NUMERICAL ANALYSIS OF LINEAR AND NONLINEAR SCHRÖDINGER EQUATIONS ON QUANTUM GRAPHS

Gracie Conte

A dissertation submitted to the faculty at the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Mathematics in the College of Arts and Sciences.

Chapel Hill
2022

Approved by:

Jeremy L. Marzuola

Yaiza Canzani

Roy Goodman

Christopher Jones

Katherine Newhall

ABSTRACT

Gracie Conte: Numerical Analysis of Linear and Nonlinear Schrödinger Equations on Quantum Graphs
(Under the direction of Jeremy L. Marzuola)

A wide variety of problems in quantum mechanics can be modeled with Schrödinger-type equations on quantum graphs. More specifically, graphs are useful for simplifying models of physical systems that feature nano-scaled branching structures. Since analytical solutions can only be found for a few trivial cases, it is necessary to consider how to accurately solve this type of problem numerically. While a wide variety of tools exist to solve these types of partial differential equations on lines, this is not well studied in the case of graphs - one critical difference between the two cases being that graphs require more complicated boundary conditions. This paper utilizes a new approach to include the boundary conditions in the discretized operator that preserves high levels of accuracy. Thus, a proper time evolution scheme must work in conjunction with a spatial operator that has incorporated boundary conditions and preserve the accuracy of our spatial component, and this is accomplished by implementing methods from differential algebraic equations. We study the numerical computation of linear and nonlinear states for Schrödinger equations on graphs, as well as numerically compute their linear stability properties. The latter result has implications for the future study of relative periodic orbits on graphs. All numerical components are being adapted into a MATLAB software package called QGLAB jointly with Roy Goodman on Github.

For my son, Landon, who loved me on my worst days
and saved all his many needs for my good days.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Jeremy Marzuola, for the many *many* hours he has spent explaining and re-explaining the broad variety of topics necessary for my dissertation. Jeremy has been the picture of patience throughout the entirety of our time working together, always knowing when he needed to teach me to think critically about the problem versus when he should sort through the dirty details with me. But beyond serving as an advisor in research, he also helped me navigate life as a graduate student and parent. He created a safe environment for me to flourish in, showing an inconceivable amount of support and kindness. I do not think I could have done this without him.

I would like to thank the remaining members of my committee: Yaiza Canzani, Roy Goodman, Chris Jones, and Katie Newhall. Along with being on my committee, Roy Goodman has been a close collaborator since I took my first step in my research. He has provided a substantial amount of guidance in my understanding of the problem, writing style, and my ability to think like an effective researcher. Katie was an irreplaceable resource during my early years in the program, providing me with an ample amount of help as I studied for comprehensive exams, and I greatly appreciate her assistance. Yaiza and Chris have both provided kind and constructive feedback and made sure I am able to answer important questions pertaining to my dissertation problem.

I would also like to thank my undergraduate advisor, David Zeigler, for exuding an astounding amount of passion for mathematics and making every class my new favorite class. He played an instrumental role in my finding passion for numerical partial differential equations. Finally, I would like to thank particularly influential instructors I had: Andres Domokos, Kimberly Elce, and Tracy Hamilton, for encouraging me to pursue a PhD and being supportive beyond belief.

Lastly, I would like to thank all my fellow graduate students, especially Aaron Barrett, Kate Daftari, and Collin Kofroth. It has been very helpful for me to talk through problems in my research and to have a supportive audience at all my presentations. I am a very nervous speaker, so having such an encouraging audience has given me the best possible environment to be successful.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

1.1 A Brief Overview

Stationary Schrödinger-type equations on graphs are what gave origin to the name "quantum graph." However, a wide variety of applications for other equations in mathematical physics have been also been studied on metric graphs including the heat [10, 34, 43] and wave [14, 27] equations. To better understand the fundamentals of these equations on quantum graphs, this thesis studies the Laplacian and equations built around the Laplacian in this context. In particular, we investigate linear spectral theory and linear Poisson problems before delving into nonlinear elliptic partial differential equations and culminating with nonlinear time-dependent problems.

Since analytical solutions can only be found for a few simple cases, it is necessary to consider how to accurately solve this type of problem numerically. Driven by a need to work with a variety of Hamiltonian operators on such a broad spectrum of graphs, a software package was developed in MATLAB called QGLAB over the course of this dissertation in collaboration with Roy Goodman. Further details about the package can be found in [40].

1.2 Physical Motivation

The Schrödinger equation plays an instrumental role in quantum mechanics, and its discovery was a significant landmark in the development of the subject. It is a linear partial differential equation that describes the wave function or state function of quantum mechanical systems. However, while it provides a way to calculate the wave function of a system and how it changes dynamically in time, it does not directly say what the wave function represents. Instead of directly giving the position, its output is interpreted as probability amplitude by squaring the absolute value of the solution which tells us the likelihood that a quantum object is at a given location. The most well-known application for the Schrödinger equation is to describe the motion of individual particles. Another main application is using the eigenvalues of the operator to find the energy levels of a system. Using Schrödinger operators in conjunction with quantum graphs allows us to make approximations for

waves propagating in thin structures.

Quantum graphs appear in various fields such as solid state physics, quantum chemistry, nanotechnology, and wave physics, as there are many scenarios in these fields in which a wave is propagated through a quasi-one-dimensional system. This class of problems can be greatly simplified (though not to a trivial extent) through the use of quantum graphs. Two key uses for quantum graphs will be described; the first use is a direct application of graphs to model physical systems, and the second regards the study of complex quantum systems in which graphs are a simpler setting.

Thin branching systems are ideal physical scenarios to model from the perspective of quantum graphs. For instance, the phenomenon of a quantum wire circuit could be described as a tube with a Y-intersection whose radius is shrunk infinitesimally until only an electron can flow through it and has been studied in the context of quantum graphs by [11, 30, 61, 70, 71, 72]. Other scenarios include: thin branching wave guides which guide waves with minimal loss of energy by restricting the transmission of energy to one direction [6, 46, 59], Anderson localization which studies the slowing down of light through the use of a photonic crystal [45, 46, 54, 55, 67] which can be represented by a Bethe lattice [52, 53], and the spectra of carbon nanotubes which have garnered interest due to their exceptional mechanical properties [9, 56, 57, 60].

Now, let us discuss a few brief examples of complex quantum systems in which quantum graphs provide a more simple framework to study the problem and produce insightful results. Bose-Einstien condensates are a state of cold matter whose behavior is studied on complex random networks through the use of graphs [1, 20, 22, 33, 64]. The quantum Hall effect, a quantized version of the Hall effect, is studied on combinatorial and quantum graphs to answer questions in quantum transport [23, 37, 46]. The Casimir effect, which is a force in quantum field theory, has been studied theoretically and experimentally in various geometries, but graphs have been used to study the "piston" geometry in particular [34, 35].

These are but a few physically motivated systems where quantum graphs can be utilized to assist in the analysis of the problem. Further examples and references are provided in [17].

1.3 Mathematical Motivation

The study of the nonlinear Schrödinger equation on quantum graphs has garnered much attention in recent history for a diverse range of topics, ranging from the investigation of ground state energy [4, 5, 39, 62] and standing wave solutions [18, 47, 48, 69], to the study of thin manifolds converging

to quantum graphs [31, 41, 65].

The interest was initially driven as a model of many physical systems, most notably as models of light propagation in optical fibers with junctions and for Bose-Einstein condensates under mean-field approximation. In the context of light propagation, it models nearly-monochromatic guided optical beams in weakly coupled waveguides with both linear and nonlinear Kerr refractive indices and no absorption [21, 36]. Meanwhile, Bose-Einstein condensates are a state of matter occurring when individual atoms are cooled to extremely low temperatures and then the atoms coalesce into a quantum mechanical entity that can be described by a wave function on a near macroscopic scale [42, 75, 76]. Subsequent to these studies, there has been significant mathematical interest in topics such as the existence, stability, and variational properties of the solutions [3, 49, 50, 66] and the impact of a graph's geometry on the solution [2, 44, 58, 69].

The effects of a graph's geometry on nonlinear states is a topic that has been explored in a number of settings. In particular, it was studied in [18] on the dumbbell graph then generalized from a computational standpoint in [39]. Similarities were found to how the nonlinear states bifurcate in certain graph settings to how they bifurcate for multiple-well potentials. Such similarities have been studied in the case of the double-well potential [63] and the triple-well potential [38]. It is these multiple-well settings in which periodic orbits arise that can be found using the excited states [38, 80]. Through accurate numerical computations of excited states, solving time dependent nonlinear equations and analysis of linear stability operators, this dissertation lays the numerical foundation for searching for relative-periodic solutions in the graph setting similar to those found for multiple-well potentials.

CHAPTER 2

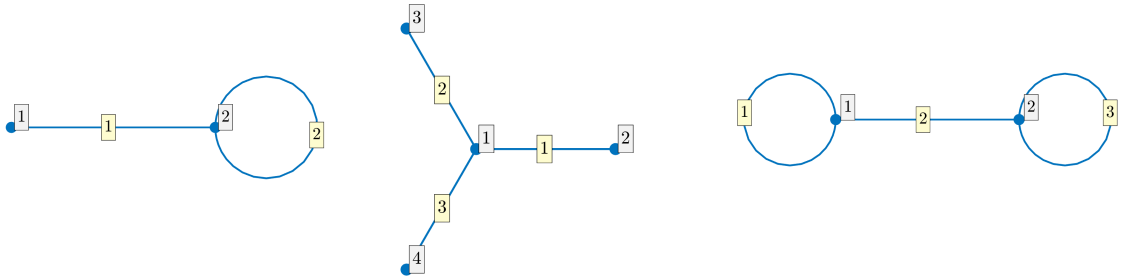
Quantum Graphs

This chapter focuses on laying the foundations of quantum graphs. After discussing fundamental definitions, we analyze the spectrum of the Laplacian on quantum graphs and extend the work done in [15] by using more general boundary conditions before closing the chapter with a few illustrative examples.

2.1 Defining a Quantum Graph

A **graph** Γ consists of a finite or countably infinite set of vertices $V = \{v_i\}$ and a set $\mathcal{E} = \{e_j\}$ of edges connecting the vertices and is denoted $(V, \mathcal{E})$. Each edge e_m can be identified with a pair (v_i, v_k) of vertices and two edges that have a vertex in common are said to be **adjacent**. Loops and multiple edges between vertices will be allowed so we can say that $\mathcal{E} \subset V \times V$. We also denote by $\mathcal{E}_v$ the set of all edges incident to the vertex v (i.e., containing v). It is assumed that the **degree** $d_v = |\mathcal{E}_v|$ of any vertex v is finite and positive. We hence exclude vertices with no edges coming in or going out. This is natural, since such vertices are irrelevant for the purposes of quantum graph.

A variety of graphs will be used in the coming sections. While the dumbbell will be the main example of this paper, the purpose of the QGLAB package is to work on all graphs so we will take a moment to introduce the three most pervasively used graphs in Figure 2.1.



(a) Lollipop (b) Star (c) Dumbbell

Figure 2.1: Common graphs

A **directed metric graph** (as depicted in Figure 2.2) assigns each edge, e_m, a finite length and

imposes a coordinate x that increases from 0 to ℓ_m as it traverses the edge from the initial vertex to the ending one. A **quantum graph** is a directed metric graph that has been equipped with a Schrödinger-type operator. But before discussing this, one should review the Schrödinger equation.

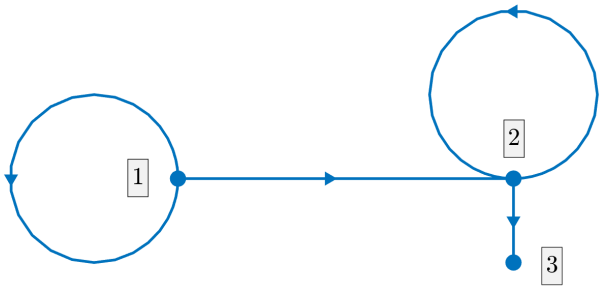


Figure 2.2: A directed graph with three vertices and four edges. Vertex 1 has degree 3, vertex 2 has degree 4, and vertex 3 has degree 1.

2.1.1 The Schrödinger Equation

The most general form of the cubic nonlinear Schrödinger equation is

$$iu_t = -u_{xx} + V(x)u + \sigma|u|^2u, \tag{2.1}$$

where $x \in \mathbb{R}$, $V(x)$ is the potential (which can be zero), and $\sigma \in \mathbb{R}$. In the case that $\sigma = 0$, this is the **linear Schrödinger equation** and if σ is nonzero, this is the **nonlinear Schrödinger equation** (NLS). The nonlinearity is said to be attractive when $\sigma < 0$ and repulsive when $\sigma > 0$; the affiliated equations are called the **focusing** and **defocusing NLS**, respectively. A more in depth discussion of the different behaviors can be found in [25].

The operator in the Schrödinger equation that does not pertain to the nonlinearity is the **Schrödinger operator**

$$L = -\partial_{xx} + V(x). \tag{2.2}$$

The Schrödinger equation is known to have **solitons** (or **traveling wave solutions**) of the form

$$u(x,t) = e^{-i\mu t}u(x), \tag{2.3}$$

where $u(x)$ is a real-valued localized function that must solve the **stationary problem**

$$u'' - V(x)u - \sigma u^3 = -\mu u, \tag{2.4}$$

as it is independent from time. (See [73, §1.3] for further information pertaining to the dynamics.) In the instance that $\sigma = 0$, we are left with the **Poisson** or **eigenvalue problem**

$$u'' - V(x)u = -\mu u. \tag{2.5}$$

Now that the Schrödinger equation (and thus the Schrödinger operator) is understood, we return to an important topic pertaining to graphs: defining how solutions must behave at the vertices.

2.1.2 Vertex Conditions

Vertices may be divided into two groups depending on how many edges they are connected to. A vertex of degree one is called a **leaf vertex**. In other words, it is a vertex connected to exactly one edge, no loops. An **internal vertex** has degree greater than one. Leaf nodes have **boundary conditions** while conditions for internal nodes are called **matching conditions**.

Matching conditions break into two subcategories. First, is the **continuity condition**. This says the solution from each edge must be equal at a shared vertex. In other words, if a vertex v_n joins edges e_j and e_k, then

$$u_j(v) = u_k(v). \tag{2.6}$$

There is also the **Kirchhoff boundary condition** which is defined by

$$\sum_{j=1}^{d_v} \omega_j u_j'(v_n) + \beta u_1(v_n) = \phi, \qquad \forall v_n \in \mathcal{V} \tag{2.7}$$

where ω_j is the **weight** of j^{th} edge and $\beta \in \mathbb{R}$. When $\beta = \phi = 0$, we call this the **current conservation condition**.

The boundary condition found at the leaf vertices is, in its most general form, a **Robin condition**

$$u_m'(v) + \alpha_m u_m(v) = 0 \tag{2.8}$$

where $\alpha_m \in \mathbb{R}$ for all $m = 1, ..., |\mathcal{E}|$. Note that the Robin condition is just a more specific case of the Kirchhoff condition. Some specific variations of the Robin condition that are most common are when $\alpha = 0$ which yields the **Neumann condition**. The **Dirichlet condition**, $u_m(v) = 0$, is obtained when $\alpha \to \infty$.

2.2 Spectrum of the Laplacian

Now that the Laplacian operator has been introduced in the context of quantum graphs, a natural problem is to investigate its spectrum. For a compact graph, the spectrum consists solely of discrete points, so the problem consists of characterizing the set of eigenvalues λ and eigenfunctions u such that

$$\Delta u = \lambda u. \tag{2.9}$$

This is known to have the analytic solution

$$u_m(x) = a_m e^{ikx} + b_m e^{ik(\ell_m - x)}, \qquad m = 1, 2, \ldots, |\mathcal{E}| \tag{2.10}$$

where $0 < \ell_m < \infty$ is the length of the m^{th} edge and $\lambda = k^2$. (This is done for ease of notation and to ensure that $\lambda > 0$.)

Since (2.10) is defined on each of our edges, there are $2|\mathcal{E}|$ a and b constants and the enforcement of the boundary conditions on each edge (as previously discussed) will result in $2|\mathcal{E}|$ equations. Solving these equations for each a_m and b_m in terms of the other a_m's and b_m's will generate a system which can be solved for k (and thus λ). This problem was solved for Neumann boundary conditions [15, §5.2]; a more generalized result is proven here for general Robin and Kirchhoff boundary conditions.

To begin, it is necessary to rewrite (2.10) with new notation that will allow us to account for general edge orientation (whether or not an edge is incoming or outgoing from a vertex v_n). Thus, we define

$$u_m = \tilde{a}_m e^{ikx} + \tilde{b}_m e^{ik(\ell_m - x)}, \tag{2.11}$$

where

$$\tilde{a}_m = \begin{cases} a_m & \text{if } u_m(v_n) = u_m(0) \\ b_m & \text{if } u_m(v_n) = u_m(\ell_m) \end{cases} \qquad \text{and} \qquad \tilde{b}_m = \begin{cases} b_m & \text{if } u_m(v_n) = u_m(0) \\ a_m & \text{if } u_m(v_n) = u_m(\ell_m) \end{cases}.$$

For the vertex conditions at v_n, first consider the Robin condition proposed in (2.8), which can be rewritten in the following way upon substituting in (2.11):

$$\sum_{j=1}^{d_{v_n}} (\tilde{a}_j - \tilde{b}_j) + \alpha_m(\tilde{a}_m + \tilde{b}_m e^{ik\ell_m}) = 0, \tag{2.12}$$

where $1 \leq m \leq d_{v_n}$. For any v_n with $d_{v_n} > 1$, there is also the continuity condition

$$\tilde{a}_1 + b_1 e^{ikx} = \cdots = \tilde{a}_d + \tilde{b}_d e^{ikx}. \tag{2.13}$$

For any n such that $1 \leq n \leq d_{v_n}$, continuity implies that

$$\sum_{j=1}^{d_{v_n}} (\tilde{a}_j + \tilde{b}_j e^{ik\ell_j}) = d_{v_n}(\tilde{a}_n + \tilde{b}_n e^{ik\ell_n}). \tag{2.14}$$

Subtracting (2.14) from (2.12) and performing algebraic simplification yields the following:

$$\sum_{j=1}^{d_{v_n}} (\tilde{a}_j - \tilde{b}_j e^{ik\ell_j}) + \alpha_m(\tilde{a}_m + b_m e^{ik\ell_m} - \sum_{j=1}^{d_{v_n}} (\tilde{a}_j + \tilde{b}_j e^{ik\ell_j}) = -d_{v_n}(\tilde{a}_n + \tilde{b}_n e^{ik\ell_j})$$

$$2\sum_{j=1}^{d_{v_n}} (\tilde{b}_j e^{ik\ell_j}) + \alpha_m(\tilde{a}_m + b_m e^{ik\ell_m} = d_{v_n}(\tilde{a}_n + \tilde{b}_n e^{ik\ell_j}),$$

which allows us to solve for $\tilde{a}_n$:

$$\tilde{a}_n = -\tilde{b}_n e^{ik\ell_n} + \frac{2}{d_{v_n}} \sum_{j=1}^{d_{v_n}} (\tilde{b}_j e^{ik\ell_j}) - \frac{\alpha_m}{d_{v_n}}(a_m + b_m e^{ik\ell}) \tag{2.15}$$

Now that we see how to solve for one variable, we can use this to construct a system of equations whose matrix form is

$$\mathbf{S}(k)\mathbf{D}(k)\mathbf{a} = \mathbf{a}, \tag{2.16}$$

where

$$\mathbf{a} = (a_1, \ldots, a_{|\mathcal{E}|}, b_1, \ldots, b_{|\mathcal{E}|})^T.$$

To define $\mathbf{S}(k)$ and $\mathbf{D}(k)$, it is necessary to consider the edge with respect to a given vertex v_n, instead of its place in the graph as a whole. In particular, think of each edge as having a "forward" direction (its original orientation) and a "backward" direction. The reversal of edge m is denoted as $\tilde{m}$. (Note that this is consistent with our definition of (2.11)!) $\mathbf{D}(k)$ is a diagonal matrix with entries are given by

$$\mathbf{D}_{jj}(k) = e^{ik\ell_j}, \tag{2.17}$$

where j cycles first through the "forward" edges and then through the "backwards" ones. $\mathbf{S}(k)$ can then be defined utilizing the idea of an edge following another edge. We say that e'_m **follows** e_m if the end-vertex of e_m is the starting vertex of e'_m. In this frame of mind, we get

$$\mathbf{S}(k)_{j',j} = \begin{cases} \dfrac{2ik}{d_{v_n} ik + \alpha_n} - 1 & \text{if } j' = \tilde{j} \text{ at } v_n \\ \dfrac{2ik}{d_{v_n} ik + \alpha_n} & \text{if } j' \text{ follows } j \text{ and } j' \neq \tilde{j} \text{ at } v_n \\ 0 & \text{otherwise.} \end{cases} \tag{2.18}$$

This may be difficult to envision generally, but there will be several example calculations in the following section to illuminate this formulation.

Since both $\mathbf{S}(k)$ and $\mathbf{D}(k)$ are in terms of k, this differs from the results in [15, Sec. 5.2] for Neumann boundary conditions where the matrix $\mathbf{S}$ is independent of k. But similar in a similar spirit, (2.16) is rearranged as

$$(\mathbf{I} - \mathbf{S}(k)\mathbf{D}(k))\mathbf{a} = 0 \tag{2.19}$$

where $\mathbf{I}$ is the identity matrix. Clearly, $\mathbf{I} - \mathbf{S}(k)\mathbf{D}(k)$ is singular so

$$\det(\mathbf{I} - \mathbf{S}(k)\mathbf{D}(k)) = 0.$$

In fact, this determinant

$$\Sigma(k) \coloneqq \det(\mathbf{I} - \mathbf{S}(k)\mathbf{D}(k)) \tag{2.20}$$

is better known as the **secular determinant**, and it allows us to solve for k and obtain our eigenvalues. As a consequence of $\mathbf{S}$ depending on k, it is necessary to modify the proof in [15, §5.4] that the secular determinant produces purely real eigenvalues despite the fact that $\Sigma(k)$ is itself a complex-valued function. The following are the necessary pieces to prove the desired statement.

Lemma 2.1. Let

$$L = \sum_{e \in E} l_e$$

denote the total length of the graph. If k is real, then the phase angle of D is equal to its determinant.

Proof. This falls out immediately by calculating the determinant as follows:

$$\det(\mathbf{D}) = \prod_{e \in E} e^{2ikl_e} = e^{2ikL}.$$

□

Lemma 2.2. Let n_{leaf} be the number of leaf vertices on a given graph, n_{int} be the number internal vertices, and m_j be the degree of the internal vertices. Further, set

$$\theta_R = \sum_{j=1}^{n_{\text{leaf}}} \arctan\left(\frac{k}{\alpha_j}\right) \qquad \text{and} \qquad \theta_K = \sum_{j=1}^{n_{int}} \arctan\left(\frac{m_j k}{\beta_j}\right),$$

where α_j is the Robin boundary condition at the leaf vertices and β_j is the Kirchhoff boundary condition at the internal vertices. The phase angle of $\mathbf{S}$ is equal to its determinant.

Proof. We begin by calculating the determinant of $\mathbf{S}$ and find the following:

$$\det(\mathbf{S}) = \prod_{j=1}^{n_{\text{leaf}}} \frac{\alpha_j + ik}{\alpha_j - ik} \prod_{j=1}^{n_{int}} \frac{m_j k + i\beta_j}{m_j k - i\beta_j}. \tag{2.21}$$

Let $z_j = \alpha_j + ik = r_j e^{i\theta_j}$ and $y_j = m_j k + i\beta_j = \rho_j e^{i\phi_j}$ where

$$\theta_j = \arg(z_j) \quad \text{and} \quad \phi_j = \arg(y_j).$$

Now z_j and y_j can be used in (2.21) to obtain the following desired result:

$$\begin{aligned}
\det(\mathbf{S}) &= \prod_{j=1}^{n_{\text{leaf}}} \frac{z_j}{\overline{z_j}} \prod_{j=1}^{n_{int}} \frac{y_j}{\overline{y_j}} \\
&= \prod_{j=1}^{n_{\text{leaf}}} \frac{r_j e^{i\theta_j}}{r_j e^{-i\theta_j}} \prod_{j=1}^{n_{int}} \frac{\rho_j e^{i\phi_j}}{\rho_j e^{-i\phi_j}} \\
&= \prod_{j=1}^{n_{\text{leaf}}} e^{2i\theta_j} \prod_{j=1}^{n_{int}} e^{2i\phi_j} \\
&= e^{2i\sum_{j=1}^{n_{\text{leaf}}} \theta_j} e^{2i\sum_{j=1}^{n_{int}} \phi_j} \\
&= e^{2i(\Theta_R + \Theta_K)}.
\end{aligned}$$

□

Lemma 2.3. **S** is a unitary matrix.

Proof. We will use the fact that a matrix is unitary if any two unique rows are orthogonal and if the dot product of each row with itself is 1.

If two edges are not adjacent to one another, then their corresponding rows from the continuity and Kirchhoff conditions will have zeros wherever the other one has constants by construction. Hence, their dot product must be zero. The harder case is when edges are adjacent.

Suppose we have edges adjacent via a vertex of degree $m > 1$. We then have unique rows of **S**, x and y, whose dot product is as follows:

$$\begin{aligned}
\vec{x} \cdot \vec{y} &= \vec{x}^T \bar{\vec{y}} \\
&= \left(\frac{\frac{\beta}{ik} - (m-2)}{m - \frac{\beta}{ik}} \right) \overline{\left(\frac{2}{m - \frac{\beta}{ik}} \right)} + \left(\frac{2}{m - \frac{\beta}{ik}} \right) \overline{\left(\frac{\frac{\beta}{ik} - (m-2)}{m - \frac{\beta}{ik}} \right)} + \sum_{j=1}^{m-2} \left(\frac{2}{m - \frac{\beta}{ik}} \right) \overline{\left(\frac{2}{m - \frac{\beta}{ik}} \right)} \\
&= \frac{\frac{2\beta}{ik} - 2m + 4 - \frac{2\beta}{ik} - 2m + 4 + \sum_{j=1}^{m-2} 4}{m^2 + \frac{\beta^2}{k^2}} \\
&= \frac{-4m + 8 + 4(m-2)}{m^2 + \frac{\beta^2}{k^2}} \\
&= 0,
\end{aligned}$$

where β is the Kirchhoff condition. We also have

$$\begin{aligned}
\vec{x} \cdot \vec{x} &= \vec{x}^T \bar{\vec{x}} \\
&= \left(\frac{\frac{\beta}{ik} - (m-2)}{m - \frac{\beta}{ik}} \right) \overline{\left(\frac{\frac{\beta}{ik} - (m-2)}{m - \frac{\beta}{ik}} \right)} + \sum_{j=1}^{m-1} \left(\frac{2}{m - \frac{\beta}{ik}} \right) \overline{\left(\frac{2}{m - \frac{\beta}{ik}} \right)} \\
&= \frac{\frac{\beta^2}{k^2} + (m-2)^2 + \sum_{j=1}^{m-1} 4}{m^2 + \frac{\beta^2}{k^2}} \\
&= \frac{\frac{\beta^2}{k^2} + m^2 - 4m + 4 + 4(m-1)}{m^2 + \frac{\beta^2}{k^2}} \\
&= 1.
\end{aligned}$$

Thus all rows coming from the continuity and Kirchhoff conditions are orthonormal.

The last case to consider is the rows that we obtain from the boundary conditions. These rows yield a row with a single non-zero entry that is also the only non-zero entry in its respective column. Thus, its dot product with any other row will be zero. Now we find the dot product of such a row, z, with itself:

$$\begin{aligned}
\vec{z} \cdot \vec{z} &= \vec{z}^T \bar{\vec{z}} \\
&= \left(\frac{\alpha + ik}{-\alpha + ik} \right) \overline{\left(\frac{\alpha + ik}{-\alpha + ik} \right)} \\
&= \left(\frac{\alpha + ik}{-\alpha + ik} \right) \left(\frac{\alpha - ik}{-\alpha - ik} \right) \\
&= \left(\frac{\alpha + ik}{-\alpha + ik} \right) \left(\frac{(-1)(-\alpha + ik)}{(-1)(\alpha + ik)} \right) \\
&= 1,
\end{aligned}$$

where α is the boundary condition for the leaf node. Thus, all rows coming from the boundary conditions are orthonormal. □

With these facts established, we may now closely follow the proof in [15, §5.4] showing that the eigenvalues of the secular determinant are, in fact, real.

Theorem 2.1. The analytic function

$$\zeta(k) = \frac{1}{\sqrt{\det(\mathbf{S}(k)\mathbf{D}(k))}} \det(I - \mathbf{S}(k)\mathbf{D}(k))$$

is real-valued for $k \in \mathbb{R}$ and has the same zeros as the secular determinant $\Sigma(k)$.

Proof. For real values of k, we know that the matrix $U = \mathbf{S}(k)\mathbf{D}(k)$ is unitary because both $\mathbf{S}(k)$ and $\mathbf{D}(k)$ are unitary matricies. So, we can rewrite

$$\zeta(k) = (\det U)^{-\frac{1}{2}} \det(I - U).$$

Using the unitary nature of U, we know $UU^* = I$ and $\det U^* = (\det U)^{-1}$. We now evaluate

$$\begin{aligned}
\overline{\zeta(k)} =& (\det U)^{\frac{1}{2}} \det(I - U^*) \\
=& (\det U)^{\frac{1}{2}} \det(UU^* - U^*) \\
=& (\det U)^{\frac{1}{2}} \det(U - I) \det(U^*) \\
=& (\det U)^{-\frac{1}{2}} \det(I - U) = \zeta(k)
\end{aligned}$$

by making use of the identity

$$\det AB = \det A \det B.$$

Therefore, $\zeta(k)$ must be real for real values of k. □

2.2.1 Example Calculations

For all of the following examples, it is useful to recall that the solution on edge e_m is given by (2.10), and we will use explicit examples of (2.6), (2.7), and (2.8).

Star Graph We begin with the graph that has the most straightforward calculation, the star graph. Recall from Figure 2.1b that it features four vertices and three edges connecting to a center

vertex. The vertex conditions in this case are

$$\begin{cases} \psi_1(0) = \psi_2(0) = \psi_3(0) & \text{Continuity condition} \\ \psi_1'(0) + \psi_2'(0) + \psi_3'(0) + \beta\psi_1(0) = 0 & \text{Kirchhoff condition} \\ \alpha_j\psi_j(l_j) + \psi_j'(l_j) = 0 & \text{Boundary condition.} \end{cases}$$

The b_j terms are easily found by substituting (2.10) into the Robin boundary conditions to get

$$\alpha_j(a_j e^{ikl_j} + b_j) + a_j ike^{ikl_j} - b_j ik = 0. \tag{2.22}$$

This is solved for b_j, which simplifies to

$$b_j = \frac{ik + \alpha_j}{ik - \alpha_j} a_j e^{ikl_j}. \tag{2.23}$$

Now, we must solve for the a_j terms; this is more involved, but still straightforward.

First, (2.10) is substituted into the continuity condition to find

$$a_1 + b_1 e^{ikl_1} = a_2 + b_2 e^{ikl_2} = a_3 + b_3^{ikl_3}. \tag{2.24}$$

When this is substituted into the Kirchhoff condition we find that

$$a_1 - b_1 e^{ikl_1} + a_2 - b_2 e^{ikl_2} + a_3 - b_3 e^{ikl_3} + \frac{\beta}{ik}\left(a_1 + b_1 e^{ikl_1}\right) = 0. \tag{2.25}$$

Rearranging (2.25), we get

$$\left(1 + \frac{\beta}{ik}\right) a_1 = \left(1 - \frac{\beta}{ik}\right) b_1 e^{ikl_1} - a_2 + b_2 e^{ikl_2} - a_3 + b_3 e^{ikl_3}, \tag{2.26}$$

and (2.24) can be used to get

$$a_2 = a_1 + b_1 e^{ikl_1} - b_2 e^{ikl_2} \tag{2.27}$$

$$a_3 = a_1 + b_1 e^{ikl_1} - b_3 e^{ikl_3}. \tag{2.28}$$

Substituting (2.27) and (2.28) into (2.26), a_1 can be found in terms of b_j:

$$\begin{aligned} a_1 &= \frac{1}{3+\frac{\beta}{ik}}\left(\left(-\frac{\beta}{ik}-1\right)b_1e^{ikl_1}+2b_2e^{ikl_2}+2b_3e^{ikl_3}\right) \\ &= \frac{-\beta-ik}{3ik+\beta}b_1e^{ikl_1}+\frac{2ik}{3ik+\beta}b_2e^{ikl_2}+\frac{2ik}{3ik+\beta}b_3e^{ikl_3}. \end{aligned} \tag{2.29}$$

A similar process can be used to find a_2 and a_3.

The system of equations generated by the a_j and b_j solutions can be turned into the following matrix system:

$$\begin{pmatrix} 0 & 0 & 0 & \frac{-\beta-ik}{3ik+\beta} & \frac{2ik}{3ik+\beta} & \frac{2ik}{3ik+\beta} \\ 0 & 0 & 0 & \frac{2ik}{3ik+\beta} & \frac{-\beta-ik}{3ik+\beta} & \frac{2ik}{3ik+\beta} \\ 0 & 0 & 0 & \frac{2ik}{3ik+\beta} & \frac{2ik}{3ik+\beta} & \frac{-\beta-ik}{3ik+\beta} \\ \frac{ik+\alpha_1}{ik-\alpha_1} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{ik+\alpha_2}{ik-\alpha_2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{ik+\alpha_3}{ik-\alpha_3} & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} e^{ikl_1} & 0 & 0 & 0 & 0 & 0 \\ 0 & e^{ikl_2} & 0 & 0 & 0 & 0 \\ 0 & 0 & e^{ikl_3} & 0 & 0 & 0 \\ 0 & 0 & 0 & e^{ikl_1} & 0 & 0 \\ 0 & 0 & 0 & 0 & e^{ikl_2} & 0 \\ 0 & 0 & 0 & 0 & 0 & e^{ikl_3} \end{pmatrix} \mathbf{a} = \mathbf{a}$$

where $\mathbf{a} = (a_1\ a_2\ a_3\ b_1\ b_2\ b_3)^T$. This matches the framework laid out for the secular determinant very nicely. Note that the involved piece of information is $\mathbf{S}(k)$, which is

$$\mathbf{S}(k) = \begin{pmatrix} 0 & 0 & 0 & \frac{-\beta-ik}{3ik+\beta} & \frac{2ik}{3ik+\beta} & \frac{2ik}{3ik+\beta} \\ 0 & 0 & 0 & \frac{2ik}{3ik+\beta} & \frac{-\beta-ik}{3ik+\beta} & \frac{2ik}{3ik+\beta} \\ 0 & 0 & 0 & \frac{2ik}{3ik+\beta} & \frac{2ik}{3ik+\beta} & \frac{-\beta-ik}{3ik+\beta} \\ \frac{ik+\alpha_1}{ik-\alpha_1} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{ik+\alpha_2}{ik-\alpha_2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{ik+\alpha_3}{ik-\alpha_3} & 0 & 0 & 0 \end{pmatrix}.$$

With this explicit example calculated, one can return to the definition of $\mathbf{S}$ in (2.18) to better understand its general construction.

For this example, we will also do the calculation for determining the phase angle, which begins with the determinant of $\mathbf{S}$:

$$\det(\mathbf{S}) = \frac{(\alpha_1 + ik)(\alpha_2 + ik)(\alpha_3 + ik)}{(\alpha_1 - ik)(\alpha_2 - ik)(\alpha_3 - ik)}.$$

Define $z_j = \alpha_j + ik = r_j e^{i\theta_j}$, where $r, \theta_j \in \mathbb{R}$. Then,

$$\theta_j = \arg(z_j).$$

We do the following calculation:

$$\det(\mathbf{S}) = \prod_{j=1}^{3} \frac{z_j}{\overline{z_j}} = \prod_{j=1}^{3} \frac{r_j e^{i\theta_j}}{r_j e^{-i\theta_j}} = \prod_{j=1}^{3} e^{2i\theta_j} = e^{2i\sum_{j=1}^{3}\theta_j} = e^{2i\Theta_R},$$

where $\Theta_R = \sum_{j=1}^{3} \arg(z_j)$.

Lollipop Graph Next, we cover the lollipop graph from Figure 2.1a, which features two vertices and two edges. In this case, the standard matching conditions are

$$\begin{cases} \psi_1(0) = \psi_1(l_1) = \psi_2(0) & \text{Continuity condition} \\ \psi_1'(0) - \psi_1'(l_1) + \psi_2'(0) + \beta\psi_1(0) = 0 & \text{Kirchhoff condition} \\ \alpha\psi_2(l_2) + \psi_2'(l_2) = 0 & \text{Boundary condition} \end{cases}$$

By substituting (2.10) into the continuity condition, we get

$$a_1 + b_1 e^{ik\ell_1} = a_1 e^{ik\ell_1} + b_1 = a_2 + b_2 e^{ik\ell_2}. \tag{2.30}$$

We also substitute (2.10) into the Kirchhoff condition to get

$$a_1 - b_1 e^{ik\ell_1} - a_1 e^{ik\ell_1} + b_1 + a_2 - b_2 e^{ik\ell_2} + \frac{\alpha_1}{ik}\left(a_1 + b_1 e^{ikl_1}\right) = 0,$$

which we rearrange into

$$\left(1+\frac{\alpha_1}{ik}\right)a_1 = \left(1-\frac{\alpha_1}{ik}\right)b_1e^{ik\ell_1} + a_1e^{ik\ell_1} - b_1 - a_2 + b_2e^{ikl_2}. \tag{2.31}$$

We then make use of (2.30) to get

$$b_1 = a_1 + b_1e^{ik\ell_1} - a_1e^{ik\ell_1}$$
$$a_2 = a_1 + b_1e^{ik\ell_1} - b_2e^{ik\ell_2}.$$

These are substituted into (2.31) which, when solved for a_1, yield

$$a_1 = \frac{2ik}{3ik+\alpha_1}a_1e^{ik\ell_1} - \frac{ik+\alpha_1}{3ik+\alpha_1}b_1e^{ik\ell_1} + \frac{2ik}{3ik+\alpha_1}b_2e^{ikl_2}. \tag{2.32}$$

A similar process can be used to find equations for a_2 and b_1.

To solve for b_2, one need only substitute (2.10) into the Robin condition

$$a_2e^{ik\ell_2} - b_2 + \frac{\alpha_2}{ik}(a_2e^{ik\ell_2} + b_2) = 0,$$

and rearrange to get

$$b_2 = \frac{ik+\alpha_2}{ik-\alpha_2}a_2e^{ik\ell_2}. \tag{2.33}$$

Our solutions for a_1, a_2, b_1 and b_2 can be summarized with the following system:

$$\begin{pmatrix} \frac{2ik}{3ik+\alpha_1} & 0 & \frac{-ik-\alpha_1}{3ik+\alpha_1} & \frac{2ik}{3ik+\alpha_1} \\ \frac{2ik}{3ik+\alpha_1} & 0 & \frac{2ik}{3ik+\alpha_1} & \frac{-ik-\alpha_1}{3ik+\alpha_1} \\ \frac{-ik-\alpha_1}{3ik+\alpha_1} & 0 & \frac{2ik}{3ik+\alpha_1} & \frac{2ik}{3ik+\alpha_1} \\ 0 & \frac{ik+\alpha_2}{ik-\alpha_2} & 0 & 0 \end{pmatrix} \begin{pmatrix} e^{ikl_1} & 0 & 0 & 0 \\ 0 & e^{ikl_2} & 0 & 0 \\ 0 & 0 & e^{ikl_1} & 0 \\ 0 & 0 & 0 & e^{ikl_2} \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} a_1 \\ a_2 \\ b_1 \\ b_2 \end{pmatrix},$$

where we recall that this implies that

$$\mathbf{S}(k) = \begin{pmatrix} \frac{2ik}{3ik+\alpha_1} & 0 & \frac{-ik-\alpha_1}{3ik+\alpha_1} & \frac{2ik}{3ik+\alpha_1} \\ \frac{2ik}{3ik+\alpha_1} & 0 & \frac{2ik}{3ik+\alpha_1} & \frac{-ik-\alpha_1}{3ik+\alpha_1} \\ \frac{-ik-\alpha_1}{3ik+\alpha_1} & 0 & \frac{2ik}{3ik+\alpha_1} & \frac{2ik}{3ik+\alpha_1} \\ 0 & \frac{ik+\alpha_2}{ik-\alpha_2} & 0 & 0 \end{pmatrix}.$$

Dumbbell Graph Recall the dumbbell graph from Figure 2.1c, which features two vertices and three edges - one bridging between the vertices and the other two edges forming loops. Notice this has no leaf vertices and thus no Robin boundary conditions. Applying standard matching conditions produces the following system of equations:

$$\begin{cases} \psi_1(0) = \psi_1(l_1) = \psi_2(0) & \text{Continuity condition} \\ \psi_2(l_2) = \psi_3(0) = \psi_3(l_3) & \\ \psi_1'(0) - \psi_1'(l_1) + \psi_2'(0) + \beta_1\psi_1(0) = 0 & \text{Kirchhoff condition} \\ \psi_3'(0) - \psi_3'(l_3) - \psi_2'(l_2) + \beta_2\psi_3(0) = 0 & \end{cases}$$

Plugging the solution into the current conditions, we get the following:

$$a_1 + b_1 e^{ikl_1} = a_1 e^{ikl_1} + b_1 = a_2 + b_2 e^{ikl_2} \tag{2.34}$$

$$a_2 e^{ikl_2} + b_2 = a_3 + b_3 e^{ikl_3} = a_3 e^{ikl_3} + b_3. \tag{2.35}$$

Similarly, we substitute this into the Kirchhoff condition to get

$$a_1 - b_1 e^{ikl_1} - (a_1 e^{ikl_1} - b_1) + a_2 - b_2 e^{ikl_2} + \frac{\beta_1}{ik}(a_1 + b_1 e^{ikl_1}) = 0 \tag{2.36}$$

$$a_3 - b_3 e^{ikl_3} - (a_3 e^{ikl_3} - b_3) - (a_2 e^{ikl_2} - b_2) + \frac{\beta_2}{ik}(a_3 + b_3 e^{ikl_3}) = 0. \tag{2.37}$$

Rearranging (2.36), we get

$$\left(1+\frac{\beta_1}{ik}\right)a_1 = a_1e^{ikl_1} + b_1e^{ikl_1} - b_1 - a_2 + b_2e^{ikl_2} - \frac{\beta_1}{ik}b_1e^{ikl_1}, \tag{2.38}$$

and we can make use of (2.34) to find

$$b_1 = a_1 + b_1e^{ikl_1} - a_1e^{ikl_2} \tag{2.39}$$

$$a_2 = a_1 + b_1e^{ikl_1} - b_2e^{ikl_3}. \tag{2.40}$$

Substituting (2.39) and (2.40), into (2.38) we can find a_1 in terms of a_1, b_1 and b_2:

$$a_1 = \frac{2}{3+\frac{\beta_1}{ik}}a_1e^{ikl_1} - \frac{-\frac{\beta_1}{ik}+1}{3+\frac{\beta_1}{ik}}b_1e^{ikl_1} + \frac{2}{3+\frac{\beta_1}{ik}}b_2e^{ikl_2}. \tag{2.41}$$

This is what will become the first row. Now, solve (2.39) and (2.40) for a_1 and substitute them into (2.41) to get equations for b_1 and a_2:

$$\begin{aligned} b_1 &= \frac{-\frac{\beta_1}{ik}-1}{3+\frac{\beta_1}{ik}}a_1e^{ikl_1} + \frac{2}{3+\frac{\beta_1}{ik}}b_1e^{ikl_1} + \frac{2}{3+\frac{\beta_1}{ik}}b_2e^{ikl_2} \\ &= \frac{-\beta_1-ik}{3ik+\beta_1}a_1e^{ikl_1} + \frac{2ik}{3ik+\beta_1}b_1e^{ikl_1} + \frac{2ik}{3ik+\beta_1}b_2e^{ikl_2}, \end{aligned}$$

$$\begin{aligned} a_2 &= \frac{2}{3+\frac{\beta_1}{ik}}a_1e^{ikl_1} + \frac{2}{3+\frac{\beta_1}{ik}}b_1e^{ikl_1} + \frac{\frac{\beta_1}{ik}-1}{3+\frac{\beta_1}{ik}}b_2e^{ikl_2} \\ &= \frac{2ik}{3ik+\beta_1}a_1e^{ikl_1} + \frac{2ik}{3ik+\beta_1}b_1e^{ikl_1} + \frac{\beta_1-ik}{3ik+\beta_1}b_2e^{ikl_2}. \end{aligned}$$

Similar manipulations can be done regarding Equations 2.35 and 2.37 to solve for b_2, a_3, and b_3. The resulting system of equations has the following $\mathbf{S}(k)$ which, as we noted in the previous two examples, is particularly interesting as it can be used to better understand (2.18):

$$\mathbf{S}(k) = \begin{pmatrix} \frac{2ik}{3ik+\beta_1} & 0 & 0 & \frac{-\beta_1-ik}{3ik+\beta_1} & \frac{2ik}{3ik+\beta_1} & 0 \\ \frac{2}{3+\frac{\beta_1}{ik}} & 0 & 0 & \frac{2}{3+\frac{\beta_1}{ik}} & \frac{-\beta_1-ik}{3ik+\beta_1} & 0 \\ 0 & \frac{2ik}{3ik+\beta_2} & \frac{2ik}{3ik+\beta_2} & 0 & 0 & \frac{-\beta_2-ik}{3ik+\beta_2} \\ \frac{-\beta_1-ik}{3ik+\beta_1} & 0 & 0 & \frac{2}{3+\frac{\beta_1}{ik}} & \frac{2ik}{3ik+\beta_1} & 0 \\ 0 & \frac{-\beta_2-ik}{3ik+\beta_2} & \frac{2ik}{3ik+\beta_2} & 0 & 0 & \frac{2ik}{3ik+\beta_2} \\ 0 & \frac{2ik}{3ik+\beta_2} & \frac{-\beta_2-ik}{3ik+\beta_2} & 0 & 0 & \frac{2ik}{3ik+\beta_2} \end{pmatrix}.$$

2.2.2 Symbolic Calculation

While eigenvalues can now be computed for any graph, it is still useful to automate the finding of the secular determinant. As mentioned earlier, it is known that the eigenvalues will be real. However, a few tricks beyond the removal of the phase angle will need to be employed numerically to ensure this.

When the secular determinant is calculated symbolically, the phase angle is removed then converted to sines and cosines. This results in a transcendental function that may still have imaginary coefficients even though the eigenvalues are, in fact, real. There are two ways for the transcendental function to produce real solutions; all coefficients are either real or purely imaginary (up to a constant). The transcendental function may be a fraction, so it is divided by the ratio of the leading coefficients. (Dividing by any constant can be done safely since the secular determinant equals zero.) This ensures that the new coefficients will now either be real or purely imaginary. The last trick is to look for imaginary coefficients. If there are none, the resulting transcendental function is purely real, and it is the secular determinant. But, if there is an imaginary coefficient, then the imaginary portion of the transcendental function is taken to be the secular determinant, as the real portion is zero.

CHAPTER 3

Accurately Defining Numerical Spatial Operators

QGLAB is a MATLAB package that has been developed jointly with Roy Goodman as the focus of this dissertation work. While it is more thoroughly documented in [40], we use this chapter to highlight the numerical methods that we built to solve linear and nonlinear stationary problems on quantum graphs using the Chebyshev discretization. This begins with defining the Laplacian in a way that incorporates boundary conditions and utilizes techniques proposed in [28] to achieve spectral accuracy. We then generalize this result to work in the context of quantum graphs and show necessary details that allow QGLAB to automatically generate the discretized Laplacian operator for any given graph. This is followed by an application to the field of spectral flow where the behavior of the Laplacian's limiting eigenfunctions (as described in [13]) is verified through the use of QGLAB. The final section overviews the tools that are implemented in QGLAB to generate numerical solutions to nonlinear, elliptic partial differential equations.

3.1 Discretizing the Laplacian

Now that the Laplacian operator has been investigated analytically, the first thing we will do numerically is accurately discretize it. Each edge is discretized by N internal points; including the end points at the vertices totals to $N+2$ points. For now, the focus will be on accurately defining things on a closed interval $[0, \ell]$, and the results are then generalized for graphs later.

For most purposes, uniformly spaced points enable us to use the finite difference scheme which is easy to understand and runs quite quickly due to the sparse nature of the operator matrices. QGLAB implements the finite difference method using standard second order centered differences with the boundary conditions enforced at so-called ghost points as discussed in [29]; more details about its implementation in QGLAB can be found in [40].

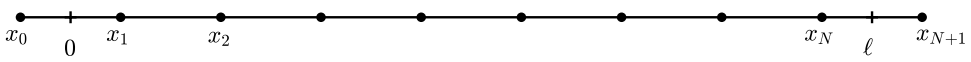


Figure 3.1: Discretization of the interval $[0, \ell]$ using ghost points

At the expense of sparsity, accuracy can be improved upon through the use of another discretization, namely **Chebyshev points of the second kind** (see Figure 3.2),

$$x_k = \frac{\ell}{2}\left(1 - \cos\left(\frac{k\pi}{N+1}\right)\right), \qquad k = 0, 1, \ldots, N, N+1. \tag{3.1}$$

Notice how these points are clustered at the end points in Figure 3.2. While the QGLAB package has been implemented in a way that allows the user to switch easily between uniformly spaced discretization points and Chebyshev points, this dissertation focuses on implementations from the perspective of Chebyshev points.

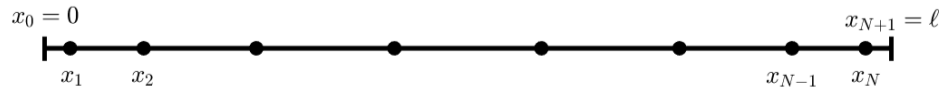


Figure 3.2: Discretization of the interval $[0, \ell]$ using Chebyshev points of the second kind.

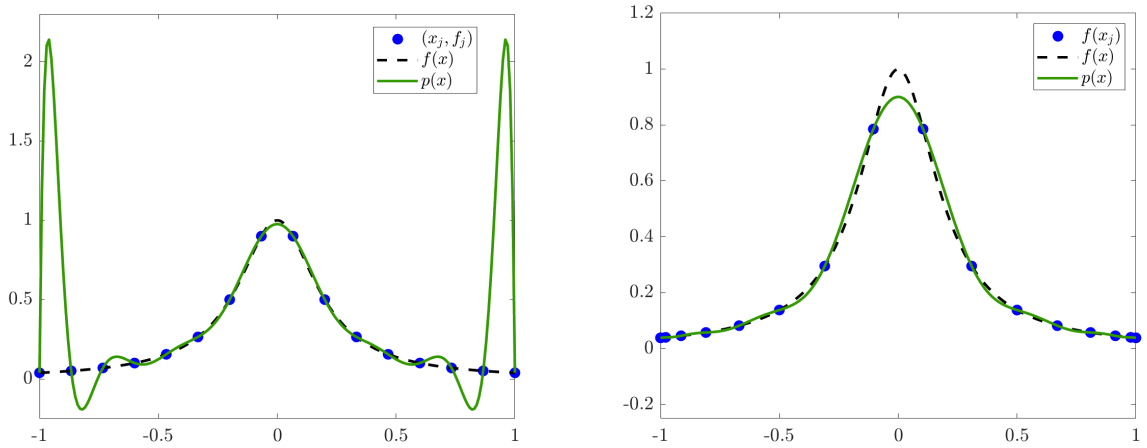
We will focused on Chebyshev points due to their high accuracy at relatively low numbers of discretization points. As an illustration of how Chebyshev points are useful, we will consider their ability to prevent the **Runge phenomenon** when performing polynomial interpolation. A polynomial can be used to interpolate the data produced by evaluating $f(x)$ at $\mathbf{x} = \{x_j\}_{j=0}^{N+1}$ via the formulation

$$f(x) \approx p(x) = \sum_{j=0}^{N+1} f_j l_j(x), \tag{3.2}$$

where $f_j = f(x_j)$ in this context and the $l_j(x)$ terms denote the Lagrange interpolating weights. Uniformly spaced points are known to produce less accurate approximations at the boundaries. However, Chebyshev points produce spectrally accurate results at the boundary points for reasonably smooth solutions. The contrast of the polynomial interpolants generated on these two different discretizations is illustrated in Figure 3.3.

The proposed interpolating polynomial in (3.2) provides the means to define the matrix representation of the first derivative matrix, since the derivative of $f(x)$ can be approximated by $p'(x)$:

$$f'(x) \approx p'(x) = \sum_{j=0}^{N+1} f_j l_j'(x) \tag{3.3}$$



(a) Uniform points

(b) Chebyshev points of the 2nd kind

Figure 3.3: Runge phenomenon

The matrix generated by evaluating $l'_j(x)$ at $\mathbf{x}$ is the differentiation matrix defined on (3.1). Thus, an approximation of $f'(x)$ at $\mathbf{x}$ can be given by

$$\mathbf{f}_{prime} \approx \mathbf{D}_{(N+2)\times(N+2)}\mathbf{f}, \tag{3.4}$$

where

$$D_{ij} = l'_j(x_i), \tag{3.5}$$

and $\mathbf{f}$ is the vector whose values are f_j. More explicit details are provided in [77]. To obtain the Laplacian, one need only square $\mathbf{D}$ to find a fairly accurate approximation.

3.1.1 Incorporating Boundary Conditions

Closed Interval The main goal is to find an accurate way to numerically formulate

$$\begin{cases} \dfrac{d^2u}{dx^2} = f(x), & 0 < x < \ell & \text{(3.6a)} \\ u'(0) + \alpha_0 u(0) = \phi_0, & \text{(3.6b)} \\ -u'(\ell) + \alpha_\ell u(\ell) = \phi_\ell. & \text{(3.6c)} \end{cases}$$

To begin discretizing this problem, (3.6a) is refomulated as

$$\mathbf{D}^2\mathbf{u} = \mathbf{f}, \tag{3.7}$$

where

$$\mathbf{u} = \begin{pmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N+1} \end{pmatrix} \quad \text{and} \quad \mathbf{f} = \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_{N+1} \end{pmatrix}. \tag{3.8}$$

But, this is a well-defined matrix system that does not account for the boundary conditions which we know will be crucial when we discuss graphs.

There are a few ways to incorporate the boundary conditions into a differentiation matrix. The most straightforward method is row replacement; it involves removing two rows (typically the first and last) of the differentiation matrix and replacing them with rows that enforce the boundary conditions. However, this is generally used in the case of uniformly spaced points, where row replacement has linear convergence (quadratic convergence is possible if done carefully as demonstrated in [50, 18]). As discussed in [77], this translates poorly to operators defined by Chebyshev points and does especially poorly for higher-order problems where it might be necessary to replace more than just two rows to properly implement the boundary conditions.

In [28], the authors developed an alternative method for incorporating boundary conditions in a differentiation matrix while preserving high spatial accuracy built around the use of Chebyshev points. The motivating observation is that we would like to work with N internal points instead of the given $N+2$ points that include the boundaries, but simply deleting the end points is no better than row replacement. Instead, the goal is to work on **Chebyshev points of the first kind**

$$\chi_k = \frac{\ell}{2}\left(1 - \cos\left(\frac{(2k-1)\pi}{2N}\right)\right) \qquad k = 1, 2, \ldots, N. \tag{3.9}$$

which are not defined at the end points, providing us with N many alternative interior points to work on (Figure 3.4).

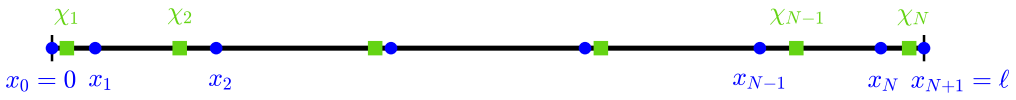


Figure 3.4: Discretization of the interval $[0, \ell]$ using Chebyshev points of the second kind (in blue) and first kind (in green)

The method relies on the barycentric resampling matrix, which provides an efficient and stable means to take a vector of values representing an interpolating polynomial defined on Chebyshev points of the second kind and resample this polynomial at Chebyshev points of the first kind. Its construction utilizes the barycentric interpolation formula proposed in [19].

Given the set of points $\mathbf{x} = \{x_k\}_{k=0}^{N+1}$, the barycentric weights are

$$w_k = \prod_{\substack{l=0 \\ l \neq k}}^{N+1} (x_k + x_l)^{-1}, \qquad k = 0, \ldots, N+1. \tag{3.10}$$

These are utilized to construct a unique interpolating polynomial,

$$p_{N+1}(x) = \frac{\sum_{k=0}^{N+1} (w_k/(x - x_k)) f_k}{\sum_{l=0}^{N+1} (w_l/(x - x_l))}, \tag{3.11}$$

which interpolates the set of data points $\{(x_k, f_k)\}_{k=0}^{N+1}$. The polynomial is evaluated at $\{x_k\}_{k=0}^{N+1}$ and again at $\{\chi_k\}_{k=1}^{N}$, so that the **barycentric resampling matrix** defined by

$$(\mathbf{P}_{\text{int}})_{j,k} = \begin{cases} \dfrac{w_k}{\chi_j - x_k} \left(\displaystyle\sum_{l=0}^{N+1} \frac{w_l}{\chi_j - x_l} \right)^{-1} & \chi_j \neq x_k \\ 1 & \chi_j = x_k. \end{cases} \tag{3.12}$$

satisfies

$$p_{N+1}(\boldsymbol{\chi}) = \mathbf{P}_{\text{int}}\ p_{N+1}(\mathbf{x}). \tag{3.13}$$

(A more generalized version of this matrix in [28, §3.1].)

This matrix is denoted as $\mathbf{P}_{\text{int}}$ since it will project the information to purely the interior points, and it has the dimensions $N \times (N+2)$. The product

$$\mathbf{L}_{\text{int}} = \mathbf{P}_{\text{int}} \mathbf{D}^2 \tag{3.14}$$

defines an $N \times (N+2)$ differentiation matrix, so applying $\mathbf{P}_{\text{int}}$ to both sides of (3.7) leaves two rows free for the implementation of boundary conditions.

The standard basis vectors, $\mathbf{e}_k$, are utilized in the implementation of the boundary conditions to pick out the correct location in a vector. For instance, it is very helpful to know $u(0) = u_0 = \mathbf{e}_1^T \mathbf{u}$,

so we can write (3.6b) as

$$\left(\mathbf{e}_1^T\mathbf{D} + \alpha_0\mathbf{e}_1^T\right)\mathbf{u} = \phi_0.$$

Similarly, we write (3.6c) as

$$\left(-\mathbf{e}_{N+2}^T\mathbf{D} + \alpha_L\mathbf{e}_{N+2}^T\right)\mathbf{u} = \phi_L.$$

The boundary conditions can be summarized in matrix form by

$$\mathbf{L}_{\text{BC}}\mathbf{u} = \boldsymbol{\phi}, \tag{3.15}$$

where

$$\mathbf{L}_{\text{BC}} = \begin{pmatrix} \mathbf{e}_1^T\mathbf{D} + \alpha_0\mathbf{e}_1^T \\ -\mathbf{e}_{N+2}^T\mathbf{D} + \alpha_L\mathbf{e}_{N+2}^T \end{pmatrix}_{2\times(N+2)} \quad \text{and} \quad \boldsymbol{\phi} = \begin{pmatrix} \phi_0 \\ \phi_L \end{pmatrix}. \tag{3.16}$$

The matrices $\mathbf{L}_{\text{int}}, \mathbf{L}_{\text{BC}}$ and $\mathbf{P}_{\text{int}}$ are used to create a single system of $N+2$ equations and $N+2$ unknowns

$$\begin{pmatrix} \mathbf{L}_{\text{int}} \\ \mathbf{L}_{\text{BC}} \end{pmatrix}\mathbf{u} = \begin{pmatrix} \mathbf{P}_{\text{int}} \\ \mathbf{0}_{2\times(N+2)} \end{pmatrix}\mathbf{f} + \begin{pmatrix} \mathbf{0}_{N\times 2} \\ \mathbf{I}_2 \end{pmatrix}\boldsymbol{\phi} \tag{3.17}$$

which is the full discretized version of (3.6).

General Graph The process generalizes to graphs fairly naturally. Suppose that we want to discretize the problem

$$\begin{cases} \dfrac{d^2u}{dx^2} = f(x), \qquad x \in e_j,\ \forall e_j \in \mathcal{E}, & \text{(3.18a)} \\ u_i(v_n) = u_j(v_n),\ \forall e_i, \qquad e_j \in \mathcal{V}_n, & \text{(3.18b)} \\ \displaystyle\sum_{e_m\in\mathcal{V}_n} w_m u_m'(v_n) + \alpha_n u_{e_{N+2}\in\mathcal{V}_n}(v_n) = \phi_n, \qquad \forall v_n \in \mathcal{V} & \text{(3.18c)} \end{cases}$$

on an arbitrary directed, metric graph where u_m is the solution on the m^{th} edge. We begin by discretizing each edge e_m with N_m Chebyshev points of the second kind that we will denote as $\mathbf{x}$. The notation $u_{mk} = u_m(x_k)$ is introduced to refer to the solution on edge m evaluated at the k^{th}

discretization point, so that $\mathbf{u}_m$ would be the discretized solution on edge m. The full discretized solution can be made using the $\mathbf{u}_m$'s, and a similar process is done to discretize $f(x)$ so that

$$\mathbf{u} = \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_{|\mathcal{E}|} \end{pmatrix}, \mathbf{f} = \begin{pmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \vdots \\ \mathbf{f}_{|\mathcal{E}|} \end{pmatrix}, \quad \text{and} \quad \boldsymbol{\phi} = \begin{pmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{|\mathcal{E}|} \end{pmatrix},$$

where $\boldsymbol{\phi}$ is the vector of nonhomogeneous boundary conditions.

Defining the Laplacian and barycentric resampling matrix on a graph are done by defining them on each edge using the same construction methods employed for the line. $\mathbf{L}_{\text{int}}^{(m)}$ is defined as the Laplacian on e_m, and $\mathbf{P}_{\text{int}}^{(m)}$ is the corresponding barycentric resampling matrix. Note that these matrices are of the sizes $(N_m + 2) \times N_m$, so that the matrices

$$\mathbf{L}_{\text{int}} = \begin{pmatrix} \mathbf{L}_{\text{int}}^{(1)} & & & \\ & \mathbf{L}_{\text{int}}^{(2)} & & \\ & & \ddots & \\ & & & \mathbf{L}_{\text{int}}^{(|\mathcal{E}|)} \end{pmatrix} \quad \text{and} \quad \mathbf{P}_{\text{int}} = \begin{pmatrix} \mathbf{P}_{\text{int}}^{(1)} & & & \\ & \mathbf{P}_{\text{int}}^{(2)} & & \\ & & \ddots & \\ & & & \mathbf{P}_{\text{int}}^{(|\mathcal{E}|)} \end{pmatrix} \tag{3.19}$$

are of dimension$(N_{\text{tot}} + 2|\mathcal{E}|) \times N_{\text{tot}}$ where $N_{\text{tot}} = \sum_{m=1}^{|\mathcal{E}|} N_m$.

Enforcing conditions at each vertex v_n requires one row for the Kirchhoff-Robin condition (3.18c) and $d_{v_n} - 1$ rows for the continuity condition (3.18b). Altogether, these form a matrix $\mathbf{L}_{\text{VC}}^{(n)}$ with d_{v_n} rows so that $\mathbf{L}_{\text{VC}}$ has $|\mathcal{E}|$ many rows. Thus, the Laplacian with encoded boundary conditions and the barycentric resampling matrix for the graph are given by

$$\mathbf{L} = \begin{pmatrix} \mathbf{L}_{\text{int}} \\ \hdashline \mathbf{L}_{\text{VC}} \end{pmatrix} = \left(\begin{array}{cccc} \mathbf{L}_{\text{int}}^{(1)} & & & \\ & \ddots & & \\ & & \mathbf{L}_{\text{int}}^{(|\mathcal{E}|)} & \\ \hdashline & & \mathbf{L}_{\text{VC}}^{(1)} & \\ & & \vdots & \\ & & \mathbf{L}_{\text{VC}}^{(|\mathcal{V}|)} & \end{array}\right) \quad \text{and} \quad \mathbf{P} = \left(\begin{array}{cccc} \mathbf{P}_{\text{int}}^{(1)} & & & \\ & \ddots & & \\ & & \mathbf{P}_{\text{int}}^{(|\mathcal{E}|)} & \\ \hdashline & & \mathbf{0}_{2|\mathcal{E}| \times (N_{\text{tot}} + 2|\mathcal{E}|)} & \end{array}\right). \tag{3.20}$$

Next, we define a **vertex condition assignment matrix** $\mathbf{M}_{\text{VCA}}$ of size $(N_{\text{tot}} + 2|\mathcal{E}|) \times |\mathcal{V}|$ whose purpose is to map the correct nonhomogeneous condition to the correct vertex condition. This means that $\mathbf{M}_{\text{VCA}}$ consists entirely of zeros except in column n which has a one in the first row corresponding to the position of the first row of $\mathbf{L}_{\text{VC}}^{(n)}$ in $\mathbf{L}$. Thus, the discrete problem can be represented in the compact form

$$\mathbf{Lu} = \mathbf{Pf} + \mathbf{M}_{\text{VCA}}\boldsymbol{\phi}. \tag{3.21}$$

Example To better visualize the construction of the Laplacian, consider the Poisson problem defined on the lollipop graph, which has the following formulation

$$\begin{cases} \dfrac{d^2u}{dx^2} = f(x), \qquad x \in \mathcal{E}_j,\ \forall e_j \in \mathcal{E}, \\ u_1(\ell_1) = u_2(0) = u_2(\ell_2), \\ u_1'(\ell_1) - u_2'(0) = u_2(\ell_2). \end{cases} \tag{3.22}$$

For this example, the edge e_1 points from v_1 to v_2 and has $N_1 = 4$, while edge e_2 points from v_2 to itself and has $N_2 = 8$ discretization points. The lollipop graph discretized with Chebyshev points of the second kind is shown in Figure 3.5. The figure also shows the structure of the nonzero entries of the matrix $\mathbf{L}$. While the construction of $\mathbf{L}_{\text{int}}$ is fairly understandable without an example, a sample calculation for $\mathbf{L}_{\text{VC}}$ is provided by considering the boundary conditions at v_2.

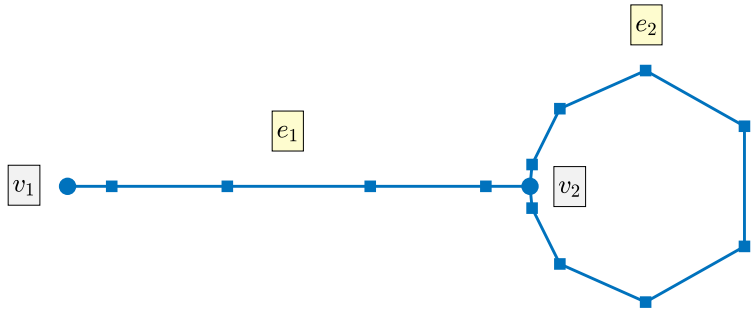


Figure 3.5: Lollipop with labeled discretized points

The continuity conditions

$$u_1(\ell_1) = u_2(0),$$
$$u_1(\ell_1) = u_2(\ell_2)$$

are easier to digest and thus done first. To write the discretized version, notice that while edge one is discretized with 4 internal points, a total of 6 points are needed for the entire edge when accounting for the end points. Thus, $u_1(\ell_1) = \mathbf{u}_6$. With this in mind, we write the discretized version of the continuity condition

$$(\mathbf{e}_6 - \mathbf{e}_7)\mathbf{u} = 0,$$
$$(\mathbf{e}_6 - \mathbf{e}_{16})\mathbf{u} = 0.$$

These correspond to the last two green lines of the Laplacian depicted in Figure 3.6.

Now, consider the Robin condition

$$w_1 u_1'(\ell_1) - w_2 u_2'(0) + w_2 u_2'(\ell_2) + \alpha_2 u_2(0) = 0,$$

which is discretized as

$$(w_1 \mathbf{e}_6 \mathbf{D}_1 - w_2 \mathbf{e}_7 \mathbf{D}_2 + w_2 \mathbf{e}_{16} \mathbf{D}_2 + \alpha_2 \mathbf{e}_7)\mathbf{u} = 0$$

and is represented in the first green line of the Laplacian from Figure 3.6. Figure 3.6 also shows the structure of the nonzero entries of $\mathbf{P}$, and we can note that the matrix $\mathbf{M}_{\text{VCA}}$ is 16×2 and its only nonzero entries are located at $(13, 1)$ and $(14, 2)$.

3.1.2 Numerical Implementation

While squaring $\mathbf{D}$ is a common way to approximate the Laplacian matrix, this introduces various numerical errors for operators implemented with a Chebyshev discretization. This section provides implementation details necessary for accurately and efficiently defining the rectangular differentiation matrices. The definition that is used for higher order differentiation matrices is given by

$$\mathbf{D}_{ij}^{(p+1)*} = \frac{1}{\chi_i - x_j}\left[\frac{(-1)^j}{2(N+1)}\left(T_{N+1}^{(p+1)}(\chi_i) - T_{N-1}^{(p+1)}(\chi_i)\right) - (p+1)\mathbf{D}_{ij}^{(p)},\right] \tag{3.23}$$

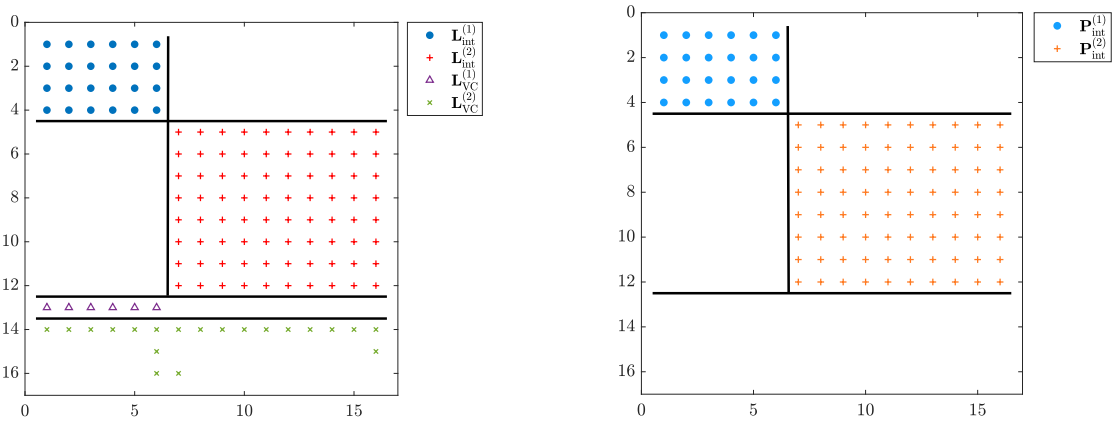


Figure 3.6: The left and right figures show the nonzero entries in $\mathbf{L}$ and $\mathbf{P}$ matrices respectively.

where the star indicates that the first term inside the square brackets is halved for $j = 0$ and $j = N + 1$, (p) denotes the p^{th} spatial derivative, $T_n = \cos(n \arccos x)$, and $\mathbf{D}^{(p+1)}$ is a $N \times (N + 2)$ rectangular matrix. If χ_i and x_i are the same, the corresponding entry becomes

$$\mathbf{D}_{ij}^{(p+1)} = (-1)^j \frac{T_{N+1}^{(p+1)}(\chi_i) - T_{N-1}^{(p+1)}(\chi_i)}{2(N+1)(p+2)}. \tag{3.24}$$

This formulation is verified in [77, §4].

Below are a few additional tricks that also reduce numerical errors and enhance the efficiency when creating rectangular differential operator.

Trigonometric Identities The nodes $\{\chi_j\}_{j=1}^{N}$ and $\{x_j\}_{j=0}^{N+1}$ could cause the $\mathbf{D}^{(p)}$ matrix to be ill-conditioned if they are extremely close together, since we divide by $\chi_j - x_k$. So, we use the identity

$$\chi_j - x_k = -2 \sin\left(\frac{\theta_j + \phi_k}{2}\right) \sin\left(\frac{\theta_j - \phi_k}{2}\right) \tag{3.25}$$

where

$$\theta_j = \frac{(2j-1)\pi}{2M} = \arctan(y_j) \quad \text{and} \quad \phi_k = \frac{(k-1)\pi}{N-1} = \arctan(x_k).$$

We refer the reader to [24] and [74] for how this reduces cancellation errors and increases accuracy.

The Flipping Trick The rectangular differentiation matrix $\mathbf{D}^{(p)}$ is skew-symmetric:

$$\mathbf{D}^{(p)}_{i,j} = (-1)^p \mathbf{D}^{(p)}_{N-1-i,N+1-j}$$

as discussed in [77]. In addition to increasing accuracy, is useful for defining $\mathbf{D}^{(p)}$ more efficiently as it utilizes the duplicity of off diagonal terms to do one calculation instead of two.

The Negative-Sum Trick The last trick implemented to increase the accuracy of our scheme is the negative-sum trick which was used in [12]. The idea is that constant functions must evaluate to zero when differentiated, so the sum of the diagonal entries is calculated by negating the sum of the rest of the row to ensure that

$$\sum_{i=0}^{N+1} \mathbf{D}_{ij} = 0. \tag{3.26}$$

Only square matrices were considered in [12], so this condition was enforced on the diagonal in the following way to increase the accuracy of the scheme:

$$\mathbf{D}_{ii} = -\sum_{\substack{j=0 \\ j\neq i}}^{N+2} \mathbf{D}_{ij}. \tag{3.27}$$

However, the idea that a constant function must be zero when differentiated must hold in the rectangular case as well and can be advantageously enforced a with a little cleverness.

Recall that each term of $\mathbf{D}^{(p)}$ is divided by the term $\chi_i - x_j$. The most ill-conditioned entry of any row would then be the one which has been divided by the smallest value of $|\chi_i - x_j|$. Replacing the most ill-conditioned value of the matrix with a value that would enforce (3.26) would then increase the accuracy of the rectangular scheme, so we assign it to be the negative of the sum of the rest of the row. In other words,

$$\mathbf{D}^{(p)}_{ij} = -\sum_{\substack{k=0 \\ k\neq j}}^{N+1} \mathbf{D}^{(p)}_{ik} \tag{3.28}$$

where j is chosen such that $|\chi_i - x_j| = \min_{j=1,..,N} |\chi_i - x_j|$.

3.2 Eigenvalues of the Laplacian

Upon establishing the numerical version of the Laplacian, the generation of numerical eigenvalues comes quickly thereafter. The analytical eigenvalue problem, namely (2.5), has the numerical

equivalent

$$\mathbf{Lv} = \lambda \mathbf{Pv}. \tag{3.29}$$

This problem can be solved fairly easily using built-in commands in MATLAB when we have Dirichlet boundary conditions. However, current algorithms involve inverting $\mathbf{L}$ to some extent, and $\mathbf{L}$ is a singular matrix in the case of no leaf vertices or when those leaf vertices do not have Dirichlet conditions. The method for accounting for the singular nature of $\mathbf{L}$ and recovering the original eigenvalues is described in the following section.

3.2.1 Shifting Method

This method will be discussed using the generalized eigenvalue problem

$$\mathbf{Av} = \lambda \mathbf{Bv} \tag{3.30}$$

as the explicit structures of $\mathbf{L}$ and $\mathbf{P}$ are not necessary and works for any singular matrix A.

Theorem 3.1. If λ is an eigenvalue for

$$(A + B)\vec{v} = \lambda B\vec{v} \tag{3.31}$$

with the associated eigenvector $\mathbf{v}$, then $\lambda - 1$ is an eigenvalue for (3.30) with the same associated eigenvector $\vec{v}$.

Proof. Suppose λ is an eigenvalue of (3.31). Then

$$\begin{aligned} (A + B)\vec{v} &= \lambda B\vec{v} \\ A\vec{v} + B\vec{v} &= \lambda B\vec{v} \\ A\vec{v} &= (\lambda - 1)B\vec{v} \end{aligned}$$

Thus $\lambda - 1$ is an eigenvalue of (3.30) with eigenvector $\vec{v}$. □

This method relies on the fact that singular square matrices exist in a subset of the space of all square matrices that has measure zero. Thus, any tiny random perturbation is almost certain to take you out of that thin subset. ("Almost certain" is actually a technical term here, meaning with

probability 1.) In the case that we are discussing, **P** serves as the perturbative term.

3.2.2 Results

The eigenvalues obtained using the shifting method were checked against the analytically obtained roots from secular determinant. Naturally, the first question is how many discretization points should we use per edge to ensure we are optimizing the accuracy of our eigenvalues. To accomplish this, the first nonzero eigenvalue of the Laplacian was examined at a variety of discretization points on the lollipop graph with Dirichlet boundary conditions and current conservation (Figure 3.7).

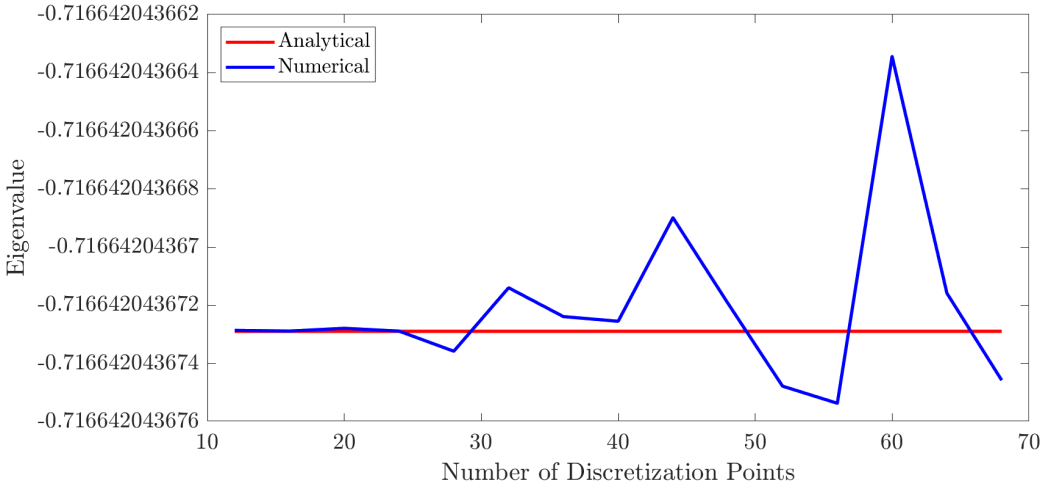


Figure 3.7: Comparison of first nonzero analytical and numerical eigenvalues

While $N = 16$ produced the most accurate result (relative error was 10^{-14}), the first nonzero eigenvalue produced by the discretizations between 12 and 68 were, at the worst, on the order of 10^{-12} accuracy. Further analysis of the spectrum revealed that 16 discretization points did not yield the best results for all the eigenvalues, but the range between 24 and 32 discretization points did best with accuracy on the order of 10^{-11} or better (see Figure 3.8).

Another comparison between the two discretization types (Chebyshev and uniform) was performed as well to ensure that the Chebyshev points are performing as expected compared to the uniform points. Figure 3.9 depicts how the first nonzero eigenvalue is more accurate over a range of discretization points, while Figure 3.10 compares the first 12 eigenvalues from each disctretization type. As expected, the eigenvalues produced by the Cheybshev discretization were substantially more accurate and needed far fewer points to achieve this accuracy.

Using $N = 32$ discretization points on each edge, the accuracy of the first ten eigenvalues were

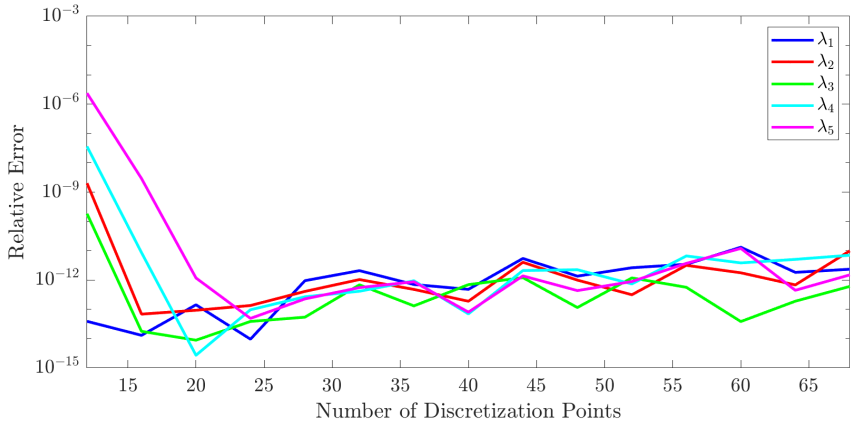


Figure 3.8: Error analysis for first five nonzero eigenvalues relative to analytically obtained eigenvalues from the secular determinant

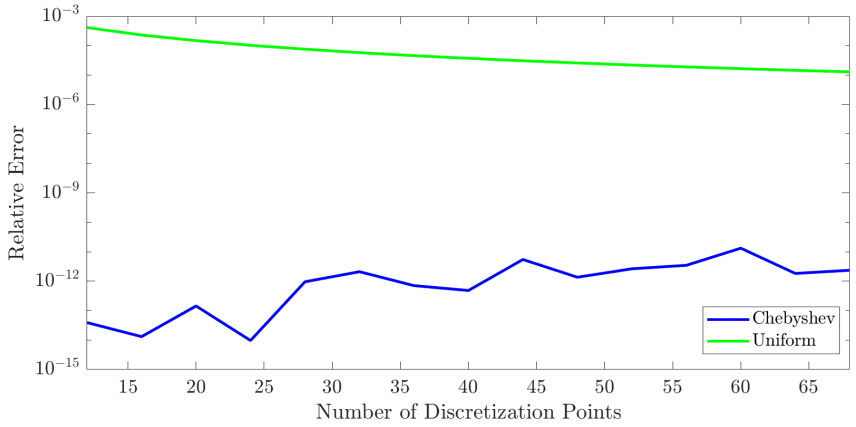


Figure 3.9: Error analysis for first nonzero eigenvalues to compare accuracy of the Chebyshev and uniform discretizations

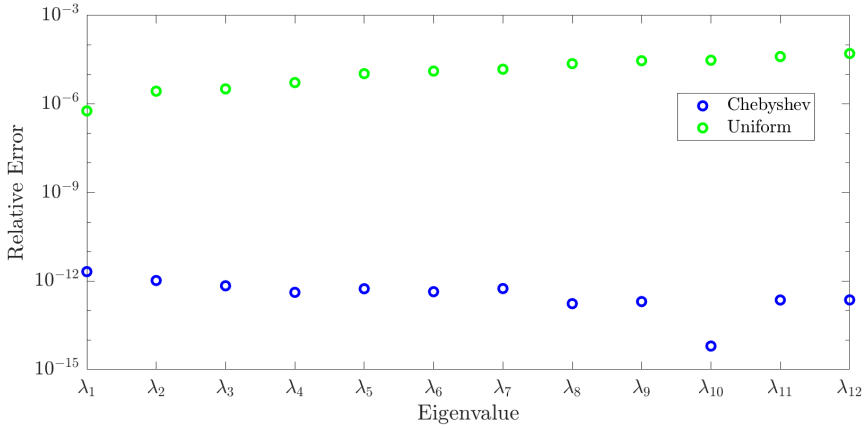


Figure 3.10: Error analysis for first 12 eigenvalues, $N = 32$ for the Chebyshev discretization while $N = 320$ for the uniform discretization

Graph	Leaf BC	Relative Error
Lollipop	Dirichlet	10^{-11}
Lollipop	Neumann	10^{-11}
Lollipop	Robin	10^{-11}
Star	Dirichlet	10^{-11}
Star	Neumann	10^{-12}
Star	Robin	10^{-12}
Dumbbell	N/A	10^{-12}

Table 3.1: First 10 eigenvalues, given $N = 32$

verified on a variety of graphs with various boundary conditions, and some of these results are summarized in Table 3.1.

Further analysis was done on other types of graphs to ensure this accuracy generalized to any graph, and the results from the other graphs matched the findings in Table 3.1. From this analysis, it is clear that 32 discretization points should be sufficient to optimize results in most cases and create a spatial operator that is operating at machine precision.

3.3 Linear Solvers: An Application to Spectral Flow

Here, we provide an application of the graph Laplacian which validates the limiting behavior of eigenvalues under spectral flow as described in [16]; this constitutes an important concept when studying the nodal deficiency of Laplace eigenfunctions. The work in [13] characterizes the eigenvalues and eigenfunctions with respect to the spectral flow parameter and derives explicit formulas for the limiting eigenfunctions. These results are the primary focus of this section.

Consider the linear Schrödinger eigenvalue problem with a potential V:

$$-\Delta u(x) + V(x)u = \lambda u(x), \qquad u(0) = u(1) = 0, \tag{3.32}$$

whose eigenvalues are $\lambda_1 < \lambda_2 \leq \lambda_3 \leq \cdots$ with corresponding eigenfunctions $\phi_1,\ \phi_2,\ \phi_3, \cdots$. One may be interested in quantifying the oscillations of ϕ_k in terms of the index k as one would do in Sturm-Liouville theory. To do this, it is helpful to study the interior zeros of ϕ_k in terms of the **nodal domain** - the segments between these interior zeros. The number of nodal domains is denoted by $\nu(\phi_k)$, and Courant's nodal domain theorem states that $\nu(\phi_k) \leq k$ (i.e. $\phi_k(x)$ has at

most $k-1$ interior zeros). Another way to say this is that the **nodal deficiency**

$$\delta^*(\phi_k) := k - \nu(\phi_k) \tag{3.33}$$

is nonnegative. If the index k^* is fixed, more information can be found about $\delta^*(\phi_{k^*})$.

This is accomplished by letting $\mathcal{Z}_{k^*} = \{x \in (0,1) \mid \phi_{k^*}(x) = 0\}$ denote the set of interior zeros for $\phi_{k^*}(x)$, which is used to define the spectral flow problem

$$-\Delta u(x) + V(x)u(x) + \sigma u(x) \sum_{k=1}^{k^*-1} \delta(x - x_k) = \lambda(\sigma)u(x), \qquad u(0) = u(1) = 0 \tag{3.34}$$

parametrized by $\sigma \in [0, \infty)$ and $x_k \in \mathcal{Z}_{k^*}$. It is proven in [13] that the nodal deficiency of ϕ_{k^*} was exactly equal to the number of eigenvalue curves, $\lambda_k(\sigma)$, that passed through λ_{k^*} as σ ranges from 0 to ∞.

This problem can be reformulated in language that easily translates to a quantum graph format by applying the Schrödinger operator $L = -\Delta + V(x)$ to each open subinterval (x_{k-1}, x_k) and imposing boundary conditions at the interval end points that both account for the delta function and enforce continuity. More succinctly, the problem becomes

$$\begin{cases} Lu(x) = \lambda(\sigma)u(x), & x \in \bigcup_{k=1}^{k^*} (x_{k-1}, x_k) \\ u(x_0 = 0) = u(x_{k^*} = 1) = 0, & \\ u(x_k^+) = u(x_k^-), & 1 \le k \le k^* - 1 \\ u'(x_k^+) - u'(x_k^-) = \sigma u(x_k), & 1 \le k \le k^* - 1 \end{cases} \tag{3.35}$$

where the superscript $+$ and $-$ denote right- and left-hand limits, respectively. This easily fits into the framework of quantum graphs where the interval endpoints $\{x_k\}_{k=0}^{k^*}$ represent vertices and the open intervals are edges. Dirichlet boundary conditions are enforced at the leaf vertices $x_0 = 0$ and $x_{k^*} = 1$, while matching conditions are applied to all of the interior vertices x_k for $1 \le k \le k^* - 1$. Through this lens, the work in [13] was numerically validated and a few results are provided below.

The results given illustrate how the value of σ affects eigenvalues for a given value of k^*. Observe in Figure 3.11 that the λ_k's for $1 \le k < k^*$ increases until they converge to λ_{k^*} as σ varies from 0

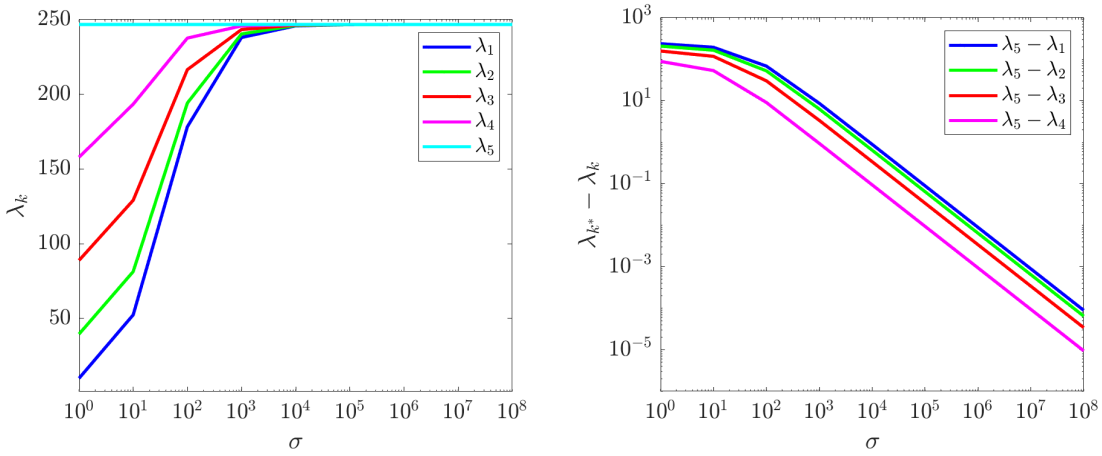


Figure 3.11: This illustrates the convergence of eigenvalues 1 through 4 to eigenvalue 5 when $k^* = 5$

to 10^8 where as λ_{k^*} stays constant. This is consistent with the findings in [16]. Fixing $\sigma = 10^8$, the corresponding eigenvectors are then depicted in Figure 3.12. The nodal deficiency of ϕ_5 can be quickly calculated by looking at Figure 3.12; ϕ_5 has 5 nodal domains, so its nodal deficiency is 0. Since all eigenvalue curves converge to λ_5, there are 0 eigenvalue curves passing through it which agrees with the findings in [13].

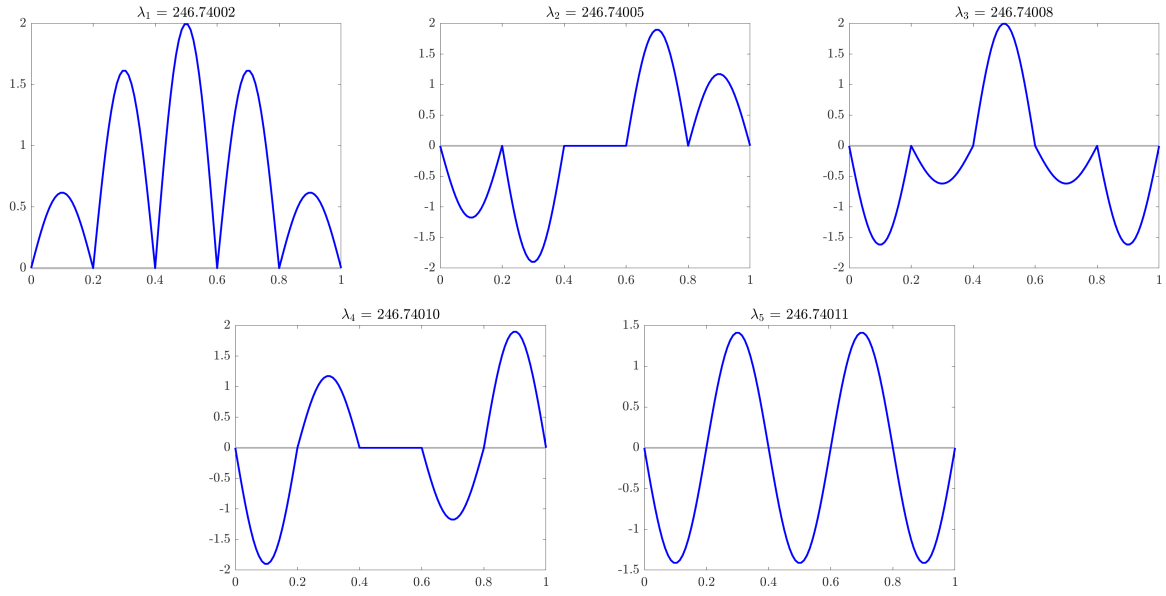


Figure 3.12: Fixing $k^* = 5$ and $\sigma = 10^8$, figures are of eigenfunctions corresponding to given eigenvalues

3.4 Solvers for Nonlinear Elliptic PDEs

Bifurcations of stationary waves are common phenomena in both conservative and dissipative nonlinear wave systems. Bifurcations of stationary waves induce qualitative changes to the wave behavior and can be used to control system outcome. Thus, their studies are both mathematically and physically important. A **bifurcation** occurs when a small smooth change to a **parameter** in the system causes the solution to have an abrupt change in its qualitative or topological behavior. The parameter's value at this point is called the **bifurcation point**. In particular, a bifurcation in a dynamical system could change the local stability properties of equilibria, cause periodic orbits to appear and disappear, or cause other invariant sets to change.

Consider a solution $(u(x), \mu)$ to the nonlinear, stationary Schrödinger problem defined in (2.4). If $\sigma = 0$ this is exactly the context covered in the previous sections. For this analysis, we will fix $\sigma = 2$ to be consistent with other works (such as [38]) so that the equation that we will work with is

$$u'' - V(x)u + 2u^3 = -\mu u. \tag{3.36}$$

To analyze our problem, we need the linearization operator (i.e. the Jacobian)

$$L_+ = \partial_{xx} - V(x) + \mu + 6u^2 \tag{3.37}$$

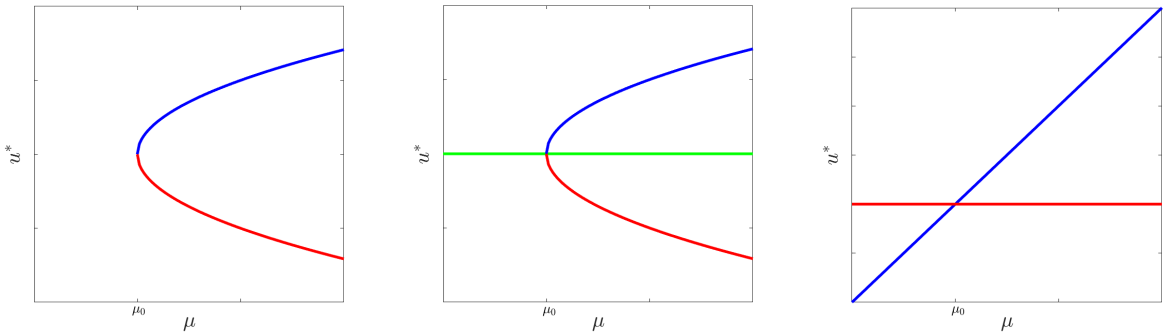
to determine bifurcations points. For a fixed $\mu = \mu_0$, the corresponding solitary wave will be denoted as $u_0(x)$, and the linearization operator is denoted as

$$L_{+0} = L_+|_{\mu=\mu_0, u=u_0}. \tag{3.38}$$

If L_{+0} is invertible, the implicit function theorem can be applied, and we know that on an open interval containing μ_0, there exists a unique family of solutions. However, if L_{+0} is singular, the implicit function theorem does not apply, and the existence or uniqueness of nearby solutions is no longer guaranteed. Thus, the point (u_0, μ_0) at which L is singular is called a bifurcation point. The three most commonly-seen bifurcations of stationary waves are the saddle-node, transcritical, and pitchfork bifurcations; these will be the focus of this section.

A **saddle-node bifurcation** (or **fold bifurcation**) occurs when two distinct equilibria collide

and disappear entirely. Visually, it has two branches (one stable, one unstable) of solutions on one side of $\mu = \mu_0$, which merge smoothly at μ_0 (Figure 3.13a). A **pitchfork bifurcation** occurs when a system transitions from one fixed point to three fixed points. It is characterized by having a single branch on one side of $\mu = \mu_0$ and three branches on the other side (Figure 3.13b). The single branch continues smoothly through the bifurcation and is the middle branch on the side with three branches. The other two branches on this side are on the top and bottom and connected smoothly at μ_0. A **transcritical bifurcation** is characterized by a fixed point that exists for all values of the bifurcation parameter and is never destroyed. But, for certain values of the parameter, there is a second fixed point, and the points exchange stability when they collide. Visually, it looks like two branches crossing each other at μ_0 (Figure 3.13c). Note that as a consequence, this ensures there will be one stable and one unstable equilibria for all values of μ except at the point of collision, μ_0. Sufficient conditions for these major types of solitary wave bifurcations are derived in [79] where the focus is the simple zero eigenvalue of L_{+0}.



(a) Saddle-node bifurcation (b) Pitchfork bifurcation (c) Transcritical bifurcation

Figure 3.13: Basic example visualizations of each of the three main bifurcation types

To find solutions to the NLS, QGLAB bifurcates off of the solutions to the nonlinear problem to obtain solutions to the linear problem. To accomplish this, QGLAB takes in a specified eigenfunction to initialize finding $u_0(x)$, then finds bifurcation points by varying μ, utilizing a Lyapunov-Schmidt reduction. The branches for both are computed using pseudo-arclength continuation. At the bifurcation points, a Newton solver is utilized to land on a new branch and find additional solutions to the NLS. Further computation details follow as described in [39]. Figure 3.14 illustrates a bifurcation diagram for the dumbbell graph where μ is varied from -1 to 0. It is known in this case that there will be two bifurcations (as shown in [62]). Note that Figure 3.14 recreates the bifurcation

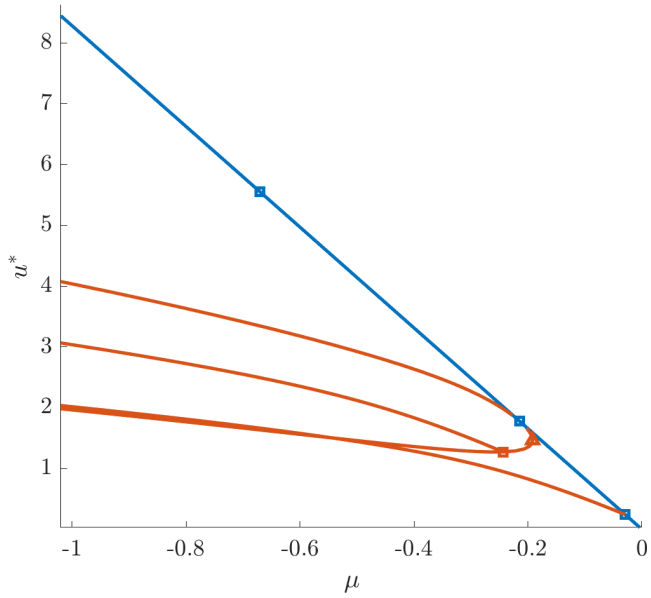
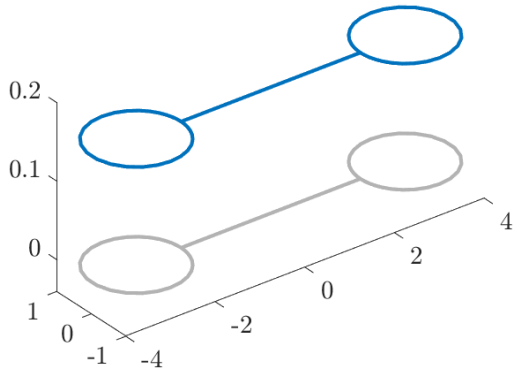


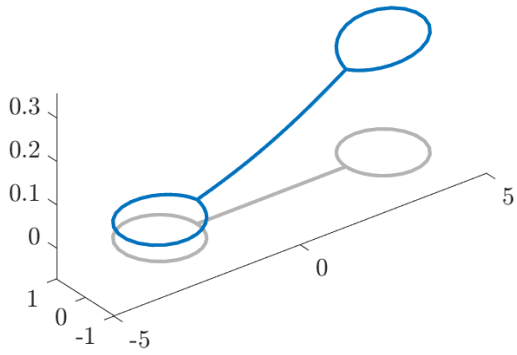
Figure 3.14: A bifurcation diagram for the dumbbell graph. he blue curve is the continuation of the first eigenfunction. The red curves were computed by following branches from bifurcation points. Saddle-node bifurcations are indicated with a triangle while squares denote branching bifurcations.

diagram found in [38, Fig 3.2], where a uniform grid (instead of Chebyshev points) was use.

With these bifurcations, we can now discuss the solutions to (3.36). Notice how the two solutions taken from the top branch (Figure 3.15a and Figure 3.16a) are both trivial solutions while both solutions taken from the bottom branch (Figure 3.15b and Figure 3.16b) exude similar behaviors (e.g. they are both monotonically increasing if the graph is read from left to right).

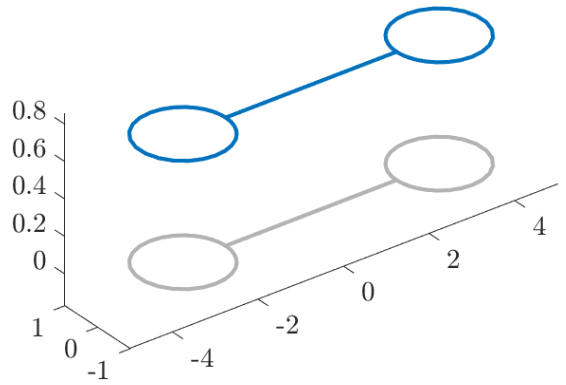


(a) Blue branch

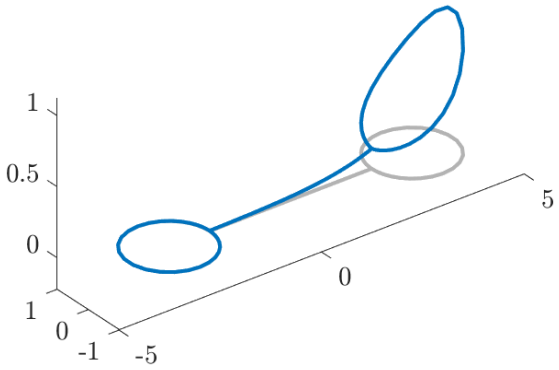


(b) Bottom branch (red)

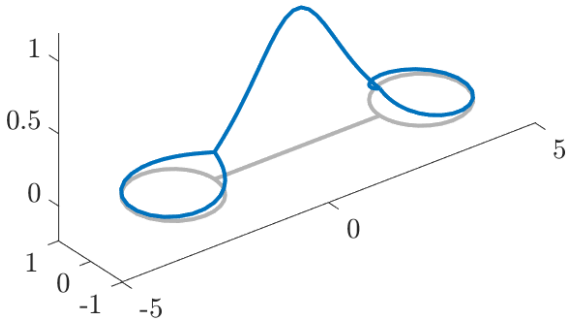
Figure 3.15: Solutions from different branches at $\mu = -0.1$



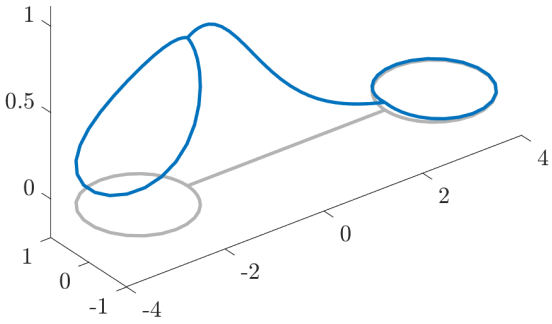
(a) Blue branch



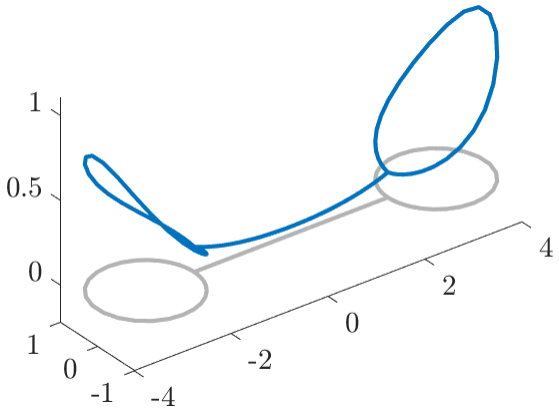
(b) Bottom branch



(c) Second from bottom branch



(d) Third from bottom branch



(e) Fourth from bottom branch

Figure 3.16: Solutions from different branches at $\mu = -0.8$

Also notice the correspondence between Figure 3.16c and Figure 3.16e. They have reversed behavior, which comes from the fact that they are results of the same bifurcation point (red triangle) but are solutions produced by going in different directions from that bifurcation point.

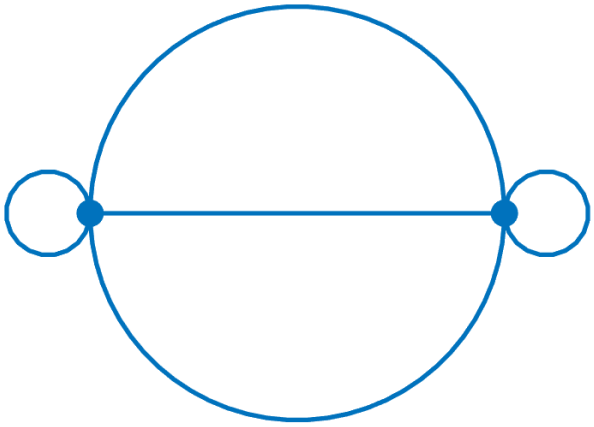


Figure 3.17: Berkolaiko-Marzuola-Pelinovsky (BMP) quantum graph layout

Another example is also calculated for the Berkolaiko-Marzuola-Pelinovsky quantum graph (Figure 3.17) which is used in [18]. The blue curve is the continuation of the first eigenfunction. The first bifurcation point is a branching bifurcation and creates the bottom red branch. The second bifurcation point is a saddle-node bifurcation, and it produces the top two red curves.

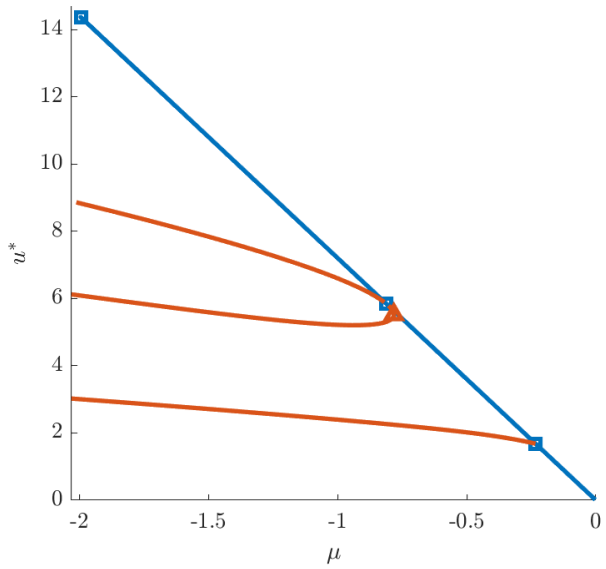
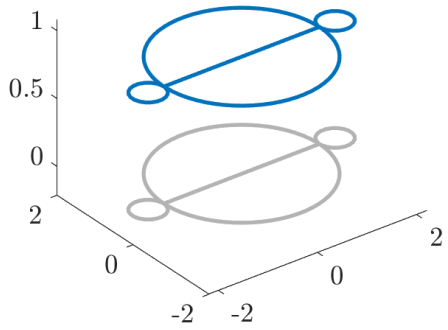
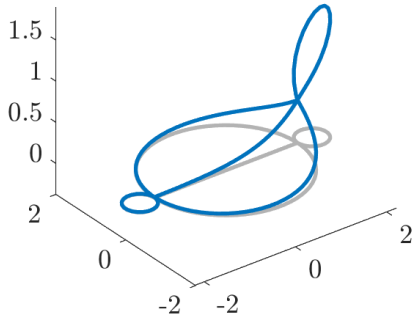


Figure 3.18: A bifurcation diagram for the BMP graph

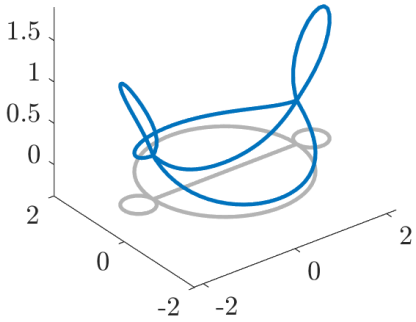
The solutions affiliated with each branch around the value of $\mu = -2$ are displayed in Figure



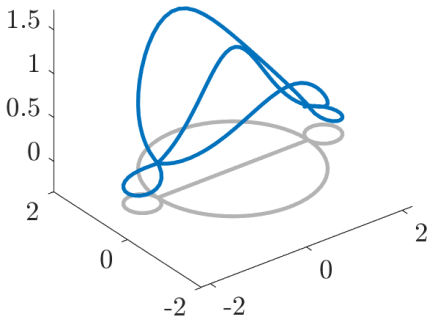
(a) Blue branch



(b) Bottom branch (red)



(c) Middle red branch



(d) Top red branch

Figure 3.19: Solutions from different branches at $\mu = -1.5$

3.19. The behavior of the saddle-node bifurcation branch solutions are as expected; they appear to be inverted versions of each other which was also observed in the dumbbell case.

While this section has discussed the archetypal bifurcations one encounters in this field, section 5.2.3 will discuss another important bifurcation called the Hamiltonian-Hopf bifurcation, which relies on analysis of linearized operators around steady states and is used in the context of time-periodic solutions. Before delving into this topic, we will first discuss the time-dependent problem.

CHAPTER 4

Time Evolution

This chapter describes two different methods implemented in QGLAB for performing time evolution on graphs. The first method utilizes known techniques from the field of differential algebraic equations, and its application to quantum graphs follows rather quickly. The second method described derives a new scheme using known Runge-Kutta methods as a basis and generalizes them to work for operators with encoded boundary conditions. It is worth noting that while quantum graphs were the motivating factor, our adapted Runge-Kutta scheme works for any system with boundary conditions.

4.1 Formulation of the Problem

Time evolution is complicated by the incorporation of boundary conditions in the spatial discretization of our operator. This encoding of the boundary conditions leads to a spatial discretization matrix which is poorly-conditioned and, thus, needs to be paired with a numerical time-evolution scheme that is more stable than accurate. Our approach here is motivated by that used in [28, §7.5] in the discussion of solving nonlinear PDE using the Chebfun package.

We formulate the PDE in the following abstract form:

$$\begin{cases} \dfrac{\partial u}{\partial t} = F(t,u), & t > 0 \qquad (4.1a) \\ l_i(u,u') = 0, & i = 1,\dots,2|\mathcal{E}| \qquad (4.1b) \\ u(x,0) = f(x), \qquad (4.1c) \end{cases}$$

where F is a (possibly nonlinear) 2nd-order differential operator in x, m is the number of edges, and l_i's are constant linear functionals that gives us necessary boundary conditions.

While the main focus for discretization points has been Chebyshev points, this formulation of the problem allows for the user to choose either uniform or Chebyshev points as their discretization choice. The key idea here is working with spatial operators that have incorporated boundary conditions

as was discussed for the definition of the Laplacian. Much like that discussion, F must also be discretized carefully. For instance, if $\mathbf{u}$ is defined using Chebyshev points of the second kind, $\mathbf{F}$ incorporates the projection from the Chebyshev points of the second kind to the first kind. Since $\mathbf{u}$ would also be defined on origonal discretization that includes the end points, $\mathbf{P}_{\text{int}}$ is multiplied to the left-hand side of the (4.1a) so it will also defined on the second, smaller set of discretization points that $\mathbf{F}$ is defined on. More details for the uniform case are discussed in [40].

The original discretization will be denoted as $\mathbf{x}$ and the smaller discretization that does not incorporate the boundary vertices will be $\boldsymbol{\chi}$. Thus, when the Chebyshev discretization is being used, $\mathbf{x}$ is points of the second kind while $\boldsymbol{\chi}$ is points of the first kind as is illustrated in Figure 3.4. When the uniform discretization is being used (recall Figure 3.1), $\mathbf{x}$ denotes the full set of discretization points, $\{x_0, x_1, \ldots, x_{N+1}\}$, and $\boldsymbol{\chi}$ is restricted to the interior points, namely $\{x_1, \ldots, x_N\}$. The discretized system can now be given as

$$\begin{cases} \mathbf{P}_{\text{int}} \frac{d\mathbf{u}}{dt} = \mathbf{F}(t, \mathbf{u}), \\ \mathbf{L}_{\text{VC}} \mathbf{u} = \mathbf{0}_{2|\mathcal{E}| \times (N_{\text{tot}} + 2|\mathcal{E}|)}, \end{cases} \tag{4.2}$$

where $\mathbf{P}_{\text{int}}$ and $\mathbf{L}_{\text{VC}}$ retain their definitions from earlier. This can be written more concisely as

$$\begin{pmatrix} \mathbf{P}_{\text{int}} \\ \mathbf{0} \end{pmatrix} \frac{d\mathbf{u}}{dt} = \begin{pmatrix} \mathbf{F}(t, \mathbf{u}) \\ \mathbf{L}_{\text{VC}} \mathbf{u} \end{pmatrix}. \tag{4.3}$$

With this numerical formulation, there were two main methods for time evolution that were considered, each with their pros and cons that will be discussed.

4.2 Differential-Algebraic Equation Method

Since boundary conditions have been encoded as algebraic conditions, this is constitutes a differential-algebraic equation, and we must decide how to evolve it in time. The problem can be written as

$$\mathbf{P}\mathbf{u}_t = \mathbf{A}(\mathbf{u}), \qquad \mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0, \tag{4.4}$$

where

$$\mathbf{A}(\mathbf{u}) = \begin{pmatrix} \mathbf{F}(t, \mathbf{u}) \\ \mathbf{L}_{\text{VC}} \mathbf{u} \end{pmatrix}.$$

With this formulation of the problem, MATLAB's built-in stiff solver `ode15s` can now be utilized by setting $\mathbf{P}$ to be the mass matrix and turning the `massing` option on. The tolerance is set so that it does not exceed the accuracy of the initial condition, $\mathbf{P}\mathbf{u}_0$.

The example of the nonlinear Schrödinger equation on the dumbbell graph

$$\begin{cases} iu_t = -u_{xx} - 2|u|^2 u & \\ u_1'(\ell_1) = 0 \quad u_1(0) = u_1(\ell_1) = u_2(0) & \text{Continuity condition} \\ u_2(\ell_2) = u_3(0) = u_3(\ell_3) & \\ -u_1'(0) + u_1'(\ell_1) - u_2'(0) + \beta_1 u_1(0) = 0 & \text{Kirchhoff condition} \\ u_2'(\ell_2) - u_3'(0) - u_2'(0) + \beta_2 u_3(\ell_3) = 0 & \\ u(0,x) = \phi(x) & \text{Initial condition} \end{cases} \tag{4.5}$$

where $\phi(x)$ is a solution to the stationary problem, was used to test the accuracy of the method; each edge was discretized with $N = 32$ points. The accuracy was determined by checking for conservation of energy and mass, where the **energy** and **mass** of the system are defined by

$$E = \sum_{j=1}^{|\mathcal{E}|} \omega_j \int_{e_j} \left(|u_j'(x)|^2 - |u_j(x)|^4 \right) dx \quad \text{and} \quad M = \sum_{j=1}^{|\mathcal{E}|} \omega_j \int_{e_j} |u_j(x)|^2 \, dx. \tag{4.6}$$

The resulting errors were on the order of 10^{-07} and 10^{-08}, respectively, as summarized in Figure 4.1.

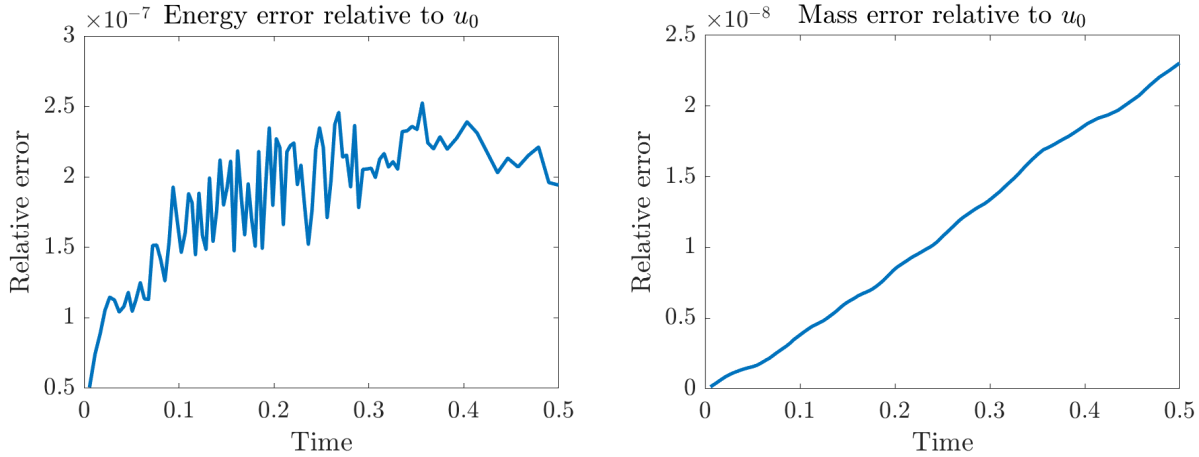


Figure 4.1: Convergence of DAE scheme

While this method runs relatively quickly and the results are sufficiently accurate, the next method improves upon these results and provides the user with more control over the time-step,

which may be desired in some cases.

4.3 Adapted Runge-Kutta Schemes for Stiff IVPs

While the previous method works by thinking about the problem through the from the perspective of DAEs, another lens one could use to view this problem through is that of a stiff initial value problem. In this context, diagonally implicit Runge-Kutta schemes are known for their stability properties. However, they are not structured for spatial operators with incorporated boundary conditions, so the goal of this section is to derive a new version of the known Runge-Kutta methods to work in this situation.

For the adapted Runge-Kutta method (ARK), the notation $\tilde{\mathbf{u}} = \mathbf{P}\mathbf{u}$ is introduced, and we pause to make two important observations that will impact the development of this algorithm.

1. Note that $\tilde{\mathbf{u}}$ and $\mathbf{F}$ are discretized by the grid $\boldsymbol{\chi}$ while $\mathbf{u}$ is discretized by the grid $\mathbf{x}$. That is, the input and output of $\mathbf{F}$ are defined on different discretizations.

2. Since our equation can be written $\tilde{\mathbf{u}}_t = \mathbf{F}(t, \mathbf{u})$, it is easiest to consider what it would mean to use a Runge-Kutta scheme to advance $\tilde{\mathbf{u}}$ and not $\mathbf{u}$.

We want to take advantage of a typical Runge-Kutta scheme, which steps the solution forward in time with a formula akin to

$$\tilde{\mathbf{u}}_{n+1} = \tilde{\mathbf{u}}_n + h \sum_{i=1}^{s} b_i \mathbf{k}_i, \tag{4.7}$$

$$t_{n+1} = t_n + h. \tag{4.8}$$

We should take care into how to properly define $\mathbf{k}_i$. *Normally*, $\mathbf{k}_i$ would be defined by setting it equal to $\mathbf{F}$, which has carefully chosen inputs (ie: $\mathbf{k}_1 = \mathbf{F}(t_n, \mathbf{u}_n)$)). However, $\tilde{\mathbf{u}}$ has rows of zeros at the bottom in place of boundary conditions, whereas $\mathbf{F}$ has been resized to leave room for boundary data to be added. This means that the dimensions of $\mathbf{k}_i$ would not be compatible for matrix addition with $\tilde{\mathbf{u}}$. For this reason, it is better to defined $\mathbf{k}_i$ by also including rows of zeros in the following way:

$$\mathbf{k} = \begin{pmatrix} \mathbf{F} \\ \\ \mathbf{0} \end{pmatrix}.$$

Before explicitly defining the $\mathbf{k}_i$'s, we should also recall that the input and output of $\mathbf{F}$ are defined on two different sets of discretization points. This presents a minor challenge since we would *normally* define a $\mathbf{k}_i$ by adding scalar multiples of $\mathbf{k}_j$'s to $\mathbf{u}_n$ (i.e. we might want to use $\mathbf{F}(t_n+c_2h, \mathbf{u}_n+ha_{21}\mathbf{k}_1)$ when defining $\mathbf{k}_2$). But by definition, the $\mathbf{k}_i$'s will be defined on $\boldsymbol{\chi}$. Since $\mathbf{u}_n$ is defined on $\mathbf{x}$, $\mathbf{k}_i$ cannot be added to the $\mathbf{u}_n$ inside of $\mathbf{F}$. We will need to find the corresponding variant of $\mathbf{k}_i$ on the same discretization as $\mathbf{u}_n$.

It is also important to remember that the boundary conditions are incorporated in $\mathbf{u}_n$ but not in $\mathbf{k}_i$. So, we will want to use a matrix that both appropriately projects our information *and* satisfies the boundary conditions. Namely, we will use

$$\mathbf{P}_{\text{VC}} = \begin{pmatrix} \mathbf{P}_{\text{int}} \\ \mathbf{L}_{\text{VC}} \end{pmatrix}, \tag{4.9}$$

and define a new variable $\boldsymbol{\kappa}_i$ by

$$\mathbf{P}_{\text{VC}}\boldsymbol{\kappa}_i = \mathbf{k}_i \tag{4.10}$$

which guarantees that $\boldsymbol{\kappa}_i$ is defined on the same set of discretization points as $\mathbf{u}_n$.

Thus we can properly define the $\mathbf{k}_i$'s in the following way:

$$\mathbf{k}_1 = \begin{pmatrix} \mathbf{F}(t_n, \mathbf{u}_n + ha_{11}\boldsymbol{\kappa}_1)) \\ \mathbf{0} \end{pmatrix} \tag{4.11a}$$

$$\mathbf{k}_2 = \begin{pmatrix} \mathbf{F}(t_n + c_2h, \mathbf{u}_n + ha_{21}\boldsymbol{\kappa}_1 + ha_{22}\boldsymbol{\kappa}_2) \\ \mathbf{0} \end{pmatrix} \tag{4.11b}$$

$$\mathbf{k}_3 = \begin{pmatrix} \mathbf{F}(t_n + c_3h, \mathbf{u}_n + h(a_{31}\boldsymbol{\kappa}_1 + a_{32}\boldsymbol{\kappa}_2 + a_{33}\boldsymbol{\kappa}_3)) \\ \mathbf{0} \end{pmatrix} \tag{4.11c}$$

$$\vdots$$

$$\mathbf{k}_s = \begin{pmatrix} \mathbf{F}(t_n + c_sh, \mathbf{u}_n + h(a_{s1}\boldsymbol{\kappa}_1 + a_{s2}\boldsymbol{\kappa}_2 + \cdots + a_{s,s-1}\boldsymbol{\kappa}_{s-1})) \\ \mathbf{0} \end{pmatrix}. \tag{4.11d}$$

We now make the final observation that the boundary conditions are not being enforced at each step. There are rows of zeros in place of the boundary conditions on both sides of (4.7), which means that

we will not want to simply have $\tilde{\mathbf{u}}_{n+1}$ as is written on the left-hand side. Since we want to have $\mathbf{L}_{\text{VC}}$ set equal to zero and also want $\mathbf{u}$ to be updated in a way that will enforce the boundary conditions, we once more make use of $\mathbf{P}_{\text{VC}}$. Now, we will not only project $\mathbf{u}$ onto the right discretization (as done in $\tilde{\mathbf{u}}$), but also include the boundary conditions in the new step:

$$\mathbf{P}_{\text{VC}}\mathbf{u}_{n+1} = \tilde{\mathbf{u}}_n + h\sum_{i=1}^{s} b_i\mathbf{k}_i.$$

Hence the adapted Runge-Kutta scheme that will iterate the solution for (4.3) through time is:

$$\mathbf{P}_{\text{VC}}\mathbf{u}_{n+1} = \mathbf{P}\mathbf{u}_n + h\sum_{i=1}^{s} b_i\mathbf{k}_i, \tag{4.12}$$

where $\mathbf{k}_i$ is defined in (4.11).

This new scheme is implemented in QGLAB uses coefficients defined by Nørsett's three-stage, 4^{th} order diagonally implicit Runge-Kutta method.

The same problem as posed in (4.5) was also used to benchmark this scheme once again using $N = 32$. Conservation of energy and mass were similarly checked and results were on the order of 10^{-7} and 10^{-9}, respectively, as summarized in Figure 4.2. While these results are slightly more accurate than that of the DAE method, this method ran much more slowly. However, this scheme does have the advantage of giving control over the time step selection to the user.

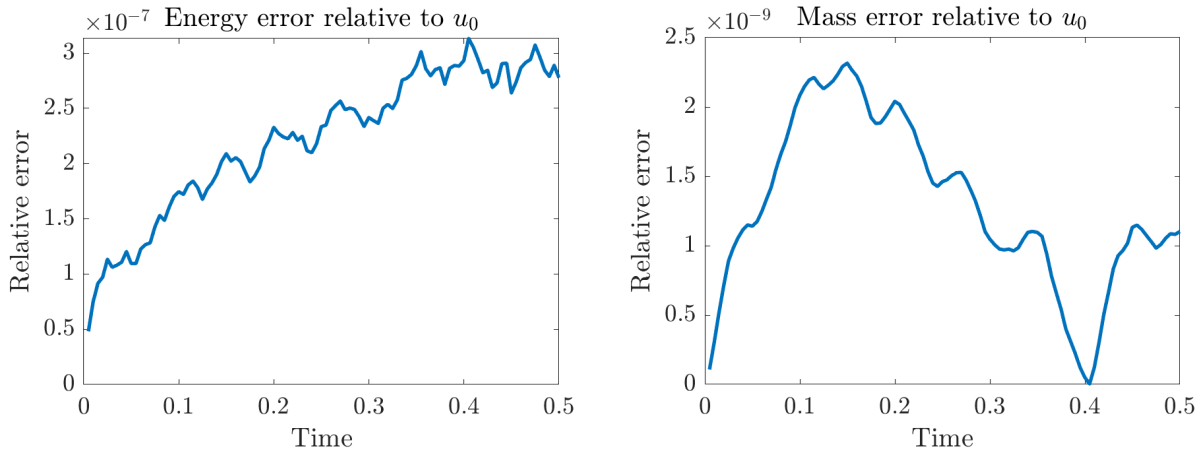


Figure 4.2: Convergence of ARK scheme

A second convergence study for the ARK method is also provided. This one is performed on the lollipop graph using the problem

$$\begin{cases} iu_t = -u_{xx} & \\ u_1(\ell_1) = u_2(0) = u_2(\ell_2) & \text{Continuity condition} \\ u_1'(\ell_1) - u_2'(0) + u_2'(\ell_2) = 0 & \text{Kirchhoff condition} \\ u_1'(0) = 0 & \text{Neumann boundary condition.} \end{cases} \tag{4.13}$$

The linear Schrödinger equation is used in this case so that the numerical results can be compared to the analytical solutions. For this convergence analysis, both the spatial and temporal regimes were analyzed for the ARK method and results are summarized in Figure 4.3. Once again we can see that 16 is not *quite* as good as $N_x = 32$. More importantly, we can observe that the accuracy increases linearly by shrinking the size of the time step.

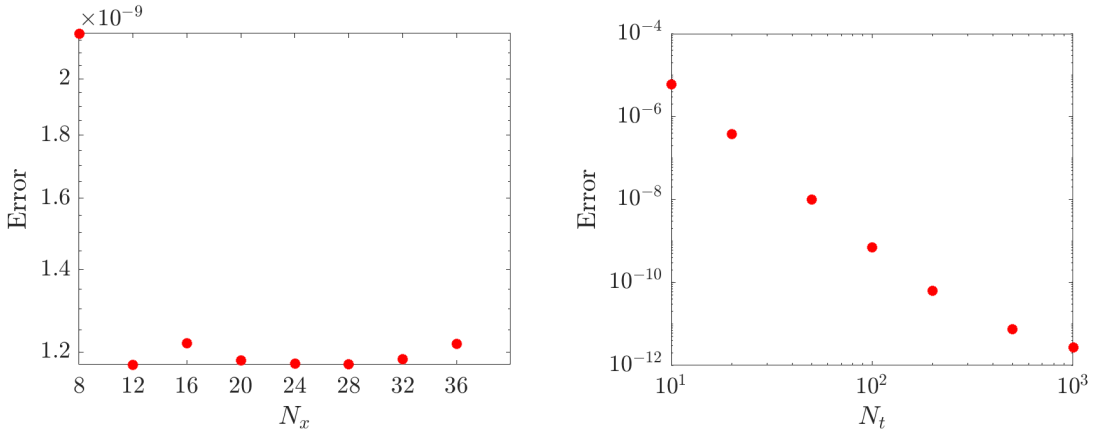


Figure 4.3: This convergence study compares the numerical results from the ARK method to the analytical solution. On the left, we fixed the time step at $\Delta t = 10^{-3}$. On the right, we fixed the number of spatial steps at $N_x = 32$.

CHAPTER 5

Linearization About Steady States and Progress Towards Time-Periodic Solutions

The study of partial differential equations from a dynamical perspective can provide illuminating perspective on the long-term behavior of solutions. In this context, one often wants to know whether a given solution is stable, and whether or not this property can be changed by introducing a small perturbation to the initial conditions. Bifurcation theory reveals that there are only a finite number of ways for a solution to transfer from being stable to unstable and in each of these scenarios, certain types of solutions and dynamics may be observed. One particular type of bifurcation that can give rise to much more complicated dynamics is the Hamiltonian-Hopf bifurcation (HHB). While the linear instabilities have been well-analyzed (see [68]), nonlinear wave dynamics in the neighborhood of such bifurcations is less known. One step in this direction was made in [38], where Hamiltonian-Hopf bifurcations of stationary, solitary waves were examined for the nonlinear Schrödinger equation with symmetric, triple-well type potentials. The phenomenon that we are particularly interested in is the investigation of time-dependent dynamics. Here, we speak of tunneling, as the solution moves between various wells. This can be periodic or chaotic, and we aim to uncover the specific circumstances under which periodic solutions arise.

In this chapter, the previously-discussed tools are now also implemented for a non-self-adjoint problem, laying necessary ground work for using QGLAB to find quasi-time-periodic solutions.

5.1 Motivation for Linearization Around Steady States

It is well-known that time periodic solutions exist for finite-dimensional Hamiltonian systems by the Lyapunov center theorem, but the goal of this section is to set up a numerical framework for pursuing a search for time-quasi-periodic solutions for an *infinite*-dimensional Hamiltonian system on a compact domain. It is reasonable to believe this is possible given that near time-periodic solutions for NLS with potentials on $\mathbb{R}$ have been found in [38, 80] which has certain implications to the expected behavior on the dumbbell graph (see [18]). Furthermore, full time-periodic solutions

for NLS with no potential on the torus were found in [78]. This result is quite relevant to our case because time-periodic solutions were found due to the geometry of the space that was being worked in. In particular, the geometry gave compactness and quantum graphs also fall under the domain of compact spaces.

5.2 Role of Linearized Operators in Finding Relative Periodic Orbits

Certain types of bifurcations can cause periodic orbits to appear and disappear. The bifurcation point where this occurs yields crucial information regarding the initial condition and period of the solution. In particular, the Lyapunov center theorem tells us that periodic orbit solutions should arise near purely imaginary eigenvalues of the linearized operator; thus, this will be our focus.

5.2.1 Complex Eigenvalues for the Cubic NLS Equation

In the context of the cubic NLS equation

$$iu_t = -u_{xx} - 2|u|^2 u, \tag{5.1}$$

we know that solitons have the form

$$u(x,t) = e^{-i\mu t} v(x), \tag{5.2}$$

where $v(x)$ is a real-valued localized function solving the associated stationary problem

$$v''(x) + 2v^3(x) = -\mu v(x), \tag{5.3}$$

and μ is a negative real-valued propagation constant (bifurcation parameter). These solitons exist as continuous families parametrized by μ, and it is assumed that the solitons are differentiable with respect to μ.

When considering the stability of these solitons, we linearize (5.1) with the normal-mode perturbation

$$u(x,t) = e^{i\mu t}\left[u(x) + f_1(x)e^{\lambda t} + \bar{f}_2(x)e^{\bar{\lambda} t}\right], \qquad f_1, f_2 \ll 1 \tag{5.4}$$

where the overbar represents complex conjugation. In order for (5.4) to be a solution for (5.1), f_1

and f_2 must satisfy the following linear eigenvalue problem

$$\mathcal{H}\begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \lambda \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}, \tag{5.5}$$

where we have the eigenvalue-eigenvector pair (λ, v) with $v = (f_1, f_2)^T$ and $\mathcal{H} = J^*\mathcal{L}^*$ which are defined by

$$J^* = i\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad \text{and} \quad \mathcal{L}^* = \begin{bmatrix} \partial_{xx} + \mu + 4u^2(x) & 2u^2(x) \\ 2u^2(x) & \partial_{xx} + \mu + 4u^2(x) \end{bmatrix}. \tag{5.6}$$

(To verify the definition of $\mathcal{L}^*$, simply plug (5.4) into (5.1) and note that only linear terms of f_1 and f_2 are kept since remaining terms are relatively negligible.)

Alternatively, the problem can also be linearized utilizing the following perturbation instead:

$$u(x,t) = e^{i\mu t}\left[u(x) + w_1(x)e^{\lambda t} + iw_2(x)e^{\lambda t}\right]. \tag{5.7}$$

If (5.7) is to also serve as a solution for (5.1), w_1 and w_2 would need to satisfy a slightly different linear eigenvalue problem than before, namely

$$J\mathcal{L}\begin{bmatrix} w_1 \\ iw_2 \end{bmatrix} = \lambda \begin{bmatrix} w_1 \\ iw_2 \end{bmatrix}, \tag{5.8}$$

where the eigenvalue-eigenvector pair would be (λ, w) with $w = (w_1, iw_2)^T$. Here J and $\mathcal{L}$ are defined in the following way:

$$J = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \quad \text{and} \quad \mathcal{L} = \begin{bmatrix} \partial_{xx} + \mu + 6u^2(x) & 0 \\ 0 & \partial_{xx} + \mu + 2u^2(x) \end{bmatrix} \tag{5.9}$$

and have similar properties to the J^* and $\mathcal{L}^*$ defined in (5.6).

Since we have linearized (5.1) in two different ways, it is important to note that (5.5) and (5.8) produce the same spectrum as is illustrated in Figure 5.2a.

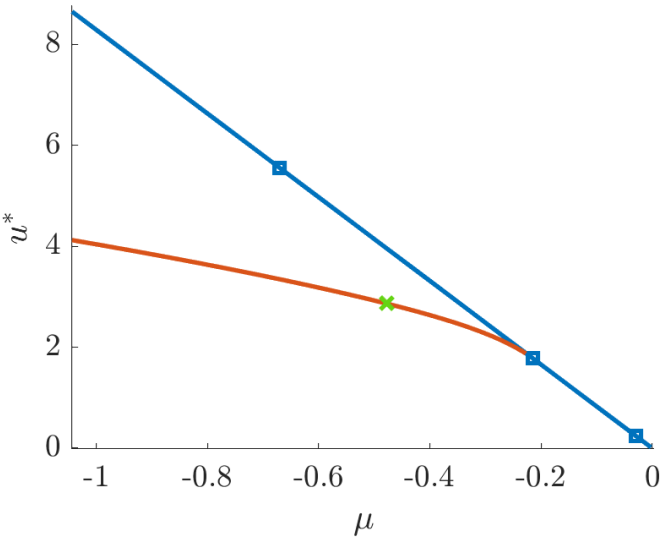
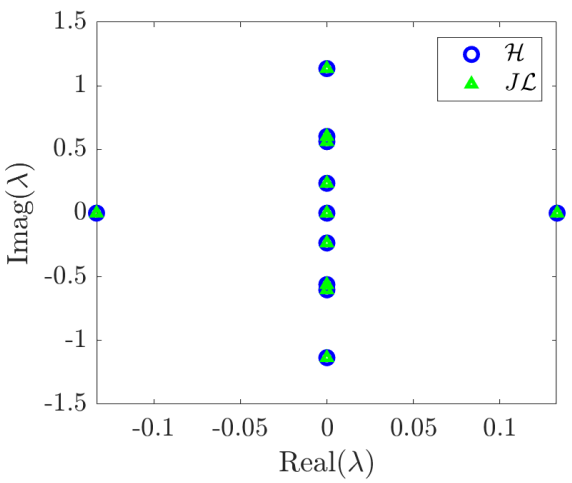
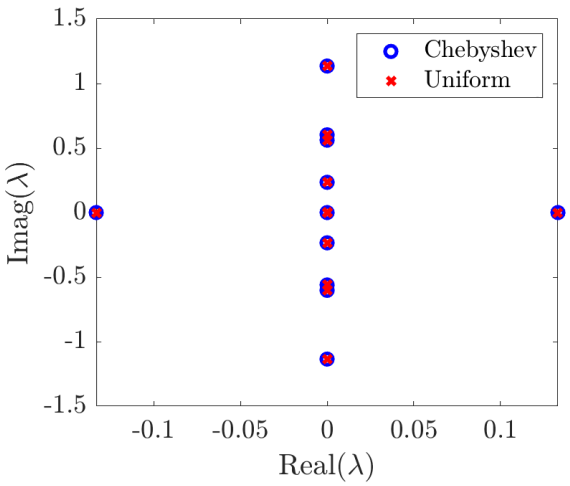


Figure 5.1: The $\mathcal{H}$ and $J\mathcal{L}$ operators were computed on the indicated branch for the indicated value of μ

The $\mathcal{H}$ and $J\mathcal{L}$ operators were numerically generated for bifurcation parameter $\mu = -0.477919$ on the branch indicated in Figure 5.1. The $\mathcal{H}$ spectrum was analyzed numerically by comparing results between the uniform and Chebyshev cases (see Figure 5.2b). The eigenvalues of $\mathcal{H}$ matched between to the two discretizations on at least the order of 10^{-5} (see Figure 5.3), which is about the same discrepancy found between the two discretizations results when the spectrum of the Laplacian

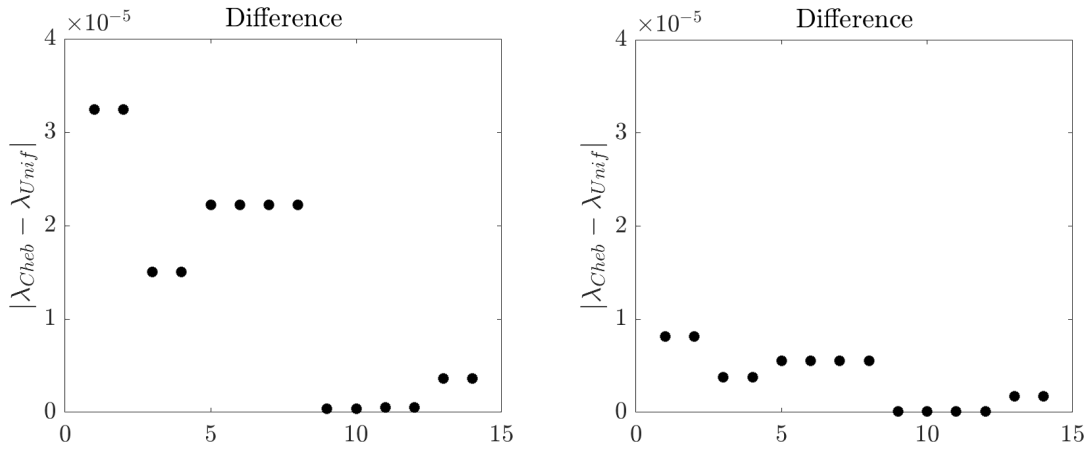


(a) Comparison of the $\mathcal{H}$ spectrum and $J\mathcal{L}$ spectrum



(b) $\mathcal{H}$ spectrum calculated using Chebysehv and uniform points

Figure 5.2: $\mathcal{H}$ and $J\mathcal{L}$ spectrum analysis



(a) $N_{\text{Cheb}} = 32$ and $N_{\text{Unif}} = 64$ (b) $N_{\text{Cheb}} = 32$ and $N_{\text{Unif}} = 128$

Figure 5.3: Number of uniform discretization points were doubled while it was static for the Chebyshev discretization

was analyzed. Doubling the number of points used for the uniform discretization causes the points to match even more closely as depicted in Figure 5.3.

5.2.2 Derivation of the Normal Form

The **normal form** of a differential equation is another differential equation that is found by making a polynomial change of variables that locally improves the nonlinear system in such a way that we can more easily recognize its dynamics (see [32]). In this section, we derive an asymptotically accurate ODE model (the normal form) for wave dynamics near the purely imaginary eigenvalue of $J\mathcal{L}$ calculated at μ_0.

Let $\mu = \mu_0$ be our bifurcation point. We assume that the full PDE solution can be expanded into the perturbation series

$$u(x,t) = e^{i\theta}\left[u_0(x) + \varepsilon u_1(x,t,T) + \varepsilon^2 u_2(x,t,T) + ...\right], \tag{5.10}$$

where

$$\theta(t,T) = \mu_0 t + \varepsilon \int_0^T \mu_1(\tau)\, d\tau + \varepsilon^2 \int_0^T \mu_2(\tau)\, d\tau + ... \tag{5.11}$$

for $T = \varepsilon t$ with small parameter $0 < \varepsilon \ll 1$. This parameter measures the deviation of the solution from the soliton $u_0(x)$. Substituting the suggested expansion into (5.1), we can develop a hierarchy of equations by grouping powers of ε. Below, calculations for u_t and u_{xx} are explicitly written to

assist in the visualization of grouping terms with respect to powers of ε:

$$u_t = i\theta_t e^{i\theta}\left[u_0 + \varepsilon u_1 + \varepsilon^2 u_2 + \ldots\right] + e^{i\theta}\left[\varepsilon u_{1t} + \varepsilon^2 u_{1T} + \varepsilon^2 u_{2t} + \varepsilon^3 u_{2T} + \ldots\right] \tag{5.12}$$

$$u_{xx} = e^{i\theta}\left[u_{0xx} + \varepsilon u_{1xx} + \varepsilon^2 u_{2xx} + \ldots\right] \tag{5.13}$$

$$\theta_t = \mu_0 + \varepsilon^2\mu_1(T) + \varepsilon^3\mu_2(T) + \ldots \tag{5.14}$$

Thus, the $\mathcal{O}(1)$ equation is

$$\begin{aligned} \mu_0 e^{i\theta}u_0 + e^{i\theta}u_{0xx} + 2e^{i\theta}|u_0|^2 u_0 &= 0 \\ \Rightarrow u_{0xx} + 2|u_0|^2 u_0 &= -\mu_0 u_0, \end{aligned} \tag{5.15}$$

which is automatically satisfied (recall (5.3)). At $\mathcal{O}(\varepsilon)$, we can develop an equation for u_1:

$$\begin{aligned} i(-i\mu_0 e^{i\theta}u_1 + e^{i\theta}u_{1t}) + e^{i\theta}u_{1xx} + 2e^{i\theta}(u_0^2\bar{u}_1 + 2|u_0|^2 u_1) &= 0 \\ iu_{1t} + u_{1xx} + \mu_0 u_1 + 4|u_0|^2 u_1 + 2u_0^2\bar{u}_1 &= 0 \\ (i\partial_t + \partial_{xx} + \mu_0 + 4u_0^2)u_1 + 2u_0^2\bar{u}_1 &= 0. \end{aligned} \tag{5.16}$$

We can take the complex conjugate of (5.16) to obtain a second equation. These terms are reminiscent of the real first row of $\mathcal{H}$ (without the i) so we evaluate $\mathcal{H}$ at $\mu = \mu_0$ to define

$$\mathcal{H}|_{\mu=\mu_0} = \mathcal{H}_0 \quad \text{and} \quad -i\mathcal{H}|_{\mu=\mu_0} = \mathcal{H}_0^{\text{real}}.$$

Then $\mathcal{H}_0^{\text{real}}$ can be used with (5.16) and its conjugate counterpart to summarize our information for u_1 as

$$(i\partial_t + \mathcal{H}_0^{\text{real}})\begin{bmatrix} u_1 \\ \bar{u}_1 \end{bmatrix} = 0. \tag{5.17}$$

Now, information about the spectrum of $\mathcal{H}_0$ can be used to find a solution for u_1.

If $i\omega$ is a simple eigenvalue of $\mathcal{H}_0$ with a single real eigenfunction $[\psi_1, \psi_2]^T$, then ω is a simple

eigenvalue of $\mathcal{H}_0^{\text{real}}$ with a single real eigenfunction $[\psi_1, \psi_2]^T$. Thus, the localized solution for u_1 is

$$u_1 = \frac{1}{2}\left(\psi_1(x)e^{i\omega t} + \psi_2(x)e^{-i\omega t}\right) \tag{5.18}$$

While one may joyously continue this process for increasing powers of ε, this first term is all that is necessary to create a sufficiently accurate initial condition for the relatively periodic solution. After modifying the equation to remove the $e^{i\mu t}$ factor, the resulting solution should be truly periodic in time. A method that could be then be used to search for time-periodic solutions in the resulting modified NLS equation is the adjoint continuation method of Ambrose-Wilkening described in [7, 8].

5.2.3 Hamiltonian-Hopf Bifurcations

While periodic solutions may be found near any purely complex eigenvalue, more interesting dynamics can be found near Hamiltonian-Hopf bifurcations. A **Hamiltonian-Hopf bifurcations** occurs in conservative wave systems, where pairs of imaginary eigenvalues in the linear-stability spectrum of stationary waves coalesce and move off of the imaginary axis, creating oscillatory instability. These linear instabilities have been well-analyzed, and it has been shown that only collisions of imaginary eigenvalues with opposite Krein signatures can induce such bifurcations by [51].

In order to discuss Krein signatures, first suppose that we have a linear Hamiltonian system that can be formulated as the spectral problem

$$J\mathcal{L}v = \lambda v, \tag{5.19}$$

where $\mathcal{L}$ is a unbounded, densely defined, self-adjoint operator in the space of square-integrable functions $L^2(\mathbb{R})$, and J is a bounded, skew-adjoint operator on $L^2(\mathbb{R})$. Hamiltonian symmetry allows us to assume that the operators $\mathcal{L}$ and J satisfy

$$J^2 = -I \tag{5.20}$$

and

$$J\mathcal{L} + \bar{\mathcal{L}}\bar{J} = 0. \tag{5.21}$$

If $\lambda_0 \in \mathbb{C}$ is a nonzero eigenvalue of (5.19) with eigenvector v_0 in the domain of $\mathcal{L}$, we define the **Krein quantity** $K(\lambda_0)$ by

$$K(\lambda_0) := \langle \mathcal{L}v_0, v_0 \rangle, \tag{5.22}$$

where $\langle \cdot, \cdot \rangle$ is the standard inner product in $L^2(\mathbb{R})$. The **Krein signature** is defined as the sign of the Krein quantity $K(\lambda_0)$ for a simple eigenvalue $\lambda_0 \in i\mathbb{R}\backslash\{0\}$. The values of λ_0 are restricted to purely complex numbers because it is **spectrally stable** when $\mathrm{Re}(\lambda_0) = 0$. If $\mathrm{Re}(\lambda_0) \neq 0$, then we say it is **spectrally unstable**.

If the parameters of the stationary NLS equation change, parameters of the spectral problem (5.19) also change. However, a simple eigenvalue $\lambda_0 \in i\mathbb{R}$ will remain on the $Re(\lambda) = 0$ axis unless it collides with another eigenvalue or a part of the continuous spectrum, due to the preservation of its multiplicity and the Hamiltonian symmetry of eigenvalues. In this case, the eigenvalue λ_0 and its Krein quantity $K(\lambda_0)$ are, at least, continuous functions of parameters of the NLS equation.

Hamiltonian symmetry also makes it easy to see that the eigenvalues of the spectral problem will occur in pairs or quadruplets. There is symmetry about the imaginary axis since $\sigma_3\mathcal{L} = \bar{\mathcal{L}}\sigma_3$ where $\sigma_3 = \mathrm{diag}(1, -1)$ (by (5.21)), and symmetry about the real axis since $\sigma_1\mathcal{L} = \bar{\mathcal{L}}\sigma_1$ where $\sigma_1 = \mathrm{antidiag}(1, 1)$. Thus, if λ is an eigenvalue of $\mathcal{L}$, then $\bar{\lambda}$ is an eigenvalue by real axis symmetry and $-\bar{\lambda}$ is as well by imaginary axis symmetry. Applying real axis symmetry to the already found eigenvalue $-\bar{\lambda}$, we get $-\lambda$. Thus a single eigenvalue λ guarantees the existence of $-\lambda$, $\bar{\lambda}$ and $-\bar{\lambda}$.

This causes purely real or purely imaginary eigenvalues to appear in $\pm\lambda$ pairs. Thus, if λ is spectrally stable (that is $\mathrm{Re}(\lambda) = 0$) then we have a spectrally stable pair $\pm\lambda$ on the imaginary axis. If λ is spectrally unstable, then it will appear as a pair on the real axis or quadruplets in the complex plane.

If two spectrally stable eigenvalues of the same Krein signature move towards each other in the parameter continuation of the spectral problem (5.19), then their coalescence will not result in the onset of instability. However, if the two spectrally stable eigenvalues have opposite Krien signatures, their coalescence is expected to result in the onset of instability, subject to technical non-degeneracy constraints discussed in [26, 51]. This fact will allow us to define conditions under which Hamiltonian-Hopf bifurcations occur.

From §5.2.1, the operators $\mathcal{L}$ and J are self-adjoint and skew-symmetric, respectively. One can

quickly verify that $J^2 = -I$ and $J\mathcal{L} + \bar{\mathcal{L}}\bar{J} = 0$; thus, we have met the necessary assumptions to discuss the Krein signature. (Same properties hold for $\mathcal{L}^*$ and J^*.)

For each spectrally stable, simple eigenvalue $\lambda_0 = i\omega$ of the spectral problem (5.5) with eigenvector $v_0 = (f_1, f_2)^T$, the Krein quantity can be explicitly written as follows:

$$K(\lambda_0) = \langle \mathcal{L}v_0, v_0 \rangle = -i\lambda_0 \langle \sigma_3 v_0, v_0 \rangle = \omega \int_{\mathbb{R}} \left[|f_1(x)|^2 - |f_2(x)|^2 \right] \, dx. \tag{5.23}$$

If λ_0 is nonzero, then the Krein signature is given by

$$K_{\lambda_0} = \operatorname{sgn} \left(\omega \int_{\mathbb{R}} \left[|f_1(x)|^2 - |f_2(x)|^2 \right] \, dx \right). \tag{5.24}$$

Similar to what was done in §5.2.2, the normal form near a HHB point (as derived in [80]) can be used to derive the initial condition that would produce a relatively-periodic solution. This initial condition could then be used in the following instead of the one that we proposed above.

The numerical method that would locate a HHB has yet to be implemented. Adding this functionality to the current bifurcation tools would provide us with the ability to search for solutions near HHB's and would fit into the existing framework easily since our linearization operators meet the listed criteria.

APPENDIX A

QGLAB EXAMPLES

A.1 Poisson Solver

The Poisson equation can be a great problem to benchmark a solver's accuracy since an analytical solution can be found relatively easily in some cases. For example, we will work on the lollipop graph to solve the vertex-value problem

$$\begin{cases} \Delta u = f, \\ u_1(0) = 0, \\ u_1'(\ell_1) - u_2'(0) + u_2'(\ell_2) + u_2(0) = 1, \end{cases}$$

where $f = \{-\pi^2 \cos(\pi x), -4\cos(2x)\}$ on edge 1 and 2, and the edges have the respective lengths 2 and π. The exact solution on the edges is then $u_{\text{exact}} = \{\cos(\pi x), \cos(2x)\}$. The structure of QGLAB makes it possible to solve this problem with just a few lines of code:

```
Phi = quantumGraphFromTemplate('multibell','nbells',1,'robinCoeff',...
      [0  1],'discretization','Chebyshev','nX',32,'nodeData',[0;  1]);
f1 = @(x) -pi^2*cos(pi*x);
f2 = @(x) -4*cos(2*x);
edgeData = Phi.applyFunctionsToAllEdges({f1,f2});
numericalSolution = Phi.solvePoisson('edgeData',edgeData);
Phi.plot(numericalSolution)
```

This yields Figure A.1. When the numerical solution is compared to the analytical one, the relative error is found on the order of 10^{-04}.

The function `quantumGraphFromTemplate` calls on a library of predefined graphs which include the lollipop (through the use of `multibell`), star, and dumbbell graphs, among numerous other examples. The plot coordinates for all graph templates are already defined which streamlines the production of visual aids. The command `quantumGraph` can be used to define graphs by specifying the source and target vertices beyond the predefined templates. The command `Phi.solvePoisson` calls on `Phi.laplacianMatrix` (which constructs the Laplacian utilizing methods described in §3.1)

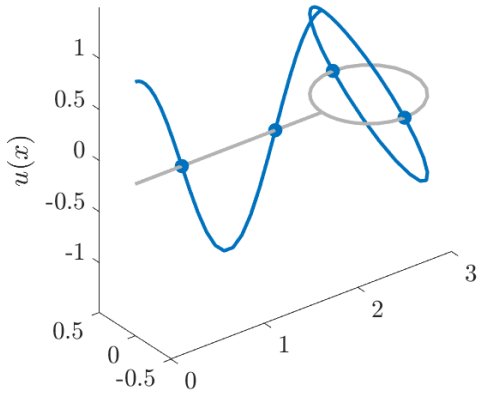


Figure A.1: Solution to nonhomogeneous Poisson problem

and solves the appropriate matrix equation using the information from `nodeData`.

A.2 Bifurcation Diagrams

Bifurcations are a fundamental tool in nonlinear partial differential equations that find solutions by branching off of the linear portion of the operator as described in §3.4. To do this with QGLAB , the quantum graph must first be constructed so a few eigenvalues and eigenfunctions can be found for the graph Laplacian.

```
tag = 'star';
Phi = quantumGraphFromTemplate(tag,'n',3,'LVec',[pi/2 1 1],...
      'discretization','Chebyshev','nx',32);

nToPlot = 4;
nDoubles = 2;
nTriples = 0;
diagramNumber=eigenfunctionsSaveData(Phi,tag,nToPlot,nDoubles,nTriples);
```

In this case, we will branch off of the first eigenvalue using the function `continueFromEig` which continues a branch from the linear limit of the chosen eigenfunction. A number of options need to be set for `continueFromEig` (and other continuation functions in general), and those are set using the function `continuerSet`.

```
opts = continuerSet([], 'LambdaThresh',-2,'NThresh',4,'plotFlag',false);
eigNumber = 1;
[bn1, bl1] = continueFromEig(tag, diagramNumber, eigNumber, opts);
bifurcationDiagram(tag, diagramNumber)
```

The labels `bn` and `bl` are chosen to stand for bifurcation number and bifurcation location, respectively. The last line produces the bifurcation diagram in Figure A.2 where it can be observed that there are two bifurcation points which are stored in `bl(1)` and `bl(2)`.

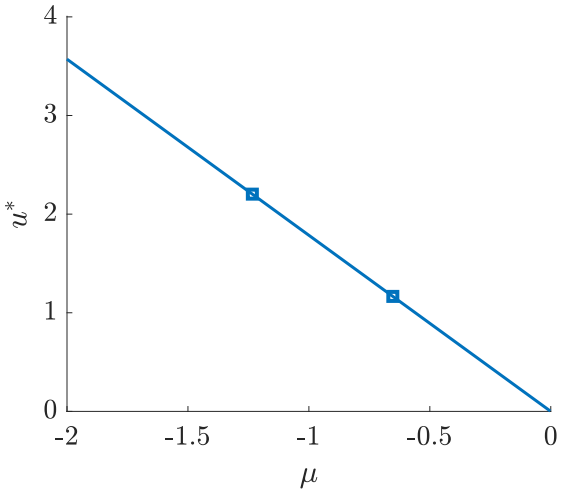


Figure A.2: Branch 1

The function `continueFromBranchPoint` branches off of the specified bifurcation point. To branch off of the first bifurcation point on branch 1, we call the desired branch `bn1` and desired branch point `bl1(1)`.

```
[bn2, bl2]=continueFromBranchPoint(tag, diagramNumber, bn1, bl1(1), 1, opts);
bifurcationDiagram(tag, diagramNumber)
```

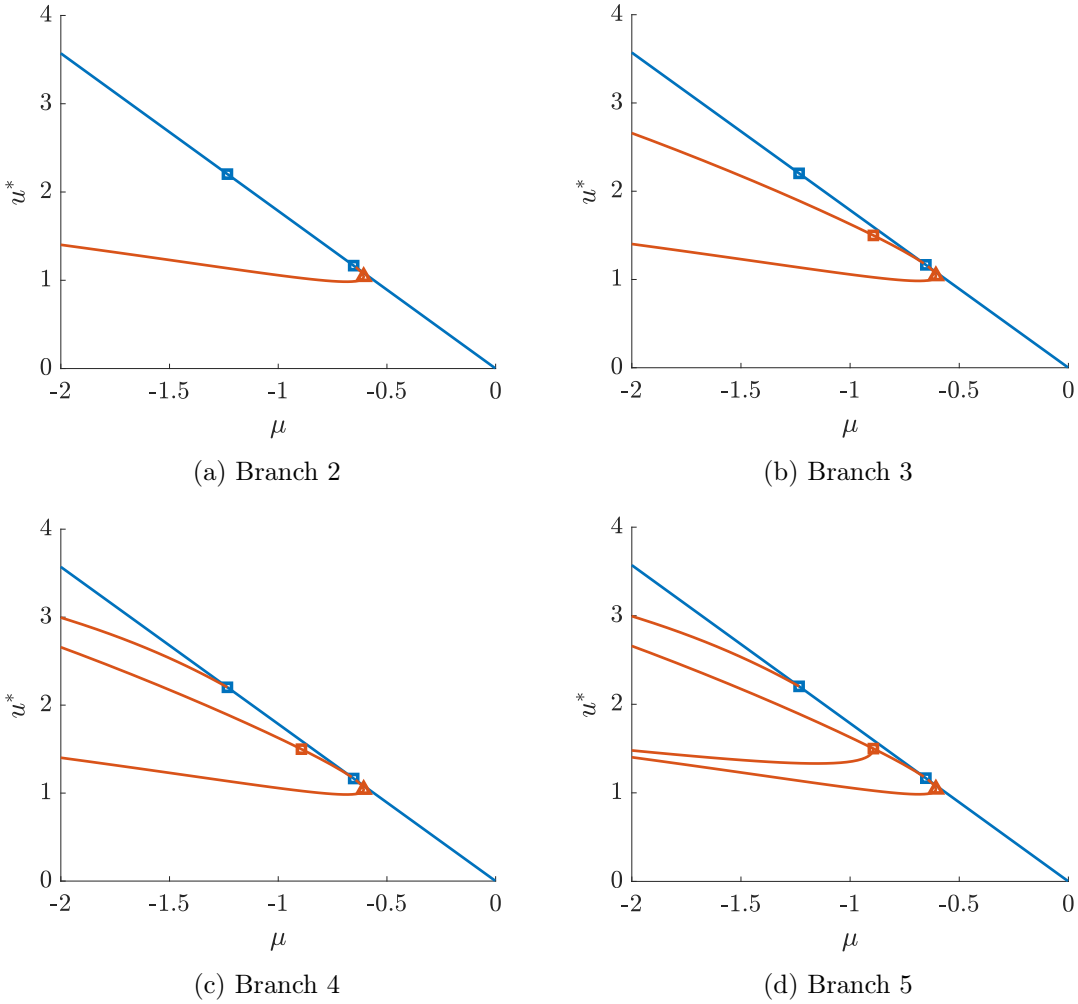
This generates Figure A.3a, where the bifurcation point is now labeled with a triangle which indicates that this is a saddle-node bifurcation point. To obtain the other branch produced by the same bifurcation point, we indicate that we would like to branch off of the same point but in the opposite direction in the following way:

```
[bn3,bl3]=continueFromBranchPoint(tag,diagramNumber,bn1,bl1(1),-1,opts);
bifurcationDiagram(tag,diagramNumber)
```

This generates Figure A.3b where a new bifurcation point is introduced on the most recently produced branch. Now, there are two bifurcation points left that can be branched off of.

The bifurcation point on the first branch is stored in `bl1(2)` and can be bifurcated off of using `continueFromBranchPoint` in the following way to generate Figure A.3c.

```
[bn4,bl4]=continueFromBranchPoint(tag,diagramNumber,bn1,bl1(2),1,opts);
bifurcationDiagram(tag,diagramNumber)
```



(a) Branch 2

(b) Branch 3

(c) Branch 4

(d) Branch 5

Figure A.3: Star bifurcation diagram branch progression

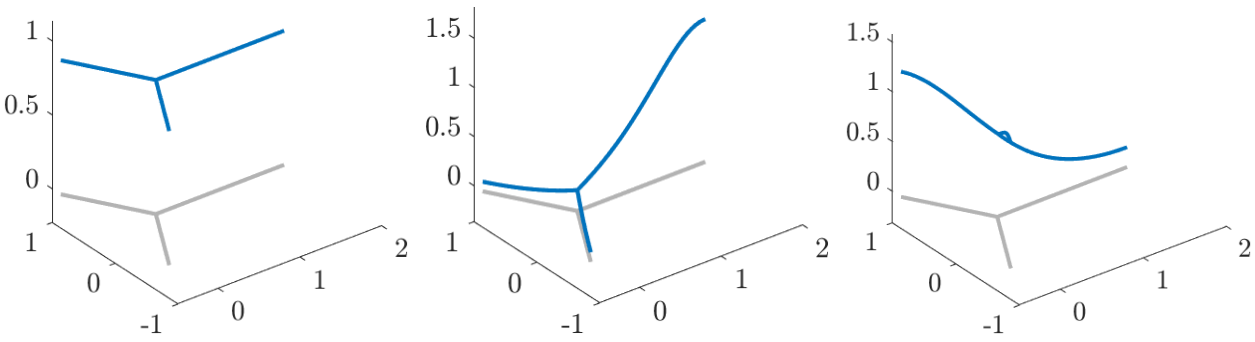
The bifurcation point on the third branch is stored in `bl3(1)`, and we can use

```
[bn5,bl5]=continueFromBranchPoint(tag,diagramNumber,bn3,bl3(1),1,opts);
bifurcationDiagram(tag,diagramNumber)
```

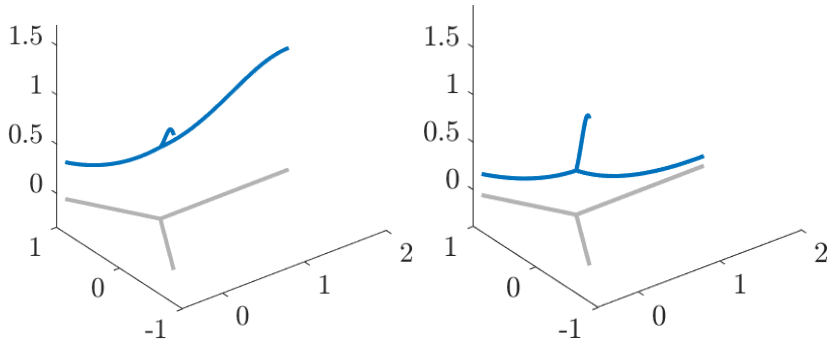
to generate our final bifurcation diagram, Figure A.3d.

The function `plotSolution` allows a user to easily obtain visuals of the solutions generated by each branch. One can specify which solution they would like a visual of, or pick the last solution generated on a branch. Below plots whichever solution is closest to $\mu = -2$, resulting in Figure A.4.

```
plotSolution(tag,diagramNumber,bn1,19)
plotSolution(tag,diagramNumber,bn2,'last')
plotSolution(tag,diagramNumber,bn3,'last')
plotSolution(tag,diagramNumber,bn4,'last')
plotSolution(tag,diagramNumber,bn5,'last')
```



(a) Branch 1 Solution (b) Branch 2 Solution (c) Branch 3 Solution



(d) Branch 4 Solution (e) Branch 5 Solution

Figure A.4: Solutions to the NLS on the star graph

A.3 Time Evolution

The functions `timeEvolve15s` and `timeEvolveARK` are the implemented time evolution schemes that were described in §4.2 and §4.3. They both evolve a solution through time given a function, initial condition, and end time; however, `timeEvolveARK` also requires that the step size be specified. While `timeEvolve15s` utilizes MATLAB's built-in command `ode15s` (a backward differentiation formula), `timeEvolveARK` utilizes the following butcher tableau designed for stiff problems:

x	x	0	0
$\frac{1}{2}$	$\frac{1}{2}-x$	x	0
$1-x$	$2x$	$1-4x$	x
	$\frac{1}{6(1-2x)^2}$	$\frac{3(1-2x)^2-1}{6(1-2x)^2}$	$\frac{1}{6(1-2x)^2}$

Table A.1: Butcher tableau for implemented ARK method

where x was chosen to be $x = 1.066$, the root of the cubic equation $x^3 - \frac{3x^2}{2} + \frac{x}{2} - \frac{1}{24} = 0$ which makes the method most stable.

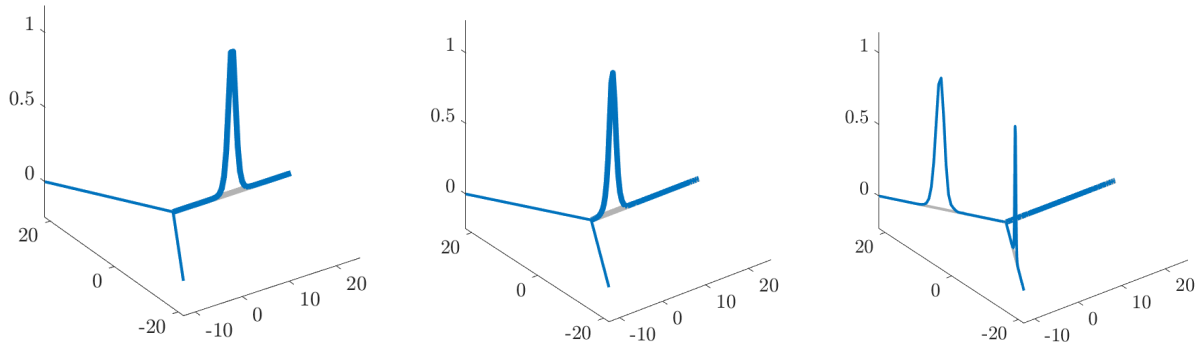
The function `timeEvolve15s` is used for the following example which demonstrates how to set up and solve the NLS on a star graph:

```
Phi = quantumGraphFromTemplate('star','LVec',8*pi,'discretization',...
    'Chebyshev','nX',80,'robinCoeff',[0 NaN NaN NaN],'weight',[2 1 1]);
M = Phi.laplacianMatrix;
B = Phi.weightMatrix;
f = @(t,z) -1i*( M*z + (2*(B*z).^2.*conj(B*z)));
y0 = Phi.applyFunctionsToAllEdges({@(x)exp(-1i*x*(-4)).*sech(x-4*pi)...
      ,0,0});
tend = 20;
[t,y] = Phi.timeEvolve15s(f,u0,tend);
```

This results in a solution where the soliton starts on edge 1, propagates towards the junction, then splits on edges 2 and 3 as illustrated in Figure A.5. The accuracy is tested by checking for conservation of energy, mass, and momentum relative to `y0`. The definitions for energy and mass are still the same as (4.6) and **momentum** is defined by

$$P(u) = \sum_{j=1}^{|\mathcal{E}|} \omega_j \int_{e_j} |u_j^*(x) u_j'(x)|^2 \, dx. \tag{A.1}$$

The accuracy results are summarized in Figure A.6.



(a) $t = 0$ (b) $t = 0.95$ (c) $t = 3.21$

Figure A.5: Propagation of a soliton towards center node with conservation of momentum

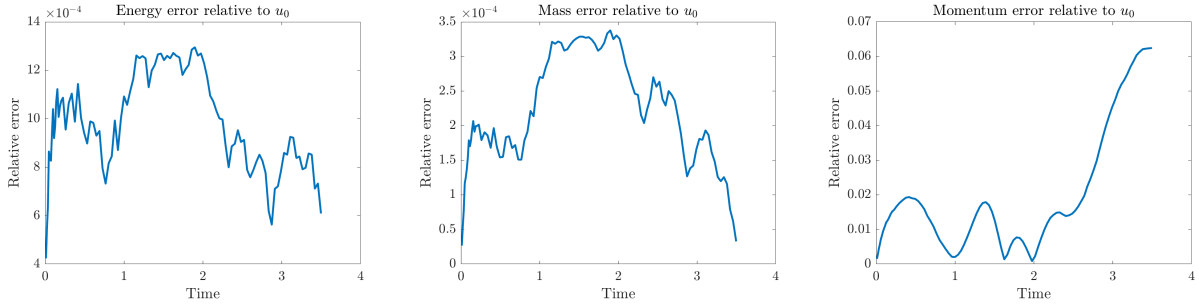
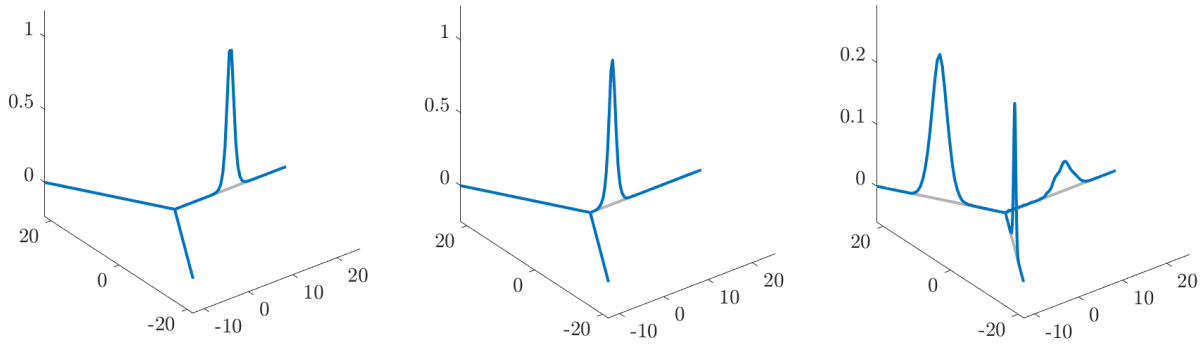


Figure A.6: Conservation of energy, mass, and momentum results

Conservation of momentum was enforced in this example by setting the the quantum graph property `weight` to `[2 1 1]`. By default, it would be `[1 1 1]`, which would not conserve momentum. This is demonstrated in Figure A.7 which illustrates that a ripple is produced going backwards along the first edge when the wave propagates through the center node.



(a) $t = 0$ (b) $t = 0.95$ (c) $t = 3.21$

Figure A.7: Propagation of a soliton towards center node without conservation of momentum.

While energy and mass are conserved in this example, momentum is not, as is summarized in Figure A.8. Notice how in the rightmost figure, there is significant jump in the relative momentum about when `t≈1.3`; this corresponds to the point in time where the wave is propagating through the junction.

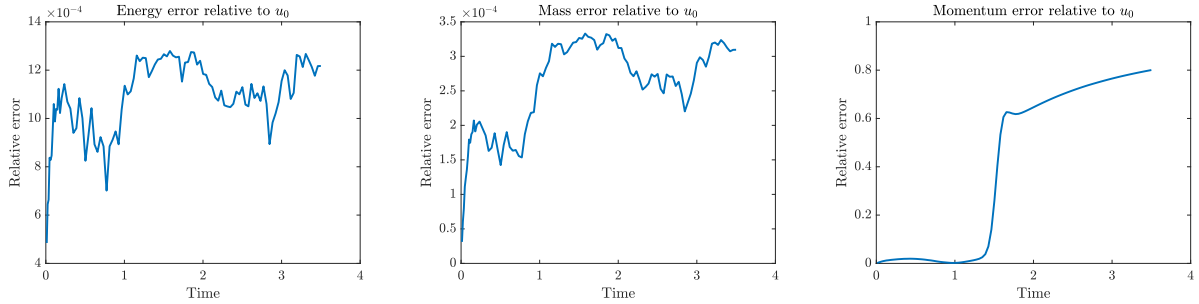


Figure A.8: Conservation of energy, mass, and momentum results.

REFERENCES

[1] Luigi Accardi, Anis Ben Ghorbal, and Nobuaki Obata, *Monotone independence, comb graphs and Bose–Einstein condensation*, Infinite Dimensional Analysis, Quantum Probability and Related Topics **7** (2004), no. 03, 419–435.

[2] Riccardo Adami, *Ground states for NLS on graphs: a subtle interplay of metric and topology*, Mathematical Modelling of Natural Phenomena **11** (2016), no. 2, 20–35.

[3] Riccardo Adami, Claudio Cacciapuoti, Domenico Finco, and Diego Noja, *Variational properties and orbital stability of standing waves for NLS equation on a star graph*, Journal of Differential Equations **257** (2014), no. 10, 3738–3777.

[4] Riccardo Adami, Enrico Serra, and Paolo Tilli, *NLS ground states on graphs*, Calculus of Variations and Partial Differential Equations **54** (2015), no. 1, 743–761.

[5] ———, *Threshold phenomena and existence results for NLS ground states on metric graphs*, Journal of Functional Analysis **271** (2016), no. 1, 201–223.

[6] Sergio Albeverio, Claudio Cacciapuoti, and Domenico Finco, *Coupling in the singular limit of thin quantum waveguides*, Journal of Mathematical Physics **48** (2007), no. 3, 032103.

[7] David M. Ambrose and Jon Wilkening, *Global paths of time-periodic solutions of the Benjamin–Ono equation connecting pairs of traveling waves*, Communications in Applied Mathematics and Computational Science **4** (2009), no. 1, 177–215.

[8] ———, *Computation of time-periodic solutions of the Benjamin–Ono equation*, Journal of Nonlinear Science **20** (2010), no. 3, 277–308.

[9] Claudio Amovilli, Frederik E. Leys, and Norman H. March, *Electronic energy spectrum of two-dimensional solids and a chain of C atoms from a quantum network model*, Journal of Mathematical Chemistry **36** (2004), no. 2, 93–112.

[10] Pascal Auscher, Thierry Coulhon, and Alexander Grigoryan, *Heat kernels and analysis on manifolds, graphs, and metric spaces*, vol. 338, American Mathematical Soc., 2003.

[11] Yshai Avishai and Jean-Marc Luck, *Quantum percolation and ballistic conductance on a lattice of wires*, Physical Review B **45** (1992), no. 3, 1074.

[12] Alvin Bayliss, Andreas Class, and Bernard J. Matkowsky, *Roundoff error in computing derivatives using the chebyshev differentiation matrix*, Journal of Computational Physics **116** (1995), no. 2, 380–383.

[13] Thomas Beck, Isabel Bors, Grace Conte, Graham Cox, and Jeremy L. Marzuola, *Limiting eigenfunctions of Sturm-Liouville operators subject to a spectral flow*, Annales Mathématiques du Québec **45** (2021), no. 2, 249–269.

[14] Mikhal I. Belishev, *Boundary controllability of a dynamical system governed by the wave equation on a class of graphs (trees)*, Journal of Mathematical Sciences **132** (2006), no. 1, 11–25.

[15] Gregory Berkolaiko, *An elementary introduction to quantum graphs*, Geometric and Computational Spectral theory **700** (2017), 41–72.

[16] Gregory Berkolaiko, Graham Cox, and Jeremy L. Marzuola, *Nodal deficiency, spectral flow, and the Dirichlet-to-Neumann map*, Letters in Mathematical Physics **109** (2019), no. 7, 1611–1623.

[17] Gregory Berkolaiko and Peter Kuchment, *Introduction to quantum graphs*, no. 186, American Mathematical Soc., 2013.

[18] Gregory Berkolaiko, Jeremy L. Marzuola, and Dmitry E. Pelinovsky, *Edge-localized states on quantum graphs in the limit of large mass*, Annales de l'Institut Henri Poincaré C, Analyse non linéaire **38** (2021), no. 5, 1295–1335.

[19] Jean-Paul Berrut and Lloyd N. Trefethen, *Barycentric Lagrange interpolation*, SIAM Review **46** (2004), no. 3, 501–517.

[20] Ginestra Bianconi and Albert-László Barabási, *Bose–Einstein condensation in complex networks*, Physical Review Letters **86** (2001), no. 24, 5632.

[21] Evgeny Bulgakov and Almas Sadreev, *Symmetry breaking in a T-shaped photonic waveguide coupled with two identical nonlinear cavities*, Physical Review B **84** (2011), no. 15, 155304.

[22] Raffaella Burioni, Davide Cassi, and Alessandro Vezzani, *Topology, hidden spectra and Bose-Einstein condensation on low-dimensional complex networks*, Journal of Physics A: Mathematical and General **35** (2002), no. 5, 1245.

[23] Markus Büttiker, *Absence of backscattering in the quantum Hall effect in multiprobe conductors*, Physical Review B **38** (1988), no. 14, 9375.

[24] Claudio Canuto, *Spectral methods and a maximum principle*, Mathematics of Computation **51** (1988), no. 184, 615–629.

[25] Thierry Cazenave, *Semilinear Schrödinger equations*, vol. 10, American Mathematical Soc., 2003.

[26] Alexander Chernyavsky, Panayotis G. Kevrekidis, and Dmitry E. Pelinovsky, *Krein signature in Hamiltonian and $\mathbb{PT}$-symmetric systems*, Parity-time Symmetry and Its Applications, Springer, 2018, pp. 465–491.

[27] René Dáger and Enrique Zuazua, *Wave propagation, observation and control in 1-d flexible multi-structures*, vol. 50, Springer Science & Business Media, 2006.

[28] Tobin A Driscoll and Nicholas Hale, *Rectangular spectral collocation*, IMA Journal of Numerical Analysis **36** (2016), no. 1, 108–132.

[29] Lennart Edsberg, *Introduction to computation and modeling for differential equations*, John Wiley & Sons, 2015.

[30] Pavel Exner, *Lattice kronig-penney models*, Physical Review Letters **74** (1995), no. 18, 3503.

[31] Pavel Exner and Olaf Post, *Approximation of quantum graph vertex couplings by scaled Schrödinger operators on thin branched manifolds*, Journal of Physics A: Mathematical and Theoretical **42** (2009), no. 41, 415305.

[32] Grégory Faye, *An introduction to bifurcation theory*, NeuroMathComp Laboratory, INRIA, Sophia Antipolis, CNRS, ENS Paris, France (2011).

[33] Francesco Fidaleo, Daniele Guido, and Tommaso Isola, *Bose–Einstein condensation on inhomogeneous amenable graphs*, Infinite Dimensional Analysis, Quantum Probability and Related Topics **14** (2011), no. 02, 149–197.

[34] Stephen A. Fulling, *Systematics of the relationship between vacuum energy calculations and heat-kernel coefficients*, Journal of Physics A: Mathematical and General **36** (2003), no. 24, 6857.

[35] Stephen A. Fulling and Justin H. Wilson, *Vacuum energy and closed orbits in quantum graphs*, Program on Analysis on Graphs and Its Applications (Isaac Newton Institute, Cambridge, 2007), Editor P. Kuchment, to appear (2008).

[36] Sven Gnutzmann, Uzy Smilansky, and Stanislav Derevyanko, *Stationary scattering from a nonlinear network*, Physical review A **83** (2011), no. 3, 033831.

[37] Nathan Goldman and Pierre Gaspard, *Quantum graphs and the integer quantum Hall effect*, Physical Review B **77** (2008), no. 2, 024302.

[38] Roy H. Goodman, *Bifurcations of relative periodic orbits in NLS/GP with a triple-well potential*, Physica D: Nonlinear Phenomena **359** (2017), 39–59.

[39] ———, *NLS bifurcations on the bowtie combinatorial graph and the dumbbell metric graph*, Disc. Cont. Dyn. Sys. A **39** (2019), 2203–2232.

[40] Roy H. Goodman, Grace Conte, and Jeremy L. Marzuola, QGlab *: A* Matlab *package for computations on quantum graphs*, (preprint).

[41] Daniel Grieser, *Thin tubes in mathematical physics, global analysis and spectral geometry*, arXiv preprint arXiv:0802.2687 (2008).

[42] Allan Griffin, David W Snoke, and Sandro Stringari, *Bose–Einstein condensation*, Cambridge University Press, 1996.

[43] Alexander Grigor'yan, *Heat kernels on manifolds, graphs and fractals*, European Congress of Mathematics, Springer, 2001, pp. 393–406.

[44] Yusuke Higuchi and Tomoyuki Shirai, *The spectrum of magnetic Schrödinger operators on a graph with periodic structure*, Journal of Functional Analysis **169** (1999), no. 2, 456–480.

[45] Peter D Hislop and Olaf Post, *Anderson localization for radial tree-like random quantum graphs*, Waves in Random and Complex Media **19** (2009), no. 2, 216–261.

[46] Norman E Hurt, *Mathematical physics of quantum wires and devices: From spectral resonances to Anderson localization*, vol. 506, Springer Science & Business Media, 2000.

[47] Adilbek Kairzhan, Robert Marangell, Dmitry E. Pelinovsky, and Ke Liang Xiao, *Standing waves on a flower graph*, Journal of Differential Equations **271** (2021), 719–763.

[48] Adilbek Kairzhan, Diego Noja, and Dmitry E. Pelinovsky, *Standing waves on quantum graphs*, arXiv preprint arXiv:2201.08114 (2022).

[49] Adilbek Kairzhan and Dmitry E. Pelinovsky, *Nonlinear instability of half-solitons on star graphs*, Journal of Differential Equations **264** (2018), no. 12, 7357–7383.

[50] Adilbek Kairzhan, Dmitry E. Pelinovsky, and Roy H. Goodman, *Drift of spectrally stable shifted states on star graphs*, SIAM Journal on Applied Dynamical Systems **18** (2019), no. 4, 1723–1755.

[51] Todd Kapitula and Keith Promislow, *Spectral and dynamical stability of nonlinear waves*, vol. 457, Springer, 2013.

[52] Abel Klein, *Absolutely continuous spectrum in the Anderson model on the Bethe lattice*, Mathematical Research Letters **1** (1994), no. 4, 399–407.

[53] ———, *Extended states in the Anderson model on the Bethe lattice*, Advances in Mathematics **133** (1998), no. 1, 163–184.

[54] Frédéric Klopp and Konstantin Pankrashkin, *Localization on quantum graphs with random vertex couplings*, Journal of Statistical Physics **131** (2008), no. 4, 651–673.

[55] ———, *Localization on quantum graphs with random edge lengths*, Letters in Mathematical Physics **87** (2009), no. 1, 99–114.

[56] Evgeny Korotyaev and Igor Lobanov, *Zigzag periodic nanotube in magnetic field*, arXiv preprint math/0604007 (2006).

[57] ———, *Schrödinger operators on zigzag nanotubes*, Annales de l'Institut Henri Poincaré **8** (2007), no. 6, 1151–1176.

[58] Evgeny Korotyaev and Natalia Saburova, *Magnetic Schrödinger operators on periodic discrete graphs*, Journal of Functional Analysis **272** (2017), no. 4, 1625–1660.

[59] Peter Kuchment and Leonid Kunyansky, *Differential operators on graphs and photonic crystals*, Advances in Computational Mathematics **16** (2002), no. 2, 263–290.

[60] Peter Kuchment and Olaf Post, *On the spectra of carbon nano-structures*, Communications in Mathematical Physics **275** (2007), no. 3, 805–826.

[61] Peter Kuchment and Hongbiao Zeng, *Convergence of spectra of mesoscopic systems collapsing onto a graph*, Journal of Mathematical Analysis and Applications **258** (2001), no. 2, 671–700.

[62] Jeremy L. Marzuola and Dmitry E. Pelinovsky, *Ground state on the dumbbell graph*, Applied Mathematics Research eXpress **2016** (2016), no. 1, 98–145.

[63] Jeremy L. Marzuola and Michael I. Weinstein, *Long time dynamics near the symmetry breaking bifurcation for nonlinear Schrödinger/gross-pitaevskii equations*, American Institute of Mathematical Sciences **28** (2010), no. 4, 1505–1554.

[64] Taku Matsui, *BEC of free bosons on networks*, Infinite Dimensional Analysis, Quantum Probability and Related Topics **9** (2006), no. 01, 1–26.

[65] Stanislav Molchanov and Boris Vainberg, *Transition from a network of thin fibers to the quantum*, Contemp. Math. **415** (2006), 227.

[66] Diego Noja and Dmitry E. Pelinovsky, *Standing waves of the quintic NLS equation on the tadpole graph*, Calculus of Variations and Partial Differential Equations **59** (2020), no. 5, 1–31.

[67] Konstantin Pankrashkin, *Localization effects in a periodic quantum graph with magnetic field and spin-orbit interaction*, Journal of Mathematical Physics **47** (2006), no. 11, 112105.

[68] Dmitry E. Pelinovsky, *Localization in periodic potentials: from Schrödinger operators to the gross–pitaevskii equation*, vol. 390, Cambridge University Press, 2011.

[69] Dmitry E. Pelinovsky and Guido Schneider, *Bifurcations of standing localized waves on periodic graphs*, Annales Henri Poincaré **18** (2017), no. 4, 1185–1211.

[70] Jacob Rubinstein and Michelle Schatzman, *Spectral and variational problems on multiconnected strips*, Comptes Rendus de l'Académie des Sciences-Series I-Mathematics **325** (1997), no. 4, 377–382.

[71] ———, *Variational problems on multiply connected thin strips i: Basic estimates and convergence of the Laplacian spectrum*, Archive for Rational Mechanics and Analysis **160** (2001), no. 4, 271–308.

[72] Klaus Ruedenberg and Charles W. Scherr, *Free-electron network model for conjugated systems. i. theory*, The Journal of Chemical Physics **21** (1953), no. 9, 1565–1581.

[73] Catherine Sulem and Pierre-Louis Sulem, *The nonlinear Schrödinger equation: self-focusing and wave collapse*, vol. 139, Springer Science & Business Media, 2007.

[74] Tao Tang and Manfred R. Trummer, *Boundary layer resolving pseudospectral methods for singular perturbation problems*, SIAM Journal on Scientific Computing **17** (1996), no. 2, 430–438.

[75] Akiyuki Tokuno, Masaki Oshikawa, and Eugene Demler, *Dynamics of one-dimensional bose liquids: Andreev-like reflection at y junctions and the absence of the aharonov-bohm effect*, Physical Review Letters **100** (2008), no. 14, 140402.

[76] E.J.G.G. Vidal, R.P.A. Lima, and M.L. Lyra, *Bose–Einstein condensation in the infinitely ramified star and wheel graphs*, Physical Review E **83** (2011), no. 6, 061137.

[77] Kuan Xu and Nicholas Hale, *Explicit construction of rectangular differentiation matrices*, IMA Journal of Numerical Analysis **36** (2016), no. 2, 618–632.

[78] Xindong Xu and Dongfeng Zhang, *Full dimension tori of Schrödinger equation*, Journal of Mathematical Physics **57** (2016), no. 11, 112702.

[79] Jianke Yang, *Classification of solitary wave bifurcations in generalized nonlinear Schrödinger equations*, Studies in Applied Mathematics **129** (2012), no. 2, 133–162.

[80] ———, *A normal form for Hamiltonian-Hopf bifurcations in nonlinear Schrödinger equations with general external potentials*, SIAM Journal on Applied Mathematics **76** (2016), no. 2, 598–617.