Algorithmic Solution of High Order Partial Differential Equations in Julia via the Fokas Transform Method

An Initial Thesis Presented in Partial Fulfillment of
the Honours Bachelor's Degree in Science

Linfan Xiao

Mathematical, Computational, and Statistical Sciences
Under the supervision of Prof. Dave Smith
Yale-NUS College
March 2019

Contents

List of Figures

List of Tables

1 Introduction

Evolution partial differential equations (PDEs) relate a quantity to its rates of change with respect to both time and position. Evolution PDEs can model a variety of phenomena in physics, such as wave propagation and particle motion. This project concerns implementing an algorithmic procedure to solve a certain class of initial-boundary value problems (IBVP) for evolution PDEs in the finite interval [1] based on the Fokas transform method [2]. The implementation is based in Julia, with a focus on symbolic results among numeric features.

1.1 Motivation

To motivate the project, suppose we want to solve some IBVPs.

Certain IBVPs (henceforth referred to as type I problems) can be solved algorithmically via classical transform pairs such as the Fourier transform (Figure 1). For example, the heat equation

$$\frac{\partial}{\partial t}u = a\frac{\partial}{\partial x^2}u, \quad -\infty < x < \infty$$

with initial condition $u(x,0) = u_0(x)$ can be solved by **TBD**

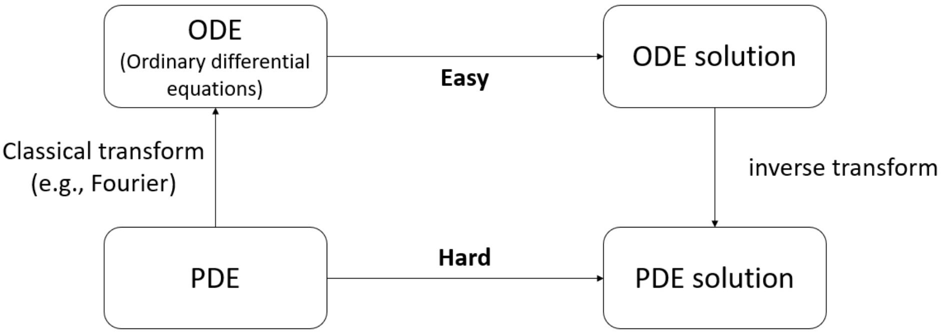

Figure 1: Solving type I IBVPs using classical transform pairs.

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 transform 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 can now be "customized" 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).

Examples of IBVPs that can be solved by the Fokas method include second order PDEs of the form

$$\frac{\partial}{\partial t}q(x,t) + a(-i)^2\frac{\partial^2}{\partial x^2}q(x,t) = 0,$$

with boundary conditions

$$\begin{aligned} q(0,t) + c_0 q(1,t) &= 0 \\ q'(0,t) + c_1 q'(1,t) &= 0, \end{aligned} \tag{1.1}$$

or

$$q(0,t) + c_0 q'(0,t) = 0$$

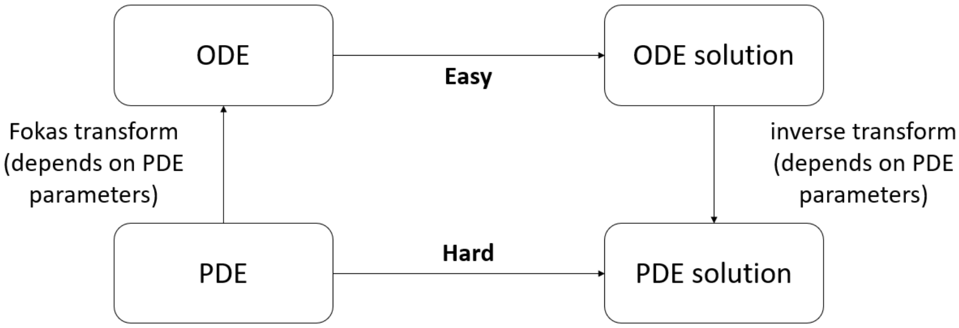
ODE
Easy
ODE solution
Fokas transform (depends on PDE parameters)
inverse transform (depends on PDE parameters)
PDE
Hard
PDE solution

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

$$q(1) + c_1 q'(1,t) = 0,$$

where $c_0, c_1 \in \hat{\mathbb{C}} := \mathbb{C} \cup \{0, \infty\}$ (in the sense that if $c_0 = \infty$ in 1.1, then dividing by c_0 makes $q(0,t)$ disappear and leaves the first equation as $q(1,t) = 0$), as well as third order PDEs of the form

$$\frac{\partial}{\partial t} q(x,t) \pm i(-i)^2 \frac{\partial^3}{\partial x^3} q(x,t) = 0,$$

with boundary conditions

$$\begin{aligned} q(0,t) + c_0 q(1,t) &= 0 \\ q'(0,t) + c_1 q'(1,t) &= 0 \\ q''(0,t) + c_2 q''(1,t) &= 0, \end{aligned}$$

where $c_j \in \hat{\mathbb{C}}$ for $j = 0, 1, 2$, or

$$\begin{aligned} q(0,t) &= 0 \\ q(1,t) &= 0 \\ q'(0,t) + c q'(1,t) &= 0, \end{aligned}$$

where $c \in \hat{\mathbb{C}}$.

In fact, as we will see later, the Fokas method is applicable to IBVPs in the above form with PDEs of any general order n. In general, compared to PDEs of higher orders, first and second order PDEs are studied and applied more frequently. The Fokas method advances the understanding of PDEs with higher orders by providing an algorithmic way to solve an entire class of IBVPs of arbitrary order. Using the Fokas method by hand, however, is laborious. It is thus of interest to implement the Fokas method as a software package that aims to supply as much computer aid as possible in the computation process. To our knowledge, despite the presence of various solvers in mathematical softwares such as Mathematica and Matlab, this would be the first time that the Fokas method is implemented.

In this project, the Fokas method is implemented in Julia, a free open-source, high-performance language for numerical computing [3]. We choose Julia to be the language of implementation for its support for high-performance numerical computing as well as its open-source property. Commercial softwares may provide access to sophisticated computational algorithms; yet due to being closed-source, there exists a lack of transparency which limits users' understanding of the nature of the software output. Therefore, the mathematical community is moving towards open-source languages like Julia, which allows users to verify the correctness of the output.

As mentioned before, the purpose of implementing the Fokas method in Julia is to

1.2 The Initial-Boundary Value Problem

To formally characterize the class of IBVPs that can be solved by the Fokas method [1, p.9], we first introduce the following definitions.

- Define linearly independent boundary forms $B_j : C^\infty[0,1] \to \mathbb{C}$ (where $C^\infty[0,1]$ denotes the class of real-valued functions differentiable for all orders on $[0,1]$) by

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

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

- Define

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

  to be the set of smooth functions ϕ satisfying the homogeneous boundary conditions $B_j\phi = 0$ for $j \in \{1, 2, \dots, n\}$.

- Define

$$S = (-i)^n \frac{d^n}{dx^n}$$

  to be a spatial differential operator of order n (i.e., a differential operator in the spatial variable x).

The Fokas method allows solving any well-posed IBVP that can be written as

$$(\partial_t + aS)q(x,t) = 0 \qquad \forall (x,t) \in (0,1) \times (0,T) \tag{1.2}$$

$$q(x,0) = q_0(x) \in \Phi \qquad \forall x \in [0,1] \tag{1.3}$$

$$q(\cdot,t) = f \in \Phi \qquad \forall t \in [0,T], \tag{1.4}$$

where a is a complex constant. For the IBVP to be well-posed, we require that $a = \pm i$ if n is odd and $\mathrm{Re}(a) \geq 0$ if n is even. Equation 1.2 is a PDE relating a quantity q to its temporal and spatial rates of change $\partial_t[q]$ and $aS[q]$. 1.3 is the initial condition where the temporal variable $t = 0$. 1.4 corresponds to the homogeneous spatial boundary conditions. An example of such IBVPs is the linear Schrödinger equation [4], which describes the wave form of a quantum system in free space. The linear Schrödinger equation with zero potential is 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. It can be written in the form of 1.2 as

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

where $q = w$, $S = \dfrac{\partial^2}{\partial x^2}$, and $a = \dfrac{h}{2mi}$. The initial condition

$$w(x,0) = w_0(x)$$

would mean that the system's wave form at initial stage is described by the wave function $w_0(x)$, and the boundary conditions

$$w(\cdot,t) = f$$

would mean that the behaviour of the system at its boundary is described by the function f (e.g., if we are considering particles moving in a box, f would describe how the particles behave at the box's boundaries).

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

$$q(x,t) = f_x(e^{-a\lambda^n t}F_\lambda(f)). \tag{1.5}$$

In the following sections, we first introduce the preliminary definitions and results

2 Preliminaries

We now introduce

2.1 Boundary Value Problems and Adjoint Boundary Value Problems

2.2 Roots of Exponential Polynomials

2.3 The Fokas Transform Pair

3 Algorithm

3.1 Constructing the Adjoint of a Homogeneous Boundary Condition

3.2 Approximating the Roots of an Exponential Polynomial on a Bounded Region

3.3 Solving the Initial-Boundary-Value Problem

4 Implementation

4.1 Platform

4.2 Examples

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] The Julia Language.

[4] Michael E. Taylor. *Partial Differential Equations I: Basic Theory*. Springer, second edition, 2018.