2024-08-14

A Sensitivity Equation Framework for Parameter Estimation in Dynamical Systems

Joshua Newey
Brigham Young University

A Sensitivity Equation Framework for Parameter Estimation in Dynamical Systems

Joshua Newey

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Jared Whitehead, Chair
Scott Glasgow
Chris Verhaaren

Department of Mathematics

Brigham Young University

ABSTRACT

A Sensitivity Equation Framework for Parameter Estimation in Dynamical Systems

Joshua Newey
Department of Mathematics, BYU
Master of Science

We present a new framework for understanding parameter estimation in dynamical systems. The approach is developed within the modeling approach of continuous data assimilation. We outline the basic assumptions that lead to our derivation. Under these assumptions we show that the parameter estimation turns into a finite dimensional nonlinear optimization problem. We show that our derivation reproduces and extends the algorithm originally developed in [9]. We then implement these methods in three example systems: the Lorenz '63 model, the two-layer Lorenz '96 model, and the Kuramoto Sivashinsky equation. So as to remain sufficiently general, our derivations are largely formal; we leave a more rigorous justification for future work.

Keywords: Data Assimilation, Parameter estimation, Differential Equations

Acknowledgements

I would like to thank all the members of my committee each of which has been an important research mentor to me in a different way. I would like to thank the Brigham Young University math department for the wonderful teaching and support through my masters degree. I would like to thank my wife Sierra for supporting me, and for helping revise this document.

Contents

List of Figures

Chapter 1. Introduction

Mathematical modeling of physical systems can be largely classified into two approaches: 1) the theory driven models that are usually derived by referencing specific aspects of physical laws (relying on only a few parameters), and 2) data driven approaches that rely on fitting large general models with many parameters. Both approaches have advantages and disadvantages. The first method generally leads to models requiring fewer parameters and can often be much simpler, but in turn requires a great deal of upfront work, creative problem solving, and intuition. The second approach has been very popular recently with the advent of machine learning, access to large data sets, and computational resources. It has the advantage of being a very general approach, but requires vast amounts of data and computational power. Also, while it often provides effective models it generally gives little understanding being largely a "black box".

In this thesis we present a new framework for understanding parameter estimation in dynamical system. The method we discuss in this thesis is a mixture of the theory driven and data driven approaches. We assume that the physical system is understood up to some relatively small number of parameters. These methods are implemented within the framework of continuous data assimilation (CDA) [4]. In particular, we show that given some basic assumptions, the parameter estimation problem reduces to a finite dimensional nonlinear optimization problem. We develop methods for parameter estimation that require concurrent simulation of the sensitivity equation, and using asymptotic approximations we develop "on the fly" methods which only require the current state of the system to do parameter estimation steps. Specifically, implementing Newton's Method within this approach reproduces an algorithm developed by Carlson, Hudson, and Larios (CHL) [9]. The general approach also allows us to develop new algorithms for "on the fly" parameter estimation.

As our main goal is to develop a general approach for parameter estimation, we do not provide a fully rigorous justification. For example, in the section on perturbation theory we do not give the precise asymptotic bounds on the terms in our perturbative expansion. A rigorous justification for the finite dimensional case should be relatively straightforward, but is reserved for future work. Given our more general approach, we rely on numerical simulation as evidence of the correctness of our results. We go through several examples to show how the implementation works in several cases. Specifically, we consider the Lorenz '63 system, the two layer Lorenz '96 system, and the Kuramoto Sivashinsky Equations system. In all cases, we show that the parameter estimation algorithms developed in this thesis converge. We investigate the robustness of the methods to various parameters upon which the algorithms depend, e.g. learning rate in gradient descent; these are often referred to as "hyperparameters".

1.1 Introduction to Continuous Data Assimilation

Data assimilation [16] is the science of combining observations and theory to optimally estimate the state of a system. We briefly walk through the nudging approach for continuous time data assimilation We consider a general continuous time dissipative dynamical system that we write generally as

$$\dot{u} + F(u) = 0. \tag{1.1}$$

Here, u is considered to be an element of some Hilbert space H. We assume that $F : H \to H$ is some known, usually nonlinear, function. If the initial conditions were known exactly, we could just directly simulate the evolution. In practice, however, there will always be some error associated with our initial conditions. The systems we are interested in will mostly be chaotic which means that this error will be magnified over time.

If the state was known exactly, we could insert a "nudging" term to push the approximate model towards the true system,

$$\dot{v} + F(v) + \mu(v - u) = 0. \tag{1.2}$$

Here v represents our simulated approximate solution. It is straightforward to see that if the solution to (1.1) is bounded then choosing μ large enough will lead to convergence of v towards u. Of course, for the same reason that it is impossible to know our initial conditions exactly, it is impossible to observe the current state of the system. It will only be possible to observe some subset of the true state of the system, usually the large scales. Letting $I_h(u) : H \to H$ be our observation function, we can modify our nudging algorithm to accommodate this idea

$$\dot{v} + F(v) + \mu(I_h(v) - I_h(u)) = 0. \tag{1.3}$$

The remarkable thing about this method is that, for suitable choices of our observation function, this method not only leads to convergence of the observed error, $\|I_h(v) - I_h(u)\|_H$, to zero but also leads to convergence of the total error, $\|v - u\|_H$, to zero. This can be due to the fact that the systems studied though often infinite dimensional have finite dimensional attractors. For simplicity, we will assume that our observation operator is a linear projection operator $I_h(v) = Pv$ onto some finite dimensional subspace of H. In general, it is ideal to choose μ as large as possible while still maintaining numerical stability of the simulation [4]; we shall see that this allows us to make some perturbative approximations that lead to relatively simple parameter estimation algorithms.

This method was originally developed for the 2D Navier Stokes equations by Azouani, Olson, and Titi (AOT) [4]. In their original paper, they gave conditions on the observation function, I_h, which allowed for convergence. It was subsequently extended to many other systems (see for example [6, 1, 2, 13, 7]). This continuous data assimilation (CDA) approach is often referred to as "nudging".

It is in the context of CDA that we introduce our new approach for parameter estimation. We will see that given a few well motivated assumptions, we are able to implement a version of any derivative based finite dimensional optimization or root finding method. In particular we show that several methods developed in previous work correspond to specific finite dimensional optimization or root finding methods. Our approach provides a general framework which allows us to examine parameter estimation algorithms in a variety of systems.

Chapter 2. Error Function and Sensitivity Equations

2.1 Problem Statement

Having reviewed the CDA approach we move on to the original work of this thesis, extending this approach to handle parameter estimation. We begin by considering a modification of the previous CDA approach where the function F is no longer exactly known. We make the modification where the simulated system has some model error which takes the form of a finite number of parameters. To denote this, write $F(u;\gamma)$ as some function from $H \times \mathbb{R}^n$ to H. Thus the system we consider is given by

$$\begin{aligned} &\dot{u} + F(u;\gamma) = 0 \\ &\dot{v} + F(v;c) + \mu P(v-u) = 0, \end{aligned} \tag{2.1}$$

where $\gamma \in \mathbb{R}^n$ is some vector of "true" parameters which are assumed unknown, and $c \in \mathbb{R}^n$ is some vector of approximate parameters. This problem was originally considered in [9] and later using a different approach in [22]. However, the approach we develop here is very different than the approach in [22], and more general than the approach in [9]. In particular, we show that our general approach reproduces the algorithm in [9] as a special case.

Our goal is to find the ideal parameter values that minimize the error functional

$$I(u,v) = \frac{1}{2}\|P(v-u)\|_H^2, \tag{2.2}$$

Much of these methods extend straightforwardly to more arbitrary error functionals, e.g. different norms on our Hilbert space, so long as they satisfy some basic assumptions such as Fréchet differentiability and convexity.

We also briefly consider extending our approach to the system

$$\begin{aligned} &\dot{u} + F(u) = 0 \\ &\dot{v} + \tilde{F}(v;c) + \mu P(v-u) = 0, \end{aligned} \tag{2.3}$$

where $\tilde{F}$ is some approximation of F which depends on some parameters c. As we discuss, this models the case where the true system is either not known exactly, or too complicated to simulate fully. In this case, the analysis becomes more complicated because there is no "true" parameter values, only (possibly) optimal ones. However, as we shall see in Section 4.3.3, the methods appear to work in this modified case as well.

2.2 Assumptions

We make several key assumptions in deriving the following algorithms. These same assumptions generally underlie many similar parameter estimation approaches [10]. These assumptions were originally motivated by numerical experimentation, but have been justified more rigorously in several cases [20, 9]. The key idea is that given these assumptions, the problem reduces to a finite dimensional nonlinear optimization problem.

2.2.1 A1 : Time independence of long time error. First we consider what happens due to the presence of parameter error in our simulated system. What will generally happen is that nudging will cause the system to converge up to some error proportional to the parameter error and then stay there. This is shown for the Lorenz '63 system in Figure 2.1.

This motivates our assumption that for some $T \gg 1$ we have

$$\frac{\partial}{\partial t} I(u,v)\bigg|_{t=T} \approx 0. \tag{2.4}$$

This assumes that the system has been nudged for sufficient time such that the error has settled to a steady state value. This leads to the following result for long time

$$I(u,v)|_{t=T} \sim E(c;v_0), \tag{2.5}$$

where $E(c;v_0)$ is some function dependent on $\mathbb{R}^n$ and possibly our initial conditions. In practice the system will tend to relax and then fluctuate around a relaxed value. It may be that a more proper derivation contains an integral over time to cancel these fluctuations. Such a modification of parameter estimation algorithms is mentioned in [20].

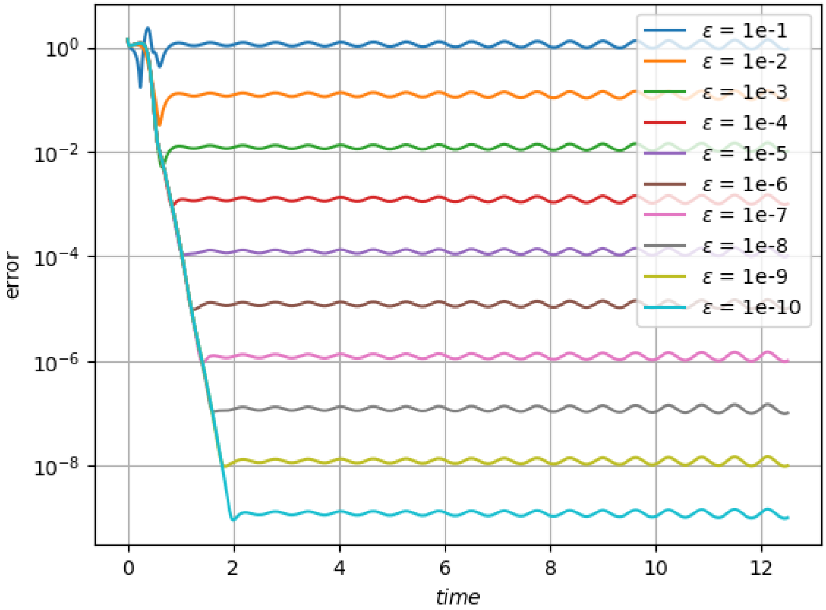

Figure 2.1: Time independence of long time error for the Lorenz '63 system while varying across c_1, here $\epsilon = \frac{\Delta c_1}{c_1}$.

2.2.2 A2: Independence of initial conditions. Another key assumption is that the long time behavior of the error is independent of initial conditions. Thus we assume that E has no dependence on initial conditions which allows us to just write $E(c)$. This is necessary because performing the parameter estimation requires regularly updating the parameter values and restarting the simulation. This assumption is clearly true if there is no parameter error as the nudging leads to convergence of the state error to zero. The key idea behind this assumption is that even in the case where there is parameter error, the nudging overwhelms the transient behavior on the initial conditions in v. This is demonstrated for the Lorenz '63 system in Figure 2.2.

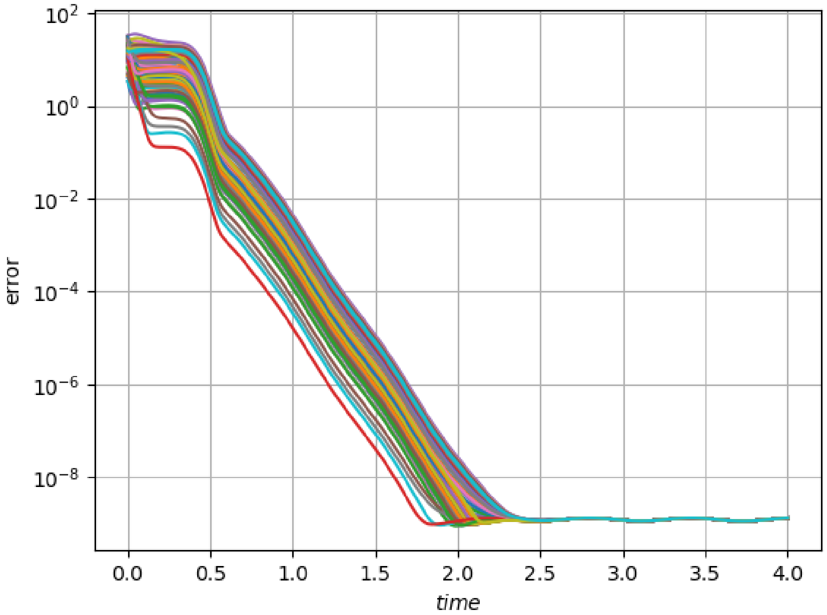

Figure 2.2: Independence of initial conditions with parameter error is $\epsilon = 10^{-10}$

2.2.3 A3: Convexity and Differentiability.

We need our function E to be sufficiently well behaved to apply the parameter estimation methods described below. This amounts to an assumption of local convexity and differentiability of our long-tine error function $E(c)$. We assume that our parameter estimation algorithm takes place in a system in which the nudging algorithm will converge. This clearly leads to the following result,

$$E(\gamma) = 0. \tag{2.6}$$

Clearly, from the definition of E, we have

$$E(c) \geq 0. \tag{2.7}$$

Thus, E has a root that is also a minimum at $c = \gamma$, i.e. the root of E at γ is at least order 2. We show the local convexity of the error in the Lorenz '63 system in Figure 2.3.

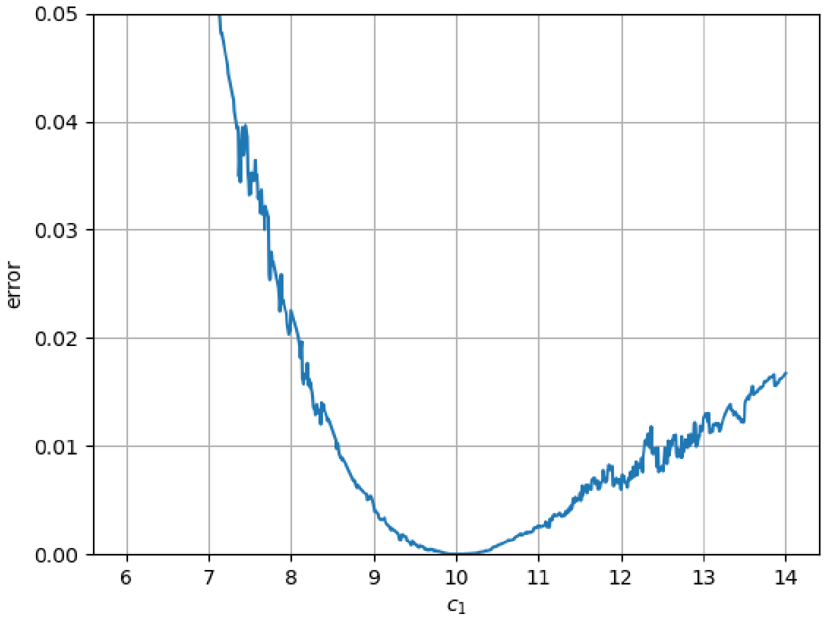

Figure 2.3: Convexity of the error function, $E(c_1))$, around the correct value $c_1 = 10$.

2.3 Derivatives of $E(\mathbf{c})$

Our goal is to implement gradient or Hessian based optimization and root finding methods on E. This requires estimating the derivatives of E. We know the form of our long term error which allows us to write the derivative of E with respect to c_i as function of $w_i := \frac{\partial v}{\partial c_i}$,

$$\frac{\partial E}{\partial c_i} = \frac{1}{2}\frac{\partial}{\partial c_i}\|P(v-u)\|_H^2\bigg|_{t=T} = \langle v-u, Pw_i\rangle_H\bigg|_{t=T}. \tag{2.8}$$

This will be valid as long as our projection operator P does not depend on c_i. Similarly if we can find the second order derivative of the solution we can find the Hessian. Indeed, defining $z_{ij} := \frac{\partial^2 v}{\partial c_i \partial c_j}$, we get the following result,

$$\frac{\partial^2 E}{\partial c_i \partial c_j} = \frac{1}{2}\frac{\partial^2}{\partial c_i \partial c_j}\|P(v-u)\|_H^2\bigg|_{t=T} = \langle v-u, Pz_{ij}\rangle_H\bigg|_{t=T} + \langle Pw_i, Pw_j\rangle_H\bigg|_{t=T}. \tag{2.9}$$

The problem then becomes how to estimate w_i and z_{ij}. This will naturally lead us to consider the sensitivity equations.

2.4 Sensitivity equations

We find the time evolution of w_i by taking the derivative of (2.1) with respect to c_i. We assume differentiability of F in both v and c. If H is infinite dimensional then a rigorous analysis would have to justify Fréchet differentiability of F in v. We move forward assuming that the formal manipulations performed here are justified. This leads us to,

$$\dot{w}_i + dF(v;c)\,[w_i] + F_{c_i}(v;c) + \mu P[w_i] = 0. \tag{2.10}$$

We use square braces to specify multilinear functions, e.g. $A[x,y]$ is a bilinear function. For single linear functions we will sometimes use this convention or sometimes us adjacency. Here, $dF[.]$ is the differential of F in v, and F_{c_i} is the partial derivative of F with respect to c_i with v held constant. Because they represent how the system is sensitive to various parameters, these equations are often referred to as "sensitivity equations" [11].

Assuming the initial conditions do not depend on our parameters, we have the following simple initial condition:

$$w_i|_{t=0} = 0. \tag{2.11}$$

We can immediately see several things from inspection of the sensitivity equations. For one thing, they are linear in w. While they are dependent on v, given a a known time dependent v we can simulate w. Operationally, this means they have the same computational difficulty as a linear time dependent system. They are also greatly simplified by having zero initial conditions. We see as well that μ acts as a decay factor on the sensitivity equations. This is important, because otherwise the sensitivity may grow with time.

We can also find the second order sensitivity equations for z_{ij} which are given by

$$\dot{z}_{ij} + dF(v)[z_{ij}] + d^2F(v)[w_i, w_j] + 2(dF)_{c_{(i}}(v)\left[w_{j)}\right] + F_{c_ic_j}(v;c) + \mu P[z_{ij}] = 0, \tag{2.12}$$

here we are using the notation $X_{(ij)} = \frac{1}{2}(X_{ij} + X_{ji})$. As in the first order case, the second order equations will also have zero initial conditions.

2.5 Asymptotics

We can make a remarkable number of simple estimates if we do perturbation theory for $\mu \to \infty$ on the sensitivity equations. In order to keep the derivation general the calculation in the next section is largely formal. It would only require a little additional work to make it rigorous in the finite dimensional case. In the infinite dimensional, i.e. partial differential equation (PDE), case the analysis will be much more involved.

2.5.1 First Order Sensitivity Equations. This is a singular perturbation problem because if we take the limit the equation becomes static. It is easiest to handle this by re-scaling the time variable as $\tau = \mu t$.

Thus we write

$$w_i(t) = W_i(\tau). \tag{2.13}$$

By the chain rule we see that we have

$$\dot{w}_i = \mu \dot{W}. \tag{2.14}$$

Plugging this into our sensitivity equations and letting $\epsilon = \frac{1}{\mu}$, we get

$$\mu \dot{W}_i + dF[W_i] + \mu P[W_i] = -F_{c_i}. \tag{2.15}$$

We start with the ansatz that W can be written as a perturbative power series in ϵ, i.e. $W = W^0 + \epsilon W^1 + \epsilon^2 W^2 + \ldots$. Following standard perturbation theory, we plug this into the differential equation and gather all terms of the same order.

The order 1 equation is

$$\begin{aligned} &\dot{W}_i^0 + PW_i^0 = 0, \\ &W_i^0|_{\tau=0} = 0. \end{aligned} \tag{2.16}$$

Clearly the only solution to this is $W_i^0 = 0$.

We next turn to the order ϵ equation

$$\dot{W}_i^1 + P[W_i^1] + dF[W_i^0] = -F_{c_i}(v). \tag{2.17}$$

This simplifies to

$$\begin{aligned}
&\dot{W}_i^1 + P[W_i^1] = -F_{c_i}(v), \\
&W_i^1(0) = 0. \qquad (2.18)
\end{aligned}$$

The solution, in the P subspace, is given by,

$$PW_i^1(\tau) = -\int_0^\tau PF_{c_i}\left(v\left(\frac{\tau'}{\mu}\right)\right)e^{\tau'-\tau}d\tau'. \qquad (2.19)$$

2.5.2 Watson's Lemma.

This gives us the following first order asymptotic approximation for w_i:

$$Pw_i(t) \sim -\frac{1}{\mu}\int_0^{\mu t} PF_{c_i}(v(\tau'/\mu))e^{\tau'-\mu t}d\tau'. \qquad (2.20)$$

If we make the change of variables $s = -\tau'/\mu + t$, this simplifies the solution to

$$Pw_i(t) \sim -\int_0^t PF_{c_i}(v(-s+t))e^{-\mu s}ds. \qquad (2.21)$$

Given the form above we can directly apply Watson's Lemma [21] to get

$$Pw_i(t) \sim -\sum_{n=0}^{\infty}\left[\left(\frac{d}{ds}\right)^n F_{c_i}(v(-s+t);c)\right]_{s=0}\frac{1}{\mu^{n+1}}. \qquad (2.22)$$

This suggests the leading order approximation

$$Pw_i(t) \sim -\frac{1}{\mu}PF_{c_i}(v(t)). \qquad (2.23)$$

2.5.3 Second Order Sensitivity Equation.

We can also look at perturbation theory for the second order sensitivity equations. We follow a similar process; plugging in our perturbative ansatz, we have

$$\begin{aligned}
(\dot{Z}_{ij}^0 + \epsilon\dot{Z}_{ij}^1 + \epsilon^2\dot{Z}_{ij}^2 + ...) + \epsilon dF[Z_{ij}^0 + \epsilon Z_{ij}^1 + \epsilon^2 Z_{ij}^2 + ...]& \\
+ P[Z_{ij}^0 + \epsilon Z_{ij}^1 + \epsilon^2 Z_{ij}^2 + ...] = -\epsilon d^2F[\epsilon W_i^1 + \epsilon^2 W_i^2 + ..., \epsilon W_j^1 + \epsilon^2 W_i^2 ...]& \\
- \epsilon 2(dF)_{c_{(j}}[\epsilon W_{i)}^1 + ...] - \epsilon F_{c_i c_j}.& \qquad (2.24)
\end{aligned}$$

The order 1 equation is given by

$$\begin{aligned}\dot{Z}^0_{ij} + PZ^0_{ij} &= 0,\\ Z^0_{ij}|_{\tau=0} &= 0.\end{aligned}\tag{2.25}$$

Again, clearly the only solution is identically zero.

The order ϵ equation is given by,

$$\begin{aligned}\dot{Z}^1_{ij} + PZ^1_{ij} &= -F_{c_i c_j}\\ Z^1_{ij}|_{\tau=0} &= 0\end{aligned}\tag{2.26}$$

With solution

$$PZ^1_{ij}(\tau) = -\int_0^\tau PF_{c_i c_j}(v(\tau'/\mu))e^{\tau'-\tau}d\tau'.\tag{2.27}$$

We can again apply Watson's Lemma to get the leading otder approximation

$$PZ^1_{ij}(\tau) \sim PF_{c_i c_j}(v(\tau/\mu);c).\tag{2.28}$$

In the case where explicit dependence of F on the parameters is linear this term will be zero. Given that many of the examples we consider have this property, in these cases, we move to second order in perturbation theory.

Consider the case where $F_{c_i c_j} = 0$. Given this assumption, the second order equation is given by

$$\begin{aligned}&\dot{Z}^2_{ij} + P[Z^2_{ij}] = -2(dF)_{c_{(i}}[W^1_{j)}],\\ &Z_ij^2|_{\tau=0} = 0.\end{aligned}\tag{2.29}$$

This gives us the equation

$$\dot{Z}^2_{ij} + P[Z^2_{ij}] = -2(dF)_{c_{(i}}[W^1_{j)}].\tag{2.30}$$

Following a similar derivation to before, we find the approximation

$$Pz_{ij} \sim \frac{1}{\mu^2}2P(dF)_{c_{(i}}(v)\left[W^1_{j)}\right].\tag{2.31}$$

If P commutes with $(dF)_{c_i}$ we can us the approximation for PW_j^1 to get

$$Pz_{ij} \sim \frac{1}{\mu^2} 2(dF)_{c_{(i}}(v) \left[PF_{c_{j)}}\right] . \tag{2.32}$$

Chapter 3. Parameter Estimation from Optimization Methods

Under some reasonable assumptions $E(c) = \frac{1}{2}\|P(v-u)\|^2\big|_{t=T\gg 1}$ as a function of c has a unique root and minima at $c = \gamma$. Given knowledge of the sensitivities w_i we can derive parameter estimation algorithms from any derivative based root finding or optimization algorithms. We follow the "relax then punch" approach where we allow the error to relax between parameter update steps [9, 10, 20]. This is required because our assumption is that the error has relaxed enough that we can do optimization on $E(c)$. This amounts to the assumption that $\Delta t = t_{k+1} - t_k$ is "large" (where t_k is our parameter update time). In practice, the system usually relaxes very quickly, so the interval between parameter updates does not need to be especially large. The behavior for large error is $\|v-u\| \sim e^{-\mu t}$, we see the characteristic relaxation time scale is $\frac{1}{\mu}$. Thus we just have to choose Δt much larger than this. In practice, because systems tend to fluctuate around a minimum rather than completely relax, there seems to be a sweet spot for the value of Δt where the system has just relaxed, but has not started to fluctuate. A good rule of thumb seems to be choosing $\Delta t > 10\frac{1}{\mu}$.

In order to compute the parameter updates we need a way of estimating the sensitivities w_i. One way to do this is by directly simulating the solution from the equations of motion numerically. We shall refer to this method as the direct simulation (DS) method. This method has the advantage of requiring less approximations, but does involve additional computation. It may still be preferable over many equation discovery methods that require running the dynamics multiple times across many different parameter values. However, the DS method will not be as useful in practice generally, but will be very useful as a demonstration tool for this thesis. The second method relies on using perturbation theory in $\mu \to \infty$, which we derived in the last section. This method, while it requires additional approximation, is more computationally efficient. It allows us to estimate the parameters using only the current state v of the nudged system. We shall refer to these methods as "on the fly" (OTF) methods. We shall see that OTF methods often lead to similar or even better convergence properties than DS methods. We briefly review several possible simple optimization and root finding algorithms and how they can be implemented as parameter estimation algorithms.

3.1 Parameter Estimation as Root Finding

3.1.1 Single Parameter Estimation: Newton's Method. We consider initially fitting a single parameter. The long time error will approach zero when the exact parameters are known. Thus we can apply any derivative based root finding methods as parameter estimation methods. The most ubiquitous of these is Newton's method; it finds the roots of a function $f(x)$ by approximating it as a linear function locally around some starting point x_0

$$f(x) \approx f(x_0) + f'(x_0) \cdot (x - x_0). \tag{3.1}$$

Next, we solve for the roots of this linear function, i.e. $f(x) = 0$.

$$x = x_0 - \frac{f(x_0)}{f'(x_0)}. \tag{3.2}$$

This suggests the iterative root finding algorithm

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}. \tag{3.3}$$

Modification for higher multiplicity roots. The root of our error function will also be a minimum, thus it will have a multiplicity of at least two. Roots of higher multiplicity slow the convergence rate of Newton's method, but an increased convergence rate can be found by multiplying the update step by a factor of the multiplicity [14]. This leads to the following modification of Newton's method when the multiplicity, m, of the root is known

$$x_{i+1} = x_i - m\frac{f(x_i)}{f'(x_i)}. \tag{3.4}$$

In our case, where the multiplicity is at least 2, this leads to the following parameter estimation algorithm.

$$c^{(k+1)} = c^{(k)} - \left.\frac{\|P[v-u]\|^2}{\langle P[v-u], w\rangle}\right|_{t_k}. \tag{3.5}$$

3.1.2 Comparison with CHL algorithm .

If we further examine this algorithm, we see remarkable similarity to the parameter estimation algorithm developed by Carlson, Hudson, and Larios (CHL) [9]. They considered a nudged system of the form

$$\begin{aligned} \dot{u} + \gamma L u + F(u) &= f \\ \dot{v} + cLv + F(v) &= f - \mu P(v-u). \end{aligned} \tag{3.6}$$

In this case taking the asymptotic approximation for Newton's method results in the following algorithm for parameter estimation. The parameter update is given by

$$c^{(k+1)} = c^{(k)} + \mu\left.\frac{\|P(v-u)\|_H^2}{\langle P(v-u), Lv\rangle_H}\right|_{t_k}. \tag{3.7}$$

We see that this agrees exactly with the CHL parameter estimation algorithm [9]. This shows us how the CHL algorithm fits into the sensitivity equaiton framework for parameter estimation.

3.1.3 Generalization to Multiple Parameters.

We next consider root finding for a function $f : \mathbb{R}^n \to \mathbb{R}$. Following the derivation of Newton's method, we approximate f as a linear function. We consider the behavior near a point $\mathbf{x_0}$ which leads to the linear system

$$-f(\mathbf{x}_0) \approx \nabla f(\mathbf{x}_0) \cdot (\mathbf{x} - \mathbf{x}_0). \tag{3.8}$$

This is an underdetermined system, but because we are making a local approximation, we can choose the closest root of the linear system that will be in the direction of steepest descent. This is given by

$$\mathbf{x} = \mathbf{x}_0 - \frac{f(\mathbf{x}_0)}{\|\nabla f(\mathbf{x}_0)\|^2} \nabla f(\mathbf{x}_0). \tag{3.9}$$

Taking the modification for double roots, this leads to the parameter estimation algorithm given by

$$c_i^{(k+1)} = c_i^{(k)} - \frac{\|P(v-u)\|_H^2}{\|\langle P(v-u), \mathbf{w}\rangle_H\|_{\mathbb{R}^n}^2} \cdot \langle P(v-u), w_i\rangle_H \bigg|_{t_k}. \tag{3.10}$$

If we take the asymptotic approximation for large μ we get the following OTF algorithm

$$c_i^{(k+1)} = c_i^{(k)} + \mu \frac{\|P(v-u)\|_H^2}{\|\langle P(v-u), F_c\rangle_H\|_{\mathbb{R}^n}^2} \cdot \langle P(v-u), F_{c_i}(v)\rangle_H \bigg|_{t_k}. \tag{3.11}$$

As we see in Section 3.2.1, this can be seen as a modification of gradient descent with a variable learning rate.

3.1.4 Root Finding vs. Optimization. It is important to make the distinction between root finding and optimization algorithms. In this thesis, we mostly assume that the functional form of the unknown dynamics is exactly known, with the only unknown being the precise value of the parameters. In this case, there are exactly correct values of parameters. However, in many applications, the exact form of the dynamics may not be known or may be impossible to simulate, and only some approximate form may be used. We briefly consider this case in section 4.3.3. In those cases, there is no longer "true" parameter values, only optimal ones, and root finding algorithms will no longer be appropriate. The issue is that if f does not go to zero when f' goes to zero, then Newton's root finding method will approach the minimum but will be pushed away from it by the singular denominator. See Figure 3.1 and compare with the behavior in Figure 4.19.

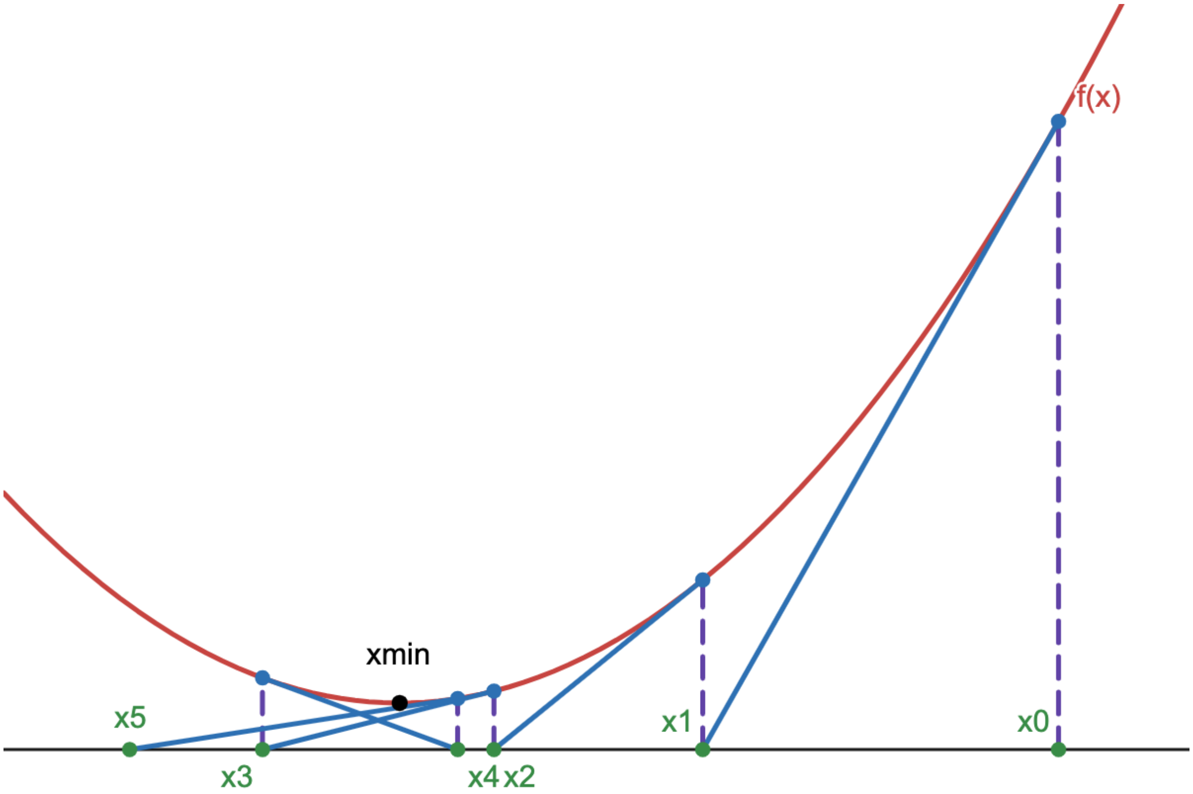

Figure 3.1: Newton's root finding method will approach a small minima but be pushed away by the denominator approaching zero.

3.2 Parameter Estimation as Optimization

3.2.1 Gradient Descent . One of the simplest optimization methods is gradient descent. It says to move in the direction of steepest descent of the error. This gives us the following optimization algorithm

$$x_{i+1} = x_i - r\nabla f(x_i) \tag{3.12}$$

where the parameter r is the learning rate. The corresponding parameter estimation algorithm is given by the following

$$c_i^{(k+1)} = c_i^{(k)} - r\langle P(v-u), Pw_i\rangle\Big|_{t_k}. \tag{3.13}$$

If we take the asymptotic approximation we get the OTF parameter estimation algorithm

$$c_i^{(k+1)} = c_i^{(k)} + \frac{r}{\mu}\langle P(v-u), PF_{c_i}\rangle\bigg|_{t_k}. \tag{3.14}$$

This algorithm only relies on values of the first derivative of our error. It can be improved upon if we can find, or at least approximate, the second derivative of our error.

3.2.2 Hessian method.

Since a minimum of the function will be a root of its derivative we can apply Newton's method to the gradient of our error function. This method is sometimes called Newton's method for optimization, or the Newton-Raphson method [5]. To avoid confusion with the root-finding Newton's method, we shall refer to this method as the Hessian method. Applying this method leads to the following iterative formula

$$x_{i+1} = x_i - (D^2 f(x_i))^{-1}\nabla f(x_i), \tag{3.15}$$

where $D^2 f$ is the Hessian of f. As such, implementing this as a parameter estimation algorithm requires knowledge of all of the second derivatives with respect to our various parameters. This algorithm assumes that the Hessian is invertible. As a parameter estimation algorithm, this becomes

$$c_i^{(k+1)} = c_i^{(k)} - \sum_j \left(\langle v-u, Pz_{ij}\rangle_H + \langle Pw_i, Pw_j\rangle_H\right)^{-1}\langle v-u, Pw_j\rangle_H\bigg|_{t_k}. \tag{3.16}$$

This requires knowledge of the solution to the second order sensitivity equations. In practice directly simulating the second order sensitivity equations grows very computationally expensive as the number of parameters increases. For example, in a system dependent on n parameters there are n first order sensitivity equation, but there are $\frac{n(n+1)}{2}$ unique second order sensitivity equations. This makes direct simulation methods more computationally prohibitive for the Hessian based optimization algorithm. One possibility is to use a mixed DS and OTF approach by using the perturbative approximation for z_{ij} in terms of w_i, but directly simulating w_i.

3.2.3 Approximations to the Hessian.

The big drawback of the Hessian algorithm is that it requires knowledge of z_{ij}. The Gauss-Newton method comes from assuming that the quadratic terms in the Hessian dominate the terms dependent on z_{ij}. This is a good approximation when $v - u$ is small or when z_{ij} is small. Making this assumption we get

$$c_i^{(k+1)} = c_i^{(k)} - \sum_j \left(\langle Pw_i, Pw_j \rangle_H\right)^{-1} \langle v - u, Pw_j \rangle_H \Big|_{t_k}. \tag{3.17}$$

The Levenberg-Marquardt algorithm is a modification of the Gauss-Newton algorithm. It is generally more robust than the Gauss-Newton algorithm [19]; for many cases, it will converge to the correct answer even when the initial guess is far off. It may also help in points where the matrix $\langle w_i, w_j \rangle$ is not invertible. It is given by the following

$$c_i^{(k+1)} = c_i^{(k)} - \sum_j \left(\langle Pw_i, Pw_j \rangle_H + \lambda \delta_{ij}\right)^{-1} \langle v - u, Pw_j \rangle_H \Big|_{t_k}. \tag{3.18}$$

The inverse given above will always be defined so long as $-\lambda$ is not an eigenvalue of $\langle Pw_i, Pw_j \rangle_H$. For appropriate choices of the cost function, the matrix $\langle Pw_i, Pw_j \rangle_H$ will be positive semi-definite and so this algorithm will always be defined. Clearly, in the limit $\lambda \to 0$ this converges to the Gauss-Newton algorithm. In the $\lambda \to \infty$ limit we have

$$\sum_j \left(\langle Pw_i, Pw_j \rangle_H + \lambda \delta_{ij}\right)^{-1} \langle v - u, Pw_j \rangle_H \sim \frac{1}{\lambda} \langle v - u, Pw_j \rangle_H = \frac{1}{\lambda} \nabla E(c). \tag{3.19}$$

Thus, the Levenberg-Marquardt algorithm comes from interpolating between Gauss-Newton and gradient descent. Using the perturbative approximation developed earlier we find the following OTF version of the Levenberg-Marquardt algorithm

$$c_i^{(k+1)} = c_i^{(k)} + \sum_j \left(\langle PF_{c_i}, PF_{c_j} \rangle_H + \lambda \delta_{ij}\right)^{-1} \langle v - u, PF_{c_j} \rangle_H \Big|_{t_k}. \tag{3.20}$$

Chapter 4. Applications

The methods outlined before this are given in full generality. We demonstrate how these methods may be implemented in several simple systems. The derivations we have given are not fully rigorous, and while a rigorous proof may be possible in the ODE case it is likely not possible in full generality in the PDE case. For this work, we forgo a full rigorous derivation. We instead rely on numerical evidence for convergence of the algorithms and leave the more rigorous analysis for future work.

4.1 Lorenz '63 System

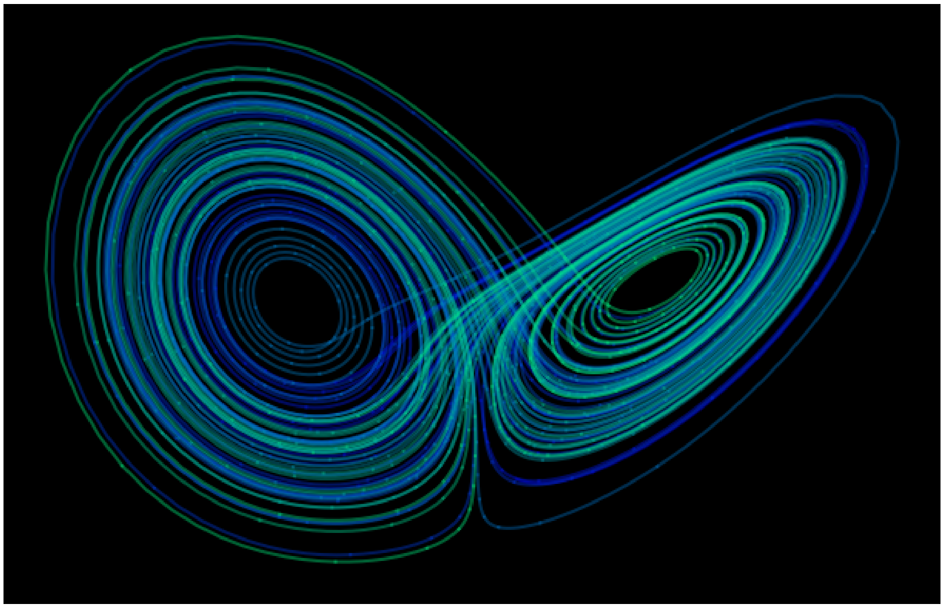

Figure 4.1: Phase space trajectory of the Lorenz '63 system.

One of the classic examples of Chaos is the Lorenz '63 system. It was originally developed by meteoroligist Edward Lorenz to model the atmosphere [17]. The dynamics of the Lorenz '63 system are shown in Figure 4.1. The equations describe a simplified model of a two dimensional fluid warmed from below and cooled from above. u_1 represents the rate of convection, u_2 is the horizontal temperature variation, and u_3 is the vertical temperature variation. The parameters γ_1, γ_2, and γ_3 are proportional to the Prandtl number, Rayleigh number, and physical size of the layer respectively. This system forms an ideal toy model for understanding chaos in dynamical systems because of it's relatively exotic behavior despite it's low dimensionality.

Because this system is low dimensional it will be easier to understand and analyze than higher dimensional dynamical systems. The differential equations for this system are given by,

$$\begin{aligned}\dot{u}_1 &= -\gamma_1(u_1 - u_2)\\ \dot{u}_2 &= u_1(\gamma_2 - u_3) - u_2\\ \dot{u}_3 &= u_1u_2 - \gamma_3u_3.\end{aligned} \tag{4.1}$$

The corresponding nudged system is

$$\begin{aligned}\dot{v}_1 &= -c_1(v_1 - v_2) - \mu(v_1 - u_1)\\ \dot{v}_2 &= v_1(c_2 - v_3) - v_2 - \mu(v_2 - u_2)\\ \dot{v}_3 &= v_1v_2 - c_3v_3 - \mu(v_3 - u_3).\end{aligned} \tag{4.2}$$

Numerically the parameter update only converges if the number of parameters being estimated is less than or equal to the rank of the observation projector. It appears that this is a fundamental restriction on these algorithms. This is because our error function maps from our parameter space into the range of P and then into $\mathbb{R}$. We see that if the dimensionality of the parameter space is less than the dimension of the range of P, then the parameters are underdetermined. This restriction is not so prohibitive in more realistic, higher dimensional systems. In practice, this means that if we want to estimate all three parameters in the Lorenz system, we have to nudge all of the variables. To focus on the parameter estimation rather than the nudging, we take our entire state to be observed. Nudging with some subset of the state was thoroughly explored in [8, 12, 10]. Indeed, [10] looked at parameter estimation using a version of the CHL algorithm to do parameter estimation on various subsets of the full parameters. We will explore the connection with the algorithm used in that paper below.

4.1.1 Direct Simulation Methods. Now, we find the following system of 9 sensitivity equations for $w_{ij} = \frac{\partial v_i}{\partial c_j}$.

$$\begin{aligned}
\dot{w}_{1j} &= -\delta_{1j}(v_1 - v_2) - c_1(w_{1j} - w_{2j}) - \mu_1 w_{1j}, \\
\dot{w}_{2j} &= w_{1j}(c_2 - v_3) + v_1(\delta_{2j} - w_{3j}) - w_{2j} - \mu_2 w_{2j}, \\
\dot{w}_{3j} &= w_{1j} v_2 + v_1 w_{2j} - \delta_{3j} v_3 - c_3 w_{3j} - \mu_3 w_{3j}. \qquad (4.3)
\end{aligned}$$

Note that because we have the same number of variables as parameters we can represent w_{ij} as a matrix, $W = [w_{ij}]$. In this case we get that the gradient of the error $E(c) = \frac{1}{2}\|\mathbf{v} - \mathbf{u}\|^2_{\mathbb{R}^3}\big|_{t=T}$ is

$$\nabla E = W^T(\mathbf{v} - \mathbf{u})\big|_{t=T}. \qquad (4.4)$$

Now, this leads to various parameter estimation algorithms which we explore below. We take the relaxation parameter to be $\mu = 100$ unless otherwise specified, and $\Delta t = 0.5$. We take the true parameter values to be $\gamma = (10, 28, 2.667)$, and we start off with initial conditions $\mathbf{u} = (0, 1, -1)$ and $\mathbf{v} = \mathbf{0}$. We start with initial parameter values $c = \frac{1}{2}\gamma$.

Gradient Descent. The gradient descent algorithm is given by

$$c^{k+1} = c^k - rW^T(\mathbf{v} - \mathbf{u})\big|_{t=t_k}. \qquad (4.5)$$

For simplicity, we take the learning rate to be constant. We experiment with various learning rates in Figure 4.2.

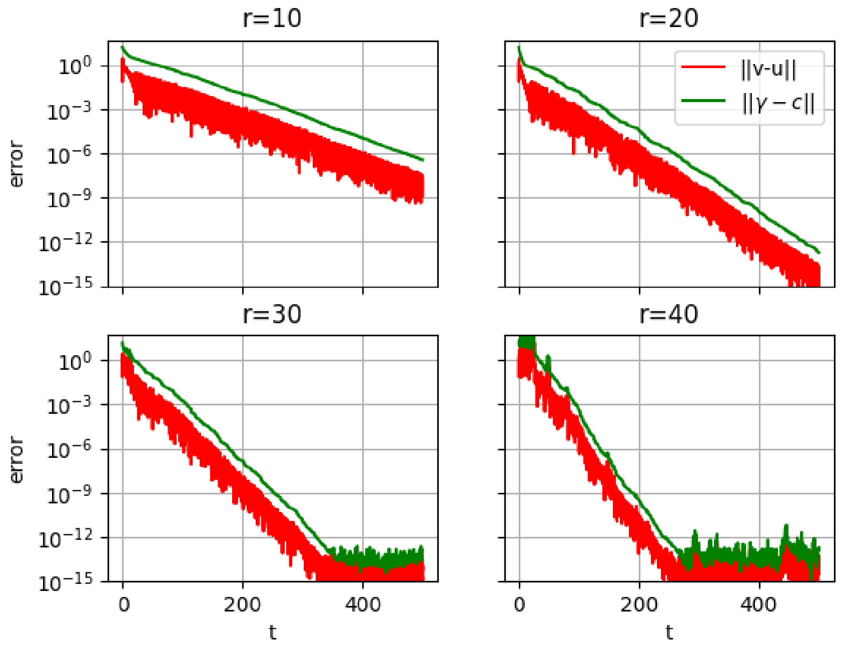

Figure 4.2: DS gradient descent with different learning rates.

We see that choosing a large learng rate leads to the faster convergence. However, if we choose the learning rate too large, for example $r = 50$, we see that we no longer see convergence of this algorithm this is shown in Figure 4.3. This is because a learning rate that is too large will consistently overshoot the minimum.

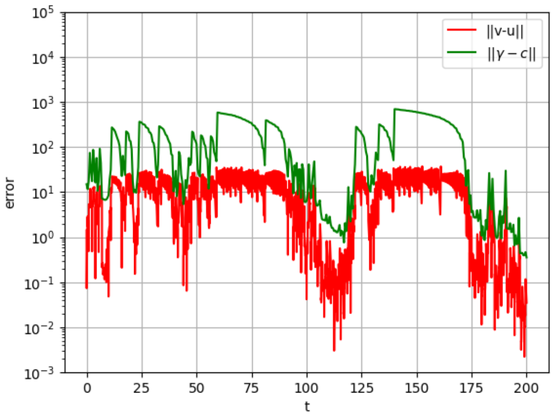

Figure 4.3: Non convergence of gradient descent with $r = 50$.

Newton's Method. Applying the multiparameter Newton's method for the Lorenz '63 system gives us the following parameter estimation algorithm.

$$c^{(k+1)} = c^{(k)} - \frac{\|\mathbf{v}-\mathbf{u}\|^2}{\|W^T(\mathbf{v}-\mathbf{u})\|^2} W^T(\mathbf{v}-\mathbf{u}). \tag{4.6}$$

As discussed earlier, this version of Newton's method is essentially a modification of gradient descent with a specific choice of variable learning rate. Figure 4.4 shows that this parameter estimation algorithm gives us significantly faster convergence than gradient descent.

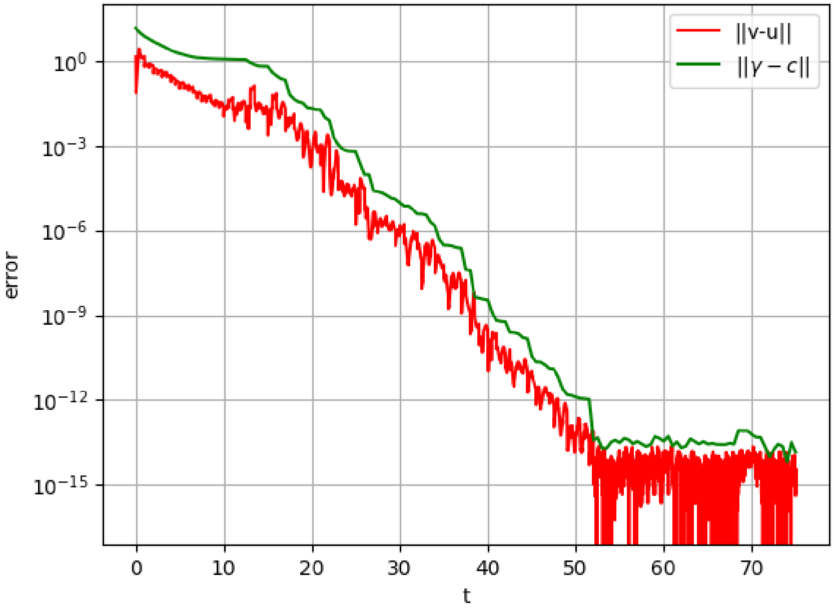

Figure 4.4: Convergence of Newton's Method

Levenberg-Marquardt. Now, the DS Hessian method requires knowledge of the 18 second order sensitivity equations. These could be directly simulated but greatly increases the computational complexity. Instead we move directly to the Levenberg-Marquardt algorithm.

$$c^{(k+1)} = c^{(k)} - (W^T W + \lambda I)^{-1} W^T(\mathbf{v}-\mathbf{u}). \tag{4.7}$$

We experiment with various values of the damping parameter λ in 4.5. We find that the fastest convergence is found with small damping. As such, we choose the parameter only large enough to ensure invertibility of the approximate Hessian. This behavior appears to be constant across the models tested here.

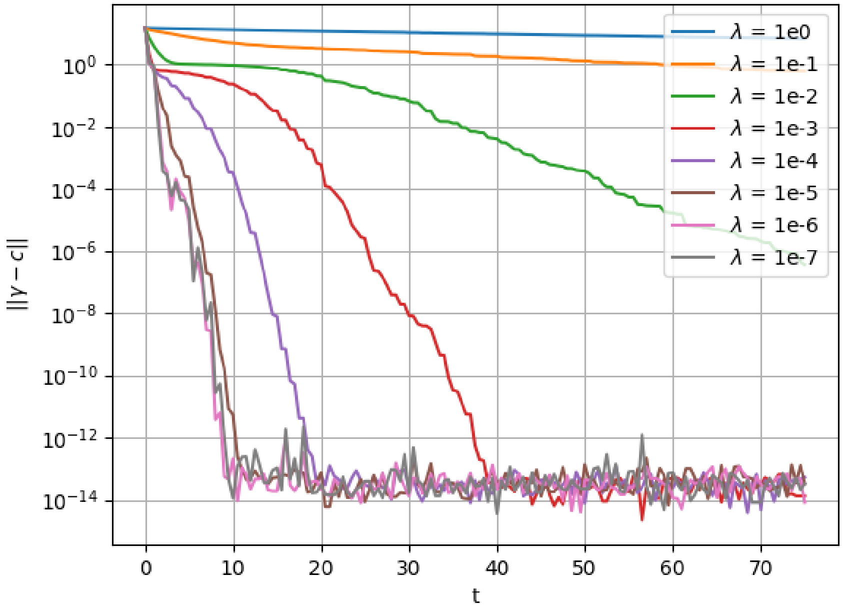

Figure 4.5: DS Levenberg-Marquardt algorithm with various values of λ.

4.1.2 OTF Methods.

Following the asymptotics derived in chapter 2.5, we find the following approximation for W which allows us to define OTF versions of the algorithms,

$$W = \begin{bmatrix} -\frac{1}{\mu_1}(v_1 - v_2) & 0 & 0 \\ 0 & \frac{1}{\mu_2}v_1 & 0 \\ 0 & 0 & -\frac{1}{\mu_3}v_3 \end{bmatrix}. \tag{4.8}$$

This allows us to compare the OTF methods with the DS methods. It can be seen that the OTF Gauss-Newton method agrees with the modification of the CHL algorithm used in [10]. We compare the convergence behavior of the DS and OTF methods in Figure 4.6. We see good agreement between the direct simulation of the sensitivity equations and the perturbative approximation.

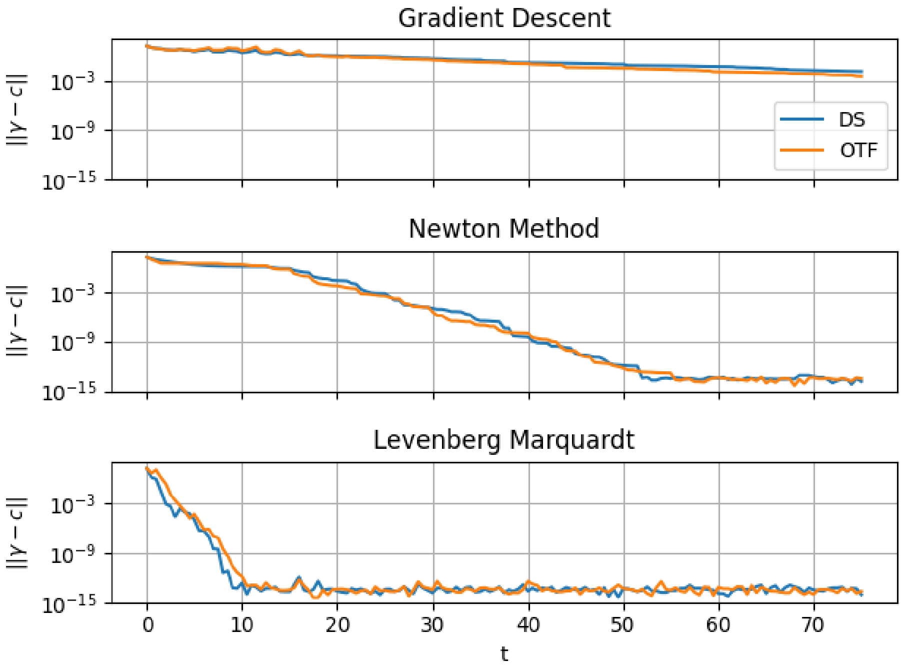

Figure 4.6: Comparison of DS and OTF methods

Interestingly, we find that the OTF method for gradient descent converges faster than the DS method. We shall see below that the OTF method converging just as fast or faster than the DS method is not so uncommon. This may be due to the fact that by doing perturbation theory in large μ, we are getting at the essential long term behavior but by directly simulating the sensitivity equations we are sensitive to the small fluctuations of the error funcitonal. We also see in Figure 4.7 that increasing the value of μ leads to greater agreement between the OTF and DS methods supporting the perturbative approximation.

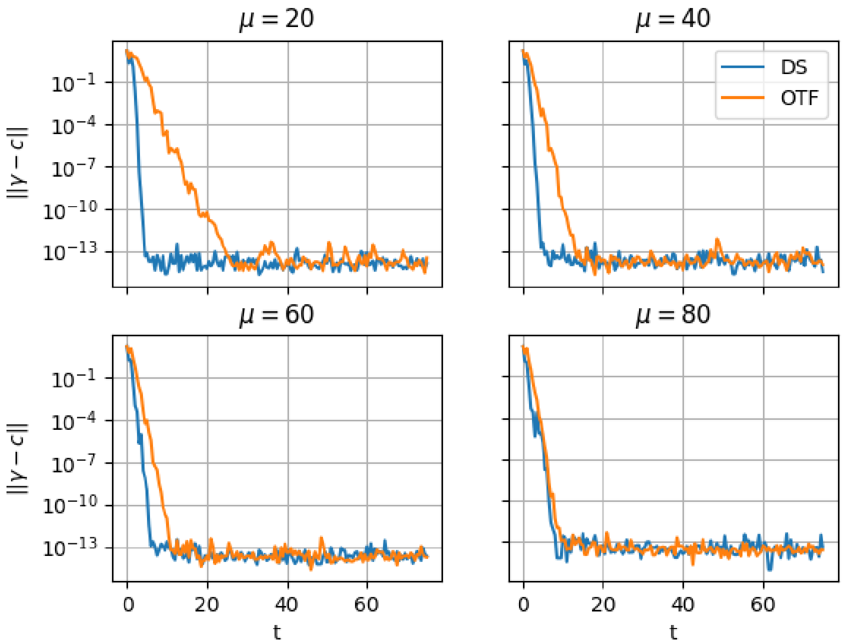

Figure 4.7: DS vs. OTF Levenberg-Marquardt method for various values of μ. Note the similar behavior for larger values of μ in agreement with perturbative results.

4.2 Two Layer Lorenz '96 Model

One thing that is important to understand is the interaction between scales of chaotic dynamical systems. The two layer Lorenz '96 model is a system of ODEs designed to capture both large scale and small scale behavior. It was originally designed to model atmospheric quantities along a single circle of latitude [18]. The dynamics of this model are plotted in Figure 4.8.

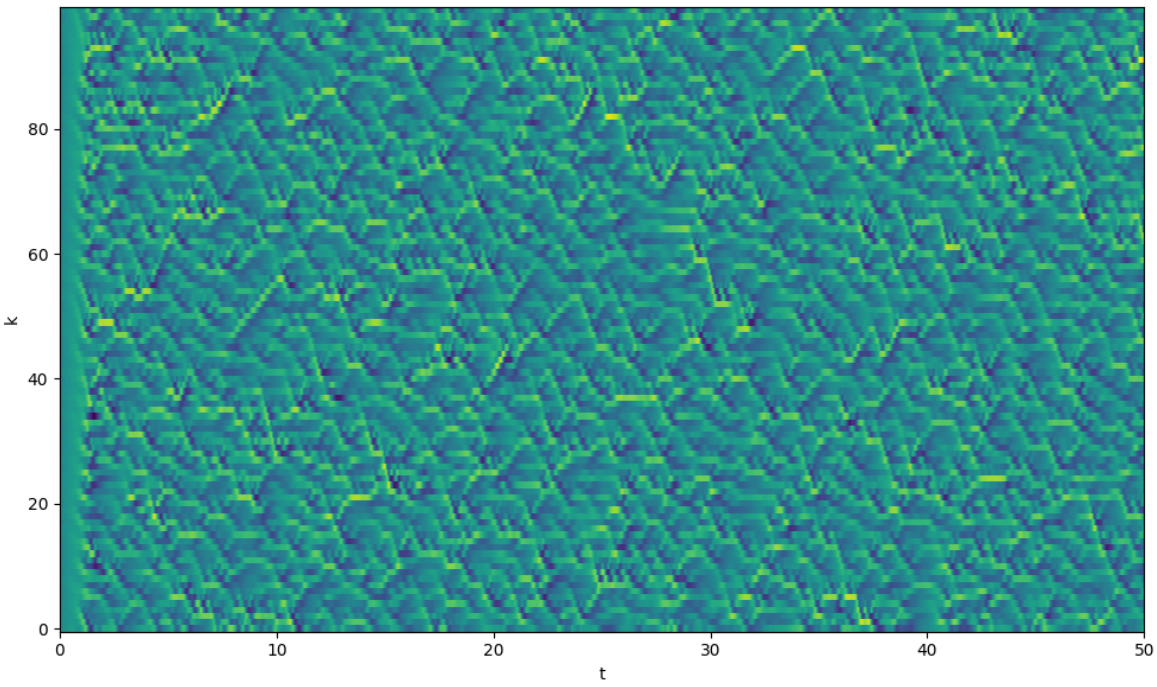

Figure 4.8: Plot of u_k^l wtih $I = 100$, $\gamma = (0.01, 0.5)$

This system has two layers, representing two different scales that are coupled together. u^l captures the large scale behavior of the system, with $k \in \{1, .., I\}$ indexing the position along the latitude circle. Each u_k^l is coupled to J small scale variables u^s, which represent the small scale behavior not accounted for by our course grained system.

$$\begin{aligned} \dot{u}_k^l &= u_{k+1}^l(u_{k-1}^l - u_{k+2}^l) + \gamma_1 \sum_{j=1}^{J} u_{kj}^s u_k^l - \gamma_2 u_k^l + F \\ \dot{u}_{kj}^s, &= -d_j u_{kj}^s - \gamma_1 (u_k^l)^2. \end{aligned} \tag{4.9}$$

Now, consider the nudged system given by

$$\begin{aligned} \dot{v}_k^l &= v_{k+1}^l(v_{k-1}^l - v_{k+2}^l) + c_1 \sum_{j=1}^{J} v_{kj}^s v_k^l - c_2 v_k^l + F - \mu(v_k^l - u_k^l), \\ \dot{v}_{kj}^s &= -d_j v_{kj}^s - c_1 (v_k^l)^2. \end{aligned} \tag{4.10}$$

Our observation operator for this system is the projection onto the large scale variables, i.e. we assume that only the large scale is observable and used for the nudging. Since the parameters d_j only appear in the small scale equations, they only directly interact with the unobservable variable, for simplicity we consider them as fixed parameters. We take them as equal to : $(d_j) = (0.2, 0.5, 1, 2, 5)$. We take the true parameter values as $\gamma = (0.01, 0.5)$, with starting approximate parameters as $c = \frac{1}{2}\gamma$. We also take $J = 5$ and $I = 40$. We take our nudging parameter as $\mu = 50$. Our initial conditions for u are randomly drawn from a normal distribution, and we take 0 initial conditions for v.

Convergence of Nudging on Large Scales. Because the parameter estimation is implemented within the nudging framework, we must check that the nudging algorithm converges for this model. We see in Figure 4.9 below that nudging with only the large scales observed leads to convergence of the entire system.

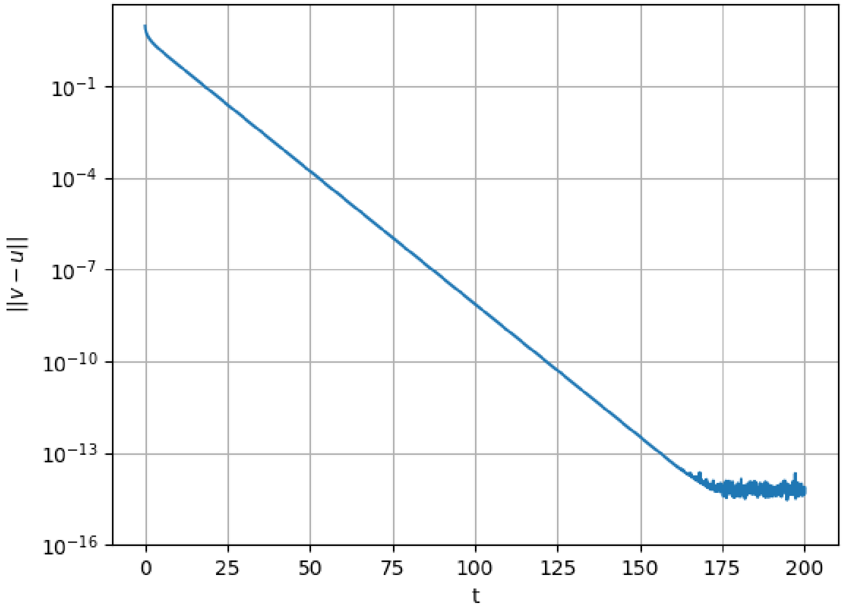

Figure 4.9: Convergence of nudging with parameters fixed at “correct” values. Only the larges scale variables are observed.

4.2.1 DS Methods.

Due to the convergence of the nudging we only need to minimize over $E = \frac{1}{2}\|P(v-u)\|^2\big|_{t=T} = \frac{1}{2}\|v^l - u^l\|^2\big|_{t=T}$. The sensitivity equations for Lorenz ’96 are given by

$$\dot{w}^l_{k,i} = w^l_{k+1,i}(v^l_{k-1} - v^l_{k+2}) + v^l_{k+1}(w^l_{k-1,i} - w^l_{k+2,i}) + \delta_{1i}\sum_{j=1}^{J} v^s_{kj} v^l_k,$$

$$+ c_1 \sum_{j=1}^{J} w^s_{kj,i} v^l_k + c_1 \sum_{j=1}^{J} v^s_{kj} w^l_{k,i} - \delta_{2i} v^l_k - c_2 w^l_{k,i} - \mu w^l_{k,i}$$

$$\dot{w}^s_{kj,i} = -d_j w^s_{kj,i} - \delta_{1,i}(v^l_k)^2 - 2c_1 v^l_k w^l_{k,i}. \tag{4.11}$$

In this model, for simplicity, we consider only the implementation of Newton's method and the Levenberg-Marquardt algorithm. We compare these algorithms in Figure 4.10. Though we should note that the algorithm does not always converge for reasons discussed in Section 4.2.3.

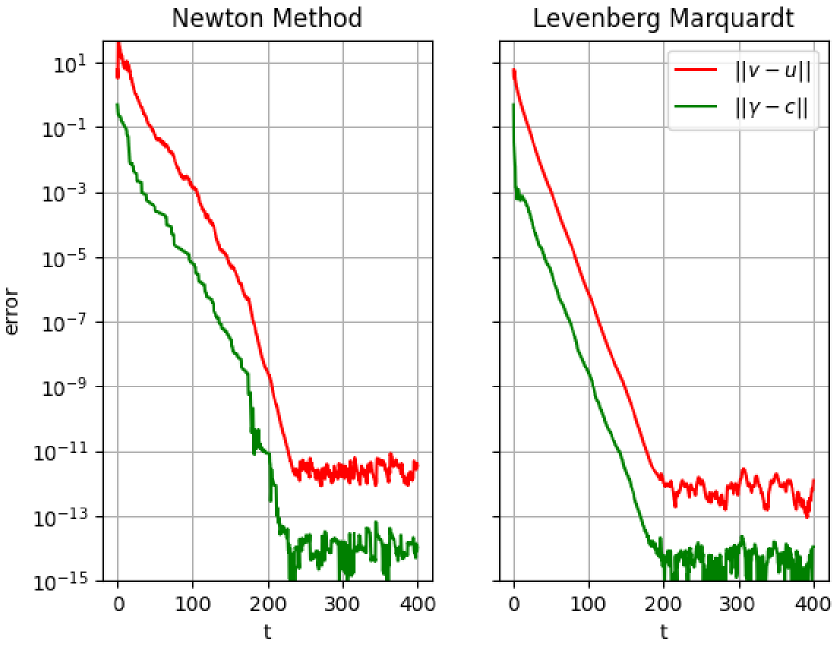

Figure 4.10: Comparison of convergence between two algorithms.

4.2.2 OTF Methods.

The perturbation theory developed earlier leads us to the following estimates for the sensitivities as $\mu \to \infty$.

$$\begin{aligned} w^l_{k,1} &\sim \frac{1}{\mu}\sum_{j=1}^{J} v^s_{kj} v^l_k, \\ w^l_{k,2} &\sim -\frac{1}{\mu} v^l_k. \end{aligned} \tag{4.12}$$

the following section shows how the parameter update algorithms are very sensitive to the hyper parameters for this system.

4.2.3 Sensitivity to Hyperparameters and Possible Failure of Convergence.

The parameter estimation algorithms in this model seem to have a good deal of sensitivity on the hyper parameters μ and Δt. We demonstrate this in Figure 4.11. From directly inspecting the results of the algorithms we find that the algorithm occasionally converges to exactly $(c_1, c_2) = (-\gamma_1, \gamma_2)$. It is not clear exactly what is going on, though it seems to indicate that the error function has a local minima or saddle point at this value. This is likely related to the fact that we are fitting the parameter value for the nonlinear term, as c_1 is the parameter which does not converge.

This is what happens in Figure 4.11 when the parameter estimation algorithm does not converge.

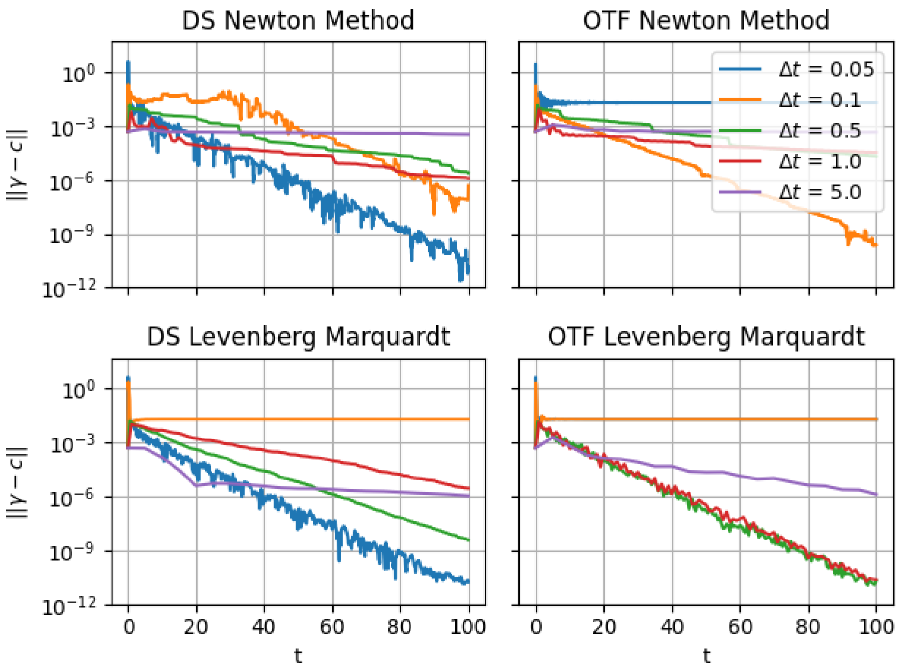

Figure 4.11: Sensitivity to parameter update interval Δt

This appears to be a feature of this specific model. Understanding this sensitivity to the hyper parameters requires further investigation.

4.3 Kuramoto-Sivashinsky Equation

Having shown the applicability of these methods in several finite dimensional systems, we now turn to the infinite dimensional case to study how these methods can be applied to the PDE case. We consider the 1 dimensional Kuramoto-Sivashinsky Equation (KSE) system

$$\dot{u} + \gamma_1 u'' + \gamma_2 u u' + \gamma_3 u^{(4)} = 0 \tag{4.13}$$

here we have $u' = \frac{\partial u}{\partial x}$. The KSE system can model a variety of physical systems. Generally, it models systems far from equillibrium such as turbulent behavior of a single flame, instabilities in reaction diffusion systems, and the flow of plasmas [15, 3]. It is mathematically interesting as a model because it is an example of a chaotic PDE with only one spatial dimension, and as such, is relatively computationally inexpensive to simulate. Despite its computational cheapness the KSE system is still displays a wider variety of interesting dynamics. We plot the dynamics of the KSE system in Figure 4.12.

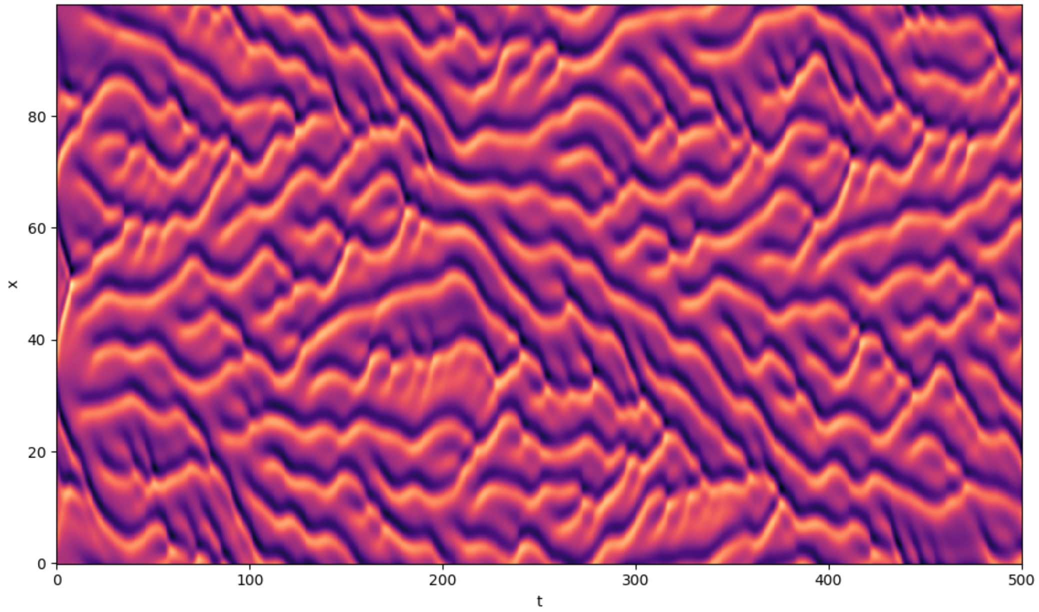

Figure 4.12: Plot of $u(x,t)$

Parameter estimation in the KSE system was studied by Pachev, Whitehead, and Mcquarrie using a different approach than the one used here [22]. The corresponding nudged system is given by

$$\dot{v} + c_1 v'' + c_2 v v' + c_3 v^{(4)} + \mu P_N(v - u) = 0. \tag{4.14}$$

Here our projection operator, P_N, is the fourier projection onto Fourier modes less than N. The inner products below are all given in the L^2 norm. Here we use $N = 32$. Because we have three parameters to fit, DS methods would greatly increase the computational complexity. We shall consider first fitting only c_1 where DS methods do not greatly increase the computational cost. This allows us to compare the DS methods with the OTF methods. After, this we move on to multi parameter estimation using OTF methods and investigate the efficacy of the parameter estimation algorithms.

4.3.1 Single Parameter Estimation. For simplicity and computational efficiency we consider only DS for single parameter estimation. In this case we only have one first order and second order sensitivity equation each. They are given by,

$$\dot{w} + v'' + c_1 w'' + c_2(wv)' + c_3 w^{(4)} + \mu P_N w = 0, \tag{4.15}$$

$$\dot{z} + 2w'' + c_1 z'' + 2c_2 ww' + c_2(zv)' + c_3 z^{(4)} + \mu P_N z = 0. \tag{4.16}$$

Numerical results. We take the true parameter values $\gamma = (1, 1, 1)$, and the hyper parameters $\Delta t = 1$ and $\mu = 25$. We take initial conditions

$$\begin{aligned} u|_{t=0} = \sin(6\pi x/L) + 0.1\cos(\pi x/L) - 0.2\sin(3\pi x/L) \\ + 0.05\cos(15\pi x/L) + 0.7\sin(18\pi x/L) - \cos(13\pi x/L). \end{aligned} \tag{4.17}$$

where our domain is $[0, L] = [0, 100]$, and we take the intial conditions of v to be zero. We also assume periodic boundary conditions for simplicity.

We can directly simulate the sensitivity equations and use this for parameter estimation. This allows us to implement parameter estimation algorithms. We also simulate the second order sensitivity equation which allows us to implement the Hessian method. We simulate the convergence of these algorithms in Figure 4.13.

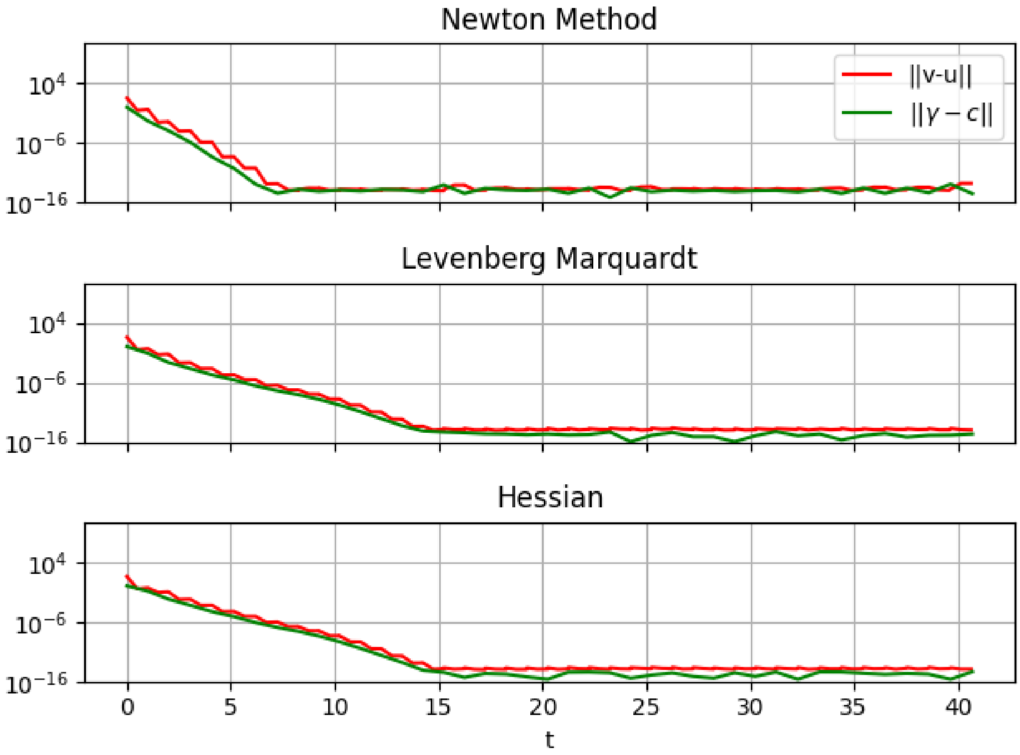

Figure 4.13: Convergence of DS algorithms with initial value $c_1 = 2$.

We see that Newton's method has the fastest convergence rate. We see similar convergence rates for the Hessian method and Levenberg-Marquardt method, justifying the approximation of the Hessian.

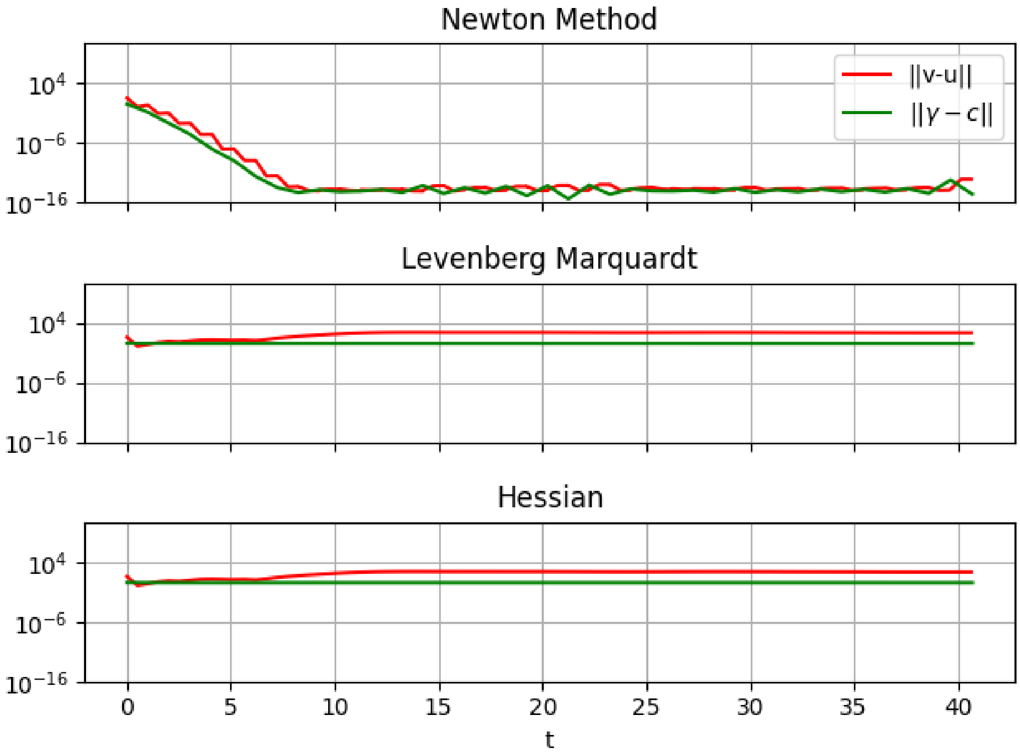

Figure 4.14: Non-convergence of optimization algorithms with initial value $c_1 = 4.5$. The optimization algorithms converge to a $c_1 \approx 4.25$.

We also see in Figure 4.14 that for specific initial guesses of the unknown parameters Levenberg-Marquardt and the Hessian method do not converge. This is likely due to the fact that they only converge to a critical point and may get stuck in a local minima or saddle point. Whereas, our error function only has one root. We compare the DS and OTF methods in Figure 4.15. We discuss the similarity between the CHL algorithm and the Gauss-Newton algorithm in the next section.

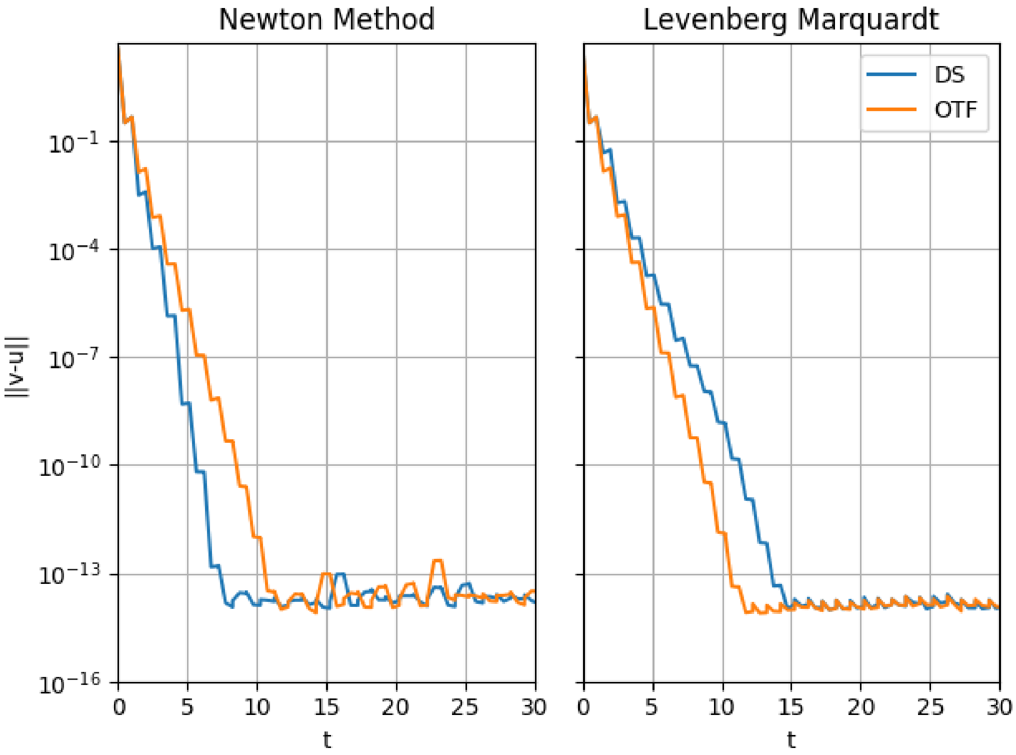

Figure 4.15: Comparison of OTF and DS methods

OTF Methods: Comparison Between CHL and Gauss-Newton. As explained in Section 3.1.2, the OTF Newton's method for a single parameter reproduces the CHL algorithm. Interestingly, the Levenberg-Marquardt algorithm with $\lambda = 0$ (i.e. the Gauss-Newton Algorithm) has a very similar form to the CHL algorithm. To see this comparison we look at them next to each other

$$c_{1,\text{CHL}}^{k+1} = c_1^k + \mu \frac{\|P[v-u]\|^2}{\langle v-u, P[v'']\rangle} \tag{4.18}$$

$$c_{1,\text{GN}}^{k+1} = c_1^k + \mu \frac{\langle v-u, P[v'']\rangle}{\|P[v'']\|^2}. \tag{4.19}$$

These are clearly very similar. They have the same sign, and roughly the same magnitude. For example, if the range of P were one dimensional they would exactly agree. However, note that the Gauss-Newton algorithm has a denominator which is easier to control, i.e. it is harder for it to be zero and it always has definite sign. From numerical experimentation shown in Figure 4.16, we see that both algorithm's have similar convergence rates for single parameter estimation.

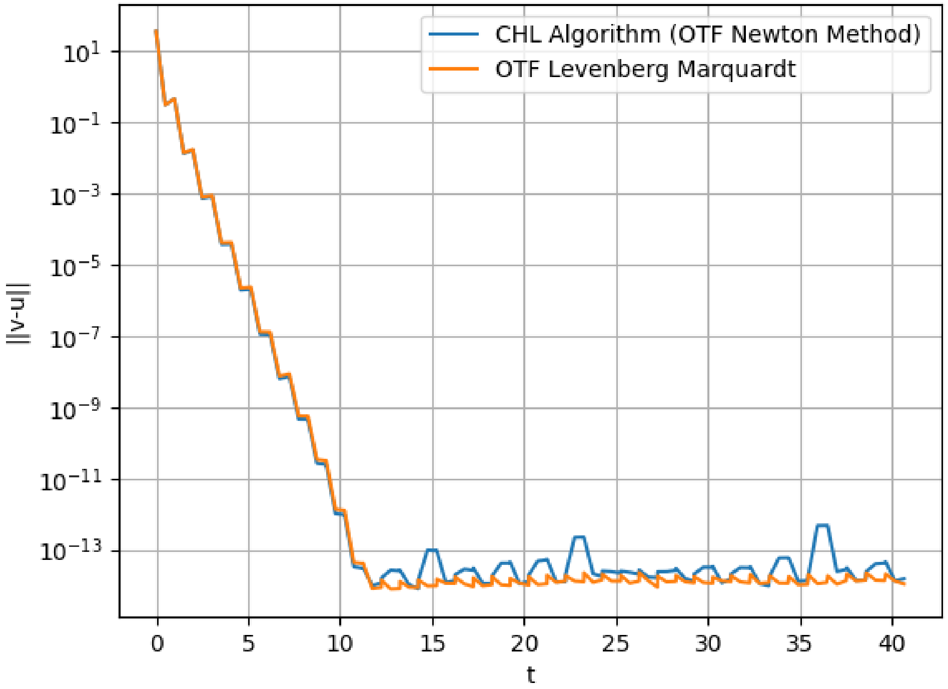

Figure 4.16: Comparison of CHL algorithm and OTF Levenberg-Marquardt.

4.3.2 OTF Methods For Multiparameter Estimation. We now move on to estimating all three parameters. We shall see, as before, that Levenberg-Marquardt generalizes very well for multiparameter estimation. The full sensitivity equations for this system are given by,

$$\dot{w}_i + \delta_{1i}v'' + \delta_{2i}vv' + \delta_{3i}v^{(4)} + c_1 w_i'' + c_2 w_i v' + c_2 v w_i' + c_3 w_i^{(4)} + \mu P_N w_i = 0. \tag{4.20}$$

We can also work out the second order sensitivity equations

$$\begin{aligned} &\dot{z}_{ij} + 2\delta_{1(i}w_{j)}'' + 2\delta_{2(i}w_{j)}v' + 2\delta_{2(i}w_{j)}'v + 2\delta_{3(i}w_{j)}^{(4)} + 2c_2 w_{(i}w_{j)}', \\ &+ c_1 z_{ij}'' + c_2 z_{ij}v' + c_2 v z_{ij}' + c_3 z_{ij}^{(4)} + \mu P_N z_{ij} = 0. \end{aligned} \tag{4.21}$$

The perturbation theory derived earlier gives us the simple approximations

$$w_i^{OTF} \sim -\frac{1}{\mu}\delta_{1i}v'' - \frac{1}{\mu}\delta_{2i}vv' - \frac{1}{\mu}\delta_{3i}v^{(4)}, \tag{4.22}$$

which can be written in vector form as

$$\begin{bmatrix} w_i \\ w_2 \\ w_3 \end{bmatrix} \sim -\frac{1}{\mu}\begin{bmatrix} v'' \\ vv' \\ v^{(4)} \end{bmatrix}, \tag{4.23}$$

and

$$\mu^2 z_{ij} \sim 2\delta_{1(i}\delta_{1j)}v^{(4)} + 2\delta_{1(i}\delta_{2j)}\left[(vv')'' + v''v' + v^{(3)}v\right] + 2\delta_{1(i}\delta_{3j)}\left[2v^{(6)}\right] + 2\delta_{2(i}\delta_{2j)}\left[v(v')^2 + (vv')'v\right] + 2\delta_{2(i}\delta_{3j)}v^{(4)}v' + 2\delta_{2(i}\delta_{3j)}\left[v^{(5)}v + (vv')^{(4)}\right] + 2\delta_{3(i}\delta_{3j)}v^{(8)}. \tag{4.24}$$

This gives the following approximation for the Hessian

$$D^2 f^{OTF} = \frac{1}{\mu}\begin{bmatrix} \|Pv''\|^2 + 2\langle v-u, P[v^{(4)}]\rangle & \langle v'', P[vv']\rangle + \langle v-u, P\left[(vv')'' + v''v' + v^{(3)}v\right]\rangle & \langle v'', Pv^{(4)}\rangle + \langle v-u, P[2v^{(6)}]\rangle \\ . & \|P[vv']\|^2 + 2\langle v-u, P\left[v(v')^2 + (vv')'v\right]\rangle & \langle v^{(4)}, P[vv']\rangle + \langle v-u, P\left[v^{(5)}v + (vv')^{(4)}\right]\rangle \\ . & . & \|P[v^{(4)}]\|^2 + 2\langle v-u, P[v^{(8)}]\rangle \end{bmatrix}. \tag{4.25}$$

This may be quite complicated to compute in practice given the number of high order derivatives involved. The Levenberg-Marquardt approximation to the Hessian is given by

$$D^2 f_{LM}^{OTF} = \frac{1}{\mu}\begin{bmatrix} \|Pv''\|^2 + \lambda & \langle v'', P[vv']\rangle & \langle v'', P[v^{(4)}]\rangle \\ . & \|P[vv']\|^2 + \lambda & \langle v^{(4)}, P[vv']\rangle \\ . & . & \|P[v^{(4)}]\|^2 + \lambda \end{bmatrix}. \tag{4.26}$$

We take our initial guess $c^{(0)} = (2, 2, 2)$. We take the true parameter values $\gamma = (1, 1, 1)$, and the hyper parameters $\Delta t = 0.5$ and $\mu = 25$. A brief summary of our result is that Newton's method does converge but very slowly, and Levenberg-Marquardt (with $\lambda = 10^{-6}$) converges very rapidly. We show the convergence of both algorithms in Figures 4.17 and 4.18.

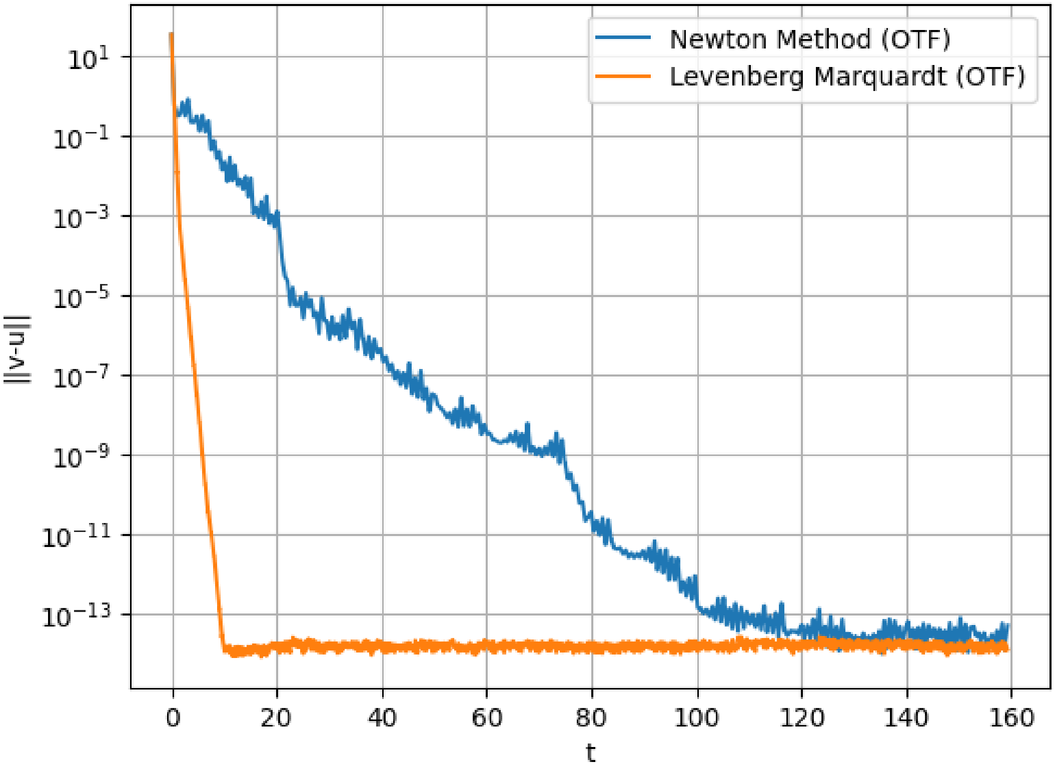

Figure 4.17: Comparison of algorithms for estimating (c_1, c_2, c_3) simultaneously.

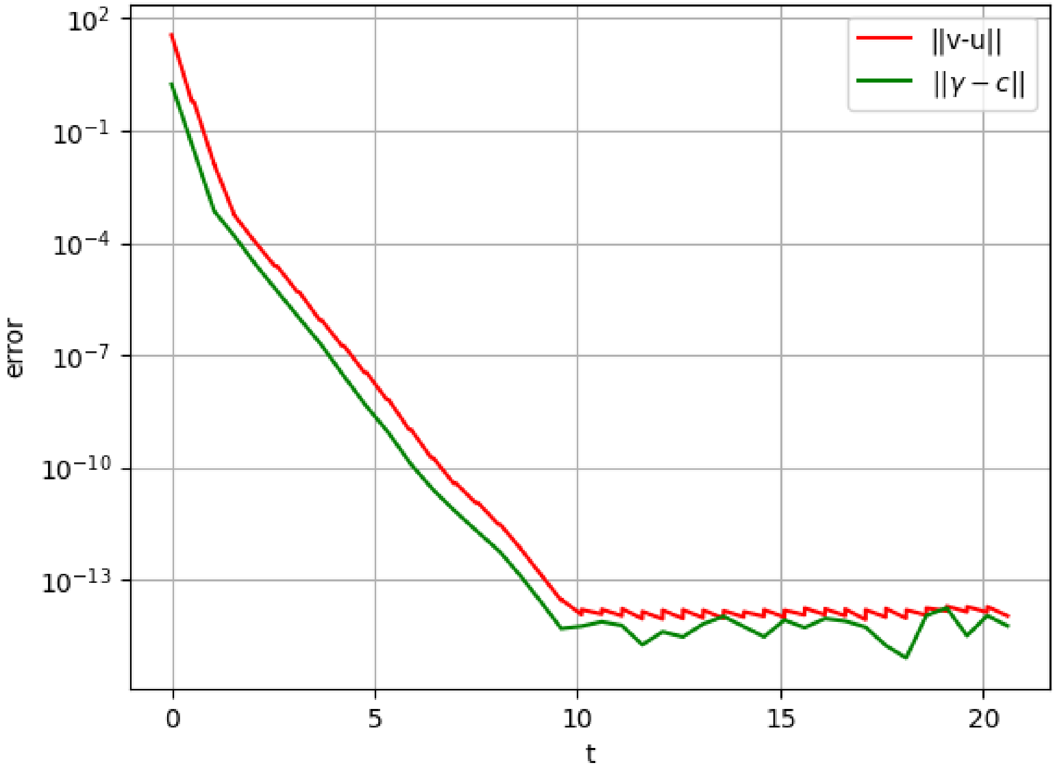

Figure 4.18: Rapid convergence of OTF Levenberg-Marquardt for multi-parameter estimation.

4.3.3 Parameter Optimization vs. Parameter Root Finding . We consider the case where the equations of motions are altered by adding additional higher order terms. Consider the modification of the true system given by

$$\dot{u} + \gamma_1 u'' + \gamma_2 uu' + \gamma_3 u^{(4)} - \epsilon u^{(6)} = 0. \tag{4.27}$$

We take ϵ to be fixed. We use this to stand in for the case where the true system is either not known exactly, or too complicated to simulate fully. For example, in a different system, the Navier Stokes equations are only a large scale approximation to the true motion of molecules in fluids which is far to complex to simulate numerically. In fact, every physical model we know of is some large scale approximation of whatever the underlying theory of the universe is. Now, given this modification of the true system, our goal is estimate (c_1, c_2, c_3) on the unmodified nudged system.

We now consider parameter estimation in this regime. We take as initial values the $c_i = \gamma_i = 1$ and we then look at the effect of implementing the parameter update schemes described in this paper. We see in Figures 4.19 and 4.20 that the Levenberg-Marquardt algorithm effectively reduces the error, but as would be expected Newton's root finding method is no longer effective. Note that although we cant expect the error to converge to zero in this case, it does lower the state error by a factor of 3 or 4.

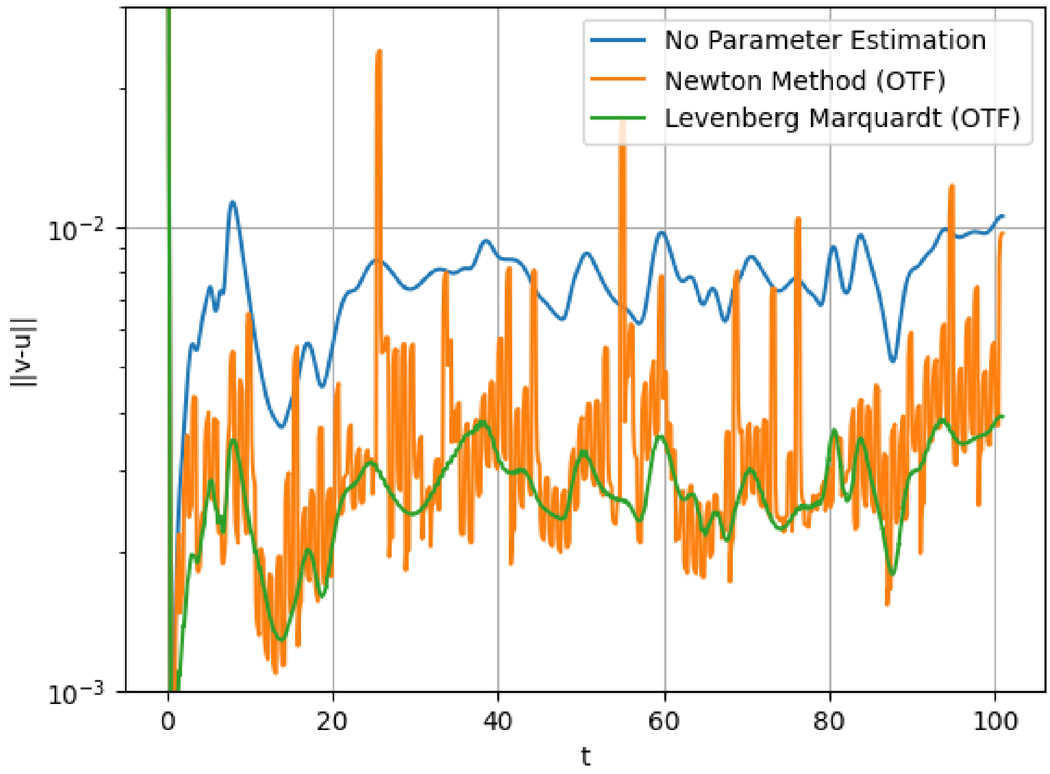

Figure 4.19: Parameter optimization with $\epsilon = 10^{-3}$.

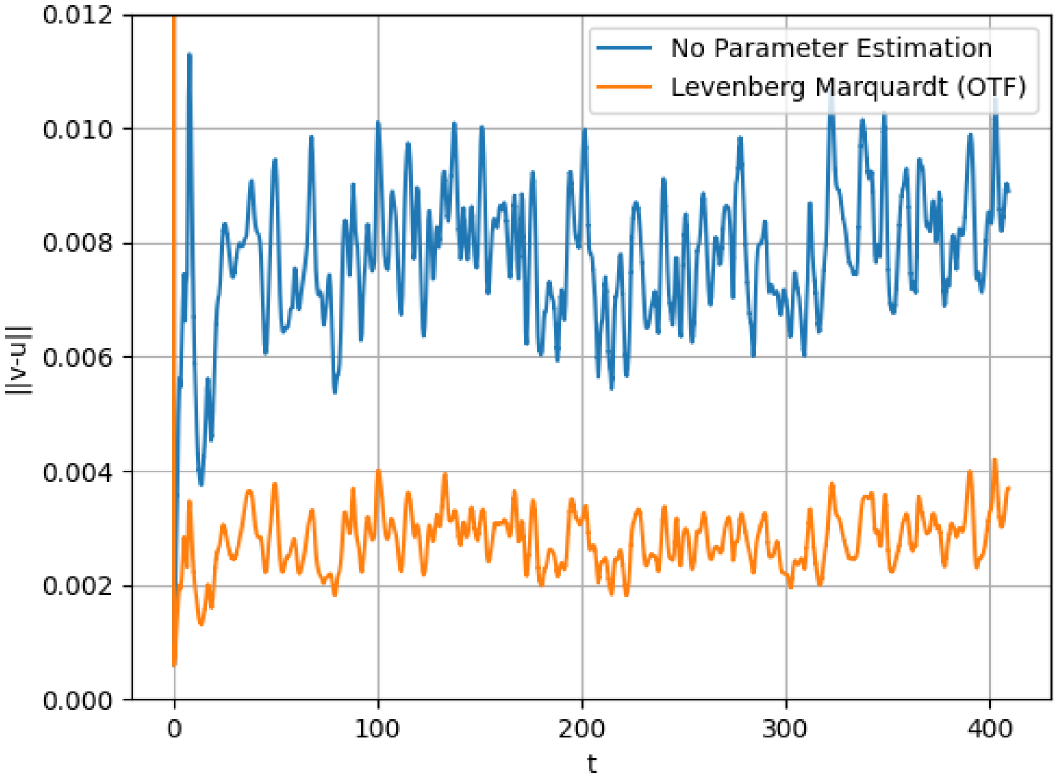

Figure 4.20: Non logarithmic plot of optimization with $\epsilon = 10^{-3}$

Chapter 5. Conclusion

5.0.1 Summary. We outline an original framework for parameter estimation within the continuous time data assimilation approach. Starting from some basic assumptions, motivated by numerical experimentation, we see that we can treat the problem as a finite dimensional optimization problem. We consider the derivative of our error functional with respect to parameters, which leads us to considering the sensitivity equations. We show that within the nudging framework, we can get perturbative approximations for the solutions to the sensitivity equations. Given these estimates, we can derive on “the fly” parameter estimation algorithms. In particular, we show that the OTF version of Newton’s method for one parameter reproduces the CHL algorithm. In addition, we develop several new parameter estimation algorithms, the most prominent of these being an implementation of the Levenberg-Marquardt algorithm. We show that for single parameter estimation, this leads to a modification of the CHL algorithm, which generalizes well to multi-parameter estimation. This algorithm also seems to compare well with another algorithm developed by Pachev, Whitehead, and McQuarrie for multi-parameter estimation because it does not require estimation of the time derivative of u [22]. Though we leave a direct comparison to this algorithm to future work.

The derivations given in this thesis are not given fully rigorously. However we have outlined the method very generally, but it also means that several assumption have to be made to carry out the derivation. We leave the fully rigorous analysis for future work, which will likely have to be carried out on a case by case basis. Instead of relying on rigorous analysis for our results, we instead demonstrate their efficacy numerically. We demonstrate the application of these methods in three cases of increasing complexity. Specifically we look at: the Lorenz '63 system, the two layer Lorenz '96 system, and the one dimensional Kuramoto-Sivashinsky equation. In all cases, we find that the methods converge for suitable choices of hyper parameters and initial conditions.

5.0.2 Next Steps. As mentioned before, further analysis is required to make the methods outlined here fully rigorous. In the finite dimensional case not this should be relatively straightforward, whereas the infinite dimensional case would require a considerable amount of further work . A rigorous analysis would also be interesting because it would allow us to analyze the convergence rate of our algorithm. Heuristically we can guess that the convergence rate should follow from the convergence of the finite dimensional optimization algorithms we implement, but there may be more subtlety. Additionally, it would be interesting to find areas where the algorithm does not converge similar to the case of the Lorenz '96 model discussed in this paper.

Additionally, it would be interesting to see if the sensitivity equations could be used to give more insight in to the algorithm developed by Pachev Whitehead and McQuarrie [22]. There are hints of a potential connection. For example, the matrix that is inverted in their algorithm is the same as the Gauss-Newton approximation of the Hessian.

Another interesting avenue of future work is to explore how we could extend the applicability of the AOT nudging algorithm. An important drawback of the current algorithm is that it is not clear how to handle noise in the measurement. In particular, if our observation operator includes a stochastic term, then choosing a large nudging parameter, μ, will amplify this noise. A proof for the convergence of nudging with noisy observations does exist in a specific case [6], but there is no general understanding of how to handle noisy observations. This is crucial because an important assumption of our derivations was that they were happening within the nudging framework. Thus, by extending the robustness and generality of the nudging, we can extend the applicability of these parameter estimation algorithms.

Bibliography

[1] Débora A. F. Albanez, Helena J. Nussenzveig Lopes, and Edriss S. Titi. Continuous data assimilation for the three-dimensional navier-stokes-α, 2014.

[2] M. U. Altaf, E. S. Titi, O. M. Knio, L. Zhao, M. F. McCabe, and I. Hoteit. Downscaling the 2d benard convection equations using continuous data assimilation, 2015.

[3] Gi Siv Ashinsky. Nonlinear analysis of hydrodynamic instability in laminar flames—i. derivation of basic equations. In *Dynamics of Curved Fronts*, pages 459–488. Elsevier, 1988.

[4] Abderrahim Azouani, Eric Olson, and Edriss S. Titi. Continuous data assimilation using general interpolant observables. *Journal of Nonlinear Science*, 24(2):277–304, APR 2014.

[5] Dimitri P Bertsekas. Nonlinear programming. *Journal of the Operational Research Society*, 48(3):334–334, 1997.

[6] Hakima Bessaih, Eric Olson, and Edriss S Titi. Continuous data assimilation with stochastically noisy data. *Nonlinearity*, 28(3):729–753, February 2015.

[7] Animikh Biswas and Randy Price. Continuous data assimilation for the three dimensional navier-stokes equations, 2020.

[8] Jordan Blocher, Vincent R Martinez, and Eric Olson. Data assimilation using noisy time-averaged measurements. *Physica D: Nonlinear Phenomena*, 376:49–59, 2018.

[9] Elizabeth Carlson, Joshua Hudson, and Adam Larios. Parameter recovery for the 2 dimensional navier-stokes equations via continuous data assimilation. *SIAM Journal on Scientific Computing*, 42(1):A250–A270, 2020.

[10] Elizabeth Carlson, Joshua Hudson, Adam Larios, Vincent R. Martinez, Eunice Ng, and Jared P. Whitehead. Dynamically learning the parameters of a chaotic system using partial observations, 2021.

[11] Robert P Dickinson and Robert J Gelinas. Sensitivity analysis of ordinary differential equation systems—a direct method. *Journal of computational physics*, 21(2):123–143, 1976.

[12] Yi Juan Du and Ming-Cheng Shiue. Analysis and computation of continuous data assimilation algorithms for lorenz 63 system based on nonlinear nudging techniques. *Journal of Computational and Applied Mathematics*, 386:113246, 2021.

[13] Aseel Farhat, Michael S Jolly, and Edriss S Titi. Continuous data assimilation for the 2d bénard convection through velocity measurements alone. *Physica D: Nonlinear Phenomena*, 303:59–66, 2015.

[14] Marco Frontini and E Sormani. Modified newton's method with third-order convergence and multiple roots. *Journal of computational and applied mathematics*, 156(2):345–354, 2003.

[15] Yoshiki Kuramoto. Diffusion-induced chaos in reaction systems. *Progress of Theoretical Physics Supplement*, 64:346–367, 1978.

[16] Kody Law, Andrew Stuart, and Kostas Zygalakis. Data assimilation. *Cham, Switzerland: Springer*, 214:52, 2015.

[17] Edward N Lorenz. Deterministic nonperiodic flow. *Journal of atmospheric sciences*, 20(2):130–141, 1963.

[18] Edward N Lorenz. Predictability: A problem partly solved. In *Proc. Seminar on predictability*, volume 1. Reading, 1996.

[19] Donald W Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, 11(2):431–441, 1963.

[20] Vincent R Martinez. Convergence analysis of a viscosity parameter recovery algorithm for the 2d navier–stokes equations. *Nonlinearity*, 35(5):2241–2287, April 2022.

[21] Peter David Miller. *Applied asymptotic analysis*, volume 75. American Mathematical Soc., 2006.

[22] Benjamin Pachev, Jared P. Whitehead, and Shane A. McQuarrie. Concurrent multiparameter learning demonstrated on the kuramoto-sivashinsky equation, 2022.