

Capstone report 1
Algorithmic solution of high order partial differential equations
in Julia via the Fokas transform method

Linfan XIAO
Supervisor: Prof. David Smith

Contents

1 Introduction	2
2 Preliminaries	2
2.1 Green's formula	3
2.2 Boundary-form formula	7
2.3 Homogeneous boundary value problem and its adjoint	12
3 Algorithm outline	15
3.1 Check input	16
3.2 Find U^+	16
3.3 Check U^+	17
4 Implementation	17
5 Next steps	18

1 Introduction

Clearly describe a class of problems that can be solved using Fokas method

Solving evolution partial differential equations typically requires ad-hoc methods and special treatments. The recently discovered “Fokas method” [1][2] allows solving many of these equations algorithmically. The first goal of the project concerns implementing the Fokas method in the Julia mathematical programming language. The first steps in the implementation involves constructing a valid adjoint boundary condition from a given homogeneous boundary condition.

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

$$\langle Tv, w \rangle = \langle v, T^*w \rangle \quad (1.1)$$

for $v \in V$, $w \in W$, where $\langle \cdot \rangle$ ~~is an~~ *denotes* inner product defined on V [3, p.204]. The “adjoint” mentioned in the first paragraph is an analogous notion defined for boundary value problems. Importantly, the adjoint problem mirrors many properties of the original problem. In particular, we may gain insights into a boundary value problem by solving its adjoint. The Fokas method makes use of this feature of the adjoint. *given* Thus, the first step in implementing it is to find out how to characterize the adjoint problem ~~from~~ the original problem.

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.

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 Preliminaries

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

$$Lx = p_0x^{(n)} + p_1x^{(n-1)} + \cdots + p_{n-1}x' + p_nx,$$

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

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

is the **adjoint** of L .

where \bar{p}_k are? Is the notation rather than defn? Think carefully about how to explain this to your reader.

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

$$\sum_{k=1}^n (M_{jk}x^{(k-1)}(a) + N_{jk}x^{(k-1)}(b)) = 0 \quad (j = 1, \dots, m) \quad (2.1)$$

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 (1.1) 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 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 Green's formula

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

$$\int_{t_1}^{t_2} (Lu) \bar{v} dt - \int_{t_1}^{t_2} u \overline{(L^+ v)} dt = [uv](t_2) - [uv](t_1) \quad (2.2)$$

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

$$[uv](t) = \sum_{m=1}^n \sum_{j+k=m-1} (-1)^j u^{(k)}(t) (p_{n-m} \bar{v})^{(j)}(t) \quad (2.3)$$

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

$$[uv](t) = \sum_{j,k=1}^n B_{jk}(t) u^{(k-1)}(t) \bar{v}^{(j-1)}(t). \quad (2.4)$$

Note that

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

Replace all 1 with l for legibility

$$= \sum_{k=1}^n \sum_{m=k}^n (-1)^{m-k} \left(\sum_{l=0}^{m-k} \binom{m-k}{l} p_{n-m}^{(m-k-l)}(t) \bar{v}^{(l)}(t) \right) u^{(k-1)}(t).$$

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 $l = j-1$, 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 $l \leq m-k$, $l = j-1$ implies $j-1 \leq m-k$. Thus, $m-k$ ranges from $j-1$ to $n-k$. Let $l' := m-k$, then $m = k+l'$, and the above equation becomes

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

Thus,

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

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

$$B(t) = \begin{bmatrix} B_{11} & B_{12} & \cdots & \cdots & p_0(t) \\ \vdots & \vdots & \cdots & -p_0(t) & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ (-1)^{n-1} p_0(t) & 0 & \cdots & 0 & 0 \end{bmatrix}. \quad (2.5)$$

*find a way to use diagonal
data to improve looking*

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 an equation of $[uv](t)$ in matrix form. We begin by introducing the following definitions.

Definition 2.6. [\[4\]](#) p.285] For vectors $f = (f_1, \dots, f_k)$, $g = (g_1, \dots, g_k)$, 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 $f = (f_1, \dots, f_k)$, $g = (g_1, \dots, g_k)$ satisfying

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

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 a semibilinear form given by

$$\begin{aligned}\mathcal{S}(f, g) &:= Sf \cdot g = \begin{bmatrix} s_{11} & \cdots & s_{1k} \\ \vdots & & \vdots \\ s_{k1} & \cdots & s_{kk} \end{bmatrix} \begin{bmatrix} f_1 \\ \vdots \\ f_k \end{bmatrix} \cdot \begin{bmatrix} g_1 \\ \vdots \\ g_k \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^k s_{1j} f_j \\ \vdots \\ \sum_{j=1}^k s_{kj} f_j \end{bmatrix} \cdot \begin{bmatrix} g_1 \\ \vdots \\ g_k \end{bmatrix} \\ &= \sum_{i=1}^k \left(\sum_{j=1}^k s_{ij} f_j \right) \bar{g}_i = \sum_{i,j=1}^k s_{ij} f_i \bar{g}_i.\end{aligned}\tag{2.6}$$

Indeed:

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

Similarly,

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

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

$$\begin{aligned}[uv](t) &= \sum_{j,k=1}^n B_{jk}(t) u^{(k-1)}(t) \bar{v}^{(j-1)}(t) \quad (\text{by (2.4)}) \\ &= \sum_{i,j=1}^n (B_{ij} u^{(j-1)} \bar{v}^{(i-1)})(t) \\ &= (B \vec{u} \cdot \vec{v})(t) \quad (\text{by (2.6)}) \\ &=: \mathcal{S}(\vec{u}, \vec{v})(t).\end{aligned}\tag{2.7}$$

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_1) 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) & \cdots & 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) & \cdots & 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) & \cdots & -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) & \cdots & 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{2.8}$$

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.

2.2 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} ($i = 1, \dots, m; j = 1, \dots, n$), define m **boundary operators (boundary forms)** U_1, \dots, U_m for functions x on $[a, b]$, for which $x^{(j)}$ ($j = 1, \dots, n-1$) exists at a and b , by

$$U_i x = \sum_{j=1}^n (M_{ij} x^{(j-1)}(a) + N_{ij} x^{(j-1)}(b)) \quad (i = 1, \dots, m) \quad (2.9)$$

U_i are **linearly independent** if the only set of complex constants c_1, \dots, 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 $c_1 = c_2 = \dots = c_m = 0$.

Definition 2.9. [4 p.286] A **vector boundary form** $U = (U_1, \dots, U_m)$ is a vector whose components are boundary forms (Definition 2.8). When U_1, \dots, 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 (2.1) can be written as

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

Indeed:

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

$$\begin{aligned}
&= \begin{bmatrix} \sum_{j=1}^n (M_{1j}x^{(j-1)}(a) + N_{1j}x^{(j-1)}(b)) \\ \vdots \\ \sum_{j=1}^n (M_{mj}x^{(j-1)}(a) + N_{mj}x^{(j-1)}(b)) \end{bmatrix} \\
&= \begin{bmatrix} U_1x \\ \vdots \\ U_mx \end{bmatrix} = \begin{bmatrix} U_1 \\ U_2 \\ \vdots \\ U_m \end{bmatrix} x = Ux.
\end{aligned}$$

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, \dots, 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

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

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 $U = (U_1, \dots, U_m)$ is any boundary form with $\text{rank}(U) = m$ and $U_c = (U_{m+1}, \dots, U_{2n})$ any form with $\text{rank}(U_c) = 2n - m$ such that (U_1, \dots, U_{2n}) has rank $2n$, then U and U_c are **complementary boundary forms**.

Note that extending U_1, \dots, U_m to U_1, \dots, 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 (2.2) 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 (2.6), we have

$$Sf \cdot g = f \cdot S^*g, \quad (2.10)$$

where S^* is the conjugate transpose of S .

Proof.

$$\begin{aligned} Sf \cdot g &= \sum_{i,j=1}^k s_{ij} f_j \bar{g}_i \quad (\text{by (2.6)}); \\ f \cdot S^*g &= \begin{bmatrix} f_1 \\ \vdots \\ f_k \end{bmatrix} \cdot \begin{bmatrix} \bar{s}_{11} & \cdots & \bar{s}_{k1} \\ \vdots & & \vdots \\ \bar{s}_{1k} & \cdots & \bar{s}_{kk} \end{bmatrix} \begin{bmatrix} g_1 \\ \vdots \\ g_k \end{bmatrix} \\ &= \begin{bmatrix} f_1 \\ \vdots \\ f_k \end{bmatrix} \cdot \begin{bmatrix} \sum_{j=1}^k \bar{s}_{j1} g_j \\ \vdots \\ \sum_{j=1}^k \bar{s}_{jk} g_j \end{bmatrix} \\ &= \sum_{i=1}^k f_i \cdot \left(\sum_{j=1}^k \bar{s}_{ji} g_j \right) \\ &= \sum_{i=1}^k f_i \cdot \left(\sum_{j=1}^k s_{ji} \bar{g}_j \right) \\ &= \sum_{i,j=1}^k s_{ji} f_i \bar{g}_j = Sf \cdot g. \quad \square \end{aligned}$$

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

$$\begin{aligned} \mathcal{S}(f, g) &= Sf \cdot g \\ &= S(F^{-1}F)f \cdot g \\ &= SF^{-1}(Ff) \cdot g \\ &= SF^{-1}\bar{f} \cdot g \\ &= \bar{f} \cdot (SF^{-1})^*g \quad (\text{by (2.10)}) \\ &= \bar{f} \cdot G^*g \\ &= \bar{f} \cdot \bar{g}. \end{aligned}$$

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

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 ,

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

(where the third equality follows from a reverse application of (2.10) 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

$$[xy](b) - [xy](a) = Ux \cdot U_c^+ y + U_c x \cdot U^+ y. \quad (2.11)$$

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

$$\tilde{U}^+ y = C^* U^+ y \quad (2.12)$$

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 (2.3)). 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 ^{only} implies that adjoint boundary conditions are unique up to linear transformation. This is why we always use “an” instead of “the” when referring to adjoint boundary conditions.

Proof. Recall from (2.8) that the left hand side of (2.11) 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 (2.5). 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 $G = (SF^{-1})^*$), we have $J = (\hat{B}H^{-1})^*$ 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, (2.11) holds.

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

2.3 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

$$Ux = 0 \tag{2.13}$$

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

$$U^+ x = 0 \tag{2.14}$$

is an **adjoint boundary condition** to (2.13).

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 (1.1).

Proposition 2.18. By Green's formula (2.2) and the boundary-form formula (2.11),

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

for all $u \in C^n$ on $[a, b]$ satisfying (2.13) and all $v \in C^n$ on $[a, b]$ satisfying (2.14).

Proof.

$$\begin{aligned} (Lu, v) - (u, L^+v) &= \int_a^b Lu \bar{v} dt - \int_a^b u(\overline{L^+v}) dt \\ &= [uv](a) - [uv](b) \quad (\text{by Green's formula (2.2)}) \\ &= Uu \cdot U_c^+v + U_cu \cdot U^+v \quad (\text{by boundary-form formula (2.11)}) \\ &= 0 \cdot U_c^+v + U_cu \cdot 0 \quad (\text{by (2.13) and (2.14)}) \\ &= 0. \end{aligned}$$

□

In some cases, the above result is treated as the definition of the adjoint problem. Here, we treat it as a property of the adjoint after defining L^+ and constructing U^+ . Yet we can still appreciate the motivation behind defining the notion of adjoint for boundary value problems. (Good definition. Expect it to be a remark.)

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

$$U^+x = P^*\xi(a) + Q^*\xi(b). \quad (2.15)$$

The following theorem is motivated by characterizing the adjoint condition (2.14) 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

$$MB^{-1}(a)P = NB^{-1}(b)Q \quad (2.16)$$

where $B(t)$ is the $n \times n$ matrix associated with the form $[xy](t)$ (2.5).

Proof. Let $\eta := (y, y', \dots, y^{(n-1)})$, then $[xy](t) = B(t)\xi(t) \cdot \eta(t)$ by (2.7).

Suppose $U^+x = 0$ is adjoint to $Ux = 0$. By definition of adjoint boundary condition (2.14), 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 (2.11) holds.

Put

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

Then by the boundary-form formula (2.11),

$$\begin{aligned} B(b)\xi(b) \cdot \eta(b) - B(a)\xi(a) \cdot \eta(a) &= (M\xi(a) + N\xi(b)) \cdot (P_c^*\eta(a) + Q_c^*\eta(b)) + \\ &\quad (M_c\xi(a) + N_c\xi(b)) \cdot (P^*\eta(a) + Q^*\eta(b)). \end{aligned}$$

By (2.10),

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

Thus,

$$\begin{aligned} B(b)\xi(b) \cdot \eta(b) - B(a)\xi(a) \cdot \eta(a) &= (P_c M + P M_c)\xi(a) \cdot \eta(a) + (Q_c M + Q M_c)\xi(a) \cdot \eta(b) \\ &\quad (P_c N + P N_c)\xi(b) \cdot \eta(a) + (Q_c N + Q N_c)\xi(b) \cdot \eta(b). \end{aligned}$$

Thus, we have

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

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 (2.16).

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 $\text{rank}(P_1^* : Q_1^*) = 2n - m$. Suppose

$$MB^{-1}(a)P_1 = NB^{-1}(b)Q_1 \tag{2.17}$$

holds.

By the fundamental theorem of linear maps [3] p.63], if V is finite-dimensional and $T \in \mathcal{L}(V, W)$, then $\dim \ker T = \dim V - \dim \text{range } T$. 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 $\dim \ker A = \dim A - \dim \text{range } A$. 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 $\dim \text{range } A$. 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 (2.17),

$$MB^{-1}(a)P_1 - NB^{-1}(b)Q_1 = 0,$$

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 $\text{rank}(P_1^* : Q_1^*) = 2n - m$,

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

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

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 $\text{rank}(H) = 2n - m$. Therefore, by (2.16), 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^* is a linear map, $U^+y = 0$ implies $U_1^+y = A^*U^+y = 0$. Since A^* is nonsingular, A^{*-1} is also a linear map, and $A^{*-1}U_1^+y = U^+y$. Thus, $U_1^+y = 0$ implies $U^+y = A^{*-1}U_1^+y = 0$. 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 $Ux = 0$. \square

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.

3 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 $U = (U_1, \dots, U_n)$ (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).

3.1 Check input

We first check that U_1, \dots, 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 $\text{rank}(M : N) \neq n$.

3.2 Find U^+

Recall from Definition 2.10 that extending U_1, \dots, U_n to U_1, \dots, U_{2n} (where (U_{n+1}, \dots, 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  $\leftarrow$   $(M : N)$ ;
    for  $i$  in  $\text{range}(\text{nrow}(E))$  do
        mat1  $\leftarrow$  vcat(mat, E[i,]);  $\triangleright$  vcat := vertical concatenation, or joining two
            matrices vertically
        if  $\text{rank}(\text{mat1}) == \text{rank}(\text{mat}) + 1$  then
            mat  $\leftarrow$  mat1
        else
            E  $\leftarrow$  E[-i,]
    return vcat(mat, E)
```

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, \dots, p_n and the interval $[a, b]$. Let B be as in (2.5) which depends on p_0, \dots, 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 $(P^* : Q^*)$ with the last n rows of J . That is, identify P^* with \tilde{M}' , the lower-left $n \times n$ submatrix of J , and Q^* with \tilde{N}' , the lower-right $n \times n$ submatrix of J . Define U^+ by

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

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

3.3 Check U^+

By Theorem 2.19 with

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

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 (2.5).

4 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, \dots, 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 $\text{rank}(M : N) = n$. 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, \dots, 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 displayed as a general method f in Julia without information for the user as to what it actually is. With symbolic expression, f will be displayed 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.

5 Next steps

With regard to the semester 1 plan in the project proposal, the progress has been on track so far. The only difference being that literature review turns out to be proceeding simultaneously with algorithm implementation, instead of sequentially, as suggested in the plan.

The next steps in the project would be to implement a non-classical transform pair (2.15a to 2.16b in [1 p.10]) that can be used to solve (complicated) initial-boundary value problems for which there does not exist a classical transform pair. Looking further ahead, when features associated with calculations are in place, we will begin implementing graphics features that allow users to visualize objects of interest.

This is very good. Clearly written and mathematically correct.

You will need to significantly expand the introduction eventually but, given that you have only just started reading [Fisher Smith 2016], it is not surprising there is not much done yet in that regard.

Very well done!

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] Emine Kesici, Beatrice Pelloni, Tristan Pryer, and David Smith. A numerical implementation of the unified Fokas transform for evolution problems on a finite interval. 2016.
- [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.