Applying Fractional Optimal Control to Real-World Problems

By:

Youssif Ahmed Abdallah

Yousef Ayman Mohtady

Marwan Ali Mohamed

Class of 2025 CS & Math. Major

Supervised by:

Dr. Salma Asaad Mohamed Shatta

Lecturer in Mathematics department

بحث ومقال بعنوان

بعض تطبيقات التحكم الأمثل في مشاكل الحياة الواقعية

اعداد

يوسف أحمد عبدالله

يوسف أيمن مهتدي

مروان علي محمد

الفرقة الرابعة شعبة الرياضيات و الحاسب

تحت إشراف

د. سلمى أسعد محمد شطا

المدرس بقسم الرياضيات

للعام الدراسي

2024-2025

الغلاف الداخلى لنسخة الكنترول

الاسم: يوسف أحمد عبدالله

يوسف أيمن مهتدي

مروان علي محمد

الفرقة: الرابعة

الشعبة: رياضة و حاسب

المشرف على المشروع: د. سلمى أسعد محمد شطا

لجنة الحكم والمناقشة :

1. د. ضحى محمود عبد التواب
2. د. ايمان إبراهيم طه إبراهيم
3. د. سلمى أسعد محمد شطا

المجموع الكلى
100

Abstract

Abstract

This study explores the importance and applications of mathematical modeling with a particular focus on optimal control in epidemic models.

We discuss fundamental concepts such as Pontryagin's Maximum Principle and the Hamiltonian function, as well as numerical techniques like the Runge-Kutta method and forward and backward Sweep Method. Additionally, we examine the role of epidemic modeling in understanding disease spread and its application in optimizing cotton leaf curl virus treatment.

Finally, we discuss the application of the Forward and Backward Sweep Method to an epidemic model, solved using Python.

Abbreviations

Abbreviations

ODEs	Ordinary Differential Equations
FDEs	Fractional Differential Equations
PMP	Pontryagin’s Maximum Principle
RK4	Runge-Kutta 4th Order Method
BVP	Boundary Value Problem
TPBVP	Two-Point Boundary Value Problem
SEIRN	Susceptible - Exposed - Infectious - Recovered - Total Population (Epidemic Model)
CLCuV	Cotton Leaf Curl Virus
IVP	Initial Value Problem
RL	Riemann-Liouville (fractional derivative/integral)
GL	GL : Grünwald-Letnikov (fractional derivative)
C	Caputo (fractional derivative)

Table of Contents

Contents

Summary

This book provides a comprehensive exploration of optimal control theory, blending rigorous mathematical foundations with practical numerical methods and real-world applications. Designed for advanced undergraduates, graduate students, and researchers in applied mathematics, engineering, and epidemiology, the text bridges theoretical concepts with computational techniques to solve dynamic optimization problems.

The journey begins with Chapter 1, which introduces the fundamental principles of optimal control, including state equations, objective functionals, and Pontryagin's Maximum Principle (PMP). Through step-by-step derivations and illustrative examples, readers learn to formulate and solve basic optimal control problems.

Chapter 2 shifts focus to numerical methodologies, detailing the Runge-Kutta 4 (RK4) method for solving ordinary differential equations and the Forward-Backward Sweep Method for boundary value problems. These tools are critical for implementing optimal control strategies computationally, enabling readers to tackle complex systems that lack analytical solutions.

Chapter 3 applies these concepts to a pressing real-world challenge: epidemic management. Using an SEIRN (Susceptible-Exposed-Infectious-Recovered-Population) model, this chapter demonstrates how optimal control theory guides vaccination strategies to minimize both infection rates and intervention costs. Case studies with varying parameters underscore the interplay between model dynamics and control policies.

By integrating theory, computation, and application, this book equips readers to address diverse optimization challenges, from engineering systems to public health crises.

Chapter 4 introduces fractional calculus, an extension of traditional calculus to non-integer orders. It explores historical roots and key definitions, including Riemann-Liouville, Grünwald-Letnikov, and Caputo derivatives and integrals. The chapter discusses the unique properties of fractional operators, such as their ability to model memory effects and non-local behavior, making them valuable for simulating complex systems.

It also covers Fractional Differential Equations (FDEs), focusing on existence, uniqueness, and applications in areas like anomalous diffusion, viscoelastic materials, biological systems, and signal processing. Finally, numerical methods like Euler's method and the Runge-Kutta method are adapted for solving FDEs, enabling practical simulations of real-world models.

Chapter 5 presents a mathematical model to control the spread of the Cotton Leaf Curl Virus (CLCuV), a major threat to cotton crops transmitted by whiteflies. The model divides populations into susceptible and infected compartments for both cotton plants and vectors, forming a four-dimensional system.

An optimal control problem is formulated to reduce infections while minimizing intervention costs, using Pontryagin's Maximum Principle to derive optimal strategies. This includes defining the Hamiltonian function and solving adjoint equations to guide effective control measures, such as insecticide use or planting resistant varieties.

The chapter highlights how mathematical modeling and optimal control can support sustainable agricultural practices to combat CLCuV.

Chapter 1

CHAPTER 1

The Basic Problem and Necessary Conditions in Optimal Control

1.1 Introduction to Optimal Control Problems

In our basic optimal control problem for ordinary differential equations, we use a control function $u(t)$ that influences a system's state $x(t)$ in a way that optimizes a given objective function. The system is typically governed by a set of differential equations that describe its evolution over time.

A standard optimal control problem consists of:

1. State Equation: A differential equation that describes how the state $x(t)$ evolves over time under the influence of the control $u(t)$.
2. Objective Functional: A function that needs to be maximized (or minimized) based on the state and control variables.
3. Initial and Terminal Conditions: Constraints on the state at the beginning and sometimes at the end of the time interval.

1.2 Mathematical Formulation

A basic optimal control problem can be written as:

$$\max_{u} J(u) = \int_{t_0}^{t_1} f\big(t, x(t), u(t)\big) dt$$

subject to the state equation:

$$x'(t) = g\big(t, x(t), u(t)\big), \qquad x(t_0) = x_0$$

where:

- $x(t)$ is the state variable that describes the system,
- $u(t)$ is the control variable that we want to optimize,
- $f(t, x, u)$ is the objective function (also called the performance index),
- $g(t, x, u)$ is the dynamics function describing the system's evolution,
- $t_0 and\ t_1$ define the time interval.

The goal is to find the control function $u^*(t)$ that maximizes or minimizes $J(u)$, depending on the problem.

Necessary Conditions for Optimality

To determine the optimal control, we derive necessary conditions using Pontryagin's Maximum Principle. These conditions involve introducing an additional variable, called the adjoint variable.

1.3 Pontryagin's Maximum Principle

Pontryagin's Maximum Principle (PMP) is one of the most fundamental results in optimal control theory. It provides necessary conditions that an optimal control must satisfy and serves as a powerful tool for solving optimal control problems.

Statement of Pontryagin's Maximum Principle

Consider a dynamical system governed by the state equation:

$$\boldsymbol{x'(t) = g\big(t, x(t), u(t)\big)}$$

where:

- $x(t)$ is the state variable (describes the system's condition at time t),
- $u(t)$ is the control variable (the decision input that we optimize),
- $g(t, x, u)$ defines how the state changes over time.

The objective is to optimize the performance functional:

$$\boldsymbol{J(u) = \int_{t_0}^{t_1} f\big(t\,, x(t), u(t)\big)dt}$$

subject to the system's constraints. PMP introduces the Hamiltonian function to analyze this problem.

1.3.1 The Hamiltonian Function

The Hamiltonian is defined as:

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

where:

- $\lambda(t)$ is the adjoint (or costate) variable, which acts like a Lagrange multiplier in calculus of variations.

1.3.2 Necessary Conditions for Optimality

If $u^*(t)$ is an optimal control and $x^*(t)$ is the corresponding optimal trajectory, then there exists an adjoint function $\lambda(t)$ such that:

1. State Equation: The optimal trajectory satisfies:

$$\boldsymbol{x'^*(t) = g\big(t, x^*(t), u^*(t)\big)}$$

2. Adjoint Equation: The costate variable satisfies:

$$\boldsymbol{\lambda'(t) = -\frac{\partial H}{\partial x}}$$

3. Optimality Condition: The optimal control maximizes the Hamiltonian:

$$\boldsymbol{H(t, x^*, u^*, \lambda) \geq H(t, x^*, u, \lambda), \forall u \in U}$$

4. Transversality Condition (if applicable): If the final state is not fixed, then

$$\boldsymbol{\lambda(t_1) = 0}$$

1.4 How to Use Pontryagin’s Maximum Principle

To solve an optimal control problem using PMP, follow these steps:

Step 1: Define the Hamiltonian

Write the Hamiltonian function:

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

Step 2: Compute the Adjoint Equation

Find $\lambda'(t)$ from:

$$\boldsymbol{\lambda'(t) = -\frac{\partial H}{\partial x}}$$

Step 3: Find the Optimal Control

Solve:

$$\boldsymbol{\frac{\partial H}{\partial u} = 0}$$

to determine $\boldsymbol{u^*(t)}$.

Step 4: Solve the System of Equations

Solve the state and adjoint equations together to find $x^*(t)$, $\lambda^*(t)$, and $u^*(t)$.

1.5 Example Application

Problem Statement

Find the control $u(t)$ that minimizes:

$$\min_{u} \int_0^1 u(t)^2 dt$$

subject to:

$$x'(t) = x(t) + u(t), \qquad x(0) = 1$$

Step 1: Define the Hamiltonian

$$f = u^2$$

$$g = x + u$$

$$H = f + \lambda g \Rightarrow H = u^2 + \lambda (x + u).$$

Step 2: Compute the Adjoint Equation

$$\lambda'(t) = -\frac{\partial H}{\partial x} = -\lambda$$

with $\lambda(1) = 0$.

Step 3: Find the Optimal Control

$$\frac{\partial H}{\partial u} = 2u + \lambda = 0 \quad \Rightarrow \quad u^* = -\frac{\lambda}{2}$$

Step 4: Solve for $\mathbf{x}^*(\mathbf{t})$ and $\boldsymbol{\lambda}^*(\mathbf{t})$

Solve the system of differential equations to determine $\boldsymbol{x}^*(\boldsymbol{t})$, $\boldsymbol{\lambda}^*(\boldsymbol{t})$, and $\boldsymbol{u}^*(\boldsymbol{t})$.

Chapter 2

CHAPTER 2

Numerical Methods for Optimal Control

2.1 Introduction

Numerical methods provide ways to find approximate solutions to mathematical problems where exact solutions are difficult or impossible to obtain. These techniques are essential in various fields such as engineering, physics, economics, and many branches of science where complex mathematical models need to be solved efficiently. The general form of initial value problem (IVP) for ordinary differential equation, where the time interval $[t_0\,, t_{final}]$:

$$\boldsymbol{y}' = \boldsymbol{f}\,(\boldsymbol{t}, \boldsymbol{y}),$$

$$\boldsymbol{y}(\boldsymbol{t_0}\,) = \boldsymbol{y_0}\,.$$

2.1 Euler's Method

The Euler method is one of the simplest and most straightforward numerical methods used to approximate solutions to ordinary differential equations (ODEs). It's particularly useful when analytical solutions are difficult to derive or when dealing with complex systems where direct integration is impractical.

Given an initial value problem (IVP) for a first-order ODE, where the time interval $[t_0, t_{final}]$ is divided into small steps of size *h*. So, we have $t_n = t_0 + n\,h\;for\;n = 0,1,2, \dots N$ and $t_N = t_f$

$$\boldsymbol{y_{n+1}} = \boldsymbol{y_n} + \boldsymbol{h}\,\boldsymbol{f}\,(\boldsymbol{t_n}\,, \boldsymbol{y_n})$$

Accuracy: The accuracy of the Euler method depends on the step size $\boldsymbol{h}$. Smaller $\boldsymbol{h}$ generally leads to more accurate results, but increases computational effort.

2.2 Runge-Kutta 4

method is a widely used numerical method for solving ordinary differential equations (ODEs). It's an improvement over the other methods and provides more accurate results by considering multiple intermediate steps within each interval. Here's how the RK4 method works:

Given an initial value problem (IVP) for a first-order ODE , where the time interval $[t_0, t_{final}]$ is divided into small steps of size h. So, we have $t_n = t_0 + n\,h$ for $n = 0,1,2,\dots N$ and $t_N = t_f$

$$\boldsymbol{y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4),}$$

$$\boldsymbol{k_1 = h\,f\,(t_n, y_n),}$$

$$\boldsymbol{k_2 = h\,f\left(t_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right),}$$

$$\boldsymbol{k_3 = h\,f\left(t_n + \frac{h}{2}, y_n + \frac{k_2}{2}\right),}$$

$$\boldsymbol{k_4 = h\,f\,(t_n + h, y_n + k_3).}$$

Here:

• k_1 is the increment based on the slope at the beginning of the interval.

• k_2 and k_3 are increments based on the slopes at the midpoint of the interval using k_1 and k_2 .

• k_4 is the increment based on the slope at the end of the interval

• y_{n+1} is the weighted average of these increments, adjusting for the curvature of the function $f(t_n, y_n)$.

Accuracy: The RK4 method is a fourth-order method, meaning that the global error (error accumulated over many steps) is typically $O(h^4)$ where h is the step size.

Advantages: RK4 strikes a good balance between accuracy and computational efficiency. It's widely used because it's relatively simple to implement yet significantly improves accuracy over simpler methods like Euler's method.

2.3 Forward-Backward Sweep Method:

The Forward-Backward Sweep Method is a numerical approach for solving boundary value problems (BVPs), particularly in optimal control theory. It addresses systems governed by ordinary differential equations (ODEs) where solutions must satisfy both initial and terminal conditions. Below is a structured overview of the method, its equations, and implementation steps.

Optimality System

For an optimal control problem, the solution must satisfy the following system:

State Equation (forward in time):

$$\boldsymbol{y}'(\boldsymbol{t}) = \boldsymbol{g}\left(\boldsymbol{t}, \boldsymbol{y}(\boldsymbol{t}), \boldsymbol{u}(\boldsymbol{t})\right), \quad \boldsymbol{y}(\boldsymbol{t_0}) = \boldsymbol{a}$$

Adjoint Equation (backward in time):

$$\boldsymbol{\lambda}'(\boldsymbol{t}) = -\frac{\partial \boldsymbol{H}}{\partial \boldsymbol{y}} = -\left(\boldsymbol{f}_y(\boldsymbol{t}, \boldsymbol{y}, \boldsymbol{u}) + \boldsymbol{\lambda}(\boldsymbol{t})\boldsymbol{g}_y(\boldsymbol{t}, \boldsymbol{y}, \boldsymbol{u})\right),$$

$$\boldsymbol{\lambda}(\boldsymbol{t_1}) = \boldsymbol{0}$$

Optimality Condition:

$$\boldsymbol{0} = \frac{\partial \boldsymbol{H}}{\partial \boldsymbol{u}} = \boldsymbol{f}_u(\boldsymbol{t}, \boldsymbol{y}, \boldsymbol{u}) + \boldsymbol{\lambda}(\boldsymbol{t})\boldsymbol{g}_u(\boldsymbol{t}, \boldsymbol{y}, \boldsymbol{u}) \ \boldsymbol{at} \ \boldsymbol{u}^*$$

Here, u^* is the optimal control derived by solving this condition. Substituting u^* into the state and adjoint equations creates a two-point boundary value problem (TPBVP).

Method Overview

The Forward-Backward Sweep Method iteratively solves the optimality system using:

Forward integration for the state $y(t)$.

Backward integration for the adjoint $\lambda(t)$.

Control updates based on the optimality condition.

Steps:

STEP 1: Initial Guess: Start with an initial guess for the control $u(t)$ over the time interval $[t_0, t_1]$.

STEP 2: Forward Sweep:

Solve the state equation $y(t)$ forward in time using the current control $u(t)$.

Use numerical methods (e.g., Runge-Kutta 4) with initial condition $y(t_0) = a$.

STEP 3: Backward Sweep:

Solve the adjoint equation $\lambda(t)$ backward in time using the transversality condition $\lambda(t_1) = 0$.

Numerical backward integration requires interpolating $y(t)$ (e.g., cubic splines).

STEP 4: Control Update:

Compute u^* using the optimality condition $0 = f + \lambda g$.

Substitute updated $y(t)$ and $\lambda(t)$ to refine $u(t)$.

STEP 5: Convergence Check:

Terminate if the relative error between successive control iterates is small:

$$\frac{\| \boldsymbol{u} - \boldsymbol{uold} \|}{\| \boldsymbol{u} \|} \leq \boldsymbol{\delta}\ (\boldsymbol{\delta} = \boldsymbol{tolerance}).$$

If not converged, repeat Steps 2–4.

Runge-Kutta 4 (RK4) Implementation

Forward Integration for State $\boldsymbol{y}(\boldsymbol{t})$:

Given $\mathrm{y}'(\mathrm{t}) = \mathrm{g}(\mathrm{t}, \mathrm{y}, \mathrm{u})$:

$$\boldsymbol{x(t+h) \approx x(t) + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)},$$

where:

$$\boldsymbol{k_1 = f\big(t, x(t)\big)},$$

$$\boldsymbol{k_2 = f\left(t + \frac{h}{2}, x(t) + \frac{h}{2}k_1,\right)},$$

$$\boldsymbol{k_3 = f\left(t + \frac{h}{2}, x(t) + \frac{h}{2}k_2\right)},$$

$$\boldsymbol{k_4 = f(t + h, x(t) + hk_3)}.$$

Backward Integration for Adjoint $\boldsymbol{\lambda(t)}$:

Given $\lambda'(\mathrm{t}) = -\frac{\partial \mathrm{H}}{\partial \mathrm{y}}$ **:**

$$\boldsymbol{\lambda(t) \approx \lambda(t+h) - \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)},$$

where:

$$\boldsymbol{k_1 = f\,\big(t, \lambda(t)\big)},$$

$$\boldsymbol{k_2 = f\,\left(t - \frac{h}{2}, \lambda(t) - \frac{h}{2}k_1\right)},$$

$$\boldsymbol{k_3 = f\,\left(t - \frac{h}{2}, \lambda(t) - \frac{h}{2}k_2\right)},$$

$$\boldsymbol{k_4 = f\,(t - h, \lambda(t) - hk_3)}.$$

Key Considerations

Interpolation: Use cubic splines to approximate $y(t)$ during the backward sweep.

Convergence: The method converges if successive control estimates stabilize (see Figure 3.1 for iterative convergence).

Error: The RK4 method has a local truncation error of $O(h^4)$.

Applications

This method is widely used in optimal control problems, such as:

Epidemiology (e.g., optimizing vaccination strategies).

Engineering (e.g., trajectory optimization).

Economics (e.g., resource allocation over time).

By iterating between forward and backward sweeps, the method efficiently balances accuracy and computational cost, making it a robust tool for solving complex optimal control systems.

2.4 Case Studies

$$\min_{u} \int_0^1 A\,x(t) - B\,u(t)^2\;dt$$

Subject to

$$x'(t) = -\frac{1}{2}x(t)^2 + C\,u(t),$$

$$x(0) = x_0 > -2, \qquad A \geq 0, \qquad B > 0.$$

the Hamiltonian is

$$H = A\,x - B\,u^2 - \frac{1}{2}\lambda x^2 + C\,\lambda\,u.$$

Using the optimality condition,

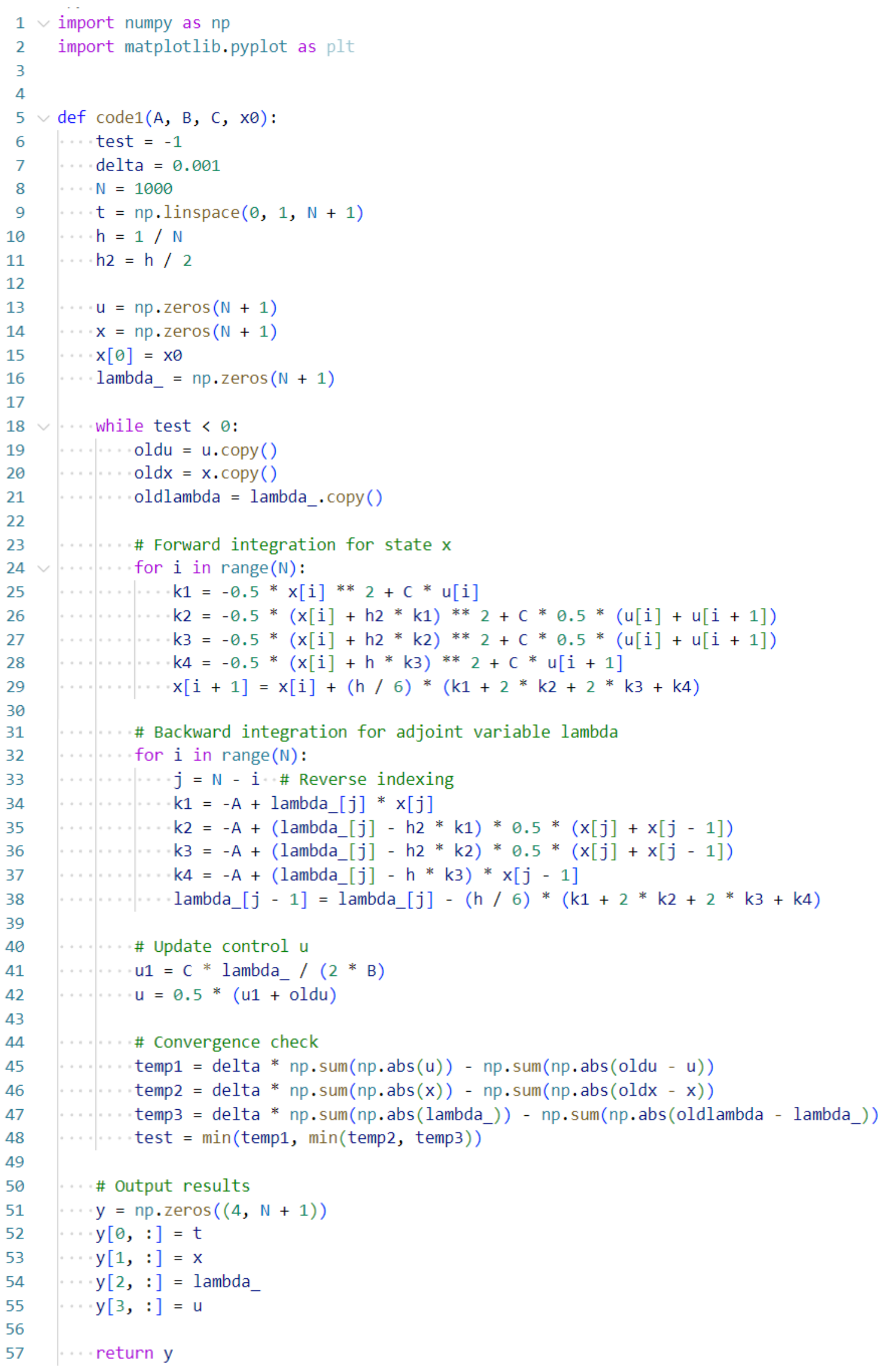
$$\mathbf{0} = \frac{\partial \boldsymbol{H}}{\partial \boldsymbol{u}} = -\mathbf{2}\boldsymbol{B}\boldsymbol{u} + \boldsymbol{C}\boldsymbol{\lambda} \Rightarrow \boldsymbol{u}^* = \frac{\boldsymbol{C}\boldsymbol{\lambda}}{\mathbf{2}\boldsymbol{B}}$$

We also easily calculate the adjoint equation to find

$$\boldsymbol{x}'(\boldsymbol{t}) = -\frac{\mathbf{1}}{\mathbf{2}}\boldsymbol{x}^2 + \boldsymbol{C}\boldsymbol{u}, \qquad \boldsymbol{x}(\mathbf{0}) = \boldsymbol{x}\mathbf{0}$$

$$\boldsymbol{\lambda}'(\boldsymbol{t}) = -\boldsymbol{A} + \boldsymbol{x}\boldsymbol{\lambda}, \qquad \boldsymbol{\lambda}(\mathbf{1}) = \mathbf{0}.$$

Code 1.py



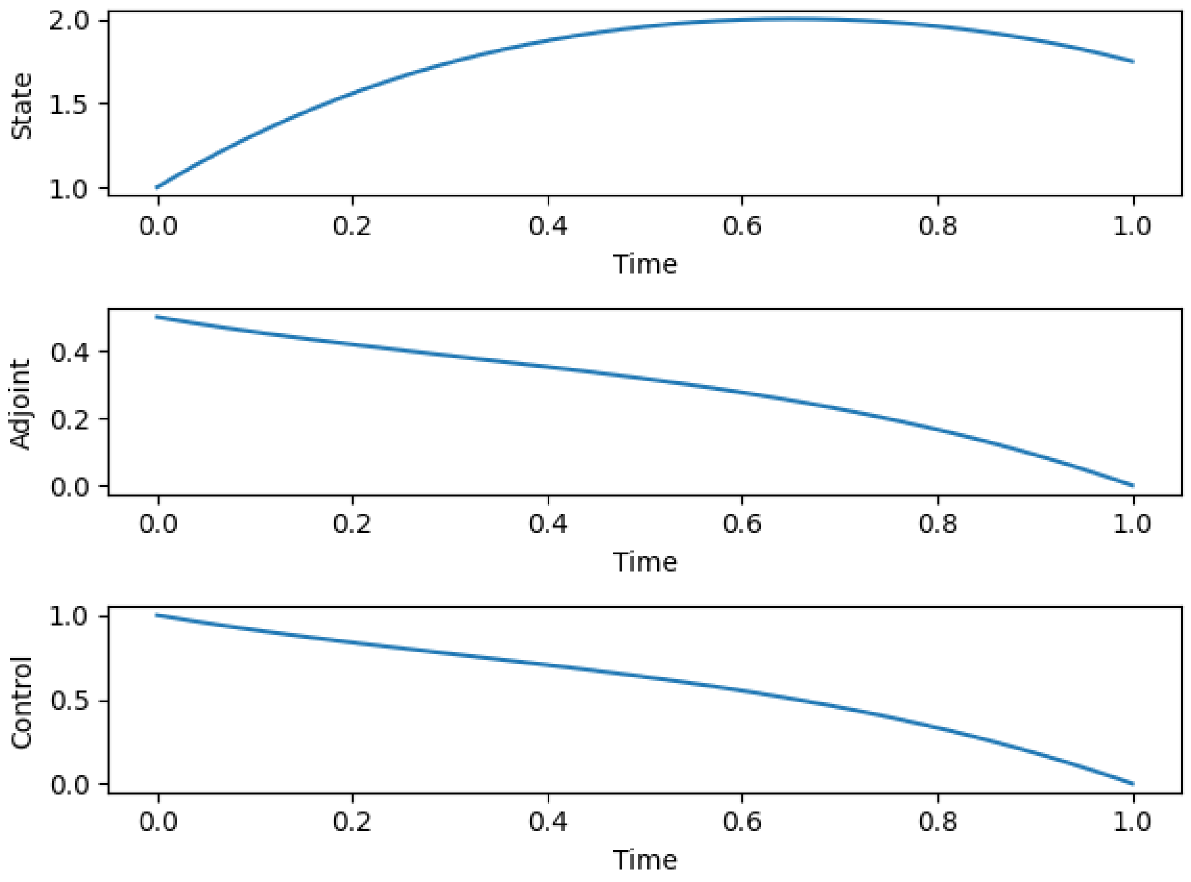
```python
import numpy as np
import matplotlib.pyplot as plt


def code1(A, B, C, x0):
    test = -1
    delta = 0.001
    N = 1000
    t = np.linspace(0, 1, N + 1)
    h = 1 / N
    h2 = h / 2

    u = np.zeros(N + 1)
    x = np.zeros(N + 1)
    x[0] = x0
    lambda_ = np.zeros(N + 1)

    while test < 0:
        oldu = u.copy()
        oldx = x.copy()
        oldlambda = lambda_.copy()

        # Forward integration for state x
        for i in range(N):
            k1 = -0.5 * x[i] ** 2 + C * u[i]
            k2 = -0.5 * (x[i] + h2 * k1) ** 2 + C * 0.5 * (u[i] + u[i + 1])
            k3 = -0.5 * (x[i] + h2 * k2) ** 2 + C * 0.5 * (u[i] + u[i + 1])
            k4 = -0.5 * (x[i] + h * k3) ** 2 + C * u[i + 1]
            x[i + 1] = x[i] + (h / 6) * (k1 + 2 * k2 + 2 * k3 + k4)

        # Backward integration for adjoint variable lambda
        for i in range(N):
            j = N - i  # Reverse indexing
            k1 = -A + lambda_[j] * x[j]
            k2 = -A + (lambda_[j] - h2 * k1) * 0.5 * (x[j] + x[j - 1])
            k3 = -A + (lambda_[j] - h2 * k2) * 0.5 * (x[j] + x[j - 1])
            k4 = -A + (lambda_[j] - h * k3) * x[j - 1]
            lambda_[j - 1] = lambda_[j] - (h / 6) * (k1 + 2 * k2 + 2 * k3 + k4)

        # Update control u
        u1 = C * lambda_ / (2 * B)
        u = 0.5 * (u1 + oldu)

        # Convergence check
        temp1 = delta * np.sum(np.abs(u)) - np.sum(np.abs(oldu - u))
        temp2 = delta * np.sum(np.abs(x)) - np.sum(np.abs(oldx - x))
        temp3 = delta * np.sum(np.abs(lambda_)) - np.sum(np.abs(oldlambda - lambda_))
        test = min(temp1, min(temp2, temp3))

    # Output results
    y = np.zeros((4, N + 1))
    y[0, :] = t
    y[1, :] = x
    y[2, :] = lambda_
    y[3, :] = u

    return y
```

This problem will utilize the PYTHON code developed in order to solve the optimal control problem.

This lab will focus on using the program to characterize the optimal control and resulting state and to ascertain how each parameter affects the solution. First, let us consider the goal of the problem. On one hand, we want to use the control u to maximize the integral of x. On the other hand, we also want to maximize the negative squared value of u. This, of course, is equivalent to minimizing the squared value of u. Thus, we must find the right balance of increasing x and keeping u as small as possible.

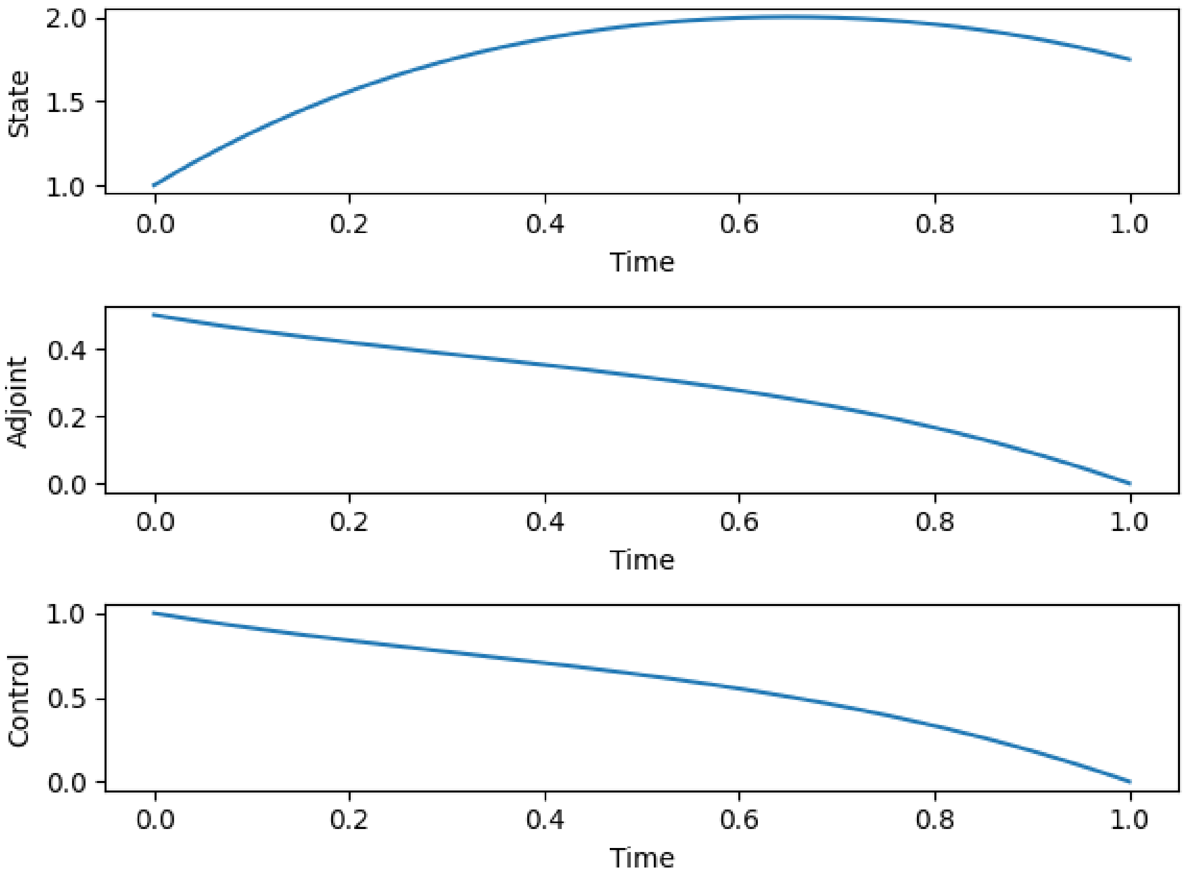
Assume

$$\boxed{A = 1, \quad B = 1, \quad C = 4, \quad x_0 = 1 .}$$



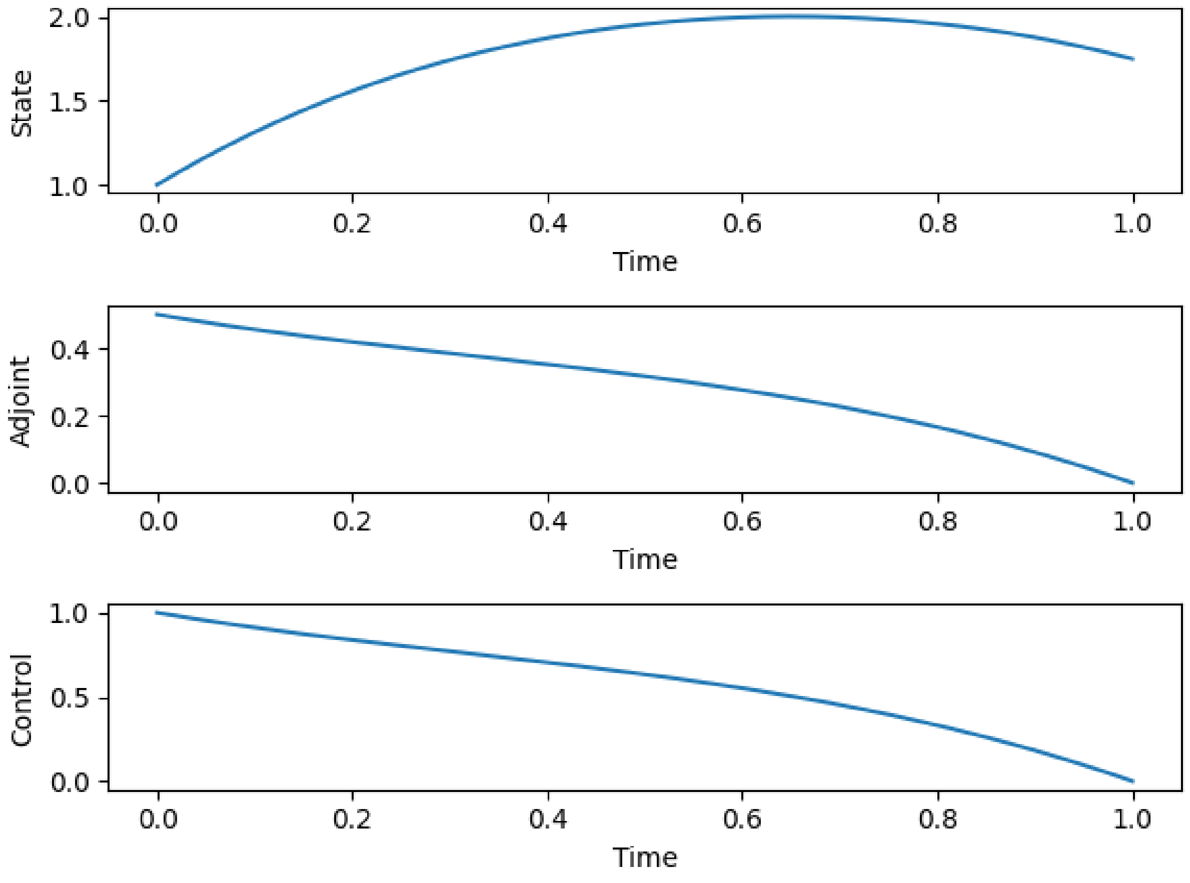
Assume

$$\boxed{A = 1, \quad B = 2, \quad C = 4, \quad x_0 = 1.}$$



Assume

$$\boldsymbol{A = 2, \quad B = 4, \quad C = 4, \quad x_0 = 1.}$$



Chapter 3

CHAPTER 3

Epidemic Model

We develop an optimal vaccination strategy for an epidemic using an SEIRN model (Susceptible-Exposed-Infectious-Recovered-Population dynamics). The goal is to minimize both the number of infectious individuals and vaccination costs over a fixed time period.

3.1 SEIRN Model Dynamics

The population is divided into four compartments:

$\boldsymbol{S(t)}$	Susceptible individuals
$\boldsymbol{E(t)}$	Exposed (latent) individuals
$\boldsymbol{I(t)}$	Infectious individuals
$\boldsymbol{R(t)}$	Recovered/immune individuals

Total Population:

$$\boldsymbol{N(t) = S(t) + E(t) + I(t) + R(t)}$$

Differential Equations:

$$\boldsymbol{S'(t) = bN(t) - dS(t) - cS(t)I(t),}$$

$$\boldsymbol{E'(t) = cS(t)I(t) - (e + d)E(t),}$$

$$\boldsymbol{I'(t) = eE(t) - (g + a + d)I(t),}$$

$$\boldsymbol{R'(t) = gI(t) - dR(t),}$$

$$\boldsymbol{N'(t) = (b - d)N(t) - aI(t).}$$

Parameters:

$\boldsymbol{b}$	Birth date
$\boldsymbol{d}$	Natural death rate
$\boldsymbol{c}$	Disease transmission rate
$\boldsymbol{e}$	Rate of progression from exposed to infectious ($\frac{1}{e} = \textit{mean latent period}$)
$\boldsymbol{g}$	Recovery rate ($\frac{1}{g} = \textit{mean infectious period}$)
$\boldsymbol{a}$	Disease-induced death rate

3.2 Optimal Control Problem

Objective: Minimize the cost functional:

$$\min_{u} \int_0^T [AI(t) + u^2(t)]dt$$

Subject to:

$$S'(t) = bN(t) - dS(t) - cS(t)I(t) - u(t)S(t), \qquad S(0) = S_0 \geq 0,$$

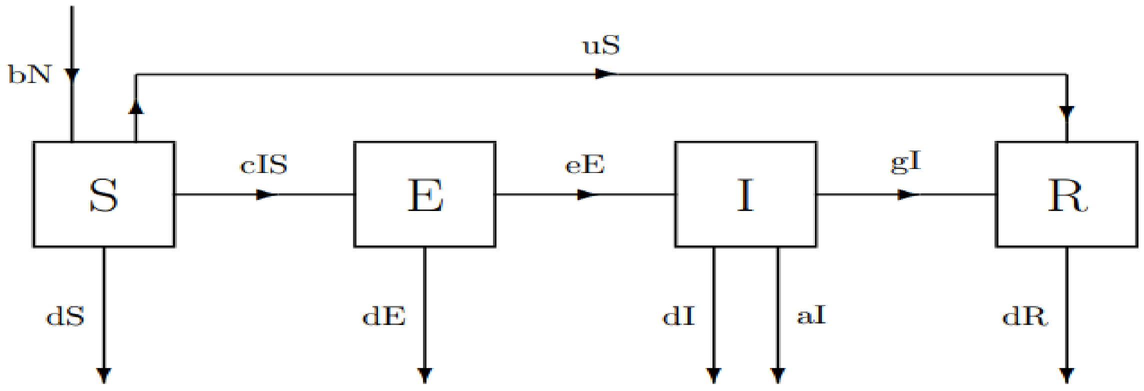
$$E'(t) = cS(t)I(t) - (e+d)E(t), \qquad E(0) = E_0 \geq 0,$$

$$I'(t) = eE(t) - (g+a+d)I(t), \qquad I(0) = I_0 \geq 0,$$

$$R'(t) = gI(t) - dR(t) + u(t)S(t), \qquad R(0) = R_0 \geq 0,$$

$$N'(t) = (b-d)N(t) - aI(t), \qquad N(0) = N_0,$$

$$Bounds\text{: } 0 \leq u(t) \leq 0.9.$$



3.2.1 Hamiltonian and Adjoint Equations

Hamiltonian:

$$H = AI(t) + u^2(t) + \lambda_1 S'(t) + \lambda_2 E'(t) + \lambda_3 I'(t) + \lambda_4 R'(t) + \lambda_5 N'(t)$$

Then:

$$H \ = \ A\, I(t) + \ u^2(t) + \ \lambda_1\big(bN(t) – \ dS(t) – \ cS(t)I(t) – \ u(t)S(t)\big)$$

$$+\lambda_2\big(cS(t)I(t) - \ (e \ + \ d)E(t)\big) + \ \lambda_3\big(eE(t) - \ (g \ + \ a \ + \ d)I(t)\big)$$

$$+\lambda_4\big(gI(t) - \ dR(t) + \ u(t)S(t)\big) + \ \lambda_5\big((b \ - \ d)N(t) - \ aI(t)\big).$$

Adjoint Equations (Backward in Time):

$$\lambda_{1'} = -\frac{\partial H}{\partial S} = \lambda_1(d + cI + u) - \lambda_2 cI - \lambda_4 u,$$

$$\lambda_2' = -\frac{\partial H}{\partial E} = \lambda_2(e + d) - \lambda_3 e,$$

$$\lambda_3' = -\frac{\partial H}{\partial I} = -A + (\lambda_1 - \lambda_2)cS + \lambda_3(g + a + d) - \lambda_4 g + \lambda_5 a,$$

$$\lambda_4' = -\frac{\partial H}{\partial R} = \lambda_4 d,$$

$$\lambda_5' = -\frac{\partial H}{\partial N} = -\lambda_1 b - \lambda_5(b - d).$$

Boundary Conditions:

$$\boldsymbol{\lambda_i(T) = 0 for\ i = 1, 2, 3, 4, 5.}$$

Optimal Vaccination Rate:

$$\boldsymbol{u^*(t) = \frac{\lambda_1 S(t) - \lambda_4 S(t)}{2}}$$

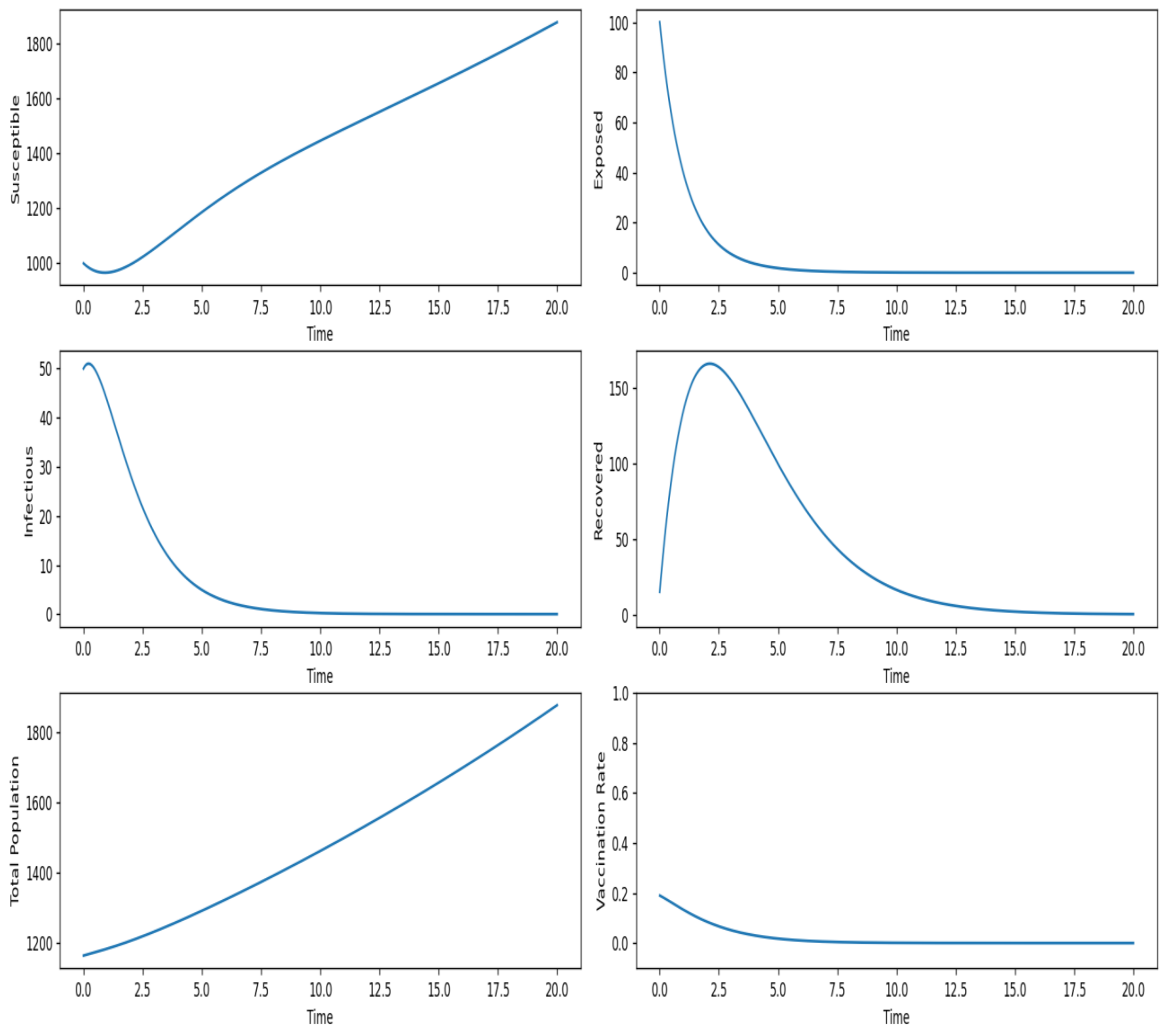
3.2.2 Case Studies

Assume

$$\boldsymbol{b} = 0.525 \quad \boldsymbol{d} = 0.5 \quad \boldsymbol{c} = 0.0001 \quad \boldsymbol{e} = 0.5$$

$$\boldsymbol{g} = 0.1 \quad \boldsymbol{a} = 0.2 \quad \boldsymbol{S_0} = 1000 \quad \boldsymbol{E_0} = 100$$

$$\boldsymbol{I_0} = 50 \quad \boldsymbol{R_0} = 15 \quad \boldsymbol{A} = 0.1 \quad \boldsymbol{T} = 20$$

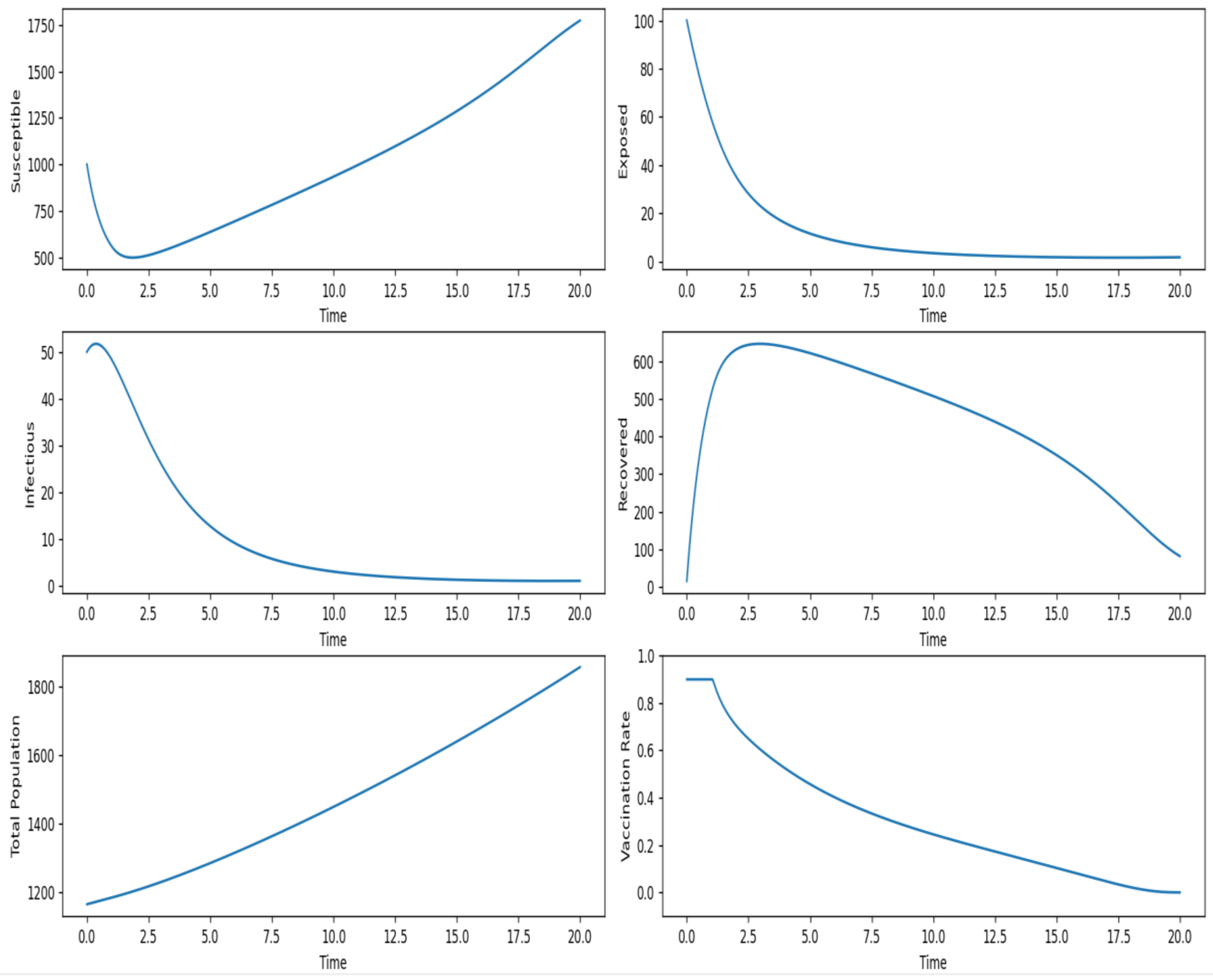


Assume

$$b = 0.525 \quad d = 0.5 \quad c = 0.001 \quad e = 0.5$$

$$g = 0.1 \quad a = 0.2 \quad S_0 = 1000 \quad E_0 = 100$$

$$I_0 = 50 \quad R_0 = 15 \quad A = 0.1 \quad T = 20$$

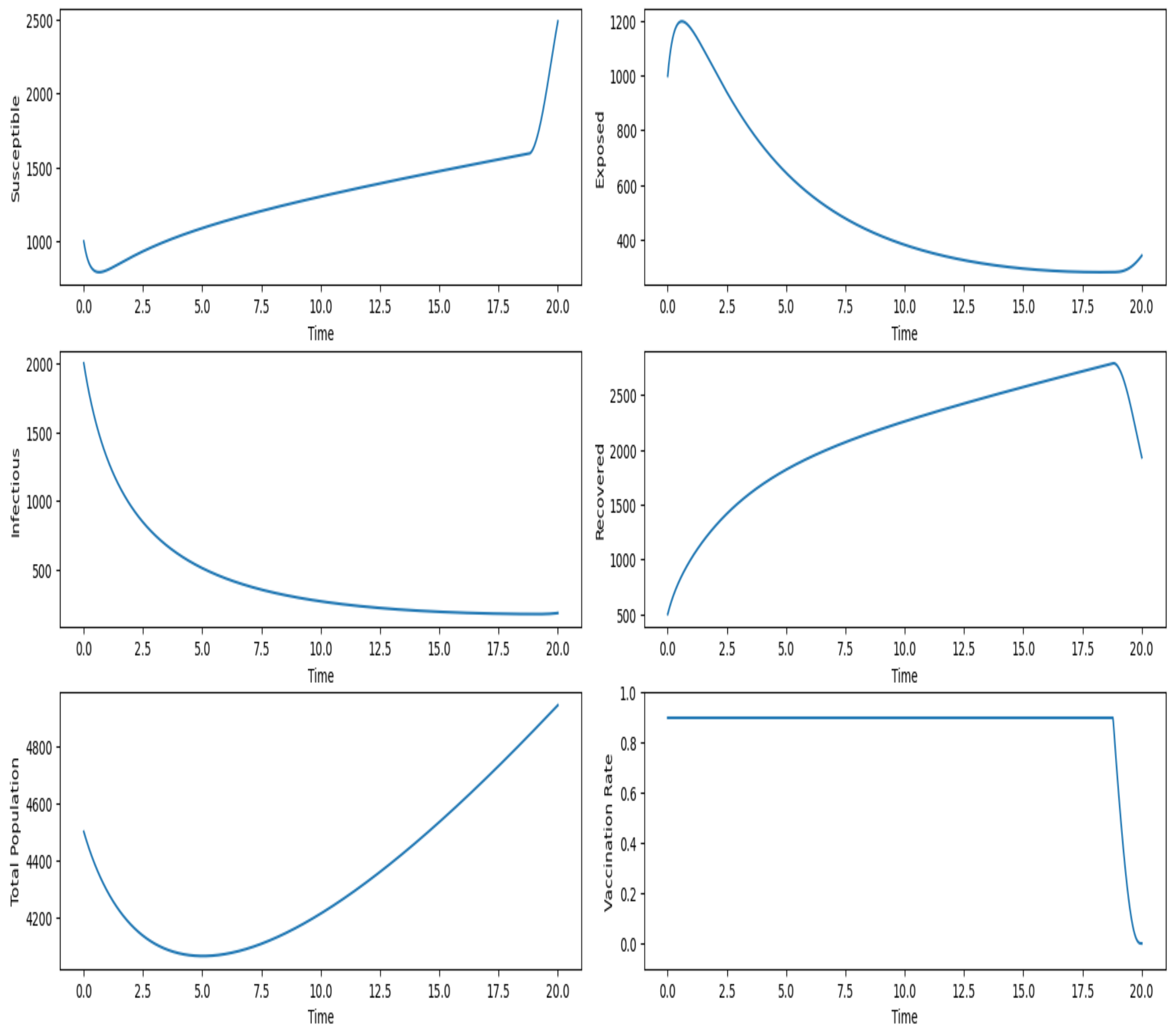


Assume

$$b = 0.525 \quad d = 0.5 \quad c = 0.001 \quad e = 0.5$$

$$g = 0.1 \quad a = 0.2 \quad S_0 = 1000 \quad E_0 = 1000$$

$$I_0 = 2000 \quad R_0 = 500 \quad A = 0.1 \quad T = 20$$

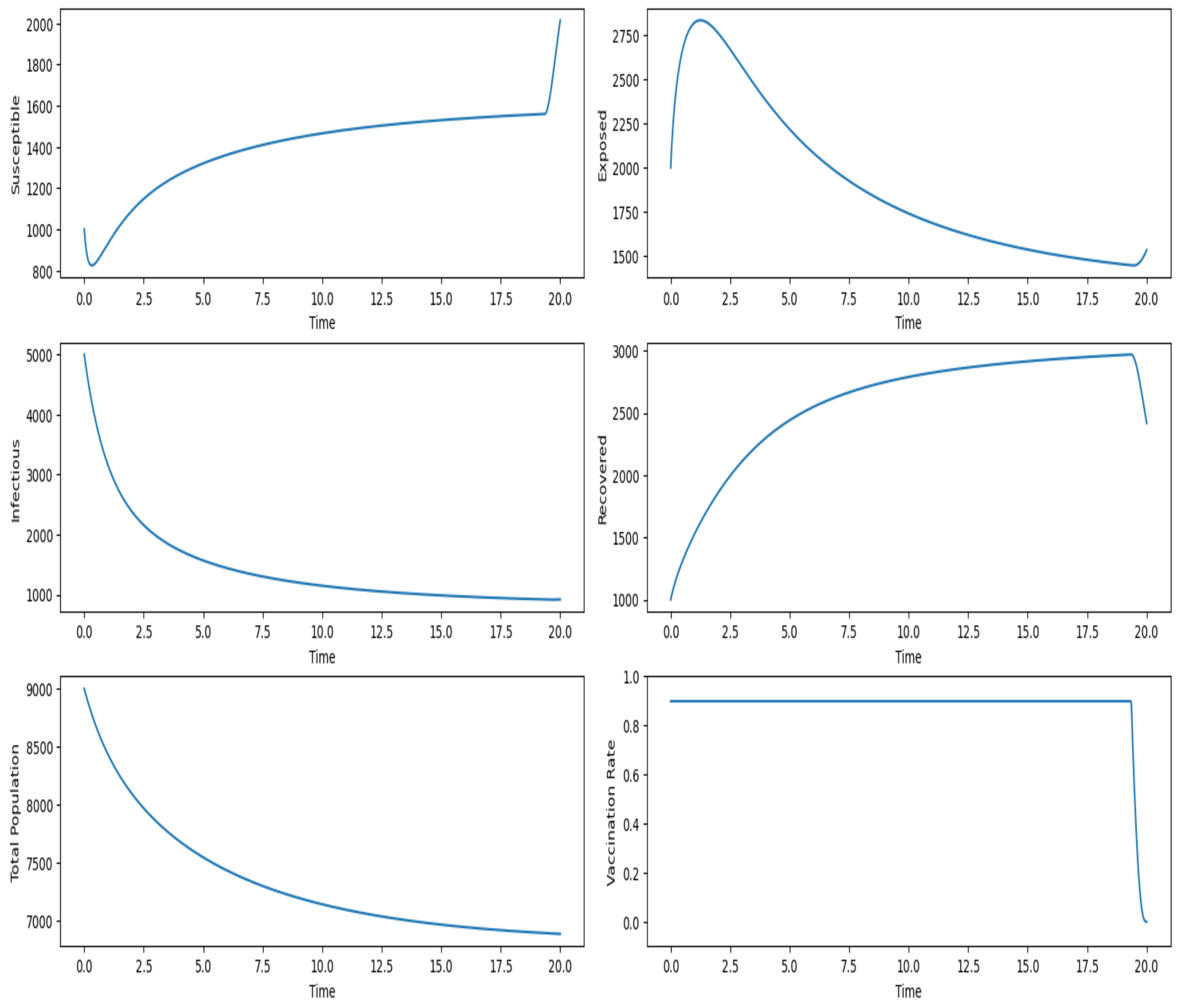


Assume

$$b = 0.525 \quad d = 0.5 \quad c = 0.001 \quad e = 0.5$$

$$g = 0.1 \quad a = 0.2 \quad S_0 = 1000 \quad E_0 = 2000$$

$$I_0 = 5000 \quad R_0 = 1000 \quad A = 0.1 \quad T = 20$$



Chapter 4

Chapter 4

Fractional Numerical Methods

4.1 Fractional calculus

4.1.1 Historical brief & applications field

Fractional calculus is a generalization of ordinary differentiation and integration to arbitrary (non-integer) order. The topics is as old as the differential calculus, and goes back to times when Leibniz and Newton invented differential calculus.

In a letter dated 30th September 1695, L'Hopital wrote to Leibniz asking him particular notation he has used in his publication for the n-th derivative of a function f

$$\frac{\boldsymbol{D^n f(x)}}{\boldsymbol{Dx^n}}$$

i.e. what would be the result if $n = \frac{1}{2}$.

Leibniz's response "An apparent paradox from which one-day useful consequence will be drawn" In this words Fractional Calculus was born. Many known mathematicians contributed to this theory over the years, among them Liouville Riemann, Caputo, Weyl, Fourier, Abel, Lacroix, Leibniz, Grunwald and Letnikov.

Fractional calculus is more than a century old as the conventional calculus, but not very popular among science and/or engineering community. The beauty of this subject is that fractional derivatives (and integrals) are not a local (or point) property (or quantity). Thereby this considers the history and non-local distributed effects.

Thereby this considers the history and non-local distributed effects. In other words, may be these topics translates the reality of nature better! Therefore, to make this subject available as popular subject to science and engineering community, it adds another dimension to understand or describe basic nature in a better way. Perhaps fractional calculus is what nature understands, and to talk with nature in this language is therefore efficient.

For past centuries, this subject was with mathematicians, and only in last few years, this was pulled to several (applied) fields of engineering and science and economics However, recent attempt is on to have the definition of fractional derivative as local operator specifically to fractal science theory. Next decade will see several applications based on this new subject, which can be thought of as superset of fractional differ integral calculus, the conventional integer order calculus being a part of it.

Fractional Calculus has recently been applied in various areas of engineering, science, finance, applied mathematics, and bio engineering. However, many researchers remain unaware of this field.

4.1.2 Some fractional order integrals and derivatives definitions, and their properties

Unfortunately, Euler's definition isn't enough to be the definition of fractional derivatives for analytical reasons (the polynomial functional space can't be span all functions we use).

The Riemann-Liouville fractional order integral and differential operators are fundamental tools in the field of fractional calculus, which extends the concepts of differentiation and integration to non-integer orders. The development of fractional calculus can be traced back to the works of mathematicians Augustin-Louis Cauchy and Liouville in the early 19th century. However, it was Bernhard Riemann and Joseph Liouville who made significant contributions to the study of fractional calculus in the mid-1800s.

Definition 1: The left-right side Riemann-Liouville's integrals with order $\alpha \geq 0$ of the function $f : (a, b) \to R$ are defined respectively as :

$$
\begin{aligned}
{}^{RL}_{a}\mathcal{I}^{\alpha}_{t} f(t) &= \frac{1}{\Gamma(\alpha)} \int_a^t \frac{f(s)}{(t-s)^{1-\alpha}} ds, \quad t \geq a \\
{}^{RL}_{t}\mathcal{I}^{\alpha}_{b} f(t) &= \frac{1}{\Gamma(\alpha)} \int_t^b \frac{f(s)}{(t-s)^{1-\alpha}} ds, \quad t \leq b
\end{aligned}
$$

Proposition: The Riemann-Liouville's integral operator in Definition 1 with order $\alpha \geq 0$ of a function $f(t)$ is a linear operator

$$
{}^{RL}\mathcal{I}^{\alpha}_{t}(\lambda f(t) + \mu g(t)) = \lambda \, {}^{RL}\mathcal{I}^{\alpha}_{t} f(t) + \mu \, {}^{RL}\mathcal{I}^{\alpha}_{t} g(t)
$$

Proposition: The repeated applying of the Riemann-Liouville's integral operator in Definition 1, is the same as composite the order, but not commutative

$$
{}^{RL}\mathcal{I}^{\alpha}_{t} \, {}^{RL}\mathcal{I}^{\beta}_{t} f(t) = {}^{RL}\mathcal{I}^{\alpha+\beta}_{t} f(t) \neq {}^{RL}\mathcal{I}^{\beta}_{t} \, {}^{RL}\mathcal{I}^{\alpha}_{t} f(t)
$$

Definition 2: The left-right side Riemann-Liouville's fractional derivatives with order $\alpha \geq 0$ of the function $f : (a,b) \rightarrow R$ are defined respectively as :

$$
\begin{aligned}
{}^{RL}_{a}\mathcal{D}^{\alpha}_{t}f(t) &= \frac{1}{\Gamma(n-\alpha)}\frac{d^n}{dt^n}\int_a^t \frac{f(s)}{(t-s)^{\alpha-n+1}}ds, \quad t \geq a \\
{}^{RL}_{t}\mathcal{D}^{\alpha}_{b}f(t) &= \frac{(-1)^n}{\Gamma(n-\alpha)}\frac{d^n}{dt^n}\int_t^b \frac{f(s)}{(t-s)^{\alpha-n+1}}ds, \quad t \leq b
\end{aligned}
$$

where $n = \lfloor \Re(\alpha) \rfloor + 1$, is a positive integer satisfying $n-1 \leq \alpha < n$ and $\Re(\alpha) \notin N_0$, if $\alpha = n \in N_0$ then the fractional Riemann-Liouville's derivatives are coincide with the classical n^{th} derivatives.

$$
\begin{aligned}
{}^{RL}_{a}\mathcal{D}^{\alpha}_{t}f(t) &= f^{(n)}(t) \\
{}^{RL}_{t}\mathcal{D}^{\alpha}_{b}f(t) &= (-1)^n f^{(n)}(t)
\end{aligned}
$$

Proposition: The repeated applying of the Riemann-Liouville's integral operator in Definition 1, then applying Riemann-Liouville's differential operator in Definition 2 with same order is the same to do nothing,

$$
{}^{RL}\mathcal{I}^{\alpha}_{t}\, {}^{RL}\mathcal{D}^{\alpha}_{t}f(t) = {}^{RL}\mathcal{D}^{\alpha}_{t}\, {}^{RL}\mathcal{I}^{\alpha}_{t}f(t) = f(t)
$$

Grünwald-Letnikov's formula is a widely used numerical method for approximating fractional derivatives. It provides a way to compute fractional derivatives of a function based on its values at equally spaced points. The formula is named after the mathematicians Carl Grünwald and Emil Letnikov, who independently derived this approach.

Definition 3: The left-right side Grünwald-Letinkov's formula for the fractional derivatives of order $\alpha \geq 0$ of the function $f: (a, b) \rightarrow R$ are defined respectively as:

$$
\begin{aligned}
{}^{GL}_{a}\mathcal{D}^{\alpha}_{t} f(t) &= \lim_{h \to 0} \frac{1}{h^{\alpha}} \sum_{j=0}^{\lfloor \frac{t-a}{h} \rfloor} w^{\alpha}_{j} f(t - jh) \\
{}^{GL}_{t}\mathcal{D}^{\alpha}_{b} f(t) &= \lim_{h \to 0} \frac{1}{h^{\alpha}} \sum_{j=0}^{\lfloor \frac{b-t}{h} \rfloor} w^{\alpha}_{j} f(t + jh)
\end{aligned}
$$

Where w_j^{α} are the normalized Grünwald's weight defined by

$$
w^{\alpha}_{j} = (-1)^{j} \binom{\alpha}{j}
$$

Definition 4: The left-right side Caputo derivatives with order $\alpha \geq 0$ of the function $f : (a, b) \rightarrow R$ are defined respectively as:

$$
\begin{aligned}
{}^{C}_{a}\mathcal{D}^{\alpha}_{t} f(t) &= \frac{1}{\Gamma(n-\alpha)} \int_{a}^{t} \frac{f^{(n)}(s)}{(t-s)^{\alpha-n+1}} ds, \quad t \geq a \\
{}^{C}_{t}\mathcal{D}^{\alpha}_{b} f(t) &= \frac{1}{\Gamma(n-\alpha)} \int_{t}^{b} \frac{f^{(n)}(s)}{(t-s)^{\alpha-n+1}} ds, \quad t \leq b
\end{aligned}
$$

where $n = \lfloor \Re(\alpha) \rfloor + 1$, is a positive integer satisfying $n - 1 \leq \alpha < n$ and $\Re(\alpha) \notin N_0$, if $\alpha = n \in N_0$ then the fractional Caputo derivatives are coincide with the classical n^{th} derivatives.

$$
\begin{aligned}
{}^{C}_{a}\mathcal{D}^{\alpha}_{t} f(t) &= f^{(n)}(t) \\
{}^{C}_{t}\mathcal{D}^{\alpha}_{b} f(t) &= (-1)^{n} f^{(n)}(t)
\end{aligned}
$$

This definition of fractional derivative is mostly used for solving fractional differential equation. So, this definition is very important to know; beside that Riemann-Liouville fractional derivative has a number of problems.

Example: Use the RL derivatives & Caputo’s derivative in Definition 2 and 4 to compute the following

1. $f(t) = c, \alpha = 0.5$

2. $f(t) = t, \alpha = 0.5$

Solution:

1. Since $\alpha = 0.5$ then $n = 1$, so we apply RL derivative first

$$
\begin{aligned}
{}^{\mathrm{RL}}_{0}\mathcal{D}^{0.5}_{t} f(t) &= \frac{d}{dt}\frac{1}{\Gamma(1)}\int_0^t \frac{c}{(t-s)^{\frac{1}{2}}} ds \\
&= \frac{c}{\sqrt{\pi}} t^{-\frac{1}{2}}
\end{aligned}
$$

second Caputo’s derivative

$$
\begin{aligned}
{}^{\mathrm{C}}_{0}\mathcal{D}^{0.5}_{t} f(t) &= \frac{1}{\Gamma(1)}\int_0^t \frac{1}{(t-s)^{\frac{1}{2}}} \frac{d}{ds}(c)\, ds \\
&= 0
\end{aligned}
$$

We claim that ${}^{RL}_{0}D^{\alpha}_{t} c \neq {}^{C}_{0}D^{\alpha}_{t} c$

2. Since $\alpha = 0.5$ then $n = 1$, so we apply RL derivative first

$$
\begin{aligned}
{}^{\mathrm{RL}}_{0}\mathcal{D}^{0.5}_{t} f(t) &= \frac{d}{dt}\frac{1}{\Gamma(1)}\int_0^t \frac{s}{(t-s)^{\frac{1}{2}}} ds \\
&= \frac{2}{\sqrt{\pi}} t^{\frac{1}{2}}
\end{aligned}
$$

Second Caputo's derivative

$$\begin{aligned}{}_{0}^{C}\mathcal{D}_{t}^{0.5}f(t) &= \frac{1}{\Gamma(1)}\int_{0}^{t}\frac{1}{(t-s)^{\frac{1}{2}}}\frac{d}{ds}(s)\,ds\\ &= \frac{2}{\sqrt{\pi}}t^{\frac{1}{2}}\end{aligned}$$

We claim that ${}_{0}^{RL}D_{t}^{\alpha}t^{n} = {}_{0}^{C}D_{t}^{\alpha}t^{n}$ for $n \geq 1$

Example: A graph of various order left side fractional of a polynomial $f(t) = t2$ with same lower point $a = 0$,see Figure 1.1

Proposition: All the differential operators in Definition 2, 3, and 4 are all linear operators

$$\mathcal{D}_{t}^{\alpha}(\lambda f(t) + \mu g(t)) = \lambda\,\mathcal{D}_{t}^{\alpha}f(t) + \mu\,\mathcal{D}_{t}^{\alpha}g(t)$$

Proposition: The repeated applying of one of the differential operators in Definition 2, 3, or 4, is the same as composite their order, but not commutative

$$\mathcal{D}_{t}^{\alpha}\,\mathcal{D}_{t}^{\beta}f(t) = \mathcal{D}_{t}^{\alpha+\beta}f(t) \neq \mathcal{D}_{t}^{\beta}\,\mathcal{D}_{t}^{\alpha}f(t)$$

Proposition: Suppose that the function f is $(n-1)$-times continuously differentiable in the interval $[a,T]$ and that $f^{(n)}$ is integrable in $[a,T]$, then for every $\alpha(0 < \alpha < n)$ the Riemann-Liouville's fractional derivative exists and coincides with the Grünwald-Letnikov's fractional derivative, and if $0 \leq m-1 \leq \alpha < m \leq n$, then for $a \leq t \leq T$ both of them satisfies

$$\mathcal{D}_{t}^{\alpha}f(t) = \sum_{k=0}^{m-1}\frac{f^{(k)}(a)(t-a)^{k-\alpha}}{\Gamma(1+k-\alpha)} + \frac{1}{\Gamma(m-\alpha)}\int_{a}^{t}\frac{f^{(m)}(s)}{(t-s)^{1-m+\alpha}}ds$$

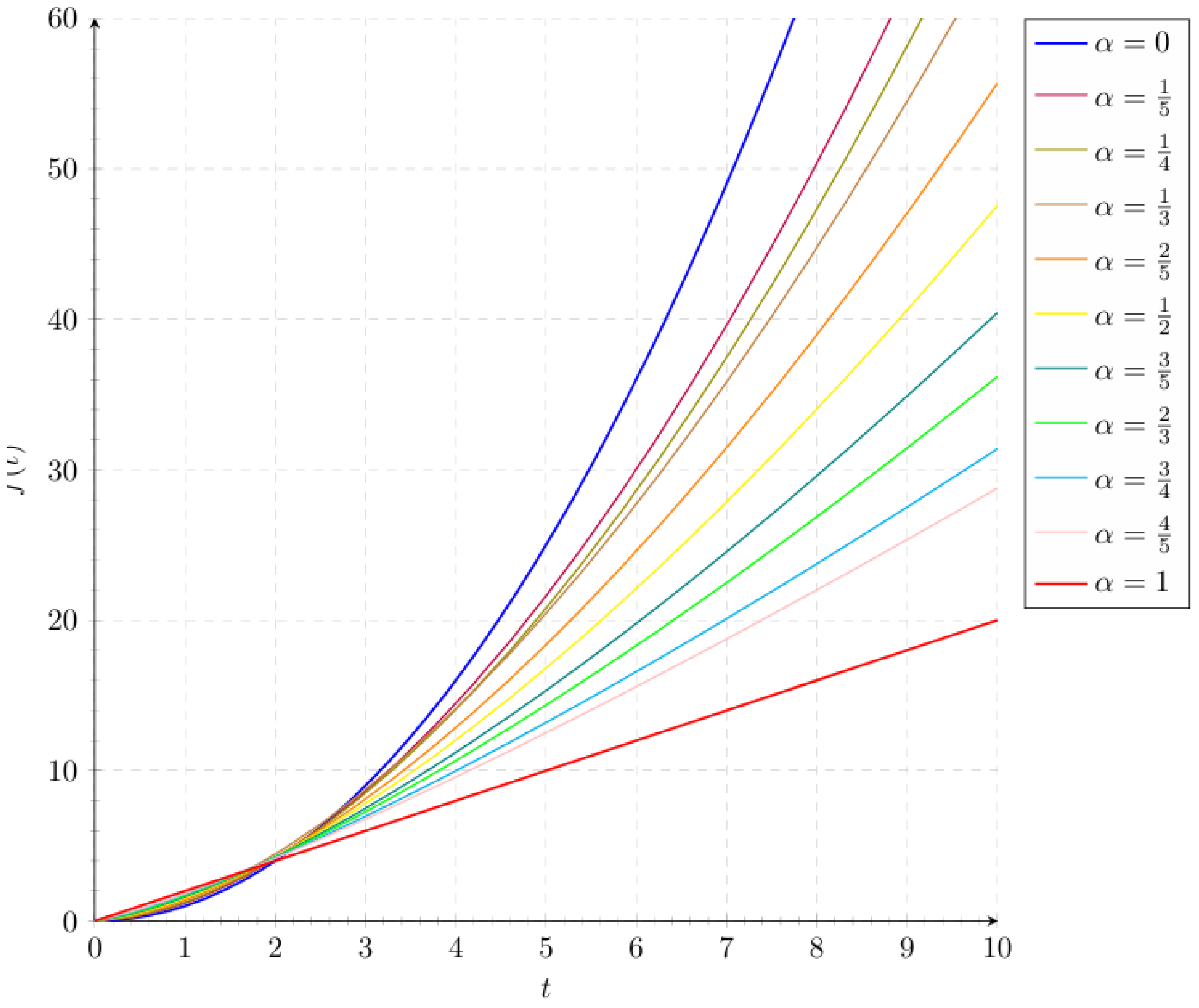


Figure 1.1: Various order α derivative for $f(t) = t^2$

Proposition: The left-right sided Riemann Liouville's fractional derivatives of a function $\boldsymbol{f(t)}$ and the the left-right sided Caputo's fractional derivatives of same $\boldsymbol{f}$ are gathered by the following equations

$$
{}^{RL}_{\ a}\mathcal{D}^{\alpha}_{t} f(t) = \sum_{k=0}^{m-1} \frac{f^{(k)}(a)(t-a)^{k-\alpha}}{\Gamma(1+k-\alpha)} + {}^{C}_{a}\mathcal{D}^{\alpha}_{t} f(t)
$$

$$
{}^{RL}_{\ t}\mathcal{D}^{\alpha}_{b} f(t) = \sum_{k=0}^{m-1} \frac{f^{(k)}(a)(b-t)^{k-\alpha}}{\Gamma(1+k-\alpha)} + {}^{C}_{t}\mathcal{D}^{\alpha}_{b} f(t)
$$

Proposition: if the function f and all of its derivatives are equal to zero at the left or the right point, $\boldsymbol{f^{(k)}(a) = 0}$, $\forall \boldsymbol{k = 0, 1, \dots, n-1}$ or $\boldsymbol{f^{(k)}(b) = 0}$, $\forall \boldsymbol{k = 0, 1, \dots, n-1}$, then

$$ {}^{RL}_{\ a}\mathcal{D}^{\alpha}_{t} f(t) = {}^{C}_{a}\mathcal{D}^{\alpha}_{t} f(t) $$
$$ {}^{RL}_{\ t}\mathcal{D}^{\alpha}_{b} f(t) = {}^{C}_{t}\mathcal{D}^{\alpha}_{b} f(t) $$

4.2 Fractional Differential Equations

Fractional differential equations are generalizations of ordinary differential equations to an arbitrary (non-integer) order. Fractional differential equations have attracted considerable interest because of their ability to model complex phenomena. These equations capture nonlocal relations in space and time with power-law memory kernels. Due to the extensive applications of OFDEs in engineering and science, research in this area has grown significantly all around the world.

4.2.1 Definitions

Definition 5 (Riemann–Liouville FDE): Let be the fractional differential equation (FDE)

$$ \mathcal{D}^{\alpha}_{a_+} y(t) = f(t, y), \quad t > a, \alpha > 0 $$

with the conditions:

$$ \mathcal{D}^{\alpha-k}_{a_+} y(a+) = b_k, \quad k = 1, \dots, n $$

Definition 6 (Caputo FDE): Let be the fractional differential equation (FDE)

$$ \mathcal{D}^{\alpha}_{a_+} y(t) = f(t, y), \quad t > a, \alpha > 0 $$

with the conditions:

$$ \mathcal{D}^{k}_{a_+} y(0) = b_k, \quad k = 1, \dots, n-1 $$

4.2.2 Existence & Uniqueness

Theorem (Existence and Uniqueness for the Caputo Problem): Let a Caputo FDE be

$$\mathcal{D}^{\alpha}y(t) = f(t,y), \quad 0 < \alpha \leq 1, \quad t > 0$$

With initial conditions:

$$y(0) = y_0$$

We consider the domain

$$D = [0,\eta] \times [y_0 - \eta, y_0 + \eta]$$

on which f satisfies:

- $\boldsymbol{f(t,y)}$ is continuous
- $|\boldsymbol{f(t,y)}| < \boldsymbol{M}$, where $\boldsymbol{M} = \mathbf{max}(\boldsymbol{t,y}) \in \boldsymbol{D}|\boldsymbol{f(t,y)}|$
- $\boldsymbol{f(t,y)}$ satisfy in $\boldsymbol{D}$ the Lipschitz condition in $\boldsymbol{y}$ if there is a constant $\boldsymbol{K}$ such that:

$$|f(t,y_2) - f(t,y_1)| < K|y_2 - y_1|$$

Then it exists δ > 0 and a function y(t) ∈ C [0, η] unique for

$$\delta = \min\left\{\eta, \left(\frac{\eta\Gamma(\alpha+1)}{M}\right)^{1/\alpha}\right\}$$

4.2.3 Applications of FDEs

Like ODEs, the fractional differential equations have wide field of applications, just generalizing the ODEs problem into fractional derivatives definitions generate enormous opportunities in modeling and simulation area also in applied statistics. Here we list some application of FDEs

Modeling anomalous diffusion: Fractional differential equations can be used to model diffusion processes that exhibit anomalous behavior, such as subdiffusion and superdiffusion. This is because fractional derivatives can capture the long-range memory effects that are often present in anomalous diffusion processes.

Controlling dynamical systems: Fractional differential equations can be used to design controllers for dynamical systems that exhibit memory effects. This is because fractional derivatives can capture the non-local interactions that are often present in such systems.

Modeling viscoelastic materials: Fractional differential equations can be used to model the behavior of viscoelastic materials, such as polymers and rubber. This is because fractional derivatives can capture the time-dependent behavior of these materials.

Modeling biological systems: Fractional differential equations can be used to model the behavior of biological systems, such as the immune system and the heart. This is because fractional derivatives can capture the memory effects and non-local interactions that are often present in biological systems.

Signal processing: Fractional differential equations can be used to process signals that exhibit long-range memory effects. This is because fractional derivatives can capture the long-term trends in such signals.

These are just a few of the many applications of fractional differential equations. As research in this area continues, we can expect to see even more applications of fractional differential equations in the future.

4.3 Numerical Methods for FDEs

4.3.1 Euler's Method

For the following problem

$$\boldsymbol{D_x^\alpha y(x) = f(x,y)}$$

$$\boldsymbol{0 < \alpha \leq 1, \qquad y(x0) = y0}$$

where $\boldsymbol{\alpha}$ is the order of the equation. The solution can be determined by one of the following

Euler's forward method

$[D_x^{-\alpha} f(x,y)]_{x=x_{n+1}}$ is approximated by the left fractional rectangular formula

$$y_{n+1} = y_0 + \sum_{j=0}^{n} b_{j,n+1} f(x_j, y_j)$$

which $b_{j,n+1}$ is defined as

$$\boldsymbol{b_{j,n+1}} = \frac{\boldsymbol{h^{\alpha}}}{\boldsymbol{\Gamma(\alpha+1)}}\left[(\mathbf{n}-\mathbf{j}+\mathbf{1})^{\alpha}-(\mathbf{n}-\mathbf{j})^{\alpha}\right]$$

Euler's backward method

$[D_x^{-\alpha} f(x,y)]_{\mathrm{x=x_{n+1}}}$ is approximated by the right fractional rectangular formula

$$y_{n+1} = y_0 + \sum_{j=0}^{n} b_{j,n+1} f(x_{j+1}, y_{j+1})$$

4.3.2 Runge-kutta Method

suppose that all conditions of existence and uniqueness of the solutions are fulfilled, the fourth order Runge–Kutta (RK4) introduced here is an extension of the Generalized Euler method for approximation of the solution of the FDE

$$\boldsymbol{y_{i+1} = y_i + \frac{1}{6}\,(k_1 + 2k_2 + 2k_3 + k_4)}$$

$$\boldsymbol{k_1 = w_{h,\alpha} f(x_i, y_i)}$$

$$\boldsymbol{k_2 = w_{h,\alpha} f(x_i + \frac{1}{4} w_{h,\alpha}, y_i + \frac{1}{4} k_1)}$$

$$\boldsymbol{k_3 = w_{h,\alpha} f(x_i + \frac{1}{4} w_{h,\alpha}, y_i + \frac{1}{4} k_2)}$$

$$\boldsymbol{k_4 = w_{h,\alpha} f(x_i + w_{h,\alpha}, y_i + k_3)}$$

Where $\boldsymbol{w_{h,\alpha}}$ is defined as following

$$\boldsymbol{w_{h,\alpha}} = \frac{\boldsymbol{h^{\alpha}}}{\boldsymbol{\Gamma(\alpha)}}$$

Chapter 5

Chapter 5

Cotton leaf curl virus model

5.1 Introduction

Cotton Leaf Curl Virus (CLCuV) poses a significant threat to agricultural ecosystems, particularly in tropical and subtropical regions, where it affects a wide range of economically important crops.

Transmitted primarily by the silverleaf whitefly (Bemisia tabaci), CLCuV causes distinctive symptoms such as leaf curling, stunted growth, reduced yield, and yellowing foliage, impacting various host plants besides cotton.

This obligate relationship underscores the intricate ecological interactions among the virus, its vector, and susceptible plant hosts, emphasizing the urgent need for comprehensive pest and disease management strategies.

To address this challenge, researchers and agricultural practitioners must continue to explore innovative approaches, including the development of resistant crop varieties and the implementation of integrated pest management practices, to mitigate the impact of CLCuV on global agriculture.

shown in Fig. 1. In this latter, we remark that infected vectors represent a critical juncture in the transmission cycle, acting as silent couriers of viral genetic material.

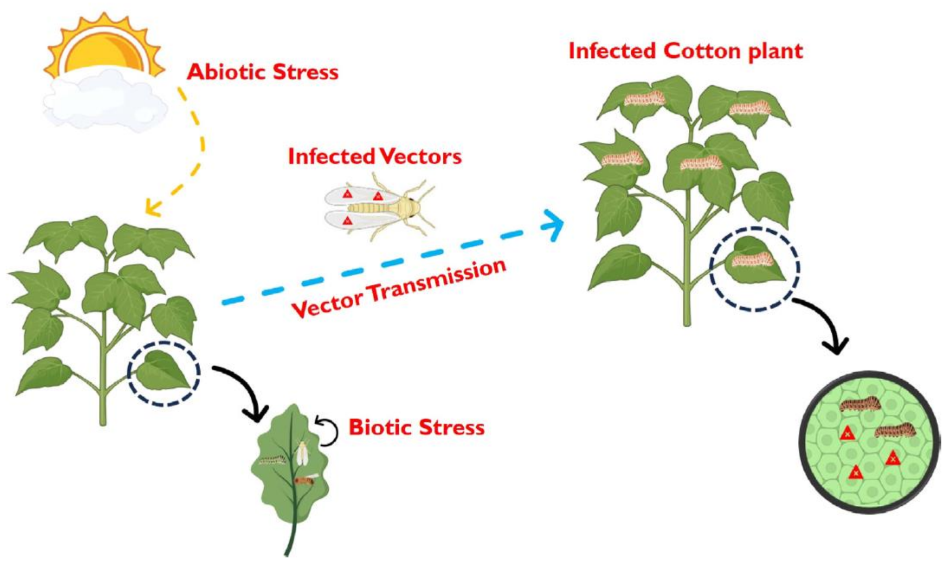


Fig. 1. Host-virus vector interaction for CLCuV.

5.2 Mathematical model

The model is formulated in four-dimensional space, demonstrating the transmission between the cotton and vector classes. The total population is divided into two main categories, namely, the cotton category and the vector category. Furthermore, the cotton and vector populations are formulated into the susceptible state, reflecting the number of cotton plants or vectors that are susceptible to being infected with the virus and the infected class for both the cotton and vector categories.

5.2.1 Model dynamics

Description of model compartments:

$\boldsymbol{S_c(t)}$	Susceptible class of cotton
$\boldsymbol{I_c(t)}$	Infected class of cotton
$\boldsymbol{X_v(t)}$	Susceptible class of vectors
$\boldsymbol{Y_v(t)}$	Infected class of vectors

State equations:

$$\boldsymbol{S_c'(t) = m_1 - \alpha_1 S_c(t) Y_v(t) - d\ \ S_c(t)}$$

$$\boldsymbol{I_c'(t) = \alpha_1 S_c(t) Y_v(t) - (d + \beta_1 + \beta_2) I_c(t)}$$

$$\boldsymbol{X_v'(t) = m_2 - \alpha_2 I_c(t) X_v(t) - \Psi X_v(t),}$$

$$\boldsymbol{Y_v'(t) = \alpha_2 I_c(t) X_v(t) - \Psi Y_v(t)}$$

Parameters:

m_1	The replanting rate of cotton
α_1	
β_1	The induced death rate of infected cotton plants
β_2	The elimination rate of infected cotton plants
m_2	
α_2	The transition rate of vulnerable vectors to infected vectors
Ψ	Natural death rate of vector population

5.3 Optimal Control Problem

5.3.1 Objective and Stat equations

Objective: Minimize the cost functional:

$$\phi(u_k) = \int_0^T \left[A_1\, I_c + A_2\, Y_v + \frac{C_1}{2} u_1^2(t)\right] dt$$

State equations:

$$S_c'(t) = m_1 - \alpha_1 S_c(t) Y_v(t) - d S_c(t)$$

$$I_c'(t) = \alpha_1 S_c(t) Y_v(t) - (d + \beta_1 + \beta_2) I_c(t)\,,$$

$$X_v'(t) = m_2 - \alpha_2 I_c(t) X_v(t) - \Psi X_v(t) - u_1(t) X_v(t),$$

$$Y_v'(t) = \alpha_2 I_c(t) X_v(t) - \Psi Y_v(t) - u_1(t)\, Y_v(t).$$

5.3.2 Hamiltonian and Adjoint Equations

Hamiltonian:

$$\begin{aligned} H = {} & A_1\, I_c + A_2\, Y_v + \frac{C_1}{2} u_1^2(t) + \lambda_1\big(m_1 - \alpha_1 S_c(t) Y_v(t) - d S_c(t)\big) \\ & + \lambda_2\,(\alpha_1 S_c(t) Y_v(t) - (d + \beta_1 + \beta_2) I_c(t)\,) \\ & + \lambda_3\,(\,m_2 - \alpha_2 I_c(t) X_v(t) - \Psi X_v(t) - u_1(t) X_v(t)) \\ & + \lambda_4\,(\alpha_2 I_c(t) X_v(t) - \Psi Y_v(t) - u_1(t)\, Y_v(t)) \end{aligned}$$

Adjoint equations:

$$\lambda_1'(t) = \alpha_1\, \lambda_2(t) Y_v(t) - \lambda_1(t)(d + \alpha_1\, Y_v(t))$$

$$\lambda_2'(t) = A_1 - \lambda_2(t)(\,\beta_1 + \beta_2 + d) + \alpha_2\, \big(\lambda_4(t) - \lambda_3(t)\big)\, X_v$$

$$\lambda_3'(t) = \alpha_2\, I_c\,(t)\, \lambda_4(t) - \lambda_3(t)(\alpha_2\, I_c(t) + \psi + u_1\,(t))$$

$$\lambda_4'(t) = A_2 + \alpha_1\big(\lambda_2(t) - \lambda_1(t)\big) S_c(t) - \lambda_4(t)(u_1(t) + \psi)$$

5.3.3 Case Studies:

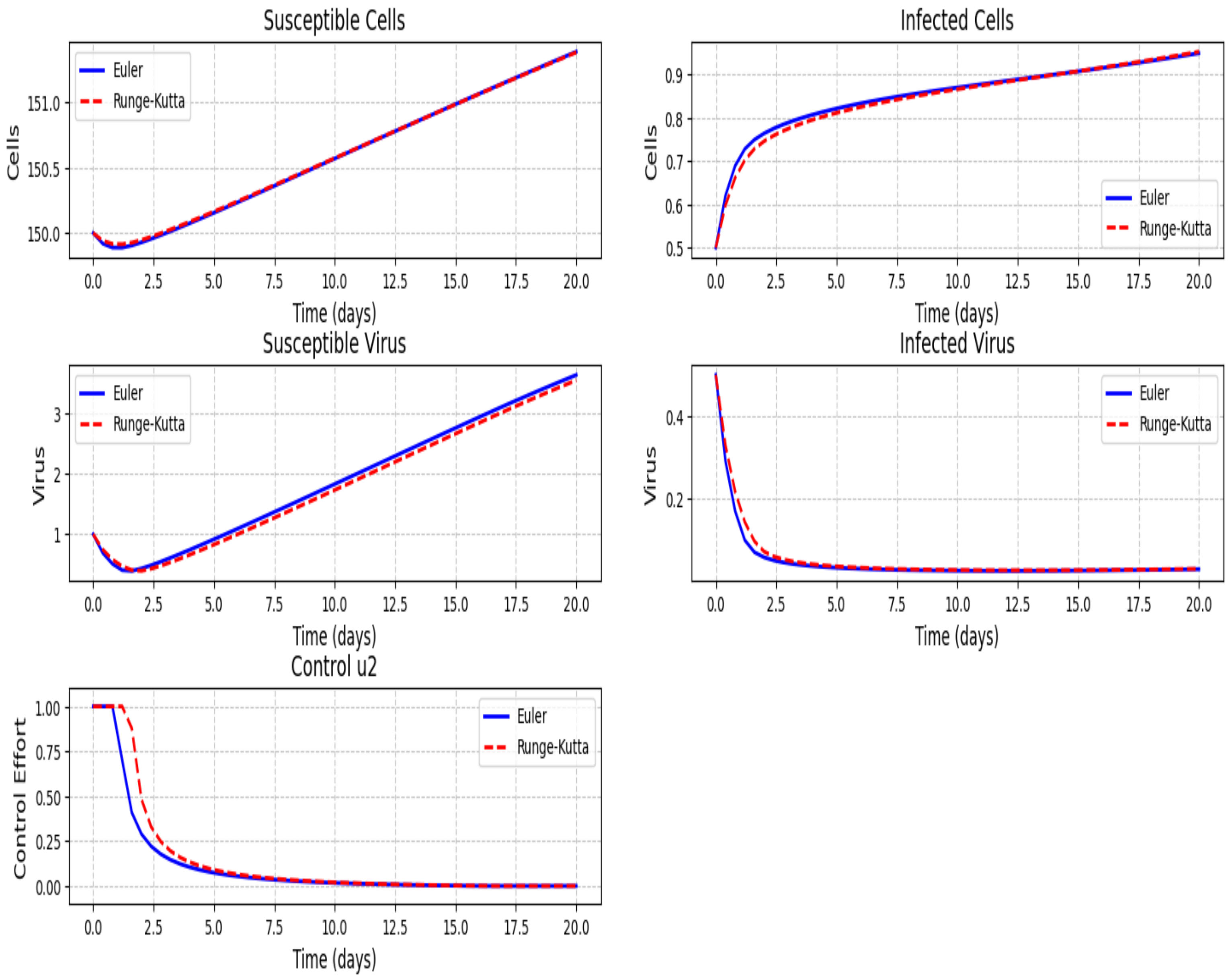
In this study, we present a comparative analysis of two numerical techniques Euler's method and the Runge-Kutta method used to solve the fractional optimal control model of CLCuV. Both methods are implemented to simulate the dynamics of susceptible and infected plant cells and viruses under control interventions. The model incorporates control functions aimed at reducing infection through strategies such as chemical treatment or biological resistance.

Assume

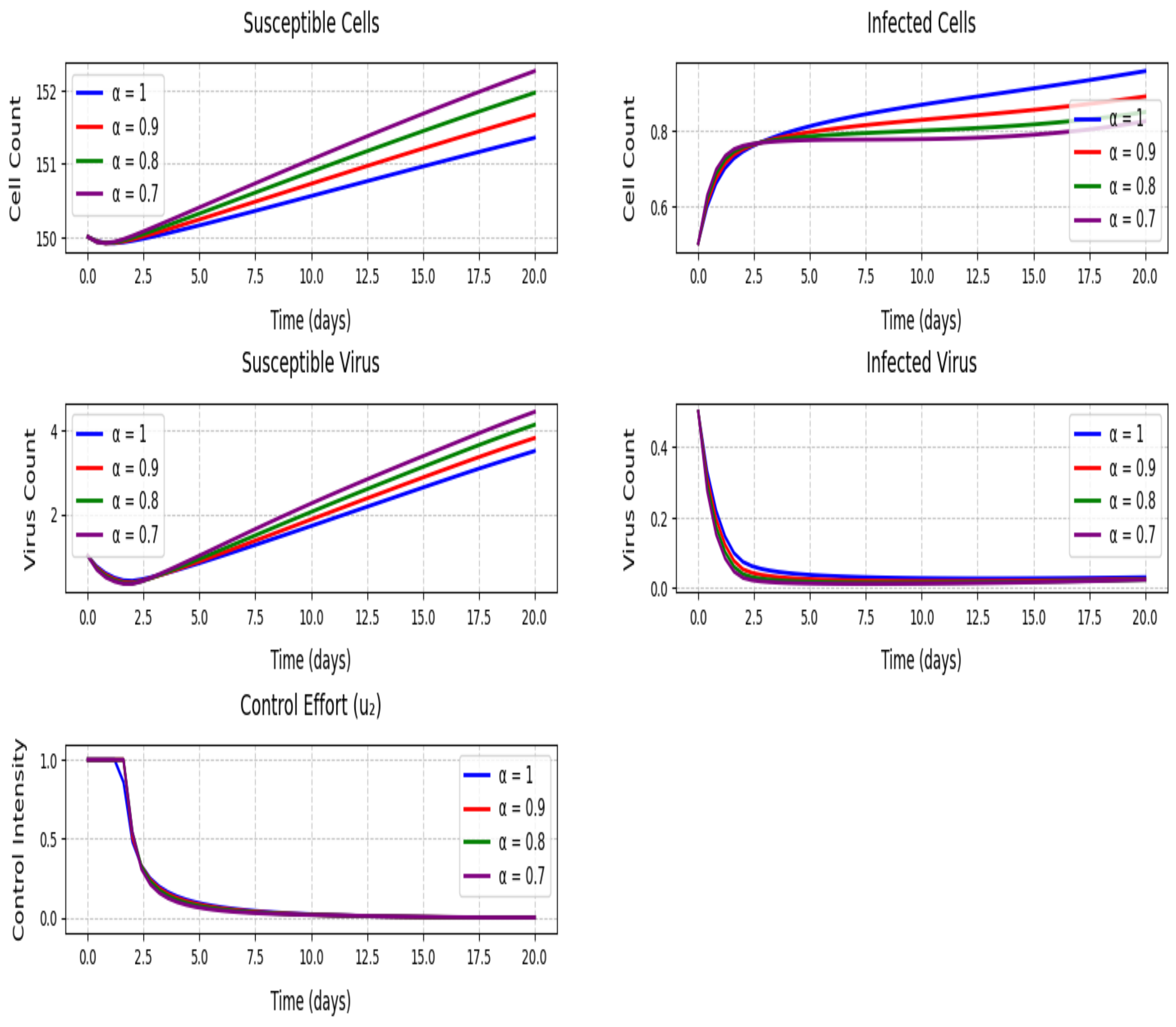
$$S_c = 150 \quad I_c = 0.5 \quad X_v = 1 \quad Y_v = 0.5 \quad m_1 = 1$$

$$\alpha_1 = 0.004 \quad d = 0.004 \quad \beta_1 = 0.002 \quad \beta_2 = 0.0024$$

$$m_2 = 0.27 \quad \alpha_2 = 0.0005 \quad \Psi = 0.03 \quad \alpha = 0.9$$



This study investigates the impact of varying fractional orders $\alpha = 1, 0.9, 0.8$, and 0.7 on the behavior of a CLCuV model under optimal control using the Runge-Kutta method.


Susceptible Cells
Infected Cells
Susceptible Virus
Infected Virus
Control Effort (u₂)
Cell Count
Virus Count
Control Intensity
Time (days)
α = 1
α = 0.9
α = 0.8
α = 0.7