Modelling CGM time series using Neural Ordinary Differential Equation

A Thesis

submitted to
Indian Institute of Science Education and Research Pune
in partial fulfillment of the requirements for the
BS-MS Dual Degree Programme

by

Alekh Ranjan Mahankudo

Indian Institute of Science Education and Research Pune
Dr. Homi Bhabha Road,
Pashan, Pune 411008, INDIA.

April, 2020

Supervisor: Dr.Pranay Goel

Certificate

This is to certify that this dissertation entitled Modelling CGM time series using Neural Ordinary Differential Equationtowards the partial fulfilment of the BS-MS dual degree programme at the Indian Institute of Science Education and Research, Pune represents study/work carried out by Alekh Ranjan Mahankudoat Indian Institute of Science Education and Research under the supervision of Dr.Pranay Goel, Associate Professor, Department of Biology , during the academic year 2019-2020.

Dr.Pranay Goel

Committee:

Dr.Pranay Goel

Dr.M.S.Santhanam

This thesis is dedicated to my parents and my sister.

Declaration

I hereby declare that the matter embodied in the report entitled Modelling CGM time series using Neural Ordinary Differential Equation are the results of the work carried out by me at the Department of Biology, Indian Institute of Science Education and Research, Pune, under the supervision of Dr.Pranay Goel and the same has not been submitted elsewhere for any other degree.

Alekh Ranjan Mahankudo

Acknowledgments

This project gave me an excellent opportunity to learn the subject. I enjoyed working on the problem and learned many things from different fields. Dr. Pranay Goel was my supervisor. He taught me how to approach a scientific problem and whenever I got stuck in any problem, he helped me out. I want to thank Dr. M.S.Santhanam for his suggestions during the mid-term evaluation. I want to thank Arjun and Sandra for always being there to help me in the project, mock presentations, and reviewing the thesis. Being new in the department, they made sure that I was comfortable in the lab.
I would also like to thank my friends Vaibhav, Sujeet, Rugwed, Rahul, Shivam, and Snehal, for supporting me throughout the project. I cannot thank my parents and my sister enough in mere words for their never-dying support and their trust in me.

Abstract

According to a government survey(2019), 11.8% of people in India have diabetes. Understanding the glucose-insulin dynamics could help in designing clinical trials and help in designing therapies for prevention. There have been attempts to model the glucose-insulin dynamics as a step in that direction. Recently deep neural networks have been used to model a dynamic system. In our work, we take an existing dynamical system (Glucose-insulin) model and incorporate a simple neural network (twice composed ReLu, with just two parameters). We show that this simple neural network (a piecewise linear term) can be used to approximate a non-linear term in the dynamical system. We introduce an algorithm to find the parameters of the neural network to fit the new dynamical system (with the neural network) to the Continous Glucose Monitoring (CGM) data. The final results show that even after replacing the non-linear term with a piecewise linear function, the glucose-insulin time series obtained are close to the one obtained from the original glucose-insulin dynamics.

Contents

Introduction

Continuous Glucose Monitoring (CGM) is a technique that records blood glucose at regular intervals. A sensor is fixed to the arm that reports blood glucose every 15 minutes for two weeks. There is an existing Glucose-Insulin model for analyzing the CGM data[1]. This existing model gives the dynamics of Glucose-insulin levels in the blood.

For modeling a continuous-time system using a neural network, there are two possible methods:

1. We approximate the system itself, that is, every hidden layer of the neural network would correspond to the value of the state at some time 't'.

2. We approximate the dynamics of the system using a neural network. Here the output of the network would be the velocity vector of the state at any time 't'.

The second approach is better for a continuous-time system, as a continuous-time system typically has a large number of time points close to infinity, which would require, in principle, an infinite layer if we go by approach 1.

While using approach 2, we can have a finite layered neural network as it needs to give just the velocity vector at any given time.

The differential equation in the second approach is what we refer to as a neural differential equation. In using a neural network for learning the dynamics of a system, we would require to train the network. Training involves using an optimization algorithm to find a set of parameters to best map inputs to outputs. In the case of differential equations, sensitivity equations have been used for finding the best set of parameters. So in the case of neural differential equations, these sensitivity equations can be used in order to train the network.

We show that a simple neural network (twice composed ReLu) can be used to replace a non-linear term in the Glucose-insulin dynamics. This opens up the possibility to use simple piecewise linear functions in place of highly non-linear functions while modeling dynamical systems.

Chapter 1

Preliminaries

1.1 Neural Network

A typical Neural Network is made up of some basic units. These units are called as Artificial Neuron/ Perceptron. A Perceptron takes an input vector (x) multiplies a weight to it element-wise, adds bias to it, and finally applies a function (called as **activation function**) on this. The value thus obtained is called the output of the Perceptron.

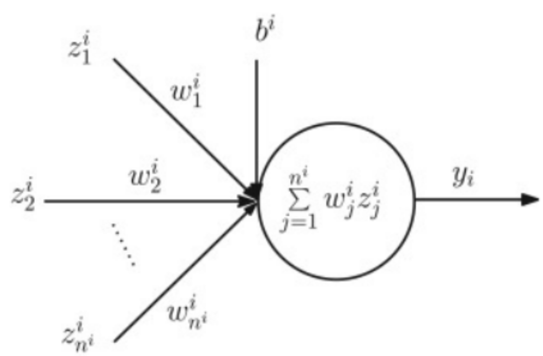


Figure 1.1: Image of a perceptron that takes an input multiplies weight to it, adds bias, and then applies a function to give the output. Source:https://www.sciencedirect.com/science/article/pii/S0149763416305176

When we stack these perceptrons one above the other, they form a hidden layer. When we take multiple hidden layers and connect each consecutive layer, then we get a neural network.

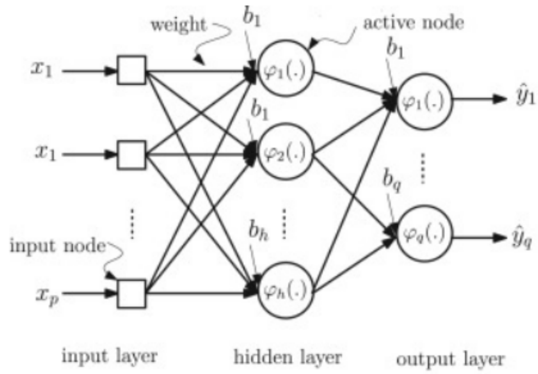


Figure 1.2: Image of a feedforward neural network with one input layer, one hidden layer, and one output layer.Source:https://www.sciencedirect.com/science/article/pii/S0149763416305176

A simple and most commonly used activation function is **Rectified Linear Unit (ReLu)**. ReLu is defined as

$$ReLu(x) = \begin{cases} x & x > 0 \\ 0 & x \leq 0 \end{cases}$$

In this work we define ReLu($a * x + b$) as **Strang**($x; a, b$). Here a, b are the parameters. Thus a n composition of ReLu would be called as Strang function of order n, represented as **Strang-n**($x; a, b$).

1.2 Neural Ordinary Differential Equations

A general Ordinary Differential Equation is of the form

$$\frac{d^n x(t)}{dt} = f(t, x, x^{(1)}, x^{(2)}, ..; parameters)$$

When we introduce a neural network on the right hand side of the above equation, it gives a Neural Ordinary Differential Equation.

$$\frac{d^n x(t)}{dt} = NeuralNetwork$$

A simple degree one Neural ODE with Neural network as composition of ReLu (with parameters a,b)can be written as

$$\frac{dx(t)}{dt} = Relu(a * Relu(a * ....Relu(a * x + b) + b)... + b)$$

Chapter 2

Sensitivity

The change in solution with respect to changes in the parameter is called as **sensitivity**. In the case of ODE's, we have various approaches for finding the sensitivity two of them are :

- Forward sensitivity method
- Backward/ Adjoint method.

2.1 Forward Sensitivity Method (Continous case)

Let $x(t)$ denote the state at time t, $x(0)$ is the initial state and θ denotes the parameter. We have

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

In this case let's take $f(x(t), t, \theta)$ to be continous with respect to time.
We need to compute the sensitivity that is $\frac{dx(t)}{d\theta}$
Let's define

$$a(t) = \frac{dx(t)}{d\theta}$$

Then,

$$\frac{da(t)}{dt} = \lim_{\epsilon \to 0} \frac{a(t+\epsilon) - a(t)}{\epsilon} \tag{2.1}$$

Now,

$$\begin{aligned}
a(t+\epsilon) &= \frac{dx(t+\epsilon)}{d\theta} \\
&= \frac{d}{d\theta}(x(t) + \epsilon \frac{dx}{dt} + O(\epsilon^2)) \\
&= \frac{d}{d\theta}(x(t) + \epsilon f(x(t), t, \theta) + O(\epsilon^2)) \\
&= \lim_{\varphi \to 0} \frac{x(t) + \epsilon f(x(t), t, \theta + \varphi) - x(t, \theta) - \epsilon f(x(t), t, \theta)}{\varphi} \\
&= \lim_{\varphi \to 0} \frac{x(t, \theta + \varphi) - x(t, \theta)}{\varphi} + \lim_{\varphi \to 0} \frac{\epsilon(f(x(t), t, \theta + \varphi) - f(x(t), t, \theta)}{\varphi} \\
&= \frac{dx(t)}{d\theta} + \lim_{\varphi \to 0} \frac{\epsilon(f(x(t), t, \theta + \varphi) - f(x(t), t, \theta)}{\varphi} \\
&= a(t) + \lim_{\varphi \to 0} \frac{\epsilon(f(x(t), t, \theta + \varphi) - f(x(t), t, \theta)}{\varphi}
\end{aligned}$$

Using the above result in equation 2.1, we get

$$\frac{da(t)}{dt} = \lim_{\epsilon \to 0} \lim_{\varphi \to 0} \frac{\epsilon(f(x(t), t, \theta + \varphi) - f(x(t), t, \theta)}{\varphi \epsilon} \tag{2.2}$$

$$= \lim_{\varphi \to 0} \frac{f(x(t), t, \theta + \varphi) - f(x(t), t, \theta)}{\varphi} \tag{2.3}$$

If f is differentiable with respect to the parameter θ then the above equation can be written as

$$\frac{da(t)}{dt} = \frac{df(x(t), t, \theta)}{d\theta} \tag{2.4}$$

$$= \frac{\partial f}{\partial x}\frac{\partial x}{\partial \theta} + \frac{\partial f}{\partial \theta} \tag{2.5}$$

Even if f is not differentiable with respect to θ, still the equation 2.3 is well defined as long as the the function is continous with respect to the parameter θ. Hence for using equation 2.3, for computing sensitivity we require that the function f (where $\frac{dx(t)}{dt} = f(x(t), t, \theta)$ be continous with respect to θ.

2.1.1 Implementation of the method

Lotka-Voltera model:

Let $x(t)$ and $y(t)$ denote the population of species1 and species2 respectively at time t.The parameters are a, b, c, d

The differential equations are

$$\begin{aligned} \frac{dx(t)}{dt} &= ax(t) - bx(t)y(t) = f_1(x(t), y(t), a, b) \\ \frac{dy(t)}{dt} &= cx(t)y(t) - dy(t) = f_2(x(t), y(t), c, d) \end{aligned}$$

Let, the sensitivity with respect to the parameter 'a' be defined as follows:

$$\begin{aligned} \frac{dx(t)}{da} &= w_1(t) \\ \frac{dy(t)}{da} &= w_2(t) \end{aligned}$$

Then we have

$$\frac{dw_1(t)}{dt} = \frac{df_1(x(t), y(t), a, b)}{da} \tag{2.6}$$

$$= \frac{\partial f_1}{\partial x(t)} \frac{\partial x(t)}{\partial a} + \frac{\partial f_1}{\partial y(t)} \frac{\partial y(t)}{\partial a} + \frac{\partial f_1}{\partial a} \tag{2.7}$$

$$= \frac{\partial f_1}{\partial x(t)} w_1(t) + \frac{\partial f_1}{\partial y(t)} w_2(t) + \frac{\partial f_1}{\partial a} \tag{2.8}$$

Similarly we also have,

$$\frac{dw_2(t)}{dt} = \frac{df_2(x(t), y(t), c, d)}{da} \tag{2.9}$$

$$= \frac{\partial f_2}{\partial x(t)}\frac{\partial x(t)}{\partial a} + \frac{\partial f_2}{\partial y(t)}\frac{\partial y(t)}{\partial a} \tag{2.10}$$

$$= \frac{\partial f_2}{\partial x(t)} w_1(t) + \frac{\partial f_2}{\partial y(t)} w_2(t) \tag{2.11}$$

Now,

$$\begin{aligned}
\frac{\partial f_1}{\partial x(t)} &= a - by(t) \\
\frac{\partial f_1}{\partial y(t)} &= -bx(t) \\
\frac{\partial f_1}{\partial a} &= x(t)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial f_2}{\partial x(t)} &= cy(t) \\
\frac{\partial f_2}{\partial y(t)} &= -d
\end{aligned}$$

So using the above expressions in equations 2.8 and 2.11.We get the following differential equations with initial conditions.

$$\begin{aligned}
\frac{dw1(t)}{dt} &= (a - by(t))w1(t) - bx(t)w2(t) \\
\frac{dw2(t)}{dt} &= cy(t)w1(t) + (cx(t) - d)w2(t) \\
w1(0) &= 0 \\
w2(0) &= 0
\end{aligned}$$

Solving this Initial value problem gives the sensitivity $w1(t)$ i.e$\frac{dx(t)}{da}$ and $w2(t)$ i.e $\frac{dy(t)}{da}$. The required sensitivities.

Computational details :

Initial time t0 = 0

Final time t1 = 50

Initial state :

x(0),y(0) = 5,5

w1(0),w2(0) = 0,0

Parameter :

b,c,d = 0.4,0.1,0.4

Change in parameter (for direct computation) = 0.000001

Time step =0.05

Computations :

The below table shows the result of the computations $w1(t)$ i.e $\frac{dx(t)}{da}$) and $w2(t)$ i.e $\frac{dy(t)}{da}$), at $t = 50$, for different values of parameter a. In the table below, we compare the sensitivity obtained by using forward method with that obtained by direct computation.

Parameter Value(a)	Direct computation (w1,w2)	Forward Sensitivity Method(w1,w2)
1.1	123.714015, 2.310671	123.712232, 2.310508
2	-7.702622, 14.212229	-7.702562 , 14.212237
3	-0.906223, 11.754277	-0.906243 , 11.754232

2.2 Forward Sensitivity (Discontinous case)

Consider the differential equation $\frac{dx(t)}{dt} = g(x, t; \theta_1, \theta_2)$

$$g(x, t; \theta_1, \theta_2) = \begin{cases} f_1(x, t; \theta_1) & 0 \leq t \leq t_0 \\ f_2(x, t; \theta_2) & t > t_0 \end{cases}$$

Here g is discontinous with respect to time.The point of discontinuity is t_0.

Now take the sensitivity as $a(t) = \frac{dx(t)}{d\theta_1}$, then we have

$$\begin{aligned}
a(t+\epsilon) &= \frac{dx(t+\epsilon)}{d\theta_1} && (2.12)\\
&= \lim_{\delta\theta_1 \to 0} \frac{x(t+\epsilon, \theta_1 + \delta\theta_1) - x(t+\epsilon, \theta_1)}{\delta\theta_1} && (2.13)\\
&= \lim_{\delta\theta_1 \to 0} \frac{x(0) + \int_0^{t+\epsilon} g(x, t'; \theta_1 + \delta\theta_1, \theta_2)dt' - x(0) - \int_0^{t+\epsilon} g(x, t'; \theta_1, \theta_2)dt'}{\delta\theta_1} && (2.14)\\
&= \lim_{\delta\theta_1 \to 0} \frac{\int_0^{t+\epsilon} g(x, t'; \theta_1 + \delta\theta_1, \theta_2)dt' - \int_0^{t+\epsilon} g(x, t'; \theta_1, \theta_2)dt'}{\delta\theta_1} && (2.15)
\end{aligned}$$

Similarly we have,

$$a(t) = \lim_{\delta\theta_1 \to 0} \frac{\int_0^{t} g(x, t'; \theta_1 + \delta\theta_1, \theta_2)dt' - \int_0^{t} g(x, t'; \theta_1, \theta_2)dt'}{\delta\theta_1} \quad (2.16)$$

Now,

$$\frac{da(t)}{dt} = \lim_{\epsilon \to 0} \frac{a(t+\epsilon) - a(t)}{\epsilon} \quad (2.17)$$

Using (2.15) and (2.16) in (2.17), we get

$$\frac{da(t)}{dt} = \lim_{\epsilon \to 0} \frac{1}{\epsilon} \lim_{\delta\theta_1 \to 0} \frac{\int_t^{t+\epsilon} g(x, t'; \theta_1 + \delta\theta_1, \theta_2) - g(x, t'; \theta_1, \theta_2)dt'}{\delta\theta_1} \quad (2.18)$$

Case I: $t, t+\epsilon \leq t_0$

$$\frac{da(t)}{dt} = \lim_{\epsilon \to 0} \frac{1}{\epsilon} \lim_{\delta\theta_1 \to 0} \frac{\int_t^{t+\epsilon} f_1(x, t'; \theta_1 + \delta\theta_1, \theta_2) - f_1(x, t'; \theta_1, \theta_2) dt'}{\delta\theta_1} \tag{2.19}$$

$$= \lim_{\epsilon \to 0} \frac{1}{\epsilon} \lim_{\delta\theta_1 \to 0} \frac{\left[f_1(x, t; \theta_1 + \delta\theta_1, \theta_2) - f_1(x, t; \theta_1, \theta_2)\right](t + \epsilon - t)}{\delta\theta_1} \tag{2.20}$$

$$= \lim_{\epsilon \to 0} \frac{1}{\epsilon} \lim_{\delta\theta_1 \to 0} \frac{\left[f_1(x, t; \theta_1 + \delta\theta_1, \theta_2) - f_1(x, t; \theta_1, \theta_2)\right]\epsilon}{\delta\theta_1} \tag{2.21}$$

$$= \lim_{\epsilon \to 0} \frac{1}{\epsilon} \frac{df_1(x, t; \theta_1)\epsilon}{d\theta_1} \tag{2.22}$$

$$= \frac{df_1(x, t; \theta_1, \theta_2)}{d\theta_1} \tag{2.23}$$

Case II: $t, t+\epsilon \geq t_0$

Similar to case I, we get the following expression

$$\frac{da(t)}{dt} = \lim_{\delta\theta_1 \to 0} \frac{f_2(x, t; \theta_1 + \delta\theta_1, \theta_2) - f_2(x, t; \theta_1, \theta_2)}{\delta\theta_1} \tag{2.24}$$

$$= \frac{df_2(x, t; \theta_1, \theta_2)}{d\theta_2} \tag{2.25}$$

Using case I we can get the value of $a(t-\epsilon)$, Now to get $a(t+\epsilon)$, we use the following equations

$$
\begin{aligned}
a(t_0+\epsilon) &= \frac{dx(t_0+\epsilon)}{d\theta_1} \\
&= \lim_{\delta\theta_1 \to 0} \frac{x(t_0+\epsilon, \theta_1+\delta\theta_1) - x(t_0+\epsilon, \theta_1)}{\delta\theta_1} \\
&= \lim_{\delta\theta_1 \to 0} \frac{x(t_0-\epsilon; \theta_1+\delta\theta_1) + \int_{t_0-\epsilon}^{t_0+\epsilon} g(x, t'; \theta_1+\delta\theta_1, \theta_2)dt' - x(t_0+\epsilon; \theta_1)}{\delta\theta_1} \\
&= \lim_{\delta\theta_1 \to 0} \frac{x(t_0-\epsilon, \theta_1) + \frac{dx(t_0-\epsilon)}{d\theta_1}\delta\theta_1 + \int_{t_0-\epsilon}^{t_0+\epsilon} g(x, t'; \theta_1+\delta\theta_1, \theta_2)dt' - x(t_0+\epsilon, \theta_1)}{\delta\theta_1} \\
&= \lim_{\delta\theta_1 \to 0} \frac{x(t_0-\epsilon, \theta_1) + a(t_0-\epsilon)\delta\theta_1 + \int_{t_0-\epsilon}^{t_0+\epsilon} g(x, t'; \theta_1+\delta\theta_1, \theta_2)dt' - x(t_0+\epsilon, \theta_1)}{\delta\theta_1} \\
&= \lim_{\delta\theta_1 \to 0} \frac{x(t_0-\epsilon, \theta_1) + a(t_0-\epsilon)\delta\theta_1 + \int_{t_0-\epsilon}^{t_0+\epsilon} g(x, t'; \theta_1+\delta\theta_1, \theta_2)dt'}{\delta\theta_1} \\
&\quad + \lim_{\delta\theta_1 \to 0} \frac{-x(t_0-\epsilon, \theta_1) - \int_{t_0-\epsilon}^{t_0+\epsilon} g(x, t'; \theta_1, \theta_2)dt'}{\delta\theta_1} \\
&= \lim_{\delta\theta_1 \to 0} \frac{a(t_0-\epsilon)\delta\theta_1 + \int_{t_0-\epsilon}^{t_0+\epsilon} g(x, t'; \theta_1+\delta\theta_1, \theta_2)dt' - \int_{t_0-\epsilon}^{t_0+\epsilon} g(x, t'; \theta_1, \theta_2)dt'}{\delta\theta_1} \\
&= a(t_0-\epsilon) + \lim_{\delta\theta_1 \to 0} \frac{\int_{t_0-\epsilon}^{t_0+\epsilon} g(x, t'; \theta_1+\delta\theta_1, \theta_2)dt' - g(x, t'; \theta_1, \theta_2)dt'}{\delta\theta_1} \\
&= a(t_0-\epsilon) + \lim_{\delta\theta_1 \to 0} \frac{\int_{t_0-\epsilon}^{t_0} f_1(x, t'; \theta_1+\delta\theta_1, \theta_2) - f_1(x, t'; \theta_1, \theta_2)dt'}{\delta\theta_1} \\
&\quad + \lim_{\delta\theta_1 \to 0} \frac{\int_{t_0}^{t_0+\epsilon} f_2(x, t'; \theta_1+\delta\theta_1, \theta_2) - f_2(x, t'; \theta_1, \theta_2)dt'}{\delta\theta_1} \\
&= a(t_0-\epsilon) + \lim_{\delta\theta_1 \to 0} \frac{f_1(x, t_0; \theta_1+\delta\theta_1, \theta_2) - f_1(x, t_0; \theta_1, \theta_2)(t_0 - t_0 + \epsilon)}{\delta\theta_1} \\
&\quad + \lim_{\delta\theta_1 \to 0} \frac{f_2(x, t_0; \theta_1+\delta\theta_1, \theta_2) - f_2(x, t_0; \theta_1, \theta_2)(t_0 + \epsilon - t_0)}{\delta\theta_1} \\
&= a(t_0-\epsilon) + \left(\frac{df_1(x, t_0; \theta_1)}{d\theta_1} + \frac{df_2(x, t_0; \theta_2)}{d\theta_1} \right) \epsilon
\end{aligned}
$$

So while computing $a(t_0+\epsilon)$ where t_0 is the point of discontinuity we can use the equation

$$
a(t_0+\epsilon) = a(t_0-\epsilon) + \left(\frac{df_1(x, t_0; \theta_1)}{d\theta_1} + \frac{df_2(x, t_0; \theta_2)}{d\theta_1} \right) \epsilon
$$

2.3 Adjoint method /Backward Sensitivity

Let $x(t)$ denote the state at time t, $x(0)$ is the initial state and $x(T)$ is the final state, θ denotes the parameter.
We have

$$\frac{dx(t)}{dt} = f(x(t), t; \theta)$$

Here $f(x(t), t; \theta)$ is continous with respect to time.

We need to compute the sensitivity $\frac{dx(T)}{d\theta}$
Lets, define

$$F(x, t, \theta) = x(T) + \int_0^T \lambda(t) \left[\frac{dx(t)}{dt} - f(x, t; \theta) \right] dt$$

Then we have

$$\frac{dF}{d\theta} = \frac{dx(T)}{d\theta} + \int_0^T \lambda(t) \left[\frac{d}{d\theta} \frac{dx(t)}{dt} - \frac{\partial f}{\partial x} \frac{\partial x}{\partial \theta} - \frac{\partial f}{\partial \theta} \right] dt \tag{2.26}$$

Now taking $\int_0^T \lambda(t) \frac{d}{d\theta} \frac{dx(t)}{dt} dt$ and using integration by parts we get

$$\begin{aligned}
\int_0^T \lambda(t) \frac{d}{d\theta} \frac{dx(t)}{dt} dt &= \int_0^T \lambda(t) \frac{d}{dt} \frac{dx(t)}{d\theta} dt \\
&= \lambda(T) \frac{dx(T)}{d\theta} - \lambda(0) \frac{dx(0)}{d\theta} - \int_0^T \frac{d\lambda(t)}{dt} \frac{dx}{d\theta} dt \\
&= \lambda(T) \frac{dx(T)}{d\theta} - \int_0^T \frac{d\lambda(t)}{dt} \frac{dx}{d\theta} dt
\end{aligned}$$

Using the above expression in equation 2.26 we get,

$$\begin{aligned}
\frac{df}{d\theta} &= \frac{dx(T)}{d\theta} + \lambda(T) \frac{dx(T)}{d\theta} - \int_0^T \frac{d\lambda(t)}{dt} \frac{dx}{d\theta} - \int_0^T \lambda(t) \left[\frac{\partial f}{\partial x} \frac{\partial x}{\partial \theta} + \frac{\partial f}{\partial \theta} \right] dt \\
&= [1 + \lambda(T)] \frac{x(T)}{d\theta} - \int_0^T \frac{dx}{d\theta} \left[\frac{d\lambda(t)}{dt} + \lambda(t) \frac{\partial f}{\partial x} \right] dt - \int_0^T \lambda(t) \frac{\partial f}{\partial \theta} dt
\end{aligned}$$

So if we choose $\lambda(t)$ such that

$$\begin{aligned}\lambda(T) &= -1 \\ \frac{d\lambda(t)}{dt} &= -\lambda(t)\frac{\partial f}{\partial x}\end{aligned}$$

Then we get,

$$\frac{dF}{d\theta} = \frac{dx(T)}{d\theta} = -\int_0^T \lambda(t)\frac{\partial f(x,t;\theta)}{\partial \theta}dt$$

This equation gives sensitivity at any time T.

2.3.1 Implementation of the method

Consider the following differential equation with a intital condition.

$$\frac{dx(t)}{dt} = 1 - px(t) \tag{2.27}$$

$$x(0) = 0 \tag{2.28}$$

So the adjoint differential equation that we get is

$$\begin{aligned}\frac{d\lambda(t)}{dt} &= -\lambda(t)\frac{\partial f(x,t;\theta)}{\partial x(t)} = \lambda(t)\theta \\ \lambda(T) &= -1\end{aligned}$$

Now

$$\frac{dx(T)}{d\theta} = -\int_0^T \lambda(t)\frac{\partial f(x,t;\theta)}{\partial \theta}dt = -\int_0^T \lambda(t)(-x(t))dt$$

This equation can be used to get sensitivity at any time t.

Computations :

Computational details:

Initial time t0 = 0

Final time T = 2

Initial state :

$x(0) = 0$

$\lambda(T)$ = -1

Change in parameter (for direct computation) = 0.000001

Time step =0.001

ODE solver = Euler

The table below shows the result of the computation of sensitivity i.e $\frac{dx(t)}{dp}$, for different values of the parameter (p) at $t = 2$. We compare the value of sensitivity obtained by adjoint method and direct computation.

Parameter Value (p)	Direct computation	Adjoint Method
2	-0.2276	-0.2276
4	-0.0623	-0.0623
5	-0.0142	-0.0142

2.4 Adjoint method /Backward Sensitivity (Discontinous Case)

Let $x(t)$ denote the state at time t, $x(0)$ is the initial state and $x(T)$ is the final state, θ denotes the parameter.

We have

$$f(x,t;\theta) = \begin{cases} f_1(x,t;\theta) & 0 \leq t \leq t_0 \\ f_2(x,t;\theta) & t > t_0 \end{cases}$$

Here f is discontinous at t_0.

We want to compute the sensitivity, $\frac{dx(T)}{d\theta}$

Let's, define

$$F(x,t,\theta) = x(T) + \int_0^T \lambda(t)\left[\frac{dx(t)}{dt} - f(x,t;\theta)\right]dt$$

Then we have

$$\frac{dF}{d\theta} = \frac{dx(T)}{d\theta} + \int_0^T \lambda(t)\left[\frac{d}{d\theta}\frac{dx(t)}{dt} - \frac{\partial f}{\partial x}\frac{\partial x}{\partial \theta} - \frac{\partial f}{\partial \theta}\right]dt \tag{2.29}$$

$$\frac{dF}{d\theta} = \frac{dx(T)}{d\theta} + \int_0^{t_0} \lambda(t)\left[\frac{d}{d\theta}\frac{dx(t)}{dt} - \frac{\partial f_1}{\partial x}\frac{\partial x}{\partial \theta} - \frac{\partial f_1}{\partial \theta}\right]dt + \int_{t_0}^T \lambda(t)\left[\frac{d}{d\theta}\frac{dx(t)}{dt} - \frac{\partial f_2}{\partial x}\frac{\partial x}{\partial \theta} - \frac{\partial f_2}{\partial \theta}\right]dt \tag{2.30}$$

Now taking $\int_0^{t_0} \lambda(t)\frac{d}{d\theta}\frac{dx(t)}{dt}dt$ and using integration by parts we get

$$\begin{aligned}
\int_0^{t_0} \lambda(t)\frac{d}{d\theta}\frac{dx(t)}{dt}dt &= \int_0^{t_0} \lambda(t)\frac{d}{dt}\frac{dx(t)}{d\theta}dt \\
&= \lambda(t_0)\frac{dx(t_0)}{d\theta} - \lambda(0)\frac{dx(0)}{d\theta} - \int_0^{t_0} \frac{d\lambda(t)}{dt}\frac{dx}{d\theta}dt \\
&= \lambda(t_0)\frac{dx(t_0)}{d\theta} - \int_0^{t_0} \frac{d\lambda(t)}{dt}\frac{dx}{d\theta}dt
\end{aligned}$$

Similarly, taking $\int_{t_0}^{T} \lambda(t)\frac{d}{d\theta}\frac{dx(t)}{dt}dt$ and using integration by parts we get

$$\int_{t_0}^T \lambda(t)\frac{d}{d\theta}\frac{dx(t)}{dt}dt = \lambda(T)\frac{dx(T)}{d\theta} - \lambda(t_0)\frac{dx(t_0)}{d\theta} - \int_{t_0}^T \frac{d\lambda(t)}{dt}\frac{dx}{d\theta}dt$$

Using the above expression in equation 2.31 we get,

$$
\begin{aligned}
\frac{dF}{d\theta} &= \frac{dx(T)}{d\theta} + \lambda(t_0)\frac{dx(t_0)}{d\theta} - \int_0^{t_0} \frac{d\lambda(t)}{dt}\frac{dx(t)}{d\theta} - \int_0^{t_0} \lambda(t)\left[\frac{\partial f_1}{\partial x}\frac{\partial x}{\partial \theta} + \frac{\partial f_1}{\partial \theta}\right] dt \\
& +\lambda(T)\frac{dx(T)}{d\theta} - \lambda(t_0)\frac{dx(t_0)}{d\theta} - \int_{t_0}^{T} \frac{d\lambda(t)}{dt}\frac{dx(t)}{d\theta} - \int_{t_0}^{T} \lambda(t)\left[\frac{\partial f_2}{\partial x}\frac{\partial x}{\partial \theta} + \frac{\partial f_2}{\partial \theta}\right] dt \\
&= \frac{dx(T)}{d\theta} + \lambda(T)\frac{dx(T)}{d\theta} - \int_0^{t_0} \frac{d\lambda(t)}{dt}\frac{dx(t)}{d\theta} - \int_{t_0}^{T} \frac{d\lambda(t)}{dt}\frac{dx(t)}{d\theta} \\
& - \int_0^{t_0} \lambda(t)\left[\frac{\partial f_1}{\partial x}\frac{\partial x}{\partial \theta} + \frac{\partial f_1}{\partial \theta}\right] dt - \int_{t_0}^{T} \lambda(t)\left[\frac{\partial f_2}{\partial x}\frac{\partial x}{\partial \theta} + \frac{\partial f_2}{\partial \theta}\right] dt \\
&= \left[1 + \lambda(T)\right]\frac{dx(T)}{d\theta} - \int_0^{t_0} \frac{dx}{d\theta}\left[\frac{\lambda(t)}{dt} + \lambda(t)\frac{\partial f_1}{\partial x}\right] dt - \int_0^{t_0} \lambda(t)\frac{\partial f_1}{\partial \theta} dt \\
& - \int_{t_0}^{T} \frac{dx}{d\theta}\left[\frac{\lambda(t)}{dt} + \lambda(t)\frac{\partial f_2}{\partial x}\right] dt - \int_{t_0}^{T} \lambda(t)\frac{\partial f_2}{\partial \theta} dt
\end{aligned}
$$

So if we choose $\lambda(t)$ such that

$$
\begin{aligned}
\lambda(T) &= -1 \\
\frac{d\lambda(t)}{dt} &= \begin{cases} -\lambda(t)\frac{\partial f_1}{\partial x} & 0 \le t \le t_0 \\ -\lambda(t)\frac{\partial f_2}{\partial x} & t > t_0 \end{cases}
\end{aligned}
$$

Then we get,

$$
\frac{dF}{d\theta} = \frac{dx(T)}{d\theta} = -\int_0^{t_0} \lambda(t)\frac{\partial f_1(x,t;\theta)}{\partial \theta} dt - \int_{t_0}^{T} \lambda(t)\frac{\partial f_2(x,t;\theta)}{\partial \theta} dt
$$

This equation gives the sensitivity at any time T.

2.5 Comparing the two methods

Let $x \in \mathbb{R}^{n_1}$ and $\theta \in \mathbb{R}^{n_2}$. Given below is the comparison between the Forward method and Backward/Adjoint method.

Forward	**Backward**
$\frac{da(t)}{dt} = \frac{\partial f}{\partial x}\frac{\partial x}{\partial \theta} + \frac{\partial f}{\partial \theta}$	$\frac{dY(t)}{dt} = \begin{bmatrix} \lambda(t)\frac{df}{dx} \\ \lambda(t)\frac{df}{d\theta} \end{bmatrix}$
Requires one forward pass	Requires one forward pass and one backward pass
$2{*}n_1(n_1{*}n_2{+}n_2)$ operations in one Euler step	$3 * n_1(n_1 + n_2)$ operations
Preferable when we need sensitivity at different time steps simulatneosly	Preferable for larger n_2, i.e large number of parameters.

Simulation 1

Let,

$$\frac{dx(t)}{dt} = \begin{cases} p_1 x(t) + p_2 & 0 \leq t \leq t_0 \\ -p_2 x(t) + p_1 & t > t_0 \end{cases}$$

Computational Details :

Initial time t0 = 0

Final time t1 = 1.1

Initial state :

$x(0) = 1$

Point of discontinuity of $\frac{dx(t)}{dt}$ at $t_0 = 1$

Parameters:

p_1, p_2=$1, 3$

Change in parameter (for direct computation) = 0.000001

Time step =0.01

ODE solver = Euler

Graph:

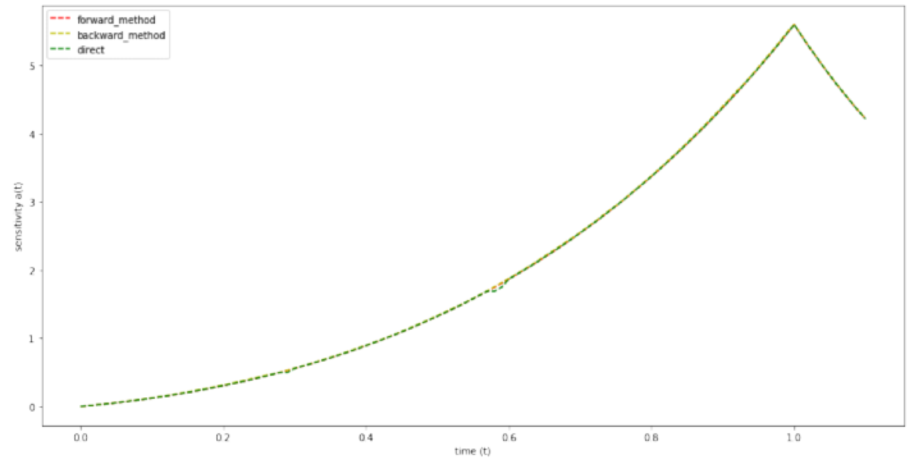


Figure 2.1: **Green**,**Red** and **Yellow** dotted-lines represents the sensitivity ($\frac{dx(t)}{dp_1}$) obtained by **direct computation**, **Forward** sensitivity method and **Backward** sensitivity method respectively. The graph shows that sensitivity obtained by all the methods are close to each other.

The table below compares the sensitivity obtained for different parameter values(p_1) at time $t = 1.1$.

Parameter Value (p_1)	Direct computation	Forward Method	Adjoint Method
1	2.2405	2.2405	2.2441
3	9.3687	9.3687	9.3935
4	12.3212	12.3212	9.3271

Simulation 2

Consider the following differential equation with a intital condition.

$$\begin{aligned} \frac{dx(t)}{dt} &= 1 - px(t) \\ x(0) &= 0 \end{aligned}$$

Computational Details :

Initial time t0 $= 0$

Final time t1 $= 1.1$

Initial state :

$x(0) = 0$

Point of discontinuity of $\frac{dx(t)}{dt}$ at $t_0 = 1$

Change in parameter (for direct computation) = 0.000001

Time step =0.01

ODE solver = Euler

Graph :

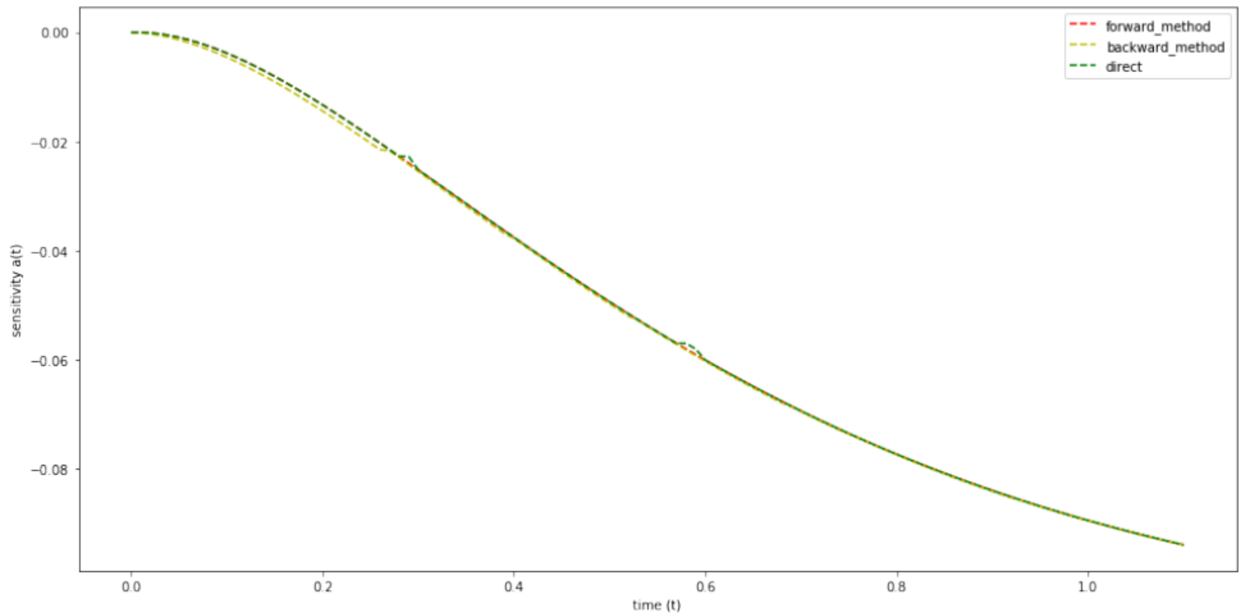


Figure 2.2: **Green**,**Red** and **Yellow** dotted-lines represents the sensitivity ($\frac{dx(t)}{da}$) obtained by **direct computation**, **Forward** sensitivity method and **Backward** sensitivity method respectively. The graph shows that all the methods concur with each other.

The below table compares the sensitivity obtained for different values of the parameter (p_1) at time $t = 1.1$.

Parameter Value (p)	Direct computation	Forward Method	Adjoint Method
1	-0.3011	-0.3011	-0.3016
3	-0.0939	-0.0939	-0.0940
5	-0.00621	-0.00621	-0.00623

Chapter 3

Neural ODE's

3.1 Using Single ReLu/ Strang-1 in a differential equation

ReLu is defined as

$$f(x) = \begin{cases} x & x > 0 \\ 0 & x \leq 0 \end{cases}$$

Let, the parameters be a and b.

$$\frac{dx(t)}{dt} = ReLu(ax(t) + b) \tag{3.1}$$

$$= Strang1(x; a, b) \tag{3.2}$$

$$= \begin{cases} ax(t) + b & ax + b > 0 \\ 0 & ax + b \leq 0 \end{cases} \tag{3.3}$$

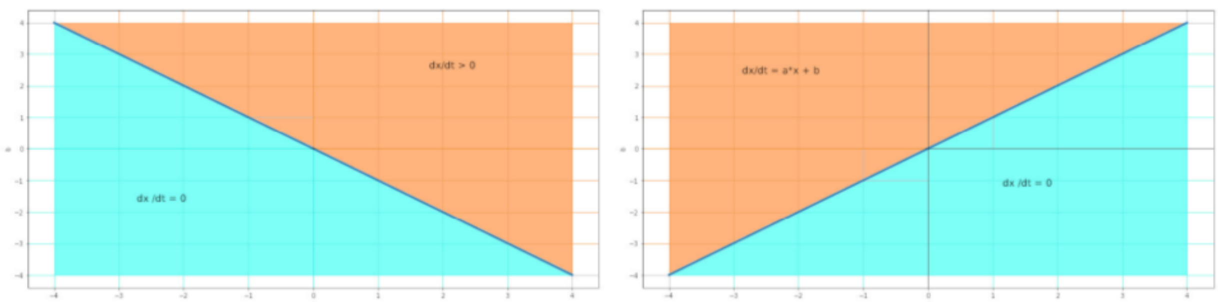
For a given initial condition x_0 we can divide the parameters space (ab plane) into regions, where $\frac{dx}{dt}$ will be zero and non-zero, these regions are Region I : $b \leq -ax_0$ and Region II : $b > -ax_0$ respectively.

So for a given x_0, if we pick a, b from Region II, then the solution of the differential eqution 3.3, i.e $x(t)$ will change with respect to time, and if we pick a, b from Region I, then $x(t)$ will remain constant w.r.t time.

So sensitvity ($\frac{dx(t)}{da}$ or $\frac{dx(t)}{db}$) will be non-zero in Region II and zero in Region I.

3.1.1 Parameter Space

The parameter space is divided into two regions. The boundary of the region depends on the initial value of x_0. The solution of an equation 3.3 i.e., $x(t)$ is sensitive to the parameters selected from one region, and it is not sensitive to the parameter selected from the other region. We define the regions in the parameter space, where any change in the parameter does not bring any change in the solution as **non-admissible region**. And the region where a change in parameter brings in a change to the solution will be called as **admissible region**.



(a) (b)

Figure 3.1: Figure (a) and (b) corresponds to $x_0 = 1$ and $x_0 = -1$ respectively.The **orange** region represents the region where any change in the parameters is reflected in the solution of the equation 3.3 i.e $x(t)$ (**admissible region**); The **blue** region represents the region where changes in the parameters are not reflected in the solution (**non-admissible region**). The **blue line** represents the boundary givn by the equation $ax_0 + b = 0$.

3.1.2 Sensitivity : Forward Method

Let $x(t)$ be the solution of the differential equation 3.3, and the sensitivity be denoted as $s(t) = \frac{dx(t)}{da}$. Solving the following IVP gives the sensitivty at any time t.

$$\frac{ds(t)}{dt} = \frac{\partial f}{\partial x}\frac{\partial x}{\partial a} + \frac{\partial f}{\partial a} \tag{3.1}$$

$$= \begin{cases} as(t) + x(t) & ax + b > 0 \\ 0 & ax + b \leq 0 \end{cases} \tag{3.2}$$

$$s(0) = 0 \tag{3.3}$$

3.1.3 Sensitivity : Backward Method

For the **backward sensitivity** method, following is the IVP.

$$\lambda(T) = -1 \tag{3.1}$$

$$\frac{d\lambda(t)}{dt} = \begin{cases} -\lambda(t)a & ax+b>0 \\ 0 & otherwise \end{cases} \tag{3.2}$$

Solving the IVP, helps in computing the sensitivity, at time 'T'.

$$\frac{dx(T)}{da} = -\int_{t_0}^{T} \lambda(t)\frac{\partial f}{\partial a}dt \tag{3.3}$$

3.2 Composition of two ReLu/Strang2

Let the differential equation be

$$\begin{aligned} \frac{dx(t)}{dt} &= Relu(aReLu(ax(t)+b)+b) \\ &= Strang2(x;a,b) \\ &= \begin{cases} a^2x(t)+ab+b & ax+b>0; a^2x+ab+b>0 \\ b & ax+b\leq 0; b>0 \\ 0 & otherwise \end{cases} \end{aligned}$$

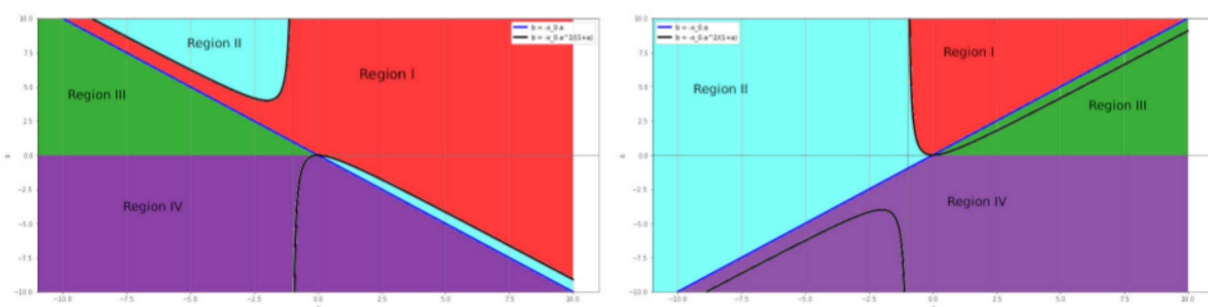
So the Initial Value Problem is given as

$$x(0) = x_0 \tag{3.4}$$

$$\frac{dx(t)}{dt} = \begin{cases} a^2x(t)+ab+b & ax+b>0; a^2x+ab+b>0 \\ b & ax+b\leq 0; b>0 \\ 0 & otherwise \end{cases} \tag{3.5}$$

3.2.1 Parameter Space

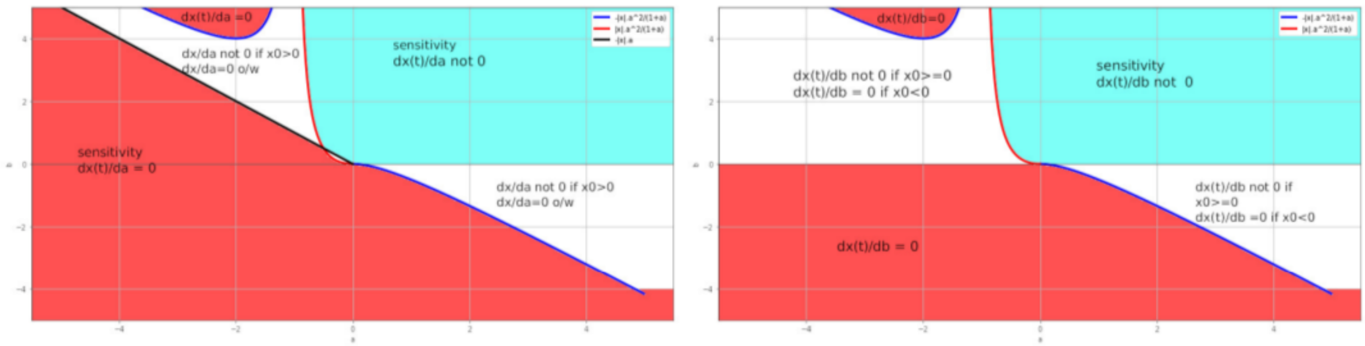
The figures below show how the ab space is divided according to the output of Strang-2 when the sign of x is positive and negative.



(a) (b)

Figure 3.2: Figure (a) and (b) corresponds to $x \geq 0$ and $x \leq 0$ respectively. The ab space is divided into different regions based on the output of Strang2 for a given x. The black boundary is given by the equation $a^2x + ab + b = 0$ and the blue boundary is given by the equation $ax + b = 0$.
The different colors corresponds to different outputs of Strang2. Let $z = Strang2(x; a, b)$, then Red region: $z = a^2x + ab + b$; Blue region: $z = 0$; Green region: $z = b$; Violet region: $z = 0$.

The figures below show how the ab space is divided based on the sensitivity of the solution of equation 3.4 and 3.5, with respect to both the parameters a and b, when the initial value x_0 is greater than zero.



(a) (b)

Figure 3.3: **Figure** (a) **and** (b) **corresponds to sensitivity** $\frac{dx}{da}$ **and** $\frac{dx}{db}$ **respectively**. The **red region** corresponds to the region where irrespective of sign of x_0,if parameters are chosen from this region the **sensitivity is zero**. The **blue region** corresponds to the region where irrespective of sign of x_0,if the parameters are chosen from this region the **sensitivity is non-zero**. The **White color** region corresponds to the region where, if the parameters are chosen the **sensitivity may be zero or non zero** depending upon the sign of x_0.
The red boundary corresponds to the equation $-a|x_0|^2 + (1+a)b = 0$; blue boundary corresponds to the equation $a|x_0|^2 + (1+a)b = 0$; Black boundary corresponds to the equation $a|x_0| + b = 0$.

3.2.2 Sensitivity

Consider the Ordinary Differential Equation

$$\begin{aligned}\frac{dx(t)}{dt} &= Relu(aReLu(ax(t)+b)+b)\\ &= f(x;a,b)\end{aligned}$$

3.2.3 Computing the senstivity w.r.t parameter a i.e $\frac{dx}{da}$

Take

$$\frac{dx(t)}{da} = s(t)$$

then the sensitivity equation for the **forward sensitivity** method is as follows:

$$\frac{ds(t)}{dt} = \frac{df}{da} \tag{3.6}$$

$$= \frac{\partial f}{\partial x(t)}\frac{\partial x(t)}{\partial a} + \frac{\partial f}{\partial a} \tag{3.7}$$

$$= \begin{cases} a^2 s(t) + b + 2ax(t) & ax + b > 0; a^2x + ab + b > 0 \\ 0 & otherwise \end{cases} \tag{3.8}$$

with the inital condition $s(t_0) = 0$.
Solving equation 3.8 with the intial condition gives us the sensitivity at any desired time 't'.

Similarly for the **backward sensitivity** method, following are the equations.

$$\lambda(T) = -1 \tag{3.9}$$

$$\frac{d\lambda(t)}{dt} = -\lambda(t)\frac{\partial f}{\partial x(t)} \tag{3.10}$$

$$= \begin{cases} -\lambda(t)a^2 & ax + b > 0; a^2x + ab + b > 0 \\ 0 & otherwise \end{cases} \tag{3.11}$$

$$s(T) = \frac{dx(T)}{da} \tag{3.12}$$

$$= -\int_{t_0}^{T} \lambda(t)\frac{\partial f}{\partial a} \tag{3.13}$$

Here

$$\frac{\partial f}{\partial a} = \begin{cases} 2ax(t) + b & a^2x + ab + b > 0 \\ 0 & otherwise \end{cases}$$

Solving equations 3.9 to 3.11 gives $\lambda(t)$. Using this in solving equation 3.13 gives sensitivity $s(t)$ i.e $\frac{dx(t)}{da}$ at any time 'T'.

3.2.4 Simulations

In these simulations we pick parameters from the different regions (Blue, Red and White) as shown in Figure 3.5 and use the sensitivity equations derived earlier to plot the sensitivity vs time graphs. We compare the sensitivity obtained by forward method, backward method and direct computation.

Case I :Parameter taken from Blue Region:

Let, $a = 1.4$, $b = 1$ and $x_0 = -1$, then we have

$$\frac{dx}{dt} = ReLu(1.4ReLu(1.4x + 1) + 1)$$

.

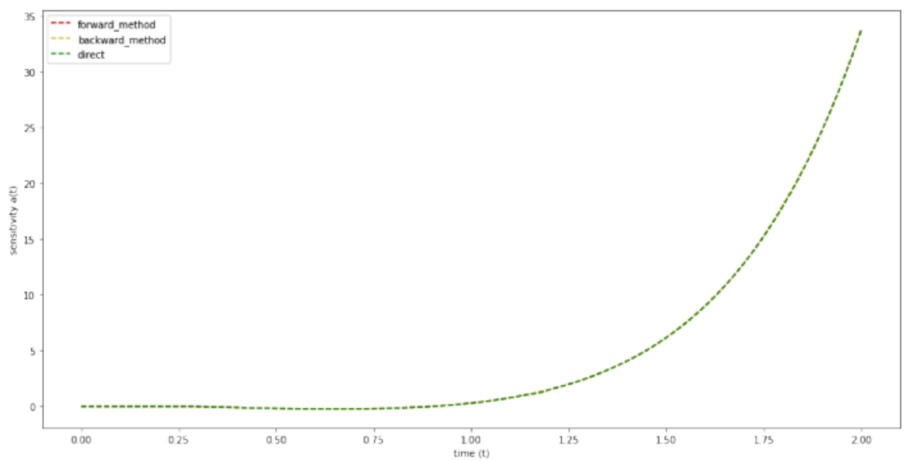


Figure 3.4: **Green**,**Red** and **Yellow** dotted-lines represents the sensitivity ($\frac{dx(t)}{da}$) obtained by **direct computation**, **Forward** sensitivity method and **Backward** sensitivity method respectively. As the **parameters were taken from the admissible region hence the sensitivity is non-zero**.

Case II :Parameter taken from Red Region:

Let, $a = -1.1$, $b = -0.4$ and $x_0 = 1$, then we have

$$\frac{dx}{dt} = ReLu(-1.1ReLu(-1.1x - 0.4) - 0.4)$$

. Here (a, b) is in non-admissible region. Here the intial at $t = 0$, $\frac{dx(t)}{dt} = 0$, so the $x(t)$ doesnot change with respect to t.

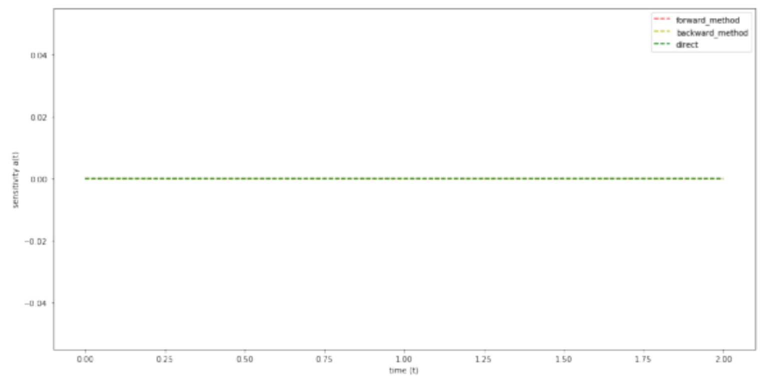


Figure 3.5: **Green**,**Red** and **Yellow** dotted-lines represents the sensitivity ($\frac{dx(t)}{da}$) obtained by **direct computation**, **Forward** sensitivity method and **Backward** sensitivity method respectively. The graph shows that **sensitivity is zero as the parameters were chosen from non-admissible region**.

Case III :Parameter taken from White Region with positive x_0:

Let,$a = -0.8$, $b = 1.2$ and $x_0 = 1.$, then we have

$$\frac{dx}{dt} = ReLu(-0.8ReLu(-0.8x + 1.2) + 1.2)$$

.

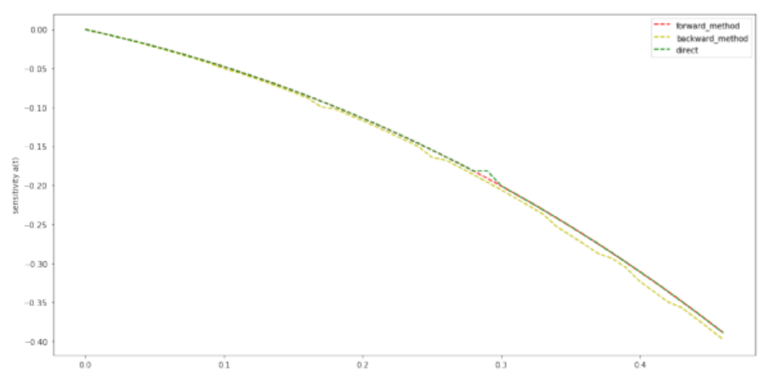


Figure 3.6: **Green**,**Red** and **Yellow** dotted-lines represents the sensitivity ($\frac{dx(t)}{da}$) obtained by **direct computation**, **Forward** sensitivity method and **Backward** sensitivity method respectively. The graph shows that **sensitivity is non-zero as** $x_0 \geq 0$.

Case IV :Parameter taken from White Region with negative x_0:

Let,$a = -0.8$, $b = 1.2$ and $x_0 = -1.$, then we have

$$\frac{dx}{dt} = ReLu(-0.8ReLu(-0.8x + 1.2) + 1.2)$$

.

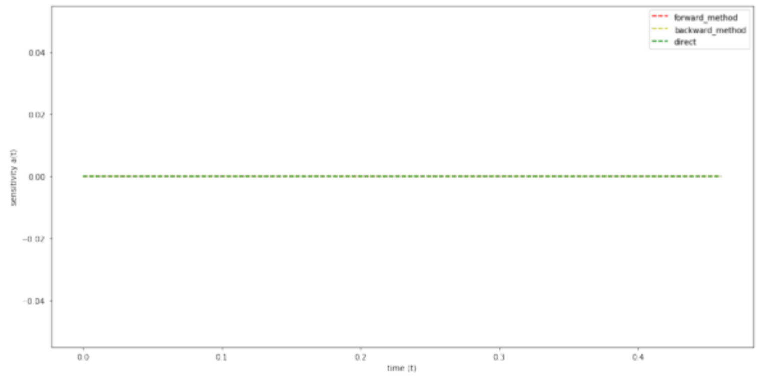


Figure 3.7: **Green**,**Red** and **Yellow** dotted-lines represents the sensitivity ($\frac{dx(t)}{da}$) obtained by **direct computation**, **Forward** sensitivity method and **Backward** sensitivity method respectively. The graph shows that **sensitivity is zero as** $x_0 \leq 0$.

3.3 Parameter Space

In this chapter we analytically define the boundaries that divide ab space into different regions where the velocity vector (Strang-n; from the ODE $\frac{dx}{dt}$ =Strang-n) takes different values. In particular define boundaries analytically that divide the ab space into regions where there is **motion** (velocity non-zero) and **no-motion** (velocity zero).

Proofs of some statements to be used later

Lemma I: For a even natural number n and $a \in \mathbb{R}$,

$$1 + a + .. + a^n > 0 \forall a \in \mathbb{R}$$

.

Proof:

If $a > 0$, then its obvious that $1 + a + .. + a^n > 0 \forall a \in \mathbb{R}$

If $a < 0$, then we have two cases.

Case I: $a < -1$

Here we can pair up the terms, $1 + (a + a^2) + (a^3 + a^4) + ... + (a^{n-2} + a^{n-1}) + a^n$, here each pair is greater than 0, hence the overall sum is greater than 0.

Case II: $-1 < a < 0$

Here we can write the summation as, $(1 + a) + (a^2 + a^3) + ... + (a^{n-1} + a^n)$, here each pair is greater than zero, hence the overall sum is also greater than 0.

Lemma II: For a odd natural number n and $a \in \mathbb{R}$,

$$1 + a + .. + a^n > 0 \forall a > -1$$

and

$$1 + a + .. + a^n < 0 \forall a < -1$$

.

Proof:

Take $y(a) = 1 + a + ... + a^n$, then we have,

$$\frac{dy}{da} = 1 + a + .. + a^{n-1} > 0 \forall a \in \mathbb{R}$$

Hence y i a monotonic function.

We also have $y(-1) = 0, y(-2) < 0, y(0) > 0$, Hence we can conclude that for $a < -1, y(a) < 0$ and for $a > -1, y(a) > 0$.

Lemma III: For $(a, b) \in \{(a, b) : x_0a^2 + (1 + a)b > 0, a > -1, b > 0\}$, we have

$$x_0a^n + (\sum_{i=0}^{i=n-1} a^i)b > 0$$

for n-even natural number.

Proof:

$$x_0a^2 + (1 + a)b > 0 \Rightarrow a^{n-2}(x_0a^2 + (1 + a)b) > 0 \Rightarrow x_0a^n + (a^{n-2} + a^{n-1})b > 0 \qquad (3.1)$$

Since $a > -1$, hence as we have proved earlier, we get,

$$1 + a + .. + a^{n-3} > 0 \Rightarrow (1 + a + .. + a^{n-3})b > 0 \quad (3.2)$$

So adding equation 1 and 2, we get $x_0 a^n + (\sum_{i=0}^{i=n-1} a^i)b > 0$.

3.3.1 Distribution of Region for Strang-n

Case I: $x_0 > 0$ and n a even number

<u>**P(n):**</u> Let $z = R^n(ax_0 + b)$. For $x_0 > 0$, and n **a even number**. a-b space is divided into four regions, they are:

Region I: $\{(a, b) : b < \frac{-x_0 a^n}{\sum_{i=0}^{i=n-1} a^i}\} \cup \{(a, b) : b < 0, a < 0\}$, here $z = 0$

Region II: $\{(a, b) : b < \frac{-x_0 a^{n-1}}{\sum_{i=0}^{i=n-1} a^i}, a < -1, b > 0\}$, here $z = b$

Region III: $\{(a, b) : b < -x_0 a, -1 < a < 0, b > 0\}$,here $z = \sum_{i=0}^{i=n-2} a^i$

Region IV: a-b space - ($I \cup II \cup III$), here $z = a^n x_0 + (1 + a + .. + a^{n-1})b$

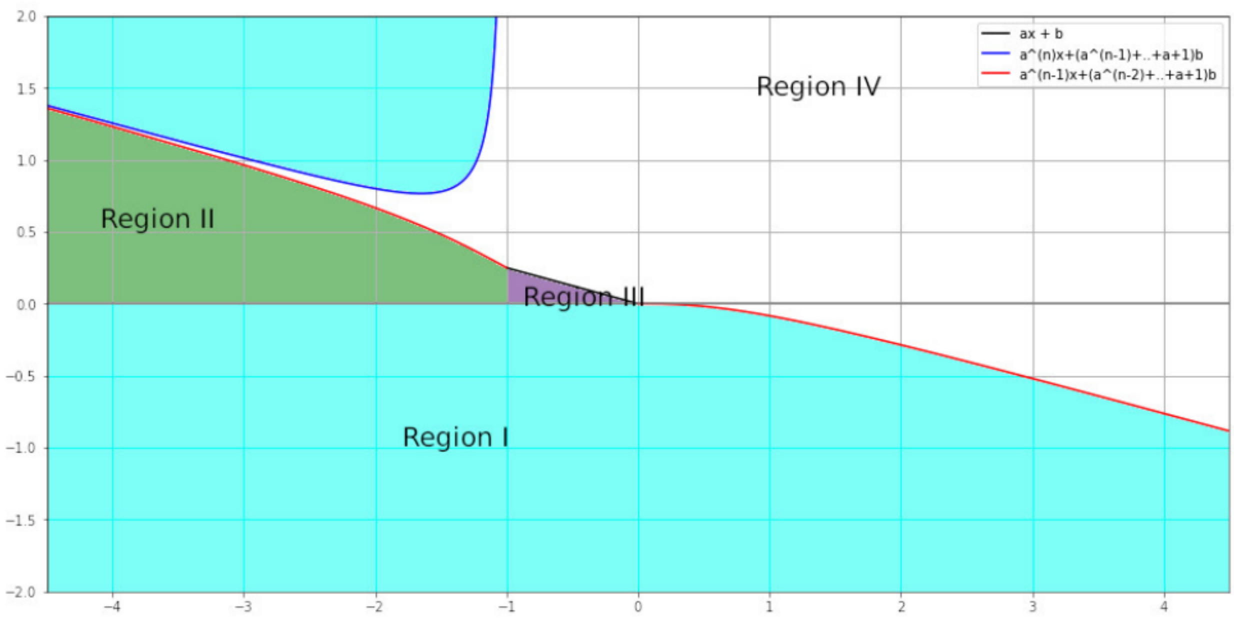


Figure 3.8: Let $z = R^n(ax_0 + b)$; Region I: $z = 0$; Region II: $z = b$; Region III: $z = 1 + a + .. + a^{n-2}$; Region IV: $z = a^n x_0 + (1 + a + .. + a^{n-1})b$
The **red boundary** correspond to the equation $a^{n-1}x + (a^{n-2} + a^{n-3} + ... + a + 1)b = 0$; The **blue boundary** corresponds to the equation $a^n x + (a^{n-1} + ... + a + 1)b = 0$; The **black boundary** corresponds to the equation $ax + b = 0$

So, the regions of no-motion is given by $A \cup B \cup C$. Where $A, B\&C$ are given by:
$A = \{(a, b) : x_0 a^n + (\sum_{i=0}^{i=n-1} a^i)b < 0, b < 0\}$
$B = \{(a, b) : b < 0, a < 0\}$
$C = \{(a, b) : x_0 a^n + (\sum_{i=0}^{i=n-1} a^i)b < 0, b > 0\}$

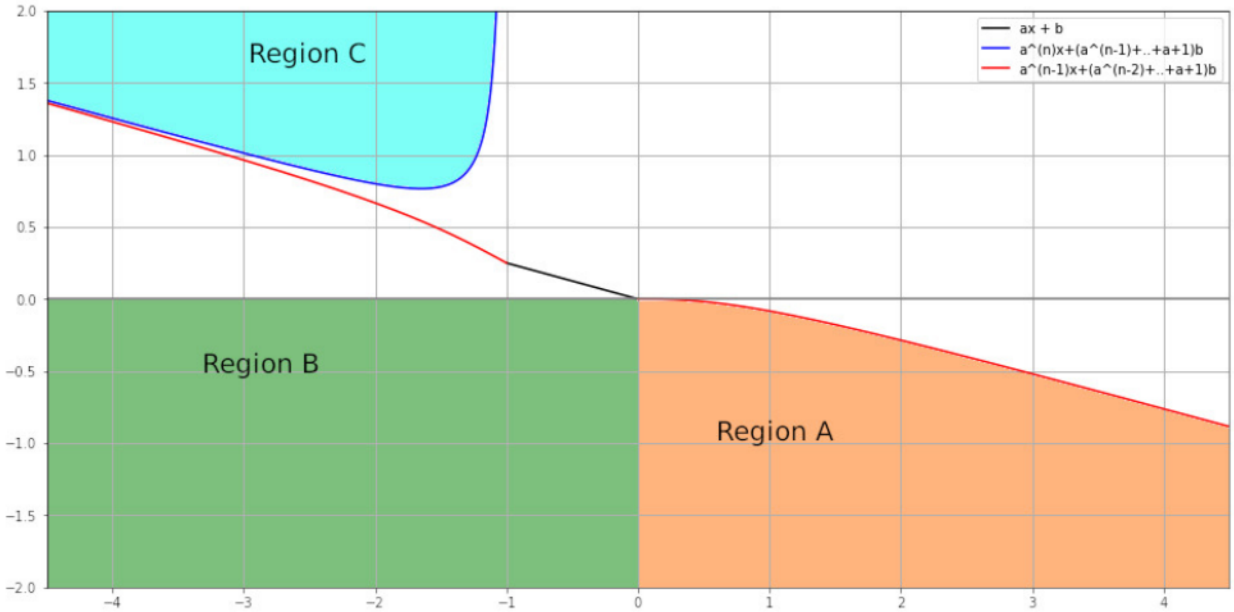


Figure 3.9: Let $z = R^n(ax_0 + b)$; Region A, B, C represents the region where $z = 0$. The **red boundary** correspond to the equation $a^{n-1}x + (a^{n-2} + a^{n-3} + ... + a + 1)b = 0$; The **blue boundary** corresponds to the equation $a^n x + (a^{n-1} + ... + a + 1)b = 0$; The **black boundary** corresponds to the equation $ax + b = 0$.

Proof:

For **P(k=2)**, we have already verified that the a-b space is divided into four regions,where the regions with no motion is given by $A \cup B \cup C$. Where $A, B\&C$ are given by:
A: $b < \frac{-x_0 a^4}{\sum_{i=0}^{i=3} a^i}, b > 0$, B: $b < \frac{-x_0 a^4}{\sum_{i=0}^{i=3} a^i}, b < 0$ and C: $b < 0, a < 0$.

Lets assume **P(k=n)** is true.
To prove **P(k=n+2)** is true.
For Strang-n, the a-b space is divided into four regions as given in figure 3.11.

Region I: $\{(a, b) : b < \frac{-x_0 a^n}{\sum_{i=0}^{i=n-1} a^i}\} \cup \{(a, b) : b < 0, a < 0\}$
Here $z = 0$
So

$$R(aR(az + b) + b) = R(aR(b) + b) = R(b) = 0$$

Region II: $\{(a,b) : b < \frac{-x_0 a^{n-1}}{\sum_{i=0}^{i=n-1} a^i}, a < -1, b > 0\}$
Here $z = b$
So

$$R(aR(az+b)+b) = R(aR(ab+b)+b) = R(b) = b$$

Region III: $\{(a,b) : b < -x_0 a, -1 < a < 0, b > 0\}$
Here $z = (1 + a + a^2 + ... + a^{n-1)})b$
So $R(aR(az+b)+b) = R(aR((1 + a + a^2 + .. + a^{n-1})b) + b)$
Here $1 + a + a^2 + .. + a^n > 0 \forall n$ for $a > -1$, (From Lemma I), hence

$$R(aR(az+b)+b) = R((1 + a + a^2 + ... + a^{n+1})b) = (1 + a + a^2 + ... + a^{n+1})b$$

Region IV: all region except I, II and III
Here $z = a^n x_0 + (1 + a + ... + a^{n-1})b$
So

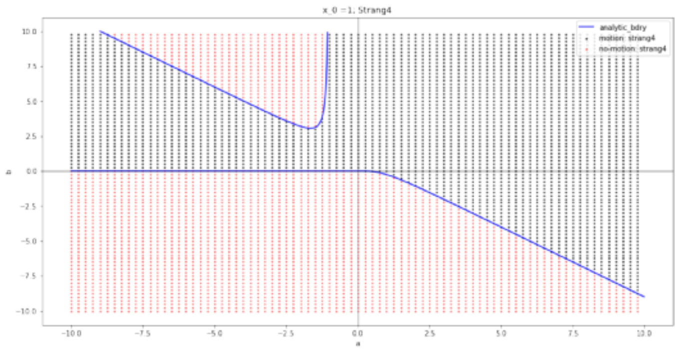
$$R(aR(az+b)+b) = \begin{cases} a^{n+2}x_0 + (\sum_{i=0}^{i=n+1} a^i)b & x_0 a^{n+2} + (\sum_{i=0}^{i=n+1} a^i)b > 0, x_0 a^{n+1} + (\sum_{i=0}^{i=n} a^i)b > 0 \\ b & x_0 a^{n+1} + (\sum_{i=0}^{i=n} a^i)b < 0, b > 0 \\ 0 & x_0 a^{n+2} + (\sum_{i=0}^{i=n+1} a^i)b < 0 \end{cases}$$

So the region where there is no-motion is given by
Region I $\cup$ (Region IV $\cap \{(a,b) : x_0 a^{n+2} + (\sum_{i=0}^{i=n+1} a^i)b < 0\}$), that is
$A \cup B \cup C$ where $A, B, \& C$ are given by:
$A = \{(a,b) : -x_0 a^{n+2} + (\sum_{i=0}^{i=n+1} a^i)b < 0, b < 0\}$
$B = \{(a,b) : b < 0, a < 0\}$
$C = \{(a,b) : -x_0 a^{n+2} + (\sum_{i=0}^{i=n+1} a^i)b < 0, b > 0\}$.

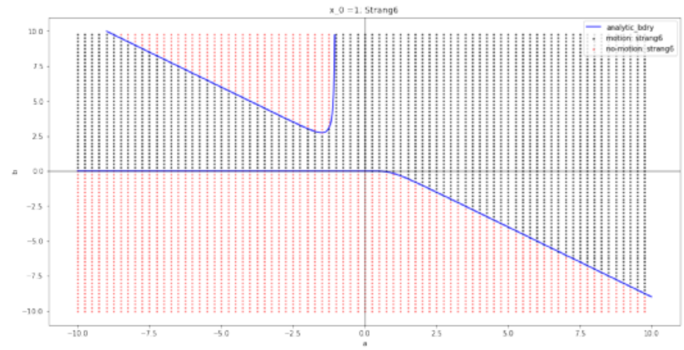
Verifying the boundaries through simulations:

The boundary is given by $b = \frac{-x_0 a^n}{\sum_{i=0}^{n-1} a^i}$ for $a > 0$ and $b = \frac{-x_0 a^n}{\sum_{i=0}^{n-1} a^i} \& b = 0$ for $a < 0$ in case of $n - even$.
The blue line represents the boundary. Parameters were taken from all over the parameter space and corresponding to each parameter, it was checked whether the velocity vector given

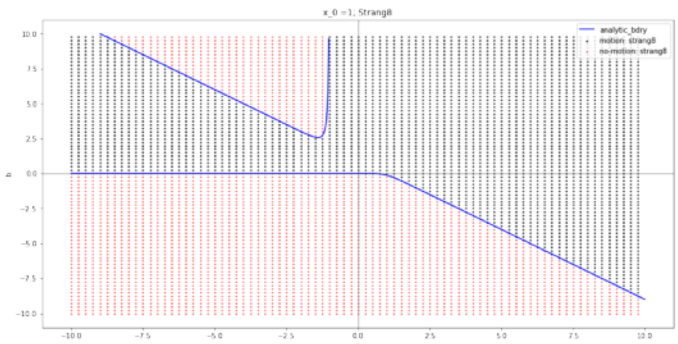
by $Strang - n(x; a, b)$ is zero or not. If it is zero we call there is **no-motion** and if it is non-zero we say there is **motion**.



(a) Strang-4



(b) Strang-6



(c) Strang-8

Figure 3.10: Parameters taken from all over the ab space. Corresponding to each parameter if the velocity vector given by $Strang - n(x; a, b)$ is noted. **Red** points represents those parameters where **velocity vector is zero**. And **black** points represents those parameters wherer **velocity vector is non-zero**.

Here it can be seen that the boundary given analytically in the above section coincides with the boundary that we get through simulation.

Case II: $x_0 > 0$ and n a odd number

P(n): Let $z = R^n(ax_0 + b)$. For $x_0 > 0$, and n **a odd number**. a-b space is divided into four regions which are:

Region I: $\{(a,b) : b < \frac{-x_0 a^n}{\sum_{i=0}^{i=n-1} a^i}, b < 0\} \cup \{(a,b) : b < 0, a < 0\} \cup \{(a,b) : b < \frac{-x_0 a^n}{\sum_{i=0}^{i=n-1} a^i}, b > 0, a < -1\}$, here $z = 0$.

Region II: $\{(a,b) : b < \frac{-x_0 a^{n-1}}{\sum_{i=0}^{i=n-1} a^i}, a < -1, b > 0\}$, here $z = b$

Region III: $\{(a,b) : b < -x_0 a, -1 < a < 0, b > 0\}$,here $z = 1 + a + .. + a^{n-2}$

Region IV: a-b space - $I \cup II \cup III$, here $z = a^n x_0 + (1 + a + .. + a^{n-1})b$

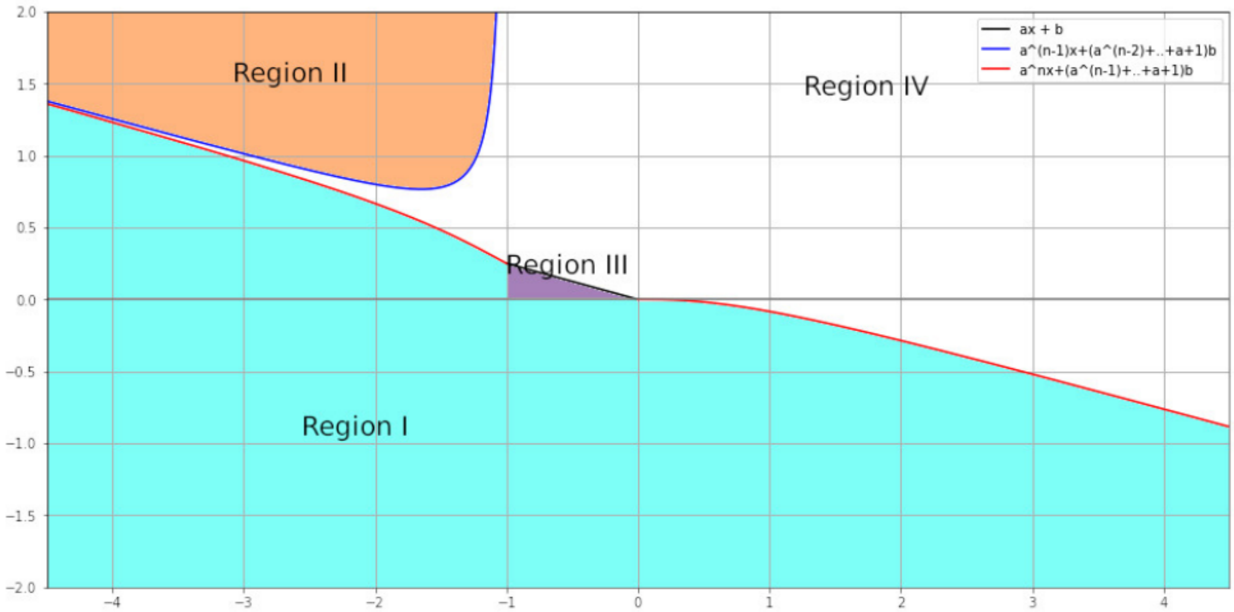


Figure 3.11: Let $z = R^n(ax_0 + b)$; Region I: $z = 0$; Region II: $z = b$; Region III: $z = 1 + a + .. + a^{n-2}$; Region IV: $z = a^n x_0 + (1 + a + .. + a^{n-1})b$
The **red boundary** correspond to the equation $a^n x + (a^{n-1} + a^{n-2} + ... + a + 1)b = 0$; The **blue boundary** corresponds to the equation $a^{n-1}x + (a^{n-2} + ... + a + 1)b = 0$; The **black boundary** corresponds to the equation $ax + b = 0$

So, the regions where there is no-motion is given by $A \cup B \cup C$. Where $A, B\&C$ are given by:

$A = \{(a,b) : x_0 a^n + (\sum_{i=0}^{i=n-1} a^i)b < 0, b < 0\}$
$B = \{(a,b) : b < 0, a < 0\}$
$C = \{(a,b) : x_0 a^n + (\sum_{i=0}^{i=n-1} a^i)b < 0, b > 0, a < -1\}$.

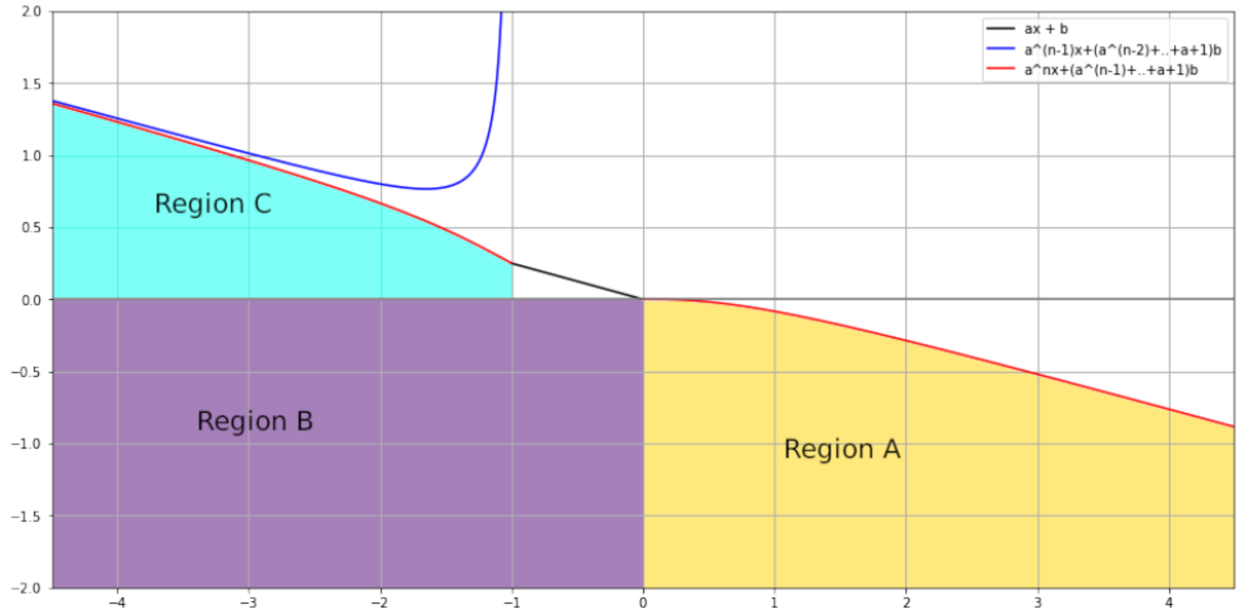


Figure 3.12: Let $z = R^n(ax_0 + b)$; Region A, B, C represents the region where $z = 0$

Proof:

For **P(n=3)**, we verified that the a-b space is divided into four regions, where the regions with no motion is given by $A \cup B \cup C$. Where $A, B \& C$ are given by:
A: $\{(a, b) : x_0a^3 + (\sum_{i=0}^{i=2} a^i)b < 0, b < 0\}$, B: $\{(a, b) : b < 0, a < 0\}$ and C: $\{(a, b) : x_0a^3 + (\sum_{i=0}^{i=2} a^i)b < 0, b > 0, a < -1\}$.

Lets assume **P(k=n)**, is true.
To prove **P(k=n+2)**, is true.
For Strang-n, the a-b space is divided into four regions as given in figure 3.14.

Region I: $\{(a, b) : b < \frac{-x_0a^n}{\sum_{i=0}^{i=n-1} a^i}, b < 0\} \cup \{(a, b) : b < 0, a < 0\} \cup \{(a, b) : b < \frac{-x_0a^n}{\sum_{i=0}^{i=n-1} a^i}, b > 0, a < -1\}$
Here $z = 0$
So

$$R(aR(az + b) + b) = R(aR(b) + b) = R(b) = 0$$

Region II: $\{(a,b) : b < \frac{-x_0 a^{n-1}}{\sum_{i=0}^{i=n-1} a^i}, a < -1, b > 0\}$
Here $z = b$
So

$$R(aR(az+b)+b) = R(aR(ab+b)+b) = R(b) = b$$

Region III: $\{(a,b) : b < -x_0 a, -1 < a < 0, b > 0\}$
Here $z = (1 + a + a^2 + ... + a^{n-1)})b$
So $R(aR(az+b)+b) = R(aR((1 + a + a^2 + .. + a^{n-1})b) + b)$
Here $1 + a + a^2 + .. + a^n > 0 \ \forall n$ for $a > -1$ (From Lemma II), hence

$$R(aR(az+b)+b) = R((1 + a + a^2 + ... + a^{n+1})b) = (1 + a + a^2 + ... + a^{n+1})b$$

Region IV: all region except I, II and III
Here $z = a^n x_0 + (1 + a + ... + a^{n-1})b$
So

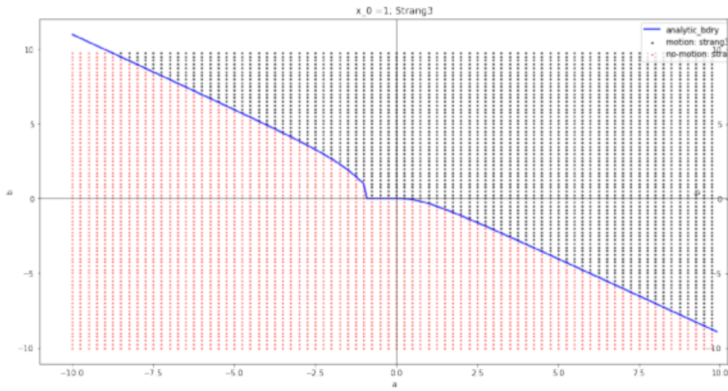
$$R(aR(az+b)+b) = \begin{cases} a^{n+2}x_0 + (\sum_{i=0}^{i=n+1} a^i)b & -x_0 a^{n+2} + (\sum_{i=0}^{i=n+1} a^i)b > 0, -x_0 a^{n+1} + (\sum_{i=0}^{i=n} a^i)b > 0 \\ b & -x_0 a^{n+1} + (\sum_{i=0}^{i=n} a^i)b < 0, b > 0 \\ 0 & -x_0 a^{n+2} + (\sum_{i=0}^{i=n+1} a^i)b < 0 \end{cases}$$

So the region where there is no-motion is given by
Region I $\cup$ (Region IV $\cap (a,b) : b < \frac{-x_0 a^{n+2}}{\sum_{i=0}^{i=n+1} a^i})$), that is
$A \cup B \cup C$ where $A, B, \& C$ are given by:
$A = \{(a,b) : x_0 a^{n+2} + (\sum_{i=0}^{i=n+1} a^i)b < 0, b < 0\}$
$B = \{(a,b) : b < 0, a < 0\}$
$C = \{(a,b) : x_0 a^{n+2} + (\sum_{i=0}^{i=n+1} a^i)b < 0, b > 0, a < -1\}$.

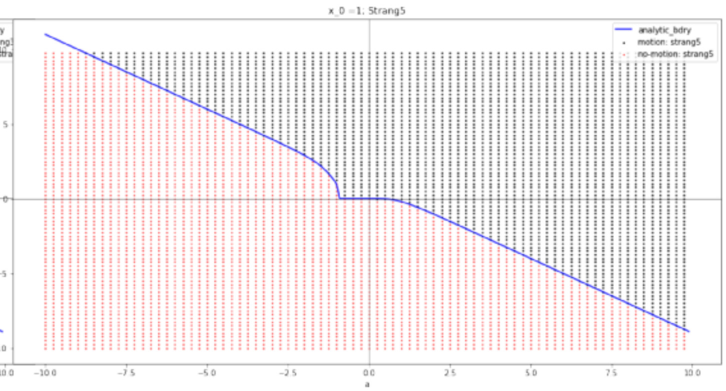
Verifying the boundaries through simulations:

The boundary is given by $b = \frac{-x_0 a^n}{\sum_{i=0}^{n-1} a^i}$ for $a < -1 \& a > 0$ and $b = 0$ for $-1 \leq a \leq 0$ in case of n-odd.

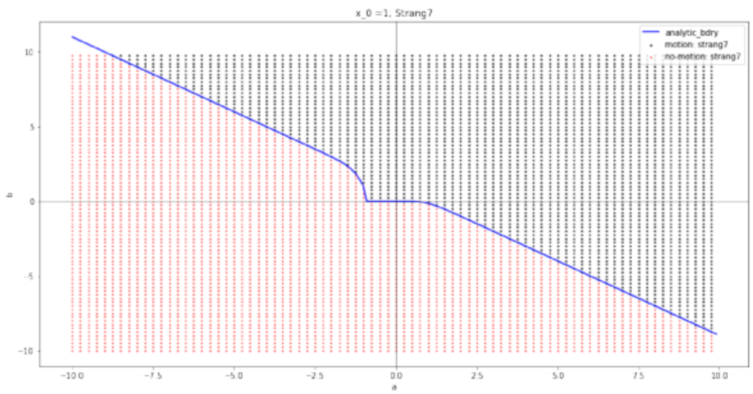
The blue line represents the boundary; the red dots represent no-motion and the black dots represents motion.



(a) Strang-3



(b) Strang-5



(c) Strang-7

Figure 3.13: Parameters taken from all over the ab space. Corresponding to each parameter if the velocity vector given by $Strang - n(x; a, b)$ is noted. **Red** points represents those parameters where **velocity vector is zero**. And **black** points represents those parameters wherer **velocity vector is non-zero**.

Here it can be seen that the boundary given analytically in the above section coincides with the boundary that we get through simulation.

Case III: $x_0 < 0$ **and** n **a odd number**

P(n):Let $z = R^n(ax_0 + b)$. For $x_0 < 0$, and n **a odd number**. a-b space is divided into five regions which are given by:

Region I: $\{(a, b) : b < 0\}$, here $z = 0$

Region II: $\{(a, b) : b < -x_0 a, b > 0\}$, here $z = 1 + a + .. + a^{n-2}$

Region III: $\{(a, b) : b > \frac{-x_0 a^2}{1+a}, b > -x_0 a\}$, here $z = a^n x_0 + (1 + a + .. + a^{n-1})b$

Region IV: $\{(a, b) : b < \frac{-x_0 a^2}{1+a}, a > -1, b > 0\}$,here $z = (1 + a + .. + a^{n-3})b$

Region V: $\{(a, b) : b > 0, a < -1\}$, here $z = b$

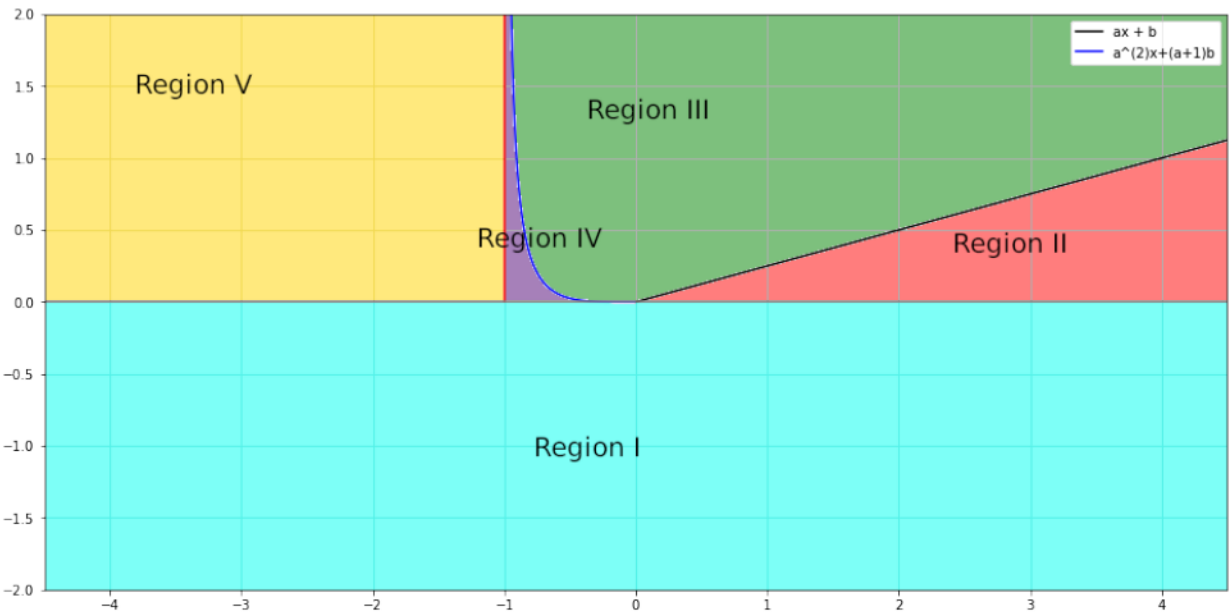


Figure 3.14: Let $z = R^n(ax_0 + b)$; Region I: $z = 0$; Region II: $z = 1 + a + .. + a^{n-2}$; Region III: $z = a^n x_0 + (1 + a + .. + a^{n-1})b$; Region IV: $z = (1 + a + .. + a^{n-3})b$; Region V: $z = b$ **Blue** boundary corresponds to the equation $a^2x + (1 + a)b$; **Black** boundary corresponds to the equation $ax + b = 0$; **Red** boundary corresponds to $a = -1$

That is the regions where there is no-motion is given by $A = \{(a, b) : b < 0\}$.

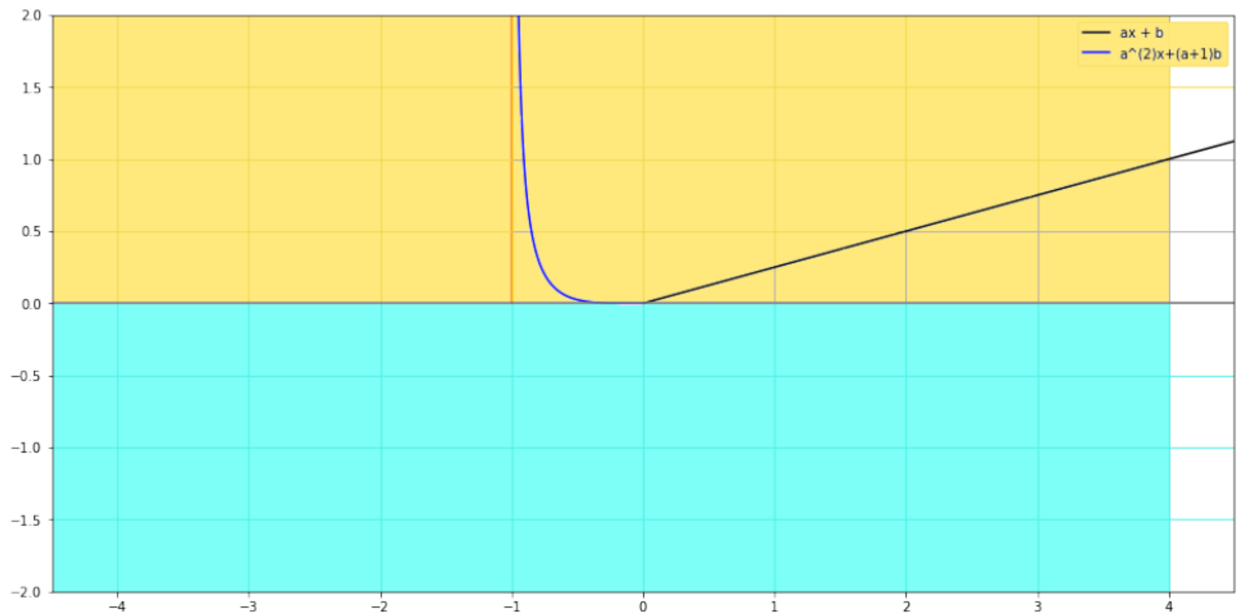


Figure 3.15: Let $z = R^n(ax_0 + b)$; Region A represents the region where $z = 0$
Blue boundary corresponds to the equation $a^2x + (1 + a)b$; **Black** boundary corresponds to the equation $ax + b = 0$; **Red** boundary corresponds to $a = -1$

Proof:

For **P(n=3)**, we verified that the a-b space is divided into five regions, where the region with no motion is given by $\{(a, b) : b < 0\}$.
Lets assume **P(k=n)** is true.
To prove **P(k=n+2)** is true.
For Strang-n, the a-b space is divided into five regions as given in figure 3.17.

Region I: $\{(a, b) : b < 0\}$
Here $z = 0$
So

$$R(aR(az + b) + b) = R(aR(b) + b) = R(b) = 0$$

Region II: $\{(a, b) : b < -x_0a, b > 0\}$
Here $z = (1 + a + a^2 + ... + a^{n-2)})b$
So $R(aR(az + b) + b) = R(aR((1 + a + a^2 + .. + a^{n-1})b) + b)$
Here $1 + a + a^2 + .. + a^{n-2} > 0 \; \forall n$ for $a > -1$, hence

$$R(aR(az+b)+b) = R((1+a+a^2+...+a^{n-1})b) = (1+a+a^2+...+a^n)b$$

Region III: $\{(a,b) : b > \frac{-x_0a^2}{1+a}, b > -x_0a\}$
Here $z = a^n x_0 + (1+a+...+a^{n-1})b$
So

$$R(aR(az+b)+b) = R(aR(a^n x_0 + (1+a+...+a^{n-1})b)+b)$$

In Region III, we have $x_0a^n + (\sum_{i=0}^{i=n-1} a^i)b > 0$ for $n \geq 2$ (From Lemma III)
Hence we get,

$$R(aR(az+b)+b) = R(aR(a^n x_0 + (1+a+...+a^{n-1})b)+b) = a^{n+2}x_0 + (1+a+...+a^{n+1})b$$

Region IV: $\{(a,b) : b < \frac{-x_0a^2}{1+a}, a > -1, b > 0\}$
Here $z = (1+a+...+a^{n-3})b$
Here $1+a+a^2+..+a^{n-2} > 0 \; \forall n$ for $a > -1$.
So

$$R(aR(az+b)+b) = R(aR((1+a+..+a^{n-2})b)+b) = (1+a+...+a^{n-2})b$$

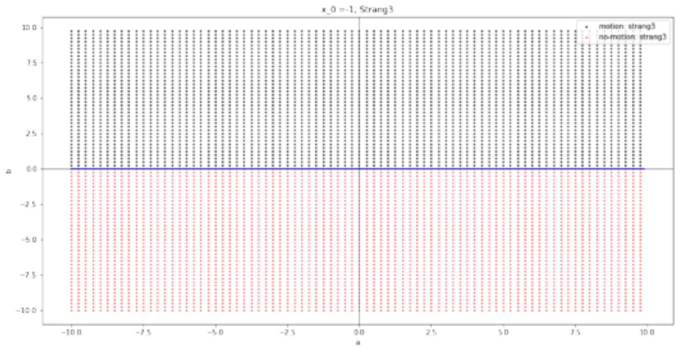
Region V: $\{(a,b) : b > 0, a < -1\}$
Here $z = b$
So

$$R(aR(az+b)+b) = R(aR(ab+b)+b) = R(b) = b$$

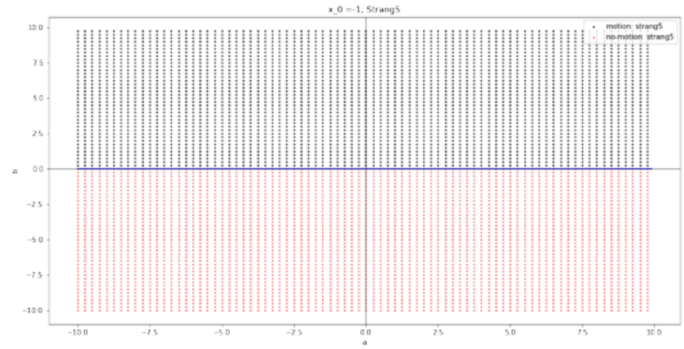
So the region where there is no-motion is given by $A = \{(a,b) : b < 0\}$.

Verifying the boundaries through simulations:

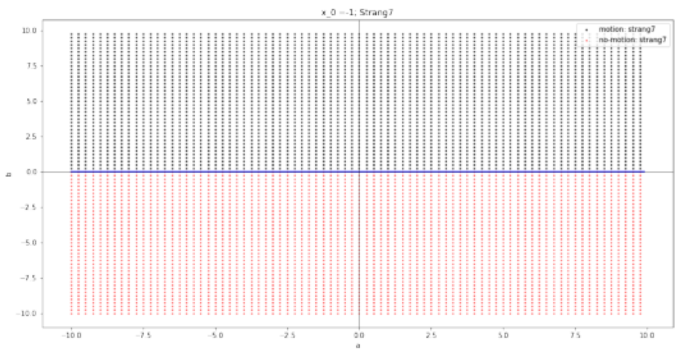
The boundary is given by $b = 0$ in case of n-odd.
The blue line represents the boundary; the red dots represent no-motion and the black dots represents motion.



(a) Strang-3



(b) Strang-5



(c) Strang-7

Figure 3.16: Parameters taken from all over the ab space. Corresponding to each parameter if the velocity vector given by $Strang - n(x; a, b)$ is noted. **Red** points represents those parameters where **velocity vector is zero**. And **black** points represents those parameters wherer **velocity vector is non-zero**.
Here it can be seen that the boundary given analytically in the above section coincides with the boundary that we get through simulation.

Case IV: $x_0 < 0$ **and** n **a even number**

P(n):Let $z = R^n(ax_0 + b)$. For $x_0 < 0$, and n **a even number**. a-b space is divided into five regions which are given by:

Region I: $\{(a, b) : b < 0\}$, here $z = 0$

Region II: $\{(a, b) : b < -x_0 a, b > 0\}$, here $z = 1 + a + .. + a^{n-2}$

Region III: $\{(a, b) : b > \frac{-x_0 a^2}{1+a}, b > -x_0 a\}$,here $z = a^n x_0 + (1 + a + .. + a^{n-1})b$

Region IV: $\{(a, b) : b < \frac{-x_0 a^2}{1+a}, a > -1, b > 0\}$, here $z = (1 + a + .. + a^{n-3})b$

Region V: $\{(a, b) : b > 0, a < -1\}$, here $z = b$

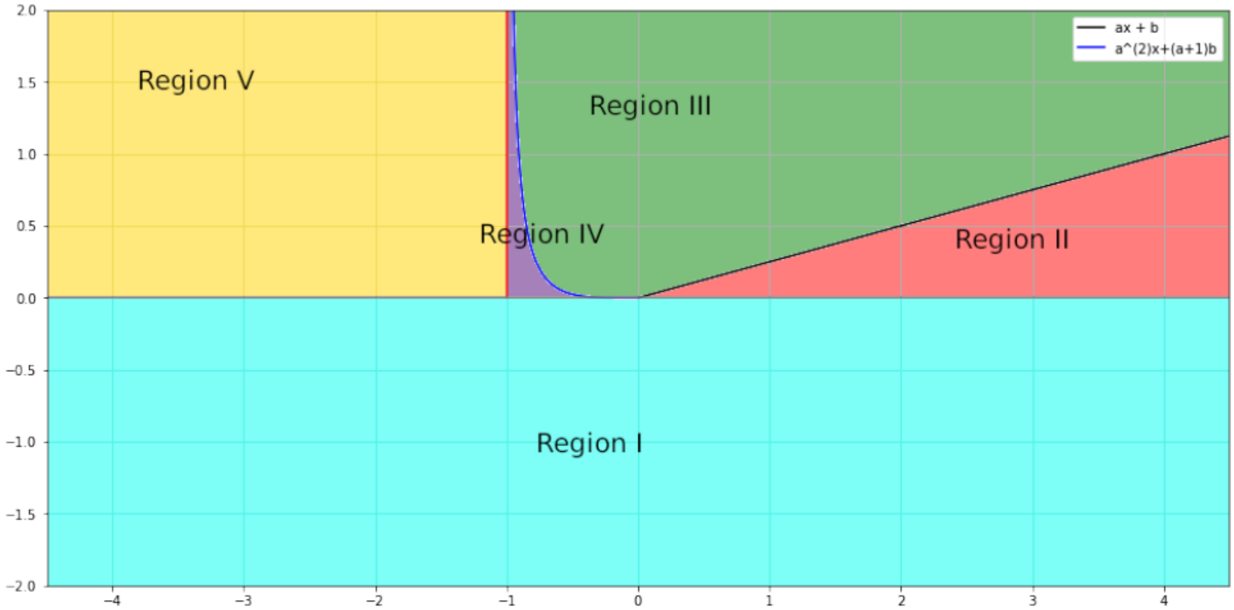


Figure 3.17: Let $z = R^n(ax_0 + b)$; Region I: $z = 0$; Region II: $z = 1 + a + .. + a^{n-2}$; Region III: $z = a^n x_0 + (1 + a + .. + a^{n-1})b$; Region IV: $z = (1 + a + .. + a^{n-3})b$; Region V: $z = b$ **Blue** boundary corresponds to the equation $a^2x + (1 + a)b$; **Black** boundary corresponds to the equation $ax + b = 0$; **Red** boundary corresponds to $a = -1$

So, the regions where there is no-motion is given by $A = \{(a, b) : b < 0\} \cup B = \{(a, b) : a < -1, b > 0\}$ (Figure 58).

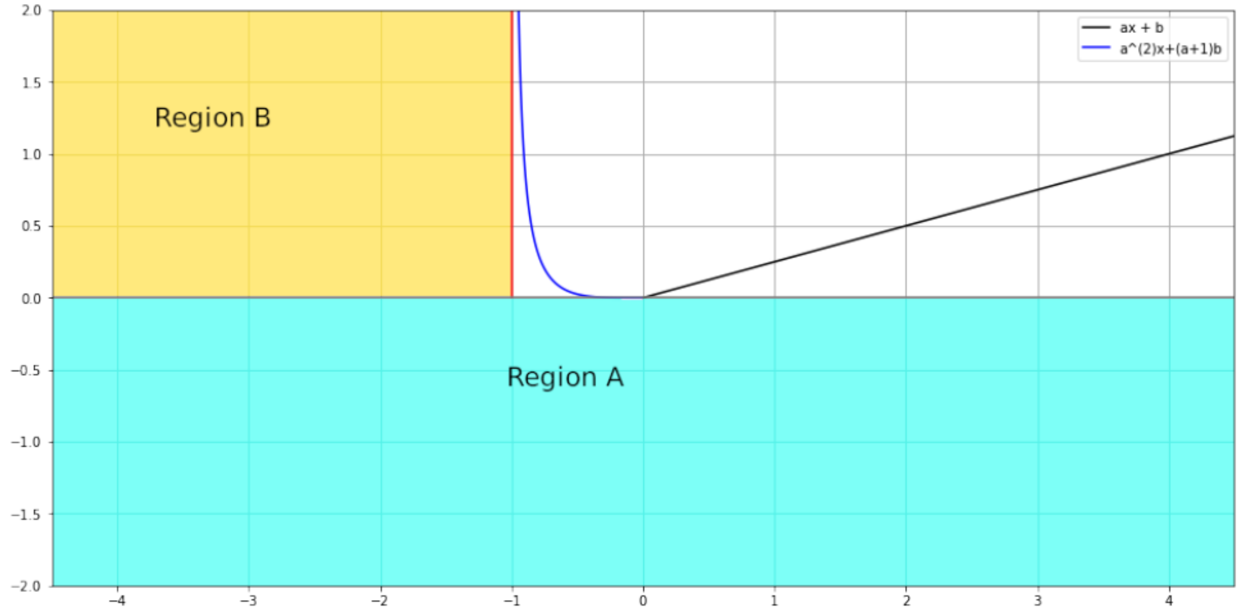


Figure 3.18: Let $z = R^n(ax_0 + b)$; Region A represents the region where $z = 0$
Blue boundary corresponds to the equation $a^2x + (1 + a)b$; **Black** boundary corresponds to the equation $ax + b = 0$; **Red** boundary corresponds to $a = -1$

Proof:

For **P(n=4)**, we verified that the a-b space is divided into five regions, where the region with no motion is given by $\{(a, b) : b < 0\} \cup \{(a, b) : a < -1, b > 0\}$.

Lets assume **P(k=n)** is true.

To prove **P(k=n+2)** is true.

For Strang-n, the a-b space is divided into five regions as given in figure 57.

Region I: $\{(a, b) : b < 0\}$

Here $z = 0$

So

$$R(aR(az + b) + b) = R(aR(b) + b) = R(b) = 0$$

Region II: $\{(a, b) : b < -x_0a, b > 0\}$

Here $z = (1 + a + a^2 + ... + a^{n-2)})b$

So $R(aR(az + b) + b) = R(aR((1 + a + a^2 + .. + a^{n-1})b) + b)$

Here $1 + a + a^2 + .. + a^{n-2} > 0$ for $a > -1$, hence

$$R(aR(az+b)+b) = R((1+a+a^2+...+a^{n-1})b) = (1+a+a^2+...+a^n)b$$

Here $z = b$
So

$$R(aR(az+b)+b) = R(aR(ab+b)+b) = R(b) = b$$

Region III: $\{(a,b) : b > \frac{-x_0a^2}{1+a}, b > -x_0a\}$
Here $z = a^nx_0 + (1+a+...+a^{n-1})b$
So

$$R(aR(az+b)+b) = R(aR(a^nx_0 + (1+a+...+a^{n-1})b)+b)$$

In Region III, we have $x_0a^n + (\sum_{i=0}^{i=n-1} a^i)b > 0$ for $n \geq 2$ (From Lemma III)
Hence we get,

$$R(aR(az+b)+b) = R(aR(a^nx_0+(1+a+...+a^{n-1})b)+b) = a^{n+2}x_0+(1+a+...+a^{n+1})b$$

Region IV: $\{(a,b) : b < \frac{-x_0a^2}{1+a}, a > -1, b > 0\}$
Here $z = (1+a+...+a^{n-3})b$
Here $1+a+a^2+..+a^{n-2} > 0\ \forall n$ for $a > -1$.
So

$$R(aR(az+b)+b) = R(aR((1+a+..+a^{n-2})b)+b) = (1+a+...+a^{n-2})b$$

Region V: $\{(a,b) : b > 0, a < -1\}$
Here $z = 0$
So

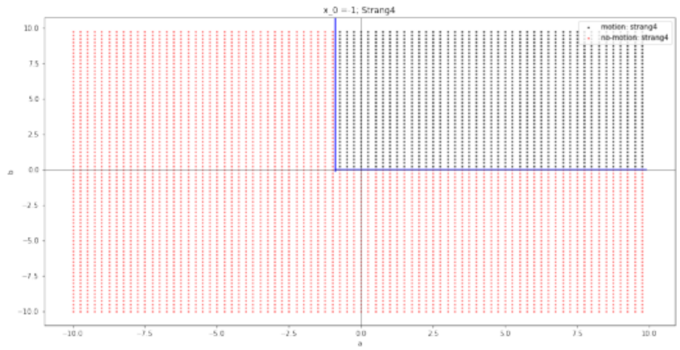
$$R(aR(az+b)+b) = R(ab+b) = 0$$

So the region where there is no-motion is given by $\{(a,b) : b < 0\} \cup \{(a,b) : a < -1, b > 0\}$.

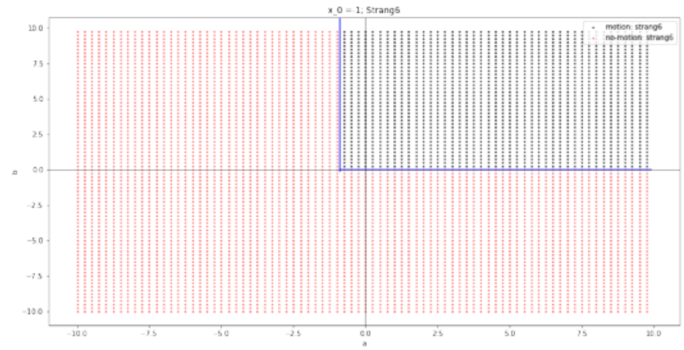
Verifying the boundaries through simulations:

The boundary is given by $b = 0$ for $a > -1$ and $a = -1$, in case of n-even.

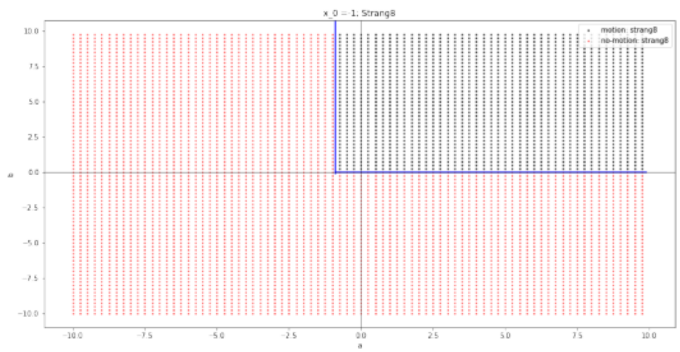
The blue line represents the boundary; the red dots represent no-motion and the black dots represents motion.



(a) Strang-4



(b) Strang-6



(c) Strang-8

Figure 3.19: Parameters taken from all over the ab space. Corresponding to each parameter if the velocity vector given by $Strang - n(x; a, b)$ is noted. **Red** points represents those parameters where **velocity vector is zero**. And **black** points represents those parameters wherer **velocity vector is non-zero**.

Here it can be seen that the boundary given analytically in the above section coincides with the boundary that we get through simulation.

Chapter 4

Glucose-Insulin dynamics

Till now, we discussed how to use sensitivity equations in the case when we have Strang function in the ODE. In other words, we have discussed the use of sensitivity equations in the context of neural ODEs. We also analytically defined the regions of the parameter (of Strang function) space, which are useful while optimization (where we need non-zero sensitivity). We shall use this knowledge in, incorporating Strang function in the Glucose-Insulin dynamics.

The Glucose-Insulin (G-I) model for the dynamics of glucose and insulin is given by:

$$\frac{dG}{dt} = R_0 - (E_{G0} + S_I I)G + k_{gut} q_{gut} \tag{4.1}$$

$$\frac{dI}{dt} = I_{max} \frac{G^2}{\alpha + G^2} - k_I I \tag{4.2}$$

The food dynamics is given by:

$$\frac{dq_{sto}}{dt} = -k_{sto} q_{sto} \tag{4.3}$$

$$\frac{dq_{gut}}{dt} = k_{sto} q_{sto} - k_{gut} q_{gut} \tag{4.4}$$

Values of the parameters of the differential equations

Parameter	Non-diabetic	Diabetic	Units
R_0	2.1	2.5	$mgdl^{-1}min^{-1}$
E_{G0}	10^{-3}	$2.5 * 10^{-3}$	min^{-1}
S_I	$3.06 * 10^{-3}$	$1.14 * 10^{-3}$	$ml\mu^{-1}U^{-1}min^{-1}$
α	10^4	10^4	mg^2dl^{-2}
I_{max}	0.28	0.93	$\mu Uml^{-1}min^{-1}$
K_I	0.01	0.06	min^{-1}
K_{sto}	0.036	0.026	min^{-1}
K_{gut}	0.098	0.026	min^{-1}

We observe that the term $I_{max}\frac{G^2}{\alpha+G^2}$ in equation 4.2 can be approximated by using $Strang2(G;a,b)$ function that is $ReLu(aReLu(aG+b)+b)$ (here a,b are the parameters), such that it fits the slope of the original term well.
So that we get the following differntial equation.

Dynamics of glucose and insulin is given by

$$\frac{dG}{dt} = R_0 - (E_{G0} + S_I I)G + k_{gut}q_{gut} \tag{4.5}$$

$$\frac{dI}{dt} = Strang2(G;a,b) - k_I I \tag{4.6}$$

The food dynamics is given by:

$$\frac{dq_{sto}}{dt} = -k_{sto}q_{sto} \tag{4.7}$$

$$\frac{dq_{gut}}{dt} = k_{sto}q_{sto} - k_{gut}q_{gut} \tag{4.8}$$

4.1 Sensitivity Equations

Let RHS of equation 4.5 and 4.6 be denoted by f_1 and f_2 respectively.
Let $s_{G1}(t) = \frac{dG(t)}{da}$, $s_{G2}(t) = \frac{dG(t)}{db}$, $s_{I1}(t) = \frac{dI(t)}{da}$, $s_{I2}(t) = \frac{dI(t)}{db}$. Using Forward sensitivity method we get the following equations.

$$\frac{ds_{G1}(t)}{dt} = \frac{\partial f_1}{dG}\frac{\partial G}{\partial a} + \frac{\partial f_1}{dI}\frac{\partial I}{\partial a} + \frac{\partial f_1}{\partial a} \quad (4.9)$$

$$= -E_{G0}s_{G1} + S_I G s_{I1} + 0 \quad (4.10)$$

$$s_{G1}(0) = 0 \quad (4.11)$$

$$\frac{ds_{I1}(t)}{dt} = \frac{\partial f_2}{dG}\frac{\partial G}{\partial a} + \frac{\partial f_2}{dI}\frac{\partial I}{\partial a} + \frac{\partial f_2}{\partial a} \quad (4.12)$$

$$= \begin{cases} a^2 s_{G1} + -K_I s_{I1} + 2aG + b & aG + b > 0; a^2G + ab + b > 0 \\ -K_I s_{I1} & otherwise \end{cases} \quad (4.13)$$

$$s_{I1}(0) = 0 \quad (4.14)$$

Solving the initial value problem given by equations $4.10 - 4.11$ and equations $4.13 - 4.14$, we can get $s_{G1}(t) = \frac{dG(t)}{da}$ at any time 't'. This will be used during the optimization.

$$\frac{ds_{G2}(t)}{dt} = \frac{\partial f_1}{dG}\frac{\partial G}{\partial b} + \frac{\partial f_1}{dI}\frac{\partial I}{\partial b} + \frac{\partial f_1}{\partial b} \quad (4.15)$$

$$= -E_{G0}s_{G2} + S_I G s_{I2} + 0 \quad (4.16)$$

$$s_{G2}(0) = 0 \quad (4.17)$$

$$\frac{ds_{I2}(t)}{dt} = \frac{\partial f_2}{dG}\frac{\partial G}{\partial b} + \frac{\partial f_2}{dI}\frac{\partial I}{\partial b} + \frac{\partial f_2}{\partial b} \quad (4.18)$$

$$= \begin{cases} a^2 s_{G2} - K_I s_{I2} + 1 + a & ax + b > 0; a^2x + ab + b > 0 \\ -K_I s_{I2} + 1 & ax + b \leq 0; b > 0 \\ -K_I s_{I2} & otherwise \end{cases} \quad (4.19)$$

$$s_{I2}(0) = 0 \quad (4.20)$$

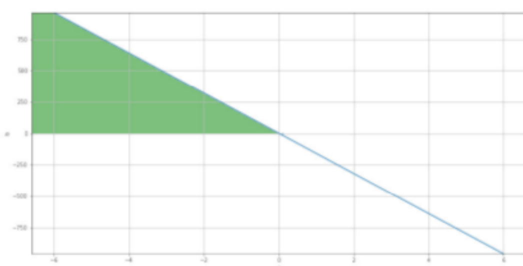
Similarly, Solving the initial value problem given by equations $4.16 - 4.17$ and equations $4.19 - 4.20$, we can get $s_{G2}(t) = \frac{dG(t)}{db}$ at any time 't'. This will be used during the optimization.

4.2 Simulations

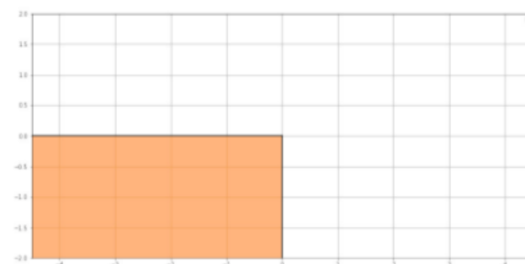
Verification of the regions in the parameter space

In this simulation we take parameters from different regions of the parameter space (admissible region and non-admissible regions) and show that only when we start with parameters from the admissible regions, we can reach the optimal parameters for the system. We show this by picking parameters from admissible regions and showing that the sensitvity is non-zero with respect to time for all time. And for parameters picked from non-admissible regions the sensitivity is zero with respect to time for some/all time.

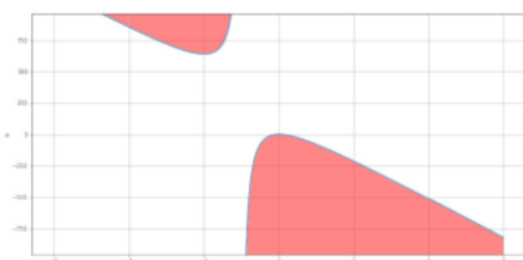
The different regions are as shown in the below figure.



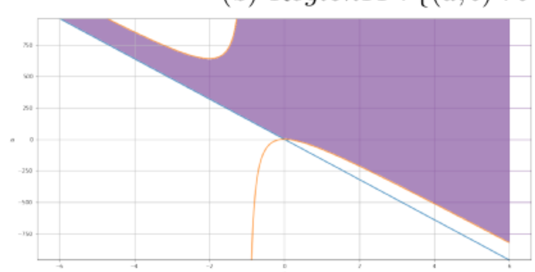
(a) $RegionI : \{(a, b) : b < -x_0 a, a < 0\}$



(b) $RegionII : \{(a, b) : b < 0, a < 0\}$



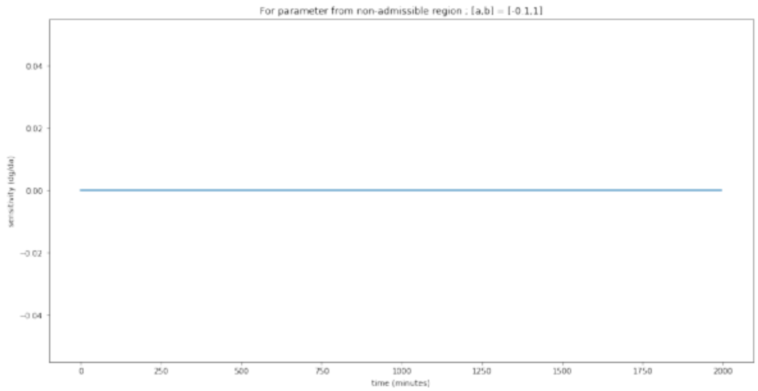
(c) $RegionIII : \{(a, b) : b < \frac{-x_0 a^2}{\sum_{i=0}^{1} a^i}\}$



(d) $RegionIV : (RegionI \cup RegionII \cup RegionIII)^c$

Region I,II,III are the non-admissible regions and Region IV is the admissible region for the parameter.

Case I : Parameter from Region I



(a) figure shows the sensitivity of Glucose with respect to parameter 'a' vs. time graph. Since the sensitivity is zero for all time 't,' that shows that any change in the parameter a does not bring change in the Glucose at any time 't.' Hence the starting from parameters from this region does not help in the optimization.

Case II : Parameter from Region II

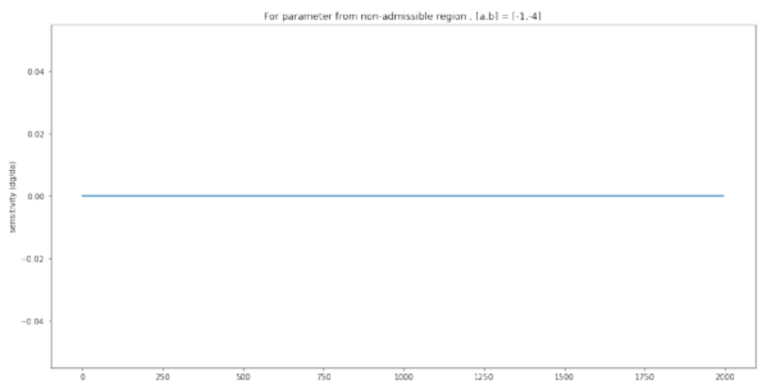


Figure 4.3: figure shows the sensitivity of Glucose with respect to parameter 'a' vs. time graph. Since the sensitivity is zero for all time 't,' that shows that any change in the parameter a does not bring change in the Glucose at any time 't.' Hence the starting from parameters from this region does not help in the optimization.

Case III : Parameter from Region III

In this region, we encounter two cases, as shown below.

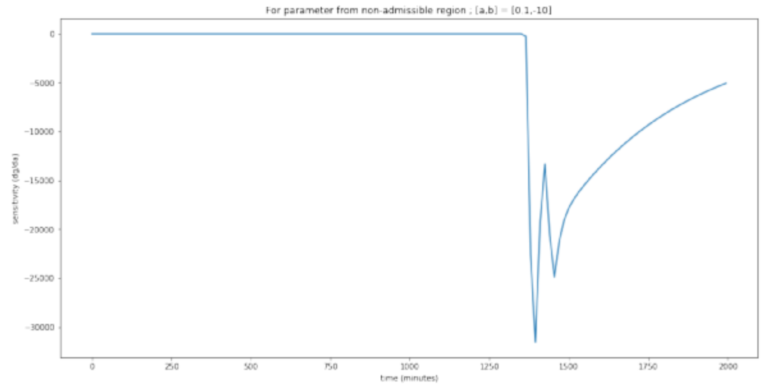


Figure 4.4: Figure shows the sensitivity of Glucose with respect to parameter 'a' vs time graph. In this case sensitivity is zero upto time '$t = 1250$'. This shows that if we start our parameters from this region we cannot fit the original glucose curve as the data upto time $t = 1250$ is not sensitive to the paramter a. So the glucose data upto time $t = 1250$ cannot be fit.

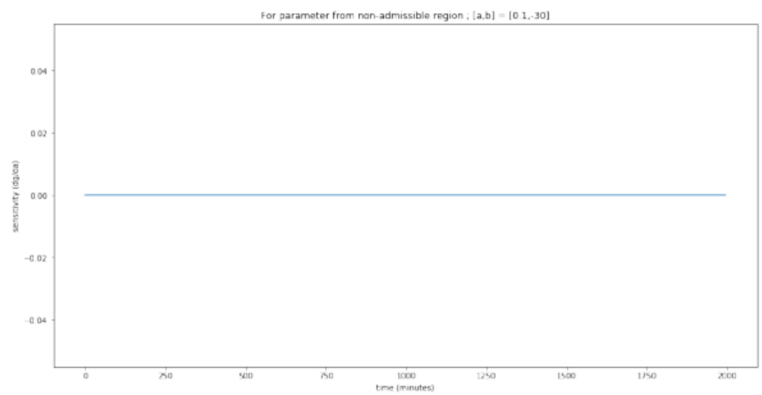


Figure 4.5: figure shows the sensitivity of Glucose with respect to parameter 'a' vs. time graph. Since the sensitivity is zero for all time 't,' that shows that any change in the parameter 'a' does not bring change in the Glucose at any time 't.'

Case IV : Parameter from Region IV

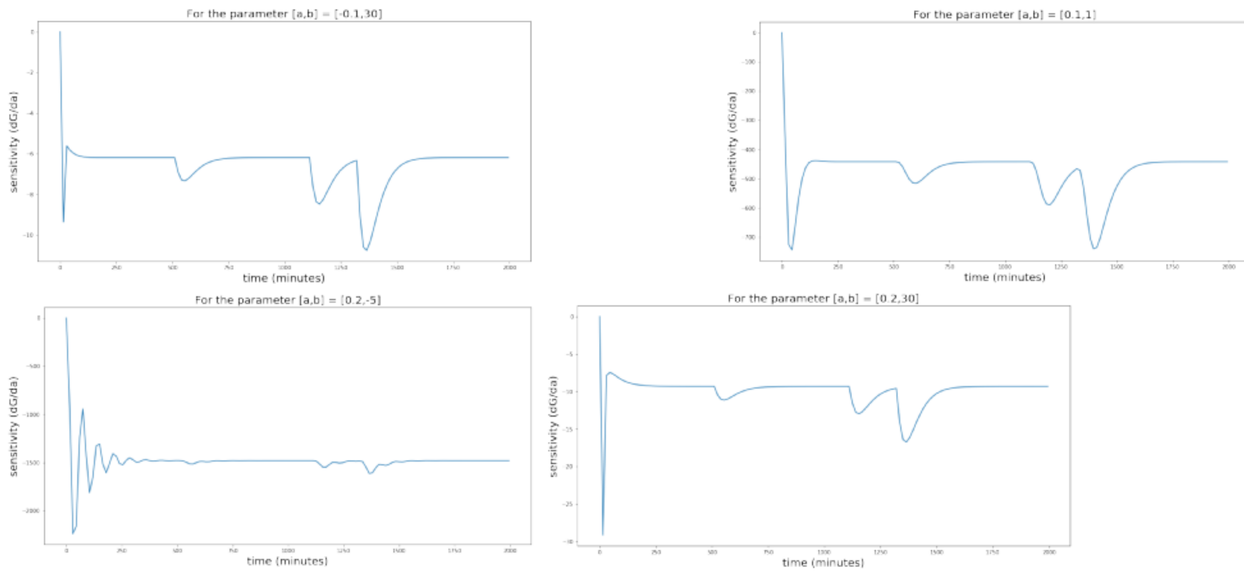


Figure 4.6: This figure shows the sensitivity of Glucose with respect to parameter 'a' vs. time graph. In this region, the sensitivity is non-zero; hence changes in the parameters reflect in the Glucose data. So taking parameters from these region works well in optimization.

Fitting the model to the data

Methodology:

- We first solve this differential equation given by equations $4.1-4.4$ to obtain $G(t)$. We then sample Glucose at a time interval of 15 minutes. This is our training data.

- In the next step, we solve the differential equation given by equations $4.5-4.8$, starting with some random parameters picked from the admissible region (as described in the previous chapter). We call the solution obtained here as $G'(t)$. Again we sample out Glucose at a time interval of 15 minutes. This is the predicted value of Glucose given by the model (obtained by incorporating Strang2 in the differential equation).

- Finally we try to minimize the mean squared difference between the actual value of Glucose($G(t)$) and the predicted value of Glucose($G'(t)$), i.e We try to minimize the objective function given by:

$$L(G'(t)) = \frac{\sum_{i=1}^{n}(G(i) - G'(i))^2}{n} \tag{4.21}$$

- The sensitivity equations will be used while computing the derivatives ($\frac{dL(G')}{da}$ and $\frac{dL(G')}{db}$). The equations are as follows:

$$\frac{dL(G')}{da} = \frac{\sum_{i=1}^{n} 2(G(i) - G'(i))(-\frac{dG'(i)}{da})}{n} \tag{4.22}$$

$$\frac{dL(G')}{db} = \frac{\sum_{i=1}^{n} 2(G(i) - G'(i))(-\frac{dG'(i)}{db})}{n} \tag{4.23}$$

In all the simulations below, we used a gradient-based optimization.

Fitting the model on a noisy data set:

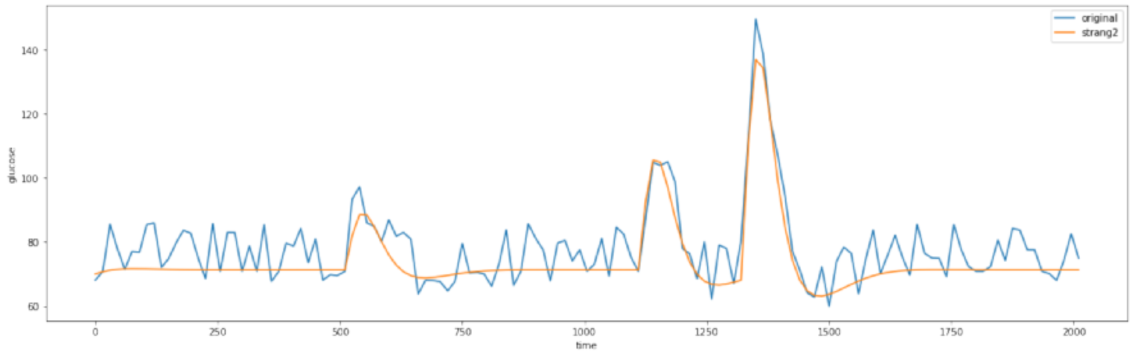
In these simulations we added noise (*from continous uniform distribution* $Unif[12,5]$) to the glucose data ($G(t)$) obtained by solving the differential equation (equations $4.1 - 4.4$) ands sampling out at an interval of 15 minutes. So our new training data becomes

$$G\hat{(}t) = G(t) + noise$$

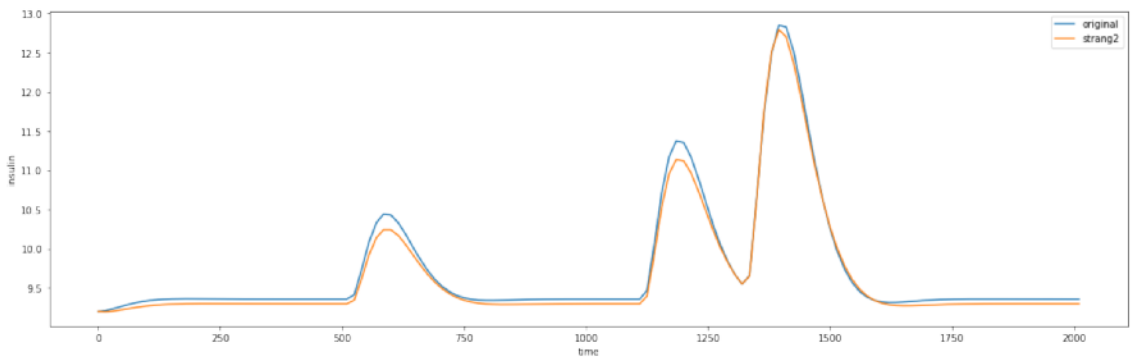
Now the objective function becomes:

$$L(G'(t)) = \frac{\sum_{i=1}^{n}(G\hat{(}i) - G'(i))^2}{n} \tag{4.24}$$

Case I: Non-diabetic

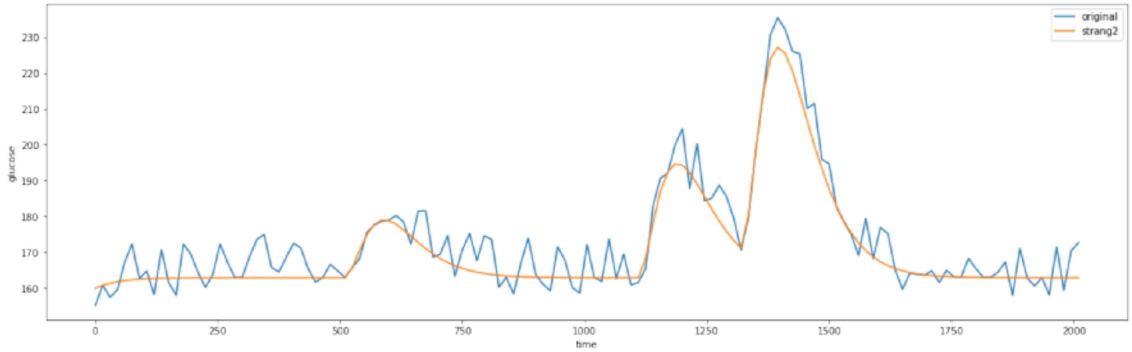


(a) Blue curve represents the simulated glucose data ($G(t)$) obtained by adding noise to the solution of the equations 4.1, 4.2, 4.3, 4.4. Initial parameter $a = 0.1, b = 0.1$ was taken from the admissible region. Equations 4.5, 4.6, 4.7, 4.8 was solved to get Glucose ($\hat{G(t)}$). The mean squared error of the two curves that is $\frac{\sum_{i=1}^{n}(\hat{G(i)} - G'(i))^2}{n}$ was optimized.The orange curve represents the Glucose value obtained by solving equations 4.5, 4.6, 4.7, 4.8, by using the final parameters obtained that is $a = 0.0381, b = -0.0088$.

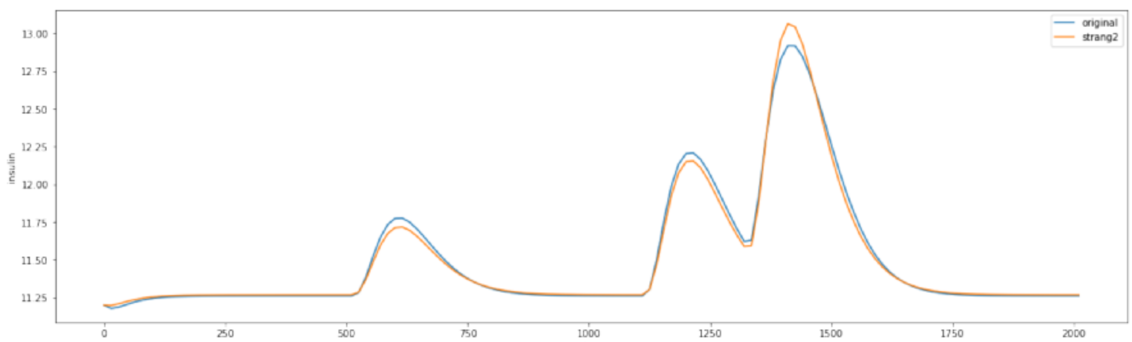


(b) Blue curve corresponds to the simulated insulin data obtained by solving equations 4.1, 4.2, 4.3, 4.4; Orange curve corresponds to the insulin value obtained after solving equations 4.5, 4.6, 4.7, 4.8 with parameter $a = 0.0381, b = -0.0088$ obtained after the optimzation.

Case II: Diabetic



(a) Blue curve represents the simulated glucose data ($G(t)$) obtained by adding noise to the solution of the equations $4.1, 4.2, 4.3, 4.4$. Initial parameter $a = 0.1, b = 0.4$ was taken from the admissible region and equations $4.5, 4.6, 4.7, 4.8$ was solved to get Glucose $(\hat{G(t)})$. The mean squared error of the two curves that is $\frac{\sum_{i=1}^{n}(\hat{G(i)} - G'(i))^2}{n}$ was optimized.The orange curve represents the Glucose value obtained by solving equations $4.5, 4.6, 4.7, 4.8$, by using the final parameters obtained that is $a = 0.0419, b = 0.3734$.



(b) Blue curve corresponds to the simulated insulin data obtained by solving equations $4.1, 4.2, 4.3, 4.4$; Orange curve corresponds to the insulin value obtained after solving equations $4.5, 4.6, 4.7, 4.8$ with parameter $a = 0.0419, b = 0.3734$ obtained after the optimzation.

The Python code for these simulations can be found at https://github.com/alekhranjan/Glucose-insulin-with-Strang2-

Chapter 5

Conclusion

In this project, we have shown how to use a simple neural network in the modeling of Glucose-insulin dynamics that corresponds to the Continous Glucose Monitoring (CGM) time series. We have also analyzed the parameter space for the Strang-n function. We found that only certain regions are appropriate to pick the parameters from while training the model. Although in this work, we have not used any real data set, since we have tested this method on a noisy data set (noise added manually), the same method can be used for a real-world data set.

Bibliography

[1] Goel P, Parkhi D, Barua A, Shah M, Ghaskadbi S. A Minimal Model Approach for Analyzing Continuous Glucose Monitoring in Type 2 Diabetes. Front Physiol. 2018;9:673. Published 2018 Jun 4. doi:10.3389/fphys.2018.00673

[2] Ibrahim Ayed and Emmanuel de Bézenac and Arthur Pajot and Julien Brajard and Patrick Gallinari. Learning Dynamical Systems from Partial Observations. CoRR. abs/1902.11136. 2019

[3] Tian Qi Chen and Yulia Rubanova and Jesse Bettencourt and David Duvenaud, Neural Ordinary Differential Equations, CoRR, abs/1806.07366, 2018,