

# Numerically computing zeros of the Evans Function

Rebekah Coggin \*  
Department of Mathematics and Statistics  
Calvin College  
Grand Rapids, MI 49546

January 26, 2015

**Abstract.** This paper presents a method of numerically computing zeros of an analytic function for the specific application of computing eigenvalues of the Sturm-Liouville problem. The Sturm-Liouville problem is an infinite dimensional eigenvalue problem that often arises in solving partial differential equations, including the heat and wave equations. To compute eigenvalues of the Sturm-Liouville problem, we construct the Evans function, whose zeros correspond to eigenvalues of the Sturm-Liouville problem. Our method requires defining a contour integral based on an rough approximation of the zero. To apply this method to find zeros of the Evans function, we make rough approximates of zeros by a finite difference calculation for eigenvalues of the Sturm-Liouville problem. For cases where the exact zeros are known, we do a comparison to find that the numerical method in this paper has an error as small as  $\mathcal{O}(10^{-16})$ .

**Keywords.** Evans function, Sturm-Liouville problem, Cauchy integral formula

## CONTENTS

1. Introduction	66
2. Finding zeros of analytic functions	66
3. Numerical computation for simple zeros	70
3.1. Example of precision	71
4. Application: The Evans Function	73
4.1. Example: Evans function construction	77
5. Numerical implementation to solve Sturm-Liouville problem	78
5.1. Finite difference eigenvalue approximation	79
5.2. Computing zeros of the Evans function	80
5.3. Accuracy of Cauchy integral method of finding zeros	82
5.4. Further improvement to accuracy	83
References	83

---

\*E-mail: [rebekah.coggin@gmail.com](mailto:rebekah.coggin@gmail.com)

## 1. INTRODUCTION

Consider the general Sturm-Liouville problem for  $a \leq x \leq b$  given by

$$-\frac{\partial}{\partial x} \left( p(x) \frac{\partial y}{\partial x} \right) + q(x)y = \lambda w(x)y \quad (1.1)$$

with boundary conditions of the form

$$b_i^1 y(a, \lambda) + b_i^2 \frac{\partial y}{\partial x}(a, \lambda) = 0, \quad b_f^1 y(b, \lambda) + b_f^2 \frac{\partial y}{\partial x}(b, \lambda) = 0. \quad (1.2)$$

On the constant coefficients  $b$ , the subscript  $i$  denotes initial conditions and the subscript  $f$  denotes final conditions. The Sturm-Liouville problem is an infinite dimensional eigenvalue problem where the goal in solving the problem is to find the eigenvalues  $\lambda$  for which there is a nontrivial solution. Each solution  $y(x, \lambda)$  associated with an eigenvalue  $\lambda$  is known as an eigenfunction. Together, these eigenvalues and eigenfunctions provide the information needed to construct the solution to the partial differential equation as an infinite series. The Sturm-Liouville problem is the eigenvalue problem that arises for both the heat equation and the wave equation [5, 7]. Specifically, the eigenvalues of the wave equation give information about frequency of oscillation, and the eigenvalues of the heat equation give information about decay rate.

In this paper, we will study the Evans function as a method of finding the eigenvalues of the Sturm-Liouville problem. The Evans function is analogous to the characteristic polynomial of a matrix. In the same way that the zeros of the characteristic polynomial precisely correspond to eigenvalues of the matrix, the zeros of the Evans function precisely correspond to eigenvalues which have a nontrivial solution to the Sturm-Liouville problem. We are interested then in finding zeros of the analytic Evans function.

The purpose of this paper is to present a method which numerically finds the zeros of an any general analytic function and show how we implement the method in MATLAB specifically to find zeros of the Evans function. Our technique uses Cauchy theory to construct an algebraic set of equations to solve for zeros of a general analytic function. This set of equations requires that the zeros we intend to find are enclosed within a simple, closed contour  $\gamma$ . The challenge in finding numerical zeros comes in defining and integrating over the contour integral  $\gamma$ . In order to define a contour  $\gamma$  which encloses zeros, our method requires knowing some rough approximation of the zeros, which we find using a finite difference scheme. We then use the trapezoidal rule for numerical integration. In solving the system by evaluating contour integrals over  $\gamma$ , we zoom in on zeros of the Evans function with much higher precision than the rough approximation.

This paper is organized in the following way. In Section 2, we use the Cauchy integral formula and the Cauchy integral theorem to derive a system of algebraic equations to solve for zeros of an analytic function enclosed by a contour. In Section 3, we show a method of numerically computing these zeros. In Section 4, we show the correspondence between eigenvalues of the Sturm-Liouville problem and zeros of the Evans function by explicitly constructing Evans function. Finally, in Section 5, we present the MATLAB program which implements the methods we developed in this paper to solve the Sturm-Liouville problem.

**Acknowledgments.** Our work was supported by the National Science Foundation under grant DMS-1108783.

## 2. FINDING ZEROS OF ANALYTIC FUNCTIONS

We will recall the following three basic facts regarding analytic functions. Let  $f$  be analytic on a simply connected domain  $D$ . Suppose that  $z_* \in D$  and  $\gamma$  is a positively oriented, simple, closed curve in  $D$  that encloses  $z_*$ . Then by the Cauchy integral formula,

$$f(z_*) = \frac{1}{2\pi i} \oint_{\gamma} \frac{f(z)}{z - z_*} dz. \quad (2.1)$$

By differentiating we also have

$$f'(z_*) = \frac{1}{2\pi i} \oint_{\gamma} \frac{f(z)}{(z - z_*)^2} dz. \quad (2.2)$$

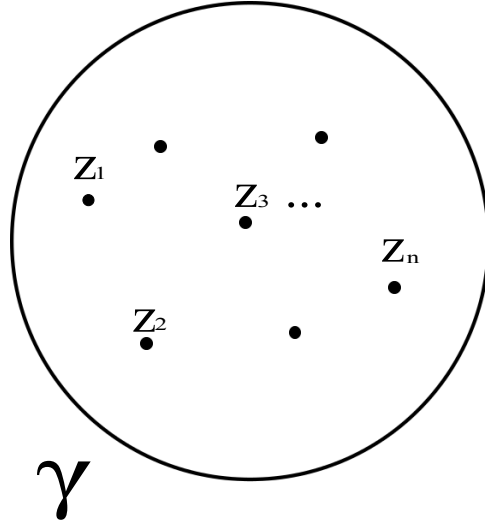
Furthermore, if we let  $\gamma$  be any simple, closed curve in  $D$ , then by the Cauchy integral theorem,

$$\oint_{\gamma} f(z)dz = 0 \tag{2.3}$$

[4, 8].

We will use the Cauchy integral formula and theorem to derive a set of equations to solve for zeros of an analytic function  $f(z)$  with  $n$  zeros at  $z_1, z_2, z_3, \dots, z_n$ . That is,  $f(z_j) = 0$  for all  $j = 1, 2, 3, \dots, n$ . Let  $\gamma$  be a simple closed contour which contains  $z_j$  for all  $j$ , as shown in Figure 1. To start building our set of equations, we construct an analytic function that is nonzero everywhere. In order to easily show that the new function has no zeros, we will assume that all zeros  $z_j$  are simple zeros. The construction could be modified for zeros of any multiplicity, but for the purpose of this paper, we will only consider simple zeros. With this assumption, we know the following is true:

$$f(z_j) = 0, \quad f'(z_j) \neq 0$$



**Figure 1:** We start with  $f(z)$  analytic with zeros  $z_1, z_2, z_3, \dots, z_n$  all enclosed by the contour  $\gamma$ .

The set of algebraic equations which we will now derive has been given elsewhere [1, 3]. We include the derivation here for the sake of completeness. We first write the Taylor expansion of  $f(z)$  about the point  $z_1$  and, since we know  $f(z_1) = 0$ , eliminate the first term. Doing so yields,

$$\begin{aligned} f(z) &= \sum_{n=0}^{\infty} \frac{f^{(n)}(z_1)}{n!} (z - z_1)^n \\ &= \sum_{n=1}^{\infty} \frac{f^{(n)}(z_1)}{n!} (z - z_1)^n \\ &= (z - z_1) \sum_{n=1}^{\infty} \frac{f^{(n)}(z_1)}{n!} (z - z_1)^{n-1} \\ &= (z - z_1) \sum_{n=0}^{\infty} \frac{f^{(n+1)}(z_1)}{(n+1)!} (z - z_1)^n. \end{aligned} \tag{2.4}$$

We will define

$$g_1(z) = \sum_{n=0}^{\infty} \frac{f^{(n+1)}(z_1)}{(n+1)!} (z - z_1)^n,$$

so that

$$f(z) = (z - z_1)g_1(z).$$

The function  $g_1(z)$  is analytic everywhere  $f(z)$  is analytic, and by construction it is nonzero at  $z = z_1$ ,  $g_1(z_1) = f'(z_1) \neq 0$ .

Now we repeat the same process to define a function  $g_2(z)$  which has the property that  $g_2(z_1), g_2(z_2) \neq 0$ . Since  $f(z_2) = 0$ , we can write

$$g_1(z_2) = \frac{f(z_2)}{(z_2 - z_1)} = 0.$$

By the product rule we differentiate  $f(z_2)$  to get

$$\begin{aligned} f'(z_2) &= g_1(z_2) + g_1'(z_2)(z_2 - z_1) \\ &= g_1'(z_2)(z_2 - z_1). \end{aligned}$$

Since  $f'(z_2) \neq 0$ , we can write

$$g_1'(z_2) = \frac{f'(z_2)}{(z_2 - z_1)} \neq 0.$$

Now that we have established both  $g_1(z_2) = 0$  and  $g_1'(z_2) \neq 0$ , we can write

$$g_1(z) = (z - z_2)g_2(z),$$

with

$$g_2(z_2) = g_1'(z_2) \neq 0.$$

In conclusion, we can rewrite  $f(z)$  in terms of  $g_2(z)$  to get

$$f(z) = (z - z_1)(z - z_2)g_2(z),$$

where  $g_2(z)$  is analytic everywhere  $f(z)$  is analytic and nonzero at  $z = z_1, z_2$ . The last statement follows from

$$0 \neq g_1(z_1) = (z_1 - z_2)g_2(z_1).$$

If we continue with this process for each of the  $n$  zeros, we have

$$f(z) = \left[ \prod_{j=1}^n (z - z_j) \right] g_n(z),$$

where  $g_n(z)$  is an analytic function such that  $g_n(z_j) \neq 0$  for all zeros  $z_j$  of  $f(z)$ . Upon removing the subscript we can thus define a nonzero analytic function  $g(z)$  as

$$g(z) = \frac{f(z)}{\prod_{j=1}^n (z - z_j)}.$$

We will use  $g(z)$  as an auxiliary function. We will manipulate  $g(z)$  into a helpful equation and then convert it back into terms of  $f(z)$ . In doing so, we will bring the zeros  $z_j$  back into the equation in such a way as to create an algebraic system for the zeros. To use  $g(z)$  in this way, we start by differentiating and dividing by itself. Since  $g(z)$  is nonzero and analytic, we have that  $g'(z)/g(z)$  is also an analytic function. Then by the Cauchy integral theorem (2.3), we have:

$$\frac{1}{2\pi i} \oint_{\gamma} \frac{g'(z)}{g(z)} dz = 0.$$

Additionally, multiplying by  $z^\ell$  for any integer  $\ell$  also gives an analytic function on which we can apply the Cauchy integral theorem, so

$$\frac{1}{2\pi i} \oint_{\gamma} z^\ell \frac{g'(z)}{g(z)} dz = 0, \quad \ell = 0, 1, 2, \dots \quad (2.5)$$

Now we would like to write Equation (2.5) in terms of  $f(z)$  and  $f'(z)$ . First, we first differentiate  $g(z)$ ,

$$\begin{aligned}
 g'(z) &= \frac{d}{dz} \frac{f(z)}{\prod_{j=1}^n (z - z_j)} \\
 &= \frac{f'(z) \prod_{j=1}^n (z - z_j) - f(z) \frac{d}{dz} \prod_{j=1}^n (z - z_j)}{\left(\prod_{j=1}^n (z - z_j)\right)^2} \\
 &= \frac{f'(z) \prod_{j=1}^n (z - z_j) - f(z) \sum_{\ell=1}^n \prod_{j=1, j \neq \ell}^n (z - z_j)}{\left(\prod_{j=1}^n (z - z_j)\right)^2} \\
 &= \frac{f'(z)}{\prod_{j=1}^n (z - z_j)} - \frac{f(z) \sum_{\ell=1}^n \prod_{j=1, j \neq \ell}^n (z - z_j)}{\left(\prod_{j=1}^n (z - z_j)\right)^2} \\
 &= \frac{f'(z)}{\prod_{j=1}^n (z - z_j)} - \frac{f(z)}{\prod_{j=1}^n (z - z_j)} \cdot \sum_{\ell=1}^n \frac{\prod_{j=1, j \neq \ell}^n (z - z_j)}{\prod_{j=1}^n (z - z_j)} \\
 &= \frac{f'(z)}{\prod_{j=1}^n (z - z_j)} - \frac{f(z)}{\prod_{j=1}^n (z - z_j)} \cdot \sum_{j=1}^n \frac{1}{z - z_j}
 \end{aligned}$$

We then divide by  $g(z)$  to get

$$\frac{g'(z)}{g(z)} = \frac{f'(z)}{f(z)} - \sum_{j=1}^n \frac{1}{z - z_j}.$$

Additionally, we can multiply through by  $z^\ell$  for any  $\ell$ ,

$$z^\ell \frac{g'(z)}{g(z)} = z^\ell \frac{f'(z)}{f(z)} - \sum_{j=1}^n z^\ell \frac{1}{z - z_j}.$$

Finally, we can write Equation (2.5) in terms of  $f(z)$  and  $f'(z)$ ,

$$\begin{aligned}
 0 &= \frac{1}{2\pi i} \oint_{\gamma} z^\ell \frac{g'(z)}{g(z)} dz \\
 &= \frac{1}{2\pi i} \oint_{\gamma} z^\ell \frac{f'(z)}{f(z)} dz - \sum_{j=1}^n \frac{1}{2\pi i} \oint_{\gamma} \frac{z^\ell}{z - z_j} dz.
 \end{aligned} \tag{2.6}$$

We now construct a set of  $n$  equations to solve for the  $n$  zeros by letting  $\ell = 1, 2, 3, \dots, n$ . By the Cauchy integral formula (2.1) we have

$$\frac{1}{2\pi i} \oint_{\gamma} \frac{z^\ell}{z - z_j} dz = z_j^\ell$$

for all  $j = 1, 2, 3, \dots, n$ . Substituting this formula into Equation (2.6) gives the result

$$z_1^\ell + z_2^\ell + \dots + z_n^\ell = M_\ell, \tag{2.7}$$

where  $M_\ell$  is the  $\ell^{\text{th}}$  moment of  $f(z)$  about  $z = 0$ ,

$$M_\ell = \frac{1}{2\pi i} \oint_\gamma z^\ell \frac{f'(z)}{f(z)} dz.$$

We summarize the results of this derivation in a useful theorem.

**Theorem 2.1.** *Let  $f$  be analytic on a simply connected domain  $D$ . Suppose that  $z_j \in D$  for  $j = 1, 2, 3, \dots, n$ , and  $f(z_j) = 0$  for all  $j$ . Further, suppose  $\gamma$  is a simple closed curve in  $D$  that encloses  $z_j$  for all  $j$ . Set*

$$M_\ell = \frac{1}{2\pi i} \oint_\gamma z^\ell \frac{f'(z)}{f(z)} dz,$$

the  $\ell^{\text{th}}$  moment of  $f(z)$  about  $z = 0$ . We can write a system of algebraic equations that solves for each zero  $z_j$  as follows:

$$\begin{aligned} z_1 + z_2 + \dots + z_n &= M_1 \\ z_1^2 + z_2^2 + \dots + z_n^2 &= M_2 \\ &\vdots \\ z_1^n + z_2^n + \dots + z_n^n &= M_n. \end{aligned} \tag{2.8}$$

The moments have interesting properties. The zeroth moment is the winding number of  $f(z)$ , and consequently provides the number of zeros located within the contour. The ratios  $M_1/M_0$  and  $M_2/M_0$  provide the center of mass and variance, respectively, of the zeros. As a final note, in the case that  $M_0 = 1$ , i.e., the contour contains within it only one zero, we only need the first equation:

**Corollary 2.2.** *Suppose the contour  $\gamma$  encloses only one zero,  $z_1$ . Then the zero is the first moment,*

$$z_1 = M_1.$$

### 3. NUMERICAL COMPUTATION FOR SIMPLE ZEROS

We have derived an algebraic system of equations to solve for  $n$  zeros enclosed within a contour  $\gamma$ . In this section, we will show a method of numerically computing the solutions of this system to find precise approximations of zeros, which requires computing the moment  $M_\ell$ . To do this, we must consider how to define a contour integral  $\gamma$  that encloses zeros of the function. We must also consider how to find  $f'(z)$  numerically given that we have  $f(z)$  explicitly. For this paper, we only apply our method to solve for a simple system where  $n = 1$ , so we will use Corollary 2.2. However we could modify the same method for contours which enclose more than one zero without much difficulty.

To begin, our method requires that a rough approximation  $z_0$  of the zero of the function  $f(z)$  is known. With this, we define the contour  $\gamma$  to be a circle  $C_R$  with radius  $R$  centered at  $z_0$

$$\gamma = C_R = \{z_0 + R \cos \theta + iR \sin \theta, \quad 0 \leq \theta \leq 2\pi\}.$$

We can then make the substitutions

$$z = z_0 + R \cos \theta + iR \sin \theta, \quad dz = i(R \cos \theta + iR \sin \theta)d\theta,$$

and rewrite Corollary 2.2 for the precise zero as

$$\begin{aligned} z_1 &= \frac{1}{2\pi i} \int_0^{2\pi} (z_0 + R \cos \theta + iR \sin \theta) \frac{f'(z_0 + R \cos \theta + iR \sin \theta)}{f(z_0 + R \cos \theta + iR \sin \theta)} i(R \cos \theta + iR \sin \theta) d\theta \\ &= \frac{R}{2\pi} \int_0^{2\pi} (z_0 + R \cos \theta + iR \sin \theta) \frac{f'(z_0 + R \cos \theta + iR \sin \theta)}{f(z_0 + R \cos \theta + iR \sin \theta)} (\cos \theta + i \sin \theta) d\theta \end{aligned} \tag{3.1}$$

We will use the trapezoidal rule to approximate the definite integral in Equation (3.1). As shown by Trefethen [9], this quadrature method is spectrally accurate when integrating a periodic function over one period. We

discretize the contour  $C_R$  by  $m_R$  points and label the points as  $z_{R,k}$ , with  $k = 1, 2, 3, \dots, m_R$ , as shown in Figure 2. The number  $m_R$  does not depend on  $R$ , but rather, the  $R$  is necessary to indicate the contour discretized by  $m_R$ . Then we evaluate the integrand of Equation (3.1) at each of these points  $z_{R,k}$ .

In order to calculate the integrand we must evaluate  $f'(z_0 + R \cos \theta + iR \sin \theta)$  at each point along the contour  $C_R$ . This is typically done using some types of finite difference scheme (e.g., centered differences); however, here we will compute the derivative using the Cauchy integral formula in Equation (2.2). To compute the derivative for each  $z_{R,k}$ , we must again define a contour  $\gamma$  for each point. We choose to make circles, call them  $C_{r,k}$ , with radius  $r \ll R$  centered at each  $z_{R,k}$  where  $k = 1, 2, 3, \dots, m_R$  so that

$$\gamma = C_{r,k} = \{z_{R,k} + r \cos \theta + ir \sin \theta, \quad 0 \leq \theta \leq 2\pi\}, \quad k = 1, 2, 3, \dots, m_R.$$

Now we make a similar substitution as previously for each of our contours  $C_{r,k}$

$$z = z_{R,k} + r \cos \theta + ir \sin \theta, \quad dz = i(r \cos \theta + ir \sin \theta)d\theta$$

so that we can rewrite the derivative of the Cauchy integral formula, Equation (2.2), as follows

$$\begin{aligned} f'(z_{R,k}) &= \frac{1}{2\pi i} \int_0^{2\pi} \frac{f(z_{R,k} + r \cos \theta + ir \sin \theta)}{(z_{R,k} + r \cos \theta + ir \sin \theta - z_{R,k})^2} i(r \cos \theta + ir \sin \theta)d\theta \\ &= \frac{1}{2\pi i} \int_0^{2\pi} \frac{f(z_{R,k} + r \cos \theta + ir \sin \theta)}{(r \cos \theta + ir \sin \theta)^2} i(r \cos \theta + ir \sin \theta)d\theta \\ &= \frac{1}{2\pi r} \int_0^{2\pi} \frac{f(z_{R,k} + r \cos \theta + ir \sin \theta)}{\cos \theta + i \sin \theta} d\theta. \end{aligned} \quad (3.2)$$

We again discretize each contour  $C_{r,k}$  with  $m_r$  points and finally use the trapezoidal rule to get the integral needed to find  $f'(z_0 + R \cos \theta + iR \sin \theta)$  at each of the discretized points along the contour  $C_R$ . We expect that the numerical evaluation will be spectrally accurate, which is typically much better than what is available using a finite difference scheme. Now we have all the pieces we need to compute Equation (3.1) to solve for  $z_1$ . We can then repeat this process to find more zeros of the function so long as we have some rough approximation  $z_0$  of each zero.

Though we could use one of several methods to compute the derivative at each point along the contour, we choose this method because of the precision it gives. Table 1 gives a comparison of the derivatives computed with this contour method compared to derivatives computed with a one-sided finite difference scheme. For this example, we use the sine function,  $f(z) = \sin(z)$ . The finite difference method is computed by  $f'(z) = (\sin(z+h) - \sin(z))/h$  with  $h = 1/n$ . We also let the same  $n = m_{r,k}$ , which is the number of points we use to discretize the contour integral. As expected, we conclude from the table that as we choose higher values of  $n$ , the precision of the contour integral method grows much faster than the finite difference method.

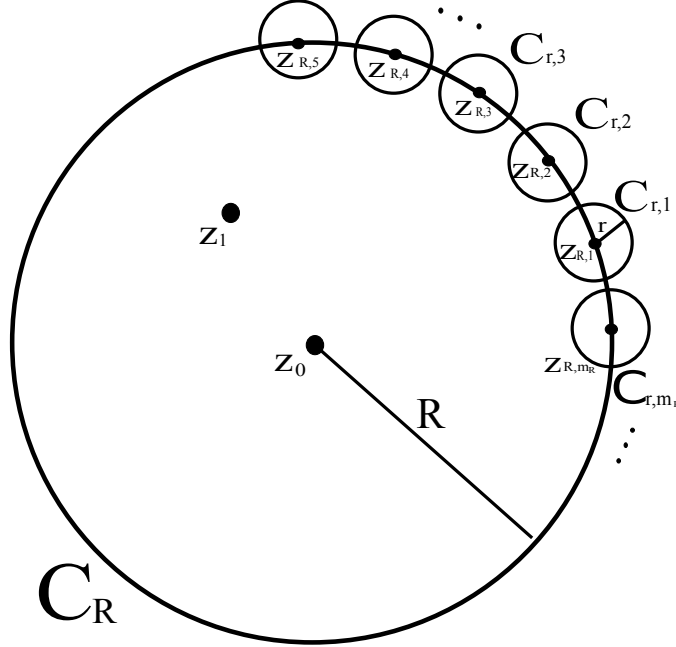
### 3.1. Example of precision

We want to demonstrate the precision of this method of numerical computation of zeros of analytic functions. We will pick the function

$$f(z) = z^3 + 5z^2 + z + 5 = (z + 5)(z + i)(z - i)$$

for our demonstration. This function is analytic with one real zero at  $z = -5$  and two complex zeros at  $z = \pm i$ . We will numerically approximate these zeros using Equation (3.1) and show a comparison with the exact zeros.

Before we can implement Equation (3.1), we must choose a contour of integration,  $C_R$ , for each zero and choose how to discretize the contours. We will let our contours  $C_R$  be circles centered 0.1 to the left of each of the exact zeros of  $f(z)$ . We must then pick a radius which ensures that exactly one zero is enclosed in each  $C_R$ . Any  $0.1 < R < 2$  will enclose exactly one zero. However, in numerically evaluating Equation (3.1), not all  $0.1 < R < 2$  will give a good approximation of the zeros. A good numerical approximation requires that the distance between the actual zeros and the center of the contour is small relative to the distance



**Figure 2:** We evaluate Equation (3.1) to find a precise approximation of zero of an analytic function  $z_1$ . We integrate along the circle  $C_R$ , which is centered at  $z_0$ , a rough approximation of a zero of  $f(z)$ , with radius  $R$ . We discretize this contour by  $m_R$  points and evaluate the integrand of Equation (3.1) at each of these points. This requires computing  $f'(z_0 + R \cos \theta + iR \sin \theta)$  at each point. For this we use the derivative of the Cauchy integral formula. For each point  $z_{R,k}$  along  $C_R$ , we define a new contour by drawing a circle, centered at that point along the contour, with radius  $r \ll R$ .

between the actual zero and the edge of the contour (see Bornemann [2] for a good discussion of this issue). For this reason, we will choose the radius  $R = 1$ , which is ten times greater than the distance between the actual zero and the approximate zero. Then our explicit contours  $C_R$  for each zero of  $f(z)$  are as follows:

$$\begin{aligned} z = -5 : \quad C_R &= \{-4.9 + 1 \cdot \cos \theta + i \cdot 1 \cdot \sin \theta, \quad 0 \leq \theta \leq 2\pi\} \\ z = +i : \quad C_R &= \{0.1 + i + 1 \cdot \cos \theta + i \cdot 1 \cdot \sin \theta, \quad 0 \leq \theta \leq 2\pi\} \\ z = -i : \quad C_R &= \{0.1 - i + 1 \cdot \cos \theta + i \cdot 1 \cdot \sin \theta, \quad 0 \leq \theta \leq 2\pi\}. \end{aligned}$$

To discretize the contours, we must also make a choice of  $m_R$  for the large contours,  $C_R$ , and  $m_{r,k}$  for the small contours for the derivative calculations,  $C_{r,k}$ . These do not need to be large to find a very precise approximation of the zeros. The following tables show the accuracy of the zeros as we vary the number of points we take on the contours. In Table 2 we do not calculate the derivatives using the derivative of the Cauchy integral formula, but instead we write the exact derivative  $f'(z) = z^2 + 5z + 1$  explicitly into the MATLAB code. In this way, we can consider choices of  $m_R$  alone. In Table 3 we will fix  $m_R = 64$  based on findings in Table 2, and consider choices of  $m_{r,k}$  alone. We will choose  $r = 0.01$  for the radius of the small contours. By Cauchy theory we see that the choice of  $r$  does not effect the derivative calculation. This essentially holds true of the numerical calculation as well. The derivatives decrease in accuracy only slightly as  $r$  grows larger. In both tables below, the second column gives the exact zeros of the functions. The third column gives the numerical approximation of the zeros given by implementing Equation (3.1). The fourth column gives the error between the first two.

From Table 2 we conclude our MATLAB program converges to an accuracy limit of  $\mathcal{O}(10^{-16})$  when using the exact derivatives. We see in Table 2 that the numerics begin to reach their limit of accuracy around  $m_R = 64$ ; that is, values of  $m_R > 64$  do not significantly improve the accuracy of the zeros. For this reason, we fix  $m_R = 64$  while we vary the value of  $m_{r,k}$  in Table 3. By Table 3 we conclude that when approximating derivatives using the Cauchy integral formula, our MATLAB program still converges to an accuracy limit of



$n$	Finite difference error	Finite difference time	Contour integral error	Contour integral time
4	0.0104	2.4800e-06 seconds	0.0083	0.0119 seconds
8	0.0026	4.1580e-06 seconds	2.7557e-06	0.0154 seconds
12	0.0012	4.2810e-06 seconds	1.6059e-10	0.0119 seconds
16	6.5091e-04	7.5290e-06 seconds	2.8866e-15	0.0123 seconds
32	1.6275e-04	2.3840e-06 seconds	9.3871e-18	0.0144 seconds
64	4.0690e-05	2.6760e-06 seconds	3.0508e-17	0.0202 seconds

**Table 1:** This table shows that Equation (3.2) numerically computes derivatives with much more precision than a finite difference method. The first column,  $n$ , gives both  $m_{r,k}$ , the number of points along the contour  $C_{r,k}$  for the Cauchy integral method and the  $n$  for  $h = 1/n$  for the finite difference method. For this example, we use the sine function and choose  $x = \pi$ . In order for the finite difference scheme to reach an accuracy of  $\mathcal{O}(10^{-16})$  we must set  $n$  to be  $\mathcal{O}(10^4)$ , compared to the less than 32 points needed for the Cauchy integral method to reach that accuracy. In this table, we also include columns giving the running time given by the MATLAB functions `tic` and `toc`.

$\mathcal{O}(10^{-16})$ . We can reach this accuracy with  $m_{r,k} = 4$ . Increasing the number of points by which we discretize our contours beyond 4 does not improve the accuracy. This example is representative of the all examples we conducted, but that more study would need to be done to generalize the applicability of the method.

## 4. APPLICATION: THE EVANS FUNCTION

Consider the general Sturm-Liouville problem for  $a \leq x \leq b$ ,

$$-\frac{\partial}{\partial x} \left( p(x) \frac{\partial y}{\partial x} \right) + q(x)y = \lambda w(x)y$$

with boundary conditions

$$b_i^1 y(a, \lambda) + b_i^2 \frac{\partial y}{\partial x}(a, \lambda) = 0, \quad b_f^1 y(b, \lambda) + b_f^2 \frac{\partial y}{\partial x}(b, \lambda) = 0.$$

The subscript  $i$  denotes initial conditions, and the subscript  $f$  denotes final conditions. We make the physically relevant assumptions that  $p(x), q(x), w(x) > 0$ . The Sturm-Liouville problem is an infinite dimensional eigenvalue problem where the goal in solving the problem is to find the eigenvalues  $\lambda$  for which there is a nontrivial solution. Each solution  $y(x, \lambda)$  associated with an eigenvalue  $\lambda$  is known as an eigenfunction.

We use the Evans function as the tool to solve for eigenvalues of the Sturm-Liouville problem. The Evans function is analogous to the characteristic polynomial of a matrix. Finding zeros of the characteristic polynomial is a common method used to find eigenvalues of a matrix. The eigenvalues have algebraic multiplicity corresponding to the multiplicity of the zero of the characteristic polynomial. We define the Evans function in such a way that its zeros are the eigenvalues of the Sturm-Liouville problem. It is the case that the order of each zero is equal to the algebraic multiplicity of the eigenvalue. However, while the characteristic polynomial can be thought of as a finite Taylor series, the Evans function will have countable infinitely many zeros.

In Section 2 we derived a set of equations to solve for unknown zeros of an analytic function and in Section 3 we showed a method to numerically compute one zero. In the next section, Section 5, we will apply this method to finding zeros of the Evans function and thus the eigenvalues of the Sturm-Liouville problem for  $a \leq x \leq b$ . In preparation for this application, in this section we will demonstrate how the Evans function is constructed.

To begin our construction, we first want to view the Sturm-Liouville problem from a geometric perspective, thus we write it as a vector-valued problem. We make the following substitution,

$$u(x, \lambda) = y(x, \lambda), \quad v(x, \lambda) = p(x) \frac{\partial y}{\partial x}(x, \lambda),$$

Value of $m_R$	Exact zero	Numerical approximation	Error	Running time
8	i	0.0015 + 1.0036i	0.0039	0.0192 seconds
	-i	0.0015 + 1.0036i	0.0039	
	-5	5.0000 + 3.3573e-16i	5.1578e-06	
12	i	0.0001 + 1.0002i	0.0002	0.0212 seconds
	-i	0.0001 + 1.0002i	0.0002	
	-5	5.0000 + 0.0000i	5.4292e-9	
16	i	0.0000 + 1.0000i	1.4957e-5	0.0221 seconds
	-i	0.0000 + 1.0000i	1.4957e-5	
	-5	5.0000 + 0.0000i	1.0365e-12	
32	i	0.0000 + 1.0000i	2.2371e-10	0.0209 seconds
	-i	0.0000 + 1.0000i	2.2371e-10	
	-5	5.0000 + 0.0000i	4.6383e-17	
64	i	0.0000 + 1.0000i	1.1242e-16	0.0251 seconds
	-i	0.0000 + 1.0000i	2.3738e-16	
	-5	5.0000 + 0.0000i	9.4825e-16	
128	i	0.0000 + 1.0000i	0.0000	0.0256 seconds
	-i	0.0000 + 1.0000i	1.2400e-16	
	-5	5.0000 + 0.0000i	1.4867e-16	

**Table 2:** In this table, we show that discretizing  $C_R$ , where  $R = 1$ , by 64 points is sufficient for good approximations of zeros. The table gives a comparison between exact zeros and numerical computations of zeros of the function  $f(z) = z^3 + 5z^2 + z + 5$  for various values of  $m_R$ . We find the numerical approximations by implementing (3.1) and calculating the exact derivatives along the contour  $C_R$  instead of approximating the derivatives. The final column gives the running time for all three zeros given by the MATLAB functions tic and toc.

which forces the boundary conditions to become

$$b_i^1 u(a, \lambda) + \frac{b_i^2}{p(a)} v(a, \lambda) = 0, \quad b_f^1 u(b, \lambda) + \frac{b_f^2}{p(b)} v(b, \lambda) = 0$$

By differentiating  $u(x, \lambda)$  and  $v(x, \lambda)$  with respect to  $x$  we have

$$\frac{\partial u}{\partial x}(x, \lambda) = \frac{\partial y}{\partial x}(x, \lambda) = \frac{v(x, \lambda)}{p(x)}$$

and

$$\begin{aligned} \frac{\partial v}{\partial x}(x, \lambda) &= \frac{\partial}{\partial x} \left[ p(x) \frac{\partial v}{\partial x}(x, \lambda) \right] \\ &= q(x)y(x, \lambda) - \lambda w(x)y(x, \lambda) \\ &= [q(x) - \lambda w(x)]u(x, \lambda). \end{aligned}$$

We have now turned a second order ODE into a first order ODE, and the Sturm-Liouville ODE can be written in vector form as

$$\frac{\partial}{\partial x} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} 0 & 1/p(x) \\ q(x) - \lambda w(x) & 0 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix}. \quad (4.1)$$

In terms of the original variables

$$\begin{pmatrix} u(x, \lambda) \\ v(x, \lambda) \end{pmatrix} = \begin{pmatrix} y(x, \lambda) \\ p(x) \frac{\partial y}{\partial x}(x, \lambda) \end{pmatrix}$$

Value of $m_{r,k}$	Exact zero	Numerical approximation	Error
1	i	0.0054 + 0.0019i	0.0057
	-i	0.0054 + 0.0019i	0.0057
	-5	0.0192 + 0.0000i	0.0192
2	i	0.0001 + 1.0002i	9.8058e-06
	-i	0.0001 + 1.0002i	9.8058e-06
	-5	5.0000 + 0.0000i	1.9231e-05
3	i	0.0000 + 1.0000i	2.4511e-15
	-i	0.0000 + 1.0000i	2.7239e-15
	-5	5.0000 + 0.0000i	3.7322e-14
4	i	0.0000 + 1.0000i	3.8069e-16
	-i	0.0000 + 1.0000i	3.7711e-16
	-5	5.0000 + 0.0000i	5.6850e-14
8	i	0.0000 + 1.0000i	1.7917e-15
	-i	0.0000 + 1.0000i	1.9235e-15
	-5	5.0000 + 0.0000i	6.2173e-14

**Table 3:** By this table, we show that discretizing  $C_{r,k}$ , where  $r = 0.01$ , by 4 points gives sufficient approximations of derivatives along  $C_R$  needed for the zero calculations. The table does this by giving a comparison between exact zeros and numerical computations of zeros of the function  $f(z) = z^3 + 5z^2 + z + 5$  for various values of  $C_{r,k}$  when we fix  $m_R = 64$ . We find the numerical approximations by implementing (3.1).

The initial conditions for this system of ODEs given by Equation (4.1) is determined by the boundary condition at  $x = a$ . Since

$$b_i^1 y(a, \lambda) + b_i^2 \frac{\partial y}{\partial x}(a, \lambda) = 0,$$

the boundary condition at  $x = a$  is satisfied anywhere on the line,

$$b_i^1 u + \frac{b_i^2}{p(a)} v = 0.$$

The vector

$$\mathbf{a} := \begin{pmatrix} -b_i^2/p(a) \\ b_i^1 \end{pmatrix}$$

is parallel to this line; consequently, any multiple of this vector can serve as an initial condition for the ODE system shown in Equation 4.1. We use the solution to the system Equation (4.1) which has the initial condition

$$\begin{pmatrix} u(a, \lambda) \\ v(a, \lambda) \end{pmatrix} = \mathbf{a}.$$

Similarly, we consider the boundary conditions at  $x = b$ . Since

$$b_f^1 y(b, \lambda) + b_f^2 \frac{y(b, \lambda)}{p(b)} = 0,$$

the boundary condition at  $x = b$  is satisfied anywhere on the line,

$$b_f^1 u + \frac{b_f^1}{p(b)} v = 0. \tag{4.2}$$

The vector

$$\mathbf{b} := \begin{pmatrix} -b_f^2/p(b) \\ b_f^1 \end{pmatrix} \tag{4.3}$$

is parallel to this line. Thus, a nontrivial solution of the Sturm-Liouville problem which satisfies the boundary conditions at  $x = a$  and  $x = b$  will satisfy

$$\begin{pmatrix} u(a, \lambda) \\ v(a, \lambda) \end{pmatrix} = \mathbf{a}, \quad \begin{pmatrix} u(b, \lambda) \\ v(b, \lambda) \end{pmatrix} \propto \mathbf{b}.$$

In order to find the eigenvalues  $\lambda$  that correspond to a nontrivial solution, we solve this differential equation with the initial conditions being the boundary conditions at  $x = a$ , and compare the solution to the boundary conditions at  $x = b$ . If  $\lambda$  is an eigenvalue, then the solution will lie on the same line as  $\mathbf{b}$ . The condition that  $\begin{pmatrix} u(b, \lambda) \\ v(b, \lambda) \end{pmatrix}$  be a scalar multiple of  $\mathbf{b}$  is equivalent to saying that the set  $\left\{ \begin{pmatrix} u(b, \lambda) \\ v(b, \lambda) \end{pmatrix}, \mathbf{b} \right\}$  is linearly dependent. We want to define the Evans function in such a way that its zeros detect these eigenvalues  $\lambda$  for which the solutions at both boundary conditions lie on the same line. Such a function is possible to construct because we know that if the solution satisfies the boundary condition at both  $x = a$  and  $x = b$ , then the determinant of a matrix with  $\begin{pmatrix} u(b, \lambda) \\ v(b, \lambda) \end{pmatrix}$  and  $\mathbf{b}$  as its columns will be zero. Based on this, we define the Evans function in the following way:

$$\begin{aligned} E(\lambda) &= \det \left( \begin{pmatrix} u(b, \lambda) \\ v(b, \lambda) \end{pmatrix} \mathbf{b} \right) \\ &= \det \begin{pmatrix} y(b, \lambda) & -b_f^2/p(b) \\ p(b) \frac{\partial y}{\partial x}(b, \lambda) & b_f^1 \end{pmatrix} \\ &= b_f^1 y(b, \lambda) + \frac{b_f^2}{p(b)} p(b) \frac{\partial y}{\partial x}(b, \lambda) \\ &= b_f^1 y(b, \lambda) + b_f^2 \frac{\partial}{\partial x} y(b, \lambda). \end{aligned} \tag{4.4}$$

The Evans function has many properties (see Kapitula and Promislow [6, Chapter 8]), including:

- the Evans function is analytic
- there exists an equivalence between order of the zeros of the Evans function and multiplicity of eigenvalues of the Sturm-Liouville problem; in particular, the order of each zero is equal to the algebraic multiplicity of the eigenvalue.

The Evans function is analogous to the characteristic polynomial of a matrix in that zeros of the Evans function are equivalent to eigenvalues of the Sturm-Liouville problem.

**Theorem 4.1.** *Consider the Sturm-Liouville problem*

$$-\frac{\partial}{\partial x} \left( p(x) \frac{\partial y}{\partial x} \right) + q(x)y = \lambda w(x)y$$

with the boundary conditions

$$b_i^1 y(a, \lambda) + b_i^2 \frac{\partial y}{\partial x}(a, \lambda) = 0, \quad b_f^1 y(b, \lambda) + b_f^2 \frac{\partial y}{\partial x}(b, \lambda) = 0.$$

Let  $y_E$  be the solution to the Sturm-Liouville ODE with initial condition

$$y_E(a, \lambda) = -\frac{b_i^2}{p(a)}, \quad \frac{\partial}{\partial x} y_E(a, \lambda) = \frac{b_i^1}{p(a)}.$$

Set the Evans function to be

$$E(\lambda) = b_f^1 y_E(b, \lambda) + b_f^2 \frac{\partial}{\partial x} y_E(b, \lambda).$$

An eigenvalue for the Sturm-Liouville problem with the proscribed boundary conditions arises if and only if it is realized as a zero of the Evans function. In other words,  $\lambda_0$  is an eigenvalue if and only if  $E(\lambda_0) = 0$ . In this case,  $y_E(x, \lambda_0)$  gives the eigenfunction associated with the eigenvalue  $\lambda_0$ .

### 4.1. Example: Evans function construction

We will now give a concrete example of constructing the Evans function for a particular Sturm-Liouville problem. For this example, we set  $a = 0, b = 1$  (so  $0 \leq x \leq 1$ ) and choose Dirichlet boundary conditions,

$$b_i^1 = 1, \quad b_i^2 = 0, \quad b_f^1 = 1, \quad b_f^2 = 0.$$

We will pick the following coefficient functions,

$$p(x) = 1, \quad q(x) = 0, \quad w(x) = 1.$$

This gives the following Sturm-Liouville problem,

$$\frac{\partial^2 y}{\partial x^2} = \lambda y, \quad y(0) = y(1) = 0. \tag{4.5}$$

We know the eigenvalues for this problem are

$$\lambda = -n^2 \pi^2, \quad n = 1, 2, \dots$$

Associated eigenfunctions are given by  $y(x) \propto \sin(n\pi x)$ . We now show how this result can be recovered using the Evans function.

In order to simplify our construction of the function  $y_E(x, \lambda)$ , we first set  $\lambda := -\gamma^2$  for  $\gamma > 0$ . We can rewrite the system as

$$\frac{\partial^2 y}{\partial x^2} = -\gamma^2 y \quad \Rightarrow \quad \frac{\partial^2 y}{\partial x^2} + \gamma^2 y = 0.$$

The general solution is

$$y(x, \gamma) = c_1 \cos(\gamma x) + c_2 \frac{\sin(\gamma x)}{\gamma}.$$

Note that writing the solution in this form leads to the limiting general solution at  $\gamma = 0$ ,

$$\lim_{\gamma \rightarrow 0^+} y(x, \gamma) = c_1 + c_2 x.$$

Considering [Theorem 4.1](#), the relevant initial condition in the construction of the Evans function is

$$y_E(0, \gamma) = 0, \quad \frac{\partial y_E}{\partial x}(0, \gamma) = 1.$$

This initial condition gives the solution

$$y_E(x, \gamma) = \frac{\sin(\gamma x)}{\gamma}.$$

Thus, the Evans function for this problem is given by

$$E(\gamma) = y_E(1, \gamma) = \frac{\sin(\gamma)}{\gamma}.$$

If  $E(\gamma_0) = 0$ , then  $\lambda_0 = -\gamma_0^2$  will be the eigenvalue for the original Sturm-Liouville problem.

We now recover the known result. We set the Evans function equal to zero to find the values of  $\gamma$  that will give eigenvalues

$$E(\gamma) = 0 \quad \longleftrightarrow \quad \gamma = n\pi, \quad n = 1, 2, \dots$$

Since  $\lambda = -\gamma^2$ , the eigenvalues  $\lambda$  are then

$$\lambda = -n^2 \pi^2, \quad n = 1, 2, \dots$$

Since the solution  $y_E(x, \lambda)$  provides the associated eigenfunction, we have

$$y_E(x, -n^2 \pi^2) = \frac{1}{n\pi} \sin(n\pi x), \quad n = 1, 2, \dots$$

## 5. NUMERICAL IMPLEMENTATION TO SOLVE STURM-LIOUVILLE PROBLEM

In this section we will present an algorithm for numerically computing the zeros of the Evans function, which is equivalent to computing eigenvalues of the Sturm-Liouville problem. We implement this algorithm with a MATLAB program which numerically calculates these zeros with high precision by employing Corollary 2.2. Since there are countably infinite eigenvalues for which there is a solution to the Sturm-Liouville problem, our program only searches for eigenvalues within a specific region based on user input. To run the MATLAB program, the user must enter the following information as shown in Figure 3:

- boundary condition coefficients  $b_i^1$ ,  $b_i^2$ ,  $b_f^1$ , and  $b_f^2$
- rectangular region which encapsulates the desired eigenvalues, with  $x$  minimum  $\leq \text{Re } \lambda \leq x$  maximum and  $y$  minimum  $\leq \text{Im } \lambda \leq y$  maximum
- interval beginning and end points; that is,  $a$  and  $b$  such that  $a \leq x \leq b$
- functions for  $p(x)$ ,  $q(x)$ , and  $w(x)$  of the Sturm-Liouville problem.

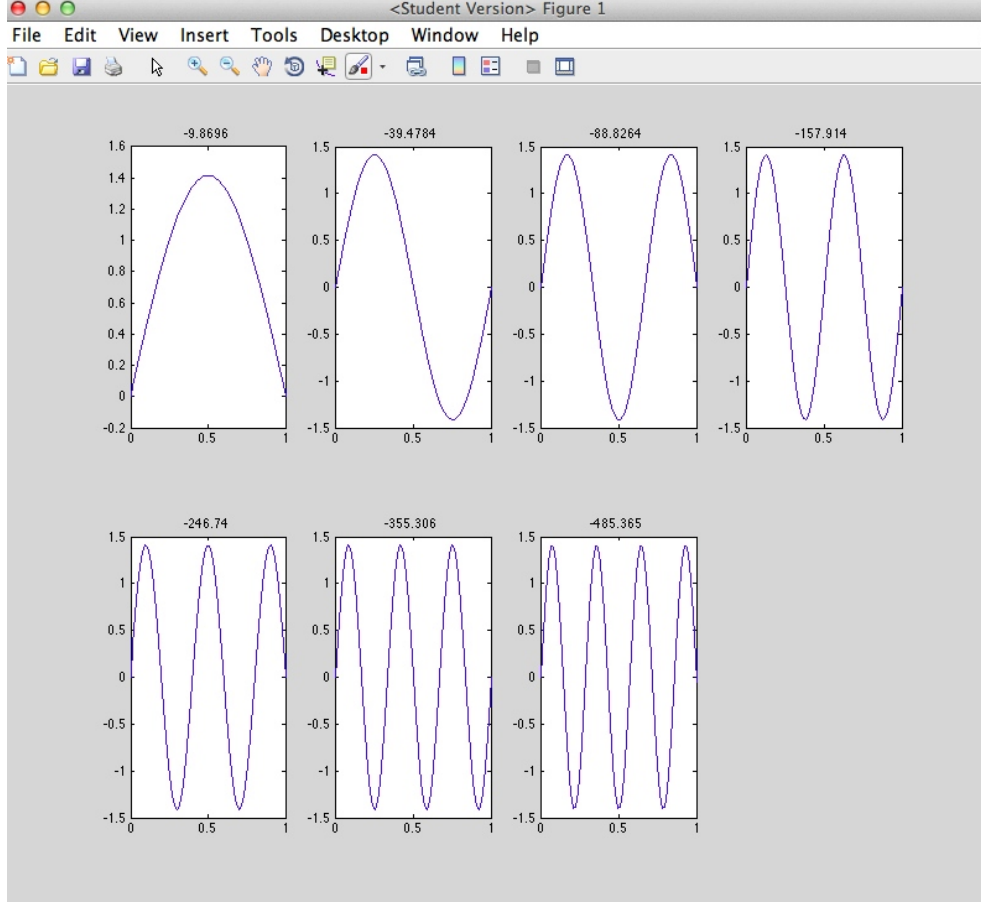
**Figure 3:** User input window for the MATLAB program used to find the eigenvalues of the Sturm-Liouville problem.

The eigenvalues within the given rectangular domain are found one at a time. Each eigenvalue is listed in the output window above a graph of the corresponding eigenfunction as shown in Figure 4. In particular, using Corollary 2.2 each eigenvalue,  $\lambda_*$ , is found via

$$\lambda_* = \frac{1}{2\pi i} \oint_{\gamma} \lambda \frac{E'(\lambda) d\lambda}{E(\lambda)}. \quad (5.1)$$

In order for the expression in (5.1) to be valid the curve  $\gamma$  must encircle only the single eigenvalue. Following the discussion in Section 3, we let  $\gamma$  be a circle  $C_R$  which encloses exactly one zero of the Evans function. We

define the circle contour  $C_R$  for each zero to be a circle centered at an approximation of the corresponding eigenvalue obtained by writing the Sturm-Liouville problem in matrix-vector form and using a finite difference method described in Subsection 5.1. We limit the list of eigenvalue approximations to only those within the user defined region and apply Equation (5.1) to each one.



**Figure 4:** The output window for the MATLAB program shows each eigenvalue of the Sturm-Liouville problem within the selected rectangular domain with a graph of its corresponding eigenfunction.

### 5.1. Finite difference eigenvalue approximation

In this subsection, we show how to find approximations of eigenvalues to use as the centers of circles which will enclose exactly one zero of the the Evans function. Using a finite difference method, we write the Sturm-Liouville problem in matrix-vector form. Equation (1.1) gives the Sturm-Liouville problem,

$$-\frac{\partial}{\partial x} \left( p(x) \frac{\partial y}{\partial x} \right) + q(x)y = \lambda w(x)y,$$

with the boundary conditions,

$$b_i^1 y(a, \lambda) + b_i^2 \frac{\partial y}{\partial x}(a, \lambda) = 0, \quad b_f^1 y(b, \lambda) + b_f^2 \frac{\partial y}{\partial x}(b, \lambda) = 0.$$

The interval is first discretized into  $N - 1$  equal subintervals with stepsize  $h = 1/(N - 1)$  where  $N$  is defined in the MATLAB program and can be changed by the user. First we will write the term  $\frac{\partial}{\partial x} \left( p(x) \frac{\partial y}{\partial x} \right)$  as

an  $N \times N$  matrix  $\mathbf{D}_p$  with the following differentiation scheme,

$$\frac{1}{h^2} \left\{ p \left( x + \frac{h}{2} \right) [y(x+h, \lambda) - y(x, \lambda)] - p \left( x - \frac{h}{2} \right) [y(x, \lambda) - y(x-h, \lambda)] \right\}.$$

This scheme must be modified at the end points,  $x = a$  or  $x = b$ . Consider  $x = a$ , so we initially have

$$\frac{1}{h^2} \left\{ p \left( a + \frac{h}{2} \right) [y(a+h, \lambda) - y(a, \lambda)] - p \left( a - \frac{h}{2} \right) [y(a, \lambda) - y(a-h, \lambda)] \right\}.$$

Since  $p(x)$  and  $y(x, \lambda)$  cannot be evaluated for  $x$  less than  $a$ , neither  $p(a - \frac{h}{2})$  nor  $y(a-h, \lambda)$  are allowed. To compensate for this, we use the linear approximations,

$$p \left( a - \frac{h}{2} \right) = p(a) + p'(a) \left( a - \frac{h}{2} \right),$$

and

$$y(a-h, \lambda) = y(a, \lambda) + \frac{\partial}{\partial x} y(a, \lambda) (a-h).$$

We can find  $\partial_x y(a, \lambda)$  with the boundary condition,

$$\frac{\partial}{\partial x} y(a, \lambda) = -\frac{b_i^1}{b_i^2} y(a, \lambda).$$

With this substitution we write an approximation at  $x = a$  as

$$\frac{1}{h^2} \left[ -p \left( \frac{h}{2} \right) + \left( \frac{b_i^1}{b_i^2} \right) h p(0) - \left( \frac{h^2}{2} \right) \left( \frac{b_i^1}{b_i^2} \right) p'(0) \right] y(0, \lambda) + \frac{1}{h^2} \left[ p \left( \frac{h}{2} \right) y(h) \right].$$

Similarly, at the right end point ( $x = b$ ),  $p(x)$  and  $y(x)$  cannot be evaluated for  $x$  greater than  $b$ . We again use the appropriate linear approximation, as well as the boundary condition, and eventually get

$$\frac{1}{h^2} \left[ \left( \frac{-b_f^1}{h b_f^2} \right) p(N) - \left( \frac{-b_f^1}{h b_f^2} \right) p'(N) - p \left( N - \frac{h}{2} \right) \right] y(N, \lambda) + \frac{1}{h^2} \left[ p \left( N - \frac{h}{2} \right) y(N-h, \lambda) \right].$$

We now have a scheme to build the complete finite difference matrix approximation,  $\mathbf{D}_p$ , of the first term of the Sturm-Liouville ODE. We also write the functions  $q(x)$  and  $w(x)$  in matrix form; specifically, we create diagonal matrices  $\mathbf{Q}$  and  $\mathbf{W}$ , respectively, for which the diagonal elements are the functions evaluated at the discretization points. Finally, we have approximations of eigenvalues of the Sturm-Liouville problem by using the MATLAB command `eig` to calculate the eigenvalues of the generalized eigenvalue problem,

$$(\mathbf{D}_p - \mathbf{Q})\mathbf{y} = \lambda \mathbf{W}\mathbf{y}.$$

## 5.2. Computing zeros of the Evans function

Now that we have rough approximations of the zeros of the Evans function via the eigenvalues of the finite difference scheme, our MATLAB program will follow the algorithm outlined in Section 3 to find the zeros much more precisely via

$$\lambda_* = \frac{1}{2\pi i} \oint_{\gamma} \lambda \frac{E'(\lambda)}{E(\lambda)} d\lambda.$$

First the program narrows the list of eigenvalues given by the finite difference approximation to a list of only eigenvalues within the user defined region. For each zero we then define a contour integral in which the zero is enclosed. As in Section 3, we define the contour  $\gamma$  to be a circle  $C_R$  centered at the finite difference approximation  $z_0$  with radius  $R$ . We must carefully consider the appropriate size of the radius  $R$ . In theory, the choice of the circle  $C_R$  does not matter so long as it encloses the zero of the Evans function which we



are locating. However, as shown in Section 3, when doing numerics, the choice of  $R$  does matter. To get a good estimate of the zero, the distance between the actual zero and the center of the contour must be small relative to the radius  $R$ . For  $|\lambda| < 100$ , a choice in the range of  $R = 1$  to  $R = 4$  is large enough to be significantly larger than the distance between the Evans function zero and the finite difference matrix approximation, and small enough to still give very precise approximations.

We can then make the substitutions

$$\begin{aligned}\lambda &= \lambda_0 + R \cos \theta + iR \sin \theta \\ d\lambda &= i(R \cos \theta + iR \sin \theta) d\theta\end{aligned}$$

where  $\lambda_0$  is initial approximation given by the finite difference matrix. Using equation (5.1) the numerical approximation to the zero,  $\lambda_*$ , is

$$\lambda_* = \frac{R}{2\pi} \int_0^{2\pi} (\lambda_0 + R \cos \theta + iR \sin \theta) \frac{E'(\lambda_0 + R \cos \theta + iR \sin \theta)}{E(\lambda_0 + R \cos \theta + iR \sin \theta)} (\cos \theta + i \sin \theta) d\theta. \quad (5.2)$$

To evaluate the integrand here, we need to compute the derivative of the Evans function along the contour. In Section 3 we found the derivatives numerically with the derivative of the Cauchy integral formula. We could use the same method here by defining circles  $C_{r,k}$  at each point  $\lambda_{R,k}$  with radius  $r$  such that  $r \ll R$ . Then, by defining  $\lambda_\theta = \lambda_0 + R \cos \theta + iR \sin \theta$  for some  $0 \leq \theta \leq 2\pi$ , we have

$$E'(\lambda_\theta) = \frac{1}{2\pi r} \int_0^{2\pi} \frac{E(\lambda_\theta + r \cos \theta + ir \sin \theta)}{\cos \theta + i \sin \theta} d\theta.$$

While this is a perfectly good method to get approximations of the derivative of the function  $E(\lambda)$  at each point along the contour  $C_R$ , for the Evans function application we can compute the derivative more quickly with a different method.

Since we must build and solve the ordinary differential equation to find  $E(\lambda)$  along the contour, we can find the derivative  $E'(\lambda)$  simultaneously by using the MATLAB ODE solver `ode45`. The Evans function requires that we calculate the ODE solution  $y_E(x, \lambda)$ , and the Evans function derivative calculation requires we find  $\partial_\lambda y_E(x, \lambda)$ . In order to see all this, we start with

$$E(\lambda) = b_f^1 y_E(b, \lambda) + b_f^2 \frac{\partial y_E}{\partial x}(b, \lambda),$$

and differentiate with respect to  $\lambda$  to get

$$E'(\lambda) = b_f^1 \frac{\partial}{\partial \lambda} y_E(b, \lambda) + b_f^2 \frac{\partial}{\partial \lambda} \frac{\partial y_E}{\partial x}(b, \lambda).$$

Now we find the initial value problem for  $\partial y_E / \partial \lambda$ . Since

$$-\frac{\partial}{\partial x} \left( p \frac{\partial y_E}{\partial x} \right) + a y_E = \lambda w y_E,$$

with

$$y_E(a, \lambda) = -\frac{b_i^2}{p(a)}, \quad \frac{\partial y_E}{\partial x}(a, \lambda) = \frac{b_i^1}{p(a)}.$$

upon differentiating with respect to  $\lambda$  we get

$$-\frac{\partial}{\partial x} \left[ p(x) \frac{\partial}{\partial x} \left( \frac{\partial y_E}{\partial \lambda} \right) \right] + q(x) \left( \frac{\partial y_E}{\partial \lambda} \right) = \lambda w(x) \left( \frac{\partial y_E}{\partial \lambda} \right) + w(x) y_E$$

with

$$\frac{\partial y_E}{\partial \lambda}(a, \lambda) = \frac{\partial}{\partial \lambda} \left( \frac{\partial y_E}{\partial x}(a, \lambda) \right) = 0$$

The new ODE to be solved is the Sturm-Liouville ODE with a nonhomogeneous term. Note that this additional term is the already found solution  $y_E$ . By solving this second ODE initial value problem for each value of  $\lambda$  along  $C_R$ , and evaluating the solution at  $x = b$ , we have the information we need to compute the integrand of Equation (5.2) at each discretized point along  $C_R$ .

*Remark 5.1.* Barker [1] uses the Evans function to numerically compute the eigenvalues for a large class of eigenvalue problems which often arise in the context of the study of stability of nonlinear waves for conservation laws. The MATLAB program that he developed for these types of eigenvalue problems, STABLAB, can be found at <http://impact.byu.edu/stablab/>. He computes the zeros of the Evans function using the same theoretical contour integration formula as presented herein. The primary difference is that where we compute the derivative of the Evans function by first solving an ODE for  $\partial y_E / \partial \lambda$ , he computes it via a second-order finite difference differentiation scheme. As we saw in Section 3, we generally expect that computing derivatives via the Cauchy formula, or via an ODE solution as done here, will lead to a more accurate approximation of the derivative than that provided for by a finite-difference scheme.

### 5.3. Accuracy of Cauchy integral method of finding zeros

For the Sturm-Liouville problem with Dirichlet boundary conditions and coefficient function  $p(x) = w(x) = 1$  for all  $0 \leq x \leq 1$  and  $q(x) = 0$  for all  $x$ , we know the eigenvalues to be

$$\lambda_j = -j^2 \pi^2, \quad j = 1, 2, 3, \dots,$$

as shown in the Example in Section 4.1. The table below compares the first five exact known zeros with the approximate zeros given by the MATLAB program. Here we chose  $R = 3$  and  $m_R = 16$  and let the error settings for MATLAB's ODE solver `ode45` be set to  $AbsTol = 10e - 12$  and  $RelTol = 10e - 10$ .

**Table 4:** Numerical zeros of simple Dirichlet problem

Exact zero	Numerical zero	Error
-9.869604401089358	-9.869604401118520	2.9162e-11
-39.478417604357430	-39.478417604388290	3.0858e-11
-88.826439609804230	-88.826439609846370	4.2149e-11
-157.9136704174297	-157.9136704174981	6.8325e-11
-246.7401100272340	-246.7401100277353	5.0133e-10

The same can be done for the Sturm-Liouville problem with Neumann boundary conditions,  $b_i^1 = b_f^1 = 0$ , and  $b_i^2 = b_f^2 = 1$ , and coefficient function  $p(x) = w(x) = 1$  for all  $x$  and  $q(x) = 0$  for all  $0 \leq x \leq 1$ . Here, we know the eigenvalues to be

$$\lambda_j = -j^2 \pi^2, \quad j = 0, 1, 2, 3, \dots$$

The table below compares the first five exact known zeros with the approximate zeros given by the MATLAB program. Again, we choose  $R = 3$  and  $m_R = 16$ .

**Table 5:** Numerical zeros of simple Neumann problem

Exact zero	Numerical zero	Error
0	1.124237731156201e-15	1.1242e-15
-9.869604401089358	-9.869604401092248	2.8901e-12
-39.478417604357430	-39.478417604365220	7.7875e-12
-88.826439609804230	-88.826439609824690	2.0463e-11
-157.9136704174297	-157.9136704244637	7.034e-09

The larger  $|\lambda|$  is, the less accurate the finite difference approximations are, thus the larger the difference between the matrix differentiation estimate and the actual zero. As mentioned in Section 3.1, the error in the zero given by the program depends on the distance between the initial zero estimate and the actual zero being small relative to the radius  $R$ . For this reason, for a fixed value of  $R$ , as the distance between the initial approximation and the actual zero grows, the error grows as well.

### 5.4. Further improvement to accuracy

If we are willing to allow more time for the MATLAB program to run, we can improve the accuracy of the zeros even further by changing the options for the ODE solver. The ODE solver is set at  $AbsTol = e - 12$  and  $RelTol = 1e - 10$  in examples in this paper. By setting these error parameters to more precise values, our ODE solver provides more accurate solutions. Table 6 shows how times and accuracy increase with more precise ODE options.

**Table 6:** Time increase due to ODE solver. For this example, we choose Dirichlet boundary conditions and set the  $x$  interval from 0 to 1,  $R = 3$ ,  $m_R = 64$  and the size of the finite difference matrix to be  $96 \times 96$ . The times given for the calculations includes the first seven eigenvalues, that is all the eigenvalues between 0 and -500. However, the error calculation is given by the absolute value between the exact value and estimated value of only the first eigenvalue. These calculations were made with a 1.3 GHz Intel Core i5 processor with 4 GB 1600 MHz DDR3 memory.

Abs tol	Rel tol	Error	Time
$10^{-16}$	$10^{-14}$	$10^{-15}$	4.45 minutes
$10^{-14}$	$10^{-12}$	$10^{-13}$	2.12 minutes
$10^{-12}$	$10^{-10}$	$10^{-11}$	0.81 minutes
$10^{-10}$	$10^{-8}$	$10^{-9}$	0.35 minutes
$10^{-8}$	$10^{-6}$	$10^{-7}$	0.16 minutes

## REFERENCES

- [1] B. Barker. Evans function computation. Master's thesis, Brigham Young U., 2009.
- [2] F. Bornemann. Accuracy and stability of computing high-order derivatives of analytic functions by Cauchy integrals. *Foundations Comp. Math.*, 11(1):1–63, 2011.
- [3] J. Bronski. Semiclassical eigenvalue distribution of the Zakharov-Shabat eigenvalue problem. *Physica D*, 97(4): 376–397, 1996.
- [4] J. Brown and R. Churchill. *Complex Variables and Applications*. McGraw-Hill, 7th edition, 2004.
- [5] R. Haberman. *Applied Partial Differential Equations with Fourier Series and Boundary Value Problems*. Pearson Education, Inc., 5th edition, 2012.
- [6] T. Kapitula and K. Promislow. *Spectral and Dynamical Stability of Nonlinear Waves*. Springer-Verlag, 2013.
- [7] J. D. Logan. *Applied Partial Differential Equations*. Springer, 2nd edition, 2004.
- [8] A. Markushevich. *Theory of Functions*. Chelsea Publishing Co., New York, 1985.
- [9] L. Trefethen. *Spectral Methods in MATLAB*. SIAM, 2000.