

# Taming of Complex Dynamical Systems

Alexander R. Grimm

Thesis submitted to the Faculty of the  
Virginia Polytechnic Institute and State University  
in partial fulfillment of the requirements for the degree of

Master of Science  
in  
Mathematics

John A. Burns, Chair  
Serkan Gugercin  
Lizette Zietsman

August 5, 2013  
Blacksburg, Virginia

Keywords: Taming, Existence, Uniqueness, Dynamical Systems, Navier Stokes, Burger's equation, Finite Elements

Copyright 2013, Alexander R. Grimm

# Taming of Complex Dynamical Systems

Alexander R. Grimm

(ABSTRACT)

The problem of establishing local existence and uniqueness of solutions to systems of differential equations is well understood and has a long history. However, the problem of proving global existence and uniqueness is more difficult and fails even for some very simple ordinary differential equations. It is still not known if the  $3D$  Navier-Stokes equation have global unique solutions and this open problem is one of the Millennium Prize Problems. However, many of these mathematical models are extremely useful in the understanding of complex physical systems. For years people have considered methods for modifying these equations in order to obtain models that still capture the observed fundamental physics, but for which one can rigorously establish global results. In this thesis we focus on a taming method to achieve this goal and apply taming to modeling and numerical problems. The method is also applied to a class of nonlinear differential equations with conservative nonlinearities and to Burgers' Equation with Neumann boundary conditions. Numerical results are presented to illustrate the ideas.

# Contents

<b>List of Figures</b>	<b>vi</b>
<b>List of Tables</b>	<b>viii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Introduction and Outline . . . . .	2
1.2 A Brief Summary of the NS Equations . . . . .	3
1.2.1 Physical Motivation . . . . .	3
1.2.2 A Derivation of Navier-Stokes Equations . . . . .	5
1.2.3 The Dimensionless Navier-Stokes Equation . . . . .	7
1.2.4 A short note on the Reynolds number . . . . .	8
1.3 Motivation . . . . .	8
1.3.1 Navier-Stokes Equations and Properties . . . . .	9
1.3.2 Taming . . . . .	11
1.4 Main Goal of This Thesis . . . . .	12
<b>2 Introduction to Dynamical Systems</b>	<b>14</b>
2.1 Discrete Dynamical Systems . . . . .	15
2.2 Dynamical Systems Generated by Differential Equations . . . . .	15
2.2.1 Equilibrium Solutions and Stability . . . . .	17
2.3 Limit sets and Attractors . . . . .	20
2.4 Dissipative Dynamical Systems . . . . .	21

2.5	Invariant Manifolds . . . . .	22
2.6	Chaos . . . . .	23
<b>3</b>	<b>The Models</b>	<b>25</b>
3.1	The BDT Models . . . . .	25
3.1.1	Properties of the BDT model . . . . .	26
3.2	Burgers equation . . . . .	34
3.2.1	Properties of the Burgers equation . . . . .	34
3.2.2	Solution Techniques . . . . .	34
3.2.3	Boundary conditions . . . . .	35
3.3	Review of Common Model Features . . . . .	36
<b>4</b>	<b>Taming for the BDT Model</b>	<b>38</b>
4.1	A Two State model . . . . .	38
4.1.1	Numerical Simulations . . . . .	40
4.2	Three state model . . . . .	44
4.2.1	Numerical Studies . . . . .	46
4.2.2	The Taming Term . . . . .	46
4.2.3	Periodic Orbits . . . . .	54
<b>5</b>	<b>Taming for Burgers Equation</b>	<b>56</b>
5.1	Burgers' Equation . . . . .	56
5.2	Numerical Approximations . . . . .	57
5.2.1	The Finite Element Method . . . . .	57
5.2.2	Finite Elements for Burgers Equation . . . . .	60
5.3	Finite Elements on Burgers equation <i>with</i> Taming . . . . .	63
5.4	Taming on the Finite Element Space . . . . .	65
5.4.1	Numerical simulations . . . . .	66
<b>6</b>	<b>Conclusions and Future Directions</b>	<b>76</b>

<b>A Bibliography</b>	<b>78</b>
<b>B Matlab Code</b>	<b>83</b>
B.1 The Lorenz system . . . . .	83
B.2 Simulating the BDT model . . . . .	84
B.3 Fitting the ellipsoid . . . . .	87
B.4 Burgers Equations & Finite Elements . . . . .	89
B.5 Matlabs pdepe solver . . . . .	96

# List of Figures

2.1	A view of the chaotic lorenz attractor with solutions to two initial points . . .	24
2.2	Time plots of x- and z-component of the chaotic orbits . . . . .	24
3.1	Solutions converging to zero for small initial conditions . . . . .	27
3.2	A solution of the BDT model going to zero with the according time plot . .	28
3.3	Slightly modified initial conditions, no longer converging to zero . . . . .	30
3.4	Views of a chaotic orbit for the BDT model. . . . .	31
3.5	Time plots of a chaotic orbit . . . . .	32
3.6	Time plots of an unbounded orbit, x-value and norm . . . . .	33
4.1	The global attractor . . . . .	41
4.2	The global attractor in the tamed case . . . . .	42
4.3	Orbits in the two-state system (no taming) . . . . .	43
4.4	Orbits in tamed two-state systemv . . . . .	44
4.5	Fitted ellipsoid to solution with least squares fitting . . . . .	49
4.6	Scaled fitted ellipsoid later used for taming . . . . .	50
4.7	Scalar product of original equation on boundary of scaled ellipsoid, scale 6 .	51
4.8	Scalar product of original equation on boundary of scaled ellipsoid, scale 12 .	52
4.9	Scalar product of <b>tamed</b> equation on boundary of scaled ellipsoid, scale 6 .	53
4.10	Scalar product of <b>tamed</b> equation on boundary of scaled ellipsoid, scale 12 .	54
4.11	Periodic orbits in the tamed and untamed case . . . . .	55
5.1	FiniteElement Nodes, N=20 . . . . .	59

5.2	Initial function used . . . . .	67
5.3	Solution to Burgers equation with Matlab <i>pdepe</i> . . . . .	68
5.4	Solution to Burgers with FEM, no taming . . . . .	68
5.5	Solution to Burgers with FEM, with initial taming using cutoff function . . .	69
5.6	Solution to Burgers with FEM, no taming . . . . .	70
5.7	Solution to tamed Burgers with FEM (projected taming term) . . . . .	70
5.8	Solution to Burgers with 3rd order taming on FEM . . . . .	71
5.9	Solution to Burgers with tamed FEM (cutoff taming term) . . . . .	71
5.10	Solution to Burgers with tamed FEM (projected taming term) . . . . .	72
5.11	Solution to Burgers with tamed FEM (3rd order taming term) . . . . .	73
5.12	Solution to Burgers with tamed FEM (cutoff taming term) . . . . .	74

# List of Tables

5.1	Parameters and setting for FEM simulations . . . . .	69
5.2	Error of FEM models with taming compared to <i>pdepe</i> solution . . . . .	74



# Chapter 1

## Introduction

In this Chapter we provide a brief overview of the Navier-Stokes equations, including their derivation. Then we discuss the issue of existence and uniqueness and various modifications of the equations that yield well-posedness for the system. In particular, we focus on the concept of taming which is the method we employ in this thesis.

Throughout the thesis we use standard notation. In particular, for any positive integer  $p$ , let  $\mathbb{R}^p$  denote  $p$ -dimensional Euclidean space with inner product

$$\langle x, y \rangle = y^T x = \sum_{i=1}^p x_i y_i.$$

The (Hilbert) space of Lebesgue square integrable functions on  $[a, b]$  is denoted by  $L^2([a, b]; \mathbb{R}^p)$  be the (Hilbert) space of Lebesgue square integrable functions on  $f : [a, b] \rightarrow \mathbb{R}^p$ . That is

$$L^2(a, b) = \left\{ f(\cdot) : f(\cdot) \text{ is measurable and } \int_a^b \|f(x)\|^2 dx < +\infty \right\}$$

endowed with the usual norm and inner product. For  $f(\cdot), g(\cdot) \in L^2([a, b]; \mathbb{R}^p)$  let

$$\langle f(\cdot), g(\cdot) \rangle = \int_a^b \langle f(x), g(x) \rangle dx \quad \text{and} \quad \|f\|_2 = \left[ \int_a^b |f(x)|^2 dx \right]^{1/2} = \sqrt{\langle f(\cdot), f(\cdot) \rangle}.$$

If  $(X, \langle \cdot, \cdot \rangle_X)$  is a Hilbert space, let  $L^2([a, b]; X)$  be the space of Lebesgue square integrable functions  $w : [a, b] \rightarrow X$  and for  $z(\cdot), w(\cdot) \in L^2([a, b]; X)$

$$\langle z(\cdot), w(\cdot) \rangle = \int_a^b \langle z(t), w(t) \rangle_X dt.$$

If  $z : [a, b] \rightarrow X$ , the the derivative of  $z(\cdot)$  at  $t$  will be denoted by

$$\frac{dz}{dt} f(t) = \partial_t z(t) = \frac{d}{dt} z(t).$$

Also the Kronecker delta is defined by

$$\delta_{ij} = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases}.$$

## 1.1 Introduction and Outline

For over 150 years the Navier-Stokes (NS) equations derived by Claude Louis Marie Henri Navier and George Gabriel Stokes have been the source of many mathematical and computational challenges. These equations describe the motion of a fluid in  $\mathbb{R}^n$  ( $n = 2$  or  $3$ ). For an incompressible fluid, the NS equations are given by

$$\begin{cases} \dot{\mathbf{u}}(t, x) = \frac{1}{Re} \Delta \mathbf{u}(t, x) - (\mathbf{u}(t, x) \cdot \nabla) \mathbf{u}(t, x) + \nabla p(t, x) + f(t, x) \\ \operatorname{div} \mathbf{u}(t, x) = \nabla \cdot \mathbf{u}(t, x) = \sum_{i=1}^n \partial_{x_i} u_i(t, x) = 0, \end{cases} \quad (1.1)$$

with initial conditions

$$\mathbf{u}(0, x) = \mathbf{u}_0(x) \quad (x \in \mathbb{R}^n). \quad (1.2)$$

Vector fields that satisfy  $\operatorname{div} \mathbf{u}(t, x) = 0$  are called divergent free and hence  $\mathbf{u}_0(x)$  is a given divergent-free vector field on  $\mathbb{R}^n$ . Also,  $Re$  is the Reynolds number and

$$\Delta = \sum_{i=1}^n \frac{\partial^2}{\partial x_i^2}$$

is the Laplacian.

The field of computational fluid dynamics (CFD) has focused on numerical solutions of these equations. Global existence and uniqueness are still unproven in three dimensions and the Clay Mathematical Institute has nominated the answer to the well posed-ness question with the Millennium Prize Problems. Over the years, the problem of well-posedness has been addressed through modifications of the NS equations in order to prove global existence and uniqueness in three dimensions. We briefly discuss the approaches by Ladyzenskaya [Lad98] and Röckner and Zhang [RZ09]. The second method is based on the concept of “taming” and, in this thesis, we focus on this approach.

Since both theory and numerical simulation of the NS equations are very complex, simpler models with similar properties are often used to test theoretical and numerical methods. In one dimension, Burgers’ equation provides such a model and has been subject to a large number of theoretical and numerical studies in both the mathematical and physical sciences communities.

The viscous Burgers’ equation is given by

$$\frac{\partial \mathbf{u}(t, x)}{\partial t} + \mathbf{u}(t, x) \frac{\partial \mathbf{u}(t, x)}{\partial x} = \mu \frac{\partial^2 \mathbf{u}(t, x)}{\partial x^2}.$$

As noted by Fletcher [Fle82] and others, Burgers’ equation captures some interesting non-linear phenomena that occur in fluid flows, such as shock behavior. Other simple ordinary differential equation (ODE) models have been used to capture non-linear dynamics similar to that observed in the Navier-Stokes equations. A simple ODE model was suggested by Baggett, Driscoll and Trefethen in [JSBT95]. This class of models is the focus of Chapter 3.

Computational science has experienced a significant growth over the past 50 years, and numerical approximations play a central role in the development of algorithms that can be applied to real problems. As noted in [ABG<sup>+</sup>02], finite precision arithmetic can produce “numerical solutions” that are not solutions to the the system of equations. In particular, Burgers’ equation with Neumann boundary conditions are “infinitely sensitive” to boundary conditions and standard numerical methods can easily produce numerical solutions that do not exist (see [ABG<sup>+</sup>02] for details). We show that by using the taming ideas in [RZ09] one can address some of these numerical issues.

## 1.2 A Brief Summary of the NS Equations

A brief overview of the physical background of fluid dynamics and the corresponding mathematical concepts are presented in this section. Also, we provide some historic notes on the subject.

### 1.2.1 Physical Motivation

In 1687 Isaac Newton published his works “Philosophiae Naturalis Principia Mathematica” [NBME33], stating among other things that;

*“...for straight, parallel and uniform flow, the shear stress between layers is proportional to the velocity gradient in the direction perpendicular to the layers.”*

Newton went on to present the first mathematical description of what became the field of fluid dynamics. Further progress was made in 1738 by Daniel Bernoulli who showed that the gradient of the pressure is proportional to the acceleration and Leonard Euler, who derived his famous equations. In conservation form the Euler equations have the form

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}(t, x)) &= 0 \\ \frac{\partial (\rho \mathbf{u}(t, x))}{\partial t} + \nabla \cdot (\mathbf{u}(t, x) \times (\rho \mathbf{u}(t, x))) + \nabla p &= 0 \\ \frac{\partial E}{\partial t} + \nabla \cdot (\mathbf{u}(t, x)(E + p)) &= 0. \end{aligned}$$

Navier (1822), Cauchy (1828), Poisson (1829) and Saint-Venant (1843) tried to add friction to describe the fluid behavior more accurately. In 1845, Stokes made the following assumptions:

- The fluid is continuous,
- The fluid is isotropic, namely the properties are independent of the direction and the frames concerned,
- The stress tensor is a linear function of the strain rates.

With these assumptions Stokes derived what is known as the incompressible NS equations and in dimensionless form are written as

$$\begin{cases} \dot{\mathbf{u}}(t, x) = \frac{1}{Re} \Delta \mathbf{u}(t, x) - (\mathbf{u}(t, x) \cdot \nabla) \mathbf{u}(t, x) + \nabla p(t, x) + f(x, t) \\ \operatorname{div}(\mathbf{u}(t, x)) = 0 \\ \mathbf{u}(0, x) = \mathbf{u}_0(x), \quad \operatorname{div}(\mathbf{u}_0) = 0, \end{cases} \quad (\text{I-NS})$$

where  $Re$  is the Reynolds-number representing the viscosity of the fluid or  $\nu = \frac{1}{Re}$  the viscosity. The equations are defined on a domain  $\Omega \subset \mathbb{R}^3$  or  $\Omega \subset \mathbb{R}^2$ , with boundary conditions

$$\mathbf{u}(t, y) = 0, \quad y \in \partial\Omega, \quad t > 0.$$

If the domain is unbounded, the boundary condition is replaced by a asymptotic condition on  $\mathbf{u}(t, x)$  as  $\|x\| \rightarrow \infty$ . For instance, it is assumed that in the far field,  $\lim_{\|x\| \rightarrow \infty} \mathbf{u}(t, x) = 0$ .

These equations can be derived by applying Newton's second law on the gravity, pressure and viscosity forces in fluids, which is presented in the next section. The individual terms are physically interpreted as

$$\begin{aligned} \dot{\mathbf{u}}(t, x) &: \text{Unsteady acceleration} \\ \mathbf{u}(t, x) \cdot \nabla \mathbf{u}(t, x) &: \text{Convective acceleration} \\ -\nabla p(t, x) &: \text{Pressure gradient} \\ \nu \Delta \mathbf{u}(t, x) &: \text{Diffusion} \\ f(t, x) &: \text{Body force} \end{aligned}$$

Although the NS equations are widely accepted as a reasonable model for many problems in fluid dynamics, there are still unanswered questions concerning the mathematical properties of these equations and how well these equations represent real fluid flows. As noted by Sir Cyril Hinshelwood (see Lighthill 1956, p.343), there is a gap between mathematical models and physical phenomena when he remarked;

*“ Fluid dynamicists were divided into hydraulic engineers who observe what cannot be explained and mathematicians who explain things that cannot be observed.”*

There is no complete theory for the 3D NS equations. However, some progress has been made on the issue of well-posedness and stability [CG09] and these results allow us to view the NS equations as dynamical systems. Also, for the 2D case, there exist a theory establishing the existence and smoothness of a classical global solution (see e.g. [LZZ13] and [JXZ05]). In the paper [Mas84] uniqueness and existence of weak solutions has been proven for all times regardless of the size of the initial data (I-NS).

Another approach to the problem of existence and uniqueness was suggested by Röckner and Zhang in [RZ09]. They add a taming term to the NS equations that “pushes” the solution back towards the origin outside of a certain region. This preserves the local behavior of the system in a neighborhood of zero but also ensures global solutions and uniqueness. We will examine the influence of similar taming terms on an ordinary differential equation found in [JSBT95].

## 1.2.2 A Derivation of Navier-Stokes Equations

As mentioned before, the Navier-Stokes equations can be derived by applying the physical conservation principles of mass and momentum (following [Bat67]). Let  $V \subset \mathbb{R}^3$  be a region of the three dimensional space with a boundary  $S = \partial V$  (sufficiently regular, e.g.  $C^2$  boundary). For introductory purposes, we assume that all functions considered are sufficiently smooth. Thus, one can interchange limits as needed and also all integrals are well defined. When the dependence on  $x$  and  $t$  is obvious, we suppress this dependence. For example,  $\rho(x, t)$  will be denoted by  $\rho$  etc., when this is clear. We begin with the basic conservation laws.

### Conservation of Mass

Mass conservation requires that the rate of accumulation of mass in a region  $V$  is balanced by the mass flux across its boundary  $S$ . Thus,

$$\frac{\partial}{\partial t} \int_V \rho dV + \int_S \rho \mathbf{n} \cdot \mathbf{u} dS = 0. \quad (1.3)$$

Here  $\rho$  is the density and  $\mathbf{u}$  the velocity vector. The surface integral can be converted into a volume integral by use of Gauss’ divergence theorem. Thus, (1.3) becomes

$$\int_V \left[ \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) \right] dV = 0,$$

which is often referred to as the *integral form of mass conservation*. Since the domain  $V$  is arbitrary, the integrand must equal zero and hence

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0. \quad (1.4)$$

Applying simple vector analysis, it follows that

$$\frac{\partial \rho}{\partial t} + \mathbf{u} \cdot \nabla \rho + \rho \nabla \cdot \mathbf{u} = 0$$

or, equivalently,

$$\frac{D\rho}{Dt} + \rho \nabla \cdot \mathbf{u} = 0, \quad (1.5)$$

where  $\frac{D\rho}{Dt} = \frac{\partial \rho}{\partial t} + \mathbf{u} \cdot \nabla \rho$  is called the *material derivative* (or *substantial derivative*) of  $\rho$ . The material derivative is frequently used in fluid dynamics and it specifies the rate of change of a physical quantity when moving with the fluid flow.

Equations (1.4) and (1.5) are called the *continuity equation*. For an incompressible fluid, the density  $\rho$  is constant with respect to both space and time, so  $\frac{D\rho}{Dt} = 0$  and hence

$$\nabla \cdot \mathbf{u} = 0. \quad (1.6)$$

This equation is called the incompressibility condition or the *divergence free condition*.

## Conservation of Momentum

The conservation of momentum states that the rate of accumulation of a momentum in the domain  $V$  plus the flux of the momentum on the boundary  $S$  has to equal the ratio of gain of momentum due to body forces and surface stress. This yields

$$\frac{\partial}{\partial t} \int_V \rho dV + \int_S \rho \mathbf{n} \cdot \mathbf{u} \otimes \mathbf{u} dS = \int_V \rho \mathbf{f} dV + \int_S \mathbf{n} \cdot \sigma dS, \quad (1.7)$$

where  $\mathbf{f}$  is the body force,  $\sigma$  the stress tensor and  $\mathbf{u} \otimes \mathbf{u}$  is the tensor product (or dyadic product). Applying Gauss' theorem, (1.7) becomes

$$\int_V \left[ \frac{\partial}{\partial t} (\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) \right] dV = \int_V (\rho \mathbf{f} + \nabla \cdot \sigma) dV.$$

Since  $V$  is arbitrary, the integrand has to be zero which implies

$$\frac{\partial}{\partial t} (\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = \rho \mathbf{f} + \nabla \cdot \sigma.$$

Again, elementary vector analysis implies that

$$\mathbf{u} \frac{\partial \rho}{\partial t} + \rho \frac{\partial \mathbf{u}}{\partial t} + \rho \mathbf{u} \cdot \nabla \mathbf{u} + \mathbf{u} \nabla \cdot (\rho \mathbf{u}) = \rho \mathbf{f} + \nabla \cdot \sigma.$$

The continuity equation (1.4) implies that the first and last term on the left hand side cancel, so that  $\mathbf{u} \frac{\partial \rho}{\partial t} = -\mathbf{u} \nabla \cdot (\rho \mathbf{u})$ . Therefore, we obtain

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho \mathbf{u} \cdot \nabla \mathbf{u} = \rho \mathbf{f} + \nabla \cdot \sigma. \quad (1.8)$$

In terms of the material derivatives, we have

$$\rho \frac{D\mathbf{u}}{Dt} = \rho \mathbf{f} + \nabla \cdot \sigma. \quad (1.9)$$

Equation (1.8) (or equation (1.9)) is called the *equation of motion*. For Newtonian fluids the constitutive relationship for the stress tensor  $\sigma$  is given by Newton's law (or Navier-Stokes law), which is written as

$$\begin{aligned} \sigma &= -p\mathbf{I} + \tau, \\ \tau &= \lambda \nabla \cdot \mathbf{u} \mathbf{I} + 2\mu \text{def } \mathbf{u}. \end{aligned} \quad (1.10)$$

Here  $\text{def } \mathbf{u} = \frac{1}{2} [\nabla \mathbf{u} + (\nabla \mathbf{u})^T]$  is the *rate-of-strain tensor*,  $p$  is the pressure,  $\mu$  is the coefficient of dynamic viscosity and  $\lambda$  is the second coefficient of viscosity.

Substituting (1.10) into (1.9) yields

$$\rho \frac{D\mathbf{u}}{Dt} = \rho \mathbf{f} - \nabla p + \mu \nabla^2 \mathbf{u} + (\lambda + \mu) \nabla (\nabla \cdot \mathbf{u}) + (\nabla \cdot \mathbf{u}) \nabla \lambda + 2 \text{def } \mathbf{u} \cdot \nabla \mu.$$

In the incompressible and homogeneous case, which is assumed throughout this thesis, both  $\mu$  and  $\lambda$  are constants and the equation above can be simplified to

$$\frac{D\mathbf{u}}{Dt} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f}.$$

Equivalently, we can write this equation as

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f}, \quad (1.11)$$

where  $\nu = \mu/\rho$  is called the *kinematic viscosity*. Finally, one arrives the Navier-Stokes equations (I-NS) for viscous incompressible flow.

### 1.2.3 The Dimensionless Navier-Stokes Equation

The derivation of the Navier Stokes equations involves absolute physical quantities. In fluid dynamics, it is often more appealing to use dimensionless forms of the governing equations. This is usually done by taking reference values for special quantities and dividing them out to arrive at a relative scale. Here  $L$  is a reference length,  $U$  a reference velocity. We then write

$$\bar{\mathbf{x}} = \frac{\mathbf{x}}{L}, \quad \bar{\mathbf{u}} = \frac{\mathbf{u}}{U}, \quad \bar{t} = \frac{U}{L} t, \quad p = \frac{p}{\rho U^2}, \quad \bar{\mathbf{f}} = \frac{L}{U^2} \mathbf{f}. \quad (1.12)$$

Substituting the expressions (1.12) into (1.6) and (1.11) yields to

$$\begin{aligned} \nabla \cdot \mathbf{u} &= 0 \\ \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} &= -\nabla p + \frac{1}{Re} \nabla^2 \mathbf{u} + \mathbf{f}, \end{aligned} \quad (1.13)$$

where the dimensionless variables are denoted by the same symbols as the corresponding dimensional variables.

### 1.2.4 A short note on the Reynolds number

The non-dimensional constant  $Re = \frac{LU}{\nu}$  is known as the *Reynolds number* and is one of the basic dimensionless numbers in fluid dynamics. The physical significance of the Reynolds number is that it can be interpreted as the ratio of inertial forces to viscous forces. Its magnitude characterizes the flow field. Most of the effort in this thesis is concerned with low Reynolds number flow; roughly of the order  $Re \in (0, 100)$ . At these low Reynolds numbers one encounters laminar flows. Airflow simulations often involve Reynolds numbers in the magnitude of  $10^5$  or  $10^6$ .

A historical note on the Reynolds number is given in [Rot90]. In 1908, Arnold Sommerfeld gave a paper in the proceedings of the the 4th International Congress of Mathematicians in Rome where he discussed turbulence [Som08]. In his presentation he provides a set of equations (known today as the Orr-Sommerfeld equations) and used he used a number “ $R$ ” that he described as

*“a pure number, we will call it the Reynolds Number.”*

The terminology introduced by Sommerfeld has not changed except for the modification  $R = Re$  used today in all branches of fluid dynamics.

### The Critical Reynolds Number

For dynamical systems, one notices a drastic change in the behavior at certain thresholds of the Reynolds number: the appearance of turbulence, non smooth orbits and such. In his original 1883 paper, Reynolds came to the conclusion that there has to be a critical velocity, at which previously existing eddies die out and the motion becomes steady. In a later 1895 paper, he tried to explicitly calculate the “lower critical” Reynolds number. His personal experience was around  $Re = 2000$  and it turns out to be  $Re_{crit} = 2020$  as of today’s experiments. Another contribution of his paper in 1895 is the first mentioning of the Reynolds’ averaged Navier-Stokes equations, the technique discussed earlier, that leads to the current version of the equations.

## 1.3 Motivation

A goal of this thesis is to present a framework that allows us to modify a basic system in such a way that one obtains “global” existence and uniqueness of the equations without changing



their behavior near a local attractor. We first recall some properties of the Navier-Stokes equations that are typical of the models we consider below. In particular, we focus on local attractors (laminar flows and chaotic or turbulent structures) that are to be preserved when the modified equations are introduced.

Shear flows arise in many applications such as separated flow of liquids, combustion, jet noise, chemical lasers, atmospheric problems, etc. The classical example of a shear flow is a flow in a pipe with turbulent behavior away from the walls. Another commonly used example is two parallel streams of different velocities that are separated by a thin plate and come together as the plate ends. The layer grows from the resulting velocity discontinuity and thickens downstream. Shear layers appear within many other flows, such as in the initial part of jets and on top of locally-separated boundary layers. They are technologically important because they generate noise and dissipate energy and are therefore a source of drag. If the two streams contain different fluids, the mixing occurs in the shear layer.

On the other hand high viscous or slow moving fluids tend to be smooth and regular. A flow that is time independent is called a *laminar flow* and is an equilibrium state. Its typical characteristic is that the fluid flows in parallel layers with no disruption between the layers (*lateral mixing*). Thus, in a laminar flow there is no cross currents, no eddies or swirls. Laminar flows provide a base flow around which one can linearize the Navier-Stokes equations. A commonly studied laminar flow is flow in a closed channel, i.e. a pipe or between flat plates. Although the time independent (laminar) flow represents a steady state solution, this solution can become “unstable” and produce a turbulent flow. For a simple heuristic interpretation, one could say that laminar flow is somewhat “smooth” and turbulent flow tends to be “rough” (see e.g. [Bat67]). Roughly speaking, turbulent flow can be thought of as an (chaotic) attractor for the dynamical system generated by the Navier-Stokes equations.

From physical experiments we know that turbulence is a natural state for many fluids and can greatly impact aerodynamic drag on planes, cars and other objects. For a more thorough introduction to turbulence see [Dav04].

### 1.3.1 Navier-Stokes Equations and Properties

The nonlinearity in the Navier-Stokes equation produces interesting physical phenomena such as turbulence. Nonlinearity is also the root cause of significant difficulties in theory, computation and applications. A standard approach in the analysis of such systems is to linearize about a (shear) flow and to use the linearized model to study stability of the system. In addition, this is a typical first step in designing stabilizing feedback controllers (e.g. [CG13]). In particular, (I-NS) is linearized around a chosen steady state nominal flow condition  $(\bar{\mathbf{u}}, \bar{\mathbf{p}})$  (see [JB01a], [JB01b]). One can decompose  $\mathbf{u}$  into a term  $\bar{\mathbf{u}}$  plus a “small” disturbance  $\tilde{\mathbf{u}}$  so that

$$\mathbf{u} = \bar{\mathbf{u}} + \tilde{\mathbf{u}} \quad \text{and} \quad \mathbf{p} = \bar{\mathbf{p}} + \tilde{\mathbf{p}}.$$

By inserting the expressions above into the NS equations (I-NS) and neglecting higher order terms, one arrives at the linear equation for  $\tilde{\mathbf{u}}$  given by

$$\frac{\partial \tilde{\mathbf{u}}}{\partial t} = -\nabla_{\tilde{\mathbf{u}}}\bar{\mathbf{u}} - \nabla_{\tilde{\mathbf{u}}}\tilde{\mathbf{u}} + \nu\nabla\tilde{\mathbf{p}} + \nu\Delta\tilde{\mathbf{u}} + \left(\nu\Delta\bar{\mathbf{u}} - \nabla_{\tilde{\mathbf{u}}}\bar{\mathbf{u}} - \nu\nabla\tilde{\mathbf{p}}\right) - \nabla_{\tilde{\mathbf{u}}}\tilde{\mathbf{u}}. \quad (1.14)$$

Although numerical simulations have been used in an attempt to answer quantitative questions about the dynamic (long time) behavior of the Navier-Stokes equations, this approach must be used with great caution. Even the simple Burgers equation shows how this approach can lead to incorrect conclusions (see [ABG<sup>+</sup>02] and [EJG13]). Note that despite all the mathematical questions that are still yet open, the Navier-Stokes equations are successfully used in turbulence models and aerodynamics to obtain very practical results through simulations.

## A Ladyzhenskaya Model

Here we discuss the ‘‘Ladyzhenskaya Model’’ which is a modified Navier-Stokes equation first suggested by Ladyzhenskaya in [Lad75]. This particular modification is worth mentioning here, because it provides a way to prove existence and uniqueness for a slightly modified version of the Navier-Stokes equations (see [Gun89, p.193ff]). Ladyzhenskaya pointed out that there might be a physical complication preventing global uniqueness and existence of solutions to the Navier-Stokes, since one might expect a physical flow to be uniquely determined by the equation and corresponding boundary conditions. To resolve this discrepancy, Ladyzhenskaya proposed an alternate model which is ‘‘close’’ to the Navier-Stokes system, but has globally unique solutions (see [Lad98], [Lad87], [Lad75]). These equations are

$$\begin{aligned} f &= -\operatorname{div}(\mathcal{A}(\mathbf{u})[\operatorname{grad} \mathbf{u} + (\operatorname{grad} \mathbf{u})^T]) + \mathbf{u} \cdot \operatorname{grad} \mathbf{u} + \operatorname{grad} p, \quad x \in \Omega \\ \operatorname{div}(\mathbf{u}) &= 0 \quad \text{in } \Omega \\ \mathbf{u} &= 0 \quad \text{on } \Gamma, \end{aligned} \quad (\text{L-NS})$$

where  $\mathbf{u}, p$  and  $f$  denote the velocity, pressure and body force respectively, and  $\Omega$  denotes an open, bounded subset of  $\mathbb{R}^p$ ,  $p \in \{2, 3\}$  with boundary  $\Gamma = \partial\Omega$ .

Ladyzhenskaya assumed  $\mathcal{A}$  to be of the form

$$\mathcal{A}(\mathbf{u}) = \nu + \nu_1 \left[ \sum_{i,j=1}^n \left( \frac{\partial \mathbf{u}_i}{\partial x_j} \right)^2 \right]^{r/2}$$

with  $r > 0$  a constant and  $\nu$  can be interpreted as the usual kinematic viscosity. Some information about the form of  $\nu_1$  and  $r$  may be gleaned from physical considerations, such as  $\nu_1 \lll \nu, \nu_1 > 0$ . Notice here that for  $\nu_1 = 0$ , we get precisely the Navier-Stokes equations. The same happens for small shear rates, i.e., when the derivatives of the velocity are small, then  $\mathcal{A}(\mathbf{u}) \approx \nu$ . However, for large shear rates the additional term in  $\nu_1$  in  $\mathcal{A}$  results in a

more highly damped system. Therefore, the physical interpretation is similar to damping. To paraphrase the approach heuristically, it is assumed that the Reynolds number is not constant (which seems more appropriate due to physical experiments), but consists of a constant term and the sum

$$\nu_1 \left[ \sum_{i,j=1}^n \left( \frac{\partial \mathbf{u}_i}{\partial x_j} \right)^2 \right]^{r/2}.$$

Hence,  $\mathcal{A}(\mathbf{u}) \approx \nu$  depends on the partial derivatives of the flow at each given point. Although this is an interesting model, we pursue a different approach which we describe below.

### 1.3.2 Taming

Ladyzhenskaya's approach is one way to obtain global existence and uniqueness for the Navier-Stokes equations. In recent years, another approach, called *taming*, has been developed in [RZ09] and [Zha09]. The main idea in these papers is to add a term to a given equation (vaguely similar to damping) that “pushes” solutions towards a given local attractor. The corresponding tamed equations remain the same near this attractor, but produce a globally defined dissipative dynamical system. In their paper [Zha09] start with the three dimensional Navier-Stokes equations

$$\begin{aligned} \nabla \cdot \mathbf{u} &= 0 \\ \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} &= -\nabla p + \frac{1}{Re} \nabla^2 \mathbf{u} + \mathbf{f}. \end{aligned} \tag{1.15}$$

A *taming function*  $g_N : \mathbb{R} \rightarrow \mathbb{R}$  is added to these equations which yields

$$\partial_t \mathbf{u}(t) = \nu \Delta \mathbf{u}(t) - (\mathbf{u}(t) \cdot \nabla) \mathbf{u}(t) + \nabla p(t) - g_N(|\mathbf{u}(t)|^2) \mathbf{u}(t) + \mathbf{f}(t), \tag{1.16}$$

where  $g_N : \mathbb{R}^+ \rightarrow \mathbb{R}^+$  is a smooth function with specific properties described below. Initial and boundary conditions are not altered.

In particular, for a given radius  $N$ ,  $g_N(\cdot)$  is defined by

$$g_N(r) = \begin{cases} 0 & r \in [0, N] \\ \frac{r-N-1/2}{\nu} & r \geq N+1 \\ 0 \leq g'_N(r) \leq C_\nu & r \geq 0 \\ |g_N^{(k)}| \leq C_{\nu,k} & r \geq 0, k \in \mathbb{N} \end{cases}$$

The function  $g_N(\cdot)$  is assumed to be smooth with all derivatives bounded by a constant only depending on  $\nu$  and the order of the derivative, but uniformly for all  $r \geq 0$ . The slope of  $g_N(\cdot)$  is  $\frac{1}{\nu}$  outside of the ball with radius  $N+1$  corresponding to the amount of “damping” added to the system. Also, by the design of  $g_N$ , namely  $g_N(r) = 0$  for  $r \in [0, N]$ , nothing is added inside a ball of radius  $N$ . This corresponds to not changing the dynamics in a neighborhood

of the origin. One could envision the value of  $N$  as a valve that adds taming only when needed. Technical reasons require only that  $N$  to be finite. However, for simulation the size of  $N$  might play an important role. Note that here the region about the origin is viewed as the “region of interest”, but by shifting this process can be applied to any region in space.

We note that the proof of global well-posedness given by Roekner and Zhang required no completely explicit form for the taming function. For  $r \in [N, N + 1]$ , we only know it is smooth with bounded derivatives. This observation allows us to use the idea for other applications and numerical computation by changing the precise form of  $g_N(\cdot)$ . Moreover, Roekner and Zhang further noted that if there exists a solution to the Navier-Stokes equation, it also locally solves the tamed equation (1.16). Weak solutions to the classical Navier-Stokes equation can be constructed with this technique.

The most important observation is that one can prove that (at least on bounded  $C_2$  domains) there exists a global attractor for (1.16). We will establish a similar result for the Baggot-Driscoll-Trefethen model discussed below and for the Burgers equation with mixed boundary conditions.

## 1.4 Main Goal of This Thesis

In this thesis we investigate taming as a means to ensure technical mathematical requirements while preserving the main character of the local dynamics for a given model. This is accomplished by adding a taming term to the governing equation that vanishes near the “interesting” dynamics (a local attractor) and provides damping further away from this object to ensure global existence and uniqueness. Thus, one obtains a dissipative dynamical system that locally matches the dynamics near the local attractor.

In this thesis we consider two models; one defined by a system of ODEs and the Burgers equation. Both models have properties in common with the Navier-Stokes equations and are excellent test cases for our study. The BDT model is a non-linear ODE (presented in the next chapter) that, despite its simplicity inherits some of the main features of the Navier-Stokes equation. In particular, one encounters chaotic behavior and extreme sensitivity to initial data. In addition, the linear portion of the system is highly non-normal which is a tripping mechanism for transition. We begin with a two state model to present the idea of taming in a very simple situation and progress to the three state model. Various taming terms are used and evaluated for this model. We show that by properly choosing a “good” taming term the local dynamics are not changed. However, by selecting other taming terms one can introduce extraneous dynamics such as periodic orbits.

We turn then to a study of Burgers’ equation and investigate how taming can deal with numerically generated false solutions and erroneous long time behavior predicted by numerical approximations (e.g., finite elements, finite differences). As noted above, finite precision arithmetic can be the root cause of these incorrect solutions and we show how taming can

be used to stabilize numerical approximations to produce correct long time dynamics.

Finally, we apply taming to reduced order models (proper orthogonal decomposition) of the finite element approximations of Burgers' equation in tamed and untamed versions. Numerical studies indicate that taming can stabilize the reduced model or at least reduce the numerical error.

The remainder of the thesis is structured as follows. In Chapter 2 we introduce the main mathematical concepts in dynamical systems. The mathematical models are described in detail in Chapter 3. The main results follow in Chapter 4 and Chapter 5 together with numerical simulations. This thesis closes with concluding remarks and future directions. Finally, the Matlab code used to produce all the numerical results is attached as an appendix.

# Chapter 2

## Introduction to Dynamical Systems

The mathematical theory of dynamical systems is introduced in this chapter. Definitions and important results are stated without proofs.

We are interested in dynamical systems generated by differential equations (DEs). If the differential equation is an ordinary differential equation (ODE), then the system evolves in a finite dimensional space of the form  $\mathbb{R}^p$ . However, if the differential equation is a partial differential equation (PDE), then the dynamical system typically evolves in an infinite dimensional space  $X$ . In order to establish that a given differential equation generates a dynamical system one must show that the DE is well-posed. Thus, one needs to establish the (global) existence, uniqueness and continuous dependence on initial data of solutions to the DE. Once it is known that the DE generates a dynamical system, powerful mathematical and numerical tools can be brought to bear to analyze the system. The focus of this work is to use this structure to study the qualitative behavior of such systems and to use the dynamical system framework as a mechanism to address numerical issues.

Since practical applied problems often yield complex dynamical systems that cannot be explicitly solved, qualitative and numerical methods are required. In particular, numerical methods are of fundamental importance in both the quantitative and qualitative analysis of real systems. The long time behavior of dynamical systems are captured in certain *invariant sets* such as equilibrium points, periodic solutions and attractors.

Although dynamical systems generated by a DE are typically continuous in time, discrete dynamical systems are also important. Discrete dynamical systems can arise by themselves in a particular application and as discrete approximations of continuous dynamical systems. Thus we briefly discuss discrete dynamical systems since they can help provide useful intuition. Although the formulations are different (iterative maps versus differential equations) to describe a dynamical system in its discrete and continuous case, respectively, many of the fundamental results apply to both types. A discrete dynamical system corresponds to a finite dimensional mapping from  $\mathbb{R}^p$  to  $\mathbb{R}^p$ . In this context, a dynamical system is a rule, which

takes an initial point  $x_0 \in \mathbb{R}^p$  and generates a (discrete) sequence of points  $x_n = x(t_n) \in \mathbb{R}^p$  depending on discrete “time steps”  $t_n$ ,  $n = 1, 2, 3, \dots$  governed by the equation at hand. In this case, one wants to make predictions about the behavior of the set of points  $x_n$  based on the governing equation and the initial point  $x_0$ . Governing equations can be of various types such as linear (matrix) equations, discrete versions of ODEs and PDEs or a general rule.

## 2.1 Discrete Dynamical Systems

Consider the system defined by the sequence of vectors  $\{x_n\}_{n=0}^{\infty} \in \mathbb{R}^n$  satisfying

$$x_{n+1} = F(x_n), \quad x_0 \in \mathbb{R}^p, \quad (2.1)$$

where  $F : D \subset \mathbb{R}^p \rightarrow \mathbb{R}^p$  is a map or *iteration* and  $D$  is the *domain of definition* of  $F$ . The vector  $x_0 \in \mathbb{R}^p$  is known as the *initial data*, *initial vector* or *initial condition*. This is an *explicit* formulation of the map  $F$ , since the step  $x_{n+1}$  is based on  $x_n$  alone.

**Definition 2.1.** *The equation (2.1) generates a **discrete dynamical system** on a subset  $E \subset \mathbb{R}^p$  if for each initial condition  $x_0 \in E$ , there exists a unique solution  $\{x_n\}_{n=0}^{\infty} \subset E$ , defined for all  $n \geq 0$ .*

Note that uniqueness is not an issue here since the function  $F$  is assumed to be well defined. What needs to be shown is  $F(x_n) \in E$  for all  $n \geq 0$ .

**Definition 2.2.** *Given a discrete dynamical system on a set  $E \subset \mathbb{R}^p$ , define the **evolution semigroup** to be the set of operators  $S^n : E \rightarrow E$ ,  $n = 0, 1, 2, 3, \dots$  defined by  $S^n x_0 = x_n$ . This operator has the following properties;*

- $S^0 = Id$ , the identity on  $\mathbb{R}^p$ ,
- $S^{n+m}x_0 = x_{n+m} = S^n x_m = S^m x_n = S^m S^n x_0$  for all  $n, m \geq 0$ .

## 2.2 Dynamical Systems Generated by Differential Equations

Consider now dynamical systems from the perspective of (ordinary) differential equations in  $\mathbb{R}^p$ . Here, the starting point is not the map  $F$  as before. In this case we start with an initial value problem for a differential equation of the form

$$\dot{\mathbf{u}}(t) = f(\mathbf{u}(t)), \quad \mathbf{u}(0) = \mathbf{u}_0 \in \mathbb{R}^p, \quad (2.2)$$

where  $f : \mathbb{R}^p \rightarrow \mathbb{R}^p$  is a given continuous function. Of course, more regularity on  $f(\cdot)$  is often required and this will be discussed when necessary.

**Definition 2.3.** A *solution* of (2.2) is a continuously differentiable function  $\mathbf{u} : I \rightarrow \mathbb{R}^p$  defined on a connected  $I \subseteq \mathbb{R}$  satisfying (i)  $0 \in I$  and (ii)  $\mathbf{u}(\cdot)$  satisfies (2.2) for  $t \in I$ .

Note that a solution  $\mathbf{u}(\cdot)$  of (2.2) is required to be defined on a connected interval containing 0 and

$$\mathbf{u}(t) = (u_1(t), u_2(t), \dots, u_p(t))^T \in \mathbb{R}^p$$

is a continuously differentiable vector function.

**Definition 2.4.** The differential equation (2.2) defines a **dynamical system** on a subset  $E \subset \mathbb{R}^p$ , if for every  $\mathbf{u}_0 \in E$  there exists a unique solution  $\mathbf{u}(\cdot)$  of (2.2), defined for all  $t \in [0, \infty)$ ,  $\mathbf{u}(t) \in E$  for all  $t \in [0, \infty)$  and this solution depends continuously on the initial data  $\mathbf{u}_0 \in E$ .

**Definition 2.5.** If the differential equation (2.2) defines a **dynamical system** on a subset  $E$ , then the **solution semigroup** is the set of operators  $\{S(t) : E \rightarrow E \text{ for } t \geq 0\}$  defined by

$$S(t)\mathbf{u}_0 = \mathbf{u}(\cdot),$$

where  $\mathbf{u}(\cdot)$  is the unique solution of (2.2) with initial data  $\mathbf{u}(0) = \mathbf{u}_0$ .

**Definition 2.6.** A set of continuous operators  $\{T(t) : E \rightarrow E \text{ for } t \geq 0\}$  is called a  **$C_0$ -semigroup** on  $E$  if

1.  $T(t + \tau) = T(t)T(\tau)$  for all  $t, \tau \geq 0$ ,
2.  $T(0) = Id$ , the identity on  $E$ ,
3.  $\lim_{t \rightarrow 0} T(t)\mathbf{u} = \mathbf{u}$  for an  $\mathbf{u} \in E$  (strong continuity at  $t = 0$ ).

In the literature, the terms *dynamical system* and  *$C_0$ -semigroup* are often used interchangeably. In the setting here, if the differential equation (2.2) defines a dynamical system on  $E$  then the corresponding solution semigroup is a  *$C_0$ -semigroup*. Consequently, one needs to have conditions on the function  $f(\cdot)$  that are sufficient for the differential equation (2.2) to define a dynamical system. The following existence and uniqueness result applies to the system (2.2) and can be found on page 102 in [SH98].

**Theorem 2.7.** Assume  $f : \mathbb{R}^p \rightarrow \mathbb{R}^p$  is globally Lipschitz on  $\mathbb{R}^p$  so that there exist a constant  $L$  such that

$$\|f(\mathbf{u}) - f(\mathbf{v})\| \leq L \|\mathbf{u} - \mathbf{v}\|, \quad \mathbf{u}, \mathbf{v} \in \mathbb{R}^p.$$

Then the initial value problem for the differential equation (2.2) has a unique solution for any  $\mathbf{u}_0 \in \mathbb{R}^p$ , this solution exists for all times  $t \geq 0$  and this solution depends continuously on the initial data. Thus, (2.2) defines a dynamical system on  $\mathbb{R}^p$ .



The previous theorem is too weak to be useful for most applications. The assumption that  $f : \mathbb{R}^p \rightarrow \mathbb{R}^p$  be globally Lipschitz is a very strong. Although weaker assumptions such as requiring that  $f : \mathbb{R}^p \rightarrow \mathbb{R}^p$  be local Lipschitz will guarantee the existence of a unique local solution, showing that this solution can be extended to all  $t \in [0, \infty)$  requires placing additional properties on  $f$ . We will make use of the following result from [SH98, p. 178] (see page 179).

**Theorem 2.8.** *If  $f : \mathbb{R}^p \rightarrow \mathbb{R}^p$  is locally Lipschitz,  $f(0) = 0$  and*

$$\langle f(\mathbf{x}), \mathbf{x} \rangle < 0 \quad \forall \mathbf{x} \neq 0,$$

*then the differential equation*

$$\frac{d}{dt} \mathbf{x}(t) = f(\mathbf{x}(t))$$

*generates a dynamical system  $S(t)$  on  $\mathbb{R}^n$ . Moreover, for each  $\mathbf{x}_0 \in \mathbb{R}^n$*

$$\lim_{t \rightarrow +\infty} \|S(t)\mathbf{x}_0\| = 0$$

*and if  $S(t_1)\mathbf{x}_0 \neq 0$ , then  $\|S(t_2)\mathbf{x}_0\| < \|S(t_1)\mathbf{x}_0\|$  for  $t_2 > t_1$ .*

Properties of a  $C_0$ -semigroup are essential those of the exponential function. For example, the solution to the linear initial value problem  $\dot{x}(t) = Ax(t)$ ,  $x(0) = x_0$  is given by  $x(t) = e^{At}x_0$ , (for a constant matrix  $A \in \mathbb{R}^{p \times p}$ ). Note that the solution semigroup is given by  $S(t) = e^{At}$  and is clearly a  $C_0$ -semigroup on  $\mathbb{R}^p$ .

The following theorem ensures that given the local Lipschitz regularity of  $f : \mathbb{R}^p \rightarrow \mathbb{R}^p$ , the dynamical system  $S(t)$  generated (2.2) is also Lipschitz continuous with respect to its initial data.

**Theorem 2.9.** *Suppose the differential equation (2.2) generates a dynamical system  $S(t)$  on  $\mathbb{R}^p$  and that  $f : \mathbb{R}^p \rightarrow \mathbb{R}^p$  is locally Lipschitz. Let  $B = B(T, U, V) \subset \mathbb{R}^p$  be a bounded set with the property that  $S(t)U, S(t)V \in B$  for all  $t \in [0, T]$  and  $L$  be the Lipschitz constant for  $f$  in  $B$ . Then:*

$$\|S(t)U - S(t)V\| \leq e^{Lt}\|U - V\| \quad \forall t \in [0, T].$$

*Thus, the dynamical system  $S(t)$  is Lipschitz continuous with respect to the initial data.*

## 2.2.1 Equilibrium Solutions and Stability

In the behavior of dynamical systems, solutions that stay at a fixed point are of particular interest. The amount and positions of fixed points and their stability characterize the dynamics of the system.

**Definition 2.10.** A point  $\bar{\mathbf{u}} \in \mathbb{R}^p$  such that  $f(\bar{\mathbf{u}}) = 0$  is called an **equilibrium (or fixed) point** of (2.2).

To compute the fixed points of a given differential equation, one needs to solve the equation  $f(\bar{\mathbf{u}}) = 0$  of (2.2). In general, solving this equation requires using a numerical algorithm such as Newton's method and can be difficult. An *equilibrium solution* is a constant solution of (2.2) so that  $\mathbf{x}_e(t)$  is an equilibrium solution if and only if

$$\mathbf{x}_e(t) \equiv \bar{\mathbf{u}},$$

where  $\bar{\mathbf{u}}$  is a fixed point.

*Stability* of a solution is concerned with the long term behavior of solutions. We recall the standard definitions of stability from the book [RKM81].

**Definition 2.11.** An equilibrium solution  $\mathbf{x}_e(t) \equiv \bar{\mathbf{u}}$  of (2.2) is called **stable** if for every  $\epsilon > 0$ , there exists a  $\delta = \delta(\epsilon) > 0$  such that if  $\mathbf{x}(t, \xi)$  is a solution of (2.2) with  $\mathbf{x}(0, \xi) = \xi$  and  $\|\xi - \bar{\mathbf{u}}\| < \delta$ , then

$$\|\mathbf{x}(t, \xi) - \bar{\mathbf{u}}\| < \epsilon \quad \text{for all } t \geq 0.$$

**Definition 2.12.** An equilibrium solution  $\mathbf{x}_e(t) \equiv \bar{\mathbf{u}}$  of (2.2) is called **unstable** if it is not stable.

**Definition 2.13.** The equilibrium solution  $\mathbf{x}_e(t) \equiv \bar{\mathbf{u}}$  is called **asymptotically stable**, if  $\mathbf{x}_e(t) \equiv \bar{\mathbf{u}}$  is stable and there exist an  $\eta > 0$  such that if  $\mathbf{x}(t, \xi)$  is a solution of (2.2) with  $\mathbf{x}(0, \xi) = \xi$  and  $\|\xi - \bar{\mathbf{u}}\| < \eta$ , then

$$\lim_{t \rightarrow \infty} \mathbf{x}(t, \xi) = \bar{\mathbf{u}}.$$

**Definition 2.14.** The equilibrium solution  $\mathbf{x}_e(t) \equiv \bar{\mathbf{u}}$  is called **exponentially asymptotically stable** if it is asymptotically stable and there exist a  $\alpha > 0$ , and for every  $\epsilon > 0$  there ex. a  $\delta = \delta(\epsilon) > 0$  such that if  $\mathbf{x}(t, \xi)$  is a solution of (2.2) with  $\mathbf{x}(0, \xi) = \xi$  and  $\|\xi - \bar{\mathbf{u}}\| < \delta$ ,

$$\|\mathbf{x}(t, \xi) - \bar{\mathbf{u}}\| \leq \epsilon e^{-\alpha t} \quad \text{for all } t \geq 0$$

Notice that all of these definitions apply specifically to the autonomous differential equation (2.2), but can be extended to nonautonomous equations (see [RKM81]). We present some standard methods to establish stability of a given equilibrium solution  $\mathbf{x}_e(t) \equiv \bar{\mathbf{u}}$ . We focus on the classical results and give a brief review of Lyapunov theory and LaSalle's Invariance Principle.

## Lyapunov theory

The first result concerns stability of linear autonomous equations and can be found on page 181 in [RKM81].

**Theorem 2.15.** Consider the linear differential equation  $\dot{\mathbf{u}}(t) = A\mathbf{x}(t)$  for  $A \in \mathbb{R}^{p \times p}$ , then  $\mathbf{x}_e(t) = 0$  is a stable equilibrium solution if all eigenvalues of  $A$  have nonpositive real parts and every eigenvalue of  $A$  that has zero real part is a simple zero of the characteristic polynomial of  $A$ . Moreover,  $\mathbf{x}_e(t) = 0$  is exponentially asymptotically stable if and only if, all the eigenvalues of  $A$  have negative real parts.

Note that this is limited to linear differential systems, with the right hand side independent of time. There are extensions to the periodic, non autonomous case where  $\dot{\mathbf{u}}(t) = A(t)\mathbf{u}(t)$  and  $A(t) = A(t + T)$  for some  $T \geq 0$ . Often this is referred to as the *Floquet Theory*. Lyapunov theory is based on energy estimates and so called Lyapunov functions and apply directly to nonlinear systems. Given the differential equation (2.2) defined by  $f : \mathbb{R}^p \rightarrow \mathbb{R}^p$  and a smooth function  $V : \mathbb{R}^p \rightarrow \mathbb{R}^1$  we define the derivative of  $V$  along  $f$ .

**Definition 2.16.** Let  $V : \mathbb{R}^p \rightarrow \mathbb{R}^1$  be a continuously differentiable with respect to all of its arguments and  $\nabla V$  denote the gradient of  $V$  with respect to  $x$ . The **derivative of  $V$  along  $f$**  is defined by

$$\dot{V}(x) = \sum_{i=1}^n \frac{\partial V}{\partial x_i}(x) f_i(x) = \langle \nabla V(x), f(x) \rangle. \quad (2.3)$$

**Definition 2.17.** A function  $V : \mathbb{R}^p \rightarrow \mathbb{R}^1$  is called a **Lyapunov function** for the dynamical system generated by (2.2), if  $V$  is positive definite and the derivative of  $V$  along  $f$  is negative semi-definite. In particular, the function  $V : \mathbb{R}^p \rightarrow \mathbb{R}^1$  is a Lyapunov function for (2.2), if

$$V(x) > 0, \text{ when } x \in \mathbb{R}^p \text{ and } x \neq 0, \quad V(0) = 0 \quad (2.4)$$

and

$$\dot{V}(x) \leq 0 \quad \text{for all } x \in \mathbb{R}^p. \quad (2.5)$$

The following two theorems follow from the more general results in [RKM81] (also see pages 128 in [SH98]).

**Theorem 2.18.** If  $f(0) = 0$  and there exists a Lyapunov function for (2.2), then the equilibrium solution  $\mathbf{x}(t) = 0$  is stable.

**Theorem 2.19.** If  $f(0) = 0$  and there exists a Lyapunov function for (2.2) satisfying

$$\dot{V}(x) < 0 \quad \text{for all } x \in \mathbb{R}^p, \quad x \neq 0, \quad (2.6)$$

then the equilibrium solution  $\mathbf{x}(t) = 0$  is asymptotically stable.

Constructing Lyapunov functions in general is often considered a form of art, since no general algorithm has yet been developed. In many practical cases, Lyapunov functions are constructed from energy functions. We return to stability after reviewing some concepts from dynamical systems theory.

## 2.3 Limit sets and Attractors

We turn now to the study of dynamical systems (i.e.,  $C_0$ -semigroups) defined on  $\mathbb{R}^p$ . In what follows we assume that  $\{S(t) : \mathbb{R}^p \rightarrow \mathbb{R}^p, t \geq 0\}$  is a  $C_0$ -semigroup defined on  $\mathbb{R}^p$ .

**Definition 2.20.** For any bounded set  $B \subset \mathbb{R}^p$ , the **action** of the semigroup  $S(t)$  on  $B$  is given by

$$S(t)B = \bigcup_{\mathbf{u} \in B} S(t)\mathbf{u}.$$

**Definition 2.21.** The **positive orbit** or **forward orbit**  $\Gamma^+(\mathbf{u})$ , through the point  $\mathbf{u} \in \mathbb{R}^p$  is the set

$$\Gamma^+(\mathbf{u}) = \{S(t)\mathbf{u} : t \geq 0\}.$$

**Definition 2.22.** A point  $x \in \mathbb{R}^p$  is called an  **$\omega$ -limit point** for a dynamical system  $S(t)$  generated by (2.2), if there exists a sequence  $\{t_i\}_{i=1}^{\infty}$ ,  $t_i \rightarrow \infty$  with  $S(t_i)\xi \rightarrow x$  as  $t_i \rightarrow \infty$ . The set of all such points for given initial data  $\xi \in U$  is called the  **$\omega$ -limit set** of  $U$ , denoted  $\omega(U)$ . Thus we have

$$\omega(\xi) = \{x \in \mathbb{R}^p : \exists t_k \rightarrow \infty \text{ with } S(t_k)\xi \rightarrow x\}.$$

Likewise, we define the  $\omega$ -limit set of a set bounded set  $B$  to be

$$\omega(B) = \{x \in \mathbb{R}^p : \exists t_k \rightarrow \infty, y_k \in B \text{ with } S(t_k)y_k \rightarrow x\}$$

We turn now to the definition of an attractor is. Loosely speaking, an attractor is a set to which all neighboring trajectories converge. Stable fixed points and stable limit cycles would be examples. Here, we refer to [Rob04] and [RKM81] for definitions. The first issue to be resolved is to precisely define what it means for two sets to approach each other. We assume that  $S(t)$  is a dynamical system on  $\mathbb{R}^p$ .

**Definition 2.23.** A set  $A \subset \mathbb{R}^p$  is said to **attract** the set  $B$  for the semigroup  $S(t)$ , if for any  $\epsilon > 0$  there is an  $t^* = t^*(\epsilon, B, A)$  such that  $S(t)B \subset \{a + \epsilon : a \in A\}$  for  $t \geq t^*$ .

**Definition 2.24.** A set  $M \subset \mathbb{R}^p$  is called a **(local) attractor**, if it is compact, invariant and attracts a neighborhood of itself. The set  $M$  is called a **global attractor**, if it attracts all bounded solutions.

A **strange attractor** is often defined to be an attractor that exhibits sensitive dependence on initial conditions, which is closely related to the concept of chaos and is briefly discussed below.

## 2.4 Dissipative Dynamical Systems

**Definition 2.25.** A dynamical system  $S(t)$  on  $\mathbb{R}^p$  is called **dissipative**, if there is a bounded, positively invariant set  $B$  with the property that, for any bounded set  $E \subset \mathbb{R}^p$  there ex. a time  $t^* = t^*(B, E) \geq 0$  such that  $S(t)E \subset B$  for all  $t > t^*$ . The set  $B$  then is called an **absorbing set**.

Again we recall the following result found on page 183 in [SH98].

**Theorem 2.26.** If a function  $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$  is locally Lipschitz and there exists an  $R > 0$  such that

$$\langle f(\mathbf{x}), \mathbf{x} \rangle < 0 \quad \forall \mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\| > R,$$

then the differential equation

$$\frac{d}{dt}x(t) = f(x(t))$$

generates a dissipative dynamical system  $S(t)$  on  $\mathbb{R}^p$  and the open ball  $B = B(0, R + \epsilon)$  is an absorbing set for any  $\epsilon > 0$ .

**Definition 2.27.** Let  $f$  be locally Lipschitz. The function  $f$  is called **dissipative**, if it satisfies the following condition:

$$\exists \alpha \geq 0, \beta > 0 : \quad \langle f(\mathbf{u}), \mathbf{u} \rangle \leq \alpha - \beta \|\mathbf{u}\|^2 \quad \forall \mathbf{u} \in \mathbb{R}^p$$

**Theorem 2.28.** If  $f : \mathbb{R}^p \rightarrow \mathbb{R}^p$  is locally Lipschitz and dissipative, then (2.2) defines a dissipative dynamical system on  $\mathbb{R}^p$  and

$$B = B(0, \sqrt{\alpha/\beta + \epsilon})$$

is an absorbing set for all  $\epsilon > 0$ .

The following results provides an explicit formula for the global attractor of a dissipative dynamical system (see [SH98]).

**Corollary 2.29.** A dissipative dynamical system with absorbing set  $B$  has the global attractor

$$\mathcal{A} = \omega(B) = \bigcap_{\tau \geq 0} S(\tau)B.$$

Note that this explicit formula might be hard to compute in practical situations. Another maybe more direct characterization of the global attractor is the following.

**Theorem 2.30.** If  $S(t)$  is a dissipative dynamical system, then the global attractor  $\mathcal{A}$  is the union of all complete bounded orbits of  $S(\bullet)$ .

## LaSalle's Invariance Principle

LaSalle's theory is an extension of Lyapunov's theory and deals with the case where (2.6) fails to hold. The main result is known as LaSalle's Invariance Principle. We assume (2.2) generates a dynamical system  $S(t)$  on  $\mathbb{R}^p$ .

**Definition 2.31.** *A trajectory of (2.2) is said to **approach the set**  $M$ , for  $M \subset \mathbb{R}^p$  if for every  $\epsilon > 0$  there exists a  $T > 0$  such that for every  $t > T$  there exists a point  $a \in M$  (possibly depending on  $t$ ) such that*

$$\|\phi(t, x_0) - a\| < \epsilon$$

**Theorem 2.32.** *Assume If  $f(0) = 0$  and  $V : \Omega \rightarrow \mathbb{R}^1$  is a continuously differentiable real valued function defined on a domain  $\Omega \in \mathbb{R}^p$  containing the origin and satisfying  $V(0) = 0$  with  $\dot{V}(x) \leq 0$  on  $\Omega$ . For a constant  $k \geq 0$ , let  $H_k$  be the component of the (sub level) sets  $S_k = \{x : V(x) \leq k\}$  which contains the origin. Suppose  $H_k$  is a closed and bounded subset of  $\Omega$ . Let  $E = \{x \in \Omega : \dot{V}(x) = 0\}$  and  $M$  be the largest invariant subset of  $E$ . Then every solution of (2.2) starting in  $H_k$  approaches the set  $M$  as  $t \rightarrow \infty$ .*

## 2.5 Invariant Manifolds

Invariant manifolds are used to examine the behavior of a dynamical system near an equilibrium point. For reasons of simplicity, we consider the finite dimensional autonomous differential system

$$\dot{\mathbf{u}}(t) = A\mathbf{u}(t), \quad \mathbf{u}(0) = U, \quad (2.7)$$

where  $A \in \mathbb{R}^{p \times p}$ .

**Definition 2.33.** *Given an equilibrium point  $\bar{\mathbf{u}}$ , the **unstable manifold** is defined to be the set*

$$W^u(\bar{\mathbf{u}}) = \{U \in \mathbb{R}^p : \mathbf{u}(t) \rightarrow \bar{\mathbf{u}} \text{ as } t \rightarrow -\infty\}.$$

*The **local unstable manifold** to be*

$$W^{u,\epsilon}(\bar{\mathbf{u}}) = \{U \in W^u(\bar{\mathbf{u}}) : \|\mathbf{u}(t) - \bar{\mathbf{u}}\| \leq \epsilon \quad \forall t \leq 0\}.$$

*The **stable manifold** is defined to be*

$$W^s(\bar{\mathbf{u}}) = \{U \in \mathbb{R}^p : \mathbf{u}(t) \rightarrow \bar{\mathbf{u}} \text{ as } t \rightarrow \infty\}$$

*and the **local stable manifold** is given by*

$$W^{s,\epsilon}(\bar{\mathbf{u}}) = \{U \in W^s(\bar{\mathbf{u}}) : \|\mathbf{u}(t) - \bar{\mathbf{u}}\| \leq \epsilon \quad \forall t \geq 0\}.$$

One is interested in the geometrical structure of the stable and unstable manifold mainly to infer properties of the dynamics of the underlying equation.

## 2.6 Chaos

There are various definitions for chaos and chaotic systems. For definitions and further reading, we refer to [Li07] or [Kie07] and the references therein. Among the first example of a chaotic system was investigated by Lorenz in 1963. The **Lorenz equation** was derived by Ed Lorenz from a drastically simplified model of convection rolls in the atmosphere. These equations also arise in models of lasers and dynamos and in the motion of water wheels (that can actually be built and tested). Lorenz discovered that the system

$$\begin{aligned}\dot{x} &= -\sigma x + \sigma y \\ \dot{y} &= rx - y - xz \\ \dot{z} &= -bz + xy,\end{aligned}\tag{LE}$$

exhibits chaotic behavior over a wide range of parameters. Here,  $\mathbf{x}(t) = [x(t), y(t), z(t)]^T \in \mathbb{R}^3$  and  $\sigma, r, b \geq 0$ . In most studies one sets  $\sigma = 10$ ,  $b = 8/3$ ,  $r = 28$  (see e.g. [Rob04]). Note that this a “simple” differential equation with a right hand side algebraic in  $x$ ,  $y$  and  $z$ . To visualize the Lorenz system, we include a plot of the Lorenz attractor in three dimensions with starting points  $x_0 = [-1, 0.1, -0.1]^T$  and  $x_0 = [-1, 0.1, -0.100001]^T$ . In particular Figure 2.1 shows the classical butterfly pattern. The rough butterfly shape of the so called “strange” attractor for the Lorenz equation is clearly visible in these plots. Even when two initial values are close, the resulting orbits may not be close at any given time  $t > 0$ . Thus, we have sensitive dependence on initial data. Another method often used to visualize chaotic behavior is the time plot of the  $x$ ,  $y$  or  $z$  components of a given orbit. Figure 2.2 contains plots of  $x(t)$ ,  $y(t)$  and  $z(t)$  for  $t > 0$ ,

The  $x(t)$ ,  $y(t)$  and  $z(t)$  components seem to be “almost” periodic, but do not exactly repeat. There is always an offset in what seems to be the periodic spikes. In practical matters this is the most often used technique to suggest chaotic behavior, since it can be plotted for any given system. We return to this point when studying the BDT equations below.

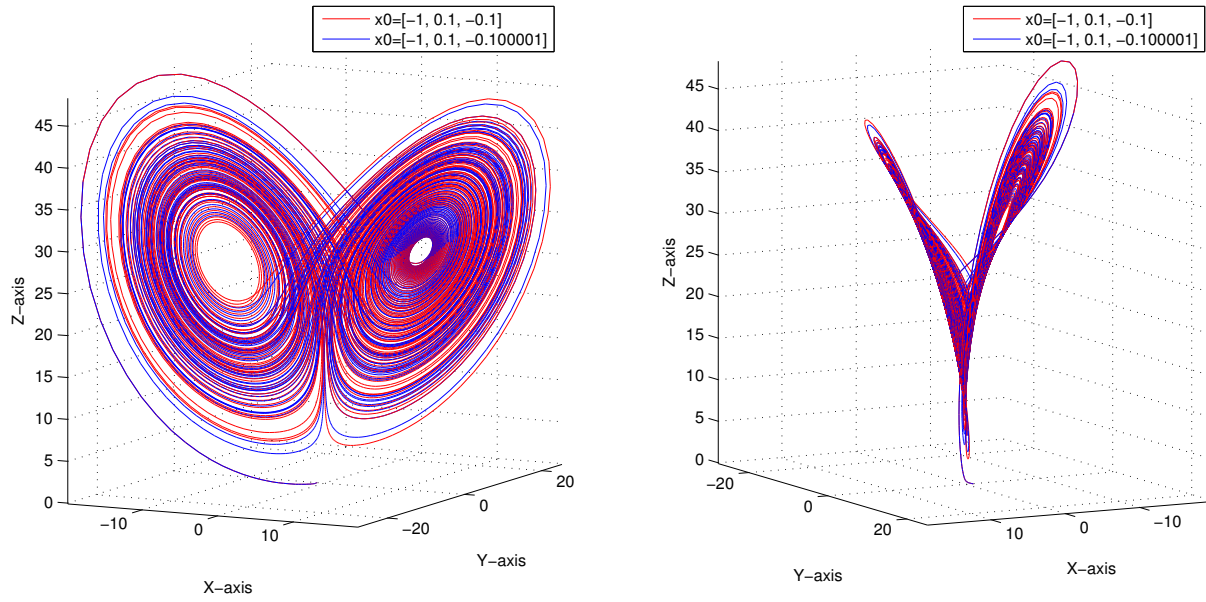


Figure 2.1: A view of the chaotic Lorenz attractor with solutions to two initial points

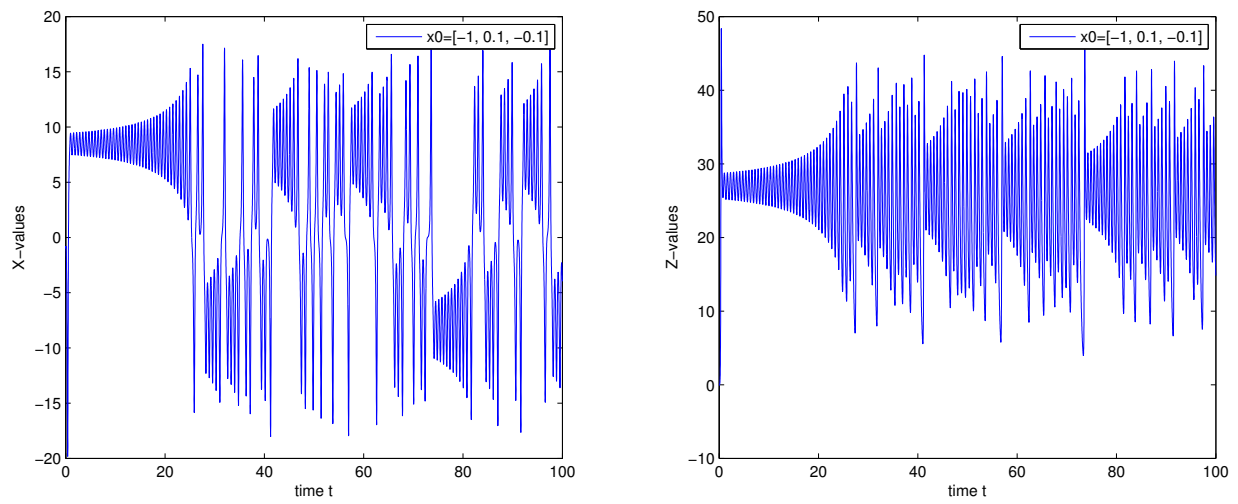


Figure 2.2: Time plots of x- and z-component of the chaotic orbits



# Chapter 3

## The Models

In this Chapter we introduce the models used in this thesis. The first model is a nonlinear ODE introduced by Baggett, Driscoll and Trefethen in 1995 [JSBT95] with a variation suggested by Waleffe in 1996 [Wal95]. We also discuss Burgers equation. All these models share some interesting properties observed for the Navier-Stokes equations.

### 3.1 The BDT Models

In [JSBT95] **Baggett, Driscoll and Trefethen** (BDT) introduced a class of ODEs of the form

$$\dot{\mathbf{u}}(t) = A\mathbf{u}(t) + g(\mathbf{u}(t))B\mathbf{u}(t),$$

where  $A$  is a real and non-normal matrix of the form

$$A = \begin{pmatrix} -2/Re & \beta & 0 \\ 0 & -2/Re & \beta \\ 0 & 0 & -2/Re \end{pmatrix}.$$

Here,  $g : \mathbb{R}^3 \rightarrow \mathbb{R}^1$  satisfies  $g(x) \geq 0$  and  $B$  is a real skew-symmetric matrix. Although the model is rather general, the case considered by [JSBT95] used  $g(\mathbf{u}) = \|\mathbf{u}\|_2$ . Waleffe pointed out in [Wal95] that the physical properties of the Navier-Stokes equations can be preserved better, if one chooses a rotational form of the nonlinearity to mimic

$$u \cdot \nabla u = \Omega \cdot u \quad \text{with} \quad \Omega = \frac{1}{2}|(\nabla u)^T - \nabla u|.$$

Therefore,  $B$  is chosen to have the form

$$B = \begin{bmatrix} 0 & -a & -b \\ a & 0 & c \\ b & -c & 0 \end{bmatrix},$$

where  $a, b, c$  are nonnegative real numbers (note that  $B^T = -B$ ). The common form of a rotation matrix can be obtained by applying the Cayley transform  $A \mapsto (I + A)(I - A)^{-1}$  to  $B$ . Note that the eigenvalues of the matrix  $A$  above are given by  $-\frac{2}{Re}$  with multiplicity 3. We modify the model in [JSBT95] to eliminate the multiplicity and set  $A$  to be

$$A = \begin{bmatrix} -\alpha & Re & 0 \\ 0 & -\beta & Re \\ 0 & 0 & -\gamma \end{bmatrix},$$

for positive  $\alpha, \beta, \gamma$ . With this modification, the matrix  $A$  has three different eigenvalues, namely  $\lambda_1 = -\alpha$ ,  $\lambda_2 = -\beta$  and  $\lambda_3 = -\gamma$ , which each have multiplicity one.

Thus, in this thesis we focus on the model

$$\frac{d}{dt} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} = \begin{bmatrix} -\alpha & Re & 0 \\ 0 & -\beta & Re \\ 0 & 0 & -\gamma \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} + \|x\|_2 \begin{bmatrix} 0 & -a & -b \\ a & 0 & c \\ b & -c & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} \quad (3.1)$$

and with

$$A = \begin{bmatrix} -\alpha & Re & 0 \\ 0 & -\beta & Re \\ 0 & 0 & -\gamma \end{bmatrix}, \quad B = \begin{bmatrix} 0 & -a & -b \\ a & 0 & c \\ b & -c & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{x} = \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix}$$

this system becomes

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + \|\mathbf{x}(t)\|_2 B\mathbf{x}(t). \quad (\text{BDT})$$

Note again, that  $B$  is skew symmetric, i.e.  $\langle Bx, x \rangle = 0$ . Thus, for any  $g : \mathbb{R}^3 \rightarrow \mathbb{R}^+$  the nonlinear term

$$G(\mathbf{x}) \triangleq \|\mathbf{x}\|_2 B\mathbf{x}$$

is conservative, i.e.

$$\langle G(\mathbf{x}), \mathbf{x} \rangle = 0 \quad (3.2)$$

and the nonlinearity does not affect the scalar product  $\langle \mathbf{x}(t), \dot{\mathbf{x}}(t) \rangle$ . This is a property of the Navier-Stokes equations and shared by the examples we consider here.

### 3.1.1 Properties of the BDT model

We will establish existence and uniqueness of solutions to the BDT equations (BDT) in later sections. Here we discuss additional properties of (BDT) that are similar to what is known for the Navier-Stokes equations and motivate the use of this model. As noted above, the conservative property of the nonlinear term is common to all models.

To illustrate some similarities in the behavior of the dynamical system, we plot some solutions with various initial conditions. In Figure 3.1 and Figure 3.2 we see that for sufficiently small

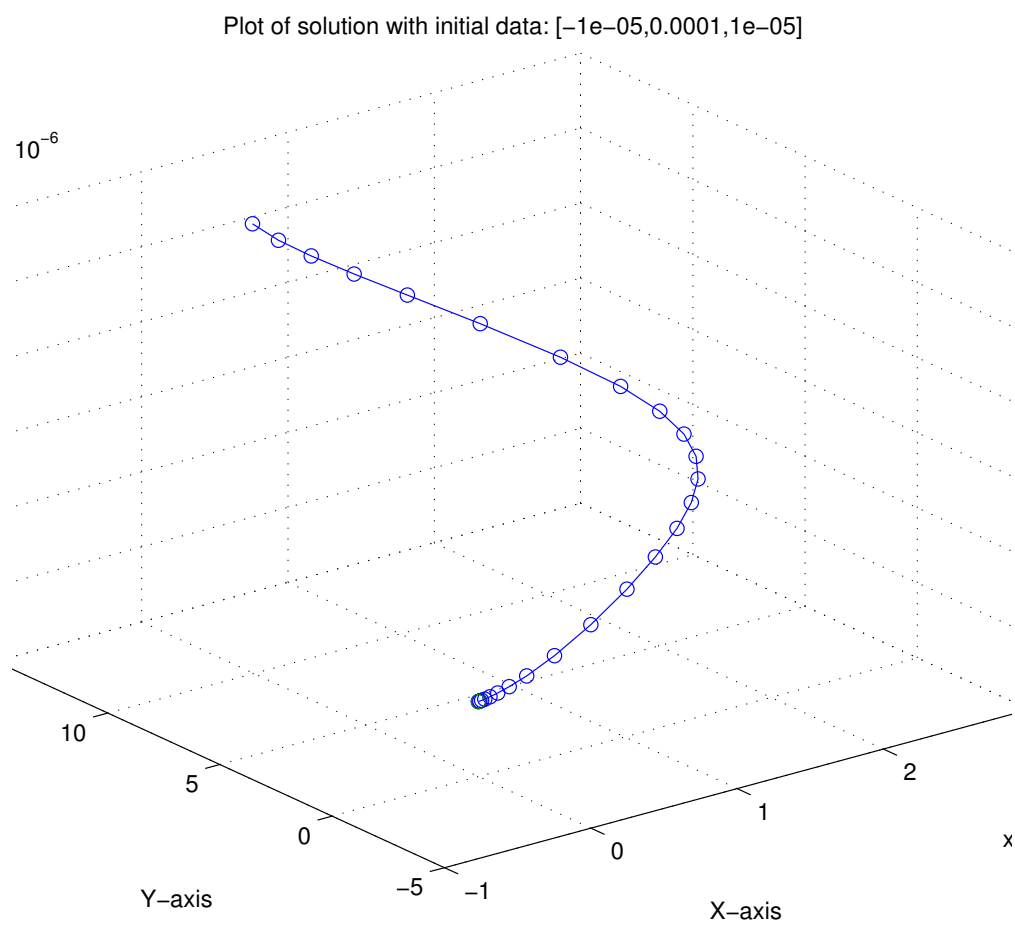


Figure 3.1: Solutions converging to zero for small initial conditions

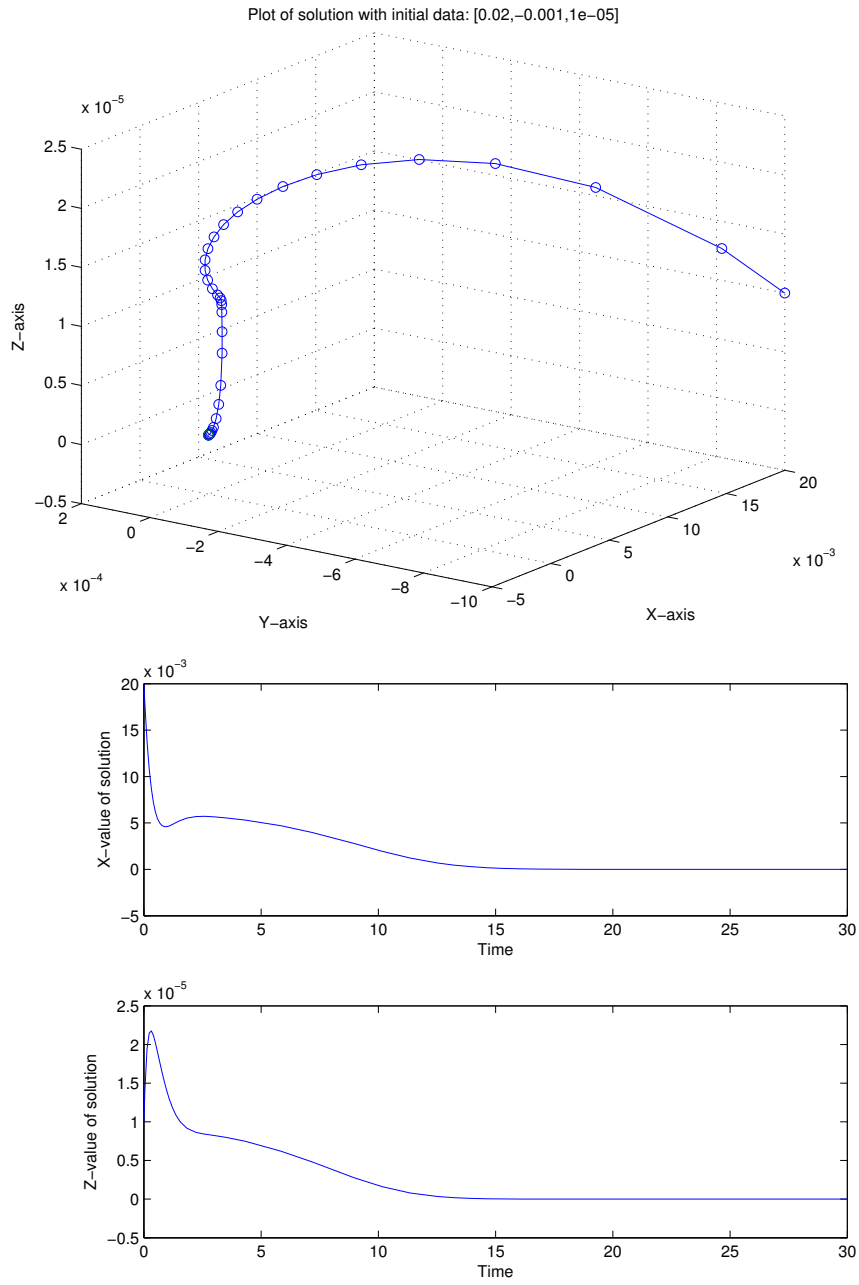


Figure 3.2: A solution of the BDT model going to zero with the according time plot

initial conditions, solutions converge to zero. Thus, the zero solution of (BDT) appears to be asymptotically stable as illustrated in the plots. In these runs, we started with setting the norm of the initial data to be less than  $10^{-3}$ .

By increasing the norm of the initial condition, the behavior of the dynamical system changes drastically. As illustrated in Figure 3.3, solutions no longer tend to zero but approach what appears to be a chaotic orbit. Thus we see a “transition” to a chaotic attractor. This behavior is also illustrated in Figure 3.4 and Figure 3.5. Note that unlike the Lorenz equation the “the shape of the (local) attractor” is similar to a dumbbell as illustrated in Figure 3.4. Time plots in Figure 3.5 indicate the chaotic nature of this attractor. In particular, there is no periodic pattern and hence the local attractor satisfies the more heuristic definition of chaotic, i.e. *aperiodic longterm behavior*.

The key observation is that even for small initial data, when the Reynolds number becomes sufficiently large one sees a transition to chaotic flows. In the 3D Navier-Stokes equations it is not known if all solution approach a global attractor because it is unknown if the exist unique global solutions. For the three state BDT model if we compute orbits for  $R = 35$  with a starting at the point  $\mathbf{x} = (8, 9, 10)^T$  we observe that the still converge the the dumbbell. The result shown in Figure 3.6. Thus, it is tempting to suggest that the dumbbell is a global attractor. However, in In this three state version of the BDT equations there exist solutions that do not converge towards the origin nor to the local attractor. In fact, as shown in Figure 3.6 there exist unbounded solutions. In this plot the initial condition  $\mathbf{x}_0 = [50, 46, 100]^T$  produces an unbounded solution.

The  $x$ - value of the solution is increasing with a recurring “spiral” pattern, whereas the norm is strictly increasing. Note the fast growth of the solution by comparing the scale of the plot to the size of the initial condition. Thus, we see that the chaotic “dumbbell” attractor is local and not a global attractor. We shall use the method of taming to modify the BDT models and generate dissipative dynamical systems such that this local attractor becomes global. This is exactly the motivation for taming as introduced by Roekner [RZ09] for the Navier-Stokes equations.

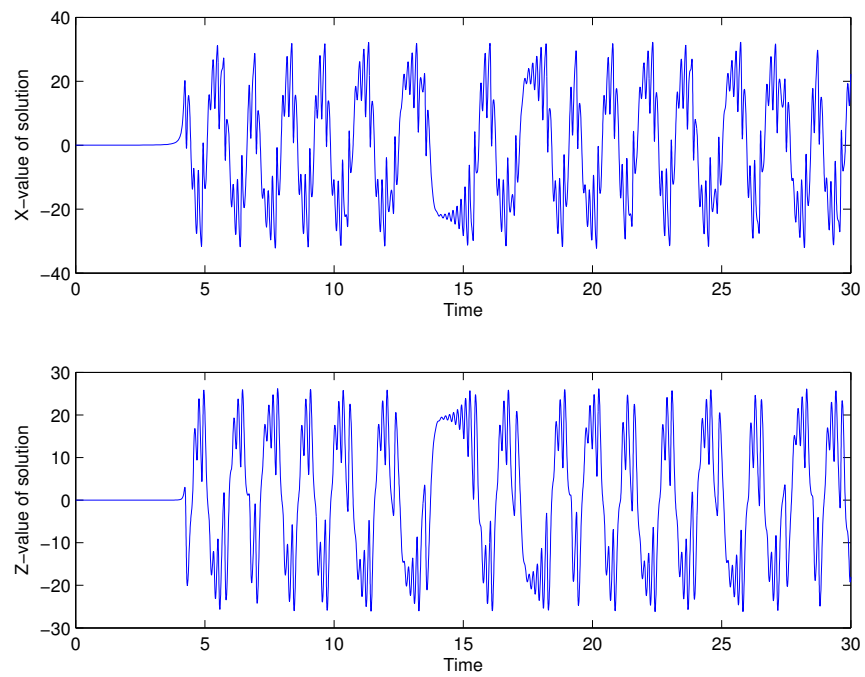
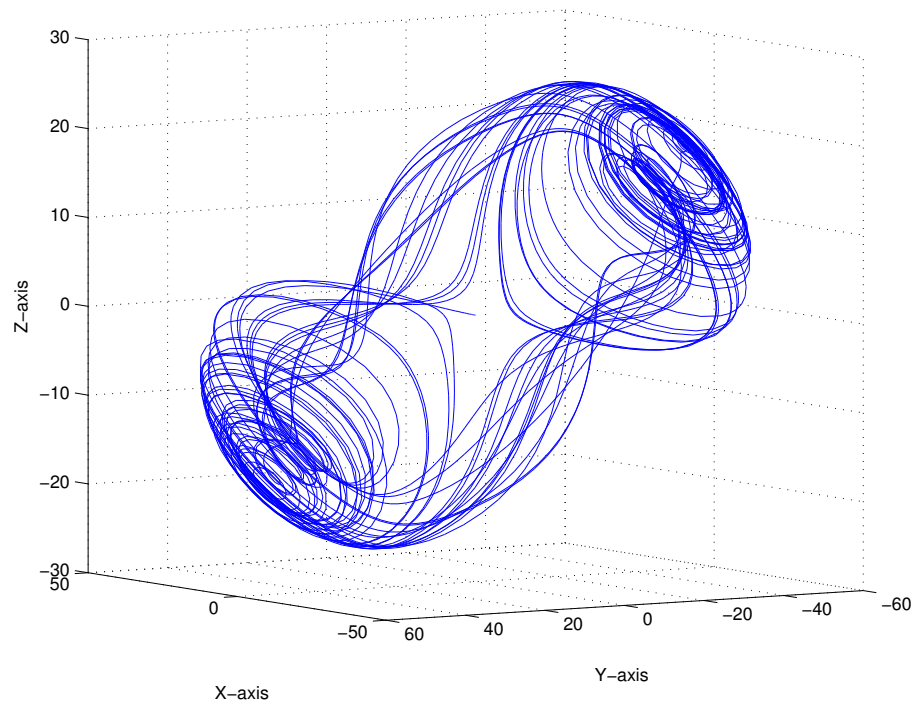


Figure 3.3: Slightly modified initial conditions, no longer converging to zero

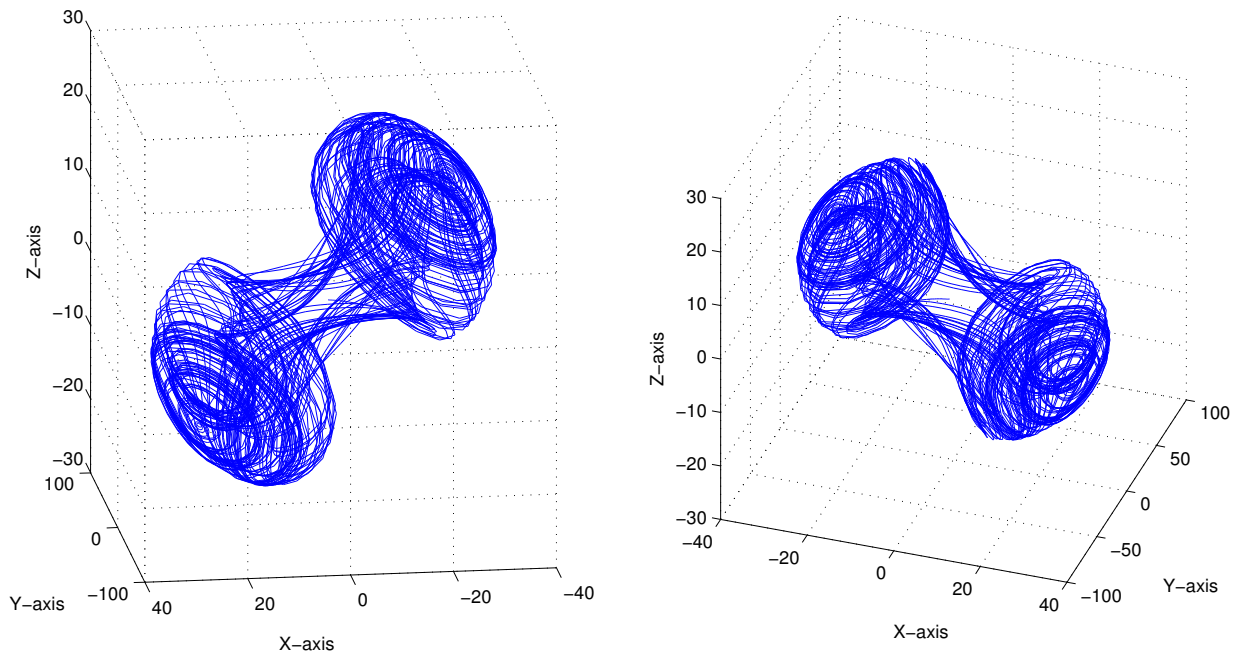


Figure 3.4: Views of a chaotic orbit for the BDT model.

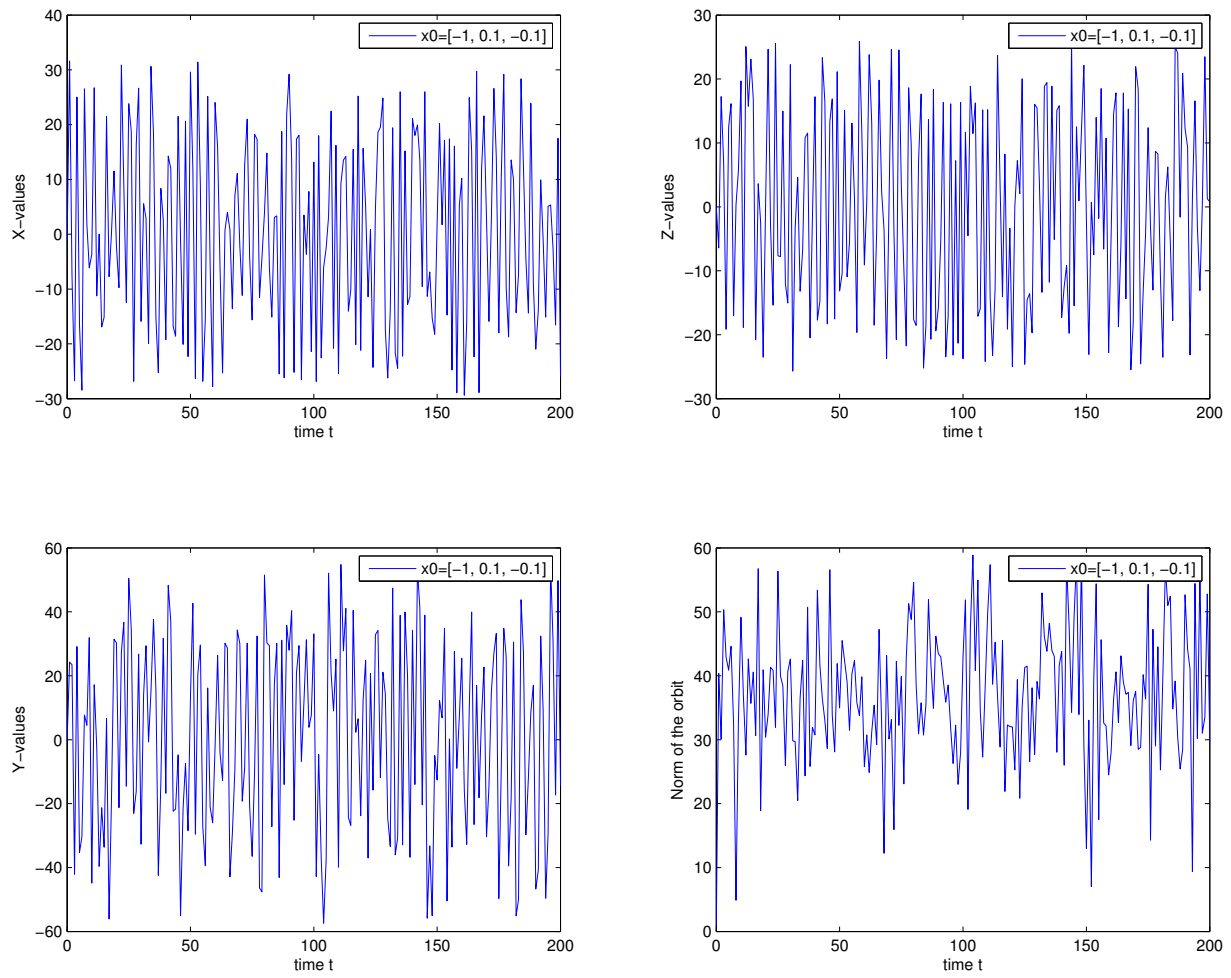


Figure 3.5: Time plots of a chaotic orbit



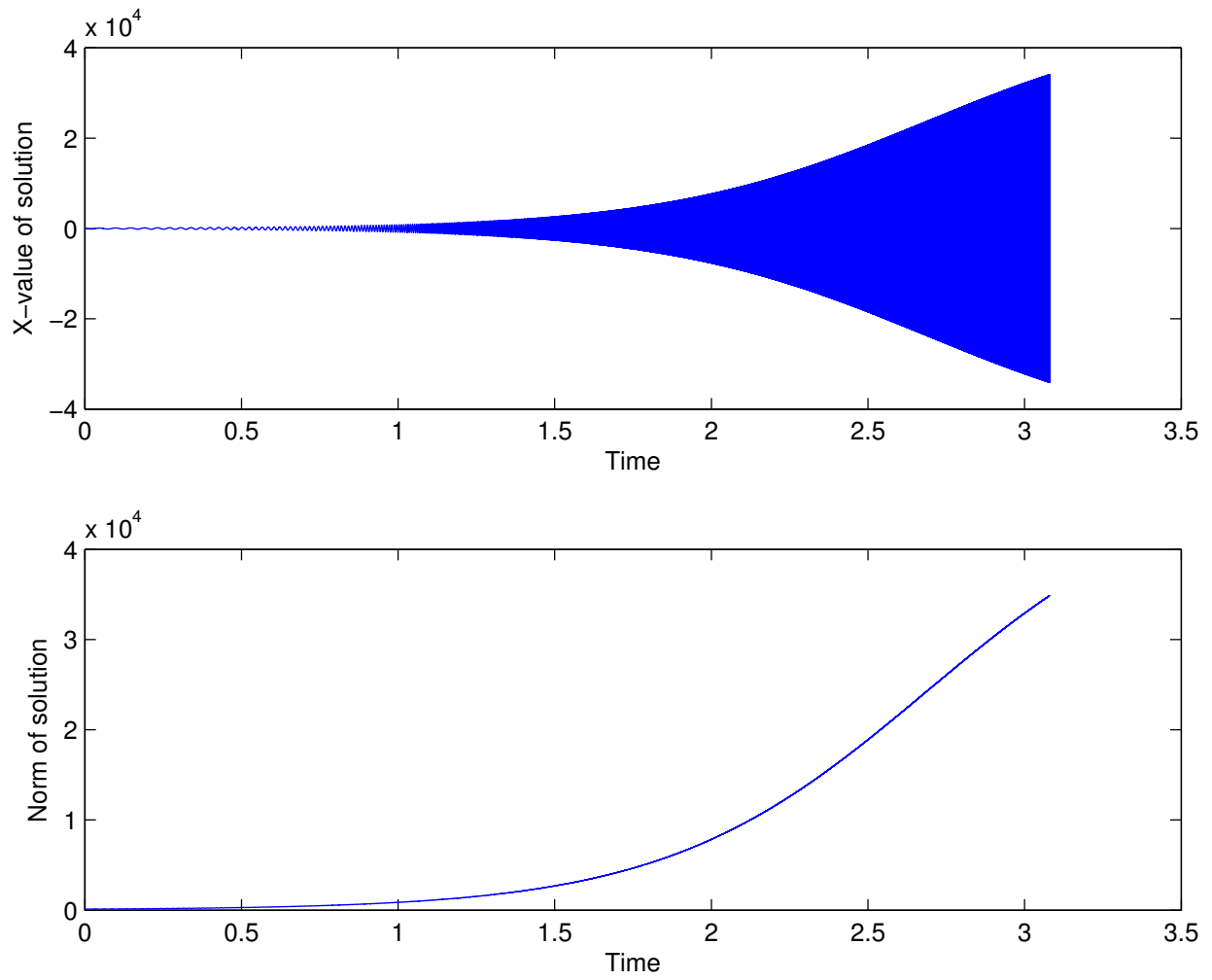


Figure 3.6: Time plots of an unbounded orbit, x-value and norm

## 3.2 Burgers equation

*Burgers equation* (named after Johannes Martinus Burgers, 1895 –1981) is an important equation in the field of fluid dynamics. We focus on the viscous Burgers equation given by

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \frac{\partial \mathbf{u}}{\partial x} = \mu \frac{\partial^2 \mathbf{u}}{\partial x^2} \quad \Leftrightarrow \quad \frac{\partial \mathbf{u}}{\partial t} + \frac{1}{2} \frac{\partial}{\partial x}(\mathbf{u}^2) = \mu \frac{\partial^2 \mathbf{u}}{\partial x^2}, \quad (3.3)$$

where the parameter  $\mu > 0$  represents the viscosity of the fluid. The special case  $\mu = 0$  is the inviscid Burgers equation

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \frac{\partial \mathbf{u}}{\partial x} = 0. \quad (3.4)$$

Burgers equation is often called a “poor man’s Navier-Stokes equation” since some important properties are shared by both systems. Also, Burgers equation is used as a model to test numerical algorithms for computational fluid dynamics.

### 3.2.1 Properties of the Burgers equation

Physically, Burgers’ equation can be used to describe the progression of a water wave from an initial profile  $x_0$  over time. Among the reasons to use this model in numerical approximations and simulations is the more realistic behavior compared to an ODE model (like the BDT model). Furthermore, the Cole-Hopf transformation provides a means to an analytic solutions for most boundary- and initial conditions by reducing it to the heat equation. The process is explained in more detail below; its essential ingredient is a change of variables. One should note here that due this fact a smoothing behavior is to be expected from Burgers’ equations similar to that of the heat equation.

### 3.2.2 Solution Techniques

We review the Cole-Hopf transformation since it is a standard method used to investigate Burgers’ equation. However, we note that the Cole-Hopf transformation cannot be easily applied to the mixed boundary conditions we consider in this paper.

#### The Cole-Hopf Transformation

For the general case of (3.3) with  $\mu > 0$ , there is a solution (discovered by Eberhard Hopf and Julian Cole independent of each other in 1950, 1951). This applies a transformation to the Burgers equation and shows the equivalence of the viscous Burgers equation with the heat equation. The theory of PDEs can then be applied to the heat equation and, transformed back, yield solutions of the viscous Burgers equation. Another consequence here is, that

the Burgers equation has many properties known for the heat equation, such as smoothness properties.

Considering the transformation

$$u = -2\nu \frac{\partial}{\partial x} \log(\phi), \quad (3.5)$$

Burgers' equation is turned into the heat equation. This yields

$$\frac{\partial \phi}{\partial t} = \nu \frac{\partial^2 \phi}{\partial x^2}, \quad (3.6)$$

which is solved with the well established PDE theory that gives an explicit formulation of the solution. The solution (after the backward transformation) to the Cauchy problem is given by

$$\mathbf{u}(t, x) = -2\nu \frac{\partial}{\partial x} \log \left( (4\nu t)^{-1/2} \int_{-\infty}^{\infty} \exp \left[ -\frac{(x-x')^2}{4\nu t} - \frac{1}{2\nu} \int_0^{x'} \mathbf{u}(x'', 0) dx'' \right] dx' \right).$$

This analytic solution it is often used as a *reference solution* for comparison with numerical algorithms. Note that the boundary conditions can become complicated.

### 3.2.3 Boundary conditions

Consider Burgers' equation with mixed boundary conditions such as a Dirichlet boundary condition at  $x = 1$  and a Neumann boundary condition at  $x = 0$ . Therefore,

$$u(t, 1) = 0 \quad \text{and} \quad \partial_x u(t, 0) = u_x(t, 0) = b \in \mathbb{R}.$$

Mixed boundary values correspond physically to fixing one side (Dirichlet) and specifying the flux of the field (Neumann) on the other side. In this thesis, the right end of the interval  $[0, 1]$  is set to 0 and set the flux on the left side at 0 to a given  $b \in \mathbb{R}$ . We consider this system because it is known that Burgers equation with a Neumann boundary condition is highly sensitive and leads to certain numeric instabilities (see [ABG<sup>+</sup>02]) We shall show how taming can help address this sensitivity.

The nonlinear term in Burgers' equation

$$z_t(t, x) + z(t, x)z_x(t, x) = \nu z_{xx}(t, x), \quad 0 < x < 1 \quad (3.7)$$

is given by the operator

$$G(z(\cdot)) = -z(\cdot) \frac{d}{dx} z(\cdot),$$

and is defined on a domain  $D(G) \subseteq L^2(0, 1)$ . The domain  $D(G)$  of the operator  $G$  depends on the boundary conditions. If  $z(\cdot) \in H^1(0, 1)$ , then it follows that

$$\langle Gz(\cdot), z(\cdot) \rangle_{L^2} = - \int_0^1 -z(x) \frac{d}{dx} z(x) z(x) dx = - \int_0^1 [z(x)]^2 \frac{d}{dx} z(x) dx = - \frac{1}{3} [z(x)]^3 \Big|_0^1. \quad (3.8)$$

If the boundary conditions are Dirichlet at both ends (i.e.,  $u(t, 0) = 0 = u(t, 1)$ ) or periodic (i.e.,  $u(t, 0) - u(t, 1) = 0$ ), then

$$-\frac{1}{3} [z(x)]^3 \Big|_0^1 = 0$$

and  $G(\cdot)$  is conservative. However, the problem with the Neumann boundary condition at the left end implies that  $D(G) = \{z(\cdot) \in H^1(0, 1) : z'(0) = z(1) = 0\}$  and in this case

$$\langle Gz(\cdot), z(\cdot) \rangle_{L^2} = -\frac{1}{3} [z(x)]^3 \Big|_0^1 = \frac{1}{3} [z(0)]^3 \neq 0.$$

Thus,  $G(\cdot)$  is no longer conservative. The case with Neumann boundary conditions at both ends is even more difficult. We shall not go into this problem here, but we note that the attractor for this problem is the set of all constant functions and hence not compact (see [BGS98] and [CT02]). Moreover, this problem has infinite sensitivity to small perturbations in the boundary conditions [EJG13].

### 3.3 Review of Common Model Features

All of the equations in this thesis are motivated by the Navier-Stokes equations and have certain similar properties. In particular, the systems have the form

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + G(\mathbf{x}(t)), \tag{3.9}$$

where  $G(\cdot)$  satisfies

$$\langle G(\mathbf{x}), \mathbf{x} \rangle = 0. \tag{3.10}$$

Condition 3.10 implies that the nonlinearity is conservative. The BDT models are of this form where

$$G(\mathbf{x}(t)) = \|\mathbf{x}(t)\|_2 B\mathbf{x}(t), \tag{3.11}$$

where  $B$  is skew symmetric.

The nonlinear term in Burgers equation is given by the operator

$$G(z(\cdot)) = -z(\cdot) \frac{d}{dx} z(\cdot),$$

and for certain boundary conditions

$$\langle G(z(\cdot)), z(\cdot) \rangle_{L^2} = 0. \tag{3.12}$$

The nonlinear operator appearing in the incompressible Navier-Stokes equations is also conservative. Another important feature of the BDT model is that the linear term in highly

non-normal. This also occurs when one linearizes the Navier-Stokes equation about a non-zero steady state flow. The combination of a non-normal linear term combined with a conservative nonlinearity produces a very sensitive system. Although the linear term in Burgers equation is self-adjoint, the nonlinear term is not conservative when a Neumann boundary condition is employed. Thus, the sensitivity comes from the Neumann boundary condition.

# Chapter 4

## Taming for the BDT Model

In this chapter, we consider the long term behavior of the dynamical system generated by the BDT model. We first discuss a two state version of the BDT model and then move to the three state model. As noted above, the BDT model appears to have a local chaotic attractor, but this attractor is not global. We then modify the BDT model by adding a taming term which we show produces a dissipative dynamical system. We end this chapter with numerical studies comparing various choices of the taming term and its influence on the dynamics of the system.

### 4.1 A Two State model

We start with a two state version of the BDT model given by

$$\begin{aligned} \dot{x}(t) &= -ax(t) + Re \cdot y(t) - y(t) \cdot \sqrt{x^2(t) + y^2(t)} \\ \dot{y}(t) &= -by(t) + x(t) \cdot \sqrt{x^2(t) + y^2(t)} \end{aligned}, \quad (4.1)$$

or equivalently, is the system

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} -a & Re \\ 0 & -b \end{bmatrix} \mathbf{x}(t) + \|\mathbf{x}(t)\| \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \mathbf{x}(t), \quad (4.2)$$

where  $\mathbf{x}(t) = (x(t), y(t))^T$ . Note that the structure is analogous to the three state BDT model, but this two state BDT model cannot produce chaotic behavior.

For the two state system, the Poincaré-Bendixon theorem gives three options a bounded solution can take:

- Approaching a critical point;

- Being a periodic orbit or
- Approaching a periodic orbit.

Here, we continue by showing the boundedness of all solutions and that the system is dissipative.

**Lemma 4.1.** *The system (4.2) generates a dissipative dynamical system.*

*Proof.* The main thing to check is the boundedness of solutions since then, we can refer to Poincaré-Bendixon theorem. Let  $x = r \cos(\varphi)$  and  $y = r \sin(\varphi)$  for  $r = \sqrt{x^2 + y^2}$ . To change the system to polar coordinates, do a change of variables

$$\begin{aligned} \dot{x}(t) &= \frac{d}{dt}[r \cdot \cos(\phi(t))] = \dot{r}(t) \cos(\phi(t)) - r(t) \dot{\phi}(t) \sin(\phi(t)) \\ \dot{y}(t) &= \frac{d}{dt}[r \cdot \sin(\phi(t))] = \dot{r}(t) \sin(\phi(t)) + r(t) \dot{\phi}(t) \cos(\phi(t)) \end{aligned} \quad (4.3)$$

Writing this in matrix form leads to

$$\begin{aligned} \begin{bmatrix} \cos(\phi(t)) & -r \sin(\phi(t)) \\ \sin(\phi(t)) & -r \cos(\phi(t)) \end{bmatrix} \begin{bmatrix} \dot{r}(t) \\ \dot{\phi}(t) \end{bmatrix} &= \begin{bmatrix} -ar(t) \cos(\phi(t)) + Re \cdot r(t) \sin(\phi(t)) - r^2 \sin(\phi(t)) \\ -br(t) \sin(\phi(t)) + r^2 \cos(\phi(t)) \end{bmatrix} \\ \Leftrightarrow \begin{bmatrix} \dot{r}(t) \\ \dot{\phi}(t) \end{bmatrix} &= \begin{bmatrix} -r(t) [b + a \cos(\phi(t))^2 - b \cos(\phi(t))^2 - \frac{1}{2} Re \sin(2\phi(t))] \\ r(t) + (a \sin(2\phi(t)))/2 - (b \sin(2\phi(t)))/2 - Re \sin(\phi(t))^2 \end{bmatrix} \end{aligned}$$

Hence we arrive at our dynamical system in polar coordinates, as

$$\begin{aligned} \dot{r}(t) &= -r(t) \left[ b + a \cos(\phi(t))^2 - b \cos(\phi(t))^2 - \frac{1}{2} Re \sin(2\phi(t)) \right] \\ \dot{\phi}(t) &= r(t) + (a \sin(2\phi(t)))/2 - (b \sin(2\phi(t)))/2 - Re \sin(\phi(t))^2 \end{aligned} \quad (4.4)$$

With the trigonometric identity  $\sin(2\phi(t)) = 2 \sin(\phi(t)) \cos(\phi(t))$ , the equation  $\dot{r}(t) \leq 0$  is equivalent to

$$b \sin^2(\phi(t)) + a \cos^2(\phi(t)) - Re \cos(\phi(t)) \sin(\phi(t)) \geq 0.$$

So the system is decreasing in radius if

$$b \sin^2(\phi(t)) + a \cos^2(\phi(t)) \geq Re \cos(\phi(t)) \sin(\phi(t)).$$

Using the fact that  $\cos(\phi(t)) \sin(\phi(t)) = \frac{1}{2} \sin(2\phi(t))$ , this leads to

$$\begin{aligned} b \sin^2(\phi(t)) + a \cos^2(\phi(t)) &\geq \frac{Re}{2} \sin(2\phi(t)) \\ \Leftrightarrow \frac{2b}{Re} \sin^2(\phi(t)) + \frac{2a}{Re} \cos^2(\phi(t)) &\geq \sin(2\phi(t)) \end{aligned}$$

Which holds for all relevant values of  $a, b, Re > 0$  considered in this thesis.  $\square$

**Lemma 4.2.** *The system (4.2) has a **global attractor** consisting of its equilibria and the according unstable manifolds.*

For the simulations in the tamed case for two states, the following tamed system is used:

$$\begin{aligned} \dot{x}(t) &= -ax(t) + Re \cdot y(t) - y(t) \cdot \sqrt{x^2(t) + y^2(t)} - \epsilon H(r(t)^2 - N)r(t)^2 x(t) \\ \dot{y}(t) &= -by(t) + x(t) \cdot \sqrt{x^2(t) + y^2(t)} - \epsilon H(r(t)^2 - N)r(t)^2 y(t) \end{aligned}, \quad (4.5)$$

where  $r(t) = \sqrt{[x(t)]^2 + [y(t)]^2}$ . The heaviside function  $H$  ensures the taming only takes effect if  $[x(t)]^2 + [y(t)]^2 \geq r^2$ .

### 4.1.1 Numerical Simulations

Since we are considering the two dimensional case, it is helpful to take a look at the behavior of (4.2) to get a feeling for the behavior of the kind of rotational nonlinearity we also encounter later on.

Two typical orbits of the equation with initial points  $\mathbf{x}_1 = (4, -7, -1)^T$  and  $\mathbf{x}_2 = (10, 20, 1)^T$  look like: In order to see how the local structure is influenced by the taming term, we have the global attractor of this equation with and without the taming here:



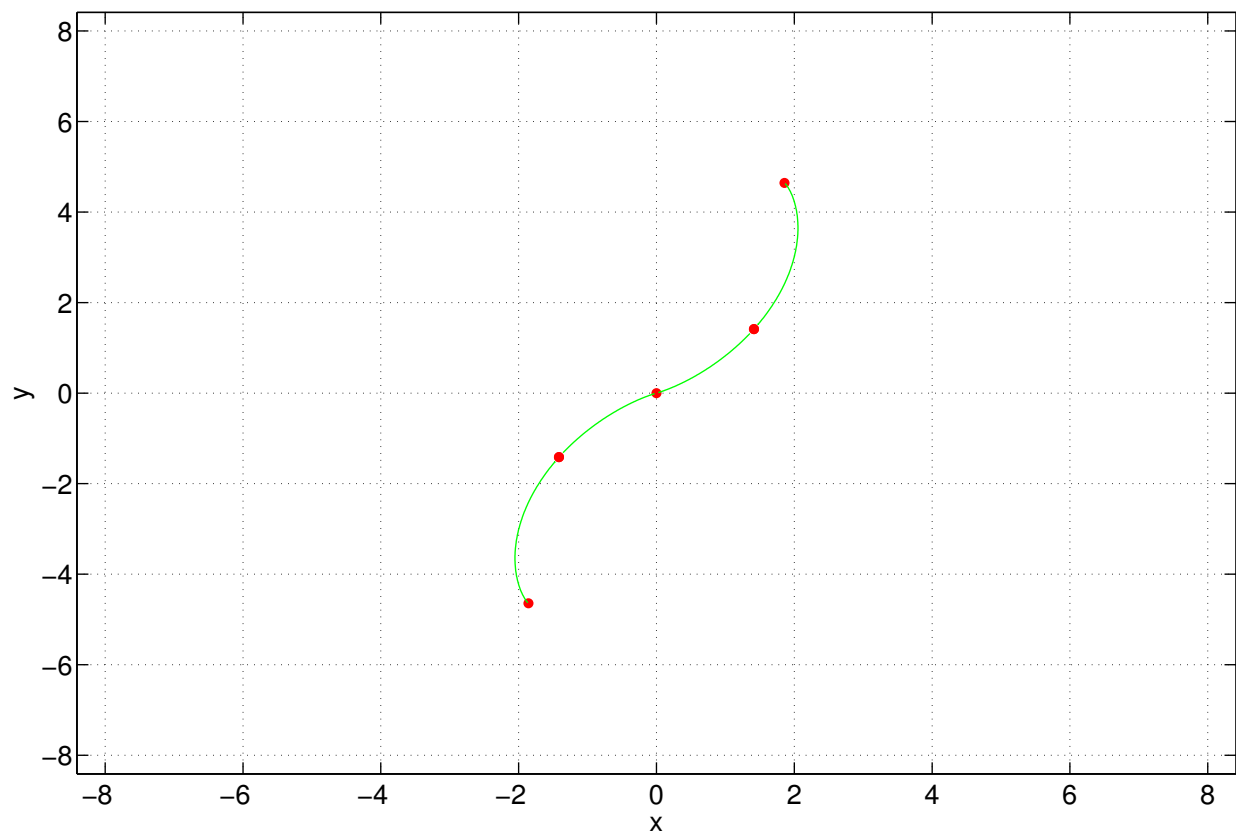


Figure 4.1: The global attractor

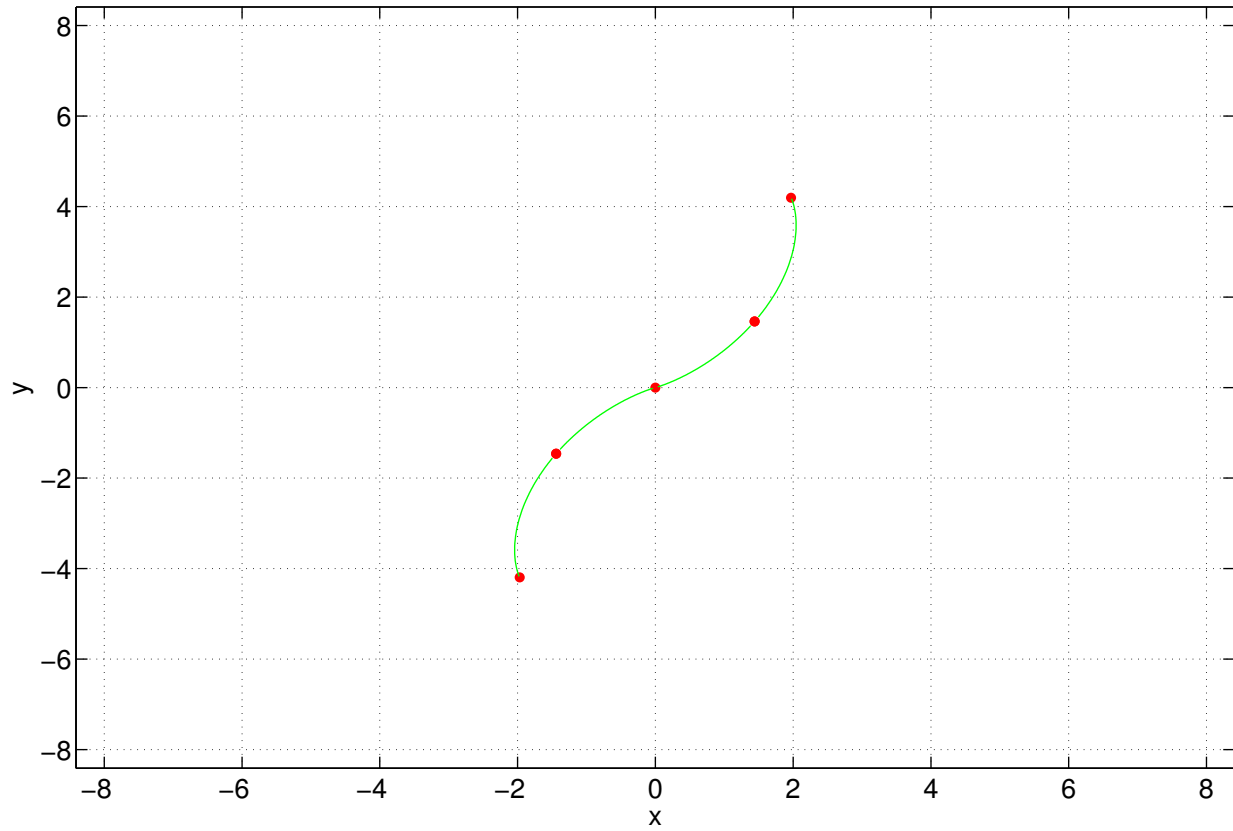


Figure 4.2: The global attractor in the tamed case

Although a global taming term is used, very little influence on the geometry of the attractor can be seen.

Using PPlane8, the influence of the taming term is shown by comparing the plot of solutions to the tamed and untamed system.

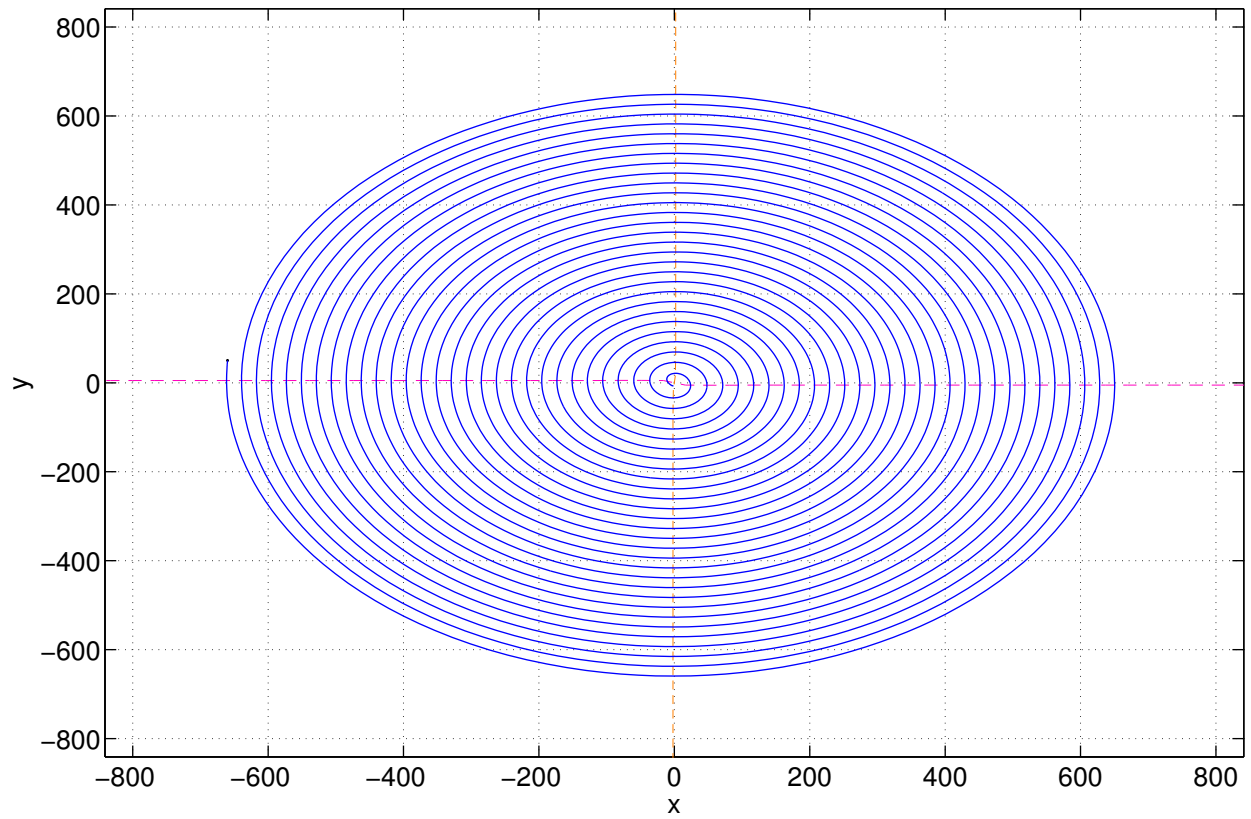


Figure 4.3: Orbits in the two-state system (no taming)

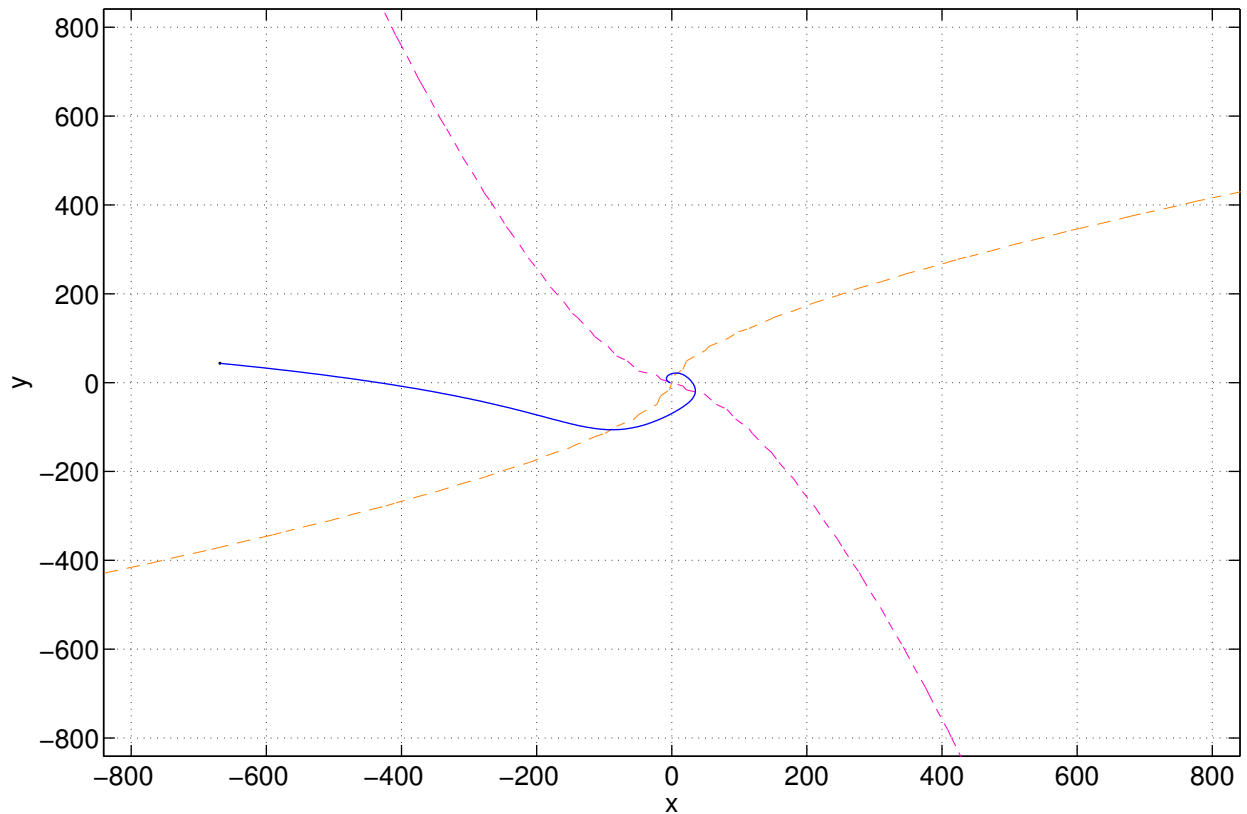


Figure 4.4: Orbits in tamed two-state system

One clearly sees that convergence speed to the (global) attractor in the 2D model is enhanced by the influence of the taming term.

Notice the scale ranging from -800 to 800 both on  $x$  and  $y$  (large scale behavior). The nullclines  $\{y \in \mathbb{R}^2 : x = 0\}$  and  $\{x \in \mathbb{R}^2 : y = 0\}$  are represented in pink and yellow in the graphs above. On the nullclines, the change in dynamics on a large scale can be seen more drastically. On a small scale, the influence of the taming term is minimal.

## 4.2 Three state model

Here, we consider three states in the model (see e.g. [TG06]) compared to two states in the previous section. Note that in three dimensions, we cannot rely on the Poincaré-Bendixon theorem to characterize the behavior of solutions. Furthermore, one encounters behavior such as chaos, which is not possible in two dimensions.

Consider the tamed version of the above mentioned equation

$$\frac{d}{dt}\mathbf{x}(t) = A\mathbf{x}(t) + G(\mathbf{x}(t)) - \epsilon\|\mathbf{x}(t)\|_2^2 \cdot \mathbf{x}(t), \quad (4.6)$$

with  $G(\mathbf{x}(t)) = \|\mathbf{x}(t)\|_2 B\mathbf{x}(t)$  - written out as

$$\frac{d}{dt}\mathbf{x}(t) = A\mathbf{x}(t) + \|\mathbf{x}(t)\|_2 B\mathbf{x}(t) - \epsilon\|\mathbf{x}(t)\|_2^2 \cdot \mathbf{x}(t), \quad (4.7)$$

with a simple cubic taming term and an  $\epsilon > 0$  small. This form of a taming term allows for more straight forward analytic results whereas the extension to taming terms with only asymptotic behavior as a requirement is obvious.

In literature, there are some attempts to compute the global attractor for the Navier-Stokes equation with regularization in three dimensions (see e.g. [KT09], [CF06], [Ros98], [MRR07], [Sel96], [ZD13]) or again considering weak formulations of Navier-Stokes, e.g. [KV07].

**Theorem 4.3.** *The tamed equation (4.6) above generates a dissipative dynamical system. Moreover, the open ball  $B(0, R + \delta)$  is an absorbing set for  $\delta > 0$ .*

*Proof.* We consider

$$f(\mathbf{x}) = A\mathbf{x} + \|\mathbf{x}\|B\mathbf{x} - \epsilon\|\mathbf{x}\|^2 \cdot \mathbf{x}$$

First of all, we establish that  $f$  satisfies a local Lipschitz condition, as can be seen the following way:

$$\begin{aligned} \|f(\mathbf{x}) - f(\mathbf{y})\| &= \left\| A\mathbf{x} + \|\mathbf{x}\|B\mathbf{x} - \epsilon\|\mathbf{x}\|^2 \cdot \mathbf{x} - A\mathbf{y} - \|\mathbf{y}\|B\mathbf{y} + \epsilon\|\mathbf{y}\|^2 \cdot \mathbf{y} \right\| \\ &\leq \|A\|\|\mathbf{x} - \mathbf{y}\| + \left\| (\|\mathbf{x}\|B - \epsilon\|\mathbf{x}\|^2)\mathbf{x} - (\|\mathbf{y}\|B - \epsilon\|\mathbf{y}\|^2)\mathbf{y} \right\| \\ &\leq \|A\|\|\mathbf{x} - \mathbf{y}\| + \left\| (MB - \epsilon M^2 I)\mathbf{x} - (MB - \epsilon M^2 I)\mathbf{y} \right\| \\ &\leq (\|A\| + \|MB - \epsilon M^2 I\|)\|\mathbf{x} - \mathbf{y}\|, \end{aligned}$$

where  $M = \max\{\|\mathbf{x}\|, \|\mathbf{y}\|\}$ . This shows that  $f$  is locally Lipschitz.

Now we show that  $\langle f(\mathbf{x}), \mathbf{x} \rangle < 0$  for  $\mathbf{x} \geq R$  for some  $R$ . Therefore evaluate  $\langle f(\mathbf{x}), \mathbf{x} \rangle$  (with  $\langle \cdot, \cdot \rangle$  being the standard scalar product on  $\mathbb{R}^3$ ):

$$\langle f(\mathbf{x}), \mathbf{x} \rangle = \langle A\mathbf{x}, \mathbf{x} \rangle + \|\mathbf{x}\|\langle B\mathbf{x}, \mathbf{x} \rangle - \epsilon\|\mathbf{x}\|^2 \langle \mathbf{x}, \mathbf{x} \rangle$$

Noting that  $B = B^T$ , we have  $\langle B\mathbf{x}, \mathbf{x} \rangle = 0$  for all  $\mathbf{x} \in \mathbb{R}^3$ .

$R(x_1x_2 + x_2x_3)$ . For all  $x$  with  $\|\mathbf{x}\| \geq R$  for some  $R > 0$  it holds Writing out  $\langle A\mathbf{x}, \mathbf{x} \rangle = -\alpha x_1^2 + Re(x_1x_2 + x_2x_3) - \beta x_2^2 - \gamma x_3^2$  it follows:

$$\begin{aligned} \langle f(\mathbf{x}), \mathbf{x} \rangle &= \langle A\mathbf{x}, \mathbf{x} \rangle - \epsilon\|\mathbf{x}\|^2 \langle \mathbf{x}, \mathbf{x} \rangle \\ &= -\alpha x_1^2 + Re(x_1x_2 + x_2x_3) - \beta x_2^2 - \gamma x_3^2 - \epsilon\|\mathbf{x}\|^4 \\ &\leq Re(x_1x_2 + x_2x_3) - \epsilon\|\mathbf{x}\|^4 < 0 \quad \text{for } \|\mathbf{x}\| \geq R. \end{aligned}$$

Thus, on a ball of radius  $R$  we have  $\langle f(\mathbf{x}), \mathbf{x} \rangle < 0$  and by a theorem in [SH98, p.183] it follows that (4.6) is dissipative.  $\square$

Theorem 4.3 implies that tamed equation (4.6) generates a dissipative dynamical system with a global attractor inside times  $t \geq 0$  can be guaranteed as well as boundedness of the solutions.

Note that the proofs holds analogously for taming functions that have the same asymptotic order with obvious modifications.

To investigate the attractor of the three dimensional Navier-Stokes equation, we refer to [KYO89], [Ros98], [Sel96], [HS00], [VC02], [JH09] and [Kac10].

### 4.2.1 Numerical Studies

For numerical studies, we consider the BDT model as introduced in Chapter 3, as

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + G(\mathbf{x}(t)) = A\mathbf{x}(t) + \|\mathbf{x}(t)\|_2 B\mathbf{x}(t).$$

Among those studies are various designs for the taming function, considerations of periodic orbits and variations of the Reynolds number. For calculations in Matlab, we fix the parameters as follows  $\alpha = 1, \beta = 1.1, \gamma = 1.2, Re = 35, a = 0.9, b = 0.3, c = 0.2$  Therefore, the resulting ODE has the form

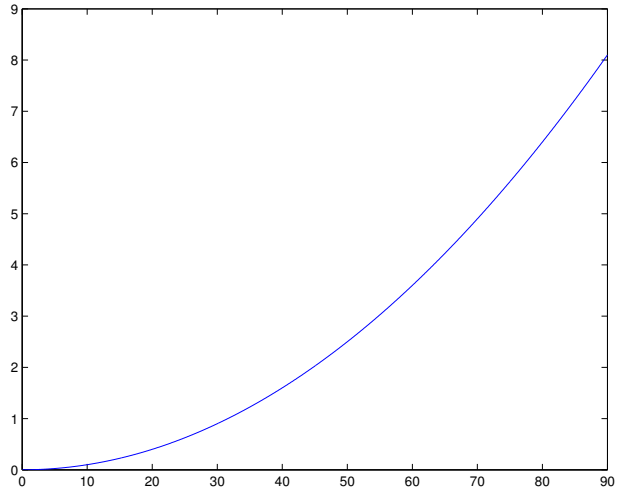
$$\dot{\mathbf{x}}(t) = \begin{bmatrix} -1.1 & 35 & 0 \\ 0 & -1.2 & 35 \\ 0 & 0 & -1.3 \end{bmatrix} \mathbf{x}(t) + \|\mathbf{x}\|_2 \begin{bmatrix} 0 & -0.9 & -0.3 \\ 0.9 & 0 & 0.2 \\ 0.3 & -0.2 & 0 \end{bmatrix} \mathbf{x}(t).$$

### 4.2.2 The Taming Term

Here the designs of the taming term is investigated further. Since there are many possible ways to construct a taming function, we will limit our focus to three specific but representative versions. Note that  $r = \|\mathbf{x}\|_2$ .

First of all the “simplest” way of taming, as used in the proof of (4.3):

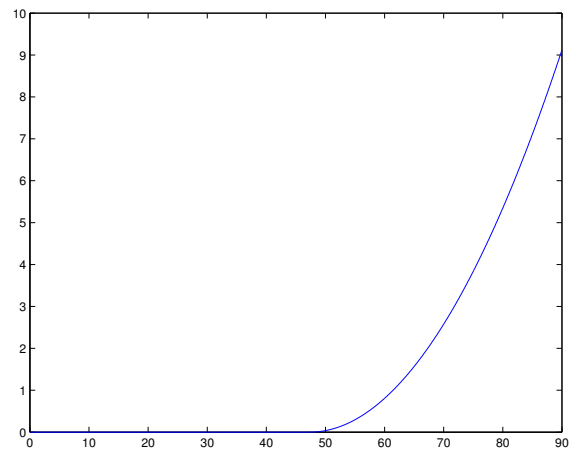
$$T_1(r) = \epsilon \cdot r^2$$



This taming term has an impact in every point besides 0. As mentioned earlier, it might be desirable not to influence the behavior of the dynamical system near the chaotic attractor; which the “simplest” case above does. Therefore, the second taming term considered is the following.

*Notation:* In the following two taming terms, we deviated from the notation before by using  $r$  instead of  $\|x\|$ . This is a result of the shape of the taming region: instead of assuming a ball of radius  $r$  centered at the origin, we assume an ellipsoid  $\mathcal{E}$  centered at the origin. The notation is adjusted to reflect this deviation.

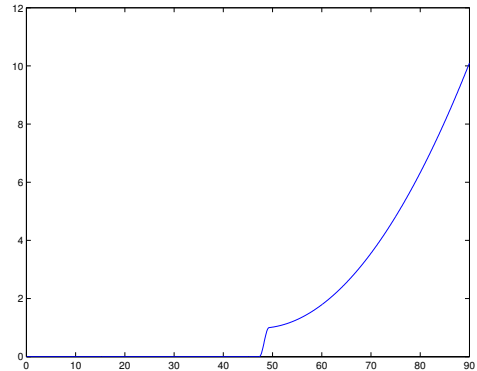
$$T_2(\|x\|) = \begin{cases} 0 & \text{if } x \in \mathcal{E} \\ \frac{(\|x\| - N_1)^2 - \frac{1}{2}}{2\nu} - \frac{1}{4\nu} & \text{outside of } \mathcal{E} \end{cases}$$



With  $N_1$  being the norm of the projection of  $x$  onto  $\mathcal{E}$  along the line  $l_a = a \cdot x, a \in \mathbb{R}$ .

In the following, we give a counterexample for a non smooth taming term to show that smoothness is needed to avoid uncharacteristic behavior. The presented function is *piecewise smooth*, hence not admissible for the procedure by Roeckner and Zhang.

$$T_3(\|x\|) = \begin{cases} 0 & \text{if } x \in \mathcal{E} \\ \frac{\sin((\|x\| - N_1)(\frac{\pi}{2}) - \frac{\pi}{2}) + 2}{2} & \text{if } N_1 \leq \|x\| \leq N_2 \\ \frac{(\|x\| - N_1)^2 - \frac{1}{2}}{2\nu} - \frac{1}{4\nu} & \text{if } x \geq N_2 \end{cases}$$



Where  $N_1$  denotes the intersection of the line through the origin with direction  $x$  and the set  $\mathcal{E}$ . A profile of the taming term is plotted on the right hand side.

For practical reasons, we examine the first version of the taming term, for the second and third, we stick closer to the idea of the taming term by Michael Roeckner in [RZ09], in the second version with a smooth taming, in the third with an “almost” smooth taming to visualize the contrast and point out the requirement in smoothness.

Also note that Roeckner and Co. do not explicitly construct the taming function but give a slope and require a smooth transition from 0 to that slope.

### Choosing the shape for $\mathcal{E}$

Roeckner and Co. use a straight ball with radius  $N$  as a region without effect of the taming term. For numerical reasons, we chose to fit the region of ineffective taming to the given equation, here the BDT model. Therefore, we try to find an ellipsoid to a given data set consisting of points of a forward orbit converging to the dumbbell attractor. Note that an ellipsoid is equivalent to a level set of a second order function from  $\mathbb{R}^3$  to  $\mathbb{R}$  (second order including mixed terms).

### Fitting the ellipsoid

We want to customize the ellipsoid  $\mathcal{E}$  to fit our dynamical system. To achieve that we fit an ellipsoid  $\mathcal{E}$  to the attractor using least squares optimization on a data set given by the points of a chaotic orbit. The resulting ellipsoid is the solution to the following equation ( $\mathbf{x} = (x, y, z)^T$ )

$$0.0014x^2 + 0.0006y^2 + 0.0019z^2 - 0.0006xy + 0.0006xz + 0.0006yz - 0.0018x + 0.0004y - 0.0004z = 1.$$



Since the fitted ellipsoid would be inside the attractor, we regard the following as a function  $g$  where  $\mathcal{E} = \{\mathbf{x} \in \mathbb{R}^3 : g(\mathbf{x}) = 2\}$ , so the ellipsoid that is actually used is a level set of the function  $g$  with RHS 2. This shows the initial fitted ellipsoid ( $\{g(\mathbf{x}) = 1\}$ ) on the left and the level set  $\{g(\mathbf{x}) = 2\}$  on the right.

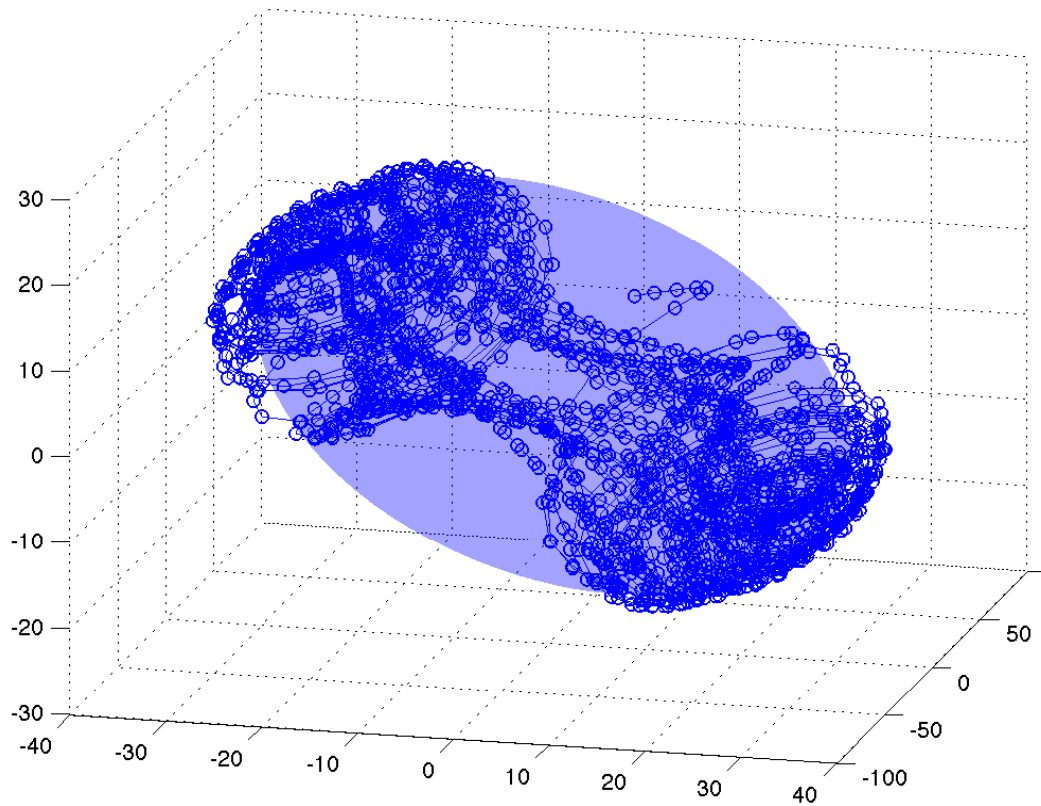


Figure 4.5: Fitted ellipsoid to solution with least squares fitting

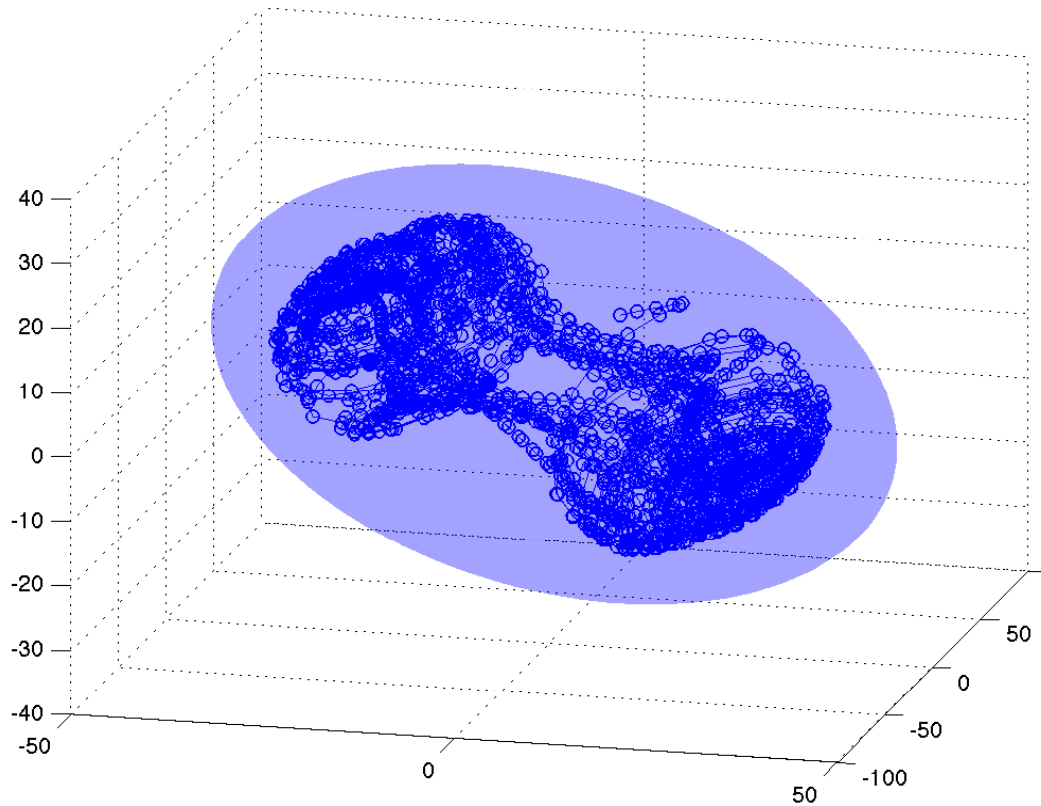


Figure 4.6: Scaled fitted ellipsoid later used for taming

### Numerical studies on the taming term

In three dimensions the problem becomes less easy to visualize and we use colorscale plots below to overcome that difficulty. As a “measure” how much the taming term effects the behavior of the dynamical system generated by (4.6), we use the following scalar product

$$\langle \dot{\mathbf{x}}, \mathbf{x} \rangle \quad \text{or} \quad \langle f(\mathbf{x}), \mathbf{x} \rangle.$$

Here we compared the effects of the taming term to the scalar product  $\langle f(x), x \rangle$  numerically for both the tamed and untamed  $f$  function. The left side is the original  $\langle Ax, x \rangle$  and the right side is  $\langle Ax - xT(x), x \rangle$ , all on the boundary of the  $\mathcal{E}$ :

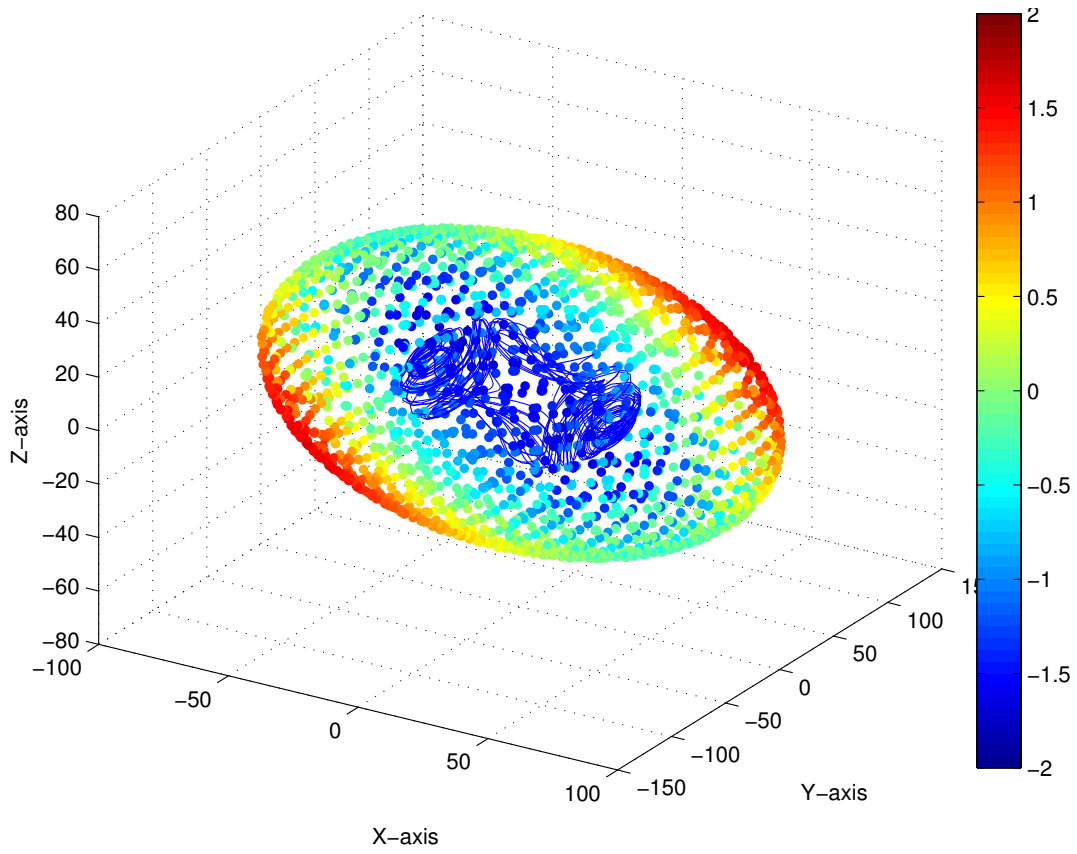


Figure 4.7: Scalar product of original equation on boundary of scaled ellipsoid, scale 6

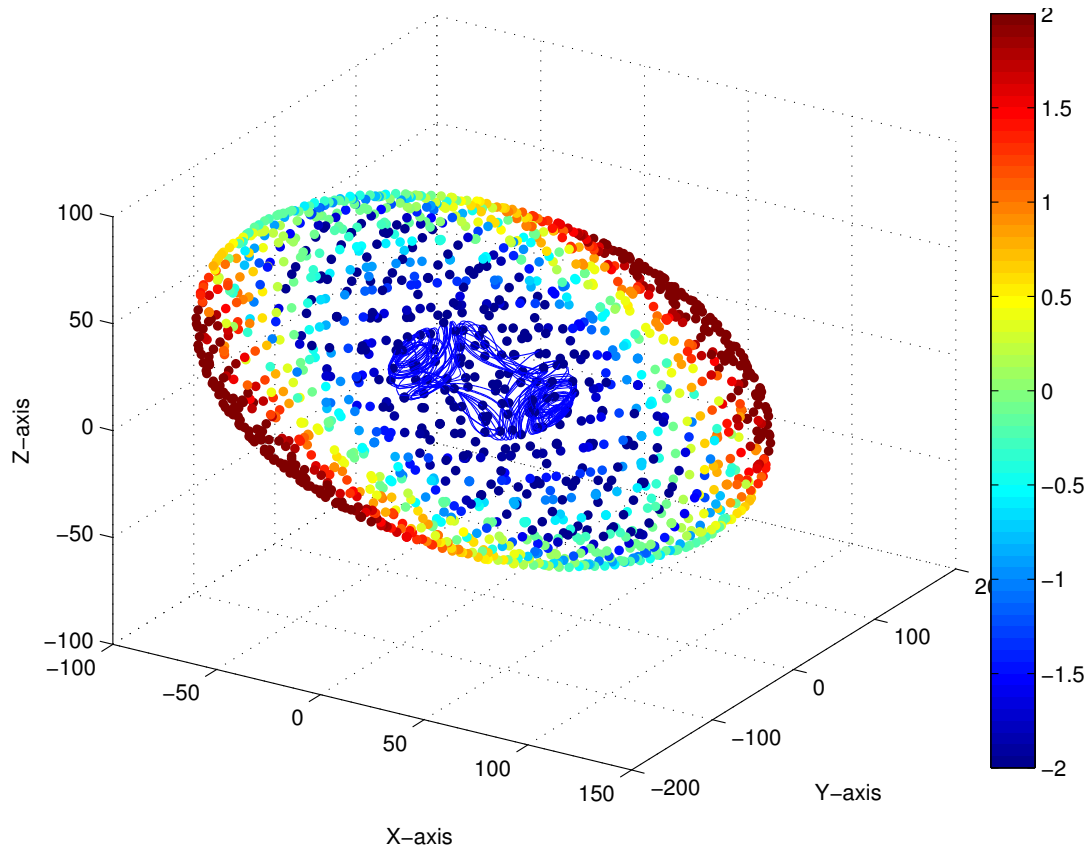


Figure 4.8: Scalar product of original equation on boundary of scaled ellipsoid, scale 12

The same orbits displayed with taming added to them are shown below.

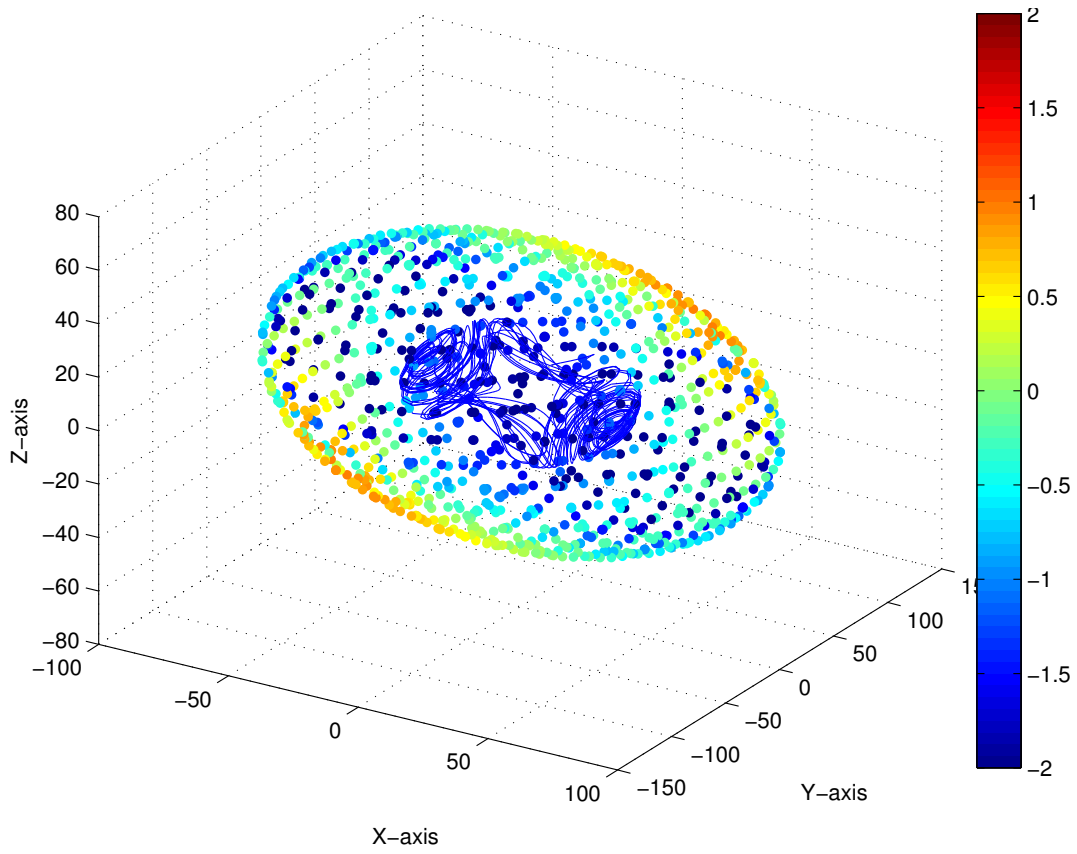


Figure 4.9: Scalar product of **tamed** equation on boundary of scaled ellipsoid, scale 6

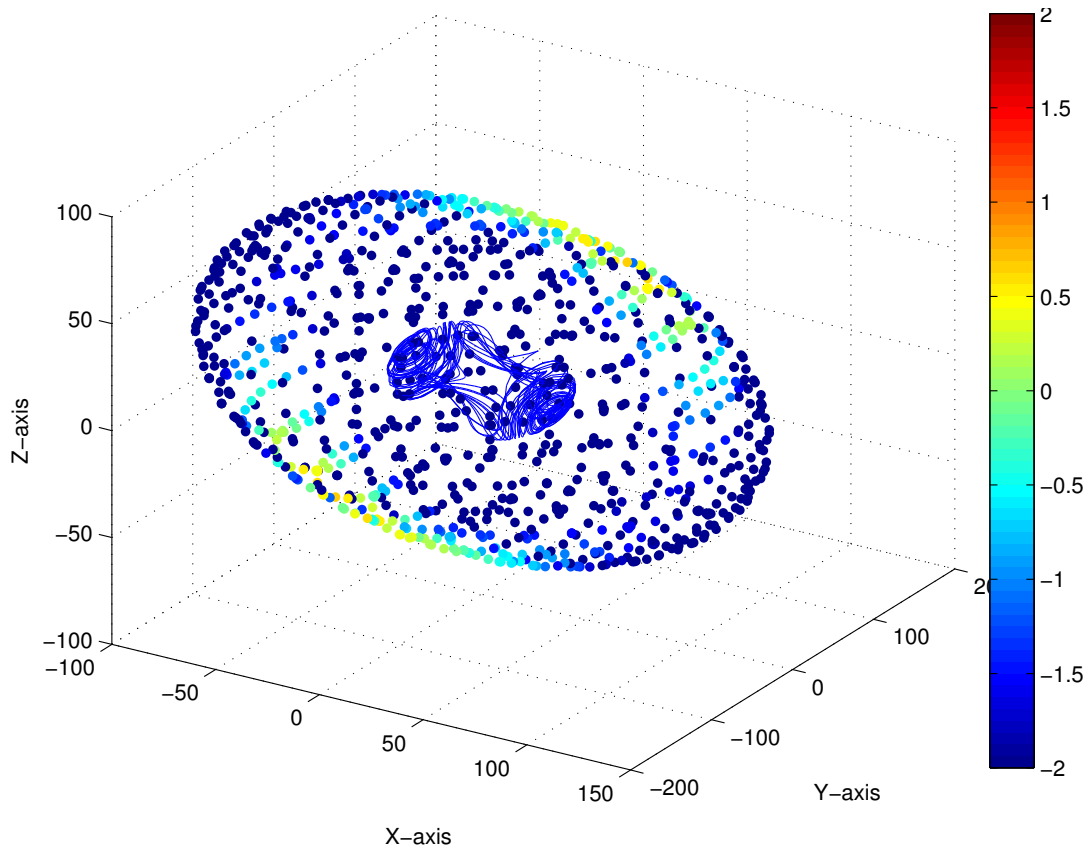


Figure 4.10: Scalar product of **tamed** equation on boundary of scaled ellipsoid, scale 12

Note that the color scale is the same in all of the four plots. The difference between the positive and negative values of the scalar product plot in the tamed and untamed version can be observed.

### 4.2.3 Periodic Orbits

Under certain parameters ( $Re = 13.42$  and others as above) and initial conditions, e.g.  $x_0 = (-5, -5, -6)^T$ , periodic behavior can be seen even for the untamed case of the BDT model. Even for the tamed case, one encounters periodic orbits that are caused by “bad” design of the taming term. The boundary of the fitted ellipsoid  $\mathcal{E}$  for instance coincides with the periodic orbit we encountered for the initial conditions given below

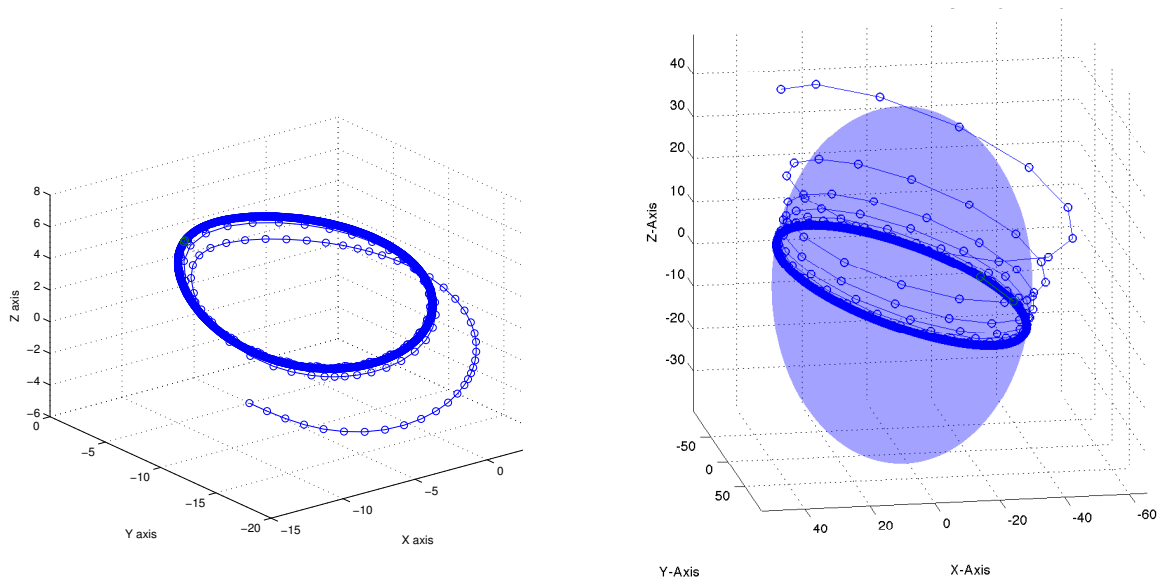


Figure 4.11: Periodic orbits in the tamed and untamed case

For the same initial condition and parameters, the untamed system has no periodic solution.

# Chapter 5

## Taming for Burgers Equation

As noted in the introduction, Burgers equation is highly sensitive to Neumann boundary conditions and standard numerical technique can produce incorrect behavior of the solutions due to finite precision arithmetic. In this section we investigate the use of taming as a mechanism to deal with these numerical issues. In particular, we apply taming to help ensure the correct long time behavior is captured by a given numerical method. The idea (as for Navier-Stokes equations) is to produce the correct asymptotic behavior without impacting the solution near the global (or local) attractor.

### 5.1 Burgers' Equation

In this section we consider Burgers equation

$$\partial_t u(t, x) + u(t, x) \partial_x u(t, x) = \nu \partial_x^2 u(t, x) \quad (5.1)$$

with Neumann-Dirichlet mixed boundary conditions

$$u_x(t, 0) = 0, \quad u(t, 1) = 0 \quad (5.2)$$

and initial condition

$$u(0, x) = u_0(x) \in L^2(0, 1). \quad (5.3)$$

The Burgers system (5.1)-(5.3) has been the subject of several papers and much is known about the behavior of this system. The following result may be found in [ABG12], [EJG13], [BGS98] and [CT02].

**Theorem 5.1.** *The Burgers system (5.1)-(5.3) generates a nonlinear dynamical system  $\{S(t), t \geq 0\}$  on  $L^2(0, 1)$  and the following results hold:*



(a) For every initial data  $u_0(\cdot) \in L^2(0, 1)$  and  $t > 0$ ,  $u(t, x) = [S(t)u_0(\cdot)](x)$  is a smooth classical solution of (5.1)-(5.3).

(b) The dynamical system  $S(t)$  is Lyapunov stable. In particular, there exists a positive continuous monotone increasing function  $a(\xi)$ ,  $\xi \geq 0$  such that  $a(0) = 0$ , and for all  $t \geq 0$  and  $\varphi(\cdot) \in L^2(0, 1)$

$$\|S(t)\varphi(\cdot)\|_{L^2} \leq a(\|\varphi(\cdot)\|). \quad (5.4)$$

(c) The only equilibrium for this system is  $u(\cdot) = 0$  which is globally asymptotically stable in  $H^1(0, 1)$ . Moreover, if  $u_0(\cdot) \in L^2(0, 1)$ , then

$$\lim_{t \rightarrow +\infty} \left\{ \sup_{0 \leq x \leq 1} |u(t, x)| \right\} = 0.$$

Theorem 5.1 above implies that the dynamical system  $S(t)$  generated by the boundary value problem (5.1)-(5.3) is a dissipative dynamical system with global attractor  $M = 0$ . In particular we have the following result.

**Corollary 5.2.** *If  $u_0(\cdot) \in L^2(0, 1)$ , then the solution  $u(t, x)$  to Burgers equation (5.1) with boundary condition (5.2) and initial condition (5.3) satisfies*

$$\lim_{t \rightarrow +\infty} \|u(t, \cdot)\|_{L^2} = 0.$$

Thus, in theory all solutions converge in  $L^2(0, 1)$  asymptotically to zero. Moreover, if  $u_0(\cdot) \in H^1(0, 1)$  the convergence is uniform in space. We shall see below that this long time behavior is not preserved for many standard numerical approximation schemes.

## 5.2 Numerical Approximations

Theorem 5.1 implies that the long term behavior of solutions to the Burgers' equation with the Neumann-Dirichlet boundary conditions (5.2) must converge to zero. However, as noted in [EJG13] most numerical methods will produce erroneous steady state solutions because of the combination of finite precision arithmetic and the sensitivity of Burgers equation due to the Neumann boundary condition. therein for the following lemma. We shall focus on finite element methods to illustrate these issues. A brief introduction to the finite element method can be found in [Joh87] and [Lay08]. However, we review the basic procedure below.

### 5.2.1 The Finite Element Method

An implementation of the finite element method usually proceeds along the following broad steps.

- Discretize the space region into a finite set of nodes, edges and elements;
- Chose *shape functions* or *interpolation functions* defined on each element;
- Formulate a weak version of the partial differential equations;
- Project the weak form of the equations onto the space spanned by the shape functions;
- Solve the projected equations for the coefficient functions of the shape functions.

We shall work through this process for Burgers equation using continuous piecewise linear shape functions. First, the interval  $[0, 1]$  is divided into  $N$  subintervals  $0 = x_0 < x_1 < \dots < x_N < x_{N+1} = 1$  of length  $h$ , so  $h = \frac{1}{N+1}$ . The basis for the shape functions are the typical “hat” functions  $\phi_j(\cdot)$  defined by:

$$\begin{aligned} \phi_0(x) &= \begin{cases} -\frac{1}{h}(x - x_1) & \text{if } x_0 \leq x \leq x_1 \\ 0 & \text{otherwise} \end{cases}, \\ \phi_j(x) &= \begin{cases} \frac{1}{h}(x - x_{j-1}) & \text{if } x_{j-1} \leq x \leq x_j \\ -\frac{1}{h}(x - x_{j+1}) & \text{if } x_j \leq x \leq x_{j+1}, \\ 0 & \text{otherwise} \end{cases}, \quad j = 1, \dots, N, \\ \phi_{N+1}(x) &= \begin{cases} \frac{1}{h}(x - x_N) & \text{if } x_N \leq x \leq x_{N+1} \\ 0 & \text{otherwise} \end{cases}. \end{aligned}$$

Note that the basis functions are continuous, linear in each subinterval  $[x_j, x_{j+1}]$  and satisfy  $\phi_j(x_i) = \delta_{i,j}$ . Figure 5.1 below shows typical hat functions for  $N = 20$ .

The following integrals appear in the calculations below and are a result of straight forward computations with the piecewise defined hat functions  $\phi$ :

$$\int_0^1 \phi_j(x)\phi_i(x)dx = \begin{cases} \frac{2}{3}h & \text{if } j = i \\ \frac{1}{6}h & \text{for } j = i \pm 1 \\ 0 & \text{otherwise} \end{cases} \quad (5.5)$$

$$\int_0^1 \phi_i(x)\phi'_j(x)dx = \begin{cases} 0 & \text{if } j = i \\ -\frac{1}{2}h & \text{for } j = i - 1 \\ \frac{1}{2}h & \text{for } j = i + 1 \\ 0 & \text{otherwise} \end{cases} \quad (5.6)$$

$$\int_0^1 \phi'_j(x)\phi'_i(x)dx = \begin{cases} 2h & \text{if } j = i \\ -h & \text{for } j = i \pm 1 \\ 0 & \text{otherwise} \end{cases} \quad (5.7)$$

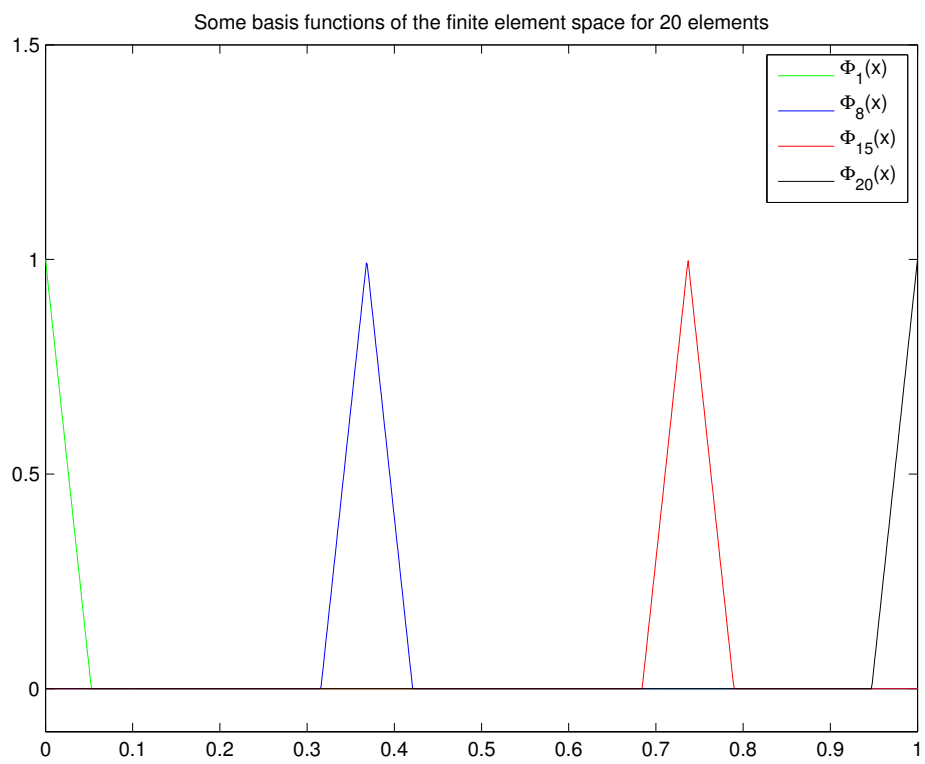


Figure 5.1: FiniteElement Nodes, N=20

Recall that the integration actually takes place over the interval  $[x_{i-1}, x_{i+1}]$ , where again  $h = 1/(N + 1)$ .

## 5.2.2 Finite Elements for Burgers Equation

Although a finite element approach to Burgers equation can be found thoroughly in literature, the individual steps are carried out in order to compare them to the tamed Burgers equation. We now formulate the Burgers equation in weak form. First, (5.1) is multiplied by a test function  $\eta(\cdot) \in H^1(0, 1)$  and integrated over  $(0, 1)$ . This produces the equation

$$\int_0^1 u_t(t, x)\eta(x)dx + \frac{1}{2} \int_0^1 \frac{\partial}{\partial x}[u(t, x)]^2\eta(x)dx = \int_0^1 \nu u_{xx}(t, x)\eta(x)dx. \quad (5.8)$$

The right hand side containing the term  $u_{xx}(t, x)$  is integrated by parts to yield

$$\nu \int_0^1 u_{xx}(t, x)\eta(x)dx = \nu [u_x(t, x)\eta(x)]_0^1 - \nu \int_0^1 u_x(t, x)\eta_x(x)dx \quad (5.9)$$

and hence one has

$$\int_0^1 u_t(t, x)\eta(x)dx + \frac{1}{2} \int_0^1 \frac{\partial}{\partial x}[u(t, x)]^2\eta(x)dx = \nu [u_x(t, x)\eta(x)]_0^1 - \nu \int_0^1 u_x(t, x)\eta_x(x)dx \quad (5.10)$$

holds for all  $\eta(\cdot) \in H^1(0, 1)$ .

Now if one approximates  $u(t, x)$  by the piecewise linear function

$$u(t, x) \approx u^N(t, x) = \sum_{i=0}^{N+1} \alpha_i(t)\phi_i(x), \quad (5.11)$$

then the Dirichlet boundary condition on the right given by  $u(t, 1) = u^N(t, 1) = 0$  implies that  $\alpha_{N+1}(t)\phi_{N+1}(1) = 0$ . Thus  $\alpha_{N+1}(t) \equiv 0$  and one only needs to approximate  $u(t, x)$  by

$$u(t, x) \approx u^N(t, x) = \sum_{i=0}^N \alpha_i(t)\phi_i(x). \quad (5.12)$$

Substituting (5.12) into (5.10) one has

$$\int_0^1 \partial_t u^N(t, x)\eta(x)dx + \frac{1}{2} \int_0^1 \frac{\partial}{\partial x}[u^N(t, x)]^2\eta(x)dx = \nu [\partial_x u^N(t, x)\eta(x)]_0^1 - \nu \int_0^1 \partial_x u^N(t, x)\eta_x(x)dx, \quad (5.13)$$

or equivalently, that

$$\begin{aligned} \int_0^1 \left( \sum_{i=0}^N \dot{\alpha}_i(t)\phi_i(x)\eta(x) \right) dx &= -\frac{1}{2} \int_0^1 \left( \sum_{i=0}^N \frac{\partial}{\partial x} [\alpha_i(t)\phi_i(x)]^2 \eta(x) \right) dx \\ &\quad - \nu \int_0^1 \left( \sum_{i=0}^N \alpha_i(t)\partial_x \phi_i(x)\partial_x \eta(x) \right) dx \\ &\quad - \nu \partial_x \phi_0(0)\eta(0) \end{aligned} \quad (5.14)$$

which must hold for any  $\eta(\cdot) \in H^1(0,1)$ . Observe that the value of  $\eta(\cdot)$  at  $x = 1$  does not come into equation (5.14) so that (5.14) need only hold for all  $\eta(\cdot) \in H_R^1(0,1) = \{\eta(\cdot) \in H^1(0,1) : \eta(1) = 0\}$ .

The important step is to substitute the basis functions  $\phi_j(x)$ ,  $j = 0, 1, \dots, N$  for  $\eta(\cdot)$  into (5.14) which produces the system of  $N + 1$  equations

$$\begin{aligned} \int_0^1 \left( \sum_{i=0}^N \dot{\alpha}_i(t) \phi_i(x) \phi_j(x) \right) dx &= -\frac{1}{2} \int_0^1 \left( \sum_{i=0}^N \frac{\partial}{\partial x} [\alpha_i(t) \phi_i(x)]^2 \phi_j(x) \right) dx \\ &\quad - v \int_0^1 \left( \sum_{i=0}^N \alpha_i(t) \partial_x \phi_i(x) \partial_x \phi_j(x) \right) dx \\ &\quad - v \partial_x \phi_0(0) \phi_j(0). \end{aligned} \quad (5.15)$$

Observe that the last term in this equation is always zero unless  $j = 0$  and in this case  $\phi_j(0) = 1$  while  $\partial_x \phi_0(0) = -\frac{1}{h} = -(N + 1)$ .

Let  $\boldsymbol{\alpha}(t) = [\alpha_0(t), \alpha_1(t), \alpha_2(t), \dots, \alpha_N(t)]^T$  denote the vector of coefficients, then the system of equations for  $\boldsymbol{\alpha}(t)$  becomes

$$M \dot{\boldsymbol{\alpha}}(t) = K \boldsymbol{\alpha}(t) + \tilde{F}(\boldsymbol{\alpha}(t)). \quad (5.16)$$

where

$$\begin{aligned} M &= \frac{h}{6} \begin{pmatrix} 2 & 1 & & & & \\ 1 & 4 & 1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & 1 & 4 & 1 & \\ & & & 1 & 4 & \end{pmatrix} \in \mathbb{R}^{[N+1] \times [N+1]}, \\ K &= \frac{1}{h} \begin{pmatrix} 1 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & -1 & 2 & -1 & \\ & & & -1 & 2 & \end{pmatrix} \in \mathbb{R}^{(N+1) \times (N+1)}, \end{aligned}$$

and the nonlinear term  $F$  is defined by

$$\tilde{F}(\boldsymbol{\alpha}) = \frac{-1}{6} \begin{bmatrix} -2[\alpha_0]^2 + \alpha_0 \alpha_1 + [\alpha_1]^2 \\ -[\alpha_0]^2 - \alpha_0 \alpha_1 + \alpha_1 \alpha_2 + [\alpha_2]^2 \\ -[\alpha_1]^2 - \alpha_1 \alpha_2 + \alpha_2 \alpha_3 + [\alpha_3]^2 \\ \vdots \\ -[\alpha_{N-2}]^2 - \alpha_{N-2} \alpha_{N-1} + \alpha_{N-1} \alpha_N + [\alpha_N]^2 \\ -[\alpha_{N-1}]^2 - \alpha_{N-1} \alpha_N \end{bmatrix}$$

Note that the standard finite element scheme generates a complex nonlinearity. The group finite element method (see [Fle82]) provides an alternate approximation to reduce this complexity. In particular, one assumes the approximation

$$u^2(t, x) \approx [u^N(t, x)]^2 \approx \sum_{j=0}^N \alpha_j^2(t) \phi_j(x)$$

and substitute this expression into (5.13). The system (5.15) now becomes

$$\begin{aligned} \int_0^1 \left( \sum_{i=0}^N \dot{\alpha}_i(t) \phi_i(x) \phi_j(x) \right) dx &= -\frac{1}{2} \int_0^1 \left( \sum_{i=0}^N [\alpha_i(t)]^2 \partial_x \phi_i(x) \phi_j(x) \right) dx \\ &\quad - v \int_0^1 \left( \sum_{i=0}^N \alpha_i(t) \partial_x \phi_i(x) \partial_x \phi_j(x) \right) dx \\ &\quad - v \partial_x \phi_0(0) \eta(0) \end{aligned} \quad (5.17)$$

which can be written as the nonlinear system

$$M \dot{\boldsymbol{\alpha}}(t) = K \boldsymbol{\alpha}(t) + \tilde{F}_G(\boldsymbol{\alpha}(t)), \quad (5.18)$$

where the nonlinear term has the form

$$\tilde{F}_G(\boldsymbol{\alpha}) = \frac{1}{4} \begin{bmatrix} [\alpha_0]^2 - [\alpha_1]^2 \\ [\alpha_1]^2 - [\alpha_3]^2 \\ [\alpha_2]^2 - [\alpha_4]^2 \\ \vdots \\ [\alpha_{N-2}]^2 - [\alpha_N]^2 \\ + [\alpha_{N-1}]^2 \end{bmatrix}. \quad (5.19)$$

For the remainder of this thesis we focus on the group finite element model (5.18). Observe that the nonlinear term  $\tilde{F}_G(\boldsymbol{\alpha})$  can be written as

$$\tilde{F}_G(\boldsymbol{\alpha}) = \frac{1}{4} \begin{pmatrix} 1 & -1 & & & & \\ 1 & 0 & -1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & & 1 & 0 & -1 \\ & & & & 1 & 0 \end{pmatrix} \begin{bmatrix} [\alpha_0]^2 \\ [\alpha_1]^2 \\ \vdots \\ [\alpha_{N-1}]^2 \\ [\alpha_N]^2 \end{bmatrix}$$

which is useful in computations. Also, the initial conditions need to be projected onto the finite element space. Therefore, one has

$$u(0, x) = u_0(x) \approx u_0^N(x) = \sum_{j=0}^N \alpha_j(0) \phi_j(x).$$

Note that  $M, K$  and  $B$  are time independent and can be computed offline and only need to be assembled once.

### 5.3 Finite Elements on Burgers equation *with* Taming

To study the tamed case, we consider the following tamed version of Burgers' equation:

$$\partial_t u(t, x) + u(t, x) \partial_x u(t, x) = \nu \partial_{xx}^2 u(t, x) - T(|u(t, x)|) u(t, x) \quad (5.20)$$

With

$$u_x(t, 1) = 0, u(t, 0) = 0$$

As well as initial conditions

$$u(0, x) = u_0(x) \in L^2(0, 1).$$

The taming term  $T(|u(t, x)|^2)u(t, x)$  is of cubic order, i.e.  $T$  qualitatively looks like

$$T : \begin{cases} T(r) = 0 & \text{for } r \leq N \\ T(r) = \frac{r-N-\frac{1}{2}}{\nu} & \text{for } r \geq N + 1, \\ 0 \leq T'(r) \leq C_\nu & \text{for } r \geq 0, \\ |T^{(k)}(r)| \leq C_{\nu,k} & \text{for } r \geq 0, k \in \mathbb{N} \end{cases} \quad (5.21)$$

for a fixed  $N$  depending only on the equation itself and constants  $C_\nu, C_{\nu,k}$  only depending on the respective subscripts.  $N$  compares to the “valve” parameter Roeckner & Co. use in [RZ09]. The requirements  $0 \leq T'(r) \leq C_\nu$  for  $r \geq 0$  and  $|T^{(k)}(r)| \leq C_{\nu,k}$  for  $r \geq 0, k \in \mathbb{N}$  ensure smoothness of the taming function as well as boundedness.

The finite element approximation is applied to the tamed Burgers' equation (5.20) following the same procedure as presented above. First, the equation is multiplied with a test function  $\eta$  and integrated over  $(0, 1)$ , which yields

$$\begin{aligned} \int_0^1 \eta(x) \partial_t u(t, x) dx + \frac{1}{2} \int_0^1 \eta(x) \partial_x [u(t, x)]^2 dx \\ = \nu \int_0^1 \eta(x) \partial_{xx}^2 u(t, x) dx - \int_0^1 \eta(x) T(|u(t, x)|^2) u(t, x) dx \end{aligned} \quad (5.22)$$

Using the same partial integration on the convective term as above gives

$$\nu \int_0^1 \eta(x) u_{xx}(t, x) dx = \nu [u_x(t, x) \eta(x)]_0^1 - \nu \int_0^1 \eta_x(x) u_x(t, x) dx, \quad (5.23)$$

and hence

$$\begin{aligned} \int_0^1 \eta(x) \partial_t u(t, x) dx + \frac{1}{2} \int_0^1 \eta(x) \partial_x [u(t, x)]^2 dx \\ = \nu [u_x(t, x) \eta(x)]_0^1 - \nu \int_0^1 \eta_x(x) u_x(t, x) dx - \int_0^1 \eta(x) T(|u(t, x)|^2) u(t, x) dx \end{aligned} \quad (5.24)$$

which holds for all  $\eta \in H^1(0,1)$ . Again approximating  $u(t, x)$  by  $u^N(t, x) = \sum_{i=0}^N \alpha_i(t)\phi_i(x)$  yields

$$\begin{aligned} & \int_0^1 \left( \sum_{j=0}^N \alpha_j(t)\phi_j(x) \right) \eta(x) + \frac{1}{2} \frac{\partial}{\partial x} \left[ \sum_{j=0}^N \alpha_j(t)\phi_j(x) \right]^2 \eta(x) dx \\ &= \nu \left[ \left( \sum_{j=0}^N \alpha_j(t)\phi_j(x) \right) \eta(x) \right]_0^1 + \nu \int_0^1 \left( \sum_{j=0}^N \alpha_j(t)\partial_x \phi_j(x) \right) \eta(x) dx \\ & \quad - \int_0^1 T(|u(t, x)|^2) \left( \sum_{j=0}^N \alpha_j(t)\phi_j(x) \right) \eta(x) dx \quad (5.25) \end{aligned}$$

The taming function is now projected onto the finite element space. This is represented by

$$T(r) = \sum_{j=0}^N \alpha_j(t)\phi_j(x),$$

for some coefficient functions  $\alpha_j$ . For the simplest version of the taming function,  $T(r) = \epsilon \cdot r$ , for some  $\epsilon > 0$ , and the group finite element approximation, this results in

$$T(|u|^2) = \epsilon \cdot |u|^2 = \epsilon \sum_{j=0}^N \alpha_j^2(t)\phi_j(x). \quad (5.26)$$

Substituting (5.26) into (5.25) and renaming one summation index to  $k$  to provide clarity yields

$$\begin{aligned} & \int_0^1 \left( \sum_{j=0}^N \alpha_j(t)\phi_j(x) \right) \eta(x) + \frac{1}{2} \frac{\partial}{\partial x} \left[ \sum_{j=0}^N \alpha_j(t)\phi_j(x) \right]^2 \eta(x) dx - \\ &= \nu \int_0^1 \left( \sum_{j=0}^N \alpha_j(t)\partial_x \phi_j(x) \right) \eta(x) dx + \nu \left[ \left( \sum_{j=0}^N \alpha_j(t)\phi_j(x) \right) \eta(x) \right]_0^1 \\ & \quad - \int_0^1 \epsilon \left( \sum_{k=0}^N \alpha_k^2(t)\phi_k(x) \right) \left( \sum_{j=0}^N \alpha_j(t)\phi_j(x) \right) \eta(x) dx \quad (5.27) \end{aligned}$$

We consider the product of sums inside there separately below. Also a specific test function  $\eta(x) = \phi_i(x)$  is used. Fixing  $i$  and let  $j$  and  $k$  vary gives

$$\begin{aligned} & \epsilon \int_0^1 \left( \sum_{k=0}^N \alpha_k^2(t)\phi_k(x) \right) \left( \sum_{j=0}^N \alpha_j(t)\phi_j(x) \right) \phi_i(x) dx = \epsilon \sum_{j,k=0}^N \alpha_k^2(t)\alpha_j(t) \int_0^1 \phi_k(x)\phi_j(x)\phi_i(x) dx \\ &= \epsilon \cdot \frac{1}{12} h \cdot \left( \alpha_{i-1}^3 + 6\alpha_i^3 + \alpha_{i+1}^3 + \alpha_i\alpha_{i-1}^2 + \alpha_i^2\alpha_{i-1} + \alpha_i^2\alpha_{i+1} + \alpha_i\alpha_{i+1}^2 \right). \end{aligned}$$

Here, the integral  $\int_0^1 \phi_k\phi_j\phi_i dx$  is computed explicitly by considering case scenarios.



Let again  $\boldsymbol{\alpha}(t) = [\alpha_1(t), \dots, \alpha_N(t)]$  the vector of coefficients, then the system can be written in matrix form as

$$M\dot{\boldsymbol{\alpha}}(t) = K\boldsymbol{\alpha}(t) + \tilde{F}(\boldsymbol{\alpha}(t)) - T\boldsymbol{\alpha}(t)^T[\boldsymbol{\alpha}]^2(t) \quad (5.28)$$

with  $K$  and  $\tilde{F}$  as in (5.16) and  $[\boldsymbol{\alpha}]^2(t)$  denotes the vector  $[\alpha_1^2(t), \alpha_2^2(t), \dots, \alpha_N^2(t)]^T$ . The matrix  $T$  then reads like

$$T\boldsymbol{\alpha}(t)^T[\boldsymbol{\alpha}]^2(t) = \begin{pmatrix} -\frac{1}{2} & \frac{1}{2} & & & & \\ -\frac{1}{2} & 0 & \frac{1}{2} & & & \\ & \ddots & \ddots & \ddots & & \\ & & -\frac{1}{2} & 0 & \frac{1}{2} & \\ & & & -\frac{1}{2} & 0 & \end{pmatrix} \begin{bmatrix} \alpha_1^3 & \alpha_1^2\alpha_2 & \dots & \alpha_1^2\alpha_N \\ \alpha_1\alpha_2^2 & \alpha_2^3 & & \vdots \\ \vdots & & \ddots & \\ \alpha_1\alpha_N^2 & & \dots & \alpha_N^3 \end{bmatrix}. \quad (5.29)$$

## 5.4 Taming on the Finite Element Space

The projected taming term takes a specific structure that can be coded up in Matlab without using the (rather large) Matrix multiplication above. Also note the specific structure involving the  $k-1, k$  and  $k+1$  term in each step. Another approach that turns out to be more effective is applying taming to the already projected equation. This means the finite element representation of Burgers' equation is tamed with a similar structure that a full projection yields. Keeping the 3rd order terms from the projected taming term results in

$$M\dot{\boldsymbol{\alpha}}(t) = K\boldsymbol{\alpha}(t) + \tilde{F}(\boldsymbol{\alpha}(t)) - \epsilon\tilde{T}\boldsymbol{\alpha}(t) \cdot |\boldsymbol{\alpha}(t)|^2 \quad (5.30)$$

Here  $\tilde{T}$  is a tridiagonal matrix of the form

$$\tilde{T} = \begin{pmatrix} 1 & 1 & & & 0 \\ 1 & 1 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 1 & 1 \\ 0 & & & 1 & 1 \end{pmatrix}$$

With the taming term now being

$$\epsilon\tilde{T}[\boldsymbol{\alpha}(t)]^3,$$

of third order as used to tame the BDT model. Here,  $[\boldsymbol{\alpha}(t)]^3$  denotes  $[\alpha_1^3(t), \dots, \alpha_N^3(t)]^T$ .

### Another version of taming on the finite element space

All the above versions of taming make use of the fact that  $\{0\}$  is the global attractor, since taming is applied at any point away from the global equilibrium solution. We want to follow

the initial concept of taming more precisely by using

$$M\dot{\boldsymbol{\alpha}}(t) = K\boldsymbol{\alpha}(t) + \tilde{F}(\boldsymbol{\alpha}(t)) - \bar{g}(\boldsymbol{\alpha}(t)) \cdot \boldsymbol{\alpha}(t) \quad (5.31)$$

for a function  $\bar{g}(\boldsymbol{\alpha}(t))$  with a predefined  $R > 0$  (the region in which no taming is applied). Resulting in

$$\bar{g}(\boldsymbol{\alpha}(t)) = \begin{cases} 0 & \text{for } \|\boldsymbol{\alpha}(t)\|_\infty < R \\ \epsilon(\|\boldsymbol{\alpha}(t)\|_\infty - R)^2 & \text{for } \|\boldsymbol{\alpha}(t)\|_\infty \geq R, \end{cases} \quad (5.32)$$

for some  $\epsilon > 0$ . This is analogous to the taming function used on the BDT model.

### 5.4.1 Numerical simulations

Simulations of the aforementioned taming on Burgers equation and the finite element model are presented in the following. Therefore, at first, a specific initial condition is considered as discussed in [EJG13], namely  $g(x) = 12.5(1 - x^6)$ . Allen et al noted, that the finite element scheme shows incorrect behavior with this initial condition due to finite precision arithmetic. Taming is used to restore the analytical correct behavior.

#### A particular numerical example

Here,  $m = 256$  elements are used as a basis for the finite element space. The simulation is run up to time  $t_{final} = 10$  with Reynolds number  $Re = 50$ . The initial function used is

$$g(x) = 12(1 - x^6),$$

as shown in the plot below.

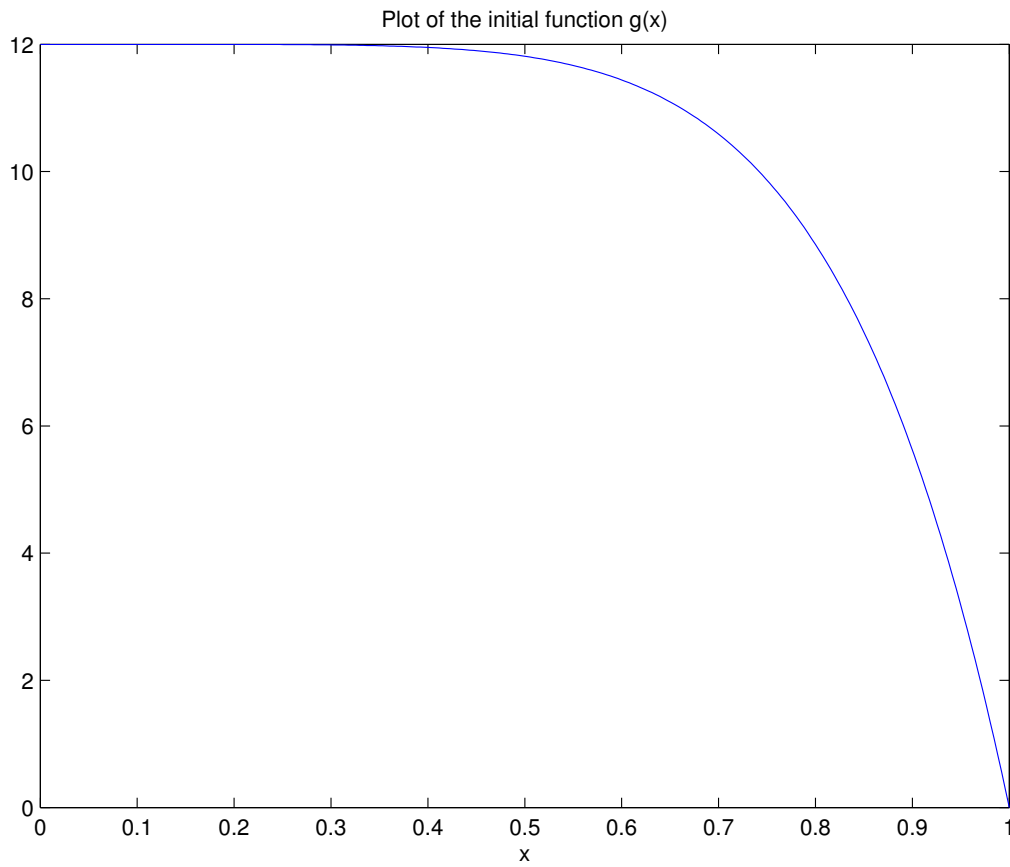


Figure 5.2: Initial function used

As a reference point, we use the Matlab internal PDE solver *pdepe*, which uses an adaptive Petrov-Galerkin projection as discretization method and plot the solution and the final state at time  $t_{final} = 10$ :

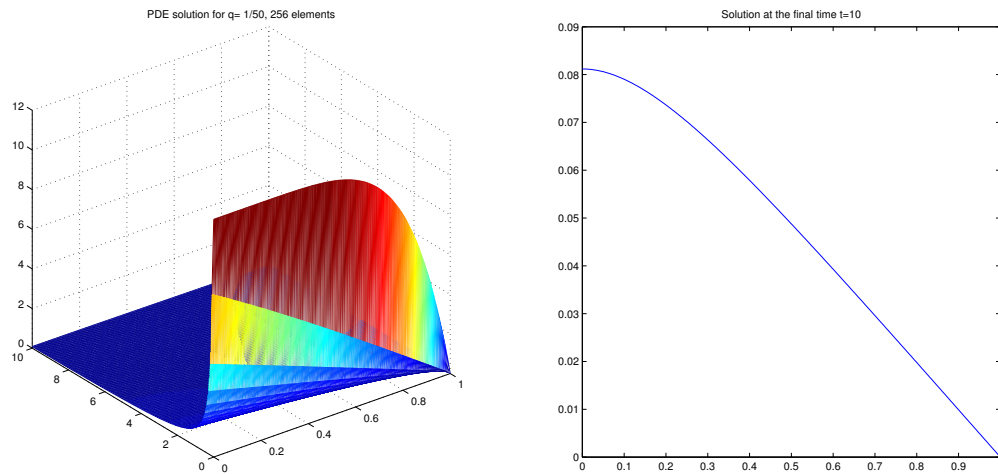


Figure 5.3: Solution to Burgers equation with Matlab *pdepe*

Since the zero function is the only global attractor, the solution to the Burgers' equation with the above initial function has to converge to zero eventually. On the contrary, when finite element discretization is applied to Burgers' equation with this initial function, it does not converge to zero but starts oscillating and tends towards a step function.

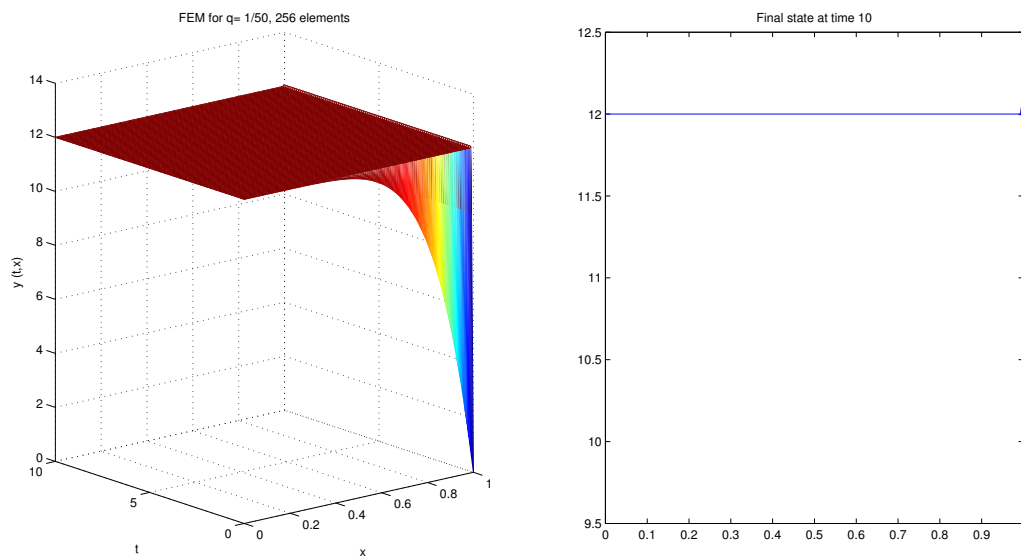


Figure 5.4: Solution to Burgers with FEM, no taming

Adding the taming term as discussed in the previous chapter to the equation, we can restore the analytical correct behavior of the solution as shown here:

Table 5.1: Parameters and setting for FEM simulations

Parameter/Function	Value	Comment
$g(x)$	$1 - x^2$	Initial function
$t_{final}$	10	Final time
$\delta_t$	200	Time steps
N	128	Finite element nodes, order of the ODE
Re	50	Reynolds number
ODE solver	ode23s	

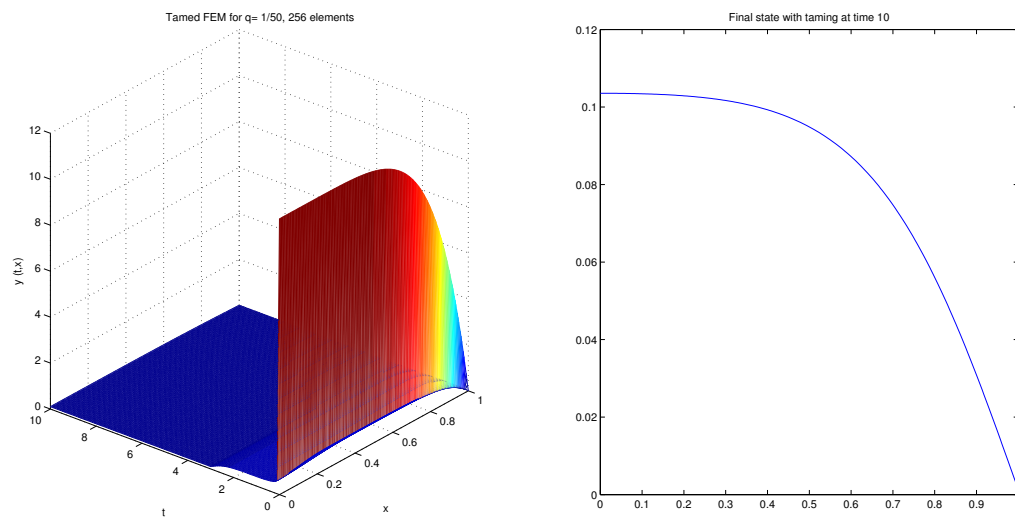


Figure 5.5: Solution to Burgers with FEM, with initial taming using cutoff function

### Different Versions of Taming

Three possible versions of taming are discussed, namely

- First tame Burgers' equation and project onto the finite element space.
- Starting out with the projected FEM on Burgers' equation and apply a 3rd order taming term similarly to what the projected taming looks like but only using 3rd order terms.
- Starting out with the projected FEM and apply a cutoff taming term.

These are the parameters used in the following simulations:

The resulting simulations are shown below.

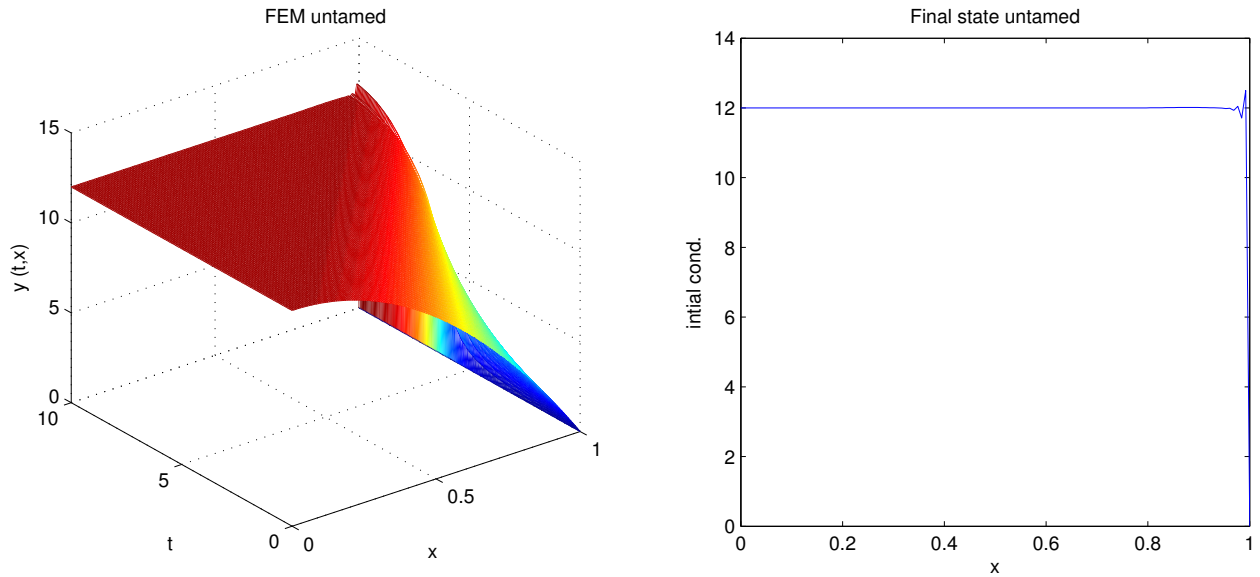


Figure 5.6: Solution to Burgers with FEM, no taming

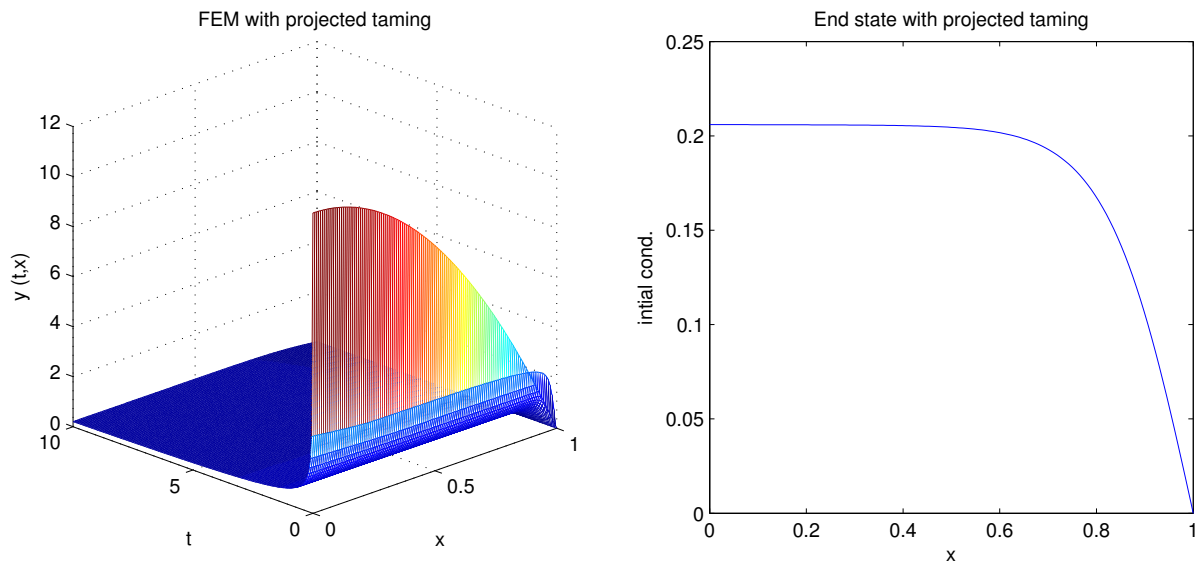


Figure 5.7: Solution to tamed Burgers with FEM (projected taming term)

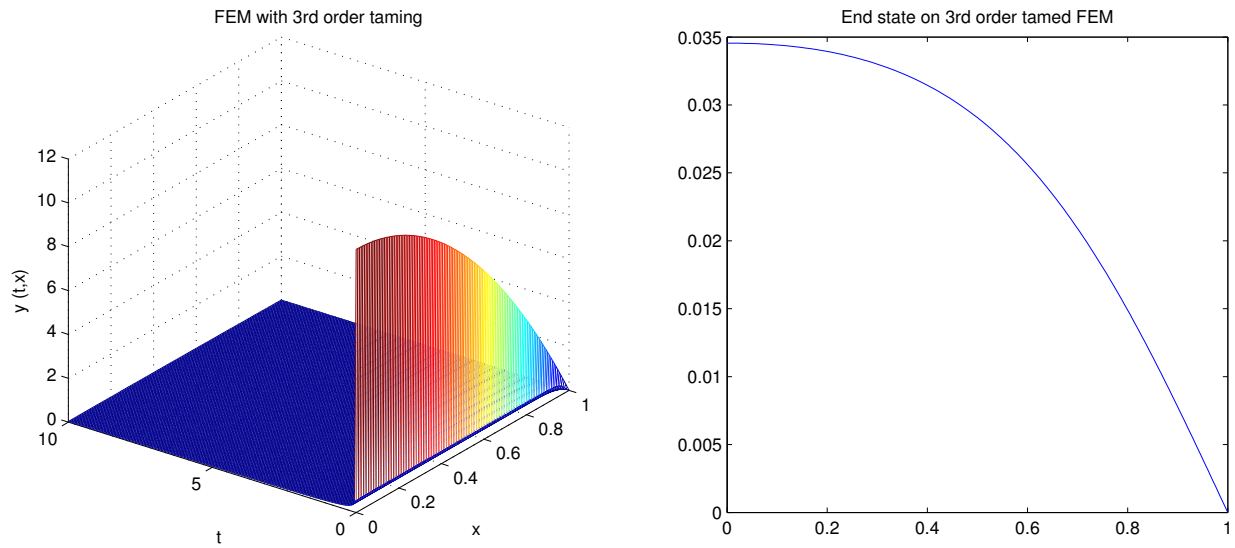


Figure 5.8: Solution to Burgers with 3rd order taming on FEM

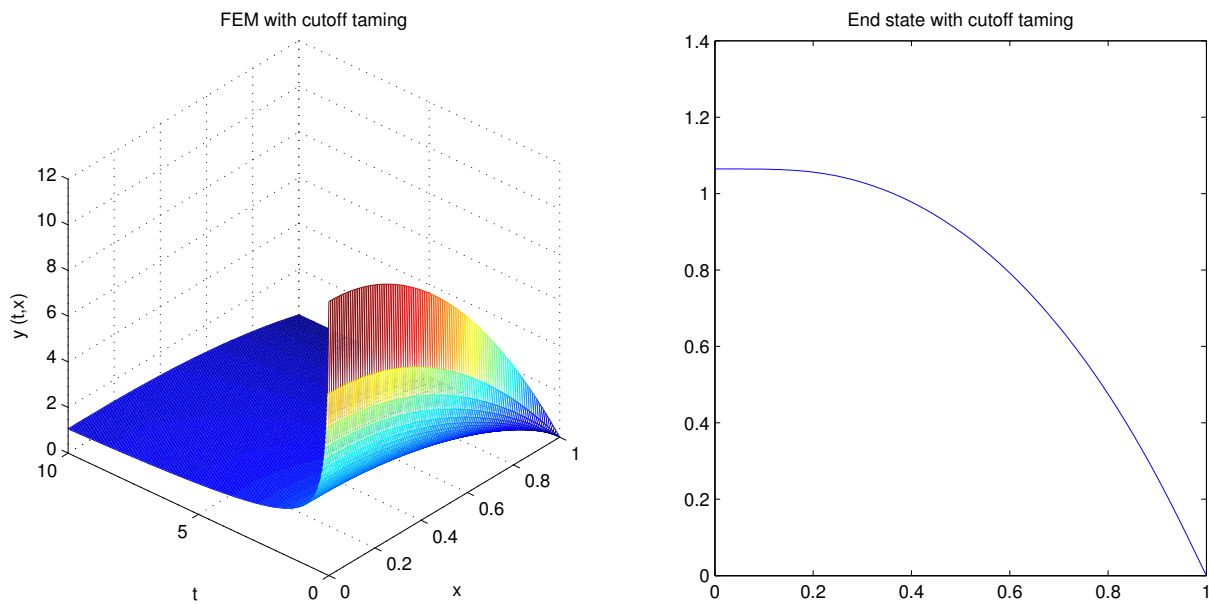


Figure 5.9: Solution to Burgers with tamed FEM (cutoff taming term)

To visualize the error between the actual solution and the numerical approximation, the *pdepe* solution is used as a reference.

Here are some error plots for the tamed scenarios. The error plot for the untamed FEM simulation is omitted, since it shows obvious wrong behavior.

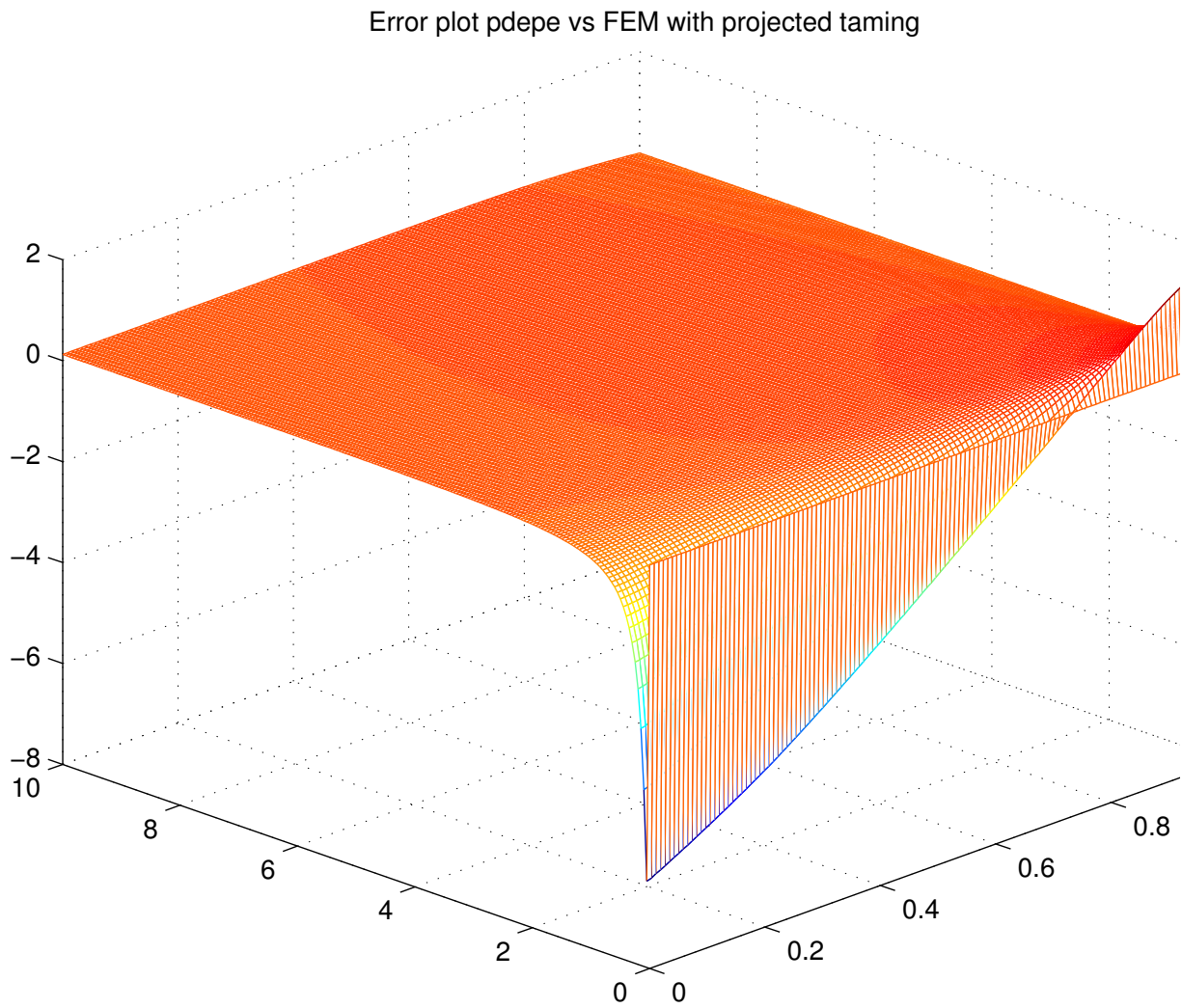


Figure 5.10: Solution to Burgers with tamed FEM (projected taming term)



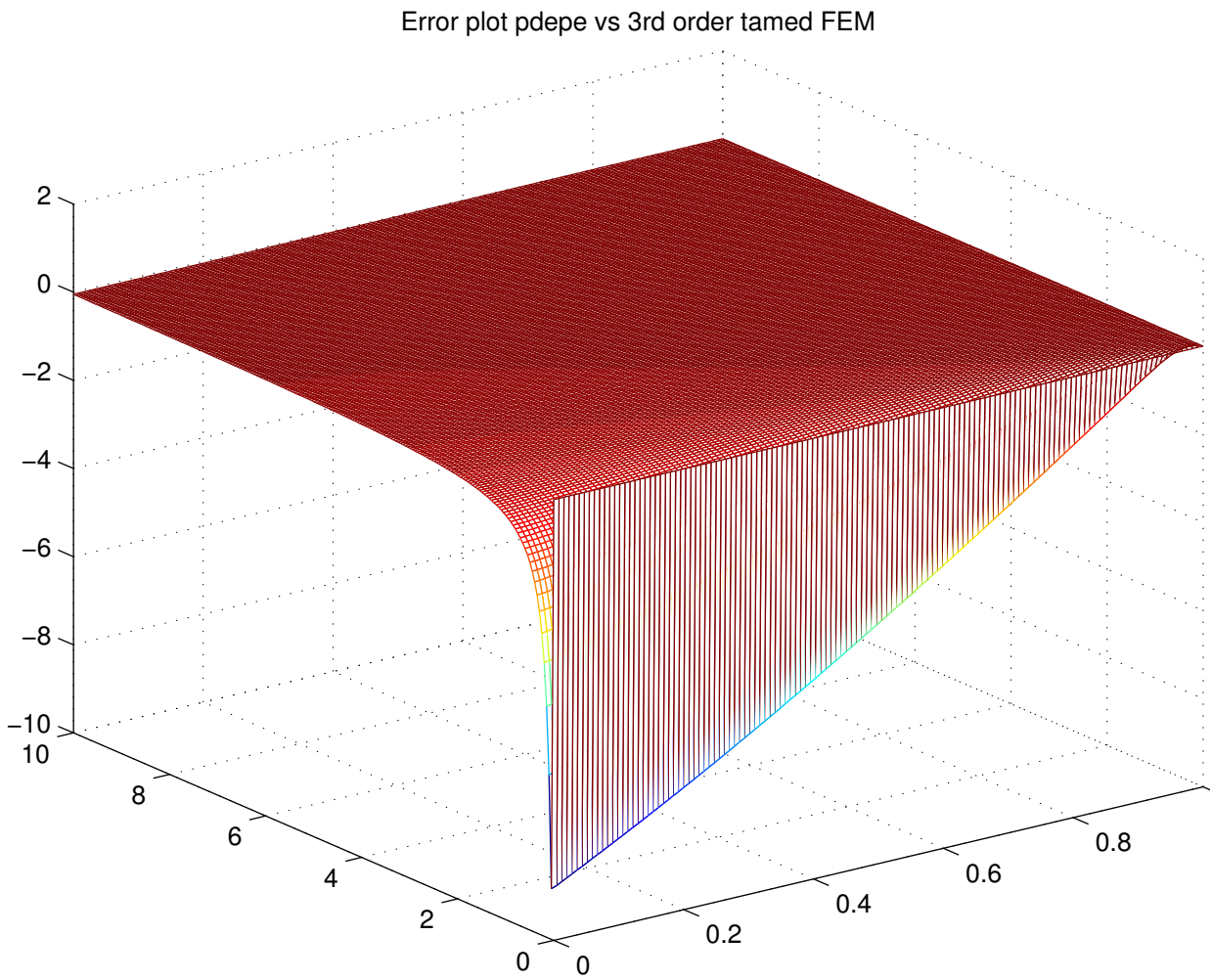


Figure 5.11: Solution to Burgers with tamed FEM (3rd order taming term)

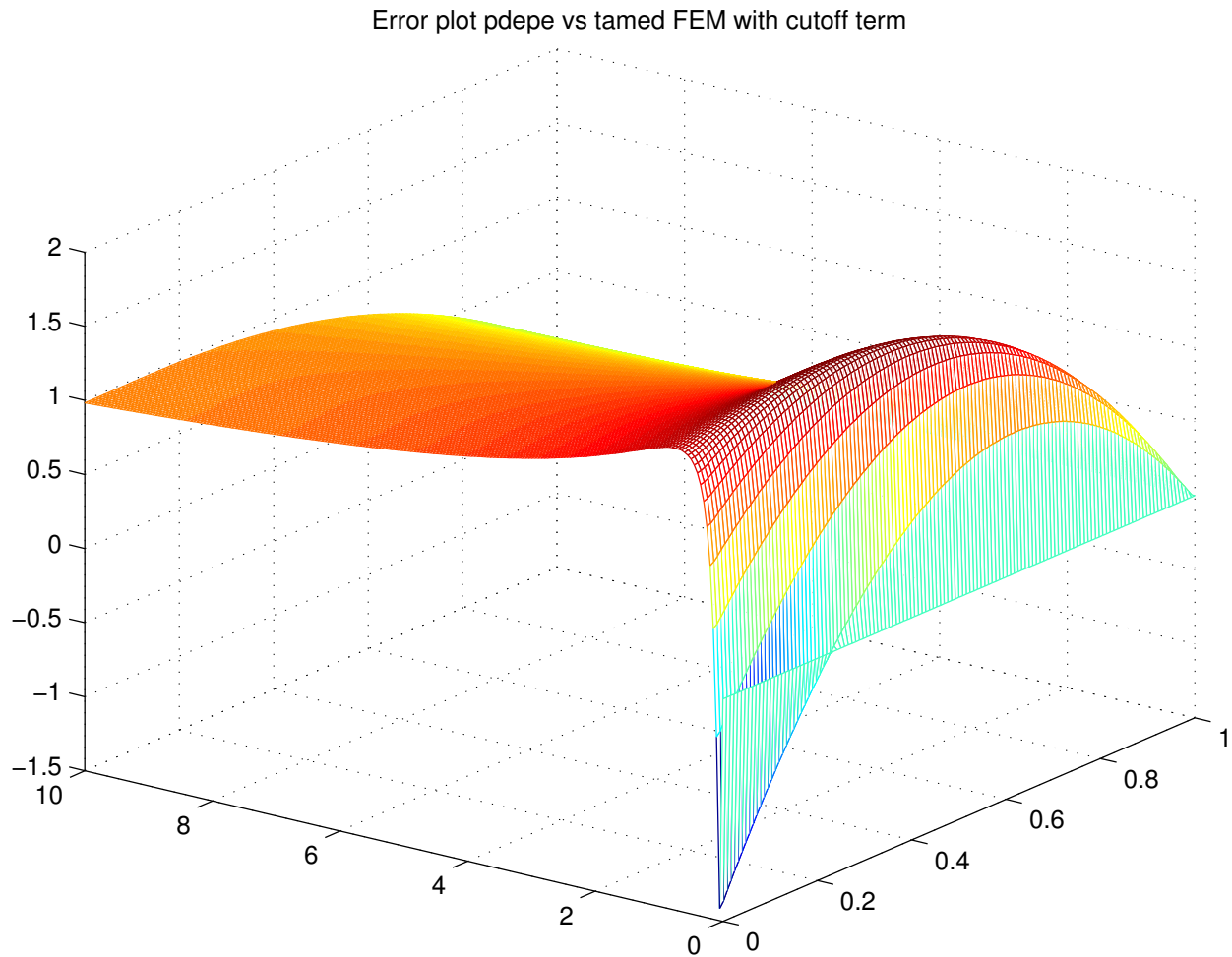


Figure 5.12: Solution to Burgers with tamed FEM (cutoff taming term)

Following the results above, here are the results of the numerical simulations.

Table 5.2: Error of FEM models with taming compared to *pdepe* solution

Taming applied	$t_{final}$	Error in $\ \cdot\ _2$	Error in $\ \cdot\ _\infty$
Projected taming term	10	1.56717	0.164217
3rd order taming on FEM	10	0.302084	0.0463144
Cutoff taming on FEM	10	8.89362	0.985508

From the plots and the errors, the 3rd order taming on the finite element space yields the most accurate results, an order of magnitude better (in the  $\|\cdot\|_\infty$ -norm as well as in the

$\|\cdot\|_2$ -norm).

# Chapter 6

## Conclusions and Future Directions

In this thesis, we applied the concept of taming to several dynamical systems that mimic many of the properties of the Navier-Stokes equations. We first studied the two state BDT model defined by two nonlinear differential equations. We established that the model was dissipative and the impact of taming was minimal. The three state BDT model does not generate a dissipative dynamical system. However, numerical studies indicated the existence of a local chaotic attractor. We proved that by taming these equations one generates a dissipative dynamical system and the local attractor becomes part of the global attractor. However, depending on the specific form of the taming term, additional asymptotic structure could be introduced near the original local chaotic attractor. In particular, one taming term introduced a periodic orbit close to the chaotic attractor. However, by a proper selection of the taming term, these periodic orbits could be eliminated.

We also applied taming to the Burgers equation with Neumann boundary condition at one boundary. The goal here was to address a numerical issue that occurs because of the high sensitivity of the solution of Burgers equation to a Neumann boundary condition. Burgers equation with a Neumann boundary condition at one end and a Dirichlet boundary condition at the other end, generates a dissipative dynamical system with the global attractor defined by the single equilibrium 0. However, as noted in [ABG<sup>+</sup>02] finite precision arithmetic causes most numerical schemes to converge to a false (numerical) equilibrium. We used two approaches based on taming as a mechanism to address this numerical issue. In the first approach, we applied taming directly to the Burgers equation and then constructed a finite element model of the tamed system. The second approach applied taming directly to the finite element model. The second approach is similar to the standard methods that add numerical dissipation to a discretization algorithm. However, the goal is different in that we are interested in computing long time solutions. Both methods eliminated the false numerical steady state solutions.

There are other possible applications for taming that might prove useful and is the subject of ongoing research. One effort is directed at using taming to address the loss of stability when

using model reduction techniques. This work is not complete but early numerical results indicate taming can help in model reduction.

The following issues will be considered in future work:

- Can taming be used to improve model reduction techniques such as proper orthogonal decomposition (POD), rational Krylov algorithms and balanced truncation?
- What is the “optimal form” of the taming term for a given application?
- Can one use taming on other numerical schemes such as finite differences to more accurately capture the asymptotic behavior of dynamical systems?
- In some sense one can view taming as a nonlinear feedback law that takes effect only for “large” states and does not impact the natural local dynamics. It would be interesting to consider taming with in the framework of control.

# Appendix A

## Bibliography

- [ABG<sup>+</sup>02] ALLEN, E. ; BURNS, J. ; GILLIAM, D. ; HILL, J. ; SHUBOV, V.: The impact of finite precision arithmetic and sensitivity on the numerical solution of partial differential equations. In: *Math. Comput. Modelling* 35 (2002), Nr. 11-12, 1165–1195. [http://dx.doi.org/10.1016/S0895-7177\(02\)00078-X](http://dx.doi.org/10.1016/S0895-7177(02)00078-X). – DOI 10.1016/S0895-7177(02)00078-X. – ISSN 0895-7177
- [ABG12] ALLEN, Edward ; BURNS, John A. ; GILLIAM, David S.: Control and sensitivity reduction for a viscous Burgers' equation. In: *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, 2012, S. 967–972
- [Bat67] BATCHELOR, George K.: *An introduction to fluid mechanics*. Cambridge University Press, 1967
- [BGS98] BYRNES, Ch. I. ; GILLIAM, D. S. ; SHUBOV, V. I.: On the global dynamics of a controlled viscous Burgers' equation. In: *J. Dynam. Control Systems* 4 (1998), Nr. 4, 457–519. <http://dx.doi.org/10.1023/A:1021815100868>. – DOI 10.1023/A:1021815100868. – ISSN 1079-2724
- [CF06] CHESKIDOV, Alexey ; FOIAS, Ciprian: On global attractors of the 3D Navier-Stokes equations. In: *Journal of Differential Equations* 231 (2006), Nr. 2, S. 714–754
- [CG09] CHEMIN, Jean-Yves ; GALLAGHER, Isabelle: Wellposedness and stability results for the Navier-Stokes equations in  $\mathcal{R}^3$ . In: *Ann. Inst. H. Poincaré Anal. Non Linéaire* 26 (2009), Nr. 2, 599–624. <http://dx.doi.org/10.1016/j.anihpc.2007.05.008>. – DOI 10.1016/j.anihpc.2007.05.008. – ISSN 0294-1449
- [CG13] CARREÑO, N. ; GUERRERO, S.: Local null controllability of the  $N$ -dimensional Navier-Stokes system with  $N - 1$  scalar controls in an arbitrary control domain. In: *J. Math. Fluid Mech.* 15 (2013), Nr. 1, 139–153. <http://dx.doi.org/10.1016/j.anihpc.2007.05.008>

1007/s00021-012-0093-2. – DOI 10.1007/s00021-012-0093-2. – ISSN 1422-6928

- [CT02] CAO, Chongsheng ; TITI, E. S.: Asymptotic Behavior of Viscous Burgers' Equations with Neumann Boundary Conditions. In: *Private Communication* (2002)
- [Dav04] DAVIDSON, Peter A.: *Turbulence: an introduction for scientists and engineers*. Oxford University Press, USA, 2004
- [EJG13] EDWARD J.ALLEN, John A. B. A. Burns ; GILIAM, David S.: Numerical approximations of the dynamical system generated by Burgers' equation with Neumann-Dirichlet boundary conditions. In: *ESAIM: Mathematical Modelling and Numerical Analysis* eFirst (2013), 7. <http://dx.doi.org/10.1051/m2an/2013084>. – DOI 10.1051/m2an/2013084. – ISSN 1290-3841
- [Fle82] FLETCHER, Clive A. J.: Burgers' equation: a model for all reasons. In: *Numerical solutions of partial differential equations (Parkville, 1981)*. Amsterdam : North-Holland, 1982, S. 139-225
- [Gun89] GUNZBURGER, Max D.: *Finite Element Methods for Viscous Incompressible Flows: A guide to theory, practice, and algorithms*. Bd. 91. 1989. – 30750 S.
- [HS00] HILL, Adrian T. ; SÜLI, Endre: Approximation of the global attractor for the incompressible Navier-Stokes equations. In: *IMA journal of numerical analysis* 20 (2000), Nr. 4, S. 633-667
- [JB01a] JOVANOVIĆ, Mihailo ; BAMIEH, Bassam: Modeling flow statistics using the linearized Navier-Stokes equations. In: *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on* Bd. 5 IEEE, 2001, S. 4944-4949
- [JB01b] JOVANOVIĆ, Mihailo ; BAMIEH, Bassam: The spatio-temporal impulse response of the linearized Navier-Stokes equations. In: *American Control Conference, 2001. Proceedings of the 2001* Bd. 3 IEEE, 2001, S. 1948-1953
- [JH09] JIANG, Jinping ; HOU, Yanren: The global attractor of g-Navier-Stokes equations with linear dampness on  $\mathbb{R}^2$ . In: *Applied Mathematics and Computation* 215 (2009), Nr. 3, S. 1068-1076
- [Joh87] JOHNSON, C.: *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Studentlitteratur, 1987 <http://books.google.com/books?id=gmbSAAAACAAJ>. – ISBN 9789144252414
- [JSBT95] J. S. BAGGETT, T. A. D. ; TREFETHEN, L. N.: A mostly linear model of transition to turbulence. In: *Physics of Fluids* 7 (1995), 833-838. [http://people.maths.ox.ac.uk/trefethen/publication/PDF/1995\\_64.pdf](http://people.maths.ox.ac.uk/trefethen/publication/PDF/1995_64.pdf)

- [JXZ05] JIANG, Song ; XIN, Zhouping ; ZHANG, Ping: Global weak solutions to 1D compressible isentropic Navier-Stokes equations with density-dependent viscosity. In: *Methods and Applications of Analysis* 12 (2005), Nr. 3, S. 239
- [Kac10] KACPRZYK, Piotr: Global attractor for the Navier-Stokes equations in a cylindrical pipe. In: *Ann. Polon. Math.* 97 (2010), Nr. 3, 201–217. <http://dx.doi.org/10.4064/ap97-3-1>. – DOI 10.4064/ap97-3-1. – ISSN 0066–2216
- [Kie07] KIEHN, RM: Turbulence and the Navier-Stokes equations. In: *arXiv preprint arXiv:0704.1596* (2007)
- [KT09] KALANTAROV, Varga K. ; TITI, Edriss S.: Global attractors and determining modes for the 3D Navier-Stokes-Voight equations. In: *Chin. Ann. Math. Ser. B* 30 (2009), Nr. 6, 697–714. <http://dx.doi.org/10.1007/s11401-009-0205-3>. – DOI 10.1007/s11401-009-0205-3. – ISSN 0252–9599
- [KV07] KAPUSTYAN, A. V. ; VALERO, J.: Weak and strong attractors for the 3D Navier-Stokes system. In: *J. Differential Equations* 240 (2007), Nr. 2, 249–278. <http://dx.doi.org/10.1016/j.jde.2007.06.008>. – DOI 10.1016/j.jde.2007.06.008. – ISSN 0022–0396
- [KYO89] KIDA, Shigeo ; YAMADA, Michio ; OHKITANI, Kohji: Route to Chaos in a Navier-Stokes Flow. Version: 1989. [http://dx.doi.org/10.1016/S0304-0208\(08\)70505-X](http://dx.doi.org/10.1016/S0304-0208(08)70505-X). In: MIMURA, Masayasu (Hrsg.) ; NISHIDA, Takaaki (Hrsg.): *Recent Topics in Nonlinear PDE IV* Bd. 160. North-Holland, 1989. – DOI 10.1016/S0304-0208(08)70505-X. – ISSN 0304–0208, 31 - 47
- [Lad75] LADYZHENSKAYA, OA: A dynamical system generated by the Navier-Stokes equations. In: *Journal of Mathematical Sciences* 3 (1975), Nr. 4, S. 458–479
- [Lad87] LADYZHENSKAYA, Olga A.: On the determination of minimal global attractors for the Navier-Stokes and other partial differential equations. In: *Russian Mathematical Surveys* 42 (1987), Nr. 6, S. 27–73
- [Lad98] LADYZHENSKAYA, Olga: Some results on modifications of three-dimensional Navier-Stokes equations. In: *Nonlinear analysis and continuum mechanics (Ferrara, 1992)*. New York : Springer, 1998, S. 73–84
- [Lay08] LAYTON, William: *Introduction to the numerical analysis of incompressible viscous flows*. Bd. 6. Society for Industrial and Applied Mathematics, 2008
- [Li07] LI, Y C.: Chaos in partial differential equations, Navier-Stokes equations and turbulence. In: *ICCM 2007* 3 (2007), S. 110–122



- [LZZ13] LIU, Shengquan ; ZHANG, Jianwen ; ZHAO, Junning: Global classical solutions for 3D compressible Navier-Stokes equations with vacuum and a density-dependent viscosity coefficient. In: *J. Math. Anal. Appl.* 401 (2013), Nr. 2, 795–810. <http://dx.doi.org/10.1016/j.jmaa.2012.12.056>. – DOI 10.1016/j.jmaa.2012.12.056. – ISSN 0022–247X
- [Mas84] MASUDA, Kyūya: Weak solutions of Navier-Stokes equations. In: *Tohoku Mathematical Journal* 36 (1984), Nr. 4, S. 623–646
- [MRR07] MARÍN-RUBIO, Pedro ; REAL, José: Attractors for 2D-Navier-Stokes equations with delays on some unbounded domains. In: *Nonlinear Anal.* 67 (2007), Nr. 10, 2784–2799. <http://dx.doi.org/10.1016/j.na.2006.09.035>. – DOI 10.1016/j.na.2006.09.035. – ISSN 0362–546X
- [NBME33] NEWTON, Isaac ; BERNOULLI, Daniel ; MACLAURIN, Colin ; EULER, Leonhard: *Philosophiae naturalis principia mathematica*. Bd. 1. excudit G. Brookman; impensis TT et J. Tegg, Londini, 1833
- [RKM81] RICHARD K. MILLER, Anthony N. M.: *Ordinary Dynamical Systems*. Courier Dover Publications, 1981 (v. 1). – ISBN 9780486462486
- [Rob04] ROBINSON, R.C.: *An Introduction to Dynamical Systems: Continuous and Discrete*. Pearson Prentice Hall, 2004 <http://books.google.com/books?id=ZGUnAQAAIAAJ>. – ISBN 9780131431409
- [Ros98] ROSA, Ricardo: The global attractor for the 2D Navier-Stokes flow on some unbounded domains. In: *Nonlinear Analysis* 32 (1998), Nr. 1, S. 71–86
- [Rot90] ROTT, N.: Note on the history of the Reynolds number. In: *Annual review of fluid mechanics, Vol. 22*. Palo Alto, CA : Annual Reviews, 1990, S. 1–11
- [RZ09] RÖCKNER, Michael ; ZHANG, Xicheng: Tamed 3D Navier-Stokes equation: existence, uniqueness and regularity. In: *Inf. Dimens. Anal. Quantum Probab. Relat. Top.* 12 (2009), Nr. 4, 525–549. <http://dx.doi.org/10.1142/S0219025709003859>. – DOI 10.1142/S0219025709003859. – ISSN 0219–0257
- [Sel96] SELL, George R.: Global attractors for the three-dimensional Navier-Stokes equations. In: *Journal of Dynamics and Differential Equations* 8 (1996), Nr. 1, S. 1–33
- [SH98] STUART, A. ; HUMPHRIES, A.R.: *Dynamical Systems and Numerical Analysis*. Cambridge University Press, 1998 (Cambridge Monographs on Applied and Computational Mathematics v. 8). <http://books.google.com/books?id=ymoQA8s5pNIC>. – ISBN 9780521645638

- [Som08] SOMMERFELD, Arnold: Ein Beitrag zur hydrodynamischen Erklärung der turbulenten Flüssigkeitsbewegung. In: *Proceedings of the 4th international mathematical congress, Rome 1908* Bd. 4, 1908, S. 116–124
- [TG06] THOMANN, Enrique A. ; GUENTHER, Ronald B.: The Fundamental Solution of the Linearized Navier–Stokes Equations for Spinning Bodies in Three Spatial Dimensions–Time Dependent Case. In: *Journal of Mathematical Fluid Mechanics* 8 (2006), Nr. 1, S. 77–98
- [VC02] VISHIK, Marko I. ; CHEPYZHOV, Vladimir V.: Trajectory and Global Attractors of Three-Dimensional Navier–Stokes Systems. In: *Mathematical Notes* 71 (2002), Nr. 1-2, S. 177–193
- [Wal95] WALEFFE, Fabian: Transition in shear flows. Nonlinear normality versus non-normal linearity. In: *Phys. Fluids* 7 (1995), Nr. 12, 3060–3066. <http://dx.doi.org/10.1063/1.868682>. – DOI 10.1063/1.868682. – ISSN 1070–6631
- [ZD13] ZHAO, CaiDi ; DUAN, JinQiao: Convergence of global attractors of a 2D non-Newtonian system to the global attractor of the 2D Navier-Stokes system. In: *Sci. China Math.* 56 (2013), Nr. 2, 253–265. <http://dx.doi.org/10.1007/s11425-012-4538-0>. – DOI 10.1007/s11425-012-4538-0. – ISSN 1674–7283
- [Zha09] ZHANG, Xicheng: A tamed 3D Navier–Stokes equation in uniform  $C^2$ -domains. In: *Nonlinear Analysis: Theory, Methods & Applications* 71 (2009), Nr. 7, S. 3093–3112

# Appendix B

## Matlab Code

### B.1 The Lorenz system

The following code used to plot the chaotic orbits of the lorenz system.

```
1 %% solving
2 options=odeset('OutputFcn','odephas3');
3 x1=[-1 .1 -0.1];
4 x2=[-1 .1 -0.100001];
5 tspan=[0:0.01:100];
6 [T X1] = ode23(@lorenz,tspan,x1);
7 [T X2] = ode23(@lorenz,tspan,x2);
8
9 %% plotting
10 close all;
11 figure;
12 subplot(1,2,1); hold on;
13 plot3(X1(:,1),X1(:,2),X1(:,3),'r-');
14 plot3(X2(:,1),X2(:,2),X2(:,3),'b-');
15 legend('x0=[-1, 0.1, -0.1]', 'x0=[-1, 0.1, -0.100001]');
16 xlabel('X-axis');ylabel('Y-axis');zlabel('Z-axis'); grid on;
17 view(27,9); hold off; axis equal;
18
19 subplot(1,2,2); hold on;
20 plot3(X1(:,1),X1(:,2),X1(:,3),'r-');
21 plot3(X2(:,1),X2(:,2),X2(:,3),'b-');
22 legend('x0=[-1, 0.1, -0.1]', 'x0=[-1, 0.1, -0.100001]');
23 xlabel('X-axis');ylabel('Y-axis');zlabel('Z-axis'); grid on;
24 view(152,9); hold off; axis equal;
25
26 figure;
27 subplot(1,2,1);
```

```

28 plot(T,X1(:,1));legend('x0=[-1, 0.1, -0.1]');
29 xlabel('time t');ylabel('X-values');
30
31 subplot(1,2,2);
32 plot(T,X1(:,3));legend('x0=[-1, 0.1, -0.1]');
33 xlabel('time t');ylabel('Z-values');
34
35 %% lorenz function
36 function xprime = lorenz(t,x);
37 %Giving back the lorenz equation
38 sig=10;
39 beta=8/3;
40 rho=28;
41 xprime=[-sig*x(1)+sig*x(2) ;
42         rho*x(1)-x(2)-x(1)*x(3) ;
43         -beta*x(3)+x(1)*x(2)];
44 end

```

## B.2 Simulating the BDT model

The following code was used to simulate the BDT-model (with varying initial conditions):

```

1 %% Setting initial conditions and plot orbit of BDT
2 options=odeset('OutputFcn','odephas3');
3 x0=[35, -20, 45];
4 %x1=[-300,-600,1000];
5 %x0=[0.02,-0.01,0.01];
6 %x0=[30,10,35]; %explodes
7 tic;
8 tspan=[0,20];
9 %[T,X] = ode23s(@poorManNS,tspan,x0);
10 %[T,X] = ode23s(@BDTTamedQuadr,tspan,x0);
11 [T,X] = ode23s(@BDTTamedSinus,tspan,x0);
12 toc;
13 %title(sprintf(' Plot of solution with initial data: [%g,%g,%g] ...
14         ',x0(1),x0(2),x0(3)))
14 xlabel('X-axis');ylabel('Y-axis');zlabel('Z-axis');
15
16 %% Plotting normal simulations
17 figure;
18 plot3(X(:,1),X(:,2),X(:,3),'b-'); hold on;
19 plot3(x0(1),x0(2),x0(3),'ro','LineWidth',2); %add initial point in red
20 xlabel('X-axis');ylabel('Y-axis');zlabel('Z-axis'); grid on;
21 title(sprintf(' Plot of solution with initial data: [%g,%g,%g] and ...
22         Re=35 ',x0(1),x0(2),x0(3)));
23 %% Plotting the simulation in 3D

```

```

24 scaleEllipsoid=2;
25 scaleScalarP=20;% 1/scaling size
26
27 maxd = max( radii )*scaleEllipsoid;
28 step = maxd / 50;
29 [ x, y, z ] = meshgrid( -maxd:step:maxd , -maxd:step:maxd , ...
    -maxd:step:maxd );
30 Ellipsoid = v(1) *x.*x + v(2) * y.*y + v(3) * z.*z + 2*v(4) *x.*y + ...
    2*v(5)*x.*z + 2*v(6) * y.*z + 2*v(7) *x    + 2*v(8)*y    + 2*v(9) * z;
31 figure; hold on;
32 p=patch( isosurface( x, y, z, Ellipsoid, scaleEllipsoid ) );
33 set( p, 'FaceColor', 'b', 'EdgeColor', 'none', 'FaceAlpha',0.2);
34 plot3(X(:,1),X(:,2),X(:,3), 'b-'); hold on;
35 plot3(x0(1),x0(2),x0(3), 'ro', 'LineWidth',2); %add initial point in red
36 xlabel('X-axis');ylabel('Y-axis');zlabel('Z-axis'); grid on;
37 title(sprintf(' Plot of solution with initial data: [%g,%g,%g] and ...
    Re=15 ',x0(1),x0(2),x0(3)));
38
39 %% Projected plots
40 normX=[];
41 for i=1:length(X)
42     normX(i)=sqrt(X(i,1)^2+X(i,2)^2+X(i,3)^2);
43 end
44
45 figure;
46 subplot(2,1,1); plot(T,X(:,1), 'b-')
47 xlabel('Time');ylabel('X-value of solution');
48 subplot(2,1,2); plot(T,normX, 'b-')
49 xlabel('Time');ylabel('Norm of solution');

```

These are function files used for the standard ODE solvers (ode23, ode45, etc.)

```

1 function xprime = BDT(t,x)
2 %Computes the right hand side of BDT Model
3 %x=x'; % transposing x for the following matrix functions
4 a=.9;
5 b=.3;
6 c=.2;
7 alpha=1;
8 beta=2.1;
9 gamma=1.2;
10 Re= 35;
11
12 A=[-alpha Re 0; 0 -beta Re; 0 0 -gamma];
13 B=[0 -a -b; a 0 c; b -c 0];
14
15 xprime=A*x+norm(x)*(B*x);
16 end

```

```

1 function xprime = BDTamed(t,x)
2 %Computes the right hand side of BDT system
3 %x=x'; % transposing x for the following matrix functions
4 a=.9;
5 b=.3;
6 c=.2;
7
8 alpha=1;
9 beta=1.1;
10 gamma=1.2;
11
12 %epsilon=0.001;
13
14 Re=35; %reynolds number
15
16 my=10;%slope of the taming function
17 N=50; %setting cutoff value for the taming function, radius from the ...
    origin
18
19 A=[-alpha Re 0; 0 -beta Re; 0 0 -gamma];
20 B=[0 -a -b; a 0 c; b -c 0];
21
22 %cutoff2(N^2,norm(x,2)^2)
23
24 % setting coefficient vector for ellipsoid
25 ellipsoidCoeff ...
    =[0.0013;0.0004;0.0013;-0.0002;-0.0001;0.0001;0.0002;-0.0001;-0.0002];
26
27 %xprime=A*x+norm(x)*(B*x)-epsilon*norm(x)^2*x;
28 %xprime=A*x+norm(x)*(B*x)-cutoff(N^2,norm(x,2)^2,my,ellipsoidCoeff)*x;
29 xprime=A*x+norm(x)*(B*x)-x*cutoffAdjustedSmooth(x,my,ellipsoidCoeff);
30 %xprime=A*x+norm(x)*(B*x)-cutoff3(norm(x))*x;
31
32 end

```

```

1 function xprime = BDTamedQuadr(t,x)
2 % Computes the right hand side of BDT system
3 a=.9;
4 b=.3;
5 c=.2;
6 alpha=1;
7 beta=1.1;
8 gamma=1.2;
9
10 % ----- important parameters -----
11 Re=15; %reynolds number
12 Thresh1=2; % initial threshold for level set
13 my=1/Re;% 1/slope of the taming function outside of a certain radius

```

```

14  % -----
15
16 A=[-alpha Re 0; 0 -beta Re; 0 0 -gamma];
17 B=[0 -a -b; a 0 c; b -c 0];
18
19 % predetermined ellipsoid, fitted to the dumbbell chaotic attractor
20 v=[0.001253052115852;4.303899528094455e-04;0.001303923399195;-2.024089915288169e-04;
21
22 r=norm(x);
23 Taming = 0;
24
25 g=@(x) ...
    v(1)*x(1)^2+v(2)*x(2)^2+v(3)*x(3)^2+2*v(4)*x(1)*x(2)+2*v(5)*x(1)*x(3)+2*v(6)*x(2)*x(
    2*v(9)*x(3);
26 f =@(N1,r) ((r-N1)^2-(1/2))/(2*my) - 1/(4*my); % full taming function
27 syms a;
28
29 if (g(x) < Thresh1)
30     Taming = 0;
31 else
32     S01=max(double(solve((g(a*x)==Thresh1),a)));
33     N1=norm(S01*x); %projection of current trajectory point onto ...
        ellipsoid
34
35     Taming = f(N1,r)-f(N1,N1);
36 end
37
38 %print the taming
39 % if (Taming > 0)
40 %     fprintf('Output vector x: %3.2g,%3.2g,%3.2g and the taming term ...
        %3.3g \n',x(1),x(2),x(3),Taming);
41 % end
42
43 % finally writing together the function
44 xprime=A*x+norm(x)*(B*x)-x*Taming;
45 end

```

## B.3 Fitting the ellipsoid

This code was used to fit the ellipsoid around the chaotic attractor:

```

1 %% solve the PMNS
2 options=odeset('OutputFcn','odephas3');
3 x0=[8 9 10];
4 tspan=[0:0.01:20];
5 [T X] = ode23s('BDT',tspan,x0,options);

```

```

6
7 %% fitting the ellipsoid
8 [ center, radii, evecs, v ] = ellipsoid_fit( X );
9
10 %% preparing
11 alpha1=-1;
12 beta1=-1.1;
13 gamma1=-1.2;
14 Re=35; %reynolds number
15 A=[alpha1 Re 0; 0 beta1 Re; 0 0 gamma1];
16
17 %% display fitted ellipsoid
18 scalesOfEllipsoid=[9];
19 parfor k=1:length(scalesOfEllipsoid)
20     scaleEllipsoid=scalesOfEllipsoid(k);
21     stepsizeQuotient = 35;
22     %scaleScalarP=800;% 1/scaling size
23
24     maxd = max( radii ) * scaleEllipsoid;
25     step = maxd / stepsizeQuotient;
26     [ x, y, z ] = meshgrid( -maxd:step:maxd , -maxd:step:maxd , ...
27         -maxd:step:maxd );
28     tic;
29     % with v=(A,B,C,D,E,F,G,H,I)
30     % Ellipsoid is described by
31     %  $Ax^2 + By^2 + Cz^2 + 2Dxy + 2Exz + 2Fyz + 2Gx + 2Hy + 2Iz = 1$ 
32     Ellipsoid = v(1) * x.*x + v(2) * y.*y + v(3) * z.*z + 2*v(4) * x.*y
33     + 2*v(5) * x.*z + 2*v(6) * y.*z + 2*v(7) * x + 2*v(8) * y + 2*v(9) * z;
34
35     % checking for positive invariance
36     figure; hold on;
37     plot3(X(:,1),X(:,2),X(:,3), 'b-')
38     xlabel('X-axis');ylabel('Y-axis');zlabel('Z-axis'); grid on;
39
40     [f,vertices] = isosurface(x, y, z, Ellipsoid, scaleEllipsoid);
41     s=20; %size of balls in surface
42     for i=1:length(vertices)
43         x1=vertices(i,:);
44         scalar = x1*A*x1.';
45         scatter3(x1(1), x1(2), x1(3),s,scalar,'fill');
46     end
47     caxis([-200000 200000])
48     colorbar('CLimMode','manual','CLim',[-200000, 200000])
49     title(sprintf('PosNeg plot for <Ax,x> on ellipsoid with scale %g ...
50         and stepsize %g', scaleEllipsoid, step));
51     view(32,29);
52 end
53 toc;

```



## B.4 Burgers Equations & Finite Elements

The main code used to produce the graphs on Burgers equation with the implementation of the finite element method:

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  %% Step 1 – Initializing Burgers FEM
3  %clear all;clc;
4  global n h m q k TFE f g M T D C B y tameEps taming_ineffectRegion x ...
   odeSystemFE
5  odeSystemFE=[];
6  % Total Spatial elements for the dicretization
7      n=256;
8  % Final time
9      final_t = 10;
10 % Total number of Time steps saved (equally spaced)
11     m = 200;
12 % Reynolds Number
13     Re = 50;
14     q=1/Re;
15
16     Δ = 0; % Neumann Boundary condition
17
18 % taming_ineffectRegion = 0.1; % margin in which taming function is ...
   ineffective
19     tameEps = 1;
20     taming_ineffectRegion = 0.01;
21 % RHS functions
22 %     f = 'f_func';
23
24 % Initial functions
25     g = @(x) 12*(1-x.^2)+Δ*x;
26     %g = @(x) x^2*(1-x)^2;
27     %g = @(x) cos(pi*x)+1;
28
29 % End Step 1
30 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
31
32 % Some adjustments for internal nodes
33     %n=n-1; % Internal nodes
34 % Compute step sizes
35     h=1/n; % space step size
36     %x = 0:h:1;
37     x=linspace(0,1,n);
38 %% Step 2
39 % Generate Mass matrix
40     M=zeros(n,n);
41     for i=1:n
42         M(i,i)=2/3;

```

```

43     if i>1
44         M(i-1,i)=1/6;
45     end
46     if i<n
47         M(i+1,i)=1/6;
48     end
49 end
50 M(1,1) = 1/3; % Dirichlet-Neumann boundary conditions
51 M=M*h;
52
53 % Generate B (Convective Term)
54 B=zeros(n,n);
55 for i=1:n
56     if i>1
57         B(i-1,i)=1/2;
58     end
59     if i<n
60         B(i+1,i)=-1/2;
61     end
62 end
63 B(1,1) = -1/2; % Dirichlet-Neumann boundary conditions
64
65 % Generate Stiffness Matrix
66 C=zeros(n,n);
67 for i=1:n
68     C(i,i)=2;
69     if i>1
70         C(i-1,i)=-1;
71     end
72     if i<n
73         C(i+1,i)=-1;
74     end
75 end
76 C(1,1) = 1; % Dirichlet-Neumann boundary conditions
77 C=C*h^(-1);
78
79 % Generate D matrix (for Neumann boundary cond.)
80 D=zeros(n,1);
81 D(1,1)=Δ;
82
83 % Finite Element Taming Matrix, tridiagonal ones
84 TFE = zeros(n,n);
85 for i=1:n
86     TFE(i,i)=1;
87     if i>1
88         TFE(i-1,i)=1;
89     end
90     if i<n
91         TFE(i+1,i)=1;
92     end
93 end

```

```

94
95 % Initial Conditions (Gauss Quadrature)
96 % Generate Initial y
97 %G = GQ_three(g);
98 %     for i=1:length(M)
99 %         G(i) = feval(g,x(i));
100 %     end
101 %G=zeros (n,1);
102 %for i=1:n
103
104 % compute inner produce <g,phi_i>
105 %     intGPhi = zeros(1,n);
106 %     for p=1:n
107 %         gphi = @(y) g(y).*phi(y,x,p); % y = integration variable
108 %         intGPhi(p) = integral(gphi,0,1);
109 %     end
110
111 % point evaluation of initial condition
112 G=zeros (n,1);
113 for i=1:n
114     G(i) = g(x(i));
115 end
116 %init_y = M\intGPhi.';
117 init_y=G;
118
119 % Taming Matrix with projection of eps-taming onto FE space
120 %T=generate_T(length(init_y));
121
122 % End Step 2
123 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
124 %% Step 3
125 t=linspace(0,final_t,m);
126
127 %options = odeset('RelTol',1e-6,'AbsTol',1e-10);
128
129 options=odeset('Mass',M);
130 %[t, yOrg] = ode15s(@FEM_Burgers_func,t,init_y,options);
131 %[t, yT ] = ode45(@FEM_Burgers_func_tamed,t,init_y,options);
132 %[t, yOrg] = ode15s(@FEM_Burgers_func_tamedFE,t,init_y,options);
133 [t, yOrg] = ode15s(@FEM_Burgers_tamed_cutoff,t,init_y,options);
134
135 % Add Dirichlet Boundary conditions
136 %tempyOrg=[yOrg';zeros(1,length(t))];
137 tempyOrg=[yOrg'];
138
139 % tempyT=[yT';zeros(1,length(t))];
140 % tempyTFE=[yTFE';zeros(1,length(t))];
141 %tempyTFE2=[yTFE2';zeros(1,length(t))];
142
143 %initial_y = [init_y;0];
144 initial_y = [init_y];

```

```

145 %x=[x,0];
146 %% FEM Plot
147 figure('name',['FEM for q= 1/',num2str(1/q),' ',num2str(n),' ...
    elements']);
148
149 % subplot(2,2,1);
150 % plot(x,initial_y);
151 % title(sprintf('Initial condition'));
152
153 subplot(1,2,2);
154 plot(x,tempyOrg(:,end));
155 title(sprintf('Final state with taming at time %g',final_t));
156
157 subplot(1,2,1);
158 mesh(x,t,tempyOrg');
159 title(sprintf('Tamed FEM for q= 1/%g, %g elements',1/q,n));
160 %axis([0 1 0 final_t 0 1.1])
161 xlabel('x')
162 ylabel('t')
163 zlabel('y (t,x)')
164
165 fprintf('2-Norm of final state: %g after %g sec ...
    \n',norm(tempyOrg(:,end),2),final_t);
166 fprintf('inf-Norm of final state: %g after %g sec ...
    \n',norm(tempyOrg(:,end),inf),final_t);
167
168 % mesh(x,t,tempyT');
169 % title(sprintf('using projected taming term'));
170 % axis([0 1 0 final_t 0 1.1])
171 % xlabel('x')
172 % ylabel('t')
173 % zlabel('y (t,x) with taming')
174 %
175 % subplot(2,2,4);
176 % mesh(x,t,tempyTFE');
177 % title(sprintf('using 3rd order finite element taming'));
178 % axis([0 1 0 final_t 0 1.1])
179 % xlabel('x')
180 % ylabel('t')
181 % zlabel('y (t,x) with FE taming')
182
183 % figure
184 % mesh(x,t,tempyTFE2');
185 % title(sprintf('using 3rd order finite element taming'));
186 % axis([0 1 0 final_t 0 1.1])
187 % xlabel('x')
188 % ylabel('t')
189 % zlabel('y (t,x) with FE taming2')
190
191 %% Plot of initial conditions
192 % figure

```

```

193 % subplot(1,2,1);
194 % plot(G);
195 % subplot(1,2,2);
196 % plot(initial_y);

```

These are the ODE files for various taming methods used to solve the FEM for Burgers' equation and generate the graphs shown in the numerical experiment section:

```

1 function [dydt]=FEM_Burgers_func(in_t, in_y)
2
3 global M B q C f D x odeSystemFE;
4
5 %F = RHS_F(f, in_t);
6 %F=f_func(x(1:end-1),in_t)';
7 %F=0;
8
9 %dydt = M\(-(1/2)*B*in_y.^2-q*(D+C*in_y));
10 dydt = -(1/2)*B*in_y.^2-q*(D+C*in_y);
11
12 odeSystemFE = [odeSystemFE,dydt];
13
14 end

```

```

1 function [ftxy]=FEM_Burgers_func_tamed(in_t, in_y)
2
3 global M B D q C f T tameEps;
4
5 %F = RHS_F(f, in_t);
6
7 in_yvec = vec( in_y.^2 * (in_y.') );
8
9 ftxy = M\(-(1/2)*B*in_y.^2-q*(D+C*in_y)) - tameEps*T*in_yvec ;
10
11 end

```

```

1 function [ftxy]=FEM_Burgers_func_tamedFE(in_t, in_y)
2
3 global M B D q C f TFE tameEps;
4
5 %F = RHS_F(f, in_t);
6
7 ftxy = M\(-(1/2)*B*in_y.^2-q*(D+C*in_y) - tameEps*TFE*in_y.^3 );
8
9 end

```

```

1 function [dydt]=FEM_Burgers_tamed_cutoff(in_t, in_y)
2
3 global B D q C tameEps taming_ineffectRegion;
4
5 r2 =norm(in_y,inf);
6 %F = RHS_F(f, in_t);
7
8 if r2 >= taming_ineffectRegion
9     dydt = -(1/2)*B*in_y.^2-q*(D+C*in_y) - ...
10         (tameEps*(r2-taming_ineffectRegion)^2)*in_y;
11     %fprintf('TAMED: Inf-norm of current state: %g, at time t=%g \n ...
12         ',r2,in_t);
13 else
14     dydt = -(1/2)*B*in_y.^2-q*(D+C*in_y);
15 end
16 end

```

This is the cutoff function used for the taming above:

```

1 function [y]=cutoffBurger1(x)
2
3 global taming_ineffectRegion TFE;
4
5 r2=norm(x,inf);
6
7 %r=norm(TFE*x.^2);
8
9 % no change inside ellipsoid
10 region1 = (r2 < taming_ineffectRegion);
11 y(region1) = 0;
12
13 % taming starts quadratically
14 region2 = (r2 >= taming_ineffectRegion);
15 y(region2) = (r2-taming_ineffectRegion)^2;
16
17 end

```

The implementation of the piecewise smooth finite element basis functions uses this matlab code.

```

1 function y=phi(t,x,j)
2 % Inputs
3 % t - sampling of the x-plane used to evaluate/plot function
4 % x - sampling of the x-plane used for finite elements, determines ...
5 % order of the finite elements (length(x)=n)

```

```

6 % j - number of the element considered
7 % Outputs:
8 % y - vector of length(t) with values of the jth element sampled over t
9
10 switch j
11     % first element
12     case 1
13         region1 = ( t ≥ x(j) ) & ( t < x(j+1) ); % radius \in [0,N]
14         y(region1) = (x(j+1)-t(region1))/(x(j+1)-x(j));
15         %y(region3) = -t(region3);
16
17         region2 = (t ≥ x(j+1));
18         y(region2) = 0;
19
20     % last element
21     case length(x)
22         region1 = ( t < x(j-1) );
23         y(region1) = 0;
24
25         region2 = (x(j-1) ≤ t) & ( t ≤ x(j) ); % radius \in [0,N]
26         y(region2) = (t(region2)-x(j-1))/(x(j)-x(j-1));
27         %y(region2) = t(region2);
28
29     % regular elements
30     otherwise
31         region1 = ( t < x(j-1) );
32         y(region1) = 0;
33
34         region2 = (x(j-1) ≤ t) & ( t < x(j) ); % radius \in [0,N]
35         y(region2) = (t(region2)-x(j-1))/(x(j)-x(j-1));
36         %y(region2) = t(region2);
37
38         region3 = ( t ≥ x(j) ) & ( t < x(j+1) ); % radius \in [0,N]
39         y(region3) = (x(j+1)-t(region3))/(x(j+1)-x(j));
40         %y(region3) = -t(region3);
41
42         region4 = (t ≥ x(j+1));
43         y(region4) = 0;
44     end
45 end

```

Which was used in the following code to generate the graph of FEM shape functions.

```

1 %% Generating finite Element functions
2 numFE=20;
3 x = linspace(0,1,numFE);
4 t=0:0.001:1;
5
6 phi_1 = phi(t,x,1);

```

```

7 phi_2 = phi(t,x,8);
8 phi_3 = phi(t,x,15);
9 phi_4 = phi(t,x,20);
10
11 figure('name','Basis function of the finite element space');
12 plot(t,phi_1,'g-',t,phi_2,'b-',t,phi_3,'r-',t,phi_4,'k-');
13 %plot(t,phi_1);
14 axis([0 1 -0.1 1.5])
15 title(sprintf('Some basis functions of the finite element space for ...
    %g elements',numFE));
16 legend('\Phi_1(x)', '\Phi_8(x)', '\Phi_{15}(x)', '\Phi_{20}(x)')

```

## B.5 Matlabs pdepe solver

Using Matlabs own *pdepe* solver for Burgers equation, this is the code used: first the main file and afterwards the support files.

```

1 %% PDE solver for Burgers equation
2 %clear all;
3 global q NeumannLeft odeSystemPdepe;
4 %odeSystemPdepe=[]; % storing DuDt for every timestep
5 % main settings:
6 Re = 50;
7 q=1/Re;
8 NeumannLeft=0;
9
10 finalT = 10;
11 timeSteps = 201;
12 spacialSteps = 257;
13
14 initialFunction = @(x) 12*(1-x^2)+NeumannLeft*x;
15 %initialFunction = @(x) sin(pi*x)
16 %initialFunction = @(x) 5*cos(pi*x);
17 %initialFunction = @(x) 50*(1/4-x)*(1/2-x)*(3/4-x);
18 %initialFunction = @(x) cos(pi*x)+1+NeumannLeft*x;
19
20 %% actually solve the PDE
21 m=0;
22 x = linspace(0,1,spacialSteps);
23 t = linspace(0,finalT,timeSteps);
24 sol = pdepe(m,@burgers_pde,initialFunction,@pdexlbc,x,t);
25 save('BurgersSolPDEPE','sol');
26
27 %% plotting the solution
28 %close all;
29 figure('name',sprintf('Re=%g, finalT=%g, Solution graph and ...

```



```

    profile',Re,t(end)));
30 subplot(1,2,1);
31 u=sol(:, :, 1);
32 mesh(x,t,u);
33 title(sprintf('PDE solution for q= 1/%g, %g elements',1/q,spacialSteps));
34
35 subplot(1,2,2);
36 plot(x,u(end, :));
37 title(sprintf('Solution at the final time t=%g',t(end)));
38
39 fprintf('2-Norm of final state: %g after %g sec ...
    \n',norm(u(end, :),2),finalT);
40 fprintf('inf-Norm of final state: %g after %g sec ...
    \n',norm(u(end, :),inf),finalT);
41
42 %% just display last state
43 % figure('name',sprintf('Re=%g, finalT=%g, Solution graph and ...
    profile',Re,t(end)));
44 % plot(x,u(end, :));
45 % title(sprintf('Solution at the final time t=%g',t(end)));

```

```

1 function [c,f,s] = burgers_pde (x,t,u,DuDx)
2 global q;
3
4 % PDE described in the following way:
5 %  $c \cdot DuDx = x^{(-m)} \cdot DDx(x^m \cdot f) + s$ 
6 % note: for m=0, it boils down to  $c \cdot DuDx = DfDx + s$ 
7
8 c=1;
9 f= q*DuDx;
10 s=u*DuDx;
11
12 end

```

```

1 function [pl,ql,pr,qr] = pdexlbc (xl,ul,xr,ur,t)
2 global q NeumannLeft;
3 % Boundary conditions described in the following way:
4 % for x=0:  $pl + ql \cdot f = 0$ 
5 % for x=1:  $pr + qr \cdot f = 0$ 
6 % with f from PDE function / here:  $f=q \cdot DuDx$ 
7 % so it boils down to:
8 % x=0:  $q \cdot ql \cdot DuDx = -pl$ 
9 % x=1:  $q \cdot qr \cdot DuDx = -pr$ 
10
11 % Neumann boundary conditions on the left
12 %  $DuDx = NeumannLeft$  for  $x = 0$ 
13 pl=-NeumannLeft;
14 ql=1/q;

```

```
15
16 % Dirichlet boundary conditions on the left
17 % pl=ul;
18 % ql=0;
19
20 % Dirichlet boundary conditions on the right
21 pr=ur;
22 qr=0;
23
24
25 end
```