INVERSE PROBLEM FOR LIGHT SPREADING APPLIED TO MEDICAL TOMOGRAPHY

Project Report Version 0 : Guidelines reformulation

Desmond Roussel Ngueguin

**Under de supervision of**

Emmanuel Franck, Laurent Navoret, and Vincent Vigon

# CONTEXT

Inverse problems are some of the most important problems in science and mathematics because of their wide range of applications in medical imaging, computer vision, radar, etc. These are problems that, given a set of observations, tell us the parameters that cause them, parameters that we cannot directly observe. In general, numerically solving an inverse problem requires advanced optimization algorithms. However, those can be difficult to implement. That is why we are introducing an approach based on machine learning and neural networks.

The problem to solve is a medical tomography inverse problem by infrared. This project will study a neural network-based method for finding the optical properties of an organ. The project was suggested and is run by the MOCO (Modélisation et Contrôle) team at IRMA (Institut de Recherche Mathématique Avancée). This team is made up of specialists in PDE analysis, control theory, high-performance scientific computing, and statistics (IRMA, n.d.). This specific project is under the guidance of IRMA researchers Emmanuel Franck, Laurent Navoret, and Vincent Vigon.

# OBJECTIVES

The **main objective** of this project is to find the optical properties of an organ by solving an inverse problem. We send an infrared light beam through an organ and we measure the signal on part of that organ. Knowing the initial conditions and the signal at all times, we can infer the organ’s properties. Namely its density, its scattering opacity, and its absorption opacity. As usually done in medical imagery, we should be able to tell if the organ if tumorous or not.

The project is divided into two parts. First, we must solve the partial differential equation (PDE) for a simplified version of the **radiative transfer equation** (1) in 1D. This will be done using the finite volumes method; it will count as our **1st specific objective**. The **2nd specific objective** will aim to solve the machine learning part of the project. Knowing the signal’s value on the boundaries of a domain at all times, the neural network will predict the properties of that domain.

(1)

Where:

* is the Stefan-Boltzmann constant
* is the thermal capacity of the medium
* is the speed of light
* is the temperature of the medium
* is the energy of the photons
* is the flux of the photons
* is the density of the medium
* is the absorption opacity
* is the scattering opacity

We should note that when the scattering opacity and the absorption opacity are high enough to be close to the speed of light (which is far larger than the observed phenomenon’s speed), the model in (1) is reduced to the following:

This model is called the **diffusion approximation** (Franck, 2020, p.7)**.** It matches a diffusion equation with a diffusion coefficient of . But more importantly, it gives us a good way of testing our program.

# ROADMAP AND DEADLINES

This section states the tasks that have already been completed and gives the planning for the remaining ones.

## Phase 1: Solving the PDE with Finite Volumes

Due to the hyperbolic nature of the PDE (1)*,* the constraints of our domain, and the need to deal with discontinuities, the **finite volumes** method is well suited to solving it. However, the well-known Rusanov scheme is not accurate enough, especially when the opacities are very high. Therefore, we will use a finite volume scheme with a splitting strategy (Franck, 2012, p. 160).

First, we must discretize our domain to form a mesh. Let and be the two real numbers such that , and an integer. We split the domain into cells of equal length to obtain a uniform mesh. At the two edges, we add two “ghost” cells. In total, we have cells. Let’s denote by the length of the intervals (the volume of the cells). For each cell , we write its center, its left edge, and its right edge.

Next, we write the splitting scheme. Two steps are required here.

### Step 1: The equilibrium part

We consider the “equilibrium” part of (1). That is, the photons are not moving, and we only take into account the relaxation terms in temperature (Franck, 2012, p. 160). This leads to all the terms with in (1) to be equal to . The equation becomes:

Writing , we solve (3 on each independent cell. The numerical scheme is given below.

Rewritten as:

Applying Cramer’s rule, we get:

Where

* , and are the value values of , and on the cell at the beginning of the step.
* , , , and
* written above is a function of and . Thus, it is actually .
* is such that

Since this step is a fixed point method, we iterate on until we get close enough to the fixed point , or more precisely . We then move to the next step.

### Step 2: Solving the rest

Once the first step converges, we move to this step with the values of and on each cell updated. We write:

With

We must also include the CFL condition below to ensure the scheme’s stability.

We can rewrite (5 as:

With

and

Once all of this is done, we move back to step 1, then back to step 2 and so on until we reach a predefined time at which we want the solution.

Concerning the deadlines, it is important to mention that considerable steps towards solving the PDE (1) have already been taken. However, a good deadline estimate for the total completion of the phase is **April 15, 2020.**

## Phase 2: Machine learning with Neural Networks

Convoluted neural networks are the machine learning choice for this phase.

### Step 1: Finding the density

Knowing not only the signals and at all times, but also the scattering and absorption opacities ( and ), we will rebuild the domain’s density . We can see that the inputs for the neural network will be 2D tensors (indexed by and ) while the output will be 1D (indexed only by ).

First, we will train and validate the neural network with the data obtained from solving the PDE (1) in the first phase. Second, we will test it hoping it generalizes well.

This part shall be completed before **April 29, 2020.**

### Setp2: The absorption and scattering opacities

This step is quite similar to the previous step with the only difference that we will be trying to predict and knowing this time. However, it requires a more complex neural network.

This part shall be completed before **May 19, 2020.**

# RESSOURCES AND BUDGET

Considerable resources are needed to complete this project. For computing purposes, a server on Atlas is available at <v100.math.unistra.fr>. Extra computing resources are also allocated through [Google Colab](https://colab.research.google.com/notebooks/welcome.ipynb).

## Phase 1: Solving the PDE

This part will be almost entirely coded in C++. We will need the [Eigen](http://eigen.tuxfamily.org/index.php?title=Main_Page) library for vector operations. A function parser like [cparse](https://github.com/cparse/cparse/wiki/Getting-Started) could also be useful. The output files will be exported in the CSV format for visualization with Matplotlib in Python.

## Phase 2: Neural Network

Since Python is the language of choice for data science, we will use it during this phase. For this to work, we will have to find a way of storing all the data from our PDE simulations in their 2D shapes, all this in one single file. We will then load the data and use it to train and test the neural network using the open-source neural-network library [Keras](https://keras.io/).

# MILESTONES

This section gives a summary for the roadmap, the objectives and the deadlines for the project.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Milestone | | Details | Tools needed | Deadline | Estimated number of hours |
| Report version 0 | | A report indicating the context and the roadmap for the project. This milestone is currently under completion. | Microsoft Office Word | April 15, 2020 | 16 hours |
| Phase 1: Finite Volumes | Numerical solution | Using the finite volumes method in 1D, we need to solve the PDE (1). This has milestone has already been completed. | VS Code, GitHub,  Eigen, Cparse | April 15, 2020 | 40 hours |
| Verification | Verify that the finite volume method is correctly implemented and solves a direct problem on a given domain. A good way to verify our splitting scheme is to test it on the diffusion approximation in (2. We will need to correct the bugs that appear in this phase. | VS Code | April 29, 2020 | 8 hours |
| Benchmarking | Compare our algorithm to known solutions in order to optimize our code for speed. This step might be done multiple times depending on the changes we make to the algorithm during verification. | VS Code | April 29, 2020 | 4 hours |
| Validation | Making sure the problem solves the correct direct problems linked to medical tomography. | VS Code, Matplotlib | April 29, 2020 | 8 hours |
| Data exporting | Writing and running a script that exports thousands of instances of a correctly solved direct problem. This requires us to run the above-optimized program a great number of times, which is the reason we need to get it right on the first try. | Altas | May 5, 2020 | 8 hours |
| Phase 2: Neural Networks | Training the model | Using the exported data, we configure and train the neural network. This is a potentially time-consuming task. Moreover, it will probably be done more than once. That is why we will do it on the Atlas server. | Atlas, Keras | May 19, 2020 | 8 hours |
| Verification | Verify that the model is properly trained. Positive indicators might be a decreasing loss value (rmse, Gini, cross-entropy), and a none underfitted nor overfitted model. | Google Colab | May 19, 2020 | 16 hours |
| Validation | Depending on the model’s score when applied to the validation data, we change the hyperparameters or chose a completely different model, sending us back to the “training” step. | Google Colab | May 19, 2020 | 8 hours |
| Benchmarking | Check that the neural network’s learning and predicting steps are fast and optimal. All this for quick reproducibility of our results on other systems. | Google Colab | May 19, 2020 | 4 hours |
| Report version 1 | | The task is to write a more complete version of the report. | MS Word | May 19, 2020 | 16 hours |
| Report version 2 | | The final version of the report, incorporating corrections indicated by the supervisors. | MS Word | May 19, 2020 | 4 hours |
| Presentation | | A slideshow to be written in PowerPoint. | MS PowerPoint | May 19, 2020 | 8 hours |

# DELIVRABLES AND OUTCOME

What is expected to be delivered at the end of this project are:

* A software that can model light spreading according to the PDE (1)
* A software that can accurately predict the optical properties on a given domain
* A typewritten report by **May 15, 2020**

All these files can be found on the GitHub repository [feelpp/csmi-m1-2020-moco-inverse](https://github.com/feelpp/csmi-m1-2020-moco-inverse) along with instructions on how to run the software.

# REFERENCES

* Franck, E. April 1, 2020. “*Projets de M1* ”. Personal notes from Emmanuel Franck summarizing the guidelines for the project.
* Franck, E. October 23, 2012. “*Construction et analyse numérique de schéma asymptotic preserving sur maillages non structurés. Application au transport linéaire et aux systèmes de Friedrichs* ”. Retrieved from <https://tel.archives-ouvertes.fr/file/index/docid/744371/filename/theseFranckv3.pdf>
* IRMA. (n.d.). “ Institut de Recherche Mathématique Avancée, UMR 7501 ". Retrieved from <http://irma.math.unistra.fr/rubrique162.html>