

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche
Scientifique
Université 8 Mai 1945 Guelma
Faculté des Mathématiques et de l'Informatique
et des Sciences de la Matière
Département de Mathématiques



Mémoire

Présenté en vue de l'obtention du diplôme de
Master Académique en Mathématiques
Option : **Equations aux Dérivées Partielles**
Et analyse numérique

Par :

Boutheyna Benabdelhafid

Intitulé

**Shooting method for boundary value problems
in ordinary differential equations**

Dirigé par : Dr. Bahloul Tarek

Devant le jury

PRESIDENT	Dr. BOULARES Hamid	Pr	Univ-Guelma
RAPPORTEUR	Dr. Bahloul Tarek	MCA	Univ-Guelma
EXAMINATEUR	Dr. ALI Ahmed	MCB	Univ-Guelma
EXAMINATEUR	Dr. MENACEUR Amor	MCA	Univ-Guelma

Session Juin 2025

Acknowledgments

*We first thank Almighty Allah for granting us the strength, courage, and willpower necessary to accomplish this modest work. We sincerely thank all those who, in one way or another, contributed to our education and to the realization of this work. We extend our heartfelt thanks to **Professor.Boulares Hamid** , from **University of May 8, 1945 – Guelma**, for graciously agreeing to evaluate this work and serve as the jury president. We also wish to express our appreciation to **Professor.Ali Ahmed and Professor.Amor MENACEUR**, for accepting to examine this work and honoring us with her participation as a jury member. We are truly grateful to everyone who contributed, in any way, to our academic journey and the development of this thesis. In order not to forget anyone, we warmly thank all those both near and far who supported us and played a role in making this modest dissertation a reality and bringing it to light.*

"A heartfelt thank you to all of you."

Dedication

In the name of my Creator, the Facilitator of my affairs and the Guardian of my path, All praise and thanks be to You, my Lord, for all the blessings You have bestowed upon me. I dedicate the fruit of my success to my ambitious self. Today, I continue my steps gently, realizing that nothing meant for me by God will ever miss me.

To my dear father, To the one who taught me that the path to success comes only through patience and perseverance, From whom I drew my strength and self-pride Thank you so much for everything, Dad.

To my beloved mother, To the one beneath whose feet lies Paradise, The invisible hand that cleared all thorns from my path, To the one who stood by me in times of weakness and sorrow My dearest mother, you have all my love.

To my dear siblings: To those who believed in my potential, the safety of my days, My support in life and my unwavering foundation, To those who always stood behind me like my shadow, To those who strengthened me and became springs from which I draw: Siham, Sarah, and Fathi.

To the joy of my heart and the source of my constant happiness,: Idris, Yazan and Meryem.

To the friend of years and companion through hardships: The one who walked with me and reached out her hand in times of weakness Zainab, you have my deepest gratitude and a place in my heart that cannot be touched by forgetfulness.

To all my loved ones, near and far, Thank you for being in my life, each in your own way.

Résumé

Cette étude utilise la méthode de tir unidimensionnelle pour résoudre numériquement un problème aux limites (BVP) gouverné par une équation différentielle ordinaire (EDO). Ces types de problèmes apparaissent dans divers domaines scientifiques et techniques, notamment la conduction thermique, la physique des semi-conducteurs, l'électrochimie, les transferts thermiques, l'élasticité, la thermoélasticité, la physique des plasmas, les matériaux à effets de mémoire et la dynamique des populations.

Mots-clés : Méthode de tir, problèmes aux limites, solutions numériques, EDO.

Abstract

This study employs the one-dimensional shooting method to numerically solve a boundary value problem (BVP) governed by an ordinary differential equation (ODE). Such BVPs arise in diverse scientific and engineering domains, including thermal conduction, semiconductor physics, electrochemistry, heat transfer, elasticity, thermoelasticity, plasma physics, materials with memory effects, and population dynamics.

Keywords: Shooting method, boundary value problems, numerical solutions, ODEs.

ملخص

تستخدم هذه الدراسة الطريقة الرقمية أحادية البعد (طريقة الرمي) لحل مسائل القيمة الحدية المحكومة بمعادلة تفاضلية عادية . تظهر مثل هذه المسائل في مجالات علمية وهندسية متعددة، مثل التوصيل الحراري، فيزياء أشباه الموصلات، الكيمياء الكهربائية، انتقال الحرارة، المرونة، المرونة الحرارية، فيزياء البلازما، المواد ذات تأثيرات الذاكرة، وديناميكيات السكان.

الكلمات المفتاحية

طريقة الرمي، مسائل القيمة الحدية، الحلول العددية، المعادلات التفاضلية العادية.

Contents

1	Introduction	1
2	Numerical Methods	3
2.1	Taylor Methods	3
2.1.1	Algorithm of the Second-Order Taylor Method	5
2.2	Runge-Kutta methods	5
2.3	Runge-Kutta Method (4th Order)	9
2.3.1	Algorithme for Runge-Kutta methods	12
2.4	Implicit Runge-Kutta Methods	13
2.5	Newton's methods	15
2.5.1	Algorithme for Newton methods	16
3	Shooting Method	18
4	Application of the Shooting Method	21
4.1	Corollary	21
4.2	Solving the Boundary Value Problem	23
4.3	System of First Order Initial Value Problems	26

Chapter 1

Introduction

The numerical resolution of differential equations is undoubtedly one of the richest and most dynamic fields in numerical analysis, offering a wide range of applications across various scientific and engineering domains. Whether in fluid mechanics, heat conduction, electromagnetism, or structural analysis, differential equations play a fundamental role in modeling physical phenomena and predicting their behavior under specific conditions.

Among the types of differential equations commonly encountered, initial value problems and boundary value problems are of particular importance. While the former can often be tackled using classic integration techniques, the latter present unique challenges due to the necessity of satisfying constraints at multiple points, typically at the boundaries of the domain. In such cases, analytical solutions are rarely obtainable, which has led to the development and refinement of numerous numerical methods.

This work is particularly focused on the shooting method, a powerful technique for solving boundary value problems associated with ordinary differential equations (ODEs). The shooting method transforms a boundary value problem into an initial value problem by guessing the missing initial condition(s), integrating the ODE, and adjusting the guess iteratively until the boundary condition at the other end is satisfied. Despite its conceptual simplicity, the method is both effective and versatile, especially when combined with robust root-finding algorithms like the secant method or Newton-Raphson method.

To build a comprehensive understanding, we begin by revisiting foundational numerical methods used for solving initial value problems. These include the Taylor method, the fourth-order Runge-Kutta method—well known for its balance between accuracy and computational efficiency—and the secant method, which aids in refining estimates for boundary conditions in the context of shooting. These techniques form the core of the first chapter, providing the necessary mathematical and computational tools for the remainder of the study.

In the second chapter, we introduce the shooting method in detail, elucidating its theoretical foundation and practical implementation. Through a guided example, we demonstrate the step-by-step procedure involved in its application and discuss its advantages and limitations. This method, as developed and analyzed in several key studies [1,2], continues to be widely used in both academic research and industrial simulations, as further illustrated by references [3–8].

The third and final chapter is dedicated to applying the shooting method

to a specific and illustrative case: Solving the boundary value problem. This classic example not only showcases the utility of the shooting method but also provides an opportunity to compare the numerical solution obtained with those derived from alternative analytical or approximate approaches. The comparative analysis highlights the precision and adaptability of the method, particularly in nonlinear contexts where analytical solutions may not exist or may be intractable.

The overarching goal of this memoir is to provide a structured and practical pathway toward mastering one of the fundamental techniques in numerical differential equation solving. Beyond the theoretical exposition, our emphasis on examples and implementation serves to bridge the gap between abstract mathematics and real-world applications.

In summary, this work is organized as follows:

- Chapter 2 presents foundational numerical methods for initial value problems.
- Chapter 3 introduces and explains the shooting method in detail.
- Chapter 4 applies the shooting method to the pendulum problem, with analysis and comparison of results.

Through this journey, we aim to demonstrate not only the mathematical rigor behind these techniques but also their practical relevance and effectiveness in solving complex boundary value problems.

Chapter 2

Numerical Methods

In this chapter, we present four numerical methods: the Taylor method and the Runge-Kutta method for solving differential equations, and the Newton method for finding the roots of nonlinear equations. These methods are used to obtain approximate solutions when analytical approaches are difficult or impossible.

2.1 Taylor Methods

We will limit ourselves, however, to the second-order Taylor method. We aim to approximate the solution at time $t = t_{n+1}$, given the value at time $t = t_n$. We immediately have:

$$\begin{aligned} y(t_{n+1}) &= y(t_n + h) \\ y(t_{n+1}) &= y(t_n) + y'(t_n)h + \frac{y''(t_n)h^2}{2} + O(h^3) \end{aligned}$$

Using the differential equation:

$$\begin{cases} y'(t) = f(t, y(t)) \\ y(t_0) = y_0 \end{cases}$$

we find:

$$y(t_{n+1}) = y(t_n) + f(t_n, y(t_n))h + \frac{f'(t_n, y(t_n))h^2}{2} + O(h^3)$$

In the previous expression, the derivative of the function $f(t, y(t))$ with respect to time appears. The chain rule gives us:

$$f'(t, y(t)) = \frac{\partial f(t, y(t))}{\partial t} + \frac{\partial f(t, y(t))}{\partial y} \cdot y'(t)$$

that is:

$$f'(t, y(t)) = \frac{\partial f(t, y(t))}{\partial t} + \frac{\partial f(t, y(t))}{\partial y} \cdot f(t, y(t))$$

Therefore, we obtain:

$$y(t_{n+1}) = y(t_n) + hf(t_n, y(t_n)) + \frac{h^2}{2} \left(\frac{\partial f(t_n, y(t_n))}{\partial t} + \frac{\partial f(t_n, y(t_n))}{\partial y} \cdot f(t_n, y(t_n)) \right) + O(h^3)$$

By neglecting terms of order 3 and higher, we arrive at the formula:

$$y(t_{n+1}) \approx y(t_n) + hf(t_n, y(t_n)) + \frac{h^2}{2} \left(\frac{\partial f(t_n, y(t_n))}{\partial t} + \frac{\partial f(t_n, y(t_n))}{\partial y} \cdot f(t_n, y(t_n)) \right)$$

which forms the basis of the Taylor method

Example 2.1.1

Find the solution of the boundary value problem

$$u'' = u + x, \quad x \in [0, 1]$$

$$u(0) = 0, \quad u(1) = 0$$

using the shooting method. Use the fourth order Taylor series method to solve the initial value problem with $h = 0.2$. We take $u'(0) = \frac{1}{2}$. The Taylor series method gives

$$\begin{aligned} u_{n+1} &= u_n + hu'_n + \frac{h^2}{2}u''_n + \frac{h^3}{6}u'''_n + \frac{h^4}{24}u^{(4)}_n \\ &= u_n + hu'_n + \frac{h^2}{2}(u_n + x_n) + \frac{h^3}{6}(u'_n + 1) + \frac{h^4}{24}(u_n + x_n) \\ &= \left(1 + \frac{h^2}{2} + \frac{h^4}{24}\right)u_n + \left(h + \frac{h^3}{6}\right)u'_n + \left(\frac{h^2}{2} + \frac{h^4}{24}\right)x_n + \frac{h^3}{6} \\ u'_{n+1} &= u'_n + hu''_n + \frac{h^2}{2}u'''_n + \frac{h^3}{6}u^{(4)}_n \\ &= \left(1 + \frac{h^2}{2}\right)u'_n + \left(h + \frac{h^3}{6}\right)u_n + \left(h + \frac{h^3}{6}\right)x_n + \frac{h^2}{2} \end{aligned}$$

With $h = 0.2$, we have

$$u_{n+1} = 0.00133 + 0.02007x_n + 1.02007u_n + 0.20133u'_n$$

$$u'_{n+1} = 0.02 + 0.20133x_n + 0.20133u_n + 1.02u'_n$$

We get:

$$\begin{aligned} u(0.2) &\approx u_2 = 0.10200, & u'(0.2) &\approx u'_2 = 0.53000 \\ u(0.4) &\approx u_3 = 0.21610, & u'(0.4) &\approx u'_3 = 0.62140 \\ u(0.6) &\approx u_4 = 0.35490, & u'(0.6) &\approx u'_4 = 0.77787 \\ u(0.8) &\approx u_5 = 0.53200, & u'(0.8) &\approx u'_5 = 1.00568 \\ u(1.0) &\approx u_6 = 0.76254, & u'(1.0) &\approx u'_6 = 1.31397 \end{aligned}$$

Let the second choice of the initial slope be $u'(0) = -\frac{1}{2}$. We get:

$$\begin{aligned} u(0.2) &\approx u_2 = -0.09934, & u'(0.2) &\approx u'_2 = -0.49000 \\ u(0.4) &\approx u_3 = -0.19464, & u'(0.4) &\approx u'_3 = -0.45953 \\ u(0.6) &\approx u_4 = -0.28171, & u'(0.6) &\approx u'_4 = -0.40738 \\ u(0.8) &\approx u_5 = -0.35601, & u'(0.8) &\approx u'_5 = -0.33145 \\ u(1.0) &\approx u_6 = -0.41250, & u'(1.0) &\approx u'_6 = -0.22869 \end{aligned}$$

We have:

$$c_2 = \frac{0 - 0.76254}{-1.17504} = 0.64895, \quad c_1 = 1 - c_2 = 0.35105$$

Hence,

$$u(x) = 0.35105u_1(x) + 0.64895u_2(x)$$

We find:

$$u(0.2) = 0.35105u_1(0.2) + 0.64895u_2(0.2) = 0.35105(0.10200) + 0.64895(-0.09934) \approx -0.02866$$

$$u(0.4) = 0.35105u_1(0.4) + 0.64895u_2(0.4) = 0.35105(0.21610) + 0.64895(-0.19464) \approx -0.05045$$

Similarly, we get:

$$u(0.6) \approx -0.05823, \quad u(0.8) \approx -0.04427, \quad u(1.0) \approx -0.000002$$

The exact solution is:

$$u(0.2) = -0.02868, \quad u(0.4) = -0.05048, \quad u(0.6) = -0.05826$$

$$u(0.8) = -0.04429, \quad u(1.0) = 0$$

2.1.1 Algorithm of the Second-Order Taylor Method

1. Given:

- a time step h ,
- an initial condition (t_0, y_0) ,
- and a maximum number of iterations N .

2. For $0 \leq n \leq N$:

$$y_{n+1} = y_n + hf(t_n, y_n) + \frac{h^2}{2} \left(\frac{\partial f(t_n, y_n)}{\partial t} + \frac{\partial f(t_n, y_n)}{\partial y} f(t_n, y_n) \right)$$

$$t_{n+1} = t_n + h$$

Print or record t_{n+1} and y_{n+1} .

3. Stop.

2.2 Runge-Kutta methods

We first explain the principle involved in the Runge-Kutta methods. By the Mean Value Theorem any solution of

$$u' = f(t, u), \quad u(t_0) = \eta_0, \quad t \in [t_0, b]$$

satisfies

$$u(t_{j+1}) = u(t_j) + hu'(t_j + \theta h) = u(t_j) + hf(t_j + \theta h, u(t_j + \theta h)), \quad 0 < \theta < 1$$

For $\theta = \frac{1}{2}$, we have

$$u(t_{j+1}) = u(t_j) + hf \left(t_j + \frac{h}{2}, u \left(t_j + \frac{h}{2} \right) \right)$$

Euler's method with spacing $h/2$ gives

$$u \left(t_j + \frac{h}{2} \right) \approx u_j + \frac{h}{2} f_j$$

Thus, we have the approximation

$$u_{j+1} = u_j + hf \left(t_j + \frac{h}{2}, u_j + \frac{h}{2} f_j \right)$$

which may be written as

$$\begin{aligned} K_1 &= hf_j \\ K_2 &= hf \left(t_j + \frac{h}{2}, u_j + \frac{1}{2} K_1 \right) \\ u_{j+1} &= u_j + K_2 \end{aligned} \tag{2.2.1}$$

Alternatively, again using Euler's method, we proceed as follows:

$$u' \left(t_j + \frac{h}{2} \right) \approx \frac{1}{2} (u'(t_j) + u'(t_j + h)) \approx \frac{1}{2} [f(t_j, u_j) + f(t_j + h, u_j + hf_j)]$$

and thus we have the approximation

$$u_{j+1} = u_j + \frac{h}{2} [f(t_j, u_j) + f(t_j + h, u_j + hf_j)] \tag{2.2.2}$$

which may be written as

$$\begin{aligned} K_1 &= hf(t_j, u_j) \\ K_2 &= hf(t_j + h, u_j + K_1) \\ u_{j+1} &= u_j + \frac{1}{2}(K_1 + K_2) \end{aligned} \tag{2.2.3}$$

This method is also called **Euler-Cauchy method**.

Either (2.2.2) or (2.2.3) can be regarded as

$$u_{j+1} = u_j + h(\text{average slope}) \tag{2.2.4}$$

This is the underlying idea of the Runge-Kutta approach. In general, we find the slope at t_j and at several other points, average these slopes, multiply by h and add the result to u_j . Thus the **Runge-Kutta method with v slopes** can be written as

$$\begin{aligned} K_1 &= hf(t_j, u_j) \\ K_2 &= hf(t_j + c_2 h, u_j + a_{21} K_1) \\ K_3 &= hf(t_j + c_3 h, u_j + a_{31} K_1 + a_{32} K_2) \\ K_4 &= hf(t_j + c_4 h, u_j + a_{41} K_1 + a_{42} K_2 + a_{43} K_3) \\ &\vdots \\ K_v &= hf \left(t_j + c_v h, u_j + \sum_{i=1}^{v-1} a_{vi} K_i \right) \end{aligned}$$

and

$$u_{j+1} = u_j + W_1 K_1 + W_2 K_2 + \cdots + W_v K_v \quad (2.2.5)$$

From (2.2.5), we may interpret the increment function as the linear combination of the slopes at t_j and at several other points between t_j and t_{j+1} . Further, knowing the values of the quantities on the right-hand side of (2.2.5), the solution value u_{j+1} may be obtained directly. Thus, (2.2.5) represents the **explicit Runge-Kutta method with v slopes**. To determine the parameters c 's, a 's and w 's in (2.2.5), we expand u_{j+1} in powers of h such that it agrees with the Taylor series expansion of the solution of the differential equation up to a certain number of terms.

Consider the following Runge-Kutta method with two slopes:

$$\begin{aligned} K_1 &= hf(t_j, u_j) \\ K_2 &= hf(t_j + c_2 h, u_j + a_{21} K_1) \\ u_{j+1} &= u_j + W_1 K_1 + W_2 K_2 \end{aligned} \quad (2.2.6)$$

where the parameters c_2, a_{21}, W_1 and W_2 are chosen to make u_{j+1} closer to $u(t_{j+1})$. Now Taylor's series gives

$$\begin{aligned} u(t_{j+1}) &= u(t_j) + hu'(t_j) + \frac{h^2}{2!}u''(t_j) + \frac{h^3}{3!}u'''(t_j) + \cdots \\ &= u(t_j) + hf(t_j, u(t_j)) + \frac{h^2}{2!}(f_t + f f_u)_t + \frac{h^3}{3!}[f_{tt} + 2f_t f_u + f^2 f_{uu} + f_u(f_t + f f_u)]_t + \cdots \end{aligned} \quad (2.2.7)$$

We also have

$$\begin{aligned} K_1 &= hf_j \\ K_2 &= hf(t_j + c_2 h, u_j + a_{21} f_j) = hf_j + h(c_2 f_t + a_{21} f f_u)_t + \frac{h^2}{2}(c_2^2 f_{tt} + 2c_2 a_{21} f_t f_u + a_{21}^2 f^2 f_{uu})_t + \cdots \end{aligned}$$

Substituting the values of K_1 and K_2 in (2.2.6), we get

$$u_{j+1} = u_j + (W_1 + W_2)hf_j + h^2(W_2 c_2 f_t + W_2 a_{21} f f_u)_t + \frac{h^3}{2}W_2(c_2^2 f_{tt} + 2c_2 a_{21} f_t f_u + a_{21}^2 f^2 f_{uu})_t + \cdots \quad (2.2.8)$$

Comparing the coefficients of various powers of h in (2.2.7) and (2.2.8), we obtain

$$\begin{aligned} W_1 + W_2 &= 1 \\ c_2 W_2 &= \frac{1}{2} \\ a_{21} W_2 &= \frac{1}{2} \end{aligned}$$

The solution of this system is

$$a_{21} = c_2, \quad W_2 = \frac{1}{2c_2}, \quad W_1 = 1 - \frac{1}{2c_2} \quad (2.2.9)$$

where $c_2 \neq 0$ is arbitrary. Substituting (2.2.9) in (2.2.8), we get

$$u_{j+1} = u_j + hf_j + \frac{h^2}{2}(f_t + f f_u)_t + \frac{h^3 c_2}{4}(f_{tt} + 2f_t f_u + f^2 f_{uu})_t + \cdots \quad (2.2.10)$$

The local truncation error is given by

$$T_{j+1} = u(t_{j+1}) - u_{j+1} = h^3 \left[\left(\frac{1}{6} - \frac{c_2}{4} \right) (f_{tt} + 2f_t f_u + f^2 f_{uu})_t + \frac{1}{6} (f_u(f_t + f f_u))_t + \dots \right] \quad (2.2.11)$$

which shows that the method (2.2.6) is of second order. The free parameter c_2 is usually taken between 0 and 1. Sometimes c_2 is chosen such that one of the W 's in the method (2.2.6) is zero or the truncation error is minimized. Such a formula is called an *optimal* formula.

It may be noted that every Runge-Kutta method should reduce to a quadrature formula when $f(t, u)$ is independent of u with W 's as weights and c 's as abscissas.

If $c_2 = \frac{1}{2}$, we get

$$u_{j+1} = u_j + hf \left(t_j + \frac{h}{2}, u_j + \frac{h}{2} f_j \right)$$

which is the Euler's method with spacing $h/2$. It reduces to the mid-point quadrature rule when $f(t, u)$ is independent of u .

For $c_2 = 1$, we get

$$u_{j+1} = u_j + \frac{h}{2} [f(t_j, u_j) + f(t_j + h, u_j + hf_j)]$$

which reduces to the trapezoidal rule when $f(t, u)$ is independent of u .

For $c_2 = \frac{2}{3}$, the truncation error is minimum. We have the optimal method

$$u_{j+1} = u_j + \frac{1}{4} h f_j + \frac{3}{4} h f \left(t_j + \frac{2}{3} h, u_j + \frac{2}{3} f_j \right) \quad (2.2.12)$$

We note that the second order method (2.2.6) requires two function evaluations for each step of integration. Similarly, we find that the third and fourth order methods require three and four function evaluations respectively for each step of integration. However, for $5 \leq v \leq 7$, the order of the methods becomes $v - 1$ only. For $v \geq 8$, the order of the method reduces further to $v - 2$.

The following two equations occur typically in all Runge-Kutta methods of the form (2.2.5)

$$c_i = \sum_{j=1}^{i-1} a_{ij}, \quad i = 2, 3, \dots, v$$

and

$$\sum_{j=1}^v W_j = 1$$

The number of unknown parameters are then $v(v + 1)/2$.

We now list the second, third and fourth order Runge-Kutta methods.

Second Order Methods

c_2	a_{21}	W_1	W_2
$\frac{1}{2}$	$\frac{1}{2}$	0	1
1	1	$\frac{1}{2}$	$\frac{1}{2}$
$\frac{2}{3}$	$\frac{2}{3}$	$\frac{1}{4}$	$\frac{3}{4}$
			(Optimal)

Third Order Methods

$$\begin{array}{c|ccc}
 c_2 & a_{21} & & \\
 c_3 & a_{31} & a_{32} & \\
 \hline
 & W_1 & W_2 & W_3
 \end{array}$$

$$\begin{array}{c|ccc}
 \frac{1}{2} & \frac{1}{3} & & \\
 \frac{3}{4} & 0 & \frac{2}{3} & \\
 \frac{3}{4} & \frac{1}{4} & 0 & \frac{3}{4}
 \end{array}
 \quad \text{Nyström}$$

$$\begin{array}{c|ccc}
 \frac{1}{2} & \frac{1}{2} & & \\
 \frac{3}{4} & 0 & \frac{3}{4} & \\
 \frac{4}{4} & \frac{2}{9} & \frac{1}{3} & \frac{4}{9}
 \end{array}
 \quad \text{Nearly optimal}$$

$$\begin{array}{c|ccc}
 1 & -1 & 2 & \\
 & \frac{1}{6} & \frac{4}{6} & \frac{1}{6}
 \end{array}
 \quad \text{Classical}$$

$$\begin{array}{c|ccc}
 \frac{1}{2} & 0 & & \\
 \frac{3}{4} & \frac{2}{3} & 0 & \\
 \frac{3}{4} & \frac{1}{4} & 0 & \frac{3}{4}
 \end{array}
 \quad \text{Heun}$$

Fourth Order Methods

$$\begin{array}{c|cccc}
 c_2 & a_{21} & & & \\
 c_3 & a_{31} & a_{32} & & \\
 c_4 & a_{41} & a_{42} & a_{43} & \\
 \hline
 & W_1 & W_2 & W_3 & W_4
 \end{array}$$

$$\begin{array}{c|ccc}
 \frac{1}{2} & \frac{1}{2} & & \\
 \frac{1}{2} & 0 & \frac{1}{2} & \\
 \frac{1}{2} & 0 & 0 & 1 \\
 \hline
 & \frac{1}{6} & \frac{2}{6} & \frac{2}{6} & \frac{1}{6}
 \end{array}
 \quad \text{Classical}$$

$$\begin{array}{c|ccc}
 \frac{1}{3} & \frac{1}{3} & & \\
 \frac{3}{4} & 1 & -1 & 1 \\
 \frac{3}{4} & \frac{1}{8} & \frac{3}{8} & \frac{3}{8} & \frac{1}{8}
 \end{array}
 \quad \text{Kutta}$$

2.3 Runge-Kutta Method (4th Order)

$$\frac{du}{dt} = f(t, u), \quad u(t_0) = \eta$$

may be written as:

$$\mathbf{u}_{j+1} = \mathbf{u}_j + \frac{1}{6}(\mathbf{K}_1 + 2\mathbf{K}_2 + 2\mathbf{K}_3 + \mathbf{K}_4)$$

where

$$\mathbf{K}_1 = \begin{bmatrix} K_{11} \\ K_{21} \\ \vdots \\ K_{n1} \end{bmatrix}, \quad \mathbf{K}_2 = \begin{bmatrix} K_{12} \\ K_{22} \\ \vdots \\ K_{n2} \end{bmatrix}, \quad \mathbf{K}_3 = \begin{bmatrix} K_{13} \\ K_{23} \\ \vdots \\ K_{n3} \end{bmatrix}, \quad \mathbf{K}_4 = \begin{bmatrix} K_{14} \\ K_{24} \\ \vdots \\ K_{n4} \end{bmatrix}$$

and

$$K_{i1} = hf(t_j, u_{1,j}, u_{2,j}, \dots, u_{n,j})$$

$$K_{i2} = hf\left(t_j + \frac{h}{2}, u_{1,j} + \frac{1}{2}K_{11}, u_{2,j} + \frac{1}{2}K_{21}, \dots, u_{n,j} + \frac{1}{2}K_{n1}\right)$$

$$K_{i3} = hf\left(t_j + \frac{h}{2}, u_{1,j} + \frac{1}{2}K_{12}, u_{2,j} + \frac{1}{2}K_{22}, \dots, u_{n,j} + \frac{1}{2}K_{n2}\right)$$

$$K_{i4} = hf(t_j + h, u_{1,j} + K_{13}, u_{2,j} + K_{23}, \dots, u_{n,j} + K_{n3}), \quad i = 1(1)n$$

In explicit form:

$$\begin{bmatrix} u_{1,j+1} \\ u_{2,j+1} \\ \vdots \\ u_{n,j+1} \end{bmatrix} = \begin{bmatrix} u_{1,j} \\ u_{2,j} \\ \vdots \\ u_{n,j} \end{bmatrix} + \frac{1}{6} \left(\begin{bmatrix} K_{11} \\ K_{21} \\ \vdots \\ K_{n1} \end{bmatrix} + 2 \begin{bmatrix} K_{12} \\ K_{22} \\ \vdots \\ K_{n2} \end{bmatrix} + 2 \begin{bmatrix} K_{13} \\ K_{23} \\ \vdots \\ K_{n3} \end{bmatrix} + \begin{bmatrix} K_{14} \\ K_{24} \\ \vdots \\ K_{n4} \end{bmatrix} \right)$$

Example 2.3.1

Solve the initial value problem:

$$u' = -2tu^2, \quad u(0) = 1$$

with $h = 0.2$ on the interval $[0, 1]$. Use the fourth-order classical Runge-Kutta method.

For $j = 0$:

$$t_0 = 0, \quad u_0 = 1$$

$$K_1 = hf(t_0, u_0) = -2(0)(1)^2 = 0$$

$$K_2 = hf\left(t_0 + \frac{h}{2}, u_0 + \frac{1}{2}K_1\right) = -2\left(\frac{0.2}{2}\right)(1)^2 = -0.04$$

$$K_3 = hf\left(t_0 + \frac{h}{2}, u_0 + \frac{1}{2}K_2\right) = -2\left(\frac{0.2}{2}\right)(0.98)^2 = -0.038416$$

$$K_4 = hf(t_0 + h, u_0 + K_3) = -2(0.2)(0.961584)^2 = -0.0739715$$

$$u(0.2) \approx u_1 = 1 + \frac{1}{6}(0 - 0.08 - 0.076832 - 0.0739715) = 0.9615328$$

For $j = 1$:

$$t_1 = 0.2, \quad u_1 = 0.9615328$$

$$K_1 = hf(t_1, u_1) = -2(0.2)(0.9615328)^2 = -0.0739636$$

$$K_2 = hf\left(t_1 + \frac{h}{2}, u_1 + \frac{K_1}{2}\right) = -2(0.2)(0.3)(0.924551)^2 = -0.1025754$$

$$K_3 = hf\left(t_1 + \frac{h}{2}, u_1 + \frac{K_2}{2}\right) = -2(0.2)(0.3)(0.9102451)^2 = -0.0994255$$

$$K_4 = hf(t_1 + h, u_1 + K_3) = -2(0.2)(0.4)(0.8621073)^2 = -0.1189166$$

$$u(0.4) \approx u_2 = 0.9615328 + \frac{1}{6}(-0.0739636 - 2(0.1025754) - 2(0.0994255) - 0.1189166)$$

$$= 0.9615328 + \frac{1}{6}(-0.0739636 - 0.2051508 - 0.1988510 - 0.1189166) = 0.8620525$$

Similarly, we get:

$$u(0.6) \approx u_3 = 0.7352784$$

$$u(0.8) \approx u_4 = 0.6097519$$

$$u(1.0) \approx u_5 = 0.5000073$$

Example 2.3.2

Solve the system of equations:

$$u' = -3u + 2v, \quad u(0) = 0$$

$$v' = 3u - 4v, \quad v(0) = \frac{1}{2}$$

with $h = 0.2$ on the interval $[0, 1]$. Use the Euler-Cauchy method.

For $j = 0$:

$$t_0 = 0, \quad u_0 = 0, \quad v_0 = 0.5$$

$$K_{11} = hf_1(t_0, u_0, v_0) = 0.2(-3 \cdot 0 + 2 \cdot 0.5) = 0.2$$

$$K_{21} = hf_2(t_0, u_0, v_0) = 0.2(3 \cdot 0 - 4 \cdot 0.5) = -0.4$$

$$K_{12} = hf_1(t_0 + h, u_0 + K_{11}, v_0 + K_{21}) = 0.2[-3(0 + 0.2) + 2(0.5 - 0.4)] = -0.08$$

$$K_{22} = hf_2(t_0 + h, u_0 + K_{11}, v_0 + K_{21}) = 0.2[3(0 + 0.2) - 4(0.5 - 0.4)] = 0.04$$

$$u(0.2) \approx u_1 = u_0 + \frac{1}{2}(K_{11} + K_{12}) = 0.06$$

$$v(0.2) \approx v_1 = v_0 + \frac{1}{2}(K_{21} + K_{22}) = 0.32$$

For $j = 1$:

$$t_1 = 0.2, \quad u_1 = 0.06, \quad v_1 = 0.32$$

$$K_{11} = hf_1(t_1, u_1, v_1) = 0.2(-3 \cdot 0.06 + 2 \cdot 0.32) = 0.092$$

$$K_{21} = hf_2(t_1, u_1, v_1) = 0.2(3 \cdot 0.06 - 4 \cdot 0.32) = -0.22$$

$$K_{12} = hf_1(t_1 + h, u_1 + K_{11}, v_1 + K_{21}) = 0.2[-3(0.06 + 0.092) + 2(0.32 - 0.22)] = -0.0512$$

$$K_{22} = hf_2(t_1 + h, u_1 + K_{11}, v_1 + K_{21}) = 0.2[3(0.06 + 0.092) - 4(0.32 - 0.22)] = 0.0112$$

$$u(0.4) \approx u_2 = u_1 + \frac{1}{2}(K_{11} + K_{12}) = 0.0804$$

$$v(0.4) \approx v_2 = v_1 + \frac{1}{2}(K_{21} + K_{22}) = 0.2156$$

Similarly, we get:

$$u(0.6) \approx u_3 = 0.082152, \quad v(0.6) \approx v_3 = 0.152456$$

$$u(0.8) \approx u_4 = 0.079309, \quad v(0.8) \approx v_4 = 0.112359$$

$$u(1.0) \approx u_5 = 0.069000, \quad v(1.0) \approx v_5 = 0.086190$$

2.3.1 Algorithm for Runge-Kutta methods

Given the initial value problem:

$$u' = f(t, u), \quad u(t_0) = u_0$$

with step size h , the algorithm proceeds as follows:

1. Initialize: $t = t_0, u = u_0$
2. For each step $j = 0, 1, 2, \dots$:

$$K_1 = h \cdot f(t_j, u_j)$$

$$K_2 = h \cdot f\left(t_j + \frac{h}{2}, u_j + \frac{1}{2}K_1\right)$$

$$u_{j+1} = u_j + K_2$$

$$t_{j+1} = t_j + h$$

2.4 Implicit Runge-Kutta Methods

The implicit Runge-Kutta method using v slopes is defined as:

$$\begin{aligned} K_i &= hf \left(t_j + c_i h, u_j + \sum_{m=1}^v a_{im} K_m \right) \\ u_{j+1} &= u_j + \sum_{m=1}^v W_m K_m \end{aligned} \quad (2.4.1)$$

where

$$c_i = \sum_{j=1}^v a_{ij}, \quad i = 1, 2, \dots, v$$

and a_{ij} , $1 \leq i, j \leq v$, $W_1, W_2, \dots, W_v$ are arbitrary parameters. The slopes K_m are defined implicitly. The number of unknown parameters are $v(v+1)$.

We now give the derivation for the case $v = 1$. We have:

$$\begin{aligned} K_1 &= hf(t_j + c_1 h, u_j + a_{11} K_1) \\ u_{j+1} &= u_j + W_1 K_1 \end{aligned} \quad (2.4.2)$$

The Taylor series gives

$$u(t_{j+1}) = u(t_j) + hu'(t_j) + \frac{h^2}{2}u''(t_j) + \dots = u(t_j) + hf(t_j, u(t_j)) + \frac{h^2}{2}(f_t + ff_u)_{t_j} + \dots$$

and

$$\begin{aligned} K_1 &= hf(t_j, u_j) + c_1 hf_t + a_{11} K_1 f_u + \dots = (hf + c_1 h^2 f_t + a_{11} hf K_1 f_u) + O(h^3) \\ &= hf + h^2(c_1 f_t + a_{11} f f_u)_{t_j} + O(h^3) \end{aligned} \quad (2.4.3)$$

Substituting (2.4.2) into (2.4.3) and comparing the coefficients of h and h^2 , we get

$$c_1 = a_{11}, \quad W_1 = 1, \quad W_1 c_1 = \frac{1}{2}$$

We obtain

$$W_1 = 1, \quad c_1 = a_{11} = \frac{1}{2}$$

The second order implicit Runge-Kutta method becomes

$$\begin{aligned} K_1 &= hf(t_j + \frac{1}{2}h, u_j + \frac{1}{2}K_1) \\ u_{j+1} &= u_j + K_1 \end{aligned} \quad (2.4.4)$$

For $v = 2$, the implicit Runge-Kutta method (2.4.1) becomes

$$\begin{aligned} K_1 &= hf(t_j + c_1 h, u_j + a_{11} K_1 + a_{12} K_2) \\ K_2 &= hf(t_j + c_2 h, u_j + a_{21} K_1 + a_{22} K_2) \\ u_{j+1} &= u_j + W_1 K_1 + W_2 K_2 \end{aligned} \quad (2.4.5)$$

where the parameter values

$$\begin{aligned} W_1 &= \frac{1}{2}, & W_2 &= \frac{1}{2} \\ c_1 &= \frac{3 + \sqrt{3}}{6}, & c_2 &= \frac{3 - \sqrt{3}}{6} \\ a_{11} &= \frac{1}{4}, & a_{12} &= \frac{1}{4} + \frac{\sqrt{3}}{6} \\ a_{21} &= \frac{1}{4} - \frac{\sqrt{3}}{6}, & a_{22} &= \frac{1}{4} \end{aligned} \tag{2.4.6}$$

lead to a fourth order method.

Example 2.4.1

Solve the initial value problem

$$u' = -2tu^2, \quad u(0) = 1$$

with $h = 0.2$ on the interval $[0, 1]$. Use the second order implicit Runge-Kutta method. The second order implicit Runge-Kutta method is given by

$$u_{j+1} = u_j + K_1, \quad j = 0, 1, 2, 3, 4$$

$$K_1 = hf \left(t_j + \frac{h}{2}, u_j + \frac{1}{2}K_1 \right)$$

which gives

$$K_1 = -h(2t_j + h) \left(u_j + \frac{1}{2}K_1 \right)^2$$

or

$$h(2t_j + h)K_1^2 + 4(1 + hu_j(2t_j + h))K_1 + 4h(2t_j + h)u_j^2 = 0$$

Solving it as a quadratic in K_1 , we get

$$K_1 = \frac{-4(1 + hu_j(2t_j + h)) + \sqrt{16 + 32hu_j(2t_j + h)}}{2h(2t_j + h)}$$

For $j = 0, t_0 = 0, u_0 = 1$,

$$K_1 = \frac{-4(1 + .2 \cdot (2 \cdot 0)) + \sqrt{16 + 32 \cdot (.2) \cdot (2 \cdot 0)}}{2 \cdot (.2) \cdot 2} = -0.0384759$$

$$u(0.2) \approx u_1 = u_0 + K_1 = 1 - 0.0384759 = 0.9615241$$

For $j = 1, t_1 = 0.2, u_1 = 0.9615241$,

$$K_1 = \frac{-4(1 + .12 \cdot 0.9615241) + \sqrt{16 + 3.84 \cdot 0.9615241}}{0.24} = -0.0997344$$

$$u(0.4) \approx u_2 = u_1 + K_1 = 0.9615241 - 0.0997344 = 0.8617897$$

Similarly, we get

$$u(0.6) \approx u_3 = 0.7343987$$

$$u(0.8) \approx u_4 = 0.6082158$$

$$u(1.0) \approx u_5 = 0.4980681$$

2.5 Newton's methods

Newton's **method** can be motivated by the mean value theorem. Let x_{n-1} approximate the root x_∞ of the equation $g(x) = 0$. According to the mean value theorem,

$$g(x_{n-1}) = \frac{g(x_{n-1}) - g(x_\infty)}{g'(z)} = g'(z)(x_{n-1} - x_\infty)$$

For some z on the interval between x_{n-1} and x_∞ . If we substitute x_{n-1} for z and the next approximant x_n for x_∞ , then this equality can be rearranged to provide the definition

$$x_n = x_{n-1} - \frac{g(x_{n-1})}{g'(x_{n-1})} \quad (2.5.1)$$

of Newton's **method**. From the perspective of functional iteration, Newton's **method** can be rephrased as $x_n = f(x_{n-1})$, where $f(x) = x - g(x)/g'(x)$. Newton's **method** applied to a typical function $g(x)$, starting from $x_0 = 1$ and moving toward the unique root of $g(x) = 0$ on $(0, \infty)$. The iterate x_n is taken as the point of intersection of the x-axis and the tangent drawn through $[x_{n-1}, g(x_{n-1})]$. The **method** fails to converge if x_0 is chosen too far to the left or right.

The local convergence properties of Newton's **method** are determined by

$$f'(x_\infty) = 1 - \frac{g'(x_\infty)}{g'(x_\infty)} + \frac{g(x_\infty)g''(x_\infty)}{g'(x_\infty)^2} = 0.$$

If we let $e_n = x_n - x_\infty$ be the current error in approximating x_∞ , then executing a second-order Taylor expansion around x_∞ yields

$$\begin{aligned} e_n &= f(x_{n-1}) - f(x_\infty) = f'(x_\infty)e_{n-1} + \frac{1}{2}f''(z)e_{n-1}^2 \\ &= \frac{1}{2}f''(z)e_{n-1}^2, \end{aligned} \quad (2.5.2)$$

where z again lies between x_{n-1} and x_∞ . Provided $f''(z)$ is continuous and x_0 is close enough to x_∞ , the error representation (2.5.2) makes it clear that Newton's **method** converges and that

$$\lim_{n \rightarrow \infty} \frac{e_n}{e_{n-1}^2} = \frac{1}{2}f''(x_\infty).$$

This property is referred to as quadratic convergence. If an iteration function $f(x)$ satisfies $0 < |f'(x_\infty)| < 1$, then a first-order Taylor expansion implies $\lim_{n \rightarrow \infty} e_n/e_{n-1} = f'(x_\infty)$, which is referred to as linear convergence.

All else being equal, quadratic convergence is preferred to linear convergence. In practice, Newton's **method** can fail miserably if started too far from a desired root x_∞ . Furthermore, it can be expensive to evaluate the derivative $g'(x)$. For these reasons, simpler, more robust methods such as bisection are often employed instead of Newton's **method**. The following two examples highlight favorable circumstances ensuring global convergence of Newton's **method** on a properly defined domain.

Example 2.5.1

We apply Newton's Method to approximate a root of the function

$$f(x) = x^3 - 2x - 5.$$

Step 1: Compute the derivative

$$f'(x) = 3x^2 - 2.$$

Step 2: Newton's Iteration Formula

Newton's iteration formula is

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

Step 3: Choose an Initial Guess

Let us choose $x_0 = 2$.

Step 4: Perform Iterations

- $x_1 = 2 - \frac{f(2)}{f'(2)} = 2 - \frac{8-4-5}{12-2} = 2 - \frac{-1}{10} = 2.1$
- $x_2 = 2.1 - \frac{f(2.1)}{f'(2.1)} \approx 2.1 - \frac{9.261-4.2-5}{13.23-2} \approx 2.1 - \frac{0.061}{11.23} \approx 2.09457$
- $x_3 \approx 2.09455$

Conclusion

After a few iterations, Newton's Method converges to a root near

$$x \approx 2.09455,$$

which satisfies $f(x) = 0$.

2.5.1 Algorithm for Newton methods

Algorithm 1 Newton's Method for Root Finding

```
1: Input: Function  $f(x)$ , derivative  $f'(x)$ , initial guess  $x_0$ , tolerance  $\varepsilon$ , maxi-
   mum iterations  $N$ 
2: Output: Approximate root  $x$ 
3: for  $n = 0$  to  $N - 1$  do
4:   Compute  $f(x_n)$  and  $f'(x_n)$ 
5:   if  $|f'(x_n)| < 10^{-12}$  then
6:     Error: Derivative too small
7:     Exit
8:   end if
9:    $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$ 
10:  if  $|x_{n+1} - x_n| < \varepsilon$  then
11:    Return  $x_{n+1}$  as the root
12:  end if
13: end for
14: Warning: Maximum iterations reached without convergence
```

Chapter 3

Shooting Method

In the **shooting method**, we solve the initial value problem

$$u'' = f(x, u, u') \quad (3.0.1)$$

$$u(a) = r_1, \quad u'(a) = \alpha \quad (3.0.2)$$

where α is some approximation of the initial slope. Using any of the methods for solving the initial value problems, the approximation $u^{(1)}(b)$ to the solution $u(b)$ is determined. This value is either smaller or larger than the required solution $u(b) = r_2$.

Let us denote $g(\alpha_0) = u^{(1)}(b) - u(b)$, where α_0 is the first approximation of α . If $g(\alpha_0) = 0$, then the condition at $x = b$ is satisfied. If this condition is not satisfied, then we repeat the above procedure using $u'(a) = \alpha_1$ to find another estimate $u^{(2)}(b)$ for $u(b)$. The process is usually repeated until the computed value at $x = b$ agrees with the boundary condition $u(b)$.

The shooting method defines a functional relationship $g(\alpha) = 0$, between $u(b)$ and the initial slope $u'(a)$. The problem is then to find the root of this equation. This root cannot be determined by the Newton-Raphson method and the secant method is often used. Secant method gives:

$$\alpha_{n+1} = \alpha_n - \left[\frac{\alpha_n - \alpha_{n-1}}{g(\alpha_n) - g(\alpha_{n-1})} \right] g(\alpha_n), \quad n = 1, 2, \dots \quad (3.0.3)$$

We have:

$$\alpha_2 = \alpha_1 - \left[\frac{\alpha_1 - \alpha_0}{u^{(2)}(b) - u^{(1)}(b)} \right] g(\alpha_1) \quad (3.0.4)$$

$$\alpha_3 = \alpha_2 - \left[\frac{\alpha_2 - \alpha_1}{u^{(3)}(b) - u^{(2)}(b)} \right] g(\alpha_2) \quad (3.0.5)$$

If the differential equation is linear, then the shooting method becomes very simple. It can be shown that the function relationship $g(\alpha) = 0$ between $u'(a)$ and $u(b)$ is also linear. We suppose that we have computed two solutions $u_1(x)$ and $u_2(x)$ of the differential equation. Both the solutions are obtained using the same initial value $u_1(a) = r_1 = u_2(a)$, but different initial slopes $u'_1(a)$ and $u'_2(a)$. Then, by the superposition principle, the solution of the differential equation can be written as:

$$u(x) = c_1 u_1(x) + c_2 u_2(x) \quad (3.0.6)$$

We have:

$$u(a) = \gamma_1 = c_1 \gamma_1 + c_2 \gamma_1 \quad (3.0.7)$$

or

$$c_1 + c_2 = 1 \quad (3.0.8)$$

and

$$u(b) = \gamma_2 = c_1 u_1(b) + c_2 u_2(b) \quad (3.0.9)$$

Solving equations, we get:

$$c_2 = \frac{\gamma_2 - u_1(b)}{u_2(b) - u_1(b)}, \quad c_1 = 1 - c_2 \quad (3.0.10)$$

Substituting back, we get the solution of the differential equation

Example

Find the solution of the boundary value problem using the **shooting method** and the fourth-order **Taylor series method** with $h = 0.2$:

$$u'' = u' + x^2, \quad x \in [0, 1], \quad u(0) = 0, \quad u(1) = 0$$

We assume $u'(0) = \alpha$, and integrate using the Taylor method.

Step 1: Derivatives

Given:

$$u'' = u' + x^2$$

$$u''' = u'' + 2x = u' + x^2 + 2x$$

$$u^{(4)} = u''' + 2 = u' + x^2 + 2x + 2$$

Step 2: Taylor series formula

$$u_{n+1} = u_n + h u'_n + \frac{h^2}{2} u''_n + \frac{h^3}{6} u'''_n + \frac{h^4}{24} u^{(4)}_n$$

$$u'_{n+1} = u'_n + h u''_n + \frac{h^2}{2} u'''_n + \frac{h^3}{6} u^{(4)}_n$$

With $h = 0.2$, we compute:

$$h^2 = 0.04, \quad h^3 = 0.008, \quad h^4 = 0.0016$$

Step 3: First trial with $u'(0) = 0.5$

$$u_0 = 0, \quad u'_0 = 0.5$$

Using the Taylor method, we compute the following (rounded):

x	$u(x)$	$u'(x)$
0.0	0.00000	0.50000
0.2	0.11080	0.61330
0.4	0.23891	0.75172
0.6	0.38746	0.92171
0.8	0.55953	1.12987
1.0	0.75804	1.38335

Step 4: Second trial with $u'(0) = -0.5$

$$u_0 = 0, \quad u'_0 = -0.5$$

Results:

x	$u(x)$	$u'(x)$
0.0	0.00000	-0.50000
0.2	-0.09080	-0.38670
0.4	-0.17891	-0.24828
0.6	-0.26171	-0.07829
0.8	-0.33753	0.12987
1.0	-0.40804	0.38335

Step 5: Linear combination to satisfy $u(1) = 0$

We seek constants c_1, c_2 such that:

$$u(x) = c_1 u_1(x) + c_2 u_2(x), \quad u(1) = 0$$

$$c_2 = \frac{0 - u_1(1)}{u_2(1) - u_1(1)} = \frac{-0.75804}{-0.40804 - 0.75804} = \frac{-0.75804}{-1.16608} \approx 0.6500$$

$$c_1 = 1 - c_2 = 0.3500$$

Step 6: Final approximate solution

$$u(x) = 0.35u_1(x) + 0.65u_2(x)$$

x	$u(x)$
0.0	0.00000
0.2	-0.0202
0.4	-0.0327
0.6	-0.0086
0.8	0.0135
1.0	0.00000

Chapter 4

Application of the Shooting Method

The shooting method is an approach for solving boundary value problems by converting them into initial value problems. This involves imposing additional conditions at a single point and iteratively adjusting them until the boundary conditions at the other end are met.

In practice, the shooting method tends to be slow. Consequently, finite difference methods are generally preferred for solving boundary value problems.

The following Corollary establishes general conditions that ensure the existence and uniqueness of a solution to a second-order boundary value problem [23,25].

4.1 Corollary

Let $p(t)$ be a C^1 -function, $p(t) : \mathbb{R}^+ \rightarrow \mathbb{R}^+$, which is non-increasing. Consider a linear two-point boundary value problem of the form

$$u'' + p(t)u' + \frac{1}{\alpha}p'(t)u = 0, \quad (4.1.1)$$

subject to the boundary conditions

$$u(0) = \zeta, \quad u'(0) = -2\gamma'\zeta,$$

where

$$\gamma' > \frac{1}{2}, \quad \zeta > 0, \quad \alpha > 1 \in \mathbb{R}^+, \quad t \geq 0.$$

Under these conditions, the problem admits a unique solution.

These results guarantee the existence and uniqueness of the solution to the two-point boundary value problem under consideration. Consider the solution of the boundary value problem (3.1.1). In the shooting method, we solve the initial value problem

$$\gamma' > \frac{1}{2}, \alpha > 1, t \geq 0 \in \mathbb{R}^+, u_0 = 1, u_T = 0$$

$$u'' + \frac{\gamma'(\alpha+1)}{2\alpha\gamma't+1}u' - \frac{2(\gamma')^2(\alpha+1)}{(2\alpha\gamma't+1)^2}u = 0$$

Clearly, the analytic solution to the IVP is given by

$$u(t) = \left(2\alpha \frac{\gamma'}{\zeta^\alpha} t + \zeta^{-\alpha}\right)^{\frac{-1}{\alpha}}$$

Consider the second-order differential equation

$$u'' = -p(t)u' - \frac{1}{\alpha}p'(t)u = f(t, u, u'), \quad (4.1.2)$$

subject to the boundary conditions

$$u(0) = \zeta, \quad u(T) = \beta. \quad (4.1.3)$$

First, Eq. (3.1.2) is written in terms of a system of two first-order differential equations:

$$\begin{cases} u' = v \\ v' = f(t, u, v) \end{cases}$$

A shooting method works with the initial value problem

$$u'' = f(t, u, u'), \quad (4.1.4)$$

subject to the boundary conditions

$$u(0) = \zeta, \quad u'(0) = \lambda. \quad (4.1.5)$$

which is usually treated as a system

$$\begin{cases} u' = v \\ v' = f(t, u, v) \\ u(0) = \zeta, \quad v(0) = \lambda \end{cases}$$

finding a value of λ that gives a solution satisfying $u(T) = \beta$.

$u_\lambda(t)$ represents the solution of the initial value problem (3.1.4)-(3.1.5) trying to find a root of the linear function

$$R(\lambda) \equiv u_\lambda(T) - \beta$$

$u_0(t)$ is the solution of (3.1.4)-(3.1.5) with $\lambda = 0$, and $w(t)$ is the solution of the corresponding homogeneous ordinary differential equation satisfying $w(0) = 0$, $w'(0) = 1$. Then

$$u_\lambda(t) = u_0(t) + \lambda w(t)$$

and thus

$$R(\lambda) \equiv u_0(T) + \lambda w(T) - \beta$$

We want to select λ so that

$$u_0(T) + \lambda w(T) = \beta$$

$R(\lambda)$ is linear. Using the equation of the straight line

$$\lambda_n = \lambda_{n-1} - \left(\frac{\lambda_{n-1} - \lambda_{n-2}}{R(\lambda_{n-1}) - R(\lambda_{n-2})} \right) R(\lambda_{n-1}), \quad \lambda_0 = \frac{u_T - u(0)}{T - 0}, \quad \lambda_1 = 1.001\lambda_0$$

To apply Secant method, Eq. (3.1.4) is written in terms of two first-order differential equations,

$$u' = v, \quad v' = -p(t)v - \frac{1}{\alpha}p'(t)u, \quad (4.1.6)$$

subject to the boundary conditions

$$u(0) = \zeta, \quad u(T) = \beta. \quad (4.1.7)$$

Let us denote the missing slope by

$$u'(0) = \lambda. \quad (4.1.8)$$

As an example, consider the solution of Eq. (3.1.4)-(3.1.5) for $\gamma' = 0.6, \zeta = 1, \alpha = 1.1, \quad p = \frac{\gamma'(\alpha+1)}{2\alpha\gamma't+1}, p' = -\frac{2(\gamma')^2(\alpha+1)}{(2\alpha\gamma't+1)^2}$. As a first approximation, let us assume the missing initial slope to be $u'(0) = \lambda_1 = -1.2$

Iteration Table:

Iteration	λ_n	$R(\lambda_n)$	Error ($ R(\lambda_n) $)
0	-1.000000	0.496862	0.496862
1	-1.200000	0.151311	0.151311
2	-1.287577	-0.000001	0.000001
3	-1.287576	0.000000	0.000000

The missing initial slope λ is approximately: -1.287575920052582

4.2 Solving the Boundary Value Problem

We aim to solve the boundary value problem (BVP):

$$u'' + p(t)u' + q(t)u = 0,$$

with boundary conditions:

$$u(0) = \zeta, \quad u(T) = \beta.$$

Define:

$$p(t) = \frac{\gamma'(\alpha+1)}{2\alpha\gamma't+1}, \quad q(t) = -\frac{2(\gamma')^2(\alpha+1)}{(2\alpha\gamma't+1)^2}.$$

Rewrite the equation as a system of first-order ordinary differential equations (ODEs). Let:

$$u_1(t) = u(t), \quad u_2(t) = u'(t).$$

Then:

$$u_1'(t) = u_2(t),$$

$$u_2'(t) = -p(t)u_2(t) + q(t)u_1(t).$$

Initialize parameters:

$$h = \frac{T}{N}, \quad u_{1,0} = \zeta, \quad u_{2,0} = 0, \quad v_{1,0} = 0, \quad v_{2,0} = 1.$$

Using the Runge-Kutta method, for each $i = 0, 1, \dots, N-1$, compute the increments for both systems:

First System ($u_{1,i}, u_{2,i}$):

$$\begin{aligned} k_{1,1} &= hu_{2,i}, \quad k_{1,2} = h[-p(t_i)u_{2,i} + q(t_i)u_{1,i}], \\ k_{2,1} &= h \left[u_{2,i} + \frac{1}{2}k_{1,2} \right], \quad k_{2,2} = h \left[-p \left(t_i + \frac{h}{2} \right) \left(u_{2,i} + \frac{1}{2}k_{1,2} \right) + q \left(t_i + \frac{h}{2} \right) \left(u_{1,i} + \frac{1}{2}k_{1,1} \right) \right], \\ k_{3,1} &= h \left[u_{2,i} + \frac{1}{2}k_{2,2} \right], \quad k_{3,2} = h \left[-p \left(t_i + \frac{h}{2} \right) \left(u_{2,i} + \frac{1}{2}k_{2,2} \right) + q \left(t_i + \frac{h}{2} \right) \left(u_{1,i} + \frac{1}{2}k_{2,1} \right) \right], \\ k_{4,1} &= h[u_{2,i} + k_{3,2}], \quad k_{4,2} = h[-p(t_i + h)(u_{2,i} + k_{3,2}) + q(t_i + h)(u_{1,i} + k_{3,1})]. \end{aligned}$$

Update the values:

$$\begin{aligned} u_{1,i+1} &= u_{1,i} + \frac{1}{6}(k_{1,1} + 2k_{2,1} + 2k_{3,1} + k_{4,1}), \\ u_{2,i+1} &= u_{2,i} + \frac{1}{6}(k_{1,2} + 2k_{2,2} + 2k_{3,2} + k_{4,2}). \end{aligned}$$

Second System ($v_{1,i}, v_{2,i}$):

$$\begin{aligned} k'_{1,1} &= hv_{2,i}, \quad k'_{1,2} = h[-p(t_i)v_{2,i} + q(t_i)v_{1,i}], \\ k'_{2,1} &= h \left[v_{2,i} + \frac{1}{2}k'_{1,2} \right], \quad k'_{2,2} = h \left[-p \left(t_i + \frac{h}{2} \right) \left(v_{2,i} + \frac{1}{2}k'_{1,2} \right) + q \left(t_i + \frac{h}{2} \right) \left(v_{1,i} + \frac{1}{2}k'_{1,1} \right) \right], \\ k'_{3,1} &= h \left[v_{2,i} + \frac{1}{2}k'_{2,2} \right], \quad k'_{3,2} = h \left[-p \left(t_i + \frac{h}{2} \right) \left(v_{2,i} + \frac{1}{2}k'_{2,2} \right) + q \left(t_i + \frac{h}{2} \right) \left(v_{1,i} + \frac{1}{2}k'_{2,1} \right) \right], \\ k'_{4,1} &= h[v_{2,i} + k'_{3,2}], \quad k'_{4,2} = h[-p(t_i + h)(v_{2,i} + k'_{3,2}) + q(t_i + h)(v_{1,i} + k'_{3,1})]. \end{aligned}$$

Update the values:

$$\begin{aligned} v_{1,i+1} &= v_{1,i} + \frac{1}{6}(k'_{1,1} + 2k'_{2,1} + 2k'_{3,1} + k'_{4,1}), \\ v_{2,i+1} &= v_{2,i} + \frac{1}{6}(k'_{1,2} + 2k'_{2,2} + 2k'_{3,2} + k'_{4,2}). \end{aligned}$$

The shooting method yields:

$$w_{2,0} = \frac{\beta - u_{1,N}}{v_{1,N}}.$$

Final solution:

$$W1 = u_{1,i} + w_{2,0}v_{1,i}, \quad W2 = u_{2,i} + w_{2,0}v_{2,i}.$$

The output is:

$$(x_i, W1, W2),$$

where $W1 \approx u(t_i)$ and $W2 \approx u'(t_i)$.

Approximate values of $u(t)$ and $v(t)$:

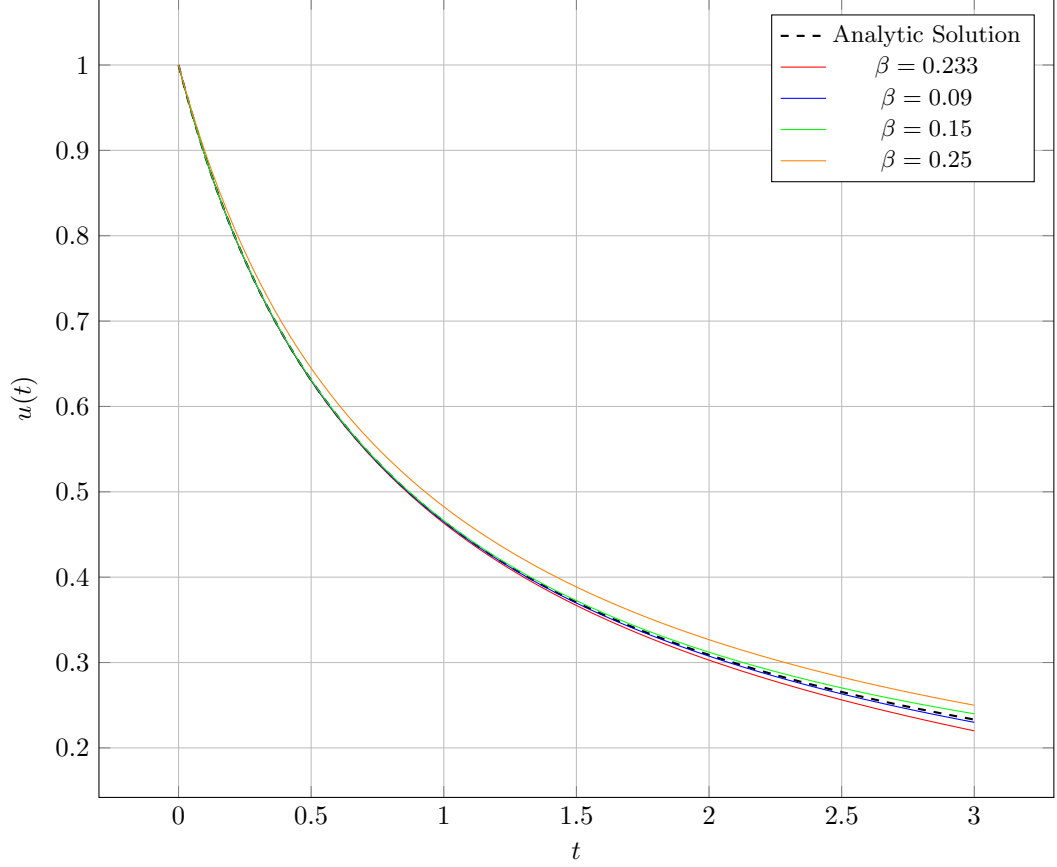
t	u(t)	v(t)	t	u(t)	v(t)	t	u(t)	v(t)
0.0000	1.000000	-1.287576	0.6061	0.532873	-0.475584	1.2121	0.316898	-0.272165
0.0303	0.962308	-1.201701	0.6364	0.518709	-0.459151	1.2424	0.308737	-0.266340
0.0606	0.927079	-1.124778	0.6667	0.505030	-0.443720	1.2727	0.300752	-0.260764
0.0909	0.894052	-1.055522	0.6970	0.491808	-0.429210	1.3030	0.292932	-0.255422
0.1212	0.862997	-0.992851	0.7273	0.479017	-0.415547	1.3333	0.285273	-0.250303
0.1515	0.833749	-0.936058	0.7576	0.466629	-0.402664	1.3636	0.277766	-0.245391
0.1818	0.806152	-0.884500	0.7879	0.454621	-0.390496	1.3939	0.270407	-0.240677
0.2121	0.780064	-0.837587	0.8182	0.442972	-0.378986	1.4242	0.263187	-0.236148
0.2424	0.755352	-0.794781	0.8485	0.431660	-0.368084	1.4545	0.256102	-0.231793
0.2727	0.731896	-0.755599	0.8788	0.420668	-0.357742	1.4848	0.249146	-0.227603
0.3030	0.709586	-0.719609	0.9091	0.409978	-0.347921	1.5152	0.242315	-0.223569
0.3333	0.688323	-0.686431	0.9394	0.399576	-0.338586	1.5455	0.235602	-0.219682
0.3636	0.668019	-0.655739	0.9697	0.389447	-0.329707	1.5758	0.229004	-0.215934
0.3939	0.648599	-0.627259	1.0000	0.379582	-0.321263	1.6061	0.222517	-0.212318
0.4242	0.629996	-0.600769	1.0303	0.369968	-0.313231	1.6364	0.216136	-0.208829
0.4545	0.612157	-0.576101	1.0606	0.360590	-0.305574	1.6667	0.209857	-0.205459
0.4848	0.595039	-0.553140	1.0909	0.351438	-0.298265	1.6970	0.203679	-0.202206
0.5152	0.578602	-0.531777	1.1212	0.342501	-0.291287	1.7273	0.197597	-0.199063
0.5455	0.562786	-0.511803	1.1515	0.333771	-0.284623	1.7576	0.191609	-0.196028
0.5758	0.547555	-0.493104	1.1818	0.325239	-0.278254	1.7879	0.185713	-0.193098

Approximate values of $u(t)$ and $v(t)$:

t	u(t)	v(t)	t	u(t)	v(t)
1.8182	0.179904	-0.190265	2.4242	0.078434	-0.148728
1.8485	0.174179	-0.187522	2.4545	0.073952	-0.147214
1.8788	0.168535	-0.184867	2.4848	0.069515	-0.145738
1.9091	0.162970	-0.182296	2.5152	0.065122	-0.144301
1.9394	0.157481	-0.179806	2.5455	0.060771	-0.142899
1.9697	0.152067	-0.177394	2.5758	0.056462	-0.141533
2.0000	0.146725	-0.175057	2.6061	0.052193	-0.140200
2.0303	0.141453	-0.172792	2.6364	0.047964	-0.138899
2.0606	0.136250	-0.170596	2.6667	0.043774	-0.137630
2.0909	0.131112	-0.168466	2.6970	0.039621	-0.136392
2.1212	0.126038	-0.166401	2.7273	0.035506	-0.135183
2.1515	0.121027	-0.164396	2.7576	0.031426	-0.134003
2.1818	0.116075	-0.162450	2.7879	0.027382	-0.132852
2.2121	0.111183	-0.160560	2.8182	0.023373	-0.131728
2.2424	0.106347	-0.158724	2.8485	0.019398	-0.130632
2.2727	0.101566	-0.156941	2.8788	0.015455	-0.129561
2.3030	0.096839	-0.155206	2.9091	0.011545	-0.128514
2.3333	0.092163	-0.153520	2.9394	0.007666	-0.127492
2.3636	0.087538	-0.151879	2.9697	0.003818	-0.126492
2.3939	0.082962	-0.150282	3.0000	-0.000000	-0.125515

$u(T)$ with $\lambda = -1.2875759200131718$: $1.6776510736171701e-12$

Numerical Solution of the BVP for Different β Values



4.3 System of First Order Initial Value Problems

Let the second order initial value problem be given as

$$u'' - 4 \frac{(\gamma')^2 (\delta + 1)}{\xi^\delta (2\delta\gamma't + 1)} u^{\delta+1} = 0 \quad (4.3.1)$$

$$u(0) = \xi, \quad u(T) = \beta,$$

Define $y_1 = u$. Then, we have the system

$$y_1' = u' = y_2, \quad y_2' = u'' = 4 \frac{(\gamma')^2 (\delta + 1)}{\xi^\delta (2\delta\gamma't + 1)} y_1^{\delta+1}$$

$$y_1(0) = \xi, \quad y_1(T) = \beta.$$

The system is given by

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}' = \begin{bmatrix} y_2 \\ 4 \frac{(\gamma')^2 (\delta+1)}{\xi^\delta (2\delta\gamma't+1)} y_1^{\delta+1} \end{bmatrix}, \quad \begin{bmatrix} y_1(0) \\ y_2(0) \end{bmatrix} = \begin{bmatrix} \xi \\ s \end{bmatrix}$$

To apply Newton's method, Eq. (3.3.1) is written in terms of two first-order differential equations,

$$\frac{du}{dt} = y_2, \quad \frac{dy_2}{dt} = f(t, y_1, y_2) \quad (4.3.2)$$

We denote the missing initial slope by

$$\frac{dy_1}{dt}(0) = s \quad \text{or} \quad y_2(0) = s \quad (4.3.3)$$

The problem is to find s such that the solution of Eq. (4.3.2), subject to the initial condition (4.3.3), satisfies the boundary condition at the second point, $u(T) = \beta$. In other words, if the solutions of the initial value problem are denoted by $u(t, s)$ and $y_2(t, s)$, one searches for the value of s such that

$$u(T, s) - \beta = \phi(s) = 0 \quad (4.3.4)$$

For Newton's method, the iteration formula for s is given by

$$s^{(n+1)} = s^{(n)} - \frac{\phi(s^{(n)})}{\frac{\partial \phi(s^{(n)})}{\partial s}} \quad (4.3.5)$$

or

$$s^{(n+1)} = s^{(n)} - \frac{u(T, s^{(n)}) - \beta}{\frac{du(T, s^{(n)})}{ds}} \quad (4.3.6)$$

Example: If we take $\beta = 0.23$, $\gamma' = 0.6$, $\delta = 1.1$ and $\xi = 1$

Iter	s	$\phi(s)$	$\frac{d\phi}{ds}$	s_{new}
1	1.00000000	5.110422×10^{20}	3.090169×10^{20}	-0.65376774
2	-0.65376774	2381.03628225	208287.52418781	-0.66519922
3	-0.66519922	1064.39906181	61357.16868982	-0.68254681
4	-0.68254681	473.57904276	18205.16799457	-0.70856025
5	-0.70856025	209.13406704	5463.43071932	-0.74683915
6	-0.74683915	91.24624437	1669.57499705	-0.80149152
7	-0.80149152	39.02747968	525.37307304	-0.87577679
8	-0.87577679	16.13575433	173.52422685	-0.96876527
9	-0.96876527	6.27357833	62.26088694	-1.06952803
10	-1.06952803	2.15774503	25.87616713	-1.15291538
11	-1.15291538	0.56118636	13.88473956	-1.19333287
12	-1.19333287	0.07179419	10.51749821	-1.20015903
13	-1.20015903	0.00160904	10.05047346	-1.20031913
14	-1.20031913	8.6×10^{-7}	10.03982902	-1.20031922

The method converged successfully to:

$$s = u'(0) = -1.200319$$

To determine ϕ' , we differentiate partially with respect to s all the equations in (3.1.4) and (3.1.5)

$$v_s'' = f_{u_s}(t, u_s, u_s')v + f_{u_s'}(t, u_s, u_s')v' \quad (4.3.7)$$

and t and s are independent,

$$v(0) = 0, \quad v'(0) = 1, \quad v = \frac{\partial u_s}{\partial s} \quad (4.3.8)$$

Recall that u_s is defined as the solution of the problem

$$u_s'' = f(t, u_s, u_s') \quad (4.3.9)$$

subject to the boundary conditions

$$u_s(0) = \xi, \quad u_s'(0) = s \quad (4.3.10)$$

In vector notation, define

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_0 \\ b_1 \end{bmatrix}.$$

Then, we can write the system as

$$\mathbf{y}' = \mathbf{f}(x, \mathbf{y}), \quad \mathbf{y}(x_0) = \mathbf{b}.$$

Let an integer $N > 0$ be chosen and set $h = (b-a)/N$. Partition the interval $[a, b]$ into N subintervals with the mesh points

$$t_i = a + (i-1)h \quad \text{for each } i = 1, \dots, N.$$

Use the notation w_{ij} for each $j = 0, 1, \dots, N$ and $i = 1, 2$ to denote an approximation $w_{1,i}$ to $u(t_i)$; $w_{2,i}$ to $u'(t_i)$. For the initial conditions, set

$$w_{1,0} = \xi; \quad w_{2,0} = (\beta - \xi)/(b - a)$$

$(\beta - \xi)/(b - a)$ is the slope of the straight line through (a, ξ) and (b, β) .

$$\begin{aligned} k_{1,1} &= hw_{2,i-1}, \quad k_{1,2} = hf(t, w_{1,i-1}, w_{2,i-1}), \\ k_{2,1} &= hf(t + h/2, w_{1,i-1} + \frac{1}{2}k_{1,1}, w_{2,i-1} + \frac{1}{2}k_{1,2}), \\ k_{3,1} &= hf(t + h/2, w_{1,i-1} + \frac{1}{2}k_{2,1}, w_{2,i-1} + \frac{1}{2}k_{2,2}), \\ k_{4,1} &= hf(t + h, w_{1,i-1} + k_{3,1}, w_{2,i-1} + k_{3,2}), \\ w_{1,i} &= w_{1,i-1} + \frac{1}{6}(k_{1,1} + 2k_{2,1} + 2k_{3,1} + k_{4,1}), \\ w_{2,i} &= w_{2,i-1} + \frac{1}{6}(k_{1,2} + 2k_{2,2} + 2k_{3,2} + k_{4,2}), \end{aligned}$$

$$k'_{1,1} = hy_2, \quad k'_{1,2} = h[f_y(t, w_{1,i-1}, w_{2,i-1})y_1 + f_y(t, w_{1,i-1}, w_{2,i-1})y_2],$$

$$k'_{2,1} = h \left(y_2 + \frac{1}{2}k'_{1,2} \right),$$

$$k'_{2,2} = hf_y(t+h/2, w_{1,i-1}, w_{2,i-1}) \left(y_1 + \frac{1}{2}k'_{1,1} \right) + f_y(t+h/2, w_{1,i-1}, w_{2,i-1}) \left(y_2 + \frac{1}{2}k'_{1,2} \right),$$

$$k'_{3,1} = h \left(y_2 + \frac{1}{2} k'_{2,2} \right),$$

$$k'_{3,2} = h f_y(t+h/2, w_{1,i-1}, w_{2,i-1})(y_1 + \frac{1}{2} k'_{2,1}) + f_y(t+h/2, w_{1,i-1}, w_{2,i-1})(y_2 + \frac{1}{2} k'_{2,2}),$$

$$k'_{4,1} = h(y_2 + k'_{3,2}),$$

$$k'_{4,2} = h f_y(t+h, w_{1,i-1}, w_{2,i-1})(y_1 + k'_{3,1}) + f_y(t+h, w_{1,i-1}, w_{2,i-1})(y_2 + k'_{3,2}).$$

$$y_1 = y_1 + \frac{1}{6}(k'_{1,1} + 2k'_{2,1} + 2k'_{3,1} + k'_{4,1}),$$

$$y_2 = y_2 + \frac{1}{6}(k'_{1,2} + 2k'_{2,2} + 2k'_{3,2} + k'_{4,2}).$$

for $\beta = 0.23$:

x_i	w_i	$\mathcal{Y}(x_i)$	$ w_i - \mathcal{Y}(x_i) $
0.000000	1.000000	1.000000	0.000000
0.300000	0.738288	0.738390	0.000102
0.600000	0.588194	0.588427	0.000233
0.900000	0.490351	0.490757	0.000406
1.200000	0.421254	0.421880	0.000626
1.500000	0.369690	0.370591	0.000900
1.800000	0.329620	0.330852	0.001232
2.100000	0.297494	0.299119	0.001625
2.400000	0.271084	0.273168	0.002084
2.700000	0.248923	0.251533	0.002610
3.000000	0.230000	0.233208	0.003208

for $\beta = 0.22$:

x_i	w_i	$\mathcal{Y}(x_i)$	$ w_i - \mathcal{Y}(x_i) $
0.000000	1.000000	1.000000	0.000000
0.300000	0.737968	0.738390	0.000423
0.600000	0.587463	0.588427	0.000964
0.900000	0.489081	0.490757	0.001675
1.200000	0.419294	0.421880	0.002585
1.500000	0.366875	0.370591	0.003716
1.800000	0.325768	0.330852	0.005084
2.100000	0.292414	0.299119	0.006705
2.400000	0.264576	0.273168	0.008591
2.700000	0.240777	0.251533	0.010756
3.000000	0.220000	0.233208	0.013208

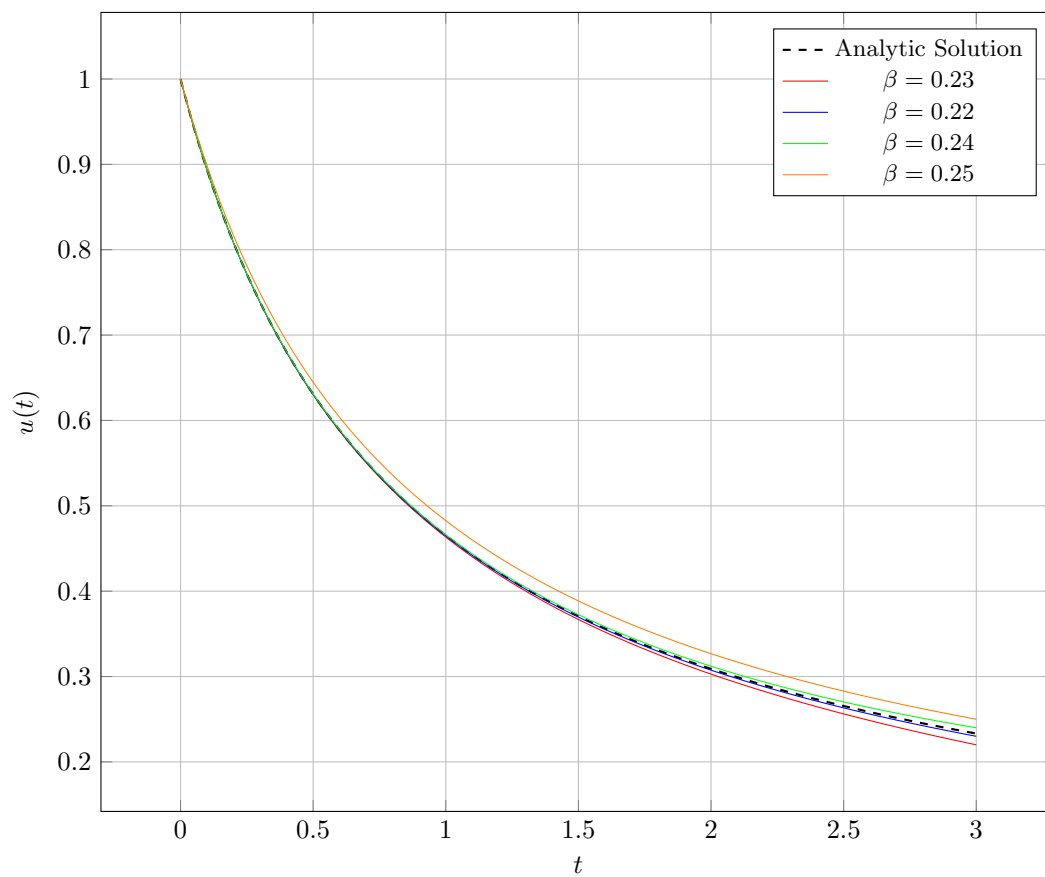
for $\beta = 0.24$:

x_i	w_i	$\mathcal{Y}(x_i)$	$ w_i - \mathcal{Y}(x_i) $
0.000000	1.000000	1.000000	0.000000
0.300000	0.738606	0.738390	0.000216
0.600000	0.588920	0.588427	0.000492
0.900000	0.491612	0.490757	0.000856
1.200000	0.423201	0.421880	0.001321
1.500000	0.372491	0.370591	0.001900
1.800000	0.333454	0.330852	0.002601
2.100000	0.302552	0.299119	0.003433
2.400000	0.277572	0.273168	0.004404
2.700000	0.257054	0.251533	0.005521
3.000000	0.240000	0.233208	0.006792

for $\beta = 0.25$:

x_i	w_i	$\mathcal{Y}(x_i)$	$ w_i - \mathcal{Y}(x_i) $
0.000000	1.000000	1.000000	0.000000
0.300000	0.738923	0.738390	0.000532
0.600000	0.589641	0.588427	0.001214
0.900000	0.492866	0.490757	0.002110
1.200000	0.425137	0.421880	0.003257
1.500000	0.375275	0.370591	0.004685
1.800000	0.337268	0.330852	0.006415
2.100000	0.307590	0.299119	0.008471
2.400000	0.284040	0.273168	0.010873
2.700000	0.265172	0.251533	0.013639
3.000000	0.250000	0.233208	0.016792

Numerical Solution of the BVP for Different β Values



General Conclusion

At the end of this work, we have conducted an in-depth study of the shooting method, relying on solid theoretical references and practical examples. This method, although classical, remains of great importance for solving boundary value problems, particularly in ordinary differential equations.

Through our analysis, we have highlighted the mathematical rigor required for its implementation, as well as the graphical tools useful for its understanding. The main objective of this work was to progressively master the procedure for applying the shooting method, which we have achieved through a structured approach supported by concrete results.

This work thus constitutes a solid foundation for further exploration of other similar numerical methods and paves the way for possible extensions in more complex or multidimensional contexts.

Bibliography

- [1] V. Komornik, *Exact controllability and stabilization the multiplier method*, John Wiley and Sons, Masson, Paris, 1994.
- [2] R. Temam, *Infinite-Dimensional Dynamical Systems in Mechanics and Physics*, Springer, 1997.
- [3] N. Alexandre, J. A. Langa and J. Robinson, *Attractors for infinite-dimensional non-autonomous dynamical systems*, Springer Science+Business Media, LLC, 2013.
- [4] Z. Enrike, “Exponential Decay for The Semilinear Wave Equation with Locally Distributed Damping,” *Communications in Partial Differential Equations*, vol. 15, no. 2, pp. 205-235, 1990.
- [5] Q. Lil and L. He1, “General decay and blow-up of solutions for a nonlinear viscoelastic wave equation with strong damping,” *Boundary Value Problems*, pp. 2-22, 2018.
- [6] L. Liu, F. Sun and Y. Wu, “Blow-up of solutions for a nonlinear Petrovsky type equation with initial data at arbitrary high energy level ,” *Boundary Value Problems*, pp. 2-18, 2019.
- [7] J. Hao and H. Wei, “Blow-up and global existence for solution of quasilinear viscoelastic wave equation with strong damping and source term,” *Boundary Value Problems*, pp. 2-12, 2017.
- [8] B. Feng, “General decay for a viscoelastic wave equation with strong time-dependent delay,” *Boundary Value Problems*, pp. 2-11, 2017.
- [9] W. Liu, “Arbitrary rate of decay for a viscoelastic equation with acoustic boundary conditions,” *Applied Mathematics Letters*, vol. 38, pp. 155–161, 2014.
- [10] S. T. Wu and H.F . Chen, “Uniform Decay of Solutions for a Nonlinear Viscoelastic Wave Equation with Boundary Dissipation,” *Journal of Function Spaces and Applications*, pp. 1-17, 2012.
- [11] S. T. Wu, “Arbitrary decays for a viscoelastic equation,” *Boundary Value Problems*, pp. 2–14, 2011.
- [12] B. Tarek, “Bounds, decay, and integrals of non-Increasing functions: A comprehensive analysis,” *IAENG International Journal of Applied Mathematics*, vol. 54, no. 8, pp. 1581–1585, 2024.

- [13] Y. Qin, *Analytic Inequalities and Their Applications in PDEs*, Operator Theory: Advances and Applications, Springer International Publishing Switzerland, 2017.
- [14] T. A. Burton , *Volterra Integral and Differential Equations Second Edition*, Elsevier B.V. All rights reserved, 2005.
- [15] P. Linz , *Analytical and Numerical Methods for Volterra Equations* , Siam Philadelphia, 1985.
- [16] Shih-Hsiang Chang , *Numerical solution of Troesch's problem by simple shooting method* , Applied Mathematics and Computation, pp 3303–3306, 216 (2010).
- [17] Sewell, Granville. *The numerical solution of ordinary and partial differential equations, 2nd ed*, John Wiley, 2005.
- [18] M. N. O. JZTBRELI and F. PASCAL *NUMERICAL SOLUTIONS OF TWO-POINT BOUNDARY VALUE PROBLEMS* , Nonlinear Analysis. Theory. Methods , Applications, Vol. 3, No. 3, pp. 395-418, 1979.
- [19] HERBERT B. KELLER *Numerical Solution of Two Point Boundary Value Problems*, Society for Industrial and Applied Mathematics, 1976.
- [20] M. K. JAIN S. R. K. IYENGAR R. K. JAIN *Numerical Methods for Scientific and Engineering Computation*, JOHN WILEY and SONS, 1985.
- [21] Basem S. Attili *THE SHOOTING METHOD FOR SOLVING SECOND ORDER FUZZY TWO-POINT BOUNDARY VALUE PROBLEMS*, International Journal of Applied Mathematics, Vol 32, No. 4, pp 663-676, 2019.
- [22] David Kincaid, Ward Cheney *Numerical Analysis* , American Mathematical Society. , 2002.
- [23] Richard L. Burden , J. Douglas Faires *Numerical Methods* , Brooks/Cole. , 2002.
- [24] Basem S. Attili , Muhammed I. Syam *Efficient shooting method for solving two point boundary value problems* , Chaos, Solitons and Fractals, 35, pp 895–903, 2008.
- [25] Joe D. Hoffman *Numerical Methods for Engineers and Scientists* , Second Edition, CRC Press Inc, 2001.
- [26] T. Y. Na, *COMPUTATIONAL METHODS IN ENGINEERING BOUNDARY VALUE PROBLEMS* , Academic Press Inc, 1979.
- [27] SUNG N. HA, *Nonlinear Shooting Method for Two-Point Boundary Value Problems* , Computers and Mathematics with Applications, 42, pp 1411-1420, 2001.
- [28] André Fortin, *Analyse numérique pour ingénieurs* , Presses Polytechnique de Montréal, 2008.
- [29] Jaan Kiusalaas, *NUMERICAL METHODS IN ENGINEERING WITH Python* , Cambridge University Press, 2005.

- [30] KELLER H. B., *Numerical Methods for Two-Point Boundary Value Problems: Shooting Methods*, Blaisdell publishing company, , 2018.
- [31] J. Chevallet, *Mathématiques: 34 problèmes corrigés posés à l'écrit du CAPES*, Vuibert, 1999.
- [32] J. Chevallet, *X, ENS, Mines, Centrale : 301 nouveaux sujets corrigés posés à l'oral de mathématiques*, Vuibert, 1999.
- [33] LANGE K., *Numerical Analysis for Statisticians*, Springer, Second Edition, 2010.
- [34] RAO S. B., SHANTHA C. K., *Numerical Methods with Programs in BASIC, FORTRAN, Pascal and C++*, Revised Edition.