AMERICAN UNIVERSITY OF BEIRUT

IDENTIFICATION OF GAS DIFFUSION COEFFICIENTS IN POLAR FIRN

by
SARA DARWICH MAAD

A thesis
submitted in partial fulfillment of the requirements
for the degree of Master of Science
to the Department of Mathematics
of the Faculty of Arts and Sciences
at the American University of Beirut

Beirut, Lebanon
September 2022

AMERICAN UNIVERSITY OF BEIRUT

IDENTIFICATION OF GAS DIFFUSION COEFFICIENTS IN POLAR FIRN

by

SARA DARWICH MAAD

Approved by:

Dr. Nabil Nassif, Professor — Advisor

Mathematics

Dr. Sophie Moufawad, Assistant Professor — Co-Advisor

Mathematics

Ahmad Sabra

Dr. Ahmad Sabra, Assistant Professor — Member of Committee

Mathematics

Dr. Faouzi Triki, Professor — Member of Committee

Grenoble-Alpes University, France

Mathematics

Date of thesis defense: September 2, 2022

AMERICAN UNIVERSITY OF BEIRUT

THESIS RELEASE FORM

Student Name: Maad Sara Darwich

Last First Middle

I authorize the American University of Beirut, to: (a) reproduce hard or electronic copies of my thesis; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes

___✓ **As of the date of submission of my thesis**

___ **After 1 year from the date of submission of my thesis .**

___ **After 2 years from the date of submission of my thesis .**

___ **After 3 years from the date of submission of my thesis .**

Sara Maad September 9, 2022

Signature Date

ACKNOWLEDGEMENTS

Finally, after a long journey of eight months with a lot of challenges, my thesis is complete!
Undertaking this topic has been a truly life-changing experience for me, and it would not have been possible without the support and guidance that I received from many people.

Special thanks must first go to my advisor, Prof. Nabil Nassif, for his support, motivation, enthusiasm, and immense knowledge throughout the whole process of completing this work. He is the reason that I have chosen this field in mathematics.
My sincere appreciation goes to my co-advisor, Dr. Sophie Moufawad, for giving me invaluable guidance, insights, moral support, and directions in writing this thesis. Thank you for all your time and for being there for me.

I would like to thank Prof. Faouzi Triki for his help and the opportunity that he provided by proposing this topic. I also want to thank Dr. Ahmad Sabra for his insightful comments.

I am extremely grateful to my friends, family, and my fiance for their love, care, encouragement, and continuous support to complete this thesis. I am very lucky to have them by my side.

ABSTRACT
OF THE THESIS OF

Sara Darwich Maad for Master of Science
Major: Mathematics

Title: Identification of Gas Diffusion Coefficients in Polar Firn

The use of data analysis cooperatively with partial differential equations is a successful technique to estimate the main parameters of diverse phenomena in many fields (medicine, biology, ecology,...). In the area of ecology, analysis of historical climate change data that leads to global warming, necessitates an estimation of various atmospheric gas concentrations, primarily CO2. In the polar regions of Greenland (Denmark) and the southern Antarctic, it is possible to retrace the histories of several atmospheric gases over the last centuries using currently obtained data through the examination of the air volumes injected into the open porosity of the Firn (compacted snow).

This thesis uses powerful modern techniques of computational mathematics based on studying an appropriate inverse problem. It consists in starting with the direct problem, where we analyze and numerically simulate a time-dependent partial differential equation that models the Firn, given its diffusion coefficients. Once the direct problem is properly solved via a robust `MATLAB` software, one then looks at recovering the diffusion coefficient on the basis of current Firn measurements. Inverse techniques reduce to solving a minimization problem on a constrained set, solved also using efficient `MATLAB` toolboxes. Successful results obtained from numerical simulations conducted on the direct and inverse problems validate the feasibility of this method to estimate Firn diffusion coefficients.

Table of Contents

ILLUSTRATIONS

TABLES

Chapter 1

Introduction and Literature Review

1.1 The Firn Problem

Ozone-depleting substances such as the chlorofluorocarbons (CFCs) emitted by human activities (refrigeration, air-conditioning, packing material...) cause large-scale damage to the stratospheric ozone layer, and warm the earth's lower atmosphere, yielding a change in global climate. Consequently, their production has been phased out by the Montreal protocol in 1987. The first-stage replacements for CFCs were Fluorinated gases (HFCs or hydrofluorocarbons). Furthermore, the emissions of F-gases are rapidly increasing. These potent greenhouse gases have an impact on global warming that is up to 23,000 times larger than that of carbon dioxide (CO2) [1]. Therefore, it is essential to have reliable monitoring and modeling techniques of historical gas emission concentrations that can provide future estimations of climate change and reduction strategies.

The historical background of atmospheres and climates could be explored in the Polar ice and snow collected from Greenland and Antarctic. It is feasible to retrace the histories of several atmospheric gases over the last centuries using obtained data from the examination of the air volumes injected into the open porosity of the Firns (compacted snow). The interpretation of the obtained data can be achieved through complicated mathematical modeling on grounds of a good understanding of the mechanics that control gas trapping in polar ice, consequently the densification and pore closure in Firns, usually over the top hundred meters of ice.

This thesis goal is to solve the mathematical model that has been derived in [1], [2], [3], [4] and produce robust and efficient solvers for reconstructing gas histories.

1.2 Mathematical Models

Considering the mass conservation equations, the concentration ρ_α^o of a gas α in open pores satisfies an initial-value, time-dependent advection-diffusion partial differential equation on a one-space dimension segment $[0, z_F]$ with Dirichlet boundary condition at 0 and a mixed one at z_F, for $z \in (0, z_F)$, $t > 0$:

$$\begin{cases} \dfrac{\partial}{\partial t}\left[\rho_\alpha^o f\right] + \dfrac{\partial}{\partial z}\left[\rho_\alpha^o f(v + w_{air})\right] + \rho_\alpha^o(\tau + \lambda) = \dfrac{\partial}{\partial z}\left[D_\alpha\left(\dfrac{\partial \rho_\alpha^o}{\partial z} - \rho_\alpha^o \dfrac{M_\alpha g}{RT_m}\right)\right], \\ \rho_\alpha^o(0,t) = \ \rho_\alpha^{atm}(t), \quad t > 0, \\ \dfrac{\partial \rho_\alpha^o(z_F, t)}{\partial z} - \dfrac{M_\alpha g}{RT_m}\rho_\alpha^o(z_F, t) = 0, \quad t > 0. \end{cases} \tag{1.1}$$

where $\rho_\alpha^{atm}(t)$ is the concentration of gas in the atmosphere (mol/m^3 of void space), $D_\alpha(z)$ is the effective diffusion coefficient of the gas α in Firn (m^2/yr) which will be considered a decreasing function and is given by:

$$D_\alpha(z) = r_\alpha c_f D_{CO2,air}(z) \tag{1.2}$$

where c_f and r_α are known constants. The constants of the model are summarized in the Table 1.1.

z_F	the depth of the Firn
f	the average volume fraction in the open pores
v	the average descending speed in the Firn
w_{air}	the average speed of the air
τ	the mass exchange rate between open and closed pores ($/yr$)
λ	the rate of radioactive decay ($/yr$)
M_α	the molar mass of the gas (kg/mol)
g	the gravitational acceleration
R	the universal constant of ideal gases ($J/mol/K$)
T_m	the mean temperature of the Firn (K)

Table 1.1: The description of the model's constants.

1.3 Objectives

The ultimate goal of this thesis is to determine the diffusion coefficient D_α of a particular gas, using data from measurements $\rho_\alpha^o(z, T)$, $z \in (0, z_F)$ made of

several gases at the end time T. To find the corresponding D_α, it is sufficient by (1.2) to study an inverse problem that finds $D_{CO2,air}$.
We seek then $D(z) \equiv D_{CO2,air}(z) \in X$, where X is either:

- $X_u = C(0, z_F)$ for unconstrained optimization,

  or

- $X_c = \{v \in C(0, z_F) \,|\, v > 0\}$ for constrained optimization.

In this thesis, we will use X_c, since the diffusion coefficients need to be positive.

To introduce the inverse problem, we let $\rho^o_\alpha(\tilde{D}; ., T)$ be the unique solution of the direct problem at the end time T, $\forall \tilde{D} \in X_c$, and $\rho^o_{\alpha,meas}(., T)$ be the measured concentration at the end time T.
And we define the following objective function:

$$\forall \tilde{D} \in X_c : V(\tilde{D}) = \sum_{\alpha \in S} \left\| \rho^o_\alpha(\tilde{D}; ., T) - \rho^o_{\alpha,meas}(., T) \right\|_2^2$$

where S is the set of all the gases in the Firn.
So we seek $D \in X_c$ such that:

$$V(D) = \min_{\tilde{D} \in X_c} V(\tilde{D})$$

1.4 Main Results

This thesis covers 5 chapters including the introduction. In chapter 2, we apply a variation method to our problem and study the existence and uniqueness of the solution by summarizing the theoretical results obtained by S. Moufawad, N. Nassif, and F. Triki in their recent work [5].

In chapter 3, we use the time and space discretizations of the direct problem introduced in [5]. For time discretization we use the Euler-Implicit scheme and for space discretization, we use the Finite Element method. On that basis, we generate the matrices of the discrete system and study the uniqueness of its solution. Then, we present and analyze the results of our numerical simulations implemented using `MATLAB`.

In chapter 4, we introduce the objective function and the `MATLAB` minimization function `fmincon` that will be used to solve the inverse problem. Then, we present and analyze the results of our `MATLAB` numerical simulations.

Finally, in chapter 5, we present our conclusion and possible future work.

Chapter 2

Theoretical Study of the Direct Problem

In this chapter, we summarize the theoretical results recently obtained by S. Moufawad, N. Nassif, and F. Triki in their recent work [5].

2.1 Definitions and Theorems

In this section, we will review some definitions and state useful theorems [6].

Definition 2.1.1 (Inner Product)**.** An inner product on a vector space V is a map $\langle .,. \rangle : V \times V \longrightarrow \mathbb{R}$ satisfying:

1. symmetry: $\langle u, v \rangle = \langle v, u \rangle \; \forall u, v \in V$
2. linearity:

$$\begin{aligned} \langle u_1 + u_2, v \rangle &= \langle u_1, v \rangle + \langle u_2, v \rangle \\ \langle \lambda u, v \rangle &= \lambda \langle u, v \rangle \end{aligned}$$

   $\forall u_1, u_2, u, v \in V$ and $\lambda \in \mathbb{R}$

3. positive definiteness: $\langle v, v \rangle \geq 0$ with equality iff $v = 0 \; \forall v \in V$.

And the norm induced by the inner product is: $\|v\| = \sqrt{\langle v, v \rangle} \; \forall v \in V$

Definition 2.1.2 (Hilbert Space)**.** A Hilbert space is a complete inner product space with respect to the norm induced by the inner product.

Definition 2.1.3 (Dual Space)**.** We denote by V^* the dual space of V, that is, the space of all bounded linear functionals on V. The norm on V^* is defined by:

$$\|f\|_{V^*} = \sup_{\substack{x \in V \\ \|x\| \leq 1}} |f(x)| = \sup_{x \in V} \frac{|f(x)|}{\|x\|}$$

Theorem 2.1.1 (Lions Theorem)**.** Let V and H be two Hilbert spaces satisfying:

$$V \subset H \subset V^* \text{ with } V^* \text{ is the dual of } V$$

with the injection from V to H is dense and continuous.
Assuming a bilinear form $A(.,.) : V \times V \to \mathbb{R}$ that satisfies:

$$\begin{cases} |A(v,w)| \leq M\|v\|_V\|w\|_V \\ |A(v,v)| \geq c_1\|v\|_V^2 - c_2\|v\|_H^2 \end{cases}$$

with M, c_1 and c_2 positive constants.
Then, for $u_0 \in H$ and $f \in L^2((0,T);V^*)$, the initial value problem

$$\begin{cases} \langle u_t, v\rangle + A(u(t),v) = \langle f(t), v\rangle \\ u(0) = u_0 \end{cases}$$

admits a unique solution u, satisfying:

$$u \in L^2((0,T);V) \cap C([0,T];H),\ \frac{du}{dt} \in L^2((0,T);V^*).$$

Theorem 2.1.2 (Riesz-Frechet Representation)**.** Given any $F \in V^*$, there exists a unique $f \in V$ such that

$$F(\varphi) = \langle f, \varphi\rangle_V \ \forall \varphi \in V.$$

Moreover,

$$|f| = \|F\|_{V^*}$$

2.2 Semi-Variational Formulation

The goal here is to put (1.1) in the semi-variational form:

$$\rho_\alpha^o \in U_{ad}\text{: } \langle(\rho_\alpha^o)_t, \phi\rangle + A(\rho_\alpha^o, \phi) = F(\phi)\ \forall \phi \in \mathcal{T} \text{ and } t > 0$$

i.e want to find $\mathcal{T}$, U_{ad}, $A(.,.)$ and $F(.)$.
Let $\phi \in \mathcal{T}$ with $\mathcal{T}$=$\{\phi \in H^1(0,z_f) \mid \phi(0) = 0\}$ and $\rho = \rho_\alpha^o$, $\mathcal{F} = v + w_{air}$, $\mathcal{G} = \tau + \lambda$ and $\mathcal{M}_\alpha = \frac{M_\alpha g}{RT_m}$, where $\mathcal{F}$, $\mathcal{G}$ and $\mathcal{M}_\alpha$ are positive constants.
So (1.1) is now given by:

$$(\rho f)_t + (\rho f\mathcal{F})_z + \rho\mathcal{G} = (D_\alpha(\rho_z - \rho\mathcal{M}_\alpha))_z \tag{2.1}$$

Multiplying (2.1) by ϕ and integrating with respect to z, we get:

$$\begin{aligned} \int f\rho_t\phi + \int f(\rho\mathcal{F})_z\phi + \int \rho\mathcal{G}\phi &= \int [D_\alpha(\rho_z - \rho\mathcal{M}_\alpha)]_z\phi \\ f\int \rho_t\phi + f\mathcal{F}\int \rho_z\phi + \mathcal{G}\int \rho\phi &= \int [D_\alpha(\rho_z - \rho\mathcal{M}_\alpha)]_z\phi \qquad (2.2) \end{aligned}$$

with $\displaystyle\int \rho_z\phi = \rho\phi\Big|_0^{z_F} - \int \rho\phi_z = \rho(z_F,t)\phi(z_F) - \int \rho\phi_z$,

and $\displaystyle\int [D_\alpha(\rho_z - \rho\mathcal{M}_\alpha)]_z\phi = D_\alpha(\rho_z - \rho\mathcal{M}_\alpha)\phi\Big|_0^{z_F} - \int D_\alpha(\rho_z - \rho\mathcal{M}_\alpha)\phi_z$

but using (1.1):

$$\begin{aligned} D_\alpha(\rho_z - \rho\mathcal{M}_\alpha)\phi\Big|_0^{z_F} &= D_\alpha[\rho_z(z_F,t) - \mathcal{M}_\alpha\rho(z_F,t)]\phi(z_F) - D_\alpha[\rho_z(0,t) - \mathcal{M}_\alpha\rho(0,t)]\phi(0) \\ &= D_\alpha(0)\phi(z_F) - 0 = 0 \end{aligned}$$

so, (2.2) becomes:

$$f\langle\rho_t,\phi\rangle + f\mathcal{F}\rho(z_F)\phi(z_F,t) - f\mathcal{F}\langle\rho,\phi_z\rangle + \mathcal{G}\langle\rho,\phi\rangle = -\langle D_\alpha(\rho_z - \rho\mathcal{M}_\alpha),\phi_z\rangle$$

Divide by f:

$$\langle\rho_t,\phi\rangle + \mathcal{F}\rho(z_F,t)\phi(z_F) - \mathcal{F}\langle\rho,\phi_z\rangle + \frac{\mathcal{G}}{f}\langle\rho,\phi\rangle = -\frac{1}{f}\langle D_\alpha\rho_z,\phi_z\rangle + \frac{\mathcal{M}_\alpha}{f}\langle D_\alpha\rho,\phi_z\rangle \tag{2.3}$$

Define the bilinear form:

$$A(\rho,\phi) = \frac{\mathcal{G}}{f}\langle\rho,\phi\rangle + \frac{1}{f}\langle D_\alpha\rho_z,\phi_z\rangle + \mathcal{F}\rho(z_F,t)\phi(z_F) - \mathcal{F}\langle\rho,\phi_z\rangle - \frac{\mathcal{M}_\alpha}{f}\langle D_\alpha\rho,\phi_z\rangle \tag{2.4}$$

Then, (2.3) becomes:

$$\langle\rho_t,\phi\rangle + A(\rho,\phi) = 0$$

Now, the problem is to find
$\rho : [0,T]\times[0,z_F] \to \mathbb{R}$ such that $\forall t > 0$ and $\rho(.,t) \in \mathcal{T} + \{\rho_\alpha^{atm}\} = U_{ad}$

$$\begin{cases} \langle\rho_t,\phi\rangle + A(\rho,\phi) = 0 \\ \rho(z,0) = \bar{\rho}(z) \end{cases} \tag{2.5}$$

where $\bar{\rho}(z)$ is a smooth function and $\rho(z,-\infty) = 0$.

2.3 Existence Theorem

To study the existence and uniqueness of the solution ρ_α^o, we use lions theorem 2.1.1, then apply it to our problem (2.5).
With a start, let's make a change of variable:

$$\tilde{\rho}(.,t) = \rho(.,t) - \rho_\alpha^{atm}(t)$$

Then (2.5) becomes:

$$\begin{cases} \langle(\tilde{\rho} + \rho_\alpha^{atm}(t))_t,\phi\rangle + A((\tilde{\rho} + \rho_\alpha^{atm}(t)),\phi) = 0 \\ \tilde{\rho}(0) = \bar{\rho}(z) - \rho_\alpha^{atm}(0) \end{cases}$$

since A(.,.) and $\langle ., . \rangle$ are bilinear we get:

$$\begin{cases} \langle \tilde{\rho}_t, \phi \rangle + A(\tilde{\rho}, \phi) = -\langle (\rho_\alpha^{atm}(t))_t, \phi \rangle - A(\rho_\alpha^{atm}(t), \phi) \\ \tilde{\rho}(0) = \bar{\rho}(z) - \rho_\alpha^{atm}(0) \end{cases}$$

with: (using (2.4))

- $A(\tilde{\rho}, \phi) = \frac{\mathcal{G}}{f}\langle \tilde{\rho}, \phi \rangle + \frac{1}{f}\langle D_\alpha \tilde{\rho}_z, \phi_z \rangle + \mathcal{F}\tilde{\rho}(z_F, t)\phi(z_F) - \mathcal{F}\langle \tilde{\rho}, \phi_z \rangle - \frac{\mathcal{M}_\alpha}{f}\langle D_\alpha \tilde{\rho}, \phi_z \rangle$
- $A(\rho_\alpha^{atm}(t), \phi) = \rho_\alpha^{atm}(t)\left(\frac{\mathcal{G}}{f}\langle 1, \phi \rangle + \mathcal{F}\phi(z_F) - \mathcal{F}\langle 1, \phi_z \rangle - \frac{\mathcal{M}_\alpha}{f}\langle D_\alpha, \phi_z \rangle\right)$

Then, to be in line with the theorem 2.1.1, we let:

- $u = \tilde{\rho}$
- $F(t, \phi) = -\langle (\rho_\alpha^{atm}(t))_t, \phi \rangle - A(\rho_\alpha^{atm}(t), \phi)$
- $u_0 = \bar{\rho}(z) - \rho_\alpha^{atm}(0)$

So, we can write the problem (2.5) as follows:

$$\begin{cases} \langle u_t, \phi \rangle + A(u(t), \phi) = F(t, \phi) \\ u(0) = u_0 \end{cases}$$

Now, assuming $D_\alpha \in C[0, z_F]$, we define the Hilbert space

$$H_\alpha^1 = \{v \in H^1 \mid \|v\|_{H_\alpha^1} < \infty\}$$

with the following inner product and norm:

$$\begin{aligned} \langle v, w \rangle_{H_\alpha^1} &= \langle D_\alpha v_z, w_z \rangle_2 + \langle v, w \rangle_2 \\ \|v\|_{H_\alpha^1}^2 &= \left\| D_\alpha^{1/2} v_z \right\|_2^2 + \|v\|_2^2 \end{aligned} \tag{2.6}$$

We can see that the injection of H_α^1 to H^1 is continuous using (2.6):

$$\|v\|_{H_\alpha^1}^2 \leq q_{\alpha,\infty}\|v\|_{H^1}^2 \text{ with } q_{\alpha,\infty} = \max\{1, \|D_\alpha\|_\infty\}$$

Lemma 2.3.1. Assuming $\dfrac{1}{D_\alpha^{1/2}} \in L^2(0, z_F)$ with:

$$q_\alpha = \left\| \frac{1}{D_\alpha^{1/2}} \right\|_2 = \left(\int_0^{z_F} \frac{1}{D_\alpha(z)} dz \right)^{1/2} < \infty$$

then, $H_{\alpha,d}^1 = \{v \in H_\alpha^1 \mid v(0) = 0\}$ is a closed subspace of H_α^1, therefore it's a Hilbert space.

Proof. let $\{v_n\} \in H^1_{\alpha,d}$ be a converging sequence with v its limit point, and let $\{v'_n\} \in L^2(0, z_F)$ be a uniformly converging sequence.
We want to show that $v \in H^1_{\alpha,d}$, i.e, $v \in L^2(0, z_F)$, $v' \in L^2(0, z_F)$ and $v(0) = 0$.
Since, $\{v_n\} \in H^1_{\alpha,d}$, then $v_n \in L^2(0, z_F)$, $v'_n \in L^2(0, z_F)$ and $v_n(0) = 0 \; \forall n$.
Moreover, $\lim\limits_{n\to\infty} v_n = v$ and $v' = \left(\lim\limits_{n\to\infty} v_n\right)' = \lim\limits_{n\to\infty} v'_n$. Thus, $v \in L^2(0, z_F)$ and $v' \in L^2(0, z_F)$.
So, it remains to show that $v(0) = 0$ using $\lim\limits_{n\to\infty} \|v - v_n\|_{H^1_\alpha} = 0$.

$$\begin{aligned} v(z) - v(0) &= \int_0^z v'(s)ds \\ v_n(z) - v_n(0) &= \int_0^z v'_n(s)ds \end{aligned}$$

By these two equations, we get:

$$\begin{aligned} -v(0) &= v_n(z) - v(z) + \int_0^z (v'(s) - v'_n(s))ds \\ |v(0)| &\leq |v_n(z) - v(z)| + \int_0^z |v'(s) - v'_n(s)|ds \end{aligned}$$

But $\int_0^z |v'(s) - v'_n(s)|ds = \int_0^z \frac{D_\alpha^{1/2}}{D_\alpha^{1/2}} |v'(s) - v'_n(s)|ds = \left\langle D_\alpha^{1/2}|v'(s) - v'_n(s)|, \frac{1}{D_\alpha^{1/2}} \right\rangle_2$

$$\begin{aligned} &\leq q_\alpha \left\| D_\alpha^{1/2} |v'(s) - v'_n(s)| \right\| \leq q_\alpha \|v - v_n\|_{H^1_\alpha} \\ \Rightarrow \quad & |v(0)| \leq |v_n(z) - v(z)| + q_\alpha \|v - v_n\|_{H^1_\alpha} \qquad (2.7) \end{aligned}$$

Integrating (2.7) with respect to z from 0 to z_F:

$$\begin{aligned} z_F|v(0)| &\leq \int_0^{z_F} |v_n(z) - v(z)| + z_F q_\alpha \|v - v_n\|_{H^1_\alpha} \\ z_F|v(0)| &\leq z_F \|v_n - v\|_2 + z_F q_\alpha \|v - v_n\|_{H^1_\alpha} \\ |v(0)| &\leq \|v_n - v\|_2 + q_\alpha \|v - v_n\|_{H^1_\alpha} \\ &\leq \|v_n - v\|_{H^1_\alpha} + q_\alpha \|v - v_n\|_{H^1_\alpha} \\ &= (1 + q_\alpha)\|v - v_n\|_{H^1_\alpha}. \qquad (2.8) \end{aligned}$$

Taking the limit:

$$|v(0)| = \lim_{n\to\infty} |v(0)| \leq (1 + q_\alpha) \lim_{n\to\infty} \|v_n - v\|_{H^1_\alpha} = 0$$

Thus, $v(0) = 0$ □

Lemma 2.3.2. Under the assumption of lemma 2.3.1 one has

$$H^1_\alpha \subset C[0, z_F] \text{ with } \|v\|_\infty \leq (1 + 2q_\alpha)\|v\|_{H^1_\alpha},\ \forall v \in H^1_\alpha$$

Proof. Using $v(z) - v(0) = \int_0^z v'(s)ds$ for $z \in [0, z_F]$, then

$$|v(z)| \leq |v(0)| + \int_0^z |v'(s)|ds$$

We can proceed like in (2.7) and (2.8):

$$\int_0^z |v'(s)|ds \leq q_\alpha \left\|D^{1/2}_\alpha v'\right\|_2 \leq q_\alpha \|v\|_{H^1_\alpha}$$

$$|v(0)| \leq |v(z)| + \int_0^z |v'(s)|ds \leq |v(z)| + q_\alpha\|v\|_{H^1_\alpha}$$

$$\int_0^{z_F} |v(0)|dz \leq \int_0^{z_F} |v(z)|dz + \int_0^{z_F} q_\alpha\|v\|_{H^1_\alpha} dz$$

$$z_F|v(0)| \leq z_F\|v\|_2 + z_F q_\alpha\|v\|_{H^1_\alpha}$$

$$\Rightarrow |v(0)| \leq \|v\|_2 + q_\alpha\|v\|_{H^1_\alpha} \leq \|v\|_{H^1_\alpha} + q_\alpha\|v\|_{H^1_\alpha} \leq (1 + q_\alpha)\|v\|_{H^1_\alpha}$$

Then, $|v(z)| \leq |v(0)| + \int_0^z |v'(s)|ds \leq (1+q_\alpha)\|v\|_{H^1_\alpha} + q_\alpha\|v\|_{H^1_\alpha} \leq (1+2q_\alpha)\|v\|_{H^1_\alpha}$

Thus, $\|v\|_\infty \leq (1 + 2q_\alpha)\|v\|_{H^1_\alpha}$. □

In order to apply Lions theorem, we let:

$$H = L^2(0, z_F) \text{ and } V = H^1_{\alpha,d}(0, z_F)$$

We have then,

$$V \subset H \subset V^*$$

with continuous injection from V to H. We need also:

1. Bi-continuity of $A(.,.)$:

$$\forall v, \phi \in H^1_\alpha : |A(v, \phi)| \leq C\|v\|_{H^1_\alpha}\|\phi\|_{H^1_\alpha}$$

2. Weak coercivity of $A(.,.)$ on $H^1_{\alpha,d}$:

$$\forall v \in H^1_{\alpha,d} : A(v, v) \geq C_0\|v\|^2_{H^1_\alpha} - C_1\|v\|^2_2$$

3. Existence of $f(t) \in V$, such that $F(t, \phi) = \langle f(t), \phi\rangle_{H^1_\alpha}$, $\forall t, \forall \phi \in H^1_\alpha$

Where C, C_0 and C_1 are positive constants independent of v and w.
Let:

$$G_f = \frac{\mathcal{G}}{f},\ f_1 = \frac{1}{f} \text{ and } M_{\alpha,f} = \frac{\mathcal{M}_\alpha}{f}$$

Then, $A(v,\phi) = G_f\langle v,\phi\rangle + f_1\langle D_\alpha v_z, \phi_z\rangle + \mathcal{F}\phi(z_F)v(z_F) - \mathcal{F}\langle v,\phi_z\rangle - M_{\alpha,f}\langle vD_\alpha, \phi_z\rangle$

1. Bi-continuity of $A(.,.)$:
   we have:

   (a) $\langle v,\phi\rangle \leq \|v\|_2\|\phi\|_2 \leq \|v\|_{H^1_\alpha}\|\phi\|_{H^1_\alpha}$

   (b) $\langle D_\alpha v_z, \phi_z\rangle \leq \left\|D_\alpha^{1/2} v_z\right\|_2 \left\|D_\alpha^{1/2}\phi_z\right\|_2 \leq \|v\|_{H^1_\alpha}\|\phi\|_{H^1_\alpha}$

   (c) $\phi(z_F)v(z_F) \leq \|\phi\|_\infty\|v\|_\infty \leq (1+2q_\alpha)^2\|v\|_{H^1_\alpha}\|\phi\|_{H^1_\alpha}$ (by lemma 2.3.2)

   (d) $|\langle v,\phi_z\rangle| \leq \|v\|_\infty \left|\left\langle \frac{1}{D_\alpha^{1/2}}, D_\alpha^{1/2}\phi_z\right\rangle_2\right| \leq (1+2q_\alpha)\|v\|_{H^1_\alpha} q_\alpha\|\phi\|_{H^1_\alpha}$
   (by lemma 2.3.2)

   (e) $|\langle vD_\alpha, \phi_z\rangle| \leq \left\|D_\alpha^{1/2}\right\|_\infty \|v\|_2 \left\|D_\alpha^{1/2}\phi_z\right\|_2 \leq \left\|D_\alpha^{1/2}\right\|_\infty \|v\|_{H^1_\alpha}\|\phi\|_{H^1_\alpha}$

   Then,

   $$|A(v,\phi)| \leq C\|v\|_{H^1_\alpha}.\|\phi\|_{H^1_\alpha}$$

   with, $C = G_f + f_1 + \mathcal{F}(1+2q_\alpha)^2 + \mathcal{F}(1+2q_\alpha)q_\alpha + M_{\alpha,f}\left\|D_\alpha^{1/2}\right\|_\infty$
   Hence, we have continuity of $A(.,.)$.

2. Coercivity of $A(.,.)$ on $H^1_{\alpha,d}$: let $v \in H^1_{\alpha,d}$

   $$A(v,v) = G_f\langle v,v\rangle + f_1\langle D_\alpha v_z, v_z\rangle + \mathcal{F}v(z_F)^2 - \mathcal{F}\langle v,v_z\rangle - M_{\alpha,f}\langle D_\alpha v, v_z\rangle$$

   We have then,

   (a) $G_f\langle v,v\rangle + f_1\langle D_\alpha v_z, v_z\rangle = G_f\|v\|_2^2 + f_1\left\|D_\alpha^{1/2}v_z\right\|_2^2 \geq \min\{G_f, f_1\}\|v\|^2_{H^1_\alpha}$

   (b) $\mathcal{F}v(z_F)^2 \geq 0$ since $\mathcal{F} \geq 0$

   (c) By Cauchy-Schwartz inequality $|\langle v_z, v\rangle| \leq \|v_z\|_2\|v\|_2$, we have:

   $$-\langle v,v_z\rangle \geq -\|v\|_2\|v_z\|_2 \geq \frac{-1}{\|D_\alpha\|_\infty^{1/2}}\left\|D_\alpha^{1/2}v_z\right\|_2\|v\|_2$$

   since $\|v_z\|_2 \leq \dfrac{\left\|D_\alpha^{1/2}v_z\right\|_2}{\|D_\alpha\|_\infty^{1/2}}$

   (d) Also, by Cauchy-Schwartz inequality:

   $$|\langle D_\alpha v, v_z\rangle| \leq \left\|D_\alpha^{1/2}v\right\|_2\left\|D_\alpha^{1/2}v_z\right\|_2 \leq \|D_\alpha\|_\infty^{1/2}\left\|D_\alpha^{1/2}v_z\right\|_2\|v\|_2$$

   $$\Rightarrow -\langle D_\alpha v, v_z\rangle \geq -\|D_\alpha\|_\infty^{1/2}\left\|D_\alpha^{1/2}v_z\right\|_2\|v\|_2$$

so for $\Gamma = \dfrac{\mathcal{F}}{\|D_\alpha\|_\infty^{1/2}} + M_{\alpha,f}\|D_\alpha\|_\infty^{1/2} > 0$:

$$A(v,v) \geq \min\{G_f, f_1\}\|v\|^2_{H^1_\alpha} - \Gamma \left\|D_\alpha^{1/2} v_z\right\|_2 \|v\|_2$$

Using the geometric inequality: $ab \leq \frac{\epsilon}{2}a^2 + \frac{1}{2\epsilon}b^2, \forall \epsilon > 0$, then

$$\begin{aligned}\left\|D_\alpha^{1/2} v_z\right\|_2 \|v\|_2 &\leq \frac{\epsilon}{2}\left\|D_\alpha^{1/2} v_z\right\|_2^2 + \frac{1}{2\epsilon}\|v\|_2^2 \\ &\leq \frac{\epsilon}{2}\|v\|^2_{H^1_\alpha} + \frac{1}{2\epsilon}\|v\|_2^2\end{aligned}$$

It implies that:

$$A(v,v) \geq \left(\min\{G_f, f_1\} - \Gamma\frac{\epsilon}{2}\right)\|v\|^2_{H^1_\alpha} - \frac{\Gamma}{2\epsilon}\|v\|_2^2$$

Choose $\epsilon > 0$ to be such that:

$$C_{0,\epsilon} = \min\{G_f, f_1\} - \Gamma\frac{\epsilon}{2} = \min\{G_f, f_1\} - \frac{\epsilon}{2}\left(\frac{\mathcal{F}}{\|D_\alpha\|_\infty^{1/2}} + M_{\alpha,f}\|D_\alpha\|_\infty^{1/2}\right) > 0$$

and, $C_{1,\epsilon} = \dfrac{\Gamma}{2\epsilon} = \dfrac{\mathcal{F}}{2\epsilon\,\|D_\alpha\|_\infty^{1/2}} + \dfrac{M_{\alpha,f}}{2\epsilon}\,\|D_\alpha\|_\infty^{1/2} > 0$

Hence, we have coercivity.

3. Existence of a function $f(t) \in L^2((0,T);V)$, such that

$$F(t,\phi) = -\left\langle (\rho_\alpha^{atm}(t))_t, \phi\right\rangle - A(\rho_\alpha^{atm}(t),\phi) = \langle f(t),\phi\rangle_{H^1_\alpha} \forall \phi \in H^1_\alpha.$$

Using bi-continuity of $A(.,)$:

$$|A(\rho_\alpha^{atm}(t),\phi)| \leq C\|\rho_\alpha^{atm}(t)\|_{H^1_\alpha}\|\phi\|_{H^1_\alpha} = C z_F^{1/2}|\rho_\alpha^{atm}(t)|\|\phi\|_{H^1_\alpha}$$

and using Cauchy-Schwartz inequality:

$$|\left\langle (\rho_\alpha^{atm}(t))_t, \phi\right\rangle \leq \|(\rho_\alpha^{atm}(t))_t\|_2\|\phi\|_2 \leq z_F^{1/2}|(\rho_\alpha^{atm}(t))_t|\|\phi\|_{H^1_\alpha}$$

Then,

$$|F(t,\phi)| \leq z_F^{1/2}\left(|(\rho_\alpha^{atm}(t))_t| + C|\rho_\alpha^{atm}(t)|\right)\|\phi\|_{H^1_\alpha} \leq \hat{C}\|\phi\|_{H^1_\alpha} \ \ \forall t,\ \forall \phi \in H^1_\alpha \tag{2.9}$$

Where $\hat{C} = z_F^{1/2}\max\{1,C\}\|\rho_\alpha^{atm}\|_{1,\infty} > 0$ and $\|\rho_\alpha^{atm}\|_{1,\infty} = \max_t\left(|(\rho_\alpha^{atm})_t| + |\rho_\alpha^{atm}|\right)$

Lemma 2.3.3. $F(t,\phi)$ is linear and continuous on H^1_α i.e

$$F(t,.) \in (H^1_\alpha)^* \subset V^* \ \forall t$$

Proof. $F(t,\phi)$ is linear in ϕ by the linearity of the L^2 inner product and the bilinearity of $A(.,.)$.
As for the continuity of $F(t,\phi)$ in H^1_α, let $\phi_n \in H^1_\alpha$ be a sequence converging to ϕ, i.e $\lim\limits_{n\to\infty} \phi_n = \phi$
Then, by (2.9):

$$|F(t,\phi_n) - F(t,\phi)| = |F(t,\phi_n - \phi)| \leq \hat{C}\|\phi_n - \phi\|_{H^1_\alpha}$$

Take now the limit as $n \to \infty$:

$$\lim_{n\to\infty} |F(t,\phi_n) - F(t,\phi)| = 0. \text{ Hence, } \lim_{n\to\infty} F(t,\phi_n) = F(t,\phi)$$

□

Now, by the Riesz-Frechet representation and by lemma 2.3.3, there exists $f(t) \in V$ such that $\forall t,\ \forall \phi \in H^1_\alpha$

$$F(t,\phi) = \langle f(t), \phi \rangle_{H^1_\alpha}$$

and

$$\|f(t)\|_{H^1_\alpha} = \|F(t,\phi)\|_{V^*} = \sup_{\phi\in V} \frac{|F(t,\phi)|}{\|\phi\|_{H^1_\alpha}} \leq \hat{C}$$

and since

$$\int_0^T \|f(t)\|^2_{H^1_\alpha} dt \leq T\hat{C}^2,$$

hence, $f(t) \in L^2((0,T);V)$.

We conclude now that the problem (1.1) admits a unique solution ρ^o_α.

Chapter 3

Numerical Implementation of the Direct Problem

In this chapter, we discretize the direct problem in space and time as discussed in [5]. Then, we generate the matrices obtained and study the existence and uniqueness of the discrete system provided that D_α is a decreasing function. Furthermore, we develop a fast and efficient solver to compute the concentration ρ_α^o using `MATLAB`.

3.1 Galerkin Formulation

To reach the Galerkin formulation of the problem, we first use the Finite Difference Euler-Implicit scheme to discretize the problem in time, then we follow it by a Finite Element space discretization.

3.1.1 Euler-Implicit Time Discretization

Over the interval $[t, t+\Delta t]$ with $0 \leq t \leq T - \Delta t$, we integrate equation (2.5) to get the following:

$$\int_t^{t+\Delta t} \langle \rho_t(z,s), \phi(z) \rangle ds + \int_t^{t+\Delta t} A(\rho(z,s), \phi(z)) ds = 0 \tag{3.1}$$

with

$$\int_t^{t+\Delta t} \langle \rho_t(z,s), \phi(z) \rangle \, ds = \left\langle \int_t^{t+\Delta t} \rho_t(z,s) ds, \phi(z) \right\rangle = \langle \rho(z, t+\Delta t) - \rho(z,t), \phi(z) \rangle \, .$$

Now (3.1) and (2.5) $\Rightarrow$

$$\begin{cases} \langle \rho(z, t+\Delta t) - \rho(z,t), \phi(z) \rangle = - \int_t^{t+\Delta t} A(\rho(z,s), \phi(z)) ds \\ \rho(z,0) = \bar{\rho}(z) \end{cases} \tag{3.2}$$

Such formulation is well-suited for semi and full discretization of the original system.
For the full discretization of the Firn equation, $\int_t^{t+\Delta t} A(\rho(z,s),\phi(z))ds$ is first discretized using an implicit right rectangular rule $\left(\int_a^b f(s)ds = (b-a)f(b)\right)$:

$$\int_t^{t+\Delta t} A(\rho(z,s),\phi(z))ds = \Delta t A(\rho(z,t+\Delta t),\phi(z))$$

Then, (3.2) will be

$$\begin{cases} \langle \rho(z,t+\Delta t) - \rho(z,t), \phi(z)\rangle = -\Delta t A\left(\rho(z,t+\Delta t),\phi(z)\right) \\ \rho(z,0) = \bar{\rho}(z) \end{cases} \tag{3.3}$$

3.1.2 Finite Element Space Discretization

Let $\mathcal{N} = \{\, z_i \mid i = 1,\dots,n\}$ be the set of nodes with $0 = z_1 < z_2 < \cdots < z_n = z_F$, and $\mathcal{E} = \{\, E_j = [z_j, z_{j+1}] \mid j = 1,\dots,n-1\}$ be the set of elements based on $\mathcal{N}$. Define the $\mathbb{P}_1$ finite element spaces:

$$X_n = \{v \in C(0,z_F) | v \in \mathbb{P}_1 \text{on } E_j, \forall j = 1,\dots,n-1\} \subset H^1(0,z_F)$$
$$\text{and } \forall\ v_n \in X_n, v_n(z) = \sum_{i=1}^{n} \varphi_i(z) v_n(z_i)$$

with $\{\varphi_i(z)\}_{i=1}^n$ finite element basis for $\mathbb{P}_1$ and:

$$\varphi_1(z) = \begin{cases} \frac{z_2 - z}{z_2 - z_1} & \text{if } z_1 \le z \le z_2, \\ 0 & \text{otherwise} \end{cases}, \qquad \varphi_n(z) = \begin{cases} \frac{z - z_{n-1}}{z_n - z_{n-1}} & \text{if } z_{n-1} \le z \le z_n, \\ 0 & \text{otherwise} \end{cases},$$
$$\varphi_i(z) = \begin{cases} \frac{z - z_{i-1}}{z_i - z_{i-1}} & \text{if } z_{i-1} \le z \le z_i, \\ \frac{z_{i+1} - z}{z_{i+1} - z_i} & \text{if } z_i \le z \le z_{i+1}, \\ 0 & \text{otherwise} \end{cases} \text{ for } i = 2,\dots,n-1$$

The galerkin approximation sequence $\{\rho_n(t)\} \in X_n + \{\rho_\alpha^{atm}\}$ to the unique solution $\rho(z,t)$ that solves (3.3) is defined by:

$$\begin{cases} \langle \rho_n(t+\Delta t) - \rho_n(t), \phi\rangle = -\Delta t A\left(\rho_n(t+\Delta t),\phi\right) \\ \rho_n(0) = \bar{\rho} \end{cases} \tag{3.4}$$

with

$$\rho_n(t) = \rho_\alpha^{atm}(t)\varphi_1 + \sum_{i=2}^{n} \rho(z_i,t)\varphi_i, \qquad (\rho_n)_z(t) = \rho_\alpha^{atm}(t)\varphi_1' + \sum_{i=2}^{n} \rho(z_i,t)\varphi_i',$$

$$\varphi_1'(z) = \begin{cases} \frac{-1}{z_2-z_1} & \text{if } z_1 \le z \le z_2, \\ 0 & \text{otherwise} \end{cases}, \qquad \varphi_n'(z) = \begin{cases} \frac{1}{z_n-z_{n-1}} & \text{if } z_{n-1} \le z \le z_n, \\ 0 & \text{otherwise} \end{cases},$$

$$\varphi_i'(z) = \begin{cases} \frac{1}{z_i-z_{i-1}} & \text{if } z_{i-1} \le z \le z_i, \\ \frac{-1}{z_{i+1}-z_i} & \text{if } z_i \le z \le z_{i+1}, \\ 0 & \text{otherwise} \end{cases} \text{for } i = 2,\dots,n-1$$

Replacing these functions in (3.4) will give:

$$\left\langle \left(\rho_\alpha^{atm}(t+\Delta t) - \rho_\alpha^{atm}(t)\right)\varphi_1, \phi\right\rangle + \sum_{i=2}^{n} \langle (\rho(z_i,t+\Delta t) - \rho(z_i,t))\,\varphi_i, \phi\rangle$$

$$= -\Delta t A\left(\rho_\alpha^{atm}(t+\Delta t)\varphi_1, \phi\right) - \Delta t A\left(\sum_{i=2}^{n} \rho(z_i,t+\Delta t)\varphi_i, \phi\right)$$

$$\Delta t A\left(\sum_{i=2}^{n} \rho(z_i,t+\Delta t)\varphi_i, \phi\right) + \sum_{i=2}^{n} \rho(z_i,t+\Delta t)\langle\varphi_i,\phi\rangle$$

$$= -\Delta t A\left(\rho_\alpha^{atm}(t+\Delta t)\varphi_1, \phi\right) + \sum_{i=2}^{n} \rho(z_i,t)\langle\varphi_i,\phi\rangle$$

$$- \left(\rho_\alpha^{atm}(t+\Delta t) - \rho_\alpha^{atm}(t)\right)\langle\varphi_1,\phi\rangle \tag{3.5}$$

Let $\phi = \varphi_j$ for $j = 2,\dots,n$, and the vector Λ(t)=$[\rho(z_2,t), \rho(z_3,t), \dots, \rho(z_n,t)]^T$ (3.5) follows then:

$$\begin{cases} \left[M + \Delta t\left(\frac{\mathcal{G}}{f}M + \frac{1}{f}S - K - \frac{\mathcal{M}_\alpha}{f}A + B\right)\right]\Lambda(t+\Delta t) = M\Lambda(t) - v_1(t) - \Delta t v_3(t) \\ \Lambda(0) = \bar{\Lambda} \end{cases} \tag{3.6}$$

For $j = 2,\dots,n$, we have the following:

- M is $(n-1)\times(n-1)$ matrix with $M_{i-1,j-1} = \langle\varphi_i,\varphi_j\rangle$ for $i,j = 2,\dots,n$ so $\sum_{i=2}^{n} \rho(z_i,t+\Delta t)\langle\varphi_i,\varphi_j\rangle = (M\Lambda(t+\Delta t))_{j-1,1}$

- $\sum_{i=2}^{n} \rho(z_i,t)\langle\varphi_i,\varphi_j\rangle = (M\Lambda(t))_{j-1,1}$

- $\langle\varphi_1,\varphi_j\rangle = \begin{cases} \langle\varphi_1,\varphi_2\rangle & \text{for } j=2 \\ 0 & \text{else} \end{cases}$

  so $(\rho_\alpha^{atm}(t+\Delta t)-\rho_\alpha^{atm}(t))\langle\varphi_1,\varphi_j\rangle = \begin{cases} (\rho_\alpha^{atm}(t+\Delta t)-\rho_\alpha^{atm}(t))\langle\varphi_1,\varphi_2\rangle & \text{for } j=2 \\ 0 & \text{else} \end{cases}$

  Let $v_1(t) = (\rho_\alpha^{atm}(t+\Delta t)-\rho_\alpha^{atm}(t))\langle\varphi_1,\varphi_2\rangle e_1$ to be a vector with length $n-1$, where $e_1=[1,0,\dots,0]^T$, then it follows that:

  $$(\rho_\alpha^{atm}(t+\Delta t)-\rho_\alpha^{atm}(t))\langle\varphi_1,\varphi_j\rangle = (v_1(t))_{j-1,1}$$

- $$\begin{aligned} A\left(\sum_{i=2}^{n}\rho(z_i,t+\Delta t)\varphi_i,\varphi_j\right) &= \frac{\mathcal{G}}{f}\sum_{i=2}^{n}\rho(z_i,t+\Delta t)\langle\varphi_i,\varphi_j\rangle + \frac{1}{f}\sum_{i=2}^{n}\rho(z_i,t+\Delta t)\langle D_\alpha\varphi_i',\varphi_j'\rangle \\ &- \mathcal{F}\sum_{i=2}^{n}\rho(z_i,t+\Delta t)\langle\varphi_i,\varphi_j'\rangle - \frac{\mathcal{M}_\alpha}{f}\sum_{i=2}^{n}\rho(z_i,t+\Delta t)\langle D_\alpha\varphi_i,\varphi_j'\rangle \\ &+ \mathcal{F}\varphi_j(z_F)\sum_{i=2}^{n}\rho(z_i,t+\Delta t)\varphi_i(z_F) \end{aligned}$$

  but $\varphi_i(z_F) = \begin{cases} 1 & \text{for } i=n, \text{ i.e } z_i=z_F \\ 0 & \text{else} \end{cases}$

  $$\begin{aligned} \Rightarrow \mathcal{F}\varphi_j(z_F)\sum_{i=2}^{n}\rho(z_i,t+\Delta t)\varphi_i(z_F) &= \mathcal{F}\varphi_j(z_F)\rho(z_F,t+\Delta t) \\ &= \begin{cases} \mathcal{F}\rho(z_F,t+\Delta t) & \text{if } j=n \\ 0 & \text{else} \end{cases} \\ &= (B\Lambda(t+\Delta t))_{j-1,1} \end{aligned}$$

  With B is a $(n-1)\times(n-1)$ zero matrix except for $B_{n-1,n-1}=\mathcal{F}$.
  Let S,K,A be the $(n-1)\times(n-1)$ matrices with entries:
  $S_{i-1,j-1}=\langle D_\alpha\varphi_i',\varphi_j'\rangle$, $K_{i-1,j-1}=\mathcal{F}\langle\varphi_i,\varphi_j'\rangle$ and $A_{i-1,j-1}=\langle D_\alpha\varphi_i,\varphi_j'\rangle$, for $i,j=2,\dots,n$.
  Then, we get:

  $$A\left(\sum_{i=2}^{n}\rho(z_i,t+\Delta t)\varphi_i,\varphi_j\right) = \left(\left(\frac{\mathcal{G}}{f}M + \frac{1}{f}S - K - \frac{M_\alpha}{f}A + B\right)\Lambda(t+\Delta t)\right)_{j-1,1}$$

- $$\begin{aligned} A\left(\rho_\alpha^{atm}(t+\Delta t)\varphi_1,\varphi_j\right) &= \rho_\alpha^{atm}(t+\Delta t)A(\varphi_1,\varphi_j) \\ &= \rho_\alpha^{atm}(t+\Delta t)\left[\frac{\mathcal{G}}{f}\langle\varphi_1,\varphi_j\rangle + \frac{1}{f}\langle D_\alpha\varphi_1',\varphi_j'\rangle - \mathcal{F}\langle\varphi_1,\varphi_j'\rangle \right. \\ &\left. -\frac{\mathcal{M}_\alpha}{f}\langle D_\alpha\varphi_1,\varphi_j'\rangle + \mathcal{F}\varphi_1(z_F)\varphi_j(z_F)\right] \end{aligned}$$

With $\varphi_1(z_F) = 0$ and the inner products are equal to 0 unless $j = 2$. Hence:

$$A\left(\rho_\alpha^{atm}(t+\Delta t)\varphi_1, \varphi_j\right) = \begin{cases} \rho_\alpha^{atm}(t+\Delta t)\left[\frac{\mathcal{G}}{f}\langle\varphi_1, \varphi_2\rangle + \frac{1}{f}\langle D_\alpha\varphi_1', \varphi_2'\rangle \right. & \\ \left. - \mathcal{F}\langle\varphi_1, \varphi_2'\rangle - \frac{\mathcal{M}_\alpha}{f}\langle D_\alpha\varphi_1, \varphi_2'\rangle\right] & \text{for } j = 2 \\ 0 & \text{else} \end{cases}$$

Let $v_3(t) := \rho_\alpha^{atm}(t+\Delta t)\left[\dfrac{\mathcal{G}}{f}\langle\varphi_1, \varphi_2\rangle + \dfrac{1}{f}\langle D_\alpha\varphi_1', \varphi_2'\rangle - \mathcal{F}\langle\varphi_1, \varphi_2'\rangle - \dfrac{\mathcal{M}_\alpha}{f}\langle D_\alpha\varphi_1, \varphi_2'\rangle\right] e_1$ to be a vector with length $n-1$, then it follows that:

$$A\left(\rho_\alpha^{atm}(t+\Delta t)\varphi_1, \varphi_j\right) = (v_3(t))_{j-1,1}$$

3.2 Matrices of the System

For generating the matrices and the vectors described above, we derive expressions for the following inner products in section 3.2.1 :
$\langle\ \varphi_i, \varphi_j\ \rangle$, $\langle\ \varphi_i, \varphi_j'\ \rangle$, $\langle\ D_\alpha\varphi_i, \varphi_j'\ \rangle$, $\langle\ D_\alpha\varphi_i', \varphi_j'\ \rangle$, for $i = 1, \ldots, n$ and for $j = 2, \ldots, n$.
Then, we generate the matrices for non-uniform meshing in section 3.2.2 and for uniform meshing in section 3.2.3.

3.2.1 Derivation of Inner Products

In this section, we will explicitly find the expressions for the following inner products: $\langle\ \varphi_i, \varphi_j\ \rangle$, $\langle\ \varphi_i, \varphi_j'\ \rangle$, $\langle\ D_\alpha\varphi_i, \varphi_j'\ \rangle$, $\langle\ D_\alpha\varphi_i', \varphi_j'\ \rangle$, for $i = 1, \ldots, n$ and for $j = 2, \ldots, n$, taking into consideration four specific cases for each inner product: $j = i-1$ with $i \neq 1, 2$; $j = i+1$ with $i \neq n$; $j = i \neq n$ and $j = i = n$, otherwise the inner product will be equal to 0.
Also, the following Note 1 will be used for approximating the inner product $\langle\ D_\alpha\varphi_i, \varphi_j'\ \rangle$.

Note 1. Given two continuous functions $f(x)$ and $g(x)$ with $g(x) \leq 0$ or $g(x) \geq 0$, we have from the Mean Value Theorem that:

$$\int_a^b f(x)g(x)dx = f(c)\int_a^b g(x)dx\ , c \in (a, b)$$

and we can approximate $f(c)$ to be $f(c) \simeq \frac{1}{2}\left[f(a) + f(b)\right]$.

1. $\langle\ \varphi_i, \varphi_j\ \rangle$:

   - for $j = i - 1$ and $i \neq 1, 2$:

$$\begin{aligned}
\langle\ \varphi_i, \varphi_j\ \rangle = \langle\ \varphi_i, \varphi_{i-1}\ \rangle &= \int_{z_{i-1}}^{z_i} \varphi_i \varphi_{i-1} dz = \int_{z_{i-1}}^{z_i} \frac{z - z_{i-1}}{z_i - z_{i-1}} \frac{z_i - z}{z_i - z_{i-1}} dz \\
&= \frac{1}{(z_i - z_{i-1})^2} \int_{z_{i-1}}^{z_i} (z z_i - z^2 - z_i z_{i-1} + z z_{i-1}) dz \\
&= \frac{1}{(z_i - z_{i-1})^2} \left(\frac{z^2}{2} z_i - \frac{z^3}{3} - z z_i z_{i-1} + \frac{z^2}{2} z_{i-1} \right) \Bigg|_{z_{i-1}}^{z_i} \\
&= \frac{1}{(z_i - z_{i-1})^2} \left(\frac{z_i^3}{2} - \frac{z_i^3}{3} - z_i^2 z_{i-1} + \frac{z_i^2}{2} z_{i-1} - \frac{z_{i-1}^2}{2} z_i + \frac{z_{i-1}^3}{3} + z_i z_{i-1}^2 - \frac{z_{i-1}^3}{2} \right) \\
&= \frac{1}{(z_i - z_{i-1})^2} \left(\frac{z_i^3}{6} - \frac{z_i^2 z_{i-1}}{2} + \frac{z_i z_{i-1}^2}{2} - \frac{z_{i-1}^3}{6} \right) = \frac{(z_i - z_{i-1})^3}{6(z_i - z_{i-1})^2} = \frac{z_i - z_{i-1}}{6}
\end{aligned}$$

   - for $j = i + 1$ and $i \neq n$:

$$\begin{aligned}
\langle\ \varphi_i, \varphi_j\ \rangle = \langle\ \varphi_i, \varphi_{i+1}\ \rangle &= \int_{z_i}^{z_{i+1}} \varphi_i \varphi_{i+1} dz = \int_{z_i}^{z_{i+1}} \frac{z_{i+1} - z}{z_{i+1} - z_i} \frac{z - z_i}{z_{i+1} - z_i} dz \\
&= \frac{1}{(z_{i+1} - z_i)^2} \int_{z_i}^{z_{i+1}} \left(z z_{i+1} - z_{i+1} z_i - z^2 + z z_i \right) dz \\
&= \frac{1}{(z_{i+1} - z_i)^2} \left(\frac{z^2}{2} z_{i+1} - z z_{i+1} z_i - \frac{z^3}{3} + \frac{z^2}{2} z_i \right) \Bigg|_{z_i}^{z_{i+1}} \\
&= \frac{1}{(z_{i+1} - z_i)^2} \left(\frac{z_{i+1}^3}{2} - z_{i+1}^2 z_i - \frac{z_{i+1}^3}{3} + \frac{z_{i+1}^2}{2} z_i - \frac{z_i^2}{2} z_{i+1} + z_{i+1}^2 z_i^2 + \frac{z_i^3}{3} - \frac{z_i^3}{2} \right) \\
&= \frac{1}{(z_{i+1} - z_i)^2} \left(\frac{z_{i+1}^3}{6} - \frac{z_{i+1}^2 z_i}{2} + \frac{z_{i+1} z_i^2}{2} - \frac{z_i^3}{6} \right) = \frac{(z_{i+1} - z_i)^3}{6(z_{i+1} - z_i)^2} = \frac{z_{i+1} - z_i}{6}
\end{aligned}$$

   - for $j = i \neq n$:

$$\begin{aligned}
\langle \varphi_i, \varphi_j \rangle = \langle\ \varphi_i, \varphi_i \rangle &= \int_{z_{i-1}}^{z_i} \varphi_i^2 dz + \int_{z_i}^{z_{i+1}} \varphi_i^2 dz \\
&= \int_{z_{i-1}}^{z_i} \left(\frac{z - z_{i-1}}{z_i - z_{i-1}} \right)^2 dz + \int_{z_i}^{z_{i+1}} \left(\frac{z_{i+1} - z}{z_{i+1} - z_i} \right)^2 dz \\
&= \frac{1}{(z_i - z_{i-1})^2} \left(\frac{(z - z_{i-1})^3}{3} \right) \Bigg|_{z_{i-1}}^{z_i} - \frac{1}{(z_{i+1} - z_i)^2} \left(\frac{(z_{i+1} - z)^3}{3} \right) \Bigg| {z_i}^{z_{i+1}} \\
&= \frac{(z_i - z_{i-1})^3}{3(z_i - z_{i-1})^2} + \frac{(z_{i+1} - z_i)^3}{3(z_{i+1} - z_i)^2} = \frac{z_i - z_{i-1} + z_{i+1} - z_i}{3} = \frac{z_{i+1} - z_{i-1}}{3}
\end{aligned}$$

- for $j = i = n$:

$$\langle \varphi_i, \varphi_j \rangle = \langle \varphi_n, \varphi_n \rangle = \int_{z_{n-1}}^{z_n} \varphi_n^2 dz \ = \int_{z_{n-1}}^{z_n} \left(\frac{z - z_{n-1}}{z_n - z_{n-1}} \right)^2 dz$$

$$= \frac{1}{(z_n - z_{n-1})^2} \left[\frac{1}{3} (z - z_{n-1})^3 \right] \Bigg|_{z_{n-1}}^{z_n} = \frac{z_n - z_{n-1}}{3}$$

Hence, we have the following result for $i, j = 2, \ldots, n$:

$$\langle \varphi_1, \varphi_2 \rangle = \frac{z_2 - z_1}{6}, \ \langle \varphi_i, \varphi_j \rangle = \begin{cases} \frac{z_i - z_{i-1}}{6} & j = i - 1, \ i \neq 2 \\ \frac{z_{i+1} - z_i}{6} & j = i + 1, \ i \neq n \\ \frac{z_{i+1} - z_{i-1}}{3} & j = i \neq n \\ \frac{z_n - z_{n-1}}{3} & j = i = n \end{cases} \tag{3.7}$$

2. $\langle \ \varphi_i, \varphi_j' \ \rangle$:

- for $j = i - 1$ and $i \neq 1, 2$:

$$\langle \ \varphi_i, \varphi_j' \ \rangle = \langle \ \varphi_i, \varphi_{i-1}' \ \rangle = \int_{z_{i-1}}^{z_i} \varphi_i \varphi_{i-1}' dz = \int_{z_{i-1}}^{z_i} \frac{z - z_{i-1}}{z_i - z_{i-1}} \frac{-1}{z_i - z_{i-1}} dz$$

$$= \frac{-1}{(z_i - z_{i-1})^2} \left[\frac{1}{2} (z - z_{i-1})^2 \right] \Bigg|_{z_{i-1}}^{z_i} = \frac{-1}{2}$$

- for $j = i + 1$ and $i \neq n$:

$$\langle \ \varphi_i, \varphi_j' \ \rangle = \langle \ \varphi_i, \varphi_{i+1}' \ \rangle = \int_{z_i}^{z_{i+1}} \varphi_i \varphi_{i+1}' dz = \int_{z_i}^{z_{i+1}} \frac{z_{i+1} - z}{z_{i+1} - z_i} \frac{1}{z_{i+1} - z_i} dz$$

$$= \frac{1}{(z_{i+1} - z_i)^2} \left[\frac{-1}{2} (z_{i+1} - z)^2 \right] \Bigg|_{z_i}^{z_{i+1}} = \frac{1}{2}$$

- for $j = i \neq n$:

$$\langle \ \varphi_i, \varphi_j' \ \rangle = \langle \ \varphi_i, \varphi_i' \ \rangle = \int_{z_{i-1}}^{z_i} \varphi_i \varphi_i' dz + \int_{z_i}^{z_{i+1}} \varphi_i \varphi_i' dz$$

$$= \int_{z_{i-1}}^{z_i} \frac{z - z_{i-1}}{z_i - z_{i-1}} \frac{1}{z_i - z_{i-1}} dz + \int_{z_i}^{z_{i+1}} \frac{z_{i+1} - z}{z_{i+1} - z_i} \frac{-1}{z_{i+1} - z_i} dz$$

$$= \frac{1}{(z_i - z_{i-1})^2} \left[\frac{1}{2} (z - z_{i-1})^2 \right] \Bigg|_{z_{i-1}}^{z_i} - \frac{1}{(z_{i+1} - z_i)^2} \left[\frac{-1}{2} (z_{i+1} - z)^2 \right] \Bigg|_{z_i}^{z_{i+1}}$$

$$= \frac{1}{2} - \frac{1}{2} = 0$$

- for $j = i = n$:

$$\langle\ \varphi_i, \varphi_j'\ \rangle = \langle\ \varphi_n, \varphi_n'\ \rangle = \int_{z_{n-1}}^{z_n} \varphi_n \varphi_n' dz = \int_{z_{n-1}}^{z_n} \frac{z - z_{n-1}}{z_n - z_{n-1}} \frac{1}{z_n - z_{n-1}} dz$$
$$= \frac{1}{(z_n - z_{n-1})^2} \left[\frac{1}{2}(z - z_{n-1})^2\right] \Bigg|_{z_{n-1}}^{z_n} = \frac{1}{2}$$

Hence, we have the following result for $i, j = 2, \ldots, n$:

$$\langle \varphi_1, \varphi_2' \rangle = \frac{1}{2}, \ \langle \varphi_i, \varphi_j' \rangle = \begin{cases} \frac{-1}{2} & j = i - 1, \, i \neq 2 \\ \frac{1}{2} & j = i + 1, \, i \neq n \\ 0 & j = i \neq n \\ \frac{1}{2} & j = i = n \end{cases} \tag{3.8}$$

3. $\langle\ D_\alpha \varphi_i, \varphi_j'\ \rangle$:

$$\langle\ D_\alpha \varphi_i, \varphi_j'\ \rangle = \int D_\alpha \varphi_i \varphi_j' dz$$

We can apply Note 1 by taking the continuous functions $f(z) = D_\alpha(z)$ and $g(z) = (\varphi_i \varphi_j')(z) \leq 0$ or ≥ 0 depending on i and j. After that, we will use the results of (3.8) in the following specific cases:

- for $j = i - 1$ and $i \neq 1, 2$:

$$\begin{aligned} \langle\ D_\alpha \varphi_i, \varphi_j'\ \rangle = \langle\ D_\alpha \varphi_i, \varphi_{i-1}'\ \rangle &= \int_{z_{i-1}}^{z_i} D_\alpha \varphi_i \varphi_{i-1}' dz \\ &\simeq \frac{1}{2}[D_\alpha(z_{i-1}) + D_\alpha(z_i)] \langle\ \varphi_i, \varphi_{i-1}'\ \rangle \\ &= -\frac{1}{4}[D_\alpha(z_{i-1}) + D_\alpha(z_i)] \end{aligned}$$

- for $j = i + 1$ and $i \neq n$:

$$\begin{aligned} \langle\ D_\alpha \varphi_i, \varphi_j'\ \rangle = \langle\ D_\alpha \varphi_i, \varphi_{i+1}'\ \rangle &= \int_{z_i}^{z_{i+1}} D_\alpha \varphi_i \varphi_{i+1}' dz \\ &\simeq \frac{1}{2}[D_\alpha(z_i) + D_\alpha(z_{i+1})] \langle\ \varphi_i, \varphi_{i+1}'\ \rangle \\ &= \frac{1}{4}[D_\alpha(z_i) + D_\alpha(z_{i+1})] \end{aligned}$$

- for $j = i \neq n$:

$$\begin{aligned}
\langle\ D_\alpha \varphi_i, \varphi'_j\ \rangle = \langle\ D_\alpha \varphi_i, \varphi'_i\ \rangle &= \int_{z_{i-1}}^{z_i} D_\alpha \varphi_i \varphi'_i dz + \int_{z_i}^{z_{i+1}} D_\alpha \varphi_i \varphi'_i dz \\
&\simeq \frac{1}{2}[D_\alpha(z_{i-1}) + D_\alpha(z_i)] \frac{1}{(z_i - z_{i-1})^2} \left[\frac{1}{2}(z - z_{i-1})^2\right]\Bigg|_{z_{i-1}}^{z_i} \\
&+ \frac{1}{2}[D_\alpha(z_i) + D_\alpha(z_{i+1})] \frac{-1}{(z_{i+1} - z_i)^2} \left[\frac{-1}{2}(z_{i+1} - z)^2\right]\Bigg|_{z_i}^{z_{i+1}} \\
&= \frac{1}{4}[D_\alpha(z_{i-1}) + D_\alpha(z_i)] - \frac{1}{4}[D_\alpha(z_i) + D_\alpha(z_{i+1})] \\
&= \frac{1}{4}[D_\alpha(z_{i-1}) - D_\alpha(z_{i+1})]
\end{aligned}$$

- for $j = i = n$:

$$\begin{aligned}
\langle\ D_\alpha \varphi_i, \varphi'_j\ \rangle = \langle\ D_\alpha \varphi_n, \varphi'_n\ \rangle &= \int_{z_{n-1}}^{z_n} D_\alpha \varphi_n \varphi'_n dz \\
&\simeq \frac{1}{2}[D_\alpha(z_{n-1}) + D_\alpha(z_n)] \langle\ \varphi_n, \varphi'_n\ \rangle \\
&= \frac{1}{4}[D_\alpha(z_{n-1}) + D_\alpha(z_n)]
\end{aligned}$$

Hence, we have the following result for $i, j = 2, \ldots, n$:

$$\langle\ D_\alpha \varphi_1, \varphi'_2\ \rangle = \frac{D_\alpha(z_1) + D_\alpha(z_2)}{4}, \tag{3.9}$$

$$\langle\ D_\alpha \varphi_i, \varphi'_j\ \rangle = \begin{cases} -\dfrac{D_\alpha(z_{i-1}) + D_\alpha(z_i)}{4} & j = i-1,\ i \neq 2 \\ \dfrac{D_\alpha(z_i) + D_\alpha(z_{i+1})}{4} & j = i+1,\ i \neq n \\ \dfrac{D_\alpha(z_{i-1}) - D_\alpha(z_{i+1})}{4} & j = i \neq n \\ \dfrac{D_\alpha(z_{n-1}) + D_\alpha(z_n)}{4} & j = i = n \end{cases} \tag{3.10}$$

4. $\langle\ D_\alpha \varphi'_i, \varphi'_j\ \rangle$:Since φ'_i and φ'_j are independent of z, we have:

$$\langle\ D_\alpha \varphi'_i, \varphi'_j\ \rangle = \varphi'_i \varphi'_j \int D_\alpha dz$$

which can be solved using the trapezoidal rule $\left(\int_a^b f(x)dx \simeq \frac{1}{2}(b-a)(f(a) + f(b))\right)$.

- for $j = i - 1$ and $i \neq 1, 2$:

$$\begin{aligned}
\langle\ D_\alpha \varphi_i', \varphi_j'\ \rangle = \langle\ D_\alpha \varphi_i', \varphi_{i-1}'\ \rangle = \int_{z_{i-1}}^{z_i} D_\alpha \varphi_i' \varphi_{i-1}' dz \\
&\simeq \frac{1}{2}\left[D_\alpha(z_{i-1}) + D_\alpha(z_i)\right] \frac{-1}{(z_i - z_{i-1})^2}(z_i - z_{i-1}) \\
&= -\frac{D_\alpha(z_{i-1}) + D_\alpha(z_i)}{2(z_i - z_{i-1})}
\end{aligned}$$

- for $j = i + 1$ and $i \neq n$:

$$\begin{aligned}
\langle\ D_\alpha \varphi_i', \varphi_j'\ \rangle = \langle\ D_\alpha \varphi_i', \varphi_{i+1}'\ \rangle = \int_{z_i}^{z_{i+1}} D_\alpha \varphi_i' \varphi_{i+1}' dz \\
&\simeq \frac{1}{2}[D_\alpha(z_i) + D_\alpha(z_{i+1})] \frac{-1}{(z_{i+1} - z_i)^2}(z_{i+1} - z_i) \\
&= -\frac{D_\alpha(z_i) + D_\alpha(z_{i+1})}{2(z_{i+1} - z_i)}
\end{aligned}$$

- for $j = i \neq n$:

$$\begin{aligned}
\langle\ D_\alpha \varphi_i', \varphi_j'\ \rangle = \langle\ D_\alpha \varphi_i', \varphi_i'\ \rangle = \int_{z_{i-1}}^{z_i} D_\alpha \varphi_i'^2 dz + \int_{z_i}^{z_{i+1}} D_\alpha \varphi_i'^2 dz \\
&\simeq \frac{1}{2}[D_\alpha(z_{i-1}) + D_\alpha(z_i)] \frac{z_i - z_{i-1}}{(z_i - z_{i-1})^2} + \frac{1}{2}[D_\alpha(z_i) + D_\alpha(z_{i+1})] \frac{z_{i+1} - z_i}{(z_{i+1} - z_i)^2} \\
&= \frac{D_\alpha(z_{i-1}) + D_\alpha(z_i)}{2(z_i - z_{i-1})} + \frac{D_\alpha(z_i) + D_\alpha(z_{i+1})}{2(z_{i+1} - z_i)}
\end{aligned}$$

- for $j = i = n$:

$$\begin{aligned}
\langle\ D_\alpha \varphi_i', \varphi_j'\ \rangle = \langle\ D_\alpha \varphi_n', \varphi_n'\ \rangle = \int_{z_{n-1}}^{z_n} D_\alpha \varphi_n'^2 dz \\
&\simeq \frac{1}{2}[D_\alpha(z_{n-1}) + D_\alpha(z_n)] \frac{1}{(z_n - z_{n-1})^2}(z_n - z_{n-1}) \\
&= \frac{D_\alpha(z_{n-1}) + D_\alpha(z_n)}{2(z_n - z_{n-1})}
\end{aligned}$$

Hence, we have the following result for $i, j = 2, \dots, n$

$$\langle\ D_\alpha \varphi_1', \varphi_2'\ \rangle = -\frac{D_\alpha(z_1) + D_\alpha(z_2)}{2(z_2 - z_1)}, \tag{3.11}$$

$$\langle\ D_\alpha \varphi_i', \varphi_j'\ \rangle = \begin{cases} -\dfrac{D_\alpha(z_{i-1}) + D_\alpha(z_i)}{2(z_i - z_{i-1})} & j = i-1,\, i \neq 2 \\ -\dfrac{D_\alpha(z_i) + D_\alpha(z_{i+1})}{2(z_{i+1} - z_i)} & j = i+1,\, i \neq n \\ \dfrac{D_\alpha(z_{i-1}) + D_\alpha(z_i)}{2(z_i - z_{i-1})} + \dfrac{D_\alpha(z_i) + D_\alpha(z_{i+1})}{2(z_{i+1} - z_i)} & j = i \neq n \\ \dfrac{D_\alpha(z_{n-1}) + D_\alpha(z_n)}{2(z_n - z_{n-1})} & j = i = n \end{cases} \tag{3.12}$$

3.2.2 Matrices for non-Uniform Meshing

Now, after finding these values, we are able to generate the vectors $v_1(t)$, $v_3(t)$ and the matrices B, K, M, A and S assuming we have a non-uniform meshing.

- Vector $v_1(t)$:
  Using the equation (3.7) and $z_1 = 0$, we get:

$$\begin{aligned} v_1(t) &= \left(\rho_\alpha^{atm}(t + \Delta t) - \rho_\alpha^{atm}(t)\right) \langle \varphi_1, \varphi_2 \rangle e_1 \\ &= \left(\rho_\alpha^{atm}(t + \Delta t) - \rho_\alpha^{atm}(t)\right) \langle \varphi_1, \varphi_2 \rangle \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}_{(n-1)\times 1} \\ &= \begin{pmatrix} (\rho_\alpha^{atm}(t + \Delta t) - \rho_\alpha^{atm}(t))\, \dfrac{z_2}{6} \\ 0 \\ \vdots \\ 0 \end{pmatrix}_{(n-1)\times 1} \end{aligned}$$

- Vector $v_3(t)$:
  Referring to (3.7), (3.8), (3.9), (3.11) and $z_1 = 0$, we get:

$$\begin{aligned} v_3(t) &= \rho_\alpha^{atm}(t + \Delta t) \left(\frac{\mathcal{G}}{f} \langle \varphi_1, \varphi_2 \rangle + \frac{1}{f} \langle\ D_\alpha \varphi_1', \varphi_2' - \mathcal{F} \langle \varphi_1, \varphi_2' \rangle - \frac{\mathcal{M}_\alpha}{f} \langle\ D_\alpha \varphi_1, \varphi_2' \right) e_1 \\ &= \rho_\alpha^{atm}(t + \Delta t) \left(\frac{\mathcal{G}(z_2 - z_1)}{6f} + \frac{-D_\alpha(z_1) - D_\alpha(z_2)}{2f(z_2 - z_1)} - \frac{\mathcal{F}}{2} - \frac{\mathcal{M}_\alpha}{4f} (D_\alpha(z_1) + D_\alpha(z_2)) \right) e_1 \\ &= \rho_\alpha^{atm}(t + \Delta t) \left(\frac{\mathcal{G}}{6f} z_2 - \left(\frac{1}{2f z_2} + \frac{\mathcal{M}_\alpha}{4f} \right) (D_\alpha(0) + D_\alpha(z_2)) - \frac{\mathcal{F}}{2} \right) \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}_{(n-1)\times 1} \end{aligned}$$

$$= \begin{pmatrix} \rho_\alpha^{atm}(t+\Delta t)\left(\frac{\mathcal{G}}{6f}z_2 - (\frac{1}{2fz_2} + \frac{\mathcal{M}_\alpha}{4f})(D_\alpha(0)+D_\alpha(z_2)) - \frac{\mathcal{F}}{2}\right) \\ 0 \\ \vdots \\ 0 \end{pmatrix}_{(n-1)\times 1}$$

- Matrix B:
  B is a $(n-1)\times(n-1)$ zero matrix except for $B_{n-1,n-1} = \mathcal{F}$, then:

$$B = \begin{pmatrix} 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \mathcal{F} \end{pmatrix}_{(n-1)\times(n-1)}$$

- Matrix K:
  K is a $(n-1)\times(n-1)$ matrix with entries: $K_{i-1,j-1} = \mathcal{F}\langle\varphi_i, \varphi_j'\rangle$, for $i,j = 2,\dots,n$. Hence, using the result of (3.8), we have:

$$\begin{aligned} K_{i-1,j-1} &= \mathcal{F}\langle\varphi_i, \varphi_j'\rangle \\ &= \mathcal{F}\begin{cases} -\frac{1}{2} & \text{if } j = i-1,\ i \neq 2, \\ \frac{1}{2} & \text{if } j = i+1,\ i \neq n \text{ or } j = i = n, \\ 0 & \text{else} \end{cases} \end{aligned}$$

  We get then,

$$K = \begin{pmatrix} 0 & \frac{\mathcal{F}}{2} & 0 & \dots & 0 \\ \frac{-\mathcal{F}}{2} & 0 & \frac{\mathcal{F}}{2} & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & \frac{-\mathcal{F}}{2} & 0 & \frac{\mathcal{F}}{2} \\ 0 & \dots & 0 & \frac{-\mathcal{F}}{2} & \frac{\mathcal{F}}{2} \end{pmatrix}_{(n-1)\times(n-1)}$$

- Matrix M:
  M is a $(n-1)\times(n-1)$ matrix with entries: $M_{i-1,j-1} = \langle\varphi_i, \varphi_j\rangle$, for $i,j = 2,\dots,n$. Then, taking the results of (3.7) and defining h_i to be the distance between two consecutive points of the space i.e $h_i = z_i - z_{i-1}$, we have:

$$\begin{aligned} M_{i-1,j-1} &= \langle\varphi_i, \varphi_j\rangle \\ &= \begin{cases} \frac{h_i}{6} & \text{if } j = i-1,\ i \neq 2, \\ \frac{h_{i+1}}{6} & \text{if } j = i+1, i \neq n, \\ \frac{h_i + h_{i+1}}{3} & \text{if } j = i \neq n, \\ \frac{h_n}{3} & \text{if } j = i = n, \\ 0 & \text{else} \end{cases} \end{aligned}$$

Which give us the matrix

$$M = \begin{pmatrix} \frac{h_2+h_3}{6} & \frac{h_3}{6} & 0 & \dots & 0 \\ \frac{h_3}{6} & \frac{h_3+h_4}{3} & \frac{h_4}{6} & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & \frac{h_{n-1}}{6} & \frac{h_{n-1}+h_n}{3} & \frac{h_n}{6} \\ 0 & \dots & 0 & \frac{h_n}{6} & \frac{h_n}{3} \end{pmatrix}_{(n-1)\times(n-1)}$$

- Matrix A:
  A is a $(n-1)\times(n-1)$ matrix with entries: $A_{i-1,j-1} = \langle D_\alpha \varphi_i, \varphi_j' \rangle$, for $i, j = 2, \dots, n$. Hence, using the equation (3.10), we get:

$$\begin{aligned} A_{i-1,j-1} &= \langle D_\alpha \varphi_i, \varphi_j' \rangle \\ &= \begin{cases} -\dfrac{D_\alpha(z_{i-1}) + D_\alpha(z_i)}{4} & \text{if } j = i-1,\ i \neq 2, \\ \dfrac{D_\alpha(z_i) + D_\alpha(z_{i+1})}{4} & \text{if } j = i+1,\ i \neq n, \\ \dfrac{D_\alpha(z_{i-1}) - D_\alpha(z_{i+1})}{4} & \text{if } j = i \neq n, \\ \dfrac{D_\alpha(z_{n-1}) + D_\alpha(z_n)}{4} & \text{if } j = i = n, \\ 0 & \text{else} \end{cases} \end{aligned}$$

  We get then,

$$A = \frac{1}{4}\begin{pmatrix} D_\alpha(z_1) & D_\alpha(z_2) & 0 & \dots & 0 \\ -D_\alpha(z_2) & D_\alpha(z_2) & D_\alpha(z_3) & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & -D_\alpha(z_{n-2}) & D_\alpha(z_{n-2}) & D_\alpha(z_{n-1}) \\ 0 & \dots & 0 & -D_\alpha(z_{n-1}) & D_\alpha(z_{n-1}) \end{pmatrix}_{(n-1)\times(n-1)}$$

$$+ \frac{1}{4}\begin{pmatrix} -D_\alpha(z_3) & D_\alpha(z_3) & 0 & \dots & 0 \\ -D_\alpha(z_3) & -D_\alpha(z_4) & D_\alpha(z_4) & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & -D_\alpha(z_{n-1}) & -D_\alpha(z_n) & D_\alpha(z_n) \\ 0 & \dots & 0 & -D_\alpha(z_n) & D_\alpha(z_n) \end{pmatrix}_{(n-1)\times(n-1)}$$

- Matrix S:
  S is a $(n-1)\times(n-1)$ matrix with entries: $S_{i-1,j-1} = \langle D_\alpha \varphi_i', \varphi_j' \rangle$, for $i, j = 2, \dots, n$. Then, taking the results of (3.12) and $h_i = z_i - z_{i-1}$, we have:

$$S_{i-1,j-1} = \langle D_\alpha \varphi_i', \varphi_j' \rangle$$

$$= \begin{cases} -\dfrac{D_\alpha(z_{i-1}) + D_\alpha(z_i)}{2h_i} & \text{if } j = i-1,\ i \neq 2, \\ -\dfrac{D_\alpha(z_i) + D_\alpha(z_{i+1})}{2h_{i+1}} & \text{if } j = i+1,\ i \neq n, \\ \dfrac{D_\alpha(z_{i-1}) + D_\alpha(z_i)}{2h_i} + \dfrac{D_\alpha(z_i) + D_\alpha(z_{i+1})}{2h_{i+1}} & \text{if } j = i \neq n, \\ \dfrac{D_\alpha(z_{n-1}) + D_\alpha(z_n)}{2h_n} & \text{if } j = i = n, \\ 0 & \text{else} \end{cases}$$

It follows that:

$$S = \begin{pmatrix} \frac{D_\alpha(z_1)}{2h_2} + \frac{D_\alpha(z_2)}{2h_3} & \frac{-D_\alpha(z_2)}{2h_3} & 0 & \cdots & 0 \\ \frac{-D_\alpha(z_2)}{2h_3} & \frac{D_\alpha(z_2)}{2h_3} + \frac{D_\alpha(z_3)}{2h_4} & \frac{-D_\alpha(z_3)}{2h_4} & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & \frac{-D_\alpha(z_{n-2})}{2h_{n-1}} & \frac{D_\alpha(z_{n-2})}{2h_{n-1}} + \frac{D_\alpha(z_{n-1})}{2h_n} & \frac{-D_\alpha(z_{n-1})}{2h_n} \\ 0 & \cdots & 0 & \frac{-D_\alpha(z_{n-1})}{2h_n} & \frac{D_\alpha(z_{n-1})}{2h_n} \end{pmatrix}_{(n-1)\times(n-1)}$$

$$+ \begin{pmatrix} \frac{D_\alpha(z_2)}{2h_2} + \frac{D_\alpha(z_3)}{2h_3} & \frac{-D_\alpha(z_3)}{2h_3} & 0 & \cdots & 0 \\ \frac{-D_\alpha(z_3)}{2h_3} & \frac{D_\alpha(z_3)}{2h_3} + \frac{D_\alpha(z_4)}{2h_4} & \frac{-D_\alpha(z_4)}{2h_4} & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & \frac{-D_\alpha(z_{n-1})}{2h_{n-1}} & \frac{D_\alpha(z_{n-1})}{2h_{n-1}} + \frac{D_\alpha(z_n)}{2h_n} & \frac{-D_\alpha(z_n)}{2h_n} \\ 0 & \cdots & 0 & \frac{-D_\alpha(z_n)}{2h_n} & \frac{D_\alpha(z_n)}{2h_n} \end{pmatrix}_{(n-1)\times(n-1)}$$

3.2.3 Matrices for Uniform Meshing

In this section we generate the vectors $v_1(t)$, $v_3(t)$ and the matrices B, K, M, A and S assuming we have a uniform meshing i.e $h_i = h$ =constant.
So the vectors and matrices in section 3.2.2 independent of h_i remain the same and for the others, the h_i will be replaced with h. Hence we have the following:

$$v_3(t) = \begin{pmatrix} \rho_\alpha^{atm}(t+\Delta t)\left(\dfrac{\mathcal{G}}{6f} z_2 - \left(\dfrac{1}{2fz_2} + \dfrac{\mathcal{M}_\alpha}{4f}\right)(D_\alpha(0) + D_\alpha(z_2)) - \dfrac{\mathcal{F}}{2}\right) \\ 0 \\ \vdots \\ 0 \end{pmatrix}_{(n-1)\times 1}$$

$$v_1(t) = \begin{pmatrix} (\rho_\alpha^{atm}(t+\Delta t) - \rho_\alpha^{atm}(t))\,\dfrac{z_2}{6} \\ 0 \\ \vdots \\ 0 \end{pmatrix}_{(n-1)\times 1}, B = \begin{pmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathcal{F} \end{pmatrix}_{(n-1)\times(n-1)}$$

$$K = \begin{pmatrix} 0 & \frac{\mathcal{F}}{2} & 0 & \dots & 0 \\ \frac{-\mathcal{F}}{2} & 0 & \frac{\mathcal{F}}{2} & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & \frac{-\mathcal{F}}{2} & 0 & \frac{\mathcal{F}}{2} \\ 0 & \dots & 0 & \frac{-\mathcal{F}}{2} & \frac{\mathcal{F}}{2} \end{pmatrix}_{(n-1)\times(n-1)} , M = h \begin{pmatrix} \frac{1}{3} & \frac{1}{6} & 0 & \dots & 0 \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ 0 & \dots & 0 & \frac{1}{6} & \frac{1}{3} \end{pmatrix}_{(n-1)\times(n-1)}$$

$$A = \frac{1}{4} \begin{pmatrix} D_\alpha(z_1) & D_\alpha(z_2) & 0 & \dots & 0 \\ -D_\alpha(z_2) & D_\alpha(z_2) & D_\alpha(z_3) & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & -D_\alpha(z_{n-2}) & D_\alpha(z_{n-2}) & D_\alpha(z_{n-1}) \\ 0 & \dots & 0 & -D_\alpha(z_{n-1}) & D_\alpha(z_{n-1}) \end{pmatrix}_{(n-1)\times(n-1)}$$

$$+ \frac{1}{4} \begin{pmatrix} -D_\alpha(z_3) & D_\alpha(z_3) & 0 & \dots & 0 \\ -D_\alpha(z_3) & -D_\alpha(z_4) & D_\alpha(z_4) & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & -D_\alpha(z_{n-1}) & -D_\alpha(z_n) & D_\alpha(z_n) \\ 0 & \dots & 0 & -D_\alpha(z_n) & D_\alpha(z_n) \end{pmatrix}_{(n-1)\times(n-1)}$$

$$S = \begin{pmatrix} \frac{D_\alpha(z_1)+D_\alpha(z_2)}{2h} & \frac{-D_\alpha(z_2)}{2h} & 0 & \dots & 0 \\ \frac{-D_\alpha(z_2)}{2h} & \frac{D_\alpha(z_2)+D_\alpha(z_3)}{2h} & \frac{-D_\alpha(z_3)}{2h} & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & \frac{-D_\alpha(z_{n-2})}{2h} & \frac{D_\alpha(z_{n-2})+D_\alpha(z_{n-1})}{2h} & \frac{-D_\alpha(z_{n-1})}{2h} \\ 0 & \dots & 0 & \frac{-D_\alpha(z_{n-1})}{2h} & \frac{D_\alpha(z_{n-1})}{2h} \end{pmatrix}_{(n-1)\times(n-1)}$$

$$+ \begin{pmatrix} \frac{D_\alpha(z_2)+D_\alpha(z_3)}{2h} & \frac{-D_\alpha(z_3)}{2h} & 0 & \dots & 0 \\ \frac{-D_\alpha(z_3)}{2h} & \frac{D_\alpha(z_3)+D_\alpha(z_4)}{2h} & \frac{-D_\alpha(z_4)}{2h} & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & \frac{-D_\alpha(z_{n-1})}{2h} & \frac{D_\alpha(z_{n-1})+D_\alpha(z_n)}{2h} & \frac{-D_\alpha(z_n)}{2h} \\ 0 & \dots & 0 & \frac{-D_\alpha(z_n)}{2h} & \frac{D_\alpha(z_n)}{2h} \end{pmatrix}_{(n-1)\times(n-1)}$$

3.3 Existence and Uniqueness of the Solution of the Finite Element Euler-Implicit Discrete System (3.6)

In this section, we show that the matrices of the system and especially for the non-uniform meshing are either positive definite or symmetric positive definite (spd). Then, we use this to prove the existence and uniqueness of the solution of the system (3.6).

Remark. An $m \times m$ matrix, N, is positive definite $\iff Z^T N Z > 0,\ \forall Z \in \mathbb{R}^m - \{0\}$. Furthermore, N is called spd when it's positive definite and symmetric i.e $N^T = N$.

Lemma 3.3.1. The matrix M is spd.

Proof.

$$MZ = \begin{pmatrix} \frac{h_2+h_3}{6} & \frac{h_3}{6} & 0 & \dots & 0 \\ \frac{h_3}{6} & \frac{h_3+h_4}{3} & \frac{h_4}{6} & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & \frac{h_{n-1}}{6} & \frac{h_{n-1}+h_n}{3} & \frac{h_n}{6} \\ 0 & \dots & 0 & \frac{h_n}{6} & \frac{h_n}{3} \end{pmatrix} \begin{pmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_{n-2} \\ Z_{n-1} \end{pmatrix}$$

$$= \begin{pmatrix} \left(\frac{h_2+h_3}{6}\right) Z_1 + \frac{h_3}{6} Z_2 \\ \frac{h_3}{6} Z_1 + \left(\frac{h_3+h_4}{3}\right) Z_2 + \frac{h_4}{6} Z_3 \\ \vdots \\ \frac{h_{n-1}}{6} Z_{n-3} + \left(\frac{h_{n-1}+h_n}{3}\right) Z_{n-2} + \frac{h_n}{6} Z_{n-1} \\ \frac{h_n}{6} Z_{n-2} + \frac{h_n}{3} Z_{n-1} \end{pmatrix}$$

$$\begin{aligned} \text{Then, } Z^T M Z = {} & \left(\frac{h_2+h_3}{6}\right) Z_1^2 + \frac{h_3}{6} Z_1 Z_2 + \frac{h_3}{6} Z_1 Z_2 + \left(\frac{h_3+h_4}{3}\right) Z_2^2 + \frac{h_4}{6} Z_2 Z_3 \\ & + \frac{h_4}{6} Z_2 Z_3 + \left(\frac{h_4+h_5}{3}\right) Z_3^2 + \ldots + \frac{h_{n-1}}{6} Z_{n-3} Z_{n-2} + \left(\frac{h_{n-1}+h_n}{3}\right) Z_{n-2}^2 \\ & + \frac{h_n}{6} Z_{n-2} Z_{n-1} + \frac{h_n}{6} Z_{n-2} Z_{n-1} + \frac{h_n}{3} Z_{n-1}^2 \\ = {} & \frac{h_2}{6} Z_1^2 + \frac{h_3}{6} Z_1^2 + \frac{h_3}{3} Z_1 Z_2 + \frac{h_3}{3} Z_2^2 + \frac{h_4}{3} Z_2^2 + \frac{h_4}{3} Z_2 Z_3 + \frac{h_4}{3} Z_3^2 + \frac{h_5}{3} Z_3^2 \\ & + \ldots + \frac{h_{n-1}}{3} Z_{n-3} Z_{n-2} + \frac{h_{n-1}}{3} Z_{n-2}^2 + \frac{h_n}{3} Z_{n-2}^2 + \frac{h_n}{3} Z_{n-2} Z_{n-1} + \frac{h_n}{3} Z_{n-1}^2 \end{aligned}$$

$$= \frac{h_2}{6}Z_1^2 + \frac{h_3}{3}\left(\frac{Z_1^2}{2} + Z_1Z_2 + \frac{Z_2^2}{2} + \frac{Z_2^2}{2}\right) + \sum_{i=2}^{n-2}\frac{h_{i+2}}{3}\left(Z_i^2 + Z_iZ_{i+1} + Z_{i+1}^2\right)$$

$$= \frac{h_2}{6}Z_1^2 + \frac{h_3}{3}\left((Z_1+Z_2)^2 + \frac{Z_2^2}{2}\right) + \sum_{i=2}^{n-2}\frac{h_{i+2}}{3}\left(\frac{Z_i^2}{2} + (Z_i + Z_{i+1})^2 + \frac{Z_{i+1}^2}{2}\right)$$

Hence, for $Z \neq 0$, $Z^TMZ > 0$ since it's a summation of positive terms. And, it's clear that M is symmetric. Therefore, M is spd. □

Lemma 3.3.2. The matrix K is positive definite and B is spd.

Proof.

$$Z^tKZ = \begin{pmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_{n-2} \\ Z_{n-1} \end{pmatrix}^T \begin{pmatrix} 0 & \frac{\mathcal{F}}{2} & 0 & \dots & 0 \\ \frac{-\mathcal{F}}{2} & 0 & \frac{\mathcal{F}}{2} & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & \frac{-\mathcal{F}}{2} & 0 & \frac{\mathcal{F}}{2} \\ 0 & \dots & 0 & \frac{-\mathcal{F}}{2} & \frac{\mathcal{F}}{2} \end{pmatrix} \begin{pmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_{n-2} \\ Z_{n-1} \end{pmatrix} = \frac{\mathcal{F}}{2} \begin{pmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_{n-2} \\ Z_{n-1} \end{pmatrix}^T \begin{pmatrix} Z_2 \\ -Z_1 + Z_3 \\ \vdots \\ -Z_{n-3} + Z_{n-1} \\ -Z_{n-2} + Z_{n-1} \end{pmatrix}$$

$$= \frac{\mathcal{F}}{2}(Z_1Z_2 - Z_1Z_2 + Z_2Z_3 - Z_2Z_3 + \dots - Z_{n-3}Z_{n-2} + Z_{n-2}Z_{n-1} - Z_{n-2}Z_{n-1} + Z_{n-1}^2)$$

$$= \frac{\mathcal{F}}{2}Z_{n-1}^2 \tag{3.13}$$

$$Z^TBZ = \begin{pmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_{n-2} \\ Z_{n-1} \end{pmatrix}^T \begin{pmatrix} 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \mathcal{F} \end{pmatrix} \begin{pmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_{n-2} \\ Z_{n-1} \end{pmatrix} = \begin{pmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_{n-2} \\ Z_{n-1} \end{pmatrix}^T \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \mathcal{F}Z_{n-1} \end{pmatrix}$$

$$=0 + \dots + 0 + \mathcal{F}Z_{n-1}^2 = \mathcal{F}Z_{n-1}^2 \tag{3.14}$$

Since $\mathcal{F} > 0$, the equations (3.13) and (3.14) are positives for $Z > 0$. Therefore, K and B are positive definite. Also, it's obvious that B is symmetric. Hence B is spd. □

Lemma 3.3.3. The matrix S is spd.

Proof. As shown in section 3.2.2, the matrix S is a summation of two matrices, let's call them S_1 and S_2. Then, $Z^TSZ = Z^TS_1Z + Z^TS_2Z$.
First, let's find Z^TS_1Z.

$$S_1Z = \begin{pmatrix} \frac{D_\alpha(z_1)}{2h_2} + \frac{D_\alpha(z_2)}{2h_3} & \frac{-D_\alpha(z_2)}{2h_3} & 0 & \dots & 0 \\ \frac{-D_\alpha(z_2)}{2h_3} & \frac{D_\alpha(z_2)}{2h_3} + \frac{D_\alpha(z_3)}{2h_4} & \frac{-D_\alpha(z_3)}{2h_4} & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & \frac{-D_\alpha(z_{n-2})}{2h_{n-1}} & \frac{D_\alpha(z_{n-2})}{2h_{n-1}} + \frac{D_\alpha(z_{n-1})}{2h_n} & \frac{-D_\alpha(z_{n-1})}{2h_n} \\ 0 & \dots & 0 & \frac{-D_\alpha(z_{n-1})}{2h_n} & \frac{D_\alpha(z_{n-1})}{2h_n} \end{pmatrix} \begin{pmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_{n-2} \\ Z_{n-1} \end{pmatrix}$$

$$= \begin{pmatrix} \left(\frac{D_\alpha(z_1)}{2h_2} + \frac{D_\alpha(z_2)}{2h_3}\right) Z_1 - \frac{D_\alpha(z_2)}{2h_3} Z_2 \\ -\frac{D_\alpha(z_2)}{2h_3} Z_1 + \left(\frac{D_\alpha(z_2)}{2h_3} + \frac{D_\alpha(z_3)}{2h_4}\right) Z_2 - \frac{D_\alpha(z_3)}{2h_4} Z_3 \\ \vdots \\ -\frac{D_\alpha(z_{n-2})}{2h_{n-1}} Z_{n-3} + \left(\frac{D_\alpha(z_{n-2})}{2h_{n-1}} + \frac{D_\alpha(z_{n-1})}{2h_n}\right) Z_{n-2} - \frac{D_\alpha(z_{n-1})}{2h_n} Z_{n-1} \\ -\frac{D_\alpha(z_{n-1})}{2h_n} Z_{n-2} + \frac{D_\alpha(z_{n-1})}{2h_n} Z_{n-1} \end{pmatrix}$$

$$\begin{aligned}
\text{So, } Z^T S_1 Z &= \left(\frac{D_\alpha(z_1)}{2h_2} + \frac{D_\alpha(z_2)}{2h_3}\right) Z_1^2 - \frac{D_\alpha(z_2)}{2h_3} Z_1 Z_2 - \frac{D_\alpha(z_2)}{2h_3} Z_1 Z_2 \\
&\quad + \left(\frac{D_\alpha(z_2)}{2h_3} + \frac{D_\alpha(z_3)}{2h_4}\right) Z_2^2 - \frac{D_\alpha(z_3)}{2h_4} Z_2 Z_3 - \frac{D_\alpha(z_3)}{2h_4} Z_2 Z_3 \\
&\quad + \left(\frac{D_\alpha(z_3)}{2h_4} + \frac{D_\alpha(z_4)}{2h_5}\right) Z_3^2 + \ldots - \frac{D_\alpha(z_{n-2})}{2h_{n-1}} Z_{n-3} Z_{n-2} \\
&\quad + \left(\frac{D_\alpha(z_{n-2})}{2h_{n-1}} + \frac{D_\alpha(z_{n-1})}{2h_n}\right) Z_{n-2}^2 - \frac{D_\alpha(z_{n-1})}{2h_n} Z_{n-2} Z_{n-1} \\
&\quad - \frac{D_\alpha(z_{n-1})}{2h_n} Z_{n-2} Z_{n-1} + \frac{D_\alpha(z_{n-1})}{2h_n} Z_{n-1}^2 \\
&= \frac{D_\alpha(z_1)}{2h_2} Z_1^2 + \frac{D_\alpha(z_2)}{2h_3} Z_1^2 - \frac{D_\alpha(z_2)}{h_3} Z_1 Z_2 + \frac{D_\alpha(z_2)}{2h_3} Z_2^2 + \frac{D_\alpha(z_3)}{2h_4} Z_2^2 \\
&\quad - \frac{D_\alpha(z_3)}{h_4} Z_2 Z_3 + \frac{D_\alpha(z_3)}{2h_4} Z_3^2 + \ldots - \frac{D_\alpha(z_{n-2})}{h_{n-1}} Z_{n-3} Z_{n-2} + \frac{D_\alpha(z_{n-2})}{2h_{n-1}} Z_{n-2}^2 \\
&\quad + \frac{D_\alpha(z_{n-1})}{2h_n} Z_{n-2}^2 - \frac{D_\alpha(z_{n-1})}{h_n} Z_{n-2} Z_{n-1} + \frac{D_\alpha(z_{n-1})}{2h_n} Z_{n-1}^2 \\
&= \frac{D_\alpha(z_1)}{2h_2} Z_1^2 + \sum_{i=1}^{n-2} \frac{D_\alpha(z_{i+1})}{h_{i+2}} \left(\frac{Z_i^2}{2} - Z_i Z_{i+1} + \frac{Z_{i+1}^2}{2}\right) \\
&= \frac{D_\alpha(z_1)}{2h_2} Z_1^2 + \sum_{i=1}^{n-2} \frac{D_\alpha(z_{i+1})}{h_{i+2}} \left(Z_i - Z_{i+1}\right)^2 \qquad (3.15)
\end{aligned}$$

Now, we find $Z^T S_2 Z$ in a similar way.

$$S_2 Z = \begin{pmatrix} \frac{D_\alpha(z_2)}{2h_2} + \frac{D_\alpha(z_3)}{2h_3} & \frac{-D_\alpha(z_3)}{2h_3} & 0 & \cdots & 0 \\ \frac{-D_\alpha(z_3)}{2h_3} & \frac{D_\alpha(z_3)}{2h_3} + \frac{D_\alpha(z_4)}{2h_4} & \frac{-D_\alpha(z_4)}{2h_4} & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & \frac{-D_\alpha(z_{n-1})}{2h_{n-1}} & \frac{D_\alpha(z_{n-1})}{2h_{n-1}} + \frac{D_\alpha(z_n)}{2h_n} & \frac{-D_\alpha(z_n)}{2h_n} \\ 0 & \cdots & 0 & \frac{-D_\alpha(z_n)}{2h_n} & \frac{D_\alpha(z_n)}{2h_n} \end{pmatrix} \begin{pmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_{n-2} \\ Z_{n-1} \end{pmatrix}$$

$$= \begin{pmatrix} \left(\frac{D_\alpha(z_2)}{2h_2}+\frac{D_\alpha(z_3)}{2h_3}\right)Z_1 - \frac{D_\alpha(z_3)}{2h_3}Z_2 \\ -\frac{D_\alpha(z_3)}{2h_3}Z_1 + \left(\frac{D_\alpha(z_3)}{2h_3}+\frac{D_\alpha(z_4)}{2h_4}\right)Z_2 - \frac{D_\alpha(z_4)}{2h_4}Z_3 \\ \vdots \\ -\frac{D_\alpha(z_{n-1})}{2h_{n-1}}Z_{n-3} + \left(\frac{D_\alpha(z_{n-1})}{2h_{n-1}}+\frac{D_\alpha(z_n)}{2h_n}\right)Z_{n-2} - \frac{D_\alpha(z_n)}{2h_n}Z_{n-1} \\ -\frac{D_\alpha(z_n)}{2h_n}Z_{n-2}+\frac{D_\alpha(z_n)}{2h_n}Z_{n-1} \end{pmatrix}$$

Then,
$$\begin{aligned} Z^TS_2Z &= \left(\frac{D_\alpha(z_2)}{2h_2}+\frac{D_\alpha(z_3)}{2h_3}\right)Z_1^2 - \frac{D_\alpha(z_3)}{2h_3}Z_1Z_2 - \frac{D_\alpha(z_3)}{2h_3}Z_1Z_2 \\ &\quad + \left(\frac{D_\alpha(z_3)}{2h_3}+\frac{D_\alpha(z_4)}{2h_4}\right)Z_2^2 - \frac{D_\alpha(z_4)}{2h_4}Z_2Z_3 - \frac{D_\alpha(z_4)}{2h_4}Z_2Z_3 \\ &\quad + \left(\frac{D_\alpha(z_4)}{2h_4}+\frac{D_\alpha(z_5)}{2h_5}\right)Z_3^2 + \ldots - \frac{D_\alpha(z_{n-1})}{2h_{n-1}}Z_{n-3}Z_{n-2} \\ &\quad + \left(\frac{D_\alpha(z_{n-1})}{2h_{n-1}}+\frac{D_\alpha(z_n)}{2h_n}\right)Z_{n-2}^2 - \frac{D_\alpha(z_n)}{2h_n}Z_{n-2}Z_{n-1} \\ &\quad - \frac{D_\alpha(z_n)}{2h_n}Z_{n-2}Z_{n-1} + \frac{D_\alpha(z_n)}{2h_n}Z_{n-1}^2 \\ &= \frac{D_\alpha(z_2)}{2h_2}Z_1^2 + \frac{D_\alpha(z_3)}{2h_3}Z_1^2 - \frac{D_\alpha(z_3)}{h_3}Z_1Z_2 + \frac{D_\alpha(z_3)}{2h_3}Z_2^2 + \frac{D_\alpha(z_4)}{2h_4}Z_2^2 \\ &\quad - \frac{D_\alpha(z_4)}{h_4}Z_2Z_3 + \frac{D_\alpha(z_4)}{2h_4}Z_3^2 + \ldots - \frac{D_\alpha(z_{n-1})}{h_{n-1}}Z_{n-3}Z_{n-2} + \frac{D_\alpha(z_{n-1})}{2h_{n-1}}Z_{n-2}^2 \\ &\quad + \frac{D_\alpha(z_n)}{2h_n}Z_{n-2}^2 - \frac{D_\alpha(z_n)}{h_n}Z_{n-2}Z_{n-1} + \frac{D_\alpha(z_n)}{2h_n}Z_{n-1}^2 \\ &= \frac{D_\alpha(z_2)}{2h_2}Z_1^2 + \sum_{i=1}^{n-2}\frac{D_\alpha(z_{i+2})}{h_{i+2}}\left(\frac{Z_i^2}{2} - Z_iZ_{i+1} + \frac{Z_{i+1}^2}{2}\right) \\ &= \frac{D_\alpha(z_2)}{2h_2}Z_1^2 + \sum_{i=1}^{n-2}\frac{D_\alpha(z_{i+2})}{h_{i+2}}\left(Z_i - Z_{i+1}\right)^2 \end{aligned} \tag{3.16}$$

Since D_α is a positive function, we get $Z^TS_1Z > 0$ by (3.15), and $Z^TS_2Z > 0$ by (3.16) for $Z > 0$. Thus, $Z^TSZ > 0$ for $Z > 0$. Plus, we have $S_1^T = S_1$ and $S_2^T = S_2$, so $S^T = S$ and S is symmetric. Therefore, S is spd. □

Lemma 3.3.4. The matrix A is positive definite assuming that D_α is a decreasing function.

Proof. Like in section 3.2.2, we can split A into a summation of two matrices A_1 and A_2, so $Z^TAZ = Z^TA_1Z + Z^TA_2Z$.

Let's start by A_1.

$$\begin{aligned}
A_1 Z =& \frac{1}{4}\begin{pmatrix} D_\alpha(z_1) & D_\alpha(z_2) & 0 & \dots & 0 \\ -D_\alpha(z_2) & D_\alpha(z_2) & D_\alpha(z_3) & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & -D_\alpha(z_{n-2}) & D_\alpha(z_{n-2}) & D_\alpha(z_{n-1}) \\ 0 & \dots & 0 & -D_\alpha(z_{n-1}) & D_\alpha(z_{n-1}) \end{pmatrix} \begin{pmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_{n-2} \\ Z_{n-1} \end{pmatrix} \\
=& \frac{1}{4}\begin{pmatrix} D_\alpha(z_1)Z_1 + D_\alpha(z_2)Z_2 \\ -D_\alpha(z_2)Z_1 + D_\alpha(z_2)Z_2 + D_\alpha(z_3)Z_3 \\ \vdots \\ -D_\alpha(z_{n-2})Z_{n-3} + D_\alpha(z_{n-2})Z_{n-2} + D_\alpha(z_{n-1})Z_{n-1} \\ -D_\alpha(z_{n-1})Z_{n-2} + D_\alpha(z_{n-1})Z_{n-1} \end{pmatrix}
\end{aligned}$$

And, $$\begin{aligned}
Z^T A_1 Z =& \frac{1}{4}[D_\alpha(z_1)Z_1^2 + D_\alpha(z_2)Z_1Z_2 - D_\alpha(z_2)Z_1Z_2 + D_\alpha(z_2)Z_2^2 + D_\alpha(z_3)Z_2Z_3 \\
& + \dots - D_\alpha(z_{n-2})Z_{n-3}Z_{n-2} + D_\alpha(z_{n-2})Z_{n-2}^2 + D_\alpha(z_{n-1})Z_{n-2}Z_{n-1} \\
& - D_\alpha(z_{n-1})Z_{n-2}Z_{n-1} + D_\alpha(z_{n-1})Z_{n-1}^2] \\
=& \frac{1}{4}\left(D_\alpha(z_1)Z_1^2 + D_\alpha(z_2)Z_2^2 + \dots + D_\alpha(z_{n-2})Z_{n-2}^2 + D_\alpha(z_{n-1})Z_{n-1}^2\right) \\
=& \frac{1}{4}\sum_{i=1}^{n-1} D_\alpha(z_i)Z_i^2
\end{aligned} \tag{3.17}$$

Now, we find $Z^T A_2 Z$.

$$\begin{aligned}
A_2 Z =& \frac{1}{4}\begin{pmatrix} -D_\alpha(z_3) & D_\alpha(z_3) & 0 & \dots & 0 \\ -D_\alpha(z_3) & -D_\alpha(z_4) & D_\alpha(z_4) & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & -D_\alpha(z_{n-1}) & -D_\alpha(z_n) & D_\alpha(z_n) \\ 0 & \dots & 0 & -D_\alpha(z_n) & D_\alpha(z_n) \end{pmatrix} \begin{pmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_{n-2} \\ Z_{n-1} \end{pmatrix} \\
=& \frac{1}{4}\begin{pmatrix} -D_\alpha(z_3)Z_1 + D_\alpha(z_3)Z_2 \\ -D_\alpha(z_3)Z_1 - D_\alpha(z_4)Z_2 + D_\alpha(z_4)Z_3 \\ \vdots \\ -D_\alpha(z_{n-1})Z_{n-3} - D_\alpha(z_n)Z_{n-2} + D_\alpha(z_n)Z_{n-1} \\ -D_\alpha(z_n)Z_{n-2} + D_\alpha(z_n)Z_{n-1} \end{pmatrix}
\end{aligned}$$

Then, $Z^T A_2 Z = \frac{1}{4}(-D_\alpha(z_3)Z_1^2 + D_\alpha(z_3)Z_1Z_2 - D_\alpha(z_3)Z_1Z_2 - D_\alpha(z_4)Z_2^2 + D_\alpha(z_4)Z_2Z_3$

$$
\begin{aligned}
&+\ldots - D_\alpha(z_{n-1})Z_{n-3}Z_{n-2} - D_\alpha(z_n)Z_{n-2}^2 + D_\alpha(z_n)Z_{n-2}Z_{n-1} \\
&- D_\alpha(z_n)Z_{n-2}Z_{n-1} + D_\alpha(z_n)Z_{n-1}^2) \\
=&\frac{1}{4}\left(-D_\alpha(z_3)Z_1^2 - D_\alpha(z_4)Z_2^2 - \ldots - D_\alpha(z_n)Z_{n-2}^2 + D_\alpha(z_n)Z_{n-1}^2\right) \\
=&\frac{1}{4}\left(D_\alpha(z_n)Z_{n-1}^2 - \sum_{i=1}^{n-2} D_\alpha(z_{i+2})Z_i^2\right) \qquad (3.18)
\end{aligned}
$$

by (3.17) and (3.18), we have:

$$
\begin{aligned}
Z^TAZ =&Z^TA_1Z + Z^TA_2Z \\
=&\frac{1}{4}\sum_{i=1}^{n-1} D_\alpha(z_i)Z_i^2 + \frac{1}{4}\left(D_\alpha(z_n)Z_{n-1}^2 - \sum_{i=1}^{n-2} D_\alpha(z_{i+2})Z_i^2\right) \\
=&\frac{1}{4}\left(\left(D_\alpha(z_{n-1}) + D_\alpha(z_n)\right)Z_{n-1}^2 + \sum_{i=1}^{n-2}\left(D_\alpha(z_i) - D_\alpha(z_{i+2})\right)Z_i^2\right)
\end{aligned}
$$

Since D_α is a positive function then $D_\alpha(z_{n-1}) + D_\alpha(z_n) > 0$ and if it's also decreasing, then $D_\alpha(z_i) - D_\alpha(z_{i+2}) > 0$ for $i = 1, \ldots, n-2$. Therefore, $Z^TAZ > 0$ for $Z > 0$. Hence, the matrix A is positive definite. □

Theorem 3.3.5. The Finite Element Euler-Implicit Discrete System (3.6) admits a unique solution assuming that D_α is a decreasing function.

Proof. Let's take the matrix $\mathscr{A}_{\Delta t} := M + \Delta t\left(\frac{\mathcal{G}}{f}M + \frac{1}{f}S - K - \frac{\mathcal{M}_\alpha}{f}A + B\right)$ and so (3.6) is equivalent to:

$$
\begin{cases}
\mathscr{A}_{\Delta t}\Lambda(t+\Delta t) = M\Lambda(t) - v_1(t) - \Delta t v_3(t) \\
\Lambda(0) = \bar{\Lambda}
\end{cases}
\qquad (3.19)
$$

Uniqueness of this system is obtained when $\mathscr{A}_{\Delta t}$ is invertible. And invertibility of $\mathscr{A}_{\Delta t}$ is obtained if $\mathscr{A}_{\Delta t}$ is positive definite. Which is done by showing that $Z^T\mathscr{A}_{\Delta t}Z > 0, \forall Z \in \mathbb{R}^{n-1} - \{0\}$.
$\mathscr{A}_{\Delta t}$ is a summation of the matrices M, S, K, A and B multiplied by positive constants. Every matrix of this summation is proved to be positive definite by the lemmas 3.3.1, 3.3.2, 3.3.3, and 3.3.4. Therefore, $\mathscr{A}_{\Delta t}$ is positive definite and the system (3.6) admits a unique solution. □

3.4 Testing

In this section, we present the results of our numerical simulations implemented using `MATLAB`. We test the direct problem for different meshing and while fixing the smallest mesh size, we compare the errors using these norms: L_∞ norm, relative L_∞ norm, L_2 norm, and relative L_2 norm. First, we test it by taking Δt to be of order h^2 i.e $\Delta t = O(h^2)$. Second, we test it using $\Delta t = O(h)$. Finally, we analyze the results obtained in both cases.

In the following testings, since we do not have the exact data, we will take $\bar{\rho}(z) = 0$, $D_\alpha(z) = -99.998z + 100$ a decreasing function, $\rho_\alpha^{atm}(t) = 2t^{1/4}$ and the following values for the constants mentioned in table 1.1:
$z_F = 1, f = 0.2, v = 200, w_{air} = 485, \tau = 10, \lambda = 0.03, M_\alpha = 0.04, g = 9.8,$
$R = 8.314, T_m = 260$

3.4.1 Case 1: $\Delta t = O(h^2)$

In this test, we will consider Δt to be of order h^2 and take four different meshing: $h_1 = \frac{1}{8}$, $h_2 = \frac{1}{16}$, $h_3 = \frac{1}{32}$ and $h_4 = \frac{1}{64}$ and three end time: $T_1 = 1$ $T_2 = 32$ and $T_3 = 128$. By running the direct problem, Algorithm A.4, for each h_i, we get the value of the concentration ρ^o_{α,h_i}. And we test convergence on the space common points (z_c) at each end time T_j for $j = 1, 2, 3$ by finding the error between each ρ^o_{α,h_i} and ρ^o_{α,h_4} for $i < 4$.

In the tables 3.1, 3.3 and 3.5, we are presenting the errors using the L_∞ norm and the relative L_∞ norm. As for the tables 3.2, 3.4 and 3.6, we are presenting the errors using the L_2 norm and the relative L_2 norm.

h_i	$\Delta t_i = O(h_i^2)$	$\left\|\rho^o_{\alpha,h_i}(z_c,T_1) - \rho^o_{\alpha,h_4}(z_c,T_1)\right\|_\infty$	$\dfrac{\left\|\rho^o_{\alpha,h_i}(z_c,T_1) - \rho^o_{\alpha,h_4}(z_c,T_1)\right\|_\infty}{\left\|\rho^o_{\alpha,h_4}(z_c,T_1)\right\|_\infty}$
$\frac{1}{8}$	$\left(\frac{1}{8}\right)^2$	0.1545	0.0772
$\frac{1}{16}$	$\left(\frac{1}{16}\right)^2$	0.0625	0.0313
$\frac{1}{32}$	$\left(\frac{1}{32}\right)^2$	0.0203	0.0101

Table 3.1: The L_∞ and relative L_∞ error at (z_c, T_1) with $\Delta t = O(h^2)$.

h_i	$\Delta t_i = O(h_i^2)$	$\left\lVert \rho^o_{\alpha,h_i}(z_c,T_1) - \rho^o_{\alpha,h_4}(z_c,T_1) \right\rVert_2$	$\dfrac{\left\lVert \rho^o_{\alpha,h_i}(z_c,T_1) - \rho^o_{\alpha,h_4}(z_c,T_1) \right\rVert_2}{\left\lVert \rho^o_{\alpha,h_4}(z_c,T_1) \right\rVert_2}$
$\frac{1}{8}$	$\left(\frac{1}{8}\right)^2$	0.2978	0.0755
$\frac{1}{16}$	$\left(\frac{1}{16}\right)^2$	0.1215	0.0308
$\frac{1}{32}$	$\left(\frac{1}{32}\right)^2$	0.0399	0.0101

Table 3.2: The L_2 and relative L_2 error at (z_c, T_1) with $\Delta t = O(h^2)$.

h_i	$\Delta t_i = O(h_i^2)$	$\left\lVert \rho^o_{\alpha,h_i}(z_c,T_2) - \rho^o_{\alpha,h_4}(z_c,T_2) \right\rVert_\infty$	$\dfrac{\left\lVert \rho^o_{\alpha,h_i}(z_c,T_2) - \rho^o_{\alpha,h_4}(z_c,T_2) \right\rVert_\infty}{\left\lVert \rho^o_{\alpha,h_4}(z_c,T_2) \right\rVert_\infty}$
$\frac{1}{8}$	$\left(\frac{1}{8}\right)^2$	0.3674	0.0772
$\frac{1}{16}$	$\left(\frac{1}{16}\right)^2$	0.1487	0.0313
$\frac{1}{32}$	$\left(\frac{1}{32}\right)^2$	0.0482	0.0101

Table 3.3: The L_∞ and relative L_∞ error at (z_c, T_2) with $\Delta t = O(h^2)$.

h_i	$\Delta t_i = O(h_i^2)$	$\left\lVert \rho^o_{\alpha,h_i}(z_c,T_2) - \rho^o_{\alpha,h_4}(z_c,T_2) \right\rVert_2$	$\dfrac{\left\lVert \rho^o_{\alpha,h_i}(z_c,T_2) - \rho^o_{\alpha,h_4}(z_c,T_2) \right\rVert_2}{\left\lVert \rho^o_{\alpha,h_4}(z_c,T_2) \right\rVert_2}$
$\frac{1}{8}$	$\left(\frac{1}{8}\right)^2$	0.7082	0.0755
$\frac{1}{16}$	$\left(\frac{1}{16}\right)^2$	0.2891	0.0308
$\frac{1}{32}$	$\left(\frac{1}{32}\right)^2$	0.0949	0.0101

Table 3.4: The L_2 and relative L_2 error at (z_c, T_2) with $\Delta t = O(h^2)$.

h_i	$\Delta t_i = O(h_i^2)$	$\left\|\rho^o_{\alpha,h_i}(z_c,T_3) - \rho^o_{\alpha,h_4}(z_c,T_3)\right\|_\infty$	$\dfrac{\left\|\rho^o_{\alpha,h_i}(z_c,T_3) - \rho^o_{\alpha,h_4}(z_c,T_3)\right\|_\infty}{\left\|\rho^o_{\alpha,h_4}(z_c,T_3)\right\|_\infty}$
$\frac{1}{8}$	$\left(\frac{1}{8}\right)^2$	0.5196	0.0772
$\frac{1}{16}$	$\left(\frac{1}{16}\right)^2$	0.2102	0.0313
$\frac{1}{32}$	$\left(\frac{1}{32}\right)^2$	0.0682	0.0101

Table 3.5: The L_∞ and relative L_∞ error at (z_c, T_3) with $\Delta t = O(h^2)$.

h_i	$\Delta t_i = O(h_i^2)$	$\left\|\rho^o_{\alpha,h_i}(z_c,T_3) - \rho^o_{\alpha,h_4}(z_c,T_3)\right\|_2$	$\dfrac{\left\|\rho^o_{\alpha,h_i}(z_c,T_3) - \rho^o_{\alpha,h_4}(z_c,T_3)\right\|_2}{\left\|\rho^o_{\alpha,h_4}(z_c,T_3)\right\|_2}$
$\frac{1}{8}$	$\left(\frac{1}{8}\right)^2$	1.0016	0.0755
$\frac{1}{16}$	$\left(\frac{1}{16}\right)^2$	0.4088	0.0308
$\frac{1}{32}$	$\left(\frac{1}{32}\right)^2$	0.1342	0.0101

Table 3.6: The L_2 and relative L_2 error at (z_c, T_3) with $\Delta t = O(h^2)$.

In all the tables , it's clear that when h gets smaller for any end time, both relative and absolute errors, for L_2 and L_∞ norms, are getting smaller. Adding that for both norms, the relative error is even smaller than the absolute error and it's value didn't change while T increases. Based on these errors, we can see that we have convergence for $\Delta t = O(h^2)$ in all different meshes knowing that we are not computing with the exact data.

3.4.2 Case 2: $\Delta t = O(h)$

In this test, we will consider Δt to be of order h and take four different meshing: $h_1 = \frac{1}{8}$, $h_2 = \frac{1}{16}$, $h_3 = \frac{1}{32}$ and $h_4 = \frac{1}{64}$ and three end time: $T_1 = 1$ $T_2 = 32$ and $T_3 = 128$. By running the direct problem, Algorithm A.4, for each h_i, we get the value of the concentration ρ^o_{α,h_i}. And we test convergence on the space common points (z_c) at each end time T_j for $j = 1, 2, 3$ by finding the error between each ρ^o_{α,h_i} and ρ^o_{α,h_4} for $i < 4$.

In the tables 3.7, 3.9 and 3.11, we are presenting the errors using the L_∞ norm and the relative L_∞ norm. As for the tables 3.8, 3.10 and 3.12, we are presenting the errors using the L_2 norm and the relative L_2 norm.

h_i	$\Delta t_i = O(h_i)$	$\left\|\rho^o_{\alpha,h_i}(z_c,T_1) - \rho^o_{\alpha,h_4}(z_c,T_1)\right\|_\infty$	$\dfrac{\left\|\rho^o_{\alpha,h_i}(z_c,T_1) - \rho^o_{\alpha,h_4}(z_c,T_1)\right\|_\infty}{\left\|\rho^o_{\alpha,h_4}(z_c,T_1)\right\|_\infty}$
$\frac{1}{8}$	$\frac{1}{8}$	0.1545	0.0772
$\frac{1}{16}$	$\frac{1}{16}$	0.0625	0.0313
$\frac{1}{32}$	$\frac{1}{32}$	0.0203	0.0101

Table 3.7: The L_∞ and relative L_∞ error at (z_c, T_1) with $\Delta t = O(h)$.

h_i	$\Delta t_i = O(h_i)$	$\left\|\rho^o_{\alpha,h_i}(z_c,T_1) - \rho^o_{\alpha,h_4}(z_c,T_1)\right\|_2$	$\dfrac{\left\|\rho^o_{\alpha,h_i}(z_c,T_1) - \rho^o_{\alpha,h_4}(z_c,T_1)\right\|_2}{\left\|\rho^o_{\alpha,h_4}(z_c,T_1)\right\|_2}$
$\frac{1}{8}$	$\frac{1}{8}$	0.2977	0.0755
$\frac{1}{16}$	$\frac{1}{16}$	0.1215	0.0308
$\frac{1}{32}$	$\frac{1}{32}$	0.0399	0.0101

Table 3.8: The L_2 and relative L_2 error at (z_c, T_1) with $\Delta t = O(h)$.

h_i	$\Delta t_i = O(h_i)$	$\left\|\rho^o_{\alpha,h_i}(z_c,T_2) - \rho^o_{\alpha,h_4}(z_c,T_2)\right\|_\infty$	$\dfrac{\left\|\rho^o_{\alpha,h_i}(z_c,T_2) - \rho^o_{\alpha,h_4}(z_c,T_2)\right\|_\infty}{\left\|\rho^o_{\alpha,h_4}(z_c,T_2)\right\|_\infty}$
$\frac{1}{8}$	$\frac{1}{8}$	0.3674	0.0772
$\frac{1}{16}$	$\frac{1}{16}$	0.1487	0.0313
$\frac{1}{32}$	$\frac{1}{32}$	0.0482	0.0101

Table 3.9: The L_∞ and relative L_∞ error at (z_c, T_2) with $\Delta t = O(h)$.

h_i	$\Delta t_i = O(h_i)$	$\left\|\rho^o_{\alpha,h_i}(z_c,T_2) - \rho^o_{\alpha,h_4}(z_c,T_2)\right\|_2$	$\dfrac{\left\|\rho^o_{\alpha,h_i}(z_c,T_2) - \rho^o_{\alpha,h_4}(z_c,T_2)\right\|_2}{\left\|\rho^o_{\alpha,h_4}(z_c,T_2)\right\|_2}$
$\frac{1}{8}$	$\frac{1}{8}$	0.7082	0.0755
$\frac{1}{16}$	$\frac{1}{16}$	0.2891	0.0308
$\frac{1}{32}$	$\frac{1}{32}$	0.0949	0.0101

Table 3.10: The L_2 and relative L_2 error at (z_c, T_2) with $\Delta t = O(h)$.

h_i	$\Delta t_i = O(h_i)$	$\left\|\rho^o_{\alpha,h_i}(z_c,T_3) - \rho^o_{\alpha,h_4}(z_c,T_3)\right\|_\infty$	$\dfrac{\left\|\rho^o_{\alpha,h_i}(z_c,T_3) - \rho^o_{\alpha,h_4}(z_c,T_3)\right\|_\infty}{\left\|\rho^o_{\alpha,h_4}(z_c,T_3)\right\|_\infty}$
$\frac{1}{8}$	$\frac{1}{8}$	0.5196	0.0772
$\frac{1}{16}$	$\frac{1}{16}$	0.2102	0.0313
$\frac{1}{32}$	$\frac{1}{32}$	0.0682	0.0101

Table 3.11: The L_∞ and relative L_∞ error at (z_c, T_3) with $\Delta t = O(h)$.

h_i	$\Delta t_i = O(h_i)$	$\left\|\rho^o_{\alpha,h_i}(z_c,T_3) - \rho^o_{\alpha,h_4}(z_c,T_3)\right\|_2$	$\dfrac{\left\|\rho^o_{\alpha,h_i}(z_c,T_3) - \rho^o_{\alpha,h_4}(z_c,T_3)\right\|_2}{\left\|\rho^o_{\alpha,h_4}(z_c,T_3)\right\|_2}$
$\frac{1}{8}$	$\frac{1}{8}$	1.0016	0.0755
$\frac{1}{16}$	$\frac{1}{16}$	0.4088	0.0308
$\frac{1}{32}$	$\frac{1}{32}$	0.1342	0.0101

Table 3.12: The L_2 and relative L_2 error at (z_c, T_3) with $\Delta t = O(h)$.

In all the tables , it's clear that when h gets smaller for any end time, both relative and absolute errors, for L_2 and L_∞ norms, are getting smaller. Adding that for both norms, the relative error is even smaller than the absolute error and it's value didn't change while T increases. Based on these errors, we can see that we have convergence for $\Delta t = O(h)$ in all different meshes knowing that we are not comparing with the exact data.

3.4.3 Analysis of the Results

By looking at all the tables above, and the fact that these are not the exact data, we can see that the errors are too small and we have convergence in all cases. Also, by comparing respectively the tables 3.1, 3.2, 3.3, 3.4, 3.5 and 3.6 to the tables 3.7, 3.8, 3.9, 3.10, 3.11 and 3.12, it's clear that the errors are exactly the same. Hence, we conclude that there is no difference between taking Δt to be of order h or of order h^2 for any end time. So, for efficiency, we will consider $\Delta t = O(h)$ in the rest of the thesis.

Chapter 4

Formulation and Implementation of the Inverse Problem

In this chapter, we attempt to find the inverse problem and handle it numerically by minimizing the objective function using a `MATLAB` function called `fmincon`.

4.1 The Objective Function

The ultimate goal of this thesis is to determine the diffusion coefficient D_α of a particular gas, using data from measurements $\rho^o_\alpha(z,T)$, $z \in (0, z_F)$ made of several gases at the end time T.
We stated in the introduction, section 1.2, that the the diffusion coefficients D_α is given by:

$$D_\alpha(z) = r_\alpha c_f D_{CO2,air}(z) \tag{4.1}$$

where c_f and r_α are known constants.
Thus, when $D_{CO2,air}$ is found, all other D_α's can be then obtained. We seek then $D(z) \equiv D_{CO2,air}(z) \in X_c := \{v \in C(0, z_F) \,|\, v > 0\}$.

Let $\rho^o_\alpha(\tilde{D}; ., T)$ be the unique solution of the direct problem at the end time T, $\forall \tilde{D} \in X_c$, and $\rho^o_{\alpha,meas}(., T)$ be the measured concentration at the end time T. As mentioned in the introduction, the inverse problem is equivalent to an optimization problem with the following objective function:

$$\forall \tilde{D} \in X_c : V(\tilde{D}) = \sum_{\alpha \in S} \left\| \rho^o_\alpha(\tilde{D}; ., T) - \rho^o_{\alpha,meas}(., T) \right\|_2^2$$

where S is the set of all the gases in the Firn.
So we seek $D \in X_c$ such that:

$$V(D) = \min_{\tilde{D} \in X_c} V(\tilde{D})$$

For computational purpose, we introduce $h, \Delta t$ and then define $\rho^o_{\alpha,h,\Delta t}(.,T)$ as the solution of the direct problem for a given D_g since we don't have the exact data.
Take this $\rho^o_{\alpha,h,\Delta t} \equiv \rho^o_{\alpha,g}$ the approximated and generated solution of the direct problem with the generated diffusion coefficient D_g. The objective function then becomes:

$$\forall \tilde{D} \in X_{c,h} : V_{h,\Delta t}(\tilde{D}) = \sum_{\alpha \in S} \left\| \rho^o_{\alpha,h,\Delta t}(\tilde{D}; ., T) - \rho^o_{\alpha,g}(., T) \right\|_2^2$$

with $X_{c,h} \subset X_c$ the set of all piecewize continuous positive functions.
So we seek $D \in X_{c,h}$ such that:

$$V_{h,\Delta t}(D) = \min_{\tilde{D} \in X_{c,h}} V_{h,\Delta t}(\tilde{D}) \tag{4.2}$$

4.2 Implementation: Use of `MATLAB` `fmincon`

Before solving an optimization problem, one must choose the appropriate `MATLAB` function. Based on [7], there are five functions to solve a nonlinear optimization problem:

1. `fminbnd`: Find minimum of single-variable function on fixed interval.

2. `fmincon`: Find minimum of constrained nonlinear multivariable function.

3. `fminsearch`: Find minimum of unconstrained multivariable function using derivative-free method.

4. `fminunc`: Find minimum of unconstrained multivariable function.

5. `fseminf`: Find minimum of semi-infinitely constrained multivariable nonlinear function.

In this problem (4.2), we will use the `MATLAB` function `fmincon` since we have a constrained nonlinear multivariable function.

4.2.1 Description of `fmincon`

In general, the function `fmincon` starts at x_0 and attempts to find a minimizer x of the problem specified by

$$\min_x f(x) \text{ such that } \begin{cases} c(x) \leq 0 \\ ceq(x) = 0 \\ A.x \leq b \\ Aeq.x = beq \\ lb \leq x \leq ub \end{cases}$$

with b and beq are vectors, A and Aeq are matrices, lb and ub can be passed as vectors or matrices, x_0 can be a scalar, vector, or matrix, $c(x)$ and $ceq(x)$ are functions that return vectors, and $f(x)$ is a function that returns a scalar. $f(x)$, $c(x)$, and $ceq(x)$ can be nonlinear functions.

In our problem, x is D and $f(x)$ is $V(D)$ and we only need D to be positive. So, $lb = 0$ and no need for the other inputs $(c(x), ceq(x), A, Aeq, b, beq$ and $ub)$.

`fmincon` have five algorithm options: `interior-point`, `sqp`, `sqp-legacy`, `active-set` and `trust-region-reflective`. We will use in this thesis just these two algorithms:

1. `interior-point`: Interior point methods or barrier methods are a certain class of algorithms to solve linear and nonlinear convex optimization problems. Violation of inequality constraints are prevented by augmenting the objective function with a barrier term that causes the optimal unconstrained value to be in the feasible space.

2. `sqp`: sqp methods solve a sequence of optimization subproblems, each of which optimizes a quadratic model of the objective subject to a linearization of the constraints. If the problem is unconstrained, then the method reduces to Newton's method for finding a point where the gradient of the objective vanishes. If the problem has only equality constraints, then the method is equivalent to applying Newton's method to the first-order optimality conditions, or Karush–Kuhn–Tucker conditions, of the problem.

4.2.2 Tolerance and Stopping Criteria

The number of iterations in an optimization depends on a solver's stopping criteria. These criteria include several tolerances one can set. Generally, a tolerance is a threshold which, if crossed, stops the iterations of a solver.
The stopping criteria of the algorithms `interior-point` and `sqp` is the same and depends on the StepTolerance and OptimalityTolerance.

The StepTolerance is a lower bound on the size of a step, meaning the relative norm of $(x_i - x_{i+1})$. If the solver attempts to take a step that is smaller than StepTolerance, the iterations end. The default value for all algorithms except `interior-point` is 10^{-6}, for the `interior-point` algorithm, the default is 10^{-10}.

The OptimalityTolerance is a tolerance for the first-order optimality measure. If the optimality measure is less than OptimalityTolerance, the iterations end. The default value is 10^{-6}. For more details on first-order optimality measure, refer to [7] on pages 3-11.

4.3 Testing

In this section, we present the results of our numerical simulations implemented using `MATLAB`. We test the inverse problem, Algorithm A.5, using the `MATLAB` function `fmincon` for different meshing, algorithms, and initial vector guess. Then, we compare the errors using the relative and absolute L_2 norm.

In this thesis, since we do not have the exact data, we will consider 3 α gases, $r_\alpha = [1,\ 2,\ 3]^T$, $c_f = 0.5$ and generate the decreasing diffusion coefficient $D_g(z) = -99.998z + 100$. By applying (4.1), we get the generated decreasing function $D_{\alpha,g}(z) = r_\alpha c_f D_g(z)$, and the other needed values are taken the same as in section 3.4. Following that, we run the direct problem, Algorithm A.4, that outputs $\rho^o_{\alpha,g}(z,T)$, the generated concentration of each gas α at the end time T.

Our aim here is to compare the error between the $D_{\alpha,g}$'s generated and the D_α's computed. Since both of these diffusion coefficients are obtained using (4.1), then this error will be the same as the error between D_g generated and D computed.

4.3.1 Case 1: Initial Vector Guess $D_0 = 0$ and $T = 1$

In this test, we found our goal D by minimizing the objective function starting by the initial vector guess $D_0 = 0$ using the `MATLAB` function `fmincon` with an OptimalityTolerance $= 10^{-8}$ and by taking the end time $T = 1$.

In the table 4.1, we are presenting, for 4 different meshes, the following results: the number of iterations, the time needed to achieve the minimum using tic-toc functions, the relative L_2 error between D and D_g, and $V(D)$ the value of the objective function at D once using the `sqp` algorithm and once again using the `interior-point` algorithm.

h	`sqp` algorithm				`interior-point` algorithm			
	iter	time (/s)	$\frac{\|D-D_g\|_2}{\|D_g\|_2}$	$V(D)$	iter	time (/s)	$\frac{\|D-D_g\|_2}{\|D_g\|_2}$	$V(D)$
$\frac{1}{4}$	75	0.145518	0.000034	6.2375e-15	75	0.230846	0.1635	1.3278e-13
$\frac{1}{8}$	132	0.490532	0.000036	1.6449e-13	148	0.788748	0.1036	4.3088e-13
$\frac{1}{16}$	303	3.368427	0.000154	3.9819e-12	291	3.434006	0.0528	2.2870e-11
$\frac{1}{32}$	1216	54.397609	0.080742	3.7647e-07	570	25.226179	0.0074	8.0227e-09

Table 4.1: Number of iterations (iter), time taken to find the minimum , relative error, and value of $V(D)$ for each mesh using the two algorithms `sqp` and `interior-point` with $D_0 = 0$ and $T = 1$.

The results in this table indicate:

- In `sqp`, the number of iterations increases for smaller h, while it remains moderate in `interior-point`. For $h = \frac{1}{4}$, the number of iterations is the same for both algorithms. In `sqp`, the number of iterations is smaller than that in `interior-point` for $h = \frac{1}{8}$, and it's the other way around for $h = \frac{1}{16}$. In the case $h = \frac{1}{32}$, the number of iterations is 2 times greater in `sqp` than in `interior-point`.

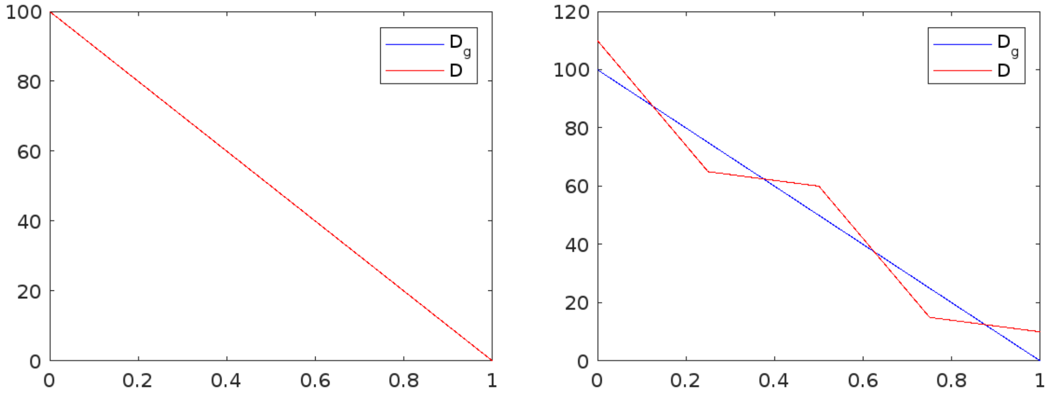
- In `sqp` and `interior-point`, the time needed to attend the minimum is increasing while h decreases. We can see that `sqp` is the fastest method by looking for example at the case $h = \frac{1}{4}$: `interior-point` and `sqp` have the same number of iterations, but `sqp` takes less time.

- On the other hand, the relative error is decreasing in `interior-point` and increasing in `sqp`. And for $h = \frac{1}{4}, \frac{1}{8}, \frac{1}{16}$, the relative error is smaller in `sqp` method than that in `interior-point`, while it's the other way around for $h = \frac{1}{32}$.

- In both algorithms, the value of the objective function at D is very small and increases for h smaller.

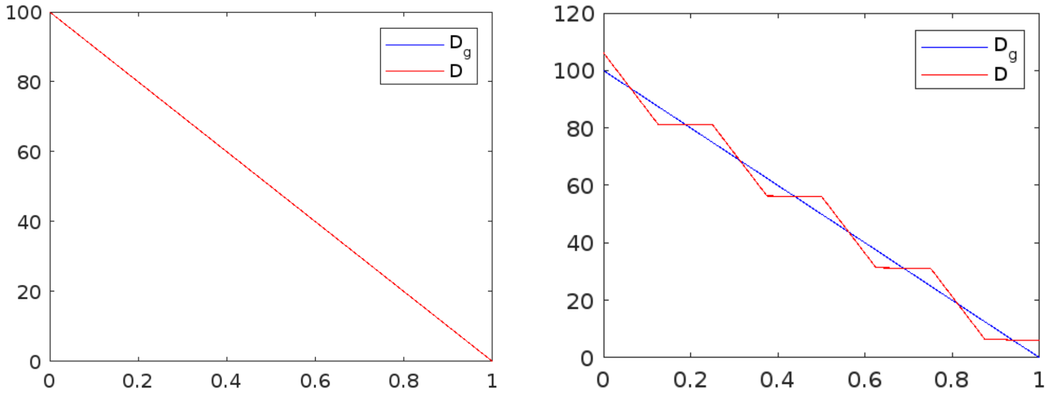
Based on the above observations and taking into consideration all of the following factors: time, relative error, and number of iteration, we can conclude that the `interior-point` method performs better for a small mesh size ($h = \frac{1}{32}$) while the `sqp` method works better for a larger mesh size h.

In figures 4.1, 4.2, 4.3 and 4.4, we plot D_g and D obtained using `sqp` algorithm (4.1a, 4.2a, 4.3a, 4.4a) versus `interior-point` algorithm (4.1b, 4.2b, 4.3b, 4.4b) for decreasing mesh sizes. Furthermore in the figure 4.5, we present the absolute error $\|D - D_g\|_2$ for every h.



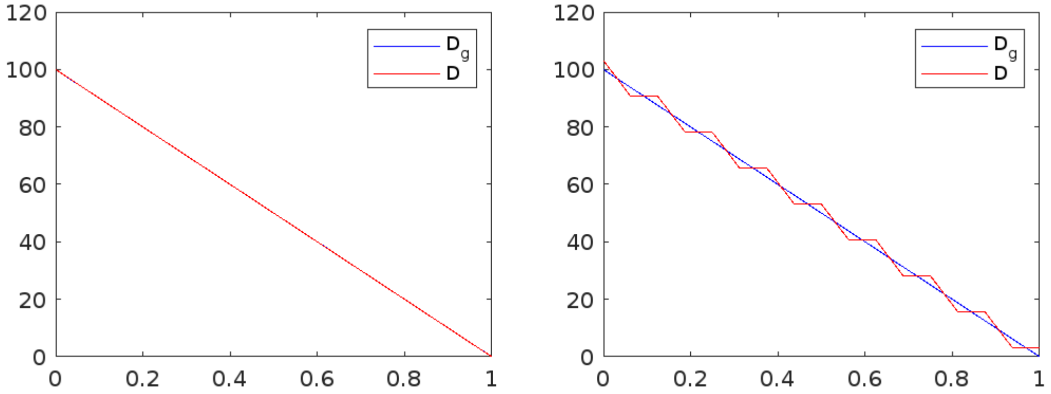
(a) `sqp` algorithm (b) `interior-point` algorithm

Figure 4.1: D and D_g for $h = \frac{1}{4}$ with $D_0 = 0$ and $T = 1$.



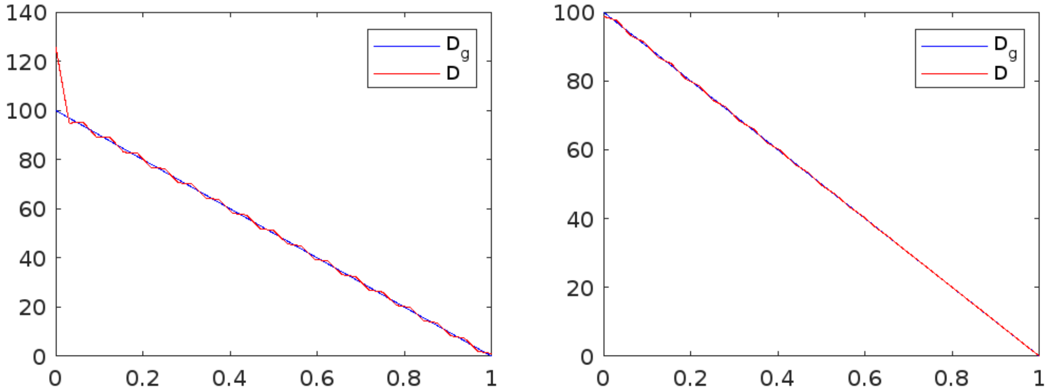
(a) `sqp` algorithm (b) `interior-point` algorithm

Figure 4.2: D and D_g for $h = \frac{1}{8}$ with $D_0 = 0$ and $T = 1$.



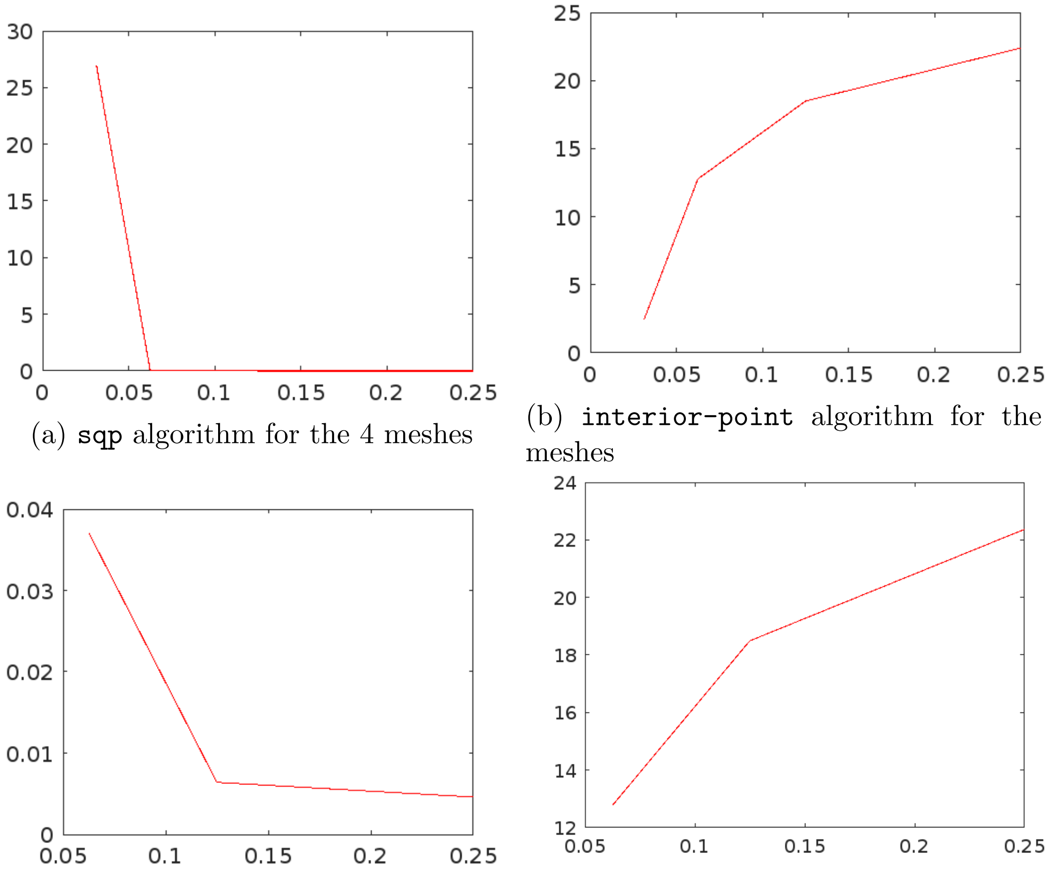
(a) `sqp` algorithm (b) `interior-point` algorithm

Figure 4.3: D and D_g for $h = \frac{1}{16}$ with $D_0 = 0$ and $T = 1$.



(a) **sqp** algorithm (b) **interior-point** algorithm

Figure 4.4: D and D_g for $h = \frac{1}{32}$ with $D_0 = 0$ and $T = 1$.



(a) **sqp** algorithm for the 4 meshes

(b) **interior-point** algorithm for the 4 meshes

(c) **sqp** algorithm for first 3 meshes

(d) **interior-point** algorithm for first 3 meshes

Figure 4.5: The absolute L_2 error: $\|D - D_g\|_2$ with $D_0 = 0$ and $T = 1$.

Based on all the figures of this case, it's clear that in the `interior-point` method, when h gets smaller, D_g and D coincide and the absolute error $\|D - D_g\|_2$ gets smaller. While in `sqp`, D_g and D coincide for the first 3 meshes and by looking to the figures 4.5a and 4.5c, the absolute error strongly decreases from $h = \frac{1}{32}$ to $h = \frac{1}{16}$ and continue decreasing moderately to $h = \frac{1}{4}$. Also, by respectively comparing figures 4.5a and 4.5c to 4.5b and 4.5d, we can observe that in the first 3 meshes the absolute error is 343 to 4800 times smaller when applying `sqp` than that when applying `interior-point`, while it's 11 times smaller when applying `interior-point` than that when applying `sqp` in the last mesh. Noticing also that D and D_g are both decreasing functions in all figures.

4.3.2 Case 2: Initial Vector Guess D_0= Random and $T = 1$

In this test, we found our goal D by minimizing the objective function for end time $T = 1$ using the `MATLAB` function `fmincon` with an Optimality Tolerance = 10^{-8} and a starting vector guess D_0 which is a random vector of length $\frac{1}{h} + 1$ with entries between 0 and 100 i.e D_0 =100*rand($\frac{1}{h} + 1$,1).

In the table 4.2, we present, for 4 different meshes, the number of iterations and time needed to achieve the minimum, the relative L_2 error between D and D_g, and $V(D)$ the value of the objective function at D once using the `sqp` algorithm and once again using the `interior-point` algorithm.

h	`sqp` algorithm				`interior-point` algorithm			
	iter	time (/s)	$\frac{\|D - D_g\|_2}{\|D_g\|_2}$	$V(D)$	iter	time (/s)	$\frac{\|D - D_g\|_2}{\|D_g\|_2}$	$V(D)$
$\frac{1}{4}$	72	0.083194	0.0016	1.8527e-15	79	0.107577	0.1410	4.2164e-14
$\frac{1}{8}$	146	0.251943	0.0095	2.9692e-12	154	0.308407	0.1063	6.2101e-13
$\frac{1}{16}$	291	1.533312	0.0025	3.6316e-09	282	1.663420	0.0446	1.8178e-11
$\frac{1}{32}$	640	13.639654	0.0293	1.1850e-07	636	14.151535	0.0209	5.9908e-08

Table 4.2: Number of iterations (iter), time taken to find the minimum, relative error, and value of $V(D)$ for each mesh using the two algorithms `sqp` and `interior-point` with D_0= random and $T = 1$.

The results in this table indicate:

- The number of iterations increases for smaller h in both algorithms. Noticing that for every h, the difference in the number of iterations between `sqp` and `interior-point` does not exceed 10.
- In `sqp` and `interior-point`, the time needed to attend the minimum is increasing while h decreases. We can see that `sqp` is the fastest method by

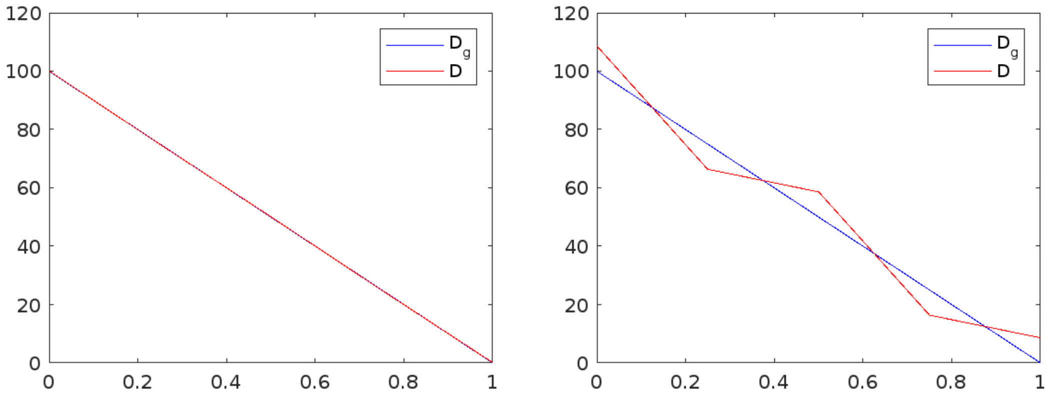
looking for example at the cases $h = \frac{1}{16}$ and $\frac{1}{32}$: the number of iterations in `sqp` is bigger than the number of iterations in `interior-point`, but `sqp` takes less time.

- As for the relative error, while h decreases, it increases in `sqp` (except for $h = \frac{1}{8}$) and decreases in `interior-point`. And for $h = \frac{1}{4}, \frac{1}{8}, \frac{1}{16}$, the relative error is 11 to 88 times smaller in `sqp` method than that in `interior-point` method, while the relative error for $h = \frac{1}{32}$ is smaller in `interior-point` than that in `sqp` with a slight difference in the values. For that, convergence in relative norm appears to be slightly better in the last mesh when applying the `interior-point` algorithm, while it's much better in the larger meshes when applying the algorithm `sqp`.

- In both algorithms, the value of the objective function at D is very small and increases for h smaller.

Based on the above observations and taking into consideration all of the following factors: time, relative error, and number of iteration, we can conclude that the `interior-point` method performs slightly better for a small mesh size ($h = \frac{1}{32}$) while the `sqp` method performs much better for a larger mesh size.

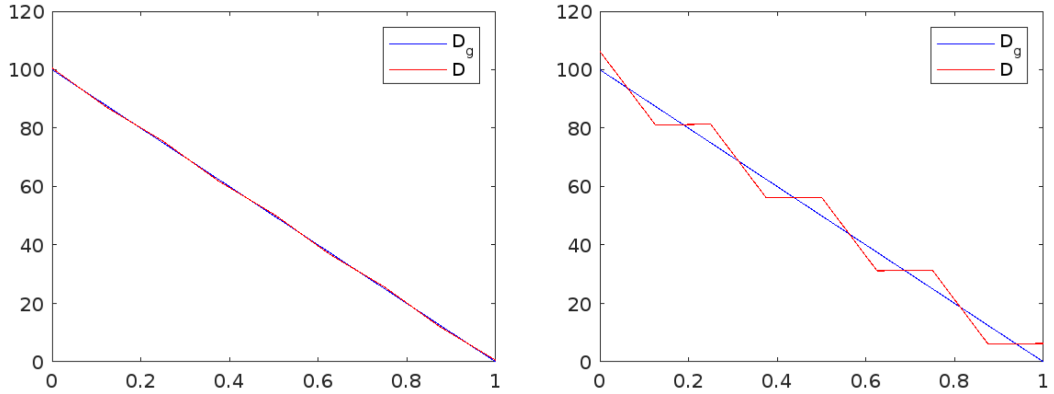
In figures 4.6, 4.7, 4.8 and 4.9, we plot D_g and D obtained using `sqp` algorithm (4.6a, 4.7a, 4.8a, 4.9a) versus `interior-point` algorithm (4.6b, 4.7b, 4.8b, 4.9b) for decreasing mesh sizes. Furthermore in the figure 4.10, we present the absolute error $\|D - D_g\|_2$ for every h.



(a) `sqp` algorithm

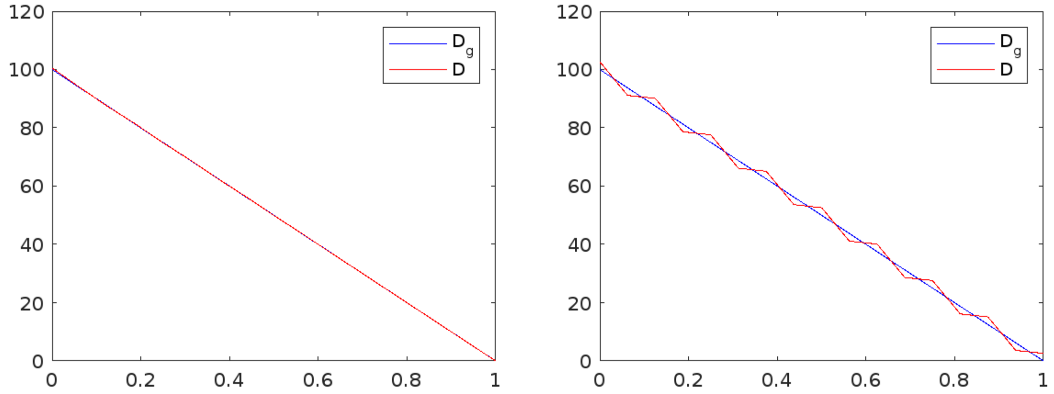
(b) `interior-point` algorithm

Figure 4.6: D and D_g for $h = \frac{1}{4}$ with D_0 =random vector and $T = 1$.



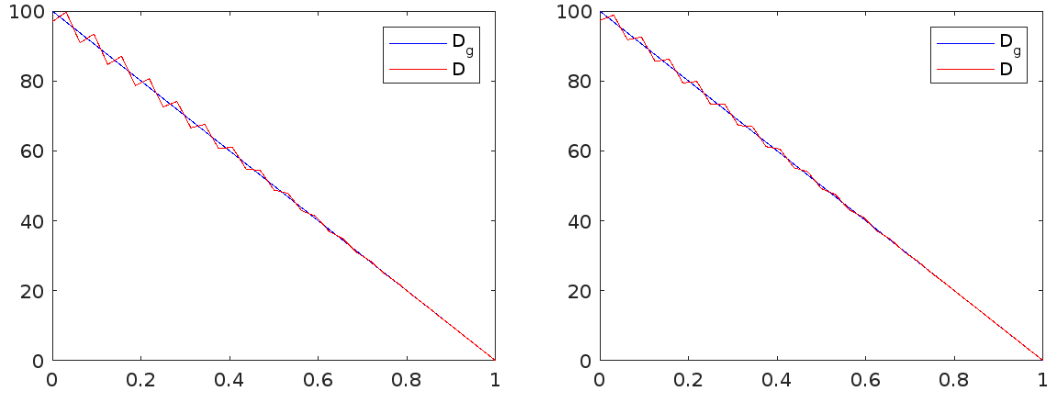
(a) `sqp` algorithm (b) `interior-point` algorithm

Figure 4.7: D and D_g for $h = \frac{1}{8}$ with D_0 =random vector and $T = 1$.



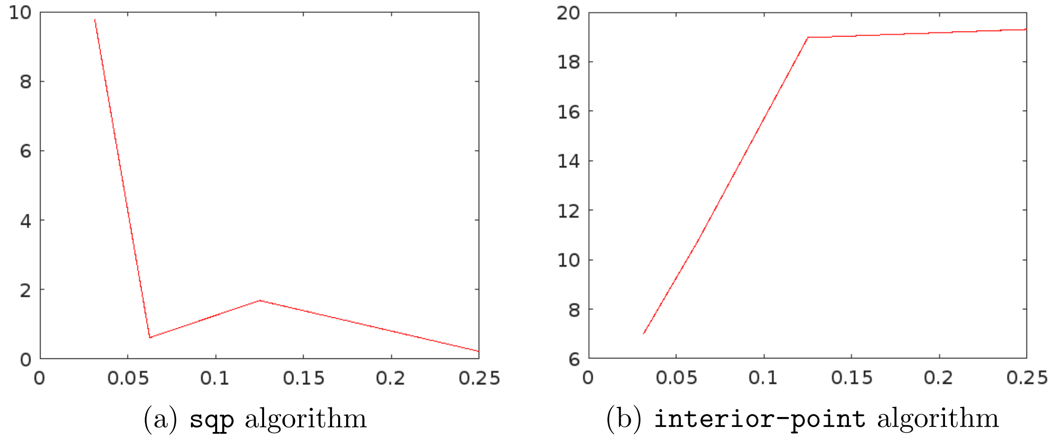
(a) `sqp` algorithm (b) `interior-point` algorithm

Figure 4.8: D and D_g for $h = \frac{1}{16}$ with D_0 =random vector and $T = 1$.



(a) `sqp` algorithm (b) `interior-point` algorithm

Figure 4.9: D and D_g for $h = \frac{1}{32}$ with D_0 =random vector and $T = 1$.



(a) sqp algorithm (b) interior-point algorithm

Figure 4.10: The absolute L_2 error: $\|D - D_g\|_2$ for all meshes for D_0 =random vector and $T = 1$.

Based on all the figures of this case, it's clear that when h is smaller, D_g and D coincide and the absolute error $\|D - D_g\|_2$ is smaller in `interior-point`. While in `sqp`, D_g and D coincide in the first 3 meshes and the absolute error decreases in general when h increases (there is a shift for $h = \frac{1}{8}$). Also, By comparing the figures 4.10a and 4.10b, we can observe that in the 3 meshes ($h = \frac{1}{4}, \frac{1}{8}, \frac{1}{16}$) the absolute error is the smallest when applying `sqp` while it's the smallest when applying `interior-point` in the remaining mesh ($h = \frac{1}{32}$). Noticing also that D and D_g are both decreasing functions in all figures.

4.3.3 Case 3: Initial Vector Guess $D_0 = 0$ and $T = 128$

In this test, we found our goal D by minimizing the objective function starting by the initial vector guess $D_0 = 0$ using the `MATLAB` function `fmincon` with an OptimalityTolerance $= 10^{-8}$ and by taking the end time $T = 128$.

In the table 4.3, we are presenting, for 3 different meshes, the following results: the number of iterations, the time needed to achieve the minimum using tic-toc functions, the relative L_2 error between D and D_g, and $V(D)$ the value of the objective function at D once using the `sqp` algorithm and once again using the `interior-point` algorithm.

h	sqp algorithm				interior-point algorithm			
	iter	time (/s)	$\frac{\|D-D_g\|_2}{\|D_g\|_2}$	$V(D)$	iter	time (/s)	$\frac{\|D-D_g\|_2}{\|D_g\|_2}$	$V(D)$
$\frac{1}{4}$	54	3.822339	0.000033	5.9323e-16	73	5.478392	0.2106	8.3478e-13
$\frac{1}{8}$	135	33.137444	0.000034	3.1353e-14	109	26.318065	0.1059	1.2162e-13
$\frac{1}{16}$	556	517.441185	0.085723	7.0749e-12	234	215.194905	0.0506	1.2001e-13

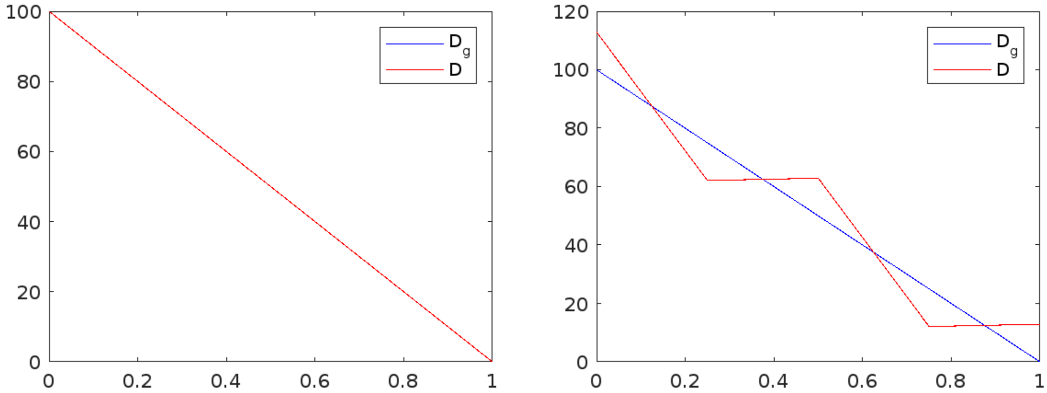
Table 4.3: Number of iterations (iter), time taken to find the minimum , relative error, and value of $V(D)$ for each mesh using the two algorithms sqp and interior-point with $D_0 = 0$ and $T = 128$.

The results in this table indicate:

- In sqp, the number of iterations increases for smaller h, while it remains moderate in interior-point. For $h = \frac{1}{4}$, the number of iterations is 1.3 times smaller in sqp than that in interior-point, and it's the other way around for $h = \frac{1}{8}$. In the case $h = \frac{1}{16}$, the number of iterations is 2 times greater in sqp than that in interior-point.
- In sqp and interior-point, the time needed to attend the minimum is increasing while h decreases.
- On the other hand, the relative error is decreasing in interior-point and increasing in sqp. And it's 6382 to 3115 times smaller for $h = \frac{1}{4}, \frac{1}{8}$, and 1.7 times greater for $h = \frac{1}{16}$ in sqp than that in interior-point.
- In both algorithms, the value of the objective function at D is very small and increases for h smaller.

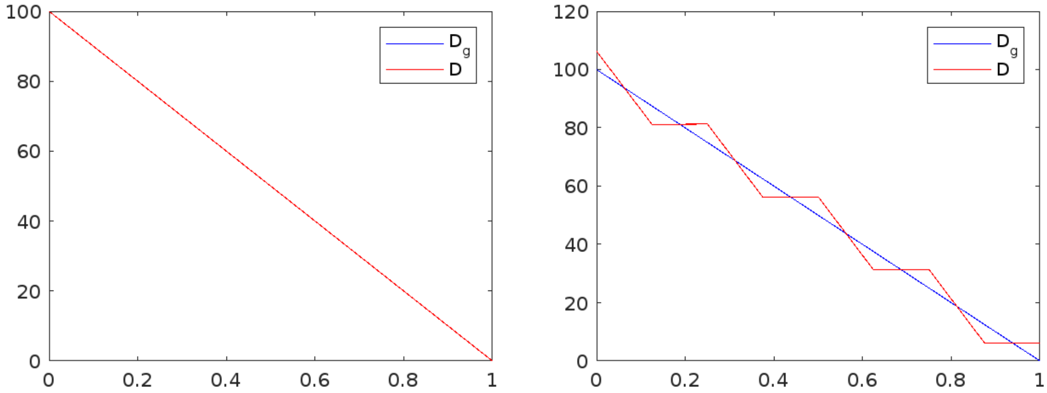
Based on the above observations and taking into consideration all of the following factors: time, relative error, and number of iteration, we can conclude that the interior-point method performs better for a moderately fine mesh ($h = \frac{1}{16}$) while the sqp method works much better for a coarse mesh ($h = \frac{1}{4}, \frac{1}{8}$).

In figures 4.11, 4.12 and 4.13, we plot D_g and D obtained using sqp algorithm (4.11a, 4.12a, 4.13a) versus interior-point algorithm (4.11b, 4.12b, 4.13b) for decreasing mesh sizes. Furthermore in the figure 4.14, we present the absolute error $\|D - D_g\|_2$ for every h.



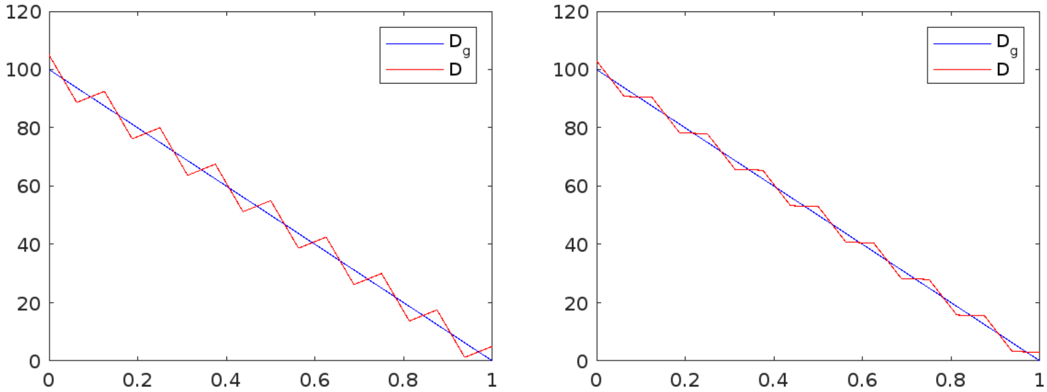
(a) `sqp` algorithm (b) `interior-point` algorithm

Figure 4.11: D and D_g for $h = \frac{1}{4}$ with $D_0 = 0$ and $T = 128$.



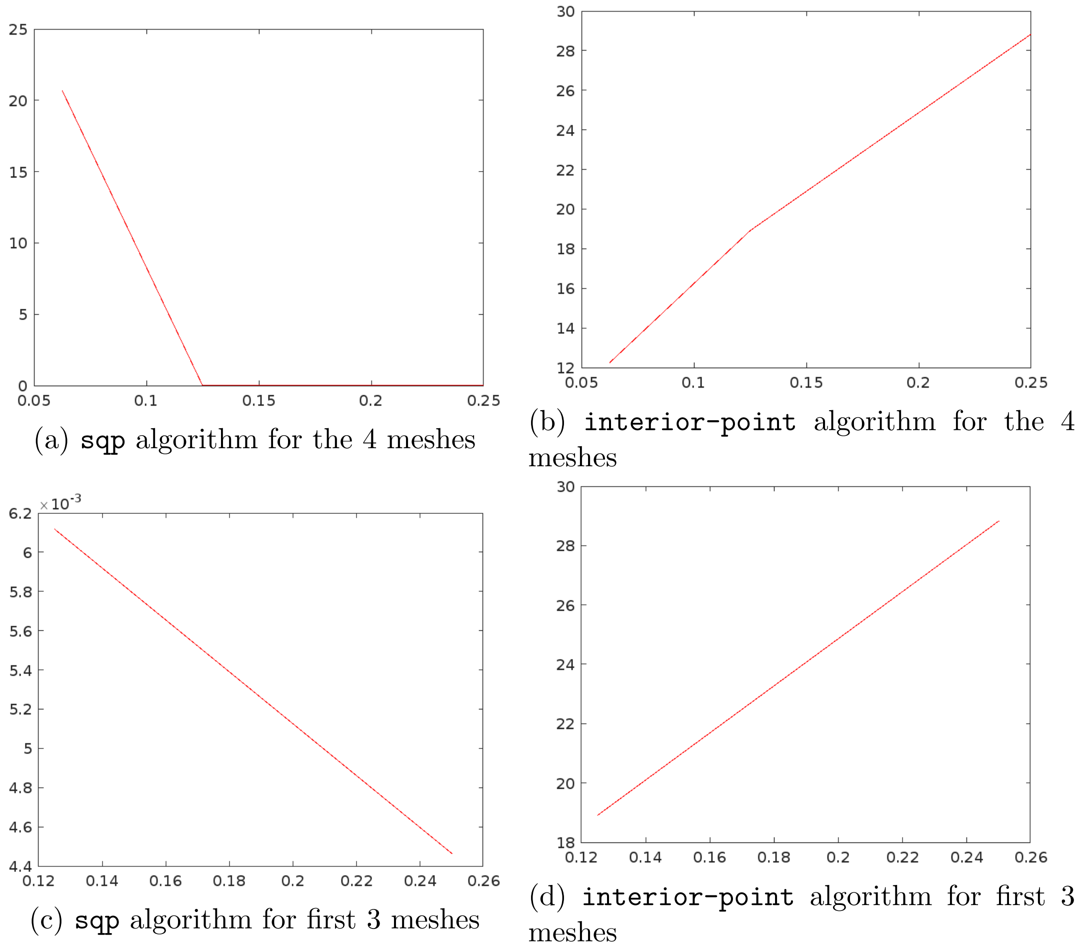
(a) `sqp` algorithm (b) `interior-point` algorithm

Figure 4.12: D and D_g for $h = \frac{1}{8}$ with $D_0 = 0$ and $T = 128$.



(a) `sqp` algorithm (b) `interior-point` algorithm

Figure 4.13: D and D_g for $h = \frac{1}{16}$ with $D_0 = 0$ and $T = 128$.



(a) `sqp` algorithm for the 4 meshes

(b) `interior-point` algorithm for the 4 meshes

(c) `sqp` algorithm for first 3 meshes

(d) `interior-point` algorithm for first 3 meshes

Figure 4.14: The absolute L_2 error: $\|D - D_g\|_2$ with $D_0 = 0$ and $T = 128$.

Based on all the figures of this case, it's clear that in the `interior-point` method, when h gets smaller, D_g and D are getting close to each other and the absolute error $\|D - D_g\|_2$ gets smaller. While in `sqp`, D_g and D coincide for the first 2 meshes and by looking to the figures 4.14a and 4.14c, the absolute error strongly decreases from $h = \frac{1}{16}$ to $h = \frac{1}{8}$ and continue decreasing moderately to $h = \frac{1}{4}$. Also, by respectively comparing figures 4.14a and 4.14c to 4.14b and 4.14d, we can observe that in the first 2 meshes the absolute error is 3100 to 6408 times smaller when applying `sqp` than that when applying `interior-point`, while it's 1.7 times smaller when applying `interior-point` than that when applying `sqp` in the last mesh. Noticing also that D and D_g are both decreasing functions in all figures.

4.3.4 Case 4: Initial Vector Guess D_0= Random and $T = 128$

In this test, we found our goal D by minimizing the objective function for end time $T = 128$ using the MATLAB function fmincon with an Optimality Tolerance $= 10^{-8}$ and a starting vector guess D_0 which is a random vector of length $\frac{1}{h} + 1$ with entries between 0 and 100 i.e D_0 =100*rand($\frac{1}{h}$ + 1,1).

In the table 4.4, we present, for 3 different meshes, the number of iterations and time needed to achieve the minimum, the relative L_2 error between D and D_g, and $V(D)$ the value of the objective function at D once using the sqp algorithm and once again using the interior-point algorithm.

h	sqp algorithm				interior-point algorithm			
	iter	time (/s)	$\frac{\|D - D_g\|_2}{\|D_g\|_2}$	$V(D)$	iter	time (/s)	$\frac{\|D - D_g\|_2}{\|D_g\|_2}$	$V(D)$
$\frac{1}{4}$	66	5.209564	0.000033	4.4347e-14	77	6.656901	0.2043	8.8150e-14
$\frac{1}{8}$	130	29.404124	0.007089	3.2351e-15	138	32.605684	0.0885	7.5593e-14
$\frac{1}{16}$	244	220.356993	0.084686	2.0513e-13	279	258.64994	0.0535	5.3287e-12

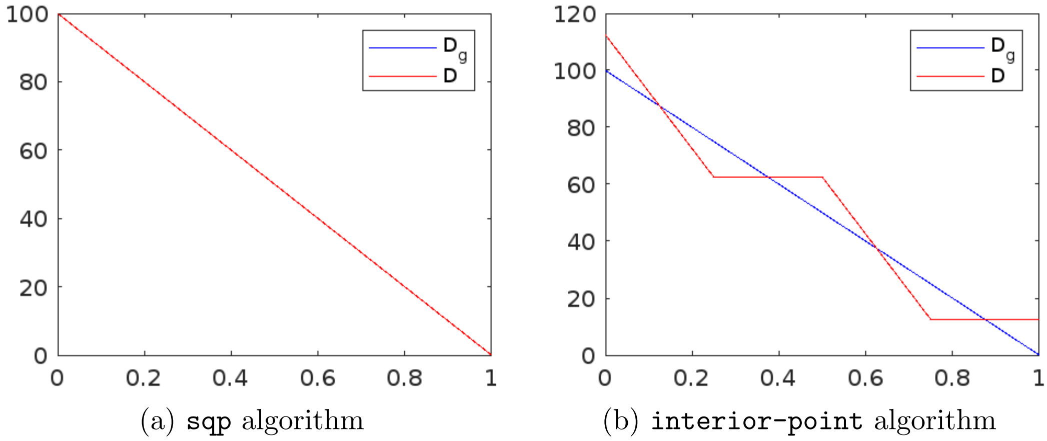
Table 4.4: Number of iterations (iter), time taken to find the minimum, relative error, and value of $V(D)$ for each mesh using the two algorithms sqp and interior-point with D_0= random and $T = 128$.

The results in this table indicate:

- The number of iterations increases for smaller h in both algorithms. Noticing that for every h, the difference in the number of iterations between sqp and interior-point does not exceed 35.
- In sqp and interior-point, the time needed to attend the minimum is increasing while h decreases.
- As for the relative error, while h decreases, it increases in sqp and decreases in interior-point. And it's 13 to 6191 times smaller for $h = \frac{1}{4}, \frac{1}{8}$ in sqp method than that in interior-point method, while the relative error for $h = \frac{1}{16}$ is 1.7 times smaller in interior-point than that in sqp.
- In both algorithms, the value of the objective function at D is very small and increases for h smaller.

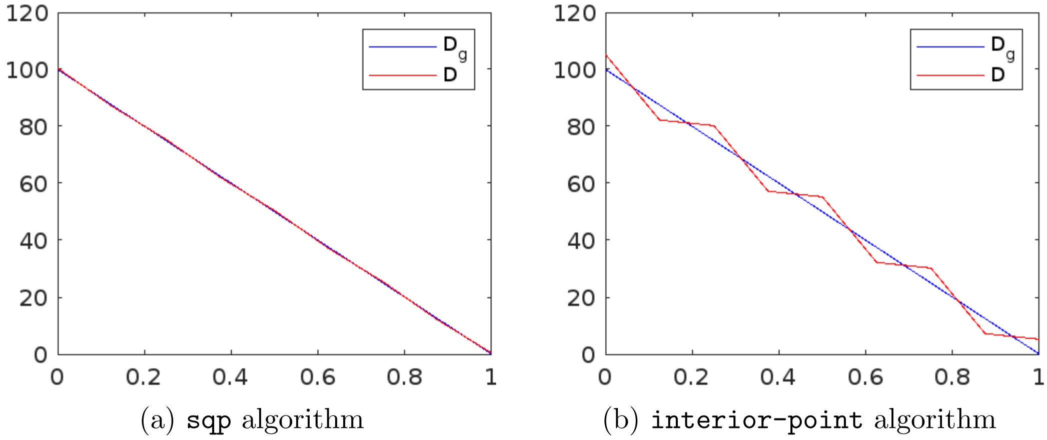
Based on the above observations and taking into consideration all of the following factors: time, relative error, and number of iteration, we can conclude that the `interior-point` method performs better for a moderately fine mesh while the `sqp` method performs much better for a coarse mesh.

In figures 4.15, 4.16 and 4.17, we plot D_g and D obtained using `sqp` algorithm (4.15a, 4.16a, 4.17a) versus `interior-point` algorithm (4.15b, 4.16b, 4.17b) for decreasing mesh sizes. Furthermore in the figure 4.18, we present the absolute error $\|D - D_g\|_2$ for every h.



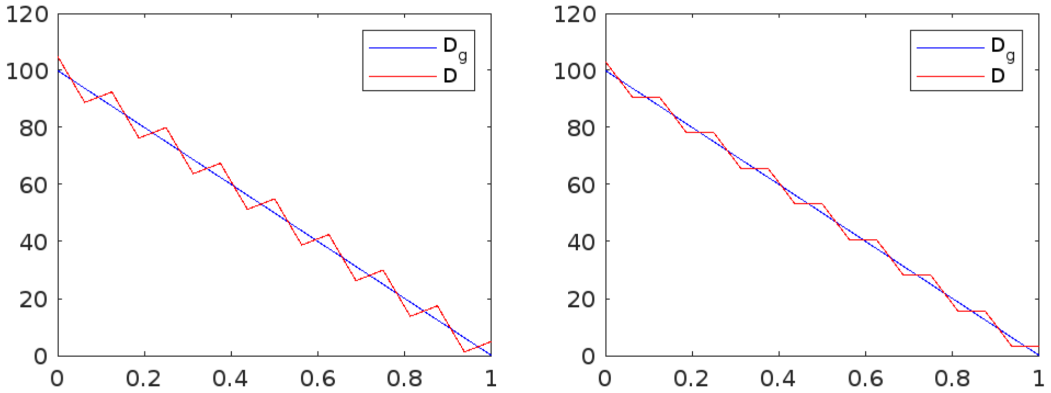
(a) `sqp` algorithm (b) `interior-point` algorithm

Figure 4.15: D and D_g for $h = \frac{1}{4}$ with D_0 =random vector and $T = 128$.



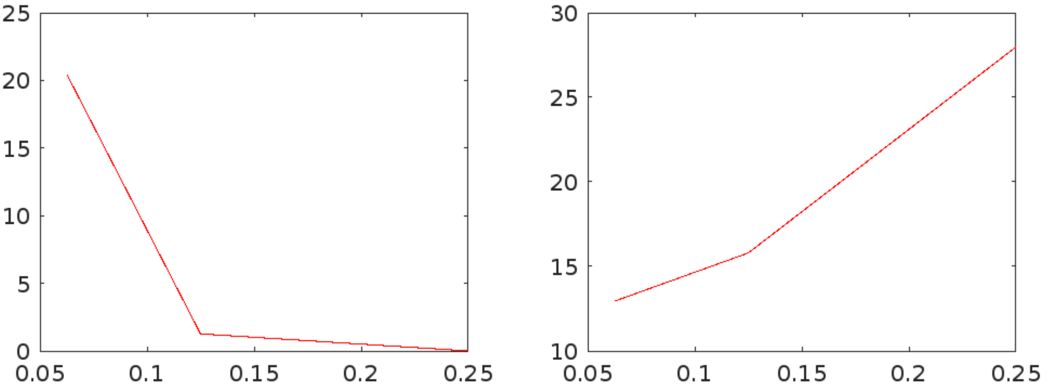
(a) `sqp` algorithm (b) `interior-point` algorithm

Figure 4.16: D and D_g for $h = \frac{1}{8}$ with D_0 =random vector and $T = 128$.



(a) `sqp` algorithm (b) `interior-point` algorithm

Figure 4.17: D and D_g for $h = \frac{1}{16}$ with D_0 =random vector and $T = 128$.



(a) `sqp` algorithm (b) `interior-point` algorithm

Figure 4.18: The absolute L_2 error: $\|D - D_g\|_2$ for all meshes for D_0 =random vector and $T = 128$.

Based on all the figures of this case, it's clear that when h is smaller, D_g and D are getting close to each other and the absolute error $\|D - D_g\|_2$ is smaller in `interior-point`. While in `sqp`, D_g and D coincide in the first 2 meshes and the absolute error decreases in general when h increases. Also, By comparing the figures 4.18a and 4.18b, we can observe that in the 2 meshes ($h = \frac{1}{4}, \frac{1}{8}$) the absolute error is the smallest when applying `sqp` while it's the smallest when applying `interior-point` in the remaining mesh ($h = \frac{1}{32}$). Noticing also that D and D_g are both decreasing functions in all figures.

4.3.5 Analysis of the Results

According to the observation of all the figures and tables, either starting by 0 or a random vector in the first 3 meshes for $T = 1$ and the first 2 meshes for

$T = 128$, we conclude that by an acceptable number of iterations, the `sqp` method achieves the best convergence in both relative and absolute norms comparing to the `interior-point` method.
Now by taking the last mesh in the cases 1, 3 and 4, the `interior-point` method gives a much better convergence in both norms than that in the `sqp` method and also uses an acceptable number of iterations. While by taking the last mesh in case 2, and by an acceptable number of iterations, the `interior-point` method gives a slightly better convergence in both norms than that in the `sqp` method. Furthermore, we conclude that the fastest algorithm in terms of time needed to find the minimum is `sqp` .

Hence, in both cases, one should apply for $h = \frac{1}{32}$ with $T = 1$ and for $h = \frac{1}{16}$ with $T = 128$, the `interior-point` algorithm to `fmincon` while apply the `sqp` algorithm to `fmincon` for a bigger h in the two end times.

Chapter 5

Conclusion

In this thesis, we studied the direct and the inverse problem of a time-dependent partial differential equation on one-space dimension $[0, z_F]$ with Dirichlet and mixed boundary condition.

First, we handled the theoretical and numerical study of the direct problem. In the theoretical part, we have reformulated the problem using a semi-variation method and shown the existence and uniqueness of its solution. As for the numerical part, we have discretized the problem in time and space and proved the existence and uniqueness of the obtained discrete system. Following that, we developed an efficient and fast solver using `MATLAB` for the direct problem and test it with generated data. This solver has given us convergence and while analyzing the results, we have concluded that taking $\Delta t = O(h)$ is the same as taking $\Delta t = O(h^2)$, noting that $\Delta t = O(h)$ will be the efficient choice.

Second, we have studied the numerical part of the inverse problem. For that, we have introduced `fmincon`, the `MATLAB` function, and two of its algorithms. Hence, we have used this function for the implementation that has recovered the diffusion coefficient $D_{CO2,air}$ and hence recovered all the D_α's. After the analysis of the obtained results, we have concluded:
For the 3 meshes $h = \frac{1}{4}, \frac{1}{8}, \frac{1}{16}$ with $T = 1$ and for the 2 meshes $h = \frac{1}{4}, \frac{1}{8}$ with $T = 128$, applying the algorithm `sqp` to `fmincon` gives the best performance in terms of time, relative and absolute norms, and number of iterations.
On the other hand and in terms of the same factors, for $h = \frac{1}{32}$ with $T = 1$, applying the `interior-point` method to `fmincon` yields slightly better performance than that in the other method when $D_0 =$ random vector and a much better one when $D_0 = 0$.
And for $h = \frac{1}{16}$ with $T = 128$, the best performance is obtained while applying the `interior-point` method to `fmincon` for both initial vector guess.
And in all the cases, we have seen that the obtained $D_{CO2,air}$ is a decreasing function and so are all the D_α's.

Future work may be another method of discretization in time for the direct problem like the Crank-Nicolson scheme. Also, it may be the theoretical study of the inverse problem and the existence and uniqueness of its solution.

Appendix A

MATLAB Codes

A.1 Function NODES

```matlab
% this function generates the vectors z, iB, t and the number m which takes
% as input:
% H: the uniform partition size
% a: first position in space
% b: last position in space
% dt: the partition for time
% T: the end time
% outputs:
% z: vector of the space points
% iB: is a boundary condition identifier, if index i correspond to Dirichlet
% boundary then iB(i)=1, otherwise iB(i)=0 needed to identify the knowns and
% unknows of the system
% m: number of meshing point for time
% t: vector of time points
function [z,iB,m,t]=NODES(H,a,b,dt,T)
t=0:dt:T;
m=length(t);
z=a:H:b;
iB=zeros(length(z),1);
iB(1)=1;
end
```

A.2 Function COEFFc

```
% this function generates the matrices M and K of the system which takes as
% input:
% h: vector of length n-1 with entries the distance between consecutifs $z_i$'s
%  (h(i-1)=z(i)-z(i-1))
% n: number of meshing points and n-1 intervals
% F: constant  needed in the generation of matrix K
% outputs:
% M and K: n*n sparse matrices (without taking into consideration that
% $z_1$ is given)
function [M,K]=COEFFc(h,n,F)
I=zeros(3*n-4,1);
J=zeros(3*n-4,1);
MIJ=zeros(3*n-4,1);
KIJ=zeros(3*n-4,1);
pt=0;
for i=2:n-1
    pt=pt+1;
    I(pt)=i; J(pt)=i-1;
    MIJ(pt)=h(i-1)/6; KIJ(pt)=-F/2;
    pt=pt+1;
    I(pt)=i; J(pt)=i;
    MIJ(pt)=(h(i-1)+h(i))/3;
    pt=pt+1;
    I(pt)=i; J(pt)=i+1;
    MIJ(pt)=h(i)/6; KIJ(pt)=F/2;
end
pt=pt+1;
I(pt)=n; J(pt)=n-1;
MIJ(pt)=h(n-1)/6; KIJ(pt)=-F/2;
pt=pt+1;
I(pt)=n; J(pt)=n;
MIJ(pt)=h(n-1)/3; KIJ(pt)=F/2;
M=sparse(I,J,MIJ);
K=sparse(I,J,KIJ);
end
```

A.3 Function COEFFv

```
% this function generates the matrices A and S of the system which takes as
% input:
% h: vector of length n-1 with entries the distance between consecutifs $z_i$'s
% (h(i-1)=z(i)-z(i-1))
% n: nomber of meshing points
% Da: vector of length n with entries diffusion coefficient Da(i) at
% position z(i)
% outputs:
% A and S: n*n sparse matrices (without taking into consideration that
% $z_1$ is given)
function [A,S]=COEFFv(h,n,Da)
I=zeros(3*n-4,1);
J=zeros(3*n-4,1);
SIJ=zeros(3*n-4,1);
AIJ=zeros(3*n-4,1);
pt=0;
for i=2:n-1
    pt=pt+1;
    I(pt)=i; J(pt)=i-1;
    AIJ(pt)=(-Da(i)-Da(i-1))/4;
    SIJ(pt)=(-Da(i)-Da(i-1))/(2*h(i-1));
    pt=pt+1;
    I(pt)=i; J(pt)=i;
    AIJ(pt)=(Da(i-1)-Da(i+1))/4;
    SIJ(pt)=  (Da(i-1)+Da(i))/(2*h(i-1))+(Da(i+1)+Da(i))/(2*h(i));
    pt=pt+1;
    I(pt)=i; J(pt)=i+1;
    AIJ(pt)=(Da(i)+Da(i+1))/4;
    SIJ(pt)=(-Da(i)-Da(i+1))/(2*h(i));
end
pt=pt+1;
I(pt)=n; J(pt)=n-1;
AIJ(pt)=(-Da(n-1)-Da(n))/4;SIJ(pt)=(-Da(n)-Da(n-1))/(2*h(n-1));
pt=pt+1;
I(pt)=n; J(pt)=n;
AIJ(pt)=(Da(n-1)+Da(n))/4;SIJ(pt)=(Da(n)+Da(n-1))/(2*h(n-1));
A=sparse(I,J,AIJ);
S=sparse(I,J,SIJ);
end
```

A.4 Function DirectPb

```
% this function generates the matrix V which takes as
% input:
% Da: vector of length n with entries diffusion coefficient Da(i)
% at position z(i)
% v0: vector of length m with entries is $\rho_\alpha^{atm}(j)$
% i.e value of $\rho(1,j)$
% M and K: n*n spareses matrices
% H: the uniform partition size
% a: first position in space
% b: last position in space
% dt: the partition for time
% T: the end time
% Gf, f1, Maf and F: constants
% outputs:
% V: n*m matrix with entries the concentration $\rho_\alpha^o$ at all
% positions of z(i) and t(j)
function[V]=DirectPb(Da,v0,M,K,H,a,b,dt,T,Gf,f1,Maf,F)
n=(b-a)/H+1;
[z,iB,m]=NODES(H,a,b,dt,T);
h=zeros(n-1,1);
h(1:n-1)=z(2:n)-z(1:n-1);
v3=1/6*Gf*z(2)-(Da(1)+Da(2))*(1/(2*z(2))*f1+1/4*Maf)-F/2;
v1=z(2)/6;
B=zeros(n,n); B(n,n)=F;
[A,S]=COEFFv(h,n,Da);
S=f1*S;
A=Maf*A;
VNK=find(iB==0);
V=zeros(n,m);
V(1,:)=v0;
V1=zeros(n-1,1);
V3=zeros(n-1,1);
AG=M+dt*(Gf*M+S-K-A+B);
AA=AG(VNK,VNK);
[L,U]=lu(AA);
 M=M(VNK,VNK);
for j=1:m-1
    V1(1,1)=(v0(j+1)-v0(j))*v1;
    V3(1,1)=(v0(j+1))*v3;
    RHS=M*V(VNK,j)-V1-dt*V3;
    V(VNK,j+1)=U\(L\RHS);
end
end
```

A.5 Function InversePb

```
% this function generates the scalar V which takes as input
% D: vector of length n with entries D(i) diffusion coefficient of gas
% CO2,air at position z(i)
% n: number of meshing points for space, m number of meshing point for time
% alpha: a vector of length l that denotes to a specific gases
% Ug: n*(l*m) matrix with entries the concentration $\rho_\alpha^o$ for the
% l gases given the diffusion coefficient Dag
% v0: vector of length m with entries is $\rho_\alpha^{atm}(j)$ i.e
% value of $\rho(1,j)$
% M and K: n*n spareses matrices
% H: the uniform partition size
% a: first position in space
% b: last position in space
% dt: the partition for time
% T: the end time
% Gf, f1, Maf, F and cf: constants
% outputs:
% V: the scalar which is the value of the objective function consedering
% the the concentration of the l gases at end time
function V = InversePb(D,n,m,alpha,Ug,v0,M,K,H,a,b,dt,T,Gf,f1,Maf,F,cf)
l=length(alpha);
Dac=zeros(n,l); % Dac: n*l matrix with entries diffusion coefficient computed
% using $D_{CO2,air}$ and Dac(:,j) represents entries of diffusion coefficient
% for alpha(j)
for j=1:l
    for i=1:n
        Dac(i,j)=alpha(j)*cf*D(i);
    end
end
Uc=zeros(n,l*m);% Uc: n*(l*m) matrix with entries the concentration $\rho_\alpha^o$
% for the l gases given the computed diffusion coefficient Dac
for j=1:l
    Uc(:,((j-1)*m+1):(j*m))=DirectPb(Dac(:,j),v0,M,K,H,a,b,dt,T,Gf,f1,Maf,F);
end
E=zeros(n,l); % E: n*l matrix with entries the error of the initial and computed
% concentration for the l gases at the end time T
for j=1:l
    E(:,j)=Ug(:,j*m)-Uc(:,j*m);
end
N=zeros(1,l); % N: vector of length l with entries the square of the $L_2$ norm
% of the error E of each gas
for j=1:l
    N(1,j)=norm(E(:,j)).^2;
end
V=sum(N); % summation of all entries of N over all the gases
end
```

A.6 Code for Section 3.4

```
%the constants:
f=0.2;f1=1/f;
Maf=f1*(0.04*9.8)/(8.314*260);
G=10+0.03;Gf=f1*G;
F=200+485;
% the domain space [a,b]:
a=0;b=1;
L=b-a;
T=1; % end time $T_1$
%T=32; % end time $T_2$
%T=128; % end time $T_3$
r=[1/8,1/16,1/32,1/64];
zc=(0:r(1):b)'; % common points between the 4 meshes
lc=length(zc);
e=length(r);
C=zeros(lc,1);
E1=zeros(lc,e);
E2=zeros(lc,e);
% for dt= order $h^2$: (section 3.4.1)
for o=1:e
    H=r(o); % H: is the uniform partition size
    dt=H.^2; % dt: is the partition for time
    n=L/H+1; % n: number of meshing points
    [z,iB,m,t]=NODES(H,a,b,dt,T);
    h=zeros(n-1,1);% h: vector of length n-1 with entries the distance between
    % consecutifs $z_i$'s needed in generating the four matrices A,S,M,K
    h(1:n-1)=z(2:n)-z(1:n-1);
    [M,K]=COEFFc(h,n,F); % generating matrices M and K
    v0=zeros(1,m); % v0: vector of length m with entries is $\rho_\alpha^{atm}(j)$
    % i.e value of $\rho(1,j)$
    for j=2:m
        v0(j)=2*(t(j)).^(1/4);
    end
    Da=zeros(n,1);% Da: vector of length n with entries diffusion coefficient
    % Da(i) at position z(i)
    f=@(z)((0.02-200)/b*z+200);
    for i=1:n
        Da(i)=f(z(i));
    end
    [U]=DirectPb(Da,v0,M,K,H,a,b,dt,T,Gf,f1,Maf,F); % generating the n*m matrix
    % U i.e the concentration $\rho_\alpha^o$  at all positions of z(i) and t(j)
    for i=1:lc
        C(i)=find(z==zc(i));
    end
    E1(:,o)=U(C,m);
end
% for dt= order h: (section 3.4.2)
```

```
for o=1:e
    H=r(o); % H: is the uniform partition size
    dt=H; % dt: is the partition for time
    n=L/H+1; % n: number of meshing points
    [z,iB,m,t]=NODES(H,a,b,dt,T);
    h=zeros(n-1,1);% h: vector of length n-1 with entries the distance between
    % consecutifs $z_i$'s needed in generating the four matrices A,S,M,K
    h(1:n-1)=z(2:n)-z(1:n-1);
    [M,K]=COEFFc(h,n,F); % generating matrices M and K
    v0=zeros(1,m); % v0: vector of length m with entries is $\rho_\alpha^{atm}(j)$
    % i.e value of $\rho(1,j)$
    for j=2:m
        v0(j)=2*(t(j)).^(1/4);
    end
    Da=zeros(n,1);% Da: vector of length n with entries diffusion coefficient
    % Da(i) at position z(i)
    f=@(z)((0.02-200)/b*z+200);
    for i=1:n
        Da(i)=f(z(i));
    end
    [U]=DirectPb(Da,v0,M,K,H,a,b,dt,T,Gf,f1,Maf,F); % generating the n*m matrix
    % U i.e the concentratio $\rho_\alpha^o$ at all positions of z(i) and t(j)
    for i=1:lc
        C(i)=find(z==zc(i));
    end
    E2(:,o)=U(C,m);
end
% the errors:
err1=zeros(4,e-1);
err2=zeros(4,e-1);
for j=1:e-1
    err1(1,j)=max(abs(E1(:,j)-E1(:,e)));
    err1(2,j)=max(abs(E1(:,j)-E1(:,e)))/max(abs(E1(:,e)));
    err1(3,j)=norm(E1(:,j)-E1(:,e));
    err1(4,j)=norm(E1(:,j)-E1(:,e))/norm(E1(:,e));
    err2(1,j)=max(abs(E2(:,j)-E2(:,e)));
    err2(2,j)=max(abs(E2(:,j)-E2(:,e)))/max(abs(E2(:,e)));
    err2(3,j)=norm(E2(:,j)-E2(:,e));
    err2(4,j)=norm(E2(:,j)-E2(:,e))/norm(E2(:,e));
end
```

A.7 Code for Section 4.3

```
%the constants:
f=0.2;f1=1/f;
Maf=f1*(0.04*9.8)/(8.314*260);
G=10+0.03;Gf=f1*G;
F=200+485;
% the domain space [a,b]:
a=0;b=1;
L=b-a;
T=1; % end time for sections 4.3.1 and 4.3.2
%T=128 % end time for sections 4.3.3 and 4.3.4
r=[1/4,1/8,1/16,1/32]; % meshes for sections 4.3.1 and 4.3.2
%r=[1/4,1/8,1/16]; % meshes for sections 4.3.3 and 4.3.4
e=length(r);
p=L/r(e)+1;
x0=100*rand(p,1);
options1 = optimoptions('fmincon', ...
    'Algorithm','sqp',...
    'TolFun',1e-08,...
    'MaxIter',10000,...
    'MaxFunEvals',300000); % input for fmincon
options2 = optimoptions('fmincon', ...
    'TolFun',1e-08,...
    'MaxIter',10000,...
    'MaxFunEvals',300000); % input for fmincon
err=zeros(1,2); % relative $L_2$ error between obtained D and the given Da
Err=zeros(2,e); % absolute $L_2$ error between obtained D and the given Da
for o=1:e
    H=r(o); % H: uniform partition size
    n=L/H+1; % n: number of meshing points
    dt=H; % dt: partition for time
    [z,iB,m,t]=NODES(H,a,b,dt,T);
    h=zeros(n-1,1);% h: vector of length n-1 with entries the distance between
    % consecutifs $z_i$'s needed in generating the four matrices A,S,M,K
    h(1:n-1)=z(2:n)-z(1:n-1);
    [M,K]=COEFFc(h,n,F); % generating matrices M and K
    v0=zeros(1,m); % v0: vector of length m with entries is $\rho_\alpha^{atm}(j)$
    % i.e value of $\rho(1,j)$
    for j=2:m
        v0(j)=2*(t(j)).^(1/4);
    end
    Da=zeros(n,1);% Da: vector of length n with entries diffusion coefficient
    % Da(i) at position z(i)
    f=@(z)((0.002-100)/b*z+100);
    for i=1:n
        Da(i)=f(z(i));
    end
    cf=0.5; %constant
    alpha=[1,2,3]; %the alpha gases
    l=length(alpha);
    Dag=zeros(n,l); % Dag: n*l matrix with entries given diffusion coefficient
    % Dag(i) at position z(i)
    for j=1:l
        for i=1:n
            Dag(i,j)=alpha(j)*cf*Da(i);
        end
    end
    Ug=zeros(n,l*m); % Ug: n*(l*m) matrix with entries the concentration
    % $\rho_\alpha^o$ for the l gases given the diffusion coefficient Dag
    for j=1:l
        Ug(:,((j-1)*m+1):(j*m))=DirectPb(Dag(:,j),v0,M,K,H,a,b,dt,T,Gf,f1,Maf,F);
    end
    % D0: initial $D_{CO2,air}$ which is a vector of length n
    D0=zeros(n,1); %for sections 4.3.1 and 4.3.3
    % D0=x0(1:(p-1)*h/L:p); %for sections 4.3.2 and 4.3.4
    AA=[]; bb=[]; Aeq=[]; beq=[];ub=[];nonlcon=[];% inputs for fmincon
    lb=zeros(n,1); % input for fmincon: lower bound for D(i) (D(i) $\geq$ 0)
    % finding $D_{CO2,air}$ by minimizing the objective function created "InversePb":
    tic
    [D1,fval1,exitflag1,output1]=fmincon(@(D)InversePb(D,n,m,alpha,Ug,v0,M,K,H, ...
        a,b,dt,T,Gf,f1,Maf,F,cf),D0,AA,bb,Aeq,beq,lb,ub,nonlcon,options1);
    toc
    tic
    [D2,fval2,exitflag2,output2]=fmincon(@(D)InversePb(D,n,m,alpha,Ug,v0,M,K,H, ...
        a,b,dt,T,Gf,f1,Maf,F,cf),D0,AA,bb,Aeq,beq,lb,ub,nonlcon,options2);
    toc
    D=zeros(n,2);
    D(:,1)=D1;D(:,2)=D2;
    for i=1:2
        Err(i,o)=norm(D(:,i)-Da);
        err(i)=norm(D(:,i)-Da)/norm(Da);
        figure();
        plot(z,Da,'b',z,D(:,i),'r')
        legend('D_g','D')
    end
end
figure();
plot(r,Err(1,:),'r')
figure();
plot(r(1:e-1),Err(1,1:e-1),'r')
figure();
plot(r,Err(2,:),'r')
figure();
plot(r(1:e-1),Err(2,1:e-1),'r')
```

Bibliography

[1] E. Witrant, P. Martinerie, C. Hogan, *et al.*, "A new multi-gas constrained model of trace gas non-homogeneous transport in firn: Evaluation and behaviour at eleven polar sites," *Atmospheric Chemistry and Physics*, vol. 12, pp. 11 465–11 483, 2011.

[2] L. Y. Yeung, L. T. Murray, P. Martinerie, *et al.*, "Isotopic constraint on the twentieth-century increase in tropospheric ozone," *Nature*, vol. 570, pp. 224–227, 2019.

[3] J. C. Laube, M. J. Newland, C. Hogan, *et al.*, "Newly detected ozone-depleting substances in the atmosphere," *Nature Geoscience*, vol. 7, pp. 266–269, 2014.

[4] D. Dahl-Jensen, M. R. Albert, A. Aldahan, *et al.*, "Eemian interglacial reconstructed from a greenland folded ice core," *Nature*, vol. 493, pp. 489–494, 2013.

[5] S. Moufawad, N. Nassif, and F. Triki, "Arxiv preprint arxiv:2207.07352," *Direct Problem of Gas Diffusion in Polar Firn*, 2022.

[6] H. Brezis, *Functional analysis, Sobolev spaces and partial differential equations*, ser. Universitext. Springer, New York, 2011, pp. xiv+599, ISBN: 978-0-387-70913-0.

[7] T. MathWorks, *Optimization Toolbox User's Guide(R2022a)*. 2022. [Online]. Available: https://www.mathworks.com/help/pdf_doc/optim/index.html.

[8] R. Sobot, "Function analysis," *Engineering Mathematics by Example*, 2021.

[9] E. Witrant and P. Martinerie, "A variational approach for optimal diffusivity identification in firns," *18th Mediterranean Conference on Control and Automation, MED'10*, pp. 892–897, 2010.

[10] F. Triki, "Coefficient identification in parabolic equations with final data," *arXiv: Analysis of PDEs*, 2020.

[11] J. J. Jang and J. K. Seo, "Detection of admittivity anomaly on high-contrast heterogeneous backgrounds using frequency difference eit," *Physiological Measurement*, vol. 36, pp. 1179–1192, 2015.

[12] P. Cochat, L. Vaucoret, and J. Sarles, "Et al," *Evidence Based Mental Health*, vol. 11, pp. 102–104, 2012.

[13] H. M. Ammari and F. Triki, "Identification of an inclusion in multifrequency electric impedance tomography," *Communications in Partial Differential Equations*, vol. 42, pp. 159–177, 2016.

[14] E. Witrant and P. Martinerie, "Input estimation from sparse measurements in lpv systems and isotopic ratios in polar firns," *IFAC Proceedings Volumes*, vol. 46, no. 2, pp. 659–664, 2013.

[15] D. Colton, R. E. Ewing, and W. Rundell, "Inverse problems in partial differential equations," 1990.

[16] H. M. Ammari, F. Triki, and C.-H. Tsou, "Numerical determination of anomalies in multifrequency electrical impedance tomography," *European Journal of Applied Mathematics*, vol. 30, pp. 481–504, 2018.

[17] E. Bonnetier, F. Triki, and C.-H. Tsou, "On the electro-sensing of weakly electric fish," *Journal of Mathematical Analysis and Applications*, vol. 464, pp. 280–303, 2018.