Elevator Pitch

Sultan Aitzhan & Dave Smith

August 5, 2019

Spiel for Not-into-Math-Type of Audience (around either 3 mins or 5 mins)

- **Presenter:** Have you taken linear algebra/FoAM? **Audience:** No!
- **What do we know on this topic?** Operator is a function, and an operator's adjoint is a function that has some nice properties in relation to its original operator. For example, consider a function on the xy-plane that rotates a given point by 90 degrees counterclockwise, say $(1,1) \mapsto (-1,1)$. In this case, the adjoint would be the function that rotates a given point by 90 degrees clockwise, i.e. $(-1,1) \mapsto (1,1)$. So, in a sense we could think of an adjoint as an "inverse" of an operator. While this is a simple example, in general operators and adjoints pervade much of mathematics and physics. In this project, we consider an ordinary differential operator on a finite interval, i.e. the result of applying this operator is a sum of derivatives of the input. The domain is characterised by functions that are specified not only at the endpoints of the interval but also in the interior; we refer to these specifications as multipoint conditions. Prior to the start, no explicit construction of an adjoint for this operator has been done.
- **What did you find or what do you expect to find?** Building on the work of Linfan Xiao from the Class of 2019, who provided an explicit construction of the adjoint when there are 2 points, we managed to extend this explicit construction for multipoint conditions. In other words, we developed an algorithmic procedure to find the adjoint operator. In doing so, we provided the necessary theoretical framework in the form of definitions and theorems.
- **What is the significance of the work and what are the further directions?** The significance of the project is that we now know how to find an adjoint of the given operator, and that we can calculate it fast using a computer. As explained in the supervisor's work, the adjoint boundary conditions may be used to solve an initial boundary value problem for the same operator with two-point boundary conditions. Thus, one possible direction is solving an initial multipoint value problem for the same operator with multipoint boundary conditions.
- **Presenter:** Would you like to learn more about the details? **Audience:** Yes!
- **What are some important aspects that we don't know, and why are they important?** The problem we considered is that of finding an adjoint to the mentioned operator. The operator itself was considered in several papers by Locker, Wilder, and Loud. While these authors conclude that a relevant adjoint exists and work out a few examples, none of them provide an explicit construction of the adjoint. When the operator is of order 2 or 3, and when the number of points is small, say 2 or 3 or 4, it might be possible to work out the adjoint operator by hand, though with some time and effort. However, if the operator is of large order, and the number of points is also large, then working out by hand becomes burdensome and takes some time. Thus, if we were to construct an adjoint, it would be advantageous to construct it in a way that is explicit and algorithmic, because then we could make use of the fast computing power that almost all modern laptops possess.
- **What did you do in your research and how did you do it?** We first conducted a literature review, readings several works on the adjoint of the multipoint operator, as well as spent some time familiarising ourselves with the way Linfan explicitly constructed the adjoint boundary conditions. We then extended Linfan's construction, supplying the theory with the necessary definitions, theorems, and proofs where needed. Finally, we implemented the procedure in programming language Julia, which we accomplished by modifying parts of Linfan's code as appropriate.

Spiel for Into-Math-Type of Audience (around either 3 mins or 5 mins)

- **Presenter:** Have you taken linear algebra/FoAM? **Audience:** Yes!
- **What do we know on this topic?** Given an operator, its adjoint is another operator that has a special association to the original operator. The importance of adjoints cannot be overstated, for they can be used in a variety of contexts: one example is deduce the solvability conditions of an arbitrary system. In this project, we consider an ordinary differential operator on a finite interval. The domain is characterised by functions that are specified not only at the endpoints of the interval but also in the interior; we refer to these specifications as multipoint conditions. Further, we assume that all of these conditions are zero. Prior to the start, no explicit construction of an adjoint for this operator has been done.
- **What did you find or what do you expect to find?** Building on the work of Linfan Xiao from the Class of 2019, who provided an explicit construction of the adjoint when there are 2 points, we managed to extend this explicit construction for multipoint conditions. In other words, we developed an algorithmic procedure to find the adjoint operator. In doing so, we provided the necessary theoretical framework in the form of definitions and theorems.
- **What is the significance of the work and what are the further directions?** The significance of the project is that we now know how to find an adjoint of the given operator, and that we can calculate it much faster using a computer. As explained in the supervisor's work, the adjoint boundary conditions may be used to define a transform pair, that in turn can be used to solve an initial boundary value problem for the same operator with two-point boundary conditions. Thus, one possible direction is solving an initial multipoint value problem for the same operator with multipoint boundary conditions, in which we could use adjoint multipoint conditions to define the relevant transform pair.
- **Presenter:** Would you like to learn more about the details? **Audience:** Yes!
- **What are some important aspects that we don't know, and why are they important?** The problem we considered is that of finding an adjoint to the mentioned operator. The operator itself was considered in several papers by Locker, Wilder, and Loud. While these authors conclude that a relevant adjoint exists and work out a few examples, none of them provide an explicit construction of the adjoint. When the operator is of order 2 or 3, and when the number of points is small, say 2 or 3, it might be possible to work out the adjoint operator by hand, though with some time and effort. However, if the operator is of large order, and the number of points is large, then construction by hand becomes burdensome and takes quite some time. Thus, if we were to construct an adjoint, it would be advantageous to construct it in a way that is explicit and algorithmic, because then we could make use of the fast computing power that almost all modern laptops possess. Further, a distinct aspect of our approach was to think of an operator as a multipoint value problem, where the action of the operator specifies the PDE, and the domain is given by the multipoint conditions. Since the multipoint conditions are zero, the problem in question becomes the homogeneous multipoint value problem. This problem is somewhat easier to tackle, because an adjoint problem was already derived for two-point conditions by Linfan.
- **What did you do in your research and how did you do it?** We first conducted a literature review, readings several works on the adjoint of the multipoint operator, as well as spent some time familiarising ourselves with the way Linfan explicitly constructed the adjoint boundary conditions. We then extended Linda's construction, supplying the theory with the necessary definitions, theorems, and proofs where needed. In addition, we developed a test that given two collections of multipoint conditions, along with the relevant operator, checks whether two collections are adjoint to each other. This allows to develop an algorithm for both generating and checking the adjoint conditions, which we finally implemented in programming language Julia, accomplished by modifying parts of Linfan's code as appropriate.