Gradient Flow and Geometric Regularization: *Notes*

Colin Roberts

July 13, 2020

1 outline

- Ask the motivating questions and give the benefits in mind
- Define the finite dimensional case
- Introduce the infinite dimensional case.
- Define the dirichlet energy functional
- Recover laplace's equation
- Heat equation
- Geodesics and Minimal surfaces using the dirichlet energy functional

Paper: Flowing maps to minimal surfaces: Existence and uniquenessof solutions

2 Statement

"Yes, great idea. The Poincare conjecture was proven using a technique like this, called "Gradient Flow". I think this would be an interesting project where we all (including me) could learn something about the application of PDEs outside the realm of physical models."CaC

Many physical phenomenon are described as being optimizers to some type of functional usually with an added constraint. A few examples: The shape of a rain drop minimizes drag with the constraint that the volume is fixed; the way a cable hangs between two poles minimizes the total energy while keeping the total length fixed; an elastic band attached at two endpoints minimizes its length between these two points where there may be obstacles in the path. The issue at hand is that these solutions can be rather difficult to find analytically even when we can prove that the solutions should exist and may even be unique. Not all is lost, however. One technique to assist in finding these solutions is called "gradient flow." This technique is powerful in computationally finding optimizers and proving existence in theory. The analogy I'll seek to describe will begin in the finite dimensional case with Lagrange multipliers and pictorially describing the flow of the gradients in this case. With the picture in mind, we can approach the (homogeneous, source free) heat equation and its long time limit and discuss how we can consider this as a gradient flow problem. We can in fact show that this directly relates to finding minimal surfaces and geodesics in flat space. One may then take an extrinsic point of view and add a constraint that our surfaces or curves must lie on manifolds embedded in flat space. For example, we can find the geodesics on the 2-sphere embedded in $\mathbb{R}^3$ using the techniques described. The last idea to note is the following: What if we think of living on the manifold instead of viewing it from the outside? That is, what if we take an intrinsic point of view versus the previously described extrinsic point of view. It turns out that the constraints are lifted and, instead, the notion of what a straight line is changes.

3 Notes on *Variational Modelling: Energies, Gradient Flows, and Large Variations*

The paper is given here: `https://arxiv.org/pdf/1402.1990.pdf`

-
-

4 Gradient Flows in Function Spaces

`https://math.stackexchange.com/questions/1687804/what-is-the-l2-gradient-flow`

One of the slides:
Given an infinite dimensional vector space space V equipped with a functional

$$\mathcal{E}\colon V \to \mathbb{R}.$$

The gradient flow is

$$\frac{\partial u}{\partial t} = -\frac{\delta \mathcal{E}}{\delta v}\bigg|_{v=u}$$

Think of

- $\frac{\partial u}{\partial t}$ as the change in "position" over time;
- $\frac{\delta \mathcal{E}}{\delta v}$ as the change in the functional over "space."

How do I formalize all of this? That is, what exactly is the relationship here? In the finite case, you have two vectors set equal and this makes total sense. Here we have two functions equal? Is the $v = u$ sensible? I'm not sure this is exactly right.

In the finite case, we are considering a derivative over "all" directions. In the infinite case, we do the same, and arrive at the weak form of a PDE. Maybe this is how we should bridge the gap.

5 Algorithm

Algorithm 1 Finite Dimensional Gradient Descent

```
procedure FinDimGradDesc
    $stringlen \leftarrow$ length of $string$
    $i \leftarrow patlen$
top:
    if $i > stringlen$ then return false
    $j \leftarrow patlen$
loop:
    if $string(i) = path(j)$ then
        $j \leftarrow j - 1.$
        $i \leftarrow i - 1.$
        goto $loop.$
        close;
    $i \leftarrow i + \max(delta_1(string(i)), delta_2(j)).$
    goto $top.$
```

Algorithm 2 Infinite Dimensional Gradient Descent

```
procedure InfDimGradDesc
    $stringlen \leftarrow$ length of $string$
    $i \leftarrow patlen$
top:
    if $i > stringlen$ then return false
    $j \leftarrow patlen$
loop:
    if $string(i) = path(j)$ then
        $j \leftarrow j - 1.$
        $i \leftarrow i - 1.$
        goto $loop.$
        close;
    $i \leftarrow i + \max(delta_1(string(i)), delta_2(j)).$
    goto $top.$
```