The Pennsylvania State University
The Graduate School

DEEP NEURAL NETWORKS AND HOMOTOPY CONTINUATION METHODS

A Dissertation in
Mathematics
by
Chunyue Zheng

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

December 2021

The dissertation of Chunyue Zheng was reviewed and approved by the following:

Wenrui Hao
Assistant Professor of Mathematics
Dissertation Advisor, Chair of Committee

Jinchao Xu
Verne M. Willaman Professor of Mathematics

Xiantao Li
Professor of Mathematics

Sencun Zhu
Associate Professor of Computer Science and Engineering

Alexei Novikov
Professor of Mathematics
Chair of the Graduate Program, Department of Mathematics

Abstract

The dissertation contains two parts. In the first part of the dissertation, we study the approximation property of deep neural networks(DNNs) and their application in numerical problems. In particular, we dive deep into the connection between DNNs with rectified linear unit (ReLU) function as the activation function and linear finite elements. By exploring the DNN representation of its nodal basis functions, we present a ReLU DNN representation of continuous piecewise linear (CPWL) functions in FEM with an estimation of the number of neurons in DNN that are needed in such a representation. Moreover, we present some numerical results for using ReLU DNNs to solve a two-point boundary problem to demonstrate the potential of applying DNN for the numerical solution of partial differential equations. In addition to the PDE example mentioned, we also apply the DNNs to approximate bifurcations of nonlinear parametric systems. After representing the solution of the nonlinear system in the form of neural networks with the parameters as input, we define an objective function and solve an optimization problem to obtain the approximation of bifurcation. We provide numerical results to demonstrate the feasibility of the method.

In the second part of the dissertation, we study the numerical methods for computing bifurcations of nonlinear parametric systems. First, we propose an adaptive step-size homotopy tracking method. We use the Puiseux series interpolation and tangent cone structure to numerically compute bifurcation points and solutions on different branches. While the adaptive homotopy tracking algorithm focuses on computing bifurcations, we sometimes only care about the path's starting and ending points. To avoid the singularities during tracking, we present a stochastic homotopy tracking algorithm that can randomly perturb the original parametric system in each step. We then show that the stochastic solution path introduced by this new method is still theoretically close to the original solution path. Various numerical examples of nonlinear systems are given to illustrate the efficiency

of these new approaches. Moreover, as an application of the adaptive homotopy tracking method, we develop a bifurcation analysis for a mathematical model of the plaque formation with a free boundary in the early stage of atherosclerosis. By performing the perturbation analysis to the radially symmetric steady-state solutions, we establish the existence of bifurcation branches for the low-density lipoprotein (LDL)/high-density lipoprotein (HDL)-cholesterol ratio and derive a theoretical condition that a bifurcation occurs for different modes. We also analyze the stability of radially symmetric steady-state solutions and conduct numerical simulations using the adaptive homotopy tracking method to verify all the theoretical results.

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to first extend my greatest gratitude to my advisor, Professor Wenrui Hao, for his patience, continuous support, and immense knowledge. His expertise is invaluable in formulating the research questions and methodologies. This dissertation would not have been possible without his advice and guidance.

I would also like to thank Professor Jinchao Xu for the discussions and mentoring during the completion of the DNN-FEM paper. His passion for mathematics has always inspired me. In addition, I would like to thank Professor Xiantao Li and Professor Sencun Zhu, for their time to serve as my committee member and their feedback on this dissertation.

I also want to thank my lovely roommates Hongxu Wei, Yixuan Zeng, and Wei Wei. When I was in my lowest days, they kept telling me to believe in myself and were always there to listen to my struggles. Having them as roommates is one of the best things that have happened in my life. Besides, I want to thank my dearest peers for their help over the years. Special shout to Zehao Guan, Enhui Huang, Haonan Zhang, and Bingjie Li. We have witnessed the growth of each other since our undergraduate days, and I am happy to see that we all made it!

Finally, I would like to thank my parents for their unconditional love. They have supported all my decisions and always have confidence in me. I cannot make this far without their support and encouragement.

This work was support in part by NSF DMS-1818769. The findings and conclusions of this dissertation do not necessarily represent the views of the National Science Foundation.

Dedication

To my parents.

Chapter 1

Introduction

1.1 Deep neural networks and applications

In recent years, deep learning models have achieved unprecedented success in various tasks of machine learning or artificial intelligence, such as computer vision, natural language processing, and reinforcement learning [1]. One primary technique in deep learning is the deep neural network (DNN). A typical DNN model is based on the compositions of linear functions and a given nonlinear activation function.

Mathematical analysis of DNN can be carried out using many different approaches. One approach is to study the approximation properties of the function class provided by DNN. The approximation property of DNN is relevant to the so-called expressive power [2] of a DNN model. Early studies of approximation properties of DNN can be traced back in [3] and [4] where the authors established some approximation properties for the function classes given by a feedforward neural network with a single hidden layer. Further error estimates for such neural networks in terms of the number of neurons can be found in [5] for sinusoidal activation functions and in [6] for more general sigmoidal activation functions. There are many other papers on this topic during the 90s and a good review of relevant works can be found in [7] and [8].

There are many different choices of activation functions. In fact, as shown in [9], a neural network with a single hidden layer can approximate any continuous function for any activation function which is not a polynomial. Among all the

activation functions, the so-called rectified linear unit (ReLU) activation function [10], namely $\mathrm{ReLU}(x) = \max(x, 0)$, has emerged to be one of the most popular activation functions used in the deep learning literature and applications. [11] presents an approximation of ReLU DNNs by relating to wavelets. Recently, [12] establish L^∞ and L^2 error bounds for functions of many variables that are approximated by linear combinations of ReLU. [13] presents rates of approximation by deep CNNs for functions in the Sobolev space $H^r(\Omega)$ with $r > 2 + d/2$. For Holder continuous functions, [14] shows that for any function f on $[0, 1]$ that can be approximated using one hidden layer with the best N-term approximation rate $\mathcal{O}(N^{-\eta})$, the two-layers ReLU neural network can improve the approximation rate to $\mathcal{O}(N^{-2\eta})$.

In this dissertation, we consider a special class of CPWL functions, namely the linear finite element (LFE) functions [15] defined on a collection of special subdomains, namely simplexes in $\mathbb{R}^d$. As every finite element function can be written as a linear combination of nodal basis functions, it suffices to study DNN representation of any given nodal basis function. To represent a nodal basis function by a DNN, we do not need to consider the complicated domain partition related with lattice representation[16], which is important in representing general piecewise linear functions in $\mathbb{R}^d$ [17]. We prove that a linear finite element function with N degrees of freedom can be represented by a ReLU DNN with at most $\mathcal{O}(d\kappa^d N)$ number of neurons with $\mathcal{O}(d)$ hidden layers where $\kappa \geq 2$) depends on the shape regularity of the underlying finite element grid. We also show (see [18]) that at least 2 hidden layers are needed for a ReLU DNN to represent any linear finite element function in a bounded domain $\Omega \subset \mathbb{R}^d$ or $\mathbb{R}^d$ when $d \geq 2$. The $\lceil \log_2(d+1) \rceil$ number of hidden layers is also optimal for $d = 2, 3$.

Another topic that will be investigated in the paper is the application of artificial neural networks for differential equations. This topic can be traced back to [19, 20, 21] in which collocation methods are studied. Recently, there are increased new research interests in the literature for the application of deep neural networks for numerical approximation of nonlinear and high dimensional PDEs as in [22, 23, 24]. Based on our result about the relationship between FEM and ReLU DNNs, we discuss the application of ReLU DNN for solving PDEs with respect to the convergence properties. In particular, we use a 1D example to demonstrate

that a Galerkin method using ReLU DNN can lead to better approximation results than the adaptive finite element method that has the same number of degrees of freedom as in the ReLU DNN.

We also present an application of neural networks in approximating the bifurcations of nonlinear parametric systems. In the second part of the dissertation, we discuss homotopy algorithms to find bifurcations. However, traditional homotopy methods may not work well when the parameter is high dimensional. The approximation properties and the Puiseux series expansions[25] inspire us to use neural networks to learn the solution paths and then solve an optimization problem to find bifurcations. This dissertation shows that a neural network with a single hidden layer can approximate the bifurcation point well numerically.

1.2 Homotopy tracking method for bifurcations of nonlinear parametric systems

Many mathematical models of natural phenomena, e.g., biology [26], physics [27, 28], and materials science [29] involve systems of nonlinear equations [30, 31, 32, 26]. Studies of these nonlinear parametric equations are formulated numerically and theoretically to focus on solution structures such as bifurcations, which occur when the parameter change causes the solution structure to change. Typical types of bifurcation include saddle-node bifurcation, transcritical bifurcation, pitchfork bifurcation, and Hopf bifurcation [33]. Although theory helps us to understand the solution structures in many cases [34, 35], the in-depth and more quantitative study of these problems often requires large-scale simulations to numerically compute bifurcations. However, computing these different bifurcation points numerically brings the same challenge. In specific, this corresponds to the real part of an eigenvalue of the Jacobian passing through zero and causes numerical challenges for Newton's and Newton-like methods [36, 37, 38]. Therefore, efficient numerical methods for computing bifurcations of large-scale systems are keys to understanding these systems.

The homotopy continuation method [39, 40, 41, 42] has been successfully used to compute bifurcations and structural stabilities for studying parametric prob-

lems. The basic idea is to trace out a one-real-dimensional solution curve described implicitly by a system of equations: given a nonlinear system $\boldsymbol{F}(\boldsymbol{u})$ to solve, one first forms a nonlinear system $\boldsymbol{G}(\boldsymbol{u})$ that is related to $\boldsymbol{F}(\boldsymbol{u})$ in a prescribed way but has known, or easily computable solutions. The systems $\boldsymbol{G}(\boldsymbol{u})$ and $\boldsymbol{F}(\boldsymbol{u})$ are combined to form a homotopy, such as the linear homotopy

$$\boldsymbol{H}(\boldsymbol{u}, t) = \boldsymbol{F}(\boldsymbol{u})(1-t) + t\boldsymbol{G}(\boldsymbol{u}) = 0, \tag{1.1}$$

where $\boldsymbol{G}(\boldsymbol{u})$ is a start system with known solutions and $\boldsymbol{F}(\boldsymbol{u})$ is the target system we want to solve. Then solutions of $\boldsymbol{F}(\boldsymbol{u}) = 0$ can be solved by tracking t from 1 to 0 via this linear homotopy. In numerical algebraic geometry (NAG), there is a well-developed theory on how to choose the start system $\boldsymbol{G}(\boldsymbol{u})$ to guarantee all the solutions of $\boldsymbol{F}(\boldsymbol{u})$ via this homotopy.

In the homotopy setup, each solution path can be tracked via the prediction/correction algorithm [43, 44, 45] which is referred as the homotopy tracking algorithm. This algorithm could become very inefficient if the parametric system is singular or near singular. To avoid the singular system, the gamma trick is proposed to construct a random homotopy setup in (1.1) by multiplying a random complex number [45]. Then the probability of hitting a singularity during the tracking is zero. Nevertheless, the system could be still near singular so that the homotopy tracking is still time-consuming [43, 44, 45]. In order to address this numerical challenge, an adaptive multi-precision path tracking algorithm [46] has been developed by adjusting precision in response to step failure according to the error estimates. An endgame algorithm [47] has also been widely used to deal with the singularities at $t = 0$. In this paper, we will develop an adaptive step-size homotopy tracking method [25] to control the tracking step-size each time. This algorithm integrates numerical methods from numerical algebraic geometry and scientific computing so that we can apply this efficient method to compute bifurcation points of large-scale nonlinear systems such as discretized systems arising from nonlinear PDEs.

All the algorithms mentioned above focus on the adaptivity of the stepsize by measuring the distance to the singularities but cannot skip these singularities during the tracking, which could be very slow and inefficient when the size of

nonlinear systems becomes large [48]. Motivated by the stochastic algorithms used in scientific computing [49, 50], e.g., the coordinate gradient descent, we present an efficient stochastic homotopy tracking method that gives the original system a random perturbation each step to avoid singularities and improve the efficiency during the tracking.

Chapter 2

ReLU Deep Neural Networks and Linear Finite Elements

2.1 Preliminaries

2.1.1 General DNN

Given $n, m \geq 1$, the first ingredient in defining a deep neural network (DNN) is (vector) linear functions of the form

$$\Theta : \mathbb{R}^n \to \mathbb{R}^m, \tag{2.1}$$

as $\Theta(x) = Wx + b$ where $W = (w_{ij}) \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$.

The second main ingredient is a nonlinear activation function, usually denoted as

$$\sigma : \mathbb{R} \to \mathbb{R}. \tag{2.2}$$

By applying the function to each component, we can extend this naturally to

$$\sigma : \mathbb{R}^n \mapsto \mathbb{R}^n.$$

Given $d, c, k \in \mathbb{N}^+$ and

$$n_1, \ldots, n_k \in \mathbb{N} \text{ with } n_0 = d, n_{k+1} = c,$$

a general DNN from $\mathbb{R}^d$ to $\mathbb{R}^c$ is given by

$$\begin{aligned} f(x) &= f^k(x), \\ f^\ell(x) &= [\Theta^\ell \circ \sigma](f^{\ell-1}(x)) \quad \ell = 1 : k, \end{aligned}$$

with $f^0(x) = \Theta(x)$. The following more concise notation is often used in computer science literature:

$$f(x) = \Theta^k \circ \sigma \circ \Theta^{k-1} \circ \sigma \cdots \circ \Theta^1 \circ \sigma \circ \Theta^0(x), \tag{2.3}$$

here $\Theta^i : \mathbb{R}^{n_i} \to \mathbb{R}^{n_{i+1}}$ are linear functions as defined in (2.1). Such a DNN is called a $(k+1)$-layer DNN, and is said to have k hidden layers. Unless otherwise stated, all layers mean hidden layers in the rest of this paper. The size of this DNN is $n_1 + \cdots + n_k$. In this paper, we mainly consider a special activation function, known as the *rectified linear unit* (ReLU), and defined as $\mathrm{ReLU} : \mathbb{R} \mapsto \mathbb{R}$,

$$\mathrm{ReLU}(x) = \max(0, x), \quad x \in \mathbb{R}. \tag{2.4}$$

A ReLU DNN with k hidden layers might be written as:

$$f(x) = \Theta^k \circ \mathrm{ReLU} \circ \Theta^{k-1} \circ \mathrm{ReLU} \cdots \circ \Theta^1 \circ \mathrm{ReLU} \circ \Theta^0(x). \tag{2.5}$$

We note that ReLU is a continuous piecewise linear (CPWL) function. Since the composition of two CPWL functions is obviously still a CPWL function, we have the following simple observation([17]).

Lemma 2.1.1. *Every ReLU DNN: $\mathbb{R}^d \to \mathbb{R}^c$ is a continuous piecewise linear function. More specifically, given any ReLU DNN, there is a polyhedral decomposition of $\mathbb{R}^d$ such that this ReLU DNN is linear on each polyhedron in such a decomposition.*

In the rest of this chapter, we will use the terminology of CPWL to define the class of functions that are globally continuous and locally linear on each polyhedron in a given finite polyhedral decomposition of $\mathbb{R}^d$.

For convenience of exposition, we introduce the following notation:

$$\begin{aligned}\mathrm{DNN_J} := \{f : f = \Theta^J \circ \mathrm{ReLU} \circ \Theta^{J-1} \cdots \mathrm{ReLU} \circ \Theta^0(x),\\ \Theta^\ell \in \mathbb{R}^{n^{\ell+1}\times(n^\ell+1)}, \quad n^0 = d, \quad n^{J+1} = 1, \quad n^\ell \in \mathbb{N}^+\}.\end{aligned} \tag{2.6}$$

Namely $\mathrm{DNN_J}$ represents the DNN model with J hidden layers and ReLU activation function with arbitrary size.

2.1.2 A shallow neural network DNN_1

We note that for $J = 0$, DNN_0 is a simple function space of global linear functions, which is often used in classic statistical analysis such as linear regression. The structure of DNN_J gets more interesting as J becomes larger. We shall now discuss the simple case when $J = 1$, namely

$$\mathrm{DNN}_1^m = \left\{ f : f = \sum_{i=1}^m \alpha_i \mathrm{ReLU}(w_i x + b_i) + \beta \right\}, \tag{2.7}$$

where $\alpha_i, b_i, \beta \in \mathbb{R}, w_i \in \mathbb{R}^{1\times d}$, for $i = 1, 2, ..., m$. Here we introduce the superscript m to denote the number of neurons. This simple neural network already has rich mathematical structures and approximation properties. Given a bounded domain $\Omega \subset \mathbb{R}^d$, we introduce the following notation

$$\mathrm{DNN}_1^m(\Omega) = \left\{ f : f(x) \in \mathrm{DNN}_1^m, \quad x \in \Omega \subset \mathbb{R}^d \right\}, \tag{2.8}$$

as the restriction of DNN_1^m on Ω.

Approximation property for the function class $\mathrm{DNN}_1^m(\Omega)$ has been much studied in the literature. For example, in [3] and [4], $\mathrm{DNN}_1^m(\Omega)$ is proved to be dense in $C^0(\Omega)$ as $m \to \infty$, which is known as universal approximation. There are also many works devoted to the asymptotic error estimates. For example, [6] established the following estimate:

$$\inf_{g\in\mathrm{DNN}_1^m(\Omega)} \|f - g\|_{0,2,\Omega} \lesssim |\Omega|^{1/2} m^{-\frac{1}{2}} \int_{\mathbb{R}^d} |\omega|_\Omega |\hat{f}(\omega)| d\omega, \tag{2.9}$$

where $|\Omega|$ denotes the volume of Ω and

$$|\omega|_\Omega = \sup_{x\in\Omega} |\omega \cdot (x - x_\Omega)|,$$

for some point $x_\Omega \in \Omega$.

For a given set of w^i and b^i, it is tempting to think the functions in DNN_1^m are generated by $\{\mathrm{ReLU}(w_i x + b_i)\}_{i=1}^m$. In such a consideration, the following result is of great theoretical interest. The proof can be found in [18].

Theorem 2.1.1. *$\{\mathrm{ReLU}(w_i x + b_i)\}_{i=1}^m$ are linearly independent if (w_i, b_i) and (w_j, b_j) are linearly independent in $\mathbb{R}^{1\times(d+1)}$ for any $i \neq j$.*

In real applications, w_i and b_i are variables. As a result, DNN_1^m is generated by variable basis functions $\{\mathrm{ReLU}(w_i x + b_i)\}_{i=1}^m$ and in particular DNN_1^m is a nonlinear space which is expected to have certain nonlinear approximation property as discussed in [51].

2.1.3 Linear finite element spaces

The finite element method (FEM), as a popular numerical method for approximating the solutions of partial differential equations (PDEs), is a well-studied subject ([15],[52]). The finite element function space is usually a subspace of the solution space, for example, the space of piecewise linear functions over a given mesh. In [17], it is shown that piecewise linear functions can be written as ReLU DNNs, which will be discussed in details later.

Assuming that $\Omega \subset \mathbb{R}^d$ is a bounded domain. We consider a special finite element function class consisting of CPWL functions with respect to a simplicial partition of Ω. Such simplicial partitions are often known as finite element grids or meshes. Some typical finite element grids are shown in Figure 2.1 for $d = 1, 2, 3$.

A finite element space is defined in association with a simplicial finite element grid $\mathcal{T}_h \subset \Omega$. A simplicial finite element grid $\mathcal{T}_h$ consists of a set of simplexes $\{\tau_k\}$ and the corresponding set of nodal points is denoted by $\mathcal{N}_h$. For a given grid $\mathcal{T}_h$, the corresponding finite element space is given by

$$V_h = \{v \in C(\Omega) : v \text{ is linear on each element } \tau_k \in \mathcal{T}_h\}. \tag{2.10}$$

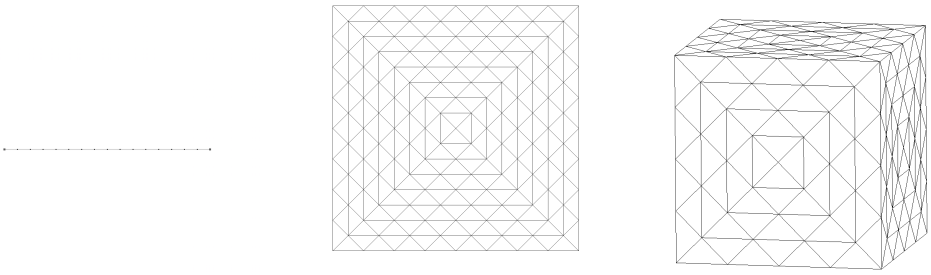


Figure 2.1: an interval, a triangle and a tedrohedron partition

Given $x_i \in \mathcal{N}_h$, it is easy to see that there exists a unique function $\phi_i \in V_h$, known as the nodal basis function, such that

$$\phi_i(x_j) = \delta_{ij}, \quad x_j \in \mathcal{N}_h. \tag{2.11}$$

A typical profile of ϕ_i is shown in Fig. 2.2 for $d = 1$ and $d = 2$.

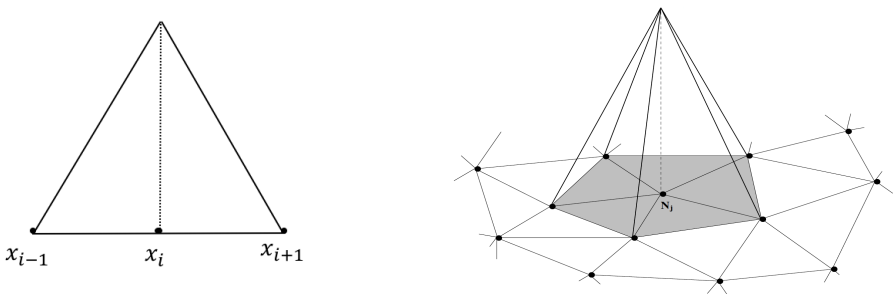


Figure 2.2: The basis function in 1D and 2D

Obviously any $v \in V_h$ can be uniquely represented in terms of these nodal basis functions:

$$v(x) = \sum_{i=1}^{N} \nu_i \phi_i(x), \tag{2.12}$$

where N is the degrees of freedom.

Given $x_i \in \mathcal{N}_h$, let $N(i)$ denote all the indices j such that τ_j contains the nodal point x_i, namely

$$N(i) = \{j : x_i \in \tau_j\},$$

and k_h denote the maximum number of neighboring elements in the grid

$$k_h \equiv d(\mathcal{T}_h) = \max_{x_i \in \mathcal{N}_h} |N(i)|. \tag{2.13}$$

Let $G(i)$ denote the support of the nodal basis ϕ_i:

$$G(i) = \bigcup_{k \in N(i)} \tau_k.$$

We say that the grid $\mathcal{T}_h$ is locally convex if $G(i)$ is convex for each i.

2.2 Linear finite element (LFE) function as a DNN

In this section, we give a constructive proof to demonstrate how a finite element function can be represented by a ReLU DNN. Our derivation and analysis are based on the representation of the finite element function as a linear combination of basis functions as follows.

2.2.1 DNN representation of finite element functions

As an illustration, we will now demonstrate how a linear finite element function associated with a locally convex grid $\mathcal{T}_h$ can be represented by a ReLU DNN. For more general grids, we refer to Remark 1 and [17].

Thanks to (2.12), it suffices to show that each basis function ϕ_i can be represented by a ReLU DNN. We first note that the case where $d = 1$ is trivial as the basis function ϕ_i with support in $[x_{i-1}, x_{i+1}]$ can be easily written as

$$\phi_i(x) = \frac{1}{h_{i-1}}\mathrm{ReLU}(x - x_{i-1}) - (\frac{1}{h_{i-1}} + \frac{1}{h_i})\mathrm{ReLU}(x - x_i) + \frac{1}{h_i}\mathrm{ReLU}(x - x_{i+1}), \tag{2.14}$$

where $h_i = x_{i+1} - x_i$.

In order to consider the cases where $d > 1$, we first prove the following lemma.

Lemma 2.2.1. *Given $x_i \in \mathcal{N}_h$, if $G(i)$ is convex, then the corresponding basis*

function can be written as

$$\phi_i(x) = \max\left\{0, \min_{k\in N(i)} g_k(x)\right\}, \tag{2.15}$$

where, for each $k \in N(i)$, g_k is the global linear function such that $g_k = \phi_i$ on τ_k.

Proof. To show (2.15) holds for all $x \in \mathbb{R}^d$, we first consider the case $x \in G(i)$, namely $x \in \tau_{k_0}$ for some $k_0 \in N(i)$. Thus

$$\phi_i(x) = g_{k_0}(x) \geq 0. \tag{2.16}$$

Let P_k be the hyperplane that passes through the $d-1$ subsimplex (of τ_k) that does not contain x_i (see the left figure in Figure 2.3). Since $G(i)$ is convex by assumption, all points in τ_{k_0} should be on the same side of the hyperplane P_k. As a result, for all $k \in N(i)$,

$$g_k(y) \geq 0 \equiv g_{k_0}(y), \quad y \in P_{k_0} \cap \tau_{k_0}.$$

By combining the above inequality with the following obvious inequality that

$$g_k(x_i) = 1 \geq 1 = g_{k_0}(x_i), \quad k \in N(i),$$

and the fact that all g_k are linear, we conclude that

$$g_k(y) \geq g_{k_0}(y), \quad \forall y \in \tau_{k_0}, k \in N(i).$$

In particular

$$g_k(x) \geq g_{k_0}(x), \quad k \in N(i).$$

This, together with (2.16), proves that (2.15) holds for all $x \in G(i)$. Thus

$$\max\left\{0, \min_{k\in N(i)} g_k(x)\right\} = g_{k_0}(x). \tag{2.17}$$

On the other hand, if $x \notin G(i)$, there exists a $\tau_k \subset G(i)$ such that τ_k contains a segment of the straight line that pass through x and x_i(see the right figure in Figure 2.3). Again let P_k be the hyperplane associated with τ_k as defined above.

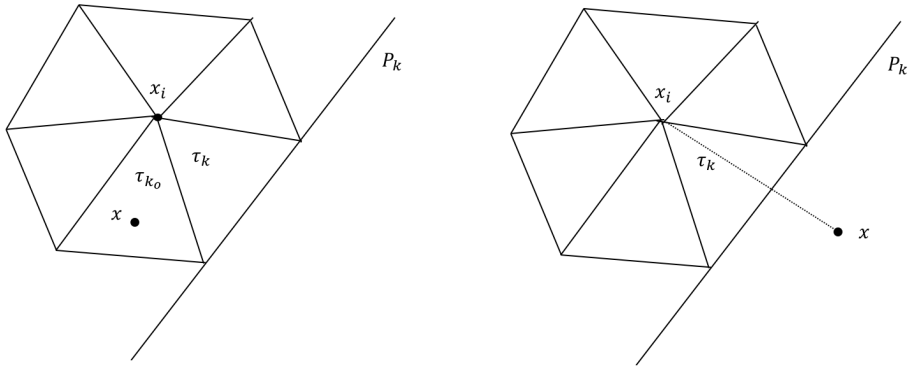


Figure 2.3: Left: $x \in G(i)$, right: $x \notin G(i)$

We note that x and x_i are on the different sides of P_k. Since

$$g_k(x_i) \geq 0, \quad g_k(y) = 0, \quad y \in P_k,$$

we then have

$$\min_{k \in N(i)} g_k(x) \leq g_k(x) \leq 0,$$

which implies

$$\max\left\{0, \min_{k \in N(i)} g_k(x)\right\} = 0 = \phi_i(x), \qquad x \notin G(i).$$

This finishes the proof of Lemma 2.2.1. □

Remark 1. *If $G(i)$ is not convex, we could also write the basis function as some max-min functions. But the form of max-min function is not as simple as the case where $G(i)$ is convex, and it depends on the shape of the support of the basis function. In some cases, we can write the basis function as the max-min-max form if $G(i)$ is a special non-convex set.*

We are now in a position to state and prove the main result in this section.

Theorem 2.2.1. *Given a locally convex finite element grid $\mathcal{T}_h$, any linear finite element function with N degrees of freedom, can be written as a ReLU-DNN with at most $k = \lceil \log_2 k_h \rceil + 1$ hidden layers and at most $\mathcal{O}(k_h N)$ number of the neurons.*

Proof. We have the following identity,

$$\min\{a,b\} = \frac{a+b}{2} - \frac{|a-b|}{2} = v \cdot \mathrm{ReLU}(W \cdot [a,b]^T), \tag{2.18}$$

where

$$v = \frac{1}{2}[1,-1,-1,-1], \qquad W = \begin{bmatrix} 1 & 1 \\ -1 & -1 \\ 1 & -1 \\ -1 & 1 \end{bmatrix}.$$

By Lemma 2.2.1, the basis function $\phi_i(x)$ can be written as:

$$\phi_i(x) = \max\left\{0, \min_{k\in N(i)} g_k(x)\right\}.$$

For convenience, we assume that

$$N(i) = \{r_1, r_2, ...r_{|N(i)|}\}.$$

Then we have

$$\begin{aligned}
\min_{k\in N(i)} g_k(x) &= \min\left\{g_{r_1}(x), ..., g_{r_{|N(i)|}}(x)\right\} \\
&= \min\left\{ \min\{g_{r_1}, ..., g_{r_{\lceil |N(i)|/2\rceil}}\}, \min\{g_{r_{\lceil |N(i)|/2\rceil+1}}, ..., g_{r_{|N(i)|}}\} \right\} \\
&= v \cdot \mathrm{ReLU}(W \cdot \begin{bmatrix} \min\{g_{r_1}, ..., g_{r_{\lceil |N(i)|/2\rceil}}\} \\ \min\{g_{r_{\lceil |N(i)|/2\rceil+1}}, ..., g_{r_{|N(i)|}}\} \end{bmatrix}).
\end{aligned}$$

According to this procedure, we get the minimum of $|N(i)|$ terms by splitting them in two, each taking the minimum over at most $\lceil |N(i)|/2 \rceil$ terms. This contributes to one ReLU hidden layer. Then we can further split the terms

$$\min\{g_{r_1}, ..., g_{r_{\lceil |N(i)|/2\rceil}}\}, \quad \min\{g_{r_{\lceil |N(i)|/2\rceil+1}}, ..., g_{r_{|N(i)|}}\}$$

until all the minimum functions contain only 1 or 2 terms.

1. If there is one term

$$\min\{a\} = a.$$

2. If there are two terms

$$\min\{a, b\} = v \cdot \mathrm{ReLU}(W \cdot [a, b]^T).$$

which is also a ReLU DNN with 1 hidden layer. So we can write a basis function as a $1 + \lceil \log_2 k_h \rceil$-hidden-layer DNN. Considering the binary-tree structure, a k-layer full binary-tree has $2^k - 1$ nodes. We can see the number of neurons is at most

$$\mathcal{O}(2^k) = \mathcal{O}(2^{1+\lceil \log_2 k_h \rceil}) = \mathcal{O}(k_h).$$

By (2.12), the piecewise linear function can be represented as a DNN with $k = 1 + \lceil \log_2 k_h \rceil$ hidden layers. The number of neurons is at most $\mathcal{O}(k_h N)$. □

We now consider a special class of the so-called shape regular finite element grid $\mathcal{T}_h$ which satisfies

$$\kappa_1 \leq \frac{r_\tau}{R_\tau} \leq \kappa_2, \quad \forall \tau \in \mathcal{T}_h, \tag{2.19}$$

for some constants κ_1 and κ_2 independent of h and d, where r_τ (R_τ) is the radius of the largest (smallest) ball contained in (containing) τ.

Corollary 2.2.1. *Given a locally convex and shape regular finite element grid $\mathcal{T}_h$, any linear finite element function with N degrees of freedom(DOFs), can be written as a ReLU-DNN with at most $\mathcal{O}(d)$ hidden layers. The number of neurons is at most $\mathcal{O}(\kappa^d N)$ for some constant $\kappa \geq 2$ depending on the shape-regularity of $\mathcal{T}_h$. The number of non-zero parameters is at most $\mathcal{O}(d\kappa^d N)$.*

We note that, using the approach described in this section, a finite element function with N DOFs can be represented by a DNN with $\mathcal{O}(N)$ number of weights. This property is expected to be useful when DNNs are used in adaptive mesh-less or vertex-less numerical discretization methods for partial differential equation, which is a subject of further study.

2.2.2 Comparison of error estimates in adaptive finite element and DNN methods

Error estimates for adaptive finite element methods are well studied in the literature. For example, an appropriately adapted linear finite element function with $\mathcal{O}(N)$ DOFs is proved to admit the following error estimate:

$$\inf_{v\in V_h} \|u-v\|_{0,2,\Omega} \leq CN^{-\frac{2}{d}} |u|_{2,\frac{2d}{d+2},\Omega}, \tag{2.20}$$

if $u \in W^{2,\frac{2d}{d+2}}(\Omega)$ and v is the interpolation based on the adapted finite element grid. More details can be founded in [53, 51].

For a shallow network DNN_1 with $\mathcal{O}(N)$ DOFs (i.e. $\mathcal{O}(\frac{N}{d})$ neurons), we have the next error estimate in (2.9) as

$$\min_{v\in \mathrm{DNN}_1^{\frac{N}{d}}(\Omega)} \|u-v\|_{0,2,\Omega} \lesssim |\Omega|^{1/2} \left(\frac{N}{d}\right)^{-\frac{1}{2}} \int_{\mathbb{R}^d} |\omega|_\Omega |\hat{u}(\omega)| d\omega. \tag{2.21}$$

In comparison, an adaptive finite element function with the same order of $\mathcal{O}(N)$ DOFs can only have convergence rate of order $\mathcal{O}(N^{-\frac{2}{d}})$.

As will be shown in §2.3, shallow neural networks DNN_1 (namely with only one hidden layer) cannot recover a linear finite element function in general, but may potentially lead to better asymptotic accuracy as the dimension d gets larger.

One idea that may help us to understand is that the shallow network is a kind of N-term or basis selection ([51])approximation scheme with $\{\sigma(w_i x + b_i)\}_{i=1}^N$ as the basis functions (as shown in Theorem 2.1.1), similar to using $\{sin(nx)\}_{n=1}^N$ as the basis functions in Fourier approximation or some others in wavelets.

For deep ReLU neural networks, our connections of FEM and ReLU DNNs in this section help us to construct a special ReLU DNN models with depth $\mathcal{O}(d)$ and parameters $\mathcal{O}(dk_dN)$ for $\mathcal{O}(N)$ DOFs. By using the approximation result for adaptive FEM, DNN approximation u_{DNN} for special structure with $\mathcal{O}(N)$ DOFs can get

$$\|u-u_{DNN}\|_{0,2,\Omega} \lesssim \left(\frac{N}{dk_d}\right)^{-\frac{2}{d}} |u|_{1,\frac{2d}{2+d},\Omega} \lesssim (N)^{-\frac{2}{d}} |u|_{1,\frac{2d}{2+d},\Omega}, \tag{2.22}$$

and $k_d = \mathcal{O}(\kappa^d)$. This shows that there exists some special deep ReLU DNN

structure which is at least as good as adaptive FEM.

2.3 LFE can not be recovered by DNN_1 for $d \geq 2$

In the previous section, we show that a finite element function can be represented by a ReLU DNN with $\log_2 k_h + 1$ hidden layers.

In view of Lemma 2.1.1 and the fact that $\mathrm{DNN}_{\mathrm{J}} \subseteq \mathrm{DNN}_{\mathrm{J}+1}$, it is natural to ask that how many layers are needed at least to recover all linear finite element functions in $\mathbb{R}^d$. In this section, we will show that

$$J_d \geq 2, \quad \text{if} \quad d \geq 2, \tag{2.23}$$

where J_d is the minimal J such that all linear finite element functions in $\mathbb{R}^d$ can be recovered by DNN_J.

In particular, we will show the following theorem.

Theorem 2.3.1. *If $\Omega \subset \mathbb{R}^d$ is either a bounded domain or $\Omega = \mathbb{R}^d$, DNN_1 can not be used to recover all linear finite element functions in Ω.*

This shows that despite it has the so-called universal approximation properties [3, 4], shallow network is not enough in the case of recovering all CPWL functions. More precisely, although the shallow ReLU DNNs are CPWL functions themselves and can approximate any CPWL functions with any accuracy, there are some CPWL functions they cannot represent exactly. As an example, a local basis function in FEM with compact support and some other simple conditions cannot be represented by ReLU DNNs with one hidden layer for dimensions greater than 2.

As for the upper bound, results in [17] provides us with one answer.

Corollary 2.3.1.

$$2 \leq J_d \leq \lceil \log_2(d+1) \rceil. \tag{2.24}$$

This also indicates that $\lceil \log_2(d+1) \rceil$ is "optimal" for $d = 2, 3$.

2.4 Application to Numerical PDEs

In this section, we discuss the application of DNNs to the numerical solution of partial differential equations (PDEs). In most of our discussion, we consider the following model problem:

$$\begin{aligned} -\Delta u &= f, \qquad x \in \Omega, \\ \frac{\partial u}{\partial \nu} &= 0, \qquad x \in \partial\Omega, \end{aligned} \tag{2.25}$$

here $\Omega \subset \mathbb{R}^d$ is a bounded domain. For simplicity of exposition, we only consider Neuman boundary condition here. As it is done in the literature, special cares need to be taken for Dirichlet boundary value problems, but we will not get into those (standard) details.

The idea of using DNN for numerical PDEs can be traced back to [54] where a collocation method is used. Similar ideas have been explored by many different authors for different types of PDEs.

For the model problem (2.25), the collocation method amounts to the following least square problem:

$$\min_{\Theta} \sum_{x_i \in \Omega} (-\Delta u_N(x_i, \Theta) - f(x_i))^2, \tag{2.26}$$

here $u_N(x, \Theta)$ is taken among the DNN function class in the form of (2.3) with a smooth activation function such as sigmoidal function and x_i are some collocation points.

Recently, [55] applied DNN for numerical PDE in the Galerkin setting which amounts to the solution of the following energy minimization problem:

$$\min_{\Theta} \int_{\Omega} (\frac{1}{2}|\nabla u_N(x, \Theta)|^2 - f u_N(x, \Theta))dx \tag{2.27}$$

Numerical experiments have demonstrated the potential of this approach. In the rest of this section, we will discuss a number of aspects of this approach from both theoretical and practical viewpoints. In particular, we will discuss its relationship with two popular finite element methods: adaptive finite element method and

moving grid method.

2.4.1 The finite element method

The finite element approximation to (2.25) can be written as

$$\min_{v\in V_h}\int_\Omega (\frac{1}{2}|\nabla v(x)|^2 - fv(x))dx, \tag{2.28}$$

where V_h is the finite element space as described in §2.1.3.

In the finite element setting, the optimization problem (2.28) is to find the coefficient (ν_i) as in (2.12) for a given finite element mesh $\mathcal{T}_h$. Some more sophisticated versions of the finite element method can be obtained by varying or optimizing $\mathcal{T}_h$ so that more accurate finite element approximation can be obtained. Roughly speaking, there are two main approaches for optimizing $\mathcal{T}_h$: one is the adaptive finite element method and the other is the moving grid finite element method.

The adaptive finite element method is, roughly speaking, to vary $\mathcal{T}_h$ by either coarsening or refining the grid. One main theoretical result is that a family of adapted grids $\mathcal{T}_h$ with $\mathcal{O}(N)$ degrees of freedom can be obtained so that the corresponding adaptive finite element approximation u_N satisfies the following error estimate

$$|u - u_h|_{1,2,\Omega} \le CN^{-\frac{1}{d}}|u|_{2,\frac{2d}{d+2},\Omega}. \tag{2.29}$$

We refer to [53, 51] for relevant details and its generalizations.

One interesting observation is that the convergence rate $\mathcal{O}(N^{-\frac{1}{d}})$ in (2.29) deteriorate badly as d increases. Of course, error estimate in the form (2.29) depends on which Sobolev or Besov function classes that the solution u belongs to, namely what norms are used in the right hand side of (2.29). But regardless what function classes for the solution u, no asymptotic error estimate seems to be known in the literature that is better than $\mathcal{O}(N^{-\frac{1}{d}})$.

The moving grid method is, on the other hand, to optimize $\mathcal{T}_h$ by varying the location of grid points while preserving topological structure of the grids (in particular the number of grid ponts remain unchanged). This approach proves to be effective in many applications, see [56, 57]. But there are very few theories on the error estimate like (2.29) in the moving grid method.

However, the H^1 approximation properties are still unclear even for DNN_1. [58] proves a similar result for H^1 error estimate for $\mathrm{DNN}_1^m(\Omega)$ with activation function $\sigma(x) = \cos(x)$. For general activation functions, or just for ReLU, it is an open problem.

2.4.2 DNN-Galerkin method

The finite element methods discussed above, including adaptive method and moving grid method, depend crucially on the underlying finite element grids. Numerical methods based on DNN, as we shall describe now, are a family of numerical methods that require no grids at all. This is reminiscent of the "mesh-less method" that have been much studied in recent years [59, 60, 61]. But the mesh-less method still requires the use of discretization points. The *DNN-Galerkin* method (as we shall call), namely the Galerkin version of the DNN-element method such as (2.27), goes one step further: it does not even need any discretization points! It is a totally point-free method!

Let us now give a brief discussion on the error estimate for the DNN-Galerkin method. We first recall a classic result by [6] for a DNN with one hidden layer of $\mathcal{O}(N)$ DOFs (i.e. $\mathcal{O}(\frac{N}{d})$ neurons),

$$\inf_{v \in \mathrm{DNN}_1^{\frac{N}{d}}(\Omega)} \|u - v\|_{0,2,\Omega} \lesssim \left(\frac{N}{d}\right)^{-\frac{1}{2}} C_u, \tag{2.30}$$

here we have

$$C_u := \int_{\mathbb{R}^d} |\omega|_\Omega |\hat{u}(\omega)| d\omega, \tag{2.31}$$

where $\hat{u}$ is the Fourier transform of any extension of the original function defined in Ω to the entire space $\mathbb{R}^d$. Here we need to point that C_u might scale with dimension d. The dependence on d is improved by [62, 63]. Especially, [63] improve this constant to be polynomial in d.

2.4.3 An 1D example: a two point-boundary value problem

As a proof of concept, let us discuss a very simple one dimensional example. We focus on the following model problem:

$$\begin{aligned} -u''(x) &= f, \qquad x \in (0,1). \\ u(0) &= u(1) = 0. \end{aligned} \tag{2.32}$$

The exact solution $u \in H^1(0,1)$ satisfies that

$$u = \arg\min_{v \in H_0^1(0,1)} E(v), \tag{2.33}$$

where

$$E(v) = \int_0^1 (\frac{1}{2}|v(x)'|^2 - fv(x))dx.$$

Given a grid

$$\mathcal{T}_N : 0 = t_0 < t_1 < ... < t_{N+1} = 1.$$

We define the space of ReLU DNNs with one hidden layer as follows:

$$U = \{u(x;t,\theta) | u(x;t,\theta) = \sum_{i=0}^{N} (\theta_{i+1} - \theta_i)\mathrm{ReLU}(x - t_i)\},$$

where θ_i is the slope of piecewise linear function in $[t_{i-1}, t_i]$. In order to satisfy the condition $u(1;t,\theta) = 0$, we have the constraint

$$\theta_0 = 0, \quad \sum_{i=0}^{N+1} \theta_{i+1}(t_{i+1} - t_i) = 0.$$

We minimize the energy norm

$$(t,\theta) = \arg\min_{t,\theta} \int_0^1 (\frac{1}{2}|u'(x;t,\theta)|^2 - fu(x;t,\theta))dx, \qquad u(x;t,\theta) \in U.$$

where $t = (t_0, t_1, ..., t_{N+1}), \theta = (\theta_0, \theta_1, ..., \theta_{N+1})$. We do the alternate iteration as

below,

$$\theta^{k+1} = \arg\min_{\theta} \int_0^1 \left(\frac{1}{2}|u'(x;t^k,\theta)|^2 - fu(x;t^k,\theta) \right) dx,$$
$$t^{k+1} = t^k - \eta \nabla_t \left(\int_0^1 \left(\frac{1}{2}|u'(x;t,\theta^{k+1})|^2 - fu(x;t,\theta^{k+1}) \right) dx \right),$$

where η is the step-length. Once t is fixed, the minimization problem is a quadratic optimization, which is the traditional finite element method. So we solve the FEM solution $u(x;t^k,\theta^{k+1})$ on grid t and then compute the slope θ_i on each $[t_{i-1},t_i]$.

Algorithm 1 Simulation $1D$ PDE

Data: Grid t, Max iteration step M.
Result: Optimal solution $u(x;t^*,\theta^*)$.
Solve θ on the grid t;
while $k \leq M$ **do**
  $g = \nabla_t \left(\int_0^1 \left(\frac{1}{2}|u'(x;t,\theta)|^2 - fu(x;t,\theta) \right) dx \right)$;
  Find η by line search;
  $t \leftarrow t - \eta g$;
  Solve θ on the grid t;
  $k \leftarrow k+1$;
end while

We choose the exact solution as

$$u(x) = x(e^{-(x-\frac{1}{3})^2/K} - e^{-\frac{4}{9}/K}),$$

with $K = 0.01$. In this numerical experiment, the learning rate $\eta = 0.5$, the max iteration step is 200, and the degrees of freedom $N = 53$.

N	$\lvert u_{uFEM} - u\rvert_1$	$\lvert u_{AFEM} - u\rvert_1$	$\lvert u_{DNN} - u\rvert_1$	$E(u_{uFEM})$	$E(u_{AFEM})$	$E(u_{DNN})$
23	0.2779	0.1375	0.1094	-0.7047	-0.7338	-0.7373
37	0.1717	0.0760	0.0663	-0.7285	-0.7404	-0.7411
53	0.1193	0.0511	0.0456	-0.7362	-0.7420	-0.7422

Table 2.1: The H^1 semi-norm error and energy

At the beginning of the simulation, we use the adaptive finite element method (AFEM) to get the adaptive grid from the uniform grid. Next we construct DNN solution with the same degrees of freedom. Then we minimize the energy and get

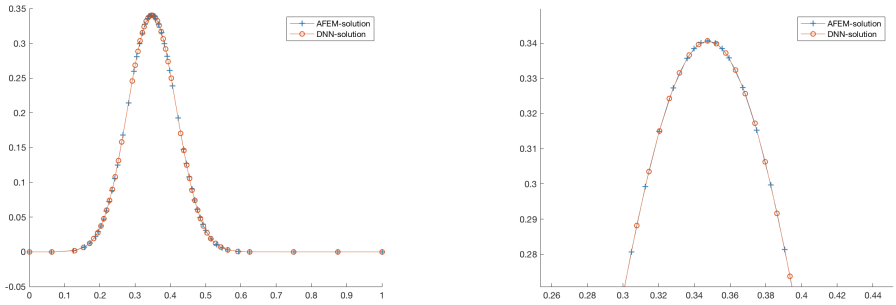


Figure 2.4: The grid of AFEM and DNN(the right figure is the top of left).

the DNN solution. We compare the energy and H^1 semi-norm error of uniform grid solution(uFEM), AFEM solution and DNN solution. From Table 2.1, the energy and H^1 semi-norm of the DNN solution are smaller than u_{AFEM} and u_{uFEM}, which implies that the DNNs can find better solution than AFEM. Figure 2.4 shows the two different grid points on the same graph, we can easily see the grid points are moving.

Chapter 3

An application of DNN in approximating bifurcations

3.1 Formulation

Generally speaking, a nonlinear parametric system is written as $\mathbf{F} : \mathbb{R}^n \times \mathbb{R}^d \to \mathbb{R}^n$,

$$\mathbf{F}(\mathbf{u}, \mathbf{p}) = \mathbf{0}, \tag{3.1}$$

where $\mathbf{p}$ is a parameter and $\mathbf{u}$ is the variable vector that depends on the parameter $\mathbf{p}$, i.e., $\mathbf{u} = \mathbf{u}(\mathbf{p})$.

We want to use neural networks to approximate the bifurcation of the above system. Given a collection of data points $\{\mathbf{p}_i, \mathbf{u}_i\}_{i=1}^K$ where $\mathbf{F}(\mathbf{u}_i, \mathbf{p}_i) = \mathbf{0}$, first we can train the neural network to approximate the solution $\mathbf{u}(\mathbf{p})$ with the loss function as below

$$f_1 = \min_{\tilde{\mathbf{u}}(\mathbf{p}) \in DNN_J} \|\tilde{\mathbf{u}}(\mathbf{p}) - \mathbf{u}(\mathbf{p})\|^2 + \frac{\lambda}{2}\|\mathbf{F}(\tilde{\mathbf{u}}(\mathbf{p}), \mathbf{p})\|^2 \tag{3.2}$$

Now with the neural network approximated solution $\tilde{\mathbf{u}}(\mathbf{p})$, we want to further approximate the bifurcations. To this end, we need to find the solution of the following minimization problem:

$$f_2 = \min_{\mathbf{p}, \mathbf{v}} \frac{\mathbf{v}^T \mathbf{F}_{\mathbf{u}}^T(\tilde{\mathbf{u}}(\mathbf{p}), \mathbf{p}) \mathbf{F}_{\mathbf{u}}(\tilde{\mathbf{u}}(\mathbf{p}), \mathbf{p}) \mathbf{v}}{\mathbf{v}^T \mathbf{v}} + \lambda_2 \|\mathbf{F}(\tilde{\mathbf{u}}(\mathbf{p})), \mathbf{p})\|^2 \tag{3.3}$$

Algorithm 2 The pseudocode of the neural network approximation algorithm.

Input: Epochs n, Number of neurons N, a solution sequence on the path $\{\mathbf{p}_i, \mathbf{u}_i\}_{i=1}^{K}$.

Output: Bifurcation on the path $\mathbf{p}_N^*$.

Define the loss function for training as (3.2);

Denote the DNN as $\mathbf{u}_N(\mathbf{p})$;

for $i = 1 : n$ **do**

  Train $\mathbf{u}_N(\mathbf{p})$;

end for

Solve optimization problem (3.3) to get $\mathbf{p}_N^*$;

3.2 Examples

In this section, we will apply Algorithm 2 to approximate bifurcations for different parametric systems. For Section 3.2.1 and Section 3.2.2, we always use one-layer neural network with ReLU as activation function.

3.2.1 Example 1

We start with the following system with $p^* = 0$ as a turning point:

$$F(x, p) = x^2 - p = 0 \tag{3.4}$$

The training data $\{(p_i, \sqrt{p_i})\}_{i=1}^{K=1600}$ is generated by randomly choose $p_i \in [0, 2]$ following a uniform distribution. By choosing different number of hidden units, we have the following result:

3.2.2 Example 2

Consider

$$F(x, \mathbf{p}, n) = x^n + \mathbf{p}_{n-1}x^{n-1} + \cdots + \mathbf{p}_0 = 0$$

Table 3.1: Error for approximating bifurcation with different widths of neural networks for (3.4)

Number of Hidden Units	$\lvert p^* - p_N^* \rvert$
N = 20	0.0970
N = 40	0.0528
N = 80	0.0237
N = 160	0.0153
N = 320	0.0100

3.2.2.1 $n = 2$

For

$$F(x, p) = x^2 + bx + c = 0 \tag{3.5}$$

Then bifurcation appears at $\mathbf{p} = (b, c) = (b, b^2/4)$.

The training data $\{b_i, c_i, \frac{-b_i+\sqrt{b_i^2-4c_i}}{2}\}_{i=1}^{K=5000}$ is generated by randomly choose $b_i \in [-2, 2]$ following a uniform distribution. c_i is drawn from the uniform distribution on $[\frac{b_i^2}{8}, \frac{b_i^2}{4}]$ to guarantee the existence of real solutions.

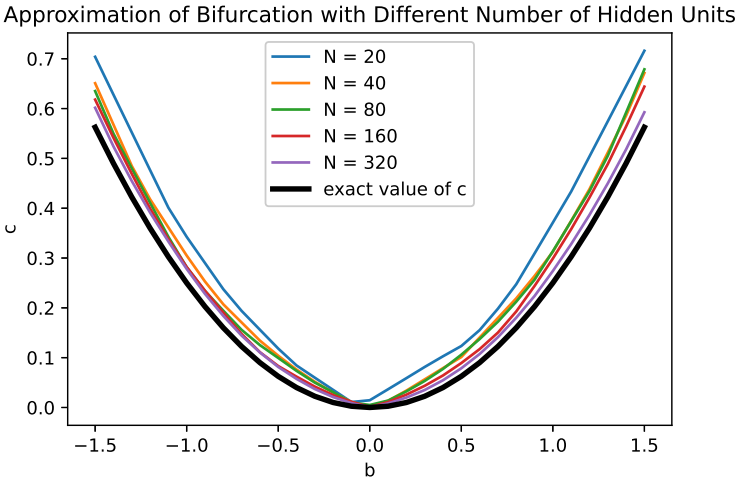


Figure 3.1: Approximation of bifurcation curve for (3.5)

3.2.2.2 $n = 3$

For

$$F(x, \mathbf{p}) = ax^3 + bx^2 + cx + d = 0 \tag{3.6}$$

here $\mathbf{p} = (b, c, d)$ is the parameter vector. Let

$$\Delta = (\frac{bc}{6a^2} - \frac{b^3}{27a^3} - \frac{d}{2a})^2 + (\frac{c}{3a} - \frac{b^2}{9a^2})^3 \tag{3.7}$$

Then the roots of $F(x)$ is

$$\begin{aligned}
x_1 =& -\frac{b}{3a} + \sqrt[3]{\frac{bc}{6a^2} - \frac{b^3}{27a^3} - \frac{d}{2a} + \sqrt{\Delta}} + \sqrt[3]{\frac{bc}{6a^2} - \frac{b^3}{27a^3} - \frac{d}{2a} - \sqrt{\Delta}} \\
x_2 =& -\frac{b}{3a} + \frac{-1+\sqrt{3}i}{2}\sqrt[3]{\frac{bc}{6a^2} - \frac{b^3}{27a^3} - \frac{d}{2a} + \sqrt{\Delta}} \\
&+ \frac{-1-\sqrt{3}i}{2}\sqrt[3]{\frac{bc}{6a^2} - \frac{b^3}{27a^3} - \frac{d}{2a} - \sqrt{\Delta}} \\
x_3 =& -\frac{b}{3a} + \frac{-1-\sqrt{3}i}{2}\sqrt[3]{\frac{bc}{6a^2} - \frac{b^3}{27a^3} - \frac{d}{2a} + \sqrt{\Delta}} \\
&+ \frac{-1+\sqrt{3}i}{2}\sqrt[3]{\frac{bc}{6a^2} - \frac{b^3}{27a^3} - \frac{d}{2a} - \sqrt{\Delta}}
\end{aligned} \tag{3.8}$$

When $\Delta = 0$, the equation has three real roots.

- If $(\frac{bc}{6a^2} - \frac{b^3}{27a^3} - \frac{d}{2a})^2 = -(\frac{c}{3a} - \frac{b^2}{9a^2})^3 \neq 0$, two of three roots are equal.
- If $(\frac{bc}{6a^2} - \frac{b^3}{27a^3} - \frac{d}{2a})^2 = -(\frac{c}{3a} - \frac{b^2}{9a^2})^3 = 0$, all three roots are equal.

Fix $a = 1$, then bifurcation appears at $(c, d) = (\frac{b^2}{3}, \frac{b^3}{27})$.

The training data is denoted as $\{b_i, c_i, d_i, -\frac{b_i}{3} + 2\sqrt[3]{\frac{b_i c_i}{6} - \frac{b_i^3}{27} - \frac{d_i}{2}}\}_{i=1}^{K=12000}$. To generate the training dataset, we first get 3000 random b_i from the uniform distribution on $[0, 2]$. For each value of b_i, c_i is drawn from the uniform distribution on $[0, \frac{b_i^2}{3}]$ for 4 times. After obtaining pairs of (b_i, c_i), apply the equality $(\frac{bc}{6a^2} - \frac{b^3}{27a^3} - \frac{d}{2a})^2 = -(\frac{c}{3a} - \frac{b^2}{9a^2})^3 = 0$ to get the corresponding value of d_i.

Here we list the integraion of c and d on the interval $[0, \frac{3}{2}]$ comparing to the exact solution:

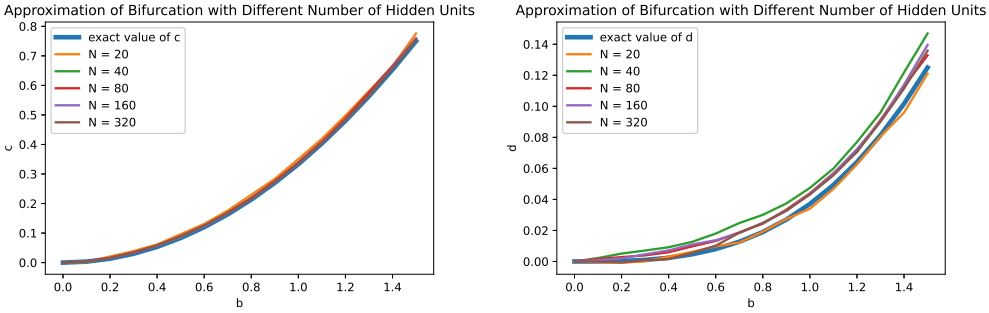


Figure 3.2: For given values of b, use neural network to approximate (c, d) for (3.6).

Hidden Units	c		d	
	exact	approximated	exact	approximated
H = 20	0.375	0.3938	0.046875	0.0457
H = 40		0.3815		0.0622
H = 80		0.3845		0.0556
H = 160		0.3806		0.0562
H = 320		0.3813		0.0532

Chapter 4

Adaptive Homotopy Tracking with Bifurcation Detection

4.1 Homotopy Continuation Method

From now on, we only consider the case when the parameter $\mathbf{p}$ in (3.1) is a scalar and denote the scalar parameter as p.

For the parametric system (3.1), the standard homotopy continuation method [64, 65] uses a predictor-corrector method to track the solution $\mathbf{u}$ as the parameter p varies. Basic prediction and correction are both accomplished by considering a local model via its Taylor expansion:

$$\mathbf{F}(\mathbf{u}+\Delta\mathbf{u}, p+\Delta p) = \mathbf{F}(\mathbf{u}, p) + \mathbf{F}_{\mathbf{u}}(\mathbf{u}, p)\Delta\mathbf{u} + \mathbf{F}_p(\mathbf{u}, p)\Delta p + \text{Higher-Order Terms},$$

where $\mathbf{F}_{\mathbf{u}} = \partial\mathbf{F}/\partial\mathbf{u}$ is the $n \times n$ Jacobian matrix and $\mathbf{F}_p = \partial\mathbf{F}/\partial p$ has size $n \times 1$.

The predictor-corrector method consists of two parts: the first one is the predictor step which gives a prediction of $\Delta\mathbf{u}$ for any given Δp based on numerical methods for solving ordinary differential equation such as Euler method, the secant predictor method, and etc; the second one is the corrector method which refines the predicted solution based on numerical methods for solving nonlinear systems such as Newton's method, conjugate gradient methods and etc (see [66] for more details). Fig 4.1 illustrates this idea using the Euler predictor and the Newton corrector. Given a solution $(\mathbf{u}_0, p_0)$ on the path, that is, $\mathbf{F}(\mathbf{u}_0, p_0) = 0$,

we plan to compute a solution at $p_1 = p_0 + \Delta p$. First we make an Euler predictor step, solving the first-order terms $\mathbf{F}_{\mathbf{u}}(\mathbf{u}_0, p_0)\Delta\mathbf{u} = -\mathbf{F}_p(\mathbf{u}_0, p_0)\Delta p$, and then letting $\tilde{\mathbf{u}}_1 = \mathbf{u}_0 + \Delta\mathbf{u}$. On the other hand, when $\|\mathbf{F}(\tilde{\mathbf{u}}_1, p_1)\|$ is not sufficiently small, one may fix p_1 to be constant by setting $\Delta p = 0$ and solving the following equation by using the Newton corrector: $\mathbf{F}_{\mathbf{u}}(\tilde{\mathbf{u}}_1, p_1)\Delta\mathbf{u} = -\mathbf{F}(\tilde{\mathbf{u}}_1, p_1)$. Repeat this corrector step until $\|\mathbf{F}(\tilde{\mathbf{u}}_1, p_1)\|$ is smaller than the chosen tolerance criterion, then we can get $\mathbf{u}_1 = \tilde{\mathbf{u}}_1 + \Delta\mathbf{u}$ and $(\mathbf{u}_1, p_1)$ is on the path.

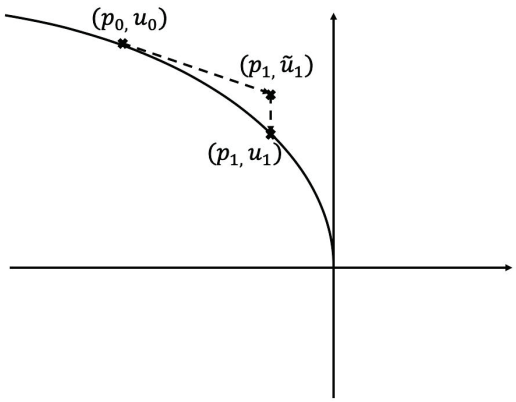


Figure 4.1: An illustration of the predictor-corrector Method.

The main concern of a numerical path-tracking algorithm is deciding which of these methods to use next and how large of a step-size Δp to use in the predictor [48, 67]. A trial-and-error approach for the step-size control is used for homotopy continuation tracking: shorten the step-size upon failure and lengthen it upon repeated successes [68, 69]. This trial-and-error approach can be computationally expensive and can lack efficiency when systems are not well-conditioned, since the step-size becomes very small. Moreover, in the path tracking process, at some critical points, the ill-conditioned Jacobian matrix $F_{\mathbf{u}}$ often causes trouble either in the prediction or in the correction process. Various computational techniques, such as pseudo-arclength continuation, Gauss-Newton continuation, and other adaptive step-size strategies [67], have been developed to handle this difficulty. For instance, the path tracking may encounter no difficulty at a turning point if the pseudo-arclength continuation is adopted. However, bifurcations of large-scale nonlinear systems are usually complex (more than turning points) and need a more sophisticated numerical method to compute.

4.2 Adaptive Homotopy Tracking with Bifurcation Detection (AHTBD)

To overcome this difficulty, an adaptive homotopy tracker is proposed to reduce the computational cost. The basic idea of this adaptive homotopy tracker is to solve the step-size simultaneously when we track the nonlinear system. For any given step-size h, we start with a point on the solution path, denoted by $(\mathbf{u}_0, p_0)$, and want to find the next point to satisfy the following augmented system:

$$\tilde{\mathbf{F}}(\mathbf{u}, p) = \begin{pmatrix} \mathbf{F}(\mathbf{u}, p) \\ g\mathbf{v}^T(\mathbf{u} - \mathbf{u}_0)(1 - s) + s(p - p_0) - h \end{pmatrix}, \tag{4.1}$$

where $g = \text{sign}(-\mathbf{v}^T\mathbf{F}_\mathbf{u}(\mathbf{u}_0, p_0)^{-1}\mathbf{F}_p(\mathbf{u}_0, p_0))/\|\mathbf{F}_\mathbf{u}(\tilde{\mathbf{u}}, \tilde{p})^{-1}\mathbf{F}_p(\tilde{\mathbf{u}}, \tilde{p})\|$, λ_{min} is the real part of the minimum eigenvalue of $\mathbf{F}_\mathbf{u}$ at $(\mathbf{u}_0, p_0)$, $\mathbf{v}$ is the corresponding eigenvector. Here $(\tilde{\mathbf{u}}, \tilde{p})$ is a generic point (i.e., randomly choosing $\tilde{p}$ to compute $\tilde{\mathbf{u}}$) [68, 69] and $\tilde{\lambda}_{min}$ is the real part of the minimum eigenvalue of $\mathbf{F}_\mathbf{u}$ at $\tilde{p}$ and $s = \left|\frac{\lambda_{min}}{\tilde{\lambda}_{min}}\right|$. Thus the next point on the path $(\mathbf{u}, p)$ is computed by solving the new augmented system $\tilde{F}$ with an adaptive step-size. In particular, when the tracking parameter p is close to a bifurcation point, λ_{min} is very small, and s approaches zero, we then have $g\mathbf{v}^T(\mathbf{u} - \mathbf{u}_0) = h$ instead of $p - p_0 = h$ which means that we change the tracking parameter from p to $\mathbf{v}^T\mathbf{u}$; when p_0 is a generic point, namely, the original system is well-conditioned, we have s be close to 1 and then $p = p_0 + h$ which is the "initial" target for the next point. Moreover, this adaptive homotopy tracking process, whose pseudocode is outlined in **Algorithm 3**, employs the Newton-Krylov method to solve the augmented nonlinear system.

Algorithm 3 The pseudocode of the adaptive tracking algorithm.

Input: A step-size h, a start point $(\mathbf{u}_0, p_0)$, and an ending parameter p_e.

Output: A solution sequence on the path $(\mathbf{u}_i, p_i)_{i=1}^N$.

Set $i = 0$;

while $(p - p_0)(p - p_e) \leq 0$ **do**

Compute the minimum eigenvalue of $\mathbf{F_u}(\mathbf{u}_i, p_i)$ and the corresponding eigenvector, $\mathbf{v}$;

Solve the augmented system (4.1) and denote the solution as $(\mathbf{u}_{i+1}, p_{i+1})$;

Set $i = i + 1$;

end while

Remark 2. *The augmented system (4.1) does not bring new singularities. In other words, if the original system is full rank, then the augmented system must be full rank. In fact, if* $\mathbf{F_u}$ *is not singular, the Jacobian matrix of the augmented system (4.1) can be written as*

$$\begin{pmatrix} \mathbf{F_u} & \mathbf{F}_p \\ g\mathbf{v}^T(1-s) & s \end{pmatrix} = \begin{pmatrix} I & 0 \\ g\mathbf{v}^T(1-s)\mathbf{F_u}^{-1} & I \end{pmatrix} \begin{pmatrix} \mathbf{F_u} & \mathbf{F}_p \\ 0 & s - g\mathbf{v}^T(1-s)\mathbf{F_u}^{-1}\mathbf{F}_p \end{pmatrix}.$$

If the original system has full rank, namely, $s \neq 0$*, then we have* $s - (1 - s)g\mathbf{v}^T\mathbf{F_u}^{-1}\mathbf{F}_p \neq 0$*, which implies that the augmented system (4.1) also has full rank. On the other hand, if* F_u *is singular, the Jacobian matrix of the augmented system could be non-singular.*

Remark 3. *The parameter tracking direction is the same as* h*. In fact, by solving*

$$\begin{pmatrix} \mathbf{F_u} & \mathbf{F}_p \\ g\mathbf{v}^T(1-s) & s \end{pmatrix} \begin{pmatrix} \Delta\mathbf{u} \\ \Delta p \end{pmatrix} = \begin{pmatrix} 0 \\ h \end{pmatrix},$$

we have

$$\Delta p = \frac{h}{s - (1-s)g\mathbf{v}^T\mathbf{F_u}^{-1}\mathbf{F}_p}.$$

Noticing the definition of g*, we have* $s - (1 - s)g\mathbf{v}^T\mathbf{F_u}^{-1}\mathbf{F}_p > 0$ *if* $s \neq 0$*, which implies that* Δp *has the same sign as* h*.*

4.2.1 Inflation Process

When the Jacobian matrix of the augmented system is ill-conditioned, the adaptive path tracking algorithm based on Newton's method is no longer satisfactory since it may converge slowly or even diverge. Once such a circumstance occurs, the deflation technique has been proposed to overcome this difficulty [70, 71]. However, the deflated system is double the size of the original nonlinear system, and sometimes even higher order derivatives need to be taken into consideration [70]. Therefore this technique is hard to apply for large-scale systems. In order to track large-scale systems, we need a different strategy, an inflation process. The motivation of the inflation technique is based on iterative methods for the ill-conditioned symmetric positive definite matrices. Let us consider a simple example with $(A + \epsilon I)x = b$ (A and b are shown below), and apply the Gauss-Seidel method with stopping criteria $\|Ax^k - b\| \leq 10^{-8}$ and $x^0 = b$. Eq. (4.2) shows the number of iterations for different value of ϵ: the number of iterations increases dramatically from 18 to $54,470$ when the matrix is ill-conditioned; the number of iterations drops to 2 when the matrix is singular. Therefore iterative methods usually are effective for a singular system, but time-consuming for a nearly singular system (see [72] for more theoretical results).

$$A = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}, b = \begin{bmatrix} -1 \\ -1 \\ 2 \end{bmatrix} \in R(A). \qquad (4.2)$$

ϵ	# of of iteration
1	18
10^{-1}	100
10^{-2}	852
10^{-3}	6,982
10^{-4}	54,470
0	2

Based on this motivation, we will inflate the nearly singular system to a singular system. More specifically, for a bifurcation point p^*, the system $\mathbf{F}(\mathbf{u}^*, p^*)$ is singular. By denoting J the Jacobian $\mathbf{F}_\mathbf{u}(\mathbf{u}, p)$, we know that J is ill-conditioned if p is close to p^* so that Newton's method becomes difficult to converge. By decomposing $\Delta\mathbf{u}$ as $\Delta\mathbf{u} = \widetilde{\Delta\mathbf{u}} + \alpha\mathbf{v}$, then we solve the following inflated system instead of

$\mathbf{F_u}(\mathbf{u},p)\Delta\mathbf{u} = -\mathbf{F}(\mathbf{u},p)$:

$$\begin{pmatrix} J^TJ & J^TJ\mathbf{v} \\ \mathbf{v}^TJ^TJ & \lambda_{min} \end{pmatrix} \begin{pmatrix} \widetilde{\Delta\mathbf{u}} \\ \alpha \end{pmatrix} = - \begin{pmatrix} J^TF(\mathbf{u},p) \\ \mathbf{v}^TJ^TF(\mathbf{u},p) \end{pmatrix}. \tag{4.3}$$

Here λ_{min} is the eigenvalue of J^TJ with the minimum norm and $\mathbf{v}$ is the corresponding eigenvector. We use J^TJ instead of J to make sure the coefficient matrix is symmetric positive semi-definite in order to guarantee the convergence of this inflation technique [72]. In fact, for any $a \in \mathbb{R}^{n\times 1}, b \in \mathbb{R}$, we have

$$\begin{aligned} (a^T,b) \begin{pmatrix} J^TJ & J^TJ\mathbf{v} \\ \mathbf{v}^TJ^TJ & \lambda_{min} \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} &= a^TJ^TJa + b\mathbf{v}^TJ^TJa + a^TJ^TJ\mathbf{v}b + \lambda_{min}b^2 \\ &= a^TJ^TJa + 2\lambda_{min}ba^T\mathbf{v} + \lambda_{min}b^2 \\ &\geq \lambda_{min}|a|^2 - 2\lambda_{min}|b||a||\mathbf{v}| + \lambda_{min}b^2 \\ &\geq \lambda_{min}(|a| - |b|)^2, \end{aligned} \tag{4.4}$$

which implies that the matrix in (4.3) is symmetric positive semi-definite. Therefore linear iterative solvers such as Gauss-Seidel or GMRES [73, 74] converge very quickly for solving the singular inflated system (4.3) [72].

Remark 4. *Since* $(\mathbf{v}^T,-1)^T$ *is in the kernel of* (4.3)*, we have a family of solutions* $(\widetilde{\Delta\mathbf{u}} + k\mathbf{v}, \alpha - k)$ *for* (4.3)*,* $\forall k$*, for any given solution pair* $(\widetilde{\Delta\mathbf{u}}, \alpha)$*. However* $\Delta\mathbf{u}$ *is unique for any* k *by the definition.*

Remark 5. *An example of the above inflation algorithm is as follows. Assume*

$$A = \begin{bmatrix} 0.7292 & 0.5737 \\ 0.3319 & 0.2613 \end{bmatrix}, b = \begin{bmatrix} 0.8782 \\ 0.3998 \end{bmatrix} \tag{4.5}$$

A is a nearly singular system. When solving the original equation $Ax = b$ *with Gauss-Seidel method with stopping criteria* $\|Ax^k - b\| \leq 10^{-8}$ *and* $x^0 = b$*, it takes 17113 iterations to get the solution. However, by using inflation as* (4.3)*, the number of iterations drop to 1.*

4.2.2 Puiseux Series Extrapolation

The power series endgame has been successfully used to handle the singularity in NAG [75, 76] for polynomial systems. This endgame technique is only used for homotopy tracking very near $t = 0$, but cannot handle the bifurcation point during the tracking. In this paper, we will develop a new numerical method based on the Puiseux Series Expansion (PSE) to approximate the bifurcation point and the solution at the bifurcation point when the nonlinear system is polynomial. The idea is to use the eigenvalue of the Jacobian matrix to interpolate the solution near the bifurcation point. In particular, at the bifurcation point, the Jacobian $\mathbf{F_u}$ has an eigenvalue with zero real part, say p_b, and several branches can come together at $(\mathbf{u}_b, p_b)$. We denote $\lambda = \min_i |real(\lambda_i)|$, where λ_i is the eigenvalue of $\mathbf{F_u}(\mathbf{u}, p)$ for any given $(\mathbf{u}, p)$. Then according to the classical Puiseux's theorem (Chapter 7 in [77] & Corollary A.3.3 in [69]) we use a Puiseux series expansion to approximate $(\mathbf{u}, p)$ in a neighborhood of $(\mathbf{u}_b, p_b)$, called the PSE operating zone. Thus the following formulation is given by

$$\mathbf{u}(\lambda) = \mathbf{u}_b + \sum_{j=1}^{\infty} \mathbf{a}_i \lambda^{j/c_1} \text{ and } p(\lambda) = p_b + \sum_{j=1}^{\infty} b_i \lambda^{j/c_2}, \tag{4.6}$$

where c_1 and c_2 are the winding numbers for path $\mathbf{u}(\lambda)$ and $p(\lambda)$, respectively. Computing the winding numbers c_1 and c_2 requires more advanced computational techniques in NAG [68, 78, 69] but can not be applied directly for large-scale nonlinear systems, e.g., the discretized polynomial systems of nonlinear PDEs. Thus in our algorithm, we make several guesses at c_1 and c_2 to get the close connection to the curvature of the paths.

Moreover, we also need to compute leading terms of the PSE, namely, $w = \min\{j|\mathbf{a}_j \neq 0\}$ and $q = \min\{j|b_j \neq 0\}$. Then (4.6) is rewritten as

$$\mathbf{u}(\lambda) = \mathbf{u}_b + \lambda^{w/c_1}(\mathbf{a}_w + \sum_{j=w+1}^{\infty} \mathbf{a}_i \lambda^{j/c_1}) \text{ and } p(\lambda) = p_b + \lambda^{q/c_2}(b_q + \sum_{j=q+1}^{\infty} b_i \lambda^{j/c_2}) \tag{4.7}$$

We will show the procedure how to estimate q/c_2, which can be extended to esti-

mate w/c_1 as well: for any constant k_1 and k_2, we have

$$p(k_1\lambda) = p_b + k_1^{q/c_2}\lambda^{q/c_2}(b_q + \sum_{j=q+1}^{\infty} b_i(k_1\lambda)^{j/c_2}),$$
$$p(k_2\lambda) = p_b + k_2^{q/c_2}\lambda^{q/c_2}(b_q + \sum_{j=q+1}^{\infty} b_i(k_2\lambda)^{j/c_2}).$$

When λ is small and $k_1 < 1, k_2 < 1$, we have

$$\frac{1-k_1^{q/c_2}}{1-k_2^{q/c_2}} \approx \frac{p(\lambda)-p(k_1\lambda)}{p(\lambda)-p(k_2\lambda)}.$$

Thus an approximation of q/c_2 is obtained by solving the following nonlinear equation:

$$f(x) := 1 - k_1^x - m(1-k_2^x) = 0,$$

where $m = \frac{p(\lambda)-p(k_1\lambda)}{p(\lambda)-p(k_2\lambda)}$. For estimating w/c_1, we multiply a random vector, α, namely, using $\alpha^T\mathbf{u}(k_1\lambda)$ and $\alpha^T\mathbf{u}(k_2\lambda)$ to repeat the above procedure. In summary, the algorithm for computing the bifurcation point based on the PSE is as follows:

Algorithm 4 Implementing PSE

Given a sequence of points on the branch, say $(\mathbf{u}^n, p^n, \lambda^n)_{n=1}^N$.
while $|\lambda| < Tol$ **do**
  Estimate the value of w/c_1 and q/c_2 by solving the nonlinear equation $f(x) = 0$;
  for $c_i = 1 : M$ **do**
    Use the first $N-1$ points to approximate the Puiseux series;
    Apply these approximations to extrapolate $(\mathbf{u}^N, p^N)$ at λ^N;
  end for
  Determine the best value of c_i by choosing the nearest extrapolating point on the paths at $\lambda = \lambda^N$;
  Use the Puiseux series to approximate $(\mathbf{u}_b, p_b)$ at $\lambda = 0$;
  if $\|(\mathbf{u}_b, p_b)\| < Tol$ **then**
    Break;
  else
    Set $\lambda = \frac{\lambda_N}{2}$, generate a new point $(\mathbf{u}^{N+1}, p^{N+1})$, and update the sequence of points;
  end if
end while

An illustrated example: We will use the following example to illustrate this PSE interpolation process:

$$F(\mathbf{u}, p) = \begin{pmatrix} x^2 - p^2 \\ (x+y)^2 - p^3 \end{pmatrix}. \tag{4.8}$$

In this example, exact solutions of one branch are

$$x = -\Big(\frac{1}{2}\Big)^{2/3}\lambda^{2/3},\ y = \Big(\frac{1}{2}\Big)^{1/3}\lambda^{2/3} + \frac{1}{2}\lambda \text{ and } \ p = \Big(\frac{1}{2}\Big)^{2/3}\lambda^{2/3},$$

where λ is the minimum eigenvalue of the Jacobian matrix. By taking $\lambda = 2$, we have our initial point $x_0 = -1$, $y_0 = 2$, and $p_0 = 1$. By taking $h = -0.1$, we collect five points on this solution path shown in Fig. 4.2. Four of them are used to compute coefficients of the Puiseux series, the other one is to determine

the winding numbers c_1 and c_2. Fig. 4.2 shows different solution trajectories by using PSE interpolation for different c_1. Then $c_1 = 3$ is the best approximation for x, y. In fact, since p is a monomial of λ, when using a different winding number c_2, the ratio q/c_2 is the same. Then the approximated bifurcation point becomes $x = -3.2 \times 10^{-5}$, $y = 1.1 \times 10^{-4}$, and $p = 3.2 \times 10^{-5}$.

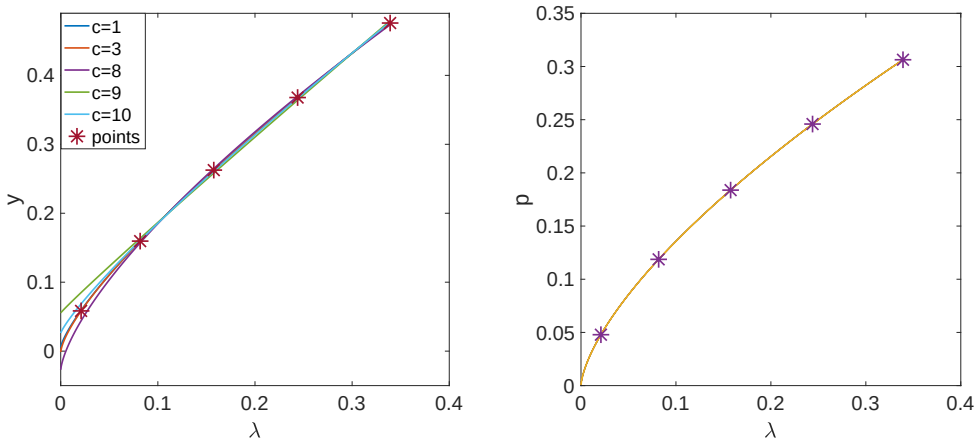


Figure 4.2: The PSE interpolation in the illustrated example (4.8). The left part shows solution trajectories of y with respect to λ for different c_1; the right part shows parameter p with respect to λ.

4.2.3 Tangent Cone

After computing the bifurcation point, the tangent cone of the bifurcation point needs to be computed in order to track along different branches by using the Lyapunov-Schmidt reduction [79, 80, 81]. The tangent cone T_* and the Jacobian matrix J_* at the bifurcation point have the following relationship

$$T_* \subseteq null(J_*),$$

which implies that the tangent cone is contained in the tangent space at a bifurcation although the tangent cone and tangent space are equal at a generic point. Then the null space of the Jacobian is computed to obtain the tangent cone at a bifurcation by using the Taylor expansion of the nonlinear system $\mathbf{F}$ in the null space of J_*. We will illustrate the procedure of computing the tangent cone by

assuming that the dimension of J_* is $n-1$. Let's denote the Jacobian $J_{\mathbf{u}}$ and the derivative J_p with respect to p at $(\mathbf{u}_0, p_0)$ as $A := [J_{\mathbf{u}}, J_p] \in R^{n\times(n+1)}$. Then we have

$$\begin{bmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \\ q_1 & q_2 \end{bmatrix} = null(A), \text{ where } \mathbf{Q}_i \in \mathbb{R}^{n\times 1} \text{ and } q_i \text{ is a scalar.}$$

Similarly, $\Lambda \in \mathbb{R}^{1\times n} = null(A^T)$. Thus we assume that

$$\Delta\mathbf{u} = a_1\mathbf{Q}_1 + a_2\mathbf{Q}_2 \text{ and } \Delta p = a_1q_1 + a_2q_2,$$

where a_i needs to be determined. We construct the following single polynomial $g(a_1, a_2)$

$$g(\mathbf{a}) = \Lambda^T F(\mathbf{u}_0 + a_1\mathbf{Q}_1 + a_2\mathbf{Q}_2, p_0 + a_1q_1 + a_2q_2).$$

By using Taylor expansion at $(0,0)$, we have

$$g(\mathbf{a}) \approx g(0,0) + \mathbf{a}^T\frac{\partial g}{\partial \mathbf{a}}(0,0) + \mathbf{a}^T H(0,0)\mathbf{a},$$

where $H(0,0)$ is the Hessian matrix of g at $(0,0)$. Then $\mathbf{a}$ satisfies the following system:

$$\mathbf{a}^T H(\mathbf{0})\mathbf{a} = 0$$
$$a_1q_1 + a_2q_2 = \Delta_p.$$

If the tangent cone has a more complex structure (such as when the dimension of the null space of the Jacobian is more than 1), we need to introduce more variables a_i and more derivatives to determine the tangent cone.

Therefore, we summarize the AHTBD method as follows and outline the flow chart in Fig. 4.3:

1. For a given initial point $(\mathbf{u}, p)$ on a solution path and a maximum step-size, solve the augmented system (4.1) to track along the path;

2. If the augmented system (4.1) becomes ill-conditioned, the inflation process is introduced;

3. Near the bifurcation point, the PSE interpolation is used to approximate the bifurcation point;

4. At the bifurcation point, the tangent cone is computed to determine the different tracking solution branches, and then repeat the first step for each path.

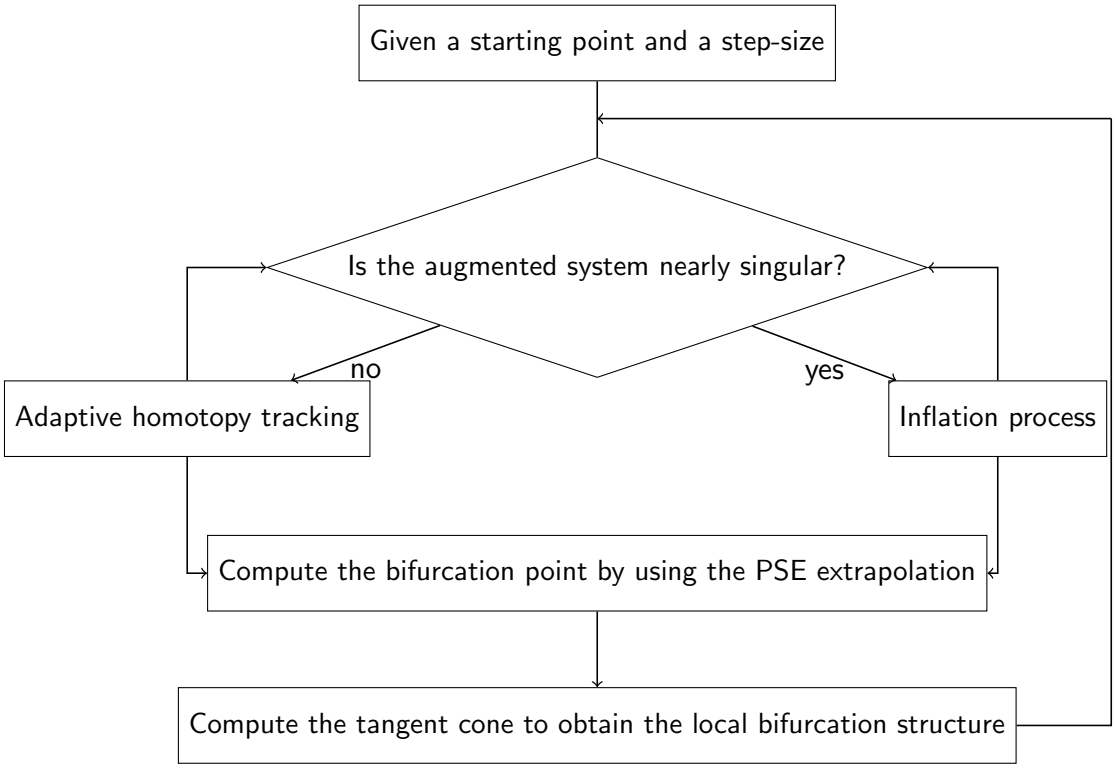


Figure 4.3: The flow chart of the AHTBD method.

4.3 Numerical Results

In this section, we apply the AHTBD method to several examples, ranging from a single equation to a system of nonlinear PDEs, to show its efficiency. Both the AHTBD method and the traditional homotopy tracking method are implemented and compared on Matlab. The traditional homotopy tracking has been implemented in various packages such as Bertini [48], HOM4PS [82], PHCpack [83] and others to handle the bifurcations. Among these existing software, Bertini has more freedom to compute the bifurcations due to the adaptive multi-precision path tracking [46] and the parallel endgame [47]. To fairly compare the AHTBD

method with the traditional homotopy tracking, we will implement both methods on Matlab.

4.3.1 Examples with complex bifurcation structures

In this subsection, we will use the AHTBD method to compute several examples with complex bifurcation structures; namely, the bifurcation point is computed first by using the adaptive homotopy tracker, and then the tangent cone algorithm is used to obtain different solution branches.

Example 1: Given

$$F(x,p) = (x-p)^4 + (x-p)(x+p), \tag{4.9}$$

we have a bifurcation point at $p = 0$. In order to compute the local bifurcation diagram at $p = 0$, we start from a point $x = 1$ and $p = 1$ to track along a solution path with the step-size $h = -0.1$. When it is close to the bifurcation, namely, $\lambda_{min} < 0.1$, we use the PSE to approximate the bifurcation point. Afterwards the tangent cone is computed: since the Jacobian F_x and the derivative F_p are both 0, the null space for $A = [F_x, F_p]$ is span$\{(0,1)^T, (1,0)^T\}$ and the null space of A^T is span$\{1\}$. Then two tangent directions are obtained, $(1,1)^T$ and $(-1,1)^T$. By setting different step-sizes, for example $h = \pm 0.1$, and choosing a tangent direction, we obtain a solution on each branch. Starting from this point, the adaptive homotopy algorithm is employed to continue tracking (see Fig. 4.4).

Example 2: The following equation represents two intersecting circles that imply complex bifurcation structures shown in Fig. 4.5:

$$F(x,p) = (x^2 + p^2 - 1)((x-1)^2 + p^2 - 1). \tag{4.10}$$

We start to track along a solution path from point $(\frac{1}{2}, \frac{\sqrt{3}}{2})$ with different tracking directions (blue point in Fig. 4.5). Fig. 4.5 shows the AHTBD tracking process with the step-size $|h| = 0.1$. It is clearly seen that the tracking is almost uniform even though there are two bifurcation points. Table 4.1 shows the comparison between the AHTBD and traditional homotopy methods when the tracking starts at point $(\frac{1}{2}, \frac{\sqrt{3}}{2})$ and ends when reaching or passing the turning point where $|p| =$

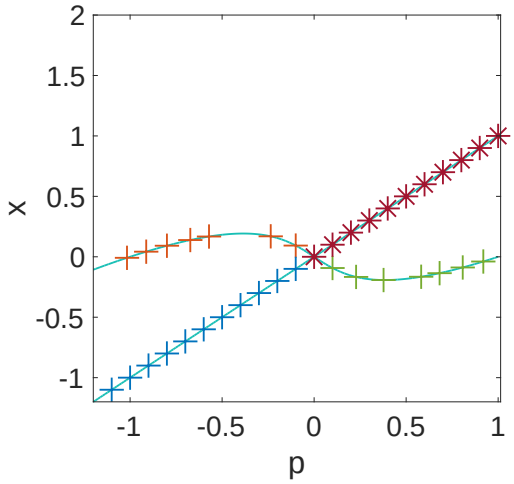


Figure 4.4: Local bifurcation diagram of (4.9): starting from the lower branch (blue points), we compute the bifurcation point first by using the PSE interpolation and then compute the tangent cone to obtain the other solution branches (green, red, and orange points).

1. The two tables have the same starting point, while the tracking direction is different. Although the traditional homotopy method may have higher accuracy for the bifurcation point, it takes many more steps to reach the end point than the AHTBD method. Moreover, the AHTBD method can pass the turning point easily (see Table 4.1 for $h = -0.1$), while the traditional method stagnates at the turning point.

4.3.2 An example of a system of nonlinear PDEs

We apply the AHTBD method to a system of nonlinear PDEs to model two species: consider a competition between two species that are ecologically identical except in their dispersal mechanisms. Let $u = u(x)$, $v = v(x)$ denote the densities of two competing species at location x. Then the study of the interaction between a resident phenotype (u) with an invader phenotype (v) can be modeled by the following system:

$$\begin{cases} \nabla(d\nabla u - \alpha u \nabla m) &= -u(m-u) & \text{in } \Omega, \\ \nabla(d\nabla v - \beta v \nabla m) &= -v(m-u) & \text{in } \Omega, \\ d\frac{\partial u}{\partial n} - \alpha u \frac{\partial m}{\partial n} &= d\frac{\partial v}{\partial n} - \beta v \frac{\partial m}{\partial n} = 0 & \text{on } \partial\Omega. \end{cases} \tag{4.11}$$

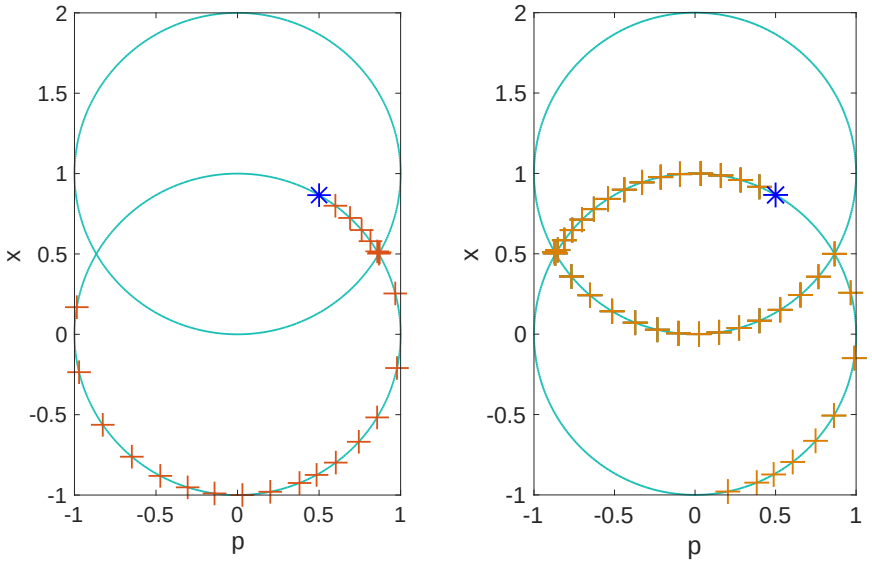


Figure 4.5: Local bifurcation diagram of (4.10). The AHTBD method is used to track from the blue point to the left and right directions.

Here $m(x)$ is the per-capita growth rate, which represents the same resources that two species are competing for. To reflect the heterogeneity of the environment, we assume that $m(x)$ is a nonconstant function to reflect the quality and quantity of resources available at the location x. In Eq. (4.11), d is two species' common random dispersal rates, and α, β are their rates of directed movement upward along the resource gradient. The boundary condition is of a no-flux type, i.e., there is no net movement across the boundary. The solution behavior of this model has been studied well in [84, 85, 86, 87]: when $\alpha = \beta$, two species co-exist, $u = v$. Bifurcation, the so-called evolutionarily stable strategy (ESS), happens on the diagonal $\alpha = \beta$, and the behavior of the solution near the bifurcation point is described in [84, 85, 87]. In reality, it is interesting to find out what happens for the bifurcation branch away from the bifurcation point, and this is where the numerical computation is needed: to find the population densities u and v as α and β moves far away from the ESS. Given $m(x) = 1 + x$, a unique positive solution of u is defined by (4.11), namely, $\tilde{u} = \tilde{u}(d, \alpha)$. By standard theory, if some rare population v is introduced into the resident population u at equilibrium (i.e. $u \equiv \tilde{u}$), then the initial (exponential) growth rate of the population of v is given

		h=0.1		h=0.05	
		Trial-and-error	AHTBD	Trial-and-error	AHTBD
# of steps		50	10	60	17
bifurcation	x	0.5002	0.5020	0.5002	0.5033
	p	0.8659	0.8671	0.8659	0.8702
endpoint	x	0.0128	−0.2097	0.0128	−0.1020
	p	0.9999	0.9778	0.9999	0.9948
		h=-0.1		h=-0.05	
		Trial-and-error	AHTBD	Trial-and-error	AHTBD
# of steps		62	19	82	32
bifurcation	x	0.5002	0.5026	0.5002	0.4831
	p	−0.8659	−0.8675	−0.8659	−0.8756
endpoint	x	0.0080	−0.1637	0.0080	0.0054
	p	−1.0000	−0.9865	1.0000	−1.0000

Table 4.1: Comparisons between AHTBD and trial-and-error tracking methods along the branches shown in Fig 4.5 with different step-sizes for h.

by λ, where $\lambda = \lambda(\alpha, \beta; d)$ is the principal eigenvalue of the problem

$$\begin{cases} \nabla \cdot (d\nabla\varphi - \beta\varphi\nabla m) + (m - \tilde{u}(d,\alpha))\varphi = \lambda\varphi & \text{in } \Omega, \\ d\frac{\partial\varphi}{\partial n} - \beta\varphi\frac{\partial m}{\partial n} = 0 & \text{on } \partial\Omega, \end{cases} \tag{4.12}$$

where the positive principal eigenfunction $\varphi = \varphi(\alpha, \beta; d)$ is uniquely determined by the normalization

$$\int_\Omega \varphi(\alpha, \beta; d) = 1. \tag{4.13}$$

In particular, when $\alpha = \beta$, we have $\varphi(\alpha, \alpha; d) = \tilde{u}$ and $\lambda(\alpha, \alpha; d) \equiv 0$ for any d, α which implies that two species u and v are identical when $\alpha = \beta$.

When we couple (4.11) and (4.13) together and discretize the system by using

the finite difference method, we have the following coupled system:

$$\mathbf{F}(\beta,\mathbf{u},\mathbf{v};\alpha) := \begin{pmatrix} \frac{2d}{h^2}u_2 - (\frac{2d}{h^2}+\frac{2\alpha}{h}+\frac{\alpha^2}{d})u_1 + u_1(m_1-u_1) \\ \frac{d}{h^2}(u_{i+1}-2u_i+u_{i-1}) - \frac{\alpha}{2h}(u_{i+1}-u_{i-1}) + u_i(m_i-u_i) \\ (-\frac{2d}{h^2}+\frac{2\alpha}{h}-\frac{\alpha^2}{d})u_N + \frac{2d}{h^2}u_{N-1} + u_N(m_N-u_N) \\ \frac{2d}{h^2}v_2 - (\frac{2d}{h^2}+\frac{2\beta}{h}+\frac{\beta^2}{d})v_1 + v_1(m_1-u_1) \\ \frac{d}{h^2}(v_{i+1}-2v_i+v_{i-1}) - \frac{\beta}{2h}(v_{i+1}-v_{i-1}) + v_i(m_i-u_i) \\ (-\frac{2d}{h^2}+\frac{2\beta}{h}-\frac{\beta^2}{d})v_N + \frac{2d}{h^2}v_{N-1} + v_N(m_N-u_N) \\ (\frac{v_1}{2}+v_2+\cdots+v_{N-1}+\frac{v_N}{2})h - 1 \end{pmatrix} = 0. \tag{4.14}$$

For any given α_0, $\mathbf{u}_0$ is solved by the discretization of (4.11). Then $\mathbf{u}_0$, $\beta_0 = \alpha_0$, $\mathbf{v}_0 = \frac{\mathbf{u}_0}{\int_\Omega \mathbf{u}_0}$ is a solution of $\mathbf{F}(\beta,\mathbf{u},\mathbf{v};\alpha) = 0$. Given initial values $(\beta_0,\mathbf{u}_0,\mathbf{v}_0,\alpha_0)$, we track along the diagonal branch $\alpha = \beta$ using α as a parameter. For our choice of $m(x)$, there is only one bifurcation. We applied the AHTBD method to track $\mathbf{F}(\beta,\mathbf{u},\mathbf{v};\alpha) = 0$, which is shown in Fig. 4.6 by starting with $\alpha_0 = 0.01$ and ending with $\alpha_0 > 0.3$. We also compared the AHTBD method with the traditional trial-

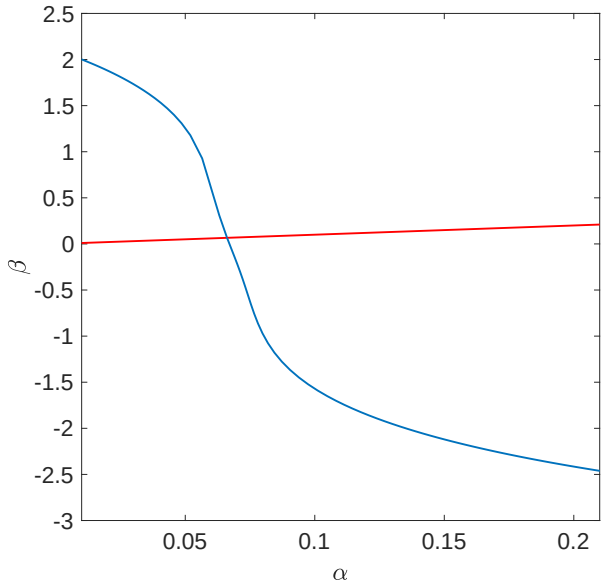


Figure 4.6: Diagram of α-β by tracking $\mathbf{F}(\beta,\mathbf{u},\mathbf{v};\alpha) = 0$ with respect to α.

and-error tracking method in Tables 4.2 & 4.3 and demonstrated that the AHTBD method is faster than the traditional homotopy tracking method for the nonlinear

PDE example.

h	Diagonal branch		Non-diagonal branch	
	Trial-and-error	AHTBD	Trial-and-error	AHTBD
0.01	88 steps(42.2808s)	25 steps(15.5970s)	88 steps(48.5956s)	26 steps(16.9497s)
0.02	70 steps(33.2463s)	16 steps(10.8033s)	70 steps(40.5963s)	15 steps(10.2496s)

Table 4.2: Comparison between the AHTBD method and the traditional trial-and-error tracking with different step-sizes for h (the number of grid points $N = 320$).

N	Diagonal branch		Non-diagonal lower branch	
	Trial-and-error	AHTBD	Trial-and-error	AHTBD
80	85 steps(5.7142s)	28 steps(2.7299s)	85 steps(6.1369s)	26 steps(2.4906s)
160	96 steps(17.9011s)	29 steps(6.2515s)	96 steps(19.1336s)	28 steps(6.6682s)
320	88 steps(42.2808s)	25 steps(15.5970s)	88 steps(48.5956s)	26 steps(16.9497s)

Table 4.3: Comparison between the AHTBD method and the traditional trial-and-error tracking for number of grid points N (the step-size is $h = 0.01$).

Chapter 5

A stochastic homotopy tracking algorithm for parametric systems of nonlinear equations

5.1 Stochastic homotopy continuation method

In section 4.1, when $\mathbf{F}_{\mathbf{u}}(\mathbf{u}, p)$ becomes singular, the solution path yields different types of bifurcations [43]. Then the numerical homotopy tracking could become very inefficient. In order to solve this numerical issue, a trial-and-error homotopy tracking method [43, 45] and an adaptive homotopy tracking method [25] have been developed to control the stepsize of p. However, the computational cost could still be very expensive when the homotopy tracking method is applied to the large-scale nonlinear systems due to the slow tracking near the singularity.

To address this challenge, we propose to solve a stochastic version of the Davidenko differential equation by introducing a noise term, namely

$$\begin{cases} \mathbf{F}_{\mathbf{u}}\big(\mathbf{u}(p,\omega),p\big)d\mathbf{u}(p,\omega) + \mathbf{F}_{p}(\mathbf{u}(p,\omega),p)dp = \mathbf{g}(\mathbf{u}(p,\omega),p)dW(p,\omega), \\ \mathbf{u}(a,\omega) = \mathbf{u}_0, \end{cases} \tag{5.1}$$

where ω is a random variable and possesses the initial condition $\mathbf{u}(a,\omega) = \mathbf{u}_0$ with probability one and $dW(p,\omega)$ denotes differential form of the Brownian motion [88]. Then, in this case, the solution path can avoid the singularity with probability one

(See Fig. 5.1 for an illustration).

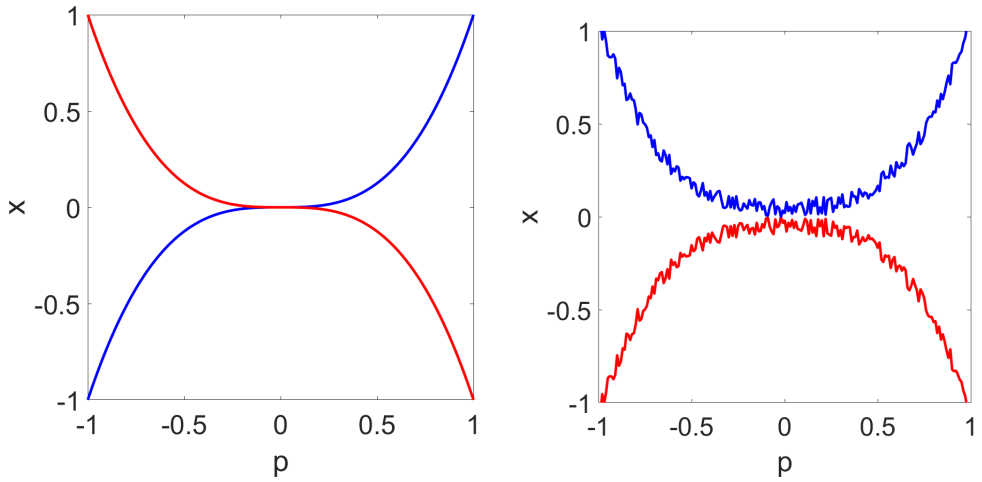


Figure 5.1: An illustration example, $x^2 - p^6 = 0$, has two solution paths $x = \pm p^3$ and one bifurcation point at $p = 0$. The traditional homotopy tracking (**Left**) hits the bifurcation point while the stochastic tracking (**Right**) can avoid the bifurcation point by tracking $x = \pm(p^3 + \xi)$, where $\xi \sim \mathcal{N}(0, 0.1)$.

In order to integrate the idea of the stochastic differential equation into the homotopy tracking, we track the solution $\mathbf{u}(p)$ from $p = a$ to $p = b$ with a stepsize Δp. Then for each $p_k = a + \Delta p \cdot k$, we solve the stochastic system below

$$\tilde{\mathbf{F}}(\mathbf{u}, p_k) = \begin{bmatrix} F_1(\mathbf{u}, p_k) \\ \vdots \\ F_{i-1}(\mathbf{u}, p_k) \\ \mathbf{u}(j) - \tilde{\mathbf{u}}_{k-1}(j) \\ F_{i+1}(\mathbf{u}, p_k) \\ \vdots \\ F_n(\mathbf{u}, p_k) \end{bmatrix} := \mathbf{F}(\mathbf{u}, p_k, \xi = (i, j)) = \mathbf{0}, \tag{5.2}$$

where $\tilde{\mathbf{u}}_{k-1}$ is the solution from previous step and $\tilde{\mathbf{F}} : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$ can be viewed as randomly chooses $n - 1$ equations from $\mathbf{F}(\mathbf{u}, p_k)$ and replaces F_i by $\mathbf{u}(j) - \tilde{\mathbf{u}}_{k-1}(j)$. Here the random variable ξ follows the uniform distribution, namely $\mathbb{P}(\xi = (i, j)) = \frac{1}{n^2}$, and quantifies the perturbations to the original system $\mathbf{F}(\mathbf{u}, p_k)$. More generally, we can randomly replace m $(1 \leq m \leq n)$ equations of $\mathbf{F}(\mathbf{u}, p)$ by $\mathbf{u}(\mathcal{J}) = \tilde{\mathbf{u}}_{k-1}(\mathcal{J})$ where $\mathcal{J} = (\mathcal{J}_1, \cdots, \mathcal{J}_m)$ is a m index. Then we define the s-th

equation of $\tilde{\mathbf{F}}$ as

$$\tilde{\mathbf{F}}_s(\mathbf{u}, p_k) = \begin{cases} \mathbf{F}_s(\mathbf{u}, p_k), & s \notin \mathcal{I} \\ \mathbf{u}(\mathcal{J}_c) - \tilde{\mathbf{u}}_{k-1}(\mathcal{J}_c), & s \in \mathcal{I} \text{ and } \mathcal{I}_c = s \end{cases}, \tag{5.3}$$

where $\mathcal{I} = (\mathcal{I}_1, \cdots, \mathcal{I}_m)$ stands for randomly choosing m equations. If $s \in \mathcal{I}$, then we find c such that $\mathcal{I}_c = s$ and replace the s-th equation by the previous value, namely, $\mathbf{u}(\mathcal{J}_c) - \tilde{\mathbf{u}}_{k-1}(\mathcal{J}_c)$. Here $\mathcal{I}$ and $\mathcal{J}$ are randomly drawn from the uniform distribution, namely $\mathbb{P}(\mathcal{I}, \mathcal{J}) = \frac{1}{(C_n^m)^2}$. We denote the set of all possible m indexes as $\mathcal{M}$.

Finally, we summarize the stochastic homotopy tracking algorithm in **Algorithm 5**. In this algorithm, we increase the number of random equations, m, if there is no solution to the stochastic system $\tilde{\mathbf{F}}(\mathbf{u}, p_{k+1}) = 0$. This is equivalent to perform a larger perturbation to the original system by solving fewer equations. Similarly, we could also increase the perturbation by setting an adaptive tolerance for $\|\mathbf{F}(\tilde{\mathbf{u}}_{k+1}, p_{k+1})\| < TOL$ by fixing the number of randomly choosing equation, m.

Algorithm 5 The pseudocode of the stochastic homotopy tracking algorithm.

Input: A step-size Δp, a threshold TOL, and a start point $(\tilde{\mathbf{u}}_0, p_0)$.
Output: A nearby solution path $(\tilde{\mathbf{u}}_k, p_k)_{k=1}^N$.
for $k = 0, \cdots, N$ **do**
  Set $m = 1$;
  Randomly choose $n - m$ equations and $n - m$ variables to form the stochastic system $\tilde{\mathbf{F}}(\mathbf{u}, p_{k+1})$ (5.2);
  Solve $\tilde{\mathbf{F}}(\mathbf{u}, p_{k+1}) = \mathbf{0}$ using the predictor-corrector method;
  if $\|\tilde{\mathbf{F}}(\tilde{\mathbf{u}}_{k+1}, p_{k+1})\| < TOL$ **then**
    Update the solution sequence;
  else
    Increase m and solve the stochastic system again.
  end if
end for

5.2 Convergence Analysis

We employ the Euler predictor and the Newton corrector [66] for the homotopy tracking algorithm: Given a solution $(\mathbf{u}_0, p_0)$ on the path, that is, $\mathbf{F}(\mathbf{u}_0, p_0) = 0$, an Euler predictor step gives

$$\mathbf{F}_{\mathbf{u}}(\mathbf{u}_0, p_0)\Delta\mathbf{u} = -\mathbf{F}_p(\mathbf{u}_0, p_0)\Delta p, \tag{5.4}$$

and then letting $\mathbf{u}_1 = \mathbf{u}_0 + \Delta\mathbf{u}$; The Newton corrector reads

$$\mathbf{F}_{\mathbf{u}}(\mathbf{u}_1, p_1)\Delta\mathbf{u} = -\mathbf{F}(\mathbf{u}_1, p_1). \tag{5.5}$$

Then we repeat this correction until $(\mathbf{u}_1, p_1)$ is on the path. The predictor-corrector method for the stochastic homotopy tracking method needs to replace $\mathbf{F}$ by $\tilde{\mathbf{F}}$ defined in (5.2) with the corresponding derivatives below:

$$\begin{aligned}
\tilde{\mathbf{F}}_p(\mathbf{u}) &= \mathbf{F}_p(\mathbf{u}, \xi = (i,j)) = \mathbf{F}_p(\mathbf{u}) - \frac{\partial F_i}{\partial p}\mathbf{e}_i, \\
\tilde{\mathbf{F}}_{\mathbf{u}}(\mathbf{u}) &= \mathbf{F}_{\mathbf{u}}(\mathbf{u}, \xi = (i,j)) = \mathbf{F}_{\mathbf{u}}(\mathbf{u}) - \mathbf{e}_i\frac{\partial F_i}{\partial \mathbf{u}}(\mathbf{u}) + E_{ij},
\end{aligned} \tag{5.6}$$

where E_{ij} is a matrix with all zero elements except the (i,j)-th element as one. For the general stochastic system (5.3) with m random equations, we have $\xi = (\mathcal{I}, \mathcal{J})$ and

$$\begin{aligned}
\tilde{\mathbf{F}}_p(\mathbf{u}) &= \mathbf{F}_p(\mathbf{u}) - \sum_{i\in\mathcal{I}}\frac{\partial F_i}{\partial p}\mathbf{e}_i \triangleq \mathbf{F}_p(\mathbf{u}) - C(\mathbf{u}, \xi), \\
\tilde{\mathbf{F}}_{\mathbf{u}}(\mathbf{u}) &= \mathbf{F}_{\mathbf{u}}(\mathbf{u}) - \sum_{i\in\mathcal{I}}\mathbf{e}_i\frac{\partial F_i}{\partial \mathbf{u}}(\mathbf{u}) + \sum_{i\in\mathcal{I}, j\in\mathcal{J}} E_{ij} \triangleq \mathbf{F}_{\mathbf{u}}(\mathbf{u}) - S(\mathbf{u}, \xi).
\end{aligned}$$

We also define the tensor $\nabla\mathbf{F}_{\mathbf{u}}(\mathbf{u})$ as follows:

$$[\nabla\mathbf{F}_{\mathbf{u}}(\mathbf{u})]_{ijk} = [\nabla^2\mathbf{F}_i(\mathbf{u})]_{jk}, \quad i,j,k \in \{1, 2, \cdots, n\}$$

and define the multiplication of the tensor with a vector, $\mathbf{b} \in \mathbb{R}^n$, as

$$[\nabla\mathbf{F}_{\mathbf{u}}(\mathbf{u})\mathbf{b}]_{ij} = \sum_{k=1}^{n}[\nabla^2\mathbf{F}_i(\mathbf{u})]_{jk}\mathbf{b}_k.$$

Then $\|\nabla \mathbf{F}_{\mathbf{u}}(\mathbf{u})\| = \max_{1\leq i\leq n} \|\nabla^2 \mathbf{F}_i(\mathbf{u})\|$. In this section, we analyze that the solution path guided by the stochastic homotopy tracking is closed to the path guided by the traditional homotopy tracking under certain conditions. This analysis is performed for Euler's prediction in Theorem 5.2.1 and for Newton's correction in Theorem 5.2.2. The proof of Theorem 5.2.1 and 5.2.2 can be found in Appendix 5.

Theorem 5.2.1 (Euler's Prediction)**.** *Suppose* $\mathbf{u}_0$ *and* $\tilde{\mathbf{u}}_0$ *are the start points for the original system* $\mathbf{F}$ *and the stochastic system* $\tilde{\mathbf{F}}$ *respectively. If we have the following assumptions*

- $\mathbf{F}_{\mathbf{u}}$ *and* $\tilde{\mathbf{F}}_{\mathbf{u}}$ *are invertible and differentiable and*

$$\|\mathbf{F}_{\mathbf{u}}\| \leq L_{\mathbf{u}}, \|\mathbf{F}_{\mathbf{u}}^{-1}\| \leq M_{\mathbf{u}} \text{ and } \|\tilde{\mathbf{F}}_{\mathbf{u}}^{-1}\| \leq M_{\mathbf{u}};$$

- $\nabla \mathbf{F}_{\mathbf{u}}$, $\nabla \tilde{\mathbf{F}}_{\mathbf{u}}$ *are continuous;*
- $\mathbf{F}_p$ *and* $\tilde{\mathbf{F}}_p$ *are differentiable and* $\|\mathbf{F}_p\| \leq M_p$*;*
- $\nabla \mathbf{F}_p$ *is continuous and* $\|\nabla \mathbf{F}_p\| \leq L_p$,

then we have

$$\|\mathbb{E}(\mathbf{u}_N - \tilde{\mathbf{u}}_N)\|^2 \leq CS_1 \|\mathbb{E}(\mathbf{u}_0 - \tilde{\mathbf{u}}_0)\|^2 + CS_2 \frac{m^2}{n^2} + \mathcal{O}(\frac{m^2 \Delta p}{n^2}), \tag{5.7}$$

where CS_1 *and* CS_2 *are constants.*

Remark 6. *For large-scale nonlinear parametric problems, when* n *is large, the error caused by the stochastic homotopy tracking becomes very small due to the* $O(\frac{1}{n^2})$ *estimate for any given* m*. Therefore, the Euler's prediction of the stochastic homotopy tracking stays closed to the prediction by the traditional homotopy tracking.*

Theorem 5.2.2 (Newton's correction)**.** *Suppose* $\mathbf{u}_k^i$ *and* $\tilde{\mathbf{u}}_k^i$ *are* i*-th Newton's iterations for solving* $\mathbf{F}(\mathbf{u}, p_k) = 0$ *and* $\tilde{\mathbf{F}}(\mathbf{u}, p_k) = 0$ *respectively. If we have the following assumptions*

- $\mathbf{F}_{\mathbf{u}}$ *and* $\tilde{\mathbf{F}}_{\mathbf{u}}$ *are invertible and differentiable and*

$$\|\mathbf{F}_{\mathbf{u}}^{-1}\| \leq M_{\mathbf{u}} \text{ and } \|\tilde{\mathbf{F}}_{\mathbf{u}}^{-1}\| \leq M_{\mathbf{u}};$$

- $\nabla \mathbf{F}_{\mathbf{u}}$, $\nabla \tilde{\mathbf{F}}_{\mathbf{u}}$ *are continuous and*

$$\|\nabla \mathbf{F}_{\mathbf{u}}\| \leq K_{\mathbf{u}} \text{ and } \|\nabla \tilde{\mathbf{F}}_{\mathbf{u}}\| \leq K_{\mathbf{u}};$$

- *The initial guesses* $\mathbf{u}_k^0$ *and* $\tilde{\mathbf{u}}_k^0$ *are in a small neighborhood of the real solutions* $\mathbf{u}_k$ *and* $\tilde{\mathbf{u}}_k$,

then we have

$$\lim_{i \to \infty} \|\mathbb{E}(\mathbf{u}_k^i - \tilde{\mathbf{u}}_k^i)\| \leq \|\mathbb{E}(\mathbf{u}_k - \tilde{\mathbf{u}}_k)\|. \tag{5.8}$$

Remark 7. *The difference of Newton's corrections between the traditional and the stochastic homotopy tracking is bounded by the difference of the solutions between the original and the stochastic systems which is pretty small for large scale systems. Thus Newton's corrections by two different homotopy tracking algorithms are near each other.*

5.3 Numerical Examples

In this section, we compare the stochastic homotopy tracking with the traditional homotopy tracking on the Matlab platform. We use the stopping criteria of $\Delta p < 10^{-7}$ for the traditional homotopy tracking method to detect the bifurcation points.

5.3.1 Example 1

We first consider a homotopy setup for solving a system of polynomial equations with the total degree start system, namely,

$$H(x, y, z; t) = t \begin{bmatrix} x^2 + y^2 + z^2 - 1 \\ x^2 - y^2 - z^2 \\ x + y + z \end{bmatrix} + (1 - t) \begin{bmatrix} x^2 - 1 \\ y^2 - 1 \\ z - 1 \end{bmatrix} = 0. \tag{5.9}$$

When $t = 0$, the solutions of $H(x, y, z; 0) = 0$ are known explicitly. The solutions of the target system, $H(x, y, z; 1) = 0$, are revealed by tracking t from 0 to 1 on the complex field. There are four solution paths needed to track from 0 to 1

	Traditional homotopy tracking	Stochastic homotopy tracking
Branch 1	1.05s (259 steps)	0.24s (11 steps)
Branch 2	0.59s (221 steps)	0.24s (11 steps)
Branch 3	0.91s (246 steps)	0.17s (11 steps)
Branch 4	0.84s (237 steps)	0.18s (11 steps)

Table 5.1: Timing comparison between traditional and stochastic homotopy tracking methods on different branches shown in Fig. 5.2.

for $\mathbf{u} = [x, y, z]^T$ shown in Fig. 5.2. The solid lines indicate the solution path of $x(t)$ for the traditional homotopy tracking, while the dashed lines represent the solution paths guided by stochastic homotopy tracking. The timing data is

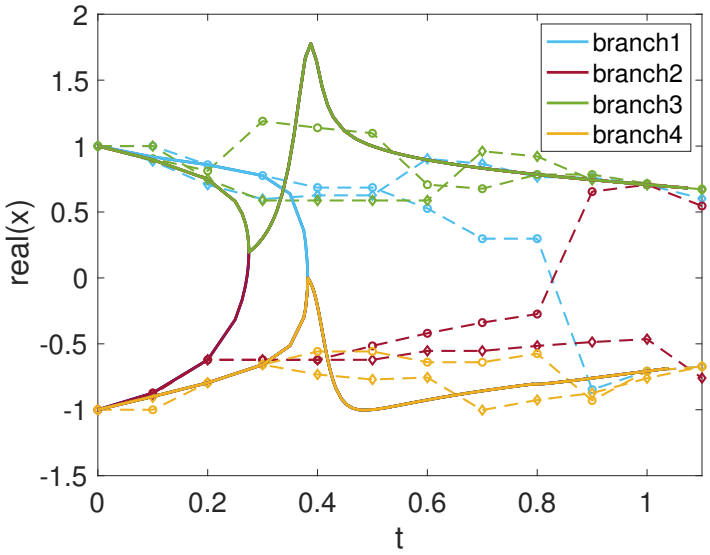


Figure 5.2: An illustration of the stochastic homotopy tracking method for tracking the solution path $x(t)$ of (5.9) on four solution branches. The solid lines are for the traditional homotopy tracking while the dashed lines are for stochastic homotopy tracking.

compared between two tracking methods is shown in Table 5.1 with $\Delta t = 0.1$ which clearly demonstrates that the stochastic homotopy tracking method is more efficient with fewer steps from $t = 0$ to $t = 1$.

n	Traditional	Stochastic
10	0.027s (24 steps)	0.013s (12 steps)
20	0.051s (22 steps)	0.022s (12 steps)
40	0.141s (30 steps)	0.076s (12 steps)
80	0.530s (29 steps)	0.272s (12 steps)

Table 5.2: Comparison between the traditional and the stochastic homotopy tracking with different number of grid points n.

5.3.2 Example 2

We consider the following 1D nonlinear boundary value problem.

$$\begin{cases} u_{xx} = u^2(u^2 - p), \\ u_x(0) = 0, u(1) = 0, \end{cases} \tag{5.10}$$

where p is the parameter. The multiple solutions become more as p gets larger. Therefore, turning points happen when p is tracked. We discretize (5.10) by using the finite difference method and have the following discretized polynomial system

$$\mathbf{F}(\mathbf{u}, p) := \begin{pmatrix} \frac{1}{h^2}(\mathbf{u}_1 - 2\mathbf{u}_1 + \mathbf{u}_2) - \mathbf{u}_1^2(\mathbf{u}_1^2 - p) \\ \frac{1}{h^2}(\mathbf{u}_{i-1} - 2\mathbf{u}_i + \mathbf{u}_{i+1}) - \mathbf{u}_i^2(\mathbf{u}_i^2 - p) \\ \frac{1}{h^2}(\mathbf{u}_{n-2} - 2\mathbf{u}_{n-1}) - \mathbf{u}_{n-1}^2(\mathbf{u}_{n-1}^2 - p) \end{pmatrix} = 0. \tag{5.11}$$

where $h = \frac{1}{n}$, $\mathbf{u} \in \mathbb{R}^{n-1}$ and $\mathbf{u}_i = u(\frac{i}{n})$ for $i = 1, 2, \cdots, n-1$. We track the parameter p from 14 down to 2 with $\Delta p = -1$ for one solution path with a turning point shown in Fig 5.3. Since the lower solution branch is close to the constant solution branch (the red line in Fig. 5.3, the stochastic homotopy tracking just switches to the constant solution branch when it is close to the turning point. Moreover, the stochastic homotopy tracking is much efficient than the traditional method by comparing the average tracking time shown in Table 5.2 for different grid points n. For the upper solution branch, since no nearby solution branch exists, the stochastic homotopy tracking has to deal with a stochastic system with a large perturbation, namely increasing m in **Algorithm 5**.

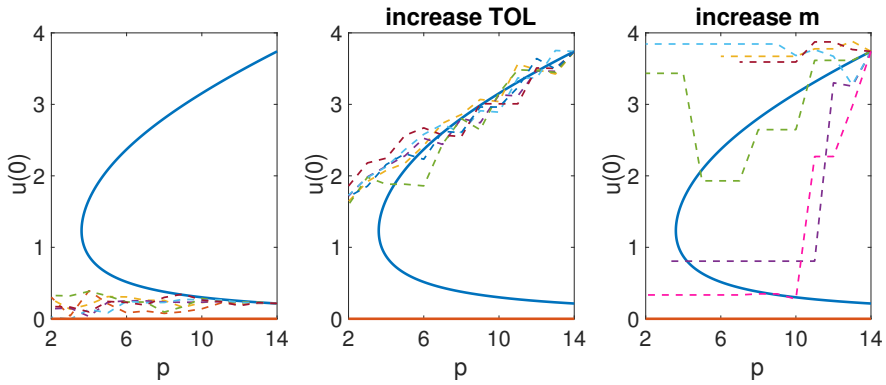


Figure 5.3: An illustration of stochastic homotopy tracking for tracking (5.10) with respect to p from 14 to 2. The lower solution branch is switched to the constant solution branch (**Left**); The upper solution branch needs a large TOL (**Middle**) or a large m (**Right**) in **Algorithm 5**.

5.3.3 Example 3

Last we consider the Schnakenberg model which is a system of partial differential equations shown below [89]:

$$\begin{cases} \dfrac{\partial u}{\partial t} = \Delta u + \eta(a - u + u^2 v), \\ \dfrac{\partial v}{\partial t} = d\Delta v + \eta(b - u^2 v), \end{cases} \tag{5.12}$$

where u is an activator and v is a substrate. The steady-state system of (5.12) with non-flux boundary condition has been well-studied in [89] and shown multiple steady-state solutions and the bifurcation structure to the diffusion parameter d. In this example, we consider the discretized steady-state system on a 1D domain $x \in [0, 1]$ with no-flux boundary conditions:

$$\mathbf{F}(\mathbf{u}, \mathbf{v}, d) := \begin{pmatrix} \frac{1}{h^2}(2\mathbf{u}_2 - 2\mathbf{u}_1) + \eta(a - \mathbf{u}_1 + \mathbf{u}_1^2\mathbf{v}_1) \\ \frac{1}{h^2}(\mathbf{u}_{i-1} - 2\mathbf{u}_i + \mathbf{u}_{i+1}) + \eta(a - \mathbf{u}_i + \mathbf{u}_i^2\mathbf{v}_i) \\ \frac{1}{h^2}(2\mathbf{u}_n - 2\mathbf{u}_{n+1}) + \eta(a - \mathbf{u}_{n+1} + \mathbf{u}_{n+1}^2\mathbf{v}_{n+1}) \\ \frac{d}{h^2}(2\mathbf{v}_2 - 2\mathbf{v}_1) + \eta(b - \mathbf{u}_1^2\mathbf{v}_1) \\ \frac{d}{h^2}(\mathbf{v}_{i-1} - 2\mathbf{v}_i + \mathbf{v}_{i+1}) + \eta(b - \mathbf{u}_i^2\mathbf{v}_i) \\ \frac{d}{h^2}(2\mathbf{v}_n - 2\mathbf{v}_{n+1}) + \eta(b - \mathbf{u}_{n+1}^2\mathbf{v}_{n+1}) \end{pmatrix} = 0. \tag{5.13}$$

where $h = \frac{1}{n}$, $\mathbf{u}, \mathbf{v} \in \mathbb{R}^{n+1}$ with $\mathbf{u}_i = u(\frac{i-1}{n})$ and $\mathbf{v}_i = v(\frac{i-1}{n})$ for $i = 1, 2, \cdots, n+1$. We introduce ghost points $\mathbf{u}_0, \mathbf{v}_0, \mathbf{u}_{n+2}$, and $\mathbf{v}_{n+2}$ at $x = 0$ and $x = 1$. The non-flux boundary conditions imply that $\mathbf{u}_0 = \mathbf{u}_2$, $\mathbf{v}_0 = \mathbf{v}_2$,$\mathbf{u}_{n+2} = \mathbf{u}_n$, and $\mathbf{v}_{n+2} = \mathbf{v}_n$.

We choose $a = 1/3,\ b = 2/3,\ \eta = 50$ and track d from 50 to 35 with different number of grid points n. As shown in Fig. 5.4, the traditional homotopy tracking method stops near the bifurcation around $d \approx 45$ with a very small tracking stepsize. However, the stochastic homotopy tracking method can avoid the bifurcation point and track down to 35. Moreover, as n goes larger, the solution path guided by the stochastic homotopy tracking gets closer to the original path. Detailed iteration comparison between two tracking methods is shown in Table 5.3 for the different number of grid points n and different tracking stepsizes Δd. It clearly shows that the stochastic homotopy tracking method becomes more efficient compared to the traditional one as the size of the system gets larger.

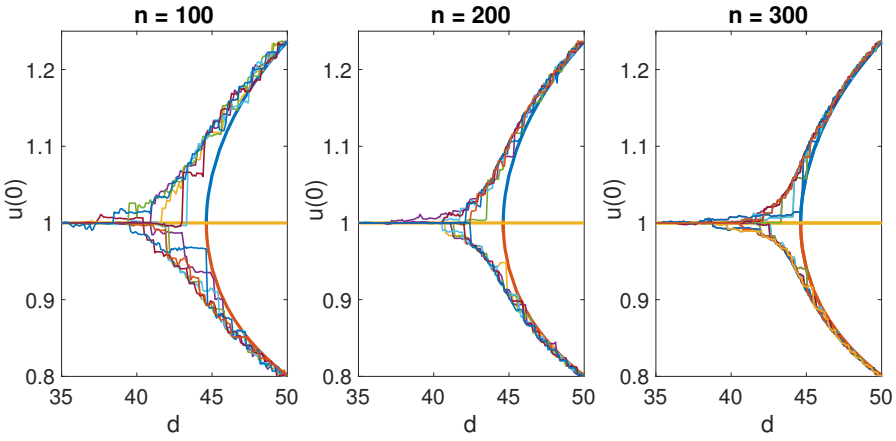


Figure 5.4: Traditional and stochastic homotopy tracking methods with different number of grid points.

		Lower branch		Upper branch	
n	Δd	Traditional	Stochastic	Traditional	Stochastic
100	-0.5	2.76s(32steps)	2.30s(31steps)	2.49s(28steps)	1.87s(31steps)
	-1	3.23s(59steps)	1.35s(16steps)	2.38s(34steps)	0.93s(16steps)
200	-0.5	12.88s(53steps)	8.83s(31steps)	10.62s(35steps)	8.93s(31steps)
	-1	9.36s(53steps)	3.08s(16steps)	7.61s(21steps)	3.88s(16steps)
300	-0.5	77.9s(90steps)	34.1s(31steps)	40.2s(34steps)	36.9s(31steps)
	-1	40.3s(90steps)	16.5s(16steps)	30.1s(34steps)	15.6s(16steps)

Table 5.3: Comparison between traditional and stochastic homotopy tracking with different number of grid points n and different step-sizes Δd.

Chapter 6

Bifurcation Analysis of a Free Boundary Model of the Plaque Formation Associated with the Cholesterol Ratio

6.1 Introduction

Atherosclerosis, known as an inflammatory disease[90, 91], is the No.1 killer of Americans. It can affect any artery in the body and most of those deaths are from heart attacks caused by fatty deposits that clog coronary arteries. These deposits, which are called plaques, consist of cholesterol, fat, and other substances[92]. As the plaque builds up, the artery wall gets thicker, which narrows the blood vessel and reduces the supply of oxygen to cells. Then the plaque may rupture and the bloodstream would carry the debris until it gets stuck, leading to the formation of thrombus. The arteries can be blocked during this process and heart attacks or strokes may occur, depending on where the plaque locates[93].

The arterial wall usually consists of three layers: the intima, media, and adventitia (see Fig. 6.1). The intima is a thin single sheet of endothelial cells. The media is composed mainly of smooth muscle cells and elastic tissue. The adventitia is the outermost connective tissue layer[94, 95]. The development of plaque begins with

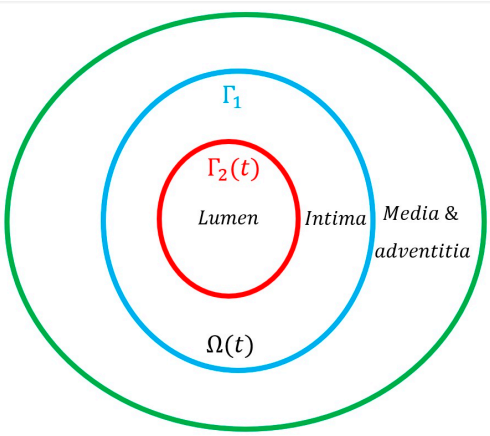


Figure 6.1: The domain of the free boundary model: $\Omega(t)$ represents the intima; the inner surface of the arterial wall, $\Gamma_2(t)$, is a free boundary; and the surface between the intima and media/adventitia, Γ_1, is fixed.

a lesion in the intima layer, initiating an inflammatory response resulting in the accumulation of LDLs[96]. Part of LDLs become oxidized LDLs by free radicals and would be ingested by macrophages differentiating from monocytes. The ingestion of large amounts of oxidized LDLs transforms macrophages into foam cells that are responsible for plaque growth. In the meanwhile, the HDLs remove cholesterol from foam cells and inhibit the oxidation of LDLs[97]. Therefore the balance of HDLs and LDLs is essential for plaque development. According to the cholesterol guideline of the AHA [98, 99], the optimal cholesterol ratio, LDL/HDL, is 3.5. A higher ratio means a higher cardiovascular risk. In other words, individuals who have a higher ratio need to work toward the optimal ratio, either by changing their lifestyles or by eating heart-healthy diets, to reduce the cardiovascular risk. In this paper, we will interpret the importance of the LDL/HDL ratio in a mathematical modeling context.

Several mathematical models have been developed to explore the relationship between cholesterol ratio and cardiovascular risk [100, 101, 102, 30]. These models characterize biological interactions among endothelial cells, monocytes, and T cells by using partial differential equations (PDEs) and address the importance of LDL

and HDL in plaque growth. Among these mathematical models, some of them are free boundary problems to describe the geometric change of the plaque in the artery [32, 30, 103]. For instance, a recent free boundary model [32] introduces a system of PDEs including LDL, HDL, macrophages, T cells, smooth muscle cells, and related cytokines and generates a "risk-map" of plaque development for any pair values of (LDL, HDL), indicating the significance of LDL and HDL in determining the growth or shrink of a plaque. Later, the effect of reverse cholesterol transport (RCT) has been added to this free boundary model [30]. Moreover, a simplified free boundary model has been analyzed theoretically on the existence of small radially symmetric stationary plaques and their stability conditions[103]. However, there is no theoretical analysis of the effect of cholesterol ratio on plaque growth for these free boundary models.

6.2 Mathematical model

We consider the plaque formation in the early stage of atherosclerosis by including the basic pathophysiology in the intima (See Fig 6.1 for the detailed domain setup). Macrophages enter the intima, $\Omega(t)$, by the chemotaxis of MCP-1 [104] and become foam cells by the uptake of oxidized LDL [105]. On the other hand, HDL removes the cholesterol from foam cells [106] which become M2 macrophages transfering back to the liver, referred to as the RCT process [107]. Then we model the density of macrophages, M, below:

$$\begin{cases} \dfrac{\partial M}{\partial t} - D\Delta M = -HM, & x \in \Omega(t), \\ \dfrac{\partial M}{\partial \mathbf{n}} = -M, & x \text{ on } \Gamma_1, \\ M = 1, & x \text{ on } \Gamma_2(t), \end{cases} \tag{6.1}$$

where H represents the concentration of HDL and HM accounts for the loss of macrophages due to the RCT process [32]. On $\Gamma_2(t)$, we use the Dirichlet boundary condition to model the recruitment of macrophages by MCP-1 and take $M = 1$ after the normalization. Since there are no macrophages in media/adventitia [32], we have $\dfrac{\partial M}{\partial \mathbf{n}} + \alpha M = 0$ on Γ_1. For simplicity, we take the flux rate $\alpha = 1$ in our model.

Plaque growth is proportional to the density of foam cells which is assumed to be a combination of LDL and macrophages in our model. Therefore we model plaque growth as

$$\nabla \cdot \mathbf{v} = LM - T, \ x \in \Omega(t), \tag{6.2}$$

where L represents the concentration of LDL and T is the clearance capacity provided by the immune system[108]: if there is too much LDL and macrophages ($LM > T$), the plaque will grow; otherwise will disappear due to the immune system. For simplicity, we treat T as a parameter in our model instead of including its dynamics.

In light of the intima's high permeability to white cells and platelets [109, 110, 111], we treat the intima as a porous medium and macrophages in the intima as a low-speed flow [30, 32] moving with a common velocity, $\mathbf{v}$. Thus the pressure, P, resulting from the movement of macrophages, follows Darcy's law $\mathbf{v} = -\nabla P$. Therefore the equation of P [30, 32] becomes:

$$\begin{cases} -\Delta P = LM - T, & x \in \Omega(t), \\ \dfrac{\partial P}{\partial \mathbf{n}} = 0 & x \text{ on } \Gamma_1, \\ P = \gamma\kappa, \ \mathbf{v_n}(t) = -\dfrac{\partial P}{\partial \mathbf{n}} & x \text{ on } \Gamma_2(t). \end{cases} \tag{6.3}$$

Since the boundary Γ_1 is fixed, we have the no-flux boundary condition. On $\Gamma_2(t)$, the pressure P is balanced by the surface-tension which is proportional to the mean curvature κ (γ is the proportionality constant, the blood pressure is considered as zero in our model for simplicity); the velocity along the normal direction $\mathbf{n}$, $\mathbf{v_n} = \mathbf{v} \cdot \mathbf{n}$ gives the free boundary moving condition [32]. Thus we summarize the free boundary model as follows

$$\begin{cases} \dfrac{\partial M}{\partial t} - D\Delta M = -HM, & x \in \Omega(t), \\ -\Delta P = LM - T, & x \in \Omega(t), \\ \dfrac{\partial P}{\partial \mathbf{n}} = 0, \dfrac{\partial M}{\partial \mathbf{n}} = -M, & x \text{ on } \Gamma_1, \\ M = 1, P = \gamma\kappa, \ \mathbf{v_n}(t) = -\dfrac{\partial P}{\partial \mathbf{n}}, & x \text{ on } \Gamma_2(t). \end{cases} \tag{6.4}$$

6.3 Radially symmetric steady-sate solutions

6.3.1 Radially symmetric solution of M

First we compute the radially symmetric steady-state solution of (6.1) by taking $\frac{\partial M}{\partial t} = 0$ and have

$$\begin{cases} M''(r) + \frac{1}{r}M'(r) - \frac{H}{D}M = 0, \\ \frac{\partial M}{\partial r}|_{r=R} = -M(R), \\ M(\rho) = 1. \end{cases} \tag{6.5}$$

By taking $z = \sqrt{\frac{H}{D}}r$, we rewrite (6.5) in terms of $u(z) = M(r)$ as

$$z^2u''(z) + zu'(z) - z^2u(z) = 0,$$

which implies

$$M_s(r) = u(z) = C_1I_0(z_r) + C_2K_0(z_r). \tag{6.6}$$

Since $I_0'(z) = I_1(z)$ and $K_0'(z) = -K_1(z)$, we solve for C_1 and C_2 by using the boundary conditions, namely,

$$\begin{cases} C_1\sqrt{\frac{H}{D}}I_1(z_R) - C_2\sqrt{\frac{H}{D}}K_1(z_R) = -(C_1I_0(z_R) + C_2K_0(z_R)), \\ C_1I_0(z_\rho) + C_2K_0(z_\rho) = 1. \end{cases} \tag{6.7}$$

Then we have

$$C_1 = \frac{\sqrt{\frac{H}{D}}K_1(z_R) - K_0(z_R)}{C(\rho, R, H, D)} \text{ and } C_2 = \frac{\sqrt{\frac{H}{D}}I_1(z_R) + I_0(z_R)}{C(\rho, R, H, D)},$$

where

$$\begin{aligned} C(\rho, R, H, D) =& (I_0(z_R)K_0(z_\rho) - I_0(z_\rho)K_0(z_R)) \\ &+ \sqrt{\frac{H}{D}}(K_1(z_R)I_0(z_\rho) + I_1(z_R)K_0(z_\rho)) \end{aligned}$$

Remark: By the maximum principle, we have $M_s(r) \geq 0$ for $\rho \leq r \leq R$, $M_s'(\rho) < 0$, and $M_s'(R) < 0$.

6.3.2 Radially symmetric solution of P

By rewriting (6.3) as $\Delta(P + \frac{DL}{H}M) = T$, we have

$$P_s(r) = -\frac{DL}{H}M_s(r) + C_3 \ln r + C_4 + \frac{1}{4}Tr^2. \tag{6.8}$$

The boundary conditions in radially symmetric case become

$$P_s(\rho) = \frac{\gamma}{\rho}, \quad \frac{\partial P_s}{\partial r}|_{r=R} = 0, \text{ and } \frac{\partial P_s}{\partial r}|_{r=\rho} = 0, \tag{6.9}$$

which are used to determine C_3, C_4, and T, namely,

$$C_3 = \frac{DL}{H}\frac{R^2\rho^2}{R^2 - \rho^2}(\frac{M_s'(\rho)}{\rho} + \frac{M_s(R)}{R}),$$

$$C_4 = \frac{\gamma}{\rho} + \frac{DL}{H} - C_3 \ln \rho - \frac{1}{4}T\rho^2,$$

and

$$T = -\frac{DL}{H}\frac{2}{R^2 - \rho^2}(RM_s(R) + \rho M_s'(\rho)). \tag{6.10}$$

For any given T, we compute ρ by solving (6.10). Therefore, the existence of ρ is critical for our model. In order to prove the existence, we solve T for any given ρ and have the following theorem.

Theorem 6.3.1. *For any given $L > 0$ and $\rho > 0$, there exists a unique $T > 0$ such that a stationary solution (M_s, P_s) is given by* (6.6) *and* (6.8)*.*

Proof. For any given ρ, it is obvious that T is uniquely determined by (6.10). Next we prove $T > 0$ by letting

$$f(r) = rM_s'(r).$$

Since

$$f'(r) = rM_s''(r) + M_s'(r) = \frac{H}{D}rM_s(r) \geq 0,$$

we have

$$f(\rho) = \rho M_s'(\rho) \leq f(R) = RM_s'(R) = -RM_s(R)$$

which implies

$$T = -\frac{DL}{H}\frac{2}{R^2-\rho^2}(f(\rho)-f(R)) \geq 0.$$

□

6.4 Bifurcation analysis and linear stability

6.4.1 The linearized system

First, we derive the linearized system of (6.4) with a perturbed domain Ω_ε to Ω, namely, $\Gamma_\varepsilon = \{r | r = \rho + \varepsilon\rho_1(\theta)\}$:

$$\begin{cases} -D\Delta M = -HM, & x \in \Omega_\varepsilon, \\ \dfrac{\partial M}{\partial \mathbf{n}} = -M, & x \text{ on } \Gamma_1, \\ M = 1, & x \text{ on } \Gamma_\varepsilon, \\ -\Delta P = LM - T, & x \in \Omega_\varepsilon, \\ \dfrac{\partial P}{\partial \mathbf{n}} = 0 & x \text{ on } \Gamma_1, \\ P = \gamma\kappa, & x \text{ on } \Gamma_\varepsilon. \end{cases} \tag{6.11}$$

By defining the following nonlinear function F based on the free boundary condition,

$$F(\rho_1, L) = \frac{\partial P}{\partial r}\Big|_{\Gamma_\varepsilon}, \tag{6.12}$$

we conclude that $\rho_1(\theta)$ induces a stationary solution if and only if $F(\rho_1, L) = 0$. Then we consider the solution of (6.11), (M, P), up to the 2nd order of ε. Justification for (6.13) can be found in Appendix B.2.

$$\begin{aligned} M(r,\theta) &= M_s(r) + \varepsilon M_1(r,\theta) + \mathcal{O}(\varepsilon^2), \\ P(r,\theta) &= P_s(r) + \varepsilon P_1(r,\theta) + \mathcal{O}(\varepsilon^2). \end{aligned} \tag{6.13}$$

Thus the boundary condition of M on Γ_ε becomes

$$\begin{aligned} 1 = M(r,\theta)|_{\Gamma_\varepsilon} &= M(\rho+\varepsilon\rho_1,\theta) \\ &= M_s(\rho+\varepsilon\rho_1+\varepsilon M_1(\rho+\varepsilon\rho_1,\theta)+\mathcal{O}(\varepsilon^2) \\ &= M_s(\rho)+\varepsilon\rho_1\frac{\partial M_s}{\partial r}(\rho)+\varepsilon M_1(\rho)+\mathcal{O}(\varepsilon^2). \end{aligned}$$

Since the mean curvature is given by

$$\kappa = \frac{2r_\theta^2 - rr_{\theta\theta}+r^2}{(r_\theta^2+r^2)^{3/2}}, \tag{6.14}$$

the linearization of κ becomes

$$\begin{aligned} \kappa|_{\Gamma_\varepsilon} =& \frac{(\rho+\varepsilon\rho_1)^2+2(\rho_\theta+\varepsilon\rho_{1\theta})^2}{((\rho+\varepsilon\rho_1)^2+(\rho_\theta+\varepsilon\rho_{1\theta})^2)^{3/2}} \\ &- \frac{(\rho+\varepsilon\rho_1)(\rho_{\theta\theta}+\varepsilon\rho_{1\theta\theta})}{((\rho+\varepsilon\rho_1)^2+(\rho_\theta+\varepsilon\rho_{1\theta})^2)^{3/2}} \\ =& \kappa_0+\varepsilon\kappa_1+\mathcal{O}(\varepsilon^2), \end{aligned} \tag{6.15}$$

where

$$\kappa_0 = \frac{2\rho_\theta^2-\rho\rho_{\theta\theta}+\rho^2}{(\rho_\theta^2+\rho^2)^{3/2}} \tag{6.16}$$

and

$$\begin{aligned} \kappa_1 =& \Big(\frac{2\rho-\rho_{\theta\theta}}{(\rho_\theta^2+\rho^2)^{3/2}} - \frac{3}{2}\frac{(\rho^2+2\rho_\theta^2-\rho\rho_{\theta\theta})2\rho}{(\rho_\theta^2+\rho^2)^{5/2}}\Big)\rho_1 \\ &+ \Big(\frac{4\rho_\theta}{(\rho_\theta^2+\rho^2)^{3/2}} - \frac{3}{2}\frac{(\rho^2+2\rho_\theta^2-\rho\rho_{\theta\theta})2\rho_\theta}{(\rho_\theta^2+\rho^2)^{5/2}}\Big)\rho_{1\theta} \\ &- \frac{\rho}{(\rho_\theta^2+\rho^2)^{3/2}}\rho_{1\theta\theta}. \end{aligned} \tag{6.17}$$

After dropping the higher order terms, we obtain the linearized system below:

$$\begin{cases} \Delta M_1 = \dfrac{H}{D} M_1 \quad \text{in} \quad \Omega, \\ M_1(\rho) = -\rho_1 \dfrac{\partial M_s(\rho)}{\partial r}, \\ \dfrac{\partial M_1}{\partial r}(R) = -M_1(R), \\ \Delta P_1 = -L M_1 \quad \text{in} \quad \Omega, \\ P_1(\rho) = -\dfrac{\gamma}{\rho^2}(\rho_1 + \rho_{1\theta\theta}), \\ \dfrac{\partial P_1}{\partial r}(R) = 0. \end{cases} \tag{6.18}$$

Assuming $\rho_1(\theta) = \cos(n\theta)$ and, by separation of variables,

$$M_1(r, \theta) = \cos(n\theta) Q_n(r), \tag{6.19}$$

we have

$$Q_n(r) = q_n(z) = \tilde{C}_1 I_n(z_r) + \tilde{C}_2 K_n(z_r) \tag{6.20}$$

which satisfies $z^2 q_n'' + z q_n' - (z^2 + n^2) q_n = 0$, $\frac{\partial Q_n}{\partial r}(R) = -Q_n(R)$, and $Q_n(\rho) = -\frac{\partial M_s(\rho)}{\partial r}$. Since $I_n'(x) = \frac{n}{x} I_n(x) + I_{n+1}(x)$ and $K_n'(x) = \frac{n}{x} K_n(x) - K_{n+1}(x)$, we have

$$\begin{aligned} Q_n'(r) =& \tilde{C}_1 \frac{\partial I_n(z_r)}{\partial r} + \tilde{C}_2 \frac{\partial K_n(z_r)}{\partial r} \\ =& \tilde{C}_1 (\frac{n}{r} I_n(z_r) + \sqrt{\frac{H}{D}} I_{n+1}(z_r)) \\ & + \tilde{C}_2 (\frac{n}{r} K_n(z_r) - \sqrt{\frac{H}{D}} K_{n+1}(z_r)), \end{aligned}$$

where

$$\begin{cases} \tilde{C}_1 = \dfrac{-M_s'(\rho)}{I_n(z_\rho) + K K_n(z_\rho)} \\ \tilde{C}_2 = \dfrac{-K M_s'(\rho)}{I_n(z_\rho) + K K_n(z_\rho)} \end{cases}$$

and

$$K = -\frac{I_n(z_R) + I_n'(z_R)}{K_n(z_R) + K_n'(z_R)}.$$

By the maximum principle, we have $Q_n(r) \geq 0$.

Similarly, we have $P_1(\rho) = \gamma(-\frac{1}{\rho^2} + \frac{n^2}{\rho^2})\cos(n\theta)$ and $\frac{\partial P_1}{\partial r}(R) = 0$. Therefore, we obtain

$$P_1 + \frac{LD}{H} M_1 = \tilde{C}_3 r^n \cos(n\theta) + \tilde{C}_4 r^{-n} \cos(n\theta), \tag{6.21}$$

where $\tilde{C}_3$ and $\tilde{C}_4$ satisfy

$$\begin{cases} -\frac{LD}{H}\frac{\partial Q_n}{\partial r}(R) + \tilde{C}_3 n R^{n-1} - \tilde{C}_4 n R^{-n-1} = 0, \\ -\frac{LD}{H} Q_n(\rho) + \tilde{C}_3 \rho^n + \tilde{C}_4 \rho^{-n} = \gamma(-\frac{1}{\rho^2} + \frac{n^2}{\rho^2}), \end{cases} \tag{6.22}$$

or

$$\begin{cases} \tilde{C}_3 = \dfrac{\gamma(n^3 - n)\rho^{-2+n} - L\frac{D}{H}(Q_n(R)R^{n+1} - nQ_n(\rho)\rho^n)}{n(\rho^{2n} + R^{2n})}, \\ \tilde{C}_4 = \tilde{C}_3 R^{2n} + L\dfrac{D}{H} Q_n(R)\dfrac{R^{n+1}}{n}. \end{cases} \tag{6.23}$$

6.4.2 Bifurcation analysis

We consider the nonlinear function F defined in (6.12) by expanding $\frac{\partial P}{\partial r}$ on Γ_ε, namely,

$$F(\rho_1, L) = \frac{\partial P}{\partial r}\Big|_{\Gamma_\varepsilon} = \varepsilon(\frac{\partial P_1}{\partial r}(\rho) + \frac{\partial^2 P_s(\rho)}{\partial r^2}\rho_1) + \mathcal{O}(|\varepsilon|^2). \tag{6.24}$$

Thus F maps (ρ_1, L) from $X^{l+3+\alpha}$ to $X^{l+\alpha}$ and is bounded for any $l \geq 0$ [112]. Furthermore, F is Fréchet differentiable and the Fréchet derivative at $(0, L)$ is given by

$$[\frac{\partial F}{\partial \rho_1}(0, L)]\cos(n\theta) = \frac{\partial P_1}{\partial r}(\rho) + \frac{\partial^2 P_s(\rho)}{\partial r^2}\rho_1. \tag{6.25}$$

Then the bifurcation condition becomes

$$\frac{\partial P_1}{\partial r}(\rho) + \frac{\partial^2 P_s(\rho)}{\partial r^2}\rho_1 = 0. \tag{6.26}$$

Since

$$\frac{\partial P_1}{\partial r}(\rho) = \cos(n\theta)[-\frac{LD}{H}\frac{\partial Q_n}{\partial r}(\rho) + n\tilde{C}_3\rho^{n-1} - n\tilde{C}_4\rho^{-n-1}] \tag{6.27}$$

and

$$\frac{\partial^2 P_s(\rho)}{\partial r^2} = T(L) - LM_s(\rho) = T(L) - L, \tag{6.28}$$

we obtain

$$\begin{aligned} F(L) =& T - L - \frac{LD}{H}\frac{\partial Q_n}{\partial r}(\rho) + n\tilde{C}_3\rho^{n-1} - n\tilde{C}_4\rho^{-n-1} \\ =& -\frac{DL}{H}\frac{2}{R^2-\rho^2}(RM_s(R)+\rho M_s'(\rho)) - L - \frac{LD}{H}Q_n'(\rho) \\ & - 2\frac{L\frac{D}{H}Q_n(R)R^{n+1}}{(\rho^{2n}+R^{2n})}\rho^{n-1} + \frac{\gamma(n^3-n)(\rho^{2n}-R^{2n})}{\rho^3(\rho^{2n}+R^{2n})} \\ & + n\frac{LD}{H}\frac{\rho^{2n}-R^{2n}}{\rho(\rho^{2n}+R^{2n})}Q_n(\rho) \\ =& 0. \end{aligned} \tag{6.29}$$

Therefore, the formula of L_n for bifurcation points is

$$L_n = \frac{C_1(n,\rho,R)}{C_2(n,\rho,R)} \tag{6.30}$$

where

$$C_1(n,\rho,R) = \frac{\gamma(n^3-n)(R^{2n}-\rho^{2n})}{\rho^3(\rho^{2n}+R^{2n})}$$

and

$$\begin{aligned} C_2(n,\rho,R) =& -\frac{D}{H}\frac{2}{R^2-\rho^2}(RM_s(R)+\rho M_s'(\rho)) - \frac{D}{H}Q_n'(\rho) \\ & - \frac{D}{H}\frac{2R^{n+1}\rho^n Q_n(R) + n(R^{2n}-\rho^{2n})Q_n(\rho)}{\rho(\rho^{2n}+R^{2n})} - 1. \end{aligned}$$

It is clear that C_1 is increasing with respect to n while the monotonicity of $C_2(n,\rho,R)$ is summarized in the following lemma.

Lemma 6.4.1. *For given $R > 0$, ρ is in a neighbor of R, namely, $\rho = R - \varepsilon$ for a small ε, $C_2(n,\rho,R) > 0$ is decreasing with respect to n .*

Proof. We rewrite $C_2(n,\rho,R)$ as

$$C_2(n,\rho,R) = -\frac{D}{H}\frac{2}{R^2-\rho^2}(RM_s(R)+\rho M_s'(\rho)) - 1 + \frac{D}{H}f(n),$$

where

$$f(n) = -Q_n'(\rho) - \frac{2R^{n+1}\rho^n}{\rho(\rho^{2n}+R^{2n})}Q_n(R) - \frac{n(R^{2n}-\rho^{2n})}{\rho(\rho^{2n}+R^{2n})}Q_n(\rho).$$

Since $\rho = R - \varepsilon$, we have

$$f(n) = \frac{H}{D} Q_n(R)\varepsilon + \mathcal{O}(\varepsilon^2).$$

By letting $F = \frac{dQ_n}{dn}$, we obtain

$$\begin{cases} -\Delta F + (\frac{n^2}{r^2} + \frac{H}{D})F = -\frac{2n}{r^2} Q_n, \\ F(\rho) = 0, \\ F(R) = \frac{dQ_n}{dn}(R), \end{cases}$$

and

$$\frac{dF}{dr}(R) = \frac{d}{dr}\frac{dQ_n}{dn}(R) = \frac{d}{dn}\frac{dQ_n}{dr}(R) = -\frac{dQ_n}{dn}(R).$$

If $\frac{dQ_n}{dn}(R) \geq 0$, then we have $\frac{dF}{dr}(R) < 0$. On the other hand, by the maximum principle, we have $\frac{dF}{dr}(R) > 0$, which leads to a contradiction. Thus $\frac{dQ_n}{dn}(R) < 0$, we have $C_2(n)$ decreases with respect to n and $F(r) \leq 0$ for all $\rho \leq r \leq R$. Moreover, since $\frac{dF}{dr}(\rho) < 0$, we have $\frac{dQ_n'(\rho)}{dn} < 0$.

Next we prove that $C_2(n, \rho, R) > 0$ when ε is small and expand $C_2(n, \rho, R)$ in terms of ε

$$\begin{aligned} C_2(n, \rho, R) =& (\frac{1}{2} M_s'(R) + Q_n(R))\varepsilon + \mathcal{O}(\varepsilon^2) \\ =& (\frac{1}{2} M_s'(R) + Q_n(\rho) + \varepsilon Q_n'(\rho))\varepsilon + \mathcal{O}(\varepsilon^2) \\ =& (\frac{1}{2} M_s'(R) - M_s'(\rho))\varepsilon + \mathcal{O}(\varepsilon^2) \\ =& -\frac{1}{2} M_s'(R)\varepsilon + \mathcal{O}(\varepsilon^2). \end{aligned}$$

Since $M_s'(R) < 0$, we have $C_2(n, \rho, R) > 0$ for a small ε.

□

Then we prove that L_n in (6.30) is a bifurcation point by verifying the following Crandall-Rabinowitz theorem [113].

Theorem 6.4.1. *Let X, Y be real Banach spaces and $F(x, \mu)$ a C^p map, $p \geq 3$, of a neighborhood $(0, \mu_0)$ in $X \times \mathbb{R}$ into Y. Suppose*

1. *$F(0, \mu) = 0$ for all μ in a neighborhood of μ_0,*

2. $KerF_x(0,\mu_0)$ *is one dimensional space, spanned by* x_0*,*

3. $ImF_x(0,\mu_0) = Y_1$ *has codimension 1,*

4. $F_{\mu x}(0,\mu_0) \notin Y_1$*.*

Then $(0,\mu_0)$ *is a bifurcation point of the equation* $F(x,\mu) = 0$ *in the following sense: In a neighborhood of* $(0,\mu_0)$*, the set of solutions of* $F(x,\mu) = 0$ *consists of two* C^{p-2} *smooth curves* $\mathcal{C}_1$ *and* $\mathcal{C}_2$ *which intersect only at the point* $(0,\mu_0)$*. Moreover* $\mathcal{C}_1$ *is the curve* $(0,\mu_0)$ *and* $\mathcal{C}_2$ *can be parameterized as follows:*

$$\mathcal{C}_2 : (x(\varepsilon),\mu(\varepsilon)), |\varepsilon| \text{ small}, \ (x(0),\mu(0)) = (0,\mu_0), x'(0) = x_0.$$

Verification. We choose the Banach spaces $X = X_1^{3+\alpha}$, $Y = X_1^{\alpha}$, $x = \rho_1$ and $\mu = L$, then have

$$[F_{\rho_1}(0,L)]\cos(n\theta) = (C_1(n,\rho,R) - LC_2(n,\rho,R))\cos(n\theta).$$

Thus the kernel space satisfies

$$\ker[F_{\rho_1}(0,L)] = \text{span}\{\cos(n\theta)\} \quad \text{if } L = L_n \tag{6.31}$$

and

$$\ker[F_{\rho_1}(0,L)] = 0 \quad \text{if } L \neq L_1, L_2, \cdots \tag{6.32}$$

which implies that $\dim(\ker[F_{\rho_1}(0,L)]) = 1$. Moreover, since that $\text{Im}[F_{\rho_1}(0,L_n)] \oplus \{\cos(n\theta)\}$ is the whole space, we have $\text{codim}(\text{Im}[F_{\rho_1}(0,L_n)]) = 1$. Finally, by differentiating with respect to L, we obtain

$$[F_{\rho_1 L}(0,L)]\cos(n\theta) = -C_2(n,\rho,R))\cos(n\theta) \notin \text{Im}[F_{\rho_1}(0,L_n)].$$

Thus all the assumptions in the Crandall-Rabinowitz theorem are satisfied. □

6.4.3 Linear Stability

We consider the linear stability via linearizing the free boundary $\Gamma_\varepsilon(t)$, $M(r,\theta,t)$, and $P(r,\theta,t)$ as follows:

$$\begin{aligned}
&\Gamma_\varepsilon : r = \rho_0(\theta) + \varepsilon\rho_1(\theta,t) + \mathcal{O}(\varepsilon^2),\\
&M(r,\theta,t) = M_0(r,\theta) + \varepsilon M_1(r,\theta,t) + \mathcal{O}(\varepsilon^2),\\
&P(r,\theta,t) = P_0(r,\theta) + \varepsilon P_1(r,\theta,t) + \mathcal{O}(\varepsilon^2).
\end{aligned} \tag{6.33}$$

The linearization of the normal direction of Γ_ε is

$$\begin{aligned}
\vec{n}|_{r=\rho_0+\varepsilon\rho_1} &= -\frac{\vec{e}_r - \frac{1}{r}(\rho_{0\theta} + \varepsilon\rho_{1\theta})\vec{e}_\theta}{\sqrt{1 + \frac{1}{r^2}(\rho_{0\theta} + \varepsilon\rho_{1\theta})^2}}\\
&= -\frac{(\rho_0 + \varepsilon\rho_1)\vec{e}_r - (\rho_{0\theta} + \varepsilon\rho_{1\theta})\vec{e}_\theta}{\sqrt{(\rho_0 + \varepsilon\rho_1)^2 + (\rho_{0\theta} + \varepsilon\rho_{1\theta})^2}}\\
&= \vec{n}_0 + \varepsilon\vec{n}_1 + \mathcal{O}(\varepsilon^2),
\end{aligned} \tag{6.34}$$

where

$$\vec{n}_0 = -\frac{\rho_0}{(\rho_0^2 + \rho_{0\theta}^2)^{\frac{1}{2}}}\vec{e}_r + \frac{\rho_{0\theta}}{(\rho_0^2 + \rho_{0\theta}^2)^{\frac{1}{2}}}\vec{e}_\theta$$

and

$$\begin{aligned}
\vec{n}_1 &= -[\frac{\rho_1\vec{e}_r - \rho_{1\theta}\vec{e}_\theta}{(\rho_0^2 + \rho_{0\theta}^2)^{\frac{1}{2}}} - \frac{(\rho_0\vec{e}_r - \rho_{0\theta}\vec{e}_\theta)(\rho_0\rho_1 + \rho_{0\theta}\rho_{1\theta})}{(\rho_0^2 + \rho_{0\theta}^2)^{\frac{3}{2}}}]\\
&= -\frac{\rho_1\rho_{0\theta}^2 - \rho_0\rho_{0\theta}\rho_{1\theta}}{(\rho_0^2 + \rho_{0\theta}^2)^{\frac{3}{2}}}\vec{e}_r - \frac{\rho_0\rho_1\rho_{0\theta} - \rho_0^2\rho_{1\theta}}{(\rho_0^2 + \rho_{0\theta}^2)^{\frac{3}{2}}}\vec{e}_\theta.
\end{aligned}$$

Then we linearize $\frac{\partial P}{\partial n}$ on the free boundary

$$\begin{aligned}
&\frac{\partial P}{\partial n}|_{r=\rho_0+\varepsilon\rho_1}\\
=&\{(P_{0r} + \varepsilon P_{1r})\vec{e}_r + \frac{1}{r}(P_{0\theta} + \varepsilon P_{1\theta})\vec{e}_\theta\}_{r=\rho_0+\varepsilon\rho_1} \cdot (\vec{n}_0 + \varepsilon\vec{n}_1)\\
=&(P_{0r}\vec{e}_r + \frac{P_{0\theta}}{\rho_0}\vec{e}_\theta) \cdot \vec{n}_0 + \varepsilon(P_{0r}\vec{e}_r + \frac{P_{0\theta}}{\rho_0}\vec{e}_\theta) \cdot \vec{n}_1\\
&+ \varepsilon[(\rho_1 P_{0rr} + P_{1r})\vec{e}_r + \frac{\rho_1\rho_0 P_{0r\theta} + P_{1\theta}\rho_0 - P_{0\theta}\rho_1}{\rho_0^2}\vec{e}_\theta] \cdot \vec{n}_0.
\end{aligned}$$

Since $(P_{0r}\vec{e}_r + \frac{P_{0\theta}}{\rho_0}\vec{e}_\theta)\cdot\vec{n}_0 = \frac{\partial P_0}{\partial n_0} = 0$, we have

$$\begin{aligned}
&[(\rho_1 P_{0rr} + P_{1r})\vec{e}_r + \frac{\rho_1\rho_0 P_{0r\theta} + P_{1\theta}\rho_0 - P_{0\theta}\rho_1}{\rho_0^2}\vec{e}_\theta]\cdot\vec{n}_0 \\
+&(P_{0r}\vec{e}_r + \frac{P_{0\theta}}{\rho_0}\vec{e}_\theta)\cdot\vec{n}_1 \\
=&-\frac{P_{0r}(\rho_1\rho_{0\theta}^2 - \rho_0\rho_{0\theta}\rho_{1\theta})}{(\rho_0^2+\rho_{0\theta}^2)^{\frac{3}{2}}} - \frac{P_{0\theta}(\rho_0\rho_1\rho_{0\theta} - \rho_0^2\rho_{1\theta})}{\rho_0(\rho_0^2+\rho_{0\theta}^2)^{\frac{3}{2}}} \\
&-\frac{\rho_0(\rho_1 P_{0rr} + P_{1r})}{(\rho_0^2+\rho_{0\theta}^2)^{\frac{1}{2}}} + \frac{\rho_{0\theta}(\rho_1\rho_0 P_{0r\theta} + P_{1\theta}\rho_0 - P_{0\theta}\rho_1)}{\rho_0^2(\rho_0^2+\rho_{0\theta}^2)^{\frac{1}{2}}}
\end{aligned}.$$

On the other hand, the velocity of Γ_ε along the normal direction is

$$v_n = \frac{-(\rho_0+\varepsilon\rho_1)_t}{\sqrt{1+\frac{1}{(\rho_0+\varepsilon\rho_1)^2}(\rho_{0\theta}+\varepsilon\rho_{1\theta})^2}} = -\frac{\varepsilon\rho_{1t}}{\sqrt{1+\frac{1}{\rho_0^2}\rho_{0\theta}^2}} + \mathcal{O}(\varepsilon^2).$$

Since $\frac{\partial P}{\partial n} = -v_n$ on Γ_ε, we obtain the following equation for ρ_1

$$\begin{aligned}
\rho_{1t} =& -[\frac{P_{0r}\rho_{0\theta}^2 + P_{0\theta}\rho_{0\theta}}{(\rho_0^2+\rho_{0\theta}^2)\rho_0} + P_{0rr} - \frac{\rho_{0\theta}P_{0r\theta}}{\rho_0^2} + \frac{\rho_{0\theta}P_{0\theta}}{\rho_0^3}]\rho_1 \\
&+ \frac{P_{0r}\rho_{0\theta} + P_{0\theta}}{\rho_0^2+\rho_{0\theta}^2}\rho_{1\theta} - [P_{1r} - \frac{P_{1\theta}\rho_{0\theta}}{\rho_0^2}].
\end{aligned}$$

Then the linearized system is

$$\begin{cases}
\dfrac{\partial M_1}{\partial t} - D\Delta M_1 = -HM_1 \\
\Delta P_1 = -LM_1 \\
\dfrac{\partial M_1}{\partial r}(R) = -M_1(R) \\
M_1(\rho_0) = -\rho_1 M_{0r}(\rho_0) \\
P_1(\rho_0) = \gamma\kappa_1 - \rho_1 P_{0r} \\
\dfrac{\partial P_1}{\partial r}(R) = 0 \\
\rho_{1t}|_{r=\rho_0} = [\dfrac{\rho_{0\theta}P_{0r\theta}}{\rho_0^2} - \dfrac{P_{0r}\rho_{0\theta}^2 + P_{0\theta}\rho_{0\theta}}{(\rho_0^2+\rho_{0\theta}^2)\rho_0} - P_{0rr} - \dfrac{\rho_{0\theta}P_{0\theta}}{\rho_0^3}]\rho_1 \\
+ \dfrac{P_{0r}\rho_{0\theta} + P_{0\theta}}{\rho_0^2+\rho_{0\theta}^2}\rho_{1\theta} - [P_{1r} - \dfrac{P_{1\theta}\rho_{0\theta}}{\rho_0^2}]
\end{cases} \tag{6.35}$$

Next we summarize the linear instability of the radially symmetric steady-state solutions in the following theorem.

Theorem 6.4.2. *For any given $L > 0$, the corresponding radially symmetric steady-state solution $(M_s(r), P_s(r), \rho_0)$ is linearly unstable. In fact, there exist initial conditions defined by*

$$\begin{aligned} &\rho_1(0) = \cos(n\theta) \\ &M_1(r,0) = u(r)\cos(n\theta) \\ &P_1(r,0) = w(r)\cos(n\theta) \end{aligned} \tag{6.36}$$

such that $\rho_1(t) \to \infty$.

Proof. We consider the solution $(\rho_1(t), M_1(r,t), P_1(r,t))$ with the following form

$$\begin{cases} \rho_1(t) = e^{at}\cos(n\theta), \\ M_1(r,t) = e^{at}u(r)\cos(n\theta), \\ P_1(r,t) = e^{at}w(r)\cos(n\theta). \end{cases} \tag{6.37}$$

Then the linearized system (6.35) is written as

$$\begin{cases} au(r) - D(\Delta u(r) - \dfrac{n^2}{r^2}u(r)) = -Hu(r) \\ \dfrac{\partial u}{\partial r}(R) = -u(R) \\ u(\rho_0) = -M_s'(\rho_0) \\ \Delta w(r) - \dfrac{n^2}{r^2}w(r) = -Lu(r) \\ w(\rho_0) = \dfrac{\gamma}{\rho_0^2}(n^2-1) \\ \dfrac{\partial w}{\partial r}(R) = 0 \\ a = -P_s'' - w_r(\rho_0) \end{cases} \tag{6.38}$$

By repeating the process in Section 6.4.1 (from (6.20) to (6.23)), we conclude

that a satisfies the following equation

$$\begin{aligned} a = & L - T(L) + \frac{LD}{H+a} u_r(\rho_0) \\ & + \frac{DL}{H+a} \frac{2R^{n+1}\rho_0^n u(R) + n(R^{2n} - \rho_0^{2n}) u(\rho_0)}{\rho_0(\rho_0^{2n} + R^{2n})}, \end{aligned} \tag{6.39}$$

where

$$u(r) = -M_s'(\rho_0)(\hat{C}_1 I_n(\hat{z}_r) + \hat{C}_2 K_n(\hat{z}_r)),\ \hat{z}_r = \sqrt{\frac{H+a}{D}} r,$$

$$\begin{cases} \hat{C}_1 = \dfrac{1}{I_n(\hat{z}_\rho) + K K_n(\hat{z}_\rho)}, \\ \hat{C}_2 = \dfrac{K}{I_n(\hat{z}_\rho) + K K_n(\hat{z}_\rho)}, \end{cases}$$

and

$$K = -\frac{I_n(\hat{z}_R) + I_n'(\hat{z}_R)}{K_n(\hat{z}_R) + K_n'(\hat{z}_R)}.$$

We consider a nonlinear function $h(a, n, L)$ defined as

$$\begin{aligned} h(a, n, L) = & L - T(L) - a + \frac{LD}{H+a} u_r(\rho_0) \\ & + \frac{DL}{H+a} \frac{2R^{n+1}\rho_0^n u(R) + n(R^{2n} - \rho_0^{2n}) u(\rho_0)}{\rho_0(\rho_0^{2n} + R^{2n})}. \end{aligned}$$

For $n \geq 0$, we have

$$h(\infty, n, L) \to -\infty \text{ and } h(0, n, L) = C_1(n, \rho_0, R) - C_2(n, \rho_0, R)L.$$

According to (6.30), L_n is monotonically increasing with respect to n. Thus for any given L, there exists n^* such that $L < L_n*$ which implies that $h(0, n^*, L) = C_1 - C_2 L > C_1 - C_2 L_{n^*} = 0$. Therefore there must be at least one positive root of $h(a, n^*, L) = 0$. By (6.37), we have $\rho_1 \to \infty$.

□

Remark 8. *In the proof, we have $n^* \geq 2$. In fact, when $n = 0, 1$, for $L \geq 0$, we have*

$$h(0, n, L) = -C_2(n, \rho_0, R)L \leq 0.$$

Moreover, when ρ_0 is in a neighborhood of R, say $\rho_0 = R - \varepsilon$, we expand $h(a, n, L)$

at $r = R$*:*

$$h(a, n, L) = -a + L - T(L) - \varepsilon L u(R) + \mathcal{O}(\varepsilon^2)$$

which decreases with respect to a *when* ε *is small. Thus* $h(a, n, L) = 0$ *does not have a positive solution for* $n = 0, 1$ *and fixed* $L \geq 0$*.*

6.5 Numerical Results

In this section, we employ numerical simulations to verify our theoretical results. Since the Laplacian operator in the 2D polar coordinate is defined as

$$\Delta = \frac{\partial^2}{\partial r^2} + \frac{1}{r}\frac{\partial}{\partial r} + \frac{1}{r^2}\frac{\partial^2}{\partial \theta^2},$$

we use the uniform grid points on the θ direction with a stepsize $\Delta\theta = \frac{2\pi}{m}$, namely, $\theta_j = j\Delta\theta$, $j = 0, \ldots, m-1$ where m is the number of grid points. For the radius on each θ_j direction, we use the uniform grid points with a stepsize h_j, namely, $r_{i,j} = \rho_j + ih_j$ with $h_j = \frac{R-\rho_j}{n}$ and $i = 0, 1, \ldots, n$, where n is the number of grid points on each radius. Then we use the central difference scheme to approximate $\frac{\partial^2}{\partial r^2}G(r_{i,j}, \theta_j)$ and $\frac{1}{r}\frac{\partial}{\partial r}G(r_{i,j}, \theta_j)$, namely,

$$\frac{\partial^2}{\partial r^2}G(r_{i,j}, \theta_j) = \frac{G(r_{i+1,j}, \theta_j) + G(r_{i-1,j}, \theta_j) - 2G(r_{i,j}, \theta_j)}{h_j^2}$$

and

$$\frac{\partial}{\partial r}G(r_{i,j}, \theta_j) = \frac{G(r_{i+1,j}, \theta_j) - G(r_{i-1,j}, \theta_j)}{2h_j}.$$

Moreover, we use nine points to approximate $\frac{\partial^2}{\partial \theta^2}$ such that the scheme has the second-order accuracy even for the non-radially symmetric case. The scheme is derived based on the Taylor expansion and shown in (B.1) in the Appendix B.1. This numerical scheme will recover the central difference scheme for the radially symmetric case.

6.5.1 Convergence Test

First we perform a convergence order test of our numerical scheme for the radially symmetric steady-state solution which has analytic formulas shown in (6.6) and (6.8). The numerical error is defined as $Err(h, \Delta\theta) = \|(M_h, P_h) - (M_S, P_S)\|_\infty$ where (M_h, P_h) is the numerical solution and (M_S, P_S) is the analytic solution. Here we choose $L = H = 3$, $D = 1$, $\gamma = 2$, $R = 2$ and $\rho = 1.6$ and show the numerical error in Table 6.1 which demonstrates the second order of convergence.

Table 6.1: Numerical errors and the order of convergence for different grid points.

$(h, \Delta\theta)$	$Err(h, \Delta\theta)$	order of convergence
$(0.2, \frac{\pi}{10})$	0.0192	-
$(0.1, \frac{\pi}{20})$	0.0044	2.1299
$(0.05, \frac{\pi}{40})$	0.0011	2.0303
$(0.025, \frac{\pi}{80})$	2.6668e-04	2.0075

Next, we test the convergence of the numerical scheme for computing bifurcation points. Numerically we use the adaptive homotopy method [25] to compute bifurcation points. More specifically, starting with a radially symmetric steady-state solution, we track along the radially symmetric solution path to L and monitor the smallest eigenvalue of the nonlinear system. When the norm of the smallest eigenvalue is less than a tolerance, e.g, 10^{-4} in our simulation, we obtain a numerical bifurcation point denote as $\tilde{L}$. The theoretical value of bifurcation point, L_n, is computed by (6.30) for any given n. Then we compute the numerical error of bifurcation points for $n = 2, 3, 4$ with different stepsize h and $\Delta\theta$ shown in Table 6.2. It is clearly shown that the numerical error gets smaller when the stepsize gets smaller which demonstrates the convergence.

6.5.2 The bifurcation structure and non-radially symmetric solutions

We numerically explore the local bifurcation structure and non-radially symmetric steady-state solutions by using the tangent cone algorithm [25]. The local bifurcation structure is shown in Fig. 6.2 for $n = 2, 3, 4$. The y axis is a projection

Table 6.2: The numerical error of bifurcation points $|\tilde{L} - L_n|$ for different n and stepsize.

$(h, \Delta\theta)$	$n = 2$	$n = 3$	$n = 4$
$(0.2, \frac{\pi}{10})$	0.7005	3.3115	1.9387
$(0.1, \frac{\pi}{20})$	0.1465	0.6722	0.1152
$(0.05, \frac{\pi}{40})$	0.0355	0.1586	0.0541
$(0.025, \frac{\pi}{80})$	0.0106	0.0305	0.0122

function defined as $\mathcal{P}(\rho(\theta)) = (\rho_{max} - \rho_{min})(\theta_{max} - \theta_{min})$ for any given $\rho(\theta)$ which quantifies the change of the free boundary. For the radially symmetric branch, we have $\mathcal{P}(\rho(\theta)) = 0$; for the non-radially symmetric branch, we have different local structures shown in Fig. 6.2 for different n. Moreover, the non-radially symmetric solutions in Fig. 6.2 are consistent with the perturbation $\cos(n\theta)$ in section 6.4.1. The color of non-radially symmetric solutions stands for the macrophage density across the domain.

6.5.3 Linear Stability

First, we verify the conclusion of Theorem 6.4.2 via checking the linear stability of radially symmetric solutions for different values of L. More specifically, we check the real part of the largest eigenvalue, $real(\lambda_{max})$, and list in Table 6.3 for different L. It shows that all the radially symmetric solutions are linearly unstable due to the positive largest eigenvalue.

Table 6.3: The real part of the largest eigenvalues of radially symmetric solutions v.s. L.

L	$real(\lambda_{max})$
$L = 0$	1.7050×10^4
$L = 3$	1.7082×10^4
$L = 15$	1.7085×10^4
$L = 120$	1.7108×10^4
$L = 200$	1.7126×10^4

Second, we check the linear stability of radially symmetric solution with radially symmetric perturbations which is the case of $n = 0$ in **Remark 1**. It can

be seen from Table 6.4 that, under radially symmetric perturbations, the radially symmetric solutions are linearly stable when L is small which is consistent with **Remark 1**. As L becomes large, there could be unstable coupled perturbations that can not be written in the form of separation of variables in (6.37). Therefore, for large L, radially symmetric solutions become unstable even with radially symmetric perturbations shown in Table 6.4.

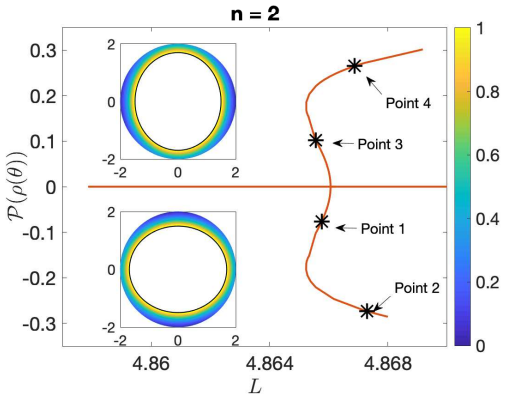
Table 6.4: The real part of the largest eigenvalues of radially symmetric solutions v.s. L under radially symmetric perturbations.

L	$real(\lambda_{max})$
5	-1.7489
10	-4.1792
50	-7.3511
100	-3.5551
140	-0.3859
150	0.4233
200	4.5619

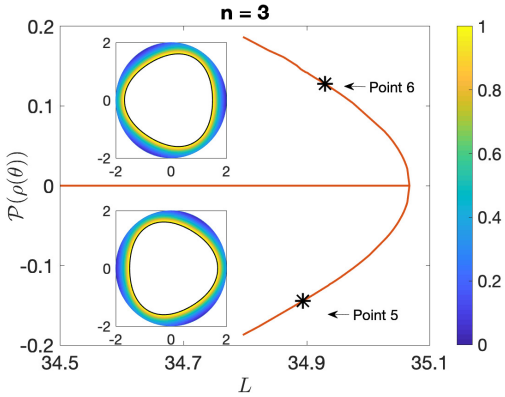
Last, we test the linear stability on the non-radially symmetric branches by choosing generic points on each branch (see points in Fig. 6.2). All the non-radially symmetric branches are linearly unstable since there exist positive real eigenvalues shown in Table 6.5.

Table 6.5: The real part of the largest eigenvalues for non-radially symmetric solutions shown in Fig. 6.2.

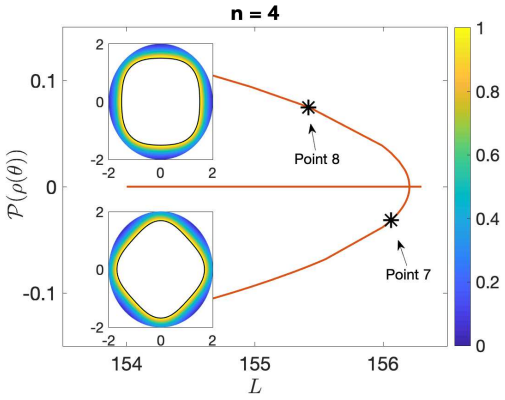
Point	$real(\lambda_{max})$
Point 1	2.1998×10^4
Point 2	2.3076×10^4
Point 3	1.9630×10^4
Point 4	2.2456×10^4
Point 5	2.1340×10^4
Point 6	2.8766×10^4
Point 7	2.3730×10^4
Point 8	2.0012×10^4



(a) $n = 2$



(b) $n = 3$



(c) $n = 4$

Figure 6.2: The bifurcation structure and non-radially symmetric steady-state solutions for different n.

Appendix A

Proofs in Chapter 5

A.1 Proof of Theorem 5.2.1

Proof. We compare the predictor step of the traditional and the stochastic homotopy tracking at $p = p_{k-1}$ and obtain

$$\begin{aligned} \mathbf{u}_k &= \mathbf{u}_{k-1} + \mathbf{F}_{\mathbf{u}}^{-1}(\mathbf{u}_{k-1})\mathbf{F}_p(\mathbf{u}_{k-1})\Delta p, \\ \tilde{\mathbf{u}}_k &= \tilde{\mathbf{u}}_{k-1} + \tilde{\mathbf{F}}_{\mathbf{u}}^{-1}(\tilde{\mathbf{u}}_{k-1})\tilde{\mathbf{F}}_p(\tilde{\mathbf{u}}_{k-1})\Delta p, \end{aligned} \tag{A.1}$$

which implies

$$\mathbf{u}_k - \tilde{\mathbf{u}}_k = \mathbf{u}_{k-1} - \tilde{\mathbf{u}}_{k-1} + \mathbf{FP}(\mathbf{u}_k, \tilde{\mathbf{u}}_k)\Delta p, \tag{A.2}$$

where $\mathbf{FP}(\mathbf{u}_k, \tilde{\mathbf{u}}_k) = \mathbf{F}_{\mathbf{u}}^{-1}(\mathbf{u}_{k-1})\mathbf{F}_p(\mathbf{u}_{k-1}) - \tilde{\mathbf{F}}_{\mathbf{u}}^{-1}(\tilde{\mathbf{u}}_{k-1})\tilde{\mathbf{F}}_p(\tilde{\mathbf{u}}_{k-1})$. Then by taking the expectation with respect to ξ, we have

$$\begin{aligned} &\|\mathbb{E}(\mathbf{u}_k - \tilde{\mathbf{u}}_k)\|^2 = \|\mathbb{E}(\mathbf{u}_{k-1} - \tilde{\mathbf{u}}_{k-1}) + \mathbb{E}\big(\mathbf{FP}(\mathbf{u}_k, \tilde{\mathbf{u}}_k)\big)\Delta p\|^2 \\ \leq &\|\mathbb{E}(\mathbf{u}_{k-1} - \tilde{\mathbf{u}}_{k-1})\|^2 + 2\|\mathbb{E}(\mathbf{u}_{k-1} - \tilde{\mathbf{u}}_{k-1})\|\|\mathbb{E}\big(\mathbf{FP}(\mathbf{u}_k, \tilde{\mathbf{u}}_k)\big)\|\Delta p \\ &+ \|\mathbb{E}\big(\mathbf{FP}(\mathbf{u}_k, \tilde{\mathbf{u}}_k)\big)\|^2\Delta p^2 \\ \leq &(1 + \Delta p)\|\mathbb{E}(\mathbf{u}_{k-1} - \tilde{\mathbf{u}}_{k-1})\|^2 + \|\mathbb{E}\big(\mathbf{FP}(\mathbf{u}_k, \tilde{\mathbf{u}}_k)\big)\|^2(\Delta p + \Delta p^2). \end{aligned} \tag{A.3}$$

Moreover, by Taylor's theorem, there exists $\mathbf{t}_{k-1}$ such that

$$\mathbf{F}_{\mathbf{u}}(\mathbf{u}_{k-1}) = \mathbf{F}_{\mathbf{u}}(\tilde{\mathbf{u}}_{k-1}) + \nabla\mathbf{F}_{\mathbf{u}}(\mathbf{t}_{k-1}) \cdot (\mathbf{u}_{k-1} - \tilde{\mathbf{u}}_{k-1}). \tag{A.4}$$

Therefore, we have

$$\begin{aligned}
&\mathbf{FP}(\mathbf{u}_k, \tilde{\mathbf{u}}_k) = \mathbf{F}_{\mathbf{u}}^{-1}(\mathbf{u}_{k-1})\big[\mathbf{F}_p(\mathbf{u}_{k-1}) - \mathbf{F}_{\mathbf{u}}(\mathbf{u}_{k-1})\mathbf{F}_{\mathbf{u}}^{-1}(\tilde{\mathbf{u}}_{k-1}, \xi_k)\mathbf{F}_p(\tilde{\mathbf{u}}_{k-1}, \xi_k)\big] \\
=&\mathbf{F}_{\mathbf{u}}^{-1}(\mathbf{u}_{k-1})\Big[\mathbf{F}_p(\mathbf{u}_{k-1}) - \Big((\mathbf{F}_{\mathbf{u}} + S)(\tilde{\mathbf{u}}_{k-1}, \xi_k) + \nabla\mathbf{F}_{\mathbf{u}}(\mathbf{t}_{k-1}) \cdot (\mathbf{u}_{k-1} - \tilde{\mathbf{u}}_{k-1})\Big) \\
&\mathbf{F}_{\mathbf{u}}^{-1}(\tilde{\mathbf{u}}_{k-1}, \xi_k)\big(\mathbf{F}_p(\tilde{\mathbf{u}}_{k-1}) - C(\tilde{\mathbf{u}}_{k-1}, \xi_k)\big)\Big] \\
=&\mathbf{F}_{\mathbf{u}}^{-1}(\mathbf{u}_{k-1})[\mathbf{F}_p(\mathbf{u}_{k-1}) - \mathbf{F}_p(\tilde{\mathbf{u}}_{k-1}) + R(\tilde{\mathbf{u}}_{k-1}, \mathbf{u}_{k-1}, \xi_k)],
\end{aligned} \tag{A.5}$$

where

$$\begin{aligned}
&R(\tilde{\mathbf{u}}_{k-1}, \mathbf{u}_{k-1}, \xi_k) \\
=&C(\tilde{\mathbf{u}}_{k-1}, \xi_k) - S(\tilde{\mathbf{u}}_{k-1}, \xi_k)\mathbf{F}_{\mathbf{u}}^{-1}(\tilde{\mathbf{u}}_{k-1}, \xi_k)(\mathbf{F}_p(\tilde{\mathbf{u}}_{k-1}) - C(\tilde{\mathbf{u}}_{k-1}, \xi_k)) \\
&- \nabla\mathbf{F}_{\mathbf{u}}(\mathbf{t}_{k-1}) \cdot (\mathbf{u}_{k-1} - \tilde{\mathbf{u}}_{k-1})\mathbf{F}_{\mathbf{u}}^{-1}(\tilde{\mathbf{u}}_{k-1}, \xi_k)(\mathbf{F}_p(\tilde{\mathbf{u}}_{k-1}) - C(\tilde{\mathbf{u}}_{k-1}, \xi_k)).
\end{aligned}$$

Moreover, there exists $\mathbf{s}_{k-1}$ such that

$$\mathbf{F}_p(\mathbf{u}_{k-1}) = \mathbf{F}_{\mathbf{u}}^{-1}(\mathbf{u}_{k-1}) + \nabla\mathbf{F}_p(\mathbf{s}_{k-1})(\mathbf{u}_{k-1} - \tilde{\mathbf{u}}_{k-1}),$$

then Eq. (A.5) becomes

$$\begin{aligned}
&\|\mathbb{E}(\mathbf{FP}(\mathbf{u}_{k-1}, \tilde{\mathbf{u}}_{k-1}))\|^2 \\
=&\|\mathbb{E}(\mathbf{F}_{\mathbf{u}}^{-1}(\mathbf{u}_{k-1})\nabla\mathbf{F}_p(\mathbf{s}_{k-1})(\mathbf{u}_{k-1} - \tilde{\mathbf{u}}_{k-1})) + \mathbb{E}(\mathbf{F}_{\mathbf{u}}^{-1}(\mathbf{u}_{k-1})R(\tilde{\mathbf{u}}_{k-1}, \mathbf{u}_{k-1}, \xi_k))\|^2 \\
\leq&2\|\mathbb{E}(\mathbf{F}_{\mathbf{u}}^{-1}(\mathbf{u}_{k-1})\nabla\mathbf{F}_p(\mathbf{s}_{k-1})(\mathbf{u}_{k-1} - \tilde{\mathbf{u}}_{k-1}))\|^2 \\
&+ 2\|\mathbb{E}(\mathbf{F}_{\mathbf{u}}^{-1}(\mathbf{u}_{k-1})R(\tilde{\mathbf{u}}_{k-1}, \mathbf{u}_{k-1}, \xi_k))\|^2) \\
\leq&2\|\mathbf{F}_{\mathbf{u}}^{-1}(\mathbf{u}_{k-1})\|^2\|\nabla\mathbf{F}_p(\cdot)\|^2\|\mathbb{E}((\mathbf{u}_{k-1} - \tilde{\mathbf{u}}_{k-1}))\|^2 \\
&+ 2\|\mathbf{F}_{\mathbf{u}}^{-1}(\mathbf{u}_{k-1})\|^2\|\mathbb{E}(R(\tilde{\mathbf{u}}_{k-1}, \mathbf{u}_{k-1}, \xi_k))\|^2
\end{aligned} \tag{A.6}$$

Since $\mathbf{F}_{\mathbf{u}}^{-1}$ and $\nabla\mathbf{F}_p$ are bounded, we have

$$\|\mathbb{E}(\mathbf{FP}(\mathbf{u}_{k-1}, \tilde{\mathbf{u}}_{k-1}))\|^2 \leq 2M_{\mathbf{u}}^2L_p^2\|\mathbb{E}((\mathbf{u}_{k-1} - \tilde{\mathbf{u}}_{k-1}))\|^2 + 2M_{\mathbf{u}}^2\|\mathbb{E}(R(\tilde{\mathbf{u}}_{k-1}, \mathbf{u}_{k-1}, \xi_k))\|^2 \tag{A.7}$$

Next we estimate $R(\tilde{\mathbf{u}}_{k-1}, \mathbf{u}_{k-1}, \xi_k)$:

$$
\begin{aligned}
&\|\mathbb{E}(R(\tilde{\mathbf{u}}_{k-1},\mathbf{u}_{k-1},\xi_k))\|^2=\|\mathbb{E}_{\xi_0\xi_1\ldots\xi_{k-1}}(\mathbb{E}_{\xi_k}R(\tilde{\mathbf{u}}_{k-1},\mathbf{u}_{k-1},\xi_k))\|^2\\
=&\Big\|\mathbb{E}\big(\frac{1}{(C_n^m)^2}\sum_{\mathcal{I},\mathcal{J}\in\mathcal{M}}R(\tilde{\mathbf{u}}_{k-1},\mathbf{u}_{k-1},\xi_k=(\mathcal{I},\mathcal{J}))\big)\Big\|^2\\
\le&3\Big(\underbrace{\|\frac{1}{(C_n^m)^2}\sum_{\mathcal{I},\mathcal{J}\in\mathcal{M}}\mathbb{E}(C(\tilde{\mathbf{u}}_{k-1},\xi_k=(\mathcal{I},\mathcal{J})))\|^2}_{A_1}\\
&+\underbrace{\|\frac{1}{(C_n^m)^2}\sum_{\mathcal{I},\mathcal{J}\in\mathcal{M}}\mathbb{E}(S(\tilde{\mathbf{u}}_{k-1},\xi_k=(\mathcal{I},\mathcal{J}))\mathbf{F}_{\mathbf{u}}^{-1}(\tilde{\mathbf{u}}_{k-1},\xi_k=(\mathcal{I},\mathcal{J}))(\mathbf{F}_p(\tilde{\mathbf{u}}_{k-1}}\\
&\underbrace{-C(\tilde{\mathbf{u}}_{k-1},\xi_k=(\mathcal{I},\mathcal{J})))\|^2}_{A_2}\\
&+\underbrace{\|\frac{1}{(C_n^m)^2}\sum_{\mathcal{I},\mathcal{J}\in\mathcal{M}}\mathbb{E}(\nabla\mathbf{F}_{\mathbf{u}}(\mathbf{t}_{k-1})\cdot(\mathbf{u}_{k-1}-\tilde{\mathbf{u}}_{k-1})\mathbf{F}_{\mathbf{u}}^{-1}(\tilde{\mathbf{u}}_{k-1},\xi_k=(\mathcal{I},\mathcal{J}))(\mathbf{F}_p(\tilde{\mathbf{u}}_{k-1})}\\
&\underbrace{-C(\tilde{\mathbf{u}}_{k-1},\xi_k=(\mathcal{I},\mathcal{J})))\|^2}_{A_3}\Big).
\end{aligned}
\tag{A.8}
$$

Since

$$
\sum_{\mathcal{I},\mathcal{J}\in\mathcal{M}}C(\tilde{\mathbf{u}}_{k-1},\xi_k=(\mathcal{I},\mathcal{J}))=\sum_{\mathcal{I},\mathcal{J}\in\mathcal{M}}\sum_{i\in\mathcal{I}}\frac{\partial F_i}{\partial p}\mathbf{e}_i=C_n^mC_{n-1}^{m-1}\mathbf{F}_p(\tilde{\mathbf{u}}_{k-1}),
$$

we have

$$
A_1=\|\frac{C_n^mC_{n-1}^{m-1}}{(C_n^m)^2}\mathbb{E}(\mathbf{F}_p(\tilde{\mathbf{u}}_{k-1}))\|^2=\|\frac{C_{n-1}^{m-1}}{C_n^m}\mathbb{E}(\mathbf{F}_p(\tilde{\mathbf{u}}_{k-1}))\|^2\le\frac{m^2}{n^2}M_p^2.
$$

Moreover, we have

$$
\begin{aligned}
A_2&\le\|\frac{1}{(C_n^m)^2}\sum_{\mathcal{I},\mathcal{J}\in\mathcal{M}}\mathbb{E}(S(\tilde{\mathbf{u}}_{k-1},\xi_k=(\mathcal{I},\mathcal{J}))\|^2\|\mathbf{F}_{\mathbf{u}}^{-1}(\tilde{\mathbf{u}}_{k-1},\cdot)\|^2\|\mathbf{F}_p(\tilde{\mathbf{u}}_{k-1})\|^2\\
&\le\frac{M_{\mathbf{u}}^2M_p^2}{(C_n^m)^4}\|\sum_{\mathcal{I},\mathcal{J}\in\mathcal{M}}\mathbb{E}(S(\tilde{\mathbf{u}}_{k-1},\xi_k=(\mathcal{I},\mathcal{J}))\|^2,
\end{aligned}
$$

By the definition of $S(\tilde{\mathbf{u}}_{k-1},\xi_k=(\mathcal{I},\mathcal{J})$, we have

$$
\begin{aligned}
\sum_{\mathcal{I},\mathcal{J}\in\mathcal{M}}\mathbb{E}(S(\tilde{\mathbf{u}}_{k-1},\xi_k=(\mathcal{I},\mathcal{J}))&=\sum_{\mathcal{J}\in\mathcal{M}}\mathbb{E}\big(\sum_{\mathcal{I}\in\mathcal{M}}(\sum_{i\in\mathcal{I}}\mathbf{e}_i\frac{\partial F_i}{\partial\mathbf{u}}(\mathbf{u})-\sum_{i\in\mathcal{I},j\in\mathcal{J}}E_{ij})\big)\\
&=C_n^mC_{n-1}^{m-1}\mathbb{E}(\mathbf{F}_{\mathbf{u}}(\tilde{\mathbf{u}}_{k-1}))-C_{n-1}^{m-1}C_{n-1}^{m-1}\mathbf{E},
\end{aligned}
$$

where $\mathbf{E}$ is the all-ones matrix. Therefore

$$A_2 \leq \frac{m^2}{n^2}(L_{\mathbf{u}}+1)^2 M_{\mathbf{u}}^2 M_p^2.$$

Similarly, we have

$$\begin{aligned} A_3 &\leq \Big\| \frac{1}{(C_n^m)^2} \sum_{\mathcal{I},\mathcal{J}\in\mathcal{M}} \nabla \mathbf{F}_{\mathbf{u}}(\mathbf{t}_{k-1}) \|^2 \|\mathbb{E}(\mathbf{u}_{k-1}-\tilde{\mathbf{u}}_{k-1})\|^2 \|\mathbf{F}_{\mathbf{u}}^{-1}(\tilde{\mathbf{u}}_{k-1},\cdot)\|^2 \|\mathbf{F}_p(\tilde{\mathbf{u}}_{k-1})\Big\|^2 \\ &\leq L_{\mathbf{u}}^2 M_{\mathbf{u}}^2 M_p^2 \|\mathbb{E}(\mathbf{u}_{k-1}-\tilde{\mathbf{u}}_{k-1})\|^2. \end{aligned}$$

Then Eq. (A.8) becomes

$$\|\mathbb{E}(R(\tilde{\mathbf{u}}_{k-1},\mathbf{u}_{k-1},\xi_k))\|^2 \leq \frac{m^2}{n^2}C_1 + C_2\|\mathbb{E}(\mathbf{u}_{k-1}-\tilde{\mathbf{u}}_{k-1})\|^2, \tag{A.9}$$

where $C_1 = 3(M_{\mathbf{u}}^2 M_p^2 (L_{\mathbf{u}}+1)^2 + M_p^2)$ and $C_2 = 3L_{\mathbf{u}}^2 M_{\mathbf{u}}^2 M_p^2$.

Then we get the estimate below

$$\|\mathbb{E}(\mathbf{FP}(\mathbf{u}_{k-1},\tilde{\mathbf{u}}_{k-1}))\|^2 \leq \frac{m^2}{n^2}M_1 + M_2\|\mathbb{E}(\mathbf{u}_{k-1}-\tilde{\mathbf{u}}_{k-1})\|^2,$$

where

$$M_1 = 2M_{\mathbf{u}}^2 C_1 \text{ and } M_2 = 2M_{\mathbf{u}}^2 L_p^2 + C_2.$$

Plugging the above results into (A.3), we have

$$\begin{aligned} &\|\mathbb{E}(\mathbf{u}_k - \tilde{\mathbf{u}}_k)\|^2 \\ \leq &(1+\Delta p)\|\mathbb{E}(\mathbf{u}_{k-1}-\tilde{\mathbf{u}}_{k-1})\|^2 + (\frac{m^2 M_1}{n^2} + M_2\|\mathbb{E}(\mathbf{u}_{k-1}-\tilde{\mathbf{u}}_{k-1})\|^2))(\Delta p + \Delta p^2) \\ \leq &\underbrace{(1+\Delta p + M_2(\Delta p+\Delta p^2))}_{\tilde{M}_1}\|\mathbb{E}(\mathbf{u}_{k-1}-\tilde{\mathbf{u}}_{k-1})\|^2 + \underbrace{\frac{m^2}{n^2}M_1(\Delta p+\Delta p^2)}_{\tilde{M}_2}. \end{aligned} \tag{A.10}$$

which implies

$$\begin{aligned}\|\mathbb{E}(\mathbf{u}_k-\tilde{\mathbf{u}}_k)\|^2 \leq & \tilde{M}_1\|\mathbb{E}(\mathbf{u}_{k-1}-\tilde{\mathbf{u}}_{k-1})\|^2+\tilde{M}_2\\ \leq & \tilde{M}_1^2\|\mathbb{E}(\mathbf{u}_{k-2}-\tilde{\mathbf{u}}_{k-2})\|^2+\tilde{M}_1\tilde{M}_2+\tilde{M}_2\\ \leq & \tilde{M}_1^k\|\mathbb{E}(\mathbf{u}_0-\tilde{\mathbf{u}}_0)\|^2+(1+\tilde{M}_1+\cdots+\tilde{M}_1^{k-1})\tilde{M}_2\\ = & \tilde{M}_1^k\|\mathbb{E}(\mathbf{u}_0-\tilde{\mathbf{u}}_0)\|^2+\frac{1-\tilde{M}_1^k}{1-\tilde{M}_1}\tilde{M}_2.\end{aligned} \tag{A.11}$$

Then we obtain the estimate of $\frac{1-\tilde{M}_1^k}{1-\tilde{M}_1}\tilde{M}_2$ as follows

$$\begin{aligned}&\frac{1-\tilde{M}_1^k}{1-\tilde{M}_1}\tilde{M}_2 \leq \frac{e^{(1+M_2)(b-a)}-1}{(1+M_2)\Delta p+M_2\Delta p^2}\tilde{M}_2\\ \leq&\frac{e^{(1+M_2)(b-a)}-1}{(1+M_2)\Delta p}(1-\frac{M_2}{1+M_2}\Delta p+\mathcal{O}(\Delta p^2))\frac{m^2M_1}{n^2}(\Delta p+\Delta p^2)\\ \leq&\frac{m^2M_1}{n^2}\frac{e^{(1+M_2)(b-a)}-1}{1+M_2}+\mathcal{O}(\frac{m^2\Delta p}{n^2})\end{aligned}$$

Thus, Eq. (A.11) becomes

$$\|\mathbb{E}(\mathbf{u}_N-\tilde{\mathbf{u}}_N)\|^2 \leq \underbrace{e^{(1+M_2)(b-a)}}_{CS_1}\|\mathbb{E}(\mathbf{u}_0-\tilde{\mathbf{u}}_0)\|^2+\underbrace{M_1\frac{e^{(1+M_2)(b-a)}-1}{1+M_2}}_{CS_2}\frac{m^2}{n^2}+\mathcal{O}(\frac{m^2\Delta p}{n^2}).$$

□

A.2 Proof of Theorem 5.2.2

Proof. We consider the i-th iteration of Newton's correction for $\mathbf{F}(\mathbf{u},p_k)=0$ and $\tilde{\mathbf{F}}(\mathbf{u},p_k)=0$. There exists $\mathbf{t}_k$ and $\tilde{\mathbf{t}}_k$ such that the following Taylor expansions hold

$$\begin{aligned}0=\mathbf{F}(\mathbf{u}_k,p_k)&=\mathbf{F}(\mathbf{u}_k^i)+\mathbf{F}_{\mathbf{u}}(\mathbf{u}_k^i)(\mathbf{u}_k-\mathbf{u}_k^i)+\frac{1}{2}(\mathbf{u}_k-\mathbf{u}_k^i)^T\nabla\mathbf{F}_{\mathbf{u}}(\mathbf{t}_k)(\mathbf{u}_k-\mathbf{u}_k^i),\\ 0=\tilde{\mathbf{F}}(\tilde{\mathbf{u}}_k,p_k)&=\tilde{\mathbf{F}}(\tilde{\mathbf{u}}_k^i)+\tilde{\mathbf{F}}_{\mathbf{u}}(\tilde{\mathbf{u}}_k^i)(\tilde{\mathbf{u}}_k-\tilde{\mathbf{u}}_k^i)+\frac{1}{2}(\tilde{\mathbf{u}}_k-\tilde{\mathbf{u}}_k^i)^T\nabla\tilde{\mathbf{F}}_{\mathbf{u}}(\tilde{\mathbf{t}}_k)(\tilde{\mathbf{u}}_k-\tilde{\mathbf{u}}_k^i).\end{aligned}$$

Thus the Newton's schemes are re-written as

$$\mathbf{u}_k^{i+1} = \mathbf{u}_k^i - \mathbf{F}_{\mathbf{u}}^{-1}(\mathbf{u}_k^i)\mathbf{F}(\mathbf{u}_k^i) = \mathbf{u}_k + \frac{1}{2}\mathbf{F}_{\mathbf{u}}^{-1}(\mathbf{u}_k^i)(\mathbf{u}_k - \mathbf{u}_k^i)^T\nabla\mathbf{F}_{\mathbf{u}}(\mathbf{t}_k)(\mathbf{u}_k - \mathbf{u}_k^i),$$
$$\tilde{\mathbf{u}}_k^{i+1} = \tilde{\mathbf{u}}_k^i - \tilde{\mathbf{F}}_{\mathbf{u}}^{-1}(\tilde{\mathbf{u}}_k^i)\tilde{\mathbf{F}}(\tilde{\mathbf{u}}_k^i) = \tilde{\mathbf{u}}_k + \frac{1}{2}\tilde{\mathbf{F}}_{\mathbf{u}}^{-1}(\tilde{\mathbf{u}}_k^i)(\tilde{\mathbf{u}}_k - \tilde{\mathbf{u}}_k^i)^T\nabla\tilde{\mathbf{F}}_{\mathbf{u}}(\tilde{\mathbf{t}}_k)(\tilde{\mathbf{u}}_k - \tilde{\mathbf{u}}_k^i).$$

Therefore,

$$\begin{aligned}
\|\mathbb{E}(\mathbf{u}_k^{i+1} - \tilde{\mathbf{u}}_k^{i+1})\| =& \Big\|\mathbb{E}\Big(\big((\mathbf{u}_k^i - \mathbf{F}_{\mathbf{u}}^{-1}(\mathbf{u}_k^i)\mathbf{F}(\mathbf{u}_k^i)\big) - \big(\tilde{\mathbf{u}}_k^i - \tilde{\mathbf{F}}_{\mathbf{u}}^{-1}(\tilde{\mathbf{u}}_k^n)\tilde{\mathbf{F}}(\tilde{\mathbf{u}}_k^n)\big)\big)\Big)\Big\| \\
=& \|\mathbb{E}(\mathbf{u}_k - \tilde{\mathbf{u}}_k) + \mathbb{E}(\frac{1}{2}\mathbf{F}_{\mathbf{u}}^{-1}(\mathbf{u}_k^i)(\mathbf{u}_k - \mathbf{u}_k^i)^T\nabla\mathbf{F}_{\mathbf{u}}(\mathbf{t}_k)(\mathbf{u}_k - \mathbf{u}_k^i)) \\
& - \mathbb{E}(\frac{1}{2}\tilde{\mathbf{F}}_{\mathbf{u}}^{-1}(\tilde{\mathbf{u}}_k^i)(\tilde{\mathbf{u}}_k - \tilde{\mathbf{u}}_k^i)^T\nabla\tilde{\mathbf{F}}_{\mathbf{u}}(\tilde{\mathbf{t}}_k)(\tilde{\mathbf{u}}_k - \tilde{\mathbf{u}}_k^i))\| \\
\leq& \|\mathbb{E}(\mathbf{u}_k - \tilde{\mathbf{u}}_k)\| + \frac{1}{2}\|\mathbf{F}_{\mathbf{u}}^{-1}(\mathbf{u}_k^i)\|\|\nabla\mathbf{F}_{\mathbf{u}}(\mathbf{t}_k)\|\mathbb{E}(\|\mathbf{u}_k - \mathbf{u}_k^i\|^2) \\
& + \frac{1}{2}\|\tilde{\mathbf{F}}_{\mathbf{u}}^{-1}(\tilde{\mathbf{u}}_k^i)\|\|\nabla\tilde{\mathbf{F}}_{\mathbf{u}}(\tilde{\mathbf{t}}_k)\|\mathbb{E}(\|\tilde{\mathbf{u}}_k - \tilde{\mathbf{u}}_k^i\|^2) \\
\leq& \|\mathbb{E}(\mathbf{u}_k - \tilde{\mathbf{u}}_k)\| + M_{\mathbf{u}}K_{\mathbf{u}}\big(\mathbb{E}(\|\mathbf{u}_k - \mathbf{u}_k^i\|^2) + \mathbb{E}(\|\tilde{\mathbf{u}}_k - \tilde{\mathbf{u}}_k^i\|^2)\big).
\end{aligned} \tag{A.12}$$

Due to the local assumption of the initial guesses, then we have the quadratic convergence of Newton's method, namely,

$$\begin{aligned}
\mathbb{E}(\|\mathbf{u}_k - \mathbf{u}_k^i\|) &\leq \alpha\mathbb{E}(\|\mathbf{u}_k - \mathbf{u}_k^{i-1}\|^2), \\
\mathbb{E}(\|\tilde{\mathbf{u}}_k - \tilde{\mathbf{u}}_k^i\|) &\leq \tilde{\alpha}\mathbb{E}(\|\tilde{\mathbf{u}}_k - \tilde{\mathbf{u}}_k^{i-1}\|^2).
\end{aligned} \tag{A.13}$$

Therefore

$$\begin{aligned}
&\|\mathbb{E}(\mathbf{u}_k^{i+1} - \tilde{\mathbf{u}}_k^{i+1})\| \\
\leq& \|\mathbb{E}(\mathbf{u}_k - \tilde{\mathbf{u}}_k)\| + M_{\mathbf{u}}K_{\mathbf{u}}\big(\alpha\mathbb{E}(\|\mathbf{u}_k - \mathbf{u}_k^{i-1}\|^4) + \tilde{\alpha}\mathbb{E}(\|\tilde{\mathbf{u}}_k - \tilde{\mathbf{u}}_k^i\|^4)\big) \\
\leq& \|\mathbb{E}(\mathbf{u}_k - \tilde{\mathbf{u}}_k)\| + M_{\mathbf{u}}K_{\mathbf{u}}\big(\alpha^i\mathbb{E}(\|\mathbf{u}_k - \mathbf{u}_k^0\|^{2^{i+1}}) + \tilde{\alpha}^n\mathbb{E}(\|\tilde{\mathbf{u}}_k - \tilde{\mathbf{u}}_k^i\|^{2^{i+1}})\big).
\end{aligned}$$

By taking the limit on both sides, we have

$$\lim_{i\to\infty}\|\mathbb{E}(\mathbf{u}_k^i - \tilde{\mathbf{u}}_k^i)\| \leq \|\mathbb{E}(\mathbf{u}_k - \tilde{\mathbf{u}}_k)\|.$$

□

Appendix B

Proofs and Formulas in Chapter 6

B.1 A numerical scheme to approximate $\frac{\partial^2 G}{\partial \theta^2}$ in Section 6.5

We use the following finite difference scheme to approximate $\frac{\partial^2 G}{\partial \theta^2}$:

$$\begin{aligned}\frac{\partial^2 G}{\partial \theta^2}(r_{i,j},\theta_j) &= a_1 G(r_{i,j},\theta_j) + a_2 G(r_{i+1,j},\theta_j) + a_3 G(r_{i-1,j},\theta_j) \\ &\quad + a_4 G(r_{i,j+1},\theta_{j+1}) + a_5 G(r_{i+1,j+1},\theta_{j+1}) \\ &\quad + a_6 G(r_{i-1,j+1},\theta_{j+1}) + a_7 G(r_{i,j-1},\theta_{j-1}) \\ &\quad + a_8 G(r_{i+1,j-1},\theta_{j-1}) + a_9 G(r_{i-1,j-1},\theta_{j-1}),\end{aligned} \tag{B.1}$$

where

$$
\begin{cases}
a_2 = \dfrac{h_{j+1}^2(r_{i,j}-r_{i,j+1})+h_{j-1}^2(r_{i,j}-r_{i,j-1})}{3h_j^2(r_{i,j+1}-2r_{i,j}+r_{i,j-1})} \\
\quad + \dfrac{(r_{i,j+1}-r_{i,j})^3+(r_{i,j-1}-r_{i,j})^3}{3h_j^2(r_{i,j+1}-2r_{i,j}+r_{i,j-1})} \\
a_1 = -2-2a_2,\ a_3 = a_2, \\
a_5 = -\dfrac{h_{j+1}^2(r_{i,j}-r_{i,j+1})+h_{j-1}^2(r_{i,j}-r_{i,j-1})}{6h_{j+1}^2(r_{i,j+1}-2r_{i,j}+r_{i,j-1})} \\
\quad - \dfrac{(2r_{i,j}-r_{i,j+1}-r_{i,j-1})^2(r_{i,j}-2r_{i,j+1}+r_{i,j-1})}{6h_{j+1}^2(r_{i,j+1}-2r_{i,j}+r_{i,j-1})}, \\
\quad - \dfrac{3h_{j+1}(r_{i,j}-r_{i,j+1})(2r_{i,j}-r_{i,j+1}-r_{i,j-1})}{6h_{j+1}^2(r_{i,j+1}-2r_{i,j}+r_{i,j-1})} \\
a_6 = -\dfrac{h_{j+1}^2(r_{i,j}-r_{i,j+1})+h_{j-1}^2(r_{i,j}-r_{i,j-1})}{6h_{j+1}^2(r_{i,j+1}-2r_{i,j}+r_{i,j-1})} \\
\quad - \dfrac{(2r_{i,j}-r_{i,j+1}-r_{i,j-1})^2(r_{i,j}-2r_{i,j+1}+r_{i,j-1})}{6h_{j+1}^2(r_{i,j+1}-2r_{i,j}+r_{i,j-1})}, \\
\quad + \dfrac{3h_{j+1}(r_{i,j}-r_{i,j+1})(2r_{i,j}-r_{i,j+1}-r_{i,j-1})}{6h_{j+1}^2(r_{i,j+1}-2r_{i,j}+r_{i,j-1})} \\
a_4 = 1-a_5-a_6, \\
a_8 = -\dfrac{h_{j+1}^2(r_{i,j}-r_{i,j+1})+h_{j-1}^2(r_{i,j}-r_{i,j-1})}{6h_{j-1}^2(r_{i,j+1}-2r_{i,j}+r_{i,j-1})} \\
\quad - \dfrac{(2r_{i,j}-r_{i,j+1}-r_{i,j-1})^2(r_{i,j}-2r_{i,j-1}+r_{i,j+1})}{6h_{j-1}^2(r_{i,j+1}-2r_{i,j}+r_{i,j-1})}, \\
\quad - \dfrac{3h_{j-1}(r_{i,j}-r_{i,j-1})(2r_{i,j}-r_{i,j+1}-r_{i,j-1})}{6h_{j-1}^2(r_{i,j+1}-2r_{i,j}+r_{i,j-1})} \\
a_9 = -\dfrac{h_{j+1}^2(r_{i,j}-r_{i,j+1})+h_{j-1}^2(r_{i,j}-r_{i,j-1})}{6h_{j-1}^2(r_{i,j+1}-2r_{i,j}+r_{i,j-1})} \\
\quad - \dfrac{(2r_{i,j}-r_{i,j+1}-r_{i,j-1})^2(r_{i,j}-2r_{i,j-1}+r_{i,j+1})}{6h_{j-1}^2(r_{i,j+1}-2r_{i,j}+r_{i,j-1})}, \\
\quad + \dfrac{3h_{j-1}(r_{i,j}-r_{i,j-1})(2r_{i,j}-r_{i,j+1}-r_{i,j-1})}{6h_{j-1}^2(r_{i,j+1}-2r_{i,j}+r_{i,j-1})} \\
a_7 = 1-a_8-a_9.
\end{cases}
$$

B.2 Justification for (6.13)

In this subsection, we justify the validity of expansions in (6.13) by showing that the $\mathcal{O}(\epsilon^2)$ terms are small. First we introduce the following Banach space

$$\begin{aligned} X^{l+\alpha} &= \{\rho_1 \in C^{l+\alpha} : \rho_1 \text{ is } 2\pi\text{-periodic}\}, \\ X_1^{l+\alpha} &= \text{closure of the linear space spanned} \\ &\qquad \text{by } \{\cos(j\theta), j = 0, 1, 2, \cdots\} \text{ in } X^{l+\alpha}. \end{aligned}$$

Then we have the following lemma:

Lemma B.2.1. *If $\rho_1 \in C^{3+\alpha}(\mathbb{R})$ and (M, P) is the solution of (6.11), then*

$$\|M - M_s\|_{C^{3+\alpha}(\bar{\Omega}_\varepsilon)} \leq C|\varepsilon|\|\rho_1\|_{C^{3+\alpha}(\mathbb{R})}, \tag{B.2}$$

$$\|P - P_s\|_{C^{1+\alpha}(\bar{\Omega}_\varepsilon)} \leq C|\varepsilon|\|\rho_1\|_{C^{3+\alpha}(\mathbb{R})}, \tag{B.3}$$

where constant C is independent of ε.

Proof. First we derive the equation of $M - M_s$ below

$$\begin{cases} \Delta(M - M_s) - \dfrac{H}{D}(M - M_s) = 0 \quad \text{in} \quad \Omega_\varepsilon, \\ \dfrac{\partial(M - M_s)}{\partial n} + (M - M_s) = 0 \quad \text{on} \quad \Gamma_1, \\ M - M_s = g_1 \quad \text{on} \quad \Gamma_\varepsilon. \end{cases} \tag{B.4}$$

From (6.11) and $M_s(\rho) = 1$, we have

$$\begin{aligned} g_1 &= M(\rho + \varepsilon\rho_1) - M_s(\rho + \varepsilon\rho_1) \\ &= 1 - M_s(\rho + \varepsilon\rho_1) \\ &= M_s(\rho) - M_s(\rho + \varepsilon\rho_1). \end{aligned} \tag{B.5}$$

By differentiating three times with respect to θ, we obtain

$$\|M - M_s\|_{C^{3+\alpha}(\Gamma_\varepsilon)} \leq C|\varepsilon|\|\rho_1\|_{C^{3+\alpha}(\mathbb{R})}. \tag{B.6}$$

The Schauder estimates then indicate that

$$\|M - M_s\|_{C^{3+\alpha}(\bar{\Omega}_\varepsilon)} \leq C|\varepsilon|\|\rho_1\|_{C^{3+\alpha}(\mathbb{R})}.$$

Since $\Gamma_\varepsilon \in C^{3+\alpha}$, the constant C is independent of ε.

Similarly, the equation of $P - P_s$ reads as

$$\begin{cases} -\Delta(P - P_s) = L(M - M_s) \quad \text{in} \quad \Omega_\varepsilon, \\ P - P_s = g_2 \quad \text{on} \quad \Gamma_\varepsilon, \\ \dfrac{\partial(P - P_s)}{\partial n} = 0 \quad \text{on} \quad \Gamma_1, \end{cases}$$

where

$$\begin{aligned} g_2 &= P(\rho + \varepsilon\rho_1) - P_s(\rho + \varepsilon\rho_1) = \gamma\kappa|_{r=\rho+\varepsilon\rho_1} - P_s(\rho + \varepsilon\rho_1) \\ &= \frac{\gamma}{\rho} - \gamma\frac{\varepsilon}{\rho^2}(\rho_1 + \rho_{1\theta\theta}) + \mathcal{O}(\varepsilon^2) - P_s(\rho + \varepsilon\rho_1) \\ &= P_s(\rho) - P_s(\rho + \varepsilon\rho_1) - \gamma\frac{\varepsilon}{\rho^2}(\rho_1 + \rho_{1\theta\theta}) + \mathcal{O}(\varepsilon^2). \end{aligned}$$

We differentiate the above equation along Γ_ε and get

$$\|P - P_s\|_{C^{1+\alpha}(\Gamma_\varepsilon)} \leq C|\varepsilon|\|\rho_1\|_{C^{3+\alpha}(\mathbb{R})}.$$

The Schauder estimates imply

$$\begin{aligned} \|P - P_s\|_{C^{1+\alpha}(\bar{\Omega}_\varepsilon)} &\leq C\|M - M_s\|_{C^{\alpha}(\bar{\Omega}_\varepsilon)} + C\|P - P_s\|_{C^{1+\alpha}(\Gamma_\varepsilon)} \\ &\leq C|\varepsilon|\|\rho_1\|_{C^{3+\alpha}(\mathbb{R})}. \end{aligned}$$

Due to the regularity of M_s and P_s and $\Gamma_\varepsilon \in C^{3+\alpha}$, we conclude the constant C is independent of ε.

□

Next, we proceed to rigorously establish (6.13). Since both M and P are defined on Ω_ε but M_1 and P_1 are defined on Ω only, we transform M_1 and P_1 to Ω_ε by Hanzawa transformation H_ε [114]:

$$(r, \theta) = H_\varepsilon(r', \theta') = (r' + \chi(r' - \rho)\varepsilon\rho_1, \theta')$$

where

$$\chi \in C^\infty, \quad \chi(z) = \begin{cases} 0 \text{ if } |z| \geq \frac{3}{4}\delta_0 \\ 1 \text{ if } |z| < \frac{1}{4}\delta_0 \end{cases}, \quad |\frac{d^k\chi}{dz^k}| \leq \frac{C}{\delta_0^k}$$

and $\delta_0 > 0$ is small. Noticing that H_ε maps Ω onto Ω_ε but keeps the annulus $\{r : \rho + \frac{3}{4}\delta_0 \leq r \leq R\}$ fixed, we set

$$\tilde{M}_1(r,\theta) = M_1(H_\varepsilon^{-1}(r,\theta)), \quad \tilde{P}_1(r,\theta) = P_1(H_\varepsilon^{-1}(r,\theta)). \tag{B.7}$$

Then we establish the following estimates.

Theorem B.2.1. *If $\rho_1 \in C^{3+\alpha}(\mathbb{R})$, (M,P) is the solution of* (6.11)*, and $(\tilde{M}_1, \tilde{P}_1)$ is defined as* (B.7)*, then*

$$\begin{aligned} &\|M - M_s - \varepsilon\tilde{M}_1\|_{C^{3+\alpha}(\bar{\Omega}_\varepsilon)} \leq C|\varepsilon|^2\|\rho_1\|_{C^{3+\alpha}(\mathbb{R})} \\ &\|P - P_s - \varepsilon\tilde{P}_1\|_{C^{1+\alpha}(\bar{\Omega}_\varepsilon)} \leq C|\varepsilon|^2\|\rho_1\|_{C^{3+\alpha}(\mathbb{R})} \end{aligned} \tag{B.8}$$

Proof. First, we compute the first and second derivatives of $\tilde{M}_1$ with respect to both r and θ:

$$\begin{aligned} &\frac{\partial\tilde{M}_1}{\partial r} = \frac{\partial M_1}{\partial r'}\frac{\partial r'}{\partial r}, \quad \frac{\partial\tilde{M}_1}{\partial\theta} = \frac{\partial M_1}{\partial r'}\frac{\partial r'}{\partial\theta} + \frac{\partial M_1}{\partial\theta'}, \\ &\frac{\partial^2\tilde{M}_1}{\partial r^2} = \frac{\partial^2 M_1}{\partial r'^2}(\frac{\partial r'}{\partial r})^2 + \frac{\partial M_1}{\partial r'}\frac{\partial^2 r'}{\partial r^2}, \\ &\frac{\partial^2\tilde{M}_1}{\partial\theta^2} = \frac{\partial^2 M_1}{\partial\theta'^2} + 2\frac{\partial^2 M_1}{\partial r'\partial\theta'}\frac{\partial r'}{\partial\theta} + \frac{\partial^2 M_1}{\partial r'^2}(\frac{\partial r'}{\partial\theta})^2 + \frac{\partial M_1}{\partial r'}\frac{\partial^2 r'}{\partial\theta^2}, \end{aligned}$$

where the derivatives of r' is derived from the Hanzawa transformation. In fact, the first derivatives are

$$\begin{aligned} 1 &= \frac{\partial r'}{\partial r} + \varepsilon\rho_1\chi'(r'-\rho)\frac{\partial r'}{\partial r}, \\ 0 &= \frac{\partial r'}{\partial\theta} + \varepsilon\rho_1\chi'(r'-\rho)\frac{\partial r'}{\partial\theta} + \varepsilon\chi(r'-\rho)\rho_{1\theta} \end{aligned}$$

thus

$$\frac{\partial r'}{\partial r} = \frac{1}{1+\varepsilon\rho_1\chi'(r'-\rho)} \text{ and } \frac{\partial r'}{\partial\theta} = -\frac{\varepsilon\chi(r'-\rho)\rho_{1\theta}}{1+\varepsilon\rho_1\chi'(r'-\rho)}.$$

Similarly, we obtain the second derivatives below

$$\begin{aligned}
\frac{\partial^2 r'}{\partial r^2} &= -\frac{\varepsilon\rho_1\chi''(r'-\rho)}{(1+\varepsilon\rho_1\chi'(r'-\rho))^2}\frac{\partial r'}{\partial r} = -\frac{\varepsilon\rho_1\chi''(r'-\rho)}{(1+\varepsilon\rho_1\chi'(r'-\rho))^3},\\
\frac{\partial^2 r'}{\partial \theta^2} &= -\frac{\varepsilon\chi(r'-\rho)\rho_{1\theta\theta}}{1+\varepsilon\rho_1\chi'(r'-\rho)} + 2\frac{\varepsilon^2\chi(r'-\rho)\chi'(r'-\rho)\rho_{1\theta}^2}{(1+\varepsilon\rho_1\chi'(r'-\rho))^2}\\
&\quad - \frac{(\chi(r'-\rho)\varepsilon\rho_{1\theta})^2\chi''(r'-\rho)\varepsilon\rho_1}{(1+\varepsilon\rho_1\chi'(r'-\rho))^3}.
\end{aligned}$$

Next we consider the estimate of $\phi = M - M_s - \varepsilon\tilde{M}_1$ which satisfies:

$$\begin{cases}
\Delta\phi - \dfrac{H}{D}\phi = \varepsilon^2\tilde{f} \quad \text{in} \quad \Omega_\varepsilon\\
\phi = g \quad \text{on} \quad \Gamma_\varepsilon\\
\dfrac{\partial\phi}{\partial r} + \phi = 0 \quad \text{on} \quad \Gamma_1
\end{cases}$$

where $\tilde{f}$ depends on various terms of Hanzawa transform above and involves up to second order derivatives of ρ_1 and M_1. By applying the Schauder estimate to (6.18), we know $M_1 \in C^{3+\alpha}$ and

$$\|\tilde{f}\|_{C^{1+\alpha}(\bar{\Omega}_\varepsilon)} \le C\|\rho_1\|_{C^{3+\alpha}(\mathbb{R})}.$$

On the boundary Γ_ε, we have

$$\begin{aligned}
g &= M(\rho+\varepsilon\rho_1) - M_s(\rho+\varepsilon\rho_1) - \varepsilon\tilde{M}_1(\rho+\varepsilon\rho_1)\\
&= 1 - M_s(\rho+\varepsilon\rho_1) + \varepsilon\rho_1\frac{\partial M_s(\rho)}{\partial r}\\
&= M_s(\rho) - M_s(\rho+\varepsilon\rho_1) + \varepsilon\rho_1\frac{\partial M_s(\rho)}{\partial r}\\
&= \mathcal{O}(\varepsilon^2)\rho_1.
\end{aligned}$$

By the Schauder theory, we obtain

$$\|M - M_s - \varepsilon\tilde{M}_1\|_{C^{3+\alpha}(\bar{\Omega}_\varepsilon)} \le C|\varepsilon|^2\|\rho_1\|_{C^{3+\alpha}(\mathbb{R})}.$$

Similarly, we write the equation of $\psi = P - P_s - \varepsilon\tilde{P}_1$ as follows

$$\begin{cases} -\Delta\psi = L\phi + \varepsilon^2\tilde{k} \quad \text{in} \quad \Omega_\varepsilon, \\ \psi = f \quad \text{on} \quad \Gamma_\varepsilon, \\ \dfrac{\partial\psi}{\partial r} = 0 \quad \text{on} \quad \Gamma_1, \end{cases}$$

where $\tilde{k}$ is based on various term of Hanzawa transform above and follows

$$\|\tilde{k}\|_{C^{1+\alpha}(\bar{\Omega}_\varepsilon)} \leq C\|\rho_1\|_{C^{3+\alpha}(\mathbb{R})}.$$

Since

$$f = P(\rho + \varepsilon\rho_1) - P_s(\rho + \varepsilon\rho_1) - \varepsilon\tilde{P}_1(\rho + \varepsilon\rho_1),$$

we have

$$\|f\|_{C^{1+\alpha}(\mathbb{R})} \leq C|\varepsilon|^2\|\rho_1\|_{C^{3+\alpha}(\mathbb{R})}.$$

Therefore, by Schauder estimates, we conclude

$$\|P - P_s - \varepsilon\tilde{P}_1\|_{C^{1+\alpha}(\bar{\Omega}_\varepsilon)} \leq C|\varepsilon|^2\|\rho_1\|_{C^{3+\alpha}(\mathbb{R})}.$$

□

Bibliography

[1] LeCun, Y., Y. Bengio, and G. Hinton (2015) "Deep learning," *nature*, **521**(7553), pp. 436–444.

[2] Cohen, N., O. Sharir, and A. Shashua (2016) "On the expressive power of deep learning: A tensor analysis," in *Conference on learning theory*, PMLR, pp. 698–728.

[3] Hornik, K., M. Stinchcombe, and H. White (1989) "Multilayer feedforward networks are universal approximators," *Neural networks*, **2**(5), pp. 359–366.

[4] Cybenko, G. (1989) "Approximation by superpositions of a sigmoidal function," *Mathematics of control, signals and systems*, **2**(4), pp. 303–314.

[5] Jones, L. K. et al. (1992) "A simple lemma on greedy approximation in Hilbert space and convergence rates for projection pursuit regression and neural network training," *The annals of Statistics*, **20**(1), pp. 608–613.

[6] Barron, A. R. (1993) "Universal approximation bounds for superpositions of a sigmoidal function," *IEEE Transactions on Information theory*, **39**(3), pp. 930–945.

[7] Ellacott, S. (1994) "Aspects of the numerical analysis of neural networks," *Acta numerica*, **3**, pp. 145–202.

[8] Pinkus, A. (1999) "Approximation theory of the MLP model," *Acta Numerica 1999: Volume 8*, **8**, pp. 143–195.

[9] Leshno, M., V. Y. Lin, A. Pinkus, and S. Schocken (1993) "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function," *Neural networks*, **6**(6), pp. 861–867.

[10] Nair, V. and G. E. Hinton (2010) "Rectified linear units improve restricted boltzmann machines," in *Icml*.

[11] Shaham, U., A. Cloninger, and R. R. Coifman (2018) "Provable approximation properties for deep neural networks," *Applied and Computational Harmonic Analysis*, **44**(3), pp. 537–557.

[12] Klusowski, J. M. and A. R. Barron (2018) "Approximation by Combinations of ReLU and Squared ReLU Ridge Functions With l^1 and l^0 Controls," *IEEE Transactions on Information Theory*, **64**(12), pp. 7649–7656.

[13] Zhou, D.-X. (2020) "Universality of deep convolutional neural networks," *Applied and computational harmonic analysis*, **48**(2), pp. 787–794.

[14] Shen, Z., H. Yang, and S. Zhang (2019) "Nonlinear approximation via compositions," *Neural Networks*, **119**, pp. 74–84.

[15] Ciarlet, P. G. (2002) *The finite element method for elliptic problems*, SIAM.

[16] Tarela, J. and M. Martinez (1999) "Region configurations for realizability of lattice piecewise-linear models," *Mathematical and Computer Modelling*, **30**(11-12), pp. 17–27.

[17] Arora, R., A. Basu, P. Mianjy, and A. Mukherjee (2016) "Understanding deep neural networks with rectified linear units," *arXiv preprint arXiv:1611.01491*.

[18] He, J., L. Li, J. Xu, and C. Zheng (2018) "Relu deep neural networks and linear finite elements," *arXiv preprint arXiv:1807.03973*.

[19] Meade Jr, A. J. and A. A. Fernandez (1994) "The numerical solution of linear ordinary differential equations by feedforward neural networks," *Mathematical and Computer Modelling*, **19**(12), pp. 1–25.

[20] ——— (1994) "Solution of nonlinear ordinary differential equations by feedforward neural networks," *Mathematical and Computer Modelling*, **20**(9), pp. 19–44.

[21] Gobovic, D. and M. E. Zaghloul (1994) "Analog cellular neural network with application to partial differential equations with variable mesh-size," in *Proceedings of IEEE International Symposium on Circuits and Systems-ISCAS'94*, vol. 6, IEEE, pp. 359–362.

[22] Han, J., A. Jentzen, and E. Weinan (2017) "Overcoming the curse of dimensionality: Solving high-dimensional partial differential equations using deep learning," *arXiv preprint arXiv:1707.02568*, pp. 1–13.

[23] Weinan, E., J. Han, and A. Jentzen (2017) "Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations," *Communications in Mathematics and Statistics*, **5**(4), pp. 349–380.

[24] Khoo, Y., J. Lu, and L. Ying (2017) "Solving parametric PDE problems with artificial neural networks," *arXiv preprint arXiv:1707.03351*.

[25] Hao, W. and C. Zheng (2020) "An adaptive homotopy method for computing bifurcations of nonlinear parametric systems," *Journal of Scientific Computing*, **82**(3), pp. 1–19.

[26] Hao, W., J. Hauenstein, B. Hu, and A. Sommese (2014) "A bootstrapping approach for computing multiple solutions of differential equations," *Journal of Computational and Applied Mathematics*, **258**, pp. 181–190.

[27] Hao, W., R. Nepomechie, and A. Sommese (2013) "Completeness of solutions of Bethe's equations," *Physical Review E*, **88**(5), p. 052113.

[28] ——— (2014) "Singular solutions, repeated roots and completeness for higher-spin chains," *Journal of Statistical Mechanics: Theory and Experiment*, **2014**(3), p. P03024.

[29] Hou, T., J. Lowengrub, and M. Shelley (2001) "Boundary integral methods for multicomponent fluids and multiphase materials," *Journal of Computational Physics*, **169**(2), pp. 302–362.

[30] Friedman, A. and W. Hao (2015) "A mathematical model of atherosclerosis with reverse cholesterol transport and associated risk factors," *Bulletin of mathematical biology*, **77**(5), pp. 758–781.

[31] Hao, W., E. Crouser, and A. Friedman (2014) "Mathematical model of sarcoidosis," *Proceedings of the National Academy of Sciences*, **111**(45), pp. 16065–16070.

[32] Hao, W. and A. Friedman (2014) "The LDL-HDL profile determines the risk of atherosclerosis: a mathematical model," *PLoS ONE*, **9**(3), p. e90497.

[33] Khalil, H. (2002) "Nonlinear systems," *Upper Saddle River*.

[34] Friedman, A. and B. Hu (2006) "Bifurcation from stability to instability for a free boundary problem arising in a tumor model," *Archive for rational mechanics and analysis*, **180**(2), pp. 293–330.

[35] ——— (2007) "Bifurcation for a free boundary problem modeling tumor growth by Stokes equation," *SIAM Journal on Mathematical Analysis*, **39**(1), pp. 174–194.

[36] Dayton, B. and Z. Zeng (2005) "Computing the multiplicity structure in solving polynomial systems," in *Proceedings of the 2005 international symposium on Symbolic and algebraic computation*, ACM, pp. 116–123.

[37] Zeng, Z. (2005) "Computing multiple roots of inexact polynomials," *Mathematics of Computation*, **74**(250), pp. 869–903.

[38] ——— (2004) "Algorithm 835: MultRoot—a Matlab package for computing polynomial roots and multiplicities," *ACM Transactions on Mathematical Software (TOMS)*, **30**(2), pp. 218–236.

[39] Li, T.-Y., T. Sauer, and J. Yorke (1989) "The cheater's homotopy: an efficient procedure for solving systems of polynomial equations," *SIAM Journal on Numerical Analysis*, **26**(5), pp. 1241–1251.

[40] Li, T.-Y. and Z. Zeng (1992) "Homotopy-determinant algorithm for solving nonsymmetric eigenvalue problems," *Mathematics of computation*, **59**(200), pp. 483–502.

[41] Morgan, A. and A. Sommese (1987) "Computing all solutions to polynomial systems using homotopy continuation," *Applied Mathematics and Computation*, **24**(2), pp. 115–138.

[42] ——— (1987) "A homotopy for solving general polynomial systems that respects m-homogeneous structures," *Applied Mathematics and Computation*, **24**(2), pp. 101–113.

[43] Bates, D., J. Hauenstein, A. Sommese, and C. Wampler (2013) *Numerically solving polynomial systems with Bertini*, vol. 25, SIAM.

[44] Leykin, A. (2011) "Numerical algebraic geometry," *Journal of Software for Algebra and Geometry*, **3**(1), pp. 5–10.

[45] Wampler, C. and A. Sommese (2005) *The Numerical solution of systems of polynomials arising in engineering and science*, World Scientific.

[46] Bates, D., J. Hauenstein, A. Sommese, and C. Wampler (2008) "Adaptive multiprecision path tracking," *SIAM Journal on Numerical Analysis*, **46**(2), pp. 722–746.

[47] Bates, J., D.and Hauenstein and A. Sommese (2011) "A parallel endgame," *Contemp. Math*, **556**, pp. 25–35.

[48] Bates, D., J. Hauenstein, A. Sommese, and C. Wampler (2006), "Bertini: Software for numerical algebraic geometry," .

[49] Cauwenberghs, G. (1993) "A fast stochastic error-descent algorithm for supervised learning and optimization," in *Advances in neural information processing systems*, pp. 244–251.

[50] Nguyen, L., H. Schmidt, A. Von Haeseler, and B. Minh (2015) "IQ-TREE: a fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies," *Molecular biology and evolution*, **32**(1), pp. 268–274.

[51] DeVore, R. A. (1998) "Nonlinear approximation," *Acta numerica*, **7**, pp. 51–150.

[52] Brenner, S. and R. Scott (2007) *The mathematical theory of finite element methods*, vol. 15, Springer Science & Business Media.

[53] Nochetto, R. H. and A. Veeser (2011) "Primer of adaptive finite element methods," in *Multiscale and adaptivity: modeling, numerics and applications*, Springer, pp. 125–225.

[54] Lagaris, I. E., A. Likas, and D. I. Fotiadis (1998) "Artificial neural networks for solving ordinary and partial differential equations," *IEEE transactions on neural networks*, **9**(5), pp. 987–1000.

[55] Weinan, E. and B. Yu (2018) "The deep Ritz method: a deep learning-based numerical algorithm for solving variational problems," *Communications in Mathematics and Statistics*, **6**(1), pp. 1–12.

[56] Li, R., T. Tang, and P. Zhang (2001) "Moving mesh methods in multiple dimensions based on harmonic maps," *Journal of Computational Physics*, **170**(2), pp. 562–588.

[57] ——— (2002) "A moving mesh finite element algorithm for singular problems in two and three space dimensions," *Journal of Computational Physics*, **177**(2), pp. 365–393.

[58] Xu, J. (2017) "Deep Neural Networks and Multigrid Methods(Lecture Notes)," .

[59] Liu, G.-R. (2009) *Meshfree methods: moving beyond the finite element method*, CRC press.

[60] Yagawa, G. and T. Yamada (1996) "Free mesh method: a new meshless finite element method," *Computational Mechanics*, **18**(5), pp. 383–386.

[61] Idelsohn, S. R., E. Onate, N. Calvo, and F. Del Pin (2003) "The meshless finite element method," *International Journal for Numerical Methods in Engineering*, **58**(6), pp. 893–912.

[62] Kurková, V. and M. Sanguineti (2002) "Comparison of worst case errors in linear and neural network approximation," *IEEE Transactions on Information Theory*, **48**(1), pp. 264–275.

[63] Mhaskar, H. N. (2004) "On the tractability of multivariate integration and approximation by neural networks," *Journal of Complexity*, **20**(4), pp. 561–590.

[64] Choi, S., D. Harney, and N. Book (1996) "A robust path tracking algorithm for homotopy continuation," *Computers & chemical engineering*, **20**(6), pp. 647–655.

[65] Watson, L., S. Billups, and A. Morgan (1987) "Algorithm 652: HOMPACK: A suite of codes for globally convergent homotopy algorithms," *ACM Transactions on Mathematical Software (TOMS)*, **13**(3), pp. 281–310.

[66] Allgower, E. and K. Georg (2003) *Introduction to numerical continuation methods*, vol. 45, SIAM.

[67] Deuflhard, P. (2011) *Newton methods for nonlinear problems: affine invariance and adaptive algorithms*, vol. 35, Springer Science & Business Media.

[68] Bates, D., J. Hauenstein, A. Sommese, and C. Wampler (2013) *Numerically solving polynomial systems with Bertini*, vol. 25, SIAM.

[69] Sommese, A. and C. Wampler (2005) *The Numerical solution of systems of polynomials arising in engineering and science*, vol. 99, World Scientific.

[70] Leykin, A., J. Verschelde, and A. Zhao (2006) "Newton's method with deflation for isolated singularities of polynomial systems," *Theoretical Computer Science*, **359**(1-3), pp. 111–122.

[71] Hauenstein, J. and C. Wampler (2013) "Isosingular sets and deflation," *Foundations of Computational Mathematics*, **13**(3), pp. 371–403.

[72] Lee, Y., J. Wu, J. Xu, and L. Zikatanov (2007) "Robust subspace correction methods for nearly singular systems," *Mathematical Models and Methods in Applied Sciences*, **17**(11), pp. 1937–1963.

[73] Xu, J. (1992) "Iterative methods by space decomposition and subspace correction," *SIAM review*, **34**(4), pp. 581–613.

[74] Xu, J., L. Chen, and R. Nochetto (2009) "Optimal multilevel methods for H (grad), H (curl), and H (div) systems on graded and unstructured grids," *Multiscale, nonlinear and adaptive approximation*, **1**(1), pp. 599–659.

[75] Bates, D., J. Hauenstein, and A. Sommese (2011) "Efficient path tracking methods," *Numerical Algorithms*, **58**(4), pp. 451–459.

[76] Morgan, A., A. Sommese, and C. Wampler (1992) "A power series method for computing singular solutions to nonlinear analytic systems," *Numerische Mathematik*, **63**(1), pp. 391–409.

[77] Fischer, G. (2001) *Plane algebraic curves*, vol. 15, American Mathematical Soc.

[78] Huber, B. and J. Verschelde (1998) "Polyhedral end games for polynomial continuation," *Numerical Algorithms*, **18**(1), pp. 91–108.

[79] Buffoni, B., J. Toland, and J. F. Toland (2003) *Analytic theory of global bifurcation: an introduction*, Princeton University Press.

[80] Chicone, C. (1994) "Lyapunov-Schmidt reduction and Melnikov integrals for bifurcation of periodic solutions in coupled oscillators," *Journal of differential equations*, **112**(2), pp. 407–447.

[81] Hao, W., J. Hauenstein, B. Hu, Y. Liu, A. Sommese, and Y.-T. Zhang (2012) "Bifurcation for a free boundary problem modeling the growth of a tumor with a necrotic core," *Nonlinear Analysis: Real World Applications*, **13**(2), pp. 694–709.

[82] Lee, T.-L., T.-Y. Li, and C.-H. Tsai (2008) "HOM4PS-2.0: a software package for solving polynomial systems by the polyhedral homotopy continuation method," *Computing*, **83**(2), pp. 109–133.

[83] Verschelde, J. (1999) "Algorithm 795: PHCpack: A general-purpose solver for polynomial systems by homotopy continuation," *ACM Transactions on Mathematical Software (TOMS)*, **25**(2), pp. 251–276.

[84] Chen, X., R. Hambrock, and Y. Lou (2008) "Evolution of conditional dispersal: a reaction-diffusion-advection model," *J Math Biol*, **57**(3), pp. 361–386.

[85] Hambrock, R. and Y. Lou (2009) "The evolution of conditional dispersal strategies in spatially heterogeneous habitats," *Bull. Math. Biol.*, **71**(8), pp. 1793–1817.

[86] Hao, W., K. Y. Lam, and Y. Lou (2017) "Concentration Phenomena in an Integro-PDE Model for Evolution of Conditional Dispersal," *Indiana University Mathematics Journal*, p. to appear.

[87] Lam, K. Y. and Y. Lou (2014) "Evolution of conditional dispersal: evolutionarily stable strategies in spatial models," *J Math Biol*, **68**(4), pp. 851–877.

[88] Arnold, L. (1974) "Stochastic differential equations," *New York*.

[89] Hao, W. and C. Xue (2020) "Spatial pattern formation in reaction–diffusion models: a computational approach," *Journal of Mathematical Biology*, pp. 1–23.

[90] Ross, R. (1999) "Atherosclerosisan inflammatory disease," *New England journal of medicine*, **340**(2), pp. 115–126.

[91] Stoll, G. and M. Bendszus (2006) "Inflammation and atherosclerosis: novel insights into plaque formation and destabilization," *Stroke*, **37**(7), pp. 1923–1932.

[92] Childbirth, H. C., B. Care, and F. Assistance "Carotid Artery Stenosis," .

[93] Moreno, P. R. (2010) "Vulnerable plaque: definition, diagnosis, and treatment," *Cardiology clinics*, **28**(1), pp. 1–30.

[94] Drake, R., A. W. Vogl, and A. W. Mitchell (2009) *Gray's Anatomy for Students E-Book*, Elsevier Health Sciences.

[95] Steve, P., P. Michelle, and K. Adele (2003), "The Leeds Histology Guide," https://www.histology.leeds.ac.uk/circulatory/arteries.php, accessed 3-July-2020.

[96] Bentzon, J. F., F. Otsuka, R. Virmani, and E. Falk (2014) "Mechanisms of plaque formation and rupture," *Circulation research*, **114**(12), pp. 1852–1866.

[97] Barter, P. (2005) "The role of HDL-cholesterol in preventing atherosclerotic disease," *European heart journal Supplements*, **7**(suppl_F), pp. F4–F8.

[98] Fernandez, M. and D. Webb (2008) "The LDL to HDL cholesterol ratio as a valuable tool to evaluate coronary heart disease risk," *Journal of the American College of Nutrition*, **27**(1), pp. 1–5.

[99] Lemieux, I., B. Lamarche, C. Couillard, A. Pascot, B. Cantin, J. Bergeron, G. Dagenais, and J. Després (2001) "Total cholesterol/HDL cholesterol ratio vs LDL cholesterol/HDL cholesterol ratio as indices of ischemic heart disease risk in men: the Quebec Cardiovascular Study," *Archives of internal medicine*, **161**(22), pp. 2685–2692.

[100] Calvez, V., A. Ebde, N. Meunier, and A. Raoult (2009) "Mathematical modelling of the atherosclerotic plaque formation," in *ESAIM: Proceedings*, vol. 28, EDP Sciences, pp. 1–12.

[101] Cobbold, C., J. Sherratt, and S. Maxwell (2002) "Lipoprotein oxidation and its significance for atherosclerosis: a mathematical approach," *Bulletin of mathematical biology*, **64**(1), pp. 65–95.

[102] McKay, C., S. McKee, N. Mottram, T. Mulholland, S. Wilson, S. Kennedy, and R. Wadsworth (2005) "Towards a model of atherosclerosis," *University of Strathclyde*, pp. 1–29.

[103] Friedman, A., W. Hao, and B. Hu (2015) "A free boundary problem for steady small plaques in the artery and their stability," *Journal of Differential Equations*, **259**(4), pp. 1227–1255.

[104] Yl-Herttuala, S., B. A. Lipton, M. E. Rosenfeld, T. Srkioja, T. Yoshimura, E. J. Leonard, J. L. Witztum, and D. Steinberg (1991) "Expression of monocyte chemoattractant protein 1 in macrophage-rich areas of human and rabbit atherosclerotic lesions," *Proc. Natl. Acad. Sci. U.S.A.*, **88**(12), pp. 5252–5256.

[105] Roy, A., U. Saqib, K. Wary, and M. S. Baig (2020) "Macrophage neuronal nitric oxide synthase (NOS1) controls the inflammatory response and foam cell formation in atherosclerosis," *Int. Immunopharmacol.*, **83**, p. 106382.

[106] Orekhov, A. N., T. Pushkarsky, Y. Oishi, N. G. Nikiforov, A. V. Zhelankin, L. Dubrovsky, V. J. Makeev, K. Foxx, X. Jin, H. S. Kruth, I. A. Sobenin, V. N. Sukhorukov, E. R. Zakiev, A. Kontush, W. Le Goff, and M. Bukrinsky (2018) "HDL activates expression of genes stimulating cholesterol efflux in human monocyte-derived macrophages," *Exp. Mol. Pathol.*, **105**(2), pp. 202–207.

[107] Meng, X. B., T. Zhu, D. H. Yang, W. Liang, G. B. Sun, and X. B. Sun (2019) "Xuezhitong capsule, an extract of Allium macrostemon Bunge, exhibits reverse cholesterol transport and accompanies high-density lipoprotein levels to protect against hyperlipidemia in ApoE-/- mice," *Ann Transl Med*, **7**(11), p. 239.

[108] Libby, P., P. M. Ridker, and A. Maseri (2002) "Inflammation and atherosclerosis," *Circulation*, **105**(9), pp. 1135–1143.

[109] Cilla, M., M. A. Martinez, and E. Pena (2015) "Effect of Transmural Transport Properties on Atheroma Plaque Formation and Development," *Ann Biomed Eng*, **43**(7), pp. 1516–1530.

[110] Cilla, M., E. Pena, and M. A. Martinez (2014) "Mathematical modelling of atheroma plaque formation and development in coronary arteries," *J R Soc Interface*, **11**(90), p. 20130866.

[111] Govindaraju, K., S. Kamangar, I. Badruddin, G. Viswanathan, A. Badarudin, and N. Salman Ahmed (2014) "Effect of porous media of the stenosed artery wall to the coronary physiological diagnostic parameter: a computational fluid dynamic analysis," *Atherosclerosis*, **233**(2), pp. 630–635.

[112] Gilbarg, D. and N. S. Trudinger (2015) *Elliptic partial differential equations of second order*, springer.

[113] Crandall, M. G. and P. H. Rabinowitz (1971) "Bifurcation from simple eigenvalues," *Journal of Functional Analysis*, **8**(2), pp. 321–340.

[114] Hanzawa, E.-i. (1981) "Classical solutions of the Stefan problem," *Tohoku Mathematical Journal, Second Series*, **33**(3), pp. 297–335.

Vita

Chunyue Zheng

Chunyue Zheng received her Bachelor of Science degree in mathematics at Wuhan University, China, in 2015. That year she began her doctoral degree in applied mathematics at Pennsylvania State University.