



IVAN FRANKO NATIONAL
UNIVERSITY OF LVIV



INTERMATHS



UNIVERSITY OF L'AQUILA

Double-Degree Master's Programme "InterMaths"

Applied and Interdisciplinary Mathematics

Master of Science
Applied Mathematics

IVAN FRANKO NATIONAL UNIVERSITY OF LVIV

Master of Science
Mathematical Engineering

UNIVERSITY OF L'AQUILA

Master's Thesis

Exploring Finite Element Method and Neural Networks for Dissipative Acoustic Problems

Supervisor

Candidate

Assoc. Prof. Mykhaylo
Shcherbatyy

Volodymyr Milchanovskyi

Student ID (UAQ): **279258**
Student ID (LVIV): **27210474C**

ACADEMIC YEAR 2022/2023

Abstract

The study of acoustics is crucial for engineering, physics, and related fields. It offers numerous research opportunities. This thesis explores the application of the finite element method and neural networks for dissipative acoustic problems.

The paper discusses the literature in this area and presents the mathematical formulation of the problem. This is followed by a description of the method of solving the system and used software.

This thesis also introduces an algorithm for simulating a contaminated environment. This algorithm is used to generate a dataset when solving a mathematical problem. After that, machine learning techniques are applied to this set of data.

The results showed that it is possible to classify data in the time domain with higher accuracy than in the frequency domain. Moreover, it can be noted that simple classifiers for this problem show better results than more complex neural networks. Still, neural networks can provide instant data analysis and decision-making capabilities for real-world examples.

Keywords: Finite Element Method, Dissipative Acoustics, Synthetic Data Generation, Data Analysis, Neural Networks

Contents

1	Introduction	5
1.1	Motivation	5
1.2	Objective	6
1.3	Thesis outline	7
2	Review of Literature	8
3	Mathematical Problem	10
3.1	Initial-boundary problem	10
3.2	Variational formulation	11
3.3	Energy balance equation	11
3.4	Variational formulation of stationary harmonic oscillations	13
4	Numerical Methods	14
4.1	Time domain problem	14
4.2	Frequency domain problem	15
4.3	FEniCSx	15
5	Numerical Simulations and Computations	17
5.1	One-dimensional problems	18
5.1.1	Propagation in Time Domain	18
5.1.2	Propagation in Frequency Domain	21
5.2	Two-dimensional problems	22
5.2.1	Propagation in Time Domain across square region	22

5.2.2	Propagation in Frequency Domain across square region	27
6	Contamination problem	32
6.1	Domain model	32
6.2	Model parameters	34
6.3	Dataset generation	39
7	Data Analysis and Neural Network Design	43
7.1	Time Domain dataset exploration and classification	43
7.2	Frequency Domain dataset exploration and classification	50
7.3	Prediction with neural networks	56
8	Conclusions	63
8.1	Summary	63
8.2	Future research directions	65
	References	66

Chapter 1

Introduction

1.1 Motivation

The study of acoustic problems is of significant importance for engineering, physics, and other related fields. Since the acoustics study covers the generation, propagation, and detection of elastic waves in different materials such as gases, liquids, and solids, it offers numerous research opportunities.

In many practical applications, acoustic waves interact with different dissipative materials. For example, in medical imaging, acoustic waves are used to produce images of the human body. In architectural acoustics, sound waves interact with the building materials, which leads to absorption and attenuation. In underwater acoustics, sound waves interact with the ocean environment, including the sea floor, marine life, and even water temperature. These interactions affect the propagation of sound. Therefore, it is used in sonar systems for object detection or navigation.

Such problems can be modeled using partial differential equations (PDE) and solved using numerical methods. The finite element method (FEM) is well-suited for modeling complex geometries and materials with spatially varying properties, making it ideal for modeling acoustic wave propagation in real-world scenarios. Another method is the boundary element method (BEM), which is

useful when modeling wave propagation in unbounded domains. For sure, these problems also can be solved using analytical methods providing closed-form solutions that are suitable for comparison and validation of numerical methods.

However, FEM can be computationally expensive and requires significant computational resources for solving large-scale problems. On the other hand, machine learning techniques, such as neural networks (NN), have shown great potential in solving complex problems in various fields.

Therefore, exploring the combination of FEM and NN for dissipative acoustic problems is an interesting research direction, which could provide efficient and accurate solutions for such problems.

1.2 Objective

The goal of this thesis is to explore the use of FEM and NN for dissipative acoustic problems. The study aims to develop numerical solutions using FEM for solving the mathematical formulation for the given problems. The computational process will be realized with the FEniCSx library, which provides a high-level interface for FEM simulations and allows for parallel computation using Message Passing Interface (MPI).

In order to guarantee precision and correctness, the numerical solutions will undergo a validation process. Upon confirmation, a synthetic dataset simulating acoustic wave propagation in both time and frequency domains will be generated. This artificial dataset will be used for training simple classifiers and neural networks with different architectures.

The overall goal is to evaluate the performance and effectiveness of neural networks in analyzing and interpreting acoustic data. By training classifiers on the simulated dataset, the thesis aims to evaluate the ability of neural networks to classify acoustic wave properties, detect anomalies, and provide insight into the dissipative acoustic problems under investigation.

1.3 Thesis outline

Before going into details, the following paragraphs provide a brief overview of this thesis and its structure:

Chapter 2 (Review of Literature) lists the relevant works related to this paper. It involves an examination of past surveys, emphasizing the principal discoveries, constraints, and lack of current studies.

Chapter 3 (Mathematical problem) outlines the mathematical problem of dissipative acoustic and provides the corresponding equation. Furthermore, this chapter includes the variational formulation for both time-dependent and steady-state harmonic oscillation problems, together with energy balance equations.

Chapter 4 (Numerical Methods) employs the Galerkin method to the previously defined variational formulation to deduce time and frequency domain problems.

Chapter 5 (Numerical Simulations and Computations) presents examples of solved problems using numerical methods. The studies in this unit cover two media, such as water and fuel oil, along with one- and two-dimensional cases.

Chapter 6 (Contamination Problem) describes the concept of fluid pollution with air bubbles. It is followed by a synthetic data generation method and its implementation.

Chapter 7 (Data Analysis and Neural Network Design) focuses on the analysis of the dataset obtained in the previous chapter. This section describes the implementation of both simple classifiers and neural networks, followed by their further analysis.

Chapter 8 (Conclusions) summarizes the results obtained in this paper and suggests directions for future research.

Chapter 2

Review of Literature

The research in this thesis covers a variety of theoretical and practical matters, such as Acoustic Problems, Finite Element Method (FEM), Neural Networks (NN), as well as Parallel and Distributed Computing. Given the breadth of these topics, it is crucial to provide a thorough literature overview for each relevant part of the research.

Lawrence E. Kinsler, in his book [1], provides a detailed explanation of the principles and concepts of acoustics. The book covers a wide range of topics, including wave equation, wave nature, propagation, reflection, underwater acoustic and sound interaction with different materials and environments.

Moving closer to the specifics of this work, the equations of wave motion in "ideal" and viscous fluid were studied in the book by Landau and Lifshitz [2]. Deducing on the time-domain displacement formulation and energy displacement can be found in the paper [3]. The author also proposes two different time implicit discretization schemes and their order of convergence.

In the paper [4], the investigation is based on previous works and considers the interaction of the acoustic fluid with an elastic solid in the presence of a viscous medium. In addition to the time domain problem, the author presents problems of harmonic oscillations and natural vibration.

Another example can be found in [5]. It describes the motion of the

viscous heat-conducting Newtonian fluid for the time domain. This paper states the variational formulation of the given problem that can be used for obtaining the solution. It would be good to consider also the presented energy equations and proof of the existence of the solution.

For these types of equations, for example, in [6], authors use the Petrov-Galerkin method [7] with the one-step recurrent scheme of Cauchy integration over time.

There are several programming software options that can be used to solve acoustic wave propagation problems using the Finite Element Method. These include *COMSOL Multiphysics*, *OpenFOAM*, *MATLAB*, and others [8, 9, 10]. Each of them has its advantages and disadvantages, such as the price in case of a commercial product or the inability to extend its functionality. Given the above, the optimal option could be to use the *FEniCSx* library, the capabilities of which can be found in [11].

Among the literature on the combination of acoustic waves and neural networks, it is possible to highlight the work done in [12]. The author describes the application of Support Vector Machines (SVM) models for problems with nonlinear viscous hydrodynamic loads.

Considering that the nature of the problem corresponds to wave motions, it is possible to use the achievements made in other fields, for example, in [13].

There also appears to be a lack of research papers specifically describing the usage of neural networks for detecting contamination with acoustic waves. While neural networks and machine learning techniques have been widely employed in various fields of acoustics, including speech recognition and audio classification, the application of neural networks for contamination detection seems to be relatively underexplored.

Chapter 3

Mathematical Problem

This chapter focuses on the mathematical formulation of the dissipative acoustic problem in viscous fluids discussed in [14, 15]. This part is crucial for the research since it provides a rigorous model for describing natural phenomena. Moreover, this also allows the derivation of further logical conclusions based on specific assumptions while evaluating the validity of the hypotheses.

3.1 Initial-boundary problem

Let fluid occupy a limited area Ω of points $x = \{x_j\}_{j=1}^d, d = 1, 2, \text{ or } 3$, and let area Ω has Lipschitz-continuous boundary Γ . The motion of the medium is described by the displacement vector $u(x, t) = \{u_i(x, t)\}_{i=1}^d$ for all $(x, t) \in \Omega \times [0, T]$ and satisfies the following equation

$$\begin{cases} \rho u_i'' - \sigma_{ij,j} = 0, \\ \sigma_{ij}(u) = \alpha_{ijkl} e_{kl}(u) + c_{ijkl} e_{kl}(u'), \\ e_{ij}(u) = (u_{i,j} + u_{j,i})/2, \text{ in } \Omega \times (0, T] \end{cases} \quad (3.1)$$

with the boundary conditions

$$\begin{cases} u_j = 0 & \text{on } \Gamma_u \times [0, T], \Gamma_u \subset \Gamma, \text{mes}(\Gamma_u) > 0 \\ \sigma_{ij} \hat{n}_j = \hat{\sigma}_i & \text{on } \Gamma_\sigma \times [0, T], \Gamma_\sigma = \Gamma \setminus \Gamma_u \end{cases} \quad (3.2)$$

and initial conditions

$$\begin{cases} u|_{t=0} = u_0 \\ u'|_{t=0} = v_0, \end{cases} \quad (3.3)$$

where ρ - density, a_{ijklm} - elasticity, c_{ijklm} - viscosity, $u_0, v_0, \hat{\sigma}_i$ - given functions and $\hat{n} = \{n_i\}_{i=1}^d$ is a unit vector of the outer normal to the boundary Γ .

3.2 Variational formulation

Consider the space $V \doteq \{v \in [H^1(\Omega)]^d : v = 0 \text{ on } \Gamma_u\}$ and $H \doteq [L^2(\Omega)]^d$ to formulate the variational problem for (3.1)-(3.3).

Let $u_0 \in V$, $v_0 \in H$ and $v \in V'$, find $\mathbf{u} \in L^2(0, T; V) \cap L^\infty(0, T; H)$ such that

$$\begin{cases} m(u''(t), v) + c(u'(t), v) + a(u(t), v) = \langle l(t), v \rangle \quad \forall t \in (0, T] \\ a(u(0) - u_0, v) = 0, \\ m(u'(0) - v_0, v) = 0 \quad \forall v \in V, \end{cases} \quad (3.4)$$

where bilinear forms and linear functional have the following form

$$\begin{cases} m(u, v) = \int_{\Omega} \rho uv \, dx \\ c(u, v) = \int_{\Omega} c_{ijklm} e_{ij}(u) e_{ij}(v) \, dx \\ a(u, v) = \int_{\Omega} a_{ijklm} e_{ij}(u) e_{ij}(v) \, dx \\ \langle l, v \rangle = \int_{\Gamma_\sigma} \hat{\sigma}_i v_i \, d\gamma \end{cases} \quad (3.5)$$

and linear continuous functional l represents the external source of mechanical energy.

3.3 Energy balance equation

To formulate the energy balance equation for the variational problem (3.4), it is necessary to introduce notations for spaces and norms. Thus by following

the convention established in [5]

$$H \doteq L^2(\Omega), \quad H \doteq H^d, \quad H(\text{div}; \Omega) : \{v \in H : \nabla \cdot v \in H\} \quad (3.6)$$

Given the assumptions made about the spaces, boundary Γ , and the continuous and symmetric nature of the bilinear forms, the following norms can be written as follows

$$\begin{aligned} \|v\|_H &\doteq \sqrt{m(v, v)} \quad \forall v \in H, \\ \|v\|_V &\doteq \sqrt{a(v, v)} \quad \forall v \in V (\text{equivalent } \|\cdot\|_{[H^1(\Omega)]^d}) \\ |v|_V &\doteq \sqrt{c(v, v)} \quad \forall v \in V \end{aligned} \quad (3.7)$$

Now given norms can be applied to the variational problem (3.4)

$$\frac{1}{2} \frac{d}{dt} \left[\|u'(t)\|_H^2 + |u(t)|_V^2 \right] + \|u'(t)\|_V^2 = \langle l(t), u'(t) \rangle, \quad \forall t \in (0, T] \quad (3.8)$$

Then by integrating over the arbitrary time interval $[0, t], 0 \leq t \leq T$ equation (3.8) can be rewritten as follows

$$\begin{aligned} \frac{1}{2} \left[\|u'(t)\|_H^2 + |u(t)|_V^2 \right] + \int_0^t \|u'(\tau)\|_V^2 d\tau &= \\ = \frac{1}{2} \left[\|u'(0)\|_H^2 + |u(0)|_V^2 \right] + \int_0^t \langle l(\tau), u'(\tau) \rangle d\tau, \quad \forall t \in [0, T] \end{aligned} \quad (3.9)$$

According to (3.9), consecutive energy expression can be written as

$$\begin{aligned} K_S [u'(t)] + P_S [u(t)] + \int_0^t D_S [u(\tau)] d\tau &= \\ = K_S [v_0] + P_S [u_0] + Q_S [u'(t)], \quad \forall t \in [0, T] \end{aligned} \quad (3.10)$$

where

$$\begin{aligned} K_S [u'(t)] &\doteq \frac{1}{2} \|u'(t)\|_H^2 - \text{kinetic energy}, \\ P_S [u(t)] &\doteq \frac{1}{2} |u(t)|_V^2 - \text{potential energy}, \\ D_S [u(t)] &\doteq \|u'(t)\|_V^2 - \text{dissipation intensity}, \\ Q_S [u'(t)] &\doteq \int_0^t \langle l(\tau), u'(\tau) \rangle d\tau - \text{external energy} \end{aligned} \quad (3.11)$$

caused by the kinetic motion of the fluid.

3.4 Variational formulation of stationary harmonic oscillations

Suppose that the oscillations in the system are caused by boundary conditions Γ_σ , that vary in time according to a harmonic law with a given circular frequency $\omega > 0$ such that, the harmonic load for the equation (3.4) can be written as

$$\hat{\sigma} = \sigma_1 + i\sigma_2 \quad (3.12)$$

Therefore, it is plausible to assume that after a certain prolonged time period, the influence of the initial conditions will be eliminated, and oscillations in the system will occur according to the same harmonic law

$$u(x, t) = u^*(x)e^{i\omega t} \quad (3.13)$$

where

$$u^*(x) = u_1(x) + iu_2(x) \quad (3.14)$$

Using (3.12) and (3.13), (3.4) can be formulated for the problem of steady-state harmonic oscillations.

Let vectors $l_1, l_2 \in L^2(V')$, $0 < \omega = 2\pi \cdot f = \text{const}$, find $u_1, u_2 \in L^2(V)$ such that

$$\begin{cases} -\omega^2 m(u_1, v_1) - \omega c(u_2, v_1) + a(u_1, v_1) = \langle l_1, v_1 \rangle & \forall v_1 \in V \\ -\omega^2 m(u_2, v_2) + \omega c(u_1, v_2) + a(u_2, v_2) = \langle l_2, v_2 \rangle & \forall v_2 \in V \end{cases} \quad (3.15)$$

where bilinear forms and linear functional are defined by (3.5).

By combining expressions in (3.15) it is possible to write balance equations as follows

$$\begin{cases} -\omega^2 \left[\|u_1\|_V^2 + \|u_2\|_V^2 \right] + \left[\|u_1\|_H^2 + \|u_2\|_H^2 \right] & = \langle l_1, v_1 \rangle + \langle l_2, v_2 \rangle \\ \omega \left[|u_1|_V^2 + |u_2|_V^2 \right] & = \langle l_1, v_1 \rangle - \langle l_2, v_2 \rangle \end{cases} \quad (3.16)$$

Chapter 4

Numerical Methods

4.1 Time domain problem

The following problem was examined in [14]. Hence, this paper only presents the final formulation that shall be used for further analysis.

Let $U^0, \dot{U}^0, \Delta t > 0, \gamma, \beta \in [0, 1]$, find $\ddot{U}^{j+1/2}$ and \dot{U}^{j+1}, U^{j+1} such that

$$\left\{ \begin{array}{l} \left[M + \Delta t \gamma C + \frac{1}{2} \Delta t^2 \beta A \right] \ddot{U}^{j+1/2} = L(t_{j+1/2}) - \\ \quad - C \dot{U}^j - A \left[U^j + \Delta t \gamma \dot{U}^{j+1} \right] \\ \dot{U}^{j+1} = U^j + \Delta t \ddot{U}^{j+1/2} \\ U^{j+1} = U^j + \frac{1}{2} \Delta (\dot{U}^j + \dot{U}^{j+1}) \\ j = 0, 1, \dots \text{ while } j \Delta t \leq T, \end{array} \right. \quad (4.1)$$

where the matrices M, A, C, and vector L are calculated by corresponding bilinear forms and linear functional defined in (3.5) as follows

$$\left\{ \begin{array}{l} M = \{m(\varphi_i, \varphi_k)\}_{i,k=0}^N \\ A = \{a(\varphi_i, \varphi_k)\}_{i,k=0}^N \\ C = \{c(\varphi_i, \varphi_k)\}_{i,k=0}^N \\ L = \{l(\varphi_i)\}_{i=0}^N \end{array} \right. \quad (4.2)$$

It should be noted that selecting the time integration step Δt , the Courant-Friedrichs-Lewy condition $\Delta t \leq \frac{h}{\alpha c}$ need to be considered, where h - mesh diameter, c - speed of sound, $\alpha > 1$ - parameter, the selection of which could depend, in particular, on the smoothness of the basis functions [16].

4.2 Frequency domain problem

A similar discussion on the problem of steady-state harmonic oscillations is described in 3.4 and was considered in [15].

Given vectors $L_1, L_2 \in H$, $0 < \omega = 2\pi \cdot f = const$, find $U_1, U_2 \in V$ such that

$$\begin{bmatrix} A - \omega^2 M & -\omega C \\ \omega C^T & A - \omega^2 M \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} = \begin{bmatrix} L_1 \\ L_2 \end{bmatrix} \quad (4.3)$$

where the matrices M , A , C , and vector L are calculated by complete analogy with (4.2).

4.3 FEniCSx

For solving problems described above with the Finite Element Method FEniCSx library was used. FEniCSx is a new version of the popular open-source library FEniCS for solving partial differential equations (PDEs). Utilizing it, it is possible to easily transfer the variational formulation of the problem into an efficient programming code using high-level Python or C++ interfaces [11, 17].

The library's design follows the principles of compactness, modularity, and extensibility, allowing users to easily integrate various solvers like PETSc, efficient mesh generation, and just-in-time compilation to enhance code performance. Additionally, the library offers the possibility of parallel execution on both a single computer and high-performance computing (HPC) systems.

To achieve precision and efficiency in computations, it is crucial to consider the tools and techniques available for addressing complex problems. Re-

search on PETSc solvers and preconditioners [18] has shown to be particularly valuable in this respect. This application can help speed up iterative methods and reduce the computational expenses for given problems.

Chapter 5

Numerical Simulations and Computations

This chapter focuses on the discussion of acoustic wave propagation in two physical domains: water and fuel oil. Both domains have unique physical characteristics that influence wave propagation.

Water

- Density: $\rho = 1000 \text{ kg/m}^3$,
- Speed of sound: $c = 1500 \text{ m/s}$
- Viscosity: $\eta = 8.94 \cdot 10^{-4} \text{ Pa} \cdot \text{s}$

Fuel oil

- Density: $\rho = 890 \text{ kg/m}^3$,
- Speed of sound: $c = 1360 \text{ m/s}$
- Viscosity: $\eta = 2.022 \text{ Pa} \cdot \text{s}$

The analysis of convergence for different mesh sizes and time integration steps can be found in [19, 20].

5.1 One-dimensional problems

Consider a line segment with a size $[0, 2.5 \text{ m}]$ and $N = 10^3$ finite elements. The domain has rigid right wall, and the pressure $p = 10^3 \text{ Pa}$ is applied to the left boundary.

5.1.1 Propagation in Time Domain

For the problem described in Section 4.1 pressure is applied for $p_t = 7 \cdot 10^{-4} \text{ s}$ with the total time $T = 3.5 \cdot 10^{-3} \text{ s}$ and time integration step $\Delta t = 2 \cdot 10^{-6} \text{ s}$.

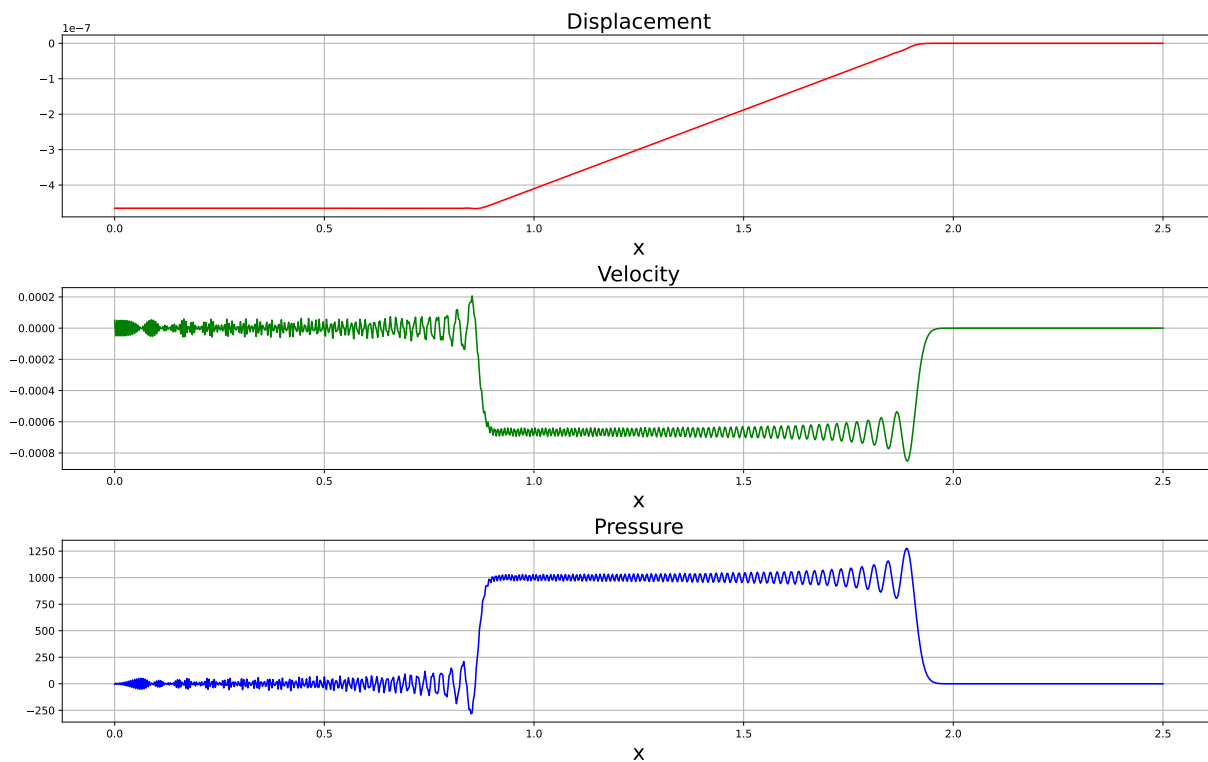


Figure 5.1: Water, $t = 1.282 \cdot 10^{-3} \text{ s}$.

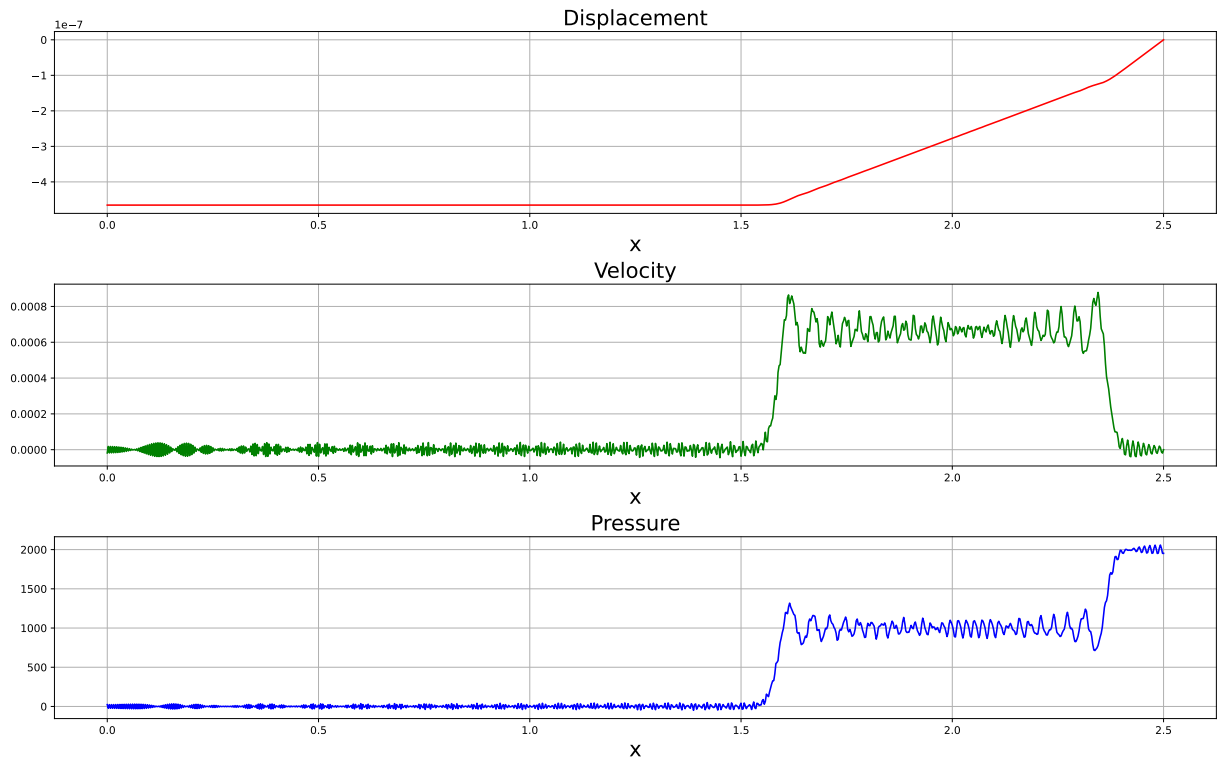


Figure 5.2: Water, $t = 2.282 \cdot 10^{-3}$ s.

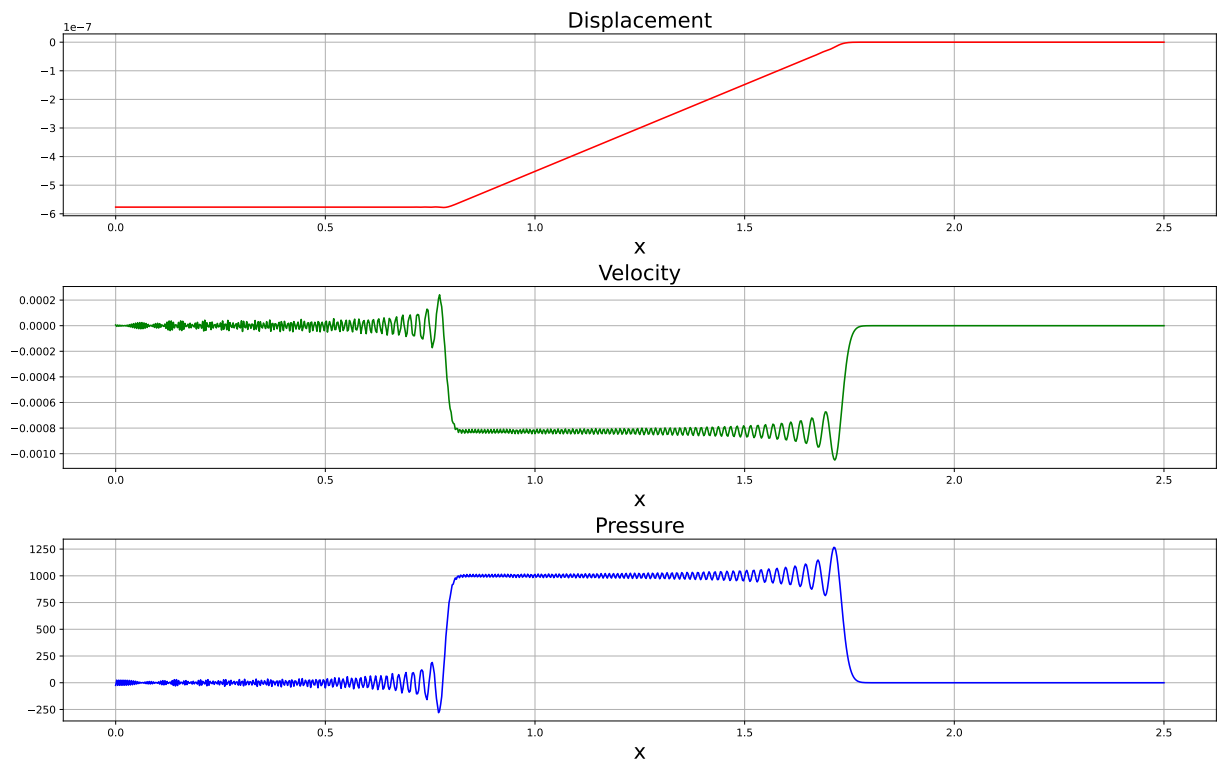


Figure 5.3: Fuel oil, $t = 1.282 \cdot 10^{-3}$ s.

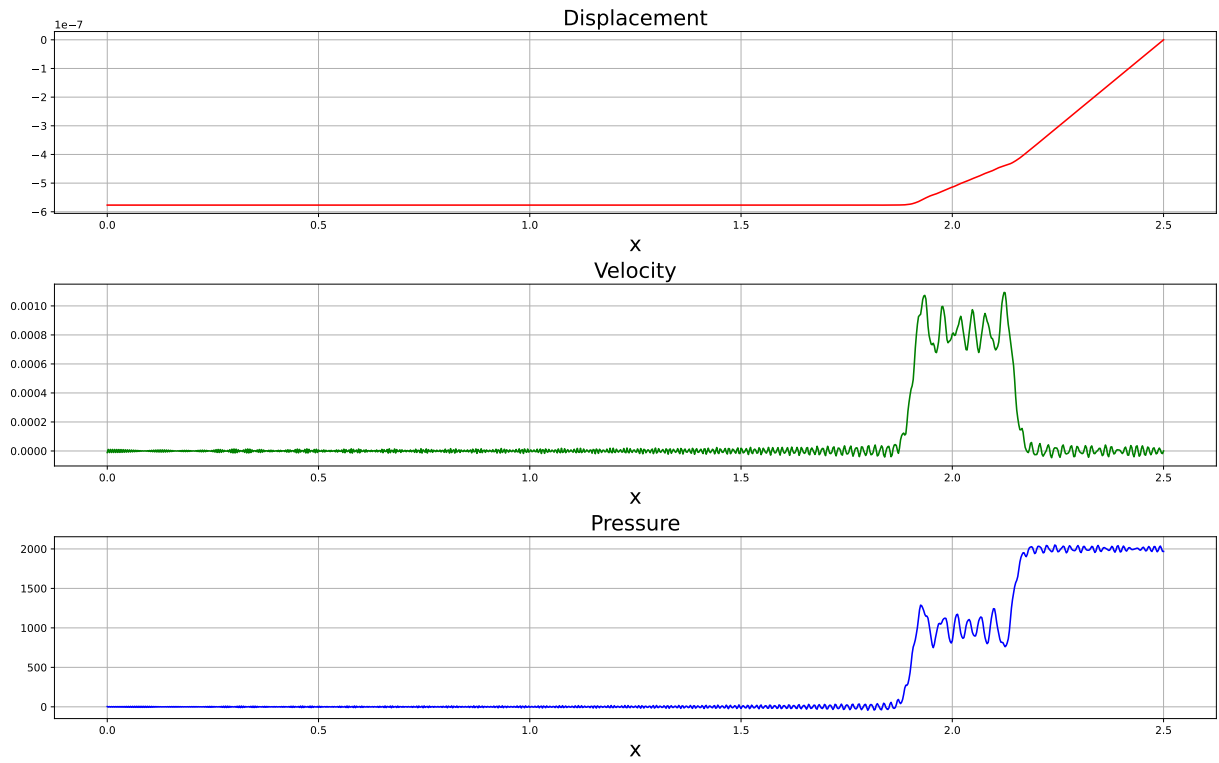


Figure 5.4: Fuel oil, $t = 2.282 \cdot 10^{-3}$ s.

Figure 5.5 show the change in energy for the system filled with water and calculated by the equation (3.10) for extended period of time $\mathbf{T} = 7 \cdot 10^{-3}$ s. The plot illustrates the exchange of potential and kinetic energies and the increase in dissipation over time.

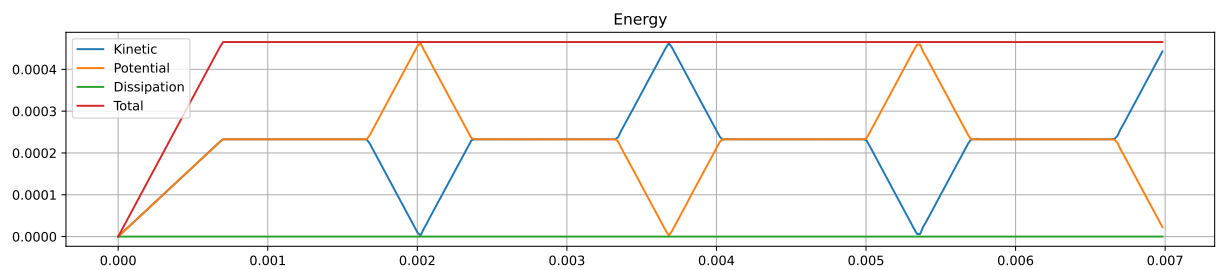


Figure 5.5: Change of energy in water.

5.1.2 Propagation in Frequency Domain

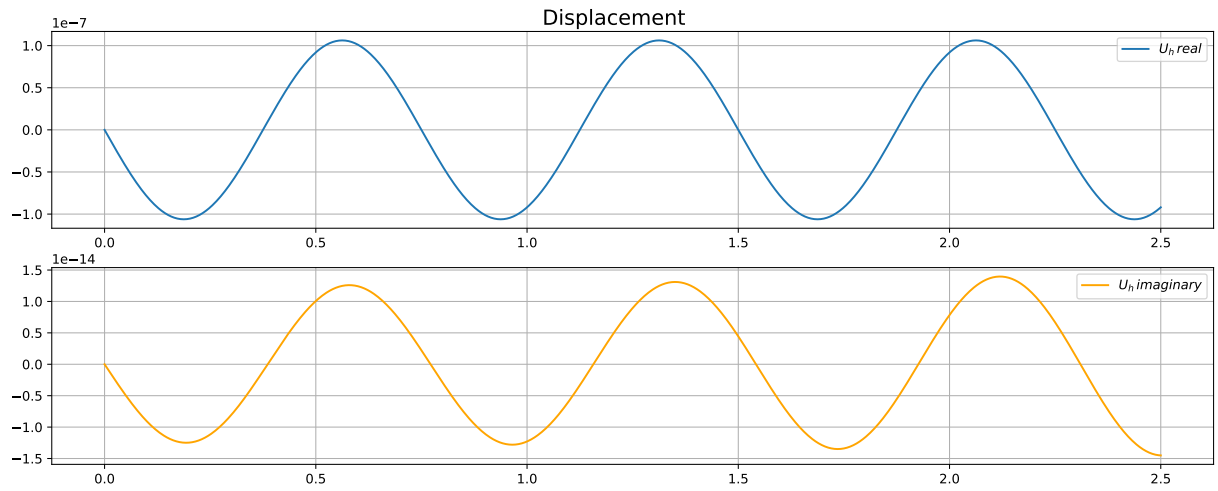


Figure 5.6: Water, $f = 2 \cdot 10^3$ Hz.

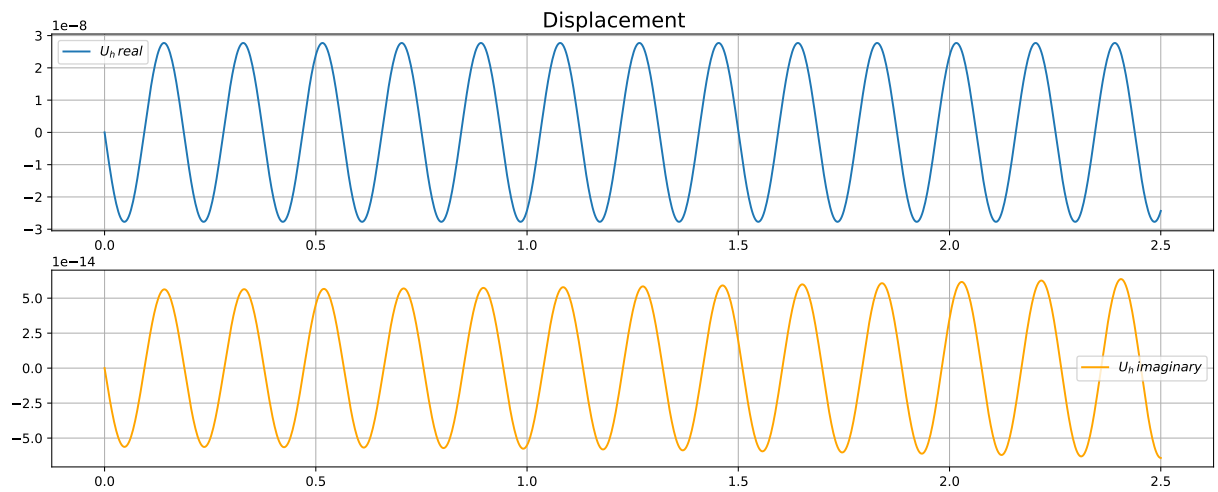


Figure 5.7: Water, $f = 8 \cdot 10^3$ Hz.

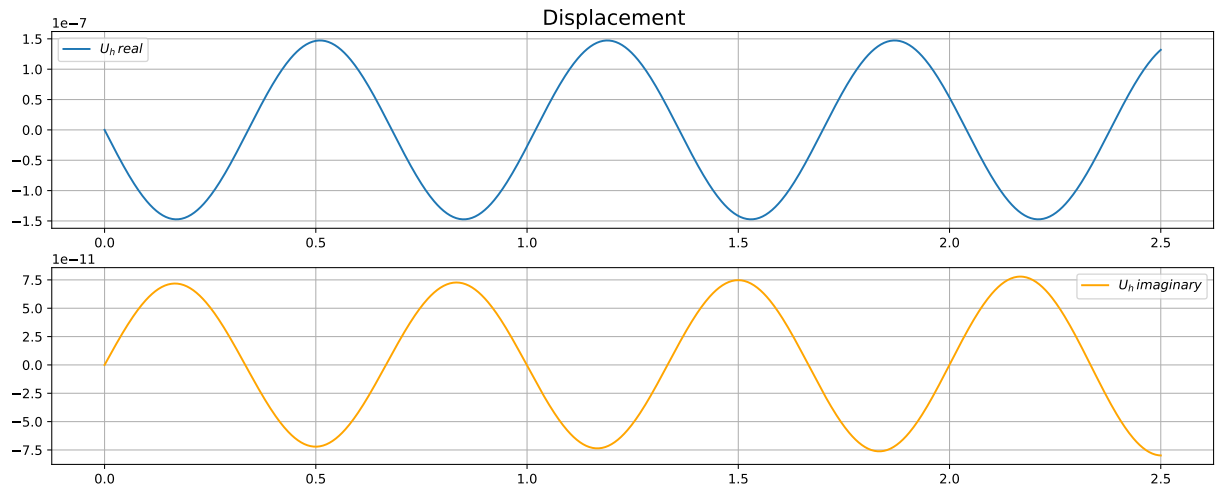


Figure 5.8: Fuel oil, $f = 2 \cdot 10^3$ Hz.

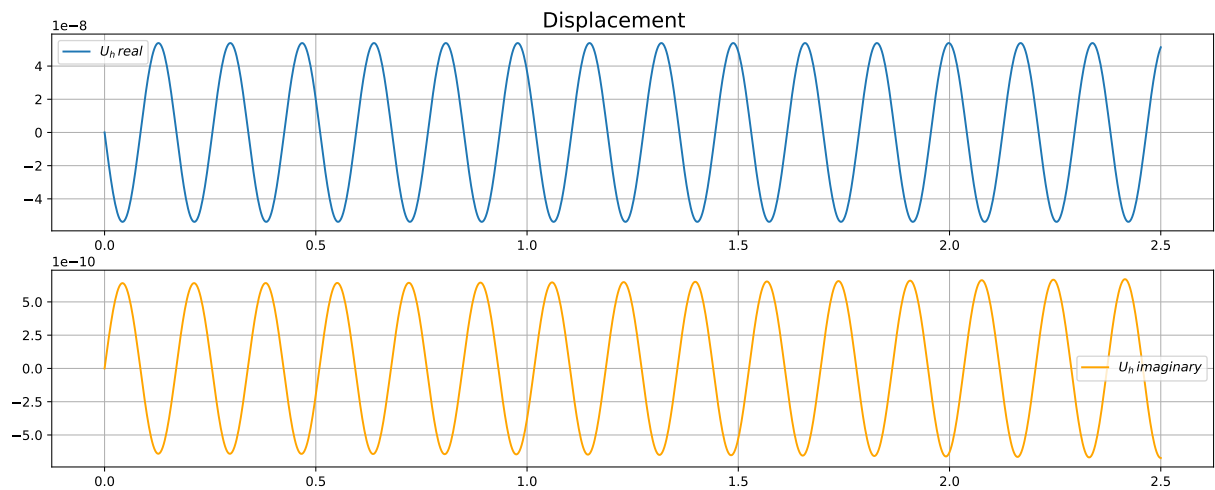


Figure 5.9: Fuel oil, $f = 8 \cdot 10^3$ Hz.

5.2 Two-dimensional problems

Consider a square region of 0.5×0.5 m with a mesh division step $h = 10^{-2}$. The domain has rigid walls, and the pressure $p = 10^3$ Pa is applied to the left boundary.

5.2.1 Propagation in Time Domain across square region

Given that the square area is considered, it is necessary to adjust the calculation parameters, in particular: $p_t = 1.4 \cdot 10^{-4}$, $T = 7 \cdot 10^{-4}$ and $\Delta t = 4 \cdot 10^{-7}$ s.

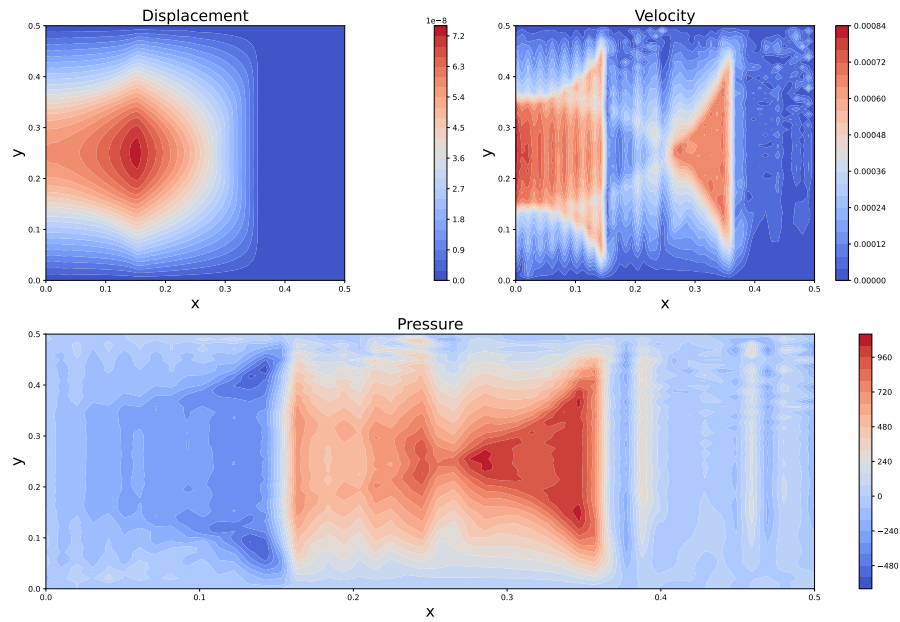


Figure 5.10: Water - 2D projection, $t = 2.4 \cdot 10^{-4}$ s.

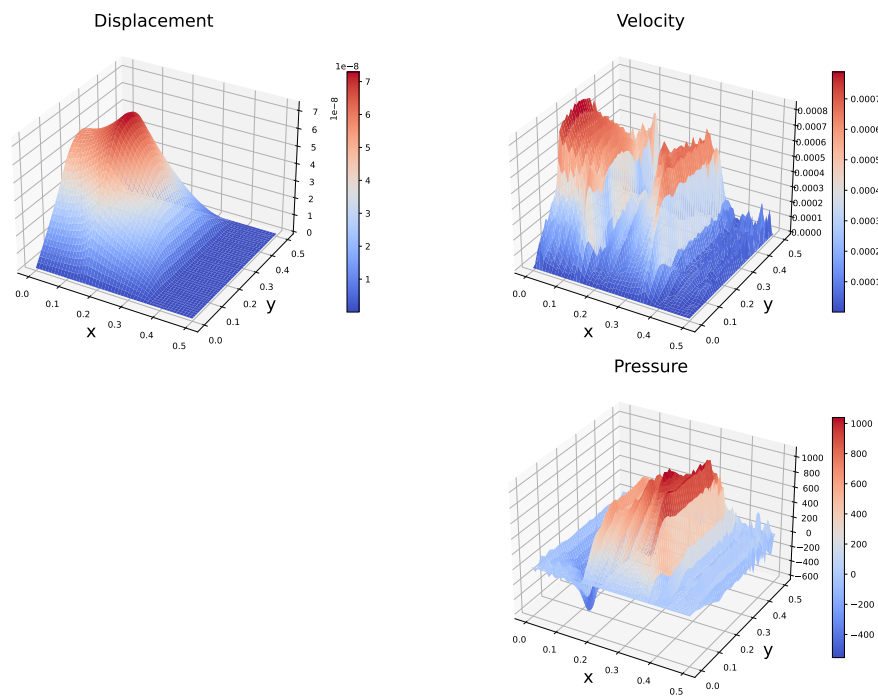


Figure 5.11: Water - 3D projection, $t = 2.4 \cdot 10^{-4}$ s.

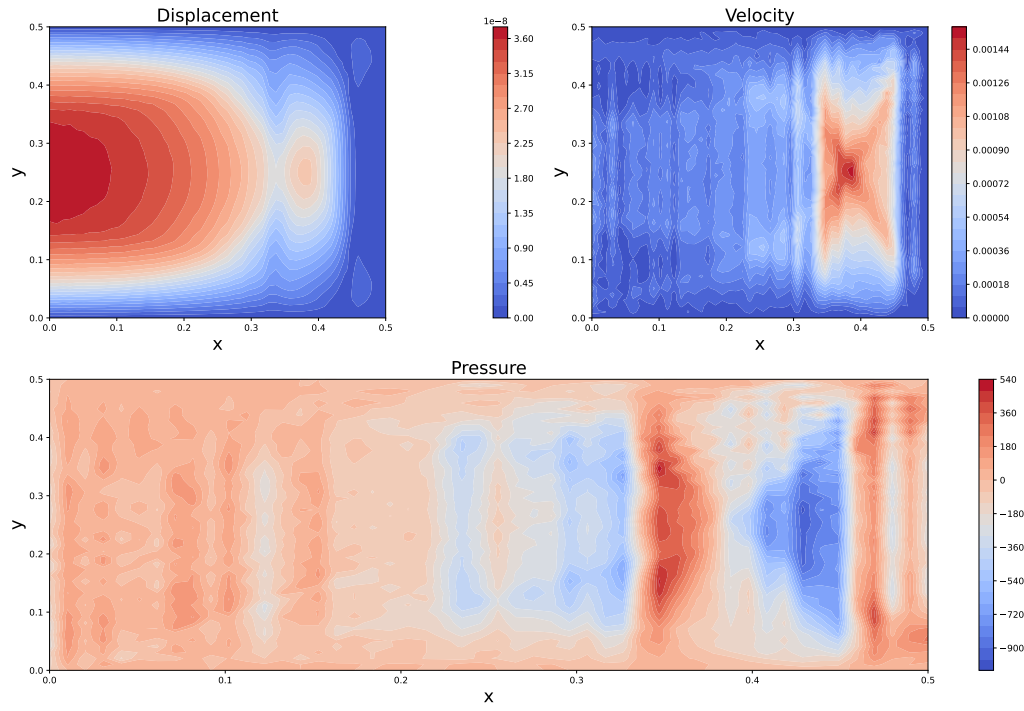


Figure 5.12: Water - 2D projection, $t = 4.4 \cdot 10^{-4}$ s.

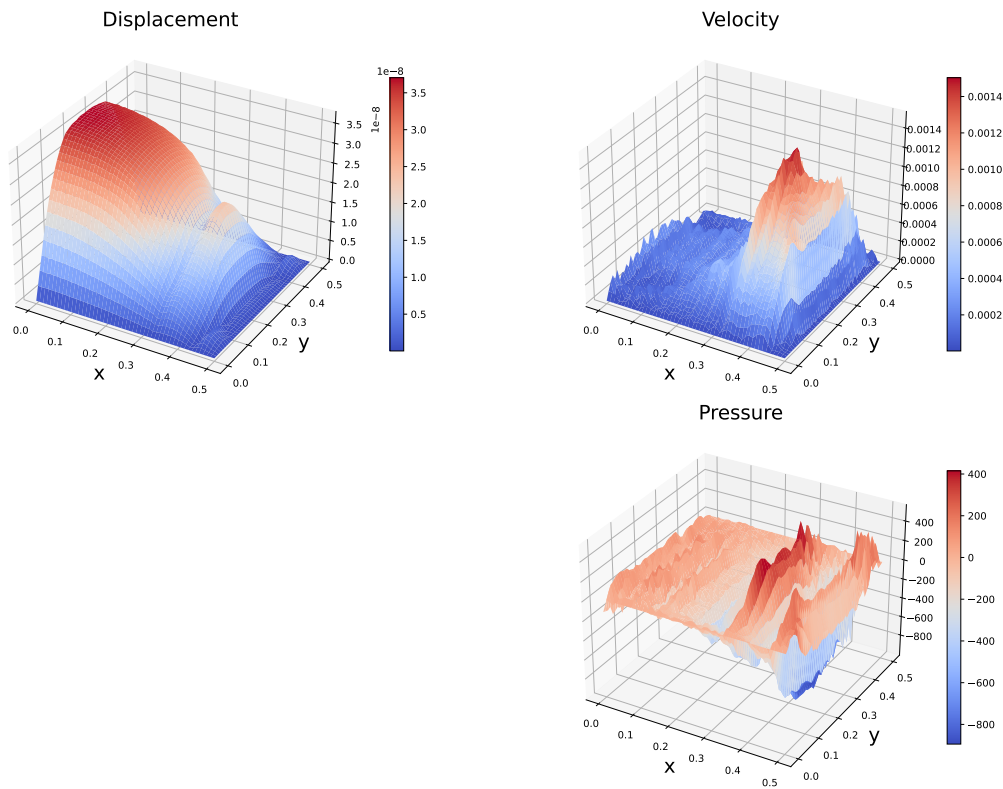


Figure 5.13: Water - 3D projection, $t = 4.4 \cdot 10^{-4}$ s.

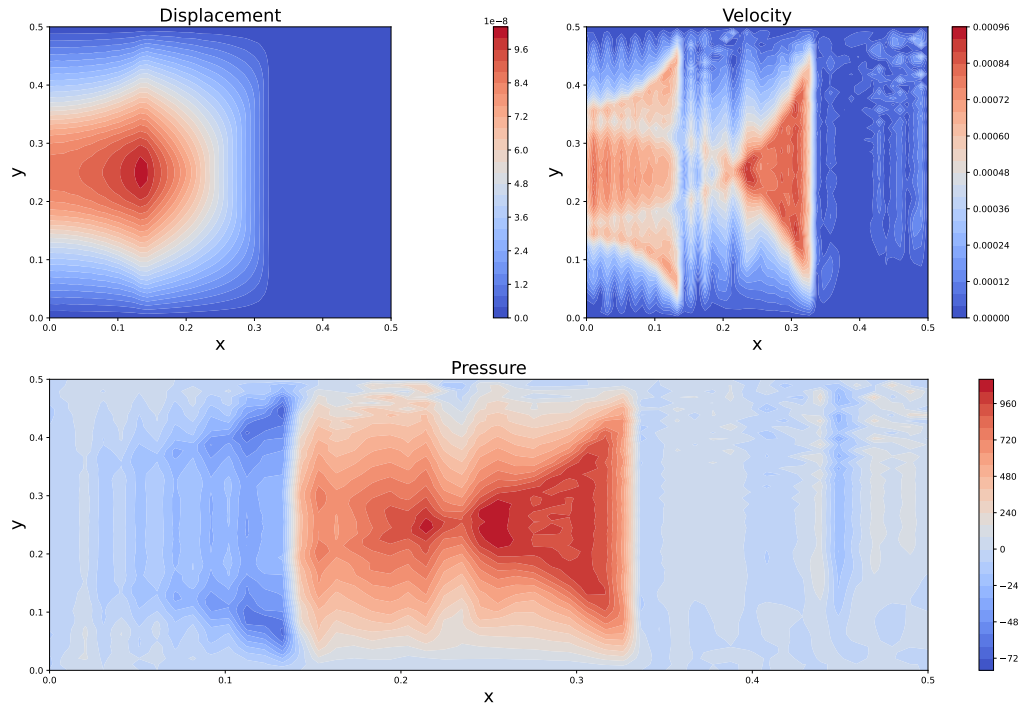


Figure 5.14: Fuel oil - 2D projection, $t = 2.4 \cdot 10^{-4}$ s.

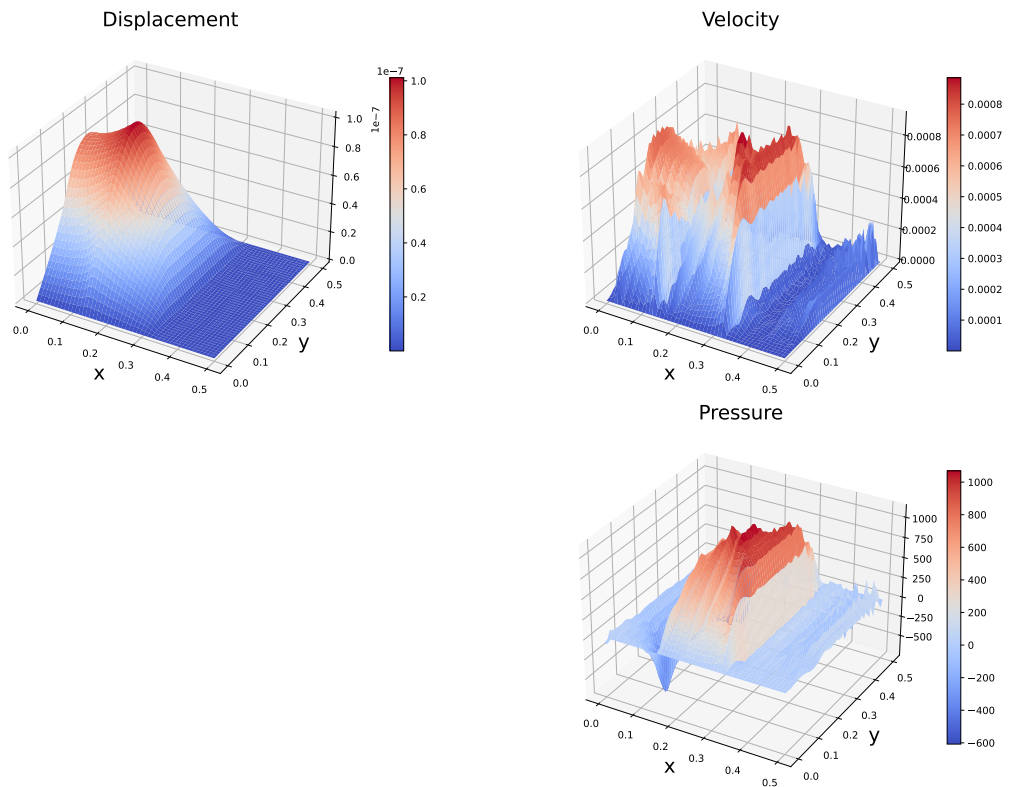


Figure 5.15: Fuel oil - 3D projection, $t = 2.4 \cdot 10^{-4}$ s.

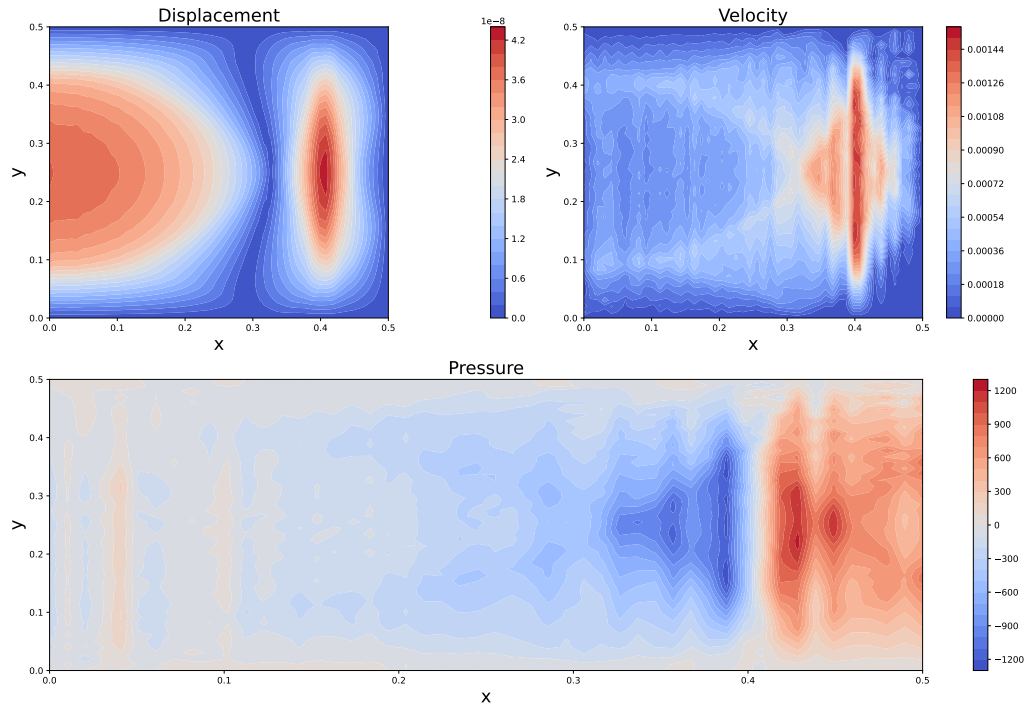


Figure 5.16: Fuel oil - 2D projection, $t = 4.4 \cdot 10^{-4}$ s.

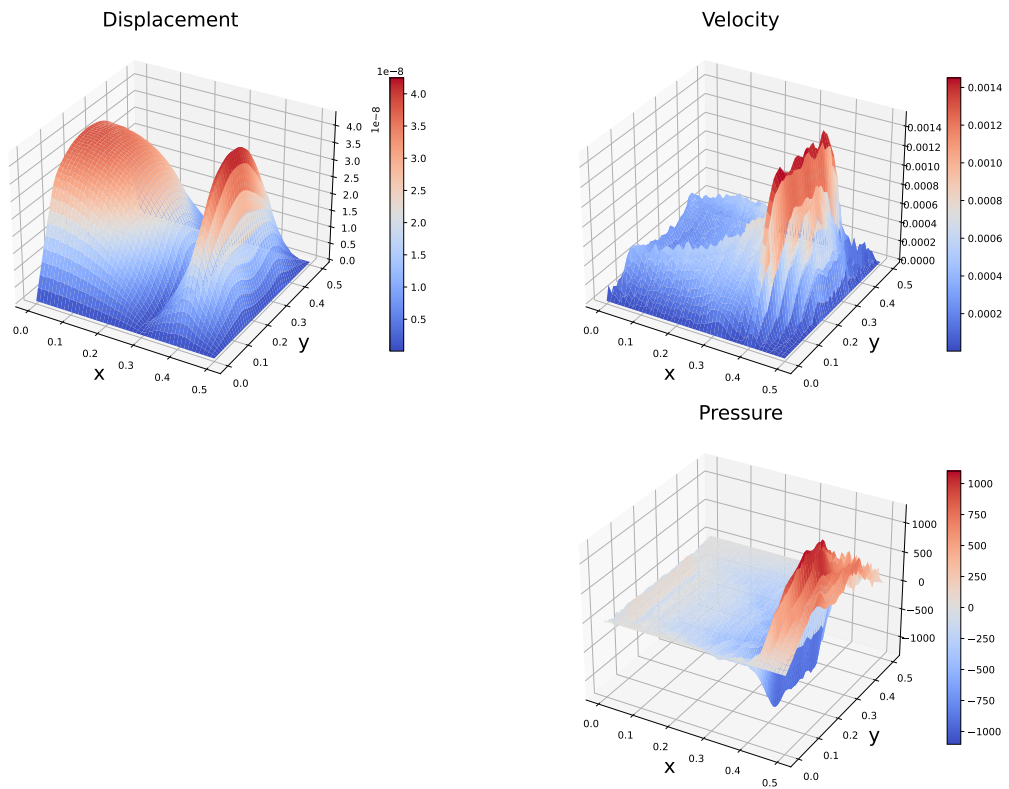


Figure 5.17: Fuel oil - 3D projection, $t = 4.4 \cdot 10^{-4}$ s.

Figure 5.18 show the change in energy for the system filled with fuel oil and calculated by the equation (3.10).

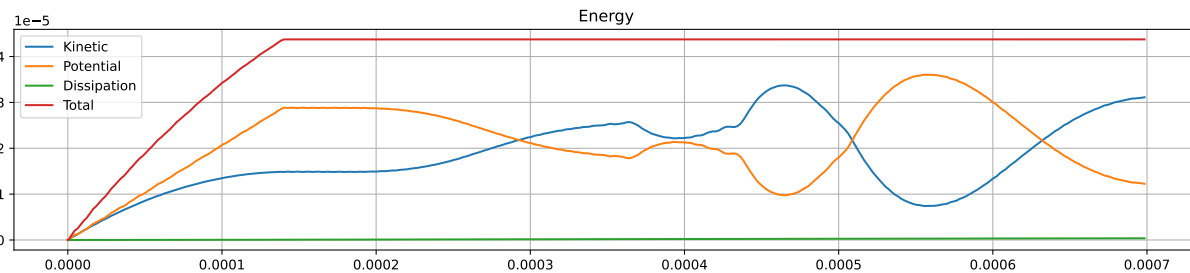


Figure 5.18: Change of energy in fuel oil.

5.2.2 Propagation in Frequency Domain across square region

The figures below show the propagation of an acoustic wave at different frequencies, where u_{h_r}, p_r – real parts of the solution, and u_{h_i}, p_i – imaginary.

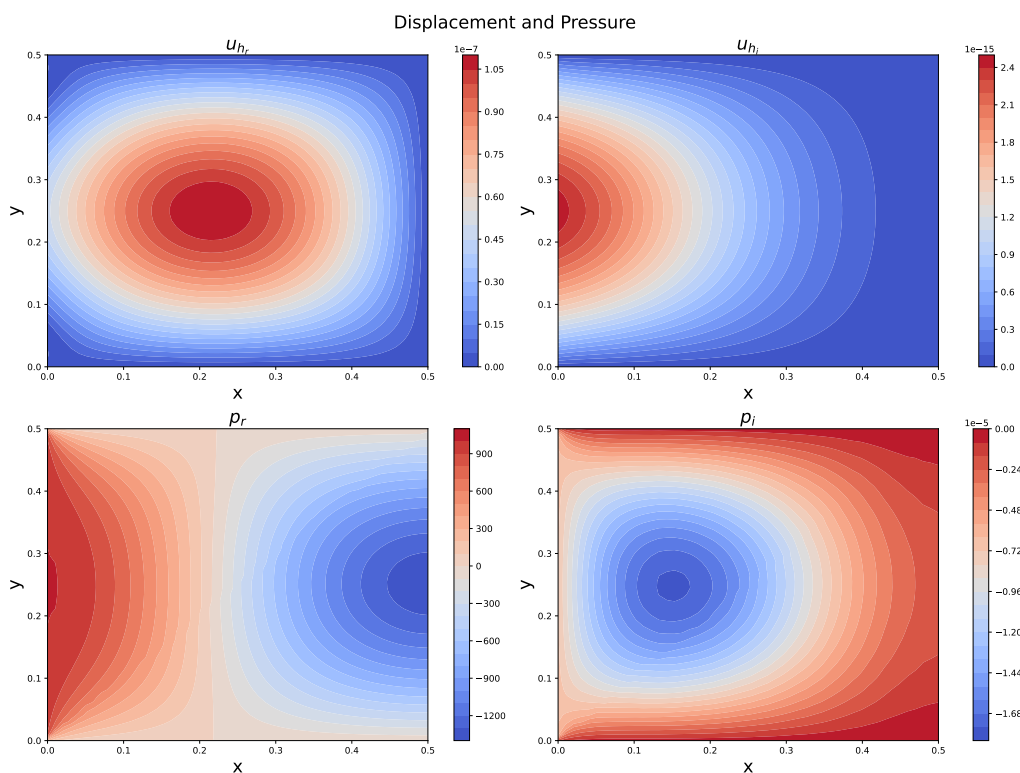


Figure 5.19: Water - 2D projection, $f = 2 \cdot 10^3$ Hz.

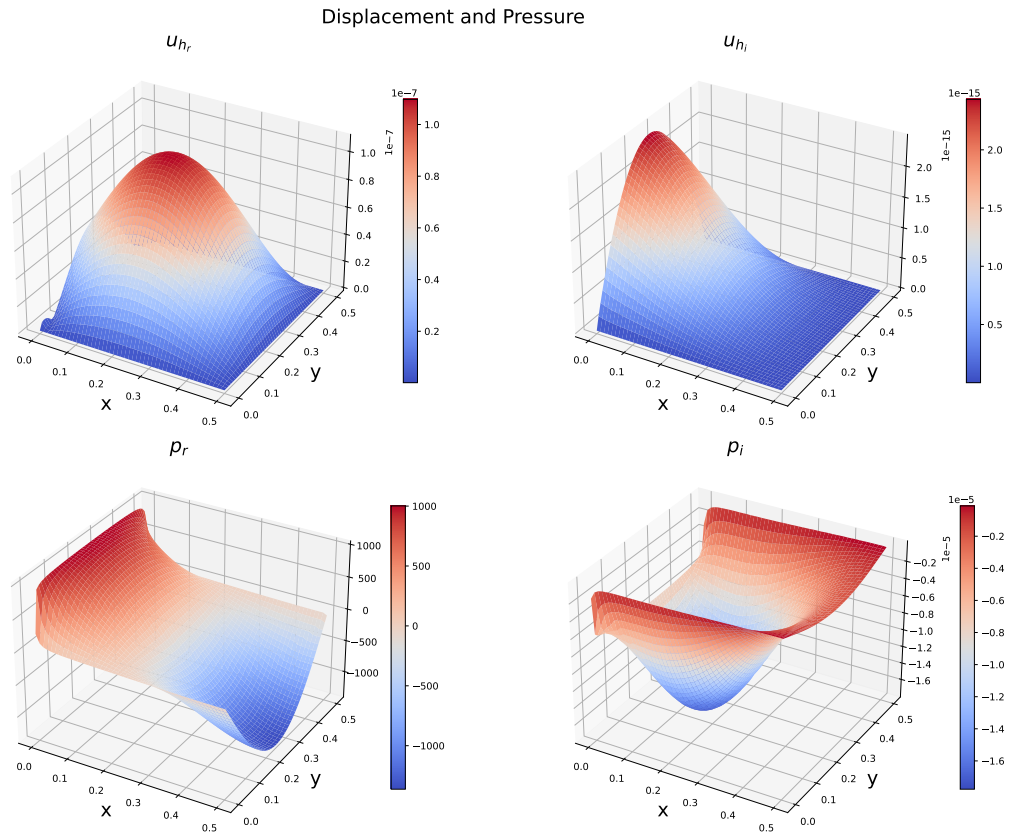


Figure 5.20: Water - 3D projection, $f = 2 \cdot 10^3$ Hz.

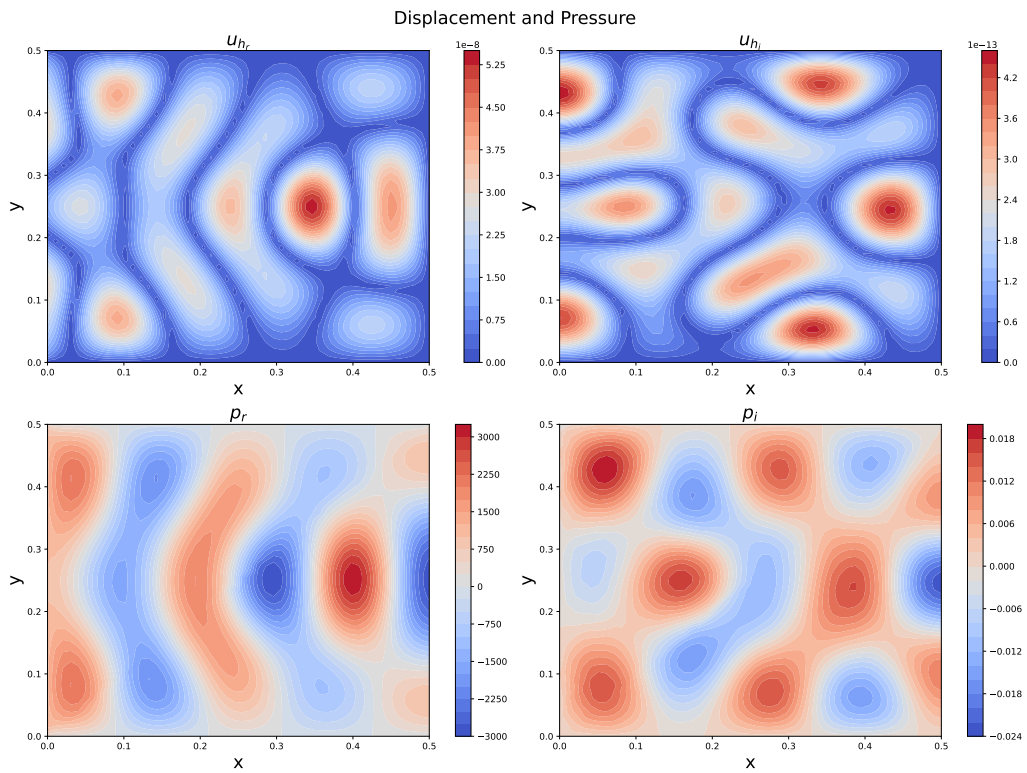


Figure 5.21: Water - 2D projection, $f = 8 \cdot 10^3$ Hz.

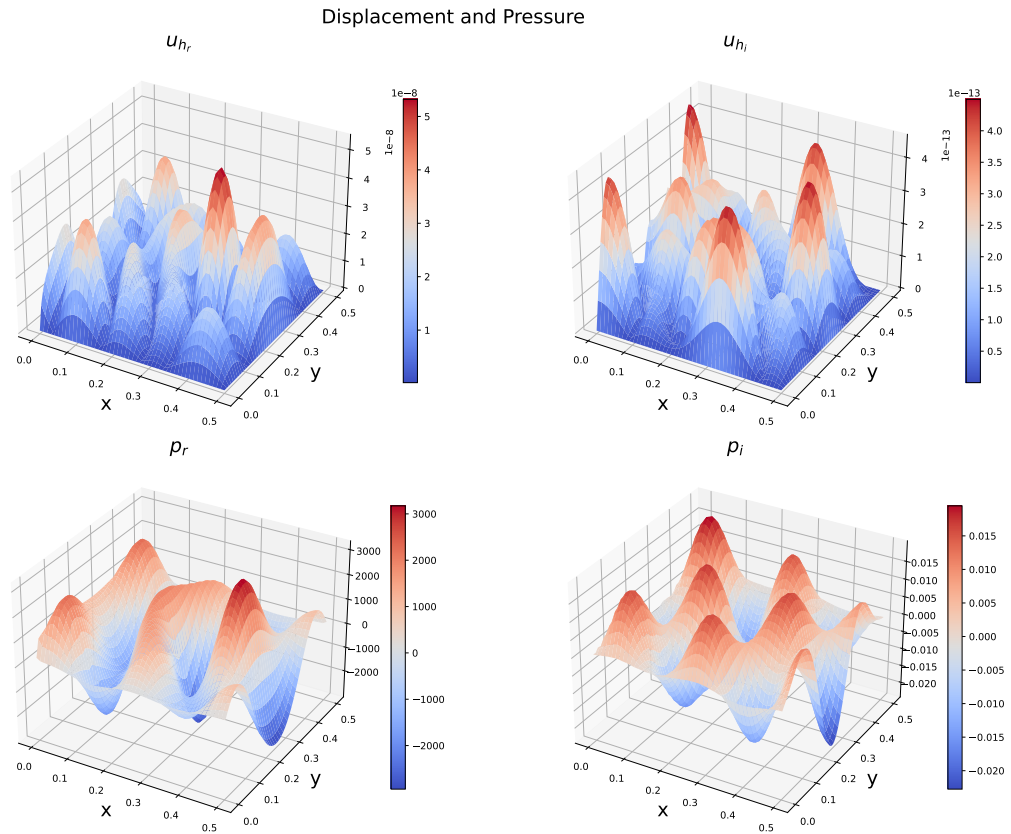


Figure 5.22: Water - 3D projection, $f = 8 \cdot 10^3$ Hz.

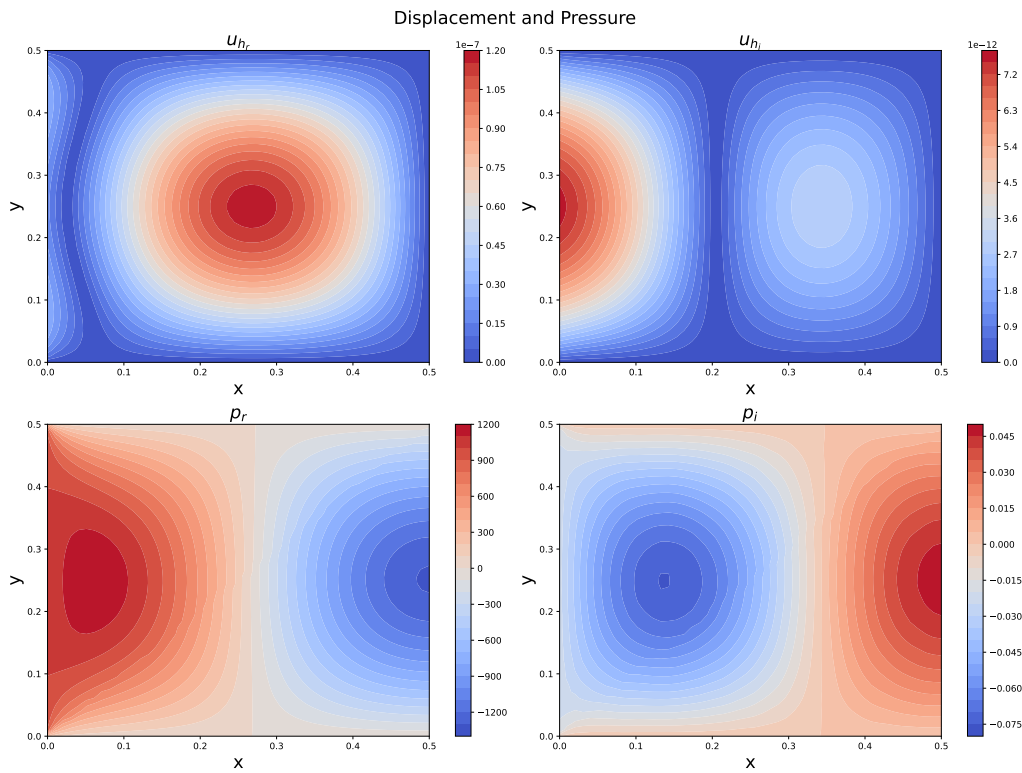


Figure 5.23: Fuel oil - 2D projection, $f = 2 \cdot 10^3$ Hz.

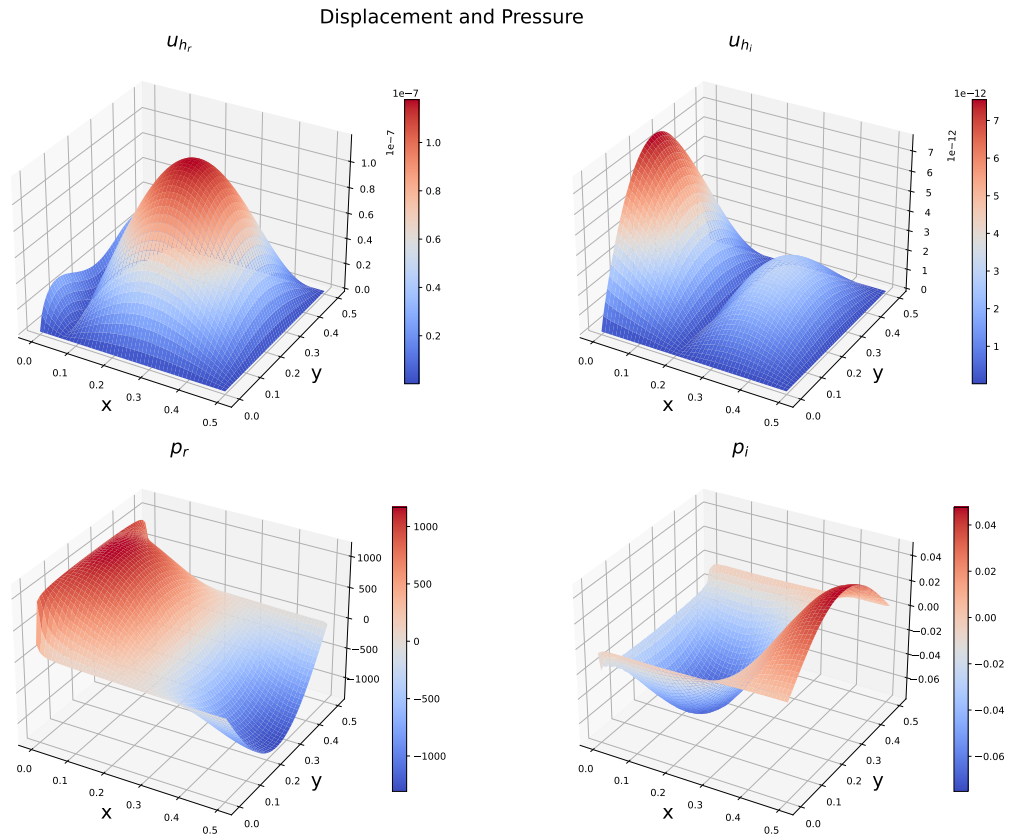


Figure 5.24: Fuel oil - 3D projection, $f = 2 \cdot 10^3$ Hz.

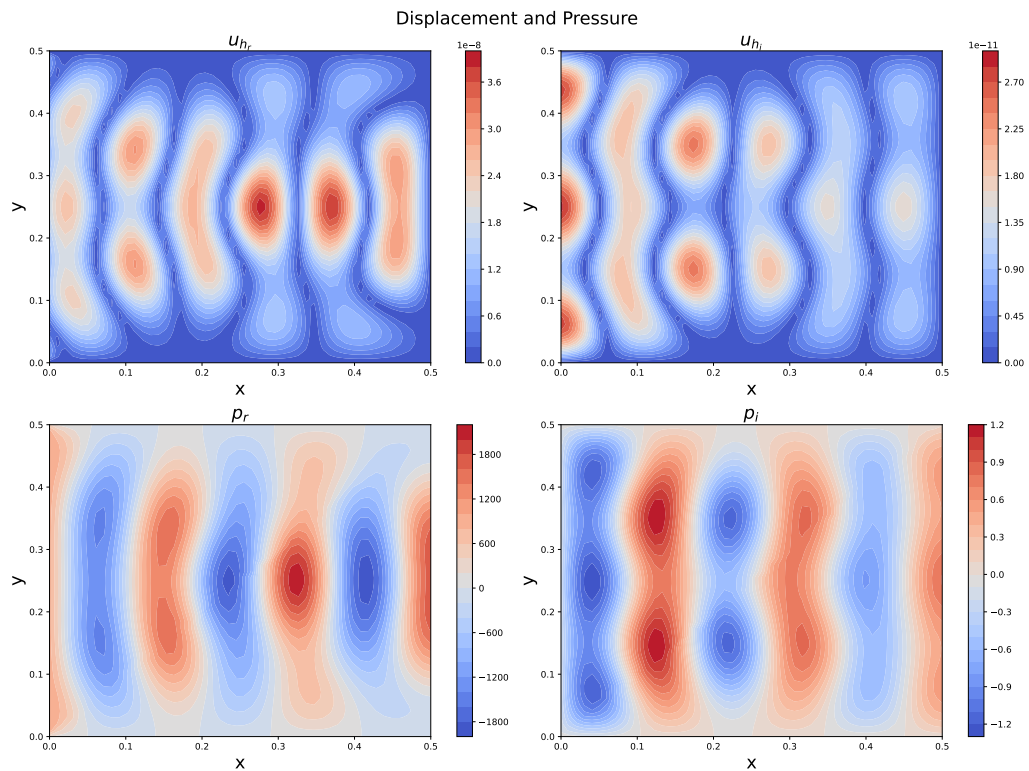


Figure 5.25: Fuel oil - 2D projection, $f = 8 \cdot 10^3$ Hz.

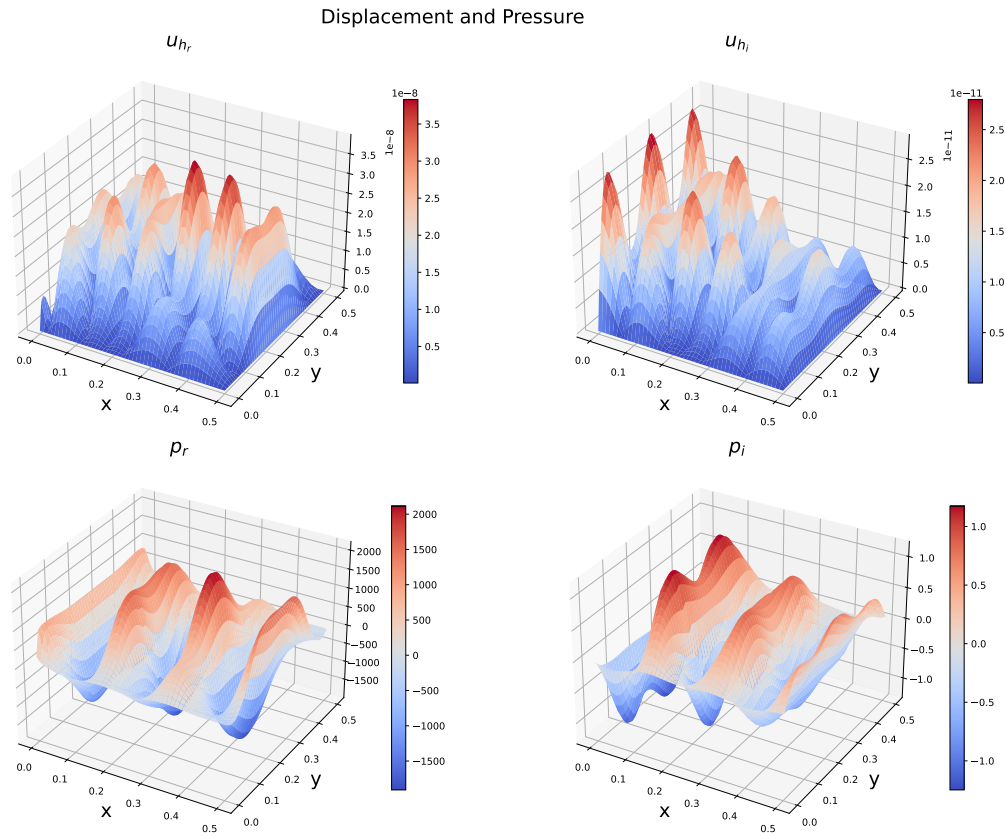


Figure 5.26: Fuel oil - 3D projection, $f = 8 \cdot 10^3$ Hz.

Chapter 6

Contamination problem

Contamination is a major issue in any study related to acoustic wave propagation. To accurately model the behavior of acoustic waves in a real-world scenario, it is essential to consider the effects of contamination on wave propagation. This is particularly important when studying more complex problems that go beyond ideal cases of homogeneous environments.

6.1 Domain model

Consider a rectangular area filled with water containing air bubbles of a given size and density. These air bubbles represent different levels of contamination inside the domain, as in Fig 6.1.

On top of that, suppose there is a device on the left edge that generates sound waves through the application of pressure on the wall. And on the opposite edge, there installed a receiver that measures pressure integral along the boundary.

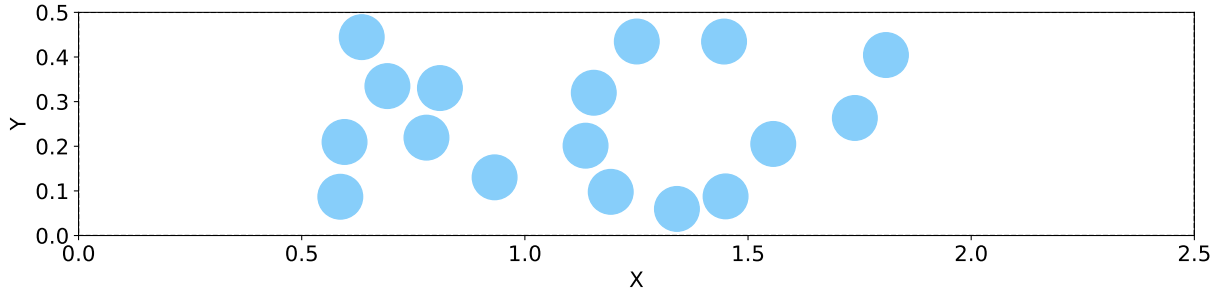


Figure 6.1: Domain model.

Assume that air bubbles are more likely to be distributed in the central segments of the domain. Thus, simulations are constructed in the following way:

1. Since the contamination can be located in different parts of the rectangle, to localize it, the domain is divided into five equal segments as in the Figure 6.2.

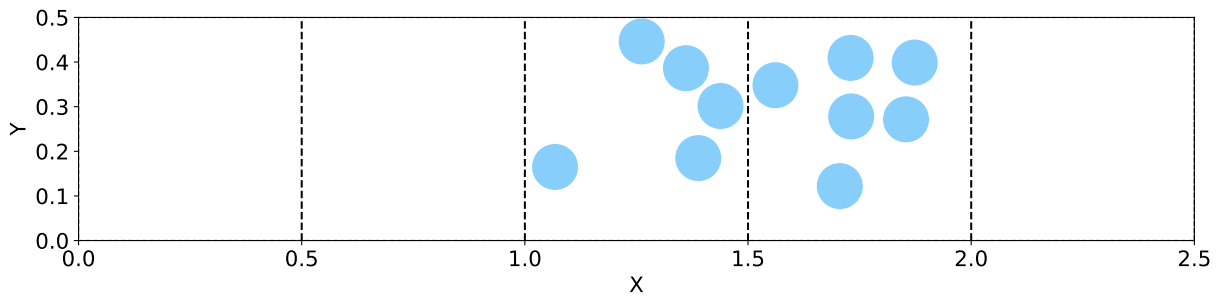


Figure 6.2: Segment division.

2. Then the number of contaminated segments is calculated using an exponential distribution with parameter $\lambda = 0.5$. "1" is added to a generated number, and the result is limited to the interval $[1, 5]$.

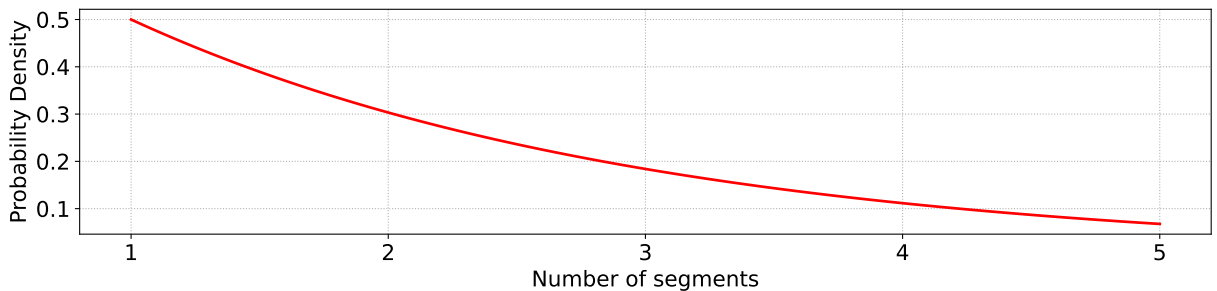


Figure 6.3: Distribution of number of selected segments.

3. After selecting the number of segments, their indexes are determined by a normal distribution with $\mu = 2$ and $\sigma = 2$ and limited to the interval $[0,4]$.

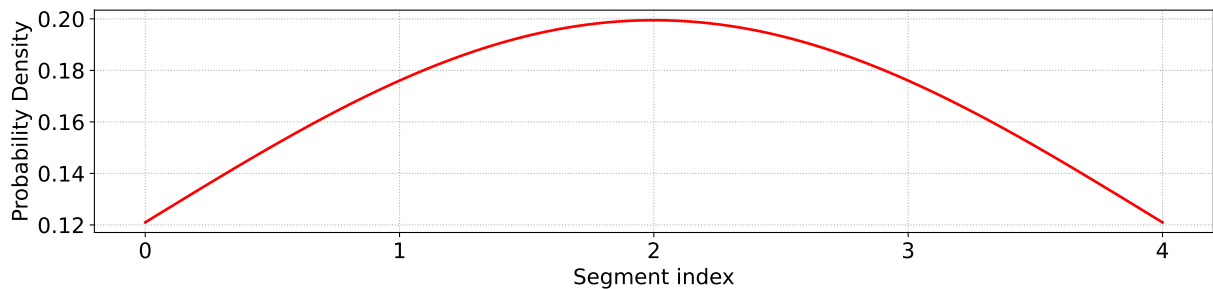


Figure 6.4: Distribution of segment indexes.

4. For each selected segment, it is assumed that the level of contamination follows a normal distribution with $\mu = 0.25$ and $\sigma = 0.25$ and limited to the interval $[0,0.5]$.

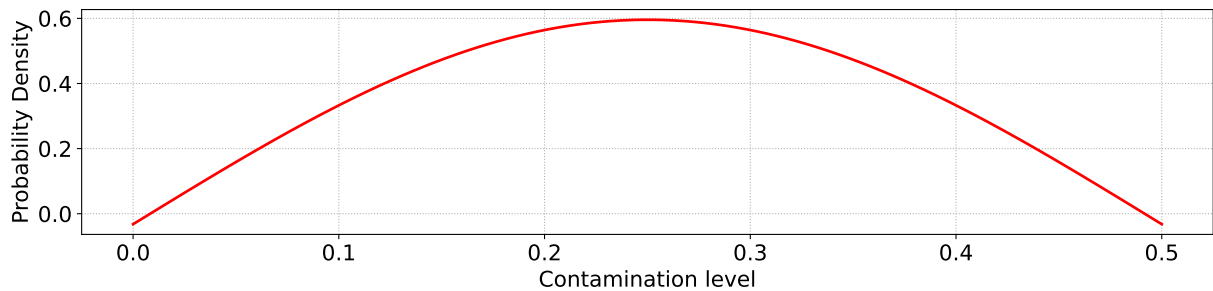


Figure 6.5: Distribution of contamination level.

5. The coordinates of each bubble, with radius $R = 0.05$, within its segment are calculated using a uniform distribution.

6.2 Model parameters

Using the region generated in Section 6.1, a number of experiments were conducted for the problem (4.1). The main objective of these experiments was to determine the time intervals in which measurements should be taken.

Initially, a set of parameters are similar to those described in Section 5.2.1, that is:

- Water — $\rho = 1000 \text{ kg/m}^3$, $c = 1500 \text{ m/s}$ and $\eta = 8.94 \cdot 10^{-4} \text{ Pa} \cdot \text{s}$
- Air — $\rho = 1.2 \text{ kg/m}^3$, $c = 343 \text{ m/s}$ and $\eta = 1.5 \cdot 10^{-5} \text{ Pa} \cdot \text{s}$
- $p = 10^3 \text{ Pa}$, $p_t = 7 \cdot 10^{-4} \text{ s}$
- $T = 3.5 \cdot 10^{-3} \text{ s}$, $\Delta t = 2 \cdot 10^{-6} \text{ s}$
- $h = 1.25 \cdot 10^{-2}$

Figure 6.6 shows the continuous recordings on the receiver throughout the entire duration of time T for **10** randomly selected regions with contamination. For consistency, the values at the receiver are additionally calculated as the average of **4** iterations with a step of Δt , that is, the time it takes to pass one finite element.

Using the above mentioned assumptions, **12** time intervals with varying values were chosen, namely: $1.344 \cdot 10^{-3}$, $1.632 \cdot 10^{-3}$, $1.784 \cdot 10^{-3}$, $1.864 \cdot 10^{-3}$, $1.944 \cdot 10^{-3}$, $2.104 \cdot 10^{-3}$, $2.240 \cdot 10^{-3}$, $2.440 \cdot 10^{-3}$, $2.600 \cdot 10^{-3}$, $2.880 \cdot 10^{-3}$, $3.080 \cdot 10^{-3}$, $3.360 \cdot 10^{-3}$.

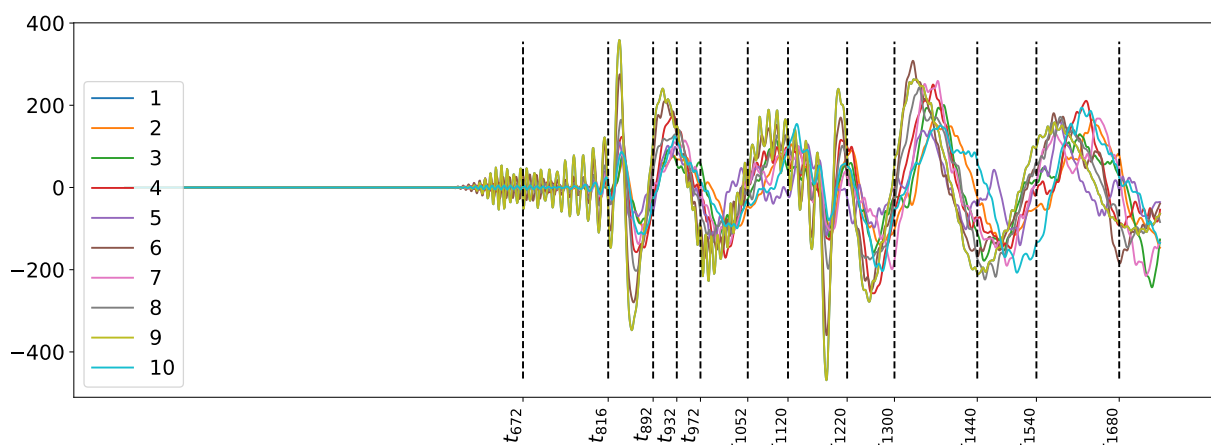


Figure 6.6: Time points.

Figure 6.7 shows the changes in acoustic pressure in both contaminated, as in Figure 6.2 and non-contaminated domains for the first four time intervals.

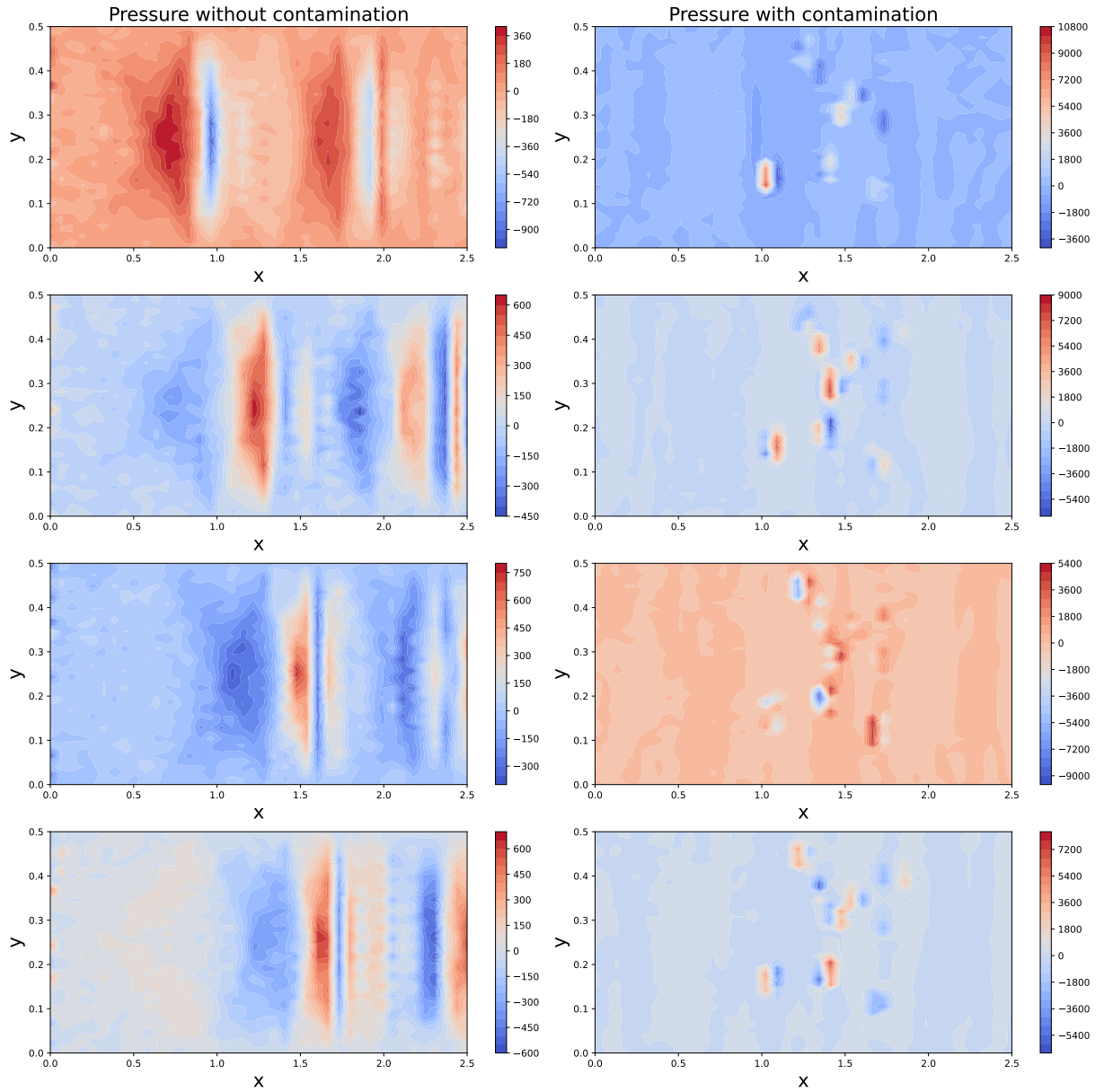


Figure 6.7: Pressure in ideal and contaminated domains.

Observing the propagation of acoustic waves in contaminated media can be challenging because of the contrasting physical characteristics of water and air. However, in Figure 6.8, to demonstrate the preservation of the wave structure, the data for the region with bubbles was normalized.

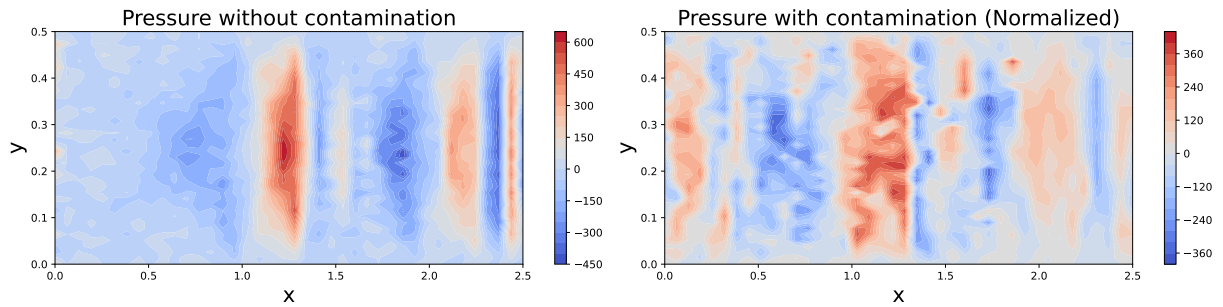


Figure 6.8: Pressure in ideal and contaminated domains with normalization.

Similarly, considering experiments conducted in Section 5.2.2, the following set of parameters was chosen for the problem (4.3):

- Water — $\rho = 1000 \text{ kg/m}^3$, $c = 1500 \text{ m/s}$ and $\eta = 8.94 \cdot 10^{-4} \text{ Pa} \cdot \text{s}$
- Air — $\rho = 1.2 \text{ kg/m}^3$, $c = 343 \text{ m/s}$ and $\eta = 1.5 \cdot 10^{-5} \text{ Pa} \cdot \text{s}$
- Frequencies — $f_1 = 1.5 \cdot 10^3$, $f_2 = 2 \cdot 10^3$, $f_3 = 3 \cdot 10^3$, $f_4 = 8 \cdot 10^3 \text{ Hz}$
- $p = 10^3 \text{ Pa}$
- $h = 6.25 \cdot 10^{-3}$

Figure 6.9 shows the changes in acoustic pressure in both contaminated, as in Figure 6.2 and non-contaminated domains for the four selected frequencies. Following the same principle, normalized values of pressure for the contaminated media are also shown in Figure 6.10.

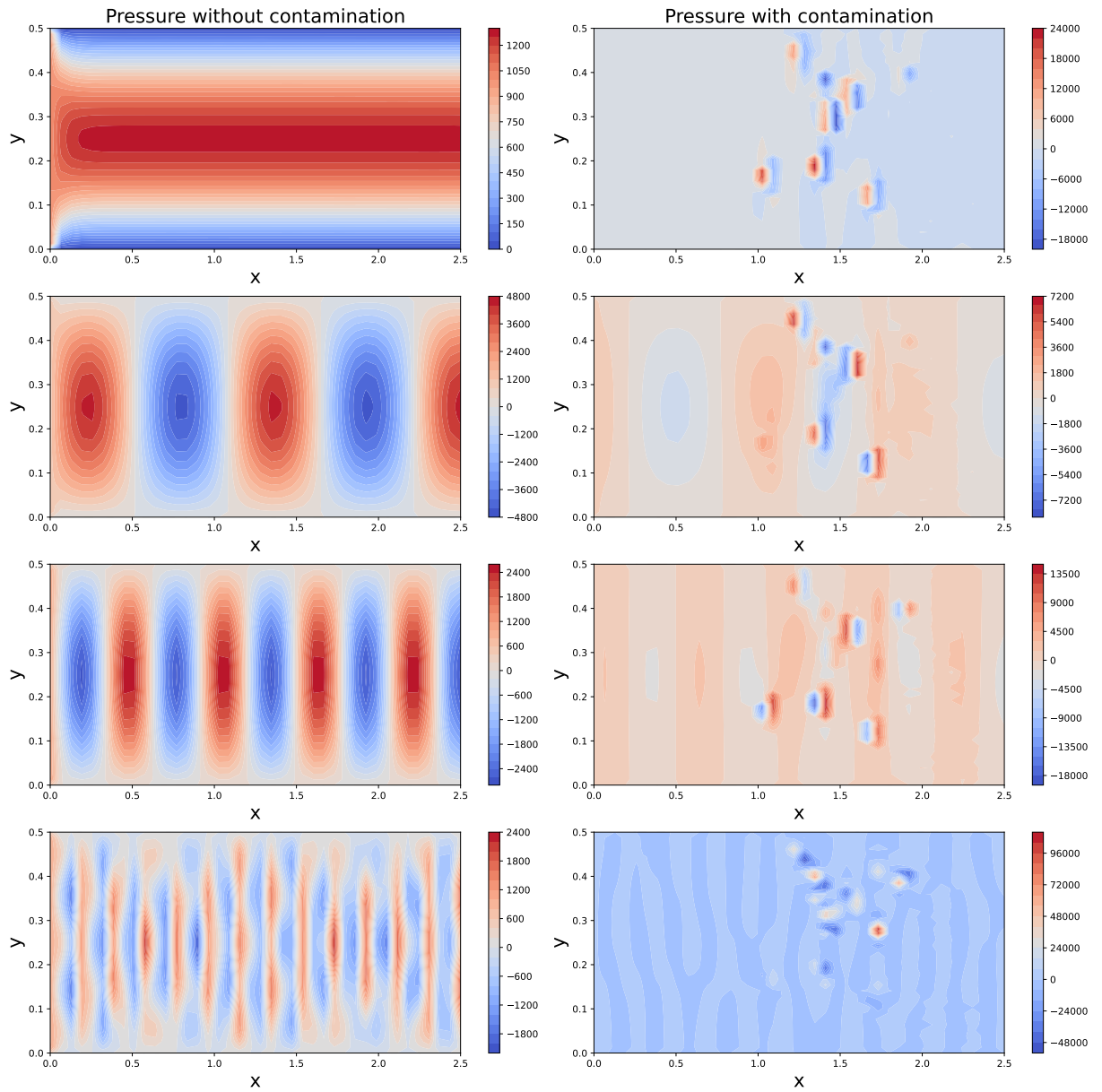


Figure 6.9: Pressure in ideal and contaminated domains.

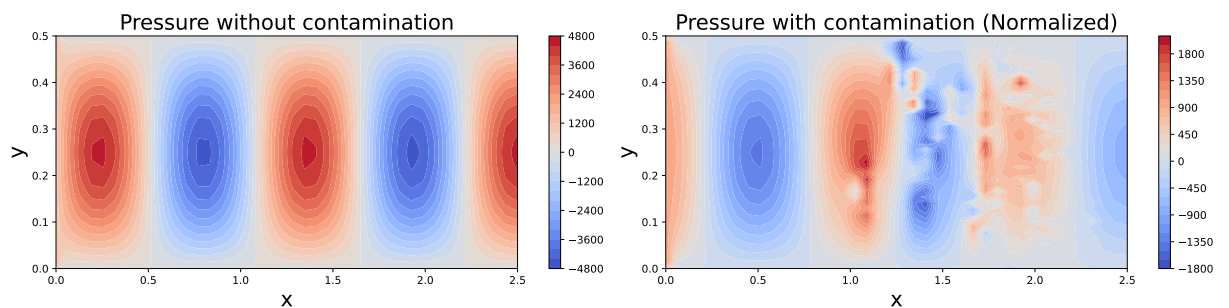


Figure 6.10: Pressure in ideal and contaminated domains with normalization.

6.3 Dataset generation

For the problems described in Section 6.1 with the parameters defined in Section 6.2, a large number of computations were required. Given the scale of the problem, distributed computing was deemed necessary for efficient and timely dataset generation.

Microsoft Azure is a cloud computing platform that provides a variety of services, including the ability to conduct distributed computations. It is a powerful tool that offers scalable and efficient solutions with a wide range of computational needs [21].

One of the key advantages of using Microsoft Azure for distributed computations is the ability to create and manage virtual machines. These virtual machines can be configured to suit specific computational needs, such as the number of cores and the amount of memory required. Additionally, Microsoft Azure allows the scaling of the number of virtual machines, making it a flexible and cost-effective solution for distributed computations.

For these computations, eight virtual machines were used, as seen in Figure 6.11. Each ran on the Ubuntu 18.04 LTS operating system and had the latest Docker image with FEniCSx library version 0.6.0 installed. All the code used can be found here [22].









<input type="checkbox"/> Name ↑↓	Type ↑↓	Resource group ↑↓	Location ↑↓	Operating system
<input type="checkbox"/>  dolfinx-01	Virtual machine	dolfinx	West Europe	Linux
<input type="checkbox"/>  dolfinx-02	Virtual machine	dolfinx	West Europe	Linux
<input type="checkbox"/>  dolfinx-03	Virtual machine	dolfinx	West Europe	Linux
<input type="checkbox"/>  dolfinx-04	Virtual machine	dolfinx	UK South	Linux
<input type="checkbox"/>  dolfinx-05	Virtual machine	dolfinx	UK South	Linux
<input type="checkbox"/>  dolfinx-06	Virtual machine	dolfinx	France Central	Linux
<input type="checkbox"/>  dolfinx-07	Virtual machine	dolfinx	France Central	Linux
<input type="checkbox"/>  dolfinx-08	Virtual machine	dolfinx	North Europe	Linux

Figure 6.11: Virtual machines.

Given the results described in [18], the computation time for the frequency domain problem can be further reduced by using parallel computing with Message Passing Interface (MPI). Listing 6.1 shows the code to start the generator for virtual machines, considering all of the above.

```

1  def main():
2      parser = argparse.ArgumentParser()
3      parser.add_argument('--name', type=str, help="Domain_name", default
4                          = 'frequency')
5      parser.add_argument('-r', nargs=2, type=int, help="Input_file_range",
6                          default=[0, 1000], metavar=('a', 'b'))
7      args = parser.parse_args()
8      if args.name == 'frequency':
9          for i in range(args.r[0], args.r[1]):
10             command = ['mpirun', '-np', '2', './RectangleModel-
11                      FrequencyDomain-MPI.py', '-i', f'{i}']
12             p = subprocess.run(command, capture_output=True, text=True)
13             print(p.stdout, end='')
14         elif args.name == 'time':
15             for i in range(args.r[0], args.r[1]):
16                 command = ['./RectangleModel-TimeDomain-MPI.py',
17                             '-i', f'{i}']

```



```

16     p = subprocess.run(command, capture_output=True, text=True)
17     print(p.stdout, end='')

```

Listing 6.1: Starting Python script

Listings 6.2 and 6.3 show functions that return initial configurations for time and frequency domain problems, respectively.

```

1  def generate_time_config(bubble_centres, bubble_lvl):
2      time_config = {
3          'mesh': {
4              'N': 40,
5              'bubble_radius': 0.05,
6              'bubble_centres': bubble_centres,
7              'bubble_lvl': bubble_lvl
8          },
9          'problem': {
10             'fluid': 'Water',
11             'contaminant': 'Air',
12             'pressure': 1e3,
13             'T': 3.5e-3,
14             'dt': 2e-6,
15             'pt': 7e-4,
16             'delay': 1.0,
17             'control_time': np.array([168, 204, 223, 233, 243, 263,
18                                     280, 305, 325, 360, 385, 420])*4
19         },
20         'petsc': {
21             'solver': 'preonly',
22             'pc': 'lu'
23         },
24         'results': {
25             'save_mesh_to_file': False,
26             'file_prefix': False,
27             'save_step': 25,
28         },
29         'logs': False
30     }
31     return time_config

```

Listing 6.2: Time domain config

```

1 def generate_frequency_config(bubble_centres , bubble_lvl):
2     frequency_config = {
3         'mesh': {
4             'N': 80,
5             'bubble_radius': 0.05,
6             'bubble_centres': bubble_centres ,
7             'bubble_lvl': bubble_lvl
8         },
9         'problem': {
10            'fluid': 'Water',
11            'contaminant': 'Air',
12            'pressure': 1e3,
13            'freq_idx': 0,
14            'control_frequencies': [1.5e3, 2e3, 3e3, 8e3]
15        },
16        'petsc': {
17            'solver': 'preonly',
18            'pc': 'lu'
19        },
20        'results': {
21            'save_mesh_to_file': False,
22            'file_prefix': False,
23        },
24        'logs': False
25    }
26    return frequency_config

```

Listing 6.3: Frequency domain config

In this way, both problems were solved for a set of **7000** contaminated media. The time taken for the frequency domain problem was **6** hours, and **8.5** hours for the time domain.

Chapter 7

Data Analysis and Neural Network Design

This chapter focuses on the exploration of the generated dataset from Chapter 6 to gain insights into its underlying structure and patterns. The aim is to develop simple classifiers and neural networks that can distinguish contamination inside the domain and accurately predict outcomes.

7.1 Time Domain dataset exploration and classification

In this and the next section, descriptive characteristics of the dataset are provided. The purpose of this step is to examine the data and find some useful information that can be used in the further construction of the neural networks.

Table 7.1 shows the initial data obtained after the dataset generation for the Time Domain problem, where z_i - contamination level in i -th segment, $i = 1..5$, and $p(t_j)$ - value of of pressure on a receiver at time $t_j, j = 1..12$.

Table 7.1: Initial Time Dataset

	z_1	z_2	z_3	z_4	z_5	$p(t_1)$	$p(t_2)$...	$p(t_{11})$	$p(t_{12})$
0	0.0	0.0	0.0	0.0	0.0	-3.95e+01	-9.85e+01	...	1.07e+02	-8.97e+01
1	0.0	0.0	0.1571	0.1885	0.0	1.47e-01	-1.06e+00	...	-4.69e+01	6.81e+01
2	0.0628	0.1885	0.0	0.0	0.0	4.92e+00	-5.57e+00	...	1.32e+01	2.23e+01
...
6991	0.0	0.0	0.0	0.0942	0.0	-8.59e+00	-2.50e+01	...	4.96e+01	-5.13e+01
6992	0.0	0.0	0.0	0.0	0.2199	-2.38e+00	-1.36e+01	...	1.26e+02	-5.74e+01
6993	0.0	0.0	0.0	0.1885	0.0	8.57e+00	-1.59e+01	...	2.94e+01	-8.13e+01

The first step in preparing the data was to convert the percentage of contamination with respect to contamination threshold c_T . That is, if the contamination is greater than 0, the segment value is *True*, otherwise - *False*, as shown in Table 7.2, where z_i^* and $p_N(t_j)$ are the transformed values.

Also, to ensure that all features have the same weight in the analysis, it is necessary to scale them to the same level. For this purpose, the ‘**Standard-Scaler**’ from the ‘*sklearn*’ library was used. This transformation helps to eliminate any bias towards features with large values that can lead to inaccurate results. It also improves numerical stability, making optimization algorithms more efficient in convergence.

Table 7.2: Transformed Time Dataset

	z_1^*	z_2^*	z_3^*	z_4^*	z_5^*	$p_N(t_1)$	$p_N(t_2)$...	$p_N(t_{11})$	$p_N(t_{12})$
0	False	False	False	False	False	-6.645584	-3.204366	...	1.318628	-0.736776
1	False	False	True	True	False	0.124733	0.570926	...	-0.720212	1.566839
2	True	True	False	False	False	0.939794	0.396187	...	0.075982	0.898236
...
6991	False	False	False	True	False	-1.367240	-0.356624	...	0.558203	-0.176201
6992	False	False	False	False	True	-0.306790	0.085066	...	1.570336	-0.265250
6993	False	False	False	True	False	1.563086	-0.004047	...	0.290597	-0.614150

In the analysis of the dataset, the ‘*ydata-profiling*’ library was used to generate summary statistics that include the frequency distribution, unique values, and a number of missing values for each dataset column. Additionally, ‘*ydata-profiling*’ offers visualization tools, like scatter plots, box plots, and histograms, which aid in dataset exploration and analysis [23]. Figure 7.1 showcases one of the examples from obtained results.

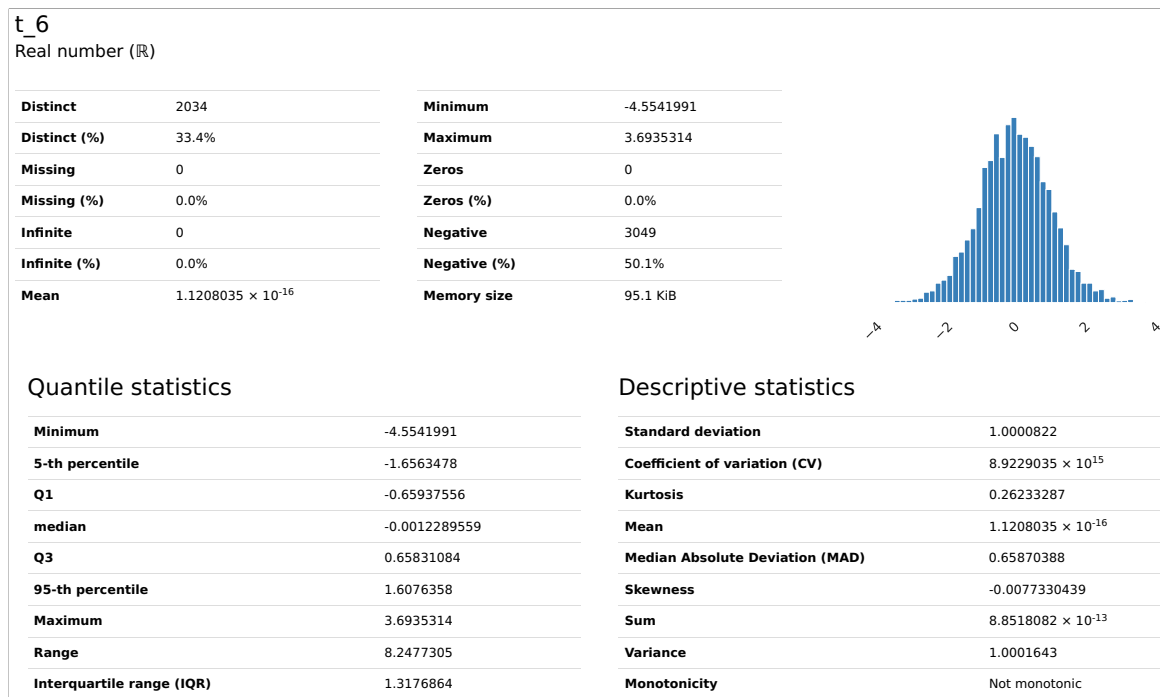


Figure 7.1: Profile report for values $p_N(t_6)$.

Figure 7.2 and Table 7.3 display the correlation coefficients between variables in a dataset. This analysis is significant for building classifiers since it can aid in feature selection and provide insights into how different features are related. Therefore, it can be seen that the first segment has the highest correlation with all time intervals. Besides, $p_N(t_2)$, $p_N(t_4)$, $p_N(t_7)$, and $p(t_{10})$ correlate with each segment more than others.

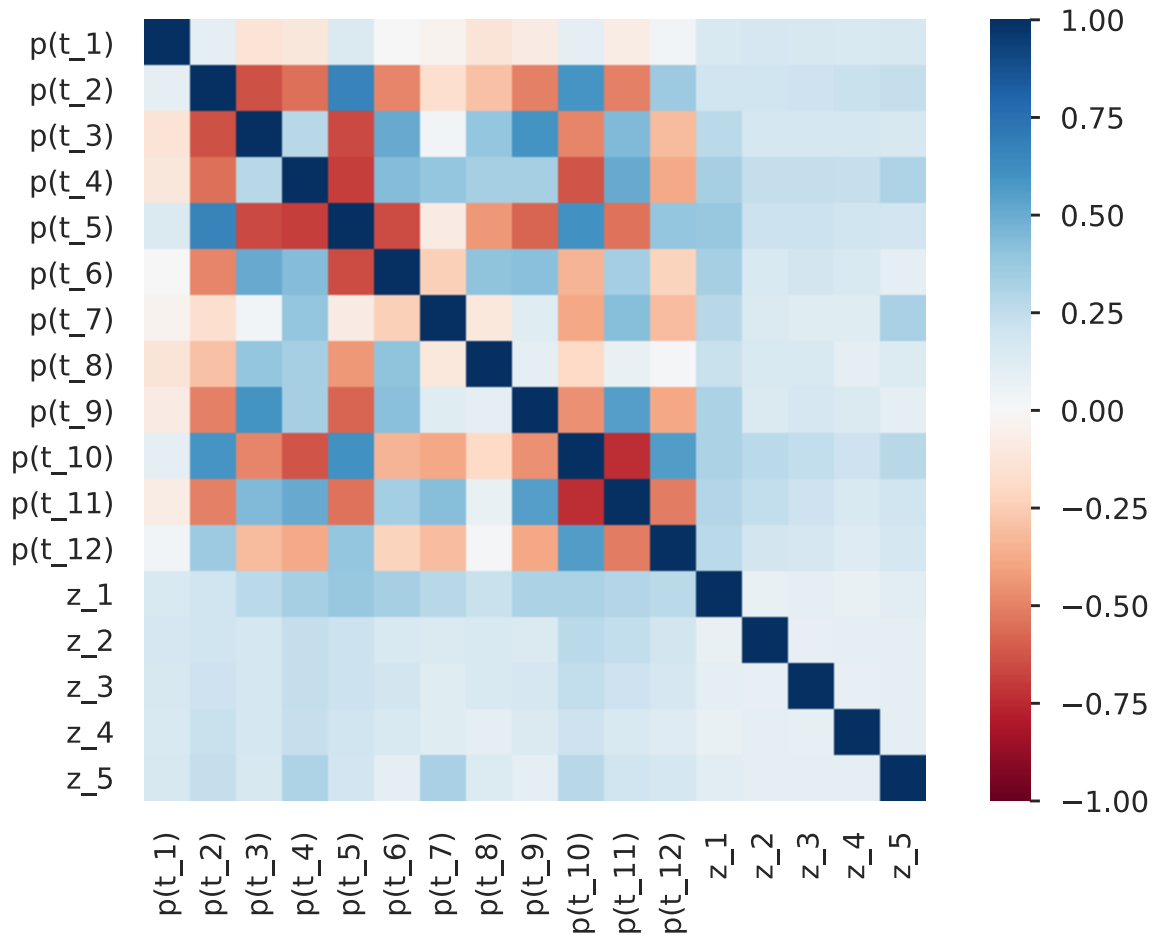


Figure 7.2: Correlation plot for Transformed Time Dataset.

Table 7.3: Correlation matrix for Transformed Time Dataset

	$p_N(t_1)$	$p_N(t_2)$	$p_N(t_3)$	$p_N(t_4)$	$p_N(t_5)$	$p_N(t_6)$	$p_N(t_7)$	$p_N(t_8)$	$p_N(t_9)$	$p_N(t_{10})$	$p_N(t_{11})$	$p_N(t_{12})$
z_1^*	0.154	0.196	0.271	0.333	0.375	0.332	0.275	0.221	0.319	0.320	0.294	0.272
z_2^*	0.169	0.199	0.178	0.241	0.218	0.151	0.143	0.151	0.142	0.271	0.247	0.184
z_3^*	0.159	0.206	0.179	0.236	0.218	0.185	0.123	0.156	0.168	0.250	0.207	0.168
z_4^*	0.153	0.223	0.178	0.228	0.197	0.151	0.120	0.096	0.141	0.210	0.150	0.131
z_5^*	0.157	0.236	0.160	0.305	0.190	0.100	0.323	0.135	0.088	0.277	0.196	0.173

Figure 7.3 shows a comparison of correlation plots at various contamination thresholds. This involved testing whether changing it from 0 to a specific value would increase correlation. This approach also makes it possible to simulate a real-world scenario where the case being considered is not entirely ideal. Based on obtained results, it can be concluded that the optimal threshold is **10%**.

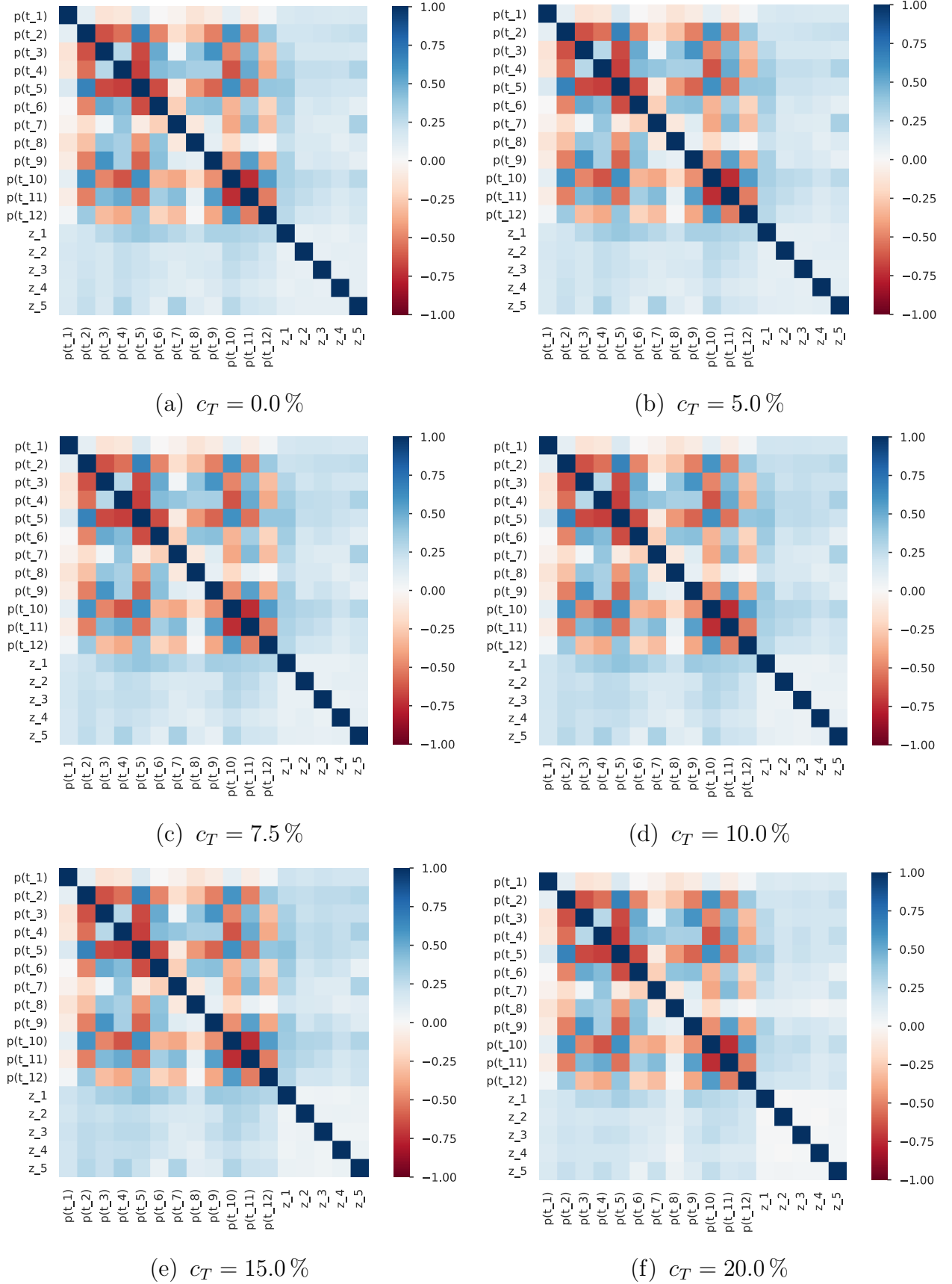
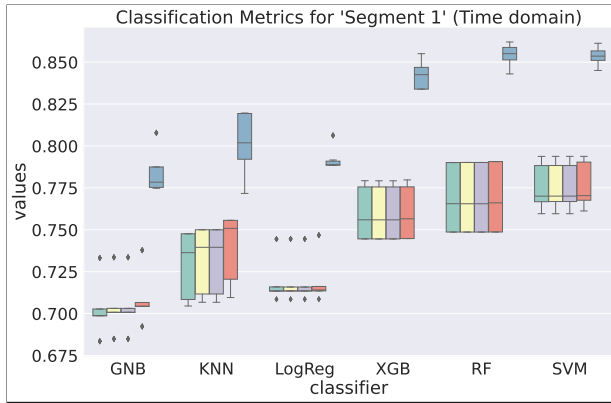


Figure 7.3: Correlation plots with different contamination thresholds c_T .

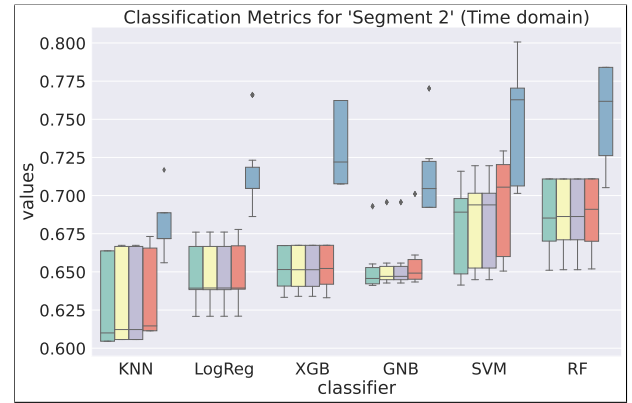
The transformed dataset was assessed for the effectiveness of simple classifiers in predicting contaminated segments using a set of 12 features. Similar comparisons have been previously made in [24]. Logistic Regression (LogReg), Random Forest (RF), K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Gaussian Naive Bayes (GNB), and Extreme Gradient Boosting (XGB) were utilized as classifiers for this task. Each classifier was trained and evaluated on both the train and test data to determine its accuracy in predicting contaminated segments.

To study the classification accuracy, a group of the following five metrics was chosen:

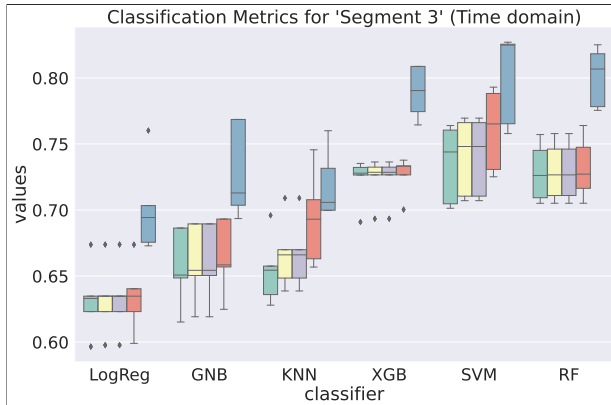
1. ***‘accuracy’***: measures the classifier’s ability to correctly identify instances as either contaminated or non-contaminated.
2. ***‘precision_weighted’***: calculates the weighted average of the precision for each class. This demonstrates how accurately the classifier predicted contaminated segments without falsely classifying non-contaminated ones.
3. ***‘recall_weighted’***: computes the weighted average of the recall for each class. A recall score indicates the prediction of a large proportion of contaminated segments without missing many actual contaminated segments.
4. ***‘f1_weighted’***: estimates the weighted harmonic mean of precision and recall for each class. A high F1 score suggests that the classifier performed well overall, with a strong balance between precision and recall.
5. ***‘roc_auc’***: This metric measures the ability of the classifier to distinguish between the contaminated and non-contaminated segments by plotting the true-positive rate (sensitivity) against the false-positive rate (1-specificity) at various threshold values.



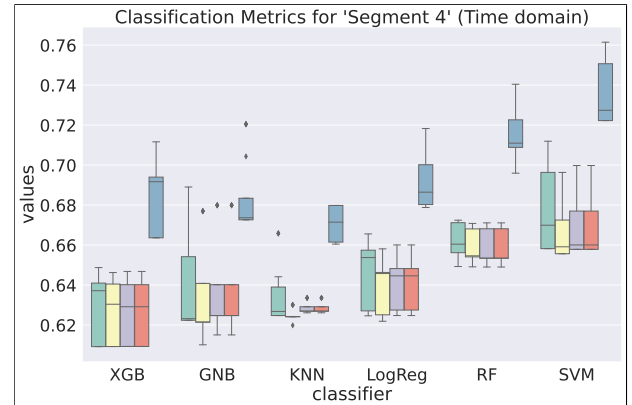
(a) Segment 1.



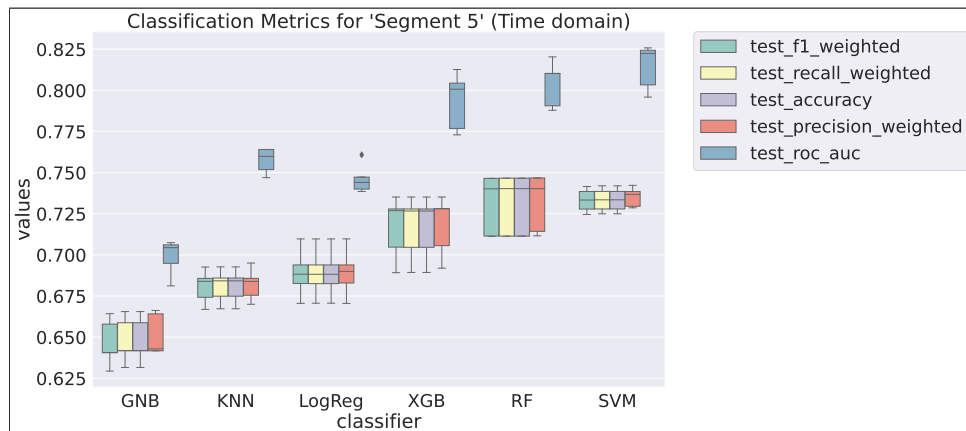
(b) Segment 2.



(c) Segment 3.



(d) Segment 4.



(e) Segment 5.

Figure 7.4: Classification metrics comparison.

The results of the evaluation of the simple classifiers on the current dataset indicate that SVM performs better, achieving an accuracy rate of 75–80%. Therefore, this implies that SVM is the most effective classifier for predicting contaminated segments in the time domain using this particular set of features.

RF and XGB were found to be the next most effective classifiers, achiev-

ing similar performance with accuracy rates of 70–75%. These results demonstrate the potential of ensemble methods in predicting contaminated segments, as RF and XGB are both ensemble-based algorithms that utilize multiple decision trees to make predictions.

The remaining classifiers, including Logistic Regression, KNN, and GNB, showed relatively worse performance with accuracy rates of 60–70%. It implies that these classifiers may not be as effective as SVM, RF, and XGB but still can be used in some cases.

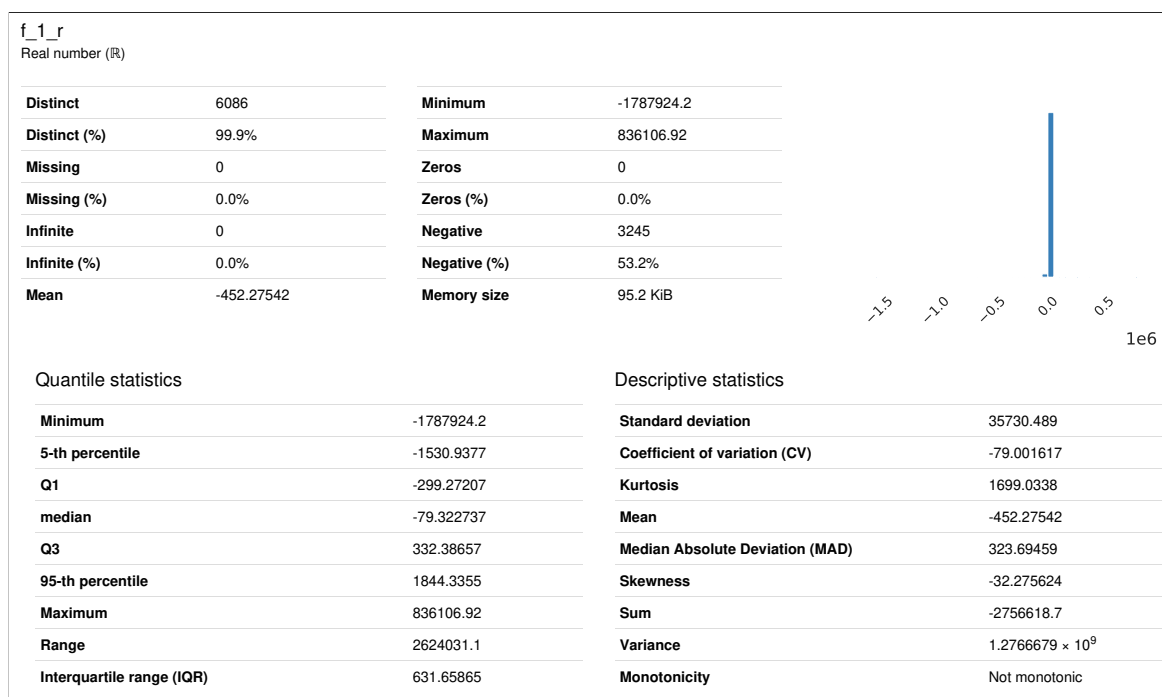
The evaluation results also showed that segments 1, 3, and 5 were better recognized by all classifiers, as expected from the correlation diagram. It also indicates that these segments may have a stronger relationship with the presence of contamination and that the selected features can effectively reflect this relationship.

7.2 Frequency Domain dataset exploration and classification

Table 7.4 shows the initial data obtained after the dataset generation for the Frequency Domain problem, where z_i - contamination level in i -th segment, $i = 1..5$, and $p_{real/imag}(f_j)$ - value of of pressure on a receiver at frequency $f_j, j = 1..4$. Similarly to the previous section, the values for the segments were also transformed according to the contamination threshold $c_T = 0$.

Table 7.4: Initial Frequency Dataset

	z_1^*	z_2^*	z_3^*	z_4^*	z_5^*	$p_{real}(f_1)$	$p_{imag}(f_1)$...	$p_{real}(f_4)$	$p_{imag}(f_4)$
0	False	False	False	False	False	4.05e+02	-2.52e-04	...	5.19e+02	-3.20e-04
1	False	False	True	True	False	-4.06e+02	-2.61e-05	...	1.22e+03	-2.07e-02
2	True	True	False	False	False	3.48e+01	-7.50e-06	...	2.28e+03	-5.13e-02
...
6996	False	False	False	True	False	3.13e+02	-1.86e-04	...	1.38e+03	1.12e-02
6997	False	False	False	False	True	4.72e+02	-3.83e-04	...	-9.99e+02	-5.01e-03
6998	False	False	False	True	False	-8.07e+02	-1.65e-04	...	-1.98e+02	-3.37e-03

Figure 7.5: Profile report for $p_{real}(f_1)$ with outliers.

The study of the value distribution of the obtained results showed some anomalies. For example, Figure 7.5 illustrates the descriptive statistics for the first value. As can be seen, certain outliers appeared during the calculations. To eliminate them, a detailed data review was performed, which led to the following assumption: ***A row is considered to have an outlier if at least one value $|p_{real}(f_i)| > 5000, i = 1..4$.***

The updated descriptive statistics are presented in Figure 7.6, and it shows considerable improvements.

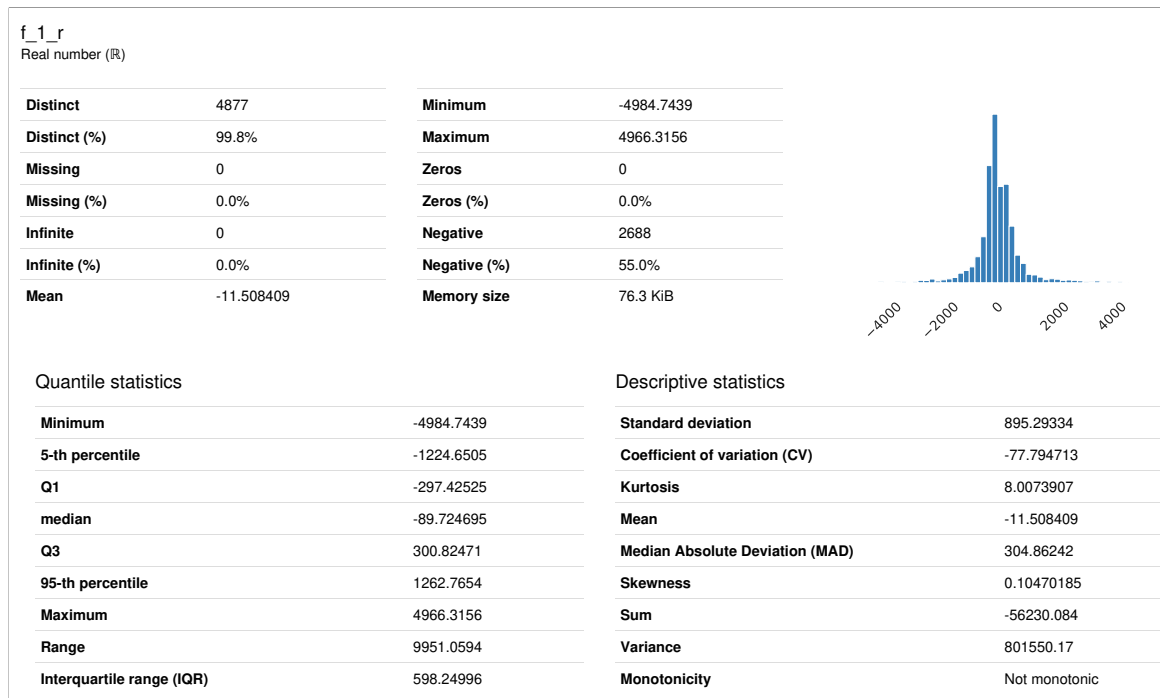


Figure 7.6: Profile report for $p_{real}(f_1)$ without outliers.

Following the steps in Section 7.1, the dataset was normalized using the same principle, and the results can be seen in Table 7.5, where $p_{real/imag}^N(f_j)$ are the transformed values.

Table 7.5: Transformed Frequency Dataset

	z_1^*	z_2^*	z_3^*	z_4^*	z_5^*	$p_{real}^N(f_1)$	$p_{imag}^N(f_1)$...	$p_{real}^N(f_4)$	$p_{imag}^N(f_4)$
0	False	False	False	False	False	0.465532	0.141150	...	0.395153	0.065420
1	False	False	True	True	False	-0.440723	0.229752	...	1.119857	-0.059636
2	True	True	False	False	False	0.051675	0.237055	...	2.212240	-0.247145
...
6992	True	False	False	False	False	-0.065389	0.244164	...	-1.001915	0.042173
6993	False	True	False	False	False	-0.161434	0.242397	...	0.016651	0.049933
6997	False	False	False	False	True	0.540650	0.089389	...	-1.171320	0.036649

Figure 7.7 and Table 7.6 show the correlation coefficients between variables in a dataset. Evaluating this diagram, it can be noted that the correlation of segments is higher at lower propagation frequencies. This is because waves at lower frequencies are more resistant to interference. It can also be seen that

the imaginary part of the solutions carries less information than the real part. And, like in the example with the time domain, the first segment has the highest correlation value.

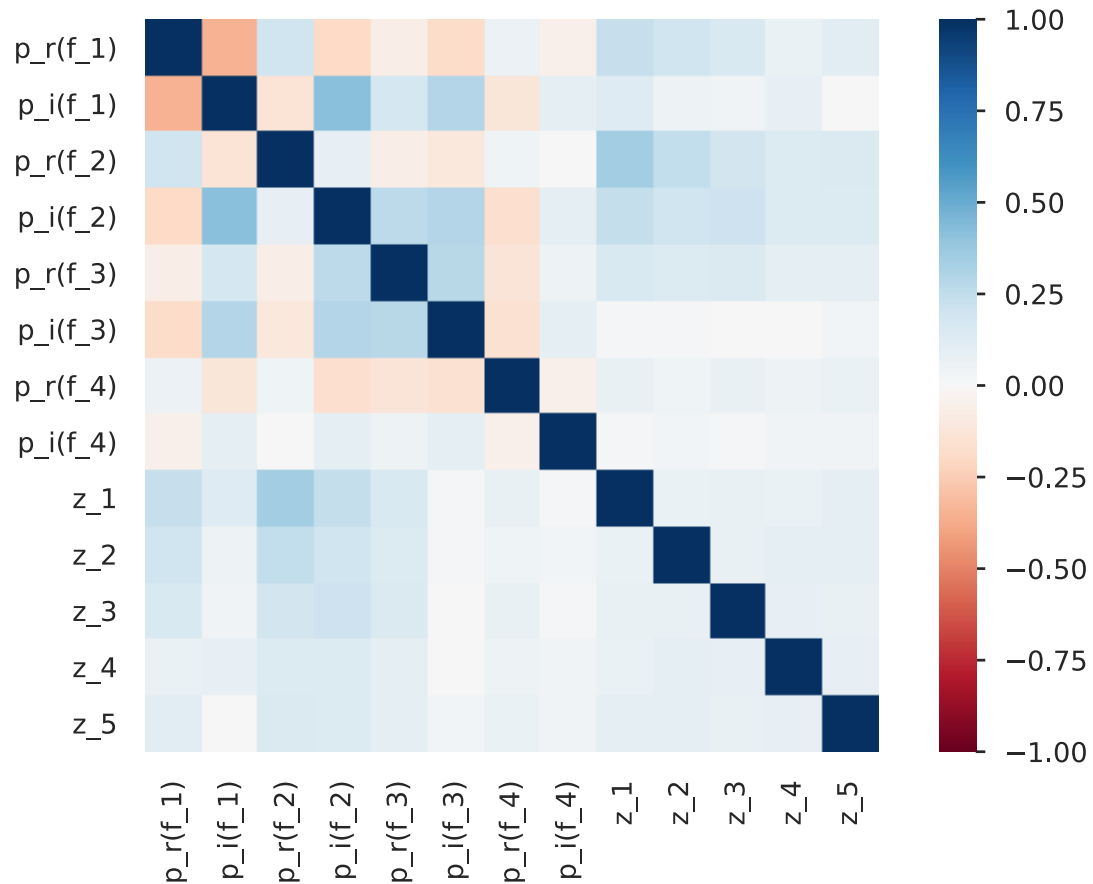


Figure 7.7: Correlation plot for transformed Frequency Dataset.

Table 7.6: Correlation matrix for transformed Frequency Dataset

	$p_{real}^N(f_1)$	$p_{imag}^N(f_1)$	$p_{real}^N(f_2)$	$p_{imag}^N(f_2)$	$p_{real}^N(f_3)$	$p_{imag}^N(f_3)$	$p_{real}^N(f_4)$	$p_{imag}^N(f_4)$
z_1^*	0.227	0.131	0.346	0.235	0.149	0.013	0.072	0.021
z_2^*	0.195	0.049	0.249	0.196	0.136	0.014	0.043	0.027
z_3^*	0.151	0.035	0.187	0.204	0.141	0.000	0.071	0.010
z_4^*	0.069	0.082	0.133	0.137	0.095	0.000	0.051	0.035
z_5^*	0.107	0.000	0.144	0.140	0.092	0.029	0.064	0.032

Figure 7.8 illustrates the correlation diagram, which shows that, also, for this problem, it is optimal to take a contamination threshold equal to 7.5 or 10 percent but to be consistent with the previous case, **10%** was chosen.

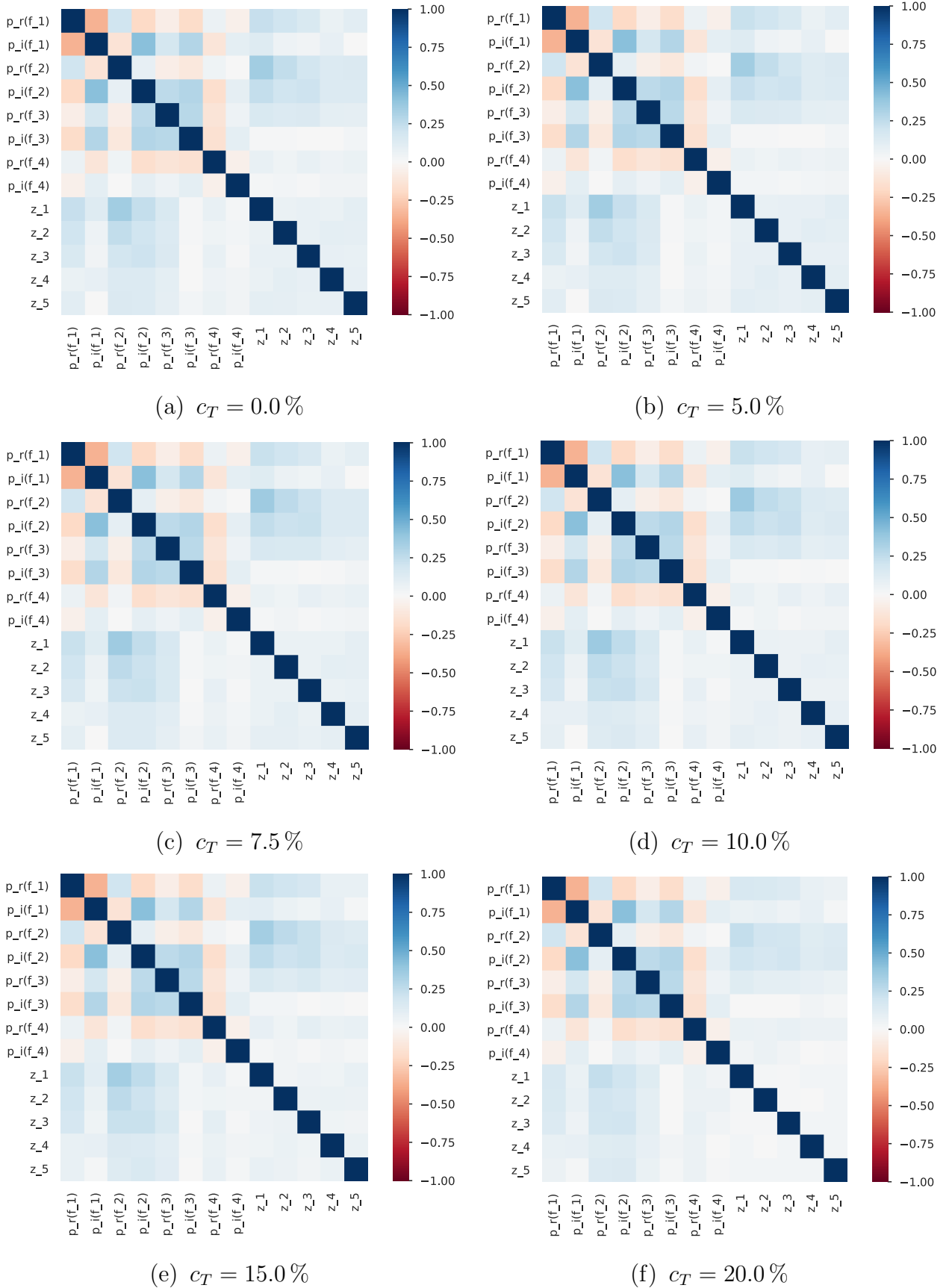
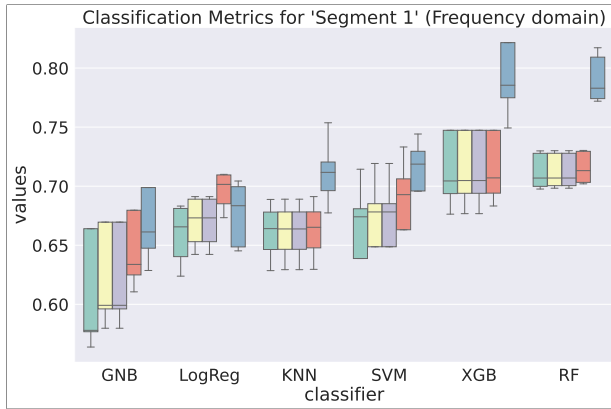
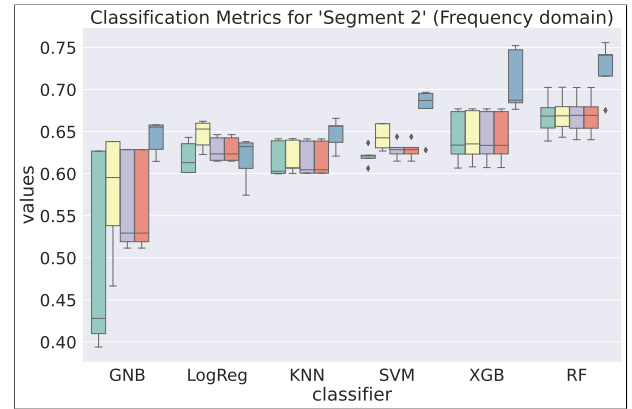


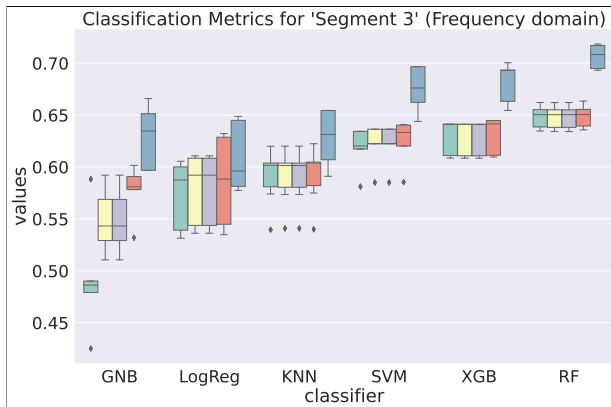
Figure 7.8: Correlation plots with different contamination thresholds c_T .



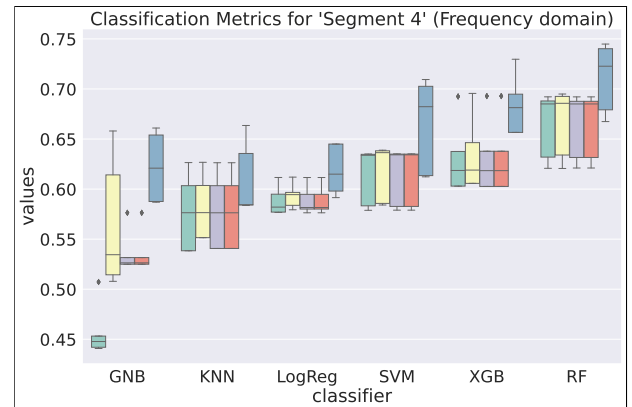
(a) Segment 1.



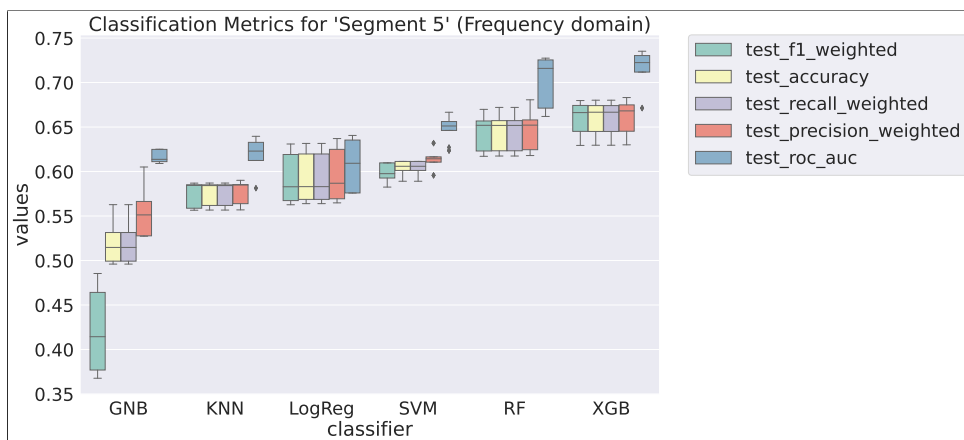
(b) Segment 2.



(c) Segment 3.



(d) Segment 4.



(e) Segment 5.

Figure 7.9: Classification metrics comparison.

The same set of comparisons was conducted for the frequency domain problem, see Figure 7.9. In this case, SVM, RF, and XGB demonstrated good performance, achieving accuracy rates of 65–70%. These findings suggest that these classifiers are robust and effective for predicting contamination based on the selected features.

On the other hand, the remaining classifiers showed relatively inferior performance, with accuracy rates ranging between 50% and 60%. This indicates that these classifiers may not be as well-suited for handling the frequency domain problem in predicting contaminated segments.

It can be observed that the performance of all classifiers degraded slightly from the first segment to the last one. This degradation implies that the classifiers may struggle to accurately classify the latter segments, potentially due to the lack of information about temporal changes, as it was in the time domain problem.

7.3 Prediction with neural networks

After conducting classification on both the time and frequency domain problems, it was decided to test more complex tools to check if they could further improve the prediction accuracy. As a result, neural networks were selected as the next phase of research. By incorporating them, the aim was to leverage the power of deep learning and its ability to capture complex relationships and patterns in the data.

The neural network models' architectures were created and defined using Keras, a high-level neural networks API. It has a user-friendly interface that allows for easy configuration of various layers, activation functions, and optimization algorithms. TensorFlow was used as the backend engine for executing the computations involved in training and evaluating the models. TensorFlow is an ideal choice for implementing deep learning algorithms as it provides efficient numerical computation and automatic differentiation capabilities[25, 26].

To work with the neural network, the working dataset was divided into three subsets. The training set consisted of 80%, and the rest was split between the validation and test sets in the ratio of 60% to 40%, respectively.

The goal of this neural network was to predict the presence of contamina-


```

22 history = model.fit(X_train,
23                     y_train,
24                     epochs=epochs,
25                     batch_size=batch_size,
26                     validation_data=(X_val, y_val),
27                     callbacks=[early_stopping, TqdmCallback(verbose=1)],
28                     verbose=0)

```

Listing 7.1: Description of ‘Model 1’

Figure 7.10 shows the training process of this model. During the analysis, it was also seen that neural networks performed poorly when there was more than one contaminated segment in the record. In this case, the prediction results reached only **47.31%** for the test set.

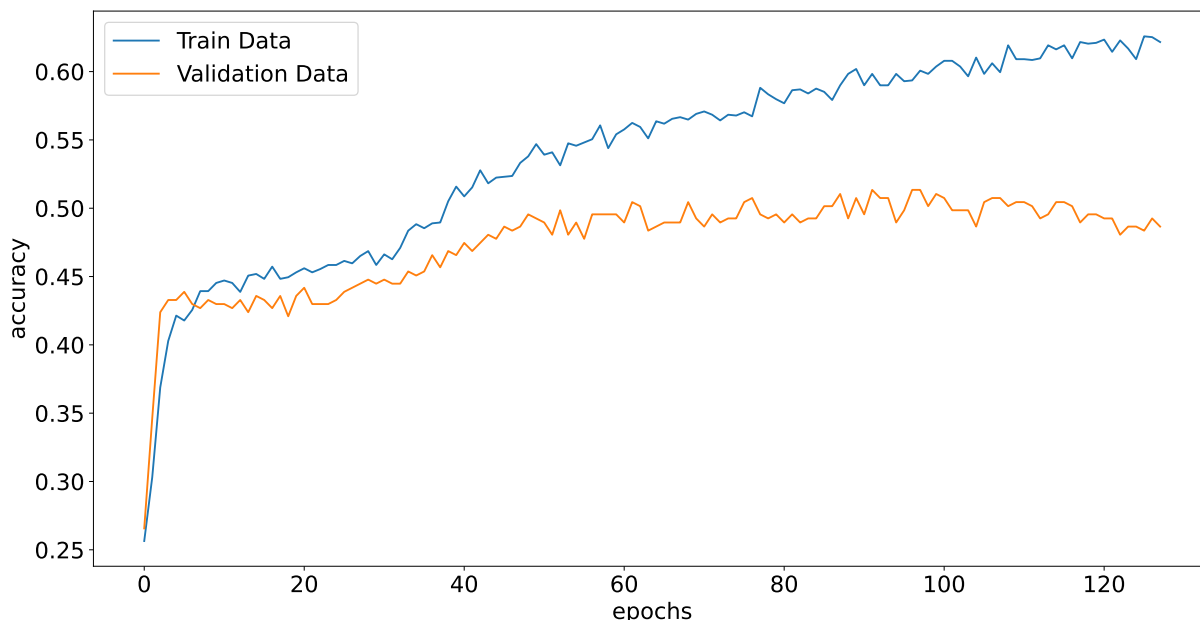


Figure 7.10: Training history for ‘Model 1’.

On the other hand, when the neural network was trained on data where only the single segment was contaminated, it gave slightly better results, see Figure 7.11. As can be noticed, it managed to predict with a probability close to 60%. This model produced an accuracy of **59.63%** on the test data, which is a rather good result given the complexity of the task.

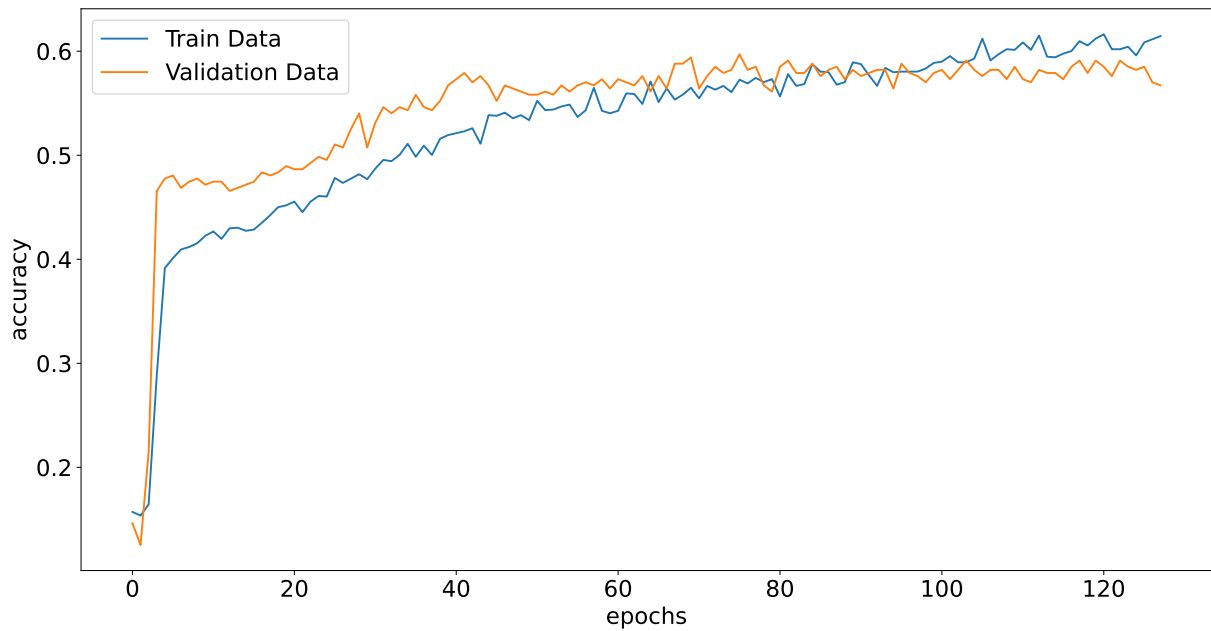


Figure 7.11: Training history for ‘Model 1‘ with single contaminated segment.

Listing 7.2 shows another variation of the neural network model with a recurrent layer. This solution is based on the fact that the values of the receiver are obtained sequentially over a certain time period, thereby allowing consideration of the temporal properties of the data.

```

1  model = Sequential([
2      Dense(32, input_shape=[X.shape[1], 1], activation='relu'),
3      Dropout(0.2),
4      SimpleRNN(64, activation='relu'),
5      Dropout(0.2),
6      Dense(16, activation='relu'),
7      Dropout(0.2),
8      Dense(8, activation='relu'),
9      Dense(5, activation=tf.keras.activations.hard_sigmoid)
10 ])
11
12 model.compile(loss='mean_squared_error',
13               optimizer='adam',
14               metrics=['accuracy'])
15
16 # train parameters
17 epochs=256
18 batch_size=64
19

```

```

20 early_stopping = EarlyStopping(monitor='val_loss',
21                               patience=64,
22                               restore_best_weights=True)
23
24 history = model.fit(X_train,
25                   y_train,
26                   epochs=epochs,
27                   batch_size=batch_size,
28                   validation_data=(X_val, y_val),
29                   callbacks=[early_stopping, TqdmCallback(verbose=1)],
30                   verbose=0)

```

Listing 7.2: Description of ‘Model 2’

The second model did not show any improvement during training, see Figure 7.12. Another drawback was that it took longer to train this model, and the prediction results were only **42.74%**.

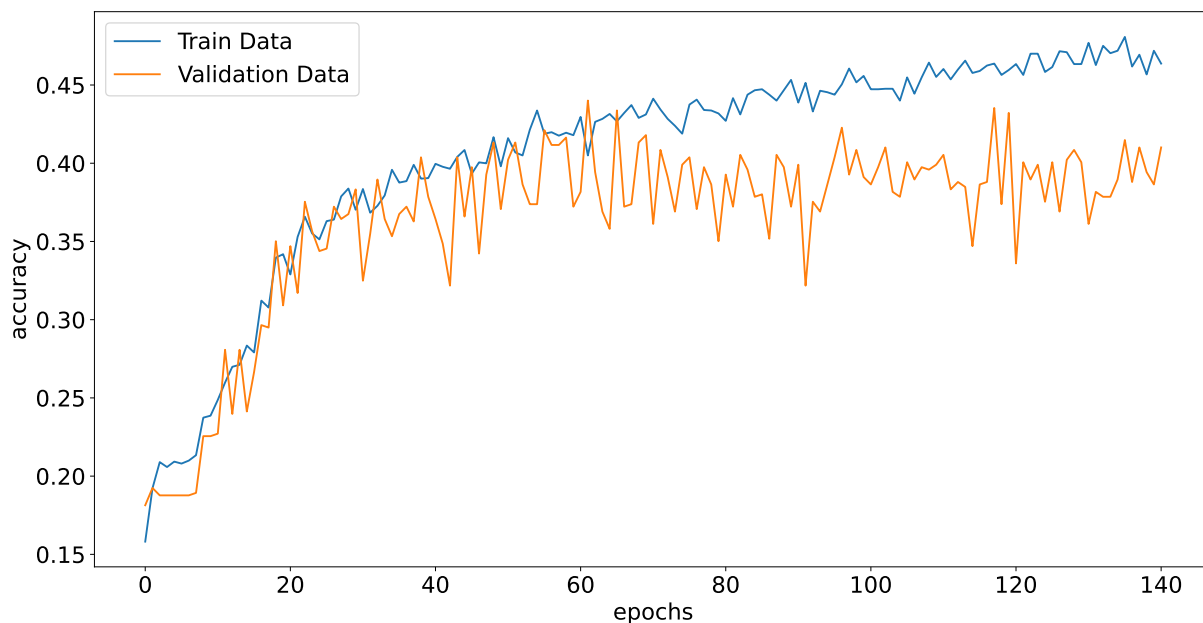


Figure 7.12: Training history for ‘Model 2’.

In Figure 7.13, some improvements in the results can be seen once again for the model with a single contaminated segment with train accuracy is equal to **49.98%**.

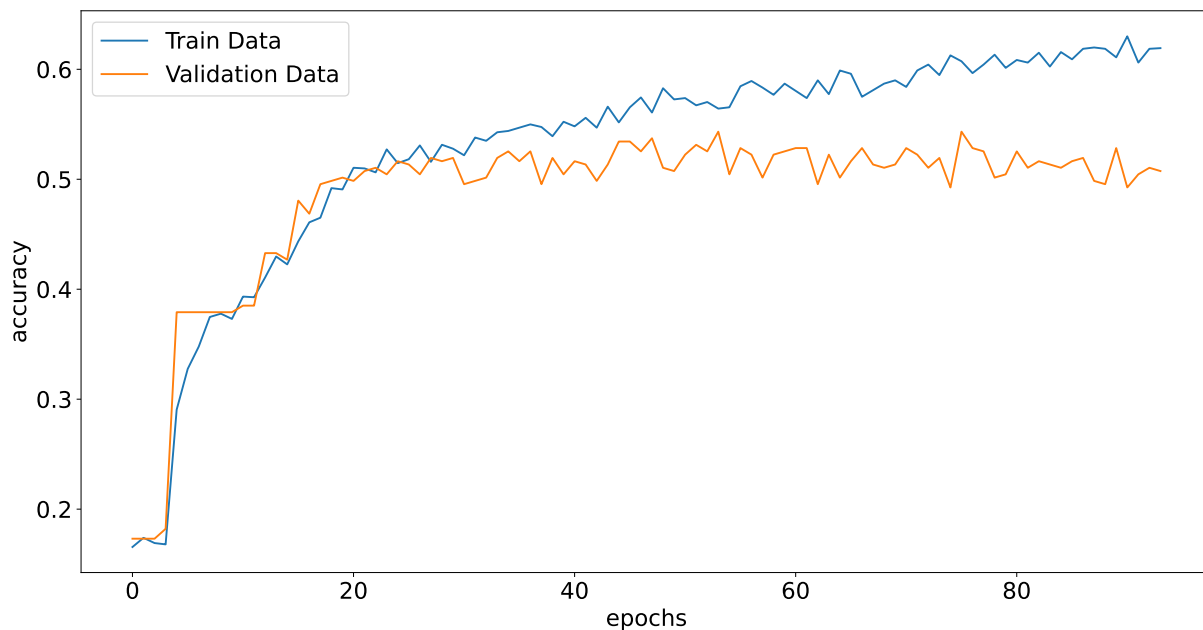


Figure 7.13: Training history for ‘Model 2’ with single contaminated segment.

For the frequency domain problem, there were also attempts to build a neural network. For this purpose, ‘*Model 1*’, shown in Listing 7.1, was used. Figure 7.14 illustrates that the training results are unsatisfactory. This situation can be justified by the fact that the specifics of the case do not allow capturing temporal changes, and the values have a low correlation, as shown in Table 7.6. The value of test accuracy in this case is equal to **44.12%**.

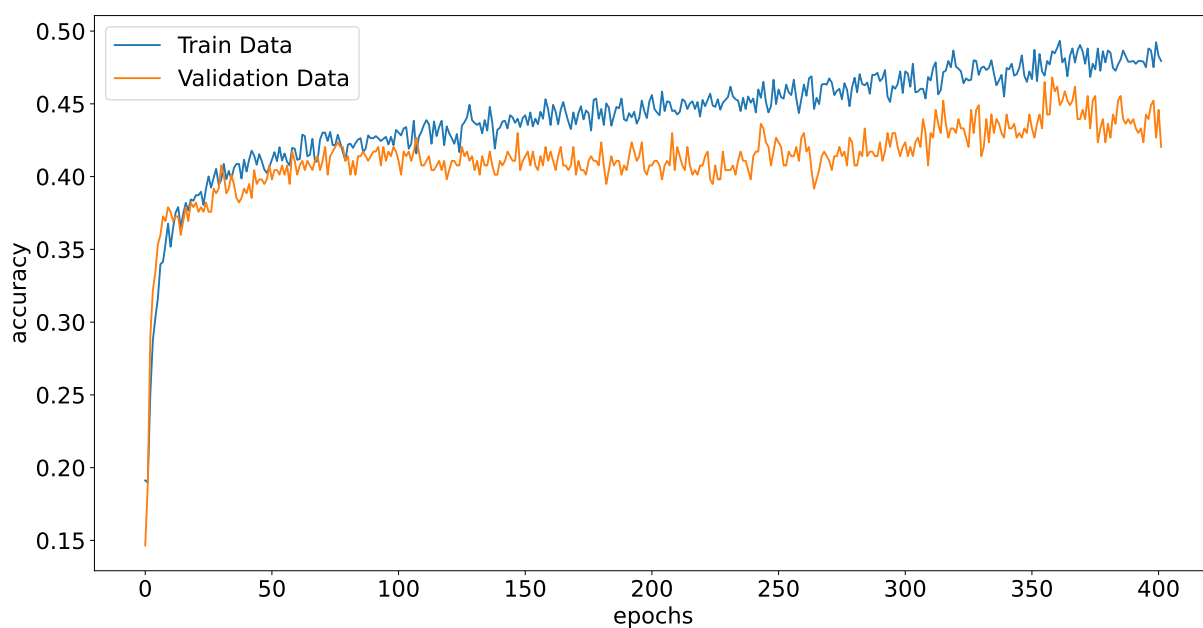


Figure 7.14: Training history for ‘Model 1’.

Considering the case of a single contaminated segment, it is possible to achieve better results. This process can be seen in Figure 7.15. And the final accuracy value during testing was **52.33%**.

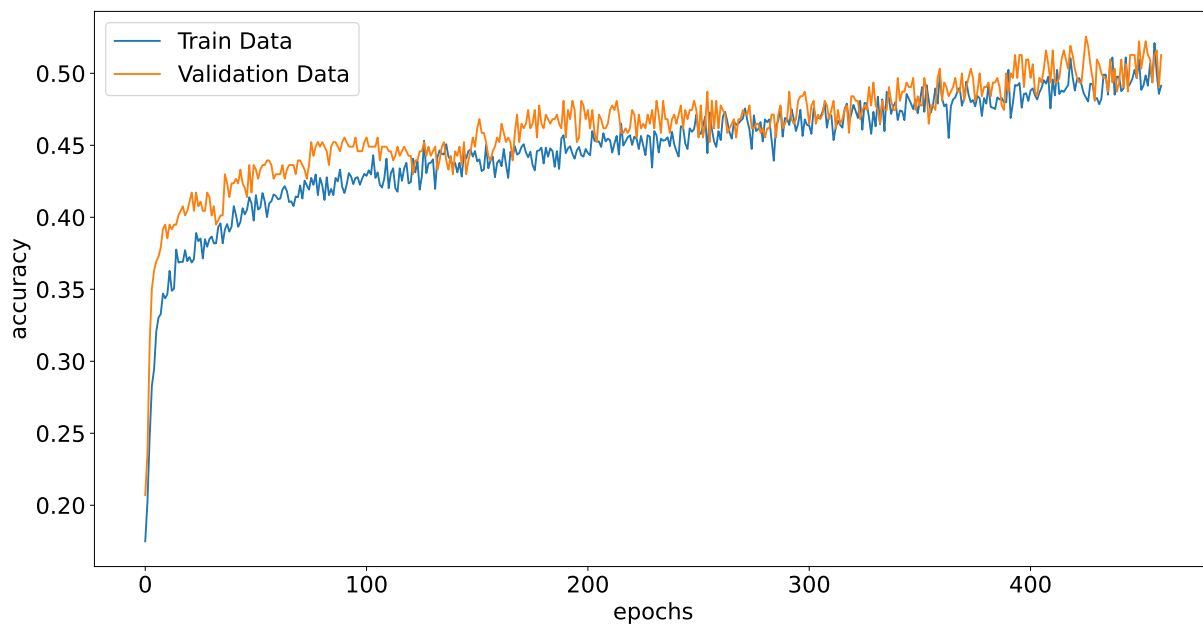


Figure 7.15: Training history for ‘Model 1‘ with single contaminated segment.

Chapter 8

Conclusions

8.1 Summary

In this thesis, a study on the application of the finite element method and neural networks was conducted. This process was logically divided into several key steps: studying the mathematical problem, solving the system, building a new model, and producing data, as well as analyzing the resulting dataset and application of machine learning.

The first step was to review the literature and describe the mathematical problem. Thus, the key research in this area was highlighted and explained the motivation behind writing the paper. Then the mathematical problem and its variation formulation were presented. And to show the correctness of the problem, the paper also includes energy equations.

After a strict formulation of the problem, the second step was to study the process of calculations. Therefore, the original problem was divided into two subproblems: Time Domain and Frequency Domain. For each of them, a discretization scheme for solving by the finite element method was presented.

Special attention should be given to the FEniCSx library, which was used for all calculations. Thus, it was possible to significantly reduce the time for implementing the solver and take advantage of parallel computing. All the

code, including Jupyter Notebooks, Python files, and generated data, is publicly available in the GitHub repository mentioned in this paper. The repository also contains the source code used in the following sections.

Next, examples of solved problems in one- and two-dimensional cases for two media with different physical properties were presented. These results show how acoustic waves behave in ideal media and give an understanding of the physical nature of the problem.

The following step was to apply the findings to a practical example. Therefore, it was assumed that there exists contamination in a certain medium. To determine it, the previous problem could be used, meaning that there is a generator at one end and a receiver at the other. Thus, values about contamination level were collected by solving this problem many times and modeling different scenarios.

The paper presents a method for generating different cases of contamination, in this case, air bubbles. This algorithm involves several sequential steps using different distribution functions. In this way, it is possible to cover the lack of data from real-world examples.

The last step, which covered a big part of the research, was to analyze the data and build neural networks. Based on this information, it is possible to derive certain conclusions about the feasibility of using machine learning algorithms and neural networks.

A descriptive study of the dataset showed the dependence of some parameters on others, the influence of the receiver's data on media contamination, and identified anomalies that could have occurred during the computations.

Therefore, for two subproblems, classifiers were used to determine the presence of contamination that exceeds a certain critical value. From the results, it can be concluded that it is possible to classify data in the time domain with a slight but higher accuracy than in the frequency domain.

As for the use of neural networks, in other words, the transformation

of the problem into a multiclassification task, the current results did not provide much accuracy. Thus, it can be concluded that the use of more complex algorithms did not lead to improved results.

Still, the usage of developed in this thesis neural networks simplifies real-case scenarios by providing instant data analysis and decision-making capabilities without requiring excessive memory and resources. The compact architecture allows for their deployment on various devices, enabling timely and resource-efficient operations.

8.2 Future research directions

The following paragraphs provide suggestions that can be used in future research.

Explore and develop new algorithms for generating contamination inside a domain. The current approach assumes non-homogeneous contamination with bubbles, but investigating alternative algorithms could provide diverse scenarios, enhancing the accuracy and applicability of the simulations.

Take outliers into account when analyzing datasets. While outliers are often considered noise, they may contain valuable information that can contribute to a better understanding of the contamination problem. Developing techniques to identify and analyze outliers could unveil unique insights and improve the overall data analysis process.

Investigate and optimize neural networks. Research can focus on exploring new architectures, activation functions, or optimization algorithms to enhance performance, efficiency, and applicability.

References

- [1] Kinsler Lawrence and Austin R Frey. *Fundamentals of Acoustics*. New York: Wiley, 1962.
- [2] L. D. Landau and E. M. Lifshitz. *Fluid mechanics by l.d. landau and e.m. lifshitz (Translated from the Russian by Sykes, J. B. and Reid, W. H.)* Pergamon Press, 1959.
- [3] Alfredo Bermúdez, Rodolfo Rodriguez, and Duarte Santamarina. “Two discretization schemes for a time-domain dissipative acoustics problem”. In: *Mathematical Models and Methods in Applied Sciences* 16 (Oct. 2006). DOI: 10.1142/S0218202506001637.
- [4] V.M. Voytovych, V.M. Horlatch, Y.V. Kondratyuk and G.A. Shynkarenko. “Displacement-based modelling of acoustic fluid-structure interaction problem”. In: *Applied problems of mechanics and mathematics APMM NASU* 2 (2005), pp. 108–118. URL: <https://ami.lnu.edu.ua/wp-content/uploads/2013/10/Displacement-Based-Modelling-of-Acoustic-Fluid-Structure-Interaction-Problem.pdf>.
- [5] I. Klymenko V. Horlatch and G. Shynkarenko. “Formulation and well-posedness of the variational problem of viscous heat-conducting fluid acoustics”. en. In: *Journal of Numerical and Applied Mathematics* 3.109 (2012), pp. 53–71.
- [6] Ivan Dyyak, Vitaliy Horlatch, and Heorhiy Shynkarenko. “Formulation and Numerical Analysis of Acoustics Problems in Coupled Thermohy-

- droelastic Systems”. In: *2019 XXIVth International Seminar/Workshop on Direct and Inverse Problems of Electromagnetic and Acoustic Wave Theory (DIPED)*. 2019, pp. 168–171. DOI: 10 . 1109 / DIPED . 2019 . 8882584.
- [7] Jacques Louis Lions Georges Duvaut. *Inequalities in Mechanics and Physics*. Grundlehren der mathematischen Wissenschaften. Springer Berlin Heidelberg, 1976. DOI: 10 . 1007 / 978 - 3 - 642 - 66165 - 5.
- [8] COMSOL Multiphysics. “Introduction to COMSOL multiphysics®”. In: *COMSOL Multiphysics, Burlington, MA, accessed Feb 9 (1998)*, p. 2018.
- [9] Christopher Greenshields. *OpenFOAM v10 User Guide*. London, UK: The OpenFOAM Foundation, 2022. URL: <https://doc.cfd.direct/openfoam/user-guide-v10>.
- [10] MATLAB. *version 7.10.0 (R2010a)*. Natick, Massachusetts: The MathWorks Inc., 2010.
- [11] Anders Logg, Kent-Andre Mardal, and Garth Wells. *Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book*. Springer Publishing Company, Incorporated, 2012. ISBN: 3642230989.
- [12] Yu Ma. *Machine learning in ocean applications : wave prediction for advanced controls of renewable energy and modeling nonlinear viscous hydrodynamics*. Massachusetts Institute of Technology, 2020. URL: <https://dspace.mit.edu/handle/1721.1/127057f>.
- [13] John Cristian Borges Gamboa. “Deep Learning for time-series analysis”. In: (Jan. 2017). arXiv: 1701.01887 [cs.LG].
- [14] Anastasiia Bondarenko and Vitaliy Horlatch. “Mathematical modeling of acoustic wave propagation in viscous fluid”. In: *International Student Scientific Conference on Applied Mathematics and Computer Sciences ISSCAMCS (May 2022)*, pp. 50–53. URL: <https://ami.lnu.edu.ua/wp-content/uploads/2022/05/ISSCAMCS-2022.pdf>.

- [15] Volodymyr Milchanovskyi and Vitaliy Horlatch. “Mathematical modeling and analysis of stationary acoustic wave propagation in viscous fluid”. In: *International Student Scientific Conference on Applied Mathematics and Computer Sciences ISSCAMCS* (May 2022), pp. 54–58. URL: <https://ami.lnu.edu.ua/wp-content/uploads/2022/05/ISSCAMCS-2022.pdf>.
- [16] R. Courant, K. Friedrichs, and H. Lewy. “On the Partial Difference Equations of Mathematical Physics”. In: *IBM Journal of Research and Development* 11.2 (1967), pp. 215–234. DOI: 10.1147/rd.112.0215.
- [17] M. W. Scroggs et al. “Construction of arbitrary order finite element degree-of-freedom maps on polygonal and polyhedral cell meshes”. In: *ACM Transactions on Mathematical Software* (2022). DOI: 10.1145/3524456.
- [18] Volodymyr Milchanovskyi. “Analysis of PETSc solvers for dissipative acoustics problems using FEniCSx”. In: *Course project* (June 2022).
- [19] Volodymyr Milchanovskyi. “Numerical Modeling of the Acoustics Problems”. In: *Course project* (June 2020).
- [20] Volodymyr Milchanovskyi. “Numerical Modeling of Acoustic Wave Propagation in Thermohydroelastic Systems”. In: *Course project* (June 2021).
- [21] Marshall Copeland et al. *Microsoft Azure: Planning, Deploying, and Managing Your Data Center in the Cloud*. 1st. USA: Apress, 2015. ISBN: 1484210441.
- [22] Volodymyr Milchanovskyi. *Dissipative-Acoustic*. <https://github.com/DthRazak/Dissipative-Acoustic>. 2023.
- [23] YData. *ydata-profiling*. <https://github.com/ydataai/ydata-profiling>. 2023.
- [24] Cole. *Quickly Compare Multiple Models*. <https://towardsdatascience.com/quickly-test-multiple-models-a98477476f0>. 2020.

- [25] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*. 2016. arXiv: 1603.04467 [cs.DC].
- [26] Antonio Gulli and Sujit Pal. *Deep learning with Keras*. Packt Publishing Ltd, 2017.