Capstone report 2
Algorithmic solution of high order partial differential equations in Julia via the Fokas transform method

Linfan XIAO
Supervisor: Prof. David Smith

Contents

1 Introduction

1.1 Motivation

Evolution partial differential equations (PDEs) relate a quantity, such as temperature, to its rates of change with respect to both time (hence "evolution") and position. Evolution PDEs can be used to describe or model a variety of phenomena in physics, such as wave propagation and particle motion. This project concerns solving a certain class of initial-boundary value problems (IBVP) for evolution PDEs in the finite interval[1] correctly and efficiently.

To motivate the project, suppose we wish to solve some IBVPs. Certain simple IBVPs (henceforth referred to as type I problems) may be solved algorithmically via classical transform pairs such as the Fourier transform (Figure 1).

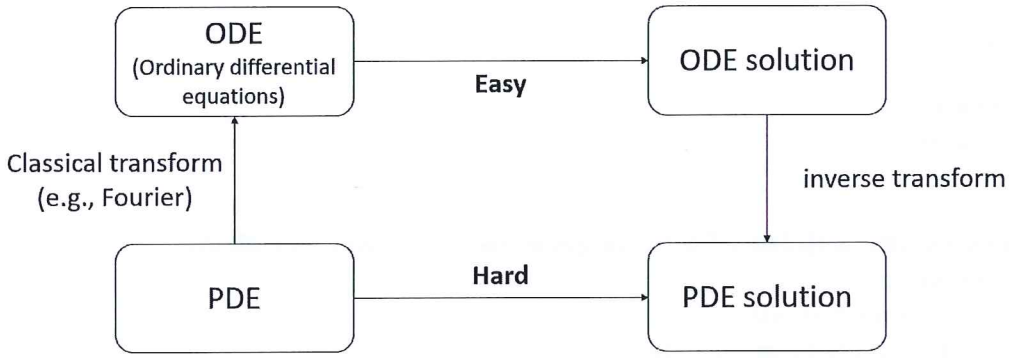

Figure 1: Solving type I IBVPs using classical transform pairs.

For more complicated IBVPs (henceforth referred to as type II problems), however, no such classical transform pairs exist. In fact, solving these IBVPs typically requires a combination of ad-hoc methods. These methods are often specific to the given problem and cannot be generalized to problems with different parameters (e.g., IBVP involving PDE of a different order or different boundary conditions).

The "Fokas method"[2] extends the idea of transform pairs to solving type II problems by constructing non-classical transform pairs based on the problem parameters. Since appropriate transform pairs may now be constructed for IBVPs with different parameters, this means that the Fokas method allows solving an entire class of IBVPs algorithmically in a manner similar to Figure 1 (Figure 2).

To characterize the class of IBVPs that can be solved by the Fokas method[1, p.9], we first define linearly independent boundary forms[1] $B_j : C^\infty[0,1] \to \mathbb{C}$ (where $C^\infty[0,1]$ denotes the class of real-valued functions differentiable for all orders on $[0,1]$) by

$$B_j\phi := \sum_{k=0}^{n-1} \left(b_{jk}\phi^{(k)}(0) + \beta_{jk}\phi^{(k)}(1) \right), \; j \in \{1,2,\ldots,n\}$$

where the boundary coefficients b_{jk}, $\beta_{jk} \in \mathbb{R}$. Furthermore, define

$$\Phi := \{\phi \in C^\infty[0,1] : \; B_j\phi = 0 \, \forall j \in \{1,2,\ldots,n\}\}$$

[1] These terms will be defined systematically in the report.

I think it would help the reader if you begin with a more informal description of the class of IBVP with some examples (I can help) for §1 and then in a new subsection of §2, clearly define the class in this way. This current presentation feels like an uneasy compromise between two different attempts to present the information, and is rather unsatisfactory as a result.

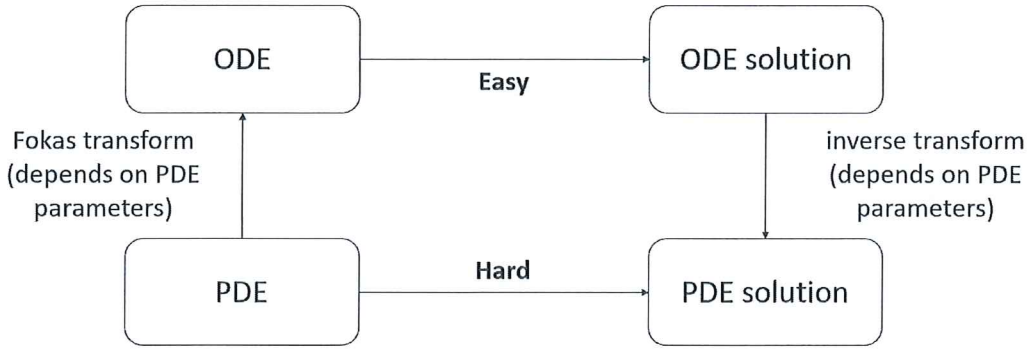

Figure 2: Solving type II IBVPs using non-classical transform pairs.

to be the set of smooth functions ϕ satisfying the homogeneous boundary conditions $B_j\phi = 0$ for all $j \in \{1, 2, \ldots, n\}$. Then the Fokas method allows solving any IBVP that can be written as

$$\begin{aligned} (\partial_t + aS)q(x,t) &= 0 \qquad && \forall (x,t) \in (0,1) \times (0,T) \\ q(x,0) &= q_0(x) && \forall x \in [0,1] \\ q(\cdot,t) &\in \Phi && \forall t \in [0,T], \end{aligned}$$

where a is a complex constant, S is a spatial differential operator (i.e., a differential operator in the spatial variable x) of order n, $(0,1) \times (0,T)$ is some domain, and $q_0 \in \Phi$ is arbitrary. (For the IBVP to be well-posed, a and n need to satisfy some additional constraints.) The first equation is a PDE relating a quantity q to its temporal and spatial rates of change $\partial_t[q]$ and $aS[q]$. The second equation is the initial condition where the temporal variable $t = 0$. The third equation corresponds to the spatial boundary conditions. An example of such IBVPs is the linear Schrödinger equation with zero potential, given by

$$ih\frac{\partial w}{\partial t} + \frac{h^2}{2m}\frac{\partial^2 w}{\partial x^2} = 0,$$

where $w(x,t)$ is the wave function, h is the Planck's constant, m is the mass of the particle, and kx describes the potential energy of the particle in the force field. Indeed, it can be written as

$$(\partial_t + aS)q(x,t) = 0$$

where $q = w$, $S = \frac{\partial^2}{\partial x^2}$, $a = \frac{h}{2mi}$.

The Fokas method allows solving the above class of IBVPs algorithmically, i.e., it provides the correct analytic solution determinisically for any IBVP that can be written in the above form. Using the Fokas method by hand, however, is laborious. Thus, the first goal of this project is to implement the Fokas method as a Julia package that allows mathematicians to quickly obtain the analytic solutions of complicated IBVPs. We choose the Julia language for its support for high-performance scientific computing as well as its popularity among the research community due to being open-source. On the other hand, solutions of complicated IBVPs usually involve infinite sums and series, and having numerical descriptions of them would help with understanding and visualizing them. Thus, the second goal of this project is to build a analytic-numerical integrator tailored to providing accurate numerical approximations of the analytic solutions of the above IBVPs.

1.2 Report outline

This report focuses on the first goal of the project, i.e., implementing the Fokas method in Julia.

For appropriate transform pair $f(x) = f_x(F)$ and $F(\lambda) = F_\lambda(f)$ found using the Fokas method, the solution to the above IBVP is given by[1, p.15]

$$q(x,t) = f_x(e^{-a\lambda^n t}F_\lambda(f)).$$

Thus, the key component of the implementation is to construct the transform pairs.

The first part of the report focuses on constructing the adjoint of a given homogeneous boundary condition in preparation for finding the Fokas transform pairs. This part of the report features revisions based on the supervisor's comments on report 1. The second part of the report focuses on implementing the Fokas transform pairs, featuring progress made since report 1.

We will conclude the report with a summary of the progress so far with respect to the plan in the project proposal. This will be followed by a brief discussion of plans for the next steps.

2 Constructing the adjoint of a homogeneous boundary condition

Given a linear map T from inner product spaces V to W (denoted as $T \in \mathcal{L}(V,W)$), the adjoint of T is the function $T^\star : W \to V$ with

$$\langle Tv, w\rangle = \langle v, T^\star w\rangle \tag{2.1}$$

for $v \in V$, $w \in W$, where $\langle\cdot,\cdot\rangle$ denotes inner products defined on V and W [3, p.204]. The "adjoint" mentioned in the first paragraph is an analogous notion defined for boundary value problems. The adjoint boundary condition associated with the adjoint problem relates to that of the original problem in a way that is important to making the Fokas transform pairs work as we would like them to. Thus, the first step in implementing the Fokas method is to construct the adjoint of a homogeneous boundary condition.

To this end, an algorithm to construct adjoint boundary conditions has been ~~created~~. The report will focus on the algorithm's development and implementation. We begin by introducing the preliminary materials on which the algorithm depends. To make the report self-contained, referenced definitions, theorems, and proofs are included in the report, with frequent supplies of the student's own remarks and proofs so as to make the results relevant to the construction algorithm.

We then present an outline of the construction algorithm, referencing definitions and theorems introduced in the preliminaries section. The outline is in higher-level, abstract form, with occasional pseudo-code illustrations when deemed necessary.

After that, we briefly discuss the implementation of the algorithm in Julia. The discussion will focus on the characterizations of key mathematical objects, notable features, and completion status.

2.1 Preliminaries

Definition 2.1. [4, p.81] A **linear differential operator** L of order n ($n > 1$) on interval $[a,b]$ is defined by

$$Lx = p_0 x^{(n)} + p_1 x^{(n-1)} + \cdots + p_{n-1}x' + p_n x,$$

where the p_k are complex-valued functions of class C^{n-k} on $[a,b]$ and $p_0(t) \neq 0$ on $[a,b]$.

Definition 2.2. [4, p.84] Given a linear differential operator L of order n as in Definition 2.1, the operator L^+ given by

$$L^+x = (-1)^n(\bar{p}_0x)^{(n)} + (-1)^{n-1}(\bar{p}_1x)^{(n-1)} + \cdots + \bar{p}_nx$$

(where $\bar{p}_k$ is the complex conjugate of p_k for $k \in \{0, \ldots, n\}$) is the **adjoint** of L.

Definition 2.3. [4, p.284] **Homogeneous boundary conditions** refer to a set of equations of the type

$$\sum_{k=1}^{n}(M_{jk}x^{(k-1)}(a) + N_{jk}x^{(k-1)}(b)) = 0 \quad (j = 1, \ldots, m) \tag{2.2}$$

where M_{jk}, N_{jk} are complex constants.

Definition 2.4. [4, p.284] A **homogeneous boundary value problem** concerns finding the solutions of

$$Lx = 0$$

on some interval $[a, b]$ which satisfy some homogeneous boundary conditions (Definition 2.3).

For a homogeneous boundary value problem π with linear differential operator L and some (homogeneous) boundary conditions, an adjoint problem π^+ involves the adjoint linear differential operator L^+ and some (homogeneous) adjoint boundary conditions. The adjoint boundary conditions are such that an equation similar to (2.1) exists for solutions of π and those of π^+, with the inner product $(\cdot)$ defined as $(u, v) := \int_a^b u\bar{v}\,dt$ for $u, v \in C^n$ on $[a, b]$. In the following sections, we seek to characterize the adjoint boundary conditions and describe an algorithm to construct them.

The construction algorithm depends on two important results, namely Green's formula and the boundary-form formula. Generally speaking, Green's formula allows characterizing a form, which, when used in the boundary-form formula, gives rise to the desired construction.

We begin with the Green's formula.

2.1.1 Green's formula

Theorem 2.5. *[4, p.284](Green's formula) For $u, v \in C^n$ on $[a, b]$,*

$$\int_{t_1}^{t_2}(Lu)\bar{v}\,dt - \int_{t_1}^{t_2}u(\overline{L^+v})\,dt = [uv](t_2) - [uv](t_1) \tag{2.3}$$

where $a \le t_1 < t_2 \le b$ and $[uv](t)$ is the form in $(u, u', \ldots, u^{(n-1)})$ and $(v, v', \ldots, v^{(n-1)})$ given by

$$[uv](t) = \sum_{m=1}^{n}\sum_{j+k=m-1}(-1)^j u^{(k)}(t)(p_{n-m}\bar{v})^{(j)}(t) \tag{2.4}$$

Using the form $[uv](t)$, we define an important $n \times n$ matrix B whose entries B_{jk} satisfy

$$[uv](t) = \sum_{j,k=1}^{n} B_{jk}(t)u^{(k-1)}(t)\bar{v}^{(j-1)}(t). \tag{2.5}$$

Note that

$$\begin{aligned}
[uv](t) &= \sum_{m=1}^{n} \sum_{j+k=m-1} (-1)^j u^{(k)}(t)(p_{n-m}\bar{v})^{(j)}(t) \\
&= \sum_{m=1}^{n} \sum_{j+k=m-1} (-1)^j u^{(k)}(t) \left(\sum_{\ell=0}^{j} \binom{j}{\ell} p_{n-m}^{(j-\ell)}(t)\bar{v}^{(\ell)}(t) \right) \\
&= \sum_{m=1}^{n} \sum_{k=0}^{m-1} (-1)^{m-1-k} u^{(k)}(t) \left(\sum_{\ell=0}^{m-1-k} \binom{m-1-k}{\ell} p_{n-m}^{(m-1-k-\ell)}(t)\bar{v}^{(\ell)}(t) \right) \\
&= \sum_{m=1}^{n} \sum_{k=1}^{m} (-1)^{m-k} \left(\sum_{\ell=0}^{m-k} \binom{m-k}{\ell} p_{n-m}^{(m-k-\ell)}(t)\bar{v}^{(\ell)}(t) \right) u^{(k-1)}(t) \quad \text{(shifting } k \text{ to } k+1) \\
&= \sum_{k=1}^{n} \sum_{m=k}^{n} (-1)^{m-k} \left(\sum_{\ell=0}^{m-k} \binom{m-k}{\ell} p_{n-m}^{(m-k-\ell)}(t)\bar{v}^{(\ell)}(t) \right) u^{(k-1)}(t).
\end{aligned}$$

To find B_{jk}, we need to extract the coefficients of $u^{(k-1)}\bar{v}^{(j-1)}$. We first note that, fixing m and k, when $\ell = j-1$, the coefficient of $\bar{v}^{(j-1)}$ is

$$\binom{m-k}{j-1} p_{n-m}^{(m-k-j+1)}(t).$$

To find the coefficient of $u^{(k-1)}\bar{v}^{(j-1)}$, we need to fix k and collect the above coefficient across all values of m. Since m goes up to n, $m-k$ goes up to $n-k$. Since $\ell \le m-k$, $\ell = j-1$ implies $j-1 \le m-k$. Thus, $m-k$ ranges from $j-1$ to $n-k$. Let $\ell' := m-k$, then $m = k+\ell'$, and the above equation becomes

$$\begin{aligned}
[uv](t) &= \sum_{k=1}^{n} \sum_{j=1}^{n} (-1)^{\ell'} \left(\sum_{\ell'=j-1}^{n-k} \binom{\ell'}{j-1} p_{n-(k+\ell')}^{(\ell'-(j-1))}(t) \right) \bar{v}^{(j-1)}(t) u^{(k-1)}(t) \\
&= \sum_{j,k=1}^{n} \left(\sum_{\ell=j-1}^{n-k} \binom{\ell}{j-1} p_{n-k-\ell}^{(\ell-j+1)}(t)(-1)^{\ell} \right) u^{(k-1)}(t)\bar{v}^{(j-1)}(t) \quad \text{(replace } \ell' \text{ by } l).
\end{aligned}$$

Thus,

$$B_{jk}(t) = \sum_{\ell=j-1}^{n-k} \binom{\ell}{j-1} p_{n-k-\ell}^{(\ell-j+1)}(t)(-1)^{\ell}.$$

We note that for $j+k > n+1$, or $j-1 > n-k$, l is undefined. This means that terms $u^{(k-1)}(t)\bar{v}^{(j-1)}(t)$ with $j+k > n+1$ does not exist in $[uv](t)$. Thus, $B_{jk}(t) = 0$. Also, for $j+k = n+1$, or $j-1 = n-k$,

$$B_{jk}(t) = \binom{j-1}{j-1} p_{j-1-(j-1)}^{(j-1-j+1)}(t)(-1)^{j-1} = (-1)^{j-1} p_0(t).$$

Thus, the matrix B has the form

$$B(t) = \begin{bmatrix} B_{11} & B_{12} & \cdots & \ddots & B_{1\,n-1} & p_0(t) \\ B_{21} & B_{22} & \cdots & \ddots & -p_0(t) & 0 \\ \vdots & \vdots & \ddots & & \vdots & \vdots \\ \vdots & \vdots & & \ddots & \vdots & \vdots \\ (-1)^{n-1}p_0(t) & 0 & \cdots & \cdots & 0 & 0 \end{bmatrix}. \tag{2.6}$$

We note that because $p_0(t) \neq 0$ on $[a,b]$ (as required in Definition 2.1), $B(t)$ is square with $\det B(t) = (p_0(t))^n \neq 0$ on $[a,b]$. Thus, $B(t)$ is nonsingular for $t \in [a,b]$.

Now we seek another matrix $\hat{B}$ that embodies both the characteristics of B and those of the interval $[a,b]$. This concerns writing the right-hand side of Green's formula in matrix form. We begin by introducing the following definitions.

Definition 2.6. [4, p.285] For vectors $f = (f_1, \ldots, f_k)$, $g = (g_1, \ldots, g_k)$, define the product

$$f \cdot g := \sum_{i=1}^{k} f_i \bar{g}_i.$$

Note that $f \cdot g = g^* f$ where * denotes conjugate transpose.

Definition 2.7. [4, p.285] A **semibilinear form** is a complex-valued function $\mathcal{S}$ defined for pairs of vectors $f = (f_1, \ldots, f_k)$, $g = (g_1, \ldots, g_k)$ satisfying

$$\mathcal{S}(\alpha f + \beta g, h) = \alpha \mathcal{S}(f,h) + \beta \mathcal{S}(g,h)$$
$$\mathcal{S}(f, \alpha g + \beta h) = \bar{\alpha} \mathcal{S}(f,g) + \bar{\beta} \mathcal{S}(f,h)$$

for any complex numbers α, β and vectors f, g, h.

We note that if

$$S = \begin{bmatrix} s_{11} & \cdots & s_{1k} \\ \vdots & & \vdots \\ s_{k1} & \cdots & s_{kk} \end{bmatrix},$$

then $Sf \cdot g$ is given by

$$\begin{aligned} \mathcal{S}(f,g) := Sf \cdot g &= \begin{bmatrix} s_{11} & \cdots & s_{1k} \\ \vdots & & \vdots \\ s_{k1} & \cdots & s_{kk} \end{bmatrix} \begin{bmatrix} f_1 \\ \vdots \\ f_k \end{bmatrix} \cdot \begin{bmatrix} g_1 \\ \vdots \\ g_k \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^{k} s_{1j} f_j \\ \vdots \\ \sum_{j=1}^{k} s_{kj} f_j \end{bmatrix} \cdot \begin{bmatrix} g_1 \\ \vdots \\ g_k \end{bmatrix} \\ &= \sum_{i=1}^{k} \left(\sum_{j=1}^{k} s_{ij} f_j \right) \bar{g}_i = \sum_{i,j=1}^{k} s_{ij} f_i \bar{g}_i. \end{aligned} \tag{2.7}$$

To see that this is a semibilinear form:

$$\mathcal{S}(\alpha f + \beta g, h) = \sum_{i,j=1}^{k} s_{ij}(\alpha f_j + \beta g_j)\bar{h}_i = \alpha \sum_{i,j=1}^{k} s_{ij} f_j \bar{h}_i + \beta \sum_{i,j=1}^{k} g_j \bar{h}_i$$

$$= \alpha Sf \cdot h + \beta Sg \cdot h = \alpha \mathcal{S}(f,h) + \beta \mathcal{S}(g,h);$$

and similarly,

$$\begin{aligned}\mathcal{S}(f, \alpha g + \beta h) &= \sum_{i,j=1}^{k} s_{ij} f_j \overline{(\alpha g_i + \beta h_i)} = \bar{\alpha} \sum_{i,j=1}^{k} s_{ij} f_j \bar{g}_i + \bar{\beta} \sum_{i,j=1}^{k} f_j \bar{h}_i \\ &= \bar{\alpha} Sf \cdot g + \bar{\beta} Sf \cdot h = \bar{\alpha} \mathcal{S}(f,g) + \bar{\beta} \mathcal{S}(f,h).\end{aligned}$$

Under a similar matrix framework, we see that $[uv](t)$ is a semibilinear form with matrix $B(t)$: Let $\vec{u} = (u, u', \ldots, u^{(n-1)})$ and $\vec{v} = (v, v', \ldots, v^{(n-1)})$. Then we have

$$\begin{aligned}[uv](t) &= \sum_{j,k=1}^{n} B_{jk}(t) u^{(k-1)}(t) \bar{v}^{(j-1)}(t) \quad \text{(by (2.5))} \\ &= \sum_{i,j=1}^{n} (B_{ij} u^{(j-1)} \bar{v}^{(i-1)})(t) \qquad (2.8) \\ &= (B\vec{u} \cdot \vec{v})(t) \quad \text{(by (2.7))} \\ &=: \mathcal{S}(\vec{u}, \vec{v})(t).\end{aligned}$$

With this notation, we can rewrite the right-hand side of Green's formula as a semibilinear form below:

$$
\begin{aligned}
[uv](t_2) - [uv](t_1) &= \sum_{j,k=1}^{n} B_{jk}(t_2) u^{(k-1)}(t_2) \bar{v}^{(j-1)}(t_2) - \sum_{j,k=1}^{n} B_{jk}(t_2) u^{(k-1)}(t_1) \bar{v}^{(j-1)}(t_1) \\
&= B(t_2)\vec{u}(t_2) \cdot \vec{v}(t_2) - B(t_1)\vec{u}(t_1) \cdot \vec{v}(t_1) \\
&= \begin{bmatrix} B_{11}(t_2) & \cdots & B_{1n}(t_2) \\ \vdots & & \vdots \\ B_{n1}(t_2) & \dots & B_{nn}(t_2) \end{bmatrix} \begin{bmatrix} u(t_2) \\ \vdots \\ u^{(n-1)}(t_2) \end{bmatrix} \cdot \begin{bmatrix} \bar{v}(t_2) \\ \vdots \\ \bar{v}^{(n-1)}(t_2) \end{bmatrix} - \\
&\quad \begin{bmatrix} B_{11}(t_1) & \cdots & B_{1n}(t_1) \\ \vdots & & \vdots \\ B_{n1}(t_1) & \dots & B_{nn}(t_1) \end{bmatrix} \begin{bmatrix} u(t_1) \\ \vdots \\ u^{(n-1)}(t_1) \end{bmatrix} \cdot \begin{bmatrix} \bar{v}(t_1) \\ \vdots \\ \bar{v}^{(n-1)}(t_1) \end{bmatrix} \\
&= \begin{bmatrix} -B_{11}(t_1) & \cdots & -B_{1n}(t_1) \\ \vdots & & \vdots \\ -B_{n1}(t_1) & \dots & -B_{nn}(t_1) \end{bmatrix} \begin{bmatrix} u(t_1) \\ \vdots \\ u^{(n-1)}(t_1) \end{bmatrix} \cdot \begin{bmatrix} \bar{v}(t_1) \\ \vdots \\ \bar{v}^{(n-1)}(t_1) \end{bmatrix} + \\
&\quad \begin{bmatrix} B_{11}(t_2) & \cdots & B_{1n}(t_2) \\ \vdots & & \vdots \\ B_{n1}(t_2) & \dots & B_{nn}(t_2) \end{bmatrix} \begin{bmatrix} u(t_2) \\ \vdots \\ u^{(n-1)}(t_2) \end{bmatrix} \cdot \begin{bmatrix} \bar{v}(t_2) \\ \vdots \\ \bar{v}^{(n-1)}(t_2) \end{bmatrix} \\
&= \begin{bmatrix} -B_{11}(t_1) & \cdots & -B_{1n}(t_1) & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & & \vdots \\ -B_{n1}(t_1) & \cdots & -B_{nn}(t_1) & 0 & \cdots & 0 \\ 0 & \cdots & 0 & B_{11}(t_2) & \cdots & B_{1n}(t_2) \\ \vdots & & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & B_{n1}(t_2) & \cdots & B_{nn}(t_2) \end{bmatrix} \begin{bmatrix} u(t_1) \\ \vdots \\ u^{(n-1)}(t_1) \\ u(t_2) \\ \vdots \\ u^{(n-1)}(t_2) \end{bmatrix} \cdot \begin{bmatrix} \bar{v}(t_1) \\ \vdots \\ \bar{v}^{(n-1)}(t_1) \\ \bar{v}(t_2) \\ \vdots \\ \bar{v}^{(n-1)}(t_2) \end{bmatrix} \\
&= \begin{bmatrix} -B(t_1) & 0_n \\ 0_n & B(t_2) \end{bmatrix} \begin{bmatrix} u(t_1) \\ \vdots \\ u^{(n-1)}(t_1) \\ u(t_2) \\ \vdots \\ u^{(n-1)}(t_2) \end{bmatrix} \cdot \begin{bmatrix} \bar{v}(t_1) \\ \vdots \\ \bar{v}^{(n-1)}(t_1) \\ \bar{v}(t_2) \\ \vdots \\ \bar{v}^{(n-1)}(t_2) \end{bmatrix} \\
&=: \hat{B} \begin{bmatrix} \vec{u}(t_1) \\ \vec{u}(t_2) \end{bmatrix} \cdot \begin{bmatrix} \vec{v}(t_1) \\ \vec{v}(t_2) \end{bmatrix}.
\end{aligned}
\tag{2.9}
$$

Recall that $\det(\lambda A) = \lambda^n \det(A)$ for $n \times n$ matrix A. Thus,

$$\det \hat{B} = \det(-B(t_1)) \det(B(t_2)) = (-1)^n \det B(t_1) \det B(t_2)$$

since $B(t_1)$ is $n \times n$. Since $B(t)$ is nonsingular for $t \in [a, b]$ (as shown before), $\hat{B}$ is nonsingular for $t_1, t_2 \in [a, b]$.

To recapitulate, given a linear differential operator L which involves functions $p_0, \dots, p_n$ and an interval $[a, b]$, from the Green's formula, we have defined a matrix B which depends on $p_0, \dots, p_n$ and

a matrix $\hat{B}$ which depends on B and $[a, b]$. These objects will be important in constructing an adjoint boundary condition using the boundary-form formula, which we now turn to.

2.1.2 Boundary-form formula

Before introducing the boundary-form formula, we need a set of definitions and results concerning boundary conditions.

Definition 2.8. [4, p.286] Given any set of $2mn$ complex constants M_{ij}, N_{ij} $(i = 1, \ldots, m;\ j = 1, \ldots, n)$, define m **boundary operators (boundary forms)** $U_1, \ldots, U_m$ for functions x on $[a, b]$, for which $x^{(j)}$ $(j = 1, \ldots, n-1)$ exists at a and b, by

$$U_i x = \sum_{j=1}^{n} (M_{ij} x^{(j-1)}(a) + N_{ij} x^{(j-1)}(b)) \quad (i = 1, \ldots, m) \tag{2.10}$$

U_i are **linearly independent** if the only set of complex constants $c_1, \ldots, c_m$ for which

$$\sum_{i=1}^{m} c_i U_i x = 0$$

for all $x \in C^{n-1}$ on $[a, b]$ is $c_1 = c_2 = \cdots = c_m = 0$.

Definition 2.9. [4, p.286] A **vectory boundary form** $U = (U_1, \ldots, U_m)$ is a vector whose components are boundary forms (Definition 2.8). When $U_1, \ldots, U_m$ are linearly independent, we say that U has rank m. We assume U has full rank below.

With the above definitions, we can now write a set of homogeneous boundary conditions (Definition 2.3) in matrix form. Define

$$\xi := \begin{bmatrix} x \\ x' \\ \vdots \\ x^{(n-1)} \end{bmatrix}; \quad U := \begin{bmatrix} U_1 \\ U_2 \\ \vdots \\ U_m \end{bmatrix}; \quad M := \begin{bmatrix} M_{11} & \cdots & M_{1n} \\ \vdots & & \vdots \\ M_{m1} & \cdots & M_{mn} \end{bmatrix}; \quad N := \begin{bmatrix} N_{11} & \cdots & N_{1n} \\ \vdots & & \vdots \\ N_{m1} & \cdots & N_{mn} \end{bmatrix}.$$

Then the set of homogeneous boundary conditions in (2.2) can be written as

$$Ux = M\xi(a) + N\xi(b).$$

Indeed:

$$\begin{aligned} M\xi(a) + N\xi(b) &= \begin{bmatrix} M_{11} & \cdots & M_{1n} \\ \vdots & & \vdots \\ M_{m1} & \cdots & M_{mn} \end{bmatrix} \begin{bmatrix} x(a) \\ x'(a) \\ \vdots \\ x^{(n-1)}(a) \end{bmatrix} + \begin{bmatrix} N_{11} & \cdots & N_{1n} \\ \vdots & & \vdots \\ N_{m1} & \cdots & N_{mn} \end{bmatrix} \begin{bmatrix} x(b) \\ x'(b) \\ \vdots \\ x^{(n-1)}(b) \end{bmatrix} \\ &= \begin{bmatrix} \sum_{j=1}^{n} M_{1j} x^{(j-1)}(a) \\ \vdots \\ \sum_{j=1}^{n} M_{mj} x^{(j-1)}(a) \end{bmatrix} + \begin{bmatrix} \sum_{j=1}^{n} N_{1j} x^{(j-1)}(b) \\ \vdots \\ \sum_{j=1}^{n} N_{mj} x^{(j-1)}(b) \end{bmatrix} \end{aligned}$$

$$= \begin{bmatrix} \sum_{j=1}^{n}(M_{1j}x^{(j-1)}(a) + N_{1j}x^{(j-1)}(b)) \\ \vdots \\ \sum_{j=1}^{n}(M_{mj}x^{(j-1)}(a) + N_{mj}x^{(j-1)}(b)) \end{bmatrix}$$

$$= \begin{bmatrix} U_1 x \\ \vdots \\ U_m x \end{bmatrix} = \begin{bmatrix} U_1 \\ U_2 \\ \vdots \\ U_m \end{bmatrix} x = Ux.$$

Based on the above, we propose another way to write U_x. Define the $m \times 2n$ matrix

$$(M : N) := \begin{bmatrix} M_{11} & \cdots & M_{1n} & N_{11} & \cdots & N_{1n} \\ \vdots & & \vdots & \vdots & & \vdots \\ M_{m1} & \cdots & M_{mn} & N_{m1} & \cdots & N_{mn} \end{bmatrix}.$$

Then $U_1, \ldots, U_m$ are linearly independent if and only if $\text{rank}(M : N) = m$, or equivalently, $\text{rank}(U) = m$. Moreover, Ux can also be written as

$$Ux = \begin{bmatrix} \sum_{j=1}^{n}(M_{1j}x^{(j-1)}(a) + N_{1j}x^{(j-1)}(b)) \\ \vdots \\ \sum_{j=1}^{n}(M_{mj}x^{(j-1)}(a) + N_{mj}x^{(j-1)}(b)) \end{bmatrix}$$

$$= \begin{bmatrix} M_{11} & \cdots & M_{1n} & N_{11} & \cdots & N_{1n} \\ \vdots & & \vdots & \vdots & & \vdots \\ M_{m1} & \cdots & M_{mn} & N_{m1} & \cdots & N_{mn} \end{bmatrix} \begin{bmatrix} x(a) \\ \vdots \\ x^{(n-1)}(a) \\ x(b) \\ \vdots \\ x^{(n-1)}(b) \end{bmatrix}$$

$$= (M : N) \begin{bmatrix} \xi(a) \\ \xi(b) \end{bmatrix}.$$

Having proposed a compact way to represent a set of homogeneous boundary conditions, we begin building our way to characterizing the notion of adjoint boundary condition. First, we need the notion of a complementary boundary form.

Definition 2.10. [4, p.287] If $U = (U_1, \ldots, U_m)$ is any boundary form with $\text{rank}(U) = m$ and $U_c = (U_{m+1}, \ldots, U_{2n})$ is any form with $\text{rank}(U_c) = 2n - m$ such that $(U_1, \ldots, U_{2n})$ has rank $2n$, then U and U_c are **complementary boundary forms**.

Note that extending $U_1, \ldots, U_m$ to $U_1, \ldots, U_{2n}$ is equivalent to embedding the matrix $(M : N)$ in a $2n \times 2n$ nonsingular matrix (recall that a square matrix is nonsingular if and only if it has full rank).

The characterization of adjoint boundary conditions is given by the boundary-form formula. The boundary-form formula is motivated by writing the right-hand side of Green's formula (2.3) as the linear combination of a boundary form U and a complementary form U_c. Before finally getting to it, we need the following propositions.

Proposition 2.11. *[4, p.287] In the context of the semibilinear form* (2.7)*, we have*

$$Sf \cdot g = f \cdot S^* g, \tag{2.11}$$

where S^* is the conjugate transpose of S.

Proof.

$$\begin{aligned}
Sf \cdot g &= \sum_{i,j=1}^{k} s_{ij} f_j \bar{g}_i \quad \text{(by (2.7))};\\
f \cdot S^* g &= \begin{bmatrix} f_1 \\ \vdots \\ f_k \end{bmatrix} \cdot \begin{bmatrix} \bar{s}_{11} & \cdots & \bar{s}_{k1} \\ \vdots & & \vdots \\ \bar{s}_{1k} & \cdots & \bar{s}_{kk} \end{bmatrix} \begin{bmatrix} g_1 \\ \vdots \\ g_k \end{bmatrix} \\
&= \begin{bmatrix} f_1 \\ \vdots \\ f_k \end{bmatrix} \cdot \begin{bmatrix} \sum_{j=1}^{k} \bar{s}_{j1} g_j \\ \vdots \\ \sum_{j=1}^{k} \bar{s}_{jk} g_j \end{bmatrix} \\
&= \sum_{i=1}^{k} f_i \cdot \overline{\left(\sum_{j=1}^{k} \bar{s}_{ji} g_j \right)} \\
&= \sum_{i=1}^{k} f_i \cdot \left(\sum_{j=1}^{k} s_{ji} \bar{g}_j \right) \\
&= \sum_{i,j=1}^{k} s_{ji} f_i \bar{g}_j = Sf \cdot g.
\end{aligned}$$

□

Proposition 2.12. *[4, p.287] Let $\mathcal{S}$ be the semibilinear form associated with a nonsingular matrix S. Suppose $\bar{f} := Ff$ where F is a nonsingular matrix. Then there exists a unique nonsingular matrix G such that if $\bar{g} = Gg$, then $\mathcal{S}(f, g) = \bar{f} \cdot \bar{g}$ for all f, g.*

Proof. Let $G := (SF^{-1})^*$, then

$$\begin{aligned}
\mathcal{S}(f, g) &= Sf \cdot g \\
&= S(F^{-1}F)f \cdot g \\
&= SF^{-1}(Ff) \cdot g \\
&= SF^{-1}\bar{f} \cdot g \\
&= \bar{f} \cdot (SF^{-1})^* g \quad \text{(by (2.11))} \\
&= \bar{f} \cdot G * g \\
&= \bar{f} \cdot \bar{g}.
\end{aligned}$$

To see that G is nonsingular, note that $\det G = \det((\overline{SF^{-1}})^T) = \det(\overline{SF^{-1}}) = \overline{\det(SF^{-1})} = \overline{\det(S)\det(F)^{-1}} \neq 0$ since S, F are nonsingular. □

Proposition 2.13. *[4, p.287] Suppose $\mathcal{S}$ is associated with the unit matrix E, i.e., $\mathcal{S}(f,g) = f \cdot g$. Let F be a nonsingular matrix such that the first j $(1 \leq j < k)$ components of $\bar{f} = Ff$ are the same as those of f. Then the unique nonsingular matrix G such that $\bar{g} = Gg$ and $\bar{f} \cdot \bar{g} = f \cdot g$ (as in Proposition 2.12) is such that the last $k-j$ components of $\bar{g}$ are linear combinations of the last $k-j$ components of g with nonsingular coefficient matrix.*

Proof. We note that for the condition on F to hold, F must have the form

$$\begin{bmatrix} E_j & 0_+ \\ F_+ & F_{k-j} \end{bmatrix}_{k \times k}$$

where E_j is the $j \times j$ identity matrix, 0_+ is the $j \times (k-j)$ zero matrix, F_+ is a $(k-j) \times j$ matrix, and F_{k-j} a $(k-j) \times (k-j)$ matrix. Let G be the unique nonsingular matrix in Proposition 2.12. Write G as

$$\begin{bmatrix} G_j & G_- \\ G_= & G_{k-j} \end{bmatrix}_{k \times k}$$

where $G_j, G_-, G_=, G_{k-j}$ are $j \times j, j \times (k-j), (k-j) \times j, (k-j) \times (k-j)$ matrices, respectively. By the definition of G,

$$f \cdot g = Ff \cdot Gg = \bar{f} \cdot Gg = G^* \bar{f} \cdot g = G^* F f \cdot g,$$

(where the third equality follows from a reverse application of (2.11) with $\bar{f}$ as f, G^* as S) which implies

$$G^* F = E_k.$$

Since

$$\begin{aligned} G^* F &= \begin{bmatrix} G_j^* & G_=^* \\ G_-^* & G_{k-j}^* \end{bmatrix} \begin{bmatrix} E_j & 0_+ \\ F_+ & F_{k-j} \end{bmatrix} \\ &= \begin{bmatrix} G_j^* + G_=^* F_+ & G_=^* F_{k-j} \\ G_-^* + G_{k-j}^* F_+ & G_{k-j}^* F_{k-j} \end{bmatrix} \\ &= \begin{bmatrix} E_j & 0_{j \times (k-j)} \\ 0_{(k-j) \times j} & E_{k-j} \end{bmatrix}. \end{aligned}$$

Thus, $G_=^* F_{k-j} = 0_+$, the $j \times (k-j)$ zero matrix. But $\det F = \det(E_j) \cdot \det(F_{k-j}) \neq 0$, so $\det F_{k-j} \neq 0$ and we must have $G_=^* = 0_+$, i.e., $G_= = 0_{(k-j) \times j}$. Thus, G is upper-triangular, and so $\det G = \det G_j \cdot \det G_{k-j} \neq 0$, which implies $\det G_{k-j} \neq 0$ and G_{k-j} is nonsingular. Hence,

$$\bar{g} = Gg = \begin{bmatrix} G_j & G_- \\ 0_{(k-j) \times j} & G_{k-j} \end{bmatrix} \begin{bmatrix} g_1 \\ \vdots \\ g_k \end{bmatrix}$$

where G_{k-j} is the nonsingular coefficient matrix such that

$$\begin{bmatrix} \bar{g}_{j-1} \\ \vdots \\ \bar{g}_k \end{bmatrix} = G_{k-j} \begin{bmatrix} g_{j-1} \\ \vdots \\ g_k \end{bmatrix}.$$

□

We are finally ready to introduce the boundary-form formula, the theorem central to the construction of adjoint boundary condition.

Theorem 2.14. *[4, p.288](Boundary-form formula) Given any boundary form U of rank m (Definition 2.8), and any complementary form U_c (Definition 2.10), there exist unique boundary forms U_c^+, U^+ of rank m and $2n-m$, respectively, such that*

$$[xy](b) - [xy](a) = Ux \cdot U_c^+ y + U_c x \cdot U^+ y. \tag{2.12}$$

If $\tilde{U}_c$ is any other complementary form to U, and $\tilde{U}_c^+, \tilde{U}^+$ the corresponding forms of rank m and $2n-m$, then

$$\tilde{U}^+ y = C^* U^+ y \tag{2.13}$$

for some nonsingular matrix C.

Remark 2.15. U^+ is the key object which will be defined later as an adjoint boundary condition to U.

Note that the existence of $[xy](t)$ implies that a linear differential operator is involved (see (2.4)). The matrices $\hat{B}$ and B in the proof also depend on this linear differential operator.

Also note that the second statement in the theorem reflects the fact that adjoint boundary conditions are unique only up to linear transformation. This is why when the need for rigor is greater than that for convenience, we take care to use "an" instead of "the" in referring to adjoint boundary condition.

Proof. Recall from (2.9) that the left-hand side of (2.12) can be considered as a semibilinear form $\mathcal{S}(f,g) = \hat{B}f \cdot g$ for vectors

$$f = \begin{bmatrix} x(a) \\ \vdots \\ x^{(n-1)}(a) \\ x(b) \\ \vdots \\ x^{(n-1)}(b) \end{bmatrix}, \quad g = \begin{bmatrix} y(a) \\ \vdots \\ y^{(n-1)}(a) \\ y(b) \\ \vdots \\ y^{(n-1)}(b) \end{bmatrix}$$

with the nonsingular matrix

$$\hat{B} = \begin{bmatrix} -B(a) & 0_n \\ 0_n & B(b) \end{bmatrix},$$

where B is as in (2.6). Recall from a previous discussion that

$$Ux = M\xi(a) + N\xi(b) = (M : N) \begin{bmatrix} \xi(a) \\ \xi(b) \end{bmatrix}$$

for M, N, ξ are as defined there. With the definition of f, we have $f = \begin{bmatrix} \xi(a) \\ \xi(b) \end{bmatrix}$ and thus

$$Ux = (M : N)f.$$

By Definition 2.10, $U_c x = (\tilde{M} : \tilde{N})f$ for two appropriate matrices $\tilde{M}, \tilde{N}$ for which

$$H = \begin{bmatrix} M & N \\ \tilde{M} & \tilde{N} \end{bmatrix}_{2n \times 2n}$$

has rank $2n$. Thus,

$$\begin{bmatrix} Ux \\ U_c x \end{bmatrix} = \begin{bmatrix} (M:N)f \\ (\tilde{M}:\tilde{N})f \end{bmatrix} = \begin{bmatrix} M & N \\ \tilde{M} & \tilde{N} \end{bmatrix} f = Hf.$$

By Proposition 2.12, there exists a unique $2n \times 2n$ nonsingular matrix J (in fact, with $S = \hat{B}$, $F = H$, $J = G$, and $G = (SF^{-1})^\star$ where * denotes the conjugate transpose, we have $J = (\hat{B}H^{-1})^\star$) such that $\mathcal{S}(f,g) = Hf \cdot Jg$. Let U^+, U_c^+ be such that

$$Jg = \begin{bmatrix} U_c^+ y \\ U^+ y \end{bmatrix},$$

then

$$[xy](b) - [xy](a) = \mathcal{S}(f,g) = Hf \cdot Jg = \begin{bmatrix} Ux \\ U_c x \end{bmatrix} \cdot \begin{bmatrix} U_c^+ y \\ U^+ y \end{bmatrix} = Ux \cdot U_c^+ y + U_c x \cdot U^+ y.$$

Thus, (2.12) holds.

The second statement in the theorem follows from Proposition 2.13 with Hf and Jg corresponding to f and g. □

2.1.3 Homogeneous boundary value problem and its adjoint

With the boundary-form formula, we are now able to fully characterize the notion of "adjoint" for boundary value problems. In this section, we begin by defining adjoint boundary condition and adjoint boundary value problem. We then explore some properties of these adjoints as relevant to the construction algorithm.

Definition 2.16. [4, p.288-89] For any boundary form U of rank m there is associated the homogeneous boundary condition

$$Ux = 0 \tag{2.14}$$

for functions $x \in C^{n-1}$ on $[a,b]$. If U^+ is any boundary form of rank $2n-m$ determined as in Theorem 2.14, then the homogeneous boundary condition

$$U^+ x = 0 \tag{2.15}$$

is an **adjoint boundary condition** to (2.14).

Putting together L, L^+ and U, U^+, we have the definition of adjoint boundary value problem.

Definition 2.17. [4, p.291] If U is a boundary form of rank m, the problem of finding solutions of

$$\pi_m : \; Lx = 0 \quad Ux = 0$$

on $[a,b]$ is a **homogeneous boundary value problem of rank** m. The problem

$$\pi_{2n-m}^+ : \; L^+ x = 0 \quad U^+ x = 0$$

on $[a,b]$ is the **adjoint boundary value problem to** π_m.

In connection with the notion of adjoint problem introduced after Definition 2.4, we now have the following property of the adjoint analogous to (2.1).

Proposition 2.18. *By Green's formula* (2.3) *and the boundary-form formula* (2.12),

$$(Lu, v) = (u, L^+v)$$

for all $u \in C^n$ *on* $[a, b]$ *satisfying* (2.14) *and all* $v \in C^n$ *on* $[a, b]$ *satisfying* (2.15).

Proof.

$$\begin{aligned}(Lu, v) - (u, L^+v) &= \int_a^b Lu\bar{v}\, dt - \int_a^b u(\overline{L^+v})\, dt \\ &= [uv](a) - [uv](b) \quad \text{(by Green's formula (2.3))} \\ &= Uu \cdot U_c^+ v + U_c u \cdot U^+ v \quad \text{(by boundary-form formula (2.12))} \\ &= 0 \cdot U_c^+ v + U_c u \cdot 0 \quad \text{(by (2.14) and (2.15))} \\ &= 0.\end{aligned}$$

□

Here, we treat the above result as a property of the adjoint after defining L^+ and constructing U^+. In some cases, it is treated as the definition of the adjoint problem, especially if the context wishes to introduce the adjoint problem using linear algebra. Historically, though, it is the adjoint problem that motivated the inner product characterization.

Now, we turn to one last result that would help us in the last step of the construction algorithm, namely checking whether an adjoint boundary condition is valid.

Just like how U is associated with two $m \times n$ matrices M, N, U^+ is associated with two $n \times (2n-m)$ matrices P, Q such that $(P^* : Q^*)$ has rank $2n - m$ and

$$U^+x = P^*\xi(a) + Q^*\xi(b). \tag{2.16}$$

The following theorem is motivated by characterizing the adjoint condition (2.15) in terms of the matrices M, N, P, Q. For our purpose, it provides a way to check whether the adjoint boundary condition found by the algorithm is indeed valid using the matrices M, N, P, Q.

Theorem 2.19. *[4, p.289] The boundary condition* $U^+x = 0$ *is adjoint to* $Ux = 0$ *if and only if*

$$MB^{-1}(a)P = NB^{-1}(b)Q \tag{2.17}$$

where $B(t)$ *is the* $n \times n$ *matrix associated with the form* $[xy](t)$ *(2.6).*

Proof. Let $\eta := (y, y', \ldots, y^{(n-1)})$, then $[xy](t) = B(t)\xi(t) \cdot \eta(t)$ by (2.8).

Suppose $U^+x = 0$ is adjoint to $Ux = 0$. By definition of adjoint boundary condition (2.15), U^+ is determined as in Theorem 2.14. But by Theorem 2.14, in determining U^+, there exist boundary forms U_c, U_c^+ of rank $2n - m$ and m, respectively, such that (2.12) holds.

Put

$$\begin{aligned}U_c x &= M_c\xi(a) + N_c\xi(b) \quad \operatorname{rank}(M_c : N_c) = 2n - m \\ U_c^+ y &= P_c^*\eta(a) + Q_c^*\eta(b) \quad \operatorname{rank}(P_c^* : Q_c^*) = m.\end{aligned}$$

Then by the boundary-form formula (2.12),

$$B(b)\xi(b) \cdot \eta(b) - B(a)\xi(a) \cdot \eta(a) = (M\xi(a) + N\xi(b)) \cdot (P_c^*\eta(a) + Q_c^*\eta(b)) +$$

$$(M_c\xi(a) + N_c\xi(b)) \cdot (P^*\eta(a) + Q^*\eta(b)).$$

By (2.11),

$$M\xi(a) \cdot P_c^*\eta(a) = P_cM\xi(a) \cdot \eta(a).$$

Thus,

$$B(b)\xi(b) \cdot \eta(b) - B(a)\xi(a) \cdot \eta(a) = (P_cM + PM_c)\xi(a) \cdot \eta(a) + (Q_cM + QM_c)\xi(a) \cdot \eta(b)$$
$$(P_cN + PN_c)\xi(b) \cdot \eta(a) + (Q_cN + QN_c)\xi(b) \cdot \eta(b).$$

Thus, we have

$$P_cM + PM_c = -B(a) \qquad P_cN + PN_c = 0_n$$
$$Q_cM + QM_c = 0_n \qquad Q_cN + QN_c = B(b).$$

Since $\det B(t) \neq 0$ on $t \in [a, b]$, $B^{-1}(a)$, $B^{-1}(b)$ exist, and thus

$$\begin{bmatrix} -B^{-1}(a)P_c & -B^{-1}(a)P \\ B^{-1}(b)Q_c & B^{-1}(b)Q \end{bmatrix} \begin{bmatrix} M & N \\ M_c & N_c \end{bmatrix} = \begin{bmatrix} E_n & 0_n \\ 0_n & E_n \end{bmatrix}.$$

Recall that $\begin{bmatrix} M & N \\ M_c & N_c \end{bmatrix}$ has full rank, which means that it is nonsingular (Definition 2.10). Thus, the two matrices on the left are inverses of each other. So we also have

$$\begin{bmatrix} M & N \\ M_c & N_c \end{bmatrix} \begin{bmatrix} -B^{-1}(a)P_c & -B^{-1}(a)P \\ B^{-1}(b)Q_c & B^{-1}(b)Q \end{bmatrix} = \begin{bmatrix} E_m & 0_+ \\ 0_- & E_{2n-m} \end{bmatrix}.$$

Therefore,

$$-MB^{-1}(a)P + NB^{-1}(b)Q = 0_+,$$

which is (2.17).

Conversely, let U_1^+ be a boundary form of rank $2n - m$ such that

$$U_1^+y = P_1^*\eta(a) + Q_1^*\eta(b)$$

for appropriate P_1^*, Q_1^* with $\operatorname{rank}(P_1^* : Q_1^*) = 2n - m$. Suppose

$$MB^{-1}(a)P_1 = NB^{-1}(b)Q_1 \tag{2.18}$$

holds.

By the fundamental theorem of linear maps [3, p.63], if V is finite-dimensional and $T \in \mathcal{L}(V, W)$, then $\dim\ker T = \dim V - \dim\operatorname{range} T$. Suppose A is a $n \times k$ matrix, then $A \in \mathcal{L}(\mathbb{R}^n, \mathbb{R}^k)$. Thus, in a homogeneous system of linear equations $Ax = 0$, we have $\dim\ker A = \dim A - \dim\operatorname{range} A$. That is, the dimension of solution space $\ker A$ is the difference between the number of unknown variables and the rank of the coefficient matrix, or $\dim\operatorname{range} A$. Therefore, letting u be a $2n \times 1$ vector, there exist exactly $2n - m$ linearly independent solutions of the homogeneous linear system $(M : N)_{m\times 2n}u = 0$. By (2.18),

$$MB^{-1}(a)P_1 - NB^{-1}(b)Q = 0,$$

and thus

$$(M : N)_{m\times 2n}\begin{bmatrix} B^{-1}(a)P_1 \\ -B^{-1}(b)Q_1 \end{bmatrix}_{2n\times(2n-m)} = 0_{m\times(2n-m)}.$$

So the $2n-m$ columns of the matrix

$$H_1 := \begin{bmatrix} B^{-1}(a)P_1 \\ -B^{-1}(b)Q_1 \end{bmatrix}$$

are solutions of this system. Since $\operatorname{rank}(P_1^* : Q_1^*) = 2n-m$,

$$\operatorname{rank}\begin{bmatrix} P_1 \\ Q_1 \end{bmatrix} = 2n-m.$$

Since $B(a)$, $B(b)$ are nonsingular, $\operatorname{rank}(H_1) = 2n-m$.

If $U^+x = P^*\xi(a) + Q^*\xi(b) = 0$ is a boundary condition adjoint to $Ux = 0$, then the matrix

$$\begin{bmatrix} -B^{-1}(a)P_c & -B^{-1}(a)P \\ B^{-1}(b)Q_c & B^{-1}(b)Q \end{bmatrix}_{2n\times 2n}$$

is nonsingular (because it has inverse $\begin{bmatrix} M & N \\ M_c & N_c \end{bmatrix}$), i.e., it has full rank. Thus, if

$$H = \begin{bmatrix} -B^{-1}(a)P \\ B^{-1}(b)Q \end{bmatrix}_{n\times(2n-m)},$$

then $\operatorname{rank}(H) = 2n-m$. Therefore, by (2.17), the $2n-m$ columns of H also form $2n-m$ linearly independent solutions of $(M : N)u = 0$, as in the case of H_1. Hence, there exists a nonsingular $(2n-m)\times(2n-m)$ matrix A such that $H_1 = HA$ (change of basis in the solution space). Thus we have

$$\begin{bmatrix} B^{-1}(a)P_1 \\ -B^{-1}(b)Q_1 \end{bmatrix} = H_1 = HA = \begin{bmatrix} B^{-1}(a)PA \\ -B^{-1}(b)QA \end{bmatrix},$$

or $P_1 = PA$, $Q_1 = QA$. Thus,

$$U_1^+y = P_1^*\eta(a) + Q_1^*\eta(b) = A^*P^*\eta(a) + A^*Q^*\eta(b) = A^*U^+y.$$

Since $A^\star$ is a linear map, $U^+y = 0$ implies $U_1^+y = A^*U^+y = 0$. Since $A^\star$ is nonsingular, $A^{\star^{-1}}$ is also a linear map, and $A^{\star^{-1}}U_1^+y = U^+y$. Thus, $U_1^+y = 0$ implies $U^+y = A^{\star^{-1}}U_1^+y = 0$. Therefore, $U^+y = 0$ if and only if $U_1^+y = 0$. Since $U^+y = 0$ is adjoint to $Ux = 0$, $U_1^+y = 0$ is adjoint to $Ux = 0$. □

To recapitulate, the boundary-form formula (Theorem 2.14) gives us the existence and construction of adjoint boundary condition, and Theorem 2.19 gives us a way to check whether a proposed adjoint boundary condition is valid. Now, we are ready to propose a construction algorithm.

2.2 Algorithm outline

Suppose we are given a homogeneous boundary value problem on $[a, b]$,

$$Lx = 0 \quad Ux = 0,$$

where L is a linear differential operator with order n (Definition 2.1) and U is a vector boundary form $U = (U_1, \ldots, U_n)$ (Definition 2.9). (For the purpose of the project, we are only interested in cases where $m = n$.) We seek to construct a valid adjoint boundary condition $U^+x = 0$ (Definition 2.16).

2.2.1 Check input

We first check that $U_1, \ldots, U_n$ are linearly independent. As noted in a previous discussion, write

$$Ux = M\xi(a) + N\xi(b),$$

then it suffices to check whether $\text{rank}(M : N) = n$. U would be considered an invalid input if $\text{rank}(M : N) \neq n$.

2.2.2 Find U^+

Recall from Definition 2.10 that extending $U_1, \ldots, U_n$ to $U_1, \ldots, U_{2n}$ (where $(U_{n+1}, \ldots, U_{2n})$ is a complementary boundary form U_c) is equivalent to embedding $(M : N)$ in a $2n \times 2n$ nonsingular matrix (where the newly added rows constitute $(\tilde{M}, \tilde{N})$ associated with U_c). We construct this $2n \times 2n$ nonsingular matrix from the identity matrix E_{2n} and $(M : N)$ as follows. We append the rows of E_{2n} one by one to $(M : N)$ and discard any row that does not make the rank of the resulting matrix increase, as shown below.

Algorithm 1: Algorithm to find U_c.

Data: $(M : N)_{n\times 2n}$ with rank n, E_{2n} with rank $2n$
Result: A $2n \times 2n$ matrix with rank $2n$ where the first n rows are $(M : N)$
begin
  mat ⟵ $(M : N)$;
  for *i in range(nrow(E))* **do**
    mat1 ⟵ vcat(mat, E[i,]) ; ▷ `vcat := vertical concatenation, or joining two matrices vertically`
    if *rank(mat1) == rank(mat)+1* **then**
      mat ⟵ mat1
    else
      E ⟵ E[-i,]
  return *vcat(mat, E)*

With this algorithm, we run the risk of adding a row that is linearly independent of the other rows, but * only just. Can we find a way around this

* Not sure exactly what the metric is to determine closeness. Any ideas? Inner product of some kind?

The output from the above algorithm is a $2n \times 2n$ matrix of the form

$$\begin{bmatrix} M & N \\ E' & E'' \end{bmatrix}$$

where the rows of E', E'' are the first n entries and the last n entries of the retained rows of E_{2n}, respectively. We identify this matrix with the matrix

$$H = \begin{bmatrix} M & N \\ \tilde{M} & \tilde{N} \end{bmatrix}$$

in Theorem 2.14 (where $\tilde{M}, \tilde{N}$ are associated with the complementary boundary form $U_c x = \tilde{M}\xi(a) + \tilde{N}\xi(b)$).

Recall that the linear differential operator L is characterized by the functions $p_0, \ldots, p_n$ and the interval $[a, b]$. Let B be as in (2.6) which depends on $p_0, \ldots, p_n$. Construct $\hat{B}$ from B and $[a, b]$ as in Theorem 2.14. Let $J := (\hat{B}H^{-1})^*$. By Theorem 2.14 and Proposition 2.12, the matrix J is of the form

$$J = \begin{bmatrix} M' & N' \\ \tilde{M}' & \tilde{N}' \end{bmatrix}$$

where $\tilde{M}', \tilde{N}'$ are associated with an adjoint U^+ and M', N' with its complement U_c^+. Thus, we can identify $(P^\star : Q^\star)$ with the last n rows of J. That is, identify $P^\star$ with $\tilde{M}'$, the lower-left $n \times n$ submatrix of J, and $Q^\star$ with $\tilde{N}'$, the lower-right $n \times n$ submatrix of J. Define U^+ by

$$U^+ x = P^\star \xi(a) + Q^\star \xi(b),$$

then we have found an adjoint U^+ to U.

2.2.3 Check U^+

By Theorem 2.19, with

$$Ux = M\xi(a) + N\xi(b), \quad U^+ x = P^\star \xi(a) + Q^\star \xi(b),$$

we can check whether the U^+ found above is indeed a valid adjoint to U by checking

$$MB^{-1}(a)P = NB^{-1}(b)Q$$

where B is as in (2.6).

2.3 Implementation

The above construction algorithm has been implemented in Julia 0.6.4. The main functions and unit tests currently occupy 600 lines of code each. Key structs (or types, objects as in more traditional object-oriented programming languages) include linear differential operator and vector boundary form. A linear differential operator (Definition 2.1) is characterized by a list of functions $p_0, \ldots, p_n$ satisfying some conditions and a tuple (a, b), where a, b are the endpoints of the interval $[a, b]$. A vector boundary form (Definition 2.9) is characterized by two $n \times n$ matrices M, N satisfying $\operatorname{rank}(M : N) = n$. Since these objects are user-defined, appropriate internal checks have been implemented to ensure the validity of user input.

Systematic unit tests have been written for the algorithm. They examine whether the algorithm can correctly construct the objects (e.g., linear differential operators, vector boundary forms, and adjoint boundary conditions) when inputs are valid, and correctly throw the pre-defined errors when inputs are invalid. For each order n of the linear differential operator L where n ranges from 1 to 10, each main functionality of the algorithm is subjected to 10 tests with randomized conditions. The algorithm has passed all tests written so far.

To use the algorithm, the user would first need to define a linear differential operator L, a vector boundary form U, and a matrix containing derivatives of the functions $p_0, \ldots, p_n$ in the definition of L. Then, finding a valid adjoint boundary condition is as simple as passing these objects to a function. To maximize workflow transparency, the implementation also contains many other functions that output various objects involved in the algorithm.

To enhance user experience, the implementation also comes with a symbolic math feature, which allows the user to keep track of various functions in the form of symbolic expression. Without this feature, a function defined as $f(x) = 10.52x^3 + 3.7x + 1$ will be stored as a general method f in Julia without information for the user as to what it actually is. With symbolic expression, f will be stored as $10.52x^3 + 3.7x + 1$, verbatim. The workflow involving symbolic expressions is parallel to that of Julia functions, ensuring that the user is able to view the symbolic expression of any output whenever desired.

3 The Fokas transform pair

Having constructed the adjoint boundary condition, we are ready to implement the Fokas transform pair F and f[1, p.10]. We will introduce the relevant definitions and discuss their implementation. Compared to the algorithm to construct adjoint boundary conditions, the work involved in this section concerns more implementation than theory. The discussion in this section will focus on main ideas only; intricacies of the implementation will go into the package documentation.

3.1 Preliminaries

Let $\alpha = e^{2\pi i/n}$. Let $M^+(\lambda)$, $M^-(\lambda)$ be the $n \times n$ matrices given by

$$M^+(\lambda) := \sum_{r=0}^{n-1} (-i\alpha^{k-1}\lambda)^r b^\star_{jr}$$
$$M^-(\lambda) := \sum_{r=0}^{n-1} (-i\alpha^{k-1}\lambda)^r \beta^\star_{jr},$$

where $b^\star_{jr}$, $\beta^\star_{jr}$ are the j,r-entry of $P^\star$ and $Q^\star$ in (2.16), indexed from 0 to $n-1$. Let M be the $n \times n$ matrix given by

$$M_{kj}(\lambda) := M^+_{kj}(\lambda) + M^-_{kj}(\lambda) e^{-\alpha^{k-1}\lambda}.$$

Let $\Delta(\lambda) := \det M(\lambda)$, which is an exponential polynomial by the definition of $M(\lambda)$. Let X^{lj} be the $(n-1) \times (n-1)$ submatrix of the block matrix $\mathbb{M} := \begin{bmatrix} M & M \\ M & M \end{bmatrix}$ where X^{lj}_{11} is $\mathbb{M}_{\ell+1,j+1}$.

For $\lambda \in \mathbb{C}$ such that $\Delta(\lambda) \neq 0$, define

$$F^+_\lambda(f) = \frac{1}{2\pi\Delta(\lambda)} \sum_{\ell=1}^{n} \sum_{j=1} (-1)^{(n-1)(\ell+j)} \det X^{lj}(\lambda) M^+_{1j}(\lambda) \int_0^1 e^{-i\alpha^{\ell-1}\lambda x} f(x)\, dx,$$
$$F^-_\lambda(f) = \frac{-e^{-i\lambda}}{2\pi\Delta(\lambda)} \sum_{\ell=1}^{n} \sum_{j=1} (-1)^{(n-1)(\ell+j)} \det X^{lj}(\lambda) M^-_{1j}(\lambda) \int_0^1 e^{-i\alpha^{\ell-1}\lambda x} f(x)\, dx.$$

Let 5ϵ be the infimum of the distances between every two zeroes of $\Delta(\lambda)$ (which exists by the theory of exponential polynomials[5]). Let ∂S denote boundary of set S. Let $\operatorname{cl} S$ denote the closure of set S. Let $\mathbb{C}^\pm$ denote the upper and lower halves of the complex plane, respectively. Define the contours

$$\Gamma^\pm_a := \partial(\{\lambda \in \mathbb{C}^\pm : \operatorname{Re}(a\lambda^n) > 0\} \setminus \bigcup_{\substack{\sigma\in\mathbb{C};\\ \Delta(\sigma)=0}} D(\sigma, 2\epsilon)) \quad (D \text{ for disk}),$$
$$\Gamma_a := \Gamma^+_a \cup \Gamma^-_a,$$
$$\Gamma^+_0 := \bigcup_{\substack{\sigma\in\operatorname{cl}\mathbb{C}^+;\\ \Delta(\sigma)=0}} C(\sigma, \epsilon) \quad (C \text{ for circle}),$$
$$\Gamma^-_0 := \bigcup_{\substack{\sigma\in\mathbb{C}^-;\\ \Delta(\sigma)=0}} C(\sigma, \epsilon),$$

$$\Gamma_0 := \Gamma_0^+ \cup \Gamma_0^-,$$
$$\Gamma := \Gamma_0 \cup \Gamma_a.$$

The Fokas transform pair is given by

$$F_\lambda : f(x) \mapsto F(\lambda) \quad F_\lambda(f) = \begin{cases} F_\lambda^+(f) & \text{if } \lambda \in \Gamma_0^+ \cup \Gamma_a^+, \\ F_\lambda^-(f) & \text{if } \lambda \in \Gamma_0^- \cup \Gamma_a^-, \end{cases}$$
$$f_x : F(\lambda) \mapsto f(x) \quad f_x(F) = \int_\Gamma e^{i\lambda x} F(\lambda)\, d\lambda,\ x \in [0,1].$$

3.2 Implementation

The above algorithm has been implemented in Julia 0.6.4 in approximately 700 lines of code. Systematic unit tests are yet to be written. Most of the work done so far has been dedicated to implementing the contour Γ, which will thus be the main subject in this section.

We note that Γ_0^+, Γ_0^- are the unions of ϵ-circles around the zeroes of $\Delta(\lambda)$. On the other hand, Γ_a^+, Γ_a^- are boundaries of the domains $\{\lambda \in \mathbb{C}^+ : \mathrm{Re}(a\lambda^n) > 0\}$ and $\{\lambda \in \mathbb{C}^- : \mathrm{Re}(a\lambda^n) > 0\}$ avoiding the 2ϵ-disks around the zeroes of $\Delta(\lambda)$. The Fokas transform pair involves integrals over these contours. Thus, to implement the transform pair, we need to characterize the contours using line segments, or list of points such that, if connected in the order in which they are listed, form the line segments in the order in which they would be integrated over in the contour integrals.

3.2.1 Contour tracing

Given a complex constant a, we find the boundary of

$$\{\lambda \in \mathbb{C} : \mathrm{Re}(a\lambda^n) > 0\}.$$

For complex number z, define $\theta_z := \arg(z)$ and $r_z := |z|$. Let $\lambda \in \mathbb{C}$. Then

$$a\lambda^n = r_a e^{i\theta_a} \cdot r_\lambda^n e^{ni\theta_\lambda} = r_a r_\lambda^n e^{i(\theta_a + n\theta_\lambda)} = r_a r_\lambda^n (\cos(\theta_a + n\theta_\lambda) + i\sin(\theta_a + n\theta_\lambda)).$$

Thus,

$$\mathrm{Re}(a\lambda^n) = r_a r_\lambda^n \cos(\theta_a + n\theta_\lambda),$$

where

$$\mathrm{Re}(a\lambda^n) > 0 \iff \cos(\theta_a + n\theta_\lambda) > 0.$$

But

$$\cos(\theta_a + n\theta_\lambda) > 0 \iff \theta_a + n\theta_\lambda \in \bigcup_{k\in\mathbb{Z}} (2\pi k - \frac{\pi}{2}, 2\pi k + \frac{\pi}{2}),$$

and so

$$\theta_\lambda \in \bigcup_{k\in\mathbb{Z}} \left(\frac{2\pi k - \frac{\pi}{2} - \theta_a}{n}, \frac{2\pi k + \frac{\pi}{2} - \theta_a}{n} \right).$$

For example, when $a = -i$ and $n = 3$,

$$\theta_\lambda \in \bigcup_{k\in\mathbb{Z}} \left(\frac{2\pi k - \frac{\pi}{2} - (-\frac{\pi}{2})}{3}, \frac{2\pi k + \frac{\pi}{2} - (-\frac{\pi}{2})}{3} \right) = \bigcup_{k\in\mathbb{Z}} \left(\frac{2\pi k}{3}, \frac{2\pi k + \pi}{3} \right).$$

This is misleading. There are infinitely many zeroes of Δ. However, in any finite region there are only finitely many. This is proved in Longer. Note that $\Delta(\lambda) = 5 - e^{-\lambda}$ is a prototypical example.

Thus, in $[0, 2\pi)$,

$$\theta_\lambda = \arg(\lambda) \in \bigcup_{k\in\{0,1,2\}} \left(\frac{2\pi k}{3}, \frac{2\pi k + \pi}{3}\right) = (0, \frac{\pi}{3}) \cup (\frac{2\pi}{3}, \pi) \cup (\frac{4\pi}{3}, \frac{5\pi}{3}).$$

Figure 3 shows some simulations of the domain $\{\lambda \in \mathbb{C} : \mathrm{Re}(a\lambda^n) > 0\}$.

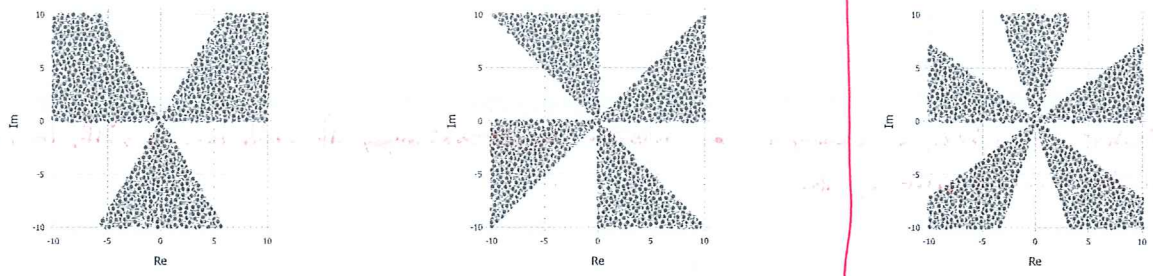

Figure 3: Simulation of sectors in the domain $\{\lambda \in \mathbb{C} : \mathrm{Re}(a\lambda^n) > 0\}$ for $a = -i$ and $n = 3, 4, 5$, respectively, by point sampling.

Now, to find the contour Γ, suppose we have been given the (finitely many) zeroes of $\Delta(\lambda)$ (we will attempt to approximate the zeroes algorithmically; for now, we assume they are user input). Each sector in the boundary is characterized by the angles of the two rays that mark its boundaries, the starting ray and the ending ray. The contour Γ is built from the boundaries of these sectors with possible deformations to include the contours encircling the zeroes. Figures 4 and 5 show the contour Γ for some examples.

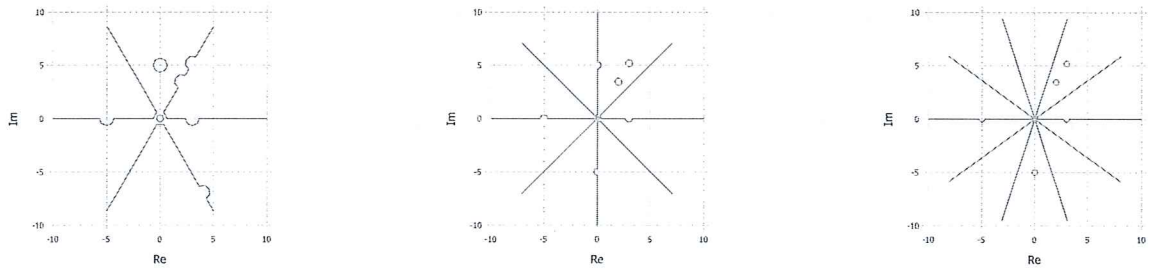

Figure 4: Plots of Γ for $a = -i$ and $n = 3, 4, 5$, respectively, with zeroes of $\Delta(\lambda)$ at $3 + \sqrt{3}i$, $2 + 2\sqrt{3}i$, $0 + 0i$, $0 + 5i$, $0 - 5i$, 3, -5, and $4 - 4\sqrt{3}i$.

Note that each sector will be assigned to Γ_a^+ or Γ_a^- depending on whether it belongs to the upper or lower half plane. If a sector overlaps with the real line (Figure 5), we divide it into the upper/lower half planes, which can then be assigned to Γ_a^+, Γ_a^-, respectively.

Let d be a sufficiently large real number chosen to represent infinity. For each sector, we initialize its boundary contour by a path consisting of three points, namely a point on the ending ray with distance d from the origin, the origin, and the a point on the starting ray with norm d (this would be the contour if no zero of $\Delta(\lambda)$ were on the sector boundary). For each zero, if it is on the boundary of some sector, we deform the sector's boundary contour to include a small n-gon of radius ϵ around the zero, where ϵ is taken to be 1/3 of the minimum of the pairwise distances between zeroes and the

1/5 earlier. Why change?

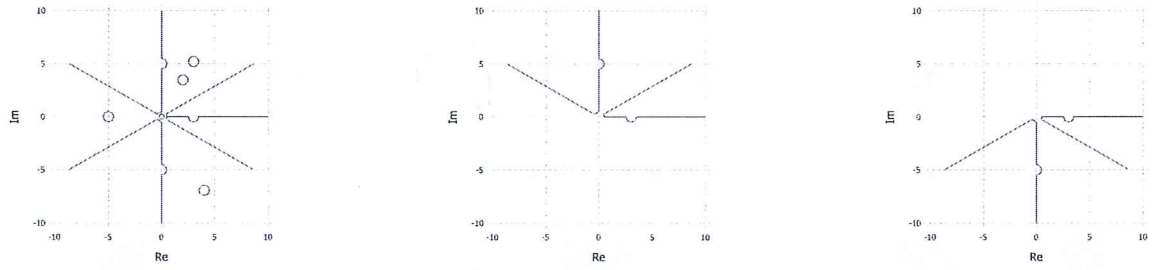

Figure 5: Plots of Γ, Γ_a^+, and Γ_a^- for $a = 1$ and $n = 3$, respectively, with the same zeroes as in Figure 4. Better to give an example for never ~~[illegible]~~ If n odd then $a \in i\mathbb{R}$, so no sector can be split by $\mathbb{R}$.

distances between each zero and the starting and ending rays of each sector (since $\Delta(\lambda)$ have finitely many zeroes, ϵ exists). If the zero is exterior to all sectors, we add an n-gon contour around it to $\Gamma_0^{\pm}$, depending on whether the zero is in the upper or lower half plane. If the zero is interior to any sector, we ignore it, since its contribution to the contour integral will be cancelled out.

After we have looped through all zeroes in this process, Γ_a would contain all sector boundary contours now appropriately deformed, and Γ_0 would contain all the n-gon contours encircling isolated zeroes. Although not reflected in the figures, care is taken to ensure that the points in Γ are listed in the order in which the contour will be integrated over. Explain how your choice of ϵ avoids ~~[illegible]~~ "overwriting" one deformation with another.

3.2.2 Chebyshev polynomial approximation

As previously mentioned, we need to integrate complicated functions and find zeroes of exponential polynomials such as $\Delta(\lambda)$ at future steps in the implementation. Dealing with these complicated functions is difficult, but we may translate it into an easier problem by approximating these complicated functions on given intervals using functions we are more familiar with. To this end, we implement a method to approximate functions using Chebyshev polynomials.

The Chebyshev polynomials are given by

$$T_n(x) = \begin{cases} \cos(n \cdot \arccos(x)) & |x| \leq 1 \\ \frac{1}{2}\left((x - \sqrt{x^2-1})^n + (x + \sqrt{x^2-1})^n\right) & |x| > 1. \end{cases}$$

Given a function f on the interval $[a, b]$, we approximate f using Chebyshev polynomial expansion. The Chebyshev approximation is best on $[-1, 1]$. Thus, we scale $[a, b]$ to $[-1, 1]$ via $g : [a, b] \to [-1, 1]$ given by $g(x) = 2(x - a)/(b - a) - 1$, obtain the Chebyshev coefficients $(a_n)_{n=0}^N$ there, and write the approximation of f on $[a, b]$ as

$$f \sim \sum_{n=0}^{N} a_n \cdot T_n(g(x)),$$

which is a finite sum of Chebyshev polynomials.

4 Next steps

With regard to the semester 1 plan in the project proposal, the progress has been on track so far. The immediate next step would be to finish implementing the Fokas method, with unit tests for contour

implementation and method to approximate the zeroes of exponential polynomials. Further next steps include organizing the implementation into a package and writing a documentation for the package. These tasks should ideally be completed before the start of semester 2, when focus will be placed on the second goal of the project as well as preparing project deliverables such as the thesis.

References

[1] D. A. Smith and A.S. Fokas. Evolution PDEs and augmented eigenfunctions. Half-line. *Journal of Spectral Theory*, 6(1):185–213, 2016.

[2] Athanassios S. Fokas. *A Unified Approach to Boundary Value Problems.* 2008.

[3] Sheldon Axler. *Linear Algebra Done Right.* Springer, third edition, 1997.

[4] Earl A. Coddington and Norman Levinson. *Theory of Ordinary Differential Equations.* McGraw-Hill Publishing, New York, 1977.

[5] R.E.Langer. The zeros of exponential sums and integrals. *Amer. Math. Soc.*, (1):213–239, 1930.