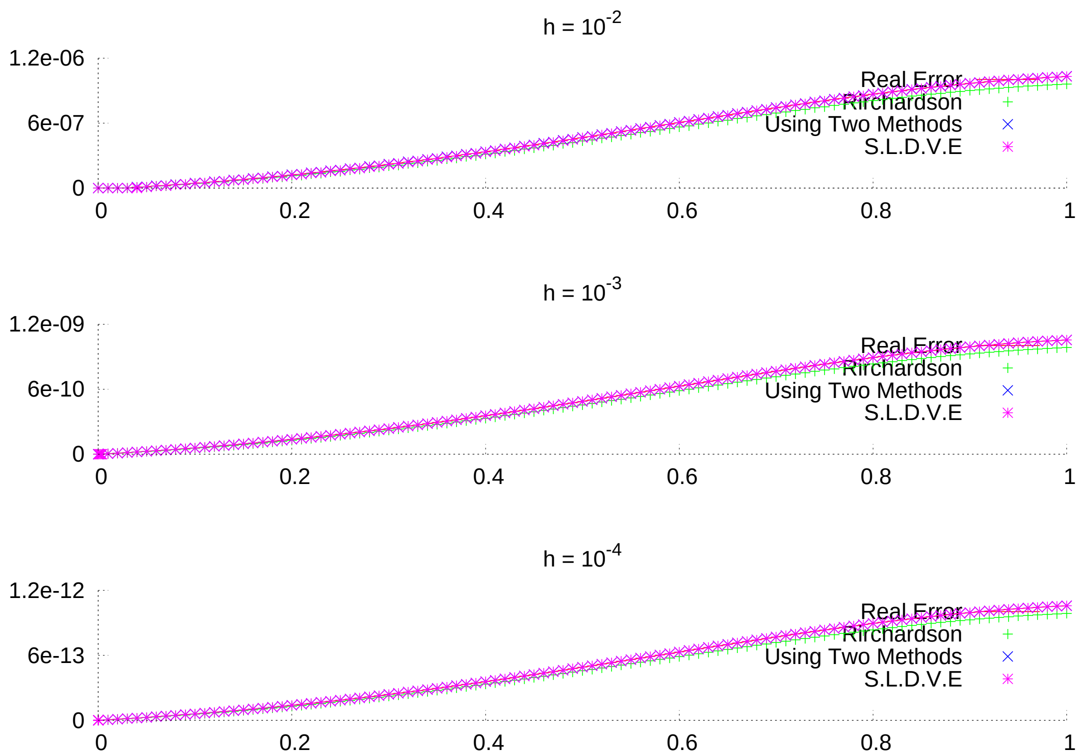

Figure 3.17: The true error and the estimates obtained for Nyström methods on the uniform grids, ODE1

3.3 Numerical Results for Nyström Methods

In this section, we present the numerical results obtained for the different global error evaluation strategies applied to Nyström methods of order 4. We choose this method as example of weakly stable method. The numerical tests were conducted on the same test problems with know solution described in Section 1.1.1.

3.3.1 Implementation on Uniform Grids

Here, we present the numerical results obtained on the uniform grids. The behaviour of the error estimates will be observed on the uniform grids with stepsizes $h = 10^{-2}$, $h = 10^{-3}$ and $h = 10^{-4}$.

1. **Problem ODE1:** The numerical results for ODE1 are given in Table 3.33 and Figure 3.17.

	$h = 10^{-2}$	$h = 10^{-3}$	$h = 10^{-4}$
Richardson	7.056e-08	7.055e-11	7.075e-14
Using 2 methods	6.455e-10	6.637e-15	1.112e-16
Zadunaisky	1.360e-06	6.984e-10	7.141e-13
SC	4.376e-07	6.979e-10	7.141e-13
SLDVE	3.296e-09	3.102e-14	3.070e-19

Table 3.33: Accuracy of the global error evaluation techniques for Nyström methods applied to ODE1 on the uniform grids.

	$h = 10^{-2}$	$h = 10^{-3}$	$h = 10^{-4}$
Richardson	0.041	0.351	3.424
Using 2 methods	0.031	0.282	2.713
Zadunaisky	0.045	0.426	4.225
SC	0.048	0.431	4.244
SLDVE	0.019	0.212	1.949

Table 3.34: CPU time (in sec) of the global error evaluation techniques for Nyström methods applied to ODE1 on the uniform grids

	$h = 10^{-2}$	$h = 10^{-3}$	$h = 10^{-4}$
Richardson	1.020e-06	6.581e-11	6.549e-14
Using 2 methods	1.064e-06	1.169e-14	1.036e-13
Zadunaisky	7.335e-05	9.838e-10	9.868e-13
SC	7.149e-05	7.731e-08	9.868e-13
SLDVE	1.214e-06	1.180e-11	1.152e-16

Table 3.35: Accuracy of the global error evaluation techniques for Nyström methods applied to ODE2 on the uniform grids.

For problem ODE1, Table 3.33 exhibits the advantage that takes of SLDVE over the other global error evaluation strategies under consideration here. The technique produced very accurate error estimate on the three grids with stepsizes $h = 10^{-2}$, $h = 10^{-3}$ and $h = 10^{-4}$. In addition to the high accuracy provided by this technique, the method integrates the equation and evaluates the global error faster than the other strategies that we consider here (See Table 3.34). Using two different methods also provided very accurate estimation of the true error for $h = 10^{-2}$ and $h = 10^{-3}$. However, when the stepsize is reduced to $h = 10^{-4}$, the order of accuracy of the error estimate computed by the using two different methods is also reducing. The technique also is not better than SLDVE in terms of CPU time. Integrating ODE1 the equation and estimating the global error using two different methods last approximately 1.5 times longer than when SLDVE is used. The Richardson extrapolation, Zadunaisky's technique and solving for the correction lead to less accurate error estimate (See Table 3.33). The CPU time for these methods are also not competitive when compared to that of SLDVE and using two different methods.

2. **Problem ODE2:** The numerical results for ODE2 are given in Table 3.35 and Figure 3.18.

   For ODE2, the Richardson extrapolation, using two methods and SLDVE computed an estimation of the global error with similar accuracy for $h = 10^{-2}$ and $h = 10^{-3}$. As the stepsize is reduce to $h = 10^{-4}$, SLDVE provided an error estimate with better accuracy than the Richardson extrapolation and using two methods (See Table 3.35). The performance of SLDVE to estimate the global error in Nyström method applied to ODE2 is also found better in terms of CPU time (See Table 3.34). Zadunaisky's technique and solving the correction lead to an error estimate with low accuracy. These technique of global error evaluation are also shown less efficient in terms of CPU time (See Table 3.34).

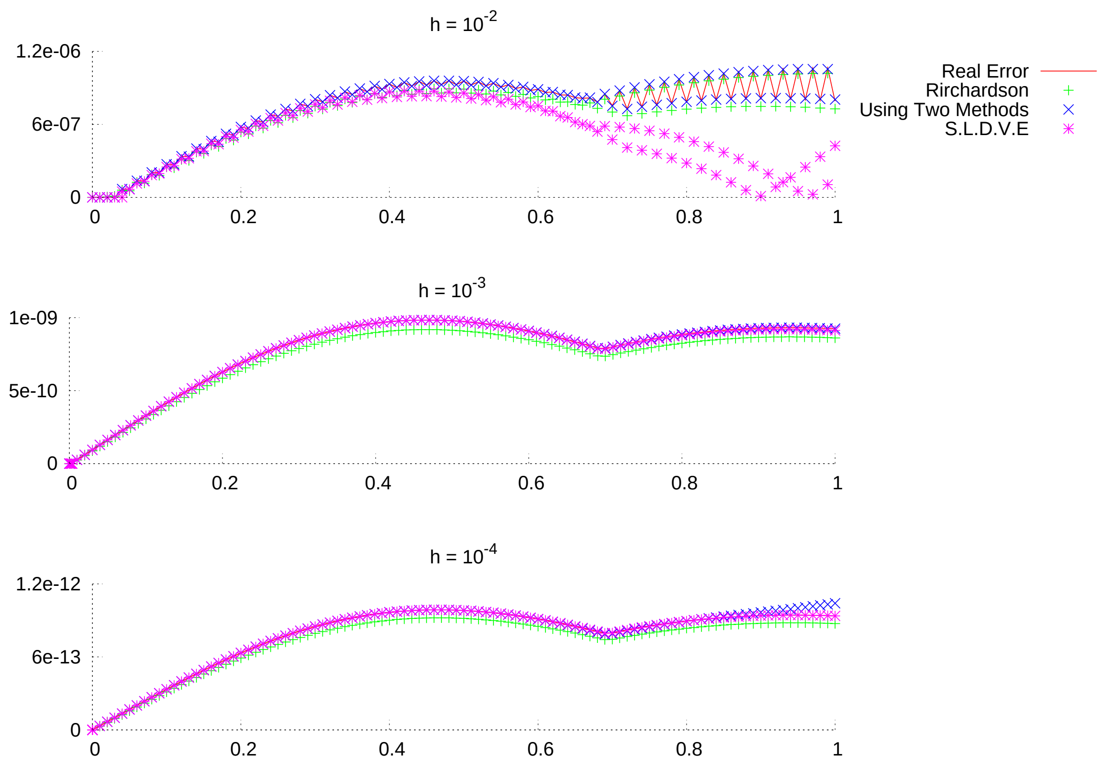

Figure 3.18: The true error and the estimates obtained for Nyström methods on the uniform grids, ODE2

	$h = 10^{-2}$	$h = 10^{-3}$	$h = 10^{-4}$
Richardson	0.067	0.648	6.456
Using 2 methods	0.057	0.491	5.290
Zadunaisky	0.072	0.706	7.149
SC	0.074	0.704	7.025
SLDVE	0.037	0.334	3.418

Table 3.36: CPU time (in sec) of the global error evaluation techniques for Nyström methods applied to ODE2 on the uniform grids

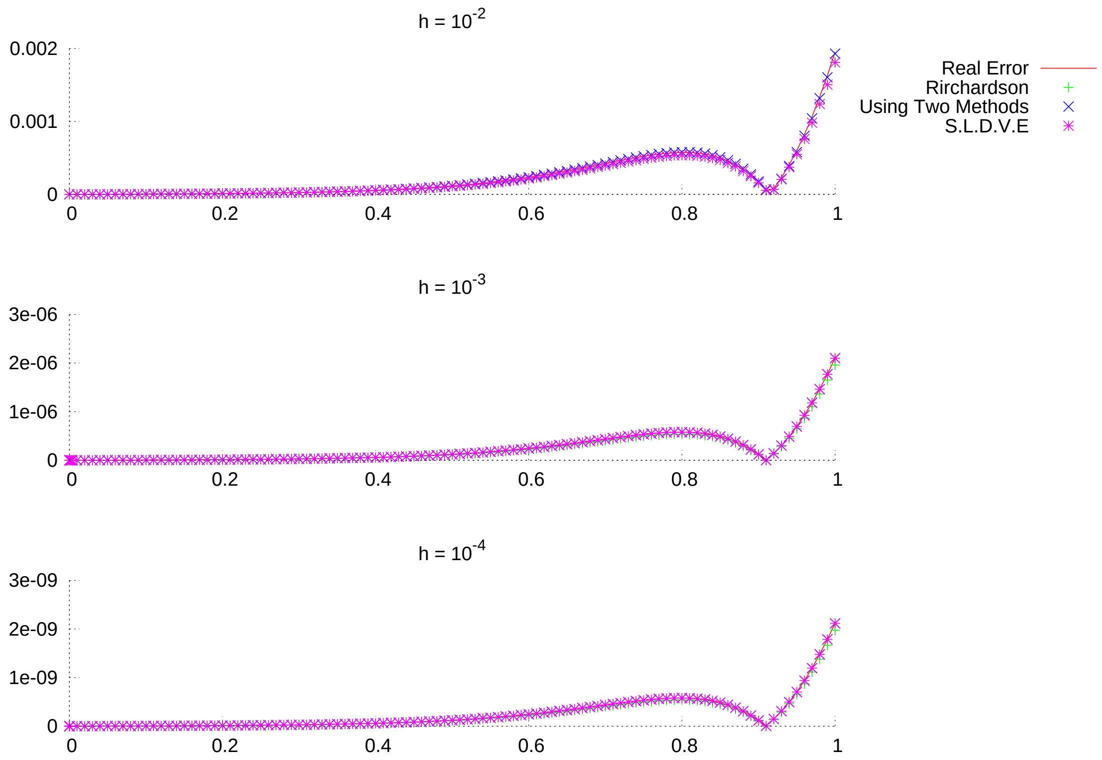

Figure 3.19: The true error and the estimates obtained for Nyström methods on the uniform grids, ODE3

3. **Problem ODE3:** The numerical results for ODE3 are given in Table 3.37 and Figure 3.19.

   In Figure 3.19, the graph for Zadunaisky's technique and solving for the correction are not plotted because of the low accuracy (See Table 3.37. SLDVE and using two methods provided an error estimate similarly accurate for the three stepsizes $h = 10^{-2}$, $h = 10^{-3}$ and $h = 10^{-4}$ (See Table 3.37). However, the performance of these two global error evaluation strategies differs significantly in terms of CPU time. SLDVE integrates the equation ODE3 and calculate the estimation of the error faster than the using two corrections. The Richardson extrapolation, Zadunaisky's technique and solving for the correction also estimated the true error with similar accuracy. These methods are shown to be less efficient than SLDVE and using two methods in terms of accuracy and CPU time when the are used to estimate the global error in the Nystrm̈ method applied to ODE3.

4. **Problem ODE4:** The numerical results for ODE4 are given in Table 3.39 and Figure 3.20.

   ODE4 was chosen as a sample of stiff ODE, and the Nyström method of order 4 which is

	$h = 10^{-2}$	$h = 10^{-3}$	$h = 10^{-4}$
Richardson	1.411e-04	1.411e-07	1.412e-10
Using 2 methods	2.050e-05	2.028e-10	3.139e-14
Zadunaisky	1.855e-03	1.852e-06	1.831e-09
SC	1.951e-03	2.101e-06	2.116e-09
SLDVE	9.931e-05	1.032e-09	1.036e-14

Table 3.37: Accuracy of the global error evaluation techniques for Nyström methods applied to ODE3 on the uniform grids.

	$h = 10^{-2}$	$h = 10^{-3}$	$h = 10^{-4}$
Richardson	0.073	0.706	7.150
Using 2 methods	0.057	0.542	5.745
Zadunaisky	0.075	0.752	7.671
SC	0.142	1.374	13.696
SLDVE	0.041	0.359	3.691

Table 3.38: CPU time (in sec) of the global error evaluation techniques for Nyström methods applied to ODE3 on the uniform grids

	$h = 10^{-2}$	$h = 10^{-3}$	$h = 10^{-4}$
Richardson	1.344e-06	7.314e-10	7.318e-13
Using 2 methods	6.639e-06	6.027e-10	1.336e-12
Zadunaisky	4.383e-03	5.551e-06	5.684e-09
SC	4.245e-03	5.538e-06	5.683e-09
SLDVE	1.697e-05	1.685e-09	1.685e-13

Table 3.39: Accuracy of the global error evaluation techniques for Nyström methods applied to ODE4 on the uniform grids.

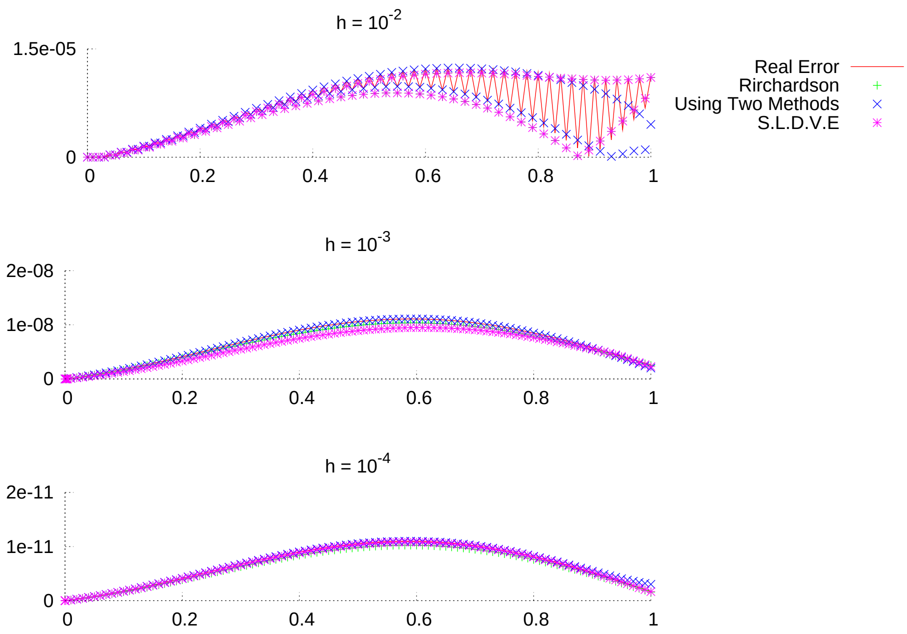

Figure 3.20: The true error and the estimates obtained for Nyström methods on the uniform grids, ODE4

	$h = 10^{-2}$	$h = 10^{-3}$	$h = 10^{-4}$
Richardson	0.037	0.351	3.520
Using 2 methods	0.030	0.280	2.792
Zadunaisky	0.046	0.435	4.285
SC	0.046	0.431	4.235
SLDVE	0.021	0.223	1.999

Table 3.40: CPU time (in sec) of the global error evaluation techniques for Nyström methods applied to ODE4 on the uniform grids

a weakly stable method is not a good method to solve ODE4 numerically. However, Figure 3.20 shows that Richardson extrapolation, using two methods and SLDVE tend to evaluate the global error more accurately when the stepsize is small. We omitted the Zadunaisky's technique and solving for the correction from the graph because of the low accuracy of the error estimate (See Table 3.39). The CPU time of the SLDVE shows that the method is more efficient than Richardson extrapolation and using two methods (See Table 3.40).

5. **Comparison**

   According to the numerical tests conducted on the different global error evaluation strategies, Richardson extrapolation, using two methods and SLDVE exhibit the same advantages on Zadunaisky's technique and solving for the correction in terms of accuracy. The figures show that the true error and the estimates provided by Richardson extrapolation, using two methods and SLDVE agree very well, except for the using two different methods applied to the stiff problem ODE4 on a grid with large stepsize $h = 10^{-2}$. In addition to the accuracy of the error estimate, SLDVE is more efficient in terms of CPU time.

3.3.2 Implementation on Non-uniform Grids

In this section, we present the results obtained for the Nyström method on the non-uniform grids. The tests were conducted on the samples of ODEs described in Section 1.1.1 and the grids are as described in 3.1.

1. **Problem ODE1:** The numerical results obtained for ODE1 are given in Table 3.41 and Figure 3.21. Figure 3.21 exhibits the ability of the using two methods to estimate the global error for Nyström methods applied to ODE1 on the non-uniform grids. For this test problem, the approximate solution is accurate of order 2 only. However, the error estimate provided by using two methods is accurate of order 4 (See Table 3.41). SLDVE and the Zadunaisky technique provided an estimate of the error accurate of order 2. The graph for solving for the correction is not presented in Figure 3.21 because of the lack of accuracy in the error estimate (See Table 3.41). In terms of CPU time, using two different methods and SLDVE present the same advantage over the other methods considered in this disseratation. Solving problem ODE1 and estimating the true error take long when the Zadunaisky technique or the solving for the correction is used as global error evaluation strategies (See 3.42).

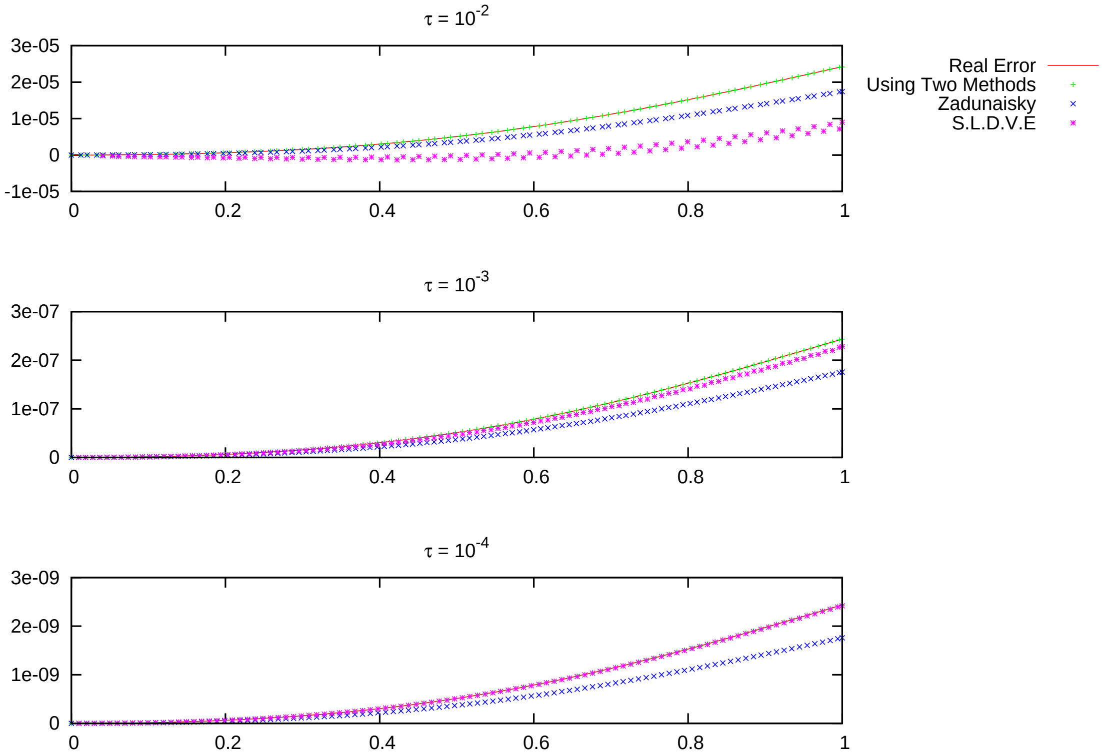

Figure 3.21: The true error and the estimates obtained for Nyström methods applied to ODE1 on the non-uniform grids

	$\tau = 10^{-2}$	$\tau = 10^{-3}$	$\tau = 10^{-4}$
Richardson	–	–	–
Using 2 methods	2.247e-08	2.756e-13	2.840e-17
Zadunaisky	1.741e-05	1.756e-07	1.758e-09
SC	6.943e-01	7.189e-01	7.214e-01
SLDVE	8.985e-06	2.281e-07	2.419e-09

Table 3.41: Accuracy of the global error evaluation techniques for Nyström methods applied to ODE1 on the non-uniform grids.

	$\tau = 10^{-2}$	$\tau = 10^{-3}$	$\tau = 10^{-4}$
Richardson	–	–	–
Using 2 methods	0.026	0.230	2.265
Zadunaisky	0.044	0.401	3.906
SC	0.042	0.397	4.192
SLDVE	0.024	0.268	2.099

Table 3.42: CPU time (in sec) of the global error evaluation for Nyström methods applied to ODE1 on the non-uniform grids

	$\tau = 10^{-2}$	$\tau = 10^{-3}$	$\tau = 10^{-4}$
Richardson	–	–	–
Using 2 methods	1.796e-06	4.592e-10	4.593e-13
Zadunaisky	1.453e-05	1.397e-07	1.391e-09
SC	6.943e-01	7.189e-01	7.214e-01
SLDVE	4.433e-08	3.853e-12	3.798e-16

Table 3.43: Accuracy of the global error evaluation techniques for Nyström methods applied to ODE2 on the non-uniform grids.

2. **Problem ODE2:** The numerical results obtained for ODE2 are given in Table 3.43 and Figure 3.22. The error estimate produced by the solving for the correction was not plotted in Figure 3.22 as it differs significantly from the real error. The accuracy of the error estimate is given in Table 3.43. According to Figure 3.22, using two methods and SLDVE provided very accurate estimation of the global error. However, Table 3.43 shows that the error estimate computed using SLDVE are the most accurate. In addition, SLDVE runs faster than using two methods (See Table 3.44).

3. **Problem ODE3:** The numerical results obtained for ODE3 are given in Table 3.45 and Figure 3.23.

   The error estimate produced by the solving for the correction was not plotted in Figure 3.23 as it differs significantly from the real error as its accuracy shows in Table 3.45. In figure 3.23, the exact error coincide with the error estimate computed using two methods and SLDVE. Table 3.45 shows that the two strategies estimated the global error with similar

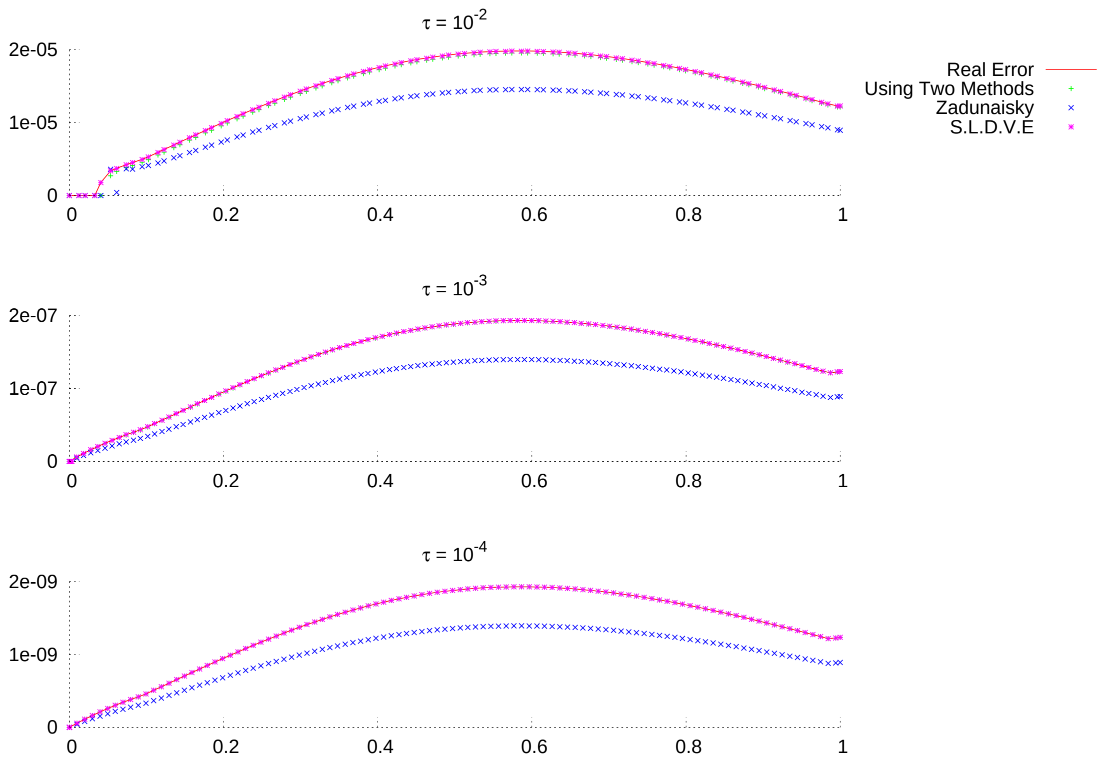

Figure 3.22: The true error and the estimates obtained for Nyström methods applied to ODE2 on the non-uniform grids

	$\tau = 10^{-2}$	$\tau = 10^{-3}$	$\tau = 10^{-4}$
Richardson	–	–	–
Using 2 methods	0.042	0.385	3.758
Zadunaisky	0.064	0.622	6.264
SC	0.066	0.617	6.281
SLDVE	0.036	0.381	3.242

Table 3.44: CPU time (in sec) of the global error evaluation for Nyström methods applied to ODE2 on the non-uniform grids

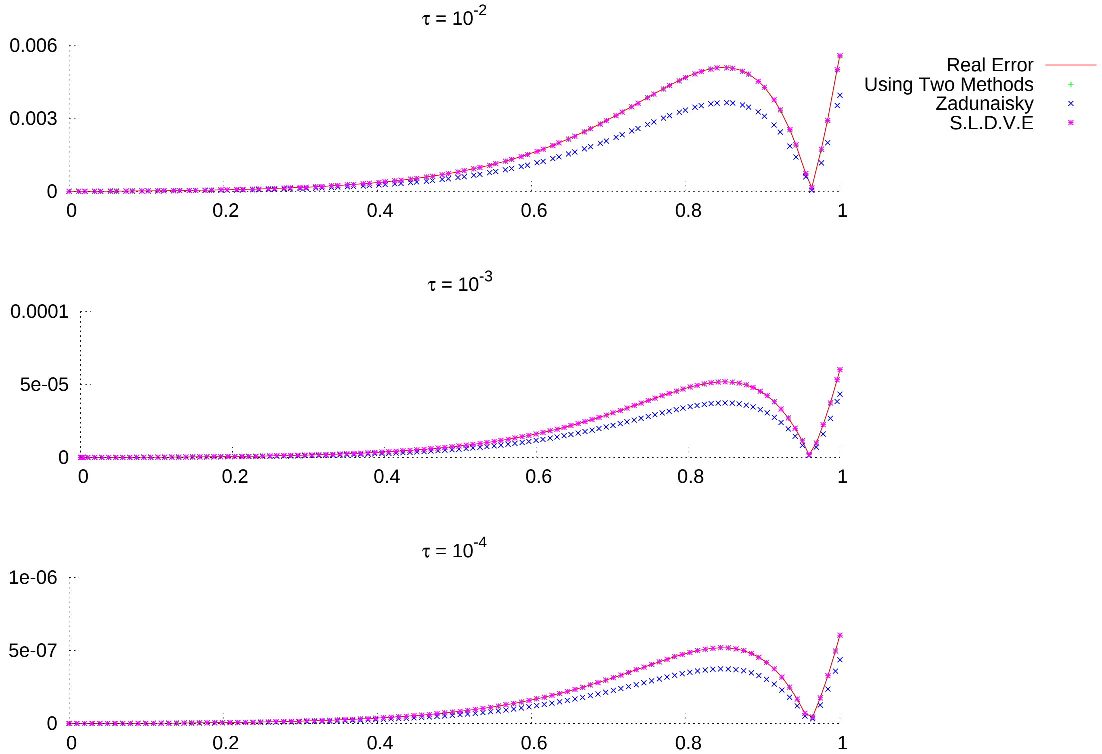

Figure 3.23: The true error and the estimates obtained for Nyström methods applied to ODE3 on the non-uniform grids

accuracy. However, Table 3.46 exhibits the advantage of SLDVE over using two different methods in terms of CPU time. The Zadunaisky's technique is shown to be less accurate than using two methods and SLDVE (see Table 3.45) and is expensive in terms of computation time (see Table 3.46).

4. **Problem ODE4:** The numerical results obtained for ODE4 are given in Table 3.47 and Figure 3.24.

   In this case also, the error estimate computed using solving for the correction was not plotted in Figure 3.24 because of the lack of accuracy (See Table 3.47). In Figure 3.24, the real error, the error estimate produced by using two methods and SLVDE coincide. However, Table 3.47 shows that the SLDVE is more accurate than using two different methods. The advantage of SLDVE is also show in Table 3.48 in terms of CPU time.

5. **Comparison**

	$\tau = 10^{-2}$	$\tau = 10^{-3}$	$\tau = 10^{-4}$
Richardson	–	–	–
Using 2 methods	5.571e-05	6.277e-09	6.354e-13
Zadunaisky	3.946e-03	4.329e-05	4.367e-07
SC	1.778e+05	1.671e+05	1.659e+05
SLDVE	3.026e-05	2.652e-09	3.617e-13

Table 3.45: Accuracy of the global error evaluation techniques for Nyström methods applied to ODE3 on the non-uniform grids.

	$\tau = 10^{-2}$	$\tau = 10^{-3}$	$\tau = 10^{-4}$
Richardson	–	–	–
Using 2 methods	0.042	0.405	4.074
Zadunaisky	0.403	0.680	6.699
SC	0.063	0.645	6.480
SLDVE	0.039	0.399	3.454

Table 3.46: CPU time (in sec) of the global error evaluation for Nyström methods applied to ODE3 on the non-uniform grids

	$\tau = 10^{-2}$	$\tau = 10^{-3}$	$\tau = 10^{-4}$
Richardson	–	–	–
Using 2 methods	2.444e-05	5.875e-09	5.885e-12
Zadunaisky	8.298e-05	7.585e-07	7.533e-09
SC	1.151e+00	1.144e+00	1.143e+00
SLDVE	6.305e-07	6.201e-11	6.201e-15

Table 3.47: Accuracy of the global error evaluation techniques for Nyström methods applied to ODE4 on the non-uniform grids.

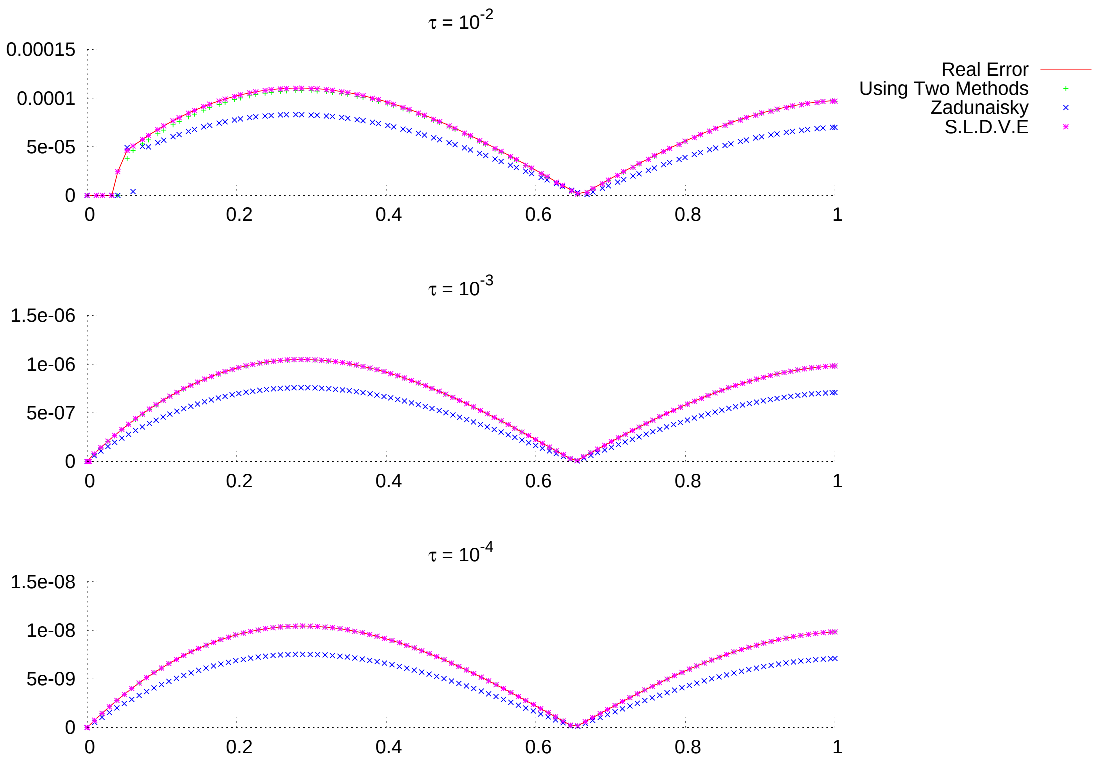

Figure 3.24: The true error and the estimates obtained for Nyström methods applied to ODE1 on the non-uniform grids

	$\tau = 10^{-2}$	$\tau = 10^{-3}$	$\tau = 10^{-4}$
Richardson	–	–	–
Using 2 methods	0.026	0.230	2.295
Zadunaisky	0.045	0.402	3.951
SC	0.045	0.396	3.969
SLDVE	0.026	0.251	2.119

Table 3.48: CPU time (in sec) of the global error evaluation for Nyström methods applied to ODE4 on the non-uniform grids

The numerical tests conducted on the non-uniform grids exhibit the advantage of SLDVE over the other methods considered in this work in terms of accuracy and computation time. Using two different methods also produced interesting results for some cases. According to the same numerical test, solving for the correction is not a good strategie to estimate the exact error of Nyström methods on non-uniform grids.

Chapter 4

Numerical Results for Index 1 DAEs

Numerical results for ODEs were presented in Chapter 3. In this chapter, we will present the numerical results for index 1 DAEs. The numerical tests were conducted on the test problems with known solution described in Section 1.2.1.

4.1 Numerical Results for Adams methods

In this section, we present and compare the numerical results obtained for different global error evaluation strategies when they are applied to Adams methods for solving semi-explicit index 1 DAE.

4.1.1 Implementation on the Uniform Grids

1. **Problem DAE1:** The numerical results for the Adams methods on the uniform grids are given in Figure 4.1 and the CPU time of the different global error evaluation strategies are given in Table 4.2. In Figure 4.1, we omitted the graph of the error estimations provided by the Zadunaisky's technique and solving for the correction as the behaviour of the true error differs significantly from the error estimates generated by these technique. The figures shows the ability of SLDVE to estimate to true error when the Adams method of order 4 is used to solve the problem DAE1. For the three different stepsizes, the error estimate and the true error agree very well. The performance of SLDVE is also satisfactory in terms of CPU

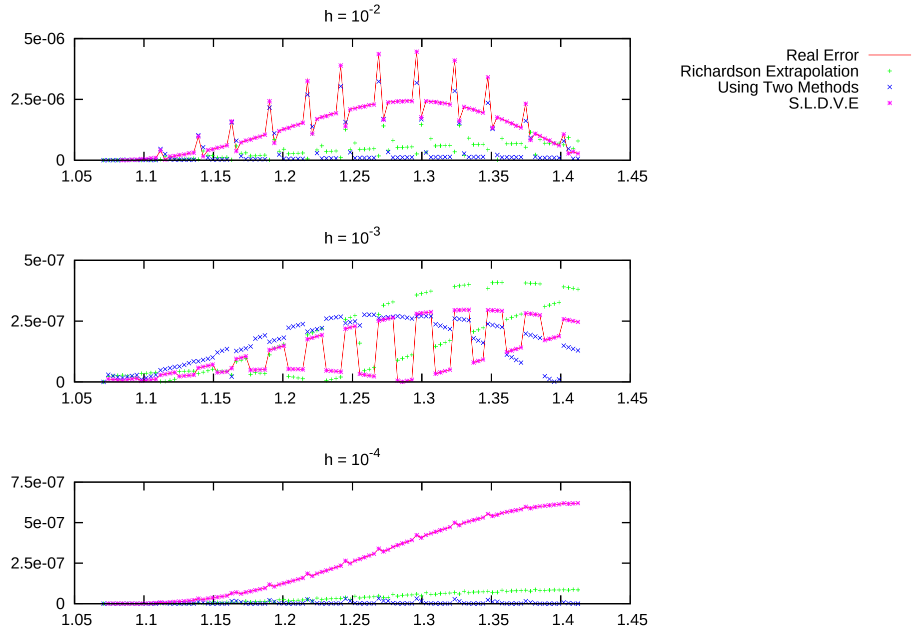

Figure 4.1: The true error and its estimates obtained for Adams methods applied to DAE1 on the uniform grids

time when the method compared with the global error evaluation strategies that we consider in this work. Table 4.2 shows that SLDVE and the using two methods are the least time consuming among the methods under consideration here.

2. **Problem DAE2:** Very similar results were obtained when the same Adams method was used to solve the stiff equation DAE2. Figure 4.2 shows that only SLDVE was able to approximate the global error correctly ; and Table 4.2 exhibits the advantage of SLDVE and using two methods in terms of CPU time.

4.1.2 Implementation on the Non-uniform Grids

1. **Problem DAE1:**

   On the non-uniform grids, we recall that the Richardson extrapolation is not applicable. The good performance of SLDVE can be seen in Figure 4.3. In this case also, the graph of the error estimate generated by the Zadunaiskys technique and the solving for the correction were omitted because of the lack of accuracy. For the Zadunaiskys technique, the error in

	$h = 10^{-2}$	$h = 10^{-3}$	$h = 10^{-4}$
Richardson	2.332e-06	4.797e-07	7.508e-07
Using 2 methods	9.037e-07	4.797e-07	7.508e-07
Zadunaisky	2.332e-06	4.797e-07	5.430e-07
SC	1.229e+00	9.531e+00	9.431e+00
SLDVE	2.402e-08	3.437e-13	9.795e-17

Table 4.1: Accuracy of the global error evaluation techniques applied to DAE1 on the uniform grids.

	$h = 10^{-2}$	$h = 10^{-3}$	$h = 10^{-4}$
Richardson	0.060	0.540	5.356
Using 2 methods	0.036	0.284	3.041
Zadunaisky	0.088	0.804	8.101
SC	0.084	0.824	8.133
SLDVE	0.037	0.357	3.028

Table 4.2: CPU time (in sec) of the global error evaluation applied to DAE1 on the uniform grids

	$h = 10^{-2}$	$h = 10^{-3}$	$h = 10^{-4}$
Richardson	1.865e-12	2.132e-16	8.133e-17
Using 2 methods	6.974e-11	2.347e-16	5.719e-17
Zadunaisky	8.568e-10	8.426e-15	5.234e-17
SC	1.569e+00	1.500e+00	1.500e+00
SLDVE	8.119e-09	4.926e-14	4.911e-19

Table 4.3: Accuracy of the global error evaluation techniques applied to DAE2 on the uniform grids.

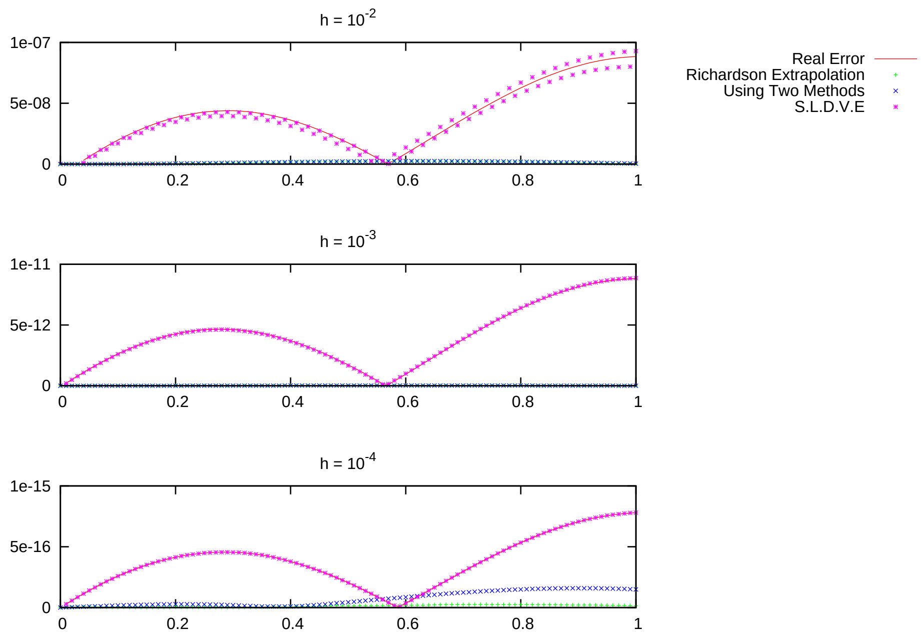

Figure 4.2: The true error and its estimates obtained for Adams methods applied to DAE2 on the uniform grids

	$h = 10^{-2}$	$h = 10^{-3}$	$h = 10^{-4}$
Richardson	0.108	0.972	9.593
Using 2 methods	0.084	0.736	7.416
Zadunaisky	0.104	1.076	10.621
SC	0.160	1.588	13.097
SLDVE	0.052	0.431	4.012

Table 4.4: CPU time (in sec) of the global error evaluation applied to DAE2 on the uniform grids

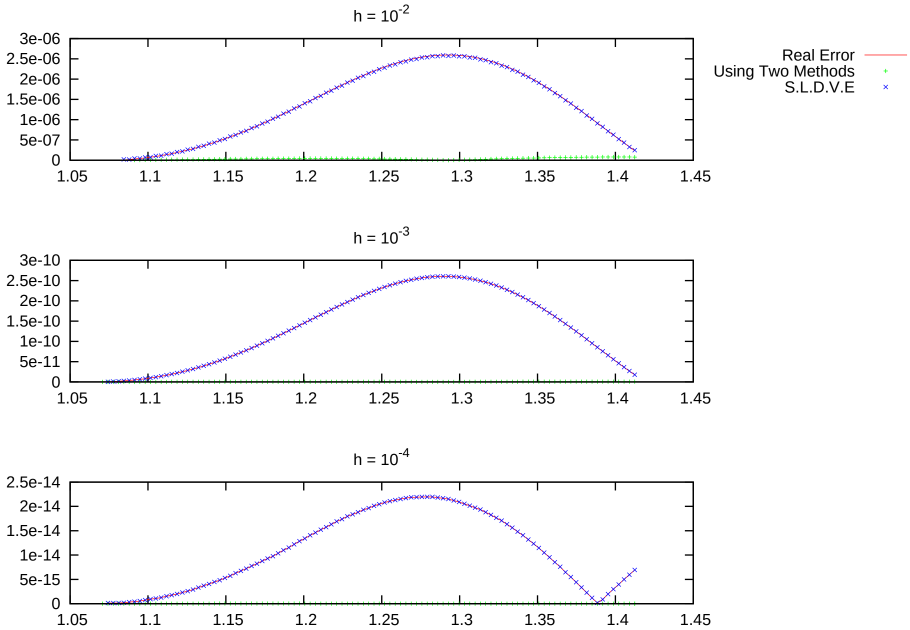

Figure 4.3: The true error and its estimates obtained for Adams methods applied to DAE1 on the non-uniform grids

the estimate is up to 226.8403 for $\tau = 10^{-2}$, and for the solving for the correction the error raised up to $1.0107e+02$. In terms of CPU time, SLDVE presents an advantage over the methods applicable on non-uniform grids (See Table 4.6) .

2. **Problem DAE2:** The behaviour of the true error and its estimates are very similar when the Adams method is applied to the stiff problem DAE2 on the non-uniform grids (See Table 4.8 and Figure 4.4).

4.2 Numerical Results for BDF formulae

4.2.1 Implementation on the Uniform Grids

1. **Problem DAE1:**

   Numerical results for the BDF formula of order 4 on the uniform grids are given in Figure 4.5 and Table 4.10. In Figure 4.5, we notice the advantage of SLDVE compared to the Richardson extrapolation and the using two methods. The true error and the error estimate

	$h = 10^{-2}$	$h = 10^{-3}$	$h = 10^{-4}$
Richardson	–	–	–
Using 2 methods	7.902e-08	9.783e-15	9.117e-15
Zadunaisky	2.268e+02	1.436e-01	1.400e-02
SC	1.010e+00	9.531e+01	9.531e+01
SLDVE	2.402e-08	1.479e-13	1.763e-18

Table 4.5: Accuracy of the global error evaluation techniques applied to DAE1 on the non-uniform grids.

	$\tau = 10^{-2}$	$\tau = 10^{-3}$	$\tau = 10^{-4}$
Richardson	–	–	–
Using 2 methods	0.056	0.544	5.880
Zadunaisky	0.088	0.736	8.013
SC	0.084	0.800	8.049
SLDVE	0.040	0.312	3.416

Table 4.6: CPU time (in sec) of the global error evaluation applied to DAE1 on the non-uniform grids

	$h = 10^{-2}$	$h = 10^{-3}$	$h = 10^{-4}$
Richardson	–	–	–
Using 2 methods	3.045e-09	3.015e-14	2.927e-18
Zadunaisky	2.268e+02	1.430e-01	1.400e-02
SC	1.571e+00	1.000e-03	1.876e-04
SLDVE	1.050e-06	1.213e-10	1.231e-14

Table 4.7: Accuracy of the global error evaluation techniques applied to DAE2 on the non-uniform grids.

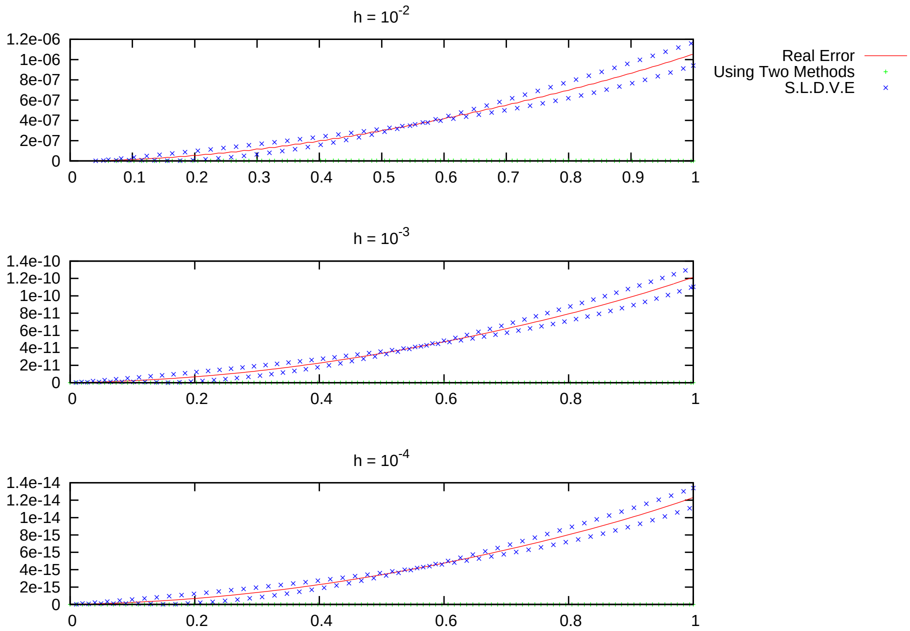

Figure 4.4: The true error and its estimates obtained for Adams methods applied to DAE2 on the non-uniform grids

	$\tau = 10^{-2}$	$\tau = 10^{-3}$	$\tau = 10^{-4}$
Richardson	–	–	–
Using 2 methods	0.080	0.756	7.928
Zadunaisky	0.164	1.624	15.981
SC	0.156	1.536	15.541
SLDVE	0.052	0.432	5.024

Table 4.8: CPU time (in sec) of the global error evaluation applied to DAE1 on the non-uniform grids

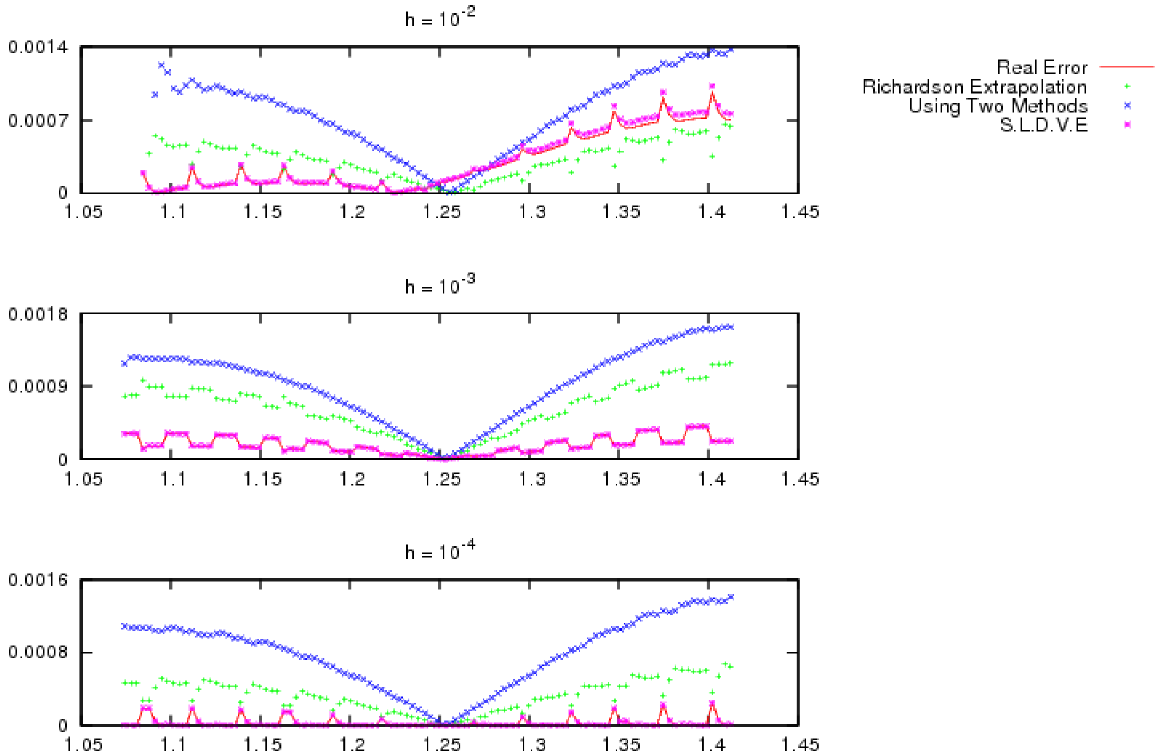

Figure 4.5: The true error and its estimates obtained for BDF formulae applied to DAE1 on the uniform grids

generated by SLDVE agree very well for all the stepsizes. The shape of the true error is conserved by the Richardson extrapolation and the using two methods, however the error estimate generated by these methods agree with the true error only when the later is sufficiently small. Table 4.10 exhibits the good performance of SLDVE in term of CPU time. In addition to the lack of accuracy in the error estimate, the Zadunaiskys technique and solving for the correction are computationally expensive when compared to SLDVE and using two methods (See Table 4.10).

2. **Problem DAE2:** For problem DAE2, Figure 4.6 shows that only the error estimate computed using SLDVE agrees with the true error. For the solving for the correction, the error in the estimate raised up to 2.0114 for $h = 10^{-2}$. For the Zadunaiskys technique, the differ-

	$h=10^{-2}$	$h=10^{-3}$	$h=10^{-4}$
Richardson	2.332e-06	4.797e-07	7.508e-07
Using 2 methods	6.316e-04	7.116e-04	6.413e-04
Zadunaisky	6.316e-04	1.436e-01	9.529e+01
SC	1.010e+02	1.436e-01	9.529e+01
SLDVE	2.402e-08	3.437e-13	9.795e-17

Table 4.9: Accuracy of the global error evaluation techniques applied to DAE1 on the uniform grids.

	$h=10^{-2}$	$h=10^{-3}$	$h=10^{-4}$
Richardson	0.100	1.020	10.129
Using 2 methods	0.080	0.708	7.332
Zadunaisky	0.108	1.068	10.677
SC	0.104	1.024	10.793
SLDVE	0.032	0.508	4.644

Table 4.10: CPU time (in sec) of the global error evaluation applied to DAE1 on the uniform grids

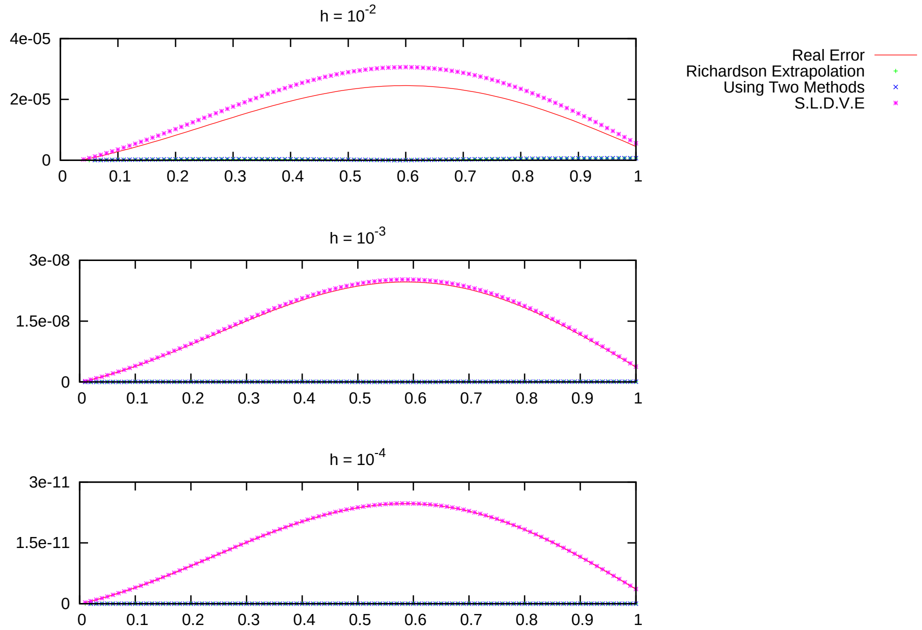

Figure 4.6: The true error and its estimates obtained for BDF formulae applied to DAE2 on the uniform grids

ence between the true error and the error estimate raised up to $1.010e+02$. The advantage of SLDVE is also shown by Table 4.12 which gives the CPU time of the different global error evaluation strategies when they are applied to the BDF formulae to solve DAE2 on the uniform grids.

4.2.2 Implementation on the Non-uniform Grids

1. **Problem DAE1:**

   On the non-uniform grids, SLDVE and using to methods approximate the true error correctly. The behaviour of the true error and the error estimate calculated by these two methods agree very well specially when τ is small (See Figure 4.7). When the Zadunaiskys technique is used, the true error differs significantly from is estimate. Namely, the difference between the true error and its estimate produced by the Zadunaiskys technique raised up to 99.62687 for $\tau = 10^{-2}$. Similar results were obtained for the solving for the correction for which the error in the approximate values of the true error was 99.6269 when $\tau = 10^{-2}$. In terms of CPU time, Table 4.14 shows the advantage of SLDVE over the methods that we consider in this

	$h = 10^{-2}$	$h = 10^{-3}$	$h = 10^{-4}$
Richardson	6.690e-07	6.678e-11	7.659e-15
Using 2 methods	6.690e-07	6.678e-11	7.659e-15
Zadunaisky	1.569e+00	1.506e-03	1.500e-04
SC	1.010e+02	1.436e-01	9.529e+01
SLDVE	6.047e-06	5.935e-10	5.925e-14

Table 4.11: Accuracy of the global error evaluation techniques applied to DAE2 on the uniform grids.

	$h = 10^{-2}$	$h = 10^{-3}$	$h = 10^{-4}$
Richardson	0.072	0.708	7.268
Using 2 methods	0.056	0.548	5.508
Zadunaisky	0.080	0.808	8.249
SC	0.088	0.804	7.993
SLDVE	0.040	0.332	3.768

Table 4.12: CPU time (in sec) of the global error evaluation applied to DAE2 on the uniform grids

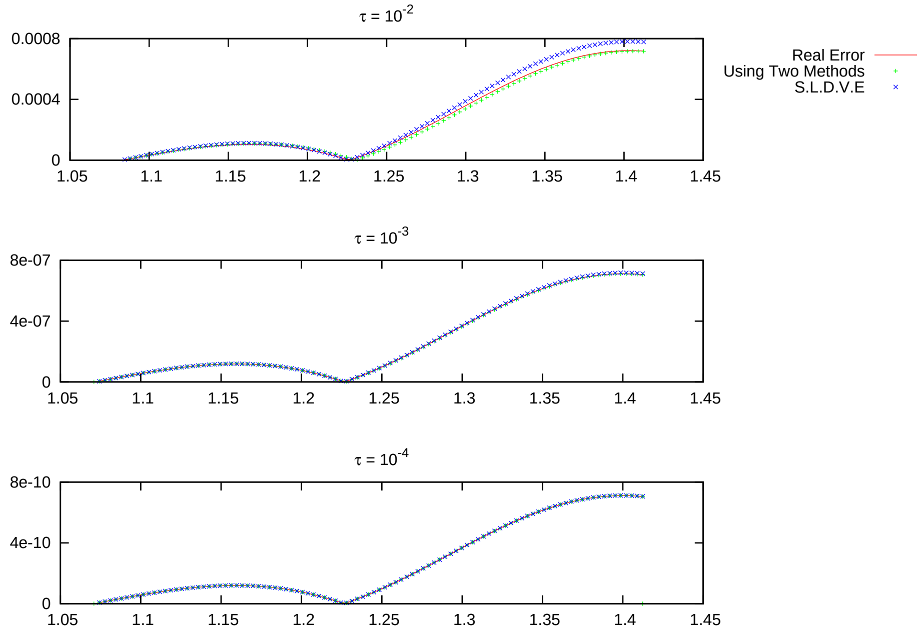

Figure 4.7: The true error and its estimates obtained for BDF formulae applied to DAE1 on the non-uniform grids

dissertations.

2. **Problem DAE2:** Similar results were obtained when the global error evaluation strategies are applied to the BDF formulae of order 4 to solve the stiff problem DAE2 on the non-uniform grids (See Table 4.16 and Figure 4.8). According to Figure 4.8, the true error and the estimates computed using SLDVE or using two methods have the same behaviour, specially when τ is small. These methods also present the same advantage in terms of CPU time (See Table 4.16).

4.3 Numerical Results for Nystöm methods

4.3.1 Implementation on the Non-uniform Grids

1. **Problem DAE1:** As shown in Figure 4.9, the numerical results for the Nyström methods are unusual. For all the values of τ, the true error as well as its estimate calculated by the different global error evaluation strategies are reduced to 0.

	$h = 10^{-2}$	$h = 10^{-3}$	$h = 10^{-4}$
Richardson	–	–	–
Using 2 methods	7.213e-04	7.121e-07	1.963e-13
Zadunaisky	9.962e+01	1.436e-01	1.427e-02
SC	9.962e+01	1.436e-01	9.529e+01
SLDVE	5.930e-05	5.838e-09	5.830e-13

Table 4.13: Accuracy of the global error evaluation techniques applied to DAE1 on the non-uniform grids.

	$\tau = 10^{-2}$	$\tau = 10^{-3}$	$\tau = 10^{-4}$
Richardson	–	–	–
Using 2 methods	0.056	0.588	5.796
Zadunaisky	0.096	0.912	8.893
SC	0.088	0.932	8.793
SLDVE	0.044	0.464	4.772

Table 4.14: CPU time (in sec) of the global error evaluation applied to DAE1 on the non-uniform grids

	$h = 10^{-2}$	$h = 10^{-3}$	$h = 10^{-4}$
Richardson	–	–	–
Using 2 methods	2.640e-05	2.653e-08	2.655e-11
Zadunaisky	1.556e+00	1.885e-03	1.876e-04
SC	1.556e+00	1.885e-03	1.876e-04
SLDVE	6.785e-06	6.655e-10	6.644e-14

Table 4.15: Accuracy of the global error evaluation techniques applied to DAE2 on the non-uniform grids.

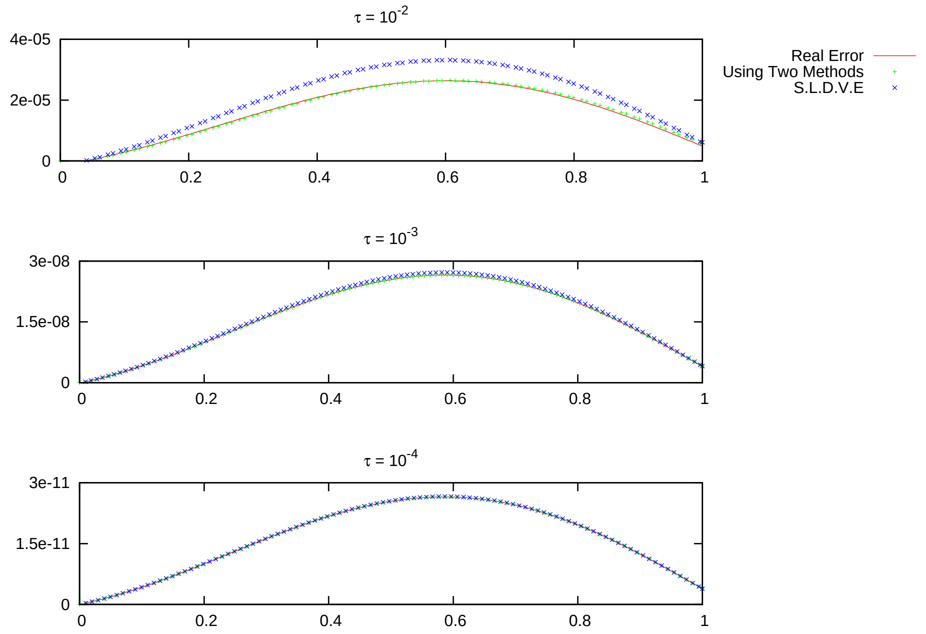

Figure 4.8: The true error and its estimates obtained for BDF formulae applied to DAE2 on the non-uniform grids

	$\tau = 10^{-2}$	$\tau = 10^{-3}$	$\tau = 10^{-4}$
Richardson	–	–	–
Using 2 methods	0.048	0.436	4.528
Zadunaisky	0.072	0.676	6.640
SC	0.072	0.648	7.180
SLDVE	0.040	0.348	3.672

Table 4.16: CPU time (in sec) of the global error evaluation applied to DAE2 on the non-uniform grids

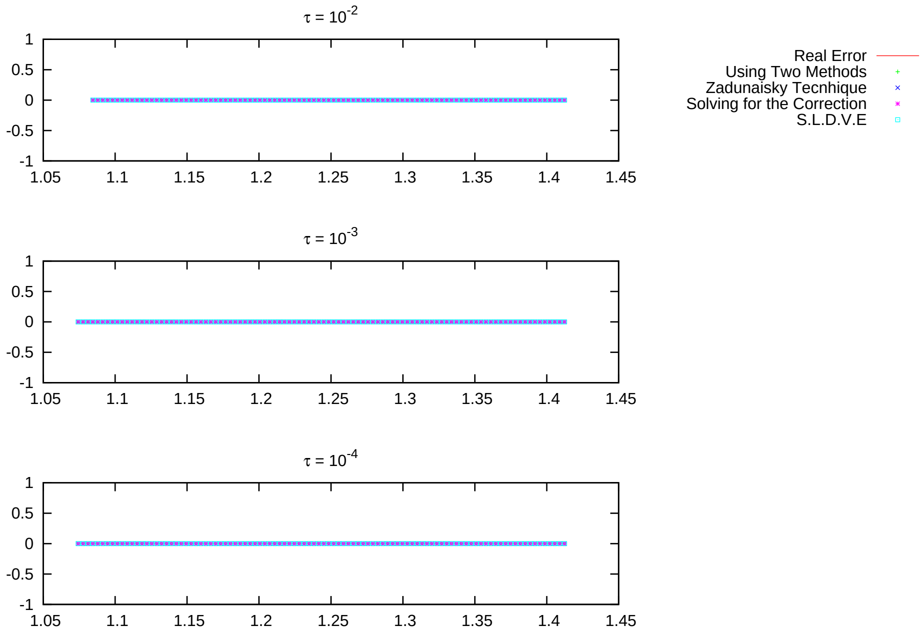

Figure 4.9: The true error and its estimates obtained for Nyström methods applied to DAE1 on the non-uniform grids

	$\tau = 10^{-2}$	$\tau = 10^{-3}$	$\tau = 10^{-4}$
Richardson	–	–	–
Using 2 methods	7.213e-04	7.121e-07	1.963e-13
Zadunaisky	9.962e+01	1.436e-01	1.427e-02
SC	9.962e+01	1.436e-01	9.529e+01
SLDVE	5.930e-05	5.838e-09	5.830e-13

Table 4.17: Accuracy of the global error evaluation techniques applied to DAE1 on the non-uniform grids.

	$\tau = 10^{-2}$	$\tau = 10^{-3}$	$\tau = 10^{-4}$
Richardson	–	–	–
Using 2 methods	0.132	1.328	13.717
Zadunaisky	0.184	1.784	17.933
SC	0.168	1.836	17.665
SLDVE	0.104	1.060	10.157

Table 4.18: CPU time (in sec) of the global error evaluation applied to DAE1 on the non-uniform grids

	$\tau = 10^{-2}$	$\tau = 10^{-3}$	$\tau = 10^{-4}$
Richardson	–	–	–
Using 2 methods	7.213e-04	7.121e-07	1.963e-13
Zadunaisky	9.962e+01	1.436e-01	1.427e-02
SC	9.962e+01	1.436e-01	9.529e+01
SLDVE	5.930e-05	5.838e-09	5.830e-13

Table 4.19: Accuracy of the global error evaluation techniques applied to DAE2 on the non-uniform grids.

2. **Problem DAE2:** Recall that the Nyström methods are weakly stable and we integrate the stiff problem DAE2 on the non-uniform grids. Figure 4.10 shows that the error estimate computed using SLDVE agrees very well with the true error. In addition to this ability to provide a good approximation of the error, SLDVE is also cheap in terms of CPU time (See Table 4.20). Only this method exhibit a good performance in terms of accuracy and CPU time. The error estimate produced by the Zadunaiskys technique and solving for the correction are not accurate, and the methods are shown to be computationally expensive.

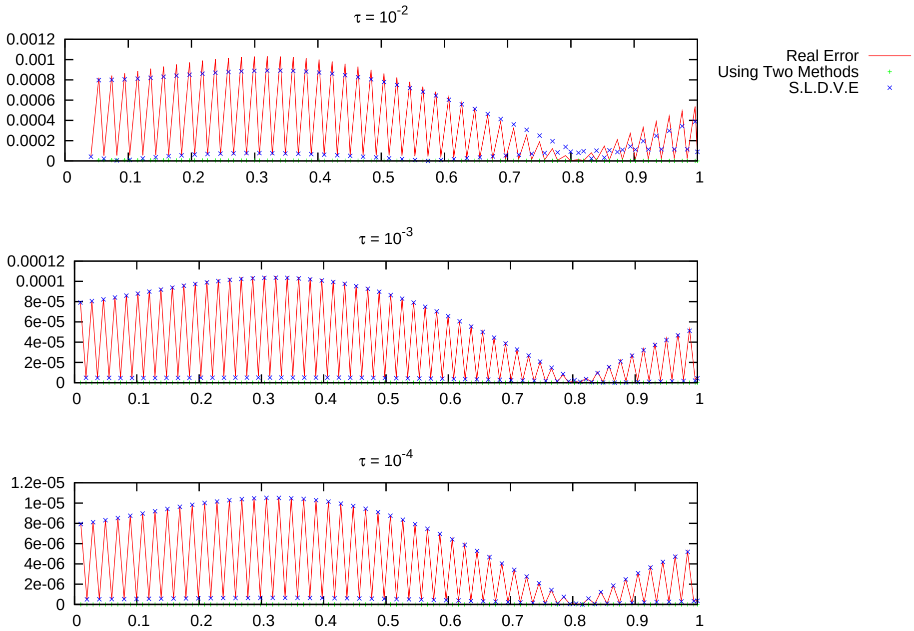

Figure 4.10: The true error and its estimates obtained for Nyström methods applied to DAE2 on the non-uniform grids

	$\tau = 10^{-2}$	$\tau = 10^{-3}$	$\tau = 10^{-4}$
Richardson	–	–	–
Using 2 methods	0.044	0.380	3.776
Zadunaisky	0.060	0.588	6.856
SC	0.060	0.588	5.748
SLDVE	0.032	0.304	3.208

Table 4.20: CPU time (in sec) of the global error evaluation applied to DAE2 on the non-uniform grids

Chapter 5

Conclusion

The main aim of this dissertation was to compare the performance of different global error evaluation techniques when they are applied to multistep methods to solve both ODEs and index 1 DAEs.

To achieve this goal, a theoretical background of the methods was first provided. We implemented different techniques for global error evaluation in C++ and considered two sets of test ODEs and index 1 DAEs problems to conduct numerical experiments. The test were performed for different types of multistep methods and different grids. We conducted the test on a personal computer with processor *Intel(R) Pentium(R) 4 CPU 3.20GHz* under *Ubuntu Linux*.

Numerical results for ODEs were presented in Chapter 3. The results showed that the SLDVE provides an excellent approximation to the error produced by multistep methods. This method exhibits advantages in terms of accuracy and CPU time. We want to point out that SLDVE was able to estimate the error with accuracy of order 5 even when the true error itself is large. This was the case, as example, when we used the Adams method of order 4 to solve the stiff ODE ODE4 on the uniform grids. This result was also confirmed when the Nyström method, which is a weakly stable multistep method, was used to solve the same problem.

Following the SLDVE are the Richardson extrapolation and the using two different methods. On uniform grids, these methods provided the same accuracy as the SLDVE, but are more expensive in terms of CPU time. The numerical results indicate that the Richardson extrapolation integrates the equation and evaluates the global error 3 times slower than SLDVE. As for using two different methods, the integration of the problem and the evaluation of the global error last 2 times longer than the SLDVE. This result can be explained by the number of right hand function calls and the resolution non-linear equations needed when implicit methods are used.

The solving for the corrections and the Zadunaisky's technique were particularly expensive and the accuracy of the global error estimate were not satisfactory. The latter result confirms the conclusion of Aïd and Levacher in [1]. Similar results were obtained for non-uniform grids.

For the index-1 DAE, similar results were obtained. The performance of SLDVE was confirmed in terms of accuracy and CPU time. This global error evaluation strategie worked well for both non-stiff and stiff problems. We specially want to point out that SLDVE was the only strategie that presents the ability to estimate the error correctly when the Nyström method of order 4 was used to solve the stiff problem DAE2 on the non-uniform grids. Richardson extrapolation and using two different methods show some interest only for some problems.

The similarity in the results for ODE and DAE can be explained by the fact that the *state space form* methods were used to solve the DAEs, that is the ODE part of the equation was solve using the usual linear multistep methods for ODEs.

Bibliography

[1] R. Aïd and L. Levacher, *Numerical investigations on global estimation for ordinary differential equations*, J. Comput. Appl. Math. 82 (1997), no. 1-2, pp. 21-39.

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

[3] J. C. Butcher, *On the Covergence of Numerical Solutions to Ordinary Differential Equations*, Math. Comp 20 (1966), no. 93, pp 1-10.

[4] J. C. Butcher, *Numerical Methods for Ordinary Differential Equations*, Wiley, New Zealand, 2003.

[5] M. Crouzeix and F. J. Lisbona, *The Convergence of Variable Step-Size, Variable-Formula, Multistep Methods*, SIAM J. Numer. Anal. 21 (1984), no. 3 pp 512-534.

[6] P. Dauflhard and F. Bornemann, *Scientific Computnig with Ordinary Differential Equations*, Springer, 2000.

[7] E. Hairer and C. Lubich, *Asymptoyic Expansions of the Global Error of Fixed-Stepsize Methods*, Numer. Math. 45 (1984), no. 3 pp 345-360.

[8] E. Hairer, S.P Nørsett and G. Wanner, *Solving Ordinary Differential Equations I: Nonstiff Problems*, Springer, Berlin, 1991.

[9] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*, Springer, Berlin, 1996.

[10] G. Yu. Kulikov and S. K. Shindin, *A Local-Global Stepsize Control for Multistep Methods Applied to Semi-Explicit Index 1 Differential-Algebraic Equations*, Korean J. Comp. App. Math 6 (1999), no. 3 pp. 463-492.

[11] G. Yu. Kulikov, *A Local-Global Version of a Stepsize Control for Runge-Kutta Methods*, Korean J.Comp. App. Math. 7 (2000), no. 2 pp. 289-318.

[12] G. Yu. Kulikov and S. K. Shindin, *A Technique for Controlling the Global Error in Multistep Methods*, Zh. Vychil. Mat. Mat. Fiz. 40 (2000), no. 9 pp. 1308-1329 [*translation in* J. Comput. Math. Math. Phys. 40 (2000), pp. 1255-1275.]

[13] G. Yu. Kulikov and S. K. Shindin, *On interpolation-Type Multistep Methods with Automatic Global Error Control*, Comp. Math. Math. Phy. 44 (2004), no. 8 pp.1388-1409.

[14] G. Yu. Kulikov and S. K. Shindin, *Global Error Estimation and Extrapolated Multistep Methods for Index 1 Differential-Algebraic Systems*, BIT Numer. Math. 45 (2005), pp 517-542.

[15] P. Kunkel and V. Mehrmann, *Differential-Algebraic Equations : Analysis and Numerical Solution*, EMS, Zürich, 2006.

[16] B. Lindberg, *Error Estimation and Iterative Improvement for Discretization Algorithms*, BIT 20 (1980), pp. 486 - 500.

[17] L. F. Shampine, *Numerical Solution of Odrinary Differential Equations*, Chapman & Hall, 1994.

[18] R. D. Skeel,*Global Error Estimation and the Backward Differentiation formulas*, Appl. Math. Comput. 5 (1989), pp 197-208.

[19] R. D. Skeel, *Analysis of Fixed-Stepsize Methods*, SIAM Numer. Anal. 13(1976), no. 5 pp 664-685.

[20] R. D. Skeel, *Thirteen Ways to Estimate Global Error*, Numer. Math. 48 (1986), pp. 1-20.

[21] P. E. Zadunaisky, *A Method for the Estimation of Errors Propagated in the Numerical Solution of a System of Ordinary Differential Equations*, International Astronomical Union. Symposium, Academic Press (1964), no. 25 p.281-288.