UNIVERSITÀ DEGLI STUDI DI BARI
ALDO MORO

Dottorato di Ricerca in Matematica

XXIII Ciclo – A.A. 2010/2011

Settore Scientifico-Disciplinare:

MAT/08 – Analisi Numerica

Tesi di Dottorato

High order finite difference schemes for the numerical solution of second order ordinary differential problems

Candidato:
Giuseppina SETTANNI

Supervisore della tesi:
Prof. P. AMODIO

Coordinatore del Dottorato di Ricerca:
Prof. L. LOPEZ

Apriró anche nel deserto una strada,
immetteró fiumi nella steppa.
(Isaia 43)

Contents

Acknowledgements

Firstly I want to be grateful to my supervisor Prof. Pierluigi Amodio for the scientific cooperation of the last four years and for the worthwhile exchange of ideas which has allowed myself to evolve in the research. Moreover, I am also grateful to Prof. Ewa B. Weinmüller for my stay of research at Vienna University of Technology, for the scientific exchange and for the international cooperation she has involved me in. I say thank both of them for their willingness.

Introduction

The present work shoots for the presentation of a code able to solve second-order ordinary differential equations that arise from different applications in physics, chemistry and engineering. Apart the restriction of the second order, we consider initial and boundary value problems, and also eigenvalue problems of differential equations, in particular we examine Sturm-Liouville problems.

Chapter 1 begins with a theoretical description of second order differential equations and eigenvalue problems. Theorems of existence and uniqueness and some definitions are given. In the last section we introduce the Boundary Value Methods, whose idea is at the base of the methods introduced in the next chapters.

Chapter 2 presents a class of methods based on generalized high order difference schemes (HOGD), which include D2ECDF, D2GFDF and D2GBDF methods which defer among them for the difference scheme chosen to discretize the first derivative. These methods are introduced for solving second order boundary value problems. Properties of the methods are also analyzed, moreover it has been realized an analysis of the conditioning which puts the basis for the generalized upwind method.

Chapter 3 takes in consideration a particular class of BVPs named singular perturbation problems (SPPs). We analyze the application of HOGD schemes, introduced in Chapter 2, to SPPs and by the conditioning analysis we introduced the class of high order generalized upwind methods HOGUPs, which are very efficient and suitable. One section is dedicated to deferred method which permits to evaluate the estimation error considering the same numerical scheme. Moreover, a stepsize variation based on the error equidistribution is used in order to guarantee the accuracy with few points of discretization. In the last section some examples on test problems taken from [29] show the efficiency of the Matlab code HOFiD_UP.

Chapter 4 contemplates the solution of second order initial value problems. The high order difference schemes considered to find the solution follow

the idea at the base of HOGD methods introduced for the boundary value problems. An analysis of the conditioning is performed and seven classes of methods depending on the choice of the main difference scheme to approximate the second and the first derivatives. Even and odd methods are considered and three different approaches allow us to obtain the solution of the problem. The estimation error is obtained using the deferred correction and it is used in order to improve the distribution of the mesh points, suggesting a simple strategy of stepsize variation discussed in the Section 4.4. Some applications are shown, the most important one is the *Flow in concrete* problem.

In Chapter 5 eigenvalue problems for second order differential equations are treated, in particular Sturm-Liouville problems. Generalized high order central difference schemes ECDFs, introduced in Chapter 2, are the main schemes applied to discretize the eigenvalue problems. Since the boundary conditions for Sturm-Liouville problems can be regular and singular, it is possible to introduce four approaches, differently among them which depend on the choice of initial and final methods. As a consequence an equivalent algebraic eigenvalue problem is obtained, following the idea of matrix methods. The technique of error equidistribution allows us to compute more accurately the eigenfunctions on a variable mesh. Some examples of regular and singular Sturm-Liouville problems show the efficiency and accuracy of the proposed approach.

Chapter 1

Second Order Differential Equations

1.1 Initial Value Problems

A second-order linear differential equation may be written in the form

$$a_0(x)y'' + a_1(x)y' + a_2(x)y = g(x) \tag{1.1.1}$$

where a_0, a_1, a_2 are real-valued continuous functions defined on some interval I. If $a_0 \neq 0$, $\forall x \in I$, then (1.1.1) may be written as

$$y'' + c_1(x)y' + c_0(x)y = f(x) \tag{1.1.2}$$

where $c_1 = a_1/a_0$, $c_0 = a_2/a_0$ and $f = g/a_0$.

It is often convenient to refer to the linear ordinary differential equation (ODE) using a *differential operator* notation

$$Ly(x) \equiv y''(x) + c_1(x)y'(x) + c_0(x)y(x). \tag{1.1.3}$$

Then (1.1.2) may be read as

$$Ly(x) = f(x). \tag{1.1.4}$$

If $f(x) = 0$ for all $x \in I$, the resulting equation $Ly(x) = 0$ is called a *homogeneous equation*. The equation (1.1.4) is called a *nonhomogeneous equation*.

Definition 1.1.1. A solution $\phi(x)$ of $Ly(x) = f(x)$ is a twice differentiable function which satisfies $L\phi(x) = f(x)$.

Theorem 1.1.2 (Existence-uniqueness theorem)**.** *Let the functions $c_0(x)$, $c_1(x)$ and $f(x)$ be continuous on an interval I. For any $x_0 \in I$ and constants y_0, y_0' a unique solution exists of the initial-value problem (IVP)*

$$\begin{aligned} &Ly = y'' + c_1(x)y' + c_0(x)y = f(x), \\ &y(x_0) = y_0, \\ &y'(x_0) = y_0'. \end{aligned} \tag{1.1.5}$$

In the following we consider some results for the second order linear homogeneous equation $Ly = 0$.

Theorem 1.1.3. *If ϕ_1 and ϕ_2 are solutions of $Ly = 0$, then a linear combination $\gamma_1\phi_1 + \gamma_2\phi_2$ (γ_1 and γ_2 given constants) is also a solution of $Ly = 0$.*

Definition 1.1.4. The functions $f_1, f_2, \ldots, f_n$ are said to be *linearly dependent* on an interval I if there exist constants $\gamma_1, \gamma_2, \ldots, \gamma_n$, not all zero, such that

$$\gamma_1 f_1(x) + \gamma_2 f_2(x) + \cdots + \gamma_n f_n(x) \equiv 0, \quad \forall\, x \in I. \tag{1.1.6}$$

Functions that are not linearly dependent are called *linearly independent*. That is, the functions $f_1, f_2, \ldots, f_n$ are linearly independent on I if the only constants satisfying (1.1.6) are $\gamma_1 = \gamma_2 = \cdots = \gamma_n = 0$.

The concept of linear independence is related to the determinant of a matrix known as the *Wronskian*.

Definition 1.1.5. The *Wronskian* of two differentiable functions $\phi_1(x)$ and $\phi_2(x)$ on an interval I is defined by the determinant

$$W(\phi_1, \phi_2; x) = \begin{vmatrix} \phi_1(x) & \phi_2(x) \\ \phi_1'(x) & \phi_2'(x) \end{vmatrix} = \phi_1(x)\phi_2'(x) - \phi_1'(x)\phi_2(x). \tag{1.1.7}$$

Theorem 1.1.6. *Let ϕ_1 and ϕ_2 be solutions of $Ly = 0$ on I. Then ϕ_1 and ϕ_2 are linearly independent on I if, and only if,*

$$W(\phi_1, \phi_2; x) \neq 0 \quad \textit{for all} \quad x \in I. \tag{1.1.8}$$

Corollary 1.1.7. *Let ϕ_1 and ϕ_2 be any two solutions of $Ly = 0$ on I. Then their Wronskian either vanishes identically or it is never zero on I.*

Theorem 1.1.8. *Let ϕ_1 and ϕ_2 be linearly independent solutions of $Ly = 0$ on an interval I. Then every solution of $Ly = 0$ can be uniquely expressed as*

$$\phi(x) = \gamma_1\phi_1(x) + \gamma_2\phi_2(x), \tag{1.1.9}$$

for γ_1 and γ_2 given constants.

Now, let us consider a general IVP written in the form

$$\begin{aligned} y''(x) &= f(x, y(x), y'(x)), \qquad x \in [a, b], \\ y(x_0) &= y_0 \\ y'(x_0) &= y_0' \end{aligned} \tag{1.1.10}$$

then from the theory of ordinary differential equations the standard theorems (see [24]) follow.

Theorem 1.1.9 (Local Existence of Solutions of Initial Value Problems)**.** *If $f(x, y, y')$ is continuous in an open and connected domain D, containing the point (x_0, y_0, y_0'), then there exists at least one twice continuously differentiable function $y(x)$, defined on some interval I containing x_0, which satisfies the initial value problem* (1.1.10)*, $(x, y(x), y'(x)) \in D$ for all $x \in I$.*

In other words the continuity of $f(x, y, y')$ guarantees the existence of at least one solution of any initial value problem on sufficiently small intervals.

Theorem 1.1.10 (Uniqueness and Continuity of Solutions of Initial Value Problems)**.** *If $f(x, y, y')$ is continuous and satisfies the Lipschitz condition in a domain D which contains the point (x_0, y_0, y_0'), this means that two non negative constants K and L exist such that, whenever (x, y, y') and (x, z, z') are in the domain D, it holds*

$$\left|f(x, y, y') - f(x, z, z')\right| \leq K\,|y - z| + L\left|y' - z'\right|, \tag{1.1.11}$$

then there is only one solution of the initial value problem (1.1.10)*. Furthermore, this solution can be uniquely continued arbitrarily close to the boundary of D. If D contains $[a, b] \times (-\infty, \infty) \times (-\infty, \infty)$, this solution exists finite for all $x \in [a, b]$.*

The additional assumption that f is Lipschitzian guarantees uniqueness and continuity.

1.2 Linear Equation with constant coefficients

We consider the equation

$$Ly = y'' + c_1 y' + c_0 y = 0, \tag{1.2.1}$$

where c_0 and c_1 are constants. We observe that

1. $e^{\gamma x}$ is a solution of the first order equation $y' = \gamma y$.
2. The derivative of the exponential function is a multiple of the function itself.

This suggests to consider $\phi(x) = e^{\lambda x}$ as a possible solution of $Ly = 0$ for an appropriate value of λ. Then, if we assume $\phi(x) = e^{\lambda x}$, substituting it in (1.2.1) we obtain

$$(\lambda^2 + c_1\lambda + c_0)e^{\lambda x} = 0.$$

Since $e^{\lambda x}$ cannot vanishes, equation (1.2.1) has a solution if

$$p(\lambda) = \lambda^2 + c_1\lambda + c_0 = 0. \tag{1.2.2}$$

$p(\lambda)$ is called the *characteristic polynomial* and $\lambda^2 + c_1\lambda + c_0 = 0$ is called the *characteristic equation* of $Ly = 0$. The roots of the polynomial $p(\lambda)$ are defined as

$$\lambda_{1,2} = \frac{-c_1 \pm \sqrt{c_1^2 - 4c_0}}{2}.$$

Then we can distinguish three cases.

1. $c_1^2 - 4c_0 > 0$: then λ_1 and λ_2 are real and distinct roots and the general solution of (1.2.1) is

$$\phi(x) = \mu_1 e^{\lambda_1 x} + \mu_2 e^{\lambda_2 x}.$$

2. $c_1^2 - 4c_0 < 0$: then λ_1 and λ_2 are complex conjugate roots, $\lambda_{1,2} = \alpha \pm i\beta$, for α and β real, and the solution of (1.2.1) is

$$\phi(x) = \mu_1 e^{(\alpha+i\beta)x} + \mu_2 e^{(\alpha-i\beta)x}.$$

   Using the Euler form $e^{\pm i\theta} = \cos\theta \pm i\sin\theta$, the solution can be also expressed as

$$\phi(x) = e^{\alpha x}(\tilde{\mu}_1 \cos\beta x + \tilde{\mu}_2 \sin\beta x).$$

   where $\tilde{\mu}_1 = \mu_1 + \mu_2$ and $\tilde{\mu}_2 = i(\mu_1 - \mu_2)$ are two new constants.

3. $c_1^2 - 4c_0 = 0$: then $\lambda_1 = \lambda_2 = \bar{\lambda}$. For Theorem 3.2.1 in [52] also $xe^{\bar{\lambda}x}$ is a solution of (1.2.1) and the general solution is written as

$$\phi(x) = \mu_1 e^{\bar{\lambda}x} + \mu_2 x e^{\bar{\lambda}x}.$$

We underline that the constants μ_1 and μ_2 are computed imposing the initial or boundary conditions to the equation (1.2.1).

1.3 Two-Point Boundary Value Problems

We consider a *two-point boundary value problem* (BVP) defined as

$$y''(x) = f(x, y(x), y'(x)), \quad x \in [a, b] \tag{1.3.1}$$

$$y(a) = \eta_1, \quad y(b) = \eta_2. \tag{1.3.2}$$

If (1.3.1)-(1.3.2) has two solutions $y_1(x)$ and $y_2(x)$, then by the first mean value theorem of the differential calculus it exists $c \in (a, b)$ such that $y_1'(c) - y_2'(c) = 0$, since the difference $y_1'(x) - y_2'(x)$ vanishes at the endpoints of the interval. In this case, $y_1(x)$ and $y_2(x)$ are both solutions of (1.3.1) and

$$y(a) = \eta_1, \quad y'(c) = m, \tag{1.3.3}$$

and

$$y'(c) = m, \quad y(b) = \eta_2, \tag{1.3.4}$$

where $m = y_1'(c) = y_2'(c)$. We can establish a relation between the uniqueness interval for the two-point boundary value problem (1.3.1)-(1.3.2) and the boundary value problems (1.3.1)-(1.3.3) and (1.3.1)-(1.3.4), as it is expressed in the following theorem [24].

Theorem 1.3.1. *Let $a < c < b$ be. If uniqueness holds for all second order differential equations* (1.3.1)*, with boundary conditions*

$$y(a) = \eta_1, \quad y'(\bar{c}) = m \tag{1.3.5}$$

whenever $\bar{c} \in (a, c]$, and if uniqueness holds for all second order differential equations (1.3.1)*, with boundary conditions*

$$y'(\bar{c}) = m \quad y(b) = \eta_2 \tag{1.3.6}$$

whenever $\bar{c} \in [c, b)$, then the uniqueness holds for all two-point boundary value problems (1.3.1)-(1.3.2)*.*

Theorem 1.3.1 shows that the interval uniqueness for the two-point boundary value problems is always at least as large as that for either of the two second boundary value problems (1.3.1)-(1.3.3) and (1.3.1)-(1.3.4). If the two boundary value problems can be solved uniquely on the interval $[a, c]$ and $[c, b]$, then the solution of (1.3.1)-(1.3.2) may be obtained on the interval $[a, b]$ by joining together the solutions of suitable chosen problems on the subinterval.

We can give a restrictive result about the existence and uniqueness as in [24].

Theorem 1.3.2. *Let us suppose $f(x, y, y')$ be continuous on $[a, b]\times(-\infty, \infty)\times(-\infty, \infty)$ and satisfy the Lipschitzian condition on $[a, b]$, so that constants K and L exist such that for any (x, y, y') and (x, z, z') in $[a, b]$,*

$$\left|f(x, y, y') - f(x, z, z')\right| \leq K\left|y - z\right| + L\left|y' - z'\right|.$$

If

$$b - a < 4\begin{cases} \dfrac{1}{(4K - L^2)^{1/2}}\cos^{-1}\dfrac{L}{2\sqrt{K}} & \text{if } \ 4K - L^2 > 0 \\ \dfrac{1}{(L^2 - 4K)^{1/2}}\cosh^{-1}\dfrac{L}{2\sqrt{K}} & \text{if } \ 4K - L^2 < 0, \\ & \quad L > 0, K > 0 \\ \dfrac{1}{L} & \text{if } \ 4K - L^2 = 0, \\ & \quad L > 0 \\ +\infty & \text{otherwise} \end{cases} \tag{1.3.7}$$

then the problem (1.3.1)-(1.3.2) *has one and only one solution.*

Proof. See [24] for the proof. □

Singular Perturbation Problems

Many phenomena in biology, chemistry, engineering and physics can be described by boundary value problems. When a mathematical model is associated with a phenomenon, generally what is essential is captured, retaining the important quantities and omitting the negligible ones which involves small parameters. The model that would be obtained maintaining the small parameters is called perturbed model. Then, we consider the following equation

$$\epsilon y'' = f(x, y, y'), \qquad x \in [a, b], \quad y \in \mathbb{R}, \tag{1.3.8}$$

subject to separated boundary conditions

$$y(a) = \eta_a, \quad y(b) = \eta_b, \tag{1.3.9}$$

where ϵ is the *perturbation parameter*, $0 < \epsilon \ll 1$, and f is a sufficiently smooth function. The problem defined in (1.3.8)-(1.3.9) is called *singular perturbation problem* (SPP).

Table 1.1: General solution behavior for

$p(x) \neq 0$	$p(x) < 0$ $p(x) > 0$	boundary layer at $x = a$ boundary layer at $x = b$
$p \equiv 0$	$q(x) > 0$ $q(x) < 0$ $q(x)$ changes sign	boundary layers at $x = a$ and $x = b$ rapidly oscillatory solution classic turning point
$p' \neq q$, $p(0) = 0$, $p(x) \neq 0$ for $x \neq 0$	$p'(0) < 0$ $p'(0) > 0$	no boundary layers, interior layer at $x = 0$ boundary layers at $x = a$ and $x = b$, no boundary layer at $x = 0$

The perturbation parameter causes very fast variations of the solution, called *layers*, in narrow regions. If the problem (1.3.8)-(1.3.9) may be rewritten in explicit form as

$$\begin{aligned} &\epsilon y'' - p(x)y' - q(x)y = r(x), \quad x \in [a, b] \\ &y(a) = \eta_1, \quad y(b) = \eta_2, \end{aligned} \tag{1.3.10}$$

where $p(x), q(x)$ and $r(x)$ are smooth, bounded functions, then we are able to state if the problem has boundary or internal layers. Distinguishing different cases depending on the properties of $p(x)$ and separating the slow solution component and the fast solution component, see [18], we can summarize the different behaviors of the solution of (1.3.10) as described in Table 1.1.

1.4 Eigenvalue Problems and Sturm-Liouville Problems

Eigenvalue problems associated with ordinary differential equations arise in considering physical problems, such as the displacement of a vibrating string or determining the temperature distribution of a heat-conducting rod. The typical equation that often occurs in eigenvalue problems is of the form

$$a_1(x)\frac{d^2y}{dx^2} + a_2(x)\frac{dy}{dx} - [a_3(x) - \lambda]\, y = 0, \quad x \in [a, b] \tag{1.4.1}$$

where $a_1(x) \neq 0$ in the interval $[a, b]$. Now, we consider

$$p(x) = \frac{1}{a_1(x)} \exp\left[\int_a^x \frac{a_2(t)}{a_1(t)}\, dt\right], \quad q(x) = \frac{a_3(x)}{a_1(x)} p(x), \quad r(x) = \frac{p(x)}{a_1(x)}. \tag{1.4.2}$$

Multiplying by $p(x)$, (1.4.1) is transformed into the self-adjoint form

$$\frac{d}{dx}\left(p\frac{dy}{dx}\right) - (q - \lambda r)\, y = 0, \tag{1.4.3}$$

which is known as a *Sturm-Liouville equation*. In terms of the self-adjoint operator

$$L = \frac{d}{dx}\left(p\frac{d}{dx}\right) - q,$$

the equation (1.4.3) can be written as

$$Ly + \lambda\, r(x) y = 0, \tag{1.4.4}$$

where λ is a parameter independent of x, and p, q and r are real-valued functions of x. We require that p is continuously differentiable in a closed interval $[a, b]$ and that q and r are continuous on the same interval.

Definition 1.4.1. The Sturm-Liouville equation (1.4.3) is called *regular* in the interval $[a, b]$ if the functions $p(x)$ and $r(x)$ are positive in the interval $[a, b]$.

Definition 1.4.2. The Sturm-Liouville equation

$$Ly + \lambda r(x) y = 0, \quad -\infty \leq a \leq x \leq b \leq \infty,$$

with the separated boundary conditions

$$\begin{aligned} \alpha_1 y(a) + \alpha_2 y'(a) &= 0, \\ \beta_1 y(b) + \beta_2 y'(b) &= 0, \end{aligned} \tag{1.4.5}$$

where the real constants α_1 and α_2 and also β_1 and β_2 are not both zero, is called *Sturm-Liouville system*.

The values of λ for which the Sturm-Liouville system has a nontrivial solution are called *eigenvalues*, and the corresponding solutions are called *eigenfunctions*.

Properties of Eigenvalues and Eigenfunctions

Definition 1.4.3. Let $\phi(x)$ and $\psi(x)$ be any real-valued integrable functions on an interval I. Then $\phi(x)$ and $\psi(x)$ are said to be *orthogonal* on I with respect to a weight function $r(x) > 0$ if and only if

$$(\phi,\ \psi) = \int_I \phi(x)\psi(x)r(x)\,dx = 0. \tag{1.4.6}$$

The interval I may be of infinite extent, or it may be either open or closed at one or both ends of the finite interval.

When $\phi = \psi$ in (1.4.6) we have the *norm* of ϕ

$$\|\phi\| = \left[\int_I \phi^2(x)r(x)\,dx\right]^{\frac{1}{2}}. \tag{1.4.7}$$

Theorem 1.4.4. *Let the coefficients p, q and r in the Sturm-Liouville system be continuous in $[a, b]$. Let the eigenfunctions ϕ_i and ϕ_k, corresponding to λ_i and λ_k, be continuously differentiable. Then ϕ_i and ϕ_k are orthogonal with respect to the weight function r in $[a, b]$.*

Proof. Since the eigenfunctions ϕ_i and ϕ_k corresponding to λ_i and λ_k satisfy the Sturm-Liouville equation, then

$$\frac{d}{dx}\left(p\phi_i'\right) - (q - \lambda_i r)\,\phi_i = 0, \tag{1.4.8}$$

and

$$\frac{d}{dx}\left(p\phi_k'\right) - (q - \lambda_k r)\,\phi_k = 0. \tag{1.4.9}$$

Multiplying (1.4.8) by ϕ_k and (1.4.9) by ϕ_i, and subtracting, it follows that

$$\begin{aligned}(\lambda_i - \lambda_k)\,r\phi_i\phi_k &= \phi_i\frac{d}{dx}\left(p\phi_k'\right) - \phi_k\frac{d}{dx}\left(p\phi_i'\right)\\ &= \frac{d}{dx}\left[\left(p\phi_k'\right)\phi_i - \left(p\phi_i'\right)\phi_k\right].\end{aligned}$$

The integration yields

$$\begin{aligned}(\lambda_i - \lambda_k)\int_a^b r\phi_i\phi_k\,dx &= \left[p\left(\phi_k'\phi_i - \phi_i'\phi_k\right)\right]_a^b\\ &= p(b)\left[\phi_k'(b)\phi_i(b) - \phi_i'(b)\phi_k(b)\right]\\ &\quad - p(a)\left[\phi_k'(a)\phi_i(a) - \phi_i'(a)\phi_k(a)\right].\end{aligned}$$

The end conditions for the eigenfunctions ϕ_i and ϕ_k are in the form

$$\beta_1\phi_i(b) + \beta_2\phi_i'(b) = 0,$$

$$\beta_1\phi_k(b) + \beta_2\phi_k'(b) = 0.$$

If $\beta_2 \neq 0$, multiplying the first condition by $\phi_k(b)$ and the second condition by $\phi_i(b)$, and subtracting to the second condition the first one, we obtain

$$\phi_k'(b)\phi_i(b) - \phi_i'(b)\phi_k(b) = 0. \tag{1.4.10}$$

In the same way, if $\alpha_2 \neq 0$, we obtain

$$\phi_k'(a)\phi_i(a) - \phi_i'(a)\phi_k(a) = 0. \tag{1.4.11}$$

As a consequence of (1.4.10) and (1.4.11) it follows that

$$(\lambda_i - \lambda_k)\int_a^b r\phi_i\phi_k\,dx = 0. \tag{1.4.12}$$

Since λ_i and λ_k are distinct eigenvalues, then

$$\int_a^b r\phi_i\phi_k\,dx = 0. \tag{1.4.13}$$

□

Theorem 1.4.5. *For a Sturm-Liouville system* (1.4.2) *all the eigenvalues are real, and the eigenfunctions may be chosen real, in each of following cases:*

- $r(x)$ *is of constant sign on* $[a, b]$*;*
- $r(x)$ *changes sign on* $[a, b]$*, while* $p(x) > 0$ *for* $x \in [a, b]$ *and* $\alpha_1\beta_1 \geq 0$*,* $\alpha_2\beta_2 \geq 0$*.*

Theorem 1.4.6. *An eigenfunction of a regular Sturm-Liouville system is unique except for a constant factor.*

Proof. See [52] for the proof. □

Theorem 1.4.7. *Let us suppose the Sturm-Liouville system be regular, so that it satisfies the hypothesis* $p(x) > 0$ *and* $r(x) > 0$ *for* $x \in [a, b]$*. Then all the eigenvalues of this system are real and may be written as a sequence* $\{\lambda_j\}$*, where* $\lambda_0 < \lambda_1 < \lambda_2 < \ldots$*,* $\{\lambda_j\} \to \infty$ *as* $j \to \infty$*. For each* j *the corresponding eigenfunction* $\phi_j(x)$*, uniquely determined up to a constant factor, has exactly* j *zeros in the interval* $[a, b]$*.*

PROOF. See [59, 52] for the proof. □

Theorem 1.4.8. *Let us suppose the Sturm-Liouville system satisfies the hypothesis $p(x) > 0$, the function $r(x) \neq 0$ changes sign, while $q(x) > 0$ for $x \in [a, b]$ and $\alpha_1\beta_1 > 0$, $\alpha_2\beta_2 > 0$. Then all the eigenvalues of this system are real and may be written as a sequence $\{\lambda_j^+\}$, $\{\lambda_j^-\}$, with $0 < \lambda_0^+ < \lambda_1^+ < \lambda_2^+ < \ldots$, $0 > \lambda_0^- > \lambda_1^- > \lambda_2^- > \ldots$, $\{\lambda_j^+\} \to \infty$ and $\{\lambda_j^-\} \to -\infty$ as $j \to \infty$. For each j the corresponding eigenfunctions $\phi_j^+(x)$ and $\phi_j^-(x)$ associated to λ_j^+ and λ_j^-, respectively, have exactly $j = 1$ zeros on the open interval (a, b).*

PROOF. See [59]. □

Singular Sturm-Liouville Systems

In the literature a Sturm-Liouville equation is called singular when it is defined on a semi-infinite or infinite interval, or when the coefficient $p(x)$ or $r(x)$ vanishes, or when one of the coefficients becomes infinite at one or both ends of a finite interval.

Remark 1.4.9. In order to give some definitions we introduce some notations. Let I be an interval of $\mathcal{R}$ and $r(x) > 0$ for all $x \in I$, then

- $L(I) = \{\phi : I \to \mathcal{C} : \int_I |\phi| \equiv \int_I |\phi(x)|dx < \infty\}$;
- $L_{loc}(I) = \{\phi : I \to \mathcal{C} : \int_\alpha^\beta |\phi(x)|dx < \infty \text{ for all } [\alpha, \beta] \subset I\}$;
- $L_r^2(I) = \{\phi : I \to \mathcal{C} : \int_I r(x)|\phi(x)|^2 dx < \infty\}$,
- $AC_{loc}(I) = \{\phi :\to \mathcal{C} : \phi \textit{ is absolutely continuous on all } \ [\alpha; \beta] \subseteq I\}$.

Let us consider $I = (a, b)$, $-\infty \leq a < b \leq \infty$, and the equation

$$-(py')' + qy = \lambda ry \quad \text{on } I,\ \lambda \in \mathcal{C}, \tag{1.4.14}$$

with the conditions

$$p, q, r : I \to \mathcal{C}, \quad 1/p, q, r \in L_{loc}(I). \tag{1.4.15}$$

Definition 1.4.10. The end-point a is said to be *regular* (**R**) if

$$1/p, q, r \in L(a, d), \quad d \in I; \tag{1.4.16}$$

otherwise it is called *singular*. Similarly, the end-point b is said to be *regular* (**R**) if

$$1/p, q, r \in L(d, b), \quad d \in I; \tag{1.4.17}$$

otherwise it is called *singular*.

In the case the endpoint is singular, there are two main classifications.

Definition 1.4.11. The endpoint a is a *limit-point* (**LP**) if a is singular and for some $\lambda \in \mathcal{C}$ there exists at least one solution y of the equation (1.4.14) satisfying

$$\int_a^d r(x)|y(x)|^2 dx = \infty$$

for some $d \in (a, b)$.

The endpoint a is a *limit-circle* (**LC**) if a is singular and for some $\lambda \in \mathcal{C}$ any solution y of the equation (1.4.14) satisfies

$$\int_a^d r(x)|y(x)|^2 dx < \infty$$

for some $d \in (a, b)$.

Moreover, in the case of LC endpoints there are two sub-cases.

The LC endpoint a is a limit-circle non-oscillatory (**LCNO**) if there exists a point $d \in (a, c)$, a real value $\lambda \in \mathcal{R}$ and a solution y of (1.4.14) such that

$$y(x) > 0 \,\text{for all}\, x \in (a, d).$$

The LC endpoint a is a limit-circle oscillatory (**LCO**) if for any real value $\lambda \in \mathcal{R}$, any non-null solution y of (1.4.14) and any $d \in (a, c]$ there exists a point $\xi \in (a, d]$ such that $y(\xi) = 0$. Similar definitions are made in b.

Remark 1.4.12. We point out that the endpoints classification is independent of λ and depends only on the interval (a, b) and the set of coefficients $\{p, q, r\}$.

Definition 1.4.13. Consider $I = (a, b)$, $-\infty \le a < b \le \infty$, $\lambda \in \mathcal{R}$, $p, q, r : I \to \mathcal{R}$, $1/p, q, r \in L_{loc}(I)$, $p \ge 0$, $r > 0$ a.e. on I. The maximal domain Δ is defined by

$$\Delta = \{f : I \to \mathcal{C} : f, pf' \in AC_{loc}(I),\ f, w^{-1}(-(pf')' + qf) \in L^2_r(I)\}.$$

Definition 1.4.14. Given $y, z \in \Delta$ the *Lagrange sesquilinear form* is defined by

$$[y, z](x) = y(x)(p\bar{z}')(x) - (\bar{z})(x)(py')(x), \quad x \in (a, b). \tag{1.4.18}$$

Initial Value Problems

Theorem 1.4.15. *Let* (1.4.14) *and* (1.4.15) *hold. Assume* p, q, r *are real valued,* $r > 0$ *a.e. and the left endpoint* a *is* R *or* LC*. Suppose* u, v *are real valued linearly independent solutions on some interval* $(a, d]$ *for some fixed real* λ_0*. Given any* $\lambda \in \mathcal{R}$ *and any* $\alpha_1, \alpha_2 \in \mathcal{R}$ *the singular initial value problem consisting of the equation*

$$-(py')' + qy = \lambda ry \quad on\, I$$

and the singular initial condition

$$[y, u](a) = \alpha_1, \quad [y, v](a) = \alpha_2$$

has a unique solution y *on* I*. Similarly at* b*.*

Proof. See in Hiton and Schaefer [41]. □

Boundary Value Problems

Consider the equation

$$-(py')' + qy = \lambda ry \quad \text{on } I = (a, b),\ -\infty \le a < b \le \infty,\ \lambda \in \mathcal{R},$$

where

$$p, q, r : I \to \mathcal{R},\ 1/p, q, r \in L_{loc}(I),\ p \ge 0,\ r > 0 \text{ a.e on } I, \tag{1.4.19}$$

and the separated boundary conditions

$$\begin{aligned} &\alpha_1 y(a) + \alpha_2 (py')(a) = 0,\ (\alpha_1, \alpha_2) \ne (0, 0),\ \alpha_1, \alpha_2 \in \mathcal{R}, \\ &\beta_1 y(b) + \beta_2 (py')(b) = 0,\ (\beta_1, \beta_2) \ne (0, 0),\ \beta_1, \beta_2 \in \mathcal{R}. \end{aligned} \tag{1.4.20}$$

Definition 1.4.16. Let u, v be real solutions of (1.4.19). Then

- u is called a *principal solution* at a if
  1. $u(t) \ne 0$ for $x \in (a, d]$ and some $d \in I$,
  2. every solution y of (1.4.19) which is not multiple of u satisfies

$$u(x) = o(y(x)) \text{ as } x \to a.$$

- v is called a *non-principal solution* at a if

1. $v(t) \neq 0$ for $x \in (a, d]$ and some $d \in I$,
2. v is not a principal solution at a.

Principal and non-principal solutions in b are defined similarly.

We underline as boundary conditions of the form (1.4.20) do not make sense when one endpoint is singular, since if a is singular $y(a)$ does not exist, in general. In the following there are some results for singular boundary conditions (see [41]).

Remark 1.4.17. (See [41]) Let (1.4.19) hold.

- Suppose each endpoint is LP. Then no boundary conditions are needed nor allowed in this case.
- Suppose a is either R or LC and b is LP. Then no boundary condition is needed nor allowed at b and the self-adjoint boundary condition in a can be expressed as follow: let $u, v \in \Delta$ be real-valued such that $[u, v](a) \neq 0$ and

$$\alpha_1[u, y](a) + \alpha_2[v, y](a) = 0,\ \alpha_1, \alpha_2 \in \mathcal{R},\ (\alpha_1, \alpha_2) \neq (0, 0). \tag{1.4.21}$$

- Suppose a is LP and b is either R or LC. Then no boundary condition is needed or allowed at a and the self-adjoint boundary condition in b can be expressed as follow: let $u, v \in \Delta$ be real-valued such that $[u, v](b) \neq 0$ and

$$\beta_1[u, y](b) + \beta_2[v, y](b) = 0,\ \beta_1, \beta_2 \in \mathcal{R},\ (\beta_1, \beta_2) \neq (0, 0). \tag{1.4.22}$$

- Suppose a is R or LC and b is also R or LC. Then the boundary condition in a and b can be expressed as follow: let $u, v \in \Delta$ be real-valued such that $[u, v](a) \neq 0$, $[u, v](b) \neq 0$ and

$$\begin{aligned} &\alpha_1[u, y](a) + \alpha_2[v, y](a) = 0,\ \alpha_1, \alpha_2 \in \mathcal{R},\ (\alpha_1, \alpha_2) \neq (0, 0), \\ &\beta_1[u, y](b) + \beta_2[v, y](b) = 0,\ \beta_1, \beta_2 \in \mathcal{R},\ (\beta_1, \beta_2) \neq (0, 0). \end{aligned} \tag{1.4.23}$$

It is important to discuss on the properties of the eigenvalues and the eigenfunctions when the endpoints are regular or singular. In the following we summarize the results in [22, 41].

Remark 1.4.18.

- Let both endpoints a and b be, independently, R or LC, then:

  1. The spectrum is always discrete and bounded below. It consists of a countably infinite sequence $\{\lambda_n : n \in \mathcal{N}_0\}$ of real eigenvalues tending to $+\infty$, which can be ordered to satisfy
  $$-\infty < \lambda_0 \leq \lambda_1 \leq \lambda_2 \leq \cdots \to \infty \tag{1.4.24}$$
  The spectrum is simple, i.e. all eigenvalues have multiplicity one, in the case of separated boundary conditions; moreover the eigenfunction y_n corresponding to λ_n has exactly n zeros in the open interval (a, b) for any $n \in \mathcal{N}_0$.
  2. If either a or b, or both, are LCO then the spectrum is unbounded by below, i.e. $\{\lambda_n : n \in \mathcal{Z}\}$, $\lim_{n\to\pm\infty} \lambda_n = \pm\infty$ and
  $$-\infty < \cdots \leq \lambda_{-2} \leq \lambda_{-1} \leq < 0 \leq \lambda_1 \leq \lambda_2 \leq \cdots \to \infty. \tag{1.4.25}$$

- Let one or both endpoints a, b be LP, then:

  1. The spectrum is always simple but may or may not be discrete, and may or may not be bounded below.
  2. If the spectrum is discrete and bounded by below then $\{\lambda_n : n \in \mathcal{N}_0\}$, $\lim_{n\to+\infty} \lambda_n = +\infty$ and
  $$-\infty < \lambda_0 \leq \lambda_1 \leq \lambda_2 \leq \cdots \to \infty. \tag{1.4.26}$$
  The eigenfunction y_n has exactly n zeros in the open interval (a, b);
  3. If the continuous spectrum is bounded by below by σ, then:
     (a) there may be no eigenvalues below σ;
     (b) there may be a finite number of eigenvalues below σ, then $\{\lambda_n : n = 0, \ldots, N, N > 0\}$ and $\lambda_n < \lambda_{n+1} \leq \sigma$ for $n = 0, 1, \ldots, N - 1$. Every eigenfunction corresponding to λ_n has exactly n zeros in the open interval (a,b);
     (c) there may be a countable infinity of eigenvalues below, then $\lambda_n < \lambda_{n+1} \leq \sigma$ and $\lim_{n\to+\infty} = \sigma$. The eigenfunction corresponding to λ_n has exactly n zeros in the open interval (a,b);
  4. There are examples when a is R and b is LP for which the spectrum is discrete but unbounded above and below,then $\{\lambda_n : n \in \mathcal{Z}\}$ with $\lambda_n < \lambda_{n+1}$ for all $n \in \mathcal{Z}$ and $\lim_{n\to\pm\infty} \lambda_n = \pm\infty$. In such cases all eigenfunctions have infinitely many zeros.

1.5 Boundary Value Methods

Naturally, we are interested in approximating a continuous problem by means of a discrete one. In this section we present a particular class of methods called *Boundary Value Methods* **BVMs** developed by D. Trigiante, see[28], which are the basic idea of the methods employed in the codes we will show in the next chapters to solve all different classes of problems discussed previously.

First of all we consider a scalar initial value problem

$$\begin{aligned} &y' = f(x,y), \quad x \in [a,b] \\ &y(a) = y_0 \end{aligned} \tag{1.5.1}$$

where the function $f(x,y)$ is continuous and satisfies the Lipschitz condition for the existence and uniqueness of the solution. Afterwards, in order to replace the continuous problem by a discrete one, we construct a mesh of equidistant points $a = x_0 < x_1 < \cdots < x_n = b$, with $x_i = a + ih$, $i = 0, \ldots, n$, and $h = (b-a)/n$, so that $\{y_i\}_{i=0}^n$ is the numerical solution to seek.

We suppose to consider numerical methods belonging to the class of *Linear Multistep Formula* **LFM** defined as

$$\sum_{j=0}^{k} \alpha_j y_{n+j} - h \sum_{j=0}^{k} \beta_j f_{n+j} = 0, \tag{1.5.2}$$

where $f_n + j$ denotes $f(x_{n+j}, y_{n+j})$ and $\alpha_k = 1$. This means that the continuous initial value problem (1.5.1) is solved by means of a discrete initial value problem, that is k initial conditions $y_0, y_1, \ldots, y_{k-1}$ are associated to the formula (1.5.2). Since only y_0 is known, the remaining values $y_1, \ldots, y_{k-1}$ are called *additional conditions* and they need to be found. Since a discrete initial value problem is generated, then these methods are simply called *Initial Value Methods* **IVM**.

Alternatively, we can decide to impose k conditions for the equation (1.5.2) in a different set of the mesh points, that is if k_1 and k_2 are two integers such that $k = k_1 + k_2$, then we can fix the value as it follows

$$y_0, y_1, \ldots, y_{k_1-1}, y_{n-k_2+1}, \ldots, y_{n-1}, y_n.$$

Consequently, the continuous initial problem is approximated by a discrete boundary value problem, for this reason the obtained methods are called *Boundary Value Methods* (BVMs), with (k_1, k_2)-boundary conditions. Moreover, it is evident as k_1 *initial conditions* and k_2 *final conditions* are required

and as in the case $k_1 = k$ and $k_2 = 0$ the class of BVMs contains the class of IVMs.

The discrete problem generated by a $k-$step BVM con $(k_1, k_2)-$boundary conditions may be written in matrix form, hence considering the matrices

$$A_n = \begin{pmatrix} \alpha_{k_1} & \dots & \alpha_k & & & \\ \vdots & \ddots & & \ddots & & \\ \alpha_0 & & \ddots & & \ddots & \\ & \ddots & & \ddots & & \alpha_k \\ & & \ddots & & \ddots & \vdots \\ & & & \alpha_0 & \dots & \alpha_{k_1} \end{pmatrix}_{(n-k_1)\times(n-k_1)},$$

$$B_n = \begin{pmatrix} \beta_{k_1} & \dots & \beta_k & & & \\ \vdots & \ddots & & \ddots & & \\ \beta_0 & & \ddots & & \ddots & \\ & \ddots & & \ddots & & \beta_k \\ & & \ddots & & \ddots & \vdots \\ & & & \beta_0 & \dots & \beta_{k_1} \end{pmatrix}_{(n-k_1)\times(n-k_1)}$$

and the vectors $y = (y_{k_1}, \dots, y_{n-1})^T$ and $f = (f_{k_1}, \dots, f_{n-1})$, we obtain the following system

$$A_n y - hB_n f = - \begin{pmatrix} \sum_{i=0}^{k_1-1}(\alpha_i y_i - h\beta_i f_i) \\ \vdots \\ \alpha_0 y_{k_1-1} - h\beta_0 f_{k_1-1} \\ 0 \\ \vdots \\ 0 \\ \alpha_k y_n - h\beta_k f_n \\ \vdots \\ \sum_{i=1}^{k_2}(\alpha_{k_1+i} y_{n-1+i} - h\beta_{k_1+i} f_{n-1+i}) \end{pmatrix}.$$

We give the following definition (see [28]).

Definition 1.5.1. A *Toeplitz band matrix*, **T**-*matrix* hereafter, is a matrix whose entries $\{t_{i,j}\}$ satisfy

$$t_{i,j} = 0 \qquad \text{for} \quad i - j > m \quad \text{or} \quad i - j > k. \tag{1.5.3}$$

Then, the typical *finite T-matrix* will be of the form

$$T_n = \begin{pmatrix} a_0 & \dots & a_k & & & \\ \vdots & \ddots & & \ddots & & \\ a_{-m} & & \ddots & & \ddots & \\ & \ddots & & \ddots & & a_k \\ & & \ddots & & \ddots & \vdots \\ & & & a_{-m} & \dots & a_0 \end{pmatrix}_{n\times n}, \tag{1.5.4}$$

where we assume $a_{-m}a_k \neq 0$. As n varies, T_n will describe a family of T-matrices sharing the same band structure.

The infinite dimensional limit matrix, called *infinite T-matrix*, will be denoted by

$$T = \begin{pmatrix} a_0 & \dots & a_k & \\ \vdots & \ddots & & \ddots \\ a_{-m} & & \ddots & \\ & \ddots & & \ddots \end{pmatrix}. \tag{1.5.5}$$

Then, we can observe as the coefficient matrices A_n and B_n are T-matrices having lower bandwidth k_1, like the number of initial conditions, and upper bandwidth k_2, like the number of final conditions. The favorable stability properties of BVMs [28] are an advantage for their employment.

Chapter 2

High Order Generalized Difference Schemes

In this chapter methods using a generalization of the high order finite difference schemes, developed by Amodio and Sgura in [14], are introduced and sometimes referred as *High Order Generalized Difference* (HOGD) schemes. Following the traditional schemes for second order BVPs, the first and the second derivatives are approximated by high order difference schemes and at the same time the framework of boundary value methods (BVMs) is considered. Three classes of methods named D2ECDF, D2GFDF and D2GBDF are introduced and the conditioning analysis is carried out showing at the end as a generalization of the upwind method can be considered as we will see in the Chapter 3.

2.1 High order finite difference schemes

Let us consider a scalar two-point boundary value problem

$$\begin{aligned} &f(x, y, y', y'') = 0, \quad x \in [a, b] \\ &y(a) = \eta_1,\ y(b) = \eta_2, \end{aligned} \tag{2.1.1}$$

where f is a sufficiently smooth function and let us assume that a unique smooth solution exists. Moreover, we choose a uniform mesh $\pi : a = x_0 < x_1 < \cdots < x_n = b$, with $x_i = x_0 + ih$, $h = (b-a)/n$ and $i = 1, \ldots, n$. Then, a mesh function $y_\pi \equiv \{y_i\}_{i=0}^n$ such that $y_i \approx y(x_i)$, $i = 0, \ldots, n$ is sought. Set $k_1, k_2, k_3, k_4 \geq 1$ we choose to approximate the second derivative by a finite

difference scheme with k_1 initial and k_2 final conditions and in the same way the first derivative with k_3 and k_4 initial and final conditions as it follows

$$y''(x_i) \approx \frac{1}{h^2} \sum_{j=-k_1}^{k_2} \alpha_{k_1+j} y_{i+j} \quad i = k_1, \ldots, n - k_2 \tag{2.1.2}$$

and

$$y'(x_i) \approx \frac{1}{h} \sum_{j=-k_3}^{k_4} \beta_{k_3+j} y_{i+j} \quad i = k_3, \ldots, n - k_4. \tag{2.1.3}$$

The coefficients α_j and β_j have to be calculated in order to achieve formulae of maximum order.

Remark 2.1.1. If $k_s = 1$, for $s = 1, \ldots, 4$, then the equations (2.1.2) and (2.1.3) represent the classical finite differences for the second and the first derivatives respectively, that is

$$y''(x_i) \approx \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2},$$

$$y'(x_i) \approx \frac{y_{i+1} - y_{i-1}}{h},$$

that are formulae of order 2.

Supposing to calculate formulae of order $k_1 + k_2 > 2$ and $k_3 + k_4 > 2$, it stands to reason to require further $m = \max(k_1 + k_2 - 2, k_3 + k_4 - 2)$ additional values of the solution other than the known boundary conditions. In order to supply to this lack, based on the idea developed for the *Boundary Value Methods* (BVM), see Section 1.5, additional schemes of the same order $k_1 + k_2$ for the second derivative and $k_3 + k_4$ for the first derivative are defined as

$$\begin{aligned} y''(x_i) &\approx \frac{1}{h^2} \sum_{j=0}^{k_1+k_2+1} \alpha_j^{(i)} y_j \quad i = 1, \ldots, k_1 - 1, \\ y''(x_i) &\approx \frac{1}{h^2} \sum_{j=0}^{k_1+k_2+1} \alpha_j^{(i-m_1)} y_{j+m_1-1} \quad i = n - k_2 + 1, \ldots, n - 1, \end{aligned} \tag{2.1.4}$$

with $m_1 = n - k_1 - k_2$ and

$$
\begin{aligned}
y'(x_i) &\approx \frac{1}{h}\sum_{j=0}^{k_3+k_4} \beta_j^{(i)} y_j \quad i = 1, \ldots, k_3 - 1, \\
y'(x_i) &\approx \frac{1}{h}\sum_{j=0}^{k_3+k_4} \beta_j^{(i-m_3)} y_{j+m_3} \quad i = n - k_4 + 1, \ldots, n - 1,
\end{aligned}
\tag{2.1.5}
$$

where $m_3 = n - k_3 - k_4$. It is outstanding to point out that the coefficients $\alpha_j^{(i)}$ and $\beta_j^{(i)}$ are computed so that the schemes (2.1.4) and (2.1.5) have the same order of (2.1.2) and (2.1.3), respectively.

Now, using the previous schemes we denote with $y_i' \approx y'(x_i)$ and $y_i'' \approx y''(x_i)$ the first and the second derivative approximations in the points of the mesh π. It is obvious that the discretization of the boundary value problem (2.1.1) yields the system of equations

$$f(x_i, y_i, y_i', y_i'') = 0 \quad \text{for}\, i = 1, \ldots, n - 1.$$

The following results (see [1, 14]) show how the coefficients of the formulae (2.1.2), (2.1.3), (2.1.4) and (2.1.5) may be computed.

Proposition 2.1.2. *For all $\nu \leq p$, the coefficients of p-steps formulae*

$$y^{(\nu)}(x_{i+l}) \approx \frac{1}{h^\nu}\sum_{j=0}^{p} c_j^{(l,\nu)}\, y_{i+j}, \tag{2.1.6}$$

approximating the νth derivative of $y(x)$ in x_{i+l}, for $l = 0, \ldots, p$, are the entries of the $(l+1)$th column of the $(p+1) \times (p+1)$ matrix

$$C^{(\nu)} = V^{-1} K^\nu V, \tag{2.1.7}$$

where V is the Vandermonde matrix

$$V = \begin{pmatrix} 1 & 1 & 1 & \ldots & 1 \\ 0 & 1 & 2 & \ldots & p \\ 0 & 1 & 2^2 & \ldots & p^2 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 1 & 2^p & \ldots & p^p \end{pmatrix}, \tag{2.1.8}$$

and

$$K = \begin{pmatrix} 0 & & & & \\ 1 & 0 & & & \\ & 2 & \ddots & & \\ & & \ddots & \ddots & \\ & & & p & 0 \end{pmatrix}. \tag{2.1.9}$$

Moreover, the formula (2.1.6) *has order* $p - \nu + 1$ *for* $\nu = 1, 2$.

Proof. The conditions for the consistency are obtained by substituting the value of the solution $y(x)$ at the grid points in (2.1.6), that is

$$y^{(\nu)}(x_{i+l}) - \frac{1}{h^\nu} \sum_{j=0}^{p} c_j^{(l,\nu)}\, y(x_{i+j}) = \tau_{i+l}, \tag{2.1.10}$$

Now it is possible to impose that the truncation error τ_{i+l} is at least $O(h^{p-\nu+1})$ by expanding (2.1.10) at $x = x_i$ as follows

$$\begin{aligned} y^{(\nu)}(x_{i+l}) - \frac{1}{h^\nu} \sum_{j=0}^{p} c_j^{(l,\nu)}\, y(x_{i+j}) = \sum_{s=\nu}^{p} \frac{(lh)^{s-\nu}}{(s-\nu)!} y^{(s)}(x_i) + \\ - \frac{1}{h^\nu} \sum_{j=0}^{p} c_j^{(l,\nu)} \sum_{s=0}^{p} \frac{j^s}{s!} h^s y^{(s)}(x_i) + O(h^{p-\nu+1}) \end{aligned} \tag{2.1.11}$$

Consequently the conditions required are

$$\begin{aligned} &\sum_{j=0}^{p} j^s c_j^{(l,\nu)} = 0, \quad 0 \le s \le \nu - 1, \\ &\sum_{j=0}^{p} j^s c_j^{(l,\nu)} = \frac{s!}{(s-\nu)!}\, l^{s-\nu}, \quad \nu \le s \le p. \end{aligned} \tag{2.1.12}$$

For each $l = 0, \ldots, p$ these relations in vector form show as the coefficients $c_0^{(l,\nu)}, \ldots, c_p^{(l,\nu)}$ are computed by solving the linear system

$$Vc = (K^\nu V) e_{l+1}, \tag{2.1.13}$$

where $c = (c_0^{(l,\nu)}, \ldots, c_p^{(l,\nu)})$ and e_l is the lth unit vector in $\mathcal{R}^{p+1}$. The entries of the Vandermonde matrix are defined, for $i, j = 0, \ldots, p$, as

$$V_{i,j} = (j)^i,$$

and from (2.1.9)

$$K_{i,j} = \begin{cases} i & i-j=1, \\ 0 & \text{otherwise.} \end{cases} \tag{2.1.14}$$

For induction it is possible to prove that

$$K^{\nu}_{i,j} = \begin{cases} i(i-1)\dots(i-\nu+1) & i-j=\nu, \\ 0 & \text{otherwise.} \end{cases} \tag{2.1.15}$$

The relation (2.1.15) is trivial when $\nu = 1$. Supposed true for $\nu - 1$, check for a generic ν. Let us note that $K^{\nu} = KK^{\nu-1}$, then using the definition (2.1.14) and the hypothesis of induction it follows that

$$K^{\nu}_{i,j} = \sum_{s=0}^{p} K_{i,s}K^{\nu-1}_{s,j} = \begin{cases} i(i-1)\dots(i-\nu+1) & i-j=\nu, \\ 0 & \text{otherwise.} \end{cases}$$

Then for $s = 0, \dots, p$ we are able to compute the sth entries of the right-side of (2.1.13)

$$\begin{aligned} ((K^{\nu}V)e_{l+1})_s &= \begin{cases} s(s-1)\dots(s-\nu+1)l^{s-\nu} & \nu \le s \le p, \\ 0 & 0 \le s \le \nu-1. \end{cases} \\ &= \begin{cases} \dfrac{s(s-1)\dots(s-\nu+1)(s-\nu)!}{(s-\nu)!}\, l^{s-\nu} & \nu \le s \le p, \\ 0 & 0 \le s \le \nu-1. \end{cases} \\ &= \begin{cases} \dfrac{s!}{(s-\nu)!}\, l^{s-\nu} & \nu \le s \le p, \\ 30 & 0 \le s \le \nu-1. \end{cases} \end{aligned}$$

Hence (2.1.13) is checked and consequently it follows that the $(l+1)$th column of the matrix $C^{(\nu)}$ in (2.1.7) contains the coefficients of a scheme of order $p - \nu + 1$ with l initial conditions. □

Proposition 2.1.3. *Let $C^{(\nu)}$ be the coefficient matrix defined in the Proposition 2.1.2, then*

$$C^{(\nu)} = C^{\nu}, \quad \textit{for all} \quad \nu \le p. \tag{2.1.16}$$

Moreover, let us consider the permutation matrix

$$J_p = \begin{pmatrix} & & & 1 \\ & & 1 & \\ & . & & \\ . & & & \\ 1 & & & \end{pmatrix}, \tag{2.1.17}$$

then the matrix $C^{(\nu)}$ *satisfies*

$$J_p C^{(\nu)} J_p = (-1)^\nu C^{(\nu)}, \tag{2.1.18}$$

namely it is centro-symmetric for even derivatives and skew-centro-symmetric for odd derivatives. Therefore, the coefficients of the j*th column are the same of those of the* $(p+2-j)$*th column, but in reverse order, and also changed in sign for the odd derivatives.*

PROOF. Set $C^{(1)} = C$ and considered (2.1.7) it follows that

$$\begin{aligned} C^{(\nu)} = V^{-1}K^\nu V &= \underbrace{V^{-1}KV\,V^{-1}KV\ldots\ldots V^{-1}KV}_{\nu} \\ &= \underbrace{CCC\ldots C}_{\nu} = C^\nu. \end{aligned}$$

It is known that the permutation matrices are orthogonal. Since the permutation matrix defined in (2.1.17) is symmetric, consequently it is clear that

$$J_p J_p = I_{p+1}. \tag{2.1.19}$$

Moreover $J_p C J_p = -C$ (see [1]), then using the (2.1.16) and (2.1.19) the (2.1.18) is checked as follows

$$J_p C^{(\nu)} J_p = J_p C^\nu J_p = \underbrace{J_pCJ_p\,J_pCJ_p\ldots\ldots J_pCJ_p}_{\nu} = (-1)^\nu C^{(\nu)}.$$

□

Proposition 2.1.4. *For* $\nu = 2$ *the formula* (2.1.6) *has order* p *even when* $l = p/2$*.*

PROOF. We consider the expansion of (2.1.6) at $x = x_{i+p/2}$, then the truncation error is defined as

$$\tau_{i+p/2} = \sum_{j=0}^{p} c_j^{(p/2,2)} \frac{(j-p/2)^{p+1}}{(p+1)!} h^{p-1} \tag{2.1.20}$$

As pointed out in the Proposition 2.1.3 for the even derivatives the coefficients are centro-symmetric, that is

$$c_j^{(p/2,2)} = c_{p-j}^{(p/2,2)} \quad j = 0, \ldots, p/2-1.$$

Then, it follows that

$$\begin{aligned}\sum_{j=0}^{p} c_j^{(p/2,2)} \frac{(j-p/2)^{p+1}}{(p+1)!} &= \sum_{j=0}^{p/2-1} c_j^{(p/2,2)} \frac{(j-p/2)^{p+1}}{(p+1)!} + \sum_{j=p/2+1}^{p} c_j^{(p/2,2)} \frac{(j-p/2)^{p+1}}{(p+1)!} \\ &= \sum_{j=0}^{p/2-1} c_j^{(p/2,2)} \frac{(j-p/2)^{p+1}}{(p+1)!} + \sum_{j=0}^{p/2-1} c_{p-j}^{(p/2,2)} \frac{(p/2-j)^{p+1}}{(p+1)!} \\ &= (-1)^{p+1} \sum_{j=0}^{p/2-1} c_j^{(p/2,2)} \frac{(p/2-j)^{p+1}}{(p+1)!} \\ &+ \sum_{j=0}^{p/2-1} c_{p-j}^{(p/2,2)} \frac{(p/2-j)^{p+1}}{(p+1)!} = 0.\end{aligned} \tag{2.1.21}$$

Hence, from (2.1.20) and (2.1.21) it achieves that $\tau_{i+p/2} = O(h^p)$, when p is even. □

Example 2.1.5. For $p = 4$ and $\nu = 1, 2$ the coefficients matrices in (2.1.7) are expressed as

$$C^{(1)} = \begin{pmatrix} -\frac{25}{12} & -\frac{1}{4} & \frac{1}{12} & -\frac{1}{12} & \frac{1}{4} \\ 4 & -\frac{5}{6} & -\frac{2}{3} & \frac{1}{2} & -\frac{4}{3} \\ -3 & \frac{3}{2} & 0 & -\frac{3}{2} & 3 \\ \frac{4}{3} & -\frac{1}{2} & \frac{2}{3} & \frac{5}{6} & -4 \\ -\frac{1}{4} & \frac{1}{12} & -\frac{1}{12} & \frac{1}{4} & \frac{25}{12} \end{pmatrix},$$

$$C^{(2)} = \begin{pmatrix} \frac{35}{12} & \frac{11}{12} & -\frac{1}{12} & -\frac{1}{12} & \frac{11}{12} \\ -\frac{26}{3} & -\frac{5}{3} & \frac{4}{3} & \frac{1}{3} & -\frac{14}{3} \\ \frac{19}{2} & \frac{1}{2} & -\frac{5}{2} & \frac{1}{2} & \frac{19}{2} \\ -\frac{14}{3} & \frac{1}{3} & \frac{4}{3} & -\frac{5}{3} & -\frac{26}{3} \\ \frac{11}{12} & -\frac{1}{12} & -\frac{1}{12} & \frac{11}{12} & \frac{35}{12} \end{pmatrix}.$$

Proposition 2.1.2 affirms that the formulae using the coefficients matrix $C^{(1)}$ have order 4, while for the approximation of the second derivative only the formula using the entries of the 3rd column of $C^{(2)}$ has order 4. It points out as the coefficients of the 3rd column give back central differences formulae of order 4. Obviously, compute formulae of order 4 for the second derivative with zero and one initial or final condition means to consider the entries of the 1st and 2nd column of $C^{(2)}$ or the $(p+1)$th and pth ones respectively, where $p = 5$ and the coefficients matrix is defined as

$$C^{(2)} = \begin{pmatrix} \frac{15}{4} & \frac{5}{6} & -\frac{1}{12} & 0 & \frac{1}{12} & -\frac{5}{6} \\ -\frac{77}{6} & -\frac{5}{4} & \frac{4}{3} & -\frac{1}{12} & -\frac{1}{2} & \frac{61}{12} \\ \frac{107}{6} & -\frac{1}{3} & -\frac{5}{2} & \frac{4}{3} & \frac{7}{6} & -13 \\ -13 & \frac{7}{6} & \frac{4}{3} & -\frac{5}{2} & -\frac{1}{3} & \frac{107}{6} \\ \frac{61}{12} & -\frac{1}{2} & -\frac{1}{12} & \frac{4}{3} & -\frac{5}{4} & -\frac{77}{6} \\ -\frac{5}{6} & \frac{1}{12} & 0 & -\frac{1}{12} & \frac{5}{6} & \frac{15}{4} \end{pmatrix}.$$

Since a BVP (2.1.1) has one initial and one final condition, it is not necessary to use the first and the last column of the matrix $C^{(\nu)}$, differently an initial value problem (IVP), as we could see in Chapter 4, has an initial condition, consequently only the first column of the matrix $C^{(\nu)}$ is not considered. For the BVP the remaining $p-1$ formulae enable to define several BVMs which are characterized by the choice to impose k conditions for the equation (2.1.1) on a different set of grid points.

Now, we give a general definition for the approach introduced.

Definition 2.1.6. Given $n+1$ equidistant points $x_0, x_1, \ldots, x_n$ in the interval $[a, b]$, a global approximation $Y^{(\nu)}$ of order p for the ν-th derivative $y^{(\nu)}(x_i)$, $i+1, \ldots, n$, using a main method with k initial conditions, is given by the vector

$$Y^{(\nu)} = \frac{1}{h^\nu} \tilde{A}_\nu \tilde{Y}, \tag{2.1.22}$$

where $Y = [y_1, \ldots, y_{n-1}]^T$ is the unknown vector, $\tilde{Y} = [y_0, Y^T, y_n]$, $y_i \approx y(x_i)$ and

$$\tilde{A}_\nu = \left(\begin{array}{c|cccc|ccccc|c}
\alpha_0^{(1,\nu)} & \alpha_1^{(1,\nu)} & \ldots & \alpha_{p+\nu-2}^{(1,\nu)} & \alpha_{p+\nu-1}^{(1,\nu)} & & & & & & \\
\vdots & \vdots & & \vdots & \vdots & & & & & & \\
\alpha_0^{(k-1,\nu)} & \alpha_1^{(k-1,\nu)} & \ldots & \alpha_{p+\nu-2}^{(k-1,\nu)} & \alpha_{p+\nu-1}^{(k-1,\nu)} & & & & & & \\
\alpha_0^{(k,\nu)} & \alpha_1^{(k,\nu)} & \ldots & \alpha_p^{(k,\nu)} & & & & & & & \\
 & \alpha_0^{(k,\nu)} & \alpha_1^{(k,\nu)} & \ldots & \alpha_p^{(k,\nu)} & & & & & & \\
 & & \ddots & & \ddots & & & & & & \\
 & & & \ddots & & \ddots & & & & & \\
 & & & & \ddots & & \ddots & & & & \\
 & & & & & \alpha_0^{(k,\nu)} & \ldots & \alpha_{k-1}^{(k,\nu)} & \alpha_p^{(k,\nu)} \\
 & & & & \alpha_0^{(k+1,\nu)} & \alpha_1^{(k+1,\nu)} & \ldots & \alpha_{p+\nu-2}^{(k+1,\nu)} & \alpha_{p+\nu-2}^{(k+1,\nu)} \\
 & & & & \vdots & \vdots & & \vdots & \vdots \\
 & & & & \alpha_0^{(p-1,\nu)} & \alpha_1^{(p-1,\nu)} & \ldots & \alpha_{p+\nu-2}^{(p-1,\nu)} & \alpha_{p+\nu-2}^{(p-1,\nu)}
\end{array}\right) \tag{2.1.23}$$

is a $(n-1) \times (n+1)$ quasi-Toeplitz matrix whose coefficients are defined in (2.1.22) and given according to Proposition 2.1.2. In particular

$$\tilde{A}_\nu = (\mathbf{a}_0^{(\nu)}, A_\nu, \mathbf{a}_k^{(\nu)}),$$

where A_ν is a $n \times n$ quasi-Toeplitz band matrix, and the first and the last columns $\mathbf{a}_0^{(\nu)}$ and $\mathbf{a}_k^{(\nu)}$ are needed to deal with the (separated) boundary conditions.

It may observe that in (2.1.22) and (2.1.23) the *main scheme* has the coefficients $(\alpha_0^{(k,\nu)}, \alpha_1^{(k,\nu)}, \ldots, \alpha_p^{(k,\nu)})$ and computes the approximations at the points x_i, for $i = k, \ldots, n-k-p$, while the approximations at the initial points x_i, for $i = 1, \ldots, k-1$ are given by $(\alpha_0^{(i,\nu)}, \alpha_1^{(i,\nu)}, \ldots, \alpha_{p+\nu-1}^{(i,\nu)})$, coefficients of the *initial methods*. *Final methods* have coefficients $(\alpha_0^{(i,\nu)}, \alpha_1^{(i,\nu)}, \ldots, \alpha_{p+\nu-2}^{(i,\nu)})$,

for $i = k+1, \ldots, p-1$, and compute the approximations at the last points x_{n-p+i}.

2.2 D2ECDF, D2GBDF and D2GFDF

It is known that the second derivative represents the symmetric part of the convection-diffusion operator associated with the ODE-BVP (2.1.1), since the odd-order schemes lose the global symmetry by requiring a different number of initial methods from final ones, then they are not taken in consideration for the approximation of $y''(x)$. Consequently, it is chosen an even order $p = 2k$ and a BVM main scheme with k initial conditions. The coefficients of the $(k+1)$th column of $C^{(2)}$ are used, since it is known from the Proposition 2.1.3 to be centro-symmetric, this is $\alpha_i^{(k,2)} = \alpha_{2k-i}^{(k,2)}$. Obviously, in this case $k_1 = k_2 = k$ and $k-1$ initial and final schemes fulfill the property

$$\alpha_i^{(j,\,2)} = \alpha_{2k+1-i}^{(2k+1-j,\,2)}, \qquad i = 0, \ldots, 2k+1, \qquad j = 1, \ldots, k-1.$$

The method obtained is an even-order generalization of the *central difference scheme*, which have been introduced in [5] for solving the Hamiltonian problems.

Definition 2.2.1. Let $Y^{(2)}$ be the global approximation of the second derivative y'' in Definition 2.1.6. The main scheme of order p (even) given by (2.1.22)-(2.1.23) with $k = p/2$ initial conditions, is called **D2**.

In regard of the D2 schemes, BVMs of even-order $p = 2k$ are considered to approximate the first derivative. This choice allows to identify three classes of methods according to the main scheme.

The first class contains a generalization of the midpoint formula for IVPs introduced in [28] and it is defined for $k_3 = k_4 = k$, this means that the main BVM is obtained by choosing the $(k+1)$th column of the matrix $C^{(1)}$. It is known from the Proposition 2.1.3 that the matrix $C^{(1)}$ is skew-centro-symmetric, consequently it is $\alpha_i^{(k,1)} = -\alpha_{2k-i}^{(k,1)}$. Moreover, the initial and final schemes have to satisfy the following property

$$\alpha_i^{(j,\,1)} = -\alpha_{2k-i}^{(2k-j,\,1)}, \qquad i = 0, \ldots, 2k, \qquad j = 1, \ldots, k-1.$$

The method of even-order defined is an extension of classical *central difference scheme* [38] for the first derivative, called *Extended Central Difference Formulae* (**ECDFs**).

The second class of methods is defined for $k_3 = k_2 + 2 = k + 1$ and consequently the $(k+2)$th column of the matrix $C^{(1)}$ is chosen to define the main BVM scheme. In this case, k initial schemes and $k-2$ final ones are necessary and it can be noted as the first $(k-1)$ initial methods are the same of ECDF, while the kth is just the main method of ECDF. For this class of methods every symmetry is lost, moreover these schemes can be considered as an extension of the classical first-order *backward difference scheme*.

The third class of formulae is defined for $k_4 = k_3 + 2 = k + 1$ and it is obtained by considering the kth column of the matrix $C^{(1)}$ as main scheme. It is plain that $k-2$ initial schemes and k final ones are required to define the overall method and the first final method corresponds to the main scheme of ECDF, while the remaining $k-1$ coincide with the final methods of ECDF. These schemes lose every symmetry, besides they can be considered as an even-order extension of the classical first-order *forward difference scheme*. Hereafter it will be assumed the following definitions.

Definition 2.2.2. Let $Y^{(1)}$ be the global approximation of the first derivative y' in Definition 2.1.6. The formula of order p (even) given by (2.1.22)-(2.1.23) is called

- *Extended Central Difference Formula* (**ECDF**) if $k = p/2$;
- *Generalized Backward Difference Formula* (**GBDF**) if $k = p/2 + 1$;
- *Generalized Forward Difference Formula* (**GFDF**) if $k = p/2 - 1$;

where k is the number of initial conditions of the main scheme.

Definition 2.2.3. For the solution of second-order ODE (2.1.1), the combinations of the D2 scheme for y'' with the three classes of methods in Definition 2.2.2 for y' are called **D2ECDF**, **D2GBF** and **D2GFDF**, respectively.

Moreover, hereafter the methods in Definition 2.2.3 will be named as *High Order Generalized Difference* (**HOGD**). In the tables Table 2.1 and Table 2.2 the coefficients of the even-order formulae used to approximate the second and the first derivative are shown, for $p = 4, 6, 8, 10$. Because of the symmetry, the formulae of order p with $p - j$ initial conditions, $j = 1, \ldots, p/2 - 1$, are omitted.

Table 2.1: Coefficients $\alpha_i^{(j,2)}$ $(i = 0, \ldots, p+1)$ to approximate the second derivative with j initial conditions $(j = 1, \ldots, p/2)$. The main BVM schemes are typed bold.

p	j	α_0	α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8	α_9	α_{10}	α_{11}
2	**1**	1	-2	1									
4	1	$\frac{5}{6}$	$-\frac{5}{4}$	$-\frac{1}{3}$	$\frac{7}{6}$	$-\frac{1}{2}$	$\frac{1}{12}$						
4	**2**	$-\frac{1}{12}$	$\frac{4}{3}$	$-\frac{5}{2}$	$\frac{4}{3}$	$-\frac{1}{12}$							
6	1	$\frac{7}{10}$	$-\frac{7}{18}$	$-\frac{27}{10}$	$\frac{19}{4}$	$-\frac{67}{18}$	$\frac{9}{5}$	$-\frac{1}{2}$	$\frac{11}{180}$				
6	2	$-\frac{11}{180}$	$\frac{107}{90}$	$-\frac{21}{10}$	$\frac{13}{18}$	$\frac{17}{36}$	$-\frac{3}{10}$	$\frac{4}{45}$	$-\frac{1}{90}$				
6	**3**	$\frac{1}{90}$	$-\frac{3}{20}$	$\frac{3}{2}$	$-\frac{49}{18}$	$\frac{3}{2}$	$-\frac{3}{20}$	$\frac{1}{90}$					
8	1	$\frac{761}{1260}$	$\frac{61}{144}$	$-\frac{201}{35}$	$\frac{341}{30}$	$-\frac{1163}{90}$	$\frac{411}{40}$	$-\frac{17}{3}$	$\frac{1303}{630}$	$-\frac{9}{20}$	$\frac{223}{5040}$		
8	2	$-\frac{223}{5040}$	$\frac{293}{280}$	$-\frac{395}{252}$	$-\frac{13}{30}$	$\frac{83}{40}$	$-\frac{319}{180}$	$\frac{59}{60}$	$-\frac{5}{14}$	$\frac{389}{5040}$	$-\frac{19}{2520}$		
8	3	$\frac{19}{2520}$	$-\frac{67}{560}$	$\frac{97}{70}$	$-\frac{89}{36}$	$\frac{23}{20}$	$\frac{7}{40}$	$-\frac{17}{90}$	$\frac{11}{140}$	$-\frac{1}{56}$	$\frac{1}{560}$		
8	**4**	$-\frac{1}{560}$	$\frac{8}{315}$	$-\frac{1}{5}$	$\frac{8}{5}$	$-\frac{205}{72}$	$\frac{8}{5}$	$-\frac{1}{5}$	$\frac{8}{315}$	$-\frac{1}{560}$			
10	1	$\frac{671}{1260}$	$\frac{29513}{25200}$	$-\frac{2341}{252}$	$\frac{3601}{168}$	$-\frac{4021}{126}$	$\frac{4231}{120}$	$-\frac{4357}{150}$	$\frac{4441}{252}$	$-\frac{643}{84}$	$\frac{2273}{1008}$	$-\frac{509}{1260}$	$\frac{419}{12600}$
10	2	$-\frac{419}{12600}$	$\frac{5869}{6300}$	$-\frac{737}{720}$	$-\frac{829}{420}$	$\frac{2089}{420}$	$-\frac{2509}{450}$	$\frac{2719}{600}$	$-\frac{569}{210}$	$\frac{2929}{2520}$	$-\frac{61}{180}$	$\frac{1517}{25200}$	$-\frac{31}{6300}$
10	3	$\frac{31}{6300}$	$-\frac{1163}{12600}$	$\frac{1583}{1260}$	$-\frac{2123}{1008}$	$\frac{97}{210}$	$\frac{323}{300}$	$-\frac{463}{450}$	$\frac{533}{840}$	$-\frac{23}{84}$	$\frac{67}{840}$	$-\frac{89}{6300}$	$\frac{29}{25200}$
10	4	$-\frac{29}{25200}$	$\frac{59}{3150}$	$-\frac{53}{315}$	$\frac{317}{210}$	$-\frac{6743}{2520}$	$\frac{103}{75}$	$\frac{1}{75}$	$-\frac{37}{315}$	$\frac{109}{1680}$	$-\frac{13}{630}$	$\frac{2}{525}$	$-\frac{1}{3150}$
10	**5**	$\frac{1}{3150}$	$-\frac{5}{1008}$	$\frac{5}{126}$	$-\frac{5}{21}$	$\frac{5}{3}$	$-\frac{5269}{1800}$	$\frac{5}{3}$	$-\frac{5}{21}$	$\frac{5}{126}$	$-\frac{5}{1008}$	$\frac{1}{3150}$	

Table 2.2: Coefficients $\alpha_i^{(j,1)}$ $(i = 0, \ldots, p)$ to approximate the first derivative with j initial conditions ($j = 1, \ldots, p/2$).

p	j	β_0	β_1	β_2	β_3	β_4	β_5	β_6	β_7	β_8	β_9	β_{10}
2	1	$-\frac{1}{2}$	0	$-\frac{1}{2}$								
4	1	$-\frac{1}{4}$	$-\frac{5}{6}$	$\frac{3}{2}$	$-\frac{1}{2}$	$\frac{1}{12}$						
4	2	$\frac{1}{12}$	$-\frac{2}{3}$	0	$\frac{2}{3}$	$-\frac{1}{12}$						
6	1	$-\frac{1}{6}$	$-\frac{77}{60}$	$\frac{5}{2}$	$-\frac{5}{3}$	$\frac{5}{6}$	$-\frac{1}{4}$	$\frac{1}{30}$				
6	2	$\frac{1}{30}$	$-\frac{2}{5}$	$-\frac{7}{12}$	$\frac{4}{3}$	$-\frac{1}{2}$	$\frac{2}{15}$	$-\frac{1}{60}$				
6	3	$-\frac{1}{60}$	$\frac{3}{20}$	$-\frac{3}{4}$	0	$\frac{3}{4}$	$-\frac{3}{20}$	$\frac{1}{60}$				
8	1	$-\frac{1}{8}$	$-\frac{223}{140}$	$\frac{7}{2}$	$-\frac{7}{2}$	$\frac{35}{12}$	$-\frac{7}{4}$	$\frac{7}{10}$	$-\frac{1}{6}$	$\frac{1}{56}$		
8	2	$\frac{1}{56}$	$-\frac{2}{7}$	$-\frac{19}{20}$	2	$-\frac{5}{4}$	$\frac{2}{3}$	$-\frac{1}{4}$	$\frac{2}{35}$	$-\frac{1}{168}$		
8	3	$-\frac{1}{168}$	$\frac{1}{14}$	$-\frac{1}{2}$	$-\frac{9}{20}$	$\frac{5}{4}$	$-\frac{1}{2}$	$\frac{1}{6}$	$-\frac{1}{28}$	$\frac{1}{280}$		
8	4	$\frac{1}{280}$	$-\frac{4}{105}$	$\frac{1}{5}$	$-\frac{4}{5}$	0	$\frac{4}{5}$	$-\frac{1}{5}$	$\frac{4}{105}$	$-\frac{1}{280}$		
10	1	$-\frac{1}{10}$	$-\frac{4609}{2520}$	$\frac{9}{2}$	$-\frac{6}{7}$	$-\frac{63}{10}$	$\frac{21}{5}$	-2	$\frac{9}{14}$	$-\frac{1}{8}$	$\frac{1}{90}$	
10	2	$\frac{1}{90}$	$-\frac{2}{9}$	$-\frac{341}{280}$	$\frac{8}{3}$	$-\frac{7}{3}$	$\frac{28}{15}$	$-\frac{7}{6}$	$\frac{8}{15}$	$-\frac{1}{6}$	$\frac{2}{63}$	$-\frac{1}{360}$
10	3	$-\frac{1}{360}$	$\frac{1}{24}$	$-\frac{3}{8}$	$-\frac{319}{420}$	$\frac{7}{4}$	$-\frac{21}{20}$	$\frac{7}{12}$	$-\frac{1}{4}$	$\frac{3}{40}$	$-\frac{1}{72}$	$\frac{1}{840}$
10	4	$\frac{1}{840}$	$-\frac{1}{63}$	$\frac{3}{28}$	$-\frac{4}{7}$	$-\frac{11}{30}$	$\frac{6}{5}$	$-\frac{1}{2}$	$\frac{4}{21}$	$-\frac{3}{56}$	$\frac{1}{105}$	$-\frac{1}{1260}$
10	5	$-\frac{1}{1260}$	$\frac{5}{504}$	$-\frac{5}{84}$	$\frac{5}{21}$	$-\frac{5}{6}$	0	$\frac{5}{6}$	$-\frac{5}{21}$	$\frac{5}{84}$	$-\frac{5}{504}$	$\frac{1}{1260}$

Time-reversal symmetry

Concerning the BVP there is an additional requirement that it is considered to be important. Numerical methods for BVPs essentially integrate the continuous problem forward and backward simultaneously but, since the continuous problem does not exhibit a preferential direction in time, it may be preferable that the numerical methods behave alike.

Let us consider the following linear and homogenous scalar problem

$$y'' - 2\gamma y' + \mu y = 0 \tag{2.2.1}$$

with constant real coefficients γ and μ and with separated boundary conditions $y(a) = \eta_1$ and $y(b) = \eta_2$. Supposed $\delta = \gamma^2 - \mu > 0$, the exact solution of (2.2.1) is given by

$$y(x) = c_1 e^{\lambda_1 x} + c_2 e^{\lambda_2 x},$$

where $\lambda_1 = \gamma + \sqrt{\delta}$, $\lambda_2 = \gamma - \sqrt{\delta}$ and the coefficients c_1 and c_2 depending on the boundary conditions are expressed by

$$c_1 = \frac{\eta_1 e^{\lambda_2 b} - \eta_2 e^{\lambda_2 a}}{e^{\lambda_1 a} e^{\lambda_2 b} - e^{\lambda_2 a} e^{\lambda_1 b}}, \qquad c_2 = \frac{\eta_2 e^{\lambda_1 a} - \eta_1 e^{\lambda_1 b}}{e^{\lambda_1 a} e^{\lambda_2 b} - e^{\lambda_2 a} e^{\lambda_1 b}}.$$

Let us suppose that the independent variable is changed for $\tau = a + b - x$, then set $y(x) = u(\tau)$ with $\tau \in [a, b]$, the equation (2.2.1) is transformed in

$$u'' + 2\gamma u' + \mu u = 0 \tag{2.2.2}$$

where $u = u(\tau)$. Moreover, the boundary conditions, like so the interval of integration, are reversed and defined as $u(a) = y(b) = \eta_2$ and $u(b) = y(a) = \eta_1$. The two parametric curves $C_\tau : \{(\tau, u(\tau)), \tau \in [a, b]\}$ and $C_x : \{(x, y(x)), x \in [a, b]\}$ coincide, this means that a point is getting ahead on C_x, it is moving back on C_τ and viceversa. Then we can affirm that the solution of the BVP (2.2.1) satisfies the property called *time isotropy* or *time-reversal symmetry* (see [28]). It would be important for the numerical methods as follows.

Remark 2.2.4 (**Time reversal symmetry condition**). *A numerical method must provide the same discrete approximation on the interval $[a, b]$ when the independent variable x of the continuous problem is transformed into $\tau = a + b - x$ and the boundary conditions are changed accordingly.*

Now, we derive the conditions so that the coefficients of the BVMs (see section 2.2) preserve the time reversal symmetry. Considering the Definition

2.1.6, let $Y = [y_1, y_2, \ldots, y_{n-1}]^T$ be the unknown solution vector, then in the vector form the approximations of the derivatives by the BVMs introduced in the previous sections are given by

$$Y''(x) \approx \frac{1}{h^2}\tilde{A}_2\tilde{Y}, \qquad Y'(x) \approx \frac{1}{h}\tilde{A}_1\tilde{Y},$$

where $\tilde{Y} = [y_0, Y^T, y_n]^T$ and $\tilde{A}_2$ and $\tilde{A}_1$ are the $(n-1)\times(n+1)$ matrices containing the coefficients of the formulae (2.1.6) for $\nu = 1, 2$.

Then the equivalent discrete problem of (2.2.1) is

$$(\tilde{A}_2 - 2\gamma h\tilde{A}_1 + \mu h^2\tilde{I})\tilde{Y} = 0, \tag{2.2.3}$$

where $\tilde{I} = [0_{n-1}, I_{n-1}, 0_{n-1}]$, 0_{n-1} is a null vector of length $n-1$ and I_{n-1} is the $(n-1)\times(n-1)$ identity matrix. Similarly, set $\tau_i = \tau_0 - ih$ and $u_i \approx u(\tau_i)$ for $i = 0, \ldots, n$, let $U = [u_1, u_2, \ldots, u_{n-1}]^T$ be the unknown solution vector of $u(\tau)$ and $\tilde{U} = [u_0, U^T, u_n]^T$, then using in reverse order the same methods the numerical approximation of (2.2.2) is given by

$$(J_{n-1}\tilde{A}_2 + 2\gamma h J_{n-1}\tilde{A}_1 + \mu h^2 J_{n-1}\tilde{I})\tilde{U} = 0, \tag{2.2.4}$$

where J_{n-1} is a permutation matrix defined in (2.1.17).

Now, we impose to the numerical method to satisfy the time reversal symmetry condition, this means that if

$$y(x_i) = u(a + b - \tau_i) \qquad \text{for} \quad i = 0, \ldots, n, \tag{2.2.5}$$

since

$$x_i = a + b - \tau_i = \tau_0 - (n-i)h = \tau_{n-i},$$

the condition (2.2.5) is approximated by

$$y_i = u_{n-i} \qquad \text{for} \quad i = 0, \ldots, n.$$

In vector form this is equivalent to require

$$\tilde{Y} = J_{n+1}\tilde{U}. \tag{2.2.6}$$

Consequently, (2.2.3) can be written as

$$(\tilde{A}_2 J_{n+1} - 2\gamma h\tilde{A}_1 J_{n+1} + \mu h^2\tilde{I}J_{n+1})\tilde{U} = 0$$

and the comparison with (2.2.4) states that the matrices $\tilde{A}_2$ and $\tilde{A}_1$ have to fulfill

$$\tilde{A}_2 = J_{n-1}\tilde{A}_2 J_{n+1}, \qquad \tilde{A}_1 = -J_{n-1}\tilde{A}_1 J_{n+1}. \tag{2.2.7}$$

Hence, in regard of the relations in (2.2.7), the numerical method is isotropic when the coefficients of the matrices $\tilde{A}_2$ and $\tilde{A}_1$ satisfy, for $i = 1, \ldots, n-1$ and $j = 1, \ldots, n+1$ the property:

$$\alpha^{(2)}_{i,j} = \alpha^{(2)}_{n-1-i,n+2-j} \qquad \alpha^{(1)}_{i,j} = -\alpha^{(1)}_{n-1-i,n+2-j}.$$

Consequently, it is clear that a numerical method preserve the time reversal symmetry when:

- the BVM main scheme is symmetric for the approximation of the y'' and skew-symmetric for the y';
- the number of initial and final methods is the same;
- the coefficients of the ith initial scheme are these of the $n-i$ final one in reverse order for y'' and also with changed sign for y'.

Since D2ECDF satisfy the property (2.2.7), we can affirm that the scheme D2ECDF is time isotropic. As a consequence, we point out that also the midpoint rule, obtained for $p = 2$, is isotropic when is applied to second-order BVPs even if it is not isotropic for first-order BVPs (see [28]).

2.3 Conditioning analysis

In this section the conditioning analysis of the discrete problem associated to the acquainted numerical methods is treated making use of the theoretical results for BVPs in [28]. Let us consider the BVP (2.2.1) and rewrite it in an equivalent problem of the first-order. By defining the vector $[y_1, y_2]^T$ with $y_1 = y$ and $y_2 = y'$ the equivalent first-order system is

$$\begin{pmatrix} y_1' \\ y_2' \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -\mu & 2\gamma \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}.$$

The conditioning analysis depends on the eigenvalues of the coefficients matrix, for this reason we proceed with computing the eigenvalues λ_1 and λ_2 by means

$$\begin{vmatrix} -\lambda & 1 \\ -\mu & 2\gamma - \lambda \end{vmatrix} = \lambda^2 - 2\gamma\lambda + \mu = 0. \tag{2.3.1}$$

The well-conditioning of the problem is preserved when it assigns initial values to the components of the solution generated by the eigenvalues with negative real parts (*decaying modes*) and final values to the components of the solution generated by the eigenvalues with positive real parts (*increasing modes*), this means that the *dichotomy* (see [18]) is maintained.

In regard of the dichotomy, since from (2.3.1) $\lambda_1\lambda_2 = \mu$, the problem will be

- *well conditioned* if and only if $\mu < 0$;
- *moderately conditioned* if $\mu = 0$;
- *ill conditioned* if $\mu > 0$ (see p. 32 $\delta > 0$).

It points out that in this analysis the oscillating solutions obtained when $\delta < 0$ are not contemplate. The results suggest us to consider the methods which yield a well conditioned discrete problem when $\mu < 0$.

It is important to clarify that, similarly the discrete problem (2.2.3) is well conditioned when the dichotomy is present with both the increasing and decreasing modes. Differently to the continuous problem, the decaying modes are generated by eigenvalues inside the unit disk of the complex plain and the increasing modes are generated by the eigenvalues outside the same disk.

The known boundary conditions allow us to extract from (2.2.3) a linear system as following

$$M \cdot Y = d, \tag{2.3.2}$$

where $M = (A_2 - 2\gamma h A_1 + \mu h^2 I)$, with A_2 and A_1 are the matrices obtained by deleting the first and the last column of the matrix $\tilde{A}_2$ and $\tilde{A}_1$, and d is just the vector involving the known boundary conditions. As affirmed in Definition 2.1.6 the matrices A_2 and A_1 are quasi-Toeplitz matrices. When n is large enough, the contribution of the initial and final methods to the condition number is minor with respect to that of main method (see [43]). Moreover, it is known that the study of the conditioning of finite Toeplitz matrix brings back to the analysis of the conditioning of infinite matrices. In [4] it is proved as following

Theorem 2.3.1. *Let $\{T_n\}$ be a family of nonsingular Toeplitz band matrices defined as in Definition 1.5.1 and $p(z) = \sum_{i=-m}^{k} a_i z^{m+i}$ the associated characteristic polynomial. Then, the family of matrices T_n is*

i) well conditioned if $p(z)$ is of type $(m, 0, k)$;

ii) weakly well conditioned if $p(z)$ is of type (m_1, m_2, k) or (m, k_1, k_2), where $m_1 + m_2 = m$ and $k_1 + k_2 = k$. In this case the conditioning number $\kappa(T_n)$ of the matrix T_n grows at most as $O(n^\mu)$, where μ is the highest multiplicity among the zeros of unit modulus;

where the polynomial $p(z)$ is the type (k_1, k_2, k_3) if it has k_1 zeros inside the unit circle in the complex plane, k_2 zeros on the circle and k_3 zeros outside it.

Hence, the study of well conditioning of the discrete problem proceeds with the analysis of the roots of the characteristic polynomial associated to the main method (see [28]), defined as following

$$\pi(z, \mu h^2, \gamma h) = \rho(z) - 2\gamma h\sigma(z) + \mu h^2 z^s, \tag{2.3.3}$$

where from the Eq. (2.1.2)-(2.1.3) $s = \max\{k_1, k_3\}$ and

$$\rho(z) = \sum_{j=-k_1}^{k_2} \alpha_{k_1+j} z^{s+j}, \qquad \sigma(z) = \sum_{j=-k_3}^{k_4} \beta_{k_3+j} z^{s+j}.$$

Set $r = \max\{k_2, k_4\}$, it follows that the characteristic polynomial in (2.3.3) has order $s + r$, where s corresponds to the number of lower diagonals of the coefficient matrix in (2.3.2) and r is the number of upper diagonals. The Theorem 2.3.1 states that the well conditioning of the matrix (2.3.2) is satisfied if the roots $\xi_i = \xi_i(\mu h^2, \gamma h)$ of the polynomial $\pi(z, \mu h^2, \gamma h)$ are such that

$$|\xi_1| \leq |\xi_2| \leq \cdots \leq |\xi_{s-1}| < |\xi_s| < 1 < |\xi_{s+1}| \leq \cdots \leq |\xi_{s+r}|.$$

The zeros $|\xi_s|$ and $|\xi_{s+1}|$ are said *principal roots*, since they generate the numerical solution, the remain roots have only a negligible contribution. Naturally we point out that if $h \to 0$, the principal roots approximate the double zeros equal one of $\rho(z)$.

Now, the aim is to find the $(\mu h^2, \gamma h)$-region of well conditioning of the discrete problem and firstly to draw the boundaries separating the well conditioning regions from the ill conditioning ones. As a consequence we solve

$$\pi(z, \gamma h, \mu h^2) = 0 \quad \text{for} \quad |z| = 1. \tag{2.3.4}$$

The polynomial (2.3.3) is linear, then it can be proceed by considering three cases:

- $z = 1$: for the consistency of the schemes the polynomial ρ and σ check $\rho(1) = \sigma(1) = 0$, this means that the line $\mu h^2 = 0$ have to be considered.
- $z = -1$: the equation (2.3.4) to analyze becomes

$$\mu h^2 - 2\gamma h\hat{\sigma} + \hat{\rho} = 0, \tag{2.3.5}$$

  where $\hat{\rho} = \rho(-1)/(-1)^s$ and $\hat{\sigma} = \sigma(-1)/(-1)^s$. Now, by means of simple calculations it is possible to try out that $\hat{\rho} < 0$ for all method and order, while $\hat{\sigma}$ will be positive for D2GBDF, negative for D2GFDF and null for D2ECDF. Therefore, the line (2.3.5) intersects the μh^2-axis at $-\hat{\rho} > 0$, and in particular for D2ECDF the line $\mu h^2 = |\hat{\rho}|$ is just parallel to the vertical axis γh. Since $\hat{\sigma} \neq 0$, for the GBDF and GFDF the intersection of the line (2.3.5) with the vertical axis is at $q = \dfrac{\hat{\rho}}{2\hat{\sigma}}$.
- z is a complex roots and $|z| = 1$: by setting $z = e^{i\theta}$ and considering the Euler formulae it is easy to check that $\rho(\bar{z}) - \bar{z}^{\,s}\rho(z)/z^s = 0$. For D2ECDF the polynomial (2.3.4) reduces to $\mu h^2 = |\hat{\rho}(z)|$, then the region covers the segment between the points $(0, 0)$ and $(0, |\hat{\rho}|)$.

It is instant to observe that D2ECDF always generates well-conditioned matrices for $\mu < 0$, this means that generalizations of the midpoint rule are stable when they are applied to second-order BVPs, but they remain unstable for IVPs (see [28]) and BVPs of first order. The situation changes for the D2GBDF and D2GFDF methods, since the conditioning regions do not cover the all left half-plain, indeed the discrete problem could be ill-conditioned for some values of $(\mu h^2, \gamma h)$ when $\mu < 0$. The ill-conditioning can be avoided by imposing a restriction on the stepsize, which is obtained solving (2.3.5). If $\hat{\delta} = (\gamma\hat{\sigma})^2 - \hat{\rho}\mu$, then the roots of the equation (2.3.5) are

$$h_1 = \frac{\gamma\hat{\sigma} - \sqrt{\hat{\delta}}}{\mu}, \qquad h_2 = \frac{\gamma\hat{\sigma} + \sqrt{\hat{\delta}}}{\mu}.$$

Since $\hat{\rho}$ and μ are both negative, it will be $\hat{\rho}\mu > 0$, therefore if $\hat{\delta} < 0$ and $\gamma\hat{\sigma} > 0$ it does not exist any restriction of the stepsize. Real positive values of h_1 and h_2, with $h_1 < h_2$, are obtained when $\hat{\delta} > 0$ and $\gamma\hat{\sigma} < 0$. The last condition suggests that $\gamma < 0$ needs to obtain a stepsize restriction ($h < h_1$) for D2GBDF , while $\gamma > 0$ is required for D2GFDF. Moreover, if we suppose $\hat{\rho}\mu \ll (\gamma\hat{\sigma})^2$, then the condition is reduced to

$$h \leq \frac{\hat{\rho}}{2\gamma\hat{\sigma}} \equiv \frac{q}{\gamma}$$

This condition is enough to guarantee the well conditioned matrices for the both methods, moreover the value of q increases with the order, as shown in the Table 2.3, this means that the well conditioning regions become wider for the high order.

Table 2.3: Maximum value of $|\gamma|\, h$ required to reach well conditioning matrices for D2GBDF $\gamma < 0$ and for D2GFDF $\gamma > 0$

Order	4	6	8	10
$\lvert q\rvert$	1	$\frac{17}{12}$	$\frac{16}{9}$	$\frac{21}{10}$

The obtained results seem to highlight as the scheme D2GBDF is an high-order extension of the backward finite differences that can be used when the coefficient γ of the first derivative (velocity) is positive, while when γ is negative then it is better to consider the D2GFDF, which is an extension of the forward finite differences. This behavior calls to mind the *upwind scheme* of the first order which applies the backward differences when $\gamma < 0$ and the forward differences when $\gamma > 0$ and suggests the idea to build by a combination of D2GBDF with D2GFDF a well conditioned method which is an *high order extension of the upwind method*, as we will see in the Chapter 3.

2.4 Conclusion

At the end we are able to underline the proprieties provided by the HOGD methods.

- All the derivatives in ODE (2.1.1) can be approximated separately by formulae of the same high order;
- the global numerical scheme provides the same high order of accuracy in each points of the interval;
- the stability proprieties depend on the choice of the main scheme used for the approximation of the first derivative;
- an high order extension of the upwind method can be applied to solve the ODE (2.1.1);
- the size of the discrete problem is always equal to the mesh size;

- implicit equations can be solved as well as the explicit ones;
- ODE of high order, this means of order greater than two, can be solved with same approaches and without increasing the size of the discrete problem.

Chapter 3

HOGUP Method for Two-Point Singular Perturbation Problems

In this chapter we treat the solution of singular perturbation problems SPPs, which have been studied for many years in applied and numerical mathematics, due to their difficulty. A wide set of codes are been developed during the years, among them we found the fortran codes: COLSYS and COLNEW [17, 25] using polynomial collocation on Gauss points; COLMOD developed from COLNEW by adding a continuation strategy and a different error estimation which is more suitable for stiff problems; TWPBVP [33] based on deferred corrections and mono-implicit Runge-Kutta methods; ACDC [30] an improvement of TWPBVP using the more stable Lobatto Runge-Kutta formulae; MIRKDC and its new implementation BVP_SOLVER [61] based on symmetric mono-implicit Runge-Kutta formulae and on the computation of a continuous solution using interpolation in order to control the defect. Moreover, matlab codes are: TOM [51] based on symmetric linear multistep formulae of high order, which are a generalization of the trapezoidal rule; BVP4c [60] and BVP5c exploiting collocation methods to compute an approximate solution which is $\mathcal{C}^1$ piecewise cubic polynomial. In [15] Amodio and Sgura show as the application of high order generalized difference (HOGD) schemes to SPP do not yield an accurate solution, so they suggest to use a high order generalized upwind HOGUP method, on the basis of the conditioning analysis. Our intention it is to apply HOGUP methods on variable meshes [6]. Therefore, we consider a technique of error equidistribution and also a de-

ferred method to build a strategy of variation stepsize, so that an accurate solution is reached by using few mesh points. We have developed a matlab code HOFiD_UP and some numerical tests, chosen in the web page of J. Cash [29], are taken in consideration.

3.1 High Order Generalized Upwind Methods

We consider singular perturbation problems defined as

$$\epsilon y'' = f(x, y, y'), \qquad x \in [a, b], \quad y \in \mathbb{R}, \tag{3.1.1}$$

subject to separated boundary conditions

$$y(a) = \eta_a, \quad y(b) = \eta_b, \tag{3.1.2}$$

where f is a sufficiently smooth function and $\epsilon > 0$ is a perturbation parameter which causes a very fast variation of the solution, called *layers*, in narrow regions, see Section 1.3.

We are interested in investigating the solution of SPPs and it could seem immediately to propose the solution of (3.1.1) by means of the methods included in the high order generalized difference (HOGD) schemes, see Chapter 2. The literature shows as the application of the classical central schemes for solving SPPs can fail, indeed in [18] it is observed that the application of 3-point central schemes with a uniform stepsize $h \gg \epsilon$ causes oscillations in the solution throughout the interval. Then, firstly we are motivated to look the behavior of the solution when formulae D2ECDFs, see Definition 2.2.3, are considered for solving SPPs as in the following example.

Example 3.1.1. Let us consider the well conditioned linear SPP, (Test Problem 4 in [29]),

$$\epsilon y'' + y' - (1 + \epsilon)y = 0, \qquad x \in [-1, 1],$$

with boundary conditions $y(-1) = 1 + \exp(-2)$, $y(1) = 1 + \exp(-2(1 + \epsilon)/\epsilon)$. The exact solution

$$y_e(x) = \exp(x - 1) + \exp\left(-\frac{1 + \epsilon}{\epsilon}(1 + x)\right)$$

has a boundary layer of width $O(\epsilon)$ at $x = -1$. We solve this problem with D2ECDF of order 6 and with a stepsize $h = 5 \cdot 10^{-2}$. For $\epsilon = 10^{-2}$ the gained solution is correct, see Figure 3.1, while it is completely wrong for $\epsilon = 10^{-5}$ (see Figure 3.2).

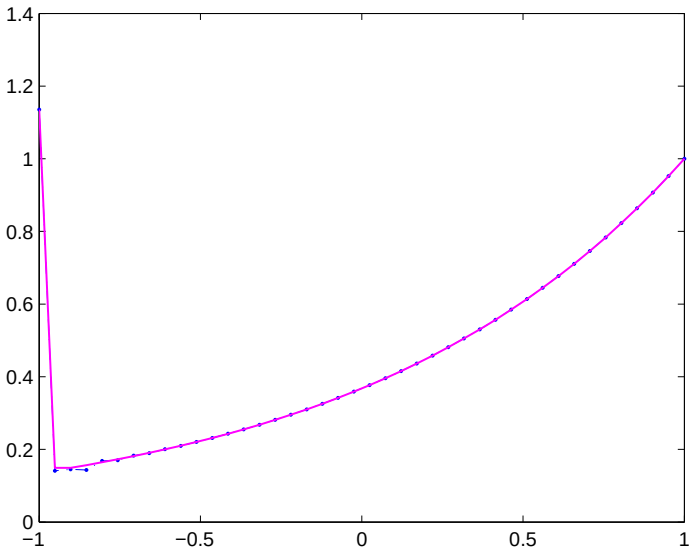


Figure 3.1: Example 3.1.1 - Numerical solution with $\epsilon = 10^{-2}$ computed by D2ECDF of order 6 with 40 equidistant points ($h = 5e-2$).

Example 3.1.1 shows that the well conditioning of the linear system obtained by discretizing the problem is not sufficient to guarantee the effectiveness of the method D2ECDF for solving singular perturbation problems. The reasons of this behavior are to seek in the properties of the characteristic polynomial.

Firstly we consider a uniform discretization of the interval $[a, b]$

$$a \leq x_1 < x_2 < \cdots < x_n \leq b, \tag{3.1.3}$$

where $x_i = x_0 + ih$, $h = (b-a)/n$. Then, we approximate the second and the first derivatives in (3.1.1) by means of formulae defined in (2.1.6), that is

$$y^{(\nu)}(x_i) \simeq y_i^{(\nu)} = \frac{1}{h^\nu} \sum_{j=-s}^{k-s} \alpha_{s+j}^{(s,\nu)} y_{i+j}, \qquad s = 1, \ldots, k-1, \tag{3.1.4}$$

where the coefficients $\alpha_{s+j}^{(s)}$ are computed imposing the maximum order p, see Proposition 2.1.2 and, moreover, $s = 0, \ldots, k$ is the number of initial conditions

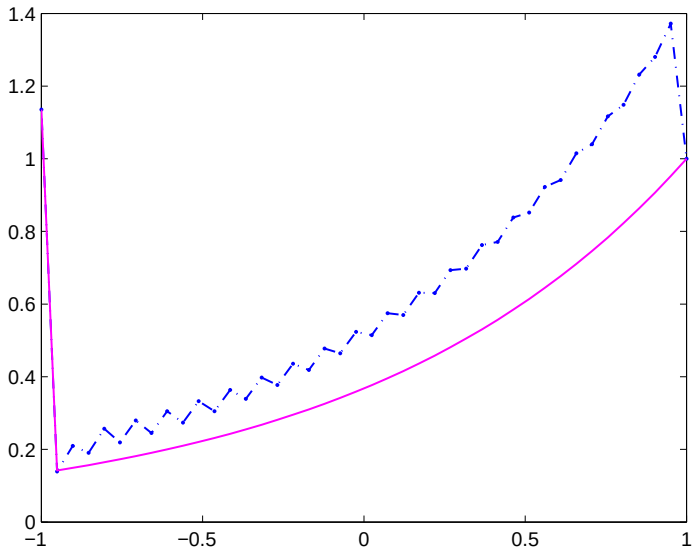


Figure 3.2: Example 3.1.1 - Numerical solution with $\epsilon = 10^{-5}$ computed by D2ECDF of order 6 with 40 equidistant points ($h = 5e-2$).

and k depends on the order p and ν. We suppose to apply a HOGD scheme to the linear test problem

$$\epsilon y'' - \gamma y' - \mu y = 0, \quad \mu \geq 0, \tag{3.1.5}$$

with separated boundary conditions (3.1.2). Then, considering Definition 2.2.3, the equation (3.1.4) can be written in vector form as

$$\left(\frac{\epsilon}{h^2}\tilde{A}_2 - \frac{\gamma}{h}\tilde{A}_1 - \mu\tilde{I}\right)\tilde{Y} = 0, \tag{3.1.6}$$

where $\tilde{I} = [0, I, 0]$ is the identity matrix of size $(n-1) \times (n-1)$ with two additional null vectors, on the left and on the right. From (3.1.6) we obtain the following linear system associated with (3.1.5)

$$M \cdot Y = d, \tag{3.1.7}$$

where $M = (\frac{\epsilon}{h^2}A_2 - \frac{\gamma}{h}A_1 - \mu I)$ is a $(n-1) \times (n-1)$ band matrix, which becomes quasi Toeplitz-matrix when n becomes large enough, and d is the

vector including the known boundary conditions. As shown in Section 2.3, the characteristic polynomial associated can be expressed as

$$\pi(z, \mu h^2, \gamma h) = \epsilon\rho(z) - \gamma h\sigma(z) - \mu h^2 z^s, \tag{3.1.8}$$

where $s = \max\{k_1, k_2\}$ and k_1 and k_2 represent the number of initial conditions of the main methods approximating the second and the first derivatives, and

$$\rho(z) = \sum_{j=-k_1}^{k-k_1} \alpha_{k_1+j}^{(k_1,2)} z^{s+j}, \qquad \sigma(z) = \sum_{j=-k_2}^{k-k_2} \alpha_{k_2+j}^{(k_2,1)} z^{s+j}.$$

Set $r = \min(k_1, k_2)$, then the polynomial $\pi(z, \mu h^2, \gamma h)$ has degree $k + s - r$. If z_i for $i = 1, \ldots, s + r$ are the roots of the characteristic polynomial, it is known from Section 2.3 that a well conditioned Toeplitz banded linear system (with s lower and r upper off-diagonals) have to satisfy the condition

$$|z_1| \leq |z_2| \leq \cdots \leq |z_{s-1}| < |z_s| < 1 < |z_{s+1}| \leq \cdots \leq |z_{s+r}|, \tag{3.1.9}$$

where z_s and z_{s+1} are the principal roots. Then, we are able to give the following result (see [15]).

Proposition 3.1.2. *Let M be the coefficient matrix defined in* (3.1.7)*. If all the roots* (3.1.9) *are real and positive, then M^{-1} has positive entries, that is M is an inverse-monotone matrix.*

Proof. From the hypothesis on the roots (3.1.9) we may approximate M by a Toeplitz band matrix which is factored as

$$M \approx (L_1 \ldots L_s) D (U_{s+1} \ldots U_{s+r}) = LDU, \tag{3.1.10}$$

where D contains a positive diagonal scaling factor, L_i, for $i = 1, \ldots, s$, is unit lower bidiagonal matrix with $-z_i$ on the lower diagonal, and U_j, for $j = s + 1, \ldots, s + r$, is unit upper bidiagonal matrix with $-z_j^{-1}$ on the upper diagonal. Since the matrices are (weakly) diagonally dominant, then all the matrices are (weakly) well conditioned. If the roots are also real and positive, then, L_i^{-1} and U_j^{-1} are positive, consequently also the product M^{-1} is positive. □

The hypotheses of Proposition 3.1.2 are too restrictive for all the HOGD schemes, indeed it is possible to check that the polynomial $\rho(z)$ in (3.1.8) satisfies it only for $p = 2, 4$. Numerically, for other orders, we have found

that the matrix A_2 in (3.1.6) has positive inverse until order $p = 14$, while the matrix A_1 never has positive inverse. Therefore for small value of $\gamma h/\epsilon$ and $\mu h^2/\epsilon$ the matrix M in (3.1.7) is inverse monotone and the approximated solution is fine. Therefore, the property is satisfied only for some value of the stepsize h proportional to ϵ^{-1}.

Even so, we consider less restrictive conditions which allow us to obtain a positive matrix M^{-1} by analyzing the principal roots of $\pi(z, \mu h^2, \gamma h)$. We can observe that the matrix L in (3.1.10) depends on the roots z_i, for $i = 1, \ldots, s$ with $|z_i| < 1$, consequently the decrease of the off-diagonal entries of L_i^{-1} is as fast as $|z_i|$ is small. Hence if z_s is positive and $z_s \gg |z_{s-1}| \geq |z_i|$, for $i = 1, \ldots, s-2$, then the contribution of the parasitic roots vanishes after few off-diagonals, so that L^{-1} is positive. Similarly, the matrix U depends on the roots z_i^{-1}, for $i = s+1, \ldots, s+r$, with $|z_i| > 1$, then if z_s is positive and $\left|z_{s+1}^{-1}\right| \gg \left|z_{s+2}^{-1}\right| \geq \left|z_i^{-1}\right|$, $i = s+3, \ldots, s+r$, then the contribution of parasitic roots is negligible on the off-diagonals and U^{-1} is positive. Therefore if L is positive and $z_s \gg \left|z_{s+1}^{-1}\right|$ or U is positive and $z_{s+1}^{-1} \gg |z_s|$, then we can affirm that M has *inverse essentially positive*.

This property is satisfied by the GBDFs if $\gamma \geq 0$ and by GFDFs when $\gamma \leq 0$. In fact, the polynomial $\sigma(z)$ corresponding to GBDFs formulae has one principal root equal to 1 and the remaining ones are much lower than 1. Indeed, numerically for the same formulae we have also checked that until order 14 L^{-1} is positive, while U^{-1} is not positive, however it has off-diagonal elements going to zero quickly. On the other hand, for GFDFs, since the roots of $\sigma(z)$ are one equal to 1, the principal root, and the others much greater than 1, it is possible to observe as U^{-1} is positive, while L^{-1} has the off-diagonal entries going to zero very fast.

Hence, we can affirm that, even if $h > \epsilon/\gamma$ for $\gamma > 0$ and respectively $\gamma < 0$ the matrix M resulting from GBDFs and GFDFs has a LDU factorization in (3.1.10), with L^{-1} positive and U^{-1} with small off-diagonal elements for GBDFs, vice versa for GFDFs. These negative elements generate only few wriggles close to the boundary layer when $\gamma h \approx \epsilon$ is chosen .

Differently the polynomial $\sigma(z)$ associated to ECDFs has -1 and 1 as principal roots, consequently if $h > \epsilon/\gamma$ then high oscillations occur and the solution can be completely modified.

Example 3.1.3. We solve the problem in Example 3.1.1 with $\epsilon = 10^{-5}$ by D2GFDF of order $p = 6$. Since $h \gg \epsilon$ some oscillations appear near the layer as shown in Figure 3.3. When the stepsize h is halved, the error outside the layer decreases following the theoretical order, as shown in Figure 3.4

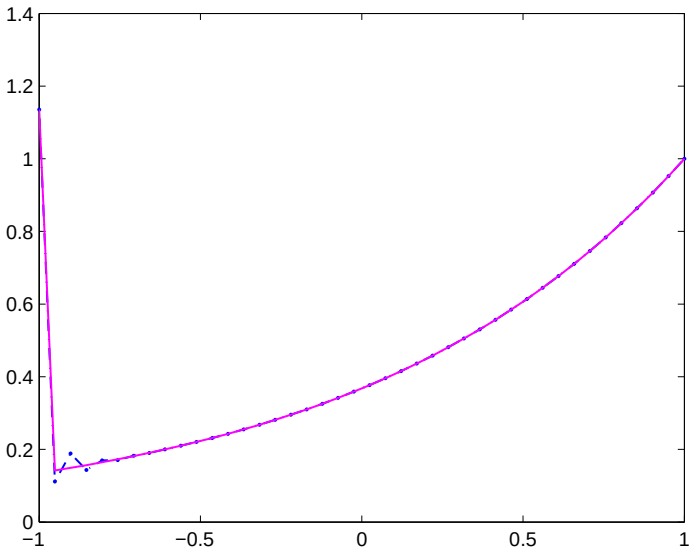


Figure 3.3: Example 3.1.3 - Numerical solution for $\epsilon = 10^{-5}$ computed by D2GFDF of order 6 with 40 equidistant points ($h = 5e-2$).

The Examples 3.1.1-3.1.3 seem to confirm that only the use of ECDFs for the approximation of the first derivative raises high oscillation in the solution. This behavior, already existing for classical central differences, has been overcome in [18] replacing the symmetric scheme for the first derivative with a one-sided scheme, keeping everything else unchanged. This substitution assures the disappearance of the spurious oscillations and motivates the introduction of the so called *upwind method* (see [18]), which approximates the first derivative by a backward or forward difference scheme according to the sign of the coefficients of y'. Certainly the first order of the accuracy represents a disadvantage of the method, even if y'' is approximated by a second order scheme we have a reduction of the global error.

Therefore in order to avoid the spurious oscillations, we decide to approximate the first derivative by GBDFs or GFDFs according to the sign of γ; this choice also preserves the well conditioning property as explain in Section 2.3. In general we can summarize this strategy in the following definition.

Definition 3.1.4 (Generalized Upwind Methods)**.** For the two-point BVP (3.1.1)-(3.1.2) we define *High Order Generalized Upwind* (**HOGUP**) method

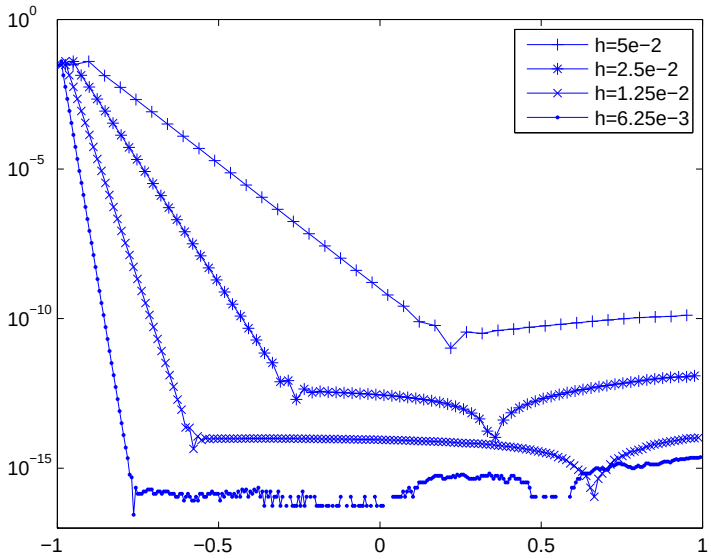


Figure 3.4: Example 3.1.3 - Absolute error for $\epsilon = 10^{-5}$, computed by D2GFDF of order 6 halving the stepsize from $h = 5e-2$ to $h = 6.5e-3$.

the global approximation of y'' by the ECDFs and $y'(x_i)$, for $i = s, n-p+s-1$, by

- GBDF when $\dfrac{\partial f}{\partial y'}(x_i) > 0$;
- GFDF when $\dfrac{\partial f}{\partial y'}(x_i) < 0$;

of the same order p, where s is the number of initial conditions.

Example 3.1.5. Let us consider the Test Problem 10 in [29]

$$\epsilon y'' + xy' = 0, \qquad x \in [-1, 1],$$

with boundary conditions $y(-1) = 0$, $y(1) = 2$. The exact solution

$$y_e(x) = 1 + \mathrm{erf}(x/\sqrt{2\epsilon})/\mathrm{erf}(1/\sqrt{2\epsilon}).$$

has a turning point of width $O(\sqrt{\epsilon})$ at $x = 0$. We consider $\epsilon = 10^{-4}$ and $h = 5e-2$ and solve the problem by D2ECDF and by the HOGUP method of

order 6. Since in the interval $[-1, 1]$ the coefficient of y' changes sign, for the HOGUP we use GBDF when $x < 0$ and GFDF when $x > 0$. Even if $h > \epsilon$, we observe as the oscillations appear only near the turning point, while the solution is completely wrong for D2ECDF, see Figure 3.5. Starting with 40 points and doubling the mesh we observe that the HOGUP check the order of convergence, see Figure 3.6.

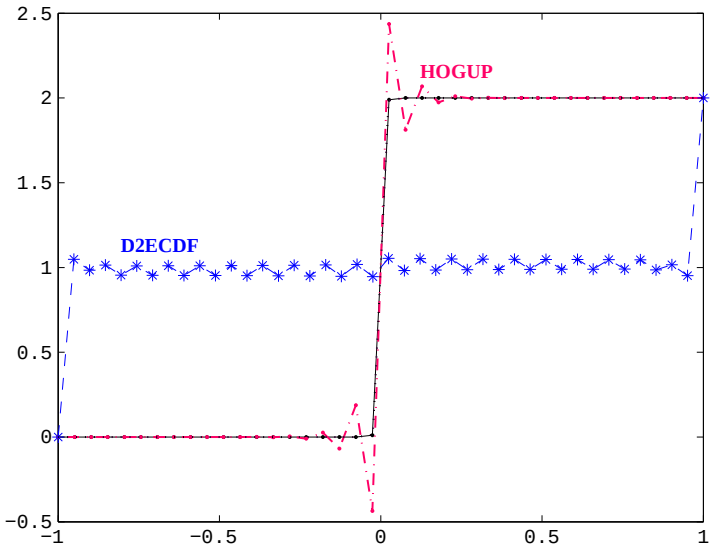


Figure 3.5: Example 3.1.5 - Numerical solution for $\epsilon = 10^{-4}$ computed by D2GFDF and HOGUP of order 6 with 40 equidistant points ($h = 5e - 2$).

Remark 3.1.6. We point out that inside the layer the error of the HOGUP method decreases with its order only when the stepsize is proportional to the size of the layer, for the Example 3.1.5 when $h = O(\sqrt{\epsilon})$, see Figure 3.6. Hence, the high order upwind method does not converge uniformly with respect to ϵ, in fact the error bound depends on ϵ. Consequently, a uniformly convergence may be reached only applying the HOGUP method on a variable mesh.

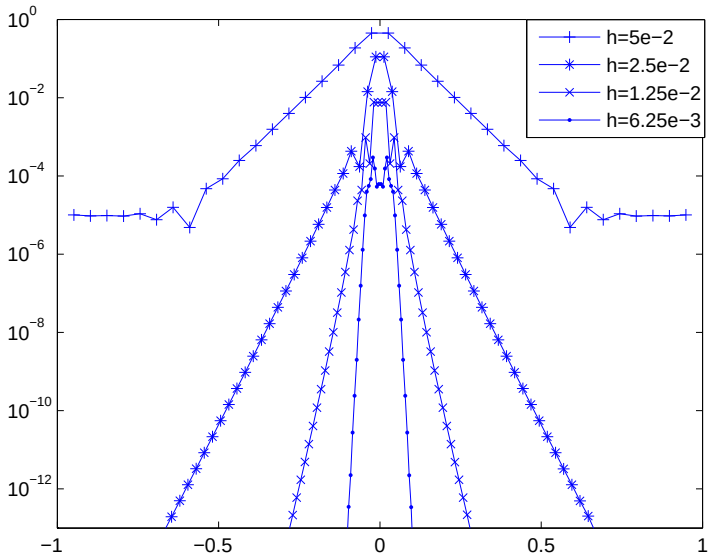


Figure 3.6: Example 3.1.5 - Absolute error for $\epsilon = 10^{-4}$, computed by HOGUP of order 6 doubling the mesh points.

3.2 HOGUP on Variable Mesh

The aim of this section is to apply the HOGUP method on grid points with variable stepsize [6]. On the other hand, we want to preserve the nice properties the method exhibits on a mesh of equidistant points. Therefore, we discretize the integration interval $[a, b]$ by means of piecewise constant grids such that the stencil used for each formula changes stepsize at most once. Moreover, we underline that in the first and last points the initial and final methods are used with constant stepsize, while the main method may be also applied on a variable mesh, thus it is necessary to compute the variable-step coefficients of the main methods as follows

$$y''(x_{i+k/2-s}) \simeq \frac{1}{h_i^2} \sum_{j=-s}^{k-s} \tilde{\alpha}_{j+s}^{(s,2)} y_{i+j}, \qquad s = 1, \ldots, k-1, \tag{3.2.1}$$

$$y'(x_{i+k/2-s+t}) \simeq \frac{1}{h_i} \sum_{j=-s}^{k-s} \tilde{\alpha}_{j+s}^{(t,s,1)} y_{i+j}, \qquad t = -1, 1, \quad s = 1, \ldots, k-1. \tag{3.2.2}$$

The coefficients $(\tilde{\alpha}_0^{(s,2)}, \ldots, \tilde{\alpha}_k^{(s,2)})$ and $(\tilde{\alpha}_0^{(t,s,1)}, \ldots, \tilde{\alpha}_k^{(t,s,1)})$ are still computed as in Proposition 2.1.2 by solving Vandermonde linear systems with the matrix

$$V = \begin{pmatrix} 1 & 1 & \ldots & 1 & 1 & \ldots & 1 \\ -s & -s+1 & \ldots & 0 & 1 & \ldots & (k-s)v \\ (-s)^2 & (-s+1)^2 & \ldots & 0 & 1 & \ldots & [(k-s)v]^2 \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ (-s)^{k+1} & (-s+1)^{k+1} & \ldots & 0 & 1 & \ldots & [(k-s)v]^{k+1} \end{pmatrix}, \quad (3.2.3)$$

where s represents the number of steps equal to h_i (the others are equal to h_{i+1}) and $v = h_i/h_{i+1}$. The coefficients are computed such that the formulae have order $p = k - \nu$, with $\nu = 1, 2$. Consequently a loss of the order occurs for the symmetric scheme involved into the approximation of y'', so that we observe a order reduction in the global approximation. For all these choices the coefficients have been computed algebraically. Moreover, we observe that the coefficients magnitude depends on the values of v, for this reason we decided to change stepsize at least every $k + 4$ points, that is we use 3 constant steps methods before changing the stepsize, if necessary, and bound v according to the values in Table 3.1.

Table 3.1: Maximum ratio between two successive steps

order	4	6	8	10
v	15	10	7	5

3.3 Deferred Correction

The *deferred correction* method [18] is widely used in several numerical schemes and it may be viewed as a special case of *defect correction* method, where defect means that one is substituting the exact solution into the difference scheme. Starting point is the idea to approximate the local truncation error resulting from applying a basic scheme, after that it is possible to solve for a corrected approximation using the same discretization.

We are interested to apply the deferred correction to high order finite difference schemes defined in Chapter 2 for the BVPs. It is known that we

approximate $y''(x_i)$ and $y'(x_i)$, for $i = 1\ldots,, n-1$, by means of high order finite difference schemes of the same even order $p = 2s$. As discussed in [7], we point out that the same results can be obtained by approximating $y'(x_i)$ and $y''(x_i)$ with two different (i.e., based on different stencils) interpolation formulae. In particular, for the second derivative and $i = s, \ldots, n-s$, we use the Lagrange polynomial of degree $2s$ (the same of the formula) interpolating $y(x)$ at the points $x_{i-s}, \ldots, x_{i+s}$,

$$q_{2s}(x) = \sum_{j=i-s}^{i+s} y_j L_j(x), \qquad \text{where} \qquad L_j(x) = \prod_{\substack{k=i-s \\ k\neq j}}^{i+s} \frac{x - x_k}{x_j - x_k}. \tag{3.3.1}$$

Then

$$y''(x_i) \approx q''_{2s}(x_i) = \sum_{j=i-s}^{i+s} \alpha_{j+s}^{(s,i)}\, y_j,$$

where $\alpha_{j+s}^{(s,i)} = L''_j(x_i)$, $j = i-s, \ldots, i+s$. Moreover, we underline that these coefficients depend on the stepsize h_i, $i = i-s, \ldots, i+s$ only in case of variable mesh. For $s > 1$ this procedure requires some adjustment in order to approximate $y''(x_i)$, for $i = 1, \ldots, s-1$ and $i = n-s+1, \ldots, n-1$. In fact, we remind that in the extreme points initial and final formulae are used, this means that for the first $s-1$ points we always have to use the stencil $x_0, \ldots, x_{2s+1}$ while for the last points the stencil must be $x_{n-2s}, \ldots, x_n$.

Alike, for the first derivative we consider a polynomial of degree $2s$ interpolating $y(x)$ in the points $x_{i-s+t}, \ldots, x_{i+s+t}$, where $t = -1, 0, 1$ depends on the chosen formula among GBDF, ECDF or GFDF, respectively. In the same way, the coefficients of these formulae derive from the first derivative of the Lagrange polynomial computed at x_i, while the approximation of $y'(x_i)$ in the first and last points of the mesh requires *ad-hoc* formulae.

In conclusion, the solution of (3.1.1)-(3.1.2) on the mesh (3.1.3) is obtained by

$$\Phi_{2s}(\mathbf{y}) = \begin{pmatrix} \epsilon A_{2s} - f(\mathbf{x}, \mathbf{y}, B_{2s}\mathbf{y}) \\ g(y_0, y_n) \end{pmatrix} = 0, \tag{3.3.2}$$

where $(\mathbf{x}, \mathbf{y})$ denotes the discrete solution, and A_{2s} and B_{2s} contain the coefficients of the formulae of order $2s$ for the second and the first derivative. In the case of a IVPs we have $g(y_0, y'_0)$. To emphasize the relation between two methods of consecutive orders it is convenient to use the Newton-like polynomial rather than the Lagrange polynomial, defined as

$$p_{2s}(x) = y\left[x_i\right] + \\ + \sum_{j=1}^{s} \left(y\left[x_{i-j}, \ldots, x_{i+j-1}\right] \prod_{k=i-j+1}^{i+j-1} \frac{x - x_k}{x_{i-j} - x_k} + y\left[x_{i-j}, \ldots, x_{i+j}\right] \prod_{k=i-j}^{i+j-1} \frac{x - x_k}{x_{i+j} - x_k} \right),$$

where, in case of constant stepsize, $y[x_j, \ldots, x_{j+k}] = \Delta^k y_j$ is the forward difference of order k. Then

$$\begin{aligned} y''(x_i) \approx p''_{2s}(x_i) &= \frac{2y[x_{i-1}, x_i, x_{i+1}]}{(x_{i+1} - x_i)(x_{i+1} - x_{i-1})} + \\ &+ 2\sum_{j=2}^{s} \left(\gamma_{j,1} y[x_{i-j}, \ldots, x_{i+j-1}] + \gamma_{j,2} y[x_{i-j}, \ldots, x_{i+j}] \right) \\ &= p''_{2s-2}(x_i) + 2\left(\gamma_{s,1} y[x_{i-s}, \ldots, x_{i+s-1}] + \gamma_{s,2} y[x_{i-s}, \ldots, x_{i+s}]\right) \end{aligned} \tag{3.3.3}$$

where

$$\gamma_{j,1} = \frac{1}{x_{i-j} - x_i} \sum_{\substack{k=i-j+1 \\ k \neq i}}^{i+j-1} \frac{1}{x_{i-j} - x_k} \prod_{\substack{r=i-j+1 \\ r \neq k,i}}^{i+j-1} \frac{x_i - x_r}{x_{i-j} - x_r}$$

and

$$\gamma_{j,2} = \frac{1}{x_{i+j} - x_i} \sum_{\substack{k=i-j \\ k \neq i}}^{i+j-1} \frac{1}{x_{i+j} - x_k} \prod_{\substack{r=i-j \\ r \neq k,i}}^{i+j-1} \frac{x_i - x_r}{x_{i+j} - x_r}.$$

The main advantage of this representation is the possibility to compute an approximation of the local truncation error with few operations. If constant stepsize is used, then the coefficients $\gamma_{j,1} = 0$ due to symmetry while $\gamma_{j,2} = \{-1/12, 1/90, -1/560, 1/3150\}$ for the even orders form 4 to 10.

With a similar reasoning, we can obtain an approximation of $y'(x_n)$ in the form

$$y'(x_i) \approx \sum_{j=1}^{s} \left(\delta_{j,1} y[x_{i-j+t}, \ldots, x_{i+j-1+t}] + \delta_{j,2} y[x_{i-j+t}, \ldots, x_{i+j+t}] \right), \tag{3.3.4}$$

where

$$\delta_{j,1} = \frac{1}{x_{i-j+t} - x_i} \prod_{\substack{k=i-j+1 \\ k \neq i-t}}^{i+j-1} \frac{x_i - x_{k+t}}{x_{i-j+t} - x_{k+t}}$$

and

$$\delta_{j,2} = \frac{1}{x_{i+j+t} - x_i} \prod_{\substack{k=i-j \\ k \neq i-t}}^{i+j-1} \frac{x_i - x_{k+t}}{x_{i+j+t} - x_{k+t}}.$$

Supposed f sufficiently smooth, from (3.3.3)-(3.3.4) it follows for the basic scheme

$$\Phi_{2s}(\mathbf{y}) = \Phi_{2s-2}(\mathbf{y}) + \tau_{2s}. \tag{3.3.5}$$

If $\hat{\mathbf{y}}$ is the exact solution of (3.1.1) or (3.1.2), then the following statements are satisfied

$$\Phi_{2s}(\hat{\mathbf{y}}) = \hat{\tau}_{2s} = o(h^{2s}), \qquad \Phi_{2s-2}(\hat{\mathbf{y}}) = \hat{\tau}_{2s-2} = o(h^{2s-2}), \tag{3.3.6}$$

then from (3.3.5)-(3.3.6) it follows that for $h \to 0$, τ_{2s} is as an approximation of the local truncation error $\hat{\tau}_{2s-2}$, so that

$$\tau_{2s} = \hat{\tau}_{2s-2} + o(h^{2s}).$$

If $\bar{\mathbf{y}}$ is the numerical solution achieved using difference schemes of order $2s-2$, then

$$\Phi_{2s-2}(\bar{\mathbf{y}}) = 0,$$

consequently by substituting $\bar{\mathbf{y}}$ in (3.3.5) we have

$$\Phi_{2s}(\bar{\mathbf{y}}) = \tau_{2s}. \tag{3.3.7}$$

It is important to point out that for (3.3.7), an approximation of the truncation error for the method of order $2s$ is gained evaluating the formulae of order $2s$ at the points of the solution reached with the lower order scheme $2s-2$. Similarly, if $\bar{\bar{\mathbf{y}}}$ is the numerical solution obtained applying the formulae of order $2s$, this means $\Phi_{2s}(\bar{\bar{\mathbf{y}}}) = 0$ and, substituting $\bar{\bar{\mathbf{y}}}$ in (3.3.5) a corrected solution $\bar{\bar{\mathbf{y}}}$ is computed by solving

$$\Phi_{2s-2}(\bar{\bar{\mathbf{y}}}) = -\tau_{2s} = -\Phi_{2s}(\bar{\mathbf{y}}). \tag{3.3.8}$$

We emphasize the significance of the result reached by the defect correction and underline that it is possible to evaluate an estimation of the truncation error concerning the difference scheme of a given order p and to compute a corrected solution yielding the same scheme, without increasing the size of the system required if a higher order is applied. In this way we are able to compute the estimation of the error without solving the problem with two consecutive orders and saving the computational cost.

3.4 Error Equidistribution

We consider the SPP (3.1.1)-(3.1.2) and an error tolerance TOL, the aim is to find a mesh

$$\pi : a = x_1 < x_2 < \cdots < x_n = b \tag{3.4.1}$$

with $h = \max_{0\le i\le n} h_i$ and $h_i = x_i - x_{i-1}$, for $i = 1, \ldots, n$, such that n is small and the error

$$e_i = |y_i - y(x_i)|, \qquad i = 0, \ldots, n \tag{3.4.2}$$

is less than TOL, where $y_\pi = \{y_i\}_{i=1}^n$ approximate solution of $y(x)$. We underline that the error may be relative or absolute, or a combination of both.

The approximation of (3.1.1)-(3.1.2) is computed by high order generalized upwind (HOGUP) schemes of order p, thus

$$|y_i - y(x_i)| = C h_i^p \, |\Psi(x_i)| + O(h^{p+1}), \tag{3.4.3}$$

where $\Psi(x)$ involves the derivatives of order higher than p. We follow the idea at the basis of the equidistribution in [18], whose principle is to minimize $\max_{0\le i\le n} |e_i|$. Considering $T_i \approx T(x_i) = C\,|\Psi(x_i)|$, we obtain an error measure $\phi_i \approx h_i T_i^{1/p}$, which varies linearly with h_i, therefore a simpler *minmax problem*

$$\min_h \max_i |\phi_i|, \qquad \sum_{i=1}^{n} h_i = b - a$$

is involved. The solution of the optimization problem is obtained by assuming all ϕ_i equal to a constant λ, so that

$$h_i = \frac{\lambda}{T_i^{1/p}}, \quad i = 0, \ldots, n, \quad \lambda = \frac{b-a}{\sum_{i=1}^{n} (T_i^{1/p})^{-1}}$$

Generalizing we consider a smooth function, therefore, as in [18], we take $T(x)^{1/p}$ as *monitor function*, then a mesh π is equidistributed with respect to the monitor function if for a constant λ

$$\int_{x_i}^{x_{i+1}} T(x)^{1/p} dx \equiv \lambda, \tag{3.4.4}$$

where

$$\lambda = \frac{\theta}{n} \tag{3.4.5}$$

and

$$\theta = \int_a^b T(x)^{1/p} dx. \tag{3.4.6}$$

We note that

$$t(x) := \frac{1}{\theta} \int_a^x T(\xi)^{1/p} d\xi \tag{3.4.7}$$

is piecewise linear, thus from (3.4.5)-(3.4.6) known x_i we may finding x_{i+1} such that

$$t(x_{i+1}) = \frac{i}{n}. \tag{3.4.8}$$

If π^* is the new mesh given by (3.4.7)-(3.4.8) then if y_π^* is the new solution, considering (3.4.4)-(3.4.5), follows that

$$h_i T(x_i)^{1/p} = \frac{\theta}{n}(1 + O(h)),$$

then (3.4.3) satisfies

$$|y_i - y(x_i)| = \left(\frac{\theta}{n}\right)^p (1 + O(h)) + O(h^p). \tag{3.4.9}$$

The new mesh size n such that the solution is equidistributed has the uniform error less than TOL, then from (3.4.9) we can predict it by choosing

$$n = \frac{\theta}{TOL^{1/p}}. \tag{3.4.10}$$

We consider also the quantities

$$r_1 = \max_{0 \le i \le n} h_i \left(\frac{T(x_i)}{TOL}\right)^{1/p} \qquad r_2 = \sum_{i=0}^{n} h_i \left(\frac{T(x_i)}{TOL}\right)^{1/p}, \qquad r_3 = \frac{r_2}{n},$$

where the ratio $\frac{r_1}{r_3}$ gives some information about the equidistribution, in fact if the ratio is large the maximum error estimate is larger than the average one, this means that the mesh is not well equidistributed. Moreover we required to check that

$$\frac{r_1}{r_3} < 1.2 \tag{3.4.11}$$

(i) If (3.4.11) is satisfied, then the mesh is sufficiently equidistributed and the new mesh is obtained doubling the points, that is we have $\pi^* = \{x_1, x_{3/2}, x_2, \ldots, x_{n-1}, x_{(n+1/2)}, x_n\}$.

(ii) If (3.4.11) is not checked, then r_2 predicts the number of mesh points satisfying the tolerance TOL and (3.4.10). Then we consider to predict the number of mesh points $n^* = \max\{\min(r_2, 1.2\,n), n/1.2\}$ in place of r_2, in order to avoid incorrect conclusion early. Then the new mesh π^* is computed by (3.4.7)-(3.4.8). We also underline that the mesh is doubled if n^* has already been used for two consecutive times.

Stepsize and order variation strategy

The order variation strategy developed in the code for solving two-point linear singular perturbation problems (3.1.1)-(3.1.2) with very small perturbation parameters ϵ consists to combine one or more methods with piecewise constant stepsize of different order, chosen among 4, 6, 8 and 10. The choice of the order is strictly connected to the desired precision. Low orders allow us to determine the first variable meshes with a suggestion on the location of the layer and relatively few points. The computed mesh can then be used by higher orders to quickly obtain better accuracy.

The following algorithm describe the variable order strategy:

Algorithm 1. function $[x, y]$ = HOFiD_UP ('problem', tol)

$ord = 4$;
$\widetilde{x} = a : (b-a)/10 : b$;
$\widetilde{y} = y_a : (y_b - y_a)/10 : y_b$;
$ltol = \max(1\text{e-}2, tol)$;
while $\|err\| < tol$
  $[x, y, err]$ = `genup` ('problem', $ord, ltol, \widetilde{x}, \widetilde{y}$);
  if $\|err\| > tol$
    $ltol = \max(\|err\|/100, tol)$;
    $ord = ord + 2$;
    $[\widetilde{x}, \widetilde{y}]$ = `adjmesh` $(x, y, ord + 4)$;
  end
end

Hence, we consider as initial number of mesh points $n = 10$ and start with method of order 4. The starting mesh is updated until we obtain a solution with a computed absolute/relative error less than 10^{-2}. This essentially means that the stepsize used is smaller than the width of the layer. Then, the process is iterated by increasing the order of the method and decreasing the exit tolerance.

Since the piecewise constant mesh depends on the order used, it is necessary to modify the output mesh when we pass from one method of lower order to one of higher order. This is made in `adjmesh` by increasing the number of constant points in each sub-interval with less than $ord + 4$ constant steps, or bringing together two sub-intervals with almost the same stepsize. Moreover the number of points in each sub-interval is increased in order to satisfy the restriction in Table 3.1.

The step variation technique, based on the equidistribution described in Section 3.4, is applied inside the `genup` function and allows us to compute, for fixed order ord and input tolerance TOL, a numerical solution with maximum error less than TOL. The error is approximated using the deferred correction in Section 3.3, that is considering the same scheme we compute the corrected solution which allow us to estimate the error.

We pay particular attention to the function `monitor` which is inside the function `genup` and allows us to compute the new grid x starting from the old grid $\widetilde{x}$ and the computed error err. It is described as follows

Algorithm 2. function $x =$ `monitor` $(err, \widetilde{x}, ord, TOL)$
$n = \text{length}(\widetilde{x}) - 1;$
$h = \widetilde{x}(2:n) - \widetilde{x}(1:n-1);$
$t = \max(err(2:n+1), err(1:n))\hat{}(1/ord);$
$r_1 = \|t\|_\infty;$
$n^* = \lfloor \|t\|_1 / TOL\hat{}(1/ord) \rfloor;$
$r_3 = \|t\|_1 / n;$
if $r_1/r_3 \leq 1.2$ & $n^* \geq 2 \cdot n$
    x is obtained halving the step-length vector h
    $n^* = 2 \cdot n$
else
    $n^* = \max(\min(n^*, \lfloor 1.2 \cdot n \rfloor), \lfloor n/1.2 \rfloor);$
    $I = [0 \ \text{cumsum}(t)]/\|t\|_1;$
    $z = 0 : 1/n^* : 1;$
    $\widehat{x} =$ `linear_interp`$(I, \widetilde{x}, z);$
    $x =$ `piecewise_grid`$(\widehat{x}, ord + 4);$
end

The function `monitor` is based on an equidistribution of the error and gives back the equidistributed mesh, which is modified next by `piecewise_grid` in order to have piecewise constant steps . The function `piecewise_grid` starts from the minimum step and determines $ord + 4$ consecutive constant steps which are able to overlay exactly some steps of the previous grid. This

procedure is iterated on the remaining part of the grid in order that the new steps are greater than or equal to those already computed and the ratio of two consecutive steps is bounded by the values in Table 3.1.

3.5 Numerical Test

In this section we give some results on the convergence behavior of the matlab code HOFiD_UP tested on four linear and two nonlinear singular perturbation problems contained in the "BVP software page" of J. Cash [29].

For our numerical experiments we have chosen to show as order 4, 6 and 8 works individually requiring three different exit tolerance, that is 10^{-4},10^{-6} and 10^{-8}. We start with a uniform initial mesh of 10 points with order 4 and 6, while the number of mesh points is 20 when order 8 is employed. Naturally, a variable stepsize is obtained using the equidistribution of the error described in in Section 3.4 and Section 3.4. The results highlight as for tolerance 10^{-4} and 10^{-6} a fine mesh is obtained with order 6, while order 8 is exploited to improve the number of mesh points when a tolerances 10^{-8} is required. Hence, order 4 seems working better with a great tolerance, as 10^{-2}. This analysis for each order helps us to set tolerances when we apply order variation strategy. Indeed, for a exit tolerance $TOL = 10^{-4}$ we choose two different order between 4, 6 and 8 and an initial tolerance 10^{-2}, differently for $TOL = 10^{-6}$ or $TOL = 10^{-8}$ we can decide to employ two or three orders, in these case for orders 4-8 the initial tolerance is 10^{-2}, for orders 6-8 the initial tolerance is 10^{-4} and for orders 4-6-8 the initial tolerances are respectively 10^{-2} for order 4 and 10^{-5} for order 6. We point out as an order variation strategy improves the number of mesh points above all for both small values of perturbation ϵ and exit tolerance TOL. For nonlinear problems a continuation strategy sometimes needs, in such cases the solution is computed in sequence for each $\epsilon_i = 10^{-i}$, $i = 1, 2, \ldots$, less than ϵ required using the method of order 4 and $TOL = 10^{-2}$. We point out that for the Example 3.5.5 no continuation is exploited, while for the last Example 3.5.6 we show both results with and without continuation.

In the following tables we carry for each order from left to right the number of steps employed to reach the fine mesh, the total number of mesh points, the number of the final mesh and the error obtained by the relation

$$\|e\|_\infty = \max_{0 \le i \le n} \frac{|y(x_i) - y_i|}{1 + |y(x_i)|},$$

when the exact solution is known, otherwise the unknown value $y(x_i)$ is sub-

stituted by $y_i^{(p+2)}$ obtained using the same method and order $p+2$. Moreover, in bold we have highlighted the finer mesh for each value of ϵ.

Example 3.5.1. Let us consider the linear test problem in Example 3.1.1, where the coefficient multiplying y' is positive in the x-domain. The Table 3.2 shows as for $TOL = 10^{-4}$ order 6 gives a finer mesh, while the Table 3.3 highlights as order 4 and 6 are a good combination for an order variation. However, this strategy is convenient to employ when the perturbation ϵ becomes small. For $TOL = 10^{-6}$ in the Table 3.4 order 6 is established to be a good choice for the small ϵ, moreover order 6 and 8 in the Table 3.5 work better for the order variation strategy. In the Table 3.6 is evident that order 8 have to be used for tolerance $TOL = 10^{-8}$, while the Table 3.7 shows as an order variation 4-6-8 allows us to considerably reduce the number of mesh points when ϵ becomes smaller.

Table 3.2: Example 3.5.1 - numerical solution with $TOL = 10^{-4}$ and fixed order.

ϵ	order 4				order 6				order 8			
10^{-1}	4	87	36	8.04e-05	3	60	30	6.08e-05	2	48	**28**	1.04e-05
10^{-2}	5	150	58	6.63e-05	5	152	**50**	2.54e-05	5	190	61	1.62e-06
10^{-3}	6	236	84	8.78e-05	7	284	**70**	7.67e-06	8	347	76	3.18e-05
10^{-4}	8	439	116	6.75e-06	8	376	**83**	1.09e-04	12	723	122	3.70e-08
10^{-5}	9	582	140	7.49e-06	10	585	**111**	8.75e-06	14	952	141	7.56e-07
10^{-6}	10	723	143	1.89e-05	11	712	**127**	5.62e-05	16	1310	186	9.01e-08
10^{-7}	11	917	172	7.06e-06	13	1039	**160**	3.71e-07	17	1515	196	1.53e-06
10^{-8}	12	1083	203	5.59e-06	14	1219	**182**	8.61e-07	18	1753	228	8.64e-05
10^{-9}	13	1315	229	1.07e-05	15	1425	**207**	4.10e-05	20	2282	273	4.12e-08
10^{-10}	14	1498	**234**	1.07e-05	16	1683	241	6.94e-05	21	2499	285	4.47e-06

Example 3.5.2. Let us consider the linear test problem 6 in [29]

$$\epsilon y'' + xy' = -\epsilon\pi^2 \cos(\pi x) - \pi x \sin(\pi x), \qquad x \in [-1, 1],$$

with boundary conditions $y(-1) = -2$, $y(1) = 0$. The exact solution

$$y_e(x) = \cos(\pi x) + \frac{\operatorname{erf}(x/\sqrt{2\epsilon})}{\operatorname{erf}(1/\sqrt{2\epsilon})}$$

has a shock layer in the turning point region near $x = 0$.

Here the coefficient of y' changes its sign in the x-domain and there is no y-term. The Table 3.8 underlines as order 6 reaches the exit tolerance $TOL = 10^{-4}$ with the finer mesh, while from the Table 3.9 we note that the

Table 3.3: Example 3.5.1 - numerical solution with $TOL = 10^{-4}$ and variable order.

ϵ	orders 4-6				orders 4-8				orders 6-8			
10^{-1}	5	96	31	1.22e-05	4	67	24	8.31e-05	4	71	**24**	8.31e-05
10^{-2}	6	197	56	5.46e-06	5	149	57	6.47e-05	6	190	**49**	3.14e-05
10^{-3}	7	248	69	8.41e-05	7	247	**68**	5.98e-06	8	317	71	1.12e-04
10^{-4}	9	377	**81**	1.55e-05	9	406	99	5.12e-07	10	504	95	2.49e-05
10^{-5}	10	497	**101**	7.18e-05	10	513	117	3.76e-06	12	765	134	4.82e-07
10^{-6}	12	723	131	1.57e-06	11	602	**122**	2.43e-05	13	916	148	2.24e-06
10^{-7}	13	955	173	1.26e-06	12	809	173	7.28e-05	14	1100	**172**	7.66e-06
10^{-8}	14	1146	**193**	3.11e-07	14	1218	234	4.31e-08	16	1488	208	3.91e-08
10^{-9}	15	1325	**208**	5.41e-07	15	1395	246	2.93e-08	17	1700	219	6.41e-07
10^{-10}	16	1463	**210**	2.86e-05	16	1489	236	1.20e-06	18	2168	307	4.01e-07

Table 3.4: Example 3.5.1 - numerical solution with $TOL = 10^{-6}$ and fixed order.

ϵ	order 4				order 6				order 8			
10^{-1}	11	592	100	9.25e-07	5	152	52	2.79e-07	3	84	**36**	2.44e-07
10^{-2}	9	517	116	1.33e-06	6	248	82	1.41e-07	5	179	**54**	1.88e-06
10^{-3}	11	956	184	6.41e-07	7	377	105	1.61e-07	8	415	**99**	2.36e-08
10^{-4}	11	1035	197	1.18e-06	8	491	**113**	1.34e-06	10	640	117	1.79e-08
10^{-5}	11	1180	246	4.77e-07	9	664	**135**	5.22e-07	12	1002	172	1.77e-09
10^{-6}	11	1232	259	1.04e-06	11	941	**170**	8.34e-08	13	1177	185	1.24e-08
10^{-7}	13	1910	342	4.29e-07	12	1149	**183**	1.38e-07	15	1610	233	7.00e-10
10^{-8}	14	2351	405	3.90e-07	13	1432	**221**	7.26e-08	16	1862	261	3.59e-09
10^{-9}	13	1974	368	1.15e-06	14	1642	**232**	1.90e-07	17	2086	269	7.13e-08
10^{-10}	14	2513	442	6.55e-07	15	1939	**267**	8.93e-08	18	2371	285	3.75e-07

Table 3.5: Example 3.5.1 - numerical solution with $TOL = 10^{-6}$ and variable order.

ϵ	order 4-8				order 6-8				order 4-6-8			
10^{-1}	5	104	**37**	8.76e-07	5	129	39	1.80e-07	7	181	49	1.98e-08
10^{-2}	6	214	65	1.51e-07	7	275	63	3.28e-08	7	258	**61**	5.56e-07
10^{-3}	8	344	97	3.09e-08	8	369	**85**	8.29e-07	9	421	92	2.78e-08
10^{-4}	9	406	99	5.12e-07	10	563	**98**	7.81e-08	11	605	122	1.79e-09
10^{-5}	11	649	136	9.44e-09	11	707	**122**	9.52e-07	12	742	133	1.23e-08
10^{-6}	12	748	**146**	3.20e-08	13	1030	171	1.19e-08	13	880	157	7.51e-08
10^{-7}	13	1018	209	1.53e-08	14	1226	**187**	6.81e-09	14	1160	205	5.17e-08
10^{-8}	14	1218	234	4.31e-08	15	1425	**206**	2.95e-08	15	1362	216	5.33e-09
10^{-9}	15	1395	246	2.93e-08	17	1915	250	2.94e-08	16	1560	**235**	3.02e-08
10^{-10}	16	1489	**236**	1.20e-06	18	2249	296	3.34e-07	18	1969	270	3.13e-07

Table 3.6: Example 3.5.1 - numerical solution with $TOL = 10^{-8}$ and fixed order.

ϵ	order 4				order 6				order 8			
10^{-1}	13	1192	400	5.22e-09	8	385	91	7.43e-09	4	134	**50**	2.18e-08
10^{-2}	14	2043	630	3.08e-09	8	465	116	1.06e-08	6	298	**91**	9.22e-10
10^{-3}	14	2412	832	1.60e-09	8	559	144	1.37e-08	9	525	**118**	8.89e-10
10^{-4}	17	3064	664	4.47e-09	9	754	186	2.66e-09	10	737	**149**	2.65e-09
10^{-5}	16	3150	704	7.87e-09	10	1039	242	1.97e-09	11	1003	**196**	4.22e-09
10^{-6}	20	5721	1126	1.56e-09	11	1357	308	9.59e-10	13	1442	**228**	2.64e-10
10^{-7}	20	5903	1162	2.36e-09	12	1719	330	1.00e-09	14	1790	**248**	5.10e-10
10^{-8}	21	6423	729	1.20e-08	13	2068	365	4.27e-09	15	2119	**293**	3.14e-09
10^{-9}	22	7405	822	4.95e-08	13	2127	345	3.43e-08	16	2421	**294**	2.88e-08
10^{-10}	21	7626	1476	5.14e-07	14	2505	330	2.51e-07	17	2758	**320**	3.91e-07

Table 3.7: Example 3.5.1 - numerical solution with $TOL = 10^{-8}$ and variable order.

ϵ	order 4-8				order 6-8				order 4-6-8			
10^{-1}	6	153	**49**	1.27e-08	6	180	50	6.39e-09	7	181	**49**	1.98e-08
10^{-2}	7	315	92	7.78e-10	8	379	92	8.66e-10	8	345	**87**	3.27e-09
10^{-3}	9	474	129	9.60e-10	9	468	99	3.91e-09	10	533	**112**	6.38e-10
10^{-4}	10	544	128	7.10e-10	11	702	126	1.25e-09	11	605	**122**	1.79e-09
10^{-5}	11	649	136	9.44e-09	12	844	137	4.13e-09	12	742	**133**	1.23e-08
10^{-6}	13	924	175	2.02e-10	14	1228	198	1.53e-10	14	1054	**174**	5.86e-10
10^{-7}	14	1250	232	3.47e-10	14	1226	**187**	6.81e-09	15	1379	219	8.17e-10
10^{-8}	15	1476	258	4.36e-09	16	1683	258	3.74e-09	15	1362	**216**	5.33e-09
10^{-9}	15	1395	246	2.93e-08	17	1915	250	2.94e-08	16	1560	**235**	3.02e-08
10^{-10}	17	1762	273	1.45e-07	18	2249	296	3.34e-07	18	1969	**270**	3.13e-07

best results for order variation are obtained with order 6 and 8, however they are worse than those obtained with order 6, this means that for great tolerance no order variation is necessary. Moreover, we observe that order 8 have to be used when $TOL = 10^{-6}$, as shown in the Table 3.10, while orders 6 and 8 are the better combination for order variation to reach the same accuracy, as we can see in the Table 3.11. Also for this problem order 8 is confirmed to be the best choice when we require tolerance 10^{-8}, see Table 3.12, however the results in the Table 3.13 show that the finer mesh, which preserve the same accuracy, is obtained with the order variation 4-6-8. Consequently, we can point out that the order variation strategy improves the efficiency of the code for number of mesh points and high precision.

Example 3.5.3. Let us consider the linear test problem 7 in [29]

$$\epsilon y'' + xy' - y = -(1 + \epsilon\pi^2)\cos(\pi x) - \pi x \sin(\pi x), \qquad x \in [-1, 1],$$

Table 3.8: Example 3.5.2 - numerical solution with $TOL = 10^{-4}$ and fixed order.

ϵ	order 4				order 6				order 8			
10^{-1}	4	76	30	1.17e-4	2	30	**20**	6.65e-5	2	60	40	1.72e-8
10^{-2}	6	218	68	5.31e-5	3	70	**40**	9.19e-5	2	60	**40**	2.19e-5
10^{-3}	7	345	94	9.69e-5	5	200	77	8.83e-6	4	173	**63**	7.76e-5
10^{-4}	8	453	119	1.32e-4	6	314	99	4.77e-5	6	400	**94**	1.98e-5
10^{-5}	8	561	155	6.84e-5	8	555	136	4.29e-6	7	579	**121**	8.34e-5
10^{-6}	11	1190	248	7.77e-5	8	581	**131**	6.72e-5	8	832	171	7.99e-6
10^{-7}	10	1045	246	5.76e-5	9	799	**172**	1.71e-5	9	1095	204	4.30e-6
10^{-8}	10	1029	243	1.04e-4	10	1060	**224**	6.06e-6	10	1337	233	3.65e-6
10^{-9}	12	1779	354	9.86e-5	11	1219	**214**	2.03e-5	11	1581	244	5.94e-5
10^{-10}	11	1478	339	5.76e-5	12	1495	**247**	1.33e-5	12	1911	282	7.21e-6
10^{-11}	12	1933	411	5.93e-5	12	1556	**270**	4.78e-5	13	2176	297	1.56e-5
10^{-12}	12	1928	406	9.54e-5	13	1848	**275**	5.02e-5	14	2546	317	9.84e-6
10^{-13}	13	2395	468	7.40e-5	14	2052	**274**	3.35e-5	15	2922	353	1.25e-5
10^{-14}	15	4063	1090	9.55e-6	15	2609	**386**	6.68e-6	16	3381	395	1.17e-6
10^{-15}	16	4093	684	1.06e-4	16	2938	422	2.02e-6	17	3782	**413**	2.00e-6
10^{-16}	15	3567	666	1.17e-4	16	2902	**345**	2.69e-5	17	3833	440	6.31e-5
10^{-17}	16	4380	737	1.06e-4	17	3377	**403**	2.12e-5	18	4307	465	1.31e-5

with boundary conditions $y(-1) = -1$, $y(1) = 1$. The exact solution

$$y_e(x) = \cos(\pi x) + x + \frac{x\mathrm{erf}(x/\sqrt{2\epsilon}) + \sqrt{2\epsilon/\pi}\exp(-x^2/2\epsilon)}{\mathrm{erf}(1/\sqrt{2\epsilon}) + \sqrt{2\epsilon/\pi}\exp(-1/2\epsilon)}$$

has a corner layer in the turning point region near $x = 0$. We observe that the coefficient multiplying y' changes sign in the x-domain. As pointed out for the previous examples with tolerance 10^{-4} it is convenient to consider order 6, see Table 3.14, while for the same precision an order variation 4-6 seems not to offer any advantage, as we can see in the Table 3.15. Tolerances 10^{-6} and 10^{-8} are satisfied more easy using only order 6 and 8 respectively, see Table 3.16 and Table 3.18. However, some improvements in the reduction of the mesh points are gained with order variation, even if the choice of orders combination depends on ϵ, as shown in Table 3.17 and Table 3.19.

Example 3.5.4. Let us consider the test problem 14 in [29]

$$\epsilon y'' - y = -\left(\epsilon\pi^2 + 1\right)\cos(\pi x), \qquad x \in [-1, 1],$$

with boundary conditions $y(-1) = y(1) = \exp(-2/\sqrt{\epsilon})$. The exact solution

$$y_e(x) = \cos(\pi x) + \exp\left((x-1)/\sqrt{\epsilon}\right) + \exp\left(-(x+1)/\sqrt{\epsilon}\right)$$

has boundary layers of width $O(\sqrt{\epsilon})$ near $x = -1$ e $x = 1$.

Table 3.9: Example 3.5.2 - numerical solution with $TOL = 10^{-4}$ and variable order.

ϵ	orders 4-6				orders 4-8				orders 6-8			
10^{-1}	4	52	20	6.65e-5	4	58	**24**	5.23e-6	4	81	34	4.95e-8
10^{-2}	5	128	47	1.85e-5	4	81	**34**	7.21e-5	4	81	**34**	7.21e-5
10^{-3}	7	275	83	4.28e-6	6	231	72	1.85e-5	8	322	**69**	7.65e-6
10^{-4}	7	329	**93**	2.28e-5	7	354	109	5.56e-5	8	465	98	1.18e-5
10^{-5}	8	507	**130**	8.73e-6	8	545	148	3.76e-6	9	659	133	5.11e-6
10^{-6}	9	725	162	8.23e-6	9	756	179	8.00e-7	10	805	**159**	8.70e-5
10^{-7}	10	896	**177**	6.95e-6	10	934	212	1.47e-6	11	1047	191	1.41e-6
10^{-8}	10	899	**189**	5.05e-5	10	964	219	4.10e-5	12	1203	197	1.33e-5
10^{-9}	11	1200	**248**	1.78e-5	11	1249	272	7.49e-6	13	1492	255	2.32e-5
10^{-10}	12	1455	258	7.20e-6	12	1535	312	6.81e-7	14	1666	**251**	2.15e-6
10^{-11}	13	1806	309	2.18e-6	13	1893	362	7.90e-8	14	1749	**284**	2.61e-4
10^{-12}	14	2134	327	4.62e-6	14	2215	382	7.41e-7	16	2266	**305**	3.01e-6
10^{-13}	14	2019	**314**	2.58e-5	14	2132	377	4.19e-5	17	2631	331	1.82e-6
10^{-14}	15	2429	366	1.09e-5	15	2525	418	5.34e-6	17	2636	**343**	3.41e-5
10^{-15}	16	2854	**412**	2.03e-6	16	2958	470	4.28e-7	18	3202	**412**	1.41e-5
10^{-16}	17	3266	462	2.90e-6	17	3372	515	1.85e-7	19	3389	**400**	4.54e-6
10^{-17}	18	3635	466	4.23e-6	17	3224	515	1.89e-4	20	3860	**455**	8.16e-7

Table 3.10: Example 3.5.2 - numerical solution with $TOL = 10^{-6}$ and fixed order.

ϵ	order 4				order 6				order 8			
10^{-1}	11	509	132	3.79e-07	3	70	**40**	4.02e-07	2	60	**40**	1.72e-08
10^{-2}	12	985	177	1.06e-06	5	206	80	5.83e-07	4	182	**71**	2.02e-07
10^{-3}	12	1712	572	1.71e-07	6	329	**111**	3.20e-07	6	373	116	5.61e-08
10^{-4}	14	2018	528	2.81e-07	7	456	133	6.46e-07	6	481	**129**	9.96e-08
10^{-5}	12	1691	490	9.83e-07	8	681	191	5.68e-07	7	673	**152**	2.49e-07
10^{-6}	19	4664	589	2.40e-06	9	882	224	1.85e-07	8	943	**190**	1.06e-07
10^{-7}	13	2984	1108	1.77e-07	9	941	**246**	1.13e-06	9	1269	264	1.12e-07
10^{-8}	16	4945	1680	7.02e-08	10	1277	304	1.58e-07	9	1397	**300**	8.02e-07
10^{-9}	14	3903	1416	3.60e-07	11	1769	412	4.02e-08	10	1718	**307**	1.37e-06
10^{-10}	14	4208	1560	2.02e-07	11	1735	370	1.36e-06	11	2114	**316**	2.64e-07
10^{-11}	15	5377	1748	2.43e-07	12	2337	486	3.79e-07	11	2256	**391**	1.06e-06
10^{-12}	19	7224	1562	4.54e-07	12	2308	449	1.76e-06	12	2617	**377**	5.64e-07
10^{-13}	17	7182	2432	1.43e-07	14	3555	693	1.03e-07	13	3053	**399**	3.58e-07
10^{-14}	16	6243	2180	6.03e-07	14	3598	637	3.51e-07	14	3674	**458**	2.24e-07
10^{-15}	20	8600	1908	5.08e-07	14	3657	632	3.80e-07	15	4299	**495**	5.24e-08
10^{-16}	22	11939	2468	3.75e-07	15	4745	852	1.06e-07	16	4853	**533**	6.03e-08
10^{-17}	21	14064	3656	2.37e-07	16	4815	654	5.79e-07	16	4998	**577**	4.12e-07

Table 3.11: Example 3.5.2 - numerical solution with $TOL = 10^{-6}$ and variable order.

ϵ	order 4-8				order 6-8				order 4-6-8			
10^{-1}	5	106	48	3.46e-09	4	81	**34**	4.95e-08	6	127	35	4.05e-08
10^{-2}	6	189	62	4.72e-07	6	212	**61**	4.78e-07	7	256	64	5.26e-07
10^{-3}	7	335	102	1.26e-07	7	388	108	7.15e-08	8	365	**90**	6.96e-07
10^{-4}	8	483	122	1.19e-07	8	553	136	3.33e-07	9	575	**127**	1.49e-07
10^{-5}	9	720	175	6.99e-08	9	699	**144**	1.49e-06	10	828	173	4.90e-08
10^{-6}	9	756	179	8.00e-07	10	909	**169**	4.72e-07	11	1096	197	5.83e-08
10^{-7}	11	1184	250	2.26e-08	11	1209	**215**	9.91e-08	12	1328	231	4.95e-08
10^{-8}	11	1200	**224**	8.76e-07	12	1637	316	6.65e-08	12	1395	267	1.90e-07
10^{-9}	12	1548	299	1.10e-07	13	1750	**284**	1.99e-07	13	1820	325	2.01e-08
10^{-10}	12	1535	**312**	6.81e-07	14	2096	319	2.53e-08	14	2072	333	3.08e-08
10^{-11}	13	1893	362	7.90e-08	14	2221	349	2.12e-07	14	2149	**343**	4.48e-07
10^{-12}	14	2215	382	7.41e-07	15	2510	**346**	4.42e-07	16	2877	365	1.62e-08
10^{-13}	15	2531	399	1.16e-07	16	2724	**354**	3.62e-07	16	2852	446	6.95e-08
10^{-14}	16	2934	**409**	9.75e-08	17	3568	492	2.52e-08	17	3295	441	4.24e-08
10^{-15}	16	2958	**470**	4.28e-07	17	3429	491	3.84e-07	17	3327	473	3.53e-07
10^{-16}	17	3372	515	1.85e-07	18	3757	**441**	1.23e-07	18	3819	553	9.10e-07
10^{-17}	18	3745	521	3.43e-08	19	4315	**467**	4.36e-07	19	4166	531	1.06e-06

Table 3.12: Example 3.5.2 - numerical solution with $TOL = 10^{-8}$ and fixed order.

ϵ	order 4				order 6				order 8			
10^{-1}	13	1301	528	1.52e-09	5	202	88	1.91e-09	3	140	**80**	3.23e-11
10^{-2}	26	6449	613	1.44e-08	8	577	147	9.61e-09	5	276	**94**	1.42e-08
10^{-3}	14	5144	2288	6.79e-10	10	990	200	1.36e-08	6	389	**121**	2.90e-08
10^{-4}	17	6296	2148	1.75e-09	14	2089	323	1.01e-08	7	708	**176**	3.73e-09
10^{-5}	14	4631	1960	3.88e-09	10	1395	468	1.01e-09	8	987	**230**	1.91e-09
10^{-6}	23	10034	2704	3.57e-09	12	2018	584	4.74e-10	9	1364	**308**	9.86e-10
10^{-7}	14	5200	2216	1.12e-08	11	1666	371	1.62e-08	9	1439	**343**	1.03e-08
10^{-8}	17	8305	3360	4.45e-09	12	2482	764	1.05e-09	10	1915	**425**	6.32e-09
10^{-9}	15	6735	2832	2.28e-08	13	3137	906	7.44e-10	11	2569	**533**	3.62e-09
10^{-10}	15	7328	3120	1.30e-08	13	2852	576	1.10e-08	11	2566	**537**	1.51e-08
10^{-11}	16	8873	3496	1.56e-08	15	5297	1434	4.59e-10	12	3453	**682**	1.03e-09
10^{-12}	20	10348	3124	2.85e-08	14	3726	**704**	5.21e-09	13	4061	740	1.01e-08
10^{-13}	17	7182	2432	1.43e-07	15	4353	**798**	1.50e-08	13	4453	827	4.00e-09
10^{-14}	16	6243	2180	6.03e-07	15	4722	865	7.72e-09	14	5071	**776**	2.26e-09
10^{-15}	21	12416	3816	3.22e-08	19	10304	2298	2.10e-10	14	5312	**861**	3.38e-09
10^{-16}	22	11939	2468	3.75e-07	16	6744	1782	1.04e-09	15	6321	**991**	3.14e-09
10^{-17}	21	14064	3656	2.37e-07	17	7792	2000	4.43e-10	15	6506	**995**	8.88e-09

Table 3.13: Example 3.5.2 - numerical solution with $TOL = 10^{-8}$ and variable order.

ϵ	order 4-8				order 6-8				order 4-6-8			
10^{-1}	5	106	**48**	3.46e-09	5	149	68	1.39e-10	7	197	70	1.04e-10
10^{-2}	8	372	104	5.83e-09	8	388	**98**	6.66e-09	9	442	103	3.28e-09
10^{-3}	8	508	140	1.30e-08	8	519	**131**	7.07e-09	10	621	139	8.94e-09
10^{-4}	9	734	206	2.09e-09	9	712	160	1.22e-08	10	729	**154**	8.24e-09
10^{-5}	10	995	235	5.11e-09	10	874	**175**	1.18e-08	11	1042	202	7.86e-09
10^{-6}	10	1037	**239**	2.74e-09	11	1206	249	1.70e-09	12	1400	281	8.63e-10
10^{-7}	11	1265	305	5.60e-09	12	1523	**282**	2.65e-09	13	1622	291	5.43e-09
10^{-8}	12	1758	378	4.06e-09	13	2018	371	9.43e-09	13	1677	**282**	5.05e-09
10^{-9}	12	1675	**382**	1.70e-08	15	2470	387	3.58e-09	14	2221	401	2.49e-09
10^{-10}	13	1966	406	3.79e-09	15	2548	422	4.74e-09	15	2480	**404**	2.96e-09
10^{-11}	14	2351	447	1.67e-09	15	2641	**411**	3.91e-09	15	2584	435	5.24e-09
10^{-12}	15	2843	548	1.12e-09	17	3702	788	1.59e-10	17	3292	**415**	8.77e-09
10^{-13}	16	3277	554	5.88e-09	17	3292	520	5.41e-09	17	3361	**509**	2.75e-09
10^{-14}	16	3228	600	7.25e-09	18	4194	627	3.83e-09	18	3871	**558**	5.37e-10
10^{-15}	17	3571	574	7.13e-09	19	4757	725	5.20e-09	18	3819	**492**	7.67e-09
10^{-16}	18	4034	652	4.64e-09	19	4399	595	3.37e-09	19	4371	**552**	6.41e-09
10^{-17}	19	4401	656	2.65e-09	20	5153	702	2.22e-08	20	4738	**572**	1.63e-08

Table 3.14: Example 3.5.3 - numerical solution with $TOL = 10^{-4}$ and fixed order.

ϵ	order 4				order 6				order 8			
10^{-1}	3	70	40	1.78e-05	2	30	**20**	7.22e-06	2	60	40	3.17e-09
10^{-2}	5	122	37	7.66e-05	2	30	**20**	3.49e-04	2	60	40	1.12e-06
10^{-3}	9	204	47	1.03e-04	4	121	51	4.26e-06	2	60	**40**	6.87e-04
10^{-4}	7	211	59	1.13e-04	4	128	**58**	7.58e-04	4	167	61	5.50e-04
10^{-5}	7	252	**74**	9.70e-05	5	213	83	1.94e-04	6	317	88	3.51e-04
10^{-6}	8	364	106	2.55e-04	5	215	**87**	4.49e-04	6	317	88	9.42e-04
10^{-7}	8	362	104	1.58e-04	5	209	**80**	7.57e-04	7	427	110	4.85e-04
10^{-8}	8	358	101	2.78e-04	5	216	**88**	3.00e-04	7	427	110	5.28e-04
10^{-9}	8	358	101	2.81e-04	5	216	**88**	5.78e-04	7	427	110	5.32e-04
10^{-10}	8	358	101	2.81e-04	5	204	**75**	8.12e-04	7	427	110	5.32e-04
10^{-11}	8	358	101	2.81e-04	5	216	**88**	3.01e-04	7	427	110	5.32e-04
10^{-12}	8	358	101	2.81e-04	5	208	**80**	8.08e-04	7	427	110	5.32e-04
10^{-13}	8	358	101	2.81e-04	5	204	**75**	8.12e-04	7	427	110	5.32e-04
10^{-14}	8	358	101	2.81e-04	5	209	**80**	7.78e-04	7	427	110	5.32e-04
10^{-15}	8	358	101	2.81e-04	5	209	**80**	7.78e-04	7	427	110	5.32e-04
10^{-16}	8	358	101	2.81e-04	5	204	**75**	8.12e-04	7	426	109	4.04e-04

Table 3.15: Example 3.5.3 - numerical solution with $TOL = 10^{-4}$ and variable order.

ϵ	orders 4-6				orders 4-8				orders 6-8			
10^{-1}	3	47	**17**	4.22e-05	3	47	**17**	2.51e-05	4	81	34	1.73e-08
10^{-2}	4	81	**34**	1.21e-05	4	81	**34**	4.46e-06	4	81	**34**	4.46e-06
10^{-3}	7	154	51	4.26e-06	6	117	48	3.03e-04	5	117	**36**	5.64e-04
10^{-4}	6	150	**58**	5.61e-04	6	167	61	9.51e-04	5	149	68	1.42e-03
10^{-5}	7	251	**82**	1.41e-04	7	252	84	8.10e-04	6	236	87	7.49e-04
10^{-6}	7	251	**82**	3.74e-04	8	352	100	5.96e-04	7	336	101	4.05e-04
10^{-7}	7	250	**81**	4.75e-04	8	352	100	6.72e-04	7	336	101	5.34e-04
10^{-8}	7	250	**81**	4.95e-04	8	352	100	6.76e-04	7	337	102	7.27e-04
10^{-9}	7	250	**81**	4.97e-04	8	352	100	6.76e-04	7	336	101	5.69e-04
10^{-10}	7	250	**81**	4.97e-04	8	352	100	6.76e-04	7	337	102	7.29e-04
10^{-11}	7	250	**81**	4.97e-04	8	352	100	6.76e-04	7	337	102	7.29e-04
10^{-12}	7	250	**81**	4.97e-04	8	352	100	6.76e-04	7	337	102	7.29e-04
10^{-13}	7	250	**81**	4.97e-04	8	352	100	6.76e-04	7	337	102	7.29e-04
10^{-14}	7	250	**81**	4.97e-04	8	352	100	6.76e-04	7	337	102	7.29e-04
10^{-15}	7	250	**81**	4.97e-04	8	352	100	6.76e-04	7	337	102	7.29e-04
10^{-16}	7	250	**81**	4.97e-04	8	352	100	6.76e-04	7	337	102	7.29e-04

Table 3.16: Example 3.5.3 - numerical solution with $TOL = 10^{-6}$ and fixed order.

ϵ	order 4				order 6				order 8			
10^{-1}	5	310	160	7.56e-08	3	70	**40**	1.12e-07	2	60	**40**	3.17e-09
10^{-2}	11	584	110	8.43e-07	4	121	51	6.33e-07	2	60	**40**	1.12e-06
10^{-3}	15	790	137	9.56e-07	5	184	**63**	9.79e-07	4	178	67	4.01e-07
10^{-4}	12	781	156	1.19e-06	6	288	90	4.23e-07	6	334	**87**	3.94e-06
10^{-5}	11	808	181	8.24e-07	7	447	126	1.28e-07	8	547	**119**	3.40e-06
10^{-6}	10	662	165	1.90e-06	8	562	**140**	1.73e-07	10	808	141	1.67e-07
10^{-7}	11	844	195	9.29e-07	9	766	176	1.26e-07	11	1020	**170**	1.56e-06
10^{-8}	12	1070	237	1.52e-06	9	759	**176**	3.96e-06	12	1174	179	2.10e-06
10^{-9}	12	1067	235	2.95e-06	10	962	213	4.68e-06	13	1428	**213**	3.32e-06
10^{-10}	13	1338	272	2.16e-06	10	1036	236	4.26e-06	13	1355	**187**	8.36e-06
10^{-11}	13	1344	278	2.51e-06	11	1194	235	1.93e-06	13	1379	**196**	8.33e-06
10^{-12}	13	1344	278	2.90e-06	10	961	207	8.26e-06	13	1385	**195**	1.01e-05
10^{-13}	13	1344	278	2.93e-06	10	1034	234	5.37e-06	13	1407	**206**	6.61e-06
10^{-14}	13	1344	278	2.94e-06	11	1199	230	3.37e-06	13	1368	**185**	9.57e-06
10^{-15}	13	1344	278	2.94e-06	10	1034	234	5.37e-06	13	1403	**195**	7.49e-06
10^{-16}	13	1344	278	2.94e-06	10	1034	234	5.38e-06	13	1365	**185**	6.72e-06

Table 3.17: Example 3.5.3 - numerical solution with $TOL = 10^{-6}$ and variable order.

ϵ	order 4-8				order 6-8				order 4-6-8			
10^{-1}	4	81	**34**	1.73e-08	4	81	**34**	1.73e-08	6	118	**34**	1.73e-08
10^{-2}	5	125	44	3.98e-07	5	125	44	3.98e-07	7	183	**40**	4.90e-07
10^{-3}	7	177	**60**	3.80e-06	6	260	78	5.96e-08	9	293	78	5.96e-08
10^{-4}	8	336	**88**	7.32e-07	7	357	90	5.39e-07	9	407	92	2.02e-07
10^{-5}	10	586	131	2.18e-07	8	524	**116**	2.32e-06	10	587	121	1.06e-07
10^{-6}	11	748	**139**	3.47e-06	10	838	146	2.08e-07	11	789	159	7.59e-08
10^{-7}	12	894	**152**	6.55e-06	11	1019	181	1.12e-06	12	943	170	1.75e-06
10^{-8}	13	1092	186	7.47e-06	12	1206	**183**	1.18e-06	13	1188	199	5.92e-07
10^{-9}	14	1247	**184**	7.15e-06	12	1162	185	1.31e-05	13	1151	187	7.29e-06
10^{-10}	15	1472	202	5.09e-06	12	1170	**177**	1.67e-05	13	1168	192	1.34e-05
10^{-11}	14	1262	197	1.32e-05	13	1404	198	5.97e-06	14	1360	**195**	4.66e-06
10^{-12}	14	1293	198	1.06e-05	13	1417	218	8.80e-06	13	1162	**188**	1.25e-05
10^{-13}	14	1299	201	1.10e-05	13	1402	206	7.47e-06	13	1169	**195**	1.34e-05
10^{-14}	15	1539	225	5.90e-06	14	1615	219	3.60e-06	14	1368	**199**	4.53e-06
10^{-15}	15	1465	209	4.43e-06	13	1414	209	8.50e-06	13	1164	**183**	1.26e-05
10^{-16}	14	1291	202	1.40e-05	13	1365	**190**	9.13e-06	15	1604	234	3.40e-06

Table 3.18: Example 3.5.3: numerical solution with $TOL = 10^{-8}$ and fixed order.

ϵ	order 4				order 6				order 8			
10^{-1}	6	630	320	4.76e-09	4	150	80	1.87e-09	2	60	**40**	3.17e-09
10^{-2}	18	2840	654	1.18e-09	7	365	99	7.45e-09	4	185	**71**	8.08e-09
10^{-3}	20	2400	379	1.93e-08	8	501	125	6.60e-09	5	277	**96**	1.12e-08
10^{-4}	19	2708	376	1.81e-08	8	534	**132**	9.10e-09	7	498	142	4.36e-09
10^{-5}	18	3022	430	1.11e-08	8	620	157	2.17e-08	9	727	**155**	3.44e-09
10^{-6}	17	2900	447	8.51e-09	8	662	182	1.92e-08	10	928	**174**	5.49e-09
10^{-7}	16	2467	413	1.01e-08	9	916	244	7.81e-09	11	1160	**197**	9.48e-09
10^{-8}	16	2486	423	9.60e-09	11	1369	291	1.83e-09	12	1455	**228**	8.10e-09
10^{-9}	16	2540	449	9.48e-09	11	1415	300	5.30e-09	13	1678	**243**	5.32e-08
10^{-10}	16	2482	437	7.95e-09	12	1793	341	2.45e-09	15	2217	**280**	5.39e-09
10^{-11}	16	2504	445	2.67e-08	13	1920	313	2.74e-08	16	2559	**300**	1.04e-08
10^{-12}	17	3039	518	1.23e-08	14	2291	358	1.80e-08	17	2845	**325**	5.22e-08
10^{-13}	18	3616	581	6.67e-09	14	2501	377	4.25e-08	17	2831	**314**	9.47e-08
10^{-14}	18	3612	582	2.49e-08	14	2464	378	9.25e-08	18	3082	**323**	7.04e-08
10^{-15}	19	4264	655	1.01e-08	15	2828	381	4.02e-08	18	3222	**338**	6.43e-08
10^{-16}	19	4266	653	1.88e-08	15	2848	348	1.86e-08	18	3171	**316**	4.22e-08

Table 3.19: Example 3.5.3 - numerical solution with $TOL = 10^{-8}$ and variable order.

ϵ	order 4-8				order 6-8				order 4-6-8			
10^{-1}	5	149	**68**	1.44e-11	5	149	**68**	1.44e-11	7	186	**68**	1.44e-11
10^{-2}	7	257	76	4.28e-09	7	257	76	4.28e-09	9	306	**70**	6.38e-09
10^{-3}	8	268	**87**	1.95e-08	7	362	103	4.16e-09	10	395	103	4.16e-09
10^{-4}	9	494	135	2.40e-09	8	514	134	2.01e-09	10	547	**130**	1.29e-09
10^{-5}	10	647	**139**	3.57e-09	9	689	145	2.07e-09	11	739	152	1.21e-09
10^{-6}	11	858	176	9.21e-09	10	921	**168**	1.34e-09	12	972	183	7.11e-10
10^{-7}	12	1093	212	1.27e-08	11	1147	**189**	1.27e-08	13	1217	230	2.59e-09
10^{-8}	14	1545	238	5.75e-10	12	1427	227	5.46e-09	14	1435	**226**	2.68e-08
10^{-9}	15	1785	237	7.97e-09	13	1593	**223**	5.00e-08	16	1994	280	8.05e-10
10^{-10}	16	2061	**268**	7.71e-09	15	2130	275	3.67e-09	17	2245	305	5.42e-09
10^{-11}	17	2446	314	4.37e-08	16	2485	**282**	8.51e-09	18	2465	291	1.57e-08
10^{-12}	18	2639	299	7.26e-08	17	2717	311	7.23e-08	19	2726	**286**	4.17e-08
10^{-13}	18	2777	**300**	8.10e-08	18	3162	349	2.39e-08	20	3216	337	1.24e-08
10^{-14}	19	3043	319	7.15e-08	18	3204	342	3.89e-08	20	3075	**314**	5.19e-08
10^{-15}	20	3293	**313**	7.20e-08	18	3195	345	6.04e-08	20	3069	337	9.28e-08
10^{-16}	19	2982	311	9.63e-08	18	3053	**310**	6.80e-08	20	3171	334	9.64e-08

This is the only problem considered with two boundary layers and without y'. As shown in Table 3.20 and Table 3.22 for tolerances $TOL = 10^{-4}$ and $TOL = 10^{-6}$, respectively, order 4 allows us to obtain better results, moreover from the Table 3.21 and the Table 3.23 we note that an order variation seems not to be suitable for these precisions. For $TOL = 10^{-8}$ both orders 6 and 8 reach the accuracy required, see Table 3.24, even if to reduce the number of mesh points it is convenient to apply an order variation strategy, as we can see in Table 3.25.

Example 3.5.5. The nonlinear problem

$$\epsilon y'' - \exp(y)y' - \pi/2 \sin(\pi x/2) \exp(2y) = 0, \qquad y(0) = y(1) = 0$$

is named Test Problem 19 in [29]. For $\epsilon \to 0$ the asymptotic solution

$$y(x) = -\ln((1 + \cos(\pi x/2))(1 - \exp(-x/(2\epsilon))/2)) + o(\epsilon).$$

The solution has a boundary layer in $x = 0$. We point out that the continuation is not considered for this problem. For $TOL = 10^{-4}$, see Table 3.26, a good accuracy is obtained with order 4, while for $TOL = 10^{-6}$ the order 6 gives the finer mesh, see Table 3.28. As suggested from the Table 3.27 and from the Table 3.29 we not have a great advantage to apply order variation strategy for large tolerance. For $TOL = 10^{-8}$ orders 6 or 8 reach the best results, see Table 3.30, even if the finer mesh is obtained with order variation 4-6-8 for many values of the perturbation ϵ.

Table 3.20: Test Problem 3.5.4: numerical solution with tol = 10^{-4} and fixed order.

ϵ	order 4				order 6				order 8			
10^{-1}	2	30	**20**	7.25e-05	2	30	**20**	1.09e-06	2	42	22	1.38e-07
10^{-2}	4	83	34	1.26e-04	4	74	**30**	1.84e-04	2	51	31	3.86e-05
10^{-3}	5	145	56	5.72e-05	5	142	56	2.44e-05	3	102	**46**	2.80e-04
10^{-4}	6	236	89	9.28e-06	6	201	**65**	1.04e-04	5	247	85	4.49e-06
10^{-5}	6	223	**76**	7.00e-05	7	300	90	8.65e-06	6	361	114	3.91e-07
10^{-6}	7	319	99	6.85e-06	7	303	**92**	1.92e-04	6	379	116	9.14e-06
10^{-7}	8	466	150	1.10e-06	8	440	**131**	3.63e-06	7	616	181	6.26e-09
10^{-8}	8	472	**147**	2.13e-06	9	627	182	4.05e-08	7	573	166	6.42e-06
10^{-9}	8	498	**144**	3.16e-05	9	652	189	3.67e-07	8	842	235	1.53e-09
10^{-10}	9	653	**170**	8.13e-07	9	653	188	1.48e-04	9	971	258	6.54e-10
10^{-11}	9	643	**184**	5.09e-05	10	863	226	7.35e-08	9	1139	284	1.48e-10
10^{-12}	10	942	**229**	3.41e-07	11	1119	279	8.32e-10	9	1085	275	5.33e-06
10^{-13}	10	864	**211**	1.13e-06	11	1142	277	9.23e-09	10	1527	347	6.45e-10
10^{-14}	10	895	**226**	7.10e-05	12	1370	314	1.61e-09	10	1419	322	1.29e-04
10^{-15}	10	1029	**266**	1.62e-04	11	1299	290	9.46e-06	11	1824	392	1.43e-09

Table 3.21: Test Problem 3.5.4: numerical solution with tol = 10^{-4} and variable order.

ϵ	orders 4-6				orders 4-8				orders 6-8			
10^{-1}	3	47	**17**	2.97e-06	3	47	**17**	6.86e-07	3	47	**17**	6.86e-07
10^{-2}	6	123	37	1.33e-05	6	108	**34**	2.04e-05	6	107	**34**	2.04e-05
10^{-3}	6	196	63	7.40e-06	6	199	62	5.64e-06	6	175	**52**	6.52e-05
10^{-4}	6	219	**72**	2.01e-04	6	233	86	5.00e-05	7	275	77	2.44e-05
10^{-5}	7	315	**92**	2.26e-05	7	331	108	3.11e-06	8	405	110	1.02e-06
10^{-6}	8	409	111	3.28e-06	8	445	134	2.27e-07	8	412	**109**	1.43e-04
10^{-7}	8	429	**113**	1.45e-04	8	445	129	1.84e-05	9	599	159	1.06e-06
10^{-8}	9	630	**158**	3.04e-07	9	654	182	6.67e-09	10	791	198	3.81e-09
10^{-9}	9	656	**158**	7.44e-06	9	683	185	6.92e-07	10	870	218	8.47e-08
10^{-10}	10	844	**191**	1.77e-08	10	860	207	3.80e-10	10	859	206	7.89e-05
10^{-11}	10	831	**188**	1.53e-05	10	856	213	1.04e-06	11	1112	249	9.66e-09
10^{-12}	11	1239	309	1.34e-09	10	945	**232**	5.27e-05	11	1085	245	2.45e-04
10^{-13}	11	1102	**238**	1.28e-07	11	1119	255	1.41e-09	12	1449	307	2.24e-10
10^{-14}	11	1141	**246**	1.93e-05	11	1184	289	1.78e-06	12	1328	272	2.44e-04
10^{-15}	11	1305	**276**	6.99e-05	11	1326	297	9.94e-06	12	1609	310	3.02e-06

Table 3.22: Test Problem 3.5.4: numerical solution with tol = 10^{-6} and fixed order.

ϵ	order 4				order 6				order 8			
10^{-1}	5	202	88	1.30e-07	2	30	20	1.09e-06	2	42	**22**	1.38e-07
10^{-2}	7	265	72	1.49e-06	6	164	51	1.17e-06	3	91	**40**	8.86e-07
10^{-3}	7	328	103	1.64e-06	6	222	77	3.62e-07	4	193	**76**	2.32e-07
10^{-4}	7	352	116	1.36e-06	7	382	134	1.38e-08	5	270	**85**	5.77e-07
10^{-5}	7	389	134	6.22e-07	7	324	**108**	1.31e-06	6	361	114	3.91e-07
10^{-6}	7	396	139	9.66e-07	8	427	**124**	3.00e-07	7	537	158	6.33e-09
10^{-7}	8	538	188	3.00e-07	9	617	**177**	6.61e-09	7	616	181	6.26e-09
10^{-8}	8	514	**167**	8.01e-07	9	627	182	4.05e-08	8	785	212	3.41e-10
10^{-9}	9	726	220	2.05e-07	9	652	**189**	3.67e-07	8	842	235	1.53e-09
10^{-10}	9	657	**174**	7.19e-07	10	893	240	2.24e-09	9	971	258	6.54e-10
10^{-11}	10	854	**211**	3.54e-07	10	863	226	7.35e-08	9	1139	284	1.48e-10
10^{-12}	10	942	**229**	3.41e-07	11	1119	279	8.32e-10	10	1420	335	3.81e-11
10^{-13}	10	864	**211**	1.13e-06	11	1142	277	9.23e-09	10	1527	347	6.45e-10
10^{-14}	11	1162	**267**	1.78e-07	12	1370	314	1.61e-09	11	1803	384	1.48e-09
10^{-15}	11	1338	**309**	7.41e-08	12	1698	399	4.82e-09	11	1824	392	1.43e-09

Table 3.23: Test Problem 3.5.4: numerical solution with tol = 10^{-6} and variable order.

ϵ	order 4-8				order 6-8				order 4-6-8			
10^{-1}	3	47	17	6.86e-07	3	47	17	6.86e-07	4	62	**15**	4.80e-07
10^{-2}	7	153	**45**	3.62e-07	7	190	52	7.85e-08	8	207	48	1.68e-07
10^{-3}	7	282	**85**	5.59e-08	7	290	88	6.93e-08	8	345	**85**	2.50e-08
10^{-4}	7	329	96	1.46e-07	8	364	**87**	3.97e-07	8	420	111	1.86e-07
10^{-5}	8	465	134	1.13e-08	9	541	**132**	1.09e-08	9	584	143	1.00e-08
10^{-6}	8	445	134	2.27e-07	9	559	147	2.25e-08	9	531	**122**	1.10e-06
10^{-7}	9	620	175	2.19e-09	9	599	**159**	1.06e-06	10	745	170	8.73e-09
10^{-8}	9	654	**182**	6.67e-09	10	836	209	5.79e-09	10	813	183	5.89e-08
10^{-9}	9	683	185	6.92e-07	10	870	218	8.47e-08	10	840	**184**	1.89e-06
10^{-10}	10	860	**207**	3.80e-10	11	1116	257	7.74e-11	11	1116	272	1.48e-09
10^{-11}	10	856	**213**	1.04e-06	11	1112	249	9.66e-09	12	1332	290	8.49e-11
10^{-12}	11	1251	306	5.93e-11	12	1423	**304**	1.49e-10	12	1561	322	7.07e-11
10^{-13}	11	1119	**255**	1.41e-09	12	1449	307	2.24e-10	12	1437	335	1.13e-08
10^{-14}	11	1184	**289**	1.78e-06	13	1721	351	1.56e-09	13	1869	379	1.22e-09
10^{-15}	12	1717	**391**	3.60e-09	13	2014	405	1.13e-09	13	2128	422	2.60e-09

Table 3.24: Test Problem 3.5.4: numerical solution with tol = 10^{-8} and fixed order.

ϵ	order 4				order 6				order 8			
10^{-1}	10	931	270	1.03e-09	3	70	40	1.77e-08	5	123	**30**	1.08e-08
10^{-2}	15	1493	344	3.08e-09	9	466	158	1.48e-10	5	215	**68**	2.35e-09
10^{-3}	13	1606	436	2.53e-09	8	452	125	9.59e-09	5	320	**118**	1.23e-09
10^{-4}	12	1419	284	1.66e-08	7	384	135	1.20e-08	5	318	**108**	1.23e-08
10^{-5}	11	1308	288	1.48e-08	7	436	**159**	5.54e-09	6	509	160	5.32e-10
10^{-6}	10	1225	324	9.33e-09	7	459	178	9.79e-09	7	621	**170**	3.74e-10
10^{-7}	9	994	305	1.32e-08	8	545	195	8.22e-09	7	616	**181**	6.26e-09
10^{-8}	9	956	313	1.65e-08	9	664	**206**	3.55e-09	8	789	217	3.31e-10
10^{-9}	10	1310	411	4.36e-09	10	891	236	9.73e-10	8	842	**235**	1.53e-09
10^{-10}	10	1161	365	8.55e-09	10	896	**240**	1.95e-09	9	971	258	6.54e-10
10^{-11}	11	1390	409	5.25e-09	11	1176	311	2.60e-10	9	1139	**284**	1.48e-10
10^{-12}	11	1562	449	3.58e-09	11	1119	**279**	8.32e-10	10	1420	335	3.81e-11
10^{-13}	11	1301	381	1.08e-08	11	1142	**277**	8.04e-09	10	1527	347	6.45e-10
10^{-14}	12	1868	519	3.46e-09	12	1370	**314**	1.61e-09	11	1803	384	1.48e-09
10^{-15}	11	1507	438	1.08e-08	12	1698	398	4.47e-09	11	1824	**392**	1.43e-09

Table 3.25: Test Problem 3.5.4: numerical solution with tol = 10^{-8} and variable order.

ϵ	order 4-8				order 6-8				order 4-6-8			
10^{-1}	9	188	**30**	1.08e-08	9	188	**30**	1.08e-08	12	236	**30**	1.08e-08
10^{-2}	9	288	74	8.50e-10	8	253	63	5.62e-09	9	268	**61**	1.16e-08
10^{-3}	7	304	**95**	7.12e-09	8	405	117	1.04e-09	9	455	110	1.29e-09
10^{-4}	8	506	160	2.08e-10	9	556	162	1.27e-10	9	555	**135**	8.75e-10
10^{-5}	8	465	134	1.13e-08	9	541	**132**	1.09e-08	9	584	143	1.00e-08
10^{-6}	9	617	170	4.92e-10	10	754	196	8.32e-11	10	685	**154**	3.40e-09
10^{-7}	9	620	175	2.19e-09	10	803	204	3.72e-10	10	745	**170**	8.73e-09
10^{-8}	9	654	**182**	6.67e-09	10	836	209	5.79e-09	11	1044	231	7.27e-11
10^{-9}	10	926	243	9.20e-11	11	1139	269	1.88e-11	11	1074	**234**	1.28e-10
10^{-10}	10	860	**207**	3.80e-10	11	1116	257	7.74e-11	11	1116	272	1.48e-09
10^{-11}	11	1148	292	3.08e-11	11	1112	**249**	9.66e-09	12	1332	290	8.49e-11
10^{-12}	11	1251	306	5.93e-11	12	1423	**304**	1.49e-10	12	1561	322	7.07e-11
10^{-13}	11	1119	**255**	1.41e-09	12	1449	307	2.24e-10	12	1437	335	1.13e-08
10^{-14}	12	1570	386	1.48e-09	13	1721	**351**	1.56e-09	13	1869	379	1.22e-09
10^{-15}	12	1717	**391**	3.60e-09	13	2014	405	1.13e-09	13	2128	422	2.60e-09

Table 3.26: Example 3.5.5 - numerical solution with $TOL = 10^{-4}$ and fixed order.

ϵ	order 4				order 6				order 8			
10^{-1}	4	65	25	7.24e-05	4	60	**21**	3.18e-05	2	45	25	3.77e-06
10^{-2}	5	145	52	5.24e-05	6	175	62	2.81e-05	5	171	**49**	2.47e-05
10^{-3}	6	223	75	6.66e-05	7	274	72	3.93e-05	8	345	**68**	2.76e-05
10^{-4}	7	309	**93**	7.39e-05	9	471	99	1.08e-05	11	580	96	9.05e-05
10^{-5}	8	431	110	8.01e-05	10	564	**98**	8.45e-05	14	949	134	2.82e-05
10^{-6}	10	692	153	1.25e-05	12	880	**152**	9.09e-06	17	1473	198	1.23e-05
10^{-7}	11	888	169	2.19e-05	13	1024	**164**	2.76e-05	19	1687	204	6.91e-05
10^{-8}	12	1116	208	1.69e-05	15	1362	**192**	6.22e-06	21	2102	225	9.73e-05
10^{-9}	13	1283	223	8.40e-06	16	1591	**213**	1.70e-05	22	2488	283	4.24e-05
10^{-10}	14	1469	**217**	1.37e-05	17	1915	258	3.37e-05	25	3464	355	3.44e-06
10^{-11}	15	1784	**262**	2.16e-05	18	2195	275	8.21e-05	26	3871	384	3.68e-05
10^{-12}	16	1950	**279**	3.51e-05	20	2867	329	2.34e-06	28	4639	404	7.77e-06
10^{-13}	17	2290	**299**	2.05e-05	21	3243	362	3.05e-06	29	5147	455	4.11e-05
10^{-14}	18	2585	**327**	1.34e-05	22	3535	376	4.34e-06	31	6214	506	1.38e-05
10^{-15}	19	2907	**321**	9.67e-05	23	3936	372	1.20e-06	33	7367	577	2.89e-06
10^{-16}	20	3219	**363**	2.05e-05	24	4394	436	8.02e-05	34	7982	610	2.74e-05

Example 3.5.6. The nonlinear problem

$$y'' - \frac{1}{\epsilon}\sinh\left(\frac{y}{\epsilon}\right) = 0, \qquad y(0) = 0, \quad y(1) = 1$$

is named Test Problem 23 in [29] and it has a boundary layer near $x = 1$. For this example we show the results with and without continuation. In the case without continuation, see Table 3.32, for $TOL = 10^{-4}$, differently from the other examples, order 8 works better than the others, while for $TOL = 10^{-6}$, see Table 3.34, it is suitable to use order 4 for many ϵ and again order 8 for smaller values of the perturbation. For $TOL = 10^{-8}$ the Table 3.36 shows the best results using order 6. As we can check in Table 3.33, Table 3.35 and Table 3.37 the order variation improves the efficiency of the code by the reduction of the mesh points only when a small tolerance, as 10^{-8}, is required, in this case orders 4-6-8 allow us to reach a good accuracy with a less number of points. From Table 3.38 to Table 3.43 we can observe the results obtained with the continuation. We point out that the convergence behavior is exactly the same discussed in the case without the continuation, even if conversely to the first case the number of steps increases just for the continuation. However, we have observed as finer meshes for $TOL = 10^{-8}$ are gained always by applying order variation 4-6-8, but with the continuation strategy, see Table 3.37 and Table 3.43.

Table 3.27: Example 3.5.5 - numerical solution with $TOL = 10^{-4}$ and variable order.

ϵ	orders 4-6				orders 4-8				orders 6-8			
10^{-1}	5	74	21	3.18e-05	4	50	**14**	9.26e-05	4	48	**14**	9.26e-05
10^{-2}	6	183	61	1.70e-05	6	174	49	3.28e-05	9	194	**45**	9.53e-05
10^{-3}	8	343	86	5.08e-06	7	274	74	8.00e-05	9	342	**61**	3.17e-05
10^{-4}	9	436	98	1.72e-05	9	454	99	2.47e-05	11	558	**95**	3.49e-05
10^{-5}	10	565	**121**	2.71e-05	10	582	124	2.19e-05	13	808	124	3.45e-05
10^{-6}	12	829	**156**	3.62e-06	12	876	171	5.09e-06	15	1200	170	2.84e-05
10^{-7}	13	1012	**184**	4.14e-06	13	1071	199	5.68e-06	16	1539	221	7.80e-05
10^{-8}	14	1151	**188**	2.01e-05	14	1196	200	1.62e-05	18	1996	257	9.63e-05
10^{-9}	15	1454	**235**	1.26e-05	15	1507	257	1.52e-05	20	2490	280	3.87e-05
10^{-10}	16	1498	**225**	9.16e-05	17	1846	281	2.78e-06	22	3086	332	8.28e-06
10^{-11}	17	1867	**266**	4.28e-05	17	1930	293	4.04e-05	24	3722	366	7.64e-06
10^{-12}	19	2205	**288**	4.55e-06	19	2279	316	5.09e-06	25	4111	388	3.97e-05
10^{-13}	20	2652	**319**	1.03e-06	20	2799	369	2.58e-06	27	4891	439	1.27e-05
10^{-14}	21	2831	**346**	8.27e-06	21	2941	393	5.23e-06	29	5873	517	1.18e-06
10^{-15}	22	3316	**396**	2.36e-06	22	3409	445	6.05e-06	30	6288	515	8.67e-05
10^{-16}	22	3429	**394**	5.58e-05	23	4010	482	3.85e-07	32	7303	580	1.01e-06

Table 3.28: Example 3.5.5 - numerical solution with $TOL = 10^{-6}$ and fixed order.

ϵ	order 4				order 6				order 8			
10^{-1}	10	378	71	6.09e-07	6	130	39	8.49e-07	3	83	**37**	3.75e-07
10^{-2}	10	629	180	3.09e-07	7	263	**81**	8.67e-07	6	297	84	6.97e-08
10^{-3}	12	1024	174	9.11e-07	7	338	93	9.37e-07	8	427	**89**	3.32e-07
10^{-4}	12	1261	221	7.16e-07	9	598	**124**	3.14e-07	11	747	132	2.37e-07
10^{-5}	13	1595	280	3.62e-07	10	757	**154**	4.18e-07	12	975	157	4.33e-07
10^{-6}	13	1651	291	6.43e-07	11	973	**169**	9.96e-07	15	1468	197	5.24e-08
10^{-7}	13	1741	308	6.64e-07	13	1443	243	3.97e-08	16	1769	**228**	2.76e-07
10^{-8}	13	1933	347	4.78e-07	14	1610	**240**	4.25e-08	17	2053	256	6.96e-07
10^{-9}	14	2716	738	8.80e-08	15	1941	277	4.68e-08	19	2571	**274**	6.67e-08
10^{-10}	14	2320	425	7.79e-07	16	2176	**283**	2.53e-07	20	2920	306	9.67e-07
10^{-11}	15	2840	467	7.69e-07	18	2779	**355**	1.19e-08	22	3806	384	4.02e-08
10^{-12}	16	3517	553	2.54e-07	18	2907	**339**	5.39e-07	23	4094	388	7.05e-08
10^{-13}	17	4068	628	6.47e-07	20	3503	**391**	1.33e-08	24	4538	416	6.01e-07
10^{-14}	17	4162	635	4.39e-07	20	3564	**363**	4.76e-07	25	5035	457	4.32e-07
10^{-15}	17	4228	644	9.77e-07	21	4049	**419**	7.00e-07	27	6124	513	2.70e-08
10^{-16}	19	5654	745	3.83e-07	22	4553	**438**	4.11e-07	28	6580	528	2.90e-07

Table 3.29: Example 3.5.5 - numerical solution with $TOL = 10^{-6}$ and variable order.

ϵ	order 4-8				order 6-8				order 4-6-8			
10^{-1}	7	136	**37**	3.21e-07	7	144	**37**	4.29e-07	8	169	**37**	4.85e-07
10^{-2}	7	247	**62**	6.93e-07	8	314	73	9.54e-07	8	340	77	5.42e-07
10^{-3}	9	427	**93**	2.46e-07	10	553	109	8.64e-08	10	562	125	1.07e-07
10^{-4}	10	594	**121**	2.94e-07	11	712	132	3.95e-07	11	693	138	1.16e-07
10^{-5}	11	740	147	2.23e-07	13	944	**144**	2.59e-07	12	846	152	1.08e-07
10^{-6}	13	1071	193	5.06e-08	14	1242	186	2.79e-07	13	1003	**172**	9.37e-07
10^{-7}	14	1299	222	4.41e-08	15	1431	209	6.02e-07	15	1395	**207**	4.56e-08
10^{-8}	15	1441	**231**	2.00e-07	17	1808	232	1.16e-07	16	1615	248	1.04e-07
10^{-9}	16	1778	**269**	8.91e-08	18	2130	280	4.71e-07	17	2027	298	7.37e-08
10^{-10}	18	2155	305	1.09e-07	19	2540	319	7.25e-08	18	2032	**293**	7.08e-07
10^{-11}	18	2254	**322**	7.30e-07	20	2889	358	1.87e-07	19	2504	341	1.54e-07
10^{-12}	20	2647	354	5.98e-08	22	3653	405	4.09e-08	21	2855	**341**	9.72e-08
10^{-13}	20	2820	**379**	9.00e-07	23	4077	426	2.93e-08	21	3070	395	4.18e-07
10^{-14}	22	3380	425	5.48e-08	24	4434	463	1.25e-08	23	3655	**417**	2.27e-07
10^{-15}	23	3894	474	2.53e-08	24	4355	**419**	5.61e-07	23	3814	481	8.11e-07
10^{-16}	23	4015	**477**	6.97e-07	26	5441	540	1.06e-07	24	4331	490	9.53e-08

Table 3.30: Example 3.5.5 - numerical solution with $TOL = 10^{-8}$ and fixed order.

ϵ	order 4				order 6				order 8			
10^{-1}	14	1096	368	1.29e-09	10	422	128	2.06e-10	5	192	**60**	8.26e-10
10^{-2}	12	1709	720	1.51e-09	10	677	214	1.08e-09	6	388	**110**	2.23e-09
10^{-3}	14	2042	692	3.95e-09	10	800	165	8.00e-09	8	535	**138**	4.38e-09
10^{-4}	14	2996	1232	7.11e-10	9	788	207	5.56e-09	11	949	**179**	1.15e-09
10^{-5}	16	3765	1216	9.05e-10	10	1008	**238**	7.46e-09	13	1460	240	2.34e-10
10^{-6}	19	4429	886	5.74e-09	11	1351	319	2.61e-09	14	1720	**234**	1.28e-09
10^{-7}	16	4678	1572	9.70e-10	12	1647	336	1.38e-09	15	2184	**305**	6.55e-10
10^{-8}	21	8140	987	9.85e-09	13	1945	**304**	1.14e-09	16	2603	339	6.04e-09
10^{-9}	17	4757	970	9.56e-09	14	2499	429	5.86e-10	17	3036	**358**	8.07e-09
10^{-10}	22	7785	1432	1.94e-09	15	2920	457	4.66e-10	18	3428	**389**	4.44e-09
10^{-11}	21	8413	1684	1.24e-09	16	3483	476	9.27e-10	20	4279	**431**	2.09e-10
10^{-12}	24	10117	1044	9.97e-09	17	3829	472	7.92e-10	21	4896	**466**	2.59e-09
10^{-13}	25	10913	1051	8.55e-09	17	3901	**430**	8.71e-09	22	5564	520	4.59e-09
10^{-14}	25	11471	1146	9.35e-09	19	4850	**536**	4.48e-10	23	6147	541	6.74e-09
10^{-15}	24	11771	2134	8.74e-10	20	5435	**558**	1.13e-09	25	7217	637	2.20e-10
10^{-16}	25	12159	1262	9.23e-09	20	5140	**469**	2.53e-09	26	8027	628	1.09e-09

Table 3.31: Example 3.5.5 - numerical solution with $TOL = 10^{-8}$ and variable order.

ϵ	order 4-8				order 6-8				order 4-6-8			
10^{-1}	9	247	61	7.14e-10	9	249	**58**	9.43e-10	10	275	**58**	1.11e-09
10^{-2}	8	420	**111**	1.72e-09	10	573	134	9.31e-10	10	550	**111**	1.48e-09
10^{-3}	9	559	**135**	2.77e-09	10	609	138	8.60e-09	11	694	**135**	2.35e-09
10^{-4}	10	699	171	8.86e-09	12	899	172	1.33e-09	12	847	**154**	1.98e-09
10^{-5}	12	979	204	5.95e-10	14	1233	223	4.69e-09	13	1030	**184**	3.18e-09
10^{-6}	13	1169	**235**	3.59e-09	15	1503	241	2.38e-09	15	1432	**235**	1.42e-09
10^{-7}	14	1351	243	9.85e-09	16	1720	261	2.04e-09	16	1632	**236**	1.40e-09
10^{-8}	16	1756	284	2.37e-09	18	2082	**273**	2.56e-09	17	1896	281	1.81e-09
10^{-9}	17	2149	339	5.60e-10	19	2469	322	1.40e-09	18	2338	**317**	7.90e-10
10^{-10}	18	2240	348	4.96e-09	20	2926	373	1.33e-09	20	2684	**346**	1.14e-09
10^{-11}	19	2675	393	3.08e-09	21	3247	**370**	3.33e-09	20	2875	371	2.89e-09
10^{-12}	21	3068	419	2.10e-09	23	4074	425	4.82e-10	22	3249	**393**	8.64e-10
10^{-13}	21	3253	417	8.99e-09	24	4567	491	5.59e-10	22	3481	**411**	9.69e-09
10^{-14}	23	3883	499	2.72e-09	25	4957	518	3.55e-10	24	4143	**465**	3.16e-09
10^{-15}	24	4415	522	3.35e-10	26	5314	531	8.74e-10	24	4316	**502**	4.63e-09
10^{-16}	24	4546	513	6.92e-09	27	6001	561	1.59e-09	25	4841	**510**	1.60e-09

Table 3.32: Example 3.5.6 - numerical solution with $TOL = 10^{-4}$ and fixed order.

ϵ	order 4				order 6				order 8			
.714e-01	9	531	**137**	9.48e-05	11	837	184	5.89e-05	11	928	196	3.94e-05
.556e-01	12	1160	230	1.81e-06	13	1540	301	1.97e-05	14	1231	**214**	9.72e-05
.455e-01	13	1590	287	3.16e-05	15	2296	385	1.96e-05	18	1870	**248**	3.53e-05
.333e-01	15	2389	325	1.31e-05	17	2915	395	4.79e-05	21	2542	**278**	4.99e-05
.294e-01	16	3011	425	7.81e-05	20	3789	434	2.28e-05	23	3271	**281**	9.82e-05
.263e-01	18	4251	465	2.48e-05	22	4641	456	3.22e-05	27	4367	**311**	8.47e-05
.238e-01	19	5162	561	7.13e-05	24	5499	481	4.66e-05	30	5776	**319**	6.57e-05
.217e-01	22	7574	663	9.93e-05	28	7165	495	8.93e-05	36	8655	**395**	6.77e-05

3.6 Conclusion

The code HOFiD_UP preserves all the advantages to discretize the second and the first derivatives in the equations separately, no transformation in an equivalent system of the first order is necessary, so that the number of mesh points represents the real dimension of the problem. The code is able to solve linear and nonlinear singular perturbation problems. For small tolerances and perturbations, on a large scale of problems, it is opportune to consider an order variation strategy, sometimes for easy problem only order 8 may be enough to reach a good accuracy and efficiency. For nonlinear problems the code uses the continuation if it needs, in fact all nonlinear problem in [29] are successfully solved. We have also observed that the results of HOFiD_UP are

Table 3.33: Example 3.5.6 - numerical solution with $TOL = 10^{-4}$ and variable order.

ϵ	orders 4-6				orders 4-8				orders 6-8			
.714e-01	12	875	188	4.95e-05	14	1022	199	2.38e-05	14	881	**183**	7.84e-05
.556e-01	13	1352	277	8.31e-05	17	1434	**234**	2.76e-05	18	1492	246	2.77e-05
.455e-01	16	2326	387	1.59e-05	20	1764	**236**	6.99e-05	21	1801	246	6.93e-05
.333e-01	18	2974	403	3.16e-05	23	2566	274	4.98e-05	24	2533	**273**	4.94e-05
.294e-01	20	3554	412	7.61e-05	26	3518	308	4.59e-05	26	3280	**276**	8.13e-05
.263e-01	23	4811	467	1.91e-05	29	4353	297	8.32e-05	30	4332	**281**	8.58e-05
.238e-01	25	5613	469	2.95e-05	32	5764	**337**	6.53e-05	31	6130	369	7.95e-05
.217e-01	29	7287	494	7.21e-05	38	8683	396	6.77e-05	38	8674	**395**	7.48e-05

Table 3.34: Example 3.5.6 - numerical solution with $TOL = 10^{-6}$ and fixed order.

ϵ	order 4				order 6				order 8			
.714e-01	11	875	**185**	5.32e-07	13	1316	255	3.49e-07	12	1703	335	5.40e-08
.556e-01	12	1176	**246**	9.42e-07	15	2183	365	5.16e-08	13	2063	387	3.72e-07
.455e-01	14	2049	**372**	2.25e-07	16	3041	493	1.51e-07	16	3334	472	4.22e-08
.333e-01	16	2925	**457**	1.60e-07	18	4361	608	3.22e-08	18	3949	472	1.88e-07
.294e-01	17	3514	**427**	5.82e-07	19	4968	608	4.88e-07	21	5069	494	2.27e-07
.263e-01	19	4771	**532**	2.20e-07	21	6498	680	1.14e-07	24	6456	**532**	7.11e-08
.238e-01	20	5742	568	7.00e-07	23	7821	713	4.52e-08	26	7611	**513**	3.45e-07
.217e-01	23	8596	706	7.80e-07	26	10745	794	2.67e-08	32	11608	**563**	2.82e-07

Table 3.35: Example 3.5.6 - numerical solution with $TOL = 10^{-6}$ and variable order.

ϵ	order 4-8				order 6-8				order 4-6-8			
.714e-01	14	1455	283	5.88e-07	14	1537	270	6.75e-07	15	1564	**251**	3.16e-07
.556e-01	16	2363	398	1.23e-07	16	2562	399	9.55e-08	16	2440	**386**	3.65e-07
.455e-01	18	2989	444	1.88e-07	18	3554	464	1.11e-07	18	3332	**436**	5.13e-07
.333e-01	20	3793	460	3.04e-07	20	4174	**448**	6.89e-07	20	4099	460	8.85e-07
.294e-01	23	5069	**486**	2.47e-07	23	5205	505	1.47e-07	23	5454	545	8.83e-08
.263e-01	26	6493	532	1.20e-07	25	6104	**520**	3.97e-07	25	6195	574	1.80e-07
.238e-01	29	7979	**553**	1.46e-07	28	7631	597	5.14e-08	27	7158	576	5.38e-07
.217e-01	34	11635	**563**	2.82e-07	32	9306	573	7.04e-07	32	9558	672	1.31e-07

Table 3.36: Example 3.5.6 - numerical solution with $TOL = 10^{-8}$ and fixed order.

ϵ	order 4				order 6				order 8			
.714e-01	16	3162	934	6.38e-10	15	1954	**338**	1.31e-09	13	2081	378	1.59e-09
.556e-01	15	2397	628	8.62e-09	16	2592	**409**	3.51e-09	14	2680	464	6.99e-09
.455e-01	17	3686	884	3.41e-09	17	3685	**578**	2.20e-09	16	4330	660	5.75e-10
.333e-01	19	5111	1136	1.61e-09	19	5257	**736**	4.29e-10	18	5950	820	1.86e-10
.294e-01	22	7261	831	9.99e-09	20	6030	**806**	2.40e-09	20	7957	960	2.04e-10
.263e-01	21	7100	1462	1.75e-09	22	8041	**973**	7.84e-10	22	10253	1112	1.01e-10
.238e-01	23	9446	1784	8.07e-10	23	9067	**1072**	2.02e-09	24	12738	1199	1.75e-10
.217e-01	25	10862	**1144**	8.98e-09	25	12597	1336	4.07e-09	27	18303	1377	4.58e-09

Table 3.37: Example 3.5.6 - numerical solution with $TOL = 10^{-8}$ and variable order.

ϵ	order 4-8				order 6-8				order 4-6-8			
.714e-01	16	2156	373	9.91e-10	16	2256	378	9.42e-10	17	2174	**323**	7.22e-10
.556e-01	17	3044	524	7.78e-10	17	3029	462	2.04e-09	17	2871	**433**	7.78e-09
.455e-01	18	3992	641	3.35e-09	19	4229	585	1.00e-09	19	3885	**526**	8.42e-09
.333e-01	19	5407	764	5.73e-09	21	4942	**608**	6.14e-09	22	5398	663	2.82e-10
.294e-01	22	7945	946	2.37e-10	24	6061	680	1.05e-09	24	6151	**660**	1.16e-09
.263e-01	23	9243	1060	9.26e-09	26	7063	738	2.26e-09	26	6973	**722**	1.52e-09
.238e-01	26	12780	1206	1.63e-10	28	8138	805	4.88e-09	28	7955	**726**	4.97e-09
.217e-01	29	18317	1363	4.57e-09	33	10765	923	1.49e-09	33	10384	**802**	1.86e-09

Table 3.38: Example 3.5.6 - numerical solution with $TOL = 10^{-4}$ and fixed order using continuation.

ϵ	order 4				order 6				order 8			
.714e-01	9	531	**137**	9.48e-05	11	837	184	5.89e-05	11	928	196	3.94e-05
.556e-01	12	1005	215	7.18e-05	14	1461	290	2.58e-05	15	1254	**209**	8.31e-05
.455e-01	15	1611	288	3.87e-05	17	2300	381	2.26e-05	19	1782	**246**	9.05e-05
.333e-01	18	2359	313	1.19e-05	20	2756	367	9.37e-05	24	2622	**286**	4.42e-05
.294e-01	20	3081	434	8.93e-05	24	3842	420	2.66e-05	27	3434	**290**	8.85e-05
.263e-01	23	4255	478	4.34e-05	27	4644	455	3.96e-05	32	4592	**319**	7.15e-05
.238e-01	26	5432	471	1.06e-05	30	5531	463	5.36e-05	36	5878	**339**	6.80e-05
.217e-01	30	7954	614	2.30e-05	36	7682	532	2.57e-05	43	8959	**395**	5.59e-05

Table 3.39: Example 3.5.6 - numerical solution with $TOL = 10^{-4}$ and variable order using continuation.

ϵ	orders 4-6				orders 4-8				orders 6-8			
.714e-01	12	875	188	4.95e-05	14	1022	199	2.38e-05	14	881	**183**	7.84e-05
.556e-01	15	1572	301	1.97e-05	18	1450	**234**	2.76e-05	19	1505	246	2.77e-05
.455e-01	18	2344	385	1.96e-05	22	1768	**231**	8.03e-05	23	1828	246	6.93e-05
.333e-01	21	2979	395	4.79e-05	26	2434	**254**	6.07e-05	27	2574	273	4.94e-05
.294e-01	25	3869	434	2.28e-05	29	3425	311	9.75e-05	30	3335	**276**	8.13e-05
.263e-01	28	4737	456	3.22e-05	34	4415	292	8.56e-05	35	4401	**281**	8.58e-05
.238e-01	31	5559	462	4.88e-05	38	5854	**337**	6.71e-05	39	5895	364	6.06e-05
.217e-01	37	7787	531	2.60e-05	44	9141	394	7.66e-05	47	8770	**378**	5.04e-05

Table 3.40: Example 3.5.6 - numerical solution with $TOL = 10^{-6}$ and fixed order using continuation.

ϵ	order 4				order 6				order 8			
.714e-01	11	875	**185**	5.32e-07	13	1316	255	3.49e-07	12	1703	335	5.40e-08
.556e-01	13	1256	**251**	8.47e-07	16	2245	390	8.10e-08	15	2447	410	5.95e-08
.455e-01	16	2037	**368**	2.58e-07	18	2890	486	3.24e-07	18	2931	426	3.65e-07
.333e-01	19	2876	**445**	1.55e-07	21	4415	595	3.49e-08	22	4131	489	1.05e-07
.294e-01	21	3556	**410**	4.69e-07	24	5323	664	3.30e-08	25	4989	491	2.52e-07
.263e-01	24	4852	525	2.60e-07	26	6441	675	1.46e-07	29	6370	**474**	2.83e-07
.238e-01	27	6382	630	1.40e-07	29	8014	715	4.51e-08	33	7936	**530**	2.72e-07
.217e-01	31	8725	771	9.44e-08	33	10520	752	6.96e-08	39	11691	**578**	3.38e-07

Table 3.41: Example 3.5.6 - numerical solution with $TOL = 10^{-6}$ and variable order using continuation.

ϵ	order 4-8				order 6-8				order 4-6-8			
.714e-01	14	1455	283	5.88e-07	14	1537	270	6.75e-07	15	1564	**251**	3.16e-07
.556e-01	17	2379	398	1.23e-07	17	2447	382	1.74e-07	18	2543	**376**	3.73e-08
.455e-01	20	2916	434	4.23e-07	20	3555	459	1.53e-07	20	3257	**418**	6.23e-07
.333e-01	24	4138	**488**	1.57e-07	24	4419	489	1.47e-07	24	4572	503	4.20e-08
.294e-01	27	4774	**446**	8.45e-07	27	5231	512	2.11e-07	27	5187	500	3.03e-07
.263e-01	31	6302	**496**	2.88e-07	30	6059	492	6.48e-07	30	6129	532	5.68e-07
.238e-01	35	8250	**546**	1.09e-07	34	7592	595	8.42e-08	33	7005	577	9.87e-07
.217e-01	41	11496	**546**	7.42e-07	39	9421	618	3.51e-07	39	9420	621	6.01e-07

Table 3.42: Example 3.5.6 - numerical solution with $TOL = 10^{-8}$ and fixed order using continuation.

ϵ	order 4				order 6				order 8			
.714e-01	16	3162	934	6.38e-10	15	1954	**338**	1.31e-09	13	2081	378	1.59e-09
.556e-01	16	2488	632	7.29e-09	17	2677	**432**	4.60e-09	15	2746	486	9.59e-09
.455e-01	19	3827	916	2.65e-09	19	3453	**543**	6.26e-09	18	3695	585	8.32e-09
.333e-01	22	5379	1206	1.29e-09	22	5369	**757**	4.22e-10	21	5645	800	2.19e-09
.294e-01	25	7215	1488	8.80e-10	25	6437	**831**	4.46e-10	24	7529	932	9.99e-10
.263e-01	27	8535	1690	7.69e-10	27	8020	**990**	5.91e-10	27	9860	1026	1.31e-09
.238e-01	30	10546	1922	7.12e-10	29	9324	**1102**	1.66e-09	30	12693	1152	1.31e-09
.217e-01	33	11441	**1170**	8.12e-09	33	13861	1426	2.74e-10	35	18774	1341	1.58e-09

Table 3.43: Example 3.5.6 - numerical solution with $TOL = 10^{-8}$ and variable order using continuation.

ϵ	order 4-8				order 6-8				order 4-6-8			
.714e-01	16	2156	373	9.91e-10	16	2256	378	9.42e-10	17	2174	**323**	7.22e-10
.556e-01	18	3060	524	7.78e-10	18	2936	457	2.97e-09	19	2958	**415**	1.11e-09
.455e-01	21	4312	648	2.26e-10	21	4221	**578**	1.23e-09	22	4382	579	2.57e-10
.333e-01	23	5636	793	3.14e-09	25	5323	653	7.57e-10	25	5146	**574**	1.17e-09
.294e-01	26	7422	914	1.20e-09	28	6036	674	1.86e-09	28	5907	**648**	3.33e-09
.263e-01	29	9742	1019	1.49e-09	31	7052	738	3.22e-09	31	6907	**694**	5.52e-09
.238e-01	32	11828	1119	2.64e-09	34	7992	748	9.12e-09	35	8339	**715**	1.32e-09
.217e-01	37	17758	1326	3.40e-09	40	10566	902	1.66e-09	40	10297	**785**	5.96e-09

comparable with those obtained by codes as ACDC and COLMOD. As regards small values of the perturbation ϵ HOFiD_UP employs less points than ACDC, while it is comparable with COLMOD. It is not possible at the moment to compare these codes on the performance time, since HOFiD_UP is a Matlab code, but this suggests a good motivation to implement a Fortran version of it.

Chapter 4

Second-Order Initial Value Problems

Most of the second order initial value problems (IVPs) arise from celestial mechanics and lack of the first derivative term, but for such problems (called conservative) ad hoc methods have been developed in order to preserve some properties of the continuous solution. Several codes exist to solve initial value problems, we remember some of them as VODE [26] using Adams and BDFs methods to handle both non stiff and stiff problems; BiMD [27] based on Blended Implicit Methods, a class of L-stable Block Implicit Methods; GAMD [42] exploting the Generalized Adams Methods in block form; RADAU and RADAU5 [39] based on implicit Runge-Kutta methods (Radau IIa), in particular RADAU5 is of order 5 and employs a stepsize control rendering a continuous solution.

In this chapter second order IVPs with nonnull derivative terms are treated since they are not integrated with classical linear multistep formulae. Moreover, a wide class of such problems arises from singular problems, i.e. not defined at some points of the domain. The methods developed so far (see [8]) follow the idea of the HOGD methods introduced in Chapter 2, so that high order finite difference schemes are employed in the approximation of the first and second derivatives. The different available choices for the main scheme allows to define seven classes of methods. Some numerical tests show the efficiency of these methods especially in the case of very difficult problems, as for example 'Flow in concrete'.

4.1 High order finite difference schemes

Let us consider the following second-order initial value problem (IVP)

$$\begin{aligned} &f(x, y, y', y'') = 0, \qquad x \in [a, b] \\ &y(a) = y_a, \quad y'(a) = y'_a, \end{aligned} \tag{4.1.1}$$

where y_a and y'_a are known initial values, f is a Lipschitz continuous function, so that for Theorem 1.1.10 a unique solution $y(x)$ exists. Moreover, let $\mathfrak{I} = \{\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_N\}$ be a subdivision of the interval $[a, b]$ in N subintervals, where $\mathcal{I}_j = [a_j, b_j]$ and

$$a = a_1 < a_2 < \cdots < a_n < b, \quad a < b_1 < b_2 < \cdots < b_n = b,$$

but $\mathcal{I}_j$ and $\mathcal{I}_{j+1}$ have at most two elements into the intersection. The idea is to solve (4.1.1) iteratively starting from $\mathcal{I}_1$ and using on $\mathcal{I}_{j+1}$ initial values obtained by $\mathcal{I}_j$.The subdivision of the interval is performed considering constant stepsize inside $\mathcal{I}_j$. The initial stepsize for the first interval $\mathcal{I}_1$ can be either preserved for the successive interval or changed as we will see in Section 4.4.

Now our aim is to find the solution of (4.1.1) on all the interval $[a, b]$ or on a part of it. For this reason, we consider a uniform discretization

$$a_1 = x_0 < x_1 < \cdots < x_n \leq b_1, \tag{4.1.2}$$

with stepsize $h = (x_n - x_0)/n$. Following the idea proposed in Section 2.1 for the solution of BVPs, the approximation of the second and the first derivatives at the mesh points is calculated by means of the high order difference formulae (2.1.6); this means that, for $i = 0, \ldots, n$,

$$y^{(\nu)}(x_i) \approx y_i^{(\nu)} = \frac{1}{h^\nu} \sum_{j=-l}^{k-l} \alpha_{j+l}^{(l,\nu)} y_{i+j}, \qquad \nu = 1, 2, \tag{4.1.3}$$

where the coefficients $\alpha_0^{(l,\nu)}, \ldots, \alpha_k^{(l,\nu)}$ are computed such that the formulae have maximum order p, as in Theorem 2.1.2, and depend on the number of initial conditions $l = 0, \ldots, k$ and ν. In (4.1.3) the index k depends on ν and the order of the formula; in particular the approximation of order p for $y^{(\nu)}(x_i)$ is in general defined on $p + \nu$ points, except for the case p even, $\nu = 2$ and $l = k/2$ where a $(p + \nu - 1)$-steps formula is required.

It is important to point out that for IVPs (4.1.3) the values of the solution and the first derivative in the left-endpoint of the interval are known, but

while the first condition on $y(x_0)$ is involved in the formulae (4.1.3), the second condition on $y'(x_0)$ at the moment is not used; however in the next section 4.3 we will see as this condition is exploit to define different approaches.

It should be recalled that the methods approximating the derivatives are based on the idea of Boundary Value Methods [28], as showed in Section 1.5. For each derivative we fix the order and derive the set of finite difference schemes (4.1.3) by changing conveniently the numbers l and $k-l$ of initial and final conditions, respectively. Among these formulae, we emphasize the *main scheme* which will be used when possible on the internal points of the mesh (4.1.2). The other formulae (or some of them) will be used once in the extreme points of the mesh.

As explained in Section 2.2, the choice of the extended central differences formulae (ECDFs) as main scheme requires to consider the same number of initial and final conditions. Moreover, in this case the coefficients of ECDFs are symmetric for the second derivative and skew-symmetric for the first derivative, as seen in Theorem 2.1.3. On the other hand, it is possible to consider as main scheme a generalization of backward (GB) or forward (GF) difference formulae , if $l-1$ or $l-2$ and $l+1$ or $l+2$ final conditions are considered, respectively, depending on the order of the method. The order of the main method is the same for the first and the second derivative approximations and it is also possible to choose between even or odd order.

Consequently, for the approximation of the second derivative we can distinguish two classes depending on the order p. The first class includes the main scheme of even order, so defined $k=p$ in the formula (4.1.3), needs to set $l=\frac{k}{2}$ to obtain the symmetric scheme ECDF. The second class includes main formula of odd order, then set $k=p+1$, we have two choices: GF defined when $l=\frac{k}{2}-1$ or GB defined for $l=\frac{k}{2}+1$.

For the first derivative, we have also to distinguish between even and odd order. For even order as in Definition 2.2.2, set $k=p$, formulae GFDF, ECDF and GBDF can be defined by choosing l among $\frac{k}{2}-1, \frac{k}{2}, \frac{k}{2}+1$, respectively. For odd order, set $k=p$, we can choose $l=\frac{k-1}{2}$ and $l=\frac{k+1}{2}$ in order to define GFDF and GBDF.

Definition 4.1.1. A global approximation of the second derivative y'' by means of formulae of order p is called

- $\mathbf{EC}_2$ if $l=p/2$ and p is even;
- $\mathbf{GB}_2$ if $l=(p-1)/2+1$ and p is odd;

- $\mathbf{GF}_2$ if $l = (p+1)/2 - 1$ and p is odd;

where l is the number of initial conditions of the main scheme.

Definition 4.1.2. A global approximation of the first derivative y' by means of formulae of order p is called

- $\mathbf{EC}_1$ if $l = p/2$ and p is even;
- $\mathbf{GB}_1$ if $l = p/2 + 1$ and p is even or if $l = (p-1)/2 + 1$ and p is odd;
- $\mathbf{GF}_1$ if $l = p/2 - 1$ and p is even od if $l = (p+1)/2 - 1$ and p is odd;

where l is the number of initial conditions of the main scheme.

Definition 4.1.3. For the solution of second-order IVP (4.1.1), the combinations of the schemes for y'' and y' given in Definition 4.1.1-4.1.2 are called for p even order $\mathbf{EC}_2\mathbf{EC}_1$, $\mathbf{EC}_2\mathbf{GF}_1$ and $\mathbf{EC}_2\mathbf{GB}_1$ and for odd order $\mathbf{GB}_2\mathbf{GB}_1$, $\mathbf{GB}_2\mathbf{GF}_1$, $\mathbf{GF}_2\mathbf{GB}_1$ and $\mathbf{GF}_2\mathbf{GF}_1$.

In Section 4.4 devoted to the numerical tests, we only consider couple of methods with the same approximation for the derivatives, namely $\mathrm{EC}_2\mathrm{EC}_1$, $\mathrm{GB}_2\mathrm{GB}_1$, and $\mathrm{GF}_2\mathrm{GF}_1$.

Example 4.1.4. In the following we consider the main schemes of order 5 and 6 for the approximation of y' and y''. The coefficients of the GB_2 (GB_1) schemes are symmetric (skew-symmetric) with respect to those of the GF_2 (GF_1) schemes for this reason they are omitted.

Order 5

$$\mathrm{GF}_2: \quad h^2\, y''(t_i) \approx -\frac{13}{180}y_{i-2} + \frac{19}{15}y_{i-1} - \frac{7}{3}y_i + \frac{10}{9}y_{i+1} + \frac{1}{12}y_{i+2} - \frac{1}{15}y_{i+3} + \frac{1}{90}y_{i+4}$$

$$\mathrm{GF}_1: \quad h\, y'(t_i) \approx \frac{1}{20}y_{i-2} - \frac{1}{2}y_{i-1} - \frac{1}{3}y_i + y_{i+1} - \frac{1}{4}y_{i+2} + \frac{1}{30}y_{i+3}$$

Order 6

$$\mathrm{EC}_2: \quad h^2\, y''(t_i) \approx \frac{1}{90}y_{i-3} - \frac{3}{20}y_{i-2} + \frac{3}{2}y_{i-1} - \frac{49}{18}y_i + \frac{3}{2}y_{i+1} - \frac{3}{20}y_{i+2} + \frac{1}{90}y_{i+3}$$

$$\mathrm{EC}_1: \quad h\, y'(t_i) \approx -\frac{1}{60}y_{i-3} + \frac{3}{20}y_{i-2} - \frac{3}{4}y_{i-1} + \frac{3}{4}y_{i+1} - \frac{3}{20}y_{i+2} + \frac{1}{60}y_{i+3}$$

$$\mathrm{GF}_1: \quad h\, y'(t_i) \approx \frac{1}{30}y_{i-2} - \frac{2}{5}y_{i-1} - \frac{7}{12}y_i + \frac{4}{3}y_{i+1} - \frac{1}{2}y_{i+2} + \frac{2}{15}y_{i+3} - \frac{1}{60}y_{i+4}$$

4.2 Conditioning

The goal of this section is to investigate the stability properties of the main schemes for the two derivatives; for this reason, let us analyze their behavior on a scalar linear problem of the form

$$y'' + \gamma y' + \mu y = 0, \tag{4.2.1}$$

where γ and μ are real numbers independent of x. The well conditioning of (4.2.1) depends on the eigenvalues λ_1 and λ_2 associated to the coefficients matrix of the equivalent first-order system,

$$\begin{pmatrix} y \\ y' \end{pmatrix}' = \begin{pmatrix} 0 & 1 \\ -\mu & -\gamma \end{pmatrix} \begin{pmatrix} y \\ y' \end{pmatrix},$$

that is on the roots of the characteristic polynomial

$$\begin{vmatrix} -\lambda & 1 \\ -\mu & -\gamma - \lambda \end{vmatrix} = \lambda^2 + \gamma\lambda + \mu = 0.$$

If $\delta = \gamma^2 - 4\mu > 0$ then $\lambda_1 = -\gamma - \sqrt{\delta}$ and $\lambda_2 = -\gamma + \sqrt{\delta}$ are the eigenvalues and the exact solution is

$$y(x) = c_1 \mathrm{e}^{\lambda_1 x} + c_2 \mathrm{e}^{\lambda_2 x}, \tag{4.2.2}$$

where c_1 and c_2 depend on the initial conditions. A well conditioned initial value problem (the solution goes to zero, see [28]) requires that λ_1 and λ_2 are non positive, that is γ and μ are non negative values. Consequently, the exact solution (4.2.2) is decreasing when $\gamma > 0$ and bounded for $\gamma = 0$. Depending on the initial conditions, $y(x)$ may be strictly positive on all the interval. The results show as it is suitable to consider methods yielding well conditioned discrete problem when $\gamma > 0$ and $\mu > 0$.

Now, let us consider the discrete problem corresponding to (4.2.1). We point out that when the size n of the grid becomes large, then the effect of the additional methods on the solution may be considered negligible. In such a case the conditioning especially depends on the main methods, therefore it is sufficient to study the roots of the characteristic polynomial

$$\pi(z, h^2\mu, h\gamma) = \rho(z) + h\gamma\sigma(z) + h^2\mu z^{\bar{s}}, \qquad \bar{s} = \max(s_1, s_2)$$

where

$$\rho(z) = \sum_{j=-s_2}^{k-s_2} \alpha_{j+s_2}^{(s_2,2)} z^{s_2+j}, \qquad \sigma(z) = \sum_{j=-s_1}^{k-s_1} \alpha_{j+s_1}^{(s_1,1)} z^{s_1+j}$$

are the polynomials associated to the main schemes discretizing, respectively, the second and the first derivative term in (4.2.1),

As a consequence of the Theorem 2.3.1, we require that the number of upper off-diagonals of the coefficient matrix of the discrete problem matches the number of roots of π outside the open unit disk [4, 28]. Since γ and μ are non negative real numbers, the boundary locus defined by

$$\pi(z, \mu h^2, \gamma h) = 0 \qquad \text{for} \quad |z| = 1, \tag{4.2.3}$$

is drawn in the quarter of the plane with $h\gamma \geq 0$ and $h^2\mu \geq 0$. The characteristic polynomial (4.2.3) is linear, then we distinguish three cases:

(i) $z = 1$: for the consistency of the schemes both $\rho(1) = \sigma(1) = 0$, then boundary is represented by the abscissa $h^2\mu = 0$;

(ii) $z = -1$: the condition (4.2.3) can be written as

$$\hat{\sigma} h\gamma + h^2\mu + \hat{\rho} = 0, \tag{4.2.4}$$

where $\hat{\rho} = \rho(-1)/(-1)^{\bar{s}}$ and $\hat{\sigma} = \sigma(-1)/(-1)^{\bar{s}}$. Through simple calculations $\hat{\rho} < 0$ for all schemes, while $\hat{\sigma} > 0$ for GB_1, $\hat{\sigma} < 0$ for GF_1 and $\hat{\sigma} < 0$ for EC_1. It is clear that the straight line described by (4.2.4) intersects the $h^2\mu$-axis at $|\rho|$ and the $h\gamma$-axis at $\dfrac{|\hat{\rho}|}{\hat{\sigma}}$. In Table 4.1 and Table 4.2 we have summarized the values of $|\hat{\rho}|$ and $\hat{\sigma}$ for different methods and orders. Moreover, the straight lines corresponding to GB_1, GF_1 and EC_1 schemes are decreasing, increasing and parallel to the $h\gamma$-axis, respectively. Hence, the use of GF_1 schemes give rise to the largest stability domain.

(iii) $|z| = 1$ and $\mathrm{Im}(z) \neq 0$: in this case we have

$$\rho(z) + h\gamma\sigma(z) + h^2\mu z^{\bar{s}} = 0,$$

$$\rho(\bar{z}) + h\gamma\sigma(\bar{z}) + h^2\mu \bar{z}^{\bar{s}} = 0,$$

then it follows through some substitutions that $\rho(\bar{z}) - \bar{z}^{\bar{s}}\rho(z)/z^{\bar{s}} = 0$. Moreover, the curve corresponding to the complex values of z of unitary modulus starts from the origin and it is quite near to the $h^2\mu$-axis and, in the case of EC_1 schemes, it coincides with the segment $0 \leq h^2\mu \leq |\rho(z)|\,/z^{\bar{s}}$.

Now, as in [28] we give some definitions on stability region.

Table 4.1: Coefficients $\hat{\sigma}$ and $\hat{\rho}$ of the straight line (4.2.4) corresponding to $\pi(-1) = 0$. Even order approximations.

		order			
		4	6	8	10
EC_2	ρ_1	16/3	272/45	2048/315	512/75
GF_1	σ_1	−8/3	−32/15	−64/35	−512/315
GB_1	σ_1	8/3	32/15	64/35	512/315
EC_1	σ_1	0	0	0	0

Table 4.2: Coefficients $\hat{\sigma}$ and $\hat{\rho}$ of the straight line (4.2.4) corresponding to $\pi(-1) = 0$. Odd order approximations.

		order			
		3	5	7	9
GF_2/GB_2	ρ_1	8/3	208/45	352/63	9278/1575
GF_1	σ_1	−4/3	−16/15	−32/35	−256/315
GB_1	σ_1	4/3	16/15	32/35	256/315

Definition 4.2.1. The region $\mathcal{D}_{k_1,k_2}$ of the complex plane defined by

$$\mathcal{D}_{k_1k_2} = \{(h^2\mu, h\gamma) \in \mathbb{C} : \pi(z, h^2\mu, h\gamma) \text{ is the type } (k_1, 0, k_2)\}$$

is called the *region of* (k_1, k_2)*-Absolute stability*

Definition 4.2.2. A main method with (k_1, k_2)-boundary conditions is said to be $A_{k_1k_2}$*-stable* if $\mathbb{C}^- \subseteq \mathcal{D}_{k_1k_2}$. Moreover, it is called to be *perfectly* $A_{k_1k_2}$*-stable* if $\mathbb{C}^- \equiv \mathcal{D}_{k_1k_2}$.

As a consequence of Definition 4.2.2, we can underline that all methods in Definition 4.1.3 are not $A_{s,k-s}$-stable, since they do not give stable methods for every value of h. As an example, we plot the stability domains for the GF_2GF_1 and GB_2GB_1 schemes of order 5 and 7 respectively in Figure 4.1 and in Figure 4.2. Then, we observe that larger stability domain belongs to higher order methods.

Remark 4.2.3. The case $\gamma = 0$ is not efficiently solved by this approach when the interval is very large, in fact, as it is known, ad hoc methods, as the symmetric linear multistep methods, are used to solve second order IVPs $y'' = f(t, y)$.

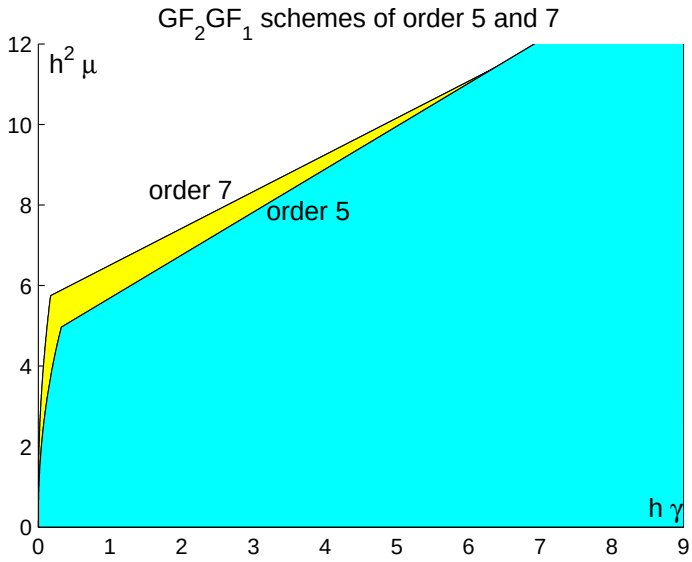


Figure 4.1: *Stability regions for the GF_2GF_1 scheme of order 5 and 7.*

4.3 Additional formulae

From the conditioning analysis of the main methods it is known that the number of roots outside the boundary locus represents the number of final conditions of the method, conversely the number of roots inside it is that corresponding to the initial conditions. Despite the continuous problem (4.1.1) has two initial conditions, its solution by means of the methods in Definition 4.1.3 should require l initial and $k-l$ final formulae.

In this section we have a stake in approximating y' and y'' at both the extreme points of (4.1.2) by using additional formulae, that for the initial value problem (4.1.1) still belong to the family (4.1.3). In fact, we follow the idea developed for the BVPs in Chapter 2 and approximate the first and the second derivatives in (4.1.1) by applying main schemes, seen in Definition 4.1.3, on the internal points x_i of the mesh (4.1.2), for $i = l, \ldots, n-k+l$, with l number of initial conditions. The approximation in the initial points x_i, for $i = 1, \ldots, l-1$ is obtained using initial formulae once, while final schemes are applied once in the points x_i , for $i = n-k+l+1, \ldots, n-1$. Then the

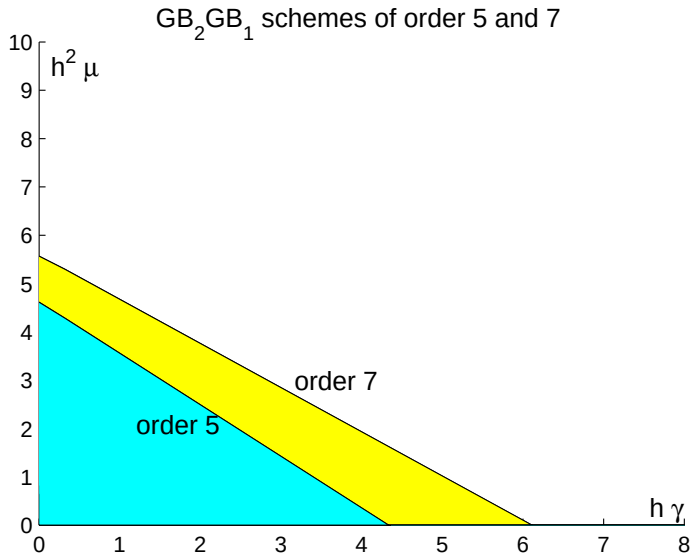


Figure 4.2: *Stability regions for GB_2GB_1 scheme of order 5 and 7.*

discrete problem of (4.1.1) can be written as

$$f(x_i, y_i, y_i', y_i'') = 0, \quad i = 1, \ldots, n-1. \tag{4.3.1}$$

We point out that $y_0 = y(x_0)$ is known, so we have all together l initial conditions, moreover (4.3.1) defines a nonlinear system of $(n-1)$ equations in n unknowns $Y = [y_1, \ldots, y_n]^T$. We need for another condition appending to (4.3.1) to find the solution $\tilde{Y} = [y_0, Y^T]^T$. Since the initial condition on $y'(x_0)$ in (4.1.1) has been not used, we use it to define two strategies and complete the system.

The first one considers the formula with zero initial conditions, obtained in (4.1.3) setting $l = 0$ and $\nu = 1$, to approximate the $y'(x_0)$ by means of an additional equation, obtaining in this way the system

$$\begin{cases} y_0 \text{ given}, \\ \dfrac{1}{h}\displaystyle\sum_{j=0}^{k} \alpha_j^{(0,1)} y_j = y_0', \\ f(x_i, y_i, y_i', y_i'') = 0, \qquad \text{for } i = 1, \ldots, n-1. \end{cases} \tag{4.3.2}$$

which provides a unique solution $\tilde{Y} = (y_0, y_1, \ldots, y_n)^T$ of the discrete problem.

The second strategy considers $y'_0 = y'(x_0)$ in Y and defines new initial formulae, using y'_0, as

$$y^{(\nu)}(x_i) \approx \frac{1}{h^\nu} \left(\bar{\alpha}_0^{(\nu,i)} h y'_0 + \sum_{j=1}^{k} \alpha_j^{(\nu,i)} y_{j-1} \right), \qquad \nu = 1, 2, \tag{4.3.3}$$

where the coefficients $(\bar{\alpha}_0^{(\nu,i)}, \alpha_0^{(\nu,i)}, \ldots, \alpha_k^{(\nu,i)})$ are computed in order to reach the maximum order, for $i = 0, \ldots, l-1$ with l number of initial conditions. A global approximation of the initial value problem (4.1.1) is obtained applying a main method, chosen in (4.1.3), with l initial and $k - l$ final conditions at all possible internal points of the grid (4.1.2), this is for $i = l, \ldots, n-k+l-1$ and using once at the initial extreme points the l formulae (4.3.3). Indeed, $k-l-1$ formulae (4.1.3) are employed once in the extreme final points, this is for $i = n-k+l, \ldots, n-1$. Consequently, an extended vector of the solution $Y = (y'_0, y_0, y_1, \ldots, y_n)^T$ is computed by solving the system

$$\begin{cases} y_0, y'_0 \text{ given}, \\ f(x_i, y_i, y'_i, y''_i) = 0, \qquad \text{for } i = 0, \ldots, n-1 \end{cases} \tag{4.3.4}$$

Both the approaches can be applied to the whole interval $[a, b]$ or to the first subinterval $\mathcal{I}_1$. In this second case we could pass on the last two values of the solution in $\mathcal{I}_1$ to $\mathcal{I}_2$ where they could be used as the known initial values of the new interval $\mathcal{I}_2$ and formulae (4.3.1) would compute $Y = (y_{-1}, y_0, y_1, \ldots, y_n)^T$ uniquely by means of

$$\begin{cases} y_0, y_{-1} \text{ given}, \\ f(x_i, y_i, y'_i, y''_i) = 0, \qquad \text{for } i = 0, \ldots, n-1. \end{cases} \tag{4.3.5}$$

Since partitioning the interval $[a, b]$ we may use different stepsize in each subinterval, then if the stepsize in $\mathcal{I}_2$ is changed, the initial value y_{-1} is computed by means of interpolation techniques from the points in the previous interval. In this case the interpolation accuracy could represent the drawback to apply this approach with variable stepsize. The idea of neglecting the value of the derivative in the last point of the interval after the interval $\mathcal{I}_1$ seems to be more natural, but this approach requires interpolation formulae that could be ill-conditioned if the order is high, since the used stepsize inside each interval is constant. Probably, a variable stepsize inside each interval would improve the situation, but we shall not consider this issue here.

Another alternative is to define a formula analogous to (4.3.2) in order to compute the approximation of $y'(x_n)$ as follows

$$\frac{1}{h}\sum_{j=-k}^{0}\alpha_j^{(n,1)}y_{n+j}=y'_n. \tag{4.3.6}$$

Then, the values of y_n and y'_n as initial condition and the same formulae (4.3.2) could be used also in the next.

Moreover, symmetry reasons suggest to consider as unknown y'_n and to define an approach similar to (4.3.3), this means that we compute the final formulae by means of

$$y^{(\nu)}(x_{n-i})\approx\frac{(-1)^\nu}{h^\nu}\left(-\bar{\alpha}_0^{(i,\nu)}hy'_n+\sum_{j=1}^{k}\alpha_j^{(i,\nu)}y_{n-j+1}\right),\qquad \nu=1,2, \tag{4.3.7}$$

where the coefficients $(\alpha_1^{(i,\nu)},\ldots,\alpha_k^{(i,\nu)},\bar{\alpha}_0^{(i,\nu)})$ for $i=0,\ldots,k-l-1$ are obtained imposing the maximum order. We observe that the coefficients in (4.3.7) for $y^{(\nu)}(x_{n-i})$ are just the same in reverse order of those in (4.3.3) for $y^{(\nu)}(x_i)$, and also changed sign for $\nu=2$. Considering as initial formulae (4.3.3) and final ones (4.3.7), set $Y=(y'_0,y_0,\ \ y_1,\ldots,y_n,y'_n)^T$, the solution vector is obtained solving the non linear system

$$\begin{cases} y_0,y'_0 \text{ given},\\ f(x_i,y_i,y'_i,y''_i)=0, \qquad \text{for } i=0,\ldots,n \end{cases} \tag{4.3.8}$$

From a numerical point of view, the last formula, even if it is described in compact form and it could be applied to the next intervals, contains values of y and y' that could be different in magnitude. Anyway, the numerical tests show that this approach gives the most accurate results.

Example 4.3.1. We consider in the following the initial formulae (4.3.3) of order 5 and 6. We leave out the final formulae (4.3.7) involving y'_n since the coefficients are, in the reverse order, the same of the initial formulae and with changed sign for the second derivative.

Order 5

$$h^2\ y''(x_0) \approx -\frac{137}{30}hy'_0 - \frac{12019}{1800}y_0 + 10y_1 - 5y_2 + \frac{20}{9}y_3 - \frac{5}{8}y_4 + \frac{2}{25}y_5$$
$$h^2\ y''(x_1) \approx \frac{13}{30}hy'_0 + \frac{3281}{1800}y_0 - \frac{41}{12}y_1 + \frac{11}{6}y_2 - \frac{5}{18}y_3 + \frac{1}{24}y_4 - \frac{1}{300}y_5$$
$$h\ y'(x_1) \approx -\frac{1}{4}hy'_0 - \frac{37}{48}y_0 + \frac{1}{6}y_1 + \frac{3}{4}y_2 - \frac{1}{6}y_3 + \frac{1}{48}y_4$$

Order 6

$$h^2\ y''(x_0) \approx -\frac{49}{10}hy'_0 - \frac{13489}{1800}y_0 + 12y_1 - \frac{15}{2}y_2 + \frac{40}{9}y_3 - \frac{15}{8}y_4 + \frac{12}{25}y_5 - \frac{1}{18}y_6$$
$$h^2\ y''(x_1) \approx \frac{77}{180}hy'_0 + \frac{2171}{1200}y_0 - \frac{203}{60}y_1 + \frac{43}{24}y_2 - \frac{13}{54}y_3 + \frac{1}{48}y_4 + \frac{1}{300}y_5 - \frac{1}{1080}y_6$$
$$h\ y'(x_1) \approx -\frac{1}{5}hy'_0 - \frac{197}{300}y_0 - \frac{1}{12}y_1 + y_2 - \frac{1}{3}y_3 + \frac{1}{12}y_4 - \frac{1}{100}y_5$$

The three approaches allow us to define an equivalent number of methods.

Definition 4.3.2. Given an initial value problem (4.1.1) and considered a subdivision $\mathfrak{I}$ of the interval $[a, b]$, the method using in each interval of the subdivision the approach (4.3.2) in combination with the formula (4.3.6) is named **D1HOGD**.

Remark 4.3.3. We point out that the D1HOGD method computes the solution applying the approach (4.3.2), after that the formula (4.3.6) is used to calculate an approximation of $y'(x_n)$. However it is preferable to approximate $y'(x_{n-1})$ by means of the formula

$$\frac{1}{h}\sum_{j=-k}^{0} \alpha_j^{(n-1,1)} y_{n+j} = y'_{n-1},$$

since the error constant for this formula is much lower than the analogous at x_n, for this reason the approximation y_n at x_n is discarded. Consequently, the successive interval has as left extreme point just x_n.

Definition 4.3.4. Given an initial value problem (4.1.1) and considered a subdivision $\mathfrak{I}$ of the interval $[a, b]$, the method using in the first interval of the subdivision or on the whole interval $[a, b]$ the approach defined by (4.3.3) and (4.3.5) and keeping in the next interval applying the approach (4.3.2) is named **ED1HOGD**.

Definition 4.3.5. Given an initial value problem (4.1.1) and considered a subdivision $\mathfrak{I}$ of the interval $[a, b]$, the method using in each interval of the subdivision the approach (4.3.8) in combination with the formulae (4.3.3)-(4.3.7) is named **EDHOGD**.

4.4 Mesh Selection Strategy

We consider the initial value problem (4.1.1) and an initial mesh π defined as in (4.1.2) covering all the interval or the subinterval $\mathcal{I}_k$ for $k = 1, \ldots, N$. The strategy uses constant stepsize inside each interval $\mathcal{I}_k$ but could change stepsize from $\mathcal{I}_k$ to $\mathcal{I}_{k+1}$.

If y_i and $\hat{y}_i$ are, respectively, the approximation and exact solution of (4.1.1) at the ith point of the kth interval, we can define the pointwise error in $\mathcal{I}_k$, for $i = 0, \ldots, n$, as

$$e_i = |\hat{y}_i - y_i|, \tag{4.4.1}$$

where $e_0 = 0$ for the initial conditions in (4.1.1).

We are interested in applying the methods of order p in Section 4.1; therefore, (4.4.1) can be expressed as

$$e_i = |\hat{y}_i - y_i| = Ch^p |\Psi(x_i)| + O(h^{p+1}), \quad i = 0, \ldots, n, \tag{4.4.2}$$

where $\Psi(x)$ is a function depending on the sth derivatives of $y(x)$, $s \geq p+1$ and h is the constant stepsize associated to the mesh π. Moreover, the maximum norm of (4.4.2) is defined as

$$\|\mathbf{e}\|_\infty = \max_{i=0,\ldots,n} |e_i| \leq \tilde{C}h^p. \tag{4.4.3}$$

Now, the intent is to choose a finer mesh in order to compute an accurate solution satisfying the condition

$$\|\mathbf{e}\|_\infty \leq TOL, \tag{4.4.4}$$

for a given relative tolerance TOL. The relations (4.4.3)-(4.4.4) allow to compute the new mesh π^* choosing as a stepsize

$$h^* = \left(\frac{\alpha \cdot TOL}{\|\mathbf{e}\|_\infty} \right)^{1/p} h, \tag{4.4.5}$$

where $\alpha = 0.9$.

We emphasize as this stepsize selection strategy guarantees to gain the desired accuracy of the solution on every subinterval or the whole interval. In the case we work on a part of all interval $\mathcal{I}_1$, after we have computed the solution on it, we proceed in approximating the solution in the next interval. Then, we have two possibility either to use the same stepsize of the interval $\mathcal{I}_1$ or a different one. This second choice consists in the stepsize estimation for the interval $\mathcal{I}_2$ by means of (4.4.5) when an accurate solution of the previous interval have been computed. Moreover, some restrictions are imposed to this estimation in order that the new stepsize is at most doubled or halved with respect to the stepsize of the previous interval. The strategy can be also summarized in the following algorithm.

Algorithm 3. *Input*: Given an IVP (4.1.1), a mesh π and the stepsize h on the interval $\mathcal{I}_1$;

1. $k = 1$;
2. while $b_k < b$ where $\mathcal{I}_k = [a_k, b_k]$;
3. compute the solution y of (4.1.1) using the basic schemes in section 4.1 and an estimate err of the error;
4. while $err > TOL$
5. $h^* = \left(\frac{\alpha \cdot TOL}{err}\right)^{1/p} h$;
6. compute the new mesh π^* of equidistant points with stepsize h^*;
7. compute y and err as in 3.;
8. end

   Output: y and err on $\mathcal{I}_k$.
9. $k = k + 1$;
10. update $h = \left(\frac{\alpha \cdot TOL}{err}\right)^{1/p} h^*$;
11. compute $h = \min\{\max(h, h^*), 2h^*\}$;
12. compute the mesh π on $\mathcal{I}_k$ using the stepsize h as defined in 11;
13. end

4.5 Numerical Tests

In this section we show some numerical tests in order to compare the methods D1HOGD, ED1HOGD and EDHOGD of order p introduced in Definition 4.3.2-4.3.4-4.3.5. We point out that D1HOGD method computes an approximation for $y'(x_{n-1})$ as specified in Remark 4.3.3 and ED1HOGD method uses the same constant stepsize in each interval. Moreover, in the numerical experiments we have used $p+4$ equidistant points in each interval covering the whole interval, where the order p ranges from 3 to 10.

For the first three examples we have firstly considered a constant stepsize implementation in order to estimate the order of convergence and to compare the methods. Then, we have solved each problem by means of a simple variable stepsize strategy with initial stepsize $h_0 = 8 \cdot 10^{-2}$ and exit tolerance $tol = 10^{-8}$.

Example 4.5.1. The first linear problem,

$$y''(x) + y'(x) = 0, \qquad x \in [0, 40],$$

has been solved with initial conditions $y(0) = 1$ and $y'(0) = -1$ or $y(0) = 2$ and $y'(0) = -1$. The roots of $z^2 + z = 0$ are -1 and 0 and, therefore, the exact solution is

$$y_e(x) = \mathrm{e}^{-x} + c_2,$$

where $c_2 = y(0) - 1$. Even if the numerical solution is monotone decreasing, it might tend to a negative value when $c_2 = 0$, this means to consider $y(0) = 1$. For this reason, in Table 4.3 we have also indicated when the numerical solution eventually becomes negative. With constant stepsize we have not observed differences between the two problems, for this reason we have discarded the table associated to the second choice of initial conditions. Vice versa, with variable stepsize, in Table 4.4 and Table 4.5 it is possible to observe that by using variable stepsize EDHOGD method requires a lower number of points than D1HOGD, moreover this number becomes much lower for the second choice of initial conditions.

Example 4.5.2. The second linear problem,

$$y''(x) - \cos x \; y'(x) + \sin x \; y(x) = 0, \quad x \in [0, 6\pi],$$

has initial conditions $y(0) = 1$ and $y'(0) = 1$. The exact solution,

$$y_e(x) = \mathrm{e}^{\sin x},$$

Table 4.3: Numerical results for Example 4.5.1 with $y(0) = 1$, constant stepsize.

	Main Scheme	Order	D1HOGD		ED1HOGD		EDHOGD	
			Error	$x : y(x) < 0$	Error	$x : y(x) < 0$	Error	$x : y(x) < 0$
$h = 8 \cdot 10^{-2}$, 500 points	GF_2GF_1	3	4.65e-05		3.90e-06	12.96	3.40e-06	
	GB_2GB_1	3	4.49e-05		8.40e-06	11.76	6.21e-06	12.00
	EC_2EC_1	4	3.31e-05	10.32	1.89e-05	10.88	2.88e-07	16.24
	GF_2GF_1	5	2.50e-08		6.70e-08	16.56	9.86e-09	18.48
	GB_2GB_1	5	6.65e-08		2.35e-08	17.60	1.65e-08	
	EC_2EC_1	6	1.44e-07	15.76	9.03e-08	16.24	3.57e-09	
	GF_2GF_1	7	1.23e-10	23.92	2.68e-10	22.08	1.88e-11	
	GB_2GB_1	7	1.23e-10	23.92	2.53e-10	22.16	1.45e-11	24.96
	EC_2EC_1	8	7.05e-10	21.12	4.80e-10	21.52	9.04e-12	
	GF_2GF_1	9	1.15e-12	27.76	1.59e-12	27.20	2.00e-14	32.56
	GB_2GB_1	9	6.31e-13	28.40	1.37e-12	27.36	5.28e-14	
	EC_2EC_1	10	3.57e-12	26.40	2.60e-12	26.72	3.36e-14	
$h = 4 \cdot 10^{-2}$, 1000 points	GF_2GF_1	3	1.54e-05		1.73e-07		5.13e-07	
	GB_2GB_1	3	1.51e-05		8.06e-07	14.04	6.89e-07	14.20
	EC_2EC_1	4	3.94e-06	12.48	1.72e-06	13.28	1.78e-08	18.96
	GF_2GF_1	5	5.36e-09		1.57e-09	20.28	3.64e-10	21.76
	GB_2GB_1	5	6.90e-09		3.31e-10	21.96	4.69e-10	
	EC_2EC_1	6	4.36e-09	19.28	2.06e-09	20.04	5.65e-11	
	GF_2GF_1	7	2.82e-12		1.40e-12	27.32	1.62e-13	
	GB_2GB_1	7	3.02e-12		1.43e-12	27.28	8.20e-14	30.16
	EC_2EC_1	8	5.24e-12	26.00	2.71e-12	26.64	6.20e-14	
	GF_2GF_1	9	1.90e-13		6.79e-14		7.33e-15	
	GB_2GB_1	9	4.30e-13	28.60	3.75e-13		1.48e-14	
	EC_2EC_1	10	2.30e-13		1.17e-13	30.20	5.53e-14	30.56

Table 4.4: Numerical results for Example 4.5.1 with $y(0) = 1$, variable stepsize.

Main Scheme	Order	D1HOGD		EDHOGD	
		Error	Mesh	Error	Mesh
GF_2GF_1	3	5.35e-06	751	3.03e-07	593
GB_2GB_1	3	3.92e-06	898	1.88e-07	817
EC_2EC_1	4	4.91e-07	1038	8.10e-09	809
GF_2GF_1	5	1.69e-08	407	1.61e-08	291
GB_2GB_1	5	2.94e-08	407	1.49e-08	331
EC_2EC_1	6	3.36e-08	434	4.40e-09	321
GF_2GF_1	7	1.11e-09	288	4.93e-10	217
GB_2GB_1	7	7.49e-10	288	1.54e-09	193
EC_2EC_1	8	3.20e-09	288	5.48e-10	217
GF_2GF_1	9	3.45e-10	223	5.39e-11	183
GB_2GB_1	9	2.74e-10	223	7.83e-11	183
EC_2EC_1	10	5.88e-10	223	8.99e-11	183

Table 4.5: Numerical results for Example 4.5.1 with $y(0) = 2$, variable stepsize.

Main Scheme	Order	D1HOGD		EDHOGD	
		Error	Mesh	Error	Mesh
GF_2GF_1	3	9.36e-06	240	7.84e-07	169
GB_2GB_1	3	6.89e-06	275	5.62e-07	209
EC_2EC_1	4	1.38e-06	268	1.75e-08	177
GF_2GF_1	5	1.25e-08	137	3.93e-08	101
GB_2GB_1	5	5.35e-08	137	4.34e-08	121
EC_2EC_1	6	1.20e-07	146	1.64e-08	111
GF_2GF_1	7	1.04e-08	112	4.31e-09	97
GB_2GB_1	7	7.13e-09	112	6.83e-09	85
EC_2EC_1	8	1.47e-08	112	2.74e-09	97
GF_2GF_1	9	1.82e-09	106	1.66e-09	85
GB_2GB_1	9	1.38e-09	106	3.18e-09	85
EC_2EC_1	10	2.23e-09	106	2.32e-09	85

has an oscillating solution with period 2π. Comparing the results obtained with constant stepsize in Table 4.6 with those using variable stepsize in Table 4.7 we can confirm that for oscillating solution it is suitable to choice equidistant stepsize for a good and cheap accuracy.

Example 4.5.3. The nonlinear problem

$$(y(x) + 1)\ y''(x) - 3(y'(x))^2 = 0, \qquad x \in [1, 10],$$

has initial conditions $y(1) = 0$ and $y'(1) = -\frac{1}{2}$. The exact solution is

$$y_e(x) = \frac{1}{\sqrt{x}} - 1.$$

In table Table 4.8 we show the results obtained with constant stepsize, but looking at Table 4.9 it seems clear that the variable stepsize allows to gain the desired accuracy with much less points and the EDHOGD method seems more suitable than D1HOGD method.

Example 4.5.4. As last example we consider an IVP which is named *Flow in concrete* problem, see [12], and defined as

$$yy'' = -xy', \quad x \in [0, 10]$$

with initial conditions $y(0) = 1$ and $y'(0) = -\gamma$, where $\gamma > 2$. The solution of the problem is obtained applying D1HOGD method with variable stepsize.

Table 4.6: Numerical results for Example 4.5.2, constant stepsize.

	Main Scheme	Order	Error D1HOGD	Error ED1HOGD	Error EDHOGD
$h = 8 \cdot 10^{-2}$, 236 points	GF_2GF_1	3	7.63e-03	1.73e-03	2.99e-03
	GB_2GB_1	3	7.31e-04	6.64e-03	3.30e-03
	EC_2EC_1	4	2.46e-03	3.83e-03	6.58e-05
	GF_2GF_1	5	1.34e-04	3.05e-04	3.27e-06
	GB_2GB_1	5	1.12e-04	2.77e-04	1.38e-05
	EC_2EC_1	6	7.66e-05	3.35e-04	8.27e-06
	GF_2GF_1	7	5.08e-06	2.50e-05	9.96e-07
	GB_2GB_1	7	5.04e-06	2.52e-05	9.32e-07
	EC_2EC_1	8	5.15e-06	2.59e-05	1.19e-06
	GF_2GF_1	9	1.57e-05	1.17e-06	3.03e-07
	GB_2GB_1	9	1.55e-05	1.23e-06	3.28e-07
	EC_2EC_1	10	1.62e-05	9.85e-07	3.28e-07
$h = 4 \cdot 10^{-2}$, 471 points	GF_2GF_1	3	9.63e-04	4.06e-04	3.78e-04
	GB_2GB_1	3	1.32e-04	6.11e-04	4.00e-04
	EC_2EC_1	4	4.03e-04	1.82e-04	4.42e-06
	GF_2GF_1	5	4.86e-06	4.11e-06	1.20e-07
	GB_2GB_1	5	4.12e-06	3.42e-06	3.39e-07
	EC_2EC_1	6	1.34e-06	4.30e-06	1.10e-07
	GF_2GF_1	7	7.10e-08	5.31e-08	3.88e-10
	GB_2GB_1	7	6.95e-08	5.43e-08	5.25e-10
	EC_2EC_1	8	5.18e-08	5.15e-08	1.67e-10
	GF_2GF_1	9	8.22e-10	9.95e-11	7.58e-12
	GB_2GB_1	9	6.06e-10	6.65e-11	2.09e-11
	EC_2EC_1	10	6.62e-10	4.85e-10	1.10e-11

Table 4.7: Numerical results for Example 4.5.2, variable stepsize.

Main Scheme	Order	D1HOGD Error	D1HOGD Mesh	EDHOGD Error	EDHOGD Mesh
GF_2GF_1	3	9.09e-05	1311	2.88e-05	1113
GB_2GB_1	3	1.70e-04	1206	2.88e-05	1113
EC_2EC_1	4	1.98e-05	1199	2.94e-06	761
GF_2GF_1	5	1.52e-06	542	4.27e-07	421
GB_2GB_1	5	1.29e-06	560	2.52e-06	351
EC_2EC_1	6	8.14e-07	542	5.56e-07	371
GF_2GF_1	7	1.08e-06	343	1.10e-07	253
GB_2GB_1	7	1.04e-06	343	9.34e-08	265
EC_2EC_1	8	7.58e-06	343	1.80e-07	253
GF_2GF_1	9	1.45e-07	275	1.22e-06	197
GB_2GB_1	9	1.56e-07	275	2.16e-06	197
EC_2EC_1	10	3.42e-07	275	1.09e-06	197

Table 4.8: Numerical results for Example 4.5.3, constant stepsize.

	Main Scheme	Order	Error D1HOGD	Error ED1HOGD	Error EDHOGD
$h = 8 \cdot 10^{-2}$, 125 points	GF_2GF_1	3	5.20e-05	6.97e-05	1.42e-05
	GB_2GB_1	3	3.23e-05	3.63e-05	6.13e-06
	EC_2EC_1	4	1.10e-04	9.47e-05	3.71e-06
	GF_2GF_1	5	7.00e-06	7.29e-06	7.35e-07
	GB_2GB_1	5	6.50e-06	6.83e-06	6.34e-07
	EC_2EC_1	6	9.29e-06	8.91e-06	7.17e-07
	GF_2GF_1	7	1.16e-06	1.17e-06	5.32e-08
	GB_2GB_1	7	1.11e-06	1.12e-06	5.57e-08
	EC_2EC_1	8	1.36e-06	1.35e-06	6.95e-08
	GF_2GF_1	9	2.39e-07	2.40e-07	1.11e-08
	GB_2GB_1	9	2.34e-07	2.35e-07	1.12e-08
	EC_2EC_1	10	2.68e-07	2.68e-07	1.24e-08
$h = 4 \cdot 10^{-2}$, 250 points	GF_2GF_1	3	4.54e-06	7.11e-06	1.75e-06
	GB_2GB_1	3	3.23e-05	2.57e-06	1.23e-06
	EC_2EC_1	4	1.31e-05	9.19e-06	2.51e-07
	GF_2GF_1	5	1.92e-07	2.27e-07	1.80e-08
	GB_2GB_1	5	1.77e-07	2.15e-07	1.20e-08
	EC_2EC_1	6	3.31e-07	2.85e-07	1.58e-08
	GF_2GF_1	7	1.31e-08	1.36e-08	4.06e-10
	GB_2GB_1	7	1.25e-08	1.30e-08	4.62e-10
	EC_2EC_1	8	1.68e-08	1.59e-08	5.54e-10
	GF_2GF_1	9	1.10e-09	1.12e-09	3.52e-11
	GB_2GB_1	9	1.07e-09	1.09e-09	3.53e-11
	EC_2EC_1	10	1.29e-09	1.26e-09	3.96e-11

Table 4.9: Numerical results for Example 4.5.3, variable stepsize.

		D1HOGD		EDHOGD	
Method	Order	Error	Mesh	Error	Mesh
GF_2GF_1	3	2.57e-06	233	2.25e-07	193
GB_2GB_1	3	2.68e-06	247	3.58e-07	153
EC_2EC_1	4	7.42e-07	240	1.72e-08	185
GF_2GF_1	5	2.65e-08	119	7.94e-09	101
GB_2GB_1	5	2.00e-08	119	5.62e-09	101
EC_2EC_1	6	6.55e-08	128	6.15e-09	101
GF_2GF_1	7	1.60e-08	90	1.74e-09	73
GB_2GB_1	7	1.43e-08	90	2.07e-09	73
EC_2EC_1	8	1.84e-08	101	2.13e-09	85
GF_2GF_1	9	1.18e-08	80	2.67e-09	71
GB_2GB_1	9	1.10e-08	80	2.77e-09	71
EC_2EC_1	10	1.32e-08	80	2.72e-09	71

Despite the previous examples highlight the better use of EDHOGD with variable stepsize, in this case since the behavior of the first derivative may be quite different from the solution one and affect the stepsize variation, the choice of the first method is preferable. In [12] the problem is solved for different values of γ from 2 to 18, since asymptotic theoretical results are confirmed numerically when γ increases. We are interested to evaluate the solution at the infinity, for this reason we compute the solution in [12] also on larger intervals than $[0, 10]$, and we can uphold that $y(10) \approx y(\infty)$. In example for $\gamma = 10$, it is $y(10) \approx 2.254440321030590e\text{-}044$, while for $\gamma = 18$ we have $y(10) \approx 1.178498689884995e\text{-}141$. In Figure 4.3-Figure 4.4 and Figure 4.5-Figure 4.6 we draw the solutions and the stepsize variations for $\gamma = 10$ and $\gamma = 18$, respectively. Very interesting is the stepsize selection, since it shows as the method works also when the stepsize becomes very small.

4.6 Conclusion

The methods introduced in Section 4.1 have the advantage to be applied to a second order IVP (4.1.1) without requiring the transformation in an equivalent system of the first order. Second and first derivatives in ODE are approximated separately and the strategy variation turns out to be very simple. The numerical results show as the methods are not more competitive than the others for IVP, but the Example 4.5.4 shows that the class of the considered methods can be very efficient when we consider problems with decreasing positive solutions which have a fast variation going to a constant value in a narrow region of the interval.

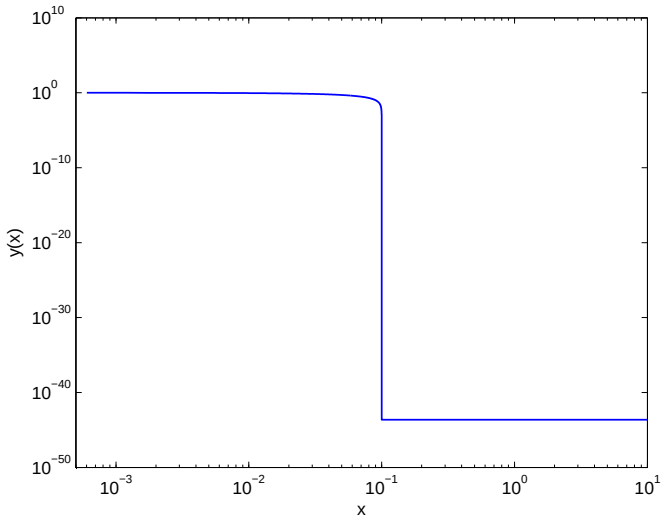


Figure 4.3: Solution for $\gamma = 10$ in logarithmic scale.

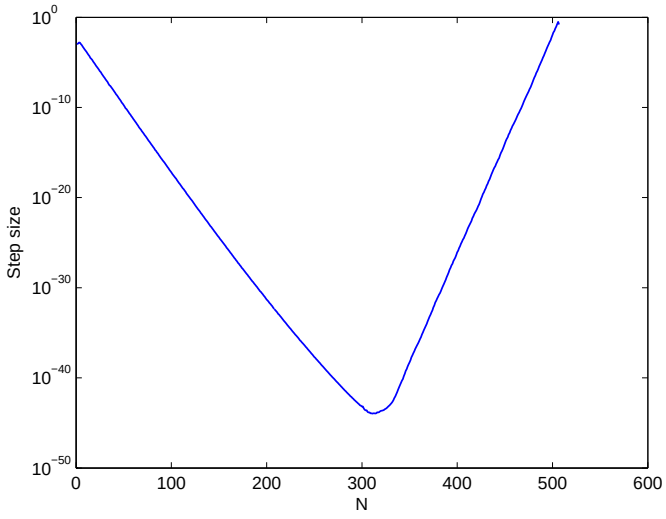


Figure 4.4: Stepsize Variation, $\gamma = 10$ and $p = 8$.

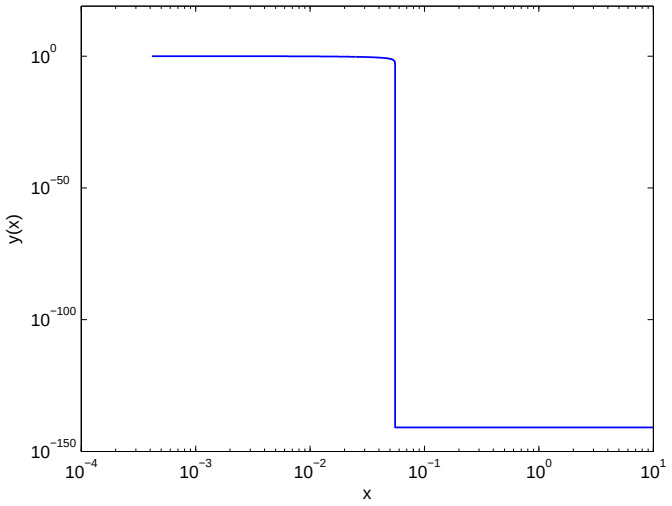


Figure 4.5: Solution for $\gamma = 18$ in logarithmic scale.

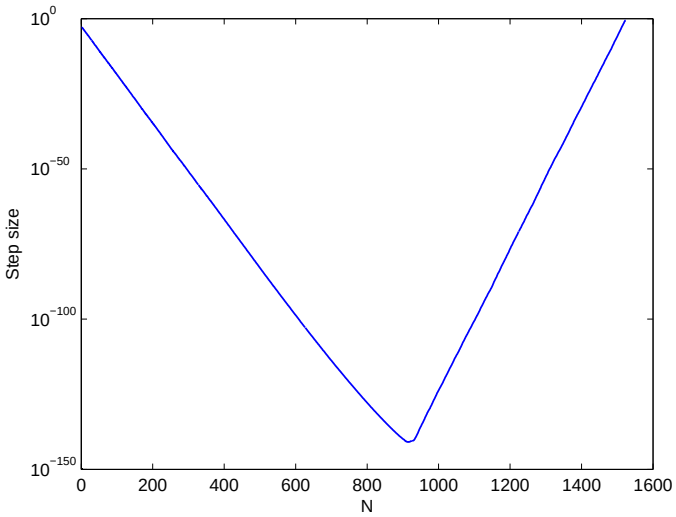


Figure 4.6: Stepsize variation, $\gamma = 18$ and $p = 8$.

Chapter 5

Sturm-Liouville Problems

Many phenomena in engineering and physics related to signal/image processing, applications in photonics, atom optics and telecommunication, quantum mechanics and electrodynamics are described by Sturm-Liouville problems and their solution assumes a great interest in mathematics and mathematical physics, so that different codes able to solve regular and singular Sturm-Liouville problems have been developed over the years. The fortran code SLEIGN introduced in [23] computes the eigenvalues and the eigenfunctions of a regular and singular Sturm-Liouville problem, and in the singular case the code automatically select a boundary condition. The code is based on the Prüfer transformation and on the knowledge of the number of zeros of the eigenfunctions. After SLEIGN, a new fortran code SLEIGN2 [20, 21, 22, 23] has been developed to compute only the eigenvalues of Sturm/Liouville problems, moreover on the contrary at the first code it requires to distinguish between regular and singular endpoints and in the singular case it is necessary to describe the boundary condition appropriately. The fortran code SLEDGE [55] computes eigenvalues and eigenfunctions of regular and singular SL problems is based on a step function approximation which use a midpoint interpolation. Another fortran code SL02FM [56] is based on coefficient approximation for the automatic solution of regular and singular Sturm/Liouville problems. An alternative matlab code is MATSLISE [47], based on CP methods [44], which solves regular Schrödinger equation, regular Sturm-Liouville problems and Schrödinger equation with distorted Coulomb potentials. Moreover, largely widespread are the application of matrix methods, which consist to apply finite differences to reduce the Sturm-Liouville problem to a matrix eigenvalue problem. For the last approach correction techniques have been

developed to improve the accuracy of the computed eigenvalues. In the last years competitive results, in comparison with corrected Numerov's method, have been reached by applying symmetric BVMs [2, 3].

In this chapter, following the idea of the matrix methods for SLPs, we discretize the continuous problem by using ECDF as main schemes [9, 10], in order to obtain an equivalent algebraic eigenvalue problem. Regular and singular SLPs are solved by the same main schemes, but with different additional conditions which depend on the regularity or singularity of the endpoints. A stepsize variation strategy, based on the equidistribution of the error, is applied in order to gain a better accuracy on the eigenfunctions estimate [11]. In the last section known Sturm-Liouville problems are solved on equidistant and variable mesh.

5.1 High order finite difference schemes

We consider a general Sturm-Liouville equation

$$-(p(x)y')' + q(x)y = \lambda r(x)y, \qquad x \in (a,b), \quad y, \lambda \in \mathbb{R}. \tag{5.1.1}$$

where $-\infty \leq a < b \leq \infty$. We assume that the set of Sturm-Liouville coefficients $p, q, r : I \to \mathbb{R}$ satisfy $p^{-1}, q, r \in L_{loc}(I)$, $p > 0$ and $r > 0$ in (a,b). The equation (5.1.1) is subjected to separated boundary conditions

$$\begin{array}{ll} a_1[y,u](a) + a_2[y,v](a) = 0, & (a_1,a_2) \neq (0,0), \\ b_1[y,u](b) + b_2[y,v](b) = 0, & (b_1,b_2) \neq (0,0), \end{array} \tag{5.1.2}$$

where $[f,g]$, see Section 1.4, is defined for two admissible functions as $[f,g](x) = f(x)(pg')(x) - (pf')(x)g(x)$, and u and v are two linearly independent solutions of the Sturm-Liouville equation for some arbitrary λ, $[u,v](a) \neq 0$ and $[u,v](b) \neq 0$.

From Section 1.4 we say that λ is the eigenvalue, whereas y is the eigenfunction associated with λ. We consider regular and singular Sturm-Liouville problems satisfying the property

$$-\infty < \lambda_0 < \lambda_1 < \lambda_2 < \lambda_3 < \ldots \quad \text{with} \quad \lim_{n\to\infty} \lambda_n = \infty; \tag{5.1.3}$$

this means that the spectrum is discrete and bounded below and only singular problems with LCNO and/or LP endpoints are considered, see Section

1.4. In addition, from Theorem 1.4.4 the associated eigenfunctions $Y_n(x)$ are orthogonal to each other with respect to the *weight function* $r(x)$,

$$\int_a^b r(x)Y_m(x)Y_n(x)dx = 0 \qquad \text{if } m \neq n, \text{ and } m, n = 1, 2, \ldots.$$

As suggested in Section 1.4, the eigenfunctions are uniquely computed according to a specified normalization function.

Furthermore, we remind that for regular Sturm-Liouville problems (5.1.1) the regular associated boundary conditions are defined as

$$\begin{array}{ll} a_1 y(a) + a_2(py')(a) = 0, & (a_1, a_2) \neq (0, 0), \\ b_1 y(b) + b_2(py')(b) = 0, & (b_1, b_2) \neq (0, 0). \end{array} \tag{5.1.4}$$

Matrix methods, which consist in applying finite differences to a Sturm-Liouville problem and reduce it to a matrix eigenvalue problem, are largely widespread [53]. However, for this approach correction techniques are required to improve the accuracy of the computed eigenvalues [16, 54]. In the last years competitive results, in comparison with corrected Numerov's method [16], have been reached by applying symmetric BVMs [2, 3]. Then, following the idea of the matrix method, we use the approach for BVPs in Chapter 2 to find the solution of a Sturm-Liouville problem (5.1.1)-(5.1.4). Therefore, we consider a uniform discretization of the interval $[a, b]$

$$a = x_0 < x_1 < \cdots < x_n = b, \tag{5.1.5}$$

where $x_i = x_0 + ih$, $i = 0, \ldots, n$, and $h = (b - a)/n$.

We approximate the second and the first derivatives in (5.1.1) by HOGD schemes, see Chapter 2. Thus, k-steps formulae are defined by

$$y^{(\nu)}(x_i) \approx y_i^{(\nu)} = \frac{1}{h^\nu} \sum_{j=-s}^{k-s} \alpha_{j+s}^{(s,\nu)} y_{i+j}, \qquad \nu = 1, 2, \tag{5.1.6}$$

see Proposition 2.1.2, where $s = 0, \ldots, k$ is the number of initial conditions. As usual, we choose the coefficients $(\alpha_0^{(s,\nu)}, \alpha_1^{(s,\nu)}, \ldots, \alpha_k^{(s,\nu)})$ in order that the formula (5.1.6) have maximum order. We use schemes of even order p, that is $k = p$ for the first derivative, while for the second derivative, $k = p$ for the scheme with symmetric stencil and $k = p + 1$ odd for the others. A main scheme is applied in all possible internal points of the interval, while additional formulae are used once in the extreme points.

For the second derivative just the schemes with symmetric stencil have the best stability properties, as shown in Section 2.3. For the first derivative, in Chapter 3 it is suggested to use schemes with $s = (k-2)/2$ or $s = (k+2)/2$ depending on the sign of the coefficient multiplying the derivative (HOGUP method). For Sturm-Liouville problems, symmetry reasons suggest to use ECDFs schemes with symmetric stencil, $s = k/2$. Then, we approximate the second and the first derivatives at the point x_i for $i = k/2, \dots, n-k$ by

$$y''(x_i) \approx y_i'' = \frac{1}{h^2} \sum_{j=-\frac{k}{2}}^{\frac{k}{2}} \alpha_{j+\frac{k}{2}}^{(k/2,2)} y_{i+j}, \qquad y'(x_i) \approx y_i' = \frac{1}{h} \sum_{j=-\frac{k}{2}}^{\frac{k}{2}} \alpha_{j+\frac{k}{2}}^{(k/2,1)} y_{i+j}. \tag{5.1.7}$$

Additional formulae approximate the solution at the remaining points at the beginning and at the end of the interval, but their choice depends on the problem regularity and singularity.

5.2 Additional Formulae

If the regular boundary conditions (5.1.4) are simply given by

$$y(a) = y(b) = 0, \tag{5.2.1}$$

then for the initial schemes the formulae (5.1.6), for $s = 1, \dots, k/2-1$, approximate the derivatives at the points x_s, while for the final conditions the discretization of the problem at the points x_{n-k+s} is obtained for $s = k/2+1, \dots, k-1$. Therefore, we discretize the BVP using D2ECDF method as in Chapter 2. In vector form if $Y = (y_1, \dots, y_{n-1})^T$ is the unknowns vector, $\tilde{Y} = (y_0, Y^T, y_n)^T$, and $\hat{A}_\nu$, for $\nu = 1, 2$, are the matrices for the global approximation of the derivatives, see Definition 2.1.6, then

$$Y^{(\nu)} = \tilde{A}_\nu \tilde{Y} \tag{5.2.2}$$

where $\tilde{A}_\nu = \hat{A}_\nu / h^\nu$, where $\tilde{A}_\nu$ is a $(n-1) \times (n+1)$ matrix.

Example 5.2.1. For the order $p = 4$

$$y'(x_i) \approx y_i' = \frac{1}{h} \begin{cases} -\frac{1}{4}y_0 - \frac{5}{6}y_1 + \frac{3}{2}y_2 - \frac{1}{2}y_3 + \frac{1}{12}y_4 \quad , & i = 1 \\ \frac{1}{12}y_{i-2} - \frac{2}{3}y_{i-1} + \frac{2}{3}y_{i+1} - \frac{1}{12}y_{i+2} \quad , & i = 2, \dots, n-2 \\ -\frac{1}{12}y_{n-4} + \frac{1}{2}y_{n-3} - \frac{3}{2}y_{n-2} + \frac{5}{6}y_{n-1} + \frac{1}{4}y_n \quad , & i = n-1 \end{cases}$$

and

$$y''(x_i) \approx y_i'' = \frac{1}{h^2} \begin{cases} \frac{5}{6}y_0 - \frac{5}{4}y_1 - \frac{1}{3}y_2 + \frac{7}{6}y_3 - \frac{1}{2}y_4 + \frac{1}{12}y_5 \quad , & i = 1 \\ -\frac{1}{12}y_{i-2} + \frac{4}{3}y_{i-1} - \frac{5}{2}y_i + \frac{4}{3}y_{i+1} - \frac{1}{12}y_{i+2} \quad , & i = 2, \ldots, n-2. \\ \frac{1}{12}y_{n-5} - \frac{1}{2}y_{n-4} + \frac{7}{6}y_{n-3} - \frac{1}{3}y_{n-2} - \frac{5}{4}y_{n-1} + \frac{5}{6}y_n \quad , & i = n-1 \end{cases}$$

Then the matrices associated to the approximation of the first and second derivatives are

$$\tilde{A}_1 = \frac{1}{h} \begin{pmatrix} -\frac{1}{4} & -\frac{5}{6} & \frac{3}{2} & -\frac{1}{2} & \frac{1}{12} & & & \\ \frac{1}{12} & -\frac{2}{3} & 0 & \frac{2}{3} & -\frac{1}{12} & & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & & \frac{1}{12} & -\frac{2}{3} & 0 & \frac{2}{3} & -\frac{1}{12} \\ & & & -\frac{1}{12} & \frac{1}{2} & -\frac{3}{2} & \frac{5}{6} & \frac{1}{4} \end{pmatrix}_{(n-1)\times(n+1)} . \quad (5.2.3)$$

and

$$\tilde{A}_2 = \frac{1}{h^2} \begin{pmatrix} \frac{5}{6} & -\frac{5}{4} & -\frac{1}{3} & \frac{7}{6} & -\frac{1}{2} & \frac{1}{12} & & \\ -\frac{1}{12} & \frac{4}{3} & -\frac{5}{2} & \frac{4}{3} & -\frac{1}{12} & & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & & -\frac{1}{12} & \frac{4}{3} & -\frac{5}{2} & \frac{4}{3} & -\frac{1}{12} \\ & & \frac{1}{12} & -\frac{1}{2} & \frac{7}{6} & -\frac{1}{3} & -\frac{5}{4} & \frac{5}{6} \end{pmatrix}_{(n-1)\times(n+1)} . \quad (5.2.4)$$

Remark 5.2.2. We point out that when the problem (5.1.1) is singular and one or both the endpoints are LP, see Section 1.4, then no boundary conditions are needed, see Section 1.4. In this case, we add to the system (5.2.2) one or two equations obtained by (5.1.6) for $s = 0$ if a is a LP point and for $s = k$ if b is a LP point. Then $\tilde{A}_\nu$ could be a $n \times (n+1)$ or $(n+1) \times (n+1)$ matrix, respectively.

Example 5.2.3. For order $p = 4$, the additional formulae to consider in the case a and/or b are LP are given by

$$\begin{aligned} y'(x_0) \approx y_0' &= \frac{1}{h} \quad -\frac{25}{12}y_0 + 4y_1 - 3y_2 + \frac{4}{3}y_3 - \frac{1}{4}y_4 \quad , \\ y'(x_n) \approx y_n' &= \frac{1}{h} \quad \frac{1}{4}y_{n-4} - \frac{4}{3}y_{n-3} + 3y_{n-2} - 4y_{n-1} + \frac{25}{12}y_n \quad , \end{aligned}$$

$$\begin{aligned} y''(x_0) \approx y_0'' &= \frac{1}{h^2} \quad \frac{15}{4}y_0 - \frac{77}{6}y_1 + \frac{107}{6}y_2 - 13y_3 + \frac{61}{12}y_4 - \frac{5}{6}y_5 \quad , \\ y''(x_n) \approx y_n'' &= \frac{1}{h^2} \quad -\frac{5}{6}y_{n-5} + \frac{61}{12}y_{n-4} - 13y_{n-3} + \frac{107}{6}y_{n-2} - \frac{77}{6}y_{n-1} + \frac{15}{4}y_n \quad . \end{aligned}$$

For the boundary conditions (5.1.2)-(5.1.4), since the first derivative in the extreme points is also involved, as in Section 4.3, the initial formulae consider the value of $y'(a) = y_0'$ and the approximations for $i = 0, \ldots, k/2-1$ are given by

$$y^{(\nu)}(x_i) \approx \frac{1}{h^\nu}\left(\bar{\alpha}_0^{(\nu,i)} h y_0' + \sum_{j=1}^{k} \alpha_j^{(\nu,i)} y_{j-1}\right), \qquad \nu = 1, 2. \tag{5.2.5}$$

On the other hand, final formulae, for $i = k/2+1, \ldots, k$, are also defined by means of the value of $y'(b) = y_n'$ as

$$y^{(\nu)}(x_{n+i-k}) \approx \frac{(-1)^\nu}{h^\nu}\left(-\bar{\alpha}_0^{(\nu,i)} h y_n' + \sum_{j=1}^{k} \alpha_j^{(\nu,i)} y_{n-j+1}\right), \qquad \nu = 1, 2. \tag{5.2.6}$$

In vector form if $\tilde{Y} = (y_0', y_0, y_1, \ldots, y_n, y_n')^T = (y_0', Y^T, y_n')^T$ and $\tilde{A}_\nu$, $\nu = 1, 2$ are the $(n+1) \times (n+2)$ matrices associated to the global approximation of the derivatives obtained with a combination of formulae (5.1.6)-(5.2.5)-(5.2.6), then we can write

$$Y^{(\nu)} = \tilde{A}_\nu \tilde{Y}. \tag{5.2.7}$$

Example 5.2.4. In this case for order $p = 4$ we obtain

$$y'(x_i) \approx y_i' = \frac{1}{h}\begin{cases} hy_0', & i = 0 \\ -\frac{h}{3}y_0' - \frac{17}{18}y_0 + \frac{1}{2}y_1 + \frac{1}{2}y_2 - \frac{1}{18}y_3 \quad , & i = 1 \\ \frac{1}{18}y_{n-3} - \frac{1}{2}y_{n-2} - \frac{1}{2}y_{n-1} + \frac{17}{18}y_n - \frac{h}{3}y_n' \quad , & i = n-1 \\ hy_n', & i = n \end{cases}$$

and

$$y''(x_i) \approx y_i'' = \frac{1}{h^2}\begin{cases} -\frac{25h}{6}y_0' - \frac{415}{72}y_0 + 8y_1 - 3y_2 + \frac{8}{9}y_3 - \frac{1}{8}y_4 \quad , & i = 0 \\ \frac{5h}{12}y_0' + \frac{257}{144}y_0 - \frac{10}{3}y_1 + \frac{7}{4}y_2 - \frac{2}{9}y_3 + \frac{1}{48}y_4 \quad , & i = 1 \\ \frac{1}{48}y_{n-4} - \frac{2}{9}y_{n-3} + \frac{7}{4}y_{n-2} - \frac{10}{3}y_{n-1} + \frac{257}{144}y_n - \frac{5h}{12}y_n' \quad , & i = n-1 \\ -\frac{1}{8}y_{n-4} + \frac{8}{9}y_{n-3} - 3y_{n-2} + 8y_{n-1} - \frac{415}{72}y_n + \frac{25h}{6}y_n' \quad , & i = n \end{cases}$$

Then the matrices are defined as

$$\tilde{A}_1 = \frac{1}{h}\begin{pmatrix} h & & & & & & & & & \\ -\frac{h}{3} & -\frac{17}{18} & \frac{1}{2} & \frac{1}{2} & -\frac{1}{18} & & & & & \\ & \frac{1}{12} & -\frac{2}{3} & 0 & \frac{2}{3} & -\frac{1}{12} & & & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & & & \\ & & & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ & & & & \frac{1}{12} & -\frac{2}{3} & 0 & \frac{2}{3} & -\frac{1}{12} & \\ & & & & & \frac{1}{18} & -\frac{1}{2} & -\frac{1}{2} & \frac{17}{18} & -\frac{h}{3} \\ & & & & & & & & & h \end{pmatrix}_{(n+1)\times(n+3)}, \tag{5.2.8}$$

$$\tilde{A}_2 = \frac{1}{h^2}\begin{pmatrix} -\frac{25h}{6} & -\frac{415}{72} & 8 & -3 & \frac{8}{9} & \frac{1}{8} & & & & \\ \frac{5h}{12} & \frac{257}{144} & -\frac{10}{3} & \frac{7}{4} & -\frac{2}{9} & \frac{1}{48} & & & & \\ & -\frac{1}{12} & \frac{4}{3} & -\frac{5}{2} & \frac{4}{3} & -\frac{1}{12} & & & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & & & \\ & & & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ & & & & -\frac{1}{12} & \frac{4}{3} & -\frac{5}{2} & \frac{4}{3} & -\frac{1}{12} & \\ & & & & \frac{1}{48} & -\frac{2}{9} & \frac{7}{4} & -\frac{10}{3} & \frac{257}{144} & -\frac{5h}{12} \\ & & & & -\frac{1}{8} & \frac{8}{9} & -3 & 8 & -\frac{415}{72} & \frac{25h}{6} \end{pmatrix}_{(n+1)\times(n+3)}. \tag{5.2.9}$$

Obviously, the absence of one of the first derivative term in the boundary conditions allows us to simplify initial and final schemes as said previously. We can distinguish two different cases.

(i) If boundary conditions involve $y'(a)$, then we consider $\tilde{Y} = [y'_0, y_0, y_1, y_2, \ldots, y_{n-1}]^T$. Formulae (5.2.5) are considered at the points x_i, $i = 0, \ldots, k/2 - 1$, while formulae (5.1.6) approximate the derivatives at the final points x_i, $i = n - k/2 + 1, \ldots, n - 1$,

$$\underbrace{y_0\, y_1\, \cdots y_{\frac{k}{2}-1}}_{\frac{k}{2}\text{ initial methods}}\, y_{\frac{k}{2}}\, \cdots y_{n-\frac{k}{2}}\, \underbrace{y_{n-\frac{k}{2}+1}\, y_{n-\frac{k}{2}+2} \cdots y_{n-1}}_{\frac{k}{2}-1\text{ final methods}}.$$

(ii) If boundary conditions involve $y'(b)$, then we have $\tilde{Y} = [y_0, y_1, \ldots, y_{n-1}, y_n, y'_n]^T$. Formulae (5.1.6) approximate the derivatives at the points x_i, $i = 1, \ldots, k/2 - 1$, while formulae (5.2.6) are considered at the final points x_i, $i = n - k/2 + 1, \ldots, n$,

$$\underbrace{y_1\, \cdots y_{\frac{k}{2}-1}}_{\frac{k}{2}-1\text{ initial methods}}\, y_{\frac{k}{2}} \cdots y_{n-\frac{k}{2}}\, \underbrace{y_{n-\frac{k}{2}+1}\, y_{n-\frac{k}{2}+2} \cdots y_{n-1} y_n}_{\frac{k}{2}\text{ final methods}}.$$

Initial Value Problem

For Theorem 1.4.15 the singular initial value problem defined by (5.1.1) with the singular initial conditions

$$\begin{array}{l} [y,u]\,(a) = \gamma_1, \\ [y,v]\,(a) = \gamma_2, \end{array} \tag{5.2.10}$$

has a unique solution on the interval $[a,b]$. As in [9], the numerical solution of (5.1.1)-(5.2.10) on the mesh (5.1.5) can be computed applying the main formula (5.1.6) at the points x_i, for $i = k/2, \ldots, n-k$, while additional formulae (5.2.5), involving $y'(a)$, are taken in consideration for the approximations at the points x_i, for $i = 0, \ldots, k/2$. Final formulae are given by (5.1.6) for $i = n-k+1, \ldots, n-1$. Therefore, in vector form the global approximation for the second and the first derivatives is given by

$$Y^{(\nu)} = \tilde{A}_\nu \tilde{Y} \tag{5.2.11}$$

where $\tilde{Y} = (y'_0, y_0, \ldots, y_n)^T$ and $\tilde{A}_\nu$, for $\nu = 1, 2$, are $n \times (n+2)$ matrices.

5.3 Algebraic solution of Sturm-Liouville problems

We apply high order finite difference schemes to Sturm-Liouville equations (5.1.1) rewritten as

$$-p(x)y'' - p'(x)y' + q(x)y = \lambda r(x)y, \qquad x \in [a,b], \quad y, \lambda \in \mathbb{R} \tag{5.3.1}$$

subjected, for simplicity, to regular boundary conditions (5.2.1). Then the discrete problem associated with (5.1.1) can be written as

$$R^{-1}\left(-PA_2 - P_1A_1 + Q\right)Y = \lambda Y \tag{5.3.2}$$

where R, P, P_1 and Q are diagonal matrices of size $n-1$ containing $r(x_i)$, $p(x_i)$, $p'(x_i)$ and $q(x_i)$, $i = 1, \ldots, n-1$, respectively. The square matrices A_1 and A_2 are extracted by $\tilde{A}_1$ and $\tilde{A}_2$ in (5.2.2) neglecting the first and the last column, which are multiplied by $y_0 = y_n = 0$ and we remind $Y = (y_1, \ldots, y_{n-1})^T$.

Then, seek eigenvalues and eigenfunctions of (5.1.1)-(5.2.1) is equivalent to solving the algebraic eigenvalue problem of the sparse matrix $M = R^{-1}\left(-PA_2 - P_1A_1 + Q\right)$. Hence, algebraic methods may be used to compute the eigenvalues.

For general regular boundary conditions (5.1.4), where a_1, a_2, b_1 and b_2 are nonnull, the idea is always to bring the computation of the eigenvalues and

eigenfunctions of the problem (5.1.1) back an equivalent algebraic problem. Then, from (5.2.7) the approximation of (5.3.1) is given by

$$\tilde{R}^{-1}\left(-\tilde{P}\tilde{A}_2-\tilde{P}_1\tilde{A}_1+\tilde{Q}I\right)\tilde{Y}=\lambda Y, \tag{5.3.3}$$

where $\tilde{Y}=(y_0',Y,y_n')^T$, with $Y=(y_0,y_1,\ldots,y_n)^T$, and $\hat{I}=[O_{n+1}\ I_{n+1}\ O_{n+1}]$ is a $(n+1)\times(n+3)$ matrix with I_{n+1} the identity matrix of size $n+1$ and O_{n+1} the null vector. Moreover, $\tilde{R}$, $\tilde{P}$, $\tilde{P}_1$ and $\tilde{Q}$ are diagonal matrices of size $n+1$ containing $r(x_i)$, $p(x_i)$, $p'(x_i)$ and $q(x_i)$, $i=0,\ldots,n$. As a consequence of (5.1.4), we consider a matrix of transformation

$$D=\begin{pmatrix} a_2p(a) & a_1 & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & b_1 & b_2p(b) \end{pmatrix}_{(n+3)\times(n+3)}.$$

such that $D\tilde{Y}=(a_1y_0+a_2p(a)y_0',\ y_0,\ \ldots,\ y_n,\ b_1y_n+b_2p(b)y_n')^T$ contains the first and the last element which are zero. Then (5.3.3) is equivalent to

$$\tilde{R}^{-1}\left(-\tilde{P}\tilde{A}_2-\tilde{P}_1\tilde{A}_1+\tilde{Q}I\right)D^{-1}D\tilde{Y}=\tilde{R}^{-1}\left(-\tilde{P}A_2-\tilde{P}_1A_1+\tilde{Q}\right)Y=\lambda Y,$$

where A_2 and A_1 are again square matrices extracted by the rectangular matrices $\tilde{A}_2D^{-1}$ and $\tilde{A}_1D^{-1}$, respectively, erasing the first and the last column. As in the previous case we obtain an algebraic eigenvalue problem, $MY=\lambda Y$, whose eigenvalues and eigenvectors are approximations of the eigenvalues and eigenfunctions of the original problem (5.3.1)-(5.1.2). We proceed in an analogous way for an eigenvalue problem with initial conditions defined in Section 5.2.

We point also out that, if a and b are LP points, then the algebraic problem has $Y=(y_0,y_1,\ldots,y_n)^T$ and $M=\tilde{R}^{-1}\left(-\tilde{P}\tilde{A}_2-\tilde{P}_1\tilde{A}_1+\tilde{Q}\right)$, with $\tilde{R}$, $\tilde{P}$, $\tilde{P}_1$ and $\tilde{Q}$ are diagonal matrices of size $n+1$ containing $r(x_i)$, $p(x_i)$, $p'(x_i)$ and $q(x_i)$, $i=0,\ldots,n$. The square matrices $\tilde{A}_2$ and $\tilde{A}_1$ are straight obtained by applying the high order difference schemes.

In general, considering a singular SLP (5.3.1) with boundary conditions (5.1.2) we can always obtain an equivalent algebraic problem of the form

$$M\cdot Y=\lambda Y. \tag{5.3.4}$$

We can summarize that the numerical method applied to (5.1.1) yields a square matrix M, therefore the computation of the eigenvalues and eigenfunctions is reduced to solve an algebraic problem (5.3.4) by means of an algebraic method. The eigenvalues of such matrix are in general good approximations of the first eigenvalues of (5.1.1)-(5.1.2).

In the case of boundary conditions depending nonlinearly on the parameter λ, the algebraic method is not enough to guarantee a good accuracy for the eigenvalues and eigenfunctions estimate. For this reason the approximations of the k-th eigenvalue λ_k and the corresponding eigenfunction are computed by the solution of a nonlinear problem with unknowns $y_1, \ldots, y_n, \lambda_k$:

$$\begin{cases} p(t_i)y_i'' + q(t_i)y_i' + r(t_i)y_i = \lambda_k y_i, & i = 1, \ldots, n, \\ \sum_{i=0}^{n} y_i^2 = 1, \end{cases}$$

where the last row is a normalization condition for the eigenfunction.

5.4 Stepsize and Order Variation Strategy

This section explains how to compute the kth eigenvalue and the corresponding eigenfunction. We observe that (see [3]) finite differences appear to maintain their accuracy only in the computation of the first eigenvalue, while a loss of order is observed for the kth, $k > 1$, eigenvalue; for example three-point scheme causes an error of $O(k^4h^2)$ in the approximation of the kth eigenvalue. Our idea is to apply a strategy to compute a good estimation of the kth eigenvalue based on the Matlab function *eigs*, which computes smallest (largest) eigenvalues in absolute value.

Firstly we consider an uniform mesh π as in (5.1.5) with $n = 30$ and start off computing the first eigenvalue λ_0 by means of the matlab function *eig* in order that its numerical relative error satisfies a given tolerance 10^{-2}, this means

$$\frac{\left|\lambda_0^{(p)} - \lambda_0^{(p+2)}\right|}{\left|\lambda_0^{(p+2)}\right|} < 10^{-2},$$

where $\lambda_0^{(p)}$ and $\lambda_0^{(p+2)}$ are approximations of the first eigenvalue obtained using two consecutive even orders p and $p + 2$.

After that using the matlab function *eigs*, which calculate the first k largest in magnitude eigenvalues of a sparse matrix starting from λ_0, we estimate the

approximation of the kth eigenvalue in order to satisfy an error tolerance equal to 10^{-3}, that is

$$\frac{\left|\lambda_k^{(p)} - \lambda_k^{(p+2)}\right|}{\left|\lambda_k^{(p+2)}\right|} < 10^{-3}, \tag{5.4.1}$$

where $\lambda_k^{(p)}$ and $\lambda_k^{(p+2)}$ are approximations with two consecutive even orders p and $p+2$. We point out that, as a consequence of (5.1.3), we calculate the smallest eigenvalues of $M - \lambda_0 I$, so that the eigenvalues are computed in the correct order. This procedure guarantees us to have a good approximation of the desired eigenvalue λ_k. If $n+1$ is the number of mesh points needed to calculate this first good approximation, then considering an uniform mesh with n equidistant intervals we compute the new approximation of λ_k on the uniform mesh; moreover, by means of the function *eigs* we evaluate only the kth eigenvalue, since we calculate the first eigenvalue close to the previous good approximation of λ_k. We can summarize this procedure in the following algorithm.

Algorithm 4. Algorithm for the first estimation of λ_k and y_k

1. $n = 30$;
2. $\widetilde{x} = a : (b-a)/n : b$;
3. $itol = 10^{-2}$;
4. $\lambda_0^{(p)} = eig(M_p)$, $\lambda_0^{(p+2)} = eig(M_{p+2})$;
5. while $err_k = \ (\lambda_0^{(p)} - \lambda_0^{(p+2)})/\lambda_0^{(p+2)} \ < itol$ & $n < nmax$;
6. $\quad n = n + 10$, update M_p and M_{p+2};
7. $\quad \lambda_0^{(p)} = eig(M_p)$, $\lambda_0^{(p+2)} = eig(M_{p+2})$;
8. end;
9. $\lambda_k^{(p)} = eigs(M_p, k, \lambda_0^{(p)})$, $\lambda_k^{(p+2)} = eig(M_{p+2}, k, \lambda_0^{(p)})$;
10. while $err_k < itol/10$ & $n < nmax$;
11. $\quad n = 2n$, update M_p and M_{p+2};
12. $\quad \lambda_k^{(p)} = eigs(M_p, k, \lambda_0^{(p)})$ and $\lambda_k^{(p+2)} = eigs(M_{p+2}, k, \lambda_0^{(p)})$;
13. end;
14. $[y_k, \lambda_k] = eigs(M_p, k, \lambda_k^{(p)})$;

For the largest eigenvalue a good accuracy of the corresponding eigenfunction on a uniform mesh requires a large number n of mesh points. In order to improve the precision of the eigenfunctions preserving a small number of mesh points, we consider a stepsize variation strategy based on the equidistribution of the error, seen in Section 3.4.

Given a mesh π as in (5.1.5) and an error tolerance TOL, our aim is to compute a variable mesh π^* with a small number of points n^* such that the error

$$e_i = \left| y_i^{(k)} - y^{(k)}(x_i) \right| < TOL, \quad i = 0, \ldots, n^*, \tag{5.4.2}$$

where $y_i^{(k)}$ is the numerical approximation of kth eigenfunction on the new mesh π^*. Estimation error is obtained considering the numerical error of two solutions computed with orders p and $p+2$.

When a good approximation of the eigenvalue λ_k has been estimated by using Algorithm 4, then if the relative error for the eigenvalue approximation and the absolute error for the eigenfunction approximation are not less than TOL, then we proceed applying a stepsize variation strategy based on the equidistribution. From Section 3.4, we know that if $T(x)$ is the monitor function, then to predict the new mesh size we have to compute

$$r_1 = \max_{0 \le i \le n} h_i \left(\frac{T(x_i)}{TOL} \right)^{1/p} \qquad r_2 = \sum_{i=0}^{n} h_i \left(\frac{T(x_i)}{TOL} \right)^{1/p}, \qquad r_3 = \frac{r_2}{n},$$

where the ratio $\frac{r_1}{r_3}$ gives some information about the equidistribution; in fact, if the ratio is large, the maximum error estimate is larger than the average one, this means that the mesh is not well equidistributed. Moreover we require to check that

$$\frac{r_1}{r_3} < 1.2 \tag{5.4.3}$$

so that we can distinguish two cases.

(i) If (5.4.3) is satisfied, then the mesh is sufficiently equidistributed and the new mesh is obtained doubling the points, that is, we have $\pi^* = \{x_1, x_{3/2}, x_2, \ldots, x_{n-1}, x_{(n+1/2)}, x_n\}$.

(ii) If (5.4.3) is not checked, then r_2 predicts the number of mesh points satisfying the tolerance TOL. Then we predict the number of mesh points $n^* = \max\{\min(n, 2.5n), n/2\}$ in place of r_2, in order to avoid incorrect conclusions early. Then the new mesh π^* is computed by

$$t(x) := \frac{1}{\theta} \int_a^x T(\xi)^{1/p} d\xi \tag{5.4.4}$$

and known x_i we may finding x_{i+1} such that

$$t(x_{i+1}) = \frac{i}{n}. \tag{5.4.5}$$

Since we want to preserve the properties of the methods on the equidistant points, we modify this value of n^* in order to obtain a piecewise variable stepsize with blocks of $p+4$ equidistant points. For this reason we consider $\lceil n^*/(k+4) \rceil$ as the predict number of mesh points and compute the new mesh by (5.4.4)-(5.4.5), after that we put $p+4$ equidistant points in each interval. We also underline that the mesh is doubled if n^* has already been used for two consecutive times.

Hence, we decide to change stepsize at least every $k+4$ points, that is we use 3 constant steps methods before changing the stepsize, if necessary, and bound the ratio of two consecutive stepsizes v according to the values in Table 5.1.

Table 5.1: Maximum ratio between two successive steps.

order	4	6	8	10
v	4	3	2	1.5

We can summarize the equidistribution algorithm as follows.

Algorithm 5.

function $x =$ `monitor` $(err, \widetilde{x}, ord, TOL)$

1. $n = \text{length}(\widetilde{x}) - 1$;
2. $h = \widetilde{x}(2:n) - \widetilde{x}(1:n-1)$;
3. $T = \max(err(2:n+1), err(1:n))\hat{}(1/ord)$;
4. $r_1 = \|T\|_\infty$;
5. $n^* = \lfloor \|T\|_1 / TOL\hat{}(1/ord) \rfloor$;
6. $r_3 = \|T\|_1 / n$;
7. if $r_1/r_3 \leq 1.2$ & $n^* \geq 2 \cdot n$
8. $\quad x$ is obtained halving the step-length vector h
9. $\quad n^* = 2 \cdot n$

10. else
11. $n^* = \max(\min(n^*, \lfloor 2.5 \cdot n \rfloor), \lfloor n/2 \rfloor)$;
12. $n^* = \lceil n^*/(k+4) \rceil$
13. $I = [0 \ \text{cumsum}(T)]/\|T\|_1$;
14. $z = 0 : 1/n^* : 1$;
15. $\widehat{x} = \texttt{linear\_interp}(I, \widetilde{x}, z)$;
16. $x = \texttt{piecewise\_grid}(\widehat{x}, ord + 4)$;
17. end

We also impose an order variation strategy which consists in requiring different error tolerances for each order; moreover, the solution obtained with each order is modified by the *monitor function* for preserving the required piecewise structure and it is used for the next order as initial mesh. The simple algorithm is the following.

Algorithm 6. Order Variation

Given $p = [4, 6, 8, 10]$ and $TOL = [TOL_1, TOL_2, TOL_3, TOL_4]$ such that $TOL_1 > TOL_2 > TOL_3 > TOL_4$.

1. $s = length(p)$;
2. for $i = 1 : s$
3. Calculate y_k, λ_k with the method of order $p(i)$ such that
$$\left\| y_k^{(p)} - y_k^{(p+2)}(x) \right\|_\infty < TOL(i) \quad \& \quad \left| \lambda_k^{(p)} - \lambda_k^{(p+2)} \right| / \left| \lambda_k^{(p+2)} \right| < TOL(i);$$
4. end

5.5 Test Problems

In this section we contemplate some examples on regular and singular Sturm-Liouville problems, defined on bounded or unbounded intervals. For the unbounded interval we transform it in a limited one. Constant stepsize allows us to estimate the numerical order of convergence, but, as we discuss in Section 5.4, for a better accuracy of the eigenfunctions associated with the larger eigenvalues, we consider variable stepsize too. We remind that the matlab

code uses the function *eigs* to compute the eigenvalues and eigenfunctions of the algebraic problem. Moreover, we also point out that for singular problems the coefficients of Sturm-Liouville problems could be not defined in the endpoints of the interval $[a, b]$, so that we consider the truncated interval $[\alpha, \beta]$ with $a < \alpha < \beta < b$, as in [50]. When a method of order p is used, the relative error for the kth eigenvalue is given by

$$E_r(\lambda_k) = \frac{\left|\lambda_k^{(p)} - \lambda_k^{(p+2)}\right|}{\left|\lambda_k^{(p+2)}\right|}, \tag{5.5.1}$$

while the absolute error for the eigenfunction associated to λ_k is defined as

$$E_a(y_k) = \left\|y_k^{(p)} - y_k^{(p+2)}\right\|_\infty. \tag{5.5.2}$$

When the theoretical value of the eigenvalue is known we may calculate also the theoretical relative error as

$$E_r^{teor}(\lambda_k) = \frac{\left|\lambda_k^{(p)} - \lambda_k\right|}{|\lambda_k|}. \tag{5.5.3}$$

Example 5.5.1. We consider the Klotter problem (see [58]) defined as

$$-y''(x) + \frac{3}{4x^2}y(x) = \lambda\frac{64\pi^2}{9x^6}y(x), \quad x \in [8/7, 8],$$
$$a = 8/7 \quad \text{regular},$$
$$b = 8 \quad \text{regular},$$

with boundary conditions $y(8/7) = y(8) = 0$. The eigenvalues are

$$\lambda_k = (k+1)^2, \qquad k = 0, 1, \ldots$$

The problem is regular and the classical D2ECDFs are applied as discussed in Section 5.2. Table 5.2 shows as the numerical order is preserved for each eigenvalue, with the exception of λ_4 and order 6, since in this case the numerical order is not reliable for the number of mesh points chosen, even if the relative error decreases. It is unmistakable as greater eigenvalues require more equidistant points than the smaller ones in order to reach a good approximations. A better accuracy is gained with higher order methods.

For this reason in Table 5.3 we show the theoretical relative error for the eigenvalues and the numerical absolute error for the eigenfunctions, which are both obtained applying the same schemes and using the stepsize and order variation strategy in Section 5.4. We choose to start with order 4 and change order for $TOL = 10^{-3}, 10^{-6}, 10^{-8}$. The exit condition is $TOL = 10^{-11}$ and it is reached with order 10, this means that we use all even order from 4 to 10. We underline as n_0 is the initial number of mesh points in order to have a good initial estimation of the eigenvalue satisfying (5.4.1); n is the final number of mesh points. In Table 5.3 we compare the results obtained by this strategy with those using the minimum stepsize h_{min} of the variable mesh as a constant stepsize for $[a,\ b]$. It is clearly visible that the number of points is much reduced if the order and stepsize variation strategy is applied, and this behavior is more evident for the largest eigenvalues. In Figure 5.1 we draw the eigenfunction associated with λ_4 on a variable mesh and exit condition $TOL = 10^{-11}$, while in the Figure 5.2 we plot the absolute error of the same eigenfunction and also the obtained stepsize variation.

Example 5.5.2. The regular problem (see [58])

$$-\left(\frac{1}{\sqrt{1-x^2}}y'\right)'(x) = \lambda\frac{1}{\sqrt{1-x^2}}y(x), \quad x \in [-1,1],$$
$$a = -1 \quad \text{regular},$$
$$b = 1 \quad \text{regular}$$

has boundary conditions $y(-1) = y(1) = 0$. It is important to specify that, though being defined regular, it has the coefficients $p(x)$ and $r(x)$ not defined in the endpoints and hence it looks singular. This example is meaningful since it shows as this 'singular' behavior wreaks a loss of order. In fact the results in Table 5.4 bear out the slow convergence and the numerical order reaches the same constant value for every eigenvalue and order of the method. Here it is known from [58] that $\lambda_0 = 3.559279966$, $\lambda_9 = 258.8005854$ and $\lambda_{24} = 1572.635284$.

Example 5.5.3. The Paine problem (see [44]) is defined as

$$-((u+x)^3y'(x))' + 4(u+x)y(x) = \lambda(u+x)^5y(x),$$

with boundary conditions $y(0) = y(-u+\sqrt{u^2+2\pi}) = 0$ and $u = \sqrt{2}$ $x \in (0,\ \ -u+\sqrt{u^2+2\pi})$. For this regular problem we apply the order and stepsize variation strategy starting with order 6 and changing order for $TOL =$

Table 5.2: Example 5.5.1 - Numerical order of convergence.

		$p=4$		$p=6$		$p=8$	
	Mesh	Error	Order	Error	Order	Error	Order
λ_0	200	1.01e-05	3.45	4.26e-07	6.16	1.61e-07	8.65
	400	9.19e-07	3.89	5.97e-09	6.91	4.01e-10	9.08
	600	1.90e-07	3.96	3.63e-10	7.35	1.01e-11	1.45
	800	6.08e-08	-	4.38e-11	-	1.55e-13	-
λ_4	300	2.88e-04	7.13	9.14e-05	12.22	2.64e-05	8.05
	600	2.05e-06	1.92	1.92e-08	2.34	9.95e-08	9.96
	900	9.42e-07	2.38	7.42e-09	7.57	1.75e-09	10.81
	1.200	4.75e-07	3.24	8.39e-10	11.22	7.82e-11	12.97
	1.500	2.31e-07	-	6.86e-11	-	4.33e-12	-
λ_{24}	2000	5.69e-05	3.19	5.53e-06	6.85	3.33e-07	7.27
	2500	2.79e-05	3.53	1.20e-06	6.92	6.58e-08	8.84
	3000	1.47e-05	3.70	3.40e-07	6.85	1.31e-08	9.51
	3500	8.30e-06	3.79	1.18e-07	6.74	3.03e-09	9.93
	4000	5.00e-06	-	4.81e-08	-	8.05e-10	-

Table 5.3: Example 5.5.1 - Numerical and theoretical relative error for the eigenvalue λ_k and absolute error for the corresponding eigenfunction obtained for variable order and stepsize strategy with exit tolerance $TOL = 10^{-11}$ and compared with constant stepsize h_{min}.

	k	0	4	24
Variable	n_0	21	41	501
	n	169	379	2102
	h_{min}	9.40e-03	3.36e-03	7.47e-04
	$E_r(\lambda_k)$	2.67e-13	2.06e-13	4.07e-14
	$E_r^{teor}(\lambda_k)$	2.67e-13	2.06e-13	4.07e-14
	$E_a(y_k)$	2.89e-13	4.47e-13	3.69e-13
Constant	$\tilde{n} = (b-a)/h_{min}$	731	2039	9177
	$E_r(\lambda_k)$	2.11e-13	5.51e-12	7.02e-12
	$E_r^{teor}(\lambda_k)$	6.17e-13	1.72e-13	1.28e-12
	$E_a(y_k)$	1.725e-13	1.70e-11	1.21e-10

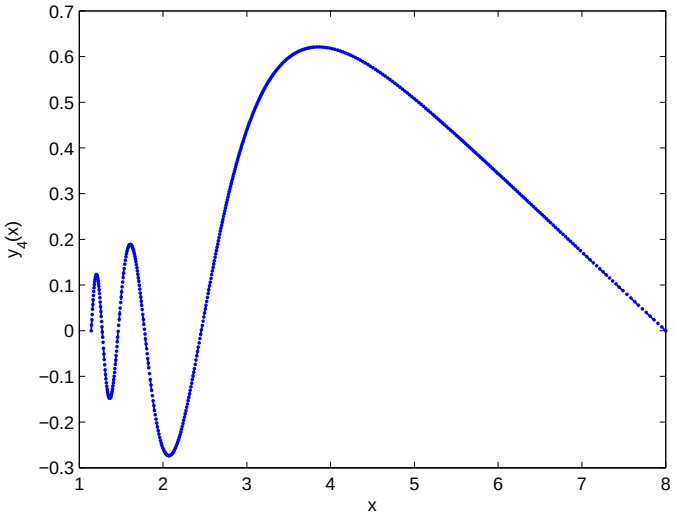


Figure 5.1: Example 5.5.1 - Eigenfunction associated to λ_4 computed with variable stepsize with exit tolerance $TOL = 10^{-11}$.

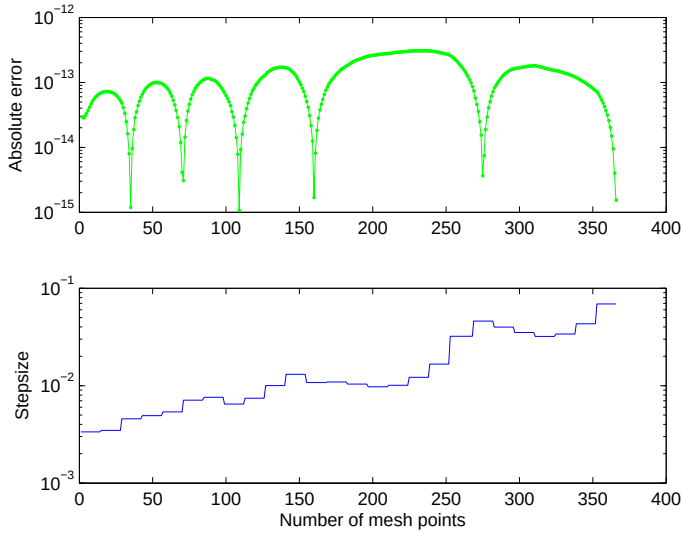


Figure 5.2: Example 5.5.1 - Absolute error for $y_4(x)$ and stepsize variation with exit tolerance $TOL = 10^{-11}$.

Table 5.4: Example 5.5.2 - Numerical order of convergence.

		$p=4$		$p=6$		$p=8$	
	Mesh	Error	Order	Error	Order	Error	Order
λ_0	1000	1.57e-05	1.50	1.41e-05	1.50	1.26e-05	1.50
	2000	5.56e-06	1.50	5.00e-06	1.50	4.44e-06	1.50
	3000	3.03e-06	1.50	2.72e-06	1.50	2.42e-06	1.50
	4000	1.97e-06	-	1.77e-06	-	1.57e-06	-
λ_9	6000	2.93e-06	1.50	2.63e-06	1.50	2.34e-06	1.50
	9000	1.59e-06	1.50	1.43e-06	1.50	1.27e-06	1.50
	12000	1.04e-06	1.50	9.31e-07	1.50	8.27e-07	1.50
	15000	7.41e-07	-	6.66e-07	-	5.92e-07	-
λ_{24}	10000	2.13e-06	1.50	1.91e-06	1.50	1.70e-06	1.50
	15000	1.16e-06	1.50	1.04e-06	1.50	9.25e-07	1.50
	20000	7.53e-07	1.50	6.76e-07	1.50	6.00e-07	1.50
	25000	5.38e-07	-	4.84e-07	-	4.30e-07	-

$10^{-3}, 10^{-6}$. The exit condition is $TOL = 10^{-9}$ and is reached with order 10. In this case we use only three consecutive even orders. The estimations obtained are comparable with those in MATSLISE [47]. In Table 5.5 we display the numerical relative and absolute errors for the eigenvalues and the eigenfunctions, respectively. Moreover, we consider n_0 to be the initial number of constant mesh points which guarantees a good initial approximation of the eigenvalue by (5.4.1) and n is the final number of points required for obtaining the exit tolerance $TOL = 10^{-9}$. In the same table we compare these results with those reached using the minimum stepsize of the variable mesh h_{min} as a constant stepsize for $[a, b]$. It is clearly visible that the order and stepsize variation strategy allows us to reach the precision required almost halving the number of the equidistant points $\tilde{n} = (b - a)/h_{min}$. We point out that, for variable stepsize, we have the same accuracy for the eigenvalues and the corresponding eigenfunctions. On the contrary, with constant stepsize the first eigenvalues are better approximated than the larger ones, and the eigenvalues approximations are much more accurate than the eigenfunctions ones.

Example 5.5.4. The truncated hydrogen equation (see [58]) is defined as,

$$-y''(x) + \left(\frac{2}{x^2} - \frac{1}{x}\right) y(x) = \lambda y(x), \quad x \in [0, 1000],$$
$$a = 0 \quad \text{LP},$$
$$b = 1000 \quad \text{regular}.$$

Table 5.5: Example 5.5.3 - Numerical and theoretical relative error for the eigenvalue λ_k and absolute error for the corresponding eigenfunction obtained for variable order and stepsize strategy with exit tolerance $TOL = 10^{-9}$ and compared with constant stepsize h_{min}.

	k	0	4	19	24
Variable	n_0	16	21	41	51
	n	99	183	519	603
	h_{min}	1.29e-02	9.89e-03	3.21e-03	2.57e-03
	$E_r(\lambda_k)$	3.89e-13	1.53e-12	1.08e-12	1.41e-12
	$E_a(y_k)$	3.72e-11	8.29e-12	4.58e-12	6.72e-12
Constant	$\tilde{n} = (b-a)/h_{min}$	245	319	979	1224
	$E_r(\lambda_k)$	8.30e-13	1.80e-13	9.31e-15	1.20e-13
	$E_a(y_k)$	6.02e-13	3.22e-13	2.17e-13	1.08e-12

The only condition is $y(1000) = 0$. The eigenvalues are

$$\lambda_k = -\frac{1}{(2k+4)^2},\ k = 0, 1, \ldots$$

Since $a = 0$ is LP, no boundary condition is required and an initial method approximates the problem in the left endpoint, see Remark 5.2.2. Since $q(x) = 2/x^2 - 1/x$ is not defined in a, we also truncate the interval (see [50]) and choose $\alpha > a$ close to zero. Moreover, the results in Table 5.6 show that the numerical order is gained, the convergence for order 8 shows results less reliable for the choice of the mesh points, even if the accuracy improves with the higher order. We emphasize that, on a truncated interval, a good eigenvalue estimation is reached until $k = 9$, since when k increases the eigenfunction is not zero in the right endpoint and oscillations range on a greater interval. In order to compare the results with the next problem, we have also drawn the eigenfunction associated to λ_4 in Figure 5.3 (above).

Example 5.5.5. The Hydrogen atom equation in Example 5.5.4 is integrated for $x \in [0, \infty]$, where

$$b = \infty \quad \text{LP}.$$

With respect to the previous example, both endpoints are LP and no boundary conditions are given, so we consider one initial and final methods to approximate the problem in both endpoints. Moreover, the upper unbounded interval

Table 5.6: Example 5.5.4 - Numerical order of convergence.

		$p=4$		$p=6$		$p=8$	
	Mesh	Error	Order	Error	Order	Error	Order
λ_0	500	2.18e-03	4.23	9.77e-04	6.89	2.00e-04	9.42
	1000	1.16e-04	2.96	8.22e-06	6.64	2.92e-07	11.15
	1500	3.49e-05	3.62	5.56e-07	6.37	3.17e-09	18.34
	2000	1.23e-05	-	8.91e-08	-	1.61e-11	-
λ_4	500	7.36e-04	4.64	1.17e-04	7.61	9.75e-06	10.18
	1000	2.95e-05	3.05	5.98e-07	6.90	8.38e-09	11.23
	1500	8.57e-06	3.68	3.64e-08	6.45	8.82e-11	14.28
	2000	2.97e-06	3.84	5.68e-09	6.24	1.45e-12	6.81
	2500	1.26e-06	-	1.41e-09	-	3.17e-13	-
λ_9	1000	1.44e-05	3.02	2.35e-07	6.98	2.54e-09	11.22
	1500	4.23e-06	3.67	1.39e-08	6.48	2.69e-11	10.95
	2000	1.47e-06	3.84	2.15e-09	6.27	1.15e-12	8.72
	2500	6.24e-07	-	5.31e-10	-	1.65e-13	-

is transformed by means of the simple change of variable

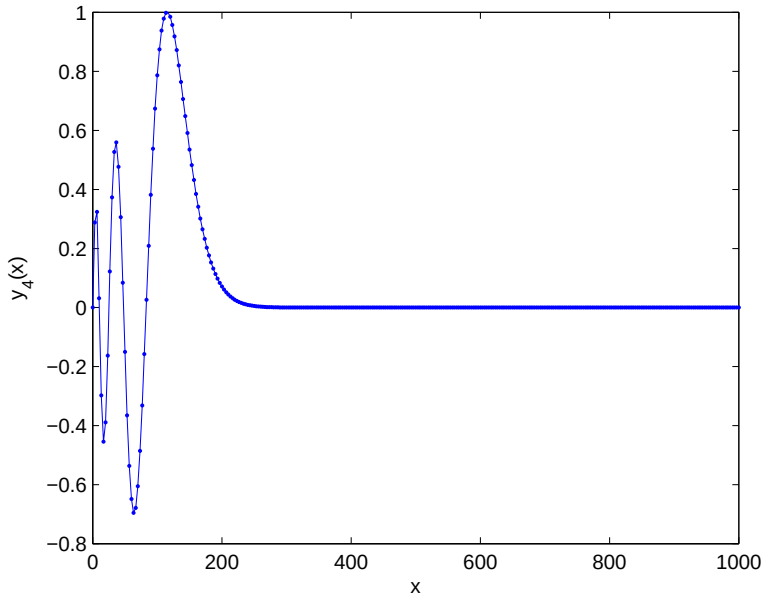
$$\tilde{x}_i = 1 - \frac{1}{\sqrt{1+x_i}} \in [0,1].$$

Obviously a constant stepsize in $[0,1]$ gives a solution with variable stepsize in the original interval, as it is possible to see in Figure 5.3 (below), in fact we truncate the solution, obtained in the interval $(0,\infty)$, in $b = 1000$, so that it is possible to compare the mesh selection with the solution of Example 5.5.4 plotted in Figure 5.3 (above). Conversely to Example 5.5.4 few points ensure a good approximation and numerical order is preserved. As noted in the other examples greatest accuracy is guaranteed by higher order methods.

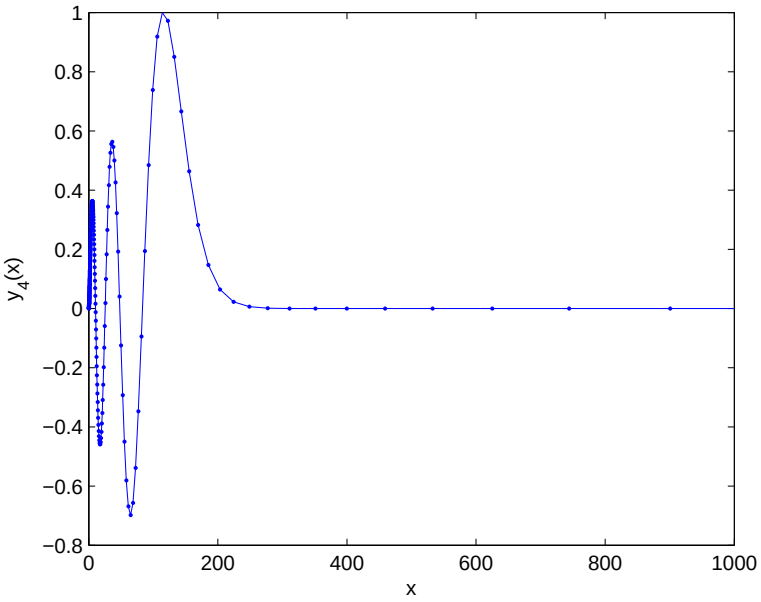
However, as shown in Table 5.7 the number of mesh points increases much more when the greatest eigenvalues are computed, for this reason we apply only the stepsize variation strategy for the smallest eigenvalues, see from Table 5.8 to Table 5.10. For the greatest eigenvalues we consider also the order variation strategy, as shown in Table 5.11 and Table 5.12.

The results in Table 5.8 and Table 5.9 seem to confirm that the higher orders 8 and 10 give a better accuracy with few points for the first eigenvalues and eigenfunctions. Moreover, comparing the results in Table 5.8 with those in Table 5.7 it is evident that we do not gain much on the number of points.

In Table 5.10 we display the stepsize and order variation strategy for the computation of the eigenvalue λ_4 and the eigenfunction y_4. We start with order 6 and $TOL = 10^{-6}$ and we reach the solution with order 10 and exit



(a) Example 5.5.4.



(b) Example 5.5.5.

Figure 5.3: Eigenfunctions $y_4(x)$ computed by the order 6 method with 300 points in the interval $[0, 1000]$ (above) and in the interval $(0, \infty)$ truncated at $b = 1000$ (below).

Table 5.7: Example 5.5.5 - Numerical order of convergence.

		$p=4$		$p=6$		$p=8$	
	Mesh	Error	Order	Error	Order	Error	Order
λ_0	100	2.79e-06	3.99	4.62e-08	5.96	1.60e-09	7.89
	150	5.53e-07	3.99	4.13e-09	5.98	6.51e-11	7.99
	200	1.75e-07	4.00	7.40e-10	5.99	6.53e-12	7.55
	250	7.19e-08	-	1.94e-10	-	1.21e-12	-
λ_4	200	3.16e-04	3.96	2.14e-05	5.90	2.21e-06	7.80
	400	2.03e-05	3.99	3.59e-07	5.97	9.88e-09	7.11
	600	4.03e-06	3.99	3.19e-08	5.98	5.53e-10	12.48
	800	1.28e-06	-	5.72e-09	-	1.52e-11	-
λ_9	500	8.14e-04	3.95	6.87e-05	5.89	7.70e-06	7.80
	1000	5.25e-05	3.98	1.16e-06	5.96	3.46e-08	7.80
	1500	1.04e-05	3.99	1.03e-07	5.98	1.46e-09	9.74
	2000	3.31e-06	-	1.85e-08	-	8.89e-11	-
λ_{24}	2000	2.76e-03	3.93	3.73e-04	5.84	6.09e-05	7.74
	4000	1.81e-04	3.97	6.53e-06	5.95	2.85e-07	7.93
	6000	3.61e-05	3.99	5.85e-07	5.97	1.15e-08	8.14
	8000	1.15e-05	-	1.05e-07	-	1.10e-09	-

condition $TOL = 10^{-10}$. The order and stepsize variation strategy give us a better accuracy with a small number of points, as a matter of fact in Table 5.7 more equidistant points are required for reaching the same precision for order 8. Moreover, if we use 350 equidistant points and order 10, then we obtain relative error $E_r(\lambda_4) = 1.27\text{e}{-09}$ for the eigenvalue and absolute error $E_a(y_4) = 2.72\text{e}{-07}$ for the eigenfunction, the precision is worst and, also increasing the number of the equidistant points until 5000, it remains unchanged. In Figure 5.4 and Figure 5.5 we show the eigenfunction $y_4(x)$ and the final stepsize variation obtained in Table 5.10.

For the eigenvalues λ_9 and λ_{24} we apply the order and stepsize variation strategy starting with order 8 and $TOL = 10^{-6}$ and we proceed using order 10 and $TOL = 10^{-10}$. In both cases, Table 5.11 and Table 5.12 show that the results are more accurate and require a small number of mesh points with respect to of those in Table 5.7 for order 8. Moreover, for λ_9 if we use 753 equidistant points, $E_r(\lambda_9) = 1.88\text{e}{-08}$ and $E_a(y_9) = 2.17\text{e}{-06}$, while taking 3264 equidistant points for λ_{24}, $E_r(\lambda_{24}) = 9.66\text{e}{-08}$ and $E_a(y_{24}) = 1.89\text{e}{-05}$. We point out also that with 5000 points we have still 9 digits correct for the eigenvalue and 7 for the eigenfunctions in the both cases. The order and variation strategy gives us a finer mesh.

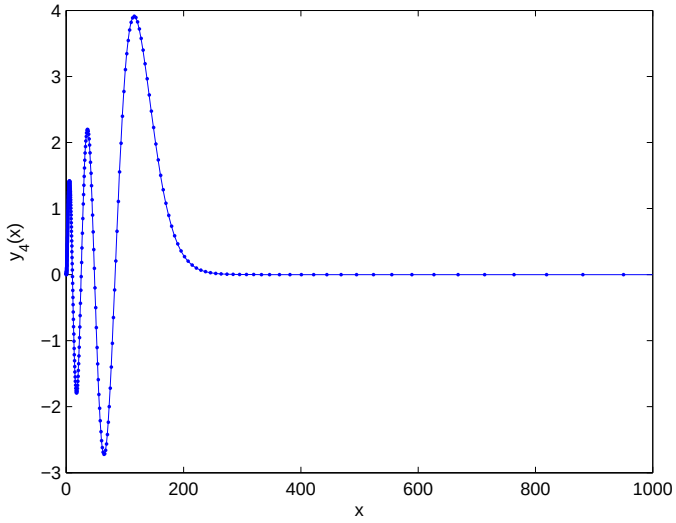


Figure 5.4: Example 5.5.5 - Eigenfunction $y_4(x)$ obtained with 341 mesh points, see Table 5.10, initial order 8 and exit condition $TOL = 10^{-10}$. The solution is truncated at $x = 1000$.

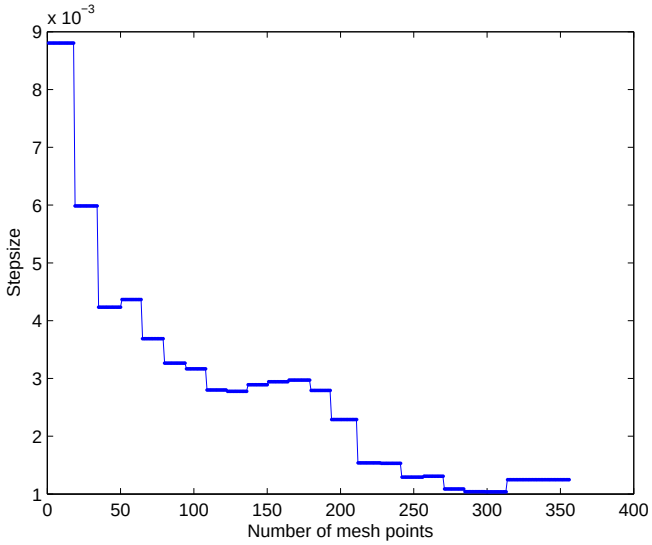


Figure 5.5: Example 5.5.5 - Stepsize variation for last step in Table 5.10 for λ_4, exit condition $TOL = 10^{-10}$.

Table 5.8: Example 5.5.5 - Mesh selection steps for the estimation of λ_0 and y_0 using the variable stepsize strategy, order $p = 8$ and exit condition $TOL = 10^{-10}$.

	$p = 8 \quad TOL = 10^{-10}$		
Mesh	31	84	191
$E_r(\lambda_0)$	7.93e-06	2.78e-09	4.72e-12
$E_r^{teor}(\lambda_0)$	1.23e-05	2.23e-09	4.73e-12
$E_a(y_0)$	2.83e-04	1.63e-07	1.86e-11
h_{min}	3.33e-02	8.96e-03	2.71e-03
h_{max}	3.33e-02	1.72e-02	1.25e-02

Table 5.9: Example 5.5.5 - Mesh selection steps for the estimation of λ_0 and y_0 using the variable stepsize strategy, order $p = 10$ and exit condition $TOL = 10^{-10}$.

	$p = 10 \quad TOL = 10^{-10}$		
Mesh	31	82	135
$E_r(\lambda_0)$	1.90e-06	3.58e-10	1.71e-12
$E_r^{teor}(\lambda_0)$	4.36e-06	5.24e-10	1.80e-12
$E_a(y_0)$	1.61e-04	2.29e-08	5.24e-11
h_{min}	3.33e-02	9.15e-03	4.15e-03
h_{max}	3.33e-02	1.74e-02	1.51e-02

Example 5.5.6. The Legendre equation (see [37]) is defined as

$$\begin{aligned} &-((1-x^2)y')'(x) + \frac{1}{4}y(x) = \lambda y(x), \quad x \in [-1,1], \\ &a = -1 \quad \text{LCNO}, \\ &b = 1 \quad \text{LCNO}. \end{aligned}$$

The boundary conditions

$$\begin{aligned} [y,u](-1) &= -(py')(-1) = 0 \\ [y,u](1) &= -(py')(1) = 0 \end{aligned}$$

are obtained from (5.1.2) setting $a_1 = b_1 = 1$, $a_2 = b_2 = 0$, $u(x) = 1$ and $v(x) = \ln((1+x)/(1-x))$. The eigenvalues are

$$\lambda_k = \left(k + \frac{1}{2}\right)^2, \quad k = 0, 1, \ldots,$$

Table 5.10: Example 5.5.5 - Mesh selection steps for the estimation of λ_4 and y_4 using the order and stepsize variation strategy, we start with order $p = 8$ and exit condition $TOL = 10^{-10}$.

	$p = 8 \;\; TOL = 10^{-6}$		$p = 10 \;\; TOL = 10^{-10}$	
Mesh	121	269	244	341
$E_r(\lambda_4)$	6.51e-05	1.30e-08	2.93e-11	4.37e-13
$E_r^{teor}(\lambda_4)$	9.66e-05	1.55e-08	1.73e-11	4.82e-13
$E_a(y_4)$	1.98e-03	5.83e-07	1.12e-08	2.73e-11
h_{min}	8.32e-03	2.40e-03	2.07e-03	9.70e-04
h_{max}	8.32e-03	6.49e-03	8.92e-03	7.93e-03

Table 5.11: Example 5.5.5 - Mesh selection steps for the estimation of λ_9 and y_9 using the order and stepsize variation strategy, we start with order $p = 8$ and exit condition $TOL = 10^{-10}$.

	$p = 8 \;\; TOL = 10^{-6}$			$p = 10 \;\; TOL = 10^{-10}$	
Mesh	241	631	672	621	753
$E_r(\lambda_9)$	8.87e-04	3.48e-08	2.06e-09	1.13e-11	5.58e-13
$E_r^{teor}(\lambda_9)$	1.61e-03	3.57e-08	2.11e-09	1.19e-11	6.16e-13
$E_a(y_9)$	3.58e-02	3.81e-06	3.33e-07	1.72e-09	2.98e-11
h_{min}	4.16e-03	9.42e-04	6.29e-04	5.52e-04	3.33e-04
h_{max}	4.16e-03	3.22e-03	4.67e-03	5.48e-03	6.26e-03

while the kth Legendre polynomial is the corresponding eigenfunction. Since the function p is null in the endpoints, an initial and a final method approximate the solution in the both endpoints substituting the boundary conditions. The results obtained for small k show that 20 points are enough for a very good approximation, consequently the numerical order is estimated for $k = 9, 24, 49$. We underline that the numerical order is preserved and sometimes exceeds the order expected. Moreover, the number of points required to reach a prescribed accuracy is proportional to k and higher order methods guarantee a better accuracy. The constant stepsize gives a good accuracy for the smallest eigenvalues, for this reason in Table 5.14 and Table 5.15 we consider an order and stepsize variation only for the largest eigenvalues λ_{24} and λ_{49}. In both cases we start using order 6 and $TOL = [10^{-3}, 10^{-6}]$ and we satisfy the exit

Table 5.12: Example 5.5.5 - Mesh selection steps for the estimation of λ_{24} and y_{24} using the order and stepsize variation strategy, we start with order $p = 8$ and exit condition $TOL = 10^{-10}$.

	$p = 8 \;\; TOL = 10^{-6}$			$p = 10 \;\; TOL = 10^{-10}$	
Mesh	961	2476	2522	2133	3264
$E_r(\lambda_{24})$	4.70e-03	8.71e-08	2.44e-09	4.63e-11	1.34e-13
$E_r^{teor}(\lambda_{24})$	1.12e-02	9.05e-08	2.47e-09	4.69e-11	4.34e-13
$E_a(y_{24})$	6.54e-01	1.45e-05	4.36e-07	1.80e-08	3.87e-11
h_{min}	1.04e-03	1.93e-04	1.03e-04	8.65e-05	3.92e-05
h_{max}	1.04e-03	9.81e-04	1.99e-03	2.80e-03	3.16e-03

condition $TOL = 10^{-8}$ with order 10, so that we use three consecutive orders 6, 8, 10. As we have noted in the other examples, the number of mesh points decreases with the order and stepsize variation strategy, on the other hand the accuracy is improved. If we consider 464 equidistant points for λ_{24} , then $E_r(\lambda_{24}) = 2.61\text{e}{-10}$ and $E_a(y_{24}) = 1.23\text{e}{-08}$, while for the eigenvalue λ_{49} we have $E_r(\lambda_{49}) = 1.13\text{e}{-09}$ and $E_a(y_{49}) = 1.52\text{e}{-07}$ with 1632 equidistant points. This means that the strategy in Section 5.3 allows us to gain a good accuracy with a finer mesh, since increase the number of the equidistant points is not enough to reach the desired precision. Therefore, in Figure 5.6 we plot the eigenfunction $y_4(x)$ obtained with order 6 and 301 equidistant points and in Figure 5.7 we draw the eigenfunction λ_{24} on a variable mesh as already explained.

A Singular Self-Adjoint Sturm–Liouville Problem

In this section we consider the solution of a singular self-adjoint Sturm-Liouville problem associated to the computation of the eigenvalues and eigenfunctions of a finite truncated Hankel transform, see[13], since in the recent years these computations have assumed a great importance in some applications. In a 2D-case, the eigenfunctions of the Fourier transform truncated to a circle can be expressed using the eigenfunctions of the finite truncated Hankel transform (FHT).

FHT are used in signal/image processing, see [46, 48, 62], and they become a powerful tool in the numerical optical analysis, as well as in the spectral estimation of the 2D-processes. They are also employed for an optimal antenna synthesis and for resolution enhancing of an optical system. Moreover, FHT

Table 5.13: Example 5.5.6 - Numerical order of convergence.

		$p=4$		$p=6$		$p=8$	
	Mesh	Error	Order	Error	Order	Error	Order
	20	3.67e-02	3.61	5.69e-03	6.10	2.14e-04	8.93
	40	3.00e-03	5.55	8.28e-05	6.80	4.38e-07	9.25
λ_9	60	3.16e-04	6.76	5.27e-06	6.78	1.03e-08	9.43
	80	4.52e-05	9.22	7.49e-07	6.74	6.83e-10	9.54
	100	5.77e-06	-	1.66e-07	-	8.13e-11	-
	100	1.13e-02	1.19	2.81e-03	2.56	1.14e-03	4.97
	200	4.94e-03	4.50	4.75e-04	5.90	3.65e-05	7.93
λ_{24}	300	7.96e-04	5.53	4.33e-05	6.37	1.47e-06	8.59
	400	1.62e-04	6.40	6.94e-06	6.52	1.24e-07	8.91
	500	3.88e-05	-	1.62e-06	-	1.70e-08	-
	500	7.03e-03	2.70	1.72e-03	4.38	4.02e-04	6.37
	1000	1.08e-03	5.06	8.24e-05	6.13	4.85e-06	8.22
λ_{49}	1500	1.39e-04	6.05	6.87e-06	6.42	1.73e-07	8.73
	2000	2.44e-05	7.23	1.08e-06	6.51	1.41e-08	8.97
	2500	4.85e-06	-	2.54e-07	-	1.90e-09	-

Table 5.14: Example 5.5.6 - Mesh selection steps for the estimation of λ_{24} and y_{24} using the order and stepsize variation strategy, we start with order $p = 6$ and exit condition $TOL = 10^{-8}$.

	$p=6 \;\; TOL=10^{-3}$		$p=8 \;\; TOL=10^{-6}$		$p=10 \;\; TOL=10^{-8}$
Mesh	301	340	282	319	464
$E_r(\lambda_{24})$	4.19e−05	3.11e−06	3.15e−08	6.77e−09	1.90e−13
$E_r^{teor}(\lambda_{24})$	4.33e−05	3.15e−06	3.17e−08	6.84e−09	1.83e−13
$E_a(y_{24})$	1.53e−03	1.40e−04	1.26e−06	1.83e−07	6.15e−11
h_{min}	6.67e−03	4.43e−03	4.23e−03	3.21e−03	1.85e−03
h_{max}	6.67e−03	8.41e−03	1.16e−02	1.29e−02	8.69e−03

Table 5.15: Example 5.5.6 - Mesh selection steps for the estimation of λ_{49} and y_{49} using the order and stepsize variation strategy, we start with order $p = 8$ and exit condition $TOL = 10^{-8}$.

	$p=6 \;\; TOL=10^{-3}$		$p=8 \;\; TOL=10^{-6}$		$p=10 \;\; TOL=10^{-8}$
Mesh	601	1290	1290	1345	1632
$E_r(\lambda_{24})$	8.91e−04	3.42e−07	1.15e−08	4.84e−10	8.85e−14
$E_r^{teor}(\lambda_{24})$	1.06e−03	3.54e−07	1.15e−08	4.87e−10	8.78e−14
$E_a(y_{24})$	1.99e−01	9.85e−05	1.19e−06	9.78e−08	4.47e−12
h_{min}	3.33e−03	9.44e−04	9.44e−04	6.86e−04	4.40e−04
h_{max}	3.33e−03	2.03e−03	2.03e−03	2.40e−03	2.36e−03

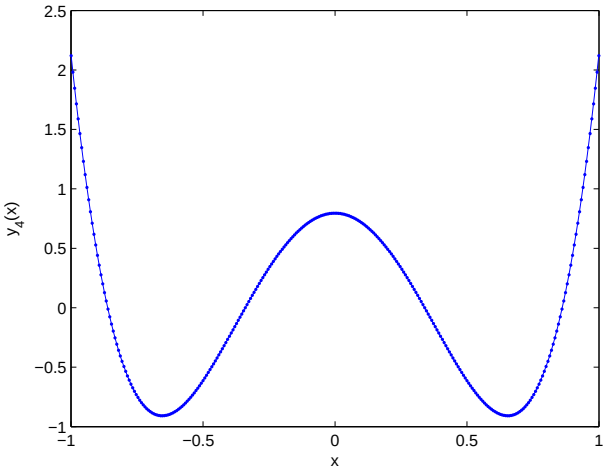


Figure 5.6: Example 5.5.6 - Numerical solution for the eigenfunction $y_4(x)$ obtained with 301 equidistant points and order $p = 6$.

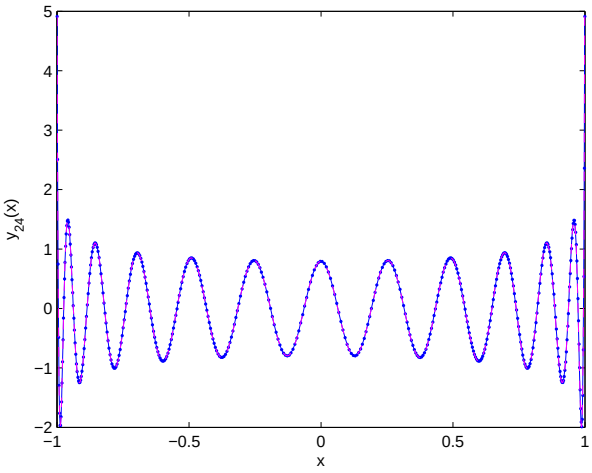


Figure 5.7: Example 5.5.6 - Eigenfunctions $y_{24}(x)$ obtained with order and stepsize variation starting with order 6 and using $TOL = 10^{-3}, 10^{-6}, 10^{-8}$. The dot line is the numerical solution, the continuous line is the theoretical solution.

eigenfunctions serve in medicine and biology for tomographic image reconstruction.

However, these functions having a great relevance in various fields are not often used in practice since they cannot be expressed in a closed form using polynomials and/or standard hypergeometric functions. Consequently, numerical techniques for FHT eigenfunctions evaluation assume great importance for the numerical analysis.

The approach proposed in [13] compute the FHT eigenfunctions as eigenfunctions of a self-adjoint Sturm–Liouville problem. Since the Sturm–Liouville problem is *singular*, its numerical treatment requires additional attention, in fact the boundary conditions have to be formulated in a proper form to guarantee that the resulting eigenvalue problem is well-posed.

The singular self-adjoint Sturm-Liouville problem in [13] is defined as

$$\frac{d}{dx}\left((1-x^2)\frac{d}{dx}y(x)\right)+Q(x)\,y(x)=0,\ Q(x)=\lambda-c^2\,x^2-\frac{m^2-1/4}{x^2},\ (5.5.4)$$

for $x \in I = (0,1)$ and bounded at its ends

$$|y(x)|<\infty,\ x\to 0^+,\quad |y(x)|<\infty,\ x\to 1^-. \tag{5.5.5}$$

Moreover, $\lambda_{m,l}$ for non–trivial solutions $y_{m,l}(x)$ of (5.5.4)-(5.5.5) are the associated eigenvalues, where $\lambda_{m,0} < \lambda_{m,1} < \ldots$ and l is the number of zeros the eigenfunction $y_{m,l}(x)$ has inside I. We also normalize the eigenfunctions by

$$\int_0^1 |y_{m,l}(x)|^2\,dx = 1. \tag{5.5.6}$$

At both ends of the interval I the problem exhibits singularities of the first kind [19, 35, 40], indeed for the boundary conditions we can say that any solution of (5.5.4) which is bounded for $x \to +0$ has the form

$$y(x) = x^{m+1/2}\,W(x), \tag{5.5.7}$$

where $W(x)$ is an analytic function and $W(0) \neq 0$ holds. Consequently

$$y(0) = 0. \tag{5.5.8}$$

Equation (5.5.7) also describes the smoothness of the solution $T(x)$ at $x = 0$ and it is clearly visible as the smoothness of $T(x)$ depends on m. The lack of smoothness in the higher solution derivatives usually causes order reductions

in the numerical methods and consequently, the loss of their efficiency. In order to avoid this loss, one can alternatively transfer the boundary condition (5.5.8) to a nearby point, where the solution $T(x)$ remains an analytic function, this means to consider an interval $[\epsilon, 1]$ with $\epsilon > 0$.

In order to compute the eigenvalues and eigenfunctions of the problem (5.5.4) we follow two approaches. The first one is used when $m > 2$ and no order reduction appears for the singularity in the left endpoint. In this cases the boundary conditions are given by (5.5.5) and the transformed condition is used at the right endpoint as follows

$$2y'(1) \,=\, \beta_1\, y(1) = \, Q(1)\, y(1). \tag{5.5.9}$$

For our methods, (5.5.9) coincides with applying formulae with zero final conditions to discretize the problem (5.5.4) at $x = 1$. The approach taking in consideration (5.5.8)-(5.5.9) and the formulae used for the regular problem in Section 5.2 gives back us an algebraic problem, therefore the first approach is a matrix method. We obtain very efficient results for $m > 2$, $l > 1$ and $c < 200$, and above all in the case $c \gg 1$, e.g. $c = 100$ and $l \sim 1$, see Figure 5.8 and Figure 5.9, where another approach based on the Prüfer angle in [13] seems to fail.

When $0 < m < 3$ the first approach shows an order reduction, since the initial formulae involve higher solution derivatives at zero. For this reason, a transfer boundary condition at zero is considered, so we compute the eigenvalues and eigenfunctions of (5.5.4) in the interval $[\epsilon, 1]$, $\epsilon > 0$. We apply the formulae used for the regular problems in Section 5.2 with the two transferred boundary condition, approximated by initial and final formulae with zero initial and final methods, respectively. We point out that the transferred boundary conditions depend nonlinearly on $\lambda_{m,l}$, then as specified in Section 5.2 we solve a nonlinear problem in which we consider also the normalization condition (5.5.6). The second approach is not a matrix method, but allows us to gain on the numerical order of convergence. In the Figure 5.10 we show the eigenfunction obtained with the second approach for $m = 0$, $l = 3$ and $c = 100$. We underline that the two approaches have been used on a uniform mesh.

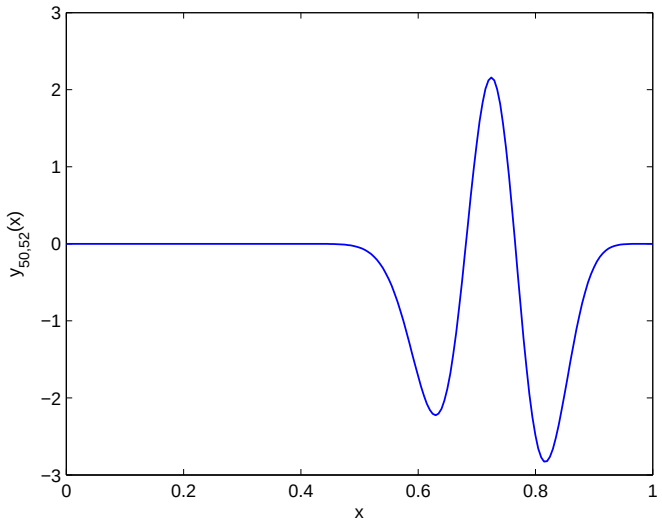


Figure 5.8: Eigenfunction $y_{50,2}$ for $c = 100$ computed with the first approach, using order 6 and 201 equidistant mesh points. $\lambda_{50,2} = 682.9516329$.

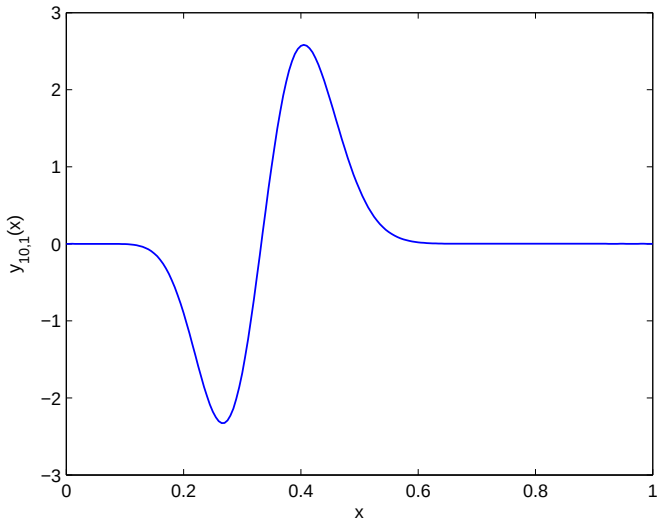


Figure 5.9: Eigenfunction $y_{10,1}$ for $c = 100$ computed with the first approach, using order 6 and 201 equidistant mesh points. $\lambda_{10,2} = -7436.51762$.

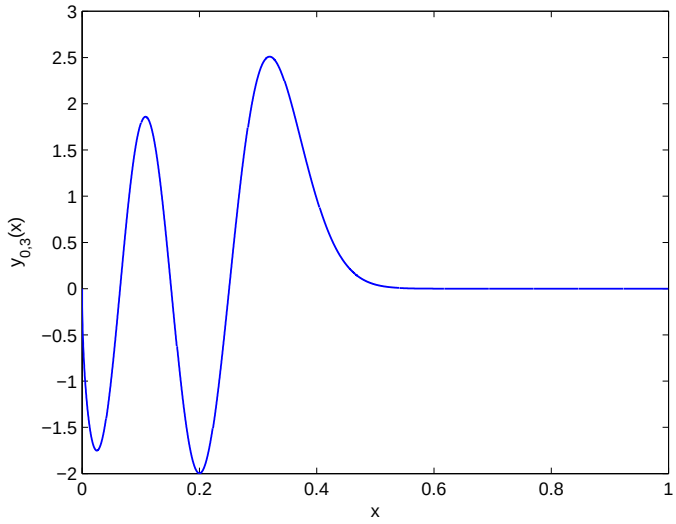


Figure 5.10: Eigenfunction $y_{0,3}$ for $c = 100$ computed with the second approach, using order 6 and 5000 equidistant mesh points. $\lambda_{10,2} = -8625.7278$.

5.6 Conclusion

The Matlab code developed allows us to solve regular and singular Sturm-Liouville problems on equidistant and variable mesh. The stepsize and order variation strategy renders the code more efficient. Since other codes Fortran are able to solve regular and singular SLPs, a comparison with these code seems to be interesting. The advantages of our code are above all connected to the solution of an algebraic problem, moreover the code works well also when a nonlinear system have to be computed.

Chapter 6

Conclusion

The methods HOGD and HOGUP introduced respectively in Chapter 2 and Chapter 3 are the starting point for the Matlab code HOFiD_UP developed to solve singular perturbation problems. We point out that the error equidistribution technique and the deferred correction allow us to obtain a variable mesh. Moreover, the stepsize and order variation strategy, together to the continuation required only for some nonlinear problems, have improved the performance of HOFiD_UP. From the comparison with other code as ACDC and COLMOD the code seems to be competitive, but a Fortran version needs for the comparison and this is the aim for the future.

The methods HOGD have permitted to introduce different classes of methods able to solve initial value problems. These methods are not comparable with the most important codes for easy problems, since the stability region are not so large. However, they are very suitable in the solution of problems which have a decreasing solution which changes rapidly in a narrow region, as in the example Flow in Concrete. For this example numerical results have confirmed some asymptotic properties of the solution, moreover we underline as other codes are not able to solve this problem.

At the end we have on the base of HOGD methods developed a Matlab code which solves regular and singular Sturm-Liouville problems. Stepsize and order variation strategies consent to reduce the number of mesh points and improve the computation of the eigenfunction, above all for the largest eigenvalues, since in this case the eigenfunctions associated have more oscillations. The advantage of this code is in the solution of an algebraic problem, even if this is not a restriction, since the code is able to obtain the eigenvalues and the eigenfunctions associated solving also a nonlinear problem, when it

is necessary. We point out that the comparison of this code with others solving regular and singular SLPs may be very interesting, but again it needs to implement a Fortran version.

Bibliography

[1] L. Aceto and D. Trigiante, *On the A-stable methods in the GBDF class*, Nonlinear Anal. Real World Appl. **3** (1) (2002), 9-23.

[2] L. Aceto, P. Ghelardoni and C. Magherini, *Boundary value methods as an extension of Numerov's method for Sturm-Liouville eigenvalue estimates*, Appl. Numer. Math. **59** (2009), 1644–1656.

[3] L. Aceto, P. Ghelardoni and C. Magherini, *BVMs for Sturm-Liouville eigenvalue estimates with general boundary conditions*, JNAIAM J. Numer. Anal. Ind. Appl. Math. **4** (2009), 113–12.

[4] P. Amodio and L. Brugnano, *The conditioning of Toeplitz band matrices*, Math. Comput. Modelling **23** (10) (1996), 29-42.

[5] P. Amodio, F. Iavernaro, *Symmetric Boundary Value Methods for second order initial and boundary value problems*, Mediterr. J. Math. **3** (2006), 383–398.

[6] P. Amodio and G. Settanni, *Variable step/order generalized upwind methods for the numerical solution of second order singular perturbation problems*, JNAIAM J. Numer. Anal. Ind. Appl. Math. **4** (2009), 65–76.

[7] P. Amodio and G. Settanni, *A deferred correction approach to the solution of singularly perturbed BVPs by high order upwind methods: implementation details*, in: Numerical analysis and applied mathematics - ICNAAM 2009. T.E. Simos, G. Psihoyios and Ch. Tsitouras (eds.), AIP Conf. Proc. **1168**, issue 1 (2009), 711–714.

[8] P. Amodio and G. Settanni, *High order finite difference schemes for the solution of second order initial value problems*, JNAIAM J. Numer. Anal. Ind. Appl. Math. **5** (2010), 3–10.

[9] P. Amodio and G. Settanni, *High order finite difference schemes for the numerical solution of eigenvalue problems for IVPs in ODEs*, in: Numerical analysis and applied mathematics - ICNAAM 2010. T.E. Simos, G. Psihoyios and Ch. Tsitouras (eds.), AIP Conf. Proc. **1281**, issue 1 (2010), 202–204.

[10] P. Amodio and G. Settanni, *A matrix method for the solution of Sturm-Liouville problems*, JNAIAM J. Numer. Anal. Ind. Appl. Math. (2011), in press.

[11] P. Amodio and G. Settanni, *A stepsize variation strategy for the solution of regular Sturm-Liouville problems*, in: Numerical Analysis and Applied Mathematics - ICNAAM 2011. T. E. Simos, G. Psihoyios and Ch. Tsitouras editors, AIP Conf. Proc. **1389**, issue B (2011), 1335–1338.

[12] P. Amodio, Ch. Budd, O. Koch, G. Settanni and E.B. Weinmüller, *Numerical Solution of the Flow in Concrete Problem*, in progress.

[13] P. Amodio, T. Levitina, G. Settanni and E.B. Weinmüller, *On the Calculation of the Finite Hankel Transform Eigenfunctions*, submitted.

[14] P. Amodio and I. Sgura, *High-order finite difference schemes for the solution of second-order BVPs*, J. Comput. Appl. Math. **176** (2005), 59–76.

[15] P. Amodio and I. Sgura, *High order generalized upwind schemes and numerical solution of singular perturbation problems*, BIT **47** (2007), 241-257.

[16] A.L. Andrew and J.W. Paine, *Correction of Numerov's eigenvalue estimates*, Numer. Math. **47** (1985), 289–300.

[17] U. Ascher, J. Christiansen and R.D. Russell, *Algorithm 569: COLSYS: Collocation Software for Boundary-Value ODEs*, ACM Trans. Math. Software **7** (1981), 223–229.

[18] U.M. Ascher, R.M.M. Mattheij and R.D. Russell, *Numerical Solution of Boundary Value Problems for ODEs*, Classics in Applied Mathematics **13**, SIAM, Philadelphia, 1995.

[19] W. Auzinger, E. Karner, O. Koch, and E.B. Weinmüller, *Collocation methods for the solution of eigenvalue problems for singular ordinary differential equations*, Opuscula Math. **26** (2006), 229–241.

[20] P.B. Bailey, *Modified Prüfer transformations*, J. Comp. Phys. **29** (1978), 306–3010.

[21] P.B. Bailey, *SLEIGN: an eigenfunction–eigenvalue code for Sturm–Liouville problems*, SAND77-2044, Sandia Laboratories, Albuquerque (1978).

[22] P.B. Bailey, W.N. Everitt and A. Zettl, *Algorithm 810: the SLEIGN2 Sturm-Liouville Code*, ACM Trans. Math. Software **27** (2001), 143–192.

[23] P.B. Bailey, M.K. Gordon and L.F. Shampine, *Automatic solution of the Sturm-Liouville problem*, ACM Trans. Math. Software **4** (1978), 193–208.

[24] P.B. Bailey, L.F. Shampine and P.E. Waltman, *Nonlinear Two Point Boundary Value Problems*, Academic Press, New York and London, 1968.

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

[26] P.N. Brown, G.D. Byrne and A.C. Hindmarsh, *VODE: A Variable Coefficient ODE Solver*, SIAM J. Sci. Stat. Comput. **10** (1989), 1038–1051.

[27] L. Brugnano and C. Magherini, *Blended Implementation of Block Implicit Methods for ODEs*, Appl. Numer. Math. **42** (2002), 29–45.

[28] L. Brugnano and D. Trigiante, *Solving Differential Problems by Multistep Initial and Boundary Value Methods*, Gordon and Breach Science Publishers, Amsterdam, 1998.

[29] J. Cash and F. Mazzia, *Algorithms for the solution of two-point boundary value problems*,
`http://www.ma.ic.ac.uk/~jcash/BVP_software/twpbvp.php`.

[30] J.R. Cash and F. Mazzia, *A new mesh selection algorithm, based on conditioning, for two-point boundary value codes*, J. Comput. Appl. Math. **184** (2005), 362–381.

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

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

[33] J.R. Cash and M.H. Wright, *Users Guide for TWPBVP: A Code for Solving Two–Point Boundary Value Problems*, `http://www.ma.ic.ac.uk/~jcash/BVP_software/twpbvp/twpbvp.pdf`.

[34] J.R. Cash and R.H. Wright, *User's guide for ACDC: An automatic continuation code for solving stiff two-point boundary value problems*, available at `http://www.ma.ic.ac.uk/~jcash/BVP_software/readme.html`.

[35] F.R. De Hoog and R. Weiss, *Difference methods for boundary value problems with a singularity of the first kind*, SIAM J. Numer. Anal. **13** (1976), 775–813.

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

[37] W.N. Everitt, *A catalogue of Sturm-Liouville differential equations*, in: Sturm-Liouville Theory. Past and Present, W.O. Amrein, A.M. Hinz and D.P. Pearson (eds.), Birkäuser, 2005.

[38] B. Fornberg and M. Ghrist, *Spatial finite difference approximations for wave-type equations*, SIAM J. Numer. Anal. **37** (1) (1999), 105-130.

[39] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems*, Springer-Verlag 2nd Editions 1996.

[40] R. Hammerling, O. Koch, C. Simon, E.B. Weinmüller, *Numerical solution of singular ODE eigenvalue problems in electronic structure computations*, J. Comput. Phys. **181** (2010), 1557–1561.

[41] D. Hiton and P.W. Schaefer, *Spectral Theory and Computational Methods of Sturm-Liouville Problems*, Lecture Notes in Pure and Applied Mathemathics, Marcel Dekker, Inc. New York, 1997.

[42] F. Iavernaro and F. Mazzia, *Solving Ordinary Differential Equations by Generalized Adams Methods: properties and implementation techniques*, Proceedings of NUMDIFF8, Appl. Num. Math. **28** (1998), 107–126.

[43] F. Iavernaro and D. Trigiante, *Preconditioning and conditioning of systems arising from boundary value methods*, Nonlinear Dynamic System Theory **1** (2001), 59-80.

[44] L.Gr. Ixaru, H. De Meyer and G. Vanden Berghe, *SLCPM12 – A program for solving regular Sturm-Liouville problems*, Comput. Phys. Comm. **118** (1999), 259–277.

[45] G. Kitzhofer, O. Koch, G. Pulverer, C. Simon and E. B. Weinmüller , *BVPSUITE1.1 A new matlab solver for singular implicit boundary value problems*, http://www.math.tuwien.ac.at/ ewa/.

[46] B. Larsson, T.V. Levitina, and E.J. Brändas, *On Generalized Prolate Spheroidal Functions*, Proc. CMMSE-2002, Alicante, Spain **II** (2002), 220–223.

[47] V. Ledoux, M. Van Daele and G. Vanden Berghe, *MATSLISE: a MATLAB package for the numerical solution of Sturm-Liouville and Schrdinger equations*, ACM Trans. Math. Software **31** (2005), 532–554.

[48] T.V. Levitina and E.J. Brändas, *Computational Techniques for Prolate Spheroidal Wave Functions in Signal Processing*, J. Comp. Meth. Sci. Eng. **1** (2001), 287–313.

[49] M. Marletta, *Certification of algorithm 700: numerical tests of the SLEIGN software for Sturm-Liouville problems*, ACM Trans. Math. Software **17** (4) (1991), 481–490.

[50] M. Marletta and J.D. Pryce, *LCNO Sturm-Liouville problems: computational difficulties and examples*, Numer. Math. **69** (3) (1995), 303-320.

[51] F. Mazzia, *BVP Software web page*, `http://www.dm.uniba.it/~mazzia/bvp/index.html`.

[52] T. Myuint-U, *Ordinary Differential equations*, North-Holland, 1978.

[53] M.R. Osborne, *A note on finite difference methods for solving the eigenvalue problems of second-order differential equations*, Math. Comp. **16** (1962), 338–346.

[54] J.W. Paine, F.R. De Hoog and R.S. Anderssen, *On the correction of finite difference eigenvalue approxiamtions for Sturm-Liouville problems*, Computing **26** (1981), 123–139.

[55] S. Pruess and C.T. Fulton, *Mathematical software for Sturm-Liouville problems*, ACM Trans. Math. Software **19** (1993), 360–376.

[56] S. Pruess and M. Marletta, *Atomatic solution of Sturm-Liouville problems using the Pruess method*, J. Comput. Appl. Math. **39** (1992), 57–78.

[57] J.D. Pryce, *Numerical solution of Sturm-Liouville problems*. Monographs on Numerical Analysis. Oxford Science Publications. The Clarendon Press, Oxford University Press, New York, 1993.

[58] J.D. Pryce, *A test package for Sturm-Liouville solvers*, ACM Trans. Math. Software **25** (1) (1999), 21-57.

[59] W.T. Reid, *Sturmian Theory for Ordinary Differential Equations*, Springer–Verlag New York Inc., 1980.

[60] L.F. Shampine, M.W. Reichelt and J. Kierzenka, *Solving Boundary Value Problems for Ordinary Differential Equations in MATLAB with bvp4c*, available at `ftp://ftp.mathworks.com/pub/doc/papers/bvp/`.

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

[62] D. Slepian, *Prolate spheroidal wave functions, Fourier analysis and uncertainty, IV: Extensions to many dimensions; generalized prolate spheroidal functions*, Bell System Tech. J. **43** (1964), 3009–3058.