INVERSE PROBLEM FOR LIGHT SPREADING APPLIED TO MEDICAL TOMOGRAPHY

Project Report Final Version

Desmond Roussel Ngueguin

Under de supervision of

Emmanuel Franck, Laurent Navoret, and Vincent Vigon

CONTENT

I. 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 prepare to 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.

II. DESCRIPTION

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 influence of the organ's properties. Namely the influence of the density, the scattering opacity, and the absorption opacity. Monte Carlo methods, Discrete Ordinate methods, Lambda Iteration (LI), and Accelerated Lambda Iteration (ALI) are some of the mode widely used methods when it comes to radiative transfer (Dullemond, 2012, p.45). However, they are complicated to implement.

The P_1 model is a moment equation for radiative transfer that is easy, fast, and generalizes well under limiting conditions. Using such a model, we will implement a splitting scheme to compute the signal's value (its evolution on the boundaries of a 1-dimensional domain, and the complete signal at the end of the simulation). Then we will visually inspect the data to understand the influence of the domain's properties on the signal.

1. Objectives

Main objective:

- Find out how the optical properties of an organ affect a signal traveling through it.

Specific Objectives:

- Understand the mathematical model for radiative transfer used.
- Numerically solve the model.
- Analyze the resulting data.

2. Tasks

This is a non-exhaustive list of the tasks completed during the project.

- Read about the mathematical model to be used (see Franck, 2012, p.169)
- Implement the splitting scheme using a finite volumes method
- Run test cases
- Analyze the data
- Write the reports
- Present the work that was done

3. Resources

The choice to use C++ to solve the P_1 model relies on it being a compiled programming language, giving us more speed, which is critically important since we will be generating thousands of lines in a data frame to be analyzed. Python was chosen as our language for data visualization because it is simply the best at doing that. Resources used are as follows:

- muParser : a C++ library for parsing expressions into functions
- **Atlas**: Computing power allocated through VPN at v100.math.unistra.fr
- **Google Colab**: for bulk data visualization in Python

Also, a tool like **Google Hangouts** has proven essential for remote project coordination and supervision, given the COVID-19-related sanitary crisis these last few months.

4. Deliverables

What was delivered at the end of the project are:

- A software named "**transfer**" that solves the numerical model
- An analysis of meaningful trends in the data generated (notebook, pdf, ...)
- A typewritten report named "**rapport_vfinale.pdf**"
- A presentation named "**presentation.pdf**"

All these files can be found on the GitHub repository feelpp/csmi-m1-2020-moco-inverse along with instructions on how to run the software.

III. ROADMAP

1. Mathematical basis

a. The radiative transfer equation (RTE)

Light traveling in a straight line through a medium interacts with the matter in multiple ways (Figure 1):

- **Emission**: Photons of light are emitted as a result of electrons in the matter jumping to a less excited state. The matter loses energy to the light beam. This phenomenon is characterized by the emission opacity σ_e. It is the inverse of the mean length between two consecutive emissions.

- **Absorption**: Photons are absorbed and electrons jump to more excited levels (or free themselves from their atoms). The matter is heated by this gain of energy. This phenomenon is characterized by the absorption opacity σ_a.
- **Scattering** (or **diffusion)**: Some photons are shifted from their direction, and sometimes change frequencies. Energy is conserved in the incoming radiation. This phenomenon is characterized by the scattering opacity σ_c.

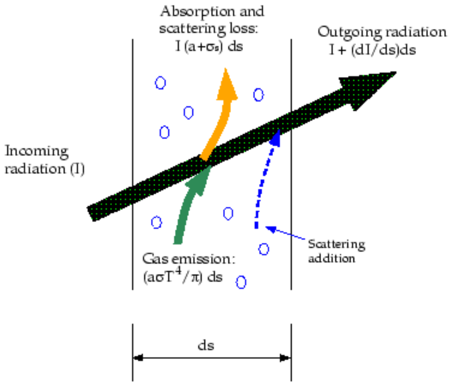


Figure 1 : Radiative transfer phenomenon in a gas (Ansys Fluent, n.d)

These opacities depend on multiple parameters among which ρ, T, and ν respectively defined as the density of the medium, its temperature, and the frequency of the photon interacted with.

We denote by $I(t, x, \Omega, \nu)$ the specific radiative intensity. As we can see, at any time t, it depends on 6 variables:

- 3 for the position x
- 2 for the direction of propagation of the photon Ω
- 1 more for the frequency of the photon ν

This quantity is directly proportionally to the number of photons found at time t in the volume dx, having a frequency in the interval of length $d\nu$, and flowing in the solid angle $d\Omega$. (Turpault, 2003).

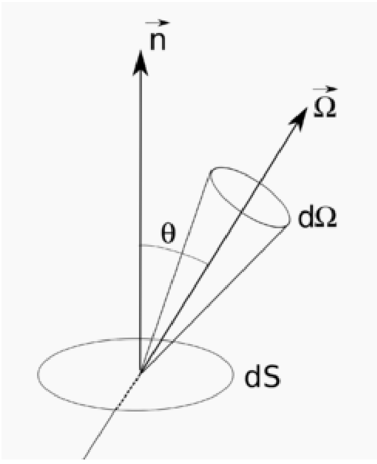


Figure 2: Solid angle used to define the specific radiative intensity

A good example of a homogeneous and isotropic intensity field is given by Planck's function:

$$B(\nu, T) = \frac{2h\nu^3}{c^2}\left[e^{\frac{h\nu}{kT}} - 1\right]^{-1}$$

Where h is Plank's constant, c is the speed of light, and k is Boltzmann's constant.

A few simplifications still have to be made before we can write the radiative transfer equation (RTE). Namely that we are at the **local thermodynamic equilibrium** (LTE). That is, the local medium (at the microscopic level) is at chemical and thermal equilibrium. Assuming matter in this state, we will write the opacities $\sigma_e, \sigma_a, \sigma_c$ as functions of just ρ and T. Let's also assume **radiative equilibrium** (which might occur in the absence of LTE); this is when matter and radiation are at equilibrium (namely $T_{radiation} = T_{matter}$). In this latter state, photons are distributed according to Planck's function at the medium's temperature.

The radiative transfer equation is obtained by making a balance of all the terms that affect the intensity of the light beam (absorption, emission, and scattering), at the microscopic level. We get (Franck, 2012, p.21):

$$\begin{aligned}
&\frac{1}{c}\frac{\partial}{\partial t} I(t,x,\Omega,\nu) + \Omega \cdot \nabla_{\mathrm{x}} I(t,x,\Omega,\nu) \\
&\qquad = \sigma_a(\rho,\Omega,\nu)\left(B(\nu,T) - I(t,x,\Omega,\nu)\right) \\
&\qquad + \frac{1}{4\pi}\int_0^{\propto}\int_{S^2} \sigma_c(\rho,\Omega,\nu)p(\Omega' \to \Omega)\big(I(t,x,\Omega',\nu) - I(t,x,\Omega,\nu)\big)\, d\Omega' \mathrm{d}\mathrm{v} \qquad (RTE)
\end{aligned}$$

In this equation, c is the speed of light, and p is the scattering's angular redistribution function. It is such that:

$$\oint p(\Omega' \to \Omega)\, d\Omega' = 1$$

It represents the probability of a photon being deviated from its original direction Ω' into our direction of interest Ω.

To obtain a kinetic P_1 model, we need to define 3 macroscopic quantities (the radiative energy, flux, and pressure) respectively as follows (Franck, p.16):

$$\begin{aligned}
E(t,x) &= \frac{4\pi}{c}\int_0^{\infty}\int_{S^2} I(t,x,\Omega,\nu)\, d\Omega dv \\
F(t,x) &= \frac{4\pi}{c}\int_0^{\infty}\int_{S^2} \Omega\, I(t,x,\Omega,\nu)\, d\Omega dv \\
P(t,x) &= \frac{4\pi}{c}\int_0^{\infty}\int_{S^2} \Omega \otimes \Omega\, I(t,x,\Omega,\nu)\, d\Omega dv
\end{aligned}$$

By computing the first and second moment of the intensity field I and taking into account that:

- Planck's function's flux is zero (since B is isotropic).
- B's integral on all frequencies ν equals aT^4 (with $a = \frac{4\sigma}{c}$, σ being the Stefan-Boltzmann constant).

We also need to link the pressure to the energy (using a closure equation like the Eddington approximation $P = \frac{1}{3}E$) (Dullemond, 2012); The RTE gives us equations (1.1) and (1.2).

Since the radiation phenomenon alone does not conserve energy, (1.1) and (1.2) need to be coupled with matter to get the conservation of energy. Luckily, we are only interested in the effects of the radiation on the matter, therefore we do not need to consider terms that are not affected by the radiation. Remembering that the probability of a photon being emitted is the same as it being absorbed (i.e. $\sigma_a = \sigma_e$ at LTE), this gives us equation (1.3). Our P_1 model is now complete:

$$\begin{cases} \partial_t E + c\partial_x F = c\sigma_a(aT^4 - E) & (1.1) \\ \partial_t F + \frac{c}{3}\partial_x E = -c\sigma_c F & (1.2) \\ \rho C_v \partial_t T = c\sigma_a(E - aT^4) & (1.3) \end{cases} \quad (1)$$

Where:

- $a = \frac{4\sigma}{c}$ with σ the Stefan−Boltzmann constant
- C_v is the thermal capacity of the medium
- c is the speed of light
- $T(t,x) > 0$ is the temperature of the medium
- $E(t,x) \in \mathbb{R}$ is the energy of the photons
- $F(t,x) \in \mathbb{R}$ is the flux of the photons
- $\rho(x) > 0$ is the density of the medium
- $\sigma_a(\rho,T) > 0$ is the absorption opacity
- $\sigma_c(\rho,T) > 0$ is the scattering opacity

b. Scheme's strengths

The P_1 scheme has numerous advantages; We will discuss the fact that it generalizes well under limiting conditions.

i. Transport limit

If the opacities are very weak i.e. $\sigma_a \approx \sigma_c \approx 0$, the photons travel through the medium without any interaction. Equations (1.1) and (1.2) lead to:

$$\partial_t(E+F) + c\partial_x\left(\frac{1}{3}E + F\right) = 0 \quad \text{and} \quad \partial_t(E-F) - c\partial_x\left(\frac{1}{3}E - F\right) = 0 \qquad (2.1)$$

These transport equations will be useful to validate our numerical scheme in the RESULTS section of this report.

ii. Diffusive limit

When the scattering opacity σ_c and the absorption opacity σ_a are high enough to be close to the speed of light c (which is far larger than the observed phenomenon's speed), the P_1 model is reduced to the following:

$$\partial_t(aT^4 + \rho C_v T) - \partial_x\left(\frac{c}{3\sigma_c}\partial_x aT^4\right) = O\left(\frac{1}{c}\right) \qquad (3.2)$$

This model is called the **diffusion approximation** (Franck, 2020, p.7).

Focusing on the energy, and considering a collision dominated medium ($\sigma_a = 0$ and $\sigma_c \approx c$), equation (1.2) can be reduced to the Fick law of diffusion:

$$F = -\frac{1}{3\sigma_c}\partial_x(cE)$$

Injecting that into (1.1), we get the **Eddington diffusion approximation** (Velarde et al., 1993, p.204):

$$\partial_t E - \partial_x\left(\frac{c}{3\sigma_c}\partial_x E\right) = 0 \qquad (4.3)$$

Both (2.2) and (2.3) match a diffusion equation with a diffusion coefficient of $\frac{c}{3\sigma_c} \approx O(1)$.

2. Solving the model with finite volumes

Due to the hyperbolic nature of the PDEs (1.1) and (1.2), the constraints of our domain, and the need to deal with discontinuities, the **finite volumes** method is well suited to solving the P_1 scheme. 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 a and b be the two real numbers such that $b > a$, and $N > 0$ an integer. We split the domain $[a, b]$ into N cells of equal length to obtain a uniform mesh. At the two edges, we add two "ghost" cells. In total, we have $N + 2$ cells. Let's denote by Δx the length of the intervals (the volume of the cells). For each cell $j \in [\![0, N + 1]\!]$, we write x_j its center, $x_{j-\frac{1}{2}}$ its left edge, and $x_{j+\frac{1}{2}}$ its right edge.

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

a. Step 1: The coupling part

We consider the "equilibrium" part of (1). That is, the photons are not moving, and we only consider the equations that are affected by matter (i.e. equations (1.1) and (1.3) where terms with the medium's temperature are involved) (Franck, 2012, p. 160). This leads to all the terms with ∂_x in (1) to be equal to 0. That equation becomes:

$$\begin{cases} \partial_t E = c\sigma_a(aT^4 - E) \\ \partial_t F = 0 \\ \rho C_v \partial_t T = c\sigma_a(E - aT^4) \end{cases} \tag{5}$$

Writing $\Theta = aT^4$, we solve (3) on each independent cell. The numerical scheme is given below.

$$\begin{cases} \dfrac{E_j^{q+1} - E_j^n}{\Delta t} = c\sigma_a\left(\Theta_j^{q+1} - E_j^{q+1}\right) \\ \dfrac{F_j^{q+1} - F_j^n}{\Delta t} = 0 \\ \rho_j C_v \mu_q \dfrac{\Theta_j^{q+1} - \Theta_j^n}{\Delta t} = c\sigma_a(E_j^{q+1} - \Theta_j^{q+1}) \end{cases}$$

Rewritten as:

$$\begin{cases} E_j^{q+1} = \alpha E_j^n + \beta \Theta_j^{q+1} \\ F_j^{q+1} = F_j^n \\ \Theta_j^{q+1} = \gamma \Theta_j^n + \delta E_j^{q+1} \end{cases}$$

Applying Cramer's rule, we get:

$$\begin{cases} E_j^{q+1} = \dfrac{\alpha E_j^n + \beta\gamma \Theta_j^n}{1 - \beta\delta} \\ F_j^{q+1} = F_j^n \\ \Theta_j^{q+1} = \dfrac{\gamma \Theta_j^n + \alpha\delta E_j^n}{1 - \beta\delta} \end{cases} \tag{6}$$

Where

- E_j^n, F_j^n and Θ_j^n are the value values of E_j, F_j and Θ_j on the cell at the beginning of the step.
- $\alpha = \frac{1}{\Delta t}\left(\frac{1}{\Delta t} + c\sigma_a\right)^{-1}$, $\beta = c\sigma_a\left(\frac{1}{\Delta t} + c\sigma_a\right)^{-1}$, $\gamma = \frac{\rho_j C_v \mu_q}{\Delta t}\left(\frac{\rho_j C_v \mu_q}{\Delta t} + c\sigma_a\right)^{-1}$, and $\delta = c\sigma_a\left(\frac{\rho_j C_v \mu_q}{\Delta t} + c\sigma_a\right)^{-1}$
- σ_a written above is a function of ρ_j and T_j^n. Thus, it is actually $\sigma_a(\rho_j, T_j^n)$.
- μ_q is such that $\mu_q = \dfrac{1}{T^{3,n} + T^n T^{2,q} + T^q T^{2,n} + T^{3,q}}$

Since this step is a fixed point method, we iterate on q until we get close enough to the fixed point $(E_j^*, F_j^*, \Theta_j^*)$, or more precisely (E_j^*, F_j^*, T_j^*). We then move to the next step.

b. Step 2: The hyperbolic part

Once the first step converges, we solve (1.2) and (1.3) as if the radiation weren't coupled with the matter, hence considering only the two hyperbolic partial differential equations in (1). We do this with the values of $E, F,$ and T on each cell updated from step 1. We write:

$$\begin{cases} \dfrac{E_j^{n+1} - E_j^*}{\Delta t} + c\dfrac{F_{j+\frac{1}{2}} - F_{j-\frac{1}{2}}}{\Delta x} = 0 \\ \dfrac{F_j^{n+1} - F_j^*}{\Delta t} + c\dfrac{E_{j+\frac{1}{2}} - E_{j-\frac{1}{2}}}{\Delta x} - cS_j' = cS_j \\ \rho_j C_v \mu_q \dfrac{T_j^{n+1} - T_j^*}{\Delta t} = 0 \end{cases} \tag{7}$$

With

$$F_{j+\frac{1}{2}} = M_{j+\frac{1}{2}}\left(\frac{F_{j+1}^n + F_j^n}{2} - \frac{E_{j+1}^n - E_j^n}{2}\right), \quad F_{j-\frac{1}{2}} = M_{j-\frac{1}{2}}\left(\frac{F_j^n + F_{j-1}^n}{2} - \frac{E_j^n - E_{j-1}^n}{2}\right)$$

$$E_{j+\frac{1}{2}} = M_{j+\frac{1}{2}}\left(\frac{E_{j+1}^n + E_j^n}{2} - \frac{F_{j+1}^n - F_j^n}{2}\right), \quad E_{j-\frac{1}{2}} = M_{j-\frac{1}{2}}\left(\frac{E_j^n + E_{j-1}^n}{2} - \frac{F_j^n - F_{j-1}^n}{2}\right)$$

$$S_j = -\frac{1}{2}\left(M_{j+\frac{1}{2}}\sigma_{j+\frac{1}{2}} + M_{j-\frac{1}{2}}\sigma_{j-\frac{1}{2}}\right)F_j^{n+1}, \quad S_j' = \frac{1}{\Delta x}\left(M_{j+\frac{1}{2}} - M_{j-\frac{1}{2}}\right)E_j^n$$

$$M_{j+\frac{1}{2}} = \frac{2}{2 + \Delta x \sigma_{j+\frac{1}{2}}}, \quad M_{j-\frac{1}{2}} = \frac{2}{2 + \Delta x \sigma_{j-\frac{1}{2}}}$$

$$\sigma_{j+\frac{1}{2}} = \frac{1}{2}\left(\sigma_c(\rho_j, T_j^n) + \sigma_c(\rho_{j+1}, T_{j+1}^n)\right), \quad \sigma_{j-\frac{1}{2}} = \frac{1}{2}\left(\sigma_c(\rho_{j-1}, T_{j-1}^n) + \sigma_c(\rho_j, T_j^n)\right)$$

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

$$\Delta t < \frac{\Delta x}{c}$$

We can rewrite (5) as:

$$\begin{cases} E_j^{n+1} = E_j^* + \alpha\left(F_{j+\frac{1}{2}} - F_{j-\frac{1}{2}}\right) \\ F_j^{n+1} = \beta F_j^* + \gamma E_j^n + \delta\left(E_{j+\frac{1}{2}} - E_{j-\frac{1}{2}}\right) \\ T_j^{n+1} = T_j^* \end{cases} \tag{8}$$

With

$$\alpha = -\frac{c\Delta t}{\Delta x},\quad \beta = \frac{1}{\Delta t}\left(\frac{1}{\Delta t} + \frac{c}{2}\left(M_{j+\frac{1}{2}}\sigma_{j+\frac{1}{2}} + M_{j-\frac{1}{2}}\sigma_{j-\frac{1}{2}}\right)\right)^{-1},\quad \gamma = \frac{c}{\Delta x}\left(M_{j+\frac{1}{2}} - M_{j-\frac{1}{2}}\right)\left(\frac{1}{\Delta t} + \frac{c}{2}\left(M_{j+\frac{1}{2}}\sigma_{j+\frac{1}{2}} + M_{j-\frac{1}{2}}\sigma_{j-\frac{1}{2}}\right)\right)^{-1}$$

$$\text{and}\ \ \delta = -\frac{c}{\Delta x}\left(\frac{1}{\Delta t} + \frac{c}{2}\left(M_{j+\frac{1}{2}}\sigma_{j+\frac{1}{2}} + M_{j-\frac{1}{2}}\sigma_{j-\frac{1}{2}}\right)\right)^{-1}$$

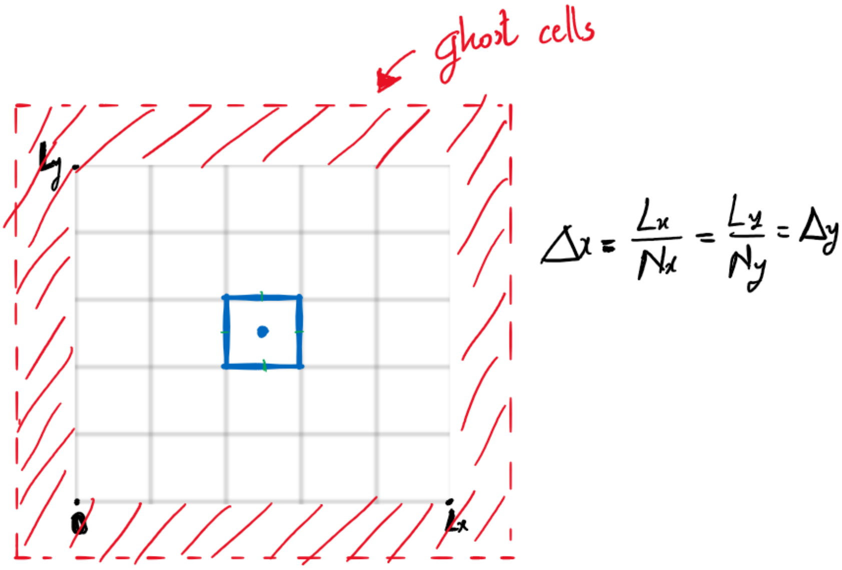
Once all of this is done, we move back to step 1, then back to step 2, and so on until we reach a predefined time t_f at which we want the solution.

c. Step 3: The hyperbolic part in 2D

In 2D, $\mathbf{F} = (F_1, F_2)$ becomes a vector, and equations (1.1) and (1.2) become

$$\begin{cases} \partial_t E + c\nabla\cdot\mathbf{F} = 0 & (1.1) \\ \partial_t \mathbf{F} + c\nabla\cdot E = -c\sigma_c\mathbf{F} & (1.2) \end{cases}$$

Let's apply the finite volumes method where control volumes are identical with the grid cells. Let's use the following uniform grid ($\Delta x = \Delta y$) for discretization:



Let Ω_j be the volume of cell j and S_j its surface:

$$\begin{cases} \partial_t \int_{\Omega_j} E + c\int_{\Omega_j} \nabla\cdot\mathbf{F} = 0 \\ \partial_t \int_{\Omega_j} \mathbf{F} + c\int_{\Omega_j} \nabla\cdot E = -c\sigma_c \int_{\Omega_j}\mathbf{F} \end{cases}$$

Using the divergence theorem, we get:

$$\begin{cases} \partial_t \int_{\Omega_j} E + c\int_{S_j} (\mathbf{F},\mathbf{n})\,dS = 0 \\ \partial_t \int_{\Omega_j} \mathbf{F} + c\int_{S_j} E\mathbf{n}\,dS = -c\sigma_c \int_{\Omega_j}\mathbf{F} \end{cases}$$

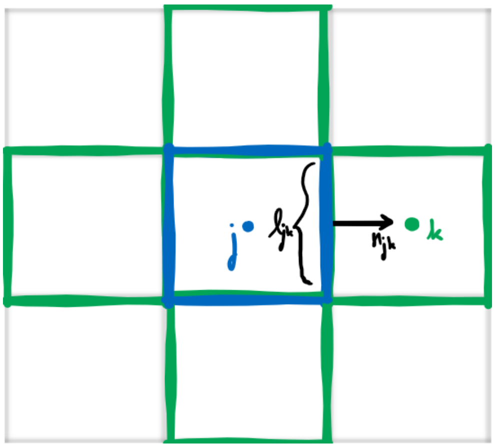
Averaging the values for $E(t,x)$ and $\mathbf{F}(t,x)$ over the cell, we have:

$$\begin{cases} \partial_t E_j + \frac{c}{|\Omega_j|}\int_{S_j} (\mathbf{F},\mathbf{n})\,dS = 0 \\ \partial_t \mathbf{F}_j + \frac{c}{|\Omega_j|}\int_{S_j} E\mathbf{n}\,dS = -\frac{c\sigma_c}{|\Omega_j|} \int_{\Omega_j}\mathbf{F} \end{cases}$$

Where:

$$E_j(t) = \frac{1}{|\Omega_j|}\int_{\Omega_j} E\,(t,x)$$
$$\mathbf{F}_j(t) = \frac{1}{|\Omega_j|}\int_{\Omega_j} \mathbf{F}\,(t,x)$$

As for the discretization, let us consider a neighbouring cell k, we define:



$$\begin{aligned}
(\mathbf{F}_{jk}, \mathbf{n}_{jk}) &= l_{jk} M_{jk} \left(\frac{\mathbf{F}_j^n \cdot \mathbf{n}_{jk} + \mathbf{F}_k^n \cdot \mathbf{n}_{jk}}{2} + \frac{E_k^n - E_j^n}{2} \right) \\
(E_{jk}\mathbf{n}_{jk}) &= l_{jk} M_{jk} \left(\frac{E_j^n + E_k^n}{2} + \frac{\mathbf{F}_k^n \cdot \mathbf{n}_{jk} - \mathbf{F}_j^n \cdot \mathbf{n}_{jk}}{2} \right) \mathbf{n}_{jk} \\
\mathbf{S}_j' &= \frac{1}{|\Omega_j|} \left(\sum_k l_{jk} M_{jk}\, \mathbf{n}_{jk} \right) E_j^n \\
\mathbf{S}_j &= -\left(\sum_k M_{jk}\sigma_{jk} \right) \mathbf{F}_j^{n+1} \\
M_{jk} &= \frac{2}{2 + \Delta x \sigma_{jk}} \\
\sigma_{jk} &= \frac{1}{2}\left(\sigma_c(\rho_j, T_j^n) + \sigma_c(\rho_k, T_k^n) \right)
\end{aligned}$$

Finally, we can write the scheme as:

$$\begin{cases}
\dfrac{E_j^{n+1} - E_j^*}{\Delta t} + \dfrac{c}{|\Omega_j|} \displaystyle\sum_k (\mathbf{F}_{jk}, \mathbf{n}_{jk}) = 0 \\
\dfrac{\mathbf{F}_j^{n+1} - \mathbf{F}_j^*}{\Delta t} + \dfrac{c}{|\Omega_j|} \displaystyle\sum_k (E_{jk}\mathbf{n}_{jk}) - c\mathbf{S}_j' = c\mathbf{S}_j
\end{cases}$$

Taking into account step 1, we can have:

$$\begin{cases}
E_j^{n+1} = E_j^* + \alpha \displaystyle\sum_k (\mathbf{F}_{jk}, \mathbf{n}_{jk}) \\
\mathbf{F}_j^{n+1} = \beta \mathbf{F}_j^* + \boldsymbol{\gamma} E_j^n + \delta \displaystyle\sum_k (E_{jk}\mathbf{n}_{jk}) \\
T_j^{n+1} = T_j^*
\end{cases} \tag{9}$$

With

$$\alpha = -\frac{c\Delta t}{|\Omega_j|}, \quad \beta = \frac{1}{\Delta t}\left(\frac{1}{\Delta t} + \sum M_{jk}\sigma_{jk}\right)^{-1}, \quad \boldsymbol{\gamma} = \frac{c}{|\Omega_j|}\left(\sum l_{jk}M_{jk}\mathbf{n}_{jk}\right)\left(\frac{1}{\Delta t} + \sum M_{jk}\sigma_{jk}\right)^{-1}$$
$$\text{and } \delta = -\frac{c}{|\Omega_j|}\left(\frac{1}{\Delta t} + \sum M_{jk}\sigma_{jk}\right)^{-1}$$

We still need the CFL condition for stability.

$$\Delta t < \frac{\Delta x}{c}$$

3. Analyzing the data

a. Step 1: Data visualization

After exporting the data from the simulations in the previous step, we will need to study them using appropriate descriptive statistics tools. We will then have to find meaningful correlations between the variables. During this step, we will try to show the results that the neural network will later find by itself. The step will serve as a preparation for the AI training and predicting phase (step 2 and step 3 below), to be completed during a subsequent internship.

b. Step 2: The density

Here we will make the AI understand the trends we found in the previous step. Knowing not only the signals $E(t,x)$, $F(t,x)$, and $T(t,x)$ at all times, but also the scattering and absorption opacities ($\sigma_c(\rho,T)$ and $\sigma_a(\rho,T)$), we will rebuild the domain's density $\rho(x)$. All the inputs for the neural network will be 1D tensors (indexed either by t or by x) and the same goes for the outputs.

c. Step 3: 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 $\sigma_c(\rho,T)$ and $\sigma_a(\rho,T)$ with $\rho(x)$ known. However, it requires a more complex neural network.

Just as the previous step, this part is no longer part of the project, we will cover it during the internship.

IV. RESULTS

1. Test cases

a. Case 1: Transport limit

This test case corresponds to the transport limit (equation (2.1)). We are interested in the signals $E+F$ and $E-F$. To perform this test, we take $\sigma_a = \sigma_c = 0$ and the initial condition for E is a gaussian distribution with its pic at the center of the domain. All the parameters for the problem are indicated in the configuration ***src/config/case_1.cfg***. Those parameters are repeated below:

```
x_min 0
x_max 1
N 100

c 1
a 1
C_v 1
CFL 0.99
precision 1e-4
t_0 0
t_f 0.5

rho 1
sigma_a 0
sigma_c 0

E_0 exp(-((x-0.5)^2)/(2*(0.05^2)))
F_0 0
T_0 1

E_l neumann
F_l neumann
T_l neumann

E_r neumann
F_r neumann
T_r neumann

E_exact 0
F_exact 0
T_exact 0

export_spatial data/df_spatial.csv
export_temporal data/df_temporal.csv
```

Figure 3: Configurations for the transport test case. The "neumann" term indicates the natural flow condition on the boundaries.

Detailed instructions on writing a configuration file are given in the ***README.md*** file at the root of the Github repository associated with this project. Now, the next step is to run the program using the command ***build/transfer src/config/case_1.cfg*** and visualize the results in ***data/case_1_spatial.csv*** .

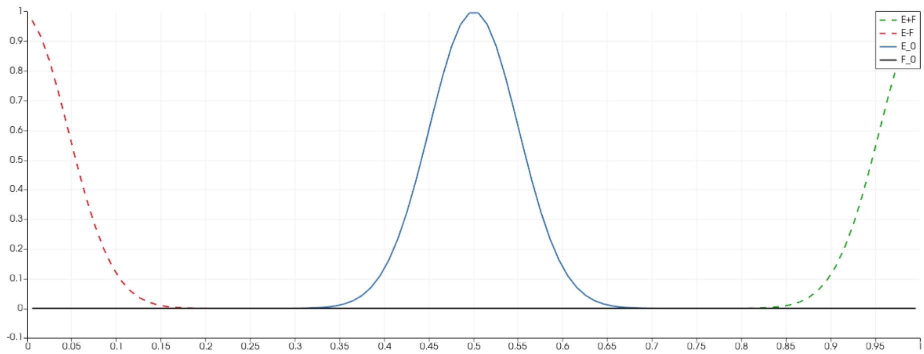


Figure 4: Transport test case. E+F is transported at the speed c=1 while E-F is transported at the speed -c=-1

b. Case 2: Diffusion limit

We will now test the diffusion limit corresponding to equation (2.3). With the scattering opacity and the speed of light taken considerably greater than the absorption opacity ($\sigma_a = 0$ and $\sigma_c = c = 1000$), the final photon energy E should be the fundamental solution to the diffusion equation:

$$\begin{cases} \dfrac{\partial}{\partial t}E(t,x) - \dfrac{\partial^2}{\partial x^2}E(t,x) = 0, & t > 0, \quad x \in [0,1] \\ E_0(x) = E(0,x) = \delta_0(x) \end{cases}$$

Written as:

$$E(t,x) = \frac{1}{\sqrt{4\pi t}} e^{-\frac{\left(x-\frac{1}{2}\right)^2}{4t}}$$

There is a problem though, placing a Dirac function is computationally difficult (since it equals $+\infty$ on its domain). To solve this, we modify that initial conditions by shifting the time variable by $t_0 = 0.002$. The initial condition becomes:

$$E_0(x) = E(t_0, x) = \frac{1}{\sqrt{4\pi t_0}} e^{-\frac{\left(x-\frac{1}{2}\right)^2}{4t_0}}$$

Configurations for this test case are given below:

```
c 1000
sigma_c 1000

x_min 0
x_max 1
N 100

a 1
C_v 1
CFL 0.99
precision 1e-6
t_0 0.002
t_f 0.02

rho 1
sigma_a 0

E_0 exp(-((x-0.5)^2)/(4*1*(t_0)))/(2*sqrt(_pi*1*(t_0)))
F_0 0
T_0 0

E_exact exp(-((x-0.5)^2)/(4*1*(t+t_0)))/(2*sqrt(_pi*1*(t+t_0)))
F_exact 0
T_exact 1

E_l neumann
F_l neumann
T_l neumann

E_r neumann
F_r neumann
T_r neumann

export_spatial data/df_spatial.csv
export_temporal data/df_temporal.csv
```

Figure 5: Configurations for the diffusion limit test case

After running the command ***build/transfer src/config/case_2.cfg***, the results will be exported into ***data/case_2_spatial.csv*** and ***data/case_2_temporal.csv*** .

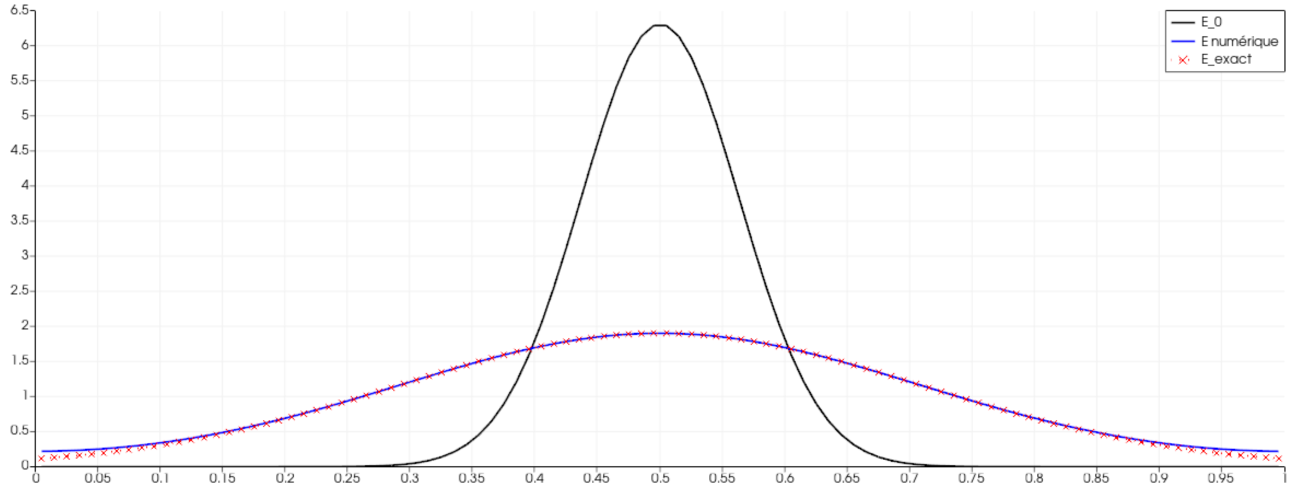


Figure 6: Spatial visualisation of the numerical and the exact solution for the diffusion limit

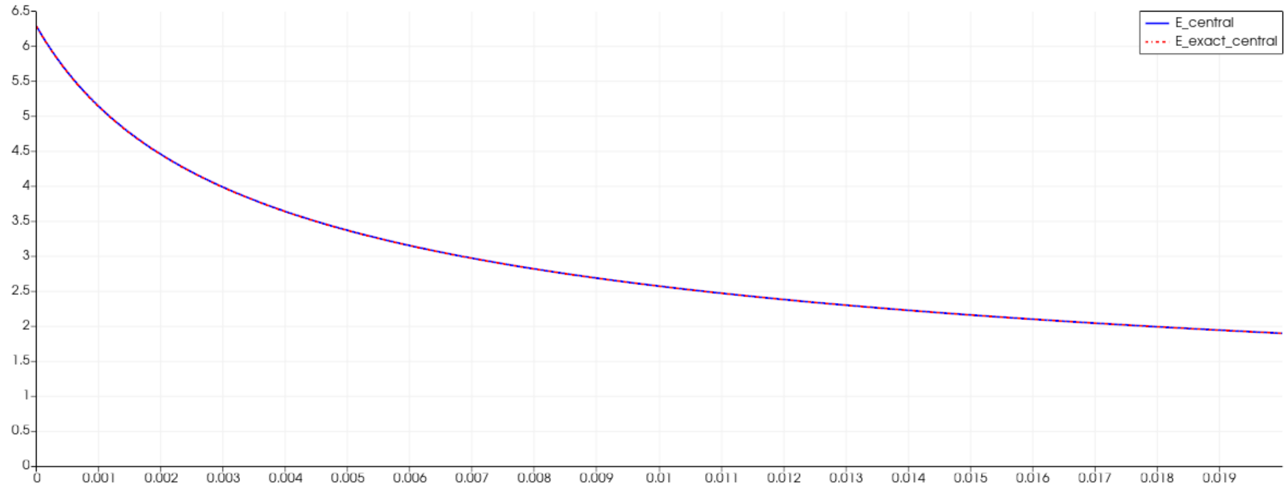


Figure 7: Temporal decrease in energy at the center of the domain for the diffusion limit

As we see (and hoped), our computed solution (in solid line) perfectly matches the expected solution (in dashed line).

c. <u>Case 3: Olson-Auer-Hall</u>

Contrary to the previous test cases, this one focuses on temperature. It tells the evolution of the radiation's and the medium's temperature. The domain $[0,3]$ is heated by a blackbody to the left (with $T_r = 0.056234\ keV$). As the body gets hotter, its absorption opacity increases, causing thermal equilibrium between the radiation and the matter (Franck, 2012, p.171). The configurations for this case can be found in ***src/config/case_3.cfg***.

```
rho 0.38214
sigma_a (1)/(T^3)
sigma_c 0

x_min 0
x_max 3
N 1000

c 172.6277
a 0.01372
C_v 0.14361
CFL 0.5
precision 1e-8
t_0 0
t_f 3.34448e-4

E_0 0.01372*(0.056234^4)
F_0 0
T_0 0.056234

E_1 0.01372*(1^4)
F_1 0
T_1 1

E_r neumann
F_r neumann
T_r neumann

E_exact 0
F_exact 0
T_exact 0

export_temporal data/df_temporal.csv
export_spatial data/df_spatial.csv
```

Figure 8: Configuration for the Olson-Auer-Hall test case

The results below are plotted in log scale for the abscissa x.

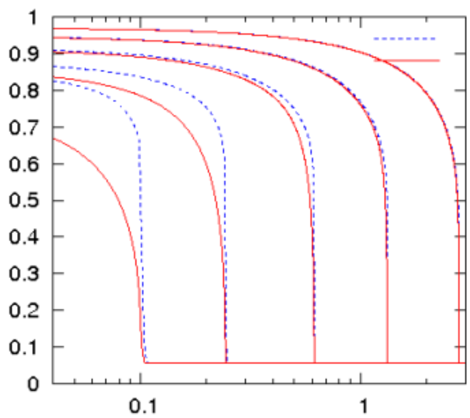


Figure 9: Expected results for t=1/c, 3/c, 10/c, 30/c, and 100/c (Franck, 2012, p.171)

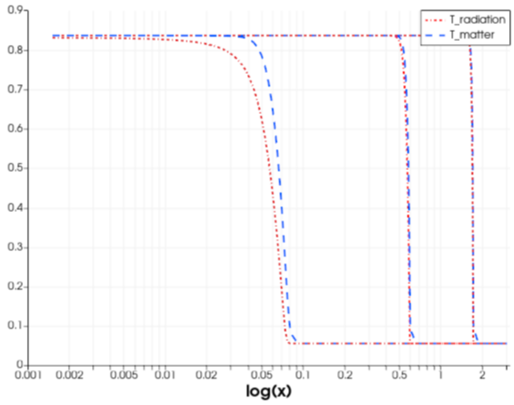


Figure 10: Obtained results for t=0.1/c, 1/c, and 3/c

As we can see, the result doesn't exactly match the reference. This is probably due is a problem in the speed of the phenomenon. This issue will be addressed in subsequent studies.

2. Data frames

The notebook ***src/notebook/analyse_des_donnees.ipnb*** analyses data (in the form of data frames) created from the program, and explains some trends (the Colab version is recommended as the Jupyter version indicated above is locally stored in the repository and will easily run into errors). We have repeated some of the results below. The data was created with the bash script ***src/simu/gauss_dump.sh***. Apart from ρ, σ_a and σ_c that change, the remaining parameters are constants. Particularly, the initial value of the signal E is the same gaussian function we used in the first test case. Initial values for F and T are respectively 0 and 1.

a. Influence of the density

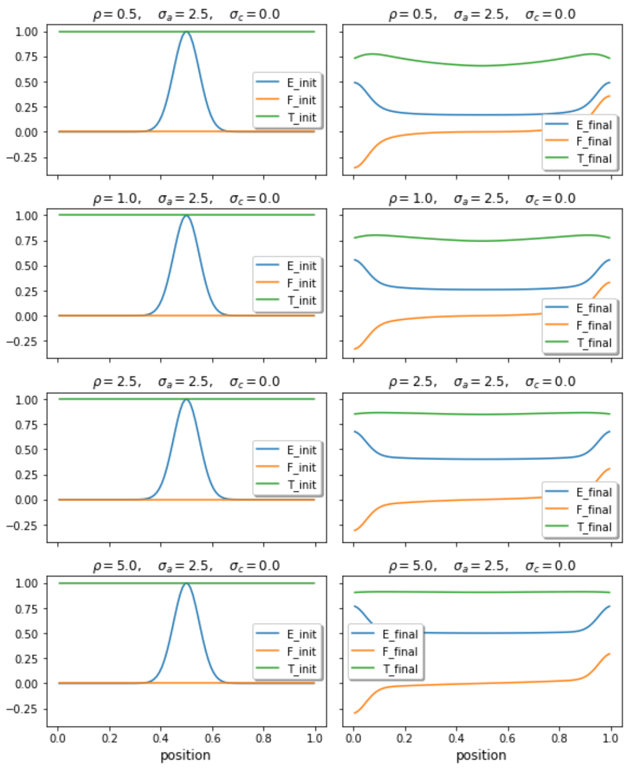


Figure 11: Influence of the density on E, F, and T for $\sigma_a = 2.5$ *and* $\sigma_c = 0$

It seems that as the density increases (Figure 11):

- The radiative energy E increases
- It doesn't change the flux F
- The temperature varies less. The equilibrium temperature seems to be higher.

b. Influence of the absorption opacity

It seems that as the opacity σ_a increases (Figure 12):

- The diffusion phenomenon is faster, but the energy E keeps the same pic value
- The flux F is attenuated during transport
- The temperature change is more important, and the thermal equilibrium is faster

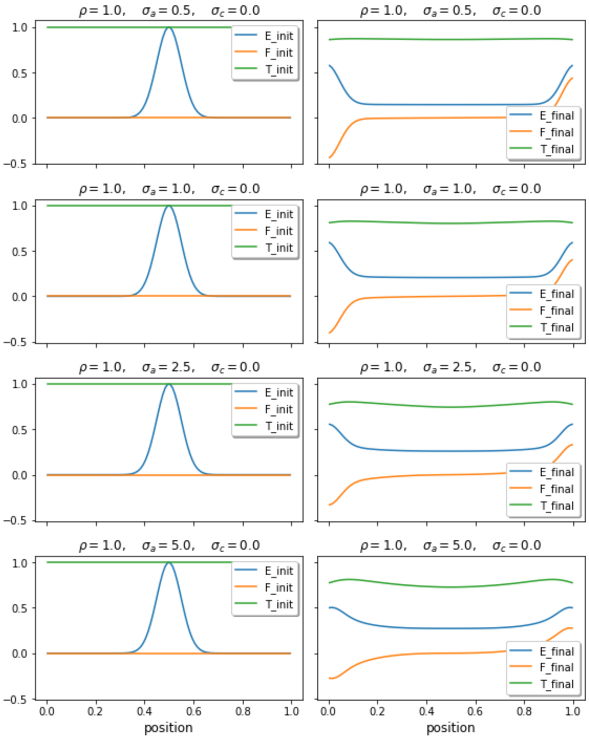


Figure 12: Influence of σ_a for $\rho = 1$ and $\sigma_c = 0$

c. <u>Influence of the scattering opacity</u>

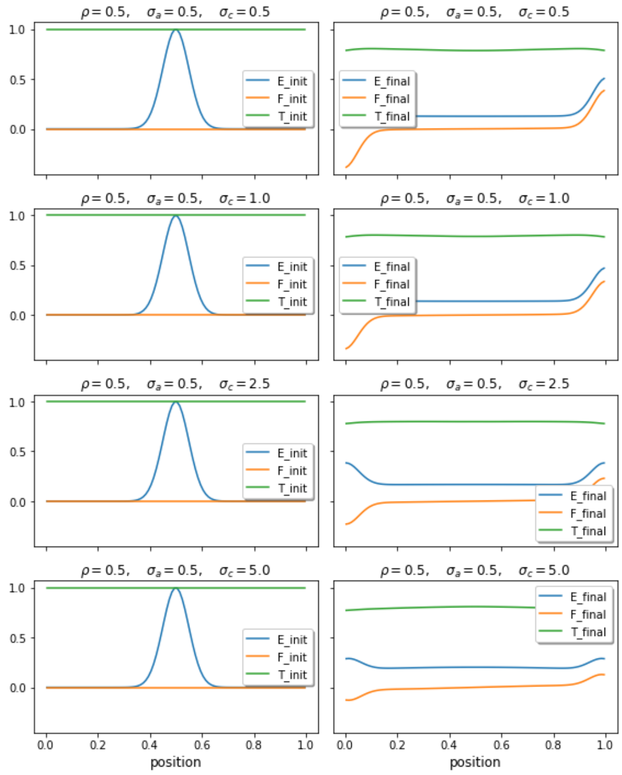


Figure 13: Influence of σ_c for $\rho = 0.5$ and $\sigma_a = 0.5$

From figure 13, it seems obvious that an increase in σ_c decreases the values of both the energy and the flux. However, to the naked eye, it is less obvious to notice any change in temperature.

Of course, these are but conjecture. Plus, the opacities usually change with the density and temperature of the medium (see test case 3). This means that their form is not as simple as constants as we assumed. However, this simple analysis shows potential trends that might be confirmed or invalidated by a convoluted neural network.

V. MILESTONES

This section gives a summary of the tasks, the goals, and the deadlines for the project. It also shows how the project has changed since it was originally assigned in late March 2020. Original definitions and time estimates are written in black, and changes are marked in red.

Milestone		Details	Tools needed	Deadline	Estimated number of hours	Effective number of hours
Understanding the model		We will seek to understand the PDE (1) and its numerical model on a slightly theoretical basis. This has already been done, but new information keeps coming every day.		April 2, 2020	8 hours	20 hours
Report version 0		A report indicating the context and the roadmap for the project. This milestone is currently under completion.	MS Word	April 15, 2020	16 hours	32 hours
Solving the splitting scheme	Solving the scheme	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~~ muParser	April 15, 2020	40 hours	60 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 (2.2) Tests will be put in place for continuous integration. We will need to find and correct all the bugs that appear in the code.	VS Code Paraview	April 29, 2020	16 hours	60 hours
	Benchmarking	Compare our algorithm to known solutions 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	4 hours
	Validation	Making sure the problem solves the correct direct problems linked to medical tomography.	VS Code Paraview	~~April 29,~~ May 22, 2020	4 hours	16 hours

	Data export	Writing and running a script that exports thousands of instances of a correctly solved direct problem. We will make sure to export the data one state of the model at a time. 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	16 hours
Analyzing the data	Studying the data	Using the exported data, we will seek to learn new information, finding interesting correlations, creating new variables, deleting outliers, cleaning out the data, and so on. During the internship, our goal will be to make the AI understand these correlations.	Google Colab Jupyter	May 19, 2020	24 hours	8 hours
	Verification	Verify that the model is properly studied. A positive indicator might be that when generating new instances of the data, we continue seeing the same trends.	Google Colab Jupyter	May 19, 2020	16 hours	4 hours
	Validation	Check that the trends we find are effectively the trends from light spreading in medical tomography. We will compare our data trends to known databases in the same area of study.	Google Colab Jupyter	May 19, 2020	4 hours	--
	Benchmarking	Check that our algorithms for analysis are fast enough to be easily repeatable 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~~ 22, 2020	8 hours	16 hours
Report final version		The final version of the report, incorporating corrections indicated by the supervisors.	MS Word	May ~~19~~ 27, 2020	4 hours	--
Presentation		A slideshow to be written in PowerPoint.	MS PowerPoint	May ~~19~~ 28, 2020	8 hours	--

The initial estimate of time to be spent on the project was **164 hours** (approximately 7 full days. However, the effective time spent on the project (which is yet to be completed) is closer to 232 hours (or 10 full days).

VI. CHALLENGES

We faced multiple challenges during this project. The main challenge was searching for information concerning the topic of light scattering coupled with matter. The radiative transfer equation is used in multiple fields (stellar atmosphere in astrophysics, nuclear fusion by inertial confinement, etc...). This led to multiple notation and nomenclatures, that became confusing at times. Even in documents that discussed our splitting scheme, a scaling factor was introduced. This is the reason we have so few test cases to validate our work.

Another challenge was linked to performance in the C++ implementation of the scheme. Speed is very important in this problem. Especially since we plan on generating a huge amount of data to be analyzed (potentially millions of lines of signals). Earlier versions of the program lacked the speed necessary to do so. The program was bloated by external libraries, especially the ones for parsing string expressions into functions. The problem has since been solved by changing from ***exprtk*** to ***muParser***. Moreover, other optimization steps that can be viewed in the code for the ***Solver*** class have proven efficient.

VII. CONCLUSION

During this project, we saw how their energy is affected when photons travel through a medium (test cases 1 and 2). We have also been able to observe how the radiative temperature evolves as it travels through a medium at radiative equilibrium (test case 3). The satisfying results encourage us to design a follow-up to this project. We will train a neural network that will then predict the optical properties of the medium.

That said, it is important to note that our numerical scheme heavily depends on the P_1 model for the radiative transfer equation. While not costly and relatively easy to solve (especially with plane 1D geometries in both optically thick and optically thin media), the P_1 model has a major disadvantage: it only gives accurate predictions when the anisotropy factor (normally $f \leq 1$) is reduced, i.e.:

$$f = \frac{\|F\|}{cE} \leq \; 0.57 \qquad \text{(Turpault, 2003, p.22)}$$

Hence the lamination of the flux, one of the radiative transfer's most fundamental properties might not be satisfied. This shortcoming will be addressed during an internship and other methods like the P_n, or M_1 model will be tested.

VIII. 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
- Turpault, R. (2003). "*Modelisation, approximation numerique et applications du transfert radiatif en desequilibre spectral couple avec l'hydrodynamique*". Mathématiques [math]. Université Sciences et Technologies - Bordeaux I, 2003. Français. tel-00004620
- Dullemond, C. P. (2012)." *What makes radiative transfer hard, and how to solve it – An introduction* ". Retrieved from http://www.ita.uni-heidelberg.de/~dullemond/lectures/radtrans_2012/Chapter_4.pdf

- Velarde, G., Ronen, Y., Martinez-Val, J, M. (1993). "*Nuclear Fusion by Inertial Confinement: A Comprehensive Treatise* ". Chapter 8. Retrieved from https://books.google.fr/books?id=wfhBW2P3ObkC&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false
- Ansys Fluent. (n.d). "Radiative Transfer Equation". Retrieved from https://www.afs.enea.it/project/neptunius/docs/fluent/html/th/node111.htm