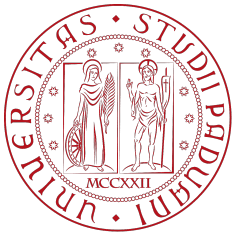

UNIVERSITÀ DEGLI STUDI DI PADOVA

Dipartimento di Scienze Economiche e Aziendali "M. Fanno"
Dipartimento di Matematica "Tullio Levi-Civita"

CORSO DI LAUREA MAGISTRALE IN

APPLIED ECONOMICS

Economic Data Analytics

TESI DI LAUREA

THE SHOOTING METHOD: AN APPLICATION TO THE RAMSEY OPTIMAL CONTROL PROBLEM

Relatore:
CH.MA PROF.SSA Alessandra Buratto

Laureanda
ELISA PREATONI
Matricola n. 2125391

Anno Accademico 2025/2026

Dichiarazione di autenticità

Dichiaro di aver preso visione del "Regolamento antiplagio" approvato dal Consiglio del Dipartimento di Scienze Economiche e Aziendali e, consapevole delle conseguenze derivanti da dichiarazioni mendaci, dichiaro che il presente lavoro non è già stato sottoposto, in tutto o in parte, per il conseguimento di un titolo accademico in altre Università italiane o straniere. Dichiaro inoltre che tutte le fonti utilizzate per la realizzazione del presente lavoro, inclusi i materiali digitali, sono state correttamente citate nel corpo del testo e nella sezione 'Riferimenti bibliografici'.

I hereby declare that I have read and understood the "Anti-plagiarism rules and regulations" approved by the Council of the Department of Economics and Management and I am aware of the consequences of making false statements. I declare that this piece of work has not been previously submitted – either fully or partially – for fulfilling the requirements of an academic degree, whether in Italy or abroad. Furthermore, I declare that the references used for this work – including the digital materials – have been appropriately cited and acknowledged in the text and in the section 'References'.

Firma/ Signature .....Elisa Peatoni.....

Abstract

Questa tesi analizza i metodi di shooting come tecniche numeriche per risolvere problemi ai valori al contorno che emergono nel controllo ottimo. Vengono esaminate le principali varianti: single shooting, multiple shooting e approcci basati su NOC. Inoltre, la ricerca discute le loro differenze pratiche in termini di stabilità, convergenza e implementazione. I metodi vengono poi implementati in Python e applicati al modello di crescita di Ramsey–Cass–Koopmans, un problema di controllo ottimo a orizzonte infinito in economia. Gli esperimenti numerici illustrano come le tecniche classiche di shooting possano ricostruire la dinamica ottimale del saddle-path e mettono in evidenza i compromessi tra single e multiple shooting, confrontandoli in termini di convergenza alla soluzione, errori terminali e sensibilità alla scelta iniziale.

This thesis investigates shooting methods as numerical techniques for solving boundary value problems arising in optimal control. We review the main variants: single shooting, multiple shooting, and NOC-based approaches. Moreover, the research discuss their practical differences in terms of stability, convergence, and implementation. The methods are then implemented in Python and applied to the Ramsey–Cass–Koopmans growth model, an infinite-horizon optimal control problem in economics. The numerical experiments illustrate how classical shooting techniques can recover the optimal saddle-path dynamics and highlight the trade-offs between single and multiple shooting, comparing them in terms of convergence to the solution, terminal errors, and sensitivity to the initial guess.

Contents

Introduction

The aim of this thesis is to study shooting method techniques and apply them to the Ramsey optimal control problem. The shooting method is a numerical technique used to solve boundary value problems by transforming them into initial value problems and iteratively guessing the initial values that satisfy the boundary conditions. This approach is especially useful for finding accurate solutions for problems where analytical methods do not work. In particular, the shooting method is useful for solving optimal control problems in different fields such as physics, mathematics, and economics. In fact, this thesis applies shooting techniques to a well-known economic model: the Ramsey-Cass-Koopmans problem. This model was chosen because it is one of the most established frameworks in economics, making it easier to present and interpret different numerical methods in a familiar setting. Moreover, it also facilitates the comparison with results obtained through other methods in the literature.

This thesis has three main objectives. The first objective is to provide an overview of dynamical systems and optimal control theory in order to clarify the context in which shooting methods are applied. The dynamical system underlying the Ramsey problem is then introduced and the steps needed to transform it into the form required for shooting methods, namely the two-point boundary value problem. The second goal of this research is to explain how the shooting method works, its main features, and its main variants. In particular, the single shooting method, the multiple shooting method, and the NOC shooting methods. Single shooting is the simplest method, on which the multiple shooting method is based. It will be analyzed how the two methods work and their differences. The NOC shooting methods are explained to provide a different option for solving optimal control problems. The last purpose of this thesis is to provide a numerical implementation in Python of the single and multiple shooting methods to solve the Ramsey problem. First, the Ramsey problem is solved with the single shooting method, then with multiple shooting, and at the end of each method a sensitivity analysis is also presented, where the initial capital is changed to see how the methods behave in different setups. In addition, a sensitivity analysis is also performed on the parameters, where the impatience rate ρ

varies.

The thesis is structured as follows. The first chapter introduces the theoretical background of dynamical systems and optimal control problems. The second chapter focuses on the shooting method, presenting and comparing its main variants, including single shooting, multiple shooting, and NOC approaches. The final chapter provides a practical application of single and multiple shooting, implemented in Python to solve the Ramsey-Cass-Koopmans model and analyze the convergence to the solution.

Chapter 1

Introduction to Dynamical System and Optimal Control

1.1 Dynamical Systems

1.1.1 From the real world to mathematical models

To study dynamical systems, we first need to clarify what a mathematical model is. Many people think that a mathematical model is simply a set of theorems expressed through mathematical symbols. However, a mathematical model is, more generally, a representation of a real system. According to Bischi et al. (2015), the modeling process consists of four stages (Figure 1.1). The first stage is the identification of a real system - which can be physical, biological, social or economical - and the selection of its main features, translating them into measurable quantities. Starting from a real situation, a specialist (for example, an economist) produces a schematic description of the system using diagrams and symbols. The second stage consists of transforming this schematic representation into a mathematical model, where the translation into symbols and operators takes place. The third stage is the mathematical study of the model, leading to outcomes such as theorems and proofs. Finally, the results are translated back into natural language and interpreted in terms of the original real system. This process does not necessarily end there: since real systems may change over time, the model may need to be re-examined and refined if the outcomes are not satisfactory. Overall, this shows the important role of mathematics in describing real-world systems and deriving useful insights.

Focusing on the topic of this thesis, dynamical systems are real systems that change over time (Bischi et al., 2015) and are described through dynamic variables. This framework has been applied in many fields. Early applications emerged in physics, where differential equations

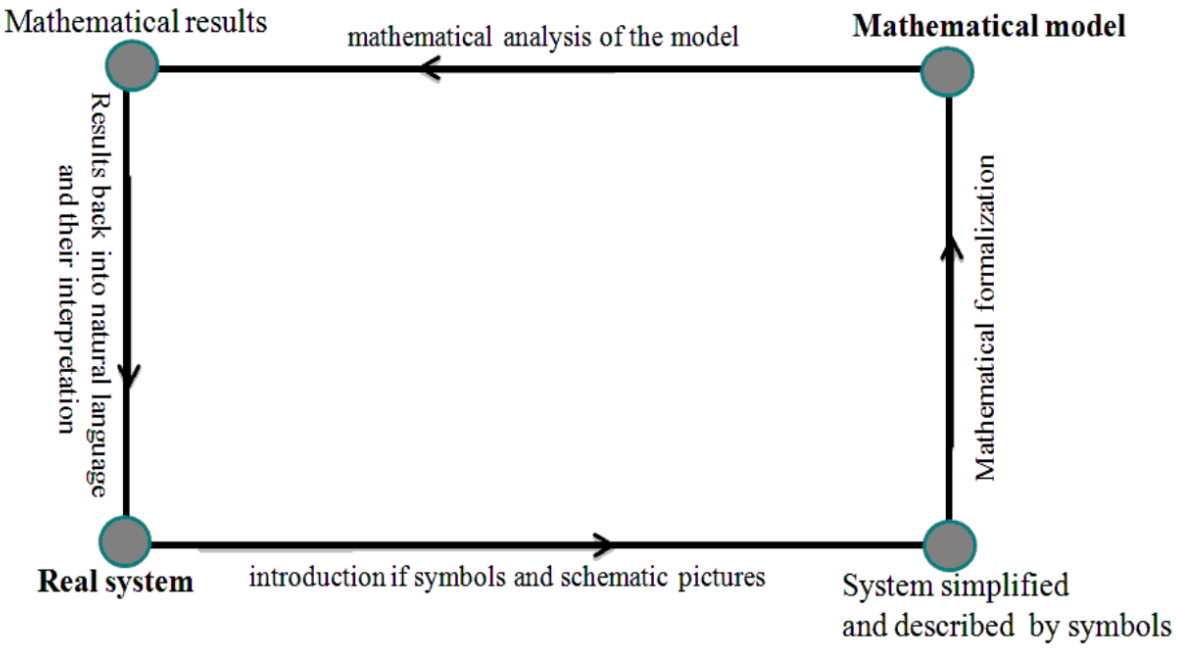

Figure 1.1: The process of mathematical modelling. Bischi et al. (2015)

were used to describe the motion of objects and planets. It has also played an important role in economics, for instance in modeling markets where prices evolve over time. In general, systems and processes whose evolution depends on time can be described by simplified dynamical models. These models characterize the system's state at a given time through time-dependent functions (Bischi et al., 2015). They describe its evolution by equations that express how state variables change over time, enabling one to try to forecast future states based on information about the state at a given time (Bischi et al., 2015). Dynamic modeling and economics have long been closely connected, and this link has become even more relevant in recent years, as advances in both fields have increased the need for mathematical models to describe and analyze how economic systems and policies evolve over time (Bischi et al., 2015). Economics has experienced a significant evolution in its modeling approaches, particularly in the theory of economic agents' decision making, which has enabled the use of mathematical methods to analyze and solve economic problems. The theory of decision making by economic agents is based on the assumption that consumers, producers and governments make decisions in order to optimally allocate scarce resources and achieve their objectives, such as utility maximization for consumers or profit maximization for producers (Mas-Colell et al., 1995). The problem began to be viewed from a different perspective: each step of the process can be interpreted as a repeated decision-making mechanism. According to Bischi et al. (2015), repeated rational decisions based on the comparison between expected and realized outcomes are referred to as "adaptive". A system is described as adaptive because agents progressively collect information

as it evolves, adjust their decisions over time, and evaluate outcomes as circumstances change. Due to this structure, adaptive economic dynamics can be studied within the theory of dynamical systems, for instance through systems of ordinary differential equations. In general, dynamical-systems tools help characterize the long-run implications of agents' decisions, including their convergence toward equilibrium states and the resulting trajectories of the model. To understand the effects of asymptotic behaviours, it is often necessary to perform a global qualitative analysis of the model. In practice, this typically requires numerical methods and computational tools for the analysis and simulation of dynamical systems. The application of dynamical-systems methods to an economic decision-making problem, supported by numerical analysis, is explored in Chapter 3 of this thesis.

1.1.2 Main definitions

In order to understand the role of dynamical systems and their application to optimal control problems, it is necessary to introduce some general definitions that will be used throughout this thesis.

The first definition is "Dynamic Systems Modeling". Before diving into the proper mathematical definition, for a better comprehension of the concept, it is useful to analyze every word of the term. The concept of "dynamic" immediately draws attention to time, which is a fundamental component in the structure of dynamical systems and in explaining how a process evolves (Irwin and Wang, 2017). This concept is used in many contexts, even in areas that differ substantially from one another. For example, some dynamical models are tested using time series data, whereas for other cases time-series observations are not required. Time-series data consists of data that depends on time, where the change in a specific variable is analyzed through repeated measurements over time(Irwin and Wang, 2017). Moreover, time-series data typically exhibit serial dependence, meaning that observations at a given time are correlated with past observations (Irwin and Wang, 2017). In other dynamic models, such as those used in cognitive decision-making, time remains central because response times and agents' choices are analyzed together. Regardless of the context, a common idea remains: dynamic systems models "assume that a system's current states are dependent on past states" (Irwin and Wang, 2017, p.2). The notion of "state" will be introduced more formally in the following definitions. In the context of DSM (Dynamic Systems Modeling), the term "systems" refers to research questions that involve the interaction of multiple components within a larger picture (Irwin and Wang, 2017). Although the interactions between components follow identifiable rules, the resulting

system-level behavior can be complex and difficult to describe in natural language. The change in one element could cause a modification in its overall dynamics (Irwin and Wang, 2017). A central question is whether the system can preserve stability over time. In general, the system stability is not fixed and it varies depending on its behavior in respect of small disturbances or the amount of variability observed across time. Moreover, systems can be classified as more "open" or "closed", depending on whether the system is influenced by external factors. Usually, social systems are always subject to external influences; however, they are often treated as closed in modeling to reduce complexity and improve the research feasibility (Irwin and Wang, 2017). This simplifying assumption will also be adopted in the case study presented in Chapter 3. Finally, in order to determine the system's rules and the interaction between its different components, the system can be described by a mathematical equation or a system of related equations (Irwin and Wang, 2017). In conclusion, the "modeling" component suggests that the "dynamic relationships between components of a system are represented as formal mathematical equations and/or implemented using computational languages" (Irwin and Wang, 2017, p. 2). In general, modeling approaches may be mathematical/computational models or statistical. Mathematical models describe a system through explicit functional relationships, providing a clear formal representation of the underlying theory(Irwin and Wang, 2017). Statistical models, instead, are "generic analytical tools" that focus on testing specific hypotheses losing the "formal representation of the theory" (Irwin and Wang, 2017, p. 2). Computational models implement mathematical models to simulate, analyze and test complex systems that are difficult to handle analytically. In the third chapter, we rely on a computational implementation to simulate and solve the Ramsey problem using the shooting method.

Here, I clarify the difference between Dynamic Systems Modeling (DSM) and Dynamical Systems. DSM describes and predicts interactions among multiple components of a system evolving over time (Irwin and Wang, 2017). A dynamical system is "a mathematical description of an evolving system, that is a real system whose state changes as time goes on" (Bischi et al., 2015, p. 7).

As for the mathematical definition, a dynamical system can be described in terms of its state, represented by a finite number n of measurable quantities called "state variables", denoted by real number

$$x_1, x_2, .., x_n$$

where $x_i \in \mathbb{R}$, $i = 1, .., n$ (Bischi et al., 2015). The collection of these variables defines the state vector $x = (x_1, .., x_n) \in \mathbb{R}$. Geometrically, in a one-dimensional dynamical system the

state can be represented by a point on a line, whereas in a two-dimensional system it corresponds to a point in the Cartesian plane (Bischi et al., 2015).

The second definition concerns the concept of "state". According to Irwin and Wang (2017), dynamic system models are characterized by three key elements: the state of the system, the state-space, and the state-transition function. The state of the system represents all information needed to describe the system at a specific moment in time (Irwin and Wang, 2017). The state-space refers to the set of all feasible states the system can take. Finally, the state-transition function specifies how the system evolves over time by mapping the different states.

Moving to a more formal definition, the "state space $M \subseteq \mathbb{R}^n$ is the set of admissible values of the state variables" (Bischi et al., 2015, p. 8). Since a dynamical system evolves over time, also the state variables are time-dependent and therefore not fixed. In particular, $x_i = x_i(t)$ for i = 1,...,n, where time t could be a real number ($t \in \mathbb{R}$) in continuous time systems, or a natural number ($t \in \mathbb{N}$) in discrete time systems. The discrete-time setting is an appropriate assumption for event-driven systems, meaning that changes in the state variables are observed at specific time steps as a consequence of discrete events (Bischi et al., 2015). We will make a deeper analysis of these two cases in the following section. In general, the main goal of dynamical systems is: knowing the state of a system at a specific point in time t_0, being able to predict its state at a different time $t \neq t_0$ (Bischi et al., 2015). The idea can be formalized by introducing an operator G such that the state at time t is obtained from the state at time t_0 through:

$$\mathbf{x}(t) = \mathbf{G}(t; x(t_0)) \tag{1.1}$$

Here, $x(t)$ denotes the state vector, collecting the state variables $x_1(t), ..., x_n(t) \in \mathbb{R}^n$, and **G** is a vector-valued mapping that acts on the state space and returns the corresponding state at time t (Bischi et al., 2015). In fact, once the "evolution operator" G is given, the future state of the system at any time $t > t_0$, as well as its past state at any time $t < t_0$, can be determined from the "initial condition" $x(t_0)$ (Bischi et al., 2015). As t changes, the vector function in 1.1 $\mathbf{x}(t)$ describes the "parametric equations for a trajectory"(Bischi et al., 2015, p. 8). Depending on the setting, the representation of a trajectory changes. In continuous time ($t \in \mathbb{R}$), the representation of the trajectory is a curve in the space $\mathbb{R}^n$ and according to the viewpoint they assume different names (Figure 1.2). In the $n + 1$ dimensional space $(\mathbb{R}^n, t)$, the curve is called an integral curve.

Whereas, in the state space $\mathbb{R}^n$, the curve is denoted as a phase curve and the arrows represent the increasing direction.

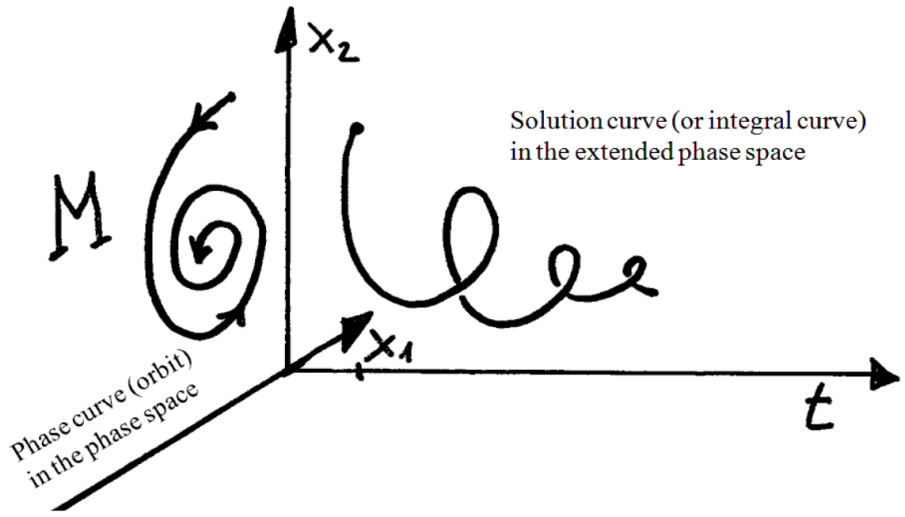

Figure 1.2: Integral curve and its projections. (Bischi et al., 2015)

In discrete time, instead, a trajectory is represented by a sequence of points. As illustrated in figure 1.3, these points describe the time evolution of the system from one point to another and they are connected by line segments to visualize its growth as time increases.

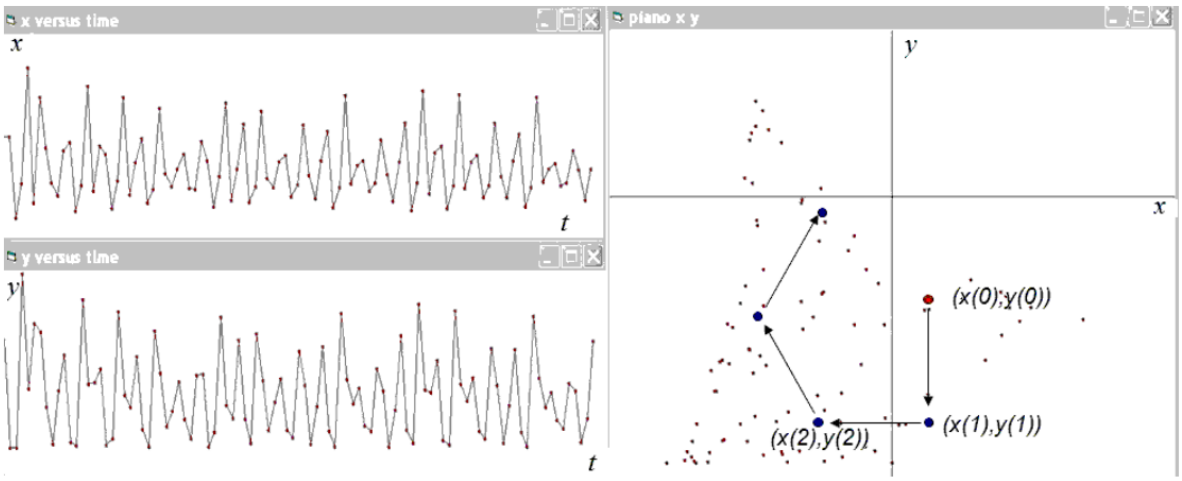

Figure 1.3: Discrete time dynamical system in two dimensions. (Bischi et al., 2015)

The third concept is related to the definitions of equilibrium and stability in dynamical systems. The notion of equilibrium is very broad and may assume different meanings across disciplines. According to Park (2007), equilibrium is defined as "a state of balance in a system that is produced and maintained by a variety of forces which may increase or decrease but they always cancel each other out, producing a steady state". For example, in physics, the definition

of equilibrium is based on Newton's law of motions and on the relationship between force, mass, length and time (Thorn and Welford, 1994). In this context, an object is in equilibrium when its velocity is constant, or when it is at rest (Thorn and Welford, 1994). The key idea is that an objects is not free from acting forces; rather the sum of all forces acting on it is equal to zero, so that they cancel each other out (Thorn and Welford, 1994). All these definitions express the same concept, although from different perspectives. In mathematics, "an equilibrium (stationary state or fixed point) $x^* = (x_1^*, ..., x_n^*)$ is a particular trajectory such that all the state variables are constant

$$\mathbf{x}(t) = \mathbf{G}(t, x^*) = x^* \tag{1.2}$$

for each $t > t_0$" (Bischi et al., 2015, p. 9). The equilibrium can be interpreted as a "trapping point', since its trajectory remains unchanged at every subsequent moment in time (Bischi et al., 2015). Indeed, if $x(t_0) = x^*$, then $x(t) = x^*$ for $t \geq t_0$. This concept can be broadened from single points to any subset of the phase space. To clarify this idea, let's consider the definition of trapping set. According to Bischi et al. (2015, p. 9), "A set $A \subseteq M$ is trapping if $x(t_0) \in A$ implies $x(t) = G(t, x(t_0)) \in A$ for each $t > t_0$". Meaning that once a trajectory enters the trapping set, it cannot leave it. A more restrictive concept is that of an invariant set. According to Bischi et al. (2015), a closed set $A \subseteq M$ is invariant if $G(t, A) = A$, i.e. each subset $A' \subset A$ is not trapping. This means that any trajectory beginning in A remains in A for all future times. Moreover, the trajectory will also touch all the points inside the invariant set. There are many kinds of invariant sets and the equilibrium point is one of those (Bischi et al., 2015). When the trajectory does not start inside the invariant set but rather in the proximity of it, different outcomes may occur (Bischi et al., 2015). The trajectory may converge to the invariant set, remain in the neighborhood of it or diverge in a different path (Bischi et al., 2015). For this reason, it becomes essential to introduce the concept of stability of an invariant set.

First, lets recall the concept of stability. The root of the word "stable" derives from the Latin adjective *stabilem*, which signifies "being able to stand firmly" (Pradeep and Shrivastava, 1990, p. 385). In other words, stability denotes the ability to resist to changes or movements (Mellodge, 2016). The notion of stability is closely related to that of equilibrium, since equilibria are often characterized according to their stability properties. Although stability does not admit a single official definition in the scientific literature, it is commonly described in terms of a system response to "small disturbances in its state" (Pradeep and Shrivastava, 1990, p. 385). Basically, the study of stability consists in perturbing a system slightly from its equilibrium and

observing its subsequent motions: if the system remains close to the equilibrium, then it is said to be stable; otherwise it will be unstable (Pradeep and Shrivastava, 1990). Figure 1.4 illustrates different notions of stability through an analogy with motion in a gravitational field, showing how an object may exhibit asymptotically stable, stable, or unstable behavior depending on its position.

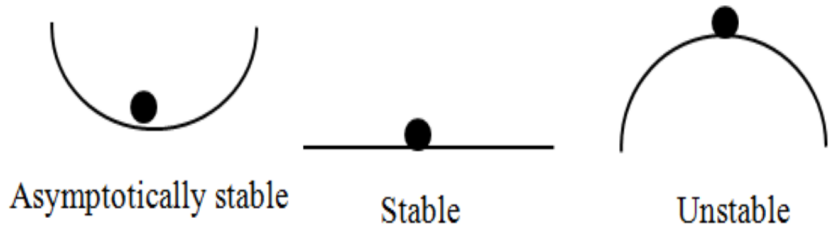

Figure 1.4: Analogy with gravitational field. (Bischi et al., 2015)

There are many ways to formalize the concept of stability. In this thesis, we are going to focus on Lyapunov stability, asymptotic stability and exponential stability.

According to Bischi et al. (2015), an invariant set A is Lyapunov stable if for each neighborhood U of A there exists another neighborhood V of A, with $V \subseteq U$, such that any trajectory starting from V remains inside U. To better understand this definition, it is useful to consider the possible behavious of trajectories depending on the stability properties of the invariant set. In particular, if a trajectory starts outside the invariant set A but sufficiently close to it, two different outcomes may occur: the trajectory may remain close to A, in which case the set is stable, or it may move away from it, indicating instability (Bischi et al., 2015). Let's now focus on a stronger notion: the asymptotic stability. An invariant set A is asymptotically stable (and it can be called an attractor) when:

- A is stable, according to the Lyapunov stability definition
- $lim_{t \longmapsto +\infty} \mathbf{G}(t, x) \in A$ for each initial condition $\mathbf{x} \in V$ (Bischi et al., 2015)

Figure 1.5 illustrates the behavior of trajectories for different stability properties of invariant sets, highlighting the role of neighborhoods and trapping behavior. In the asymptotically stable case, the trajectory starts sufficiently close to the invariant set A, but outside it, then in the long run converge to it (Bischi et al., 2015). At first glance, the second condition of asymptotic stability may appear stronger than the first one. However, Lyapunov stability is a fundamental requirement: it may happen that trajectories starting close to A eventually converge to it, but leave any prescribed neighborhood $U \supset A$ before doing so (Bischi et al., 2015). In this case,

the Lyapunov stability condition will be violated. Hence, for asymptotical stability, Lyapunov stability is a necessary condition (Bischi et al., 2015).

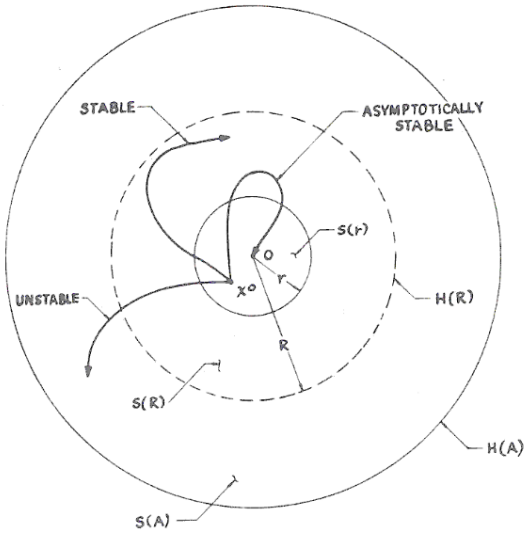

Figure 1.5: Stability, asymptotic stability and instability. (Bischi et al., 2015)

These definitions are expressed in terms of neighborhoods, but can be equivalently formulated using the Euclidean norm in $\mathbb{R}^n$, defined as $||x|| = \sqrt{\sum_{i=1}^{n} x_i^2}$ and $||x-y|| = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$ (Bischi et al., 2015). So, we can now restate the definitions that refer to an equilibrium point. An equilibrium $\bar{x}$ can be Lyapunov stable, if for every $\epsilon > 0$ there exists $\delta > 0$ such that if $|x - \bar{x}| < \delta$ then $|\Phi_t(x) - \bar{x}|$ for all $t \geq 0$ (Hunter, 2011). Moreover, if the equilibrium $\bar{x}$ is Lyapunov stable and, in addition, "there exists $n > 0$ such that if $|x - \bar{x}| < n$ then $|\Phi_t(x) \rightarrow \bar{x}|$ when $t \rightarrow \infty$" (Hunter, 2011, p. 9). It is important to note that the Lyapunov stability does not imply asymptotic stability. However, Lyapunov stability is a necessary condition for asymptotic stability: without it, a trajectory could converge to the equilibrium only after large excursions away from it, which would violate the notion of Lyapunov stability (Hunter, 2011). To better understand the concept, figure 1.6 illustrates typical phase portraits associated with stable, asymptotically stable, and unstable equilibria in two-dimensional dynamical systems.

Regarding exponential stability, "an equilibrium point x^* is exponentially stable if there exist $\alpha > 0$ and $\lambda > 0$ such that $||x(t) - x^*|| < \alpha e^{-\lambda t}||x(0) - x^*||$ for all $t \geq 0$" (Mellodge, 2016, p.179). This definition is stronger than asymptotic stability. In fact, exponential stability of an equilibrium point implies both Lyapunov stability and asymptotical stability. From the point of view of the trajectory, in the case of exponential stability the system converges to the equilibrium point at an exponential rate, meaning that the speed of convergence can be measured with an exponential function (Mellodge, 2016). Up to this point, we have referred to local definitions and how the behavior of a dynamical system when the initial conditions lie in a neighborhood

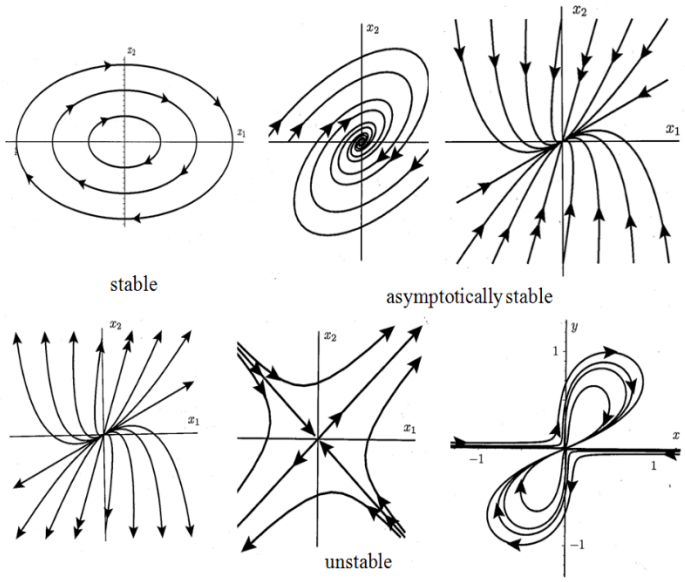

Figure 1.6: Phase portraits. (Bischi et al., 2015)

of an invariant set. However, it is also important to investigate whether the trajectory returns to the equilibrium after larger perturbations and how far from the equilibrium the system can be displaced while still converging back to the equilibrium (Bischi et al., 2015). This question leads to the next concept: the basin of attraction.

The fourth notion we introduce is the Basin of attraction. According to Bischi et al. (2015), the basin of attraction of an attractor A is the set of all points $x \in M$ such that

$$lim_{t \longmapsto +\infty} \mathbf{G}(t, x) \in A \tag{1.3}$$

Therefore, the basin of attraction can be written as

$$B(A) = \left\{ x \in M \;\middle|\; \lim_{t \to +\infty} \mathbf{G}(t, x) \in A \right\} \tag{1.4}$$

In other words, the basin of attraction is the set of initial states in the phase space whose trajectories come together to the same attractor (Nouri and Seyyedsalehi, 2023). In particular, an attractor A is referred to as a global attractor when $B(A) = M$, meaning that every point in the state space M converges to A (Bischi et al., 2015). The size of the basin of attraction is often interpreted as an indicator of the robustness of the system with respect to exogenous perturbations (Bischi et al., 2015). However, this interpretation is not always accurate. In fact, according to its shape, a larger basin of attraction does not necessary imply greater robustness. This behavior is illustrated, for example, in the left sketch of figure 1.7. Additionally, there are

cases in which a stable equilibria may be more sensitive to perturbation than an unstable but controlled equilibrium, as shown in the right sketches of figure 1.7). Furthermore, the speed

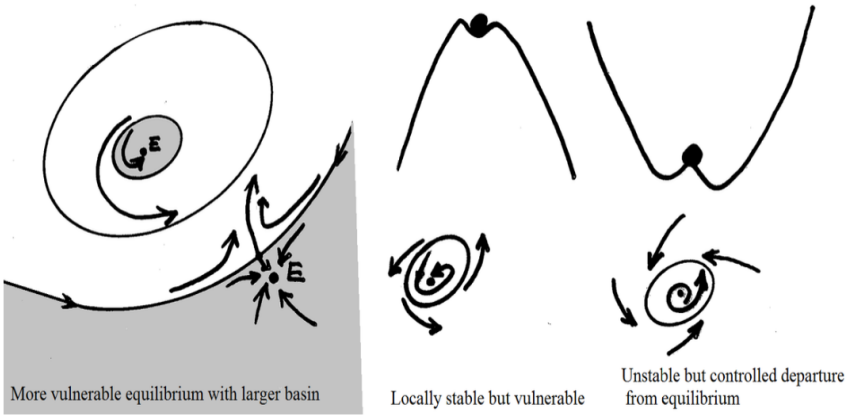

Figure 1.7: Local stability extent. (Bischi et al., 2015)

at which the trajectories converge to the attractor is another important factor to consider. In particular, the time required for the effect of a perturbation to decay plays a crucial role (Bischi et al., 2015). The problem is that in some cases the convergence could take too long and this could affect the result in the real system. For instance, when studying the response of an economic system to a crisis, the time needed to recover and return to the pre-crisis equilibrium is a key factor. If this recovery time is too long, it may lead to significant economic consequences.

To fully understand the behavior of dynamical systems, several factors must be taken into account. An important one is the representation of the trajectories. However, this task can be challenging, since the evolution of the operator **G** is often difficult to represent due to the complexity of the initial conditions and the difficulty in expressing it through simple functions (Bischi et al., 2015). In general, dynamical systems are formulated through evolution equations, or laws of motion, that describe how the system changes over small time intervals (Bischi et al., 2015). In continuous time settings, these local evolution laws are typically expressed by systems of ordinary differential equations (ODE), whereas in discrete time settings they are described by systems of difference equations. The system of ordinary differential equations is formulated as follows (Bischi et al., 2015):

$$\begin{cases} \dfrac{dx_i(t)}{dt} = f_i\big(x_1(t), \ldots, x_n(t); \alpha\big), \quad i = 1, \ldots, n, \\ x_i(t_0) = \bar{x}_i \end{cases} \tag{1.5}$$

where:

- the time derivative $\frac{dx_i(t)}{dt}$ represents the rate of change of the state variable $x_i(t)$ with respect to time;

- the functions $f_i(x_i(t), ..x_j(t))$ describe how the state variable x_i (self–interaction) and the other state variables x_j, $j \neq i$ (cross–interaction), affect its rate of change;

- $\alpha = (\alpha_1, ...\alpha_m), \alpha_i \in \mathbb{R}$ formulates m real parameters fixed along a trajectory that can take different numerical values to expose exogenous influence on dynamical systems, such as how different policies impact the outside environment.

Whenever the parameter α_i assumes some variations, these modifications are called structural modifications, since they can modify both the shape of the functions f_i and the property of the trajectory (Bischi et al., 2015). Usually, in dynamical systems instead of using the Leibniz notation $\frac{dx}{dt}$, it is used the Newton's notation $\dot{x}$. So, the differential equations in 1.5 can be rewritten as follows:

$$\left\{\dot{x}_i = f_i\big(x_1(t), \ldots, x_n(t); \alpha\big), \quad i = 1, \ldots, n \right. \tag{1.6}$$

Even if differential equations can be of the order greater than one, they can always be reformulated into differential equation systems of order one in the form of 1.5 with the introduction of auxiliary variables (Bischi et al., 2015). For example, this second order differential equation

$$\ddot{x}_i + \alpha\dot{x} + bx(t) = 0 \tag{1.7}$$

with initial conditions $x(0) = x_0$ and $\dot{x}(0) = v_0$ can be rewritten in the form 1.6 with the definitions of these variables: $x_1(t) = x(t)$ and $x_2(t) = \dot{x}(t)$. So, the equivalent system becomes:

$$\begin{cases} \dot{x_1} = x_2, \\ \\ \dot{x_2} = -bx_1 - ax_2 \end{cases} \tag{1.8}$$

According to Bischi et al. (2015), the model can be called nonautonomous when along the trajectory the parameters α vary with respect to time, for example, when $\alpha_i = \alpha_i(t)$ are functions of time. However, in this case, the non autonomous model can also be rewritten in the normal form 1.5 as an equivalent autonomous model of dimension $n+1$. The key is always to introduce a dynamic variable $x_{n+1} = t$ whose evolving time is controlled by the first order differential equation $\dot{x}_{n+1} = 1$ (Bischi et al., 2015).

An important property of a system of ordinary differential equations is the theorem of existence and uniqueness of solutions. In particular, "if the functions f_i have continuous partial

derivatives $\frac{df_i}{dx_k}$ in M and $x(t_0) \in M$, then there exists a unique solution $x_i(t), i = 1, .., n$, of the system 1.5 such that $x(t_0) = \bar{x}$, and each $x_i(t)$ is a continuous function" (Bischi et al., 2015, p. 15). This condition implies local Lipschitz continuity with respect to the state variables.

With respect to discrete-time systems, the evolution equations are formulated through systems of difference equations. Difference equations describe the time evolution of the system as a sequence of discrete points starting from a given initial condition (Bischi et al., 2015)

$$\begin{cases} x_i(t+1) = f_i(x_1(t), ...x_n(t); \alpha), \quad i = 1, ..., n \\ x_i(0) = \bar{x_i} \end{cases} \tag{1.9}$$

As in the continuous time case, difference equation of a higher order can always be rewritten as a system of first order difference equations. For example, the second-order difference equation

$$x(t+1) + ax(t) + bx(t-1) = 0 \tag{1.10}$$

with initial conditions $x(-1) = x_0$ and $x(0) = x_1$ can be restated as the following first order system:

$$\begin{cases} x(t+1) = -ax(t) - by(t) \\ y(t+1) = x(t) \end{cases} \tag{1.11}$$

where $y(t) = x(t-1)$, starting from the initial conditions $x(0) = x_1$, $y(0) = x_0$. Nonautonomous difference equations can also be transformed into equivalent autonomous systems. For instance, the nonautonomous equation:

$$x_{(t+1)} = f(x_t, t) \tag{1.12}$$

can be reformulated as the autonomous system:

$$\begin{cases} x_{(t+1)} = f(x_t, y_t) \\ y_{(t+1)} = y_t + 1 \end{cases} \tag{1.13}$$

where $y_t = t$. In general, the study of ordinary differential equations 1.5 and difference equations 1.9 represents the main approach to dynamical systems in continuous and discrete time, respectively. Both provide a local description of how systems evolve over time (Bischi et al., 2015). However, for the purpose of this thesis, we focus exclusively on differential equations since the analysis concerns continuous-time dynamical systems.

1.1.3 Types of Dynamical Systems

Dynamical systems play a fundamental role in mathematics and in the context of this thesis, as they provide a framework to analyze, predict, and control the behavior of the system (Mellodge, 2016). It is therefore important to distinguish between the different types of dynamical system that exist, since the methods to solve a system depend on its specific characteristics. In fact, dynamical systems can be classified according to their features. In this paragraph, we discuss the main classifications that are relevant for the problem analyzed in chapter 3 and clarify which characteristics are associated to that model.

Let us begin with the most important distinction in dynamical systems, which concern the concepts of time and state space. In particular, a dynamical system can be classified as either continuous-time or discrete-time. These concepts were already introduced in general terms in the previous section; here, they are specified in more detail. Time is considered continuous when represented as a continuum of successive time instants (Ossimitz and Mrotzek, 2008). In contrast, time is discrete when modeled as a sequence of equidistant points with unique consecutive values (Irwin and Wang, 2017). In continuous time systems, the evolution of the system over a given time-span is described by a continuous function over time (Ossimitz and Mrotzek, 2008). Usually, these functions are differentiable and, as illustrated in the previous section, can be specified by differential equations. That is, the changes in the state variable are captured by the first derivative of the state function. In this specific model, it is not meaningful to partition time into discrete points and intervals since its main characteristic is to get distinct values for each instant in time. For example, consider the different heights of a car after passing over a bump (Ossimitz and Mrotzek, 2008). The height of the car can be observed at any moment, and the results are in figure 1.8. The curve produced by the changing height of the car is smooth and the time is represented on a continuous axis.

With regard to discrete time models, time is represented as a sequence of equally spaced time points or intervals, and the system state is only observed at those specific instants. A key feature of this model is that both the number of time intervals and time points are finite and defined. Moreover, the number of time points is equal to the number of time intervals plus one (Ossimitz and Mrotzek, 2008). To better understand the discrete model, it can be seen as a bank account in which an amount is invested at a fixed interest rate and updated periodically. The invested amount changes only once a month, so for every month there is a single value. In this case, only the value at the end of the month is relevant, while all the other values in-between are not. As illustrated in figure 1.9, the distinction between the two modeling approaches becomes

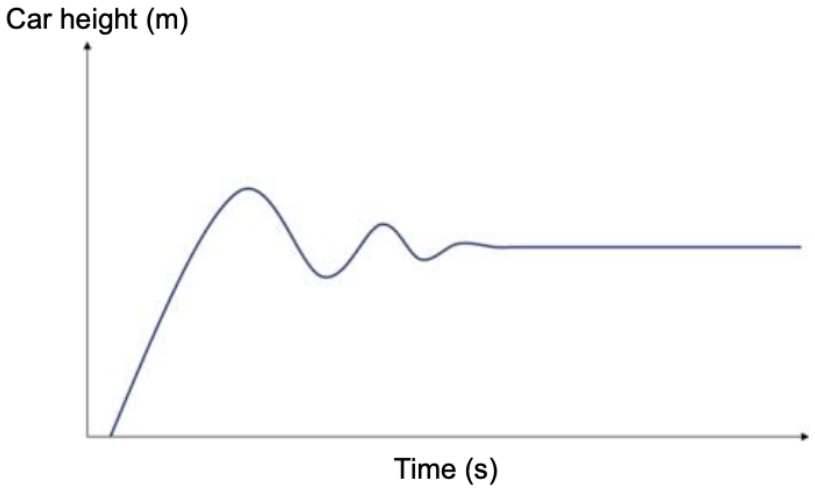

Figure 1.8: Car's height evolution over time as a continuous-time signal. (Mellodge, 2016)

evident. On the right there is the continuous time model, which is represented by a linear time axis; while on the left the discrete time model is represented by an axis divided into equally distributed time points and time intervals. A practical guideline to choose between the two frameworks is to examine the available data (Ossimitz and Mrotzek, 2008). If the number of given data is finite, then the discrete model will be applied. On the other hand, if the data are continuous in time and can always be observed at any point in time, as waves in signals, then the continuous model is the right fit.

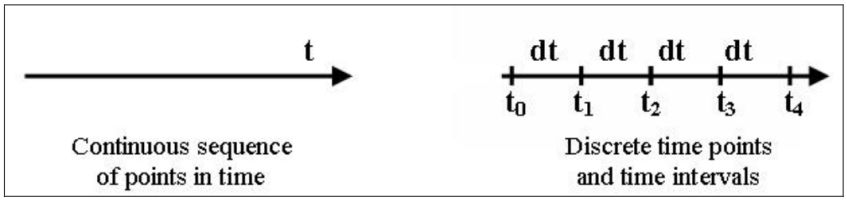

Figure 1.9: Continuous Time VS Discrete Time. (Ossimitz and Mrotzek, 2008)

To conclude, we present the corresponding mathematical definitions. A continuous-time dynamical system on $\mathbb{R}^n$ is defined by a system of ordinary differential equations

$$\dot{x} = f(x), \qquad x \in M, \tag{1.14}$$

where M is an open subset of $\mathbb{R}^n$ or, more generally, an n-dimensional manifold (Benettin, 2012). A discrete-time dynamical system is defined by an iterated map (or system of difference equations)

$$x_{t+1} = \Phi(x_t), \tag{1.15}$$

which generates the trajectory

$$x_t = \Phi^t(x_0), \qquad t \in \mathbb{N}, \tag{1.16}$$

where Φ^t denotes the t-th iterate of the map Φ (Benettin, 2012). As already specified, in the third chapter the analyzed problem will be a continuous time dynamical system.

The next distinction is based on the linearity of a dynamical system. This differentiation is based on the relationships between the state variables. A dynamic system can be linear or non linear. For sure linear systems are more simple than non linear ones. In any case, it is not always true that a linear dynamic system model produces a linear output. In fact, the behavior of the model depends on the influence of the previous system states and on compliance with the laws of superposition (Mellodge, 2016). A linear system can be described in different ways: if it has a transfer function, if the sinusoidal input generates a sinusoidal output of the same frequency, and if it obeys the law of superposition (Mellodge, 2016). In order to check for system's linearity, the most common way is to verify whether it follows the law of superposition. This law has two components: scaling and additivity. According to Mellodge (2016, p. 8), for the scaling property, "if the system's input $x(t)$ results in an output $y(t)$, then an input of $\alpha x(t)$ will result in the output $\alpha y(t)$ for any value α". This means that if the system's input is multiplied by a value, the output will also be multiplied by the same value. Moreover, in a linear system, it does not matter whether the value is multiplied before or after the system, the output will not change. For the additivity part, "if a system's input $x_1(t)$ results in $y_1(t)$ and $x_2(t)$ results in $y_2(t)$, then if the input is $x_1(t) + x_2(t)$, the output is $y_1(t) + y_2(t)$" (Mellodge, 2016, p. 9). In other words, the signal can be summed before or after entering the system and the output signal will be the same in both cases. In mathematics, a linear system can be written in matrix form (Bischi et al., 2015):

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} \tag{1.17}$$

where

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}; x(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}; \dot{x}(t) = \begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{pmatrix} \tag{1.18}$$

On the other hand, a nonlinear system is a system that is not linear (Mellodge, 2016). In other words, a nonlinear system can be identified when the linearity conditions do not hold. For example, this system $y = mx + b$ where m and b are nonzero constants, is a linear equation, but it is a non linear system. This is because the scaling property does not hold. In fact, if we consider $x_0(t)$ the input of the system and $y_0(t)$ its output and multiply the input by α, the corresponding

output y_1 will be

$$y_1 = m(\alpha x_0) + b \tag{1.19}$$

However, if we multiply the output by α, the results will be different

$$\alpha y_0 = \alpha(mx_0 + b) = m(\alpha x_0) + \alpha b \tag{1.20}$$

Since y_1 is different from αy_0, the scaling property does not hold and the system is nonlinear. The same happens for the additivity property. If we consider the same structure as before and consider two inputs $x_1(t)$ and $x_2(t)$ and the corresponding outputs $y_1(t)$ and $y_2(t)$. Then the sum of the inputs in the system will result in this output:

$$y_s = m(x_1 + x_2) + b \tag{1.21}$$

while

$$y_1 + y_2 = mx_1 + b + mx_2 + b = m(x_1 + x_2) + 2b \tag{1.22}$$

Since the outputs are different, the additivity problem does not apply, hence the system is non linear. One of the properties of non linear systems is that they can admit more than one equilibrium point at a time (Bischi et al., 2015). The last specification for a nonlinear system is the definition of a bifurcation. According to Mellodge (2016, p. 227), bifurcation is the "change in behavior resulting from a small change in a parameter". Behavior is usually defined as the change in the number of equilibrium points or in the type of equilibrium points, if stable or unstable (Mellodge, 2016). The change in the parameters is referred to as a threshold that separates the system into two: above it, the system assumes a particular behavior, and below it exhibits another. The Ramsey model analyzed in this thesis belongs to the class of nonlinear dynamical systems, where nonlinearity arises from exponential functional forms appearing in the utility function and in the capital accumulation dynamics.

The next classification concerns whether a dynamical system is autonomous or nonautonomous, depending on how the coefficient of a model is treated. In fact, it can be treated either as a constant or as a function of time itself (Irwin and Wang, 2017). In the case where the coefficient behaves as a constant, the system will be time-invariant or autonomous. Meaning the system does not depend on time. In particular, the dynamic system is defined by coefficients that do not change over time (Mellodge, 2016). On the other hand, if the coefficients of a system change over time, it means the system is nonautonomous or time-variant. Suggesting that the

system explicitly depends on time. Of course, the relationship between the components in a system could be constant for some time and then vary. For example, the interest rate in a bank account can be constant for several years, meaning that the system is autonomous over that period, and then it could vary due to policy changes, turning the system into a nonautonomous one. From a system-theoretic perspective, a common way to identify time invariance is to examine whether a shift in time in the input is also reflected in the output(Mellodge, 2016). In fact, if the input of $x(t)$ generates an output $y(t)$, and the input $x(t-\tau)$ produces the output $y(t-\tau)$ for any value of τ, then the system is autonomous. However, in the context of dynamical systems defined by differential equations, the distinction is made directly in the mathematical formulation of the model. As discussed in the previous section, any nonautonomous system can always be reformulated into an equivalent autonomous one by introducing a dynamic variable.

The last distinction concerns deterministic and stochastic dynamical systems. In a deterministic model, given the current state of the system, the future state of the system can be fully defined (Irwin and Wang, 2017). In contrast, in a stochastic model, the future state of the system cannot be predicted with certainty, but only in probabilistic terms (Irwin and Wang, 2017). Usually, in real world systems, the dynamic systems are stochastic since there are uncertainty, imprecision, and unpredictability that cannot be anticipated. In these cases, dynamic systems are affected by random disturbances and variables, commonly referred as noise. As a consequence, the future state of the dynamical system is predicted based on some known probabilistic properties that have been assumed (Mellodge, 2016). In this thesis, the analysis is restricted to deterministic dynamical systems.

To conclude this section, we will take into account the dimensionality of dynamical systems. The dimensionality of a dynamical system corresponds to the number of dynamical variables that determine the dimension of the phase space (Strogatz, 2024). Depending on this number of variables, the dynamic system can be classified as one-dimensional, two-dimensional, or n-dimensional. One-dimensional dynamical systems are characterized by a single state variable and can be represented by a vector field on a line. This category is the simplest type of dynamical system, but it also presents some limitations. In fact, one-dimensional systems can only exhibit simple behavior, like fixed points and bifurcation, hence are used to study at a theoretical level than to a practical one. Two-dimensional dynamic systems are characterized by two variables and can be represented in the phase plane (Bischi et al., 2015). Dynamics can be qualitatively studied using tools such as nullclines, which allow one to identify equilibrium

points and stability (Bischi et al., 2015). Some examples of dynamical systems in two dimensions can be the pendulum and the linear oscillator (Strogatz, 2024). In particular, two-dimensional dynamic models can exhibit oscillatory behavior. Finally, the n-dimensional dynamical system, with $n > 2$, involves more than two dynamic variables (Bischi et al., 2015). The representation of this dynamic system is really complex, and it is necessary to use a numerical tool for its analysis. Moreover, when the variables are more than three, nonlinear dynamical systems may present chaotic trajectories and weird attractors, which are often associated with fractals (Strogatz, 2024). The Ramsey problem analyzed in the third chapter belongs to this category of n-dimensional nonlinear dynamical systems. For this reason, we need numerical techniques, such as the shooting method implemented in python, to solve the model.

1.2 Optimal Control Problems

In the previous section, we analyzed the main types of dynamical systems and their main characteristics. It is now important to understand how to control and modify the behavior of dynamical systems in order to achieve the desired objectives (Grass et al., 2008). Optimal control theory, together with the maximum principle for finding optimal trajectories, is what is needed. In particular, these tools, combined with numerical methods, will be fundamental to solve the Ramsey problem in the last chapter of this thesis. The optimal control theory was founded in order to solve mathematics and engineering problems. This theory is formed by two parts that were discovered subsequently: control theory and optimization. Control theory had its foundation in the middle of the twentieth century through the development of automatic control, a field that was extensively studied and needed to be recognized as official (Grass et al., 2008). Many scholars from all fields have contributed to the development of this theory, and, in fact, it is one of the most interdisciplinary theories in science. Among the most influential contributions, which we will explore in more detail in the following sections, are Bellman's dynamic programming and Pontryagin's Maximum Principle (Grass et al., 2008). The other component of optimal control is optimization. This mathematical field studies the "improvement of a decision variable in order to minimize cost or to maximize a utility" (Grass et al., 2008, p. 102). Historically, control theory is based on the calculus of variations, and from this foundation, after the second world war, it has been extended as optimization theory (Grass et al., 2008). One of the first optimization problems was Fernat's derivation of the law of refraction to solve a minimum time problem in 1662. Later, Newton applied optimization theory to define the optimal shape of a ship's bow (Grass et al., 2008). Extensions of optimization theory also

appeared in dynamic economics, especially with Evans' pricing problem for monopolists and Ramsey (1928), who solved the capital accumulation problem by applying Euler's equation (Grass et al., 2008). Pontryagin's Maximum Principle also has its foundation in engineering problems, in particular in solving steering aircraft in the Soviet Union in the 1950s (Grass et al., 2008). Many researchers contributed to the creation of the Maximum Principles and it has many independent roots. For sure one thing is certain, optimal control is an extension of the calculus of variations and it has Russian roots. For instance, the control variable denoted by u originates from the Russian word *upravlenije*, which means control (Grass et al., 2008). After all these studies, optimal control theory has started to be applied to all possible fields, such as economics and management, and from there researchers continue to expand their analysis into new areas and new technologies.

1.2.1 Problem formulation

In this section, the mathematical description of the standard optimal control problem is presented. The standard optimal control problem is characterized by different components: the state variable of the system and the state dynamics, the initial and the terminal time, the control value and the control region, the constraints, the instantaneous payoff or profit, the scrap or salvage value (or terminal payoff) and the discount rate (Grass et al., 2008). The optimal control problem in standard form is defined as:

$$\max_{u(\cdot)} \quad \int_0^T e^{-rt}\, g\big(x(t), u(t), t\big)\, dt \;+\; e^{-rT}\, S\big(x(T), T\big) \tag{1.23a}$$

$$\text{s.t.} \quad \dot{x}(t) = f\big(x(t), u(t), t\big), \qquad t \in [0, T] \tag{1.23b}$$

$$x(0) = x_0 \tag{1.23c}$$

$$x_i(T) = x_i^T, \qquad i = 1, \ldots, n' \tag{1.23d}$$

$$x_i(T) \geq x_i^T, \qquad i = n' + 1, \ldots, n'' \tag{1.23e}$$

$$x_i(T) \text{ free}, \qquad i = n'' + 1, \ldots, n \tag{1.23f}$$

$$u(t) \in \Omega\big(x(t), t\big), \qquad \forall\, t \in [0, T]. \tag{1.23g}$$

Let's analyze each component of the optimal control problem. First, the state of the system at time t is defined by a n-dimensional real column vector (Grass et al., 2008):

$$x(t) = (x_1(t), x_2(t), ..., x_n(t))^\top \in \mathbb{R}^n, \quad t \in [0, T] \tag{1.24}$$

In the optimal control problem, the evolution of this state variable 1.23b is presented and it is denoted as state dynamics. Here, T refers to the *terminal time* of the process and 0 refers to the initial time. The terminal time can also be ∞, which is called an infinite time horizon problem, and it will be considered for the problem in chapter 3. There are some consequences when $T = +\infty$: first, no condition is usually imposed on the state 1.24, but sometimes it is required that $lim_{t\longmapsto+\infty} x(t) \geq \bar{x}$; and second, there is no scrap value $S(x(T),(T)) = 0$ (Bischi et al., 2015). The next component is the control u. The control represents a choice variable that the decision-maker can continuously set within the set Ω (Bischi et al., 2015). In the optimal control problem, the control u is in both instantaneous profit and as a constraint in 1.23g. Moreover, the control is piecewise continuous and must be within the control region Ω (Grass et al., 2008). As we saw, the state can be represented by the ODE 1.23b with the initial condition 1.23c and together with the choice of the control, it present the trajectory of the state (Bischi et al., 2015). According to Grass et al. (2008, p. 105), the ODE represents "how the current state $x(t)$ and the decision-maker's action $u(t)$ at time t influence the rate of change of the state at time t". The ODE is subject to the following terminal constraints 1.23d, 1.23e and 1.23f; where $n' \geq 0, n'' \geq 0, n' + n'' \leq n$. Finally, the main component: the objective function 1.23a. The objective function is the total discounted payoff over $[0, T]$, and it can be written as:

$$V(u(\cdot)) = \int_0^T e^{-rt}\, g\big(x(t), u(t), t\big)\, dt \;+\; e^{-rT}\, S\big(x(T), T\big). \tag{1.25}$$

The continuous function $g(x(t), u(t), t)$, where $g : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}$, denotes the instantaneous profit that comes from the application of the control $u(t)$ at time t when the current state $x(t)$ and the discount rate r are nonnegative (Grass et al., 2008). The term $S(x(T), T)$ defines the salvage or scrap value and describes the payoff or the reward for being in the state $x(T)$ at the end of the horizon T (Grass et al., 2008). Moreover, the salvage value $S : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$ is assumed to be continuously differentiable with respect to x and T.

Before moving on to the next sections, it is necessary to make some clarification. First of all, a necessary condition to find the optimal state and control is that they need to be admissible. According to Grass et al. (2008, p. 105), "The pair of functions $(x(\cdot), u(\cdot)) : [0, T] \to \mathbb{R}^n$ X $\mathbb{R}^m$ is called an *admissible pair* of 1.23b if $u(\cdot)$ is piecewise continuous, for which the left and the right-hand limit exist and $x(\cdot)$ is continuous and piecewise continuously differentiable, which satisfies for all points t where $u(\cdot)$ is continuous" with $x(t)$ as state trajectories, 1.23c as initial conditions, $u(\cdot)$ as admissible control and 1.23d-1.23e as terminal conditions. "For

conventional reasons we decide that at a point of discontinuity t of the control $u(\cdot)$:

$$u(t) := u(t^{-}), \tag{1.26}$$

and the control is defined to be continuous at time zero and the terminal time T" (Grass et al., 2008, p. 106).

Moreover, the solution and the control need to be optimal. According to Grass et al. (2008, p. 108), "Let $(x^*(\cdot), u^*(\cdot))$ be an admissible pair for OC 1.23. If

$$V(u^*(\cdot)) \geq V(u(\cdot)), \quad \text{for all admissible controls} \quad u(\cdot); \tag{1.27}$$

then $(x^*(\cdot), u^*(\cdot))$ is called an optimal solution (path) for OC 1.23, $u^*(\cdot)$ is called an optimal control and $x^*(\cdot)$ is called an optimal state trajectory. A part of the optimal solution $(x^*(\cdot), u^*(\cdot))$ is called an (optimal) arc. The value function for OC 1.23 with respect to the initial state x_0 is denoted by"

$$V^*(x_0) := V(u^*(\cdot)). \tag{1.28}$$

In general settings, in the optimal solution of an OC problem 1.23: the control is assumed to be measurable and the state function is absolutely continuous (Grass et al., 2008). One last useful clarification, a minimization problem can always be transformed into a maximization problem by multiplying by -1 (Grass et al., 2008). For example:

$$x_0 = argmin\{g(x)\} \quad \text{iff} \quad x_0 = argmax\{-g(x)\}. \tag{1.29}$$

In general, the optimal control problem aims to maximize the objective function $V(\cdot)$ in 1.25 over the admissible controls $u(\cdot)$, while taking into consideration that the system is influenced by the system dynamics 1.23b, the initial condition 1.23c and the terminal conditions 1.23d - 1.23f (Grass et al., 2008). Furthermore, the state variable x is not directly affected by the decision-maker choice, rather it summarizes the result of previous decisions, since it already contains all the information to make an optimal decision (Grass et al., 2008). The only way to influence the evolution of the system is by changing the control variable $u(t)$. The optimal solutions to an optimal control problem could be zero, one, or multiple. In the next section, we discuss the necessary conditions to find optimal solutions for the OC problem provided by Pontryagin's Maximum Principle.

1.2.2 Necessary conditions: Pontryagin's Maximum Principle

Pontryagin's Maximum Principle provides the necessary conditions that optimal solutions for an optimal control(OC) problem need to satisfy (Grass et al., 2008). In order to solve an OC problem, the optimal solution needs to exist and satisfy the Maximum Principle conditions. The resulting system of ordinary differential equations of dimension $2n$ will define the necessary conditions for optimal solutions. The Pontryagin's Maximum Principle is thus defined. "Let $(x^*(\cdot), u^*(\cdot))$ be an optimal solution of OC 1.23 with free terminal state. Then there exists a continuous and piecewise continuously differentiable function $\lambda(\cdot)$, with $\lambda(t) \in \mathbb{R}^n$ satisfying for all $t \in [0, T]$:

$$H\big(x^*(t), u^*(t), \lambda(t), t\big) = \max_{u \in \Omega(x^*(t), t)} H\big(x^*(t), u, \lambda(t), t\big). \tag{1.30}$$

and at every point t where $u(\cdot)$ is continuous

$$\dot{\lambda}(t) = r\lambda(t) - H_x(x^*(t), u^*(t), \lambda(t), t). \tag{1.31}$$

Furthermore the transversality condition

$$\lambda(T) = S_x(x^*(T), T) \tag{1.32}$$

holds, where

$$H(x, u, \lambda, t) := g(x, u, t) + \lambda f(x, u, t) \tag{1.33}$$

is called the Hamiltonian of the problem OC 1.23 (Grass et al., 2008). In this theorem, the variable λ is the costate or adjoint variable, which is a vector in $\mathbb{R}^n$, while 1.31 is the costate equation, one of the necessary conditions of the Maximum Principle. Other main components of the Maximum Principle are the canonical system and its extremal solution, together with the specification of the maximum principle. "Let

$$\hat{u}(t) \in argmax_{u \in \Omega(x(t), t)} H(x(t), u, \lambda(t), t); \tag{1.34}$$

then the ODE

$$\dot{x}(t) = H_\lambda(x(t), \hat{u}(t), \lambda(t), \lambda_0, t) = f(x(t), \hat{u}(t), t) \tag{1.35}$$

$$\dot{\lambda}(t) = r\lambda(t) - H_x(x(t), \hat{u}(t), \lambda(t), \lambda_0, t), \tag{1.36}$$

is called the canonical system of OC 1.23" (Grass et al., 2008, p. 109). Let us consider $(\hat{x}(\cdot), \hat{\lambda}(\cdot))$ as a solution of the canonical system 1.35-1.36, which satisfies the initial condition 1.23c, the transversality condition 1.32, and the terminal constraints 1.23d-1.23f; then $\hat{x}(\cdot)$ is called an extremal state or trajectory and together with u, thus $(\hat{x}(\cdot), \hat{u}(\cdot))$ is called an extremal solution of the OC problem 1.23 (Grass et al., 2008). In general, the canonical system, the initial condition and the transversality condition define a $2n$ boundary value problem. In particular, the initial and the transversality conditions determine the boundary conditions. To sum up, the necessary conditions that an optimal solution must satisfy are as follows (Bischi et al., 2015):

$$\begin{cases} \dot{x} = f(x^*, u^*, t) = H_\lambda \quad \text{(state equation)} \\ \dot{\lambda} = r\lambda(t) - [g_x(x^*, u, t) + \lambda f_x(x^*, u^*, t)] = r\lambda(t) - H_x \quad \text{(costate equation)} \\ u^* = argmaxH(x^*, u, \lambda^*, t) \quad \text{(maximum principle)}, u \in \Omega \\ x^*(0) = x_0 \quad \text{(initial condition)} \\ (a)\lambda^*(T) = S_x(x^*(T), T) \quad \text{(transversality condition when x(T) is free)} \\ (b)\lambda^*(T) \geq S_x(x^*(T), T) \quad \text{(transversality condition when } x(T) \geq x_T) \end{cases} \tag{1.37}$$

As can be seen, the transversality condition may vary according to the terminal conditions in $x(T)$. In fact, if $x(T)$ is free, then the transversality condition is $\lambda^*(T) = S_x(x^*(T), T)$; if $x(T) \geq x_T$, then the transversality condition becomes $\lambda^*(T) \geq S_x(x^*(T), T)$; however, if $x(T) = x_T$, then there is no transversality condition imposed. Furthermore, it could happen that there is no scrap value, hence $S(x_T, T) = 0$. In this case, the transversality conditions will change in $\lambda^*(T) = 0$ when $x(T)$ is free and, in $\lambda^*(T) \geq 0$ when $x(T) \geq x_T$ (Bischi et al., 2015). An important remark that Bischi et al. (2015) highlights regards the admissible controls. In particular, the control $u(t)$ is not assumed to be continuous and this could affect the optimal control. In the sense that it could be discontinuous, therefore, the control has jumps called switching points. Nevertheless, both the maximum principle continues to hold for all the points where the control $u(t)$ is continuous, and the costate equation holds when the control is continuous. Another remark of Bischi et al. (2015) concerns the terminal time T, which could not be specified in some cases, and it is a variable of the problem. In those time-optimal control problems, the goal is to find the smallest possible time to reach a given point starting from the initial condition. One last remark is related to possible constraints that can be included in the problem. The most usual ones are the mixed inequality constraints, where in these cases there

are inequalities like $q(x(t), u(t), t) \geq 0$ for each $t \in [0, T]$, and the "pure-state" constraints of the form $s(x(t), t) \geq 0$ (Bischi et al., 2015). In these cases, the maximum principle is accordingly reformulated.

An important function, that will be useful in the third chapter, is the Maximized Hamiltonian. That is determined as "Let $H(x, u, \lambda, t)$ be the Hamiltonian 1.33 then the function

$$H^*(x, \lambda, t) = \max_{u \in \Omega(x^*(t), t)} H\big(x, u, \lambda, t\big). \tag{1.38}$$

is called the maximized Hamiltonian" (Grass et al., 2008, p. 110). This definition is important because, in autonomous problems, the maximized Hamiltonian is constant along the optimal trajectory. Since the Ramsey problem is autonomous, it is important to show when an optimal control problem is autonomous. Such as described for differential equations, the optimal control problem is classified as autonomous when its functions $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ and the control region $\Omega(x)$ do not explicitly depend on time (Grass et al., 2008).

In order to solve the optimal control problem, there are different operations to perform. First, one must ensure the existence of an optimal solution and then identify the trajectory that maximizes the objective functional among the candidates satisfying the necessary conditions (Grass et al., 2008). To conclude, solving an optimal control problem consists of:

1. Modeling the optimal control problem
2. Maximization of the Hamiltonian
3. Deriving the canonical system
4. Solving the resulting boundary value problem.

1.2.3 Sufficient conditions

Pontryagin's Maximum Principle provides the necessary conditions that an optimal solution has to satisfy. These conditions allow us to identify possible candidates for the optimal solution (Bischi et al., 2015). Moreover, to ensure that the candidate is an optimal, the sufficient conditions must be met. In particular, these conditions work under specific concavity or convexity of the function. For the purpose of this thesis, two sufficient conditions will be presented: the Mangasarian sufficient condition and the Arrow's sufficient condition. The Mangasarian sufficient condition says: "Consider a candidate solution of the optimal control problem 1.23, for

example an admissible control u^*, the corresponding admissible path x^* and the costate variable λ^*, obtained through Pontryagin's Maximum Principle.

- If the Hamiltonian H in 1.23 is concave in x and u for all $t \in [0, T]$ then u^* is an optimal control and x^* is an optimal path
- If the Hamiltonian H in 1.23 is strictly concave in x and u for all $t \in [0, T]$ then u^* is the unique optimal control and x^* is the unique optimal path" (Bischi et al., 2015, p. 199).

The corollary states that if the instant payoff $g(x, u, t)$ is concave in x and u for all $t \in [0, T]$, if one of the following conditions holds:

- for all $t \in [0, T]$, $f(x, u, t)$ is concave in x and u and $\lambda^* \geq 0$;
- for all $t \in [0, T]$, $f(x, u, t)$ is concave in x and u and $\lambda^* \leq 0$;
- $f(x, u, t)$ is linear in x and u

then, u^* is an optimal control and x^* is an optimal path (Bischi et al., 2015). The maximized Hamiltonian can be specifically written for a concave system and can be used as a sufficient condition (Bischi et al., 2015):

$$H_M(x, \lambda, t) = \max_u H(x, u, \lambda, t) = \max_u [g(x, u, t) + \lambda f(x, u, t)] \tag{1.39}$$

To conclude, Arrow's sufficient condition states that : "Consider a candidate solution of the optimal control problem 1.23, for example an admissible control u^*, the corresponding admissible path x^* and the costate variable λ^*, obtained through Pontryagin's Maximum Principle. If the maximized Hamiltonian H_M in 1.39 is concave in x for all $t \in [0, T]$, then u^* is an optimal control and x^* is an optimal path" (Bischi et al., 2015, p. 200).

1.2.4 Hamilton–Jacobi–Bellman Equation

Pontryagin's Maximum Principle provides a way to solve the OC problem, however, is not the only approach. In fact, another way to solve an optimal control problem is dynamic programming. This approach relies on the principle of optimality, which allows to group optimal control problems in a larger class of dynamic decision problems (Grass et al., 2008). The class of problems (OCF) that we are considering does not take into account the discount

rate r and is formulated as follows:

$$\max_{u(\cdot)} \quad \left\{ \int_t^T g\big(x(s), u(s), s\big)\, ds \;+\; S\big(x(T), T\big) \right\} \tag{1.40a}$$

$$\text{s.t.} \quad \dot{x}(s) = f\big(x(s), u(s), s\big), \qquad s \in [t, T] \tag{1.40b}$$

$$x(t) = \xi. \tag{1.40c}$$

For any (ξ, t), we assumed that the optimal control problem presents an optimal solution. The value function 1.40a, also called the Bellman function $W(\xi, t)$, is formulated as

$$W(\xi, t) = \max_{u(\cdot)} \left\{ \int_t^T g(x(s), u(s), s) ds + S(x(T), T) \right\} \tag{1.41}$$

The Hamilton-Jacobi-Bellman equation is based on the principle of optimality, formulated by Bellman (1957) as follows: "An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision". In mathematical terms, the principle of optimality states: "Assume that for each (ξ, t) with $t \in [0, T]$, $\xi \in \mathbb{R}^n$ there exists an admissible solution $(x^*(\cdot), u^*(\cdot))$ of the problem 1.43a. Then $(x^*(\cdot), u^*(\cdot))$ is an optimal solution for the problem of class OCF 1.43a with $x(t) = \xi$ iff

$$W(\xi, t) = \int_t^s g\big(x^*(\tau), u^*(\tau), \tau\big)\, d\tau + W\big(x^*(s), s\big) \tag{1.42a}$$

$$W\big(x^*(T), T\big) = S\big(x^*(T), T\big) \tag{1.42b}$$

This relation 1.42 allows us to derive information on the relative change of $W(\xi, t)$ with respect to ξ when s tends to t" (Grass et al., 2008, p. 128). This results in the HJB equation: "Let the Bellman function $W(\xi, t)$ be continuously differentiable with respect to ξ and t. Furthermore, for every initial point ξ and time t there exists an admissible control $u^*(\cdot)$ with a corresponding state trajectory $x^*(\cdot)$. Then $(x^*(\cdot), u^*(\cdot))$ is an optimal solution of OCF 1.42 iff the Bellman function $W(\xi, t)$ satisfies the Hamilton-Jacobi-Bellman equation (HJB):

$$-W_t(\xi, t) = \max_u \{ g(\xi, u, t) + W_x(\xi, t) f(\xi, u, t) \} \tag{1.43a}$$

$$\text{and} \quad W(x^*(T), T) = S(x^*(T), T). \tag{1.43b}$$

for all $(\xi, t) \in \mathbb{R}^n \times [0, T]$ for which $u^*(\cdot)$ is continuous" (Grass et al., 2008, p. 128).

While Pontryagin's Maximum Principle transforms the optimal control problem into a boundary value problem, the HJB equation transforms it into a nonlinear partial differential equation. That is why dynamic programming is often difficult to solve explicitly. For this reason, in the present thesis we adopt Pontryagin's approach.

1.2.5 Boundary Value Problem VS Initial Value Problem

In this thesis, we already introduced the term *"Boundary Value Problem"*, but now it needs a proper formulation, together with the *"Initial Value Problem"* definition. The main difference between Initial Value Problems (IVP) and Boundary Value Problems (BVP) is the conditions to which they are subject. IVP is subject only to an initial condition such as here 1.23c, while BVP is subject to the initial condition, and the transversality condition which together give a boundary in which is contained the set of all possible solutions. The IVP is characterized by an ordinary differential equation with an initial condition specified at a single initial point

$$\dot{x}(t) = f(x(t), t), \quad x \in \mathbb{R}^n, \quad t \in [0, T] \tag{1.44}$$

$$\text{s.t.} \quad x(0) = x_0 \tag{1.45}$$

where $x_0 \in \mathbb{R}^n$ is a given constant vector (Hunter, 2011). In this case, under the assumption of regularity, the existence and uniqueness of the solution are guaranteed (Hunter, 2011). On the other hand, the BVP could be characterized by an ODE or a partial differential equation, and the conditions are imposed in more than one point in the time interval. In fact, the two boundary points can be defined as

$$b(x(0), x(T)) = 0 \tag{1.46}$$

where b is a vector-valued function in $\mathbb{R}^n \to \mathbb{R}^n$ and is called boundary condition (Grass et al., 2008). The ODE in 1.44 together with the condition 1.46 is a two-point boundary value problem. An important structural difference between IVP and BVP is that BVP, depending on the additional time constraint T, could admit no solution, one solution, or many solutions (Grass et al., 2008). This difference makes solving BVP numerically more complicated than solving IVP. As we saw in Pontryagin's Maximum Principle section, after applying the Principle, the canonical system can be defined as a system of differential equations subject to an initial condition and a transversality condition. This transforms the optimal control problem into a two point boundary problem. Since solving BVP is numerically complicated, a common strategy is to transform the BVP into an equivalent IVP. This is the idea of the Shooting method: guess

the initial condition, hence solving the IVP, and then adjust the guess to satisfy the terminal boundary conditions (Ascher et al., 1995).

Chapter 2

The Shooting Method

In this chapter, we develop the theoretical foundation of the shooting method as a numerical technique to solve boundary value problems. We explore the classical single shooting method together with its most relevant variants such as the multiple shooting, the NOC shooting and NOC multiple shooting. We conclude the chapter with a comparison of the different methods.

As already explained in the previous chapter (1), boundary value problems, due to their structure, are quite complex to solve numerically. However, the shooting method is one of the most useful and efficient methods in handling them. In fact, the shooting method is based on reducing the BVP to an initial value problem, which has a simpler structure, and solving it by guessing the initial condition (Edun and Akinlabi, 2021). The IVP is hence solved using an iterative solution and the method continues with the same process, pursuing the goal of finding a satisfactory level for the second boundary condition.

Although the shooting method performs well on relatively basic problems, its effectiveness strongly depends on the stability properties of the IVP created (Edun and Akinlabi, 2021). For this reason, many scholars discovered some variation of the shooting in order to provide a solution to all the different problems that may arise. This is the reason why more than one variant of the method is presented in this thesis, to give a comprehensive overview of it.

2.1 Single Shooting Method

2.1.1 Motivation and idea

In optimal control, methods are often classified as direct or indirect. In the indirect method, the necessary conditions come from Pontryagin's Maximum Principle, by obtaining a canonical system with its initial and transversality conditions which give rise to a two-point boundary value

problem that can be solved by different approaches, such as the shooting method (Grass et al., 2008). In contrast, direct methods use time discretization to turn the optimal control problem into a "finite-dimensional nonlinear program (NLP)", which standard SQP methods can solve (Grass et al., 2008). Usually direct methods are used in large-scales and real-time implementations; meanwhile, for long-term optimal behavior, indirect methods are often preferred (Grass et al., 2008). Since the Ramsey problem is formulated as an infinite horizon optimal control problem, this thesis adopts an indirect approach based on Pontryagin's Maximum Principle and solves the resulting boundary value problem using the shooting method.

The shooting method is a classical numerical technique used to solve boundary value problems (BVPs). The shooting method is an iterative method (Kong et al., 2020). This method is named after an analogy with target shooting. Basically, such as a cannon that shoots a target 2.2, the shooting method tries to guess the boundary condition and check where it land; if it did not hit the target, it will try again to change its direction based on the error (Kong et al., 2020). The method continues to shoot as long as it will get the closest possible to the target. That is why the shooting method is iterative, it keeps shooting and checking until the solution is found.

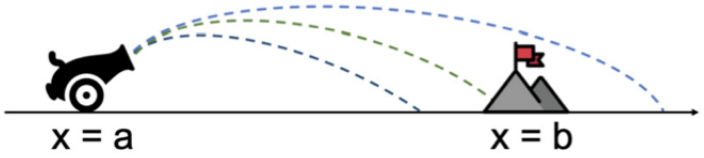

Figure 2.1: Target shooting analogy to the shooting method. (Kong et al., 2020).

From a mathematical perspective, the shooting method aims to transform an ODE boundary value problem into an equivalent initial value problem (Kong et al., 2020). While an IVP is easier to solve with just one or two guesses the problem is solved, a boundary value problem is more complicated. In fact, the methods to solve the IVP, like the Runge-Ketta one, do not work for the BVP since not all the initial value conditions are available to solve the ODE and get a unique solution (Kong et al., 2020). Therefore, the shooting method was invented to beat this difficulty (Kong et al., 2020). By trying to shoot the target until the solution for the BVP is sufficiently satisfactory.

The shooting method is a family composed of variants of the algorithm used to solve different BVPs. Now we concentrate on the most simple one: the single shooting method. This algorithm repeatedly simulates "a mathematical model of the physical system" (Vyasarayani et al., 2012, p. 1) with the same span of the optimal control problem, compares the simulated response with the measured one, and, after each simulation, updates the estimate for the parameter

until it reaches the same response, or get really close to it (Vyasarayani et al., 2012).

2.1.2 Process steps and numerical algorithms

In order to solve the boundary value problem using the shooting approach, there is a procedure to follow. First, let us recall a general second-order ODE given $f(a) = f_a$ and $f(b) = f_b$ (Grass et al., 2008)

$$f''(x) = F(x, f(x), f'(x)) \tag{2.1}$$

The steps to follow are:

1. Guess $f'(a) = \alpha$ and $f(a) = f_a$ to turn the ODE in 2.1 into an initial value problem transforming the two conditions into one value $x = a$.

2. Solve IVP with some numerical algorithms like Runge-Kutta or Euler method and then integrate to the boundary b to find $f(b) = f_\beta$ by solving a root-finding problem, with Newton or Secant method.

3. The last step is to compare the results to achieve $f_\beta - f_b = 0$. Usually, the first guess is never accurate; therefore, the initial guess is adjusted and the process is repeated until the error is acceptable (Grass et al., 2008).

The first step is the transformation from a boundary value problem into an initial value problem. In the transformation form BVP to IVP there could be some stability problems due to the sensitivity of the system to initial conditions. In general, the transformation of the BVP is based on the idea of inserting one condition for the final boundary condition, in fact the second order BVP:(Edun and Akinlabi, 2021)

$$\begin{cases} y'' = f(x, y(x), y'(x)) \\ y(a) = \alpha, \quad y(b) = \beta \quad \text{on the interval } [a, b] \end{cases} \tag{2.2}$$

is transformed into a pair of initial value problems, with the condition $y'(a) = s$ which represents the slope:

$$\begin{cases} y'' = f(x, y(x), y'(x)) \\ y(a) = \alpha, \quad y'(a) = s \end{cases} \tag{2.3}$$

Where the boundary conditions are specified at the origin and are homogeneous at the initial point (Klamkin, 1970). The key in transforming the BVP into an IVP is to change the final

boundary condition into a condition of the slope of the initial condition, and the final condition in the BVP becomes the target to hit with the shooting method hence $y(b) = \beta$.

The second step is the real shooting part. It is divided into two step: the first is solving IVP and the second is solving the root-finding problem. To solve the IVP there are many methods, but the most used ones are the Euler method and the Runge-Kutta method. The explicit Euler formula is one of the most intuitive methods to solve the initial value problem. The idea is that in any state $(t_j, S(t_j))$ the formula uses F at the state to "point" at the next state and shift in that direction for a distance of h (Kong et al., 2020). According to Kong et al. (2020), assuming a function $F(t, S(t))$ that computes $\frac{dS(t)}{dt}$, a numerical grid t of the interval $[t_0, t_f]$, and an initial state value $S_0 = S(t_0)$, the steps to solve the initial value problem and compute $S(t_j)$ for every t_j are the following:

1. Store $S_0 = S(t_0)$ in an array, S
2. Compute $S(t_1) = S_0 + hF(t_0, S_0)$
3. Store $S_1 = S(t_1)$ in S
4. Compute $S(t_2) = S_1 + hF(t_1, S_1)$
5. Store $S_2 = S(t_2)$ in S
6. ....
7. Compute $S(t_f) = S_{f-1} + hF(t_f, S_f)$
8. Store $S_f = S(t_f)$ in S

where S is an approximation of the solution of the initial value problem. This method integrates the solution of the ODE.

The Runge-Kutta methods are one of the most used to solve ODEs. The idea is to get a more accurate solution than the Euler's method (Kong et al., 2020). While in Euler's function it will be needed to insert more terms and get more complicated calculations, in Runge-Kutta methods it is easier. The RK method is based on truncate Taylor series without requiring the computation of higher derivatives (Kong et al., 2020). The two more used methods are the second order Runge Kutta method and the fourth order RK method. The second order RK will define

$$k_1 = F(t_j, S(t_j)) \tag{2.4}$$

$$k_2 = F(t_j + ph, S(t_j) + qhk_1) \tag{2.5}$$

where we shall have:

$$S(t_{j+1}) = S(t_j) + \frac{1}{2}(k_1 + k_2)h \tag{2.6}$$

For the fourth-order Runge Kutta method the output is the same but for four points k_1, k_2, k_3 and k_4 where the weighted average is the approximation of the solution (Kong et al., 2020). In the numerical implementation of the Ramsey problem in chapter 3, the IVP will be solved in python using the function "solve_ivp". In any case, the output obtained from this step, in the notation introduced in 2.3, is the solution of the IVP that is $y(b; s)$ and satisfies $y'(a) = s$, together with the initial boundary condition $y(a) = \alpha$ (Filipov et al., 2017). Usually, this solution does not satisfy the second boundary condition $y(b) = \beta$, which leads to the second part of the shooting procedure: the root-finding part.

In order to find the solution to the boundary condition, we need to solve the root finding problem, that is, finding the zero. Finding the zero actually means to find the solution with the smallest possible error. Here, the error is the deviation from the expected value (Kong et al., 2020). The difference

$$\phi(s) = y(b; s) - \beta \tag{2.7}$$

is the "deviation from the second boundary condition" (Filipov et al., 2017, p. 11). The goal is to find s such that $\Phi(s) = 0$. Therefore, the IVP solution $y(b; s)$ is the solution of the BVP (Filipov et al., 2017). For this reason, the shooting method is a root finding procedure to find the root s of $\phi(s) = 0$ (Filipov et al., 2017). To approximate the roots, there are different numerical methods, the most famous ones are the Newton method, the Secant method, and the Bisection method. In general, in these methods the next root approximation s_{k+1} is found like this:

$$s_{k+1} = s_k - \frac{\phi(s_k)}{k} \tag{2.8}$$

where s_k is the current approximation (Filipov et al., 2017). The difference between the methods is how its calculated the value of k. In the Newton method, the value k is the slope of the tangent to the line ϕ, hence $k = \phi'(s_k)$. The common formula used for the Newton method is:

$$s_{k+1} = s_k - \frac{\phi(s_k)}{\phi'(s_k)} \tag{2.9}$$

For the Secant method, the value k is the slope of the secant to the line ϕ. In these methods, since they are based on the current approximation, the value k is readjusted at each iteration

(Filipov et al., 2017).

Moreover, the bisection method is based on the iterative use of the intermediate value theorem to find roots (Kong et al., 2020). Basically, the theorem is used to bisect the interval containing the root. For example, given the interval $[a, b]$, let $m = \frac{b+a}{2}$ be the middle point of $[a, b]$, with conditions $f(a) > 0$ and $f(b) < 0$, for the intermediate value theorem, which ensures that the root is in this interval (Kong et al., 2020). If m is the root, then the method is finished; if not, the method continues to divide the interval and iterate the process until the solution is found and the error is acceptable (Kong et al., 2020). For this step, in Python the function "fsolve/root_scalar/root" is usually used.

The third step is to check the solution to see if it satisfies $f_\beta - f_b = 0$ and if the residual $|\phi(s)|$ is sufficiently small. If it does not satisfy the boundary condition, simply repeat from step one with an updated guess.

2.1.3 Mathematical formulation

After explaining the process of the shooting method, we now present its general mathematical formulation. Let us consider again a second order BVP (Edun and Akinlabi, 2021):

$$\begin{cases} y'' = f(x, y(x), y'(x) \\ y(a) = \alpha, \quad y(b) = \beta \end{cases} \tag{2.10}$$

in the interval $[a, b]$. The BVP becomes the corresponding IVP:

$$\begin{cases} y'' = f(x, y(x), y'(x) \\ y(a) = \alpha \quad y'(a) = s \end{cases} \tag{2.11}$$

To apply the shooting method, we transform the BVP into a family of initial value problems (IVPs). We replace the unknown boundary condition at $x = b$ with an unknown initial slope:

$$\begin{cases} y'' = f(x, y(x), y'(x)), \\ y(a) = \alpha, \quad y'(a) = s, \end{cases} \tag{2.12}$$

where $s \in \mathbb{R}$ is an unknown parameter. Let us assume that $\phi(s)$ is the error and that it can be defined as:

$$\phi(s) = y(b; s) - \beta \tag{2.13}$$

The goal is to find s that satisfies $\phi(s) = 0$. In a graph, β is the target value at $x = b$: we adjust the guessed slope until the computed trajectory hits it. This was the first step in the procedure.

Moving to the shooting part, let us make an initial guess for the value $y'(a)$ (Edun and Akinlabi, 2021). Let us assume to be $y'(a) = s_0$ and then solve the IVP using either the Euler or the Runge-Kutta method. The resulting IVP will be:

$$\begin{cases} y'' = f(x, y(x), y'(x) \\ y(a) = a \quad y'(a) = s_0 \end{cases} \tag{2.14}$$

After carrying out the IVP solution $y(b; s)$ and checking if it satisfies the target β, we can plan the next step. If the solution satisfies the second boundary condition $y(b) = \beta$, then $y(b; s)$ is the solution of the BVP. Always checking whether the error $|\phi(s)|$is acceptable. If this is not so, we make another guess, for example s_1 and the resulting solved IVP is:

$$\begin{cases} y'' = f(x, y(x), y'(x) \\ y(a) = a \quad y'(a) = s_1 \end{cases} \tag{2.15}$$

The process is repeated until the target β is hit. As we explained in the last section, the value of the estimates are linearly correlated to the previous guesses (Edun and Akinlabi, 2021). In order to achieve a better value for s_2, we can use the estimation interpolation formula, hence the secant method:

$$s_{n+1} = s_n - \frac{\phi(s_n)(s_n - s_{n-1})}{\phi(s_n) - \phi(s_{n-1})} \tag{2.16}$$

with $n \geq 1$. In making this assumption, we always check the error $y(b) - \beta$ and when it is acceptable, means that the solution is found.

Let us now consider a practical example from Edun and Akinlabi (2021) to better understand the method. Consider the boundary value problem:

$$\begin{cases} y'' = 2y^3 - 6y - 2x^3, \quad 1 \leq x \leq 2 \\ y(1) = 2, \quad y(2) = 2,5 \end{cases} \tag{2.17}$$

and let $y'(1) = \alpha$ Considering these two initial guesses: $\alpha_0 = 4$ and $\alpha_2 = 1,5$. The two IVPs will be:

IVP1:

$$\begin{cases} y'' = 2y^3 - 6y - 2x^3, \quad 1 \leq x \leq 2 \\ y(1) = 2, \quad y'(1) = 4 \end{cases} \tag{2.18}$$

IVP2:

$$\begin{cases} y'' = 2y^3 - 6y - 2x^3, \quad 1 \leq x \leq 2 \\ y(1) = 2, \quad y'(1) = 1,5 \end{cases} \tag{2.19}$$

Assume $y' = z$ and $z' = 2y^3 - 6y - 2x^3$. With the Euler method and with $h = \frac{b-a}{4} = \frac{2-1}{4} = 0,25$ and using $y_{i+1} = y_i - hz_i$ and $z_{i+1} = z_i - h(2y_i^3 - 6y_i - 2x_i^3)$, we reached $x_4 = 2$, $y_4 = 6,42505$ but our target was $\beta = 2,5$. Assuming this results, we use the secant method to find a more accurate value for the slope $y'(0)$:

$$\alpha_k = \alpha_{k-1} - \frac{(y(b, \alpha_{k-1}) - \beta)(\alpha_{k-1} - \alpha_{k-2})}{(y(b, \alpha_{k-1}) - (y(b, \alpha_{k-2})} \tag{2.20}$$

The results of the guess α_2 will not hit the target. The process continues until the correct guess is found: $\alpha_4 = 0,06$. To see all procedures, please refer to Edun and Akinlabi (2021). Here, the table resulting from the correct initial guess α_4 and shows the Euler method until the boundary condition is satisfied $y(2) = 2,5$.

i	x_i	Euler (y_i) $(\alpha_4 = 0.06)$	Exact	$\lvert\varepsilon\rvert$	z_i
0	1	2.00000	2.00000	0.00000	0.06000
1	1.25	2.01500	2.05000	0.03500	0.56000
2	1.5	2.15500	2.16667	0.01167	0.65161
3	1.75	2.31790	2.32142	0.00352	0.73554
4	2	2.50179	2.50000	0.00179	

Figure 2.2: Comparison of the exact solution and the shooting method using Euler. (Edun and Akinlabi, 2021).

To conclude this section, we present one last example. This example helps to better understand from a graphical point of view how the shooting method works. We take the problem from Kong et al. (2020): we need to launch a rocket, where the altitude is $y(t)$ and the gravity is $g = 9,8m/s^2$. If after 5 seconds we want the rocket to reach $50m$ off the ground, what should be the velocity of launching? To solve the problem, we first describe it as a BVP for a second-order ODE:

$$\begin{cases} \frac{d^2y}{dt^2} = -g \\ y(0) = 0, \quad y(5) = 50 \end{cases} \tag{2.21}$$

In order to find $y'(0)$ at the launch, "we will reduce the order of the function" (Kong et al., 2020, chapter 23.2) and introduce the velocity $v(t) = y'(t)$, which becomes:

$$\begin{cases} \frac{dy}{dt} = v \\ \frac{dv}{dt} = -g \end{cases} \tag{2.22}$$

Following the Python implementation provided by Kong et al. (2020), I recreated the two initial guesses and the final solution, computed with the "solve_ivp" formula and a root-finding step. In figure 2.3, the first graph (a) represents the shooting method with an initial guess of $v = 25m/s$ but we can see that it undershoots the target. In the second graph (b), the velocity is $v = 40m/s$, but this time it overshoots the target. Finally, the root finding procedure converges to the velocity $v = 34,4999m/s$, which in the third graph (c) shows that the slope hits the target at $t = 5s$.

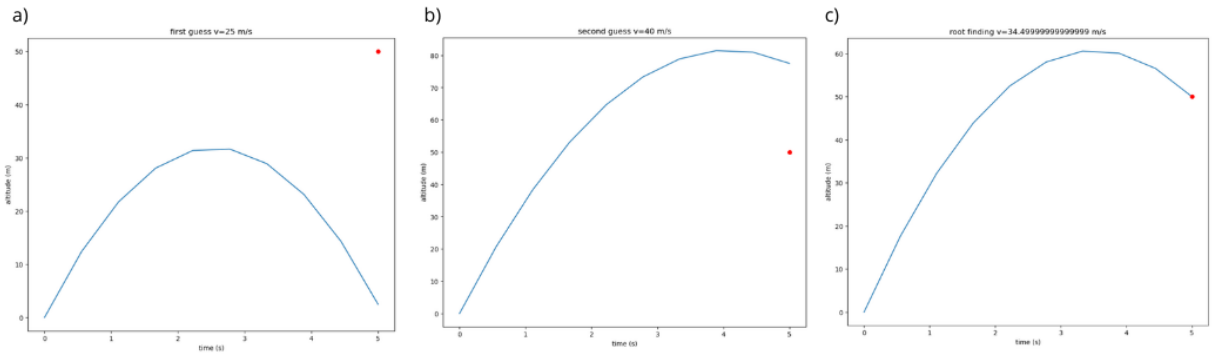

Figure 2.3: Graphical illustration of single shooting with successive guesses of the initial slope. Personal elaboration based on Kong et al. (2020).

The same logic will be applied in Chapter 3 to the canonical system of the Ramsey model, where the unknown initial condition corresponds to the costate variable.

2.2 Multiple Shooting Method

2.2.1 Motivation and idea

According to Morrison et al. (1962) the multiple shooting method can be viewed as a compromise between the single shooting method and approaches based on implicit difference equations. This method derives from the single shooting method and solves more specific problems that are too complicated for the simple method. The idea is to divide the time interval $[0, T]$ into k equal subintervals (Morrison et al., 1962). Then, each subinterval is solved with a single-shooting method procedure (Vyasarayani et al., 2012). Moreover, constraints enforce continuity between

consecutive segments, ensuring that the the final state of one subinterval is sufficiently close to the initial state of the next one (Vyasarayani et al., 2012). Therefore, by dividing the long integration into many smaller ones, multiple-shooting typically improves numerical stability and reduces sensitivity to initial guess, thus reducing the risk of divergence when identifying parameters (Vyasarayani et al., 2012).

The single shooting method has some difficulties that the multiple shooting method tries to solve, which are (Osborne, 1969):

1. the instability of the initial value problem for the system of differential equations, especially with long horizons;

2. the difficulty in making accurate initial guesses in nonlinear problems.

In general, the multiple shooting approach is more robust than single shooting method, especially when the time interval is long. By dividing the interval, the convergence to the solution becomes more precise. However, continuity may suffer, since the condition needs to meet between the following subsets (Morrison et al., 1962). In figure 2.4, it is shown how multiple shooting works from a graphical point of view. As we can see, the time axis is divided into equal segments and for each segment is represented the initial guess, the forward integration, the defect between the initial guess and the forward integration, and the true solution.

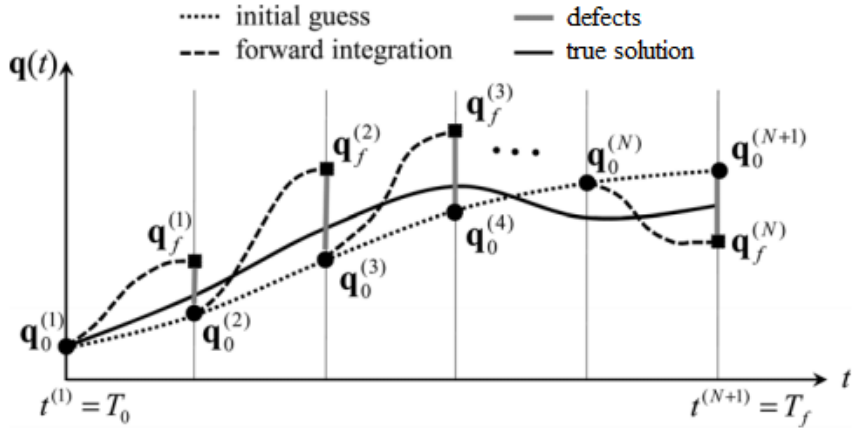

Figure 2.4: Multiple shooting method. Brüls et al. (2014).

In the third Chapter, we apply multiple shooting to solve the Ramsey problem and compare the results with those obtained via single shooting. Since the Ramsey problem is nonlinear and defined over an infinite horizon, multiple shooting is expected to provide improved stability and accuracy, especially in the numerical solution of the associated boundary value problem.

2.2.2 Process steps

The multiple shooting method has a process similar to the single shooting approach, but introduces additional structure in order to improve numerical stability and convergence. The main difference is the subdivision of the time interval and in the imposition of continuity conditions between subintervals. In particular, the steps to solve the BVP with the multiple shooting method are (Fraser-Andrews and Oberle, 1999):

1. Dividing the time interval $[0, T]$ in k subintervals
2. Introducing additional unknown initial values at the beginning of each segment
3. Solving the initial value problem for each subinterval
4. Imposing continuity properties at the ends of each subinterval with the beginning of the following one
5. Solve the nonlinear problem for the initial values, usually using the Newton method.

The steps (3) and (5) are theoretically similar to the single shooting method. The main difference is that while in single shooting the IVP is integrated starting from a single unknown initial vector, in multiple shooting several intermediate initial vectors are introduced (step 2). The reason for the division of the interval is to avoid instability in the differential equation that could otherwise "blow up" before the IVP can be integrated (Morrison et al., 1962). For the theory about steps (3) and (6) please refer to the last section and keep in mind the dimensionality of the unknown variables.

Let us start with the first step: the division of the interval in k segments. The main idea is to separate the problem into smaller intervals in order to inhibit the effect of the initial guess on the exponential growth of the system (Clark and Petzold, 1989). This limits the sensitivity to the initial guess and the propagation of errors throughout the interval. These issues arise in the transformation of the ODE from BVP to IVP, and gives instability to the IVP (Carraro et al., 2014).

The second step after dividing the interval $[0, T]$ into k subintervals is to define different dependent variables for each created segment (Morrison et al., 1962). Therefore, after segmentation, there are $n \times k$ dependent variables in the interval of integration $[0, T/k]$. An important remark that Morrison et al. (1962) highlights is that changing the sign of $\dot{y}$ reverse the direction of integration in that particular interval. This means that when the integration is unstable in only one direction of the problem and the sign $\dot{y}$ is changed in half the subinterval, then that

part will be in a stable direction (Morrison et al., 1962). For this reason, the intervals can be divided into smaller segments so that the integration does not let the instability influence the problem. Moreover, it reduces the exponential factors that influence the algorithm and stabilize it (Carraro et al., 2014).

Step (3) involves guessing the initial value for all the established subintervals. The procedure is the same as for the single shooting method, just applied to smaller intervals.

As said before the multiple shooting method is a mix between the single shooting method and finite difference approach. From the finite difference approach, multiple shooting takes the properties that the solution is kept stable at the splitting points (Morrison et al., 1962). These are the matching convergence conditions of step (4) in the process. The important thing is that the portion of the same original variable matches in the interior splitting nodes (Morrison et al., 1962). In fact, if y_i is defined as the variable in the interval $[(p-1)T/k; pT/k]$ and y_j is the same variable in the interval $[pT/k; (p+1)T/k]$ and if in the later interval the integration direction is reversed, then $y_i(0)$ and $y_j(0)$ are guessed and it is introduced the following condition $y_i(T/k) = y_j(T/k)$ (Morrison et al., 1962). Moreover, if y_m defines the variable in the subsequent interval $[(p+1)T/k; (p+2)T/k]$, then another condition is introduced: $y_j(0) = y_m(0)$ and so on.

After the matching conditions of subsequent intervals are defined, we can move to the last step. The application of a Newton-type method to integrate the solution. The Netwon's method is applied to all the integrated subintervals, applying all the initial conditions and the convergence conditions to enforce continuity in the internal nodes (Carraro et al., 2014).

2.2.3 Mathematical formulation

The formulation of the multiple shooting method will follow the theoretical framework of Osborne (1969), while using a more modern notation provided by Bock and Plitt (1984). Let us consider the system of n differential equations by Osborne (1969):

$$\frac{dx}{dt} = f(x,t) \tag{2.23}$$

subject to the boundary conditions

$$B_a x(a) + B_b x(b) = \beta \tag{2.24a}$$

(2.24b)

where $B_a, B_b \in \mathbb{R}^{r\times n}$ and $\beta \in \mathbb{R}^r$. The solution of this nonlinear problem can be written as $\mathbf{x}(t, t_{i-1}, s_j)$, or more simply $\mathbf{x}_i(t)$ (Osborne, 1969).

After the description of the problem, let us divide the time interval into equal subintervals. Taking into account the interval $[a, b]$, according to Bock and Plitt (1984) it is divided into the different mesh points as follows:

$$a = t_0 < t_1 < \cdots < t_m = b \tag{2.25}$$

Then, let us introduce the unknown state variables for all the subintervals and the relative IVP

$$\begin{cases} x_j'(t) = f(t, x_j(t)), \quad t \in [t_j, t_{j+1}], \\ x_j(t_j) = s_j. \end{cases} \tag{2.26}$$

as function of s_j. For each segment we define the approximation of the solution

$$\Phi_j(s_j) := x_j(t_{j+1};\, t_j, s_j), \qquad j = 0, \ldots, m-1. \tag{2.27}$$

The next step is to introduce the matching conditions that impose the continuity of the system trajectory (Bock and Plitt, 1984)

$$s_{j+1} - \Phi_j(s_j) = 0, \qquad j = 0, \ldots, m-1. \tag{2.28}$$

together with the boundary conditions for the extremes of the subintervals

$$B_a\, s_0 + B_b\, s_m = \beta. \tag{2.29}$$

Therefore, the multiple shooting method transforms a BVP into a nonlinear system $F(S) = 0$ where the continuity conditions are given by

$$F(S) = \begin{bmatrix} B_a s_0 + B_b s_m - \beta \\ s_1 - \Phi_0(s_0) \\ s_2 - \Phi_1(s_1) \\ \vdots \\ s_m - \Phi_{m-1}(s_{m-1}) \end{bmatrix}. \tag{2.30}$$

The last step is to solve $F(S) = 0$ with Newton's method (Osborne, 1969).

$$J(S^{(k)})\,\Delta S^{(k)} = -F(S^{(k)}), \qquad S^{(k+1)} = S^{(k)} + \Delta S^{(k)}. \tag{2.31}$$

With this formulation, the multiple shooting approach proves that by adding the extra variables s_j the convergence and numerical stability are improved in nonlinear problems (Bock and Plitt, 1984). Improving numerical stability prevents the growth of errors throughout the system that could be caused by incorrect initial guesses and could cause instability in the ODE system (Bock and Plitt, 1984).

2.3 NOC Shooting Methods

According to Fraser-Andrews and Oberle (1999), the NOC Shooting method is a technique to solve optimal control problems. The term NOC refers to Numerical Optimisation Center, which is the English university where they develop this specific technique (University of Hertfordshire Hatfield, UK) (Fraser-Andrews and Oberle, 1999). Since this thesis is going to apply the shooting methods to an optimal control problem, the Ramsey problem, we wanted to give a complete overview of all the possible shooting methods applicable. In particular, the NOC shooting is a specific shooting method for optimal control. In general, after the Pontryagin's Maximum Principle is applied to the optimal control problem and the problem is reduced to a boundary value problem, the NOC shooting methods are applied (Fraser-Andrews and Oberle, 1999). Such as for the classical formulation where the multiple shooting method is a variant of the single shooting method, also for the NOC family, there exist both the NOC shooting method and the multiple NOC shooting method. In this section, we are going to briefly analyze both of them (Fraser-Andrews and Oberle, 1999). In particular, we will analyze their both theoretical and mathematical definitions, together with their principal features.

2.3.1 NOC Shooting

The NOC shooting method (Fraser-Andrews and Oberle, 1999) is an indirect technique for solving optimal control problems. After deriving the necessary conditions via Pontryagin's Minimum Principle, the optimal control problem is reduced to a multipoint boundary value problem. The key idea of NOC shooting, as described in Fraser-Andrews and Oberle (1999), is to parameterize the unknown quantities (such as missing initial values, and discontinuity parameters) and to reformulate the problem as a finite-dimensional constrained minimization

problem.

Before defining the NOC shooting, here the optimal control problem and the system of equations with the relative multipoint boundary conditions that we take into consideration are defined. Consider the following objective function I to minimize (Fraser-Andrews and Oberle, 1999):

$$I = \int_a^w L(u,x)dt + \Psi_a(x(a)) + \Psi_w(x(w)), \tag{2.32}$$

where x is the state in dimensions n, u is the continuous control in dimension d, L and Ψ are smooth functions and a, w are unknown (Fraser-Andrews and Oberle, 1999). This objective function is subject to the state equation and the two-point boundary conditions (Fraser-Andrews and Oberle, 1999):

$$\dot{x} = f(u,x) \tag{2.33}$$

$$r_{aw}(x(a), x(w)) = 0 \tag{2.34}$$

After the definition of the optimal control problem, the Hamiltonian, the adjoint equations and the transversality conditions are determined. The Pontryagin's Minimum Principle is applied, and the outcome is a set of multipoint boundary conditions. This is the process that we saw in Chapter 1 and that is developed in Fraser-Andrews and Oberle (1999). Here, to explain the NOC shooting, the multipoint boundary value problem is directly presented. Starting from the following system of equations (Fraser-Andrews and Oberle, 1999):

$$\dot{y} = f(u,y), \quad u = u(y), \quad t \in [a,w], \tag{2.35}$$

with the set of unknown parameters $\tau = \{\tau_k : k = 1,..,s\}$ which are the arc nodes that satisfy the following condition (Fraser-Andrews and Oberle, 1999):

$$a = \tau_1 < \tau_2 < .. < \tau_{s-1} < \tau_s = w \tag{2.36}$$

The form of (f,u) that changes at τ becomes (Fraser-Andrews and Oberle, 1999):

$$\dot{y} = f_k(u,y), \quad u = u_k(y), \quad t \in [\tau_k, \tau_{k+1}], \quad k = 1,..,s-1, \tag{2.37}$$

where the functions are expressed with the explicit form of the arc (f_k, u_k).

Finally, the multipoint boundary value problem can be defined by the conditions for the

inner arc nodes:

$$A_k(y(\tau_k^-), y(\tau_k^+), v_k) = 0 \quad k = 2, .., s-1, \tag{2.38}$$

where the parameters v_k define any discontinuity in p at τ_k and A_k is a smooth function that "express the continuity property of y" (Fraser-Andrews and Oberle, 1999, p. 302). Together with the two point boundary conditions:

$$r(y(a), y(w)) = 0 \tag{2.39}$$

This formulation of the BVP will also be used for the definition of NOC multiple shooting.

Regarding NOC shooting, this technique uses 2.35 - 2.39 to minimize the objective function I in 2.32 (Fraser-Andrews and Oberle, 1999). The unknown parameters are collected in a finite-dimensional vector

$$q = (\bar{\tau}, \bar{y}(a), v),$$

which contains the switching times, the missing initial values, and possible discontinuity parameters. The objective functional I is then regarded as a function of q, i.e., $I = I(q)$ (Fraser-Andrews and Oberle, 1999). The objective is to find $I(q)$ for any q. To evaluate $I(q)$, the augmented system (state, costate, and possibly the integral state I) is integrated along each arc, starting from the initial guess for $\bar{y}(a)$. In addition, it is checked that the multipoint boundary conditions 2.39 are satisfied at each arc node τ_k (Fraser-Andrews and Oberle, 1999). For minimization, the condition of point constraints 2.38 is transformed into the following equality constraints:

$$g_{eq}(\bar{\tau}, \bar{y(a)}) = 0 \tag{2.40}$$

To complete the integration, also the condition 2.39 is defined with the same terms $g_{eq}(\bar{\tau}, \bar{y(a)})$ (Fraser-Andrews and Oberle, 1999).

To summarize, the NOC shooting method, according to Fraser-Andrews and Oberle (1999) involves:

1. estimate $q = (\bar{\tau}, \bar{y(a)}, v)$
2. minimize I with respect to q, subject to $\dot{y} = f_k(u_k, y), k = 1, ..., s-1$ and to conditions at τ.

To conclude this section on NOC shooting, its main features are presented (Fraser-Andrews and Oberle, 1999).

1. The optimal control problem is reduced to a finite-dimensional constrained minimization problem in the parameter vector q.

2. Switching times and other incidental parameters are included directly in the minimization variables.

3. The state and adjoint equations are integrated along each arc, either forward or backward in time.

4. Missing initial values of adjoint variables are estimated as part of the parameter vector.

5. To compute minimization, there are many algorithms available.

2.3.2 Multiple NOC Shooting

Multiple NOC shooting extends the NOC shooting framework by introducing additional shooting nodes along the trajectory (Fraser-Andrews and Oberle, 1999). The method combines the parametrization strategy of NOC shooting with the interval decomposition characteristic of classical multiple shooting (Fraser-Andrews and Oberle, 1999).

For the definition, let us recall the objective function I 2.32 to minimize and the unknown parameters for the arc nodes 2.36. Now, let

$$\theta = \{\theta_j : j = 1, \ldots, m\}$$

be a finite set of points representing the multiple shooting nodes, satisfying

$$a = \theta_1 < \theta_2 < \cdots < \theta_{m-1} < \theta_m = w. \tag{2.41}$$

The nodes $\{\theta_2, \ldots, \theta_{m-1}\}$ are defined in terms of the arc nodes τ (Fraser-Andrews and Oberle, 1999). In general, the multiple shooting nodes and the arc nodes may coincide. However, the multiple shooting intervals are often smaller than the arc intervals, so that none of the arc nodes τ_k lies strictly inside a θ-subinterval (Fraser-Andrews and Oberle, 1999).

At each multiple shooting node θ_j, an additional parameter vector γ_j of dimension n is introduced. This vector represents the initial value of y when the integration is restarted at θ_j (Fraser-Andrews and Oberle, 1999).

With this additional parametrization, the NOC shooting parameter vector becomes

$$q = (\bar{\tau}, \bar{y}(a), v, \gamma). \tag{2.42}$$

The integration is performed as described for NOC shooting in the previous section. However, two different situations may occur:

Case 1: $\theta_j \neq \tau_k$**.**

If a multiple shooting node does not coincide with an arc node, the state variable y is required to be continuous at θ_j, namely

$$y(\theta_j^-) = y(\theta_j^+). \tag{2.43}$$

In this case, the integration is restarted at θ_j using γ_j as the new initial value, and condition (2.43) is transformed into an equality constraint in the minimization problem (Fraser-Andrews and Oberle, 1999).

Case 2: $\theta_j = \tau_k$**.**

If a multiple shooting node coincides with an arc node, then the inner arc conditions (2.38) must be satisfied. In this case, the integration is restarted at τ_k with γ_j as the initial value of y, and the corresponding arc condition is also transformed into an equality constraint in the minimization problem (Fraser-Andrews and Oberle, 1999).

In both cases, the two-point boundary condition (2.39) is added as an additional equality constraint (Fraser-Andrews and Oberle, 1999). The resulting problem is therefore a finite-dimensional constrained minimization problem in the parameter vector q.

The multiple NOC shooting, can then be summarized as (Fraser-Andrews and Oberle, 1999):

1. estimate $q = (\bar{\tau}, \bar{y}(a), \nu, \gamma)$
2. minimize I with respect to q, subject to $\dot{y} = f_k(u_k, y)$, $k = 1, ..., s-1$ and to conditions at τ and θ, adding the restart integration at multiple shooting points

Finally, the main feature of multiple NOC shooting combine some features of multiple shooting with others from NOC shooting. From multiple shooting, the method takes the following features (Fraser-Andrews and Oberle, 1999):

1. The trajectory is divided in subtrajectories and then integrated directly over their subintervals.
2. Integration can be performed along different segments of the trajectory, forward or backward.

3. The first estimates are made at the beginning of each subinterval, with the foreknowledge of a "rough solution of an application"(Fraser-Andrews and Oberle, 1999, 304).

4. The integrations in the subintervals are independent and this allows to solve difficulties in the first estimate of the parameters.

On the other hand, from NOC shooting the main features this method inherits are (Fraser-Andrews and Oberle, 1999):

1. The optimal control problem is reduced to a finite-dimensional constrained minimization problem in the parameter vector q.

2. To compute minimization, there are many algorithms available.

2.4 Comparison of methods

Until now we have provided an explanation for how the different shooting methods work. To conclude this chapter, it is necessary to make a final comparison to understand when it is useful to apply one method or another. Each method has its peculiar aspects. The choice of the method depends on the characteristics of the boundary value problem and on the numerical properties of the system under consideration.

Let us start with the most classic comparison: Single Shooting VS Multiple Shooting. The main structural differences between these two methods are the division in subintervals of the time horizon, and in the introduction of matching conditions between subintervals in the multiple shooting formulation (Osborne, 1969). These differences have some consequences. The most obvious one is that the single shooting method is conceptually and numerically simpler than the multiple one. This is because the single shooting method is integrated starting from a single unknown initial vector, while multiple shooting methods introduce several initial vectors, one for each subinterval (Morrison et al., 1962). Although this characteristic could make the single shooting method simpler, it also increases the sensitivity to initial guess, especially when the problem is complex, i.e., with a nonlinear problem (Osborne, 1969). This is because in single shooting it is difficult to make a good initial guess for all the horizon of the IVP (Osborne, 1969). While in multiple shootings, due to the partition of the interval, the guesses are made for each interval, which makes the guess more accurate. According to (Osborne, 1969), a difficulty of the single shooting method is the instability of the initial value problem for the system of ODE. Since an error in the initial guess could have an exponential growth along the integration

interval (Clark and Petzold, 1989). However, the stability of the IVP can be mitigated by the multiple shooting method, which, due to its structure, an unstable behavior can be detected in a subinterval and assessed directly (Morrison et al., 1962). The convergence conditions imposed in the multiple shooting method are useful to enforce the continuity of the solution and prevent the propagation of error across the horizon (Carraro et al., 2014). This improves the numerical stability and convergence, especially in long-horizon and nonlinear problems, but it also complicates the problem by augmenting the dimensionality of nonlinear systems. In conclusion, the two methods have both advantages and disadvantages, the real challenge is to understand when to use which.

Another interesting comparison, discussed by Fraser-Andrews and Oberle (1999) is between multiple shooting applied to optimal control problems and NOC shooting. One advantage of multiple shooting over NOC shooting is that some ODE are more stable with the multiple shooting formulation. By subdividing the trajectory into smaller segments and integrating separately over each subinterval, multiple shooting reduces the risk of instability that may arise when integrating along the entire trajectory, as in NOC shooting Fraser-Andrews and Oberle (1999). The second advantage is that the solution is given by combining all the initial value estimates for each subinterval Fraser-Andrews and Oberle (1999). If not, there could be the same problem of single shooting, that is more difficult to estimate the value at the beginning, and this may cause overflow of the algorithm (Fraser-Andrews and Oberle, 1999). The last advantage is that it is easier to manage the difficulties in estimation (Fraser-Andrews and Oberle, 1999). Since subintervals are independent in multiple shooting, it allows local estimation errors to remain confined within a subinterval rather than affecting the full horizon (Fraser-Andrews and Oberle, 1999).

On the other hand, NOC shooting offers advantages in terms of formulation. It directly transforms the optimal control problem into a finite-dimensional constrained minimization problem (Fraser-Andrews and Oberle, 1999). This makes the method more flexible and often easier to adapt to different optimal control structures(Fraser-Andrews and Oberle, 1999). Moreover, the estimation of missing initial values may require less prior knowledge of the solution, although possibly at the cost of reduced numerical precision (Fraser-Andrews and Oberle, 1999).

Finally, multiple NOC shooting combines the structural stability benefits of multiple shooting with the parametrization framework of NOC shooting (Fraser-Andrews and Oberle, 1999). By introducing additional shooting nodes and restart parameters, it improves robustness

with respect to numerical instability. However, this improvement comes at the price of increased dimensionality of the parameter vector and a larger set of equality constraints (Fraser-Andrews and Oberle, 1999). In contrast, classical NOC shooting remains simpler and involves fewer parameters, which may result in a more straightforward implementation (Fraser-Andrews and Oberle, 1999).

In conclusion, there is no better method; however, there is a method that fits better according to the problem. In particular, based on the structure of the optimal control problem, the length of the time horizon, the degree of nonlinearity, and the desired trade-off between numerical stability and computational complexity.

In the next Chapter, we will apply single and multiple shooting to the Ramsey problem to show how the classical method can solve an economic optimal control problem. The NOC shooting family is an advanced method that could be interesting for further studies and is therefore not pursued further here.

Chapter 3

Case study: Application to the Ramsey optimal control problem

3.1 The Ramsey-Cass-Koopmans model

3.1.1 Historical and economic background

The Ramsey-Cass-Koopmans model is one of the most famous dynamic macroeconomic models, a neoclassical optimal growth model (Attanasio, 2015). The model was initially developed by Frank Ramsey in 1928 in its last publication, "A mathematical theory of savings" (Ramsey, 1928). In this article, Ramsey (1928)'s goal is to answer the question: how much income a nation should save. He answers this question by deriving a general rule, that is, "the rate of saving multiplied by the marginal utility of money should always be equal to the amount by which the total net rate of enjoyment of utility falls short of the maximum possible rate of enjoyment" (Ramsey, 1928, p.543). Ramsey (1928) starts by delineating the theory of consumption based on individual savings choices. The initial problem introduces three variables: capital (k), consumption rate $c(t)$ and labor supply $h(t)$. The goal is to achieve maximum utility taking into account the trade-off between work and consumption (Attanasio, 2015). Then, Ramsey (1928) introduces the discounting of future utilities, where he assumes that "future utilities and disutilities are discounted at a constant rate" (Ramsey, 1928, p. 549). In an infinite-horizon setting, the discount factor also captures the utility of subsequent generations (Attanasio, 2015). Basically, the initial Ramsey formulation refers to a planning problem, where a planner chooses to allocate resources optimally and tries to maximize the utility of the household (Gourinchas, 2015). From this initial formulation and idea given by Ramsey (1928), other scholars extended the theory. Cass and Koopmans in 1965 reformulated the model in a decentralized competitive equilibrium

framework, where households maximize utility and firms maximize profits (Gourinchas, 2015). In particular, in their model (Gourinchas, 2015), given prices and wages, the "households supply labor, hold capital and consume optimally" [p.2], while the firms maximize their profit by renting capital and hiring labor. The two methods are identical because they have the same conditions about no market imperfection, where the welfare theorem holds (Gourinchas, 2015). This is why the model is usually referred to as "Ramsey-Cass-Koopmans model". The model provides two formulations: one for the firms and one for the households. The model we study in the application, sections 3.2-3.3, is the maximization of the household utility, the planner problem. It's formulation will be explained in the next section 3.1.2.

To conclude this initial section, we are going to present the economic idea of maximizing utility. According to Mas-Colell et al. (1995), the utility maximization problem is a consumer's decision problem. Where the consumer is assumed to have rational, continuous and nonsatiated preferences, while the utility $u(x)$ is assumed to be a continuous function representing the preferences (Mas-Colell et al., 1995). The utility maximization problem can be defined as

$$\begin{cases} \max_{x \geq 0} \quad u(x) \\ \text{s.t.} \quad p \cdot x \leq w \end{cases} \tag{3.1}$$

where the consumer's problem is to choose the most preferred bundle given prices $p >> 0$ and wealth level $w > 0$ (Mas-Colell et al., 1995). The consumer has a Walsarian budget set that she can choose from to maximize her utility, that is $B_{p,w} = \{x \in \mathbb{R}^L_+ : p \cdot x \leq w\}$ (Mas-Colell et al., 1995). In other words, the goal of the utility maximization problem is for the consumer to choose, based on her preferences, the combination that most satisfies her utility considering the budget available to her. The problem is that some times the preferences that could bring her more satisfaction may not meet the budget. If it is lower than the budget, it means the utility is not maximized and if it goes above means that it is not a feasible solution. The figure 3.1 illustrated in Mas-Colell et al. (1995) shows the budget constraint and two indifference curves of the consumer's preferences. The curve below the budget does not maximize the utility, in fact, $\hat{u} < u(x(p, w))$, while the one that perfectly meets the budget constraint is the optimal choice where the consumer's utility is maximized $\{y \in \mathbb{R}^2_+ : u(y) = u(x(p, w))\}$.

The static utility maximization framework provides the foundation for the intertemporal problem studied in the Ramsey model (Mas-Colell et al., 1995). However, while the classical consumer problem considers a single-period choice under a budget constraint, the Ramsey

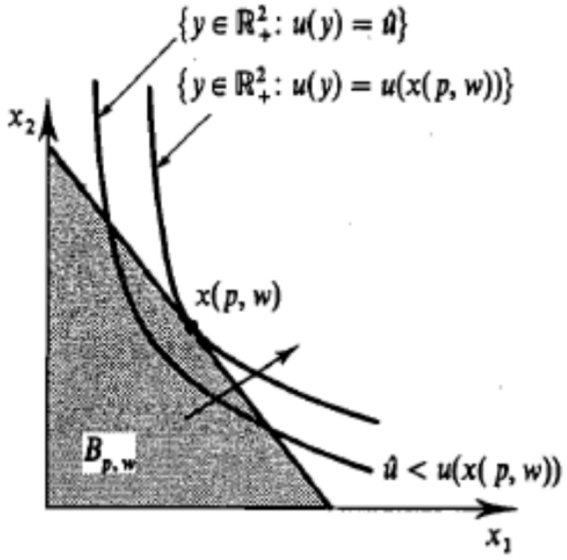

Figure 3.1: Utility maximization problem. Mas-Colell et al. (1995).

model extends this framework to an infinite horizon setting, where the representative household chooses a time path of consumption in order to maximize discounted lifetime utility (Ramsey, 1928).

3.1.2 Mathematical formulation and canonical system

The Ramsey-Cass-Koopmans model describes a representative household that maximizes lifetime utility by choosing an optimal consumption path (Barro and Sala-i Martin, 2004). In general, the household maximizes:

$$U = \int_{t=0}^{\infty} e^{-\rho t} u(C(t)) L(t) dt \tag{3.2}$$

where ρ is the discount rate. The model integrates the utility defined over consumption per worker $u(C(t))$ (Gourinchas, 2015). In fact, $C(t)$ represents the consumption per worker and $C(t)L(t)$ is the aggregate consumption, with $L(t)$ representing labor supply (or population). However, for the purpose of this thesis, we consider the per-capita formulation and assume a constant population, so $L(t)$ is omitted and $n = 0$.

Therefore, the planner problem can be written as:

$$\max_{c(t)} U = \int_{0}^{\infty} e^{-\rho t} u(c(t)) dt \tag{3.3}$$

subject to the capital accumulation constraint:

$$\dot{k}(t) = f(k(t)) - c(t) - \delta k(t), \tag{3.4}$$

with initial condition $k(0) > 0$.

Considering the corresponding Cobb-Douglas production function:

$$f(k) = k^{\alpha}, \qquad 0 < \alpha < 1, \tag{3.5}$$

The flow utility is defined as:

$$u(c(t)) = \frac{c(t)^{1-\theta} - 1}{1 - \theta}, \quad \theta > 0, \quad \rho > 0 \tag{3.6}$$

where θ is the coefficient of relative risk aversion. The coefficient θ describes how the household ranks different lotteries (Gourinchas, 2015). Moreover, θ is the inverse of the instantaneous elasticity of intertemporal substitution (IES) $\sigma = 1/\theta$. In a deterministic environment, this means that when θ is low, IES is high therefore the "marginal utility fall more slowly and the household is more willing to substitute consumption over time" (Gourinchas, 2015, p.4). However, when θ is high, then IES is low and the household falls rapidly (Gourinchas, 2015). Since the coefficient is constant, it is usually referred to as CRRA (Constant Relative Risk Aversion) (Gourinchas, 2015).

The assumption $\rho > 0$ is the convergence condition which ensures that the lifetime utility is well defined (Gourinchas, 2015). The general convergence condition is $\rho - n - (1 - \theta)g > 0$, however since we are considering the per-capita case $n = 0, g = 0$, the condition reduces to $\rho > 0$.

Let us apply Pontryagin's Maximum Principle to this problem. The current-value Hamiltonian is:

$$H(c(t), k(t), \lambda(t)) = u(c(t)) + \lambda(t)[f(k(t)) - c(t) - \delta k(t)] \tag{3.7}$$

The first-order condition with respect to $c(t)$ is:

$$\frac{\partial H}{\partial c} = u'(c(t)) - \lambda(t) = 0 \tag{3.8}$$

which implies:

$$u'(c(t)) = c(t)^{-\theta} = \lambda(t) \tag{3.9}$$

This equation describes $\lambda(t)$ as the marginal utility of consumption. Which leads to the co-state equation that is given by:

$$\dot{\lambda}(t) = \rho\lambda(t) - \lambda(t)\left(f'(k(t)) - \delta\right) \tag{3.10}$$

Using the expression for $\lambda(t)$ and differentiating, we obtain the Euler equation:

$$\frac{\dot{c}(t)}{c(t)} = \frac{1}{\theta}(f'(k(t)) - \delta - \rho) \tag{3.11}$$

which represents the optimal consumption. Substituting the Cobb-Douglas production function:

$$\dot{c} = \frac{1}{\theta}(\alpha k^{\alpha-1} - \delta - \rho)c \tag{3.12}$$

$$\dot{k} = k^{\alpha} - c - \delta k \tag{3.13}$$

where α is the capital elasticity in production and δ is the depreciation rate. This is the canonical dynamical system of the Ramsey model with initial condition $k(0) > 0$ and transversality condition:

$$\lim_{t\to\infty} e^{-\rho t}\lambda(t)k(t) = 0 \tag{3.14}$$

The steady states are obtained by imposing $\dot{c} = 0$ and $\dot{k} = 0$:

$$\alpha(k^*)^{\alpha-1} = \rho + \delta \tag{3.15}$$

which implies (Barro and Sala-i Martin, 2004):

$$k^* = \left(\frac{\alpha}{\rho + \delta}\right)^{\frac{1}{1-\alpha}} \tag{3.16}$$

$$c^* = (k^*)^{\alpha} - \delta k^* \tag{3.17}$$

The steady state is saddle-point stable. A saddle point, according to Grass et al. (2008), is an equilibrium $\hat{x}$ that satisfies $n_+ > 0$ and $n_- > 0$. An equilibrium can exhibit saddle point stability, like in the Ramsey model, where stability refers to the convergence of the saddle paths. Linearization of the system around the steady state yields one positive and one negative

eigenvalue, implying that a stable manifold exists, which is the set of points from which the solution converges (Grass et al., 2008). In infinite time horizon, like in this case, "stable path are candidates for optimal solutions" (Grass et al., 2008, p. 349).

This formulation we just saw refers to when the steady states impose the equilibrium but outside the equilibrium the consumption $c(t)$ could increase or decrease in time. This depends on the growth rate of the optimal consumption flow (Viscolani, 2025). First of all, we give an economic interpretation to the components in the Euler equation in 3.11. In fact, $f'(k)$ is the marginal productivity of capital, δ is obsolescence, and ρ is the degree of impatience (Viscolani, 2025). In particular, ρ is the preference for consuming today instead of in the future. The higher ρ, the higher is the impatience and the will of consuming today. When impatience exceeds the marginal productivity of capital, net of obsolescence, then the growth rate of optimal consumption is negative and the consumption is decreasing (Viscolani, 2025).

$$\dot{c}(t) < 0 \quad \Longleftrightarrow \quad f'(k(t)) - \delta < \rho \tag{3.18}$$

If the net returns $f'(k(t)) - \delta$ is above impatience, then the growth rate of the optimal consumption flow is positive (Viscolani, 2025). Meaning, consumption grows over time.

$$\dot{c}(t) \geq 0 \quad \Longleftrightarrow \quad f'(k(t)) - \delta \geq \rho \tag{3.19}$$

where the sign of $\dot{c}$ depends on the value of $f'(k(t))$ and consumption is increasing (Viscolani, 2025). We will see these properties of the Ramsey problem in the next section, where we will perform a sensitivity analysis and study the behavior of the model.

In equilibrium we formulated the Ramsey-Cass-Koopmans model as a two-point boundary value problem, since it has generated a two dimensional dynamical system with an initial condition on $k(0)$ and an asymptotic condition at infinity. Therefore, it can be solved using shooting methods.

3.2 Numerical implementation via Single Shooting

In this section, we solve the Ramsey-Cass-Koopmans two-point boundary value problem (BVP) that we formulated in the last section, with the single shooting method. The section is organized into three parts: a preliminary initial-guess analysis, a robust solution procedure, and a sensitivity discussion.

To solve the problem with the single-shooting method, we report two different procedures with different purposes.

- **Bounded minimization** (`minimize_scalar`, method `bounded`) is used as a preliminary search. It solves

$$\min_{c_0 \in [c_{\min}, c_{\max}]} |k(T; c_0) - k^*| .$$

  This provides a candidate value for c_0, but it just minimized the error and does not guarantee that the terminal condition is exactly satisfied.

- **Root finding** (`root_scalar`, method `brentq`) is used as the *robust final step*. It solves directly

$$F(c_0) = k(T; c_0) - k^* = 0,$$

  on a bracket where F changes sign. Under this condition, Brent's method is robust and converges reliably.

Therefore, the first procedure is exploratory, while the second one is the actual solver for the boundary condition.

3.2.1 Initial guess analysis

Before solving the Ramsey problem with the single shooting method, we import the required libraries: plotting tools, `solve_ivp` for integrating the ODE system, and `root`/`root_scalar` for scalar root finding. We then assign the model and numerical parameters: Ramsey parameters, a finite horizon $T = 120$ to approximate the infinite horizon, initial capital k_0, and ODE tolerances.

Listing 3.1: Ramsey model parameters

```
# Ramsey parameters
alpha = 0.33   # output elasticity of capital
delta = 0.08   # depreciation rate
rho = 0.04     # discount rate
theta = 2.0    # relative risk aversion (CRRA)

# Numerical horizon (infinite-horizon approximation)
T = 120.0

# Initial capital
```

```
k0 = 0.5


# ODE tolerances
RTOL = 1e-8
ATOL = 1e-10
```

The steady state is computed as

$$k^* = \left(\frac{\alpha}{\rho + \delta}\right)^{\frac{1}{1-\alpha}}, \qquad c^* = (k^*)^\alpha - \delta k^*,$$

which gives:

```
k* = 4.526061, c* = 1.283756
```

Next, we test two initial guesses for $c(0)$ and plot them. For each trial value of c_0, the Ramsey system is numerically integrated over $[0, T]$ using `scipy.integrate.solve_ivp` (with tolerances `RTOL` and `ATOL`). The shooting residual is then defined as

$$F(c_0) = k(T; c_0) - k^*.$$

Numerically, the enforced terminal condition is $k(T) \approx k^*$, while c^* is shown only as a reference line.

```
Initial guesses used: [0.44969456, 0.45369456]
```

The trajectories and their terminal offsets relative to the target are shown in figure 3.2.

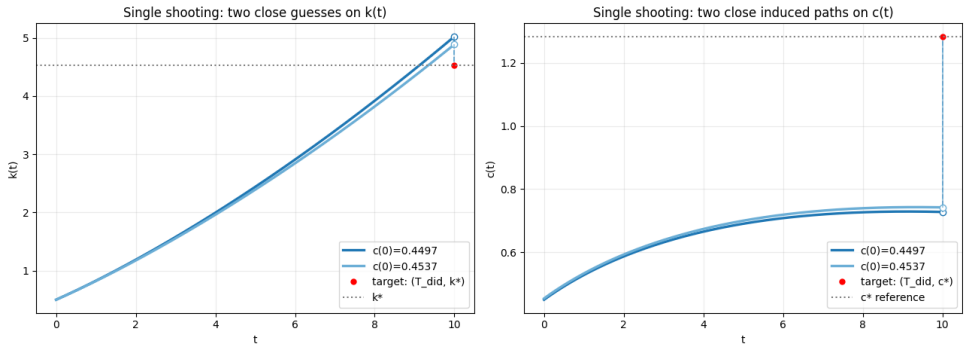

Figure 3.2: Output Initial Guesses. Personal elaboration.

A separate optimization attempt using bounded minimization gives:

```
Single shooting - optimization attempt:
  convergence (|error| < 1e-2): False
```

```
  c(0): 0.50283454
  terminal error k(T)-k*: 2.380e+01
  time (s): 0.2077
  function evaluations: 28
```

This attempt does not satisfy the terminal condition accurately.

3.2.2 Robust solution with the single shooting method

We now compute $c(0)$ with a robust strategy. Define the shooting residual

$$F(c_0) = k(T; c_0) - k^*.$$

After finding a sign-changing bracket for F, we solve $F(c_0) = 0$ with `root_scalar(..., method="brentq")`. The Brent method is a robust bracketing algorithm (bisection + secant + inverse quadratic interpolation), and it requires opposite signs at the bracket endpoints.

```
rs = root_scalar(terminal_error, bracket=bracket, method="brentq")
c0_single = rs.root
single_success = rs.converged
single_nfev = rs.function_calls
single_time = time.perf_counter() - t0

sol_single = simulate_given_c0(c0_single, T_local=T, n_points=800)
err_single = terminal_error(c0_single)
single_success = bool(single_success and sol_single.success and np.
   isfinite(err_single))
```

The resulting trajectory is shown:

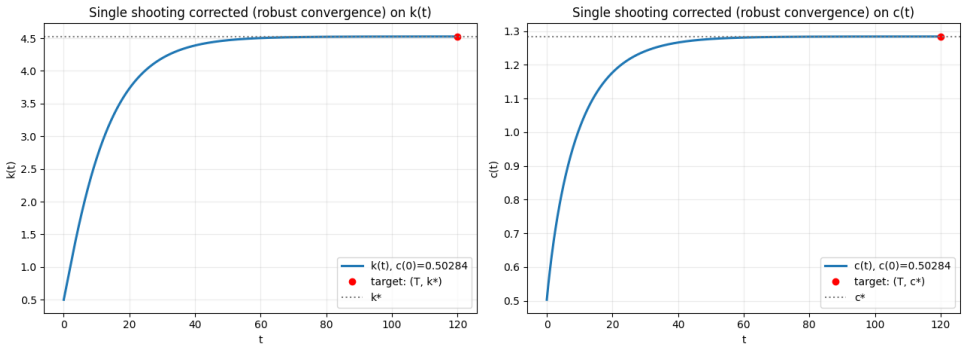

Figure 3.3: Output Robust Solution. Personal elaboration.

The output is:

```
Single shooting corrected:
  convergence: True
  bracket used: (0.5008403361, 0.5130252101)
  c(0): 0.50283810
  terminal error k(T)-k*: -3.170e-06
  time (s): 0.2560
  function evaluations: 35
```

Hence, the robust single-shooting solution satisfies the boundary condition with a very small terminal error.

3.2.3 Sensitivity analysis with single shooting method

In the previous section, we obtained a robust transition path with the single-shooting method that converges to the steady state. We now study two regimes based on the sign of the initial consumption growth rate, namely $\dot{c}(0) > 0$ and $\dot{c}(0) < 0$, to understand how the model dynamics change when $\dot{c}$ changes sign.

To do this, we vary k_0 while keeping the other parameters fixed. The choice of k_0 is based on the threshold level k_{thr}, obtained from the Euler condition:

$$f'(k) - \delta - \rho = 0.$$

With Cobb–Douglas technology, $f'(k) = \alpha k^{\alpha-1}$, so:

$$\alpha k^{\alpha-1} - \delta - \rho = 0.$$

Solving for k gives:

$$k_{\text{thr}} = \left(\frac{\alpha}{\delta + \rho}\right)^{\frac{1}{1-\alpha}}. \tag{3.20}$$

With our baseline parameters, this yields $k_{\text{thr}} \approx 4.5$.

We then define two cases around this threshold: - $\dot{c}(0) > 0$ when $k_0 < k_{\text{thr}}$, - $\dot{c}(0) < 0$ when $k_0 > k_{\text{thr}}$.

Therefore, we use the following scenarios for the sensitivity analysis:

```
k_threshold = (alpha / (rho + delta)) ** (1.0 / (1.0 - alpha))
single_scenarios = [
```

```
    {'name': 'Case A (dot c(0) > 0)', 'alpha': alpha, 'delta': delta,
        'rho': rho, 'theta': theta, 'k0': 1.0},
    {'name': 'Case B (dot c(0) < 0)', 'alpha': alpha, 'delta': delta,
        'rho': rho, 'theta': theta, 'k0': 6.0},
]
```

For each case, we compute the steady state, integrate the Ramsey ODE with `solve_ivp`, define the terminal error, and solve for $c(0)$ using `root_scalar` (Brent method), as in the robust single-shooting section:

```
rs_loc = root_scalar(terr, bracket=bracket, method='brentq')
c0_sol = rs_loc.root
ok = rs_loc.converged
nfev = rs_loc.function_calls
elapsed = time.perf_counter() - t0_loc

sol_fin = simulate(c0_sol, n_points_local=n_points)
err_fin = terr(c0_sol)
ok = bool(ok and sol_fin.success and np.isfinite(err_fin))
```

In the output, we also report:

```
'g0': dc_over_c(k0_local, alpha_s, delta_s, rho_s, theta_s)
```

where

$$g_0 = \frac{\dot{c}}{c}$$

Its sign directly identifies whether $\dot{c}(0) > 0$ or $\dot{c}(0) < 0$.

The results are shown in Figure 3.4.

The corresponding numerical summary is:

```
--- Single shooting sensitivity summary (dot c criterion) ---
Case A (dot c(0) > 0): k0=1.00, rho=0.0400, g0=1.050e-01 (dot c(0)>0)
   , c0=0.667084, err=-2.088e-08, time=0.1490s
Case B (dot c(0) < 0): k0=6.00, rho=0.0400, g0=-1.033e-02 (dot c(0)
   <0), c0=1.460509, err=-1.729e-06, time=0.1283s
```

These results confirm the theory discussed in Section 3.1.2. In Case A ($(k_0 < k_{\text{thr}})$), $(g_0 > 0)$ implies a positive initial growth rate of optimal consumption, so the trajectory of $(c(t))$ is increasing. In this regime, impatience (ρ) does not exceed the net marginal productivity of

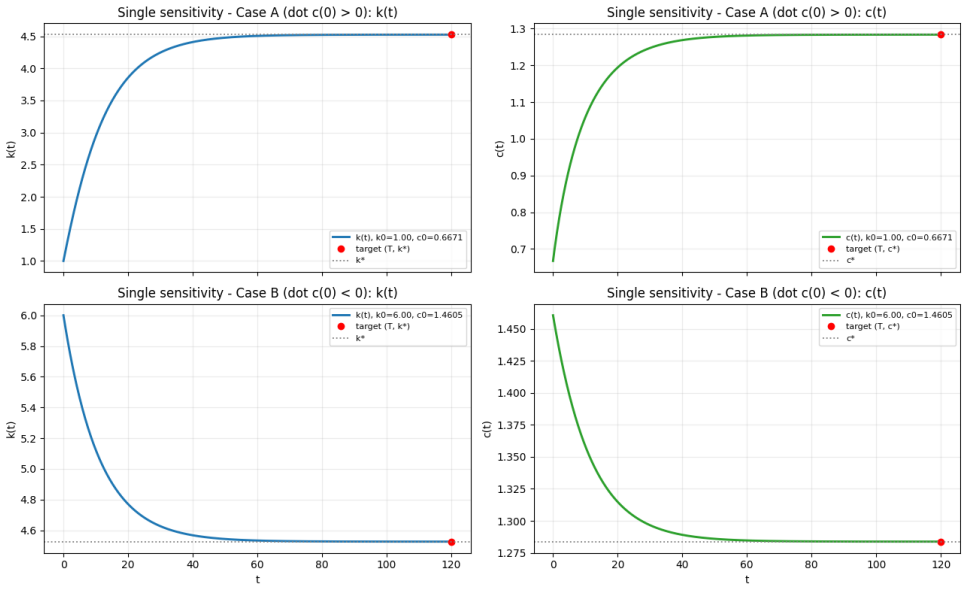

Figure 3.4: Output sensitivity analysis via single shooting. Personal elaboration.

capital (i.e., marginal productivity net of depreciation), so consumption is sustained and tends to rise over time. In Case B ($(k_0 > k_{\text{thr}})$), with $(k_0 = 6)$ in our experiment), $(g_0 < 0)$ implies a negative initial growth rate of optimal consumption, so $(c(t))$ declines. Here, impatience is relatively stronger than the net marginal productivity, which shifts consumption toward earlier periods and leads to lower consumption growth later on.

Economically, this follows from:

$$\frac{\dot{c}}{c} = \frac{1}{\theta}\left(\alpha k^{\alpha-1} - \delta - \rho\right). \tag{3.21}$$

When the net marginal product of capital exceeds impatience, consumption initially grows; when impatience exceeds the net marginal product, consumption initially declines.

3.3 Numerical implementation via Multiple Shooting

Here we follow the same path as in the single shooting method. First, we present a non-convergence example, then the robust solution, and finally the sensitivity analysis.

3.3.1 Non-convergence example

The main difference of this method, compared to the previous one, is that the time interval is divided into subintervals. At each node, continuity is enforced, and the terminal condition $k(T) \approx k^*$ is imposed.

First, we force an unfavorable numerical setup (poor initial guess + few iterations). The goal is to show that convergence is not automatic in multiple shooting either. The problem setup is the same as in the single shooting method, so the parameters are the same. Here, we only add the number of intervals, which is 15. Then, we solve the Ramsey ODE with `scipy.integrate.solve_ivp`. The resulting plot is shown in Figure 3.5.

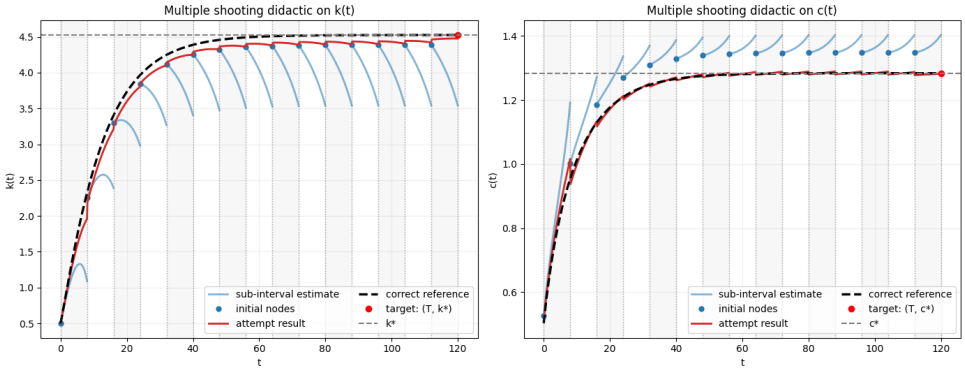

Figure 3.5: Output Non-Convergence Example. Personal elaboration.

In the plot, we can see that the time interval is divided into subintervals and that there are guesses at each node. The resulting estimates do not converge to the solution and do not satisfy the continuity conditions. In fact, they diverge, and it is not possible to obtain a continuous solution in this example.

The output of the example is:

```
  Multiple shooting - attempt:
convergence: False
message: The number of calls to function has reached maxfev = 1.
terminal error k(T)-k*: -4.900e-02
time (s): 0.2451
```

The output reflects the plot. In fact, the error is relatively large and the method does not converge.

3.3.2 Robust solution with multiple shooting

Now, we provide a robust solution to the Ramsey model with the multiple shooting method. As in single shooting, we solve for the unknown initial condition and internal node values by minimizing the residual equations. In this case, we use `root(..., method="hybr")` (not `root_scalar`) to solve the nonlinear system of continuity and terminal conditions.

```
nodes_guess = unpack_nodes_generic(x0)
```

```
segments_guess = reconstruct_segments(nodes_guess)

t0 = time.perf_counter()
rm = root(residual_multiple_generic, x0, method="hybr")
multi_time = time.perf_counter() - t0

nodes_star = unpack_nodes_generic(rm.x)
t_ms, k_ms, c_ms = reconstruct_path(nodes_star)
err_multi = k_ms[-1] - k_star
```

We set initial guesses at each node and then impose continuity conditions. The final plot is shown in Figure 3.6.

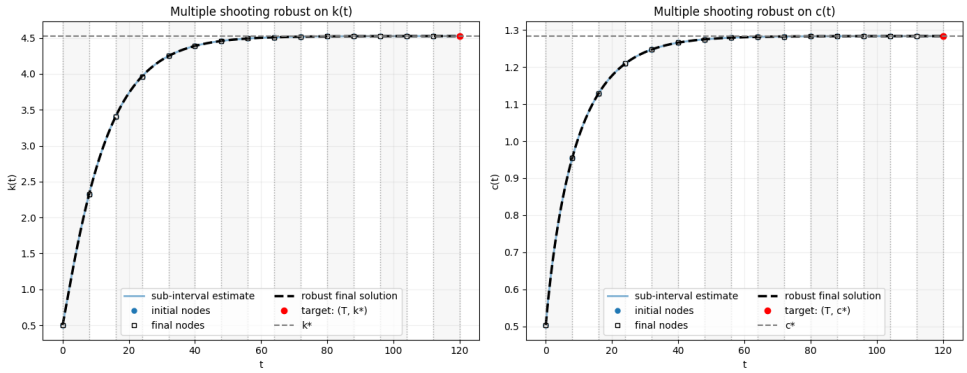

Figure 3.6: Output Robust Solution. Personal elaboration.

The final output is:

```
   Multiple shooting (robust):
  convergence: True
  message: The solution converged.
  c(0): 0.50283810
  terminal error k(T)-k*: 0.000e+00
  time (s): 0.1597
  function evaluations: 34
```

The combination of model parameters and the infinite-horizon approximation yields a terminal error numerically equal to zero (up to machine precision), indicating an excellent numerical solution.

3.3.3 Sensitivity analysis

The same sensitivity analysis is performed using the multiple-shooting method. The motivation and economic interpretation are the same as in Section 3.2.3. We use the same two scenarios, defined by the sign of $\dot{c}(0)$ (equivalently $g_0 = \dot{c}(0)/c(0)$), with different initial capital values, $k_0 = 1$ and $k_0 = 6$.

The main difference is numerical. In multiple shooting, the time horizon is split into subintervals through temporal nodes. For each subinterval, the Ramsey ODE is integrated with `solve_ivp`; then continuity conditions between adjacent nodes are imposed, together with the terminal condition on capital, $k(T) \approx k^*$. A robust initial guess is obtained from the single-shooting solution and then interpolated on the nodes. The nonlinear system is solved with `root`, first with method `hybr`, and, if needed, with `lm`.

```
rm_loc = None
t0_loc = time.perf_counter()
for method in ['hybr', 'lm']:
    trial = root(residual, x0_loc, method=method)
    rm_loc = trial
    if trial.success:
        break
elapsed = time.perf_counter() - t0_loc

t_all = np.array(t_all)
k_all = np.array(k_all)
c_all = np.array(c_all)
err_fin = k_all[-1] - k_star_s
ok = bool(rm_loc.success and np.isfinite(err_fin))
```

The full trajectory is then reconstructed by concatenating all subinterval segments, and the final terminal error is computed.

The final plot is shown in Figure 3.7.

The output is:

```
--- Multiple shooting sensitivity summary (dot c criterion) ---
Case A (dot c(0) > 0): k0=1.00, rho=0.0400, g0=1.050e-01 (dot c(0)>0)
   , c0=0.667084, err=1.776e-15, time=0.0883s, nfev=24
Case B (dot c(0) < 0): k0=6.00, rho=0.0400, g0=-1.033e-02 (dot c(0)
   <0), c0=1.460509, err=-2.665e-14, time=0.0704s, nfev=24
```

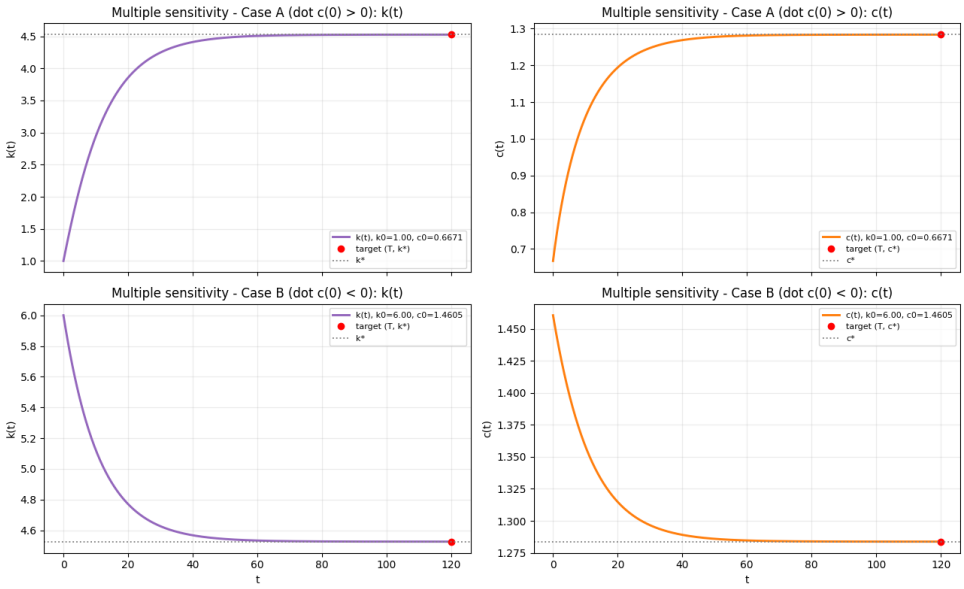

Figure 3.7: Output sensitivity analysis via multiple shooting. Personal elaboration.

The results are consistent with theory. In Case A, consumption has a positive initial growth rate; in Case B, consumption has a negative initial growth rate. Accordingly, g_0 is positive in the first case and negative in the second. A comparison with single shooting is provided in the next section.

3.4 Numerical comparison and discussion

This section compares the numerical results obtained with robust single shooting and robust multiple shooting.

First, we compare the robust solutions, with same parameters and same horizon, across the two methods. The main result is that single shooting is more sensitive to the initial guess: small changes in $c(0)$ may lead to noticeably different trajectories before root-finding correction. After convergence, single shooting provides an accurate solution, but it remains more exposed to local instability. Multiple shooting is generally more robust because the horizon is split into subintervals and continuity constraints are imposed at each node. This increases the number of unknowns and the computational burden, but usually improves numerical stability and reduces terminal error.

In our setup, both methods converge to the same economic path. For $k_0 = 0.5$, the optimal initial consumption is approximately $c(0) \approx 0.501$. The multiple-shooting terminal error is close to machine precision.

Table 3.1: Baseline comparison: robust single shooting vs robust multiple shooting

Method	Convergence	$c(0)$	$k(T) - k^*$	Time (s)
Single shooting (corrected)	True	0.50283810	-3.170×10^{-6}	0.2540
Multiple shooting (robust)	True	0.50283810	0.000×10^{0}	0.1597

Next, we report the initial-condition sensitivity analysis performed in the last sections. Model parameters are fixed, while k_0 is changed to generate two regimes:

$$\frac{\dot{c}(0)}{c(0)} = \frac{1}{\theta}\left(\alpha k_0^{\alpha-1} - \delta - \rho\right) \quad \Rightarrow \quad \dot{c}(0) \gtrless 0.$$

Compared with single shooting, multiple shooting again yields smaller terminal errors and more stable convergence.

Table 3.2: Sensitivity comparison (dot c criterion): single vs multiple shooting

Method	Case	k_0	ρ	g_0	$c(0)$	$k(T) - k^*$
Single	A ($\dot{c}(0) > 0$)	1.00	0.0400	1.050×10^{-1}	0.667084	-2.088×10^{-8}
Single	B ($\dot{c}(0) < 0$)	6.00	0.0400	-1.033×10^{-2}	1.460509	-1.729×10^{-6}
Multiple	A ($\dot{c}(0) > 0$)	1.00	0.0400	1.050×10^{-1}	0.667084	1.776×10^{-15}
Multiple	B ($\dot{c}(0) < 0$)	6.00	0.0400	-1.033×10^{-2}	1.460509	-2.665×10^{-14}

Finally, for completeness, we include a parametric sensitivity analysis by varying ρ (with fixed $k_0 = 6.0$). We choose one low-ρ case with $\dot{c}(0) > 0$ and one high-ρ case with $\dot{c}(0) < 0$, using

$$\rho_{\text{thr}} = \alpha k_0^{\alpha-1} - \delta.$$

Hence:

$$\rho < \rho_{\text{thr}} \Rightarrow \dot{c}(0) > 0, \qquad \rho > \rho_{\text{thr}} \Rightarrow \dot{c}(0) < 0.$$

For this analysis the plot is shown in figure 3.8;

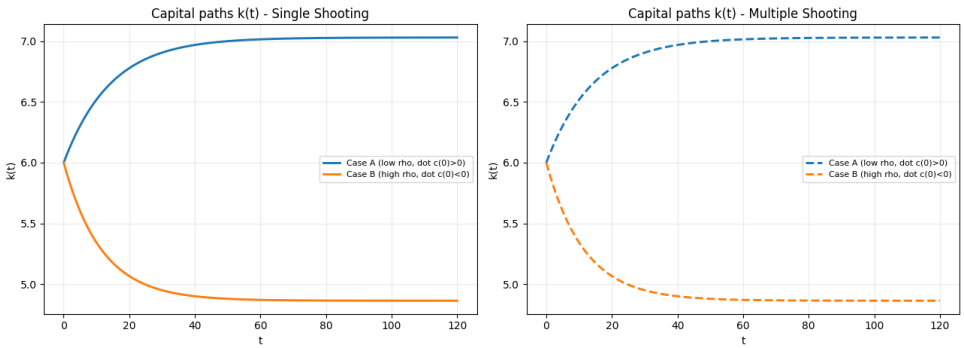

Figure 3.8: Output parameter sensitivity analysis. Personal elaboration.

and the output is represented in this table 3.3:

Table 3.3: Two-case parametric sensitivity on ρ ($k_0 = 6.0$, $\rho_{\text{thr}} = 0.019347$)

Method	Case	ρ	g_0	$c(0)$	$k(T) - k^*$	Time (s)
Single	A ($\dot{c}(0) > 0$)	0.009347	5.000×10^{-3}	1.254594	-6.495×10^{-9}	0.0531
Single	B ($\dot{c}(0) < 0$)	0.034347	-7.500×10^{-3}	1.425257	1.940×10^{-9}	0.1595
Multiple	A ($\dot{c}(0) > 0$)	0.009347	5.000×10^{-3}	1.254594	5.329×10^{-14}	0.0601
Multiple	B ($\dot{c}(0) < 0$)	0.034347	-7.500×10^{-3}	1.425257	-5.684×10^{-14}	0.0645

From the plot, we can see that the results confirmed the theory and show both the increasing consumption in the case where the impatience is low and a decreasing consumption when the impatience is high. In particular, when ρ is high, the impatience is greater, therefore, encouraging earlier consumption, which reduces the initial growth of consumption.

From an economic point of view, both numerical methods identify the same saddle-path equilibrium, i.e. the unique stable trajectory converging to (k^*, c^*). This is also consistent with the transversality condition of the infinite-horizon Ramsey problem that we saw in section 3.1.2.

Overall, both methods recover the expected economic dynamics, but multiple shooting is numerically more robust, especially in sensitivity exercises. Moreover, in the cases that we present the multiple shooting method was faster than the single shooting one, but it is not always guaranteed, it depends on the problem setup.

3.5 Limitations and alternative numerical approaches

This study highlights the main strengths and weaknesses of shooting-based methods for solving the Ramsey boundary value problem (BVP).

A first limitation is the high sensitivity of single shooting to the initial guess for $c(0)$. Because the method propagates one initial-value trajectory over the full horizon, small initialization errors may generate large terminal deviations, especially in nonlinear systems and in long-horizon settings (Osborne, 1969). In practice, this may produce instability, slow convergence, or failure of the root-finding step.

Multiple shooting improves robustness by splitting the horizon into subintervals and imposing continuity conditions at internal nodes. This reduces the impact of local instability and, in our experiments, yields significantly smaller terminal errors. However, the method is not free of limitations: it introduces many additional unknowns, increases the dimensionality of the nonlinear system, and may still face convergence difficulties in highly nonlinear or poorly initialized problems (Morrison et al., 1962). Therefore, robustness is gained at the cost of higher algebraic complexity.

More generally, both shooting approaches inherit typical BVP challenges: dependence on initialization, potential non-convergence, and sensitivity to discretization choices (horizon length, node density, and tolerances). These aspects are particularly relevant in infinite-horizon models approximated numerically on a finite horizon.

For these reasons, alternative numerical strategies are relevant for future work. A first option is the time-elimination method, often used in macroeconomic dynamics to transform the system and simplify numerical treatment, by eliminating time from the equations (Barro and Sala-i Martin, 2004). A second option is the finite difference method, which discretizes derivatives on a grid and solves the resulting global algebraic system; this approach can be more stable in some BVP settings (Wu, 2024). A third option is collocation, which approximates the solution by basis functions and enforces the differential equations at selected nodes, typically achieving good accuracy with relatively few points (Wu, 2024). Finally, the finite element method provides a flexible weak-form framework and is widely used for complex BVPs in applied mathematics and engineering (Wu, 2024).

Overall, future research could compare these methods on the same Ramsey setup, evaluating convergence reliability, terminal accuracy, and computational cost under identical numerical settings. Moreover, it could also be interesting to apply different shooting methods techniques, such as NOC shooting and direct shooting. Finally, another in-depth analysis could be to construct a heat map over the initial guess $c(0)$ and the horizon T.

Conclusion

The main objectives of this thesis, described in the introduction, were: (i) to provide the theoretical background on dynamical systems and optimal control needed for the Ramsey problem, (ii) to review the different shooting methods, and (iii) to implement and compare these methods numerically in the Ramsey-Cass-Koopmans framework.

The analysis shows that the Ramsey model can be formulated as a continuous-time, nonlinear, autonomous, deterministic dynamic system. Using Pontryagin's Maximum Principle, the planner's problem is transformed into a two-point boundary value problem, which is suitable for shooting-based numerical methods.

The numerical comparison highlights that single shooting is more sensitive to the initial guess, especially in nonlinear and long-horizon settings. Multiple shooting, by splitting the horizon into subintervals and enforcing continuity conditions, is computationally more structured and generally more robust.

Finally, with the numerical implementation, the comparison between single and multiple shooting methods becomes clear. In fact, the results show that the multiple shooting method provides the most robust solution to the Ramsey problem, achieving convergence to the solution with the smallest terminal error (in the setup $k(T) - k^* = 0.00$). Moreover, the sensitivity analysis provides further insight into the robustness of the model by stressing it with alternative parameter configurations, varying both k_0 and ρ.

In conclusion, this thesis aims to provide a methodological and applied contribution to the numerical solution of an economic optimal control problem. In the future, shooting methods could be used more widely to solve boundary value problems that cannot be solved analytically. Future work may extend the comparison to alternative global methods (e.g., finite differences, collocation, and finite elements) and to other economic models.

Code availability

The final notebook used for the numerical simulations is available at: https://github.com/elipreatoni-ux/Ramsey-Shooting-Thesis/blob/main/ramsey_shooting_final.ipynb (accessed on 1 March 2026).

Bibliography

Ascher, U. M., Mattheij, R. M. M., and Russell, R. (1995). Numerical solution of boundary value problems for ordinary differential equations. In *Classics in applied mathematics*.

Attanasio, O. (2015). Frank ramsey's a mathematical theory of saving. *The Economic Journal*, 125.

Barro, R. and Sala-i Martin, X. (2004). *Economic growth second edition*. Cambridge MA.: The MIT Press.

Bellman, R. (1957). *Dynamic Programming*. Princeton University Press.

Benettin, G. (2012). Una passeggiata tra i sistemi dinamici. Lecture notes for the Scuola Galileiana di Studi Superiori.

Bischi, G. I., Radi, F., and Lamantia, D. (2015). Lecture notes on dynamical systems in economics and finance. Lecture notes.

Bock, H. G. and Plitt, K. J. (1984). A multiple shooting algorithm for direct solution of optimal control problems. *IFAC Proceedings Volumes*, 17:1603–1608.

Brüls, O., Bastos, G. J., and Seifried, R. (2014). A stable inversion method for feedforward control of constrained flexible multibody systems. *Journal of Computational and Nonlinear Dynamics*, 9(1):011014.

Carraro, T., Geiger, M., and Rannacher, R. (2014). Indirect multiple shooting for nonlinear parabolic optimal control problems with control constraints. *SIAM Journal on Scientific Computing*, 36:A452–A481.

Clark, K. D. and Petzold, L. R. (1989). Numerical solution of boundary value problems in differential-algebraic systems. *Siam Journal on Scientific and Statistical Computing*, 10:915–936.

Edun, I. and Akinlabi, G. (2021). Application of the shooting method for the solution of second order boundary value problems. *Journal of Physics: Conference Series*, 1734:012020.

Filipov, S., Gospodinov, I., and Faragó, I. (2017). Shooting-projection method for two-point boundary value problems. *Applied Mathematics Letters*, 72.

Fraser-Andrews, G. and Oberle, H. (1999). A multiple-shooting technique for optimal control. *Journal of Optimization Theiry and Applications*, 102(2):299–313.

Gourinchas, P.-O. (2015). Notes for econ202a: The ramsey-cass$_k oopmansmodel. Lecturenotes$.

Grass, D., Caulkins, J., Feichtinger, G., Tragler, G., and Behrens, D. (2008). *Optimal Control of Nonlinear Processes: With Applications in Drugs, Corruption, and Terror*. Springer-Verlag Berlin Heidelberg.

Hunter, J. K. (2011). Introduction to dynamical systems. *UCDavis Mathematics MAT A*, 207:1–33.

Irwin, M. and Wang, Z. (2017). Dynamic systems modeling. *The international encyclopedia of communication research methods*, pages 1–12.

Klamkin, M. S. (1970). Transformation of boundary value problems into initial value problems. *Journal of Mathematical Analysis and Applications*, 32(2):308–330. Submitted by R. Bellman; received October 29, 1969.

Kong, Q., Siauw, T., and Bayen, A. (2020). *Python Programming and Numerical Methods: A Guide for Engineers and Scientists*. Academic Press, London, UK, 1st edition.

Mas-Colell, A., Whinston, M. D., and Green, J. R. (1995). *Microeconomic Theory*. Oxford University Press.

Mellodge, P. (2016). *A Practical Approach to Dynamical Systems for Engineers*. Elsevier Ltd.

Morrison, D. D., Riley, J. D., and Zancanaro, J. F. (1962). Multiple shooting method for two-point boundary value problems. *Communications of the ACM*, 5:613 – 614.

Nouri, A. and Seyyedsalehi, S. A. (2023). Eigen value based loss function for training attractors in iterated autoencoders. *Neural Networks*, 161:575–588.

Osborne, M. R. (1969). On shooting methods for boundary value problems. *Journal of Mathematical Analysis and Applications*, 27:417 – 433.

Ossimitz, G. and Mrotzek, M. (2008). The basics of system dynamics: discrete vs. continuous modelling of time. In *Proceedings of the 26th International Conference of the System Dynamics Society*.

Park, C. (2007). *A Dictionary of Environment and Conservation*. Oxford University Press, 1 edition. Online version updated in 2012.

Pradeep, S. and Shrivastava, S. K. (1990). Stability of dynamical systems – an overview. *Journal of Guidance, Control, and Dynamics*, 13(3):385–393.

Ramsey, F. P. (1928). A mathematical theory of saving. *The Economic Journal*, 38(152):543–559.

Strogatz, S. (2024). *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*. CRC Press, Taylor Francis Group.

Thorn, C. and Welford, M. (1994). The equilbrium concept in geomorphology. *Annals of the Association of American Geographers*, 84.

Viscolani, B. (2025). Neoclassical economic growth (ramsey ii). Lecture notes, Mathematics for Economics course, Degree Programme in Mathematics. Unpublished course slides.

Vyasarayani, C. P., Uchida, T., and McPhee, J. (2012). Single-shooting homotopy method for parameter identification in dynamical systems. *Phys. Rev. E*, 85:036201.

Wu, Y. (2024). Comparing and contrast different numerical approaching to the ivp and bvp problems. *Highlights in Science, Engineering and Technology*, 115:491–511.