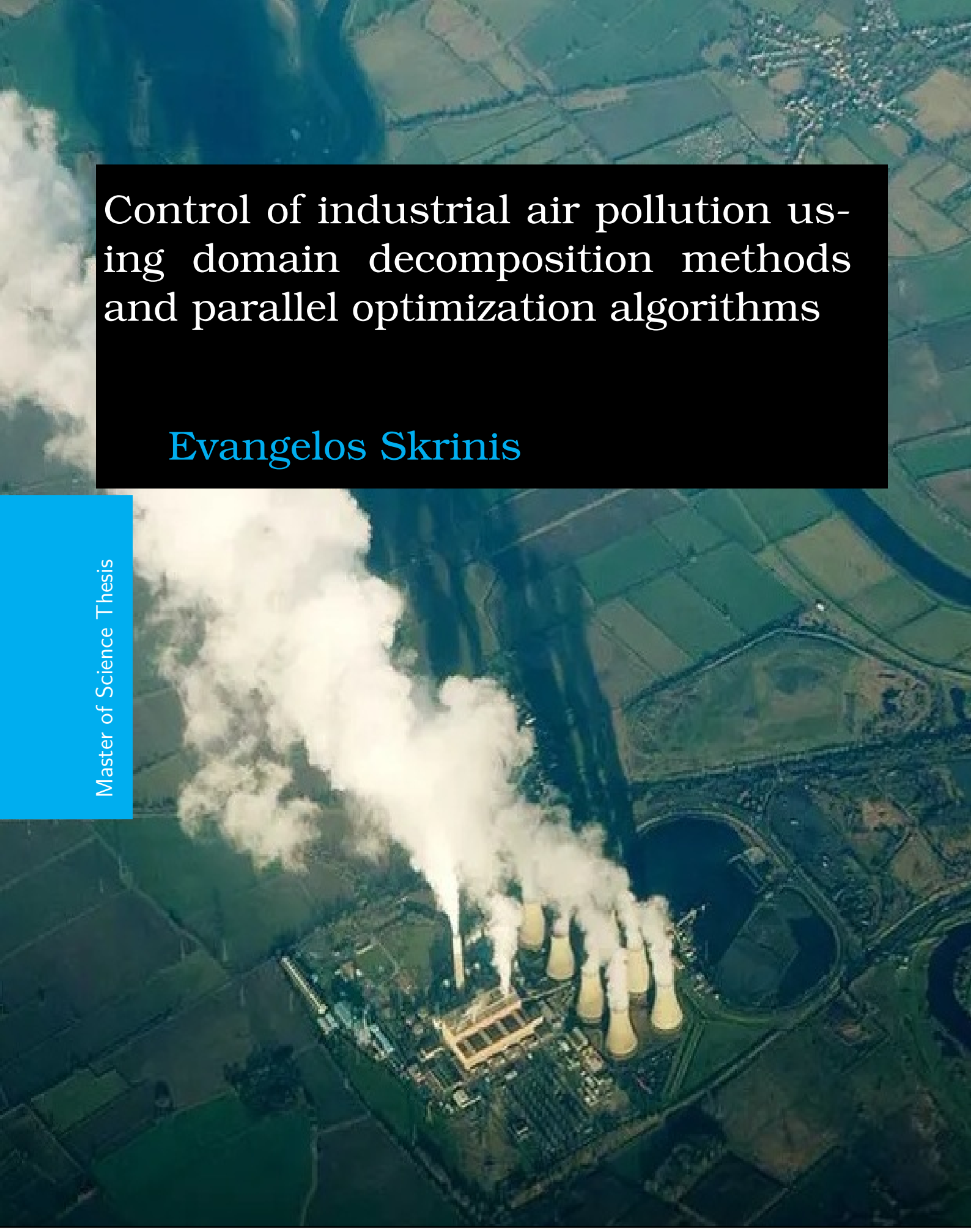


# Control of industrial air pollution using domain decomposition methods and parallel optimization algorithms

Evangelos Skrinis

Master of Science Thesis

TU Delft Delft University of Technology

Delft Center for Systems and Control

# Control of industrial air pollution using domain decomposition methods and parallel optimization algorithms

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

Evangelos Skrinis

September 11, 2026

Faculty of Mechanical Engineering (ME) · Delft University of Technology


Delft Center for
Systems and Control

Delft University of Technology
Department of
Delft Center for Systems and Control (dcsc)

The undersigned hereby certify that they have read and recommend to the Faculty of Mechanical Engineering (ME) for acceptance a thesis entitled

Control of industrial air pollution using domain decomposition methods and parallel optimization algorithms

by

Evangelos Skrinis

in partial fulfillment of the requirements for the degree of

Master of Science Systems and Control

Dated: September 11, 2026

Supervisor:

Dr. D. (Dimitris) Boskos

Committee::

Dr. O. (Omid) Nejadseyfi

# Abstract

Air-pollution is one of the most critical ecological and public health challenges of the 21st century. Its pre-emptive, passive, and reactive control is a fundamental aspect of environmental management, that is aimed at minimising the release of harmful substances and their impact on human health and natural systems. An effective pollution control strategy involves the scientific modelling of the pollution dispersion so as to either prevent pollutants from being emitted or mitigate the effects after their release in specified areas of interest. One mathematical framework suitable for this purpose, includes the solution of partial differential equations in order to accurately simulate the physical phenomenon in question. By coupling this model with an optimisation scheme, it is possible to design targeted interventions, such as the real-time adjustment of emission rates, the optimal placement of industrial facilities, and other related countermeasures. These actions can guarantee that the concentration of a pollutant will be within specified safe margins inside the regions of interest. The main objectives of this thesis are to propose an accurate mathematical model that describes the dispersion of air-pollutants in a wide range of scenarios, and implement it successfully within an appropriate parallel optimisation scheme, while evaluating its convergence under specific assumptions. The formulation of an optimisation methodology under those requirements, will result in a large-scale numerical problems with different properties depending on the application parameters, and with various limitations regarding their numerical solution. In an attempt to overcome them, this thesis examines the use of PDE-constrained optimisation methods in combination with parallel techniques so as to achieve the minimisation of the air-pollutants' effects along with the distribution of the computational burden. More specifically, it will provide insight on the implementation of the augmented Lagrangian technique with the use of a Newton method in the formulation of a robust and accurate optimisation framework. In addition, emphasis will be given in the utilisation of parallel algorithms based on domain decomposition methodologies (DDM) in order to produce faster results, improve scalability, and allow the time-sensitive regulation of the pollutants' emission rate.

# Contents

# List of Figures

# List of Tables

# Preface

Dear reader,

The following essay is my thesis, the final step in the completion of the master's programme Systems and Control at TU Delft.

The completion of my studies at Delft has been neither an easy nor a linear process. There were days when I was overwhelmed and even questioned my choice to pursue this field of study. At the same time, it was an incredibly rewarding experience that pushed me to my limits and forced me to grow as an engineer.

The topics of modelling complicated and multivariable phenomena, which require extensive knowledge from various fields of science, along with the design of control responses based on optimal strategies have always triggered a profound interest in me. Even from my undergraduate studies as a mechanical engineer, I was inclined to dig deeper whenever those problems surfaced in my way. Thanks to my supervisor, professor Boskos, I was given the opportunity to tackle a real-world problem focused on air pollution and work on a optimisation framework that can be applied in a plethora of similar problem formulations as well.

My journey in this thesis involved many hours of theoretical analysis and research. This study allowed me to acquire the background knowledge required for the comprehension of the presented scientific notions and the formulation of the proposed methodology. This it was followed by more hours of troubleshooting between FreeFEM++ and MATLAB, since there were times in which the code would not run, the results would not make sense, or the theoretical procedures could not be successfully implemented. The present thesis is the result of all these efforts.

So dear reader, as you go through these pages, I hope that I will manage to pass on my interest in this scientific field and that you will enjoy it, as much as I enjoyed the whole process of researching for it and writing it.

# Acknowledgements

I would like to thank my supervisor Dr. D. (Dimitris) Boskos for his assistance during the writing of this thesis. Without his guidance, I would not have been able to overcome the challenges that I encountered along the way. He was always available and willing to answer my numerous questions and allowed me to work on a topic that I am profoundly interested in.

I would also like to thank Dr. O. (Omid) Nejadseyfi for participating in my thesis committee, taking the time to read and evaluate this work.

Finally, I am extremely grateful to my family and friends for their continuous support throughout all the years of my Master's study.

Delft, University of Technology
September 11, 2026

Evangelos Skrinis

*Πάντα στον νου σου να 'χεις την Ιθάκη.*
*Το φθάσιμον εκεί είν' ο προορισμός σου.*
*Αλλά μη βιάζεις το ταξίδι διόλου.*

*Καλύτερα χρόνια πολλά να διαρκέσει*
*και γέρος πια ν' αράξεις στο νησί,*
*πλούσιος με όσα κέρδισες στον δρόμο,*
*μη προσδοκώντας πλούτη να σε δώσει η Ιθάκη.*

*Ιθάκη*, Κ. Π. Καβάφης

# Chapter 1

# Introduction

This chapter establishes the underlying scope, the motivation and the main objectives of this thesis. Section 1-1 will break down and provide insight into the main challenges associated with the problem of optimal air pollution management. It includes a brief overview of the requirements of an optimisation scheme focused on the minimisation of pollutant concentrations and their socioeconomic impacts. Furthermore, some of the existing methodologies utilised for the optimal control of air pollution will be discussed, along with their respective advantages and limitations. Then, in Section 1-2 we will present the main research questions that are addressed in this thesis. Finally, an outline of the structure and the remaining chapters is provided in Section 1-3.

## 1-1 Background and Motivation

Air pollution is the phenomenon of the accumulation of particles or gases in the atmosphere, at concentrations high enough to alter the atmospheric chemistry and impair ecological systems. Pollutants, such as $SO_2$, $NO_2$, CO, and $PM_{2.5-10}$, can originate from natural processes (volcanic eruptions, forest fires), biogenic sources (marine and terrestrial ecosystems), and anthropogenic activities (energy production, agriculture, transportation, industrial manufacturing processes, urbanisation, etc.)[1]. The atmospheric propagation of these pollutants is dependent on meteorological variables (temperature, wind speed, boundary layer height, etc.) and on the morphology of the terrain through a variety of physical processes such as diffusion, advection, chemical conversion, and many other [2]. At present, air pollution is an unavoidable by-product of human activity and after more than two hundred years the rapid industrialisation and technological advancement, mankind constitutes its main contributor [3]. However, the impact of poor air quality conditions is detrimental to human health with both economic and societal implications. The total cost of sick days, chronic diseases, increased mortality in combination with a decreased fertility rate [4] will continue to accumulate for both the society and the economy unless action is taken. Fortunately, the time period from the last quarter of the twentieth century until 'recently', can be distinguished by an increased

awareness of humanity's impact on Earth. As a result, the improvement of air quality, through the minimisation of the adverse effects of human activities, has been a worldwide concern. To that end, a plethora of systems for the prediction, early warning, and control of air pollution, in a variety of anthropogenic sources and environmental circumstances has been a priority for many countries and international regulatory organisations such as EEA,US EPA and others [5, 6].

An optimal pollution-abatement policy should involve the control of human activities that generate emissions and at the same time support essential societal, economic, and industrial functions. Bringing those processes to a halt, relocating them, or even blindly regulating them, would indeed negate the effects of air-pollution but it would also generate substantial costs in different areas of interest such as supply chains, manufacturing, essential services, etc. [7]. As a result, the objective of minimizing the adverse effects of the dispersion of pollutants in the atmosphere must also take into account both the implementation costs of the proposed countermeasures and their toll in the economy and the society[8]. To that end, the formulation of an optimisation framework that is able to accommodate different cost functions and, most importantly, a variety of constraints would be of paramount importance. The ideal framework should be able to support both short-term control strategies (early warning, emergency response) and long-term planning (scenario testing, policy assessment) while taking into account the results of cost-benefit analysis. This can be achieved in an optimisation scheme (PDE-constrained optimisation) by utilising complex mathematical models for the accurate prediction of the underlying physical phenomena, and constraints along with cost functions for the enforcement of specific policies [9, 10]. The selected model, used for the simulation of the transport of air pollution, must be sufficiently accurate, so as to provide reliable quantitative prediction data in real-world conditions. At the same time, it should be simulated within a time frame that allows for dynamic responses in continuously evolving phenomena and its form should accommodate the implementation of fast-converging optimisation algorithms [11, 12].

Analogous requirements are extended to the optimisation scheme that will be implemented. In particular, the ideal methodology should be adaptable to the aforementioned specifications, meaning that it should be able to incorporate different types and number of optimisation variables, state & control constraints and conflicting objective functions. However, this formulation is desirable only as long as it is able to provide a computationally tractable framework that retains sufficient fidelity to the underlying physical model. Although there are many methodologies, that are able to accommodate PDE-constrained optimisation problems, a common disadvantage is the increase of the computational, as the spatial and temporal resolution increases and more complicated characteristics are introduced. A partial solution to this limitation could be the use of domain decomposition methods (DDM). DDM are basically a parallel numerical technique focused on the solution of partial differential equations. They are able to split up the whole domain into smaller subdomains and thus find a quickly global solution from the results of the local problems that can be solved independently and in parallel [13, 14]. The insight behind this idea, lies in the fact that a PDE-constrained optimisation could retain most of the numerical properties of the discretised state equation and as a result, a parallel solution through DDM for the large discretised optimisation problem could be designed.

Based on, either the decomposition of the whole optimal problem [15], or the PDEs that govern it [16], it is possible to parallelise some procedures and create preconditioners that

speed up selected solvers (e.g. Krylov) [17]. In fact, several approaches have been proposed for the utilisation of DDM and the spatial or temporal decomposition of the PDE-constrained optimisation problems [18, 19]. Although, the specific details of all these methodologies may differ, their common characteristic, is that they are implemented in a particular class of linear PDEs and, most importantly, without any equality or inequality constraints [20, 21, 22, 23]. This means that they cannot be implemented successfully to air pollution control problems and other applications that require complicated, sometimes non-linear, PDEs to accurately describe the underlying processes, and depend on constraints to restrict the systems variables from exceeding operational or physical limits.

## 1-2 Research Questions

In an attempt to address the gap that was identified in the previous section, this thesis will endeavour to formulate an optimisation framework capable of accommodating a broad range of requirements, focus on air pollution control, utilising domain decomposition methodologies. As a result, the main research question will be formulated as follows:

> ***How can an optimisation framework based on domain decomposition methodologies be formulated to accommodate the requirements of air pollution control?***

The analysis of this research question can be broken into four tasks, each of which must be addressed for the development of the proposed framework.

Firstly, it is important to *identify the main requirements for the optimal control of air pollution and its impacts*. This includes the different types of cost functionals, control variables and, constraints that are crucial for the formulation of the PDE and the optimisation problem. Only after these requirements have been established can the necessary attributes of the optimisation algorithm be determined.

Secondly, the focus is directed on the *search of PDE-constrained optimisation methods* that can accommodate those attributes, can ensure the computational tractability of the problem as the size and the complexity increases and whose framework enables the implementation of domain decomposition methods.

Thirdly, *a variety of domain decomposition methods must be assessed* in order to determine whether they could reduce the computational cost and improve the efficiency and the scalability of the selected optimisation algorithm.

Finally, the last question addresses *how can all these components be integrated into a unified methodology* that can incorporate all the necessary requirements of air pollution control.

## 1-3 Thesis Outline

The rest of this thesis is composed by six additional chapters.Chapter 2 examines the main approaches to air-pollution modelling and their integration into optimisation frameworks. Furthermore, it formulates a case study that will be used as a reference in the proposed methodology. Then at Chapter 3, we review the optimisation schemes that could be coupled

with the described models. PDE-constrained optimisation algorithms in combination with techniques that enhance their adaptability and tractability of the overall methodology such as the augmented Lagrangian approach are analysed. This is followed by an overview and the analysis of the classical and contemporary methods associated with DDM in Chapter 4. Emphasis is given on the substructuring methods that are focused on the solution of PDEs. All these aforementioned techniques are integrated into one unified framework that is formulated and explained in Chapter 5. Later, on Chapter 6 the implementation of the proposed algorithm in the case of air pollution control is addressed. Furthermore, the numerical results along with the efficiency comparison between case studies are presented and interpreted. Finally the concluding remarks and suggestions for future work and improvements on proposed methodology are summarised in Chapter 7.

# Chapter 2

# Air Pollution

In this chapter we present a review of air-pollution along with the fundamental concepts that are used to simulate its underlying physical phenomena. Section 2-1 includes an overview of the adverse effects and consequences of air-pollution and highlights the need for accurate prediction models imbedded in optimisation frameworks. Afterwards, Section 2-2 focuses on the different types of forecasting models and it provides insight on their classification and implementation. Following this, Section 2-3 discusses the main requirements of an optimisation problem with respect to the corresponding model and air-pollution in general. Finally, in Section 2-4 we present and analyse the reference model that will be used for the numerical examples of the proposed optimisation framework.

## 2-1 Overview of Air-Pollution

Air-pollution should not be regarded as an abstract concept that affects only a minority of highly susceptible individuals. It can result in both long-term and short-term effects, impacting global and regional areas alike, and it could require widely spread policies in order to deal with its aftermath. In other instances, it is possible to counteract its negative consequences through local, in situ decisions [24]. For example, the accumulation of air pollutants over many decades can have an adverse impact even on the whole planet and its climate, and thus requires the cooperation of many countries over different continents in order to counter it. On the other hand, the same pollutants can pose an immediate risk to human well-being if they exceed specific concentration limits even for a short period of time in a specific region [25]. Simultaneously, the prolonged exposure to even slightly elevated concentration can prove to be detrimental to people's health [26].

Depending on the scope of interest of a specific research, most of these harmful effects can be identified and accurately estimated for the ecological systems, the economic activities and the health humans. In particular, healthcare expenses, lost productivity, water quality deterioration, abatement expenditure, decreased agricultural production, societal impacts, and many other direct or indirect consequences have quantifiable costs that can be reflected to

key performance indicators. These KPIs can provide an evaluation basis for the assessment of the different policies and control decisions based on the predicted outcome and their effect on a selected objective [5]. To protect human health, the environment, and mitigate the societal burden, the World Health Organization (WHO) along with several countries have identified and issued guidelines for the concentration of sulphur dioxide ($SO_2$), nitrogen dioxide ($NO_2$), carbon monoxide (CO), ozone ($O_3$), lead (Pb), particulate matter with aerodynamic diameters less than 2.5 $mm$ and 10 $mm$ ($PM_{2.5-10}$) and benzene ($C_6H_6$) as some of the most prominent sources of air-pollution [27, 28]. The process of correlating the concentration of these pollutants with macroeconomic indexes is a complicated, multi-stage procedure that requires the interdisciplinary cooperation of a plethora of fields (environmental science, medical science, exposure assessment, econometrics, etc.). However, any attempt to assess, predict, or control the effects of air-pollution, must begin with the creation of an accurate model of the pollutant's dispersion through the atmosphere[29, 30]. In order to compute the required concentrations, one must take into account the emission sources that produce the air pollutants, the atmospheric processes of diffusion, transport, and chemical transformation as well as the response of the receptors agents. The main focus of this chapter is the description of mathematical models that are able to achieve this and of the necessary requirements to their implementation in optimisation frameworks.

## 2-2 Dispersion Models of Air-Pollution

The transportation of pollutants in the air is a result of a variety of physical and chemical processes (advection, turbulent diffusion, deposition, chemical reactions, aerosol dynamics etc.) and is dependent on a plethora of variables (temperature, wind speed & direction, geomorphological relief, source type, particle diameter, etc.) [2]. These pollutants can travel over many distances in varying timescales, impacting ecosystems on their path and even infiltrate indoor spaces far away from the emission sources. Due to differences in the size of the regions and/or time domains, the availability of data, the required accuracy and robustness, the available computational power and many other factors, several Air Quality Forecasting (AQF) methodologies have been developed. They can be broadly classified into a variety of different groups based on their properties (deterministic, parametric, statistical, hybrid, etc),

One of the first contemporary systematic methods that developed a tool for the prediction of the air-pollution in the atmosphere was documented in the middle of the twentieth century and it served as an early warning mechanism. Based on the idea that meteorological stagnation is conducive to the accumulation of air pollutants in the atmosphere, these methodologies made qualitative predictions of the air quality level, based on the meteorological measurements in the atmosphere [11, 31]. Since then, the advances in the sciences of meteorology, environmental statistics, and computer engineering, as well as the decreased cost and the technological improvement of measuring instruments, have paved the way for the development of more sophisticated pollutant dispersion models [11, 32].

Statistical models can analyse a vast amount of data from many different sources and rapidly predict the concentration of air pollutants without any previous knowledge of the physical, chemical, or biological processes that affect their dispersion. Multi-variate regression models (e.g. LUR [33]), ARIMA models, and principal component analysis methods are just some of

the more prominent frameworks that fall under this category [34, 35]. Most of these methodologies focus on the prediction of the air quality based either on the past measurements and variations from computed values or on the correlation of specific meteorological, geographical and other features to the pollutants' concentration. The availability of measurement data along with the implementation of machine learning techniques has enabled the development of black box models with high accuracy and rapid result generation [36, 37, 38].

Numerical models, on the other hand, utilise mathematical approaches to simulate the internal mechanisms of the underlying physical, chemical and biological processes that are present at the dispersion of pollution in the atmosphere. Depending on the required accuracy, these techniques can result in large coupled differential or partial differential equations with a spatial domain ranging from an indoor area to the whole atmosphere and a temporal range from minutes to years. the diagram of a complete model of air-pollution can be seen in Figure 2-1 [39]. Depending on the application, the selected mathematical expressions could

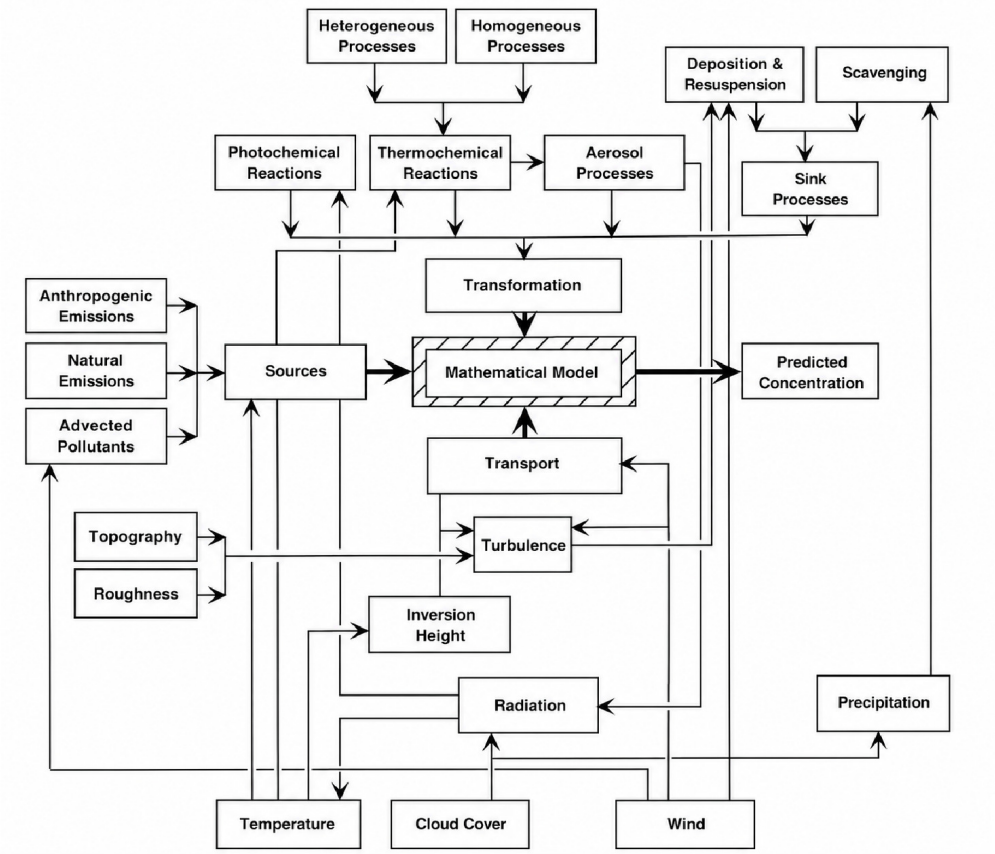



**Figure 2-1:** Components of a comprehensive air quality model.

be simplified by taking into consideration only the principal phenomena such the processes of advection, diffusion, wet and dry deposition and chemical reaction. Furthermore, the spatio-temporal domain can be adjusted to fit the corresponding problem at hand. However there are problems, that based on the kind of the pollutant and the type of research, have to be capable of representing more elaborate mechanisms (aerosol micro-kinetics, gravitational settling, effect of boundary layers) under more complex environments (street canyons) [40] .

## 2-3 Air-Pollution Optimal Control Requirements

One of the goals of this thesis, is the implementation of a model for the dispersion of pollution in the atmosphere, into an optimisation framework, that is aimed in the mitigation air pollutants and their effects. In order to achieve this in a systematic way, it is crucial that the model can facilitate numerical comparisons between different cases. These outputs are necessary for the strategic design of countermeasures that could efficiently suppress the effects of air-pollution [30]. Advanced and proactive risk assessment plans have the potential to yield substantial societal and economic benefits, positively affecting individuals, companies, and whole countries in general. As a result, any qualitative approaches have to be excluded from a proposed optimisation framework. The lack of precise prediction of the pollutants' concentration would lead to a large margin of error since too much idle information is left unused. However, this is not the sole requirement. Accuracy, robustness and computational efficiency play a critical role and are essential characteristics of a model planned to me used within an optimisation framework.

Statistical and machine learning approaches are able to produce quantitative results with the added benefit of rapid computation since the computationally heavy task (training) has been already performed online. However, these types of methods have specific prerequisites in terms of data quality and sample size. When they are not met, the accuracy of the proposed model will decrease. Even when these requirements are met, the simulation's uncertainty will increase if the optimisation process leads the system outside of its training set [41]. Thus, an optimisation framework, based on these models, would require high-resolution and high-quality datasets. These, however, that are not always available or easily accessible across a variety of scenarios, pollutants or spatial and temporal resolutions. Furthermore, most of these models are treated as a 'black box' with little to no insight on the interpretability between the produced concentrations and the controllable inputs.

For these reasons, numerical models are used for the representation of the phenomena of the pollution dispersion in the proposed optimisation framework in this thesis. They are also capable of generating quantitative approximations of the pollutant concentration in the atmosphere with different levels of accuracy depending on the application [42]. The knowledge of the governing dynamics allows the model to capture abrupt emissions or rapidly changing atmospheric conditions when the transient response is significant as well as the steady state under a wide variety of circumstances. Furthermore, it is possible to derive explicit information regarding the sensitivity of the pollutant dispersion with respect to selected parameters. This is extremely advantageous in PDE-constrained optimisation for the utilisation of the gradient based methods. Their only limitation lies in the computational cost that arise from their numerical solution. As the complexity of the model and the accuracy requirements increase, so does the necessary time it takes to solve the underlying equations. One of the goals of this thesis is to attempt to resolve this obstacle with the appropriate optimisation framework and the utilisation of domain decomposition techniques.

Apart from the choice of the model, there are many other attributes, associated with air-pollution, that have to be clarified before the formulation of an optimisation problem. One of them, is the type of the emission sources, as it may be required to represent a road (line source), an industrial or agricultural area (area source) or the residential buildings (multiple distributed sources). Closely related to this, is the identification of the control and

mitigation strategies. Through these actions, the optimisation algorithm attempts to reduce the effects of pollutants. Apart from the obvious restriction of the emission rate (reduce industrial output, road access, etc.), more advanced options could be required depending on the application. The position of the emission source, its maximum operating level, the adjustment of the composition of energy production, when available, should be intergraded to a general PDE-constrained optimisation framework. Furthermore, any constraints to either the control or state variables that correspond to safety and operational limits should not be excluded from the problem. Finally, the cost function of the optimisation problem could be as simple as penalising pollutant concentrations that exceed a preset safety threshold or could include competing objectives. For example, an advanced cost-benefit analysis could provide the environmental and health-related costs associated with the increased pollution along with the required resources and additional cost of mitigation policies. All these characteristics should be considered in the choice of an optimisation algorithm.

## 2-4 Air-Pollution Model Based on PDE

The case study of this thesis, as it is previously mentioned, is the dispersion of pollutants through the air. In order to accurately simulate all the related physical phenomena under a broad range of different scenarios, a mathematical model based on partial differential equations is utilised. For that reason, the advection-diffusion-reaction equation was selected since it is the basis of many contemporary CTM [43]. It is a well-studied model for representing all the relevant natural or anthropogenic processes of emission, atmospheric transport, diffusion, turbulent dispersion along with more complicated phenomena such as chemical reactions, aerosol dynamics, photolysis, and more [39].

Depending on the application, the pollutant in question, and the required accuracy, it is possible to select which terms of the PDE are essential for the proposed model. The general form of this equation is:

$$\boxed{\frac{\partial c(\mathbf{x},t)}{\partial t} - \nabla \cdot (\mathbf{K}\nabla c(\mathbf{x},t)) + \nabla \cdot (\mathbf{V}c(\mathbf{x},t)) = \mathbf{q}(\mathbf{x},t) + R(c,\mathbf{x},t)} \tag{2-1}$$

with

- $c(\mathbf{x},t)$ is the pollutant concentration measured in an unit such as $^{kg}/_{m^3}$ or $^{mol}/_{m^3}$,
- $\mathbf{x}$ is the spatial coordinates and t is the temporal coordinate,
- $\mathbf{V}$ is the wind velocity $[^{m}/_{s}]$, and
- $\mathbf{K}$ is the diffusivity tensor $[^{m^2}/_{s}]$.

Based on this formulation, the first term $^{\partial c}/_{\partial t}$ denotes the temporal accumulation of the pollutant's concentration. Then, the principal terms that contribute to the pollutants' transport are presented. Based on Fick's laws [44], the advection $\nabla \cdot (\mathbf{V}c)$ and the diffusion $-\nabla(\mathbf{K}\nabla c)$ terms denote the advective and diffusive flux, respectively and describe their correlation with the concentration's gradient. Finally, $\mathbf{q}$ represents the existence of pollutant emission sources

or sinks whereas $R(c, \mathbf{x}, t)$ term denotes any additional required physical or chemical process, as described beforehand. This expression can be critical to specific applications since it is the key to a realistic description of processes that are more complicated than simple advection or diffusion. These complex mechanisms of chemical reactions, deposition, aerosol dynamics, and others can be represented by non-linear and/or coupled PDEs dependent on several other parameters such as temperature, pressure gradients, the Coriolis force, the type of terrain and soil properties, etc [39].

As it is mentioned previously, the described formulation is able to capture a wide range of scenarios with different operational requirements. One common example is that the advection component can incorporate the case of compressible atmospheric flow whereas the diffusion component capable of modelling anisotropic and turbulent dispersion via the values of the tensor $\mathbf{K}$. The terms:

$$\mathbf{K} = \begin{bmatrix} K_{xx} & K_{xy} & K_{xz} \\ K_{yx} & K_{yy} & K_{yz} \\ K_{zx} & K_{zy} & K_{zz} \end{bmatrix}, \qquad \nabla(\mathbf{V}c) = \mathbf{V} \cdot \nabla c + c \cdot \nabla \mathbf{V}$$

could be also dependent on the spatial or temporal coordinates ($\mathbf{K}(c, \mathbf{x}, t)$) as well as on additional parameters used to describe complex processes such as turbulence or deposition. Similarly, the term $\mathbf{q}(\mathbf{x}, t)$ is usually represented mathematically as an ideal point source or sink via the Dirac function. However, it is possible to extend this form into a line source in case a road acts as input or even into a distributed source in a subdomain $\Omega_r$ in the case of various industrial pollution areas, chimneys, or others as seen below:

$$q(\mathbf{x}, t) = \begin{cases} q_r(\mathbf{x}, t) & \mathbf{x} \in \Omega_r \\ 0 & \mathbf{x} \notin \Omega_r. \end{cases}$$

If emissions can be controlled, then they can take the format $q(\mathbf{x}, t; u) = u(t) \cdot q(\mathbf{x}, t)$, with $u(t)$ denoting the available actions that limit or increase the production of pollutants.

### Proposed Case Study

It has become evident that the advection-diffusion-reaction equation can facilitate a wide range of problems. It can also be augmented with data or measurement-derived parameters that manage to capture more complicated phenomena. The purpose of the proposed air-pollution model is to be implemented in a PDE-constrained optimisation scheme. To that end, it is advantageous to formulate a model that will be complex enough to accurately depict the dispersion of air pollutants but also simple enough, so as to not hinder the implementation of a novel methodology and its numerical benchmark. After an initial case study is complete, the method's effectiveness can be analysed for a more general form of the PDE.

Based on this framework, the steady state 2D advection-diffusion equation is presented. The main changes consist of the lack of a reaction term along with any time-varying terms. Among those is the temporal accumulation term based on the variable t, $\frac{\partial c}{\partial t} = 0$. Although transient models provide a more comprehensive representation of atmospheric pollutant transport, steady-state formulations are more preferable for continuous emission scenarios with slowly varying meteorological conditions. Furthermore, the $\mathbf{q}$ represents the controllable distributed

source from three industrial chimneys and is equal to the normalised emission rate with respect to an appropriate emission volume. Moreover, the spatial coordinates $\mathbf{x}$ are limited into two dimensions. $\mathbf{V}$ will denote the stable median air-velocity with constant intensity and direction. Lastly, the diffusion will be assumed to be isotropic and independent of any other parameter. All the aforementioned changes can be mathematically represented as follows:

$$\mathbf{K} = \nu \cdot I_{2\times 2} \implies -\nabla \cdot (\mathbf{K}\nabla c(\mathbf{x})) = -\nu\nabla^2 c(\mathbf{x}),$$
$$\nabla \cdot (\mathbf{V}c(\mathbf{x})) = \mathbf{V} \cdot \nabla c(\mathbf{x}) + \cancel{c(\mathbf{x})\nabla \cdot \mathbf{V}} = \mathbf{V} \cdot \nabla c(\mathbf{x}),$$
$$q(\mathbf{x}; u) = \begin{cases} u(\mathbf{x}) & \mathbf{x} \in \Omega_r \\ 0 & \mathbf{x} \notin \Omega_r. \end{cases}$$

As a result, the derived partial differential equation that is proposed for the initial case study of the formulated optimisation framework is seen in the Equation 2-2. It describes the steady state of the advection and diffusion of the air pollutant on a two-dimensional plane with a vertical distance from the ground equal to the effective height $H$.

$$\begin{aligned} -\nu\nabla^2 c(\mathbf{x}) + \mathbf{V} \cdot \nabla c(\mathbf{x}) &= \mathbf{q}(\mathbf{x}; u) && \text{in } \Omega \\ c(\mathbf{x}) &= 0 && \text{on } \Gamma_{\text{in}} \\ \nu\frac{\partial c}{\partial n} &= 0 && \text{on } \Gamma_{\text{out}} \end{aligned} \tag{2-2}$$

The boundary conditions in this PDE describe a very specific situation as well. The concentration of the pollutant in zero in the inflow domain boundary($\Gamma_{\text{in}} := \{\partial\Omega | \mathbf{V} \cdot \mathbf{n} < 0\}$) so as to ensure that all the pollutant in the domain is derived exclusively from the emission sources. Furthermore, at the outflow domain boundary($\Gamma_{\text{out}} := \{\partial\Omega | \mathbf{V} \cdot \mathbf{n} \geq 0\}$) the concentration is assumed to have a state that does not change in the outward direction $\mathbf{n}$

### Extension into three dimensions

The aforementioned model is the first step in the design of the PDE-constrained optimisation method. The selected form PDE will be advantageous for the initial case study since it captures the principal physical phenomena that dominate dispersion process while it is simple enough to allow benchmark testing of complex optimisation techniques. However, there are other models that can provide higher accuracy to the proposed methodology with respect to the representation of real-world scenarios. Most of the previously described examples, like the CTM models with coupled mechanics in three dimensions that treat the whole atmosphere as a unified system [6], can be used to achieve this. Therefore, it is essential to ensure the effectiveness of the formulated optimisation framework for more sophisticated air-pollution models. In order to achieve this goal, an extension of the Equation 2-2 is introduced based on the well-established approach of the Gaussian plume.

The Gaussian plume method, along with a plethora of its variants has been used by engineers for numerous applications that demand the modelling of turbulent dispersion of natural or human-induced pollutants [39, 45]. This mathematical model can be derived from the advection-diffusion equation under the mild assumptions of minor variations in the ground topography and neglecting diffusion along the airflow direction in favour of the advection

terms [45]. Then, based on the variation, different hypotheses are implemented so as to accurately simulate a variety of more elaborate processes. The standard interpretation is described by Equation 2-3 and the result can be seen in Figure 2-2[45]. According to the Gaussian plume model the concentration is equal to

$$c(x,y,z) = \frac{Q}{2\pi u_x \sigma_y(x)\sigma_z(x)} \exp\left(-\frac{y^2}{2\sigma_y^2(x)}\right)\left[\exp\left(-\frac{(z-H)^2}{2\sigma_z^2(x)}\right) + \exp\left(-\frac{(z+H)^2}{2\sigma_z^2(x)}\right)\right] \quad (2\text{-}3)$$

where $c(x,y,z)$ is the concentration of pollutant at the point $(x,y,z)$, $Q$ is the constant emission rate of the pollutant, $u_x$ is the mean wind speed in $x-$direction, and $H$ is the effective height of the plume centre-line that is expressed as $H = H_s + \Delta H$, with $H_s$ the physical height of the emission source and $\Delta H$ the plume rise height caused by the initial momentum and buoyancy of the emitted gases. The coordinates $x$, $y$, and $z$ denote the downwind, crosswind, and vertical directions respectively whereas the functions $\sigma_y(x)$ and $\sigma_z(x)$ are the standard deviation terms which describe the spread of the plume due to atmospheric turbulence and are dependent on the distance from the source as well as the class of atmospheric stability.

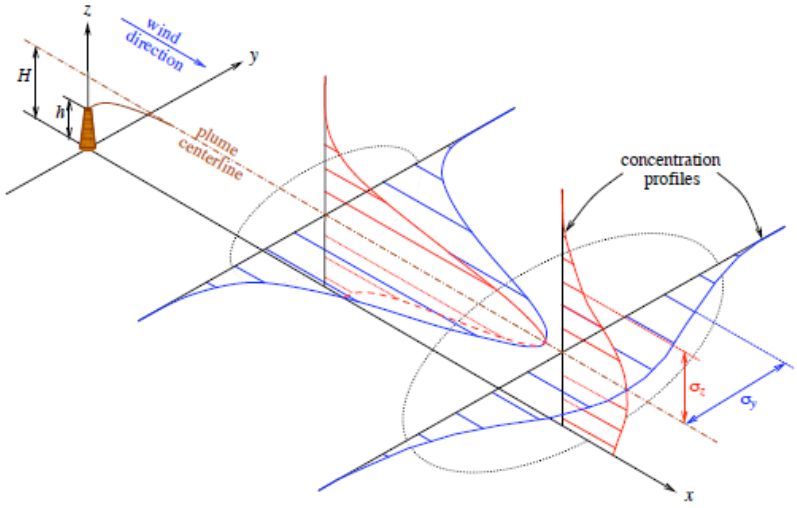



**Figure 2-2:** The Gaussian plume emitted from a point source.

Using this formulation as a basis, it is possible to implement additional features such as multiple sources of varying type, different levels of terrain absorbing capacity, and introduce the notions of the inversion layer, height-dependent parameters and time-varying sources.

Instead of transitioning directly to a PDE that describes the pollutants' behaviour in three dimensions, the aforementioned advection-diffusion equation can be projected into the vertical direction with the assistance of the Gaussian plume. This quasi-3D model utilises the solution $\mathbf{c}(x,y)$ of (2-2) and multiplies it by the term:

$$\pi(z) = \left[\exp\left(-\frac{(z-H)^2}{2\sigma_z^2}\right) + \exp\left(-\frac{(z+H)^2}{2\sigma_z^2}\right)\right]. \quad (2\text{-}4)$$

As a result the complete solution will have the following form:

$$\mathbf{c}_{2.5D}(x,y,z) = \pi(z) \cdot \mathbf{c}(x,y), \quad (2\text{-}5)$$

with variable $z$ denoting the distance from the ground. In most cases of environmental control, the concentration value at $z = 0$ (the ground level) is required for the computation of the degree of air-pollution. Under this assumption, the solution can be reformulated as follows:

$$\mathbf{c}_{2.5D}(x, y, 0) = \pi(0) \cdot \mathbf{c}(x, y) = 2 \exp\left(\frac{-H^2}{2\sigma_z^2}\right) \cdot \mathbf{c}(x, y) \tag{2-6}$$

#### Extensions of the quasi-3D model

In the previous section, several additional attributes are presented on the premise that their implementation can augment the capabilities of the air-pollutant dispersion models. Among these, emphasis should be given in the cases of a non-uniform air velocity field and the dependence of the diffusion parameters on variables such as the velocity, distance from the source and atmospheric stability. In those cases, the air speed will be substituted with the term $\mathbf{V} \coloneqq \mathbf{V}(\mathbf{x})$ and all the terms that are used to describe turbulent diffusion will also be subject to additional parameters. More specifically, based on the Pasquill-Gifford dispersion model, the atmospheric conditions can be categorised into stable, neutral and unstable according to a variety of factors such as the wind speed, the solar irradiance, the type of the terrain, etc [39]. Depending on the stability class, both the vertical $\sigma_z(\mathbf{x})$ and the lateral $\sigma_y(\mathbf{x})$ will be computed using formulas based on the distance from the emission source $\mathbf{r}$ as seen in Table 2-1 [46]. Then, the diffusion coefficient can be computed via the formula:

$$\sigma^2(x) = \frac{2}{u} \int_0^x K(\xi)\, d\xi. \tag{2-7}$$

| Stability Class | Lateral Dispersion $\sigma_y(x)$ | Vertical Dispersion $\sigma_z(x)$ |
|---|---|---|
| Extremely unstable | $0.22r\,(1 + 0.0001r)^{-1/2}$ | $0.20r$ |
| Slightly unstable | $0.11r\,(1 + 0.0001r)^{-1/2}$ | $0.08r\,(1 + 0.0002r)^{-1/2}$ |
| Neutral | $0.08r\,(1 + 0.0001r)^{-1/2}$ | $0.06r\,(1 + 0.0015r)^{-1/2}$ |
| Stable | $0.04r\,(1 + 0.0001r)^{-1/2}$ | $0.016r\,(1 + 0.0003r)^{-1}$ |

**Table 2-1:** Briggs' dispersion coefficients for rural terrain

Another way to enhance the performance of the described model is to include the reaction terms described via the expression $R(c, \mathbf{x}, t)$. This general term is able to facilitate the modelling of a vast range of phenomena. For example, the plethora of chemical reactions, that occur in the atmosphere, could constitute an important factor for the whole model. Many of the pollutants that are produced by human activities are highly reactive and create new molecules continuously, as they come in contact with other atmospheric gases. As a matter of fact, in some cases the by-products of these transformations would consist a more dangerous pollutant and thus they would have to be monitored. One way to achieve the representation of the chemical reactions in the partial differential equation, is with the term $\lambda_{chem} \cdot \mathbf{c}(\mathbf{x})^{\alpha_{chem}}$. The parameter $\lambda_{chem}$ includes the information regarding the reaction rate and can be associated with either the pollutant's lifetime or the chemical kinematics of the

process whereas the $\alpha_{chem}$ denotes the chemical reaction's order [47]. Similarly to the described expression, phenomena such as photolysis ($\lambda_{phot}\mathbf{c}$), dry or wet deposition ($\lambda_{dry/wet}\mathbf{c}$) and gravitational settling ($\lambda_{grav}\nabla\mathbf{c}$) can be implemented in the PDE if it is necessitated by the selected application.

## Synopsis of Chapter 2

This chapter presented the fundamental concepts of air-pollution and highlighted the principal modelling approaches used to simulate the dispersion of pollutants in the atmosphere. Then we discussed the requirements of optimal air-pollution control with respect to its various scenarios. Finally, emphasis was given to the formulation of the mathematical model of air-pollution (advection-diffusion) that will be used as a case study for the numerical implementation of the proposed methodology. This model is analysed and assessed with respect to its accuracy, interpretability, computational efficiency, and suitability for integration into air-pollution optimisation frameworks.

# Chapter 3

# PDE-Constrained Optimisation

In this chapter the fundamental concepts of PDE-constrained optimisation are presented and its application to the control of air-pollution is thoroughly examined. Section 3-1 introduces the primary components of an optimisation framework that aims into the development of optimal mitigation strategies. Then, Section 3-2 provides the main characteristics of the proposed optimisation algorithm. we emphasise on the attributes of the optimality conditions, discretisation, decision variables, and handling of the constraints that, more often than not, are dictated from the nature, prerequisites, and goals of the problem.

## 3-1 General Formulation of the Optimisation Scheme

PDE-Constrained Optimisation refers to the solution of optimal control problems that are governed by partial differential equations. In engineering and many other applied sciences, PDEs are used in order to model the behaviour of a wide range of systems, from heat transfer, electromagnetism, and fluid mechanics to structural analysis, economic processes, and population dynamics [10]. Optimal Control Problems (OCPs) involve the identification of a control function $u$ and a corresponding state function $c$, that will drive the objective functional $\mathcal{J}(c,u)$ to its minimum while they are subject to constraints (by physical or artificial laws). The state equations are represented by the operator $\mathcal{E}(c,u)$, which may denote algebraic, ordinary differential, or a partial differential equations while the state and control functions may be restricted within specific admissible sets $Y_{\text{ad}}, U_{\text{ad}}$. Any equality constraints are embedded in the $\mathcal{E}$ term whereas any additional inequality constraints are represented by the $G(c,u)$ operator. As a result the following abstract and general form of an OCP arises [48]:

$$\begin{aligned} \min_{c,u} \quad & \mathcal{J}(c,u) \\ \text{s.t.} \quad & \mathcal{E}(c,u) = 0, \quad \mathcal{Q}(c,u) \leq 0 \\ & u \in U_{\text{ad}} \quad c \in C_{\text{ad}} \end{aligned} \tag{3-1}$$

This framework can incorporate a vast range of problems and thus, it can accommodate the requirements of the many applications and different scenarios of air pollution control.

For example, physical coefficients, boundary/initial conditions, source/sink terms, material parameters, as well as the geometrical or even the topological configuration of the domain can act as control variables depending on the situation and the associated goals [10]. All these choices, in addition to the structure of the PDE and the cost functional $\mathcal{J}$, hold a pivotal role in the methodologies that will be used to obtain an optimal solution and can be expressed via the Equation 3-1.

For that reason, PDE-constrained optimisation is a very well suited methodology to tackle the problem of mitigating the air-pollution effects on a specified area of interest [49, 50]. First of all, as it is mentioned beforehand, it is designed around the minimisation of cost functional that is constrained by a partial differential equation. The air pollution underlying dynamics will be described by a set of PDEs, so it will be prudent to take advantage of the transport model and implement an optimisation algorithm that can facilitate this. Furthermore, this methodology satisfies all the requirements of air pollution optimal control as described in Chapter 2. The cost functional does not have any restrictions and as a result it could be composed of either two quadratic penalties that are localised some parts of the domain $\Omega_{\text{target}}$ and $\Omega_u$ :

$$\mathcal{J}(c,u) = \frac{1}{2}\int_{\Omega_{\text{target}}} (c - c_{\text{target}})^2 \, d\Omega + \int_{\Omega_u} u^2 \, d\Omega, \tag{3-2}$$

or a general function $\phi$ defined over the whole domain :

$$\mathcal{J}(c,u) = \int_0^T \int_\Omega \phi(\mathbf{x}, t, c(\mathbf{x},t), u(\mathbf{x},t)) \; dx \, dt. \tag{3-3}$$

Moreover, the different types of control actions can easily be taken into account as long as they are included in the state equation and/or the cost functional. Finally, any type of restrictions on the state $c$ and the control $u$ can be represented in this optimisation framework with equality or inequality constraints.

## 3-2 Development of PDE-Constrained Optimisation Framework

Even though the air pollution control problems, embedded with a numerical model, can be expressed naturally in a PDE-constrained optimisation framework, that does not necessarily mean that they can be easily solved. This will depend from the innate attributes of the problem, meaning the characteristics of the PDE, the cost functional, the type of constraints as well as the chosen optimisation algorithm. There are many different techniques that can be used to solve an infinite dimensional optimisation problem that is subject to a PDE.

### Design Choices

First of all, in the general form of the OCP seen at (3-1), $c$ and $u$ do not denote variables but functions. The well-posedness of the optimal control problem requires that these functions lie within specific function spaces. However, instead of searching for the functions $c(\mathbf{x},t)$ and $u(\mathbf{x},t)$ that satisfy the PDE directly everywhere, it is much more advantageous to relax this requirement and search for the weak solution of the system. The weak solution, in the case of the advection-diffusion PDE (2-2), indicates a function whose first derivative exist,

it is integrable and it satisfies the variational form of the governing equation. Generally, optimisation problems with non-linear terms or non-continuous parameters may not have a strong solution. So, by loosening this requirement, it is possible to derive to an optimal function that would be unattainable otherwise. Furthermore, the weak solution is the basis of the finite elements method that will be utilised, later on, for the discretisation of the optimisation problem.

The discretisation of the problem is necessary since, even in its weak form, the analytical solution of the optimal control problem is not always possible and it is rarely a simple procedure. However, the exact stage, in which the discretisation takes place, constitutes a design choice. The numerical approximation of $c$ and $u$ the can be performed in a different stage of the optimisation algorithm, leading to the distinction between the two different methodologies [9, 48]:

- Optimise-then-discretise (OtD)
- Discretise-then-optimise (DtO)

In OtD methods the optimality system is derived at the continuous level and then it is solved numerically. On the other hand, in DtO the state problem, the objective function and the constraints are discretised, which is then followed by an optimisation method that will be applied on the derived algebraic system. These two strategies do not necessarily yield the same result depending on the properties of the state and control functions and the selected optimisation scheme [10].

Another design choice of the optimisation algorithm, is related to the optimisation variables. Based on the handling of the state $c$ there are two categories of optimisation frameworks. On the one hand, there are the Reduced Space Methods in which the state equation is solved with respect to the state variable $c$. Then the state is eliminated from the cost functional and the constraints, leaving $u$ the sole decision variable. On the other hand, All-at-Once Methods treat both the state and the control variables as optimisation variables and the state equation is handled as an equality constraint. Both of these methods exhibit specific advantages and disadvantages. It is evident that the first approach would require the solution of the PDE that describes the state equation every time the evaluation of the objective function or the gradient is required. Furthermore, information regarding the second order optimality conditions becomes expensive to compute which usually forces the use of more slowly converging algorithms and the state constraints become more difficult to handle. This will lead to an increase of the computational cost at every iteration step towards the optimal solution, especially if the underlying PDE is non-linear [51]. However, this approach reduces the number of optimisation variables and therefore simplifies the resulting optimisation processes. In addition, it removes the state equation from the constraints, since it is satisfied directly through the computation of state variables with respect to the control variables. In the latter case, both state $c$ and control $u$ are treated as independent optimisation variables. As result, the optimisation problem avoids the numerical solution of a PDE at every time that the evaluation of the state variable is necessary. Moreover, all the constraints can be incorporated directly into the optimisation algorithm. Nevertheless, its efficient solution would require the selection of an appropriate solver and preconditioning. This is necessary since the derived linear system of the optimality conditions has a much higher dimension due to the additional variables and it is usually in a saddle-point structure [52].

### Optimality Conditions

A convenient way to handle the state equation as a constraint, is the introduction of the Lagrangian functional and the formulation of the new optimisation problem. By defining the Lagrangian multiplier $p$, the first order optimality conditions of the new unconstrained problem are expressed through the Karush-Kuhn-Tucker (KKT) conditions. In the general case, the saddle points of the Lagrangian minimisation are candidates for the optimum of the original optimisation problem. Furthermore, it is possible to derive information regarding the sensitivity of the cost functional based on the multiplier function $p$ [53]. Assuming that the optimisation problem is constrained only by the state equation,

$$\min_{c,u} \quad \mathcal{J}(c,u) \qquad \text{s.t. } \mathcal{E}(c,u) = 0, \tag{3-4}$$

the Lagrangian functional is defined as:

$$\mathcal{L}(c,u,p) = \mathcal{J}(c,u) + \langle p, \mathcal{E}(c,u) \rangle \,. \tag{3-5}$$

Then the KKT conditions are described via the system of equations:

$$\begin{aligned} \delta_p \mathcal{L} &= 0 \\ \delta_y \mathcal{L} &= 0 \\ \delta_u \mathcal{L} &= 0. \end{aligned} \tag{3-6}$$

The Newton linearisation of these equations will lead to the following system:

$$\begin{bmatrix} \nabla^2_{cc}\mathcal{L} & \nabla^2_{cu}\mathcal{L} & \mathcal{E}^*_c \\ \nabla^2_{uc}\mathcal{L} & \nabla^2_{uu}\mathcal{L} & \mathcal{E}^*_u \\ \mathcal{E}_y & \mathcal{E}_u & 0 \end{bmatrix}_{(c^k,u^k,p^k)} \begin{bmatrix} \delta c^k \\ \delta u^k \\ \delta p^k \end{bmatrix} = - \begin{bmatrix} \nabla_c \mathcal{L}(c^k,u^k,p^k) \\ \nabla_u \mathcal{L}(c^k,u^k,p^k) \\ \mathcal{E}(c^k,u^k) \end{bmatrix}. \tag{3-7}$$

### Handling of Constraints - Augmented Lagrangian

The resulting saddle point system is large and sparse so the corresponding solver must be chosen very carefully in order to ensure a fast convergence to the optimum. To that end, in order to improve the conditioning of the derived matrix and implement any additional equality and inequality constraints, the use of the augmented Lagrangian technique is suggested. In constrained optimisation problems, the existence of inequality constraints perplexes the derivation of an optimal solution. There are methods that can account for them directly, such as projected gradient, interior point or active-set [54]. However, in the augmented Lagrangian methodology there is the option to "include" them in the cost functional. This will lead to the derivation of an unconstrained optimisation problem in which fast converging algorithms can then be implemented [55, 56]. The augmented Lagrangian preserves the KKT conditions of the Lagrangian optimisation problem since on the feasible points the penalty terms vanish.

Any equality constraint (linear or non-linear) can be easily included in the augmented Lagrangian $\mathcal{L}_\rho$ via the implementation of a quadratic penalty and a multiplier term, similar to (3-5):

$$\mathcal{L}_\rho(c,u,p) = \mathcal{J}(c,u) + \langle p, \mathcal{E}(c,u) \rangle + \frac{\rho}{2} \left\| \mathcal{E}(c,u) \right\|^2 . \tag{3-8}$$

Inequality constraints can also be reformulated into additional equality constrains with the introduction of the slack variables $z$. For example, the optimisation problem:

$$\begin{aligned} &\min_{c,u} \quad \mathcal{J}(c,u) \\ \text{s.t.} \quad &\mathcal{Q}_j(c,u) \leq 0, \quad j=1,\dots,p \quad \mathcal{Q}=(\mathcal{Q}_1,\dots,\mathcal{Q}_p), \end{aligned} \tag{3-9}$$

can be reformulated as follows:

$$\begin{aligned} &\min_{c,u} \quad \mathcal{J}(c,u) \\ \text{s.t.} \quad &\mathcal{Q}_j(c,u) + z_j^2 = 0, \quad j=1,\dots,p. \end{aligned} \tag{3-10}$$

Under this form, typical and robust methodologies associated with the equality constrained optimization problems can be implemented. Considering the augmented Lagrangian approach, as before, with $\mu_i$ as the multipliers, the aforementioned formula (3-8) arises:

$$\mathcal{L}_\rho(c,u,z,\mu) = \mathcal{J}(c,u) + \sum_{j=1}^{p} \mu_j(\mathcal{Q}_j(c,u)+z_j^2) + \frac{\rho}{2}\sum_{j=1}^{p}|\mathcal{Q}_j(c,u)+z_j^2|^2 \tag{3-11}$$

The introduction of the slack variables $z$ and the term $z^2$ simplifies the form of the augmented Lagrangian and nullifies the need for any constraint on the values of the optimisation variables. However, it leads to an increase of the total number of optimisation variables and, as a result, the size of the derived linear system. To that end, we attempted to remove the $z$ variable. To achieve this we break down the optimisation into two steps. First, the minimisation of the augmented Lagrangian with respect to the $z$ variable followed by the optimisation of the derived expression with respect to $(c,u)$. In this case, the minimisation of $\mathcal{L}_\rho(c,u,z,\lambda,\mu)$ with respect to $z$ can be done explicitly as long as $c,u$are fixed. Thus, the minimisation can be written as:

$$\begin{aligned} \min_z \mathcal{L}_\rho(c,u,z,\mu) &= \min_z \left[\mathcal{J}(c,u) + \sum_{j=1}^{p} \mu_j(\mathcal{Q}_j(c,u)+z_j^2) + \frac{\rho}{2}\sum_{j=1}^{p}|\mathcal{Q}_j(c,u)+z_j^2|^2\right] \implies \\ \min_z \mathcal{L}_\rho(c,u,z,\mu) &= \min_z \left[\sum_{j=1}^{p} \mu_j(\mathcal{Q}_j(c,u)+z_j^2) + \frac{\rho}{2}\sum_{j=1}^{p}(\mathcal{Q}_j(c,u)+z_j^2)^2\right]. \end{aligned}$$

The optimum of this expression is located at:

$$\begin{aligned} z_j^\star &= \arg\min_{z_j\in\mathbb{R}} \left[\mu_j(\mathcal{Q}_j(c,u)+z_j^2) + \frac{\rho}{2}(\mathcal{Q}_j(c,u)+z_j^2)^2\right] \implies \\ (z_j^\star)^2 &= \max\left(0,\ -\mathcal{Q}_j(c,u) - \frac{\mu_j}{\rho}\right), \end{aligned} \tag{3-12}$$

and by replacing $z_j^\star$ in the previous formula (3-11), the augmented Lagrangian is now exclusively dependent on $(c,u)$, as seen in :

$$\begin{aligned} \widetilde{\mathcal{L}}_\rho(c,u,\mu) &:= \mathcal{L}_\rho(c,u,z^\star(c,u,\mu),\mu) \implies \\ \widetilde{\mathcal{L}}_\rho(c,u,\mu) &= \mathcal{J}(c,u) + \frac{1}{2\rho}\sum_{j=1}^{p}\left(\max\{0,\mu_j+\rho\mathcal{Q}_j(c,u)\}^2 - \mu_j^2\right). \end{aligned} \tag{3-13}$$

In the general case, with both equality and inequality constraints, the additional slack variables $z^2$ are replaced by the $\max(0, -\mathcal{Q}_j(c,u) - \mu_j/\rho)$ expression. Then, the complete augmented Lagrangian, with two Lagrangian multipliers $p$, $\mu$ can be compressed into the form:

$$\widetilde{\mathcal{L}_\rho}(c,u,p,\mu) = \mathcal{J}(c,u) + \langle p, \mathcal{E}(c,u)\rangle + \frac{\rho}{2}\left\|\mathcal{E}(c,u)\right\|^2 + \frac{1}{2\rho}\sum_{j=1}^{p}\left(\max\{0, \mu_j + \rho\mathcal{Q}_j(c,u)\}^2 - \mu_j^2\right).$$

After its formulation, each optimisation iteration consists of two steps. First, the minimisation of $\widetilde{\mathcal{L}_\rho}$ with respect to the state and control variables is performed (3-14a) and then, based on the produced outcome, the update of the Lagrangian multipliers is executed (3-14b):

$$(c^{k+1}, u^{k+1}) = \arg\min_{c,u} \widetilde{\mathcal{L}_{\rho_k}}(c,u,p^k,\mu^k) \tag{3-14a}$$

$$\begin{aligned} p^{k+1} &= p^k + \rho^k E(c^{k+1}, u^{k+1}), \\ \mu_j^{k+1} &= \mu_j^k \cdot \exp\left(\rho^k \mathcal{Q}_j(c^{k+1}, u^{k+1})\right), \\ \rho^{k+1} &= \Phi\left(\rho^k, c^{k+1}, c^k, u^{k+1}, u^k\right). \end{aligned} \tag{3-14b}$$

The minimisation of (3-14a) constitutes the most computationally demanding step of the described optimisation algorithm and its convergence could require many iterations. Thus, we decided that Newton optimisation algorithm should be utilised to speed up the whole procedure [53]. The main issue that hinders the effectiveness of this methodology is the requirement of a twice differentiable $\widetilde{\mathcal{L}_\rho}$ as well as the size of linear system that is produced. However, the terms :

$$\frac{1}{2\rho}\sum_{j=1}^{p}\left(\max\{0, \mu_j + \rho\mathcal{Q}_j(c,u)\}^2 - \mu_j^2\right),$$

are only once differentiable in the whole domain. As a result, the presented reformulation of the inequality constraints cannot accommodate the implementation of the Newton optimisation algorithm. More specifically, the elimination of the additional slack variables was achieved through the loss of the Lagrangian's smoothness. Therefore, the fastest optimisation methods like Newton, that utilise information from the Hessian of the Lagrangian, will not provide accurate results close to the non-differentiable points. One way to resolve this issue is to use inexact Newton methods, sacrificing the convergence speed of the whole process.

However, it is attempted to resolve this obstacle by incorporating a continuous differentiable function like the exponential, as a penalty function of the augmented Lagrangian [53] seen in Figure 3-1. Handling the inequality constraints $G_i$ with this approach, will result in a new formula for the augmented Lagrangian function that is continuously differentiable. Namely, we have:

$$\boxed{\mathcal{L}_\rho(c,u,p) = \mathcal{J}(c,u) + \langle p, \mathcal{E}(c,u)\rangle + \frac{\rho}{2}\left\|E(c,u)\right\|^2 + \sum_j \frac{\mu_j}{\rho}\left[\exp\left(\rho\cdot\mathcal{Q}_j(c,u)\right) - 1\right]}. \tag{3-15}$$

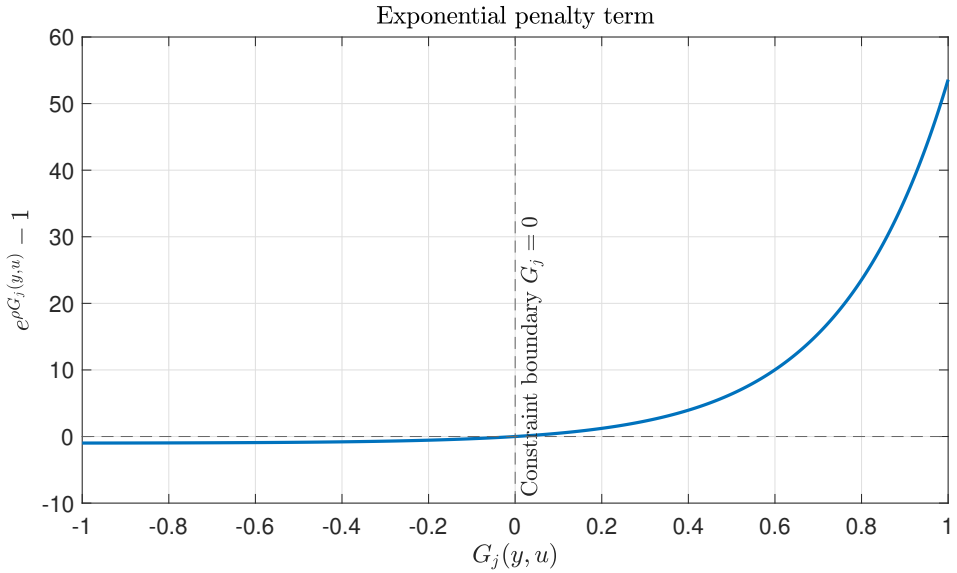



**Figure 3-1:** Exponential Penalty Term of $\mathcal{Q}_j(c,u) \leq 0$.

Following this, the augmented Lagrangian algorithm will take the following form:

$$(c^{k+1}, u^{k+1}) = \arg\min_{c,u} \widetilde{\mathcal{L}_{\rho_k}}(c, u, p^k, \mu^k) \tag{3-16a}$$

$$\begin{aligned} p^{k+1} &= p^k + \rho^k E(c^{k+1}, u^{k+1}), \\ \mu_j^{k+1} &= \mu_j^k \cdot \exp\left(\rho^k \mathcal{Q}_j(c^{k+1}, u^{k+1})\right), \\ \rho^{k+1} &= \Phi\left(\rho^k, c^{k+1}, c^k, u^{k+1}, u^k\right). \end{aligned} \tag{3-16b}$$

The update of the penalty parameter $\rho^k$ is dictated by the comparison of the norms of the residual of the equality constraints and norm of the violation of the inequality constraints. Between the outer iterations, whenever these values do not decrease fast enough, the parameter $\rho$ will have to be increased in general. Otherwise it keeps the same value.

So the proposed methodology provides a clear path to the handling of an optimisation problem with any type of equality and inequality constraint based on the rapid Newton-step algorithm. Nevertheless, it is a valid concern that in the All-at-Once approach, a large number of DoFs is required in order to achieve adequate accuracy in the numerical solution. As a result, the direct inversion of the Hessian matrix of the augmented Lagrangian function can be very costly computationally. To overcome this obstacle, domain decomposition techniques will be implemented in the Newton algorithm. These methodologies exploit the inner structure of PDEs to simplify them into smaller problems that can be solved in parallel. The idea to use DDM was drawn from the fact that the Hessian, that is derived from the augmented Lagrangian of a PDE-constrained optimisation problem, will share many similarities with the numerical solution of the same PDE. Since DDMs are used for the numerical solution of PDEs, then they could potentially speed up the derivation of the minima in the optimisation problem in Equation 3-16a.

## Synopsis of Chapter 3

The main goal of this chapter is to present the decision choices of the PDE-constrained optimisation algorithms that can accommodate the requirements of optimal control of air pollution, as discussed previously. Emphasis is given on the analysis of the techniques that can enhance the adaptability and robustness of the overall methodology such as the augmented Lagrangian approach with Newton step iterations. These methods in combination to the choice of the DtO technique and the All-at-Once approach motivate the implementation of domain decomposition methodologies that will accelerate the solution of the derived large linear systems.

# Chapter 4

# Domain Decomposition Methodologies

This chapter introduces the fundamental concepts of Domain Decomposition Methods (DDMs) and investigates their applicability to the optimal control of air pollution. Section 4-1 presents an overview of the DDM along with the motivation behind its selection for the solution of the large and sparse linear system derived from the augmented Lagrangian optimisation problem. Next, Section 4-2 provides insight on the general principles that are shared between various domain decomposition methods. Finally, in Section 4-3 the selected approach (substructuring method) is analysed, with emphasis on the numerical solution of a partial differential equation.

## 4-1 Overview of Domain Decomposition Methods

Domain Decomposition can be classified as a divide-and-conquer technique for the solution of partial differential equations using parallel architectures. The methodology is based on the premise that, by splitting the computational domain into smaller subdomains and solving the corresponding problems locally, it is possible to reconstruct the global solution. This leads to the creation of a family of subproblems with reduced size that are coupled either directly, through the values of the unknown variables at the interfaces, or weakly at the expense of introducing an iterative converging process among the subdomains [57] that guarantees the solution continuity.

This framework can be applied directly on "continuous" PDEs and leads to the decomposition of the global analytical solution, but it can also facilitate the efficient implementation of classical numerical techniques on discretised differential equations. In the continuous case, DDM assists by breaking down hard geometries into analytically solvable domains, localising singularities, and easily handling heterogeneous or discontinuous coefficients. In the discrete case, DDM provide an alternative solution to the robust but slow numerical solvers used for large algebraic systems [58, 59]. This implementation is far more interesting since, complicated and non-linear problems related to fluid dynamics, electromagnetism, continuum mechanics, and many other fields can only be solved via the discretisation of the underlaying equations. In

most cases, the accuracy of these simulations is directly related to the number of DOFs available in each method. As the spatial or temporal resolution increases, so does the dimension of the derived linear system, and as a result the computational costs related to processing power and memory requirements of the specified solver [14].

The reason why DDMs are very promising in numerical analysis, is based on the fact that they can seamlessly incorporate one very critical component into their algorithms: parallel computations. Each domain decomposition method shares the same principal idea, the splitting the problem's domain $\Omega$ into the subdomains $\Omega_1, \Omega_2, \ldots, \Omega_N$ and the computation of the solution of a PDE on each of the $\Omega_i$. The key aspect of this technique is that it coordinates the results from each of the subdomain to ensure their convergence to the original global solution [60]. This is achieved via the appropriate coupling through interface conditions and/or coarse grid corrections. Its main advantages are that it can be used to improve efficiency (with respect to time and memory), modularity (different solver or mesh in each region), and scalability while decreasing the demand for processing power. To that reason, different procedures have been developed over the years to cover a wide variety of applications.

## 4-2 General Principles of Domain Decomposition Methodologies

Even though a a wide range of domain decomposition methods have emerged, their analysis and formulation can be studied under a common framework. As the first step, a partial differential equation $\mathcal{L}u = f$ in $\Omega$ with the boundary condition $\mathcal{B}u = g$ on $\partial\Omega$ is assumed, with $\mathcal{L}$ denoting the differential operator and $\mathcal{B}$ the boundary operator of a well-posed problem. In order to decompose the domain, the subdomains $\Omega_i$ will be chosen such as $\Omega = \bigcup_{i=1}^{N} \Omega_i$ with interface $\Gamma_j := \partial\Omega_j \setminus \partial\Omega$. In the following examples, the case of two subdomains is presented ($\Gamma_i = \Gamma_j = \Gamma$), in order to simplify the comprehension of the discussed mathematical concepts. By introducing the operator $T\mathcal{T}_i$ that corresponds to the transmission/coupling conditions between the subdomains and the outward directions $n_i$ (from $\partial\Omega_i$) the following two distinct techniques arise.

#### Alternating DDM with general transmission conditions

Given $u_2^{(k)}$, the values $u_1^{(k+1)}$ followed by $u_2^{(k+1)}$ are computed sequentially from:

$$\begin{cases} \mathcal{L}u_1^{(k+1)} = f & \text{in } \Omega_1 \\ \mathcal{B}u_1^{(k+1)} = g & \text{on } \partial\Omega \cap \partial\Omega_1 \\ \mathcal{T}_1 u_1^{(k+1)} = \mathcal{T}_1 u_2^{(k)} & \text{on } \Gamma \end{cases} \qquad \begin{cases} \mathcal{L}u_2^{(k+1)} = f & \text{in } \Omega_2 \\ \mathcal{B}u_2^{(k+1)} = g & \text{on } \partial\Omega \cap \partial\Omega_2 \\ \mathcal{T}_2 u_2^{(k+1)} = \mathcal{T}_2 u_1^{(k+1)} & \text{on } \Gamma. \end{cases} \tag{4-1}$$

#### Parallel DDM with general transmission conditions

Given $u_1^{(k)}$, the values $u_2^{(k)}$, $u_1^{(k+1)}$ and $u_2^{(k+1)}$ are computed simultaneously from

$$\begin{cases} \mathcal{L}u_1^{(k+1)} = f & \text{in } \Omega_1 \\ \mathcal{B}u_1^{(k+1)} = g & \text{on } \partial\Omega \cap \partial\Omega_1 \\ \mathcal{T}_1 u_1^{(k+1)} = \mathcal{T}_1 u_2^{(k)} & \text{on } \Gamma \end{cases} \qquad \begin{cases} \mathcal{L}u_2^{(k+1)} = f & \text{in } \Omega_2 \\ \mathcal{B}u_2^{(k+1)} = g & \text{on } \partial\Omega \cap \partial\Omega_2 \\ \mathcal{T}_2 u_2^{(k+1)} = \mathcal{T}_2 u_1^{(k)} & \text{on } \Gamma. \end{cases} \tag{4-2}$$

The transmission conditions of both the aforementioned methodologies could receive (but are not limited) any of the presented forms:

$$\begin{aligned}
&\text{Dirichlet–Dirichlet (classical):} && \mathcal{T}_i = I \\
&\text{Dirichlet–Neumann :} && \mathcal{T}_i = I, \mathcal{T}_j = \partial_{n_j} \\
&\text{Neumann–Neumann:} && \mathcal{T}_i = \partial_{n_i} \\
&\text{Robin :} && \mathcal{T}_i = \partial_{n_i} + \alpha_i I \\
&\text{Ventcell (includes tangential derivatives):} && \mathcal{T}_i = \partial_{n_i} + \alpha_i I + \beta_i \Delta_\tau.
\end{aligned} \tag{4-3}$$

The expression $\partial_{n_i}$ is the normal derivative on $\partial\Omega_i$ and $\Delta_\tau$ is the Laplace operator on $\Gamma$. The choice of the interface operator, is one of the most important in the domain decomposition methods. It is responsible for the process of data exchange between the sub-domains and as a result is directly responsible for the convergence of the whole procedure. In the classical approaches facilitate the transmission of the Dirichlet conditions to enforce matching values at the interface region. However, this option may be a sub-optimal strategy or even not acceptable choice in complicated scenarios. This is a very important, since an ill-suited interface operator can lead to a dramatic drop in the solver's speed and accuracy or even cause the divergence of the solution. To address this issue different types of interface operators were introduced. Neumann, Robin, or even higher-order artificial conditions on the interface can be beneficial and increase the convergence rate of the whole procedure [61, 62].

### 4-2-1 Overlapping Domain Decomposition Methods

Another one main categorisation of DDMs is based on the distinction between overlapping and non-overlapping domains as shown in Figure 4-1[63]. In the case of overlapping decomposition, at least one subdomain $\Omega_i$ shares a common intersecting region with one or more other subdomains. As a result, information will be exchanged between those overlapping regions and this data will iteratively lead the whole system to the global solution. The simplest example of this methodology is the original alternating method proposed by H. Schwarz for the numerical solution of the Poisson equation in a geometrically complex 2D domain as seen in Algorithm 1. The formulation of the global answer is derived with data from each one of the subdomains as they are combined, along with a weighted average over the overlapping region.

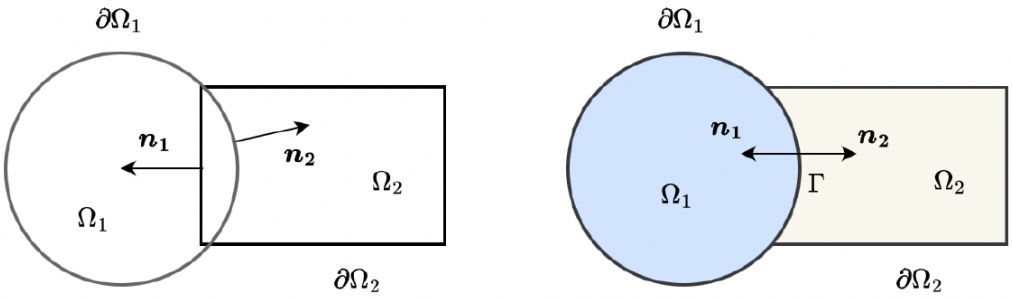



**Figure 4-1:** DDM with overlapping and non-overlapping subdomains.

Several extensions of the classical Schwarz approach have been developed since, such as the Additive Schwarz Method (ASM), Restricted Additive Schwarz (RAS), and Optimized Restricted Additive Schwarz (ORAS), with small differentiations on transmission conditions and sequence of updates. In these types of DDMs, the rate of convergence is not dependent only on the type of PDEs and the transmission conditions. Another prerequisite is the overlap of the subdomains. This requirement will increase the computational cost of the whole procedure since the same variables will have to be computed multiple times by different solvers in the shared domains. Generally, as the overlapping area increases, so does the convergence speed of the algorithm. This attribute hinders the scalability of this methodology.

**Algorithm 1** Classical overlapping Schwarz method (Alternating Method)

1: Choose an initial guess $u_2^{(0)}$ on $\Omega_2$
2: **For** $k = 0, 1, 2, \ldots$

$$\text{Solve in } \Omega_1: \quad \Delta u_1^{k+1} = f \quad \text{in } \Omega_1, \quad \text{with boundary conditions}$$
$$u_1^{k+1} = g \quad \text{on } \partial\Omega_1 \cap \partial\Omega, \qquad u_1^{k+1} = u_2^{k} \quad \text{on } \Gamma_1 : \partial\Omega_1 \cap \partial\Omega_2$$

$$\text{Solve in } \Omega_2: \quad \Delta u_2^{k+1} = f \quad \text{in } \Omega_2, \quad \text{with boundary conditions}$$
$$u_2^{k+1} = g \quad \text{on } \partial\Omega_2 \cap \partial\Omega, \qquad u_2^{k+1} = u_1^{k+1} \quad \text{on } \Gamma_2 : \partial\Omega_2 \cap \partial\Omega_1$$

Check convergence (e.g. interface residual or iterate difference)
**End For**

### 4-2-2 Non-Overlapping Domain Decomposition Methods

In non-overlapping domain decomposition methods, the partition of the domain leads to disjoint regions that only meet at their interfaces. The global solution is reconstructed from the sum of the local solutions of the subdomains and their continuity can be enforced via a variety of methods. For example, it can be enforced strongly via Neumann or/and Dirichlet boundary conditions or weakly via the introduction of additional constraints and variables in the subdomains' interface. The formulation of these interface conditions necessitates an in-depth analysis, since it encloses the fundamental properties of this approach. It can affect the convergence speed, the numerical size of the problem, the solver's robustness, and it must be specifically designed for most of the distinct cases.

In order to better understand the properties and requirements of this methodology, the previous case with two subdomains will be used again as a example. Assuming that the PDE under consideration is of second order, then in can be reformulated in the equivalent multi-domain form [61]:

$$\begin{gathered} \mathcal{L}u_1 = f \quad \text{in } \Omega_1, \qquad \mathcal{L}u_2 = f \quad \text{in } \Omega_2, \\ u_1 = g \quad \text{on } \partial\Omega_1 \cap \partial\Omega, \qquad u_2 = g \quad \text{on } \partial\Omega_2 \cap \partial\Omega \\ u_1 = u_2 \quad \text{on } \Gamma \\ \frac{\partial u_2}{\partial n} = \frac{\partial u_1}{\partial n} \quad \text{on } \Gamma. \end{gathered} \tag{4-4}$$

It is evident that in the presented framework, the optimal boundary conditions of the two subproblems are intertwined with each other. The partition between the subdomains $\Omega_1$ and $\Omega_2$ has led to the artificial interface $\Gamma$ and in order to solve the PDE in each of the subproblems, the knowledge of their boundary conditions (both in $\partial\Omega$ and $\Gamma$) is necessary. Wherever the subdomain's boundary coincides with the border of the whole domain $\Omega$, the boundary conditions are known. It is logical that, in those regions the properties of the local solution must be the same as those of the initial problem and thus, both the local and the global solutions must share the same boundary conditions ($u_i = g \quad \text{on} \quad \partial\Omega_i \cap \partial\Omega$).

However, the interface $\Gamma$ is not included in those regions. In this artificial boundary, the optimal conditions of each subdomain would be an 'absorbing boundary condition' [57], meaning an operator that is able to reproduce the precise response for each of the subdomain's adjacent regions. In order to achieve this in the $1^{st}$ sub-problem and compute $u_1$, it would require the computation of the normal derivatives from the $2^{nd}$ subdomain that were produced by the use of the $u_1$ as boundary condition on $\Gamma$. In other words, each sub-domain would require to have information regarding how its own solution affects the response of other sub-domains. The optimal coupling conditions can therefore be expressed through the **Steklov-Poincaré operator (S-P)** [61]. The idea behind it, is to eliminate the interior unknowns of a subdomain and produce its flux at the interface (similar to a Dirichlet-to-Neumann (DtN) operator).

If we define $S$ as the S-P operator of the presented example, the Equation 4-4 can be written as follows:

$$\begin{matrix} \mathcal{L}u_i = f \quad \text{in } \Omega_i \\ u_i = 0 \quad \text{on } \partial\Omega_i \cap \partial\Omega \\ u_i = \lambda \quad \text{on } \Gamma \quad \text{for i=1,2} \end{matrix} \quad \text{with} \quad \begin{matrix} S\lambda = \psi \quad \text{on } \Gamma \\ S\lambda := \dfrac{\partial}{\partial n}H_1\lambda - \dfrac{\partial}{\partial n}H_2\lambda, \quad \psi := \dfrac{\partial}{\partial n}T_2 f - \dfrac{\partial}{\partial n}T_1 f. \end{matrix} \tag{4-5}$$

The introduced operators $H_i$ and $T_i$ derive from the solution of the following PDEs:

$$\begin{cases} \mathcal{L}\left(H_i\lambda\right) = 0 & \text{in } \Omega_i \\ H_i\lambda = 0 & \text{on } \partial\Omega_i \cap \partial\Omega \\ H_i\lambda = \lambda & \text{on } \Gamma \end{cases} \quad \begin{cases} -\mathcal{L}\left(T_i f\right) = f & \text{in } \Omega_i \\ T_i f = 0 & \text{on } \partial\Omega_i \cap \partial\Omega \\ T_i f = 0 & \text{on } \Gamma. \end{cases} \tag{4-6}$$

Under the described circumstances, the use of the *Steklov-Poincaré operator* as a coupling condition allows the local solutions to converge to the global solution, by enforcing boundary conditions that make the interface 'invisible' and precisely reproduce the responses of the adjacent subdomains. To achieve this, the operator utilises the data not only from the subdomains' interface (typical DtN operator), but it requires the solution of an additional boundary value problem of the neighbouring subdomains. Its formulation depends strongly on the PDE in question as well as the domain of the problem. The analytical derivation of this operator is not straightforward and it can differ significantly depending on the initial problem.

To overcome these issues, instead of the ideal transmission conditions, a combination of lower order differential operators that are local and cheaper to implement can be used. This is performed in an attempt to estimate the S-P operator through asymptotic or series-based approximations. However, these imperfect interface conditions cannot remove every 'artificial reflection' of the error as it bounces of the interface and will only converge to zero iteratively

as the subproblems keep correcting each other. The prevailing non-overlapping methods that attempt this include the Neumann-Neumann (NN,BNN), the Balancing Domain Decomposition (BDD,BDDC), and the Finite Element Tearing and Interconnecting methods (FETI) [64].

The aforementioned difficulties associated with the formulation of the optimal interface conditions through the analytical computation of the Stekloc-Poincareóperator arise primarily at the continuous level. For discretised PDE formulations, the explicit derivation of the S-P operator is algebraically equivalent to the Schur complement of the locally interior variables. This observation consists the basis of the substructuring methods, which we present in the next section.

## 4-3 Algebraic Formulation of Substructuring Domain Decomposition Methods

Substructuring methods are a family of non-overlapping domain decomposition techniques that solve a linear system by eliminating subdomains' interiors DOFs. Closely connected to the Schur complement, this framework is implemented for the parallel computation of the solution of large and sparse linear systems that are derived from the discretization of PDEs. Their main principle lies in the local elimination of the interior degrees of freedom and the formulation of a system based on the interface's unknowns that will be equivalent to the global problem. According to the methodology that is used to enforce the interface continuity between the subdomains, these algorithms can be categorised as primal, dual, or hybrid.

A variety of processes can be used to discretise the physical problem (finite differences, finite elements, finite volume, etc.). Regardless of this choice, the linear system that emerges will have the general form $A\,\mathbf{u} = f$. Due to this specific choice of discretisation, the inner structure of the matrix $A$ will be similar to the Figure 4-2**a**.

Following the partition of the domain $\Omega$ into non-overlapping subdomains $\Omega = \bigcup_{i=1}^{N} \Omega_i$, the variables can be reordered accordingly. The unknowns DOFs of each subdomain $\Omega_i$ are divided into the interior DOFs $\mathbf{u}_{I_i}$ that lie either inside the subdomain or at the Neumann boundary ($\partial\Omega \cap \partial\Omega_i$) and the local interface DOFs $\mathbf{u}_{\Gamma_i}$ that lie at the subdomains' interface. Thus, for $N$ subdomains, the following system, that is equivalent to the previous $A\,\mathbf{u} = f$, arises:

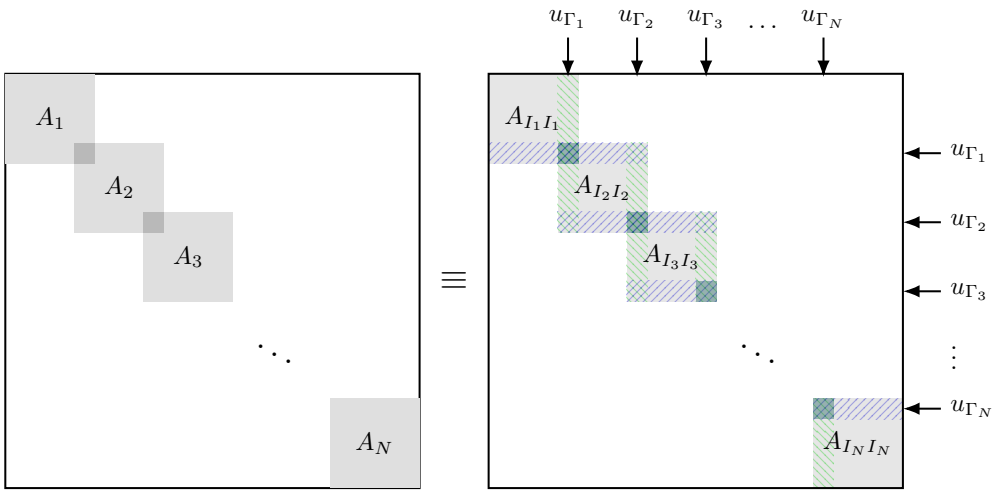
$$\underbrace{\begin{bmatrix} A_{I_1I_1} & 0 & 0 & 0 & A_{I_1\Gamma} \\ 0 & A_{I_2I_2} & \cdots & 0 & A_{I_2\Gamma} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & A_{I_NI_N} & A_{I_N\Gamma} \\ A_{\Gamma 1} & A_{\Gamma 2} & \cdots & A_{\Gamma N} & A_{\Gamma\Gamma} \end{bmatrix}}_{\mathbf{A}_R} \underbrace{\begin{bmatrix} \mathbf{u}_{I_1} \\ \mathbf{u}_{I_2} \\ \vdots \\ \mathbf{u}_{I_N} \\ \mathbf{u}_{\Gamma} \end{bmatrix}}_{\mathbf{u}_R} = \underbrace{\begin{bmatrix} F_{I_1} \\ F_{I_2} \\ \vdots \\ F_{I_N} \\ F_{\Gamma} \end{bmatrix}}_{\mathbf{F}_R} \implies \tag{4-7}$$

$$A_R\mathbf{u_R} = \mathbf{F_R}$$

with $\mathbf{u}_\Gamma = \begin{bmatrix} \mathbf{u}_{\Gamma_1}^\top & \mathbf{u}_{\Gamma_2}^\top & \dots, & \mathbf{u}_{\Gamma_N}^\top \end{bmatrix}^\top$. The matrix $A_R$ has now the inner structure seen in Figure 4-5. This was achieved via the identification of the interface variables and the reordering of all the DoFs so that they are placed at the bottom of $\mathbf{u_R}$ (Figure 4-3), followed by the consistent reordering of the equations of the linear system (Figure 4-4).

In order to enhance the understanding of this procedure, a schematic representation of the described processes is presented in the Figure 4-2, Figure 4-3, Figure 4-4 and Figure 4-5.

The discretisation of the PDE will produce the matrix A whose structure can be seen in Figure 4-2(**a**). The grey shaded regions represent subdomains' contributions whereas all the other white regions are equal to zero. Each sub-matrix $A_{I_i}$ can be derived from the discretisation of the same PDE at the $\Omega_i$ subdomain. But, in the reconstruction of the global $A$ matrix, the subdomains that share an interface, will contribute to the same area. This area is denoted with a darker shade of gray.




**(a)** Original Structure of Matrix $A$

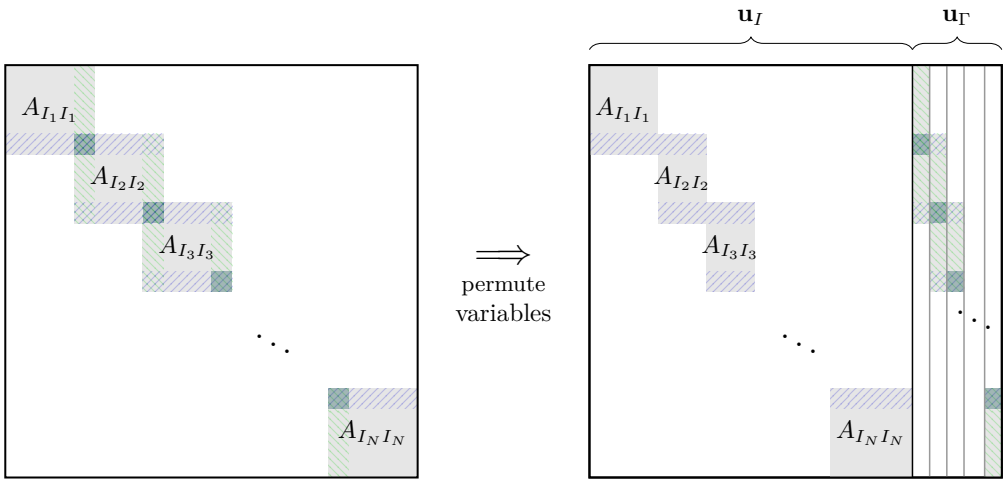
**(b)** Identification of Interface DoFs and Equations

**Figure 4-2:** The structure of the matrix $A$ derived from the discretisation of the PDE.

By identifying the associated interface unknowns (green columns), and the corresponding equations (blue rows) on Figure 4-2(**b**) or Figure 4-3(**a**), one can see how matrix A will have to change due to the dictated permutations.

At first, the permutation of the variables is performed with the help of the permutation matrix P, which will lead to the change of the A matrix as seen in Figure 4-3 and Equation 4-8:

$$A\mathbf{u} = f \xrightarrow{\text{Permute variable}} AP^{-1} \underbrace{P\mathbf{u}}_{\mathbf{u}_R} = f \implies AP^{-1}\mathbf{u}_R = f \tag{4-8}$$



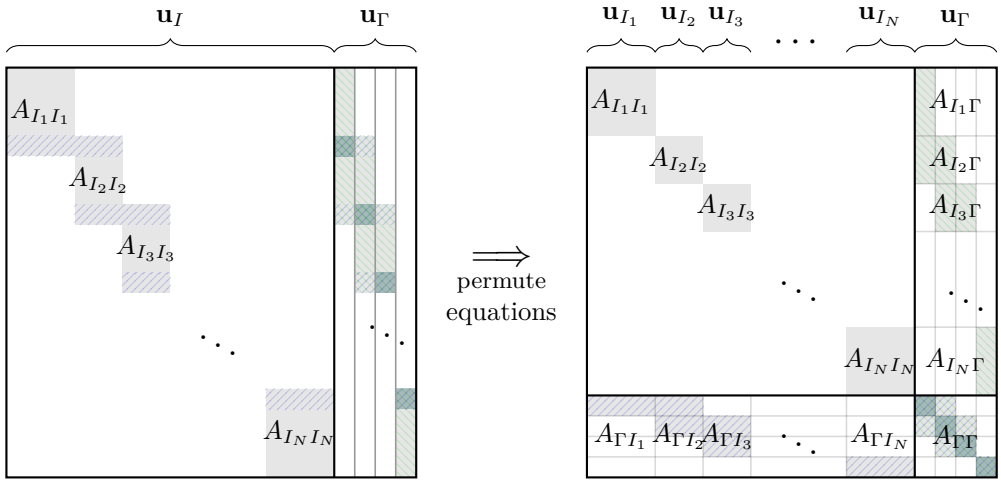

**(a)** Identification of Interface DoFs and Equations

**(b)** Permutation of the Variables

**Figure 4-3:** Identification and permutation of the interface degrees of freedom.

Then, the reordering of the equations is performed. These equations, that are associated with the original position of the interface variables, can be reordered successfully with the use of the same permutation matrix P. The final form $A_R$ can be seen in Figure 4-4 and Equation 4-9:

$$AP^{-1}\mathbf{u}_R = f \xrightarrow{\text{Permute equations}} \underbrace{PAP^{-1}}_{A_R}\mathbf{u}_R = \underbrace{Pf}_{F_R} \implies A_R\mathbf{u_R} = \mathbf{F_R} \tag{4-9}$$




**(a)** Permutation of the Variables

**(b)** Permutation of the Equations Structure of $A_R$

**Figure 4-4:** Permutation of the corresponding equations, resulting in the matrix $A_R$.

By eliminating all the interior $u_{I_i}$, the formulation of Schur Complement will lead to the following system:

$$S\,\mathbf{u}_\Gamma = g \quad \text{with} \qquad S = A_{\Gamma\Gamma} - \sum_{i=1}^{N} A_{\Gamma I_i} A_{I_i I_i}^{-1} A_{I_i \Gamma} \quad \text{and} \quad g = F_\Gamma - \sum_{i=1}^{N} A_{\Gamma I_i} A_{I_i I_i}^{-1} F_{I_i}. \tag{4-10}$$

The described reordering can be applied relatively easy once the original matrix $A$ has been formulated by the FEM software for the underlying PDE in the whole domain. However, it is also possible to parallelise the formulation of the Schur complement system. Instead of computing the matrix that corresponds to the whole discretised PDE, we can receive the necessary information by computing the equivalent matrices of the local discretised subproblems:

$$\underbrace{\begin{bmatrix} A_{I_i I_i} & A_{I_i \Gamma_i} \\ A_{\Gamma_i I_i} & A_{\Gamma_i \Gamma_i} \end{bmatrix}}_{\mathbf{A}_{\text{local}}} \underbrace{\begin{bmatrix} \mathbf{u}_{I_i} \\ \mathbf{u}_{\Gamma_i} \end{bmatrix}}_{\mathbf{u}_{\text{local}}} = \underbrace{\begin{bmatrix} F_{I_i} \\ F_{\Gamma_i} \end{bmatrix}}_{\mathbf{F}_{\text{local}}}. \tag{4-11}$$

Attention must be given to the presented matrices and their indexes. While $A_{I_i I_i}$ represent the same matrices seen in (4-7), there is a difference in the matrices with the $\Gamma_i$ index. They are specifically correlated to the interface variables of the $i^{th}$ subdomain. On the other hand, in (4-7) the $\Gamma$ index of the matrices refers to all the interface variables of the subdomains.

In the local system (4-11), in order to eliminate the interior DOFs of each independent solver, the inverse of $A_{I_i I_i}$ must be computed so as to form the local Schur complement (4-12):

$$\boxed{S_i\,\mathbf{u}_{\Gamma_i} = g_i} \qquad \begin{aligned} S_i &:= A_{\Gamma_i \Gamma_i} - A_{\Gamma_i I_i}\, A_{I_i I_i}^{-1} A_{I_i \Gamma_i} \\ g_i &:= F_{\Gamma_i} - A_{\Gamma_i I_i}\, A_{I_i I_i}^{-1} F_{I_i}. \end{aligned} \tag{4-12}$$

These local Schur operators are then used to reconstruct the global Schur complements system that couples all the subdomains as seen in Equation 4-13. Physically, the $S_i$ matrix computes the flux of the $u_i$ in the interface of the subdomains. Then the global interface system is formulated, based on the restriction operators $R_i$ that match the interface local DOFs to the global set of interface DOFs.

$$\boxed{S\,\mathbf{u}_\Gamma \;=\; g} \qquad S = \sum_{i=1}^{N} R_i^\top S_i R_i, \quad g = \sum_{i=1}^{N} R_i^\top g_i. \tag{4-13}$$

This smaller interface problem is the discretised version of the S-P operator. Because each interface node carries one global DOF, the continuity across all subdomains is enforced strongly. The restriction operators $R$ make sure that all the different equations, from each subdomain that are related to the same node, will collapse to a single expression. The dimension of this linear system ends up equal to the total number of interface DOFs, that is often many orders of magnitude smaller than the initial system. Once the numerical solver converges, each interior unknown $u_{I_i}$ can be computed locally as follows:

$$u_{I_i} \;=\; A_{I_i I_i}^{-1}(F_{I_i} - A_{I_i \Gamma_i} u_{\Gamma_i}), \qquad i = 1, 2, \ldots, N. \tag{4-14}$$

In the general case, Figure 4-5 shows how the reordered matrix is obtained after grouping the interior degrees of freedom of each subdomain and placing the interface degrees of freedom on the bottom. The diagonal blocks, inside the dotted line, correspond to the subdomains' interior unknowns, while the final rows and columns represent their coupling with the DoFs on the interface. An example of the substructuring methods is seen in Algorithm 2.

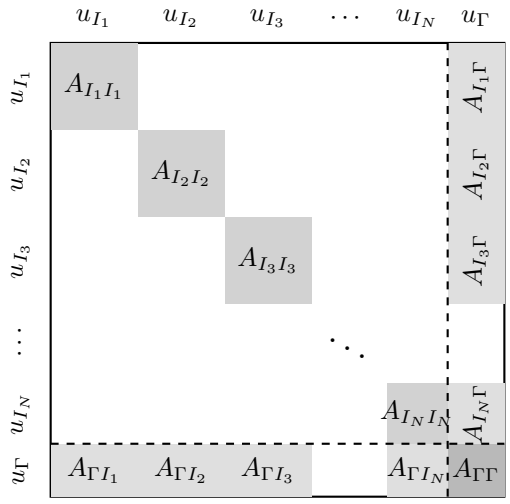



**Figure 4-5:** Reordered matrix $A_R$

Following the discretisation of the PDE, the described algorithmic steps describe how this non-overlapping domain decomposition method, with optimal transmission conditions, boils down to the solution of the interface system $S\mathbf{u}_\Gamma = g$. However, this system is not as easy to solve as it seems. The size of the matrix $S$ is directly related to the number of the subdomains' interface DoFs. It is much lower than the total number of DoFs but it can still rise dramatically as the size of the whole domain rises, as the number of its partitions increases, or even as the mesh size reduces. In addition, after eliminating all interior DoFs, $S$ is a dense matrix and, as a result, its formulation and storage is expensive in terms of processing power and required memory. A direct solver will work but it will require reasonable time and resources so an iterative framework (Krylov solver) may be more attractive depending on the application. Even then, the advantage of parallel computation can be used not only in the formulation of $S$, but in the creation of a valid preconditioner [65].

Based on the previous analysis, there are indications that the substructuring methods can be utilised in the solution of PDE-constrained optimisation problems associated with air-pollution. This is based to the fact that, the large sparse algebraic system from the discretisation of the governing transport equations shares similar characteristics to the linear system derived from the Newton step of the discretised augmented Lagrangian. The structure of the algebraic system of a PDE allows the elimination of the interior degrees of freedom of each subdomain locally. It is shown that the global problem can be reduced to a Schur-complement system, dependent only on the interfaces variables. The same logic can be attempted in the linear system that is derived from the optimisation with respect to the interface state variables and the control inputs. This structure is advantageous for optimisation schemes with

multiple iterations since the same domain partition can be reused throughout the optimisation process and, contrary to the overlapping techniques that duplicate some DoFs, the total number of variables does not increase. As a result, substructuring methods could potentially reduce both computational time and memory requirements while improving the scalability of large-scale optimisation frameworks for air-pollution control.

**Algorithm 2 : Substructuring Domain Decomposition Method**

1: **Splitting of the whole domain $\Omega$** Partition of the global mesh into non-overlapping subdomains $\{\Omega_k\}_{k=1}^{K}$ and computation of the local matrices and vectors $(A_k, F_k)$.

2: **Local DOFs identification on each subdomain:** Reorder unknowns into interior $I_k$ and interface $\Gamma_k$ and remove the DOFs associated with global constraints :

$$A_k = \begin{bmatrix} A_{I_k I_k} & A_{I_k \Gamma_k} \\ A_{\Gamma_k I_k} & A_{\Gamma_k \Gamma_k} \end{bmatrix}, \qquad F_k = \begin{bmatrix} F_{I_k} \\ F_{\Gamma_k} \end{bmatrix}.$$

3: **Compute local $g_k$ and assemble global g:**

$$g \leftarrow \sum_{k=1}^{K} R_k^\top g_k \qquad g_k = F_{\Gamma_k} - A_{\Gamma_k I_k} (A_{I_k I_k})^{-1} F_{I_k}$$

4: **Compute local $S_k$ and assemble the global Schur Complement operator:**

$$S \leftarrow \sum_{k=1}^{K} R_k^\top S_k R_k, \qquad S_k := A_{\Gamma_k \Gamma_k} - A_{\Gamma_k I_k} (A_{I_k I_k})^{-1} A_{I_k \Gamma_k}.$$

▷ Discrete equivalent to Steklov-Poincare Operator

5: **Solve the interface linear system:**

$$S u_\Gamma = g$$

▷ formulation of preconditioner M is possible

6: **On each domain $k$, compute the local interior DOFs :**

$$u_{I_i} = A_{I_i I_i}^{-1} (F_{I_i} - A_{I_i \Gamma_i} u_{\Gamma_i})$$

7: **Return complete solution** $u = (u_{I_1}, \ldots, u_{I_K}, u_\Gamma)$.

## Synopsis of Chapter 4

The main goal of this chapter is the familiarisation of the reader with the main principles of the domain decomposition methods. Emphasis is given to the substructuring techniques, as an alternative methodology for the solution of the large and sparse linear systems which are derived from the discretisation of partial differential equations. By exploiting the inner structure of the discretised matrix $A$, the whole procedure can be parallelised and thus improve the overall computational efficiency. This characteristic is what motivates the selection of DDM as an approach that can be extended to the proposed PDE-constrained optimisation problem, in order to mitigate its cost and improve its performance.

# Chapter 5

# A Domain-Decomposition Framework for PDE-Constrained Optimisation

In this chapter the implementation of domain decomposition methods on the framework of PDE-constrained optimisation is analysed. In Section 5-1, an overview of existing methodologies is presented. Then the formulation of the proposed method is introduced. Section 5-2 describes in detail all the decision choices, their effects on a generalised case of OCP and the steps of the algorithm that will generate the numerical solution of the optimisation problem.

## 5-1 Existing Domain Decomposition Strategies for PDE-Constrained Optimisation

As the underlying physical models in optimisation schemes become larger and more complicated, there exists a compelling need for faster and more efficient numerical solvers. The conducted analysis of the previous chapters has revealed that the interior structure and attributes of DDMs have many similarities to the derived linear systems of these optimisation problems. In addition DDMs support parallel computations which makes them ideal for the formulation of a scalable numerical framework. There have been many attempts to combine PDE-constrained optimisation and domain decomposition methodologies but the derived approaches are restrictive in terms of the type of state equation, cost functional and, more importantly, in the existence of additional constraints. The main categories of the already established DDM implementations are outlined in the following section.

#### Domain Decomposition Methods in Reduced-Space Optimisation

One of these approaches is based on overcoming the practical limitations of the reduced-space formulation of PDE constrained optimisation problems. A typical strategy for the simplification of these optimisation problems, consists of the elimination of the state variables

and the implementation of a suitable optimiser with respect to only the control variables. This procedure is then followed by the discretisation of the whole system. For example, the minimisation of a cost functional $\mathcal{J}(c,u)$ that is subject to a state equation, denoted as $\mathcal{E}$, could be replaced via an equivalent optimisation problem with the $u$ as its exclusive decision variable (5-1). The reformulation

$$\min_{u,c} \mathcal{J}(c,u) \;\; s.t.\; \mathcal{E}(c,u)=0 \implies \min_{u} \mathcal{J}(c(u),u) := \min_{u} \widehat{\mathcal{J}}(u) \tag{5-1}$$

however, means that each evaluation of the objective function and its derivatives would require the solution of the state equation. This solution can be derived via the discretisation of the PDE and must be performed at every iteration. In some applications, the use of complicated, non-linear equations with tens of thousands (if not more) degrees of freedom is not uncommon. As a result, the reduced-space approach of the optimisation scheme would grind a halt since PDE's solution constitutes a numerical bottleneck. This is exactly where the DDMs can intervene, as they were primarily developed as a fast and parallel alternative solver of PDEs. Following the computation of the state $c$ from the values of $u$ at each iteration (through the solution of $\mathcal{E}(c,u)=0$), the minimisation of the cost function has yet to be performed. To that end, one can use typical optimisation solvers such as gradient descent, conjugate gradient, etc [66, 67, 68].

Another characteristic approach is the formulation of Lagrangian and the use of the adjoint function $p$ :

$$\begin{aligned} &\mathcal{L}(c,u,p) = \mathcal{J}(c,u) + \langle \mathcal{E}(c,u),\, p\rangle \\ &\textit{Computation of the adjoint } p \quad \mathcal{L}_c(c,u,p) := {}^{\partial\mathcal{L}}/_{\partial c} = 0 \implies \\ &\mathcal{J}_c(c,u) + \mathcal{E}_c(c,u)^* p = 0 \implies \mathcal{E}_c(c,u)^* p = -\mathcal{J}_c(c,u) \end{aligned} \tag{5-2}$$

so as to derive the necessary information regarding the sensitivity of the cost functional with respect to $u$. The PDE that describes $p$ has almost identical characteristics with the state equation. Consequently, following the replacement of $c$ with $c(u)$, DDMs could be used in this case to parallelise and speed up the computation procedure of $p$ [69]. More details on this technique are provided in Appendix A-1-1.

### Domain Decomposition Methods in the Coupled Optimality System

A different case of the integration of DDM into the PDE constrained optimisation framework, is based on the principle of *optimise-then-decompose*. More specifically, the use of of the aforementioned Lagrangian (5-2) will introduce the adjoint function $p$ and assist in formulating the optimality conditions. The search of the stationary points of the produced Lagrangian $\mathcal{L}$ is equivalent to the search of potential optimal states of the cost functional $\mathcal{J}$. The first-order optimality conditions (KKT) can be written as follows:

$$\begin{aligned} \nabla_p \mathcal{L}(c,u,p) &= \mathcal{E}(c,u) = 0 && \text{state equation} \\ \nabla_c \mathcal{L}(c,u,p) &= \mathcal{J}_c(c,u) + \mathcal{E}_c(c,u)^* p = 0 && \text{adjoint equation} \\ \nabla_u \mathcal{L}(c,u,p) &= \mathcal{J}_u(c,u) + \mathcal{E}_u(c,u)^* p = 0 && \text{control optimality condition} \end{aligned} \tag{5-3}$$

In some cases the optimal control $u^*$ can be explicitly derived from the optimality condition $\nabla_u \mathcal{L}(c,u,p)=0$. The optimal control will be dependent on the state and adjoint variables,

and when it is applied, it is guaranteed to drive the system to its optimal state. Unlike the previous example where the only decision variable that remained was $u$, here it will be substituted. Replacing the $u$ with the optimal input $u^*$ in the other two optimality conditions will result into a coupled system of PDEs dependent only on $c$ and $p$. With the use of DDM, this system can then be decomposed into subdomains under suitable interface conditions and be solved either iteratively or in parallel [70, 15]. After its convergence, the the state and the adjoint can be reconstructed from all the local solutions. Additional information can be found in Appendix A-1-2.

### Domain Decomposition in the Optimal Control Problem

Finally, another option for the integration of DDM into a PDE-constrained optimisation framework is based on a *decompose-then-optimise* strategy that provides greater flexibility in the design of the optimisation algorithm and can result in increased efficiency and scalability. At first, the underlying PDE domain is decomposed into a set of subdomains. For any PDE, the solution of the state variable $c$ depends not only on the control $u$, but also on the conditions on the boundary of the whole domain. These conditions are dictated by the the application requirements and the phenomenon that is described. However, the physical boundary of each subdomain is not equivalent with the physical boundary of the whole problem. As a result, each of the local problems will not be well-defined unless further conditions are implemented to their interface boundary with other subdomains. Thus, local solution $c_i$ at each subdomain will be dependent on the local input $u_i$, acting as control, and the local interface variables $c_{\Gamma_i}$ or $^{\partial c_{\Gamma_i}}/_{\partial n}$, acting as boundary conditions on the interface. Now, the local problems are well defined and their solution can be expressed as a function of the local controls $u_i$ and the interface value $c_{\Gamma_i}$ or $^{\partial c_{\Gamma_i}}/_{\partial n}$ [23]. Nevertheless, it is not guaranteed that the synthesis of these solutions will be equivalent to the global solution. In order to achieve this, the following equation must hold true:

$$\begin{aligned} \mathcal{E}(c_i, u_i) = 0 \quad \text{in } \Omega_i, \qquad & \mathcal{E}(c_j, u_j) = 0 \quad \text{in } \Omega_j, \\ c_i = g \quad \text{on } \partial\Omega_i \cap \partial\Omega, \qquad & c_j = g \quad \text{on } \partial\Omega_j \cap \partial\Omega \\ c_i = c_j \quad \text{on } \Gamma_{ij} \qquad & \frac{\partial c_j}{\partial n_j} = -\frac{\partial c_i}{\partial n_i} \quad \text{on } \Gamma_{ij} \end{aligned} \tag{5-4}$$

with $g$ denoting the global boundary conditions and the last two equations representing the continuity between the solutions of different subdomains. Based on that, the local problems can be defined and are equivalent with the global problem. Then, the optimisation problem can be reduced and expressed based, not only on the local inputs $u_i$ but also on either the value of $c$ or its outward flux on the interface between the subdomains. The term that is chosen as a decision variable can be imbedded into the optimisation problem in a way that it a priori satisfies one of the conditions of continuity in (5-4). The other term enters the optimisation problem as an additional constraint. As a result, this methodology formulates a reduced optimisation problem with decision variables the local inputs $u_i$ and a corresponding interface term along with an additional constraints to enforce continuity. The corresponding cost functional will be equal to the sum of the original objective function at each subdomain. Domain decomposition methods can be utilised for the parallel computation of the interior state variables $c_I$. This technique is discussed in greater detail in Appendix A-1-3.

## 5-2 Formulation of the Proposed Methodology

The proposed optimisation methodology is intended to provide a general framework for the solution of optimisation problems governed by partial differential equations. Unlike the described attempts, rather than restricting the formulation to a particular objective functional, state equation, or set of constraints, the problem is initially considered in its abstract form. This general formulation will allow us to incorporate a broad range of equality and inequality constraints, something that is absent from the existing methodologies lack of. Taking into consideration all the information presented in Chapter 3, the general representation of the optimisation problem will be:

$$\begin{aligned} &\min_{c\in V, u\in U} \mathcal{J}(c,u) \\ \text{s.t.} \quad &\mathcal{E}(c,u)=0, \quad \mathcal{D}(c,u)=0, \quad \mathcal{Q}(c,u)\leq 0 \end{aligned} \tag{5-5}$$

where $c$ denotes the state variable, $u$ the control variable, and $\mathcal{J}: V \times U \to \mathbb{R}$ an arbitrary objective functional. The $\mathcal{E}$ operator represents the underlying PDE along with the boundary conditions whereas the operators $\mathcal{D}$ and $\mathcal{Q}$ describe any additional equality and inequality constraints, respectively.

### 5-2-1 Discretisation of the Optimisation Problem

The chosen optimisation method will be based on a *discretize-then-optimize* strategy in contrast to the aforementioned implementations of domain decomposition at PDE-constrained optimisation of Section 5-1. This approach offers greater flexibility in the handling of the optimisation terms and thus it could reveal similarities between the optimisation problem and the solution of the underlying PDE. The identification of similarities in their structure will provide insight on novel ways to decompose the OCP and exploit it via DDM. For the same reason, a *all at once* technique will be utilised and both the state and the control will be treated as optimisation variables. For the first step, the governing PDE, the objective functional, as well as the constraints functions are discretised. By doing so, the original infinite-dimensional problem is reformulated into a finite-dimensional optimisation problem. The discretisation is performed utilising one of the various finite element methods. This choice was made due to the ability of FEM to accurately capture a wide range of PDEs along with their adaptability to various requirements and domain changes. Furthermore, the weak formulation of the PDEs, which forms the basis of the mathematical analysis of optimal control problems, can be naturally intergraded into FEM.

The discrete state and control spaces, denoted as $V_h$ and $U_h$ respectively, are finite-dimensional subspaces of the continuous spaces $V$ and $U$. They are derived through the basis functions $\phi_i(\mathbf{x})$ and $\psi_i(\mathbf{x})$ which result in the approximation of $c$ and $u$ as follows:

$$\begin{aligned} V_h \subset V \qquad U_h \subset U \qquad V_h = \text{span}\{\phi_1,\dots,\phi_{n_c}\} \quad U_h = \text{span}\{\psi_1,\dots,\psi_{n_u}\} \\ c_h(\mathbf{x}) = \sum_{i=1}^{n_c} c_i\,\phi_i(\mathbf{x}) \in V_h \qquad u_h(\mathbf{x}) = \sum_{j=1}^{n_u} u_j\,\psi_j(\mathbf{x}) \in U_h \end{aligned} \tag{5-6}$$

Here, $c_i$ and $u_j$ denote the degrees of freedom ($n_c$ and $n_u$ collectively) that are associated with the corresponding basis functions. For a more concise mathematical representation, they will

be grouped into the following vectors:

$$\mathbf{c} = \begin{bmatrix} c_1 & c_2 & \cdots & c_{n_c} \end{bmatrix}^T \in \mathbb{R}^{n_c} \qquad \mathbf{u} = \begin{bmatrix} u_1 & u_2 & \cdots & u_{n_u} \end{bmatrix}^T \in \mathbb{R}^{n_u}.$$

After the substitution of these expressions in the formulation of Equation 5-5 the following finite dimensional optimisation problem arises.

$$\begin{aligned} &\min_{\mathbf{c},\mathbf{u}} \mathcal{J}_h(\mathbf{c},\mathbf{u}) \\ &\text{s.t.} \quad \mathcal{E}_h(\mathbf{c},\mathbf{u}) = 0, \quad \mathcal{D}_h(\mathbf{c},\mathbf{u}) = 0, \quad \mathcal{Q}_h(\mathbf{c},\mathbf{u}) \leq 0 \end{aligned} \tag{5-7}$$

In Equation 5-7, $\mathcal{E}_h$ represents the discrete form of the state equation PDE, $\mathcal{J}_h$ denotes the discretised term associated with the cost functional and $\mathcal{D}_h$ includes any additional discretised equality constraints. Special attention should be given to the term $\mathcal{Q}_h$, which is associated with the inequality constraints of the continuous optimisation problem. The discretisation of these expressions is not as simple as substituting $c_h(\mathbf{x})$ and $u_h(\mathbf{x})$ and then solving for $\mathbf{c}$ and $\mathbf{u}$. The reformulation of the continuous inequality constraints (e.q. $c_h(\mathbf{x}) \leq c_{max}$) into a set of nodal discreet constraints (e.q. $c_i \leq c_{max}$) is equivalent only under specific assumptions associated with the finite element discretisation. The use of $P_0$ or $P_1$ Lagrange-type elements guarantees that the value of the polynomial basis function will not exceed the value at the node and, as a result, the aforementioned reformulation is valid. However, there are instances where the application requires the use of $P_2$ or higher order elements to guarantee the accurate simulation of the underlying physics. Then nodal constraints are not sufficient unless further simplifications are made or different basis functions (e.q. Bernstein-Bézier) are utilised.

### 5-2-2 Formulation of the Augmented Lagrangian Method

After the discretisation, the resulting finite dimensional and constrained optimisation problem can be solved with typical optimisation techniques. To that end, in order to transform it into an unconstrained optimisation problem, an augmented Lagrangian methodology will be implemented. It is worth mentioning, that any degrees of freedom that are allocated to the compliance of Dirichlet boundary conditions or associated with any additional the equality constraints ($\mathcal{D}_h$) will have to be eliminated from the problem or be treated numerically. For the sake of compactness, it will be assumed that the $\mathcal{D}_h$ is incorporated into the $\mathcal{E}_h$ term. Under this simplification, all the equality constraints contribute two terms in the augmented Lagrangian, one multiplier term and one quadratic penalty term.

On the other hand, the treatment of the inequality constraints is not so straightforward. Typical methods based on the introduction of additional slack variables that will not only increase the size of the problem but produce some strictly once-differentiable terms as seen in Section 3-2. The main drawback of this approach is the exclusion of Newton optimisation methods since there exist areas in the space, where the Hessian is not well-defined. As a result, either approximations (semi-smooth Newton) or different types of optimisation techniques should be used. In order to avoid this and be able to apply the rapidly convergent Newton algorithm, a different approach is proposed, and a continuously differentiable exponential penalty function is introduced 3-2. Taking into consideration all the attributes mentioned

above, the augmented Lagrangian for the general discrete problem can be written as seen below:

$$\mathcal{L}_\rho(\mathbf{c},\mathbf{u},\lambda,\mu) = \mathcal{J}_h(\mathbf{c},\mathbf{u}) + \lambda^T\mathcal{E}_h(\mathbf{c},\mathbf{u}) + \frac{\rho}{2}\mathcal{E}_h(\mathbf{c},\mathbf{u})^T\mathcal{E}_h(\mathbf{c},\mathbf{u}) + \\ + \sum_{j=1}^{m}\frac{\mu_j}{\rho}\left[\exp\left(\rho\mathcal{Q}_h(\mathbf{c},\mathbf{u})_j\right) - 1\right]. \tag{5-8}$$

In the previous equation $\lambda$ denotes the multipliers associated with the equality constraints, whereas $\mu_i$ represent the multipliers associated with the inequality constraints whose total number is adding to $m$ (number of inequality constraints). Finally, the new term $\rho$ is the penalty parameter of the optimisation methodology.

So far, the described technique offers a systematic approach to create the augmented Lagrangian equation from the minimisation problem of a cost function under any type equality and inequality constraints. The next steps of the methodology require the solution of equivalent optimisation problem seen in Equation 5-10.

$$\begin{aligned} &\min_{\mathbf{c},\mathbf{u}} \mathcal{J}_h(\mathbf{c},\mathbf{u}) \\ \text{s.t.}\quad &\mathcal{E}_h(\mathbf{c},\mathbf{u}) = 0 \quad \mathcal{Q}_h(\mathbf{c},\mathbf{u}) \leq 0 \end{aligned} \quad \xrightarrow{\text{Augmented Lagrangian}} \quad \mathcal{L}_\rho(\mathbf{c},\mathbf{u},\lambda,\mu) \tag{5-9}$$

The resulting optimisation procedure is composed by two reoccurring steps. At first, for fixed values of the Lagrange multipliers and penalty parameters, an unconstrained minimisation of the augmented Lagrangian is performed with respect to the state and control variables. Once an acceptable minimiser has been obtained, the multipliers and the penalty parameter will be updated according to the residuals of the corresponding constraints. This process will be repeated until all the preset optimality criteria are satisfied.

Step 1 - Inner Iteration: Solve the optimisation problem with constant $\lambda, \mu$

$$\left(\mathbf{c}^{k+1},\mathbf{u}^{k+1}\right) = \arg\min_{\mathbf{c},\mathbf{u}} \mathcal{L}_\rho\left(\mathbf{c}^k,\mathbf{u}^k,\lambda^k,\mu^k\right)$$

Step 2 - Outer Iteration: Update the multipliers and penalty parameters

$$\begin{aligned} \lambda^{k+1} &= \lambda^k + \rho^k\,\mathcal{E}_h\left(\mathbf{c}^{k+1},\mathbf{u}^{k+1}\right) \\ \mu_j^{k+1} &= \mu_j^k \exp\left(\rho^k\,\mathcal{Q}_h\left(\mathbf{c}^{k+1},\mathbf{u}^{k+1}\right)_j\right) \quad j = 1,\ldots,m \\ \rho^{k+1} &= f\left(\rho^k,\mathbf{c}^{k+1},\mathbf{u}^{k+1},\lambda^{k+1},\mu^{k+1}\right) \end{aligned} \tag{5-10}$$

### 5-2-3 Newton-Step Update of the Inner Optimisation Problem

In order to compute the minimiser determined by the first step of (5-10), a Newton method is proposed. Newton methods exploit additional information associated with the Hessian matrix and can exhibit substantially faster local convergence, compared to simple gradient methods. However, each iteration demands the solution of a larger and more complicated linear system. More specifically, at the iteration $k$, the gradient $g_k$ and the Hessian $H_k$ of the augmented

Lagrangian with respect to the optimisation variables $\mathbf{c}$ and $\mathbf{u}$ are formulated while all the other parameters are treated as constants. Then, the Newton step $[\Delta\mathbf{c}, \Delta\mathbf{u}]^T$ is computed based on the following equation:

$$\boxed{H_k\left(\mathbf{c}^k, \mathbf{u}^k\right)\begin{bmatrix}\Delta\mathbf{c}\\ \Delta\mathbf{u}\end{bmatrix} = -g_k\left(\mathbf{c}^k, \mathbf{u}^k\right)} \tag{5-11}$$

By denoting:

$$E_c := \frac{\partial \mathcal{E}_h}{\partial \mathbf{c}}\left(\mathbf{c}^k, \mathbf{u}^k\right) \quad E_u := \frac{\partial \mathcal{E}_h}{\partial \mathbf{u}}\left(\mathbf{c}^k, \mathbf{u}^k\right) \quad \text{and} \quad s_j^k := \mu_j^k \exp\left(\rho \mathcal{Q}_h\left(\mathbf{c}^k, \mathbf{u}^k\right)_j\right)$$

the gradient $g_k$ is computed from the differentiation of the augmented Lagrangian function (5-8), with respect to $\mathbf{c}$ and $\mathbf{u}$ decision variables:

$$\boxed{\begin{aligned} g_k\left(\mathbf{c}^k, \mathbf{u}^k\right) := \nabla_{(\mathbf{c},\mathbf{u})}\mathcal{L}_\rho\left(\mathbf{c}^k, \mathbf{u}^k\right) := \begin{bmatrix}\nabla_{\mathbf{c}}\mathcal{L}_\rho\left(\mathbf{c}^k, \mathbf{u}^k\right)\\ \nabla_{\mathbf{u}}\mathcal{L}_\rho\left(\mathbf{c}^k, \mathbf{u}^k\right)\end{bmatrix} \quad \textit{with}\\ \nabla_{\mathbf{c}}\mathcal{L}_\rho\left(\mathbf{c}^k, \mathbf{u}^k\right) = \nabla_{\mathbf{c}}\mathcal{J}_h\left(\mathbf{c}^k, \mathbf{u}^k\right) + E_c^T\left(\lambda^k + \rho\mathcal{E}_h\left(\mathbf{c}^k, \mathbf{u}^k\right)\right) + \sum_{j=1}^{m} s_j^k \nabla_{\mathbf{c}}\mathcal{Q}_h\left(\mathbf{c}^k, \mathbf{u}^k\right)_j\\ \nabla_{\mathbf{u}}\mathcal{L}_\rho\left(\mathbf{c}^k, \mathbf{u}^k\right) = \nabla_{\mathbf{u}}\mathcal{J}_h\left(\mathbf{c}^k, \mathbf{u}^k\right) + E_u^T\left(\lambda^k + \rho\mathcal{E}_h\left(\mathbf{c}^k, \mathbf{u}^k\right)\right) + \sum_{j=1}^{m} s_j^k \nabla_{\mathbf{u}}\mathcal{Q}_h\left(\mathbf{c}^k, \mathbf{u}^k\right)_j \end{aligned}} \tag{5-12}$$

The Hessian will be computed in a similar manner from the differentiation of the gradient of the same augmented Lagrangian function. It is derived via the following formula:

$$\boxed{H_k\left(\mathbf{c}^k, \mathbf{u}^k\right) := \nabla^2_{(\mathbf{c},\mathbf{u})}\mathcal{L}_\rho\left(\mathbf{c}^k, \mathbf{u}^k\right) := \begin{bmatrix}\nabla^2_{\mathbf{cc}}\mathcal{L}_\rho\left(\mathbf{c}^k, \mathbf{u}^k\right) & \nabla^2_{\mathbf{cu}}\mathcal{L}_\rho\left(\mathbf{c}^k, \mathbf{u}^k\right)\\ \nabla^2_{\mathbf{uc}}\mathcal{L}_\rho\left(\mathbf{c}^k, \mathbf{u}^k\right) & \nabla^2_{\mathbf{uu}}\mathcal{L}_\rho\left(\mathbf{c}^k, \mathbf{u}^k\right)\end{bmatrix}} \tag{5-13a}$$

By denoting some additional terms,

$$E_{\mathbf{cc}} := \nabla^2_{\mathbf{cc}}\mathcal{E}_h\left(\mathbf{c}^k, \mathbf{u}^k\right) \quad E_{\mathbf{cu}} := \nabla^2_{\mathbf{cu}}\mathcal{E}_h\left(\mathbf{c}^k, \mathbf{u}^k\right) \quad E_{\mathbf{uc}} := \nabla^2_{\mathbf{uc}}\mathcal{E}_h\left(\mathbf{c}^k, \mathbf{u}^k\right) \quad E_{\mathbf{uu}} := \nabla^2_{\mathbf{uu}}\mathcal{E}_h\left(\mathbf{c}^k, \mathbf{u}^k\right)$$
$$D_s^k := \operatorname{diag}\left(s_1^k, \ldots, s_m^k\right) \quad Q_{\mathbf{c}} := \nabla_{\mathbf{c}}\mathcal{Q}_h\left(\mathbf{c}^k, \mathbf{u}^k\right) \quad Q_{\mathbf{u}} := \nabla_{\mathbf{u}}\mathcal{Q}_h\left(\mathbf{c}^k, \mathbf{u}^k\right)$$
$$Q_{\mathbf{cc}} := \nabla^2_{\mathbf{cc}}\mathcal{Q}_h\left(\mathbf{c}^k, \mathbf{u}^k\right) \quad Q_{\mathbf{cu}} := \nabla^2_{\mathbf{cu}}\mathcal{Q}_h\left(\mathbf{c}^k, \mathbf{u}^k\right) \quad Q_{\mathbf{uc}} := \nabla^2_{\mathbf{uc}}\mathcal{Q}_h\left(\mathbf{c}^k, \mathbf{u}^k\right) \quad Q_{\mathbf{uu}} := \nabla^2_{\mathbf{uu}}\mathcal{Q}_h\left(\mathbf{c}^k, \mathbf{u}^k\right)$$

the blocks of the Hessian can be formulated as follows:

$$
\begin{aligned}
\nabla^2_{\mathbf{cc}}\mathcal{L}_\rho\left(\mathbf{c}^k,\mathbf{u}^k\right) &= \nabla^2_{\mathbf{cc}}\mathcal{J}_h\left(\mathbf{c}^k,\mathbf{u}^k\right) + \rho E_{\mathbf{c}}^T E_{\mathbf{c}} + \sum_{i=1}^{n}\left[\left(\lambda^k + \rho\mathcal{E}_h\left(\mathbf{c}^k,\mathbf{u}^k\right)\right)\left(E_{\mathbf{cc}}\right)_i\right] + \\
&\quad + s^k Q_{\mathbf{cc}} + \rho Q_{\mathbf{c}}^T D_s^k Q_{\mathbf{c}} \\
\nabla^2_{\mathbf{cu}}\mathcal{L}_\rho\left(\mathbf{c}^k,\mathbf{u}^k\right) &= \nabla^2_{\mathbf{cu}}\mathcal{J}_h\left(\mathbf{c}^k,\mathbf{u}^k\right) + \rho E_{\mathbf{c}}^T E_{\mathbf{u}} + \sum_{i=1}^{n}\left[\left(\lambda^k + \rho\mathcal{E}_h\left(\mathbf{c}^k,\mathbf{u}^k\right)\right)\left(E_{\mathbf{cu}}\right)_i\right] + \\
&\quad + s^k Q_{\mathbf{cu}} + \rho Q_{\mathbf{c}}^T D_s^k Q_{\mathbf{u}} \\
\nabla^2_{\mathbf{uc}}\mathcal{L}_\rho\left(\mathbf{c}^k,\mathbf{u}^k\right) &= \nabla^2_{\mathbf{uc}}\mathcal{J}_h\left(\mathbf{c}^k,\mathbf{u}^k\right) + \rho E_{\mathbf{u}}^T E_{\mathbf{c}} + \sum_{i=1}^{n}\left[\left(\lambda^k + \rho\mathcal{E}_h\left(\mathbf{c}^k,\mathbf{u}^k\right)\right)\left(E_{\mathbf{uc}}\right)_i\right] + \\
&\quad + s^k Q_{\mathbf{uc}} + \rho Q_{\mathbf{u}}^T D_s^k Q_{\mathbf{c}} \\
\nabla^2_{\mathbf{uu}}\mathcal{L}_\rho\left(\mathbf{c}^k,\mathbf{u}^k\right) &= \nabla^2_{\mathbf{uu}}\mathcal{J}_h\left(\mathbf{c}^k,\mathbf{u}^k\right) + \rho E_{\mathbf{u}}^T E_{\mathbf{u}} + \sum_{i=1}^{n}\left[\left(\lambda^k + \rho\mathcal{E}_h\left(\mathbf{c}^k,\mathbf{u}^k\right)\right)\left(E_{\mathbf{uu}}\right)_i\right] + \\
&\quad + s^k Q_{\mathbf{uu}} + \rho Q_{\mathbf{u}}^T D_s^k Q_{\mathbf{u}}
\end{aligned}
\tag{5-13b}
$$

### 5-2-4 Attempted Decomposition of the Derived Linear System

After the derivation of the formulas that produce the gradient and the Hessian ((5-12),(5-13)) of the augmented Lagrangian equation (5-8), the next step is to solve the linear system in Equation 5-11 that computes the Newton-step correction of the state and the control variables. This solution would normally require the inversion of a large and sparse matrix and would have to be repeated at every inner iteration since both the Hessian and the gradient could be dependent on the current $c_k$ or $u_k$. Following the typical *discretise-then-optimise* method of a PDE-constrained optimisation problem, this step would be considered the computational bottleneck of the proposed methodology. To that end, at the next stage, DDM techniques are utilised in an attempt to decompose the problem and speed up the solution of the derived linear system.

From the analysis of the structure of the Hessian matrix, several interesting features have emerged. First of all, as the nature of the described problem dictates, the state variable is used to describe a characteristic of the simulated system and, as a result, $\mathbf{c}$ will have to be defined in the whole domain. On the other hand, the control variable $\mathbf{u}$ represents an action designed to influence the state of the system. This action, mathematically is usually modelled as distributed and is applied in the whole domain. In real world scenarios however, it is spatially localised. In the described example of air-pollution, the control will be limited from the location and the surface of the industrial chimneys and can either increase or decrease their output. Therefore, it is safe to assume that the value of $\mathbf{u}$ will be a priori equal to zero over most parts of the domain. The crucial deduction from these observations is that the number of DoFs associated with the state $\mathbf{c}$ is immensely larger compared to the the control $\mathbf{u}$. Consequently, the term that predominantly contributes to the Hessian is the upper-left block, which is equal to $\nabla^2_{\mathbf{cc}}\mathcal{L}_\rho\left(\mathbf{c}^k,\mathbf{u}^k\right)$. If this term can be decomposed and reordered into interior and interface variables, then by adding the fewer control inputs as "artificial"

interface variables, it would be possible to decompose the complete Hessian. For this reason, the attention of the subsequent analysis was directed towards the aforementioned matrix and its structure.

From all the terms of $\nabla^2_{\mathbf{cc}}\mathcal{L}_\rho$, the product of $\rho E_{\mathbf{c}}^T E_{\mathbf{c}}$ is the one that is standing out. The reason for this is that the term $E_{\mathbf{c}}$ is also used for the solution of the underlying PDE that is described via the state equation. More specifically, as it is mentioned beforehand, $E_{\mathbf{c}}$ denotes the derivative of the PDE residual in the discretised formulation. Based on their definition and the discretisation with FEM, it can be proven that all the other terms of $\nabla^2_{\mathbf{cc}}\mathcal{L}_\rho$ have a beneficial structure similar to what was seen in Figure 4-2. This means that they can be successfully manipulated by a DDM and its inverse matrix can be assembled via the local contributions of the decomposed domain. The same holds true for the $E_{\mathbf{c}}$ term as it is directly associated with the PDE solution. However, the term that appears in the $\nabla^2_{\mathbf{cc}}\mathcal{L}_\rho$ is not the $E_{\mathbf{c}}$ but the product $\rho E_{\mathbf{c}}^T E_{\mathbf{c}}$. Unfortunately, there is no guarantee that this term will retain the same attributes as the $E_{\mathbf{c}}$. As a result, directly applying a non-overlapping substructuring DDM to the derived Hessian will, most likely, fail to produce the block diagonal form of the interior variables seen in Figure 4-5. This structure is required for the parallelisation of the whole procedure.

### 5-2-5 Proposed State Variable Transformation

To overcome this issue and recover a structure with promising characteristics, the proposed methodology introduces a transformation of the state variables on the equation that computes the Newton-step update (5-11). More specifically, instead of solving the equation with respect to $\mathbf{x} = \begin{bmatrix}\Delta\mathbf{c}^\top & \Delta\mathbf{u}^\top\end{bmatrix}^T$ a new set of variables $\mathbf{x}^*$ will be defined as the vector

$$\mathbf{x}^* = \begin{bmatrix} E_{\mathbf{c}}\Delta\mathbf{c} \\ \Delta\mathbf{u} \end{bmatrix} = \underbrace{\begin{bmatrix} E_{\mathbf{c}} & 0 \\ 0 & I \end{bmatrix}}_{\mathbf{P}} \begin{bmatrix} \Delta\mathbf{c} \\ \Delta\mathbf{u} \end{bmatrix} = P\mathbf{x}. \tag{5-14}$$

By executing this reformulation, the Hessian will have to be altered too, in order to accommodate the change in the variables and satisfy the original equation.

$$\begin{aligned} H_k\left(\mathbf{c}^k, \mathbf{u}^k\right)\mathbf{x} &= -\,g_k\left(\mathbf{c}^k, \mathbf{u}^k\right) \implies \\ \underbrace{H_k\left(\mathbf{c}^k, \mathbf{u}^k\right)P^{-1}}_{H_k^*}\underbrace{P\mathbf{x}}_{\mathbf{x}^*} &= -\,g_k\left(\mathbf{c}^k, \mathbf{u}^k\right) \implies \\ H_k^*\left(\mathbf{c}^k, \mathbf{u}^k\right)\mathbf{x}^* &= -\,g_k\left(\mathbf{c}^k, \mathbf{u}^k\right) \end{aligned} \tag{5-15}$$

So, the value of the $H_k^*$ is computed as follows:

$$H_k^*\left(\mathbf{c}^k, \mathbf{u}^k\right) := \begin{bmatrix} \nabla^2_{\mathbf{cc}}\mathcal{L}_\rho\left(\mathbf{c}^k, \mathbf{u}^k\right)E_{\mathbf{c}}^{-1} & \nabla^2_{\mathbf{cu}}\mathcal{L}_\rho\left(\mathbf{c}^k, \mathbf{u}^k\right) \\ \nabla^2_{\mathbf{uc}}\mathcal{L}_\rho\left(\mathbf{c}^k, \mathbf{u}^k\right)E_{\mathbf{c}}^{-1} & \nabla^2_{\mathbf{uu}}\mathcal{L}_\rho\left(\mathbf{c}^k, \mathbf{u}^k\right) \end{bmatrix}, \tag{5-16}$$

and a new Newton-step equation arises.

The main advantage of this method lies in the fact, that it changes the structure of $\nabla^2_{\mathbf{cc}}\mathcal{L}_\rho$ from the original Hessian into a more favourable one. This is achieved through the presence of the $E_\mathbf{c}$ term and the "amplification" of the its beneficial structure. More specifically, by denoting as

$$\begin{aligned}\mathbf{M_{cc}}\left(\mathbf{c}^k,\mathbf{u}^k\right) = \nabla^2_{\mathbf{cc}}\mathcal{J}_h\left(\mathbf{c}^k,\mathbf{u}^k\right) + \sum_{i=1}^{n}\left[\left(\lambda^k + \rho\mathcal{E}_h\left(\mathbf{c}^k,\mathbf{u}^k\right)\right)(E_{\mathbf{cc}})_i\right] + \\ + s^k Q_{\mathbf{cc}} + \rho Q_\mathbf{c}^T D_s^k Q_\mathbf{c}\end{aligned} \tag{5-17}$$

which gathers all the terms of $\nabla^2_{\mathbf{cc}}\mathcal{L}_\rho(\mathbf{c},\mathbf{u})$ from (5-13b), besides $\rho E_\mathbf{c}^T E_\mathbf{c}$, it can be seen that the upper left block of $H_k^*$ will be equal to:

$$\begin{aligned}\nabla^2_{\mathbf{cc}}\mathcal{L}_\rho\left(\mathbf{c}^k,\mathbf{u}^k\right)E_\mathbf{c}^{-1} = \mathbf{M_{cc}}\left(\mathbf{c}^k,\mathbf{u}^k\right)E_\mathbf{c}^{-1} + \rho E_\mathbf{c}^T E_\mathbf{c} E_\mathbf{c}^{-1} \implies \\ \nabla^2_{\mathbf{cc}}\mathcal{L}_\rho\left(\mathbf{c}^k,\mathbf{u}^k\right)E_\mathbf{c}^{-1} = \mathbf{M_{cc}}\left(\mathbf{c}^k,\mathbf{u}^k\right)E_\mathbf{c}^{-1} + \rho E_\mathbf{c}^T.\end{aligned} \tag{5-18}$$

From Equation 5-18, we can see that the $E_\mathbf{c}^T$ term is present into $\nabla^2_{\mathbf{cc}}\mathcal{L}_\rho E_\mathbf{c}^{-1}$ but the $E_\mathbf{c}$ is missing. However, the aforementioned beneficial structure of $E_\mathbf{c}$ is symmetric (Figure 4-2a, Figure 5-2a) and as a result the transpose $E_\mathbf{c}^T$ shares this beneficial structure. The improvement of the inner structure of $H_k^*$ comes at the expense of an additional computational cost at every iteration. From now on, at every iteration for which $\mathbf{x}^*$ is computed, in order to retrieve the isolated values of $\Delta\mathbf{c}$ the large linear system

$$E_\mathbf{c}\left(\mathbf{c}^k,\mathbf{u}^k\right)\Delta\mathbf{c} = \mathbf{x}_\mathbf{c}^* \quad \text{with} \quad \mathbf{x}^* = \begin{bmatrix}\mathbf{x}_\mathbf{c}^* \\ \Delta\mathbf{u}\end{bmatrix}, \tag{5-19}$$

will have to be solved. The $\mathbf{x}_\mathbf{c}^*$ term is derived from the parts of the solution of $\mathbf{x}^*$ that are associated with the state variable $\mathbf{c}$. However, the specific type of the linear system in (5-19) is basically the same as the solution of the $\mathcal{E}(c,u)$ PDE. As it is mentioned many times now, the structure of $E_\mathbf{c}$ is compatible with the utilisation of the substructuring decomposition method and thus, the computation of the $\Delta\mathbf{c}$ at each iteration can occur through parallel processes, which reduces the overall computational costs of changing from $\mathbf{x}$ to $\mathbf{x}^*$.

### 5-2-6 Proposed Reordering into Interior and Interface Variables

So far, the proposed methodology has ensured the presence of $E_\mathbf{c}^T$ within $\nabla^2_{\mathbf{cc}}\mathcal{L}_\rho E_\mathbf{c}^{-1}$. Nevertheless, this alone is not enough for the beneficial structure properties of $E_\mathbf{c}$ to be transferred into the upper left block of $H_k^*$. In order to accomplish this goal, the aforementioned property of $E_\mathbf{c}$ will have to be "amplified". This is achieved via the nullification of the effects from all other terms of $\nabla^2_{\mathbf{cc}}\mathcal{L}_\rho E_\mathbf{c}^{-1}$. As seen in Equation 5-18, the terms that are included in $\mathbf{M_{cc}}E_\mathbf{c}^{-1}$ not only do not share the desired structure (Figure 5-2b), but also require the computation of the inverse of $E_\mathbf{c}$ in order to compute them.

To eliminate their effects two possible interventions have been identified. The first one is the reordering of the corresponding DoFs so as to transport the counterproductive structure out of the upper left block of the Hessian. From the view of the substructuring domain decomposition methods this is equivalent to the transition of the corresponding DoFs from interior variables into "artificial" interface variables. By doing so, they are no longer considered

as a part of the upper left block of $H_k^*$ and their removal, although it does change $E_{\mathbf{c}}^T$ it does not affect its beneficial structure (Figure 5-3).

In order to provide some further insight on the steps of the algorithm, we create a schematic representation of the system reformulation. Assuming a physical domain $\Omega$ similar to Figure 5-1 that is split into N subdomains ($\Omega_i$) and it includes areas of interest that are associated with the cost functional, inequality constraints or non-linear phenomena (red areas).

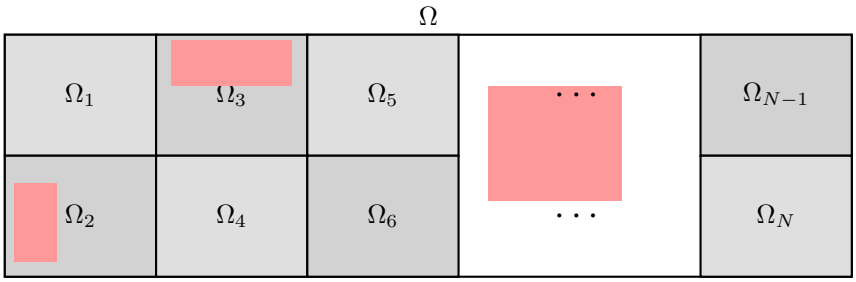
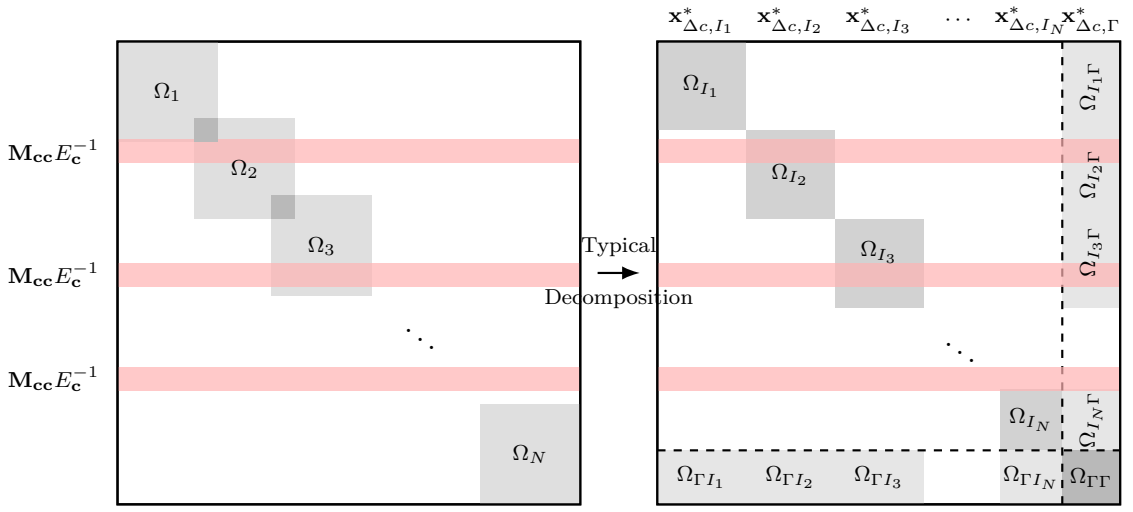



**Figure 5-1:** Decomposed physical domain, with spatially localised areas of interest.

Then, based on the mesh from the finite elements method and the proposed transformation of the state variables, the upper left block if the transformed Hessian (5-16) will have a structure similar to the one seen in the Figure 5-2(a). In this figure, the gray areas denotes as $\Omega_i$ represent the values of the $E_c^\top$ matrix that are associated with the $\Omega_i$ subdomain. The red stripes denote the values of the $\mathbf{M_{cc}}E_{\mathbf{c}}^{-1}$ terms. It is evident that the advantageous structure of $E_c^\top$ is perturbated by the values of the $\mathbf{M_{cc}}E_{\mathbf{c}}^{-1}$ terms . Independently of the individual structure of $\mathbf{M_{cc}}$, its multiplication with $E_{\mathbf{c}}^{-1}$ will spread its values all over the domain. On the other hand, the values of $E_c^\top$ are localised in the regions that correspond to the subdomains $\Omega_i$.

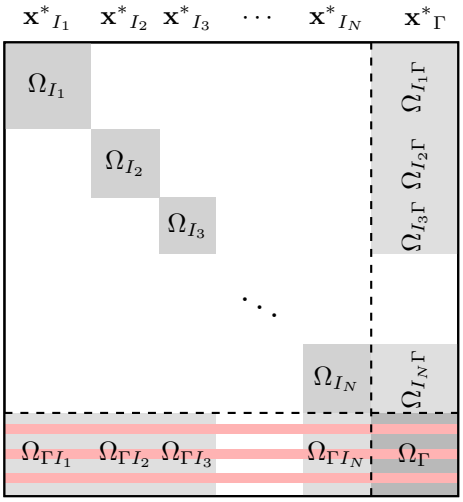



**(a)** Original structure of $\nabla^2_{\mathbf{cc}}\mathcal{L}_\rho E_{\mathbf{c}}^{-1}$ **(b)** Physical decomposition of $\nabla^2_{\mathbf{cc}}\mathcal{L}_\rho E_{\mathbf{c}}^{-1}$

**Figure 5-2:** The structure of the upper-left block matrix of $\mathbf{H}_k^*$.

If a typical substructuring decomposition algorithm, based on the physical interfaces of the domain, is performed then the interior matrix, denoted via the dashed line (Figure 5-2**b**), will not be block diagonal. As a result its computation cannot be parallelised like the case of Figure 4-5.

To overcome this issue, as it is mentioned beforehand, the degrees of freedom that correspond to the red stripes can be considered as an "artificial" interface variable and thus be transported outside the interior matrix. The procedure is similar to the one described in Section 4-3 and the result can be seen in Figure 5-3. There some of the interior matrices $\Omega_{I_i}$ are smaller that before since DoFs were removed from them but now they are block diagonal.




Proposed decomposition including the "artificial" interface variables

**Figure 5-3:** Proposed Decomposition of $\nabla^2_{\mathbf{cc}}\mathcal{L}_\rho E_{\mathbf{c}}^{-1}$

The disadvantages of this technique lies with the increase of the interface variables $\mathbf{x}^*{}_\Gamma$. In the described DDM framework (Section 4-3), the interior variables (in this case $\mathbf{x}^*{}_I$) are eliminated and the large and sparse linear system is transformed into a smaller but denser one, with size equal to the number of the interface variables. When this number increases, not only the effect of parallelisation is reduced but also the matrix derived from the Schur complement may have a comparable computational cost with the original one.

As a result, this technique is only valid for the terms of $\mathbf{M_{cc}}$ which affect the $H_k^*$ locally as seen in the schematic representation. This means that, the majority of $\mathbf{M_{cc}}E_{\mathbf{c}}^{-1}$ will have to be a priori equal to zero with the exception of the rows associated with a preselected set of nodes of the mesh. This preselected set denotes the corresponding DoFs that have to be removed to "amplify" the structure of $E_{\mathbf{c}}^T$. Terms with this property can be the Hessian of the cost functional with respect to $\mathbf{c}$ ($\nabla^2_{\mathbf{cc}}\mathcal{J}_h$), assuming that its area of interest is a subregion of the whole domain, the inequality terms ($s^k Q_{\mathbf{cc}} + \rho Q_{\mathbf{c}}^T D_s^k Q_{\mathbf{c}}$) associated with the state variable at a specific region and the contribution of the second-order derivative of the state equation $\left[\sum_{i=1}^n \left\{\left(\lambda^k + \rho\mathcal{E}_h\left(\mathbf{c}^k, \mathbf{u}^k\right)\right)(E_{\mathbf{cc}})_i\right\}\right]$ when its effect is spatially localised.

However, the identification of these terms is not sufficient for the proposed methodology. Prior to the reordering of the associated DoFs, the products

$$\nabla^2_{\mathbf{cc}}\mathcal{J}_h E_{\mathbf{c}}^{-1}, \qquad s^k Q_{\mathbf{cc}} E_{\mathbf{c}}^{-1}, \quad \rho Q_{\mathbf{c}}^T D_s^k Q_{\mathbf{c}} E_{\mathbf{c}}^{-1} \quad \text{and} \quad \sum_{i=1}^{n} \left\{ \left( \lambda^k + \rho \mathcal{E}_h \left( \mathbf{c}^k, \mathbf{u}^k \right) \right) (E_{\mathbf{cc}})_i \right\} E_{\mathbf{c}}^{-1} \tag{5-20}$$

will have to be computed. Normally, that would be a computationally expensive process but since $E_{\mathbf{c}}$ is the matrix associated with the PDE solution and all terms of $\mathbf{M_{cc}}$ are considered to be mostly empty, there are ways to design a procedure that will speed up this computation. It is safe to assume that $\mathbf{M}^*_{\mathbf{cc}}$ will retain the property of its components and it will be composed from a relative small number (compared to the total number of DoFs) of non-zero rows. Then each of these rows can be isolated and with $\mathbf{Z_{cc}}$ the solution of the product $\mathbf{M_{cc}} E_{\mathbf{c}}^{-1}$ the equations, that arise for each row, are:

$$[\mathbf{M_{cc}}]_i E_{\mathbf{c}}^{-1} = [\mathbf{Z_{cc}}]_i \implies [\mathbf{M_{cc}}]_i = [\mathbf{Z_{cc}}]_i E_{\mathbf{c}} \implies [\mathbf{M_{cc}}]_i^T = E_{\mathbf{c}}^T [\mathbf{Z_{cc}}]_i^T . \tag{5-21}$$

So, for each of these rows, a linear system is formulated with a structure that is favourable for substructuring domain decomposition methods due to $E_c^\top$. As a result, the small number of non-zero rows of $\mathbf{Z_{cc}}$ can be computed fast and in parallel.

The last step, before the implementation of the proposed domain decomposition method is the computation of the $\nabla^2_{\mathbf{uc}}\mathcal{L}_\rho E_{\mathbf{c}}^{-1}$ term. This is a by-product of the selection of the new set of variables $\mathbf{x}^*$. Based on Equation 5-13b the aforementioned term will be equal to

$$\begin{aligned} \nabla^2_{\mathbf{uc}}\mathcal{L}_\rho \left( \mathbf{c}^k, \mathbf{u}^k \right) E_{\mathbf{c}}^{-1} &= \nabla^2_{\mathbf{uc}}\mathcal{J}_h \left( \mathbf{c}^k, \mathbf{u}^k \right) E_{\mathbf{c}}^{-1} + \rho E_{\mathbf{u}}^T E_{\mathbf{c}} E_{\mathbf{c}}^{-1} + \\ + \sum_{i=1}^{n} \left[ \left( \lambda^k + \rho \mathcal{E}_h \left( \mathbf{c}^k, \mathbf{u}^k \right) \right) (E_{\mathbf{uc}})_i \right] E_{\mathbf{c}}^{-1} + s^k Q_{\mathbf{uc}} E_{\mathbf{c}}^{-1} + \rho Q_{\mathbf{u}}^T D_s^k Q_{\mathbf{c}} E_{\mathbf{c}}^{-1} &\implies \\ \nabla^2_{\mathbf{uc}}\mathcal{L}_\rho \left( \mathbf{c}^k, \mathbf{u}^k \right) E_{\mathbf{c}}^{-1} = \left( \nabla^2_{\mathbf{uc}}\mathcal{J}_h \left( \mathbf{c}^k, \mathbf{u}^k \right) + s^k Q_{\mathbf{uc}} + \rho Q_{\mathbf{u}}^T D_s^k Q_{\mathbf{c}} \right) E_{\mathbf{c}}^{-1} + \\ + \sum_{i=1}^{n} \left[ \left( \lambda^k + \rho \mathcal{E}_h \left( \mathbf{c}^k, \mathbf{u}^k \right) \right) (E_{\mathbf{uc}})_i \right] E_{\mathbf{c}}^{-1} + \rho E_{\mathbf{u}}^T . \end{aligned} \tag{5-22}$$

The last term of the (5-22) is actually simplified from the multiplication with $E_{\mathbf{c}}^{-1}$ and all the others will have to be computed based on the techniques previously described this section (5-21).

### Alternative Approach

The second intervention that could eliminate the effect of $\mathbf{M_{cc}}$ is the removal of the terms that interfere with the beneficial structure of $E_{\mathbf{c}}^T$. For example, eliminating

$$\sum_{i=1}^{n} \left[ \left( \lambda^k + \rho \mathcal{E}_h \left( \mathbf{c}^k, \mathbf{u}^k \right) \right) (E_{\mathbf{cc}})_i \right] \tag{5-23}$$

could be a result of simplification in the formulation of the PDE-constrained optimisation problem or a simplification of the optimality conditions (quasi-Newton step methods).

### 5-2-7 Formulation of the Decomposed Linear System

By following the presented steps of the proposed methodology until this stage, all the blocks of the $H_k^*$ will have been computed. The next phase of the solution requires the reordering of the variables from Equation 5-15. This reordering will occur in order to enforce the advantageous structure of $E_{\mathbf{c}}$ into the upper left block of the $H_k^*$. This is necessary since, based on Section 4-3, in order to apply the substructuring domain decomposition methods in Equation 5-15 the matrix $H_k^*$ must have the form

$$H_k^*\left(\mathbf{c}^k,\mathbf{u}^k\right)=\begin{bmatrix} H_{II}^*\left(\mathbf{c}^k,\mathbf{u}^k\right) & H_{I\Gamma}^*\left(\mathbf{c}^k,\mathbf{u}^k\right)\\ H_{\Gamma I}^*\left(\mathbf{c}^k,\mathbf{u}^k\right) & H_{\Gamma\Gamma}^*\left(\mathbf{c}^k,\mathbf{u}^k\right)\end{bmatrix} \tag{5-24}$$

and $H_{II}^*$ must be block diagonal. In the solution of a stand-alone PDE the reordering of the variables is based on the selected non-overlapping decomposition of the physical domain. Standard substructuring methods split the state variables $\mathbf{c}$ into the interface variables $\mathbf{c_\Gamma}$ that are associated with the sub-domains interfaces and the interior variables $\mathbf{c_I}$ that exist inside the sub-domains. If this action is repeated here, based on (4-5), it will ensure that $E_{\mathbf{c}}$ is block diagonal. Then the same can be said for $E_{\mathbf{c}}^T$. However, the additional terms of $H_{II}^*$ break this property in the general case as seen in Figure 5-2(b). To resolve this issue, as it was suggested in the previous paragraph, some additional variables that typically would belong in the interior group will have to be considered as "artificial" interface variables.

As a result, in the proposed optimisation framework, the definition of the decomposition of the problem is not limited to the decomposition of the physical domain and the corresponding geometrical interface. Depending on the nature of the optimal problem, interior and interface variables may adapt to accommodate the convergence of the substructuring DDM. So, in the general case, the interface variables $\mathbf{x}_{\mathbf{\Gamma}}^*$ will include $\mathbf{\Delta u}$, the $\mathbf{\Delta c}$ at the nodes on the geometrical interface of the domain as well as some originally interior variables $\mathbf{\Delta c}$ that are associated with localised objective functions and constraints. Finally, the form of the linear system that leads to the computation of the Newton step update will be:

$$\begin{bmatrix} H_{II}^*\left(\mathbf{c}^k,\mathbf{u}^k\right) & H_{I\Gamma}^*\left(\mathbf{c}^k,\mathbf{u}^k\right)\\ H_{\Gamma I}^*\left(\mathbf{c}^k,\mathbf{u}^k\right) & H_{\Gamma\Gamma}^*\left(\mathbf{c}^k,\mathbf{u}^k\right)\end{bmatrix}\begin{bmatrix}\mathbf{x}_{\mathbf{I}}^*\\ \mathbf{x}_{\mathbf{\Gamma}}^*\end{bmatrix}=-\begin{bmatrix}\mathbf{g_k}\left(\mathbf{c^k},\mathbf{u^k}\right)_{\mathbf{I}}\\ \mathbf{g_k}\left(\mathbf{c^k},\mathbf{u^k}\right)_{\mathbf{\Gamma}}\end{bmatrix}. \tag{5-25}$$

A schematic representation of the full $H_k^*$ matrix can be seen in Figure 5-4. According to our assumption about the size of the $H_k^*$, it must be mostly identical to its upper left term that is seen in Figure 5-3. Indeed, when the other terms $\nabla^2_{\mathbf{uc}}\mathcal{L}_\rho E_{\mathbf{c}}^{-1}$,$\nabla^2_{\mathbf{cu}}\mathcal{L}_\rho$,$\nabla^2_{\mathbf{uu}}\mathcal{L}_\rho$ are included (blue parts) the general structure of the matrix does not change. In fact Equation 5-25 has the exact same structure described in Equation 4-11 and thus it can be partitioned and solved based on the described substructuring domain decomposition algorithm Section 4-3.

### 5-2-8 Solution of the Decomposed Linear System

The derived Schur complement equation will be

$$\boxed{S_\Gamma \mathbf{x}_{\mathbf{\Gamma}}^* = \mathbf{b}_\Gamma} \tag{5-26a}$$

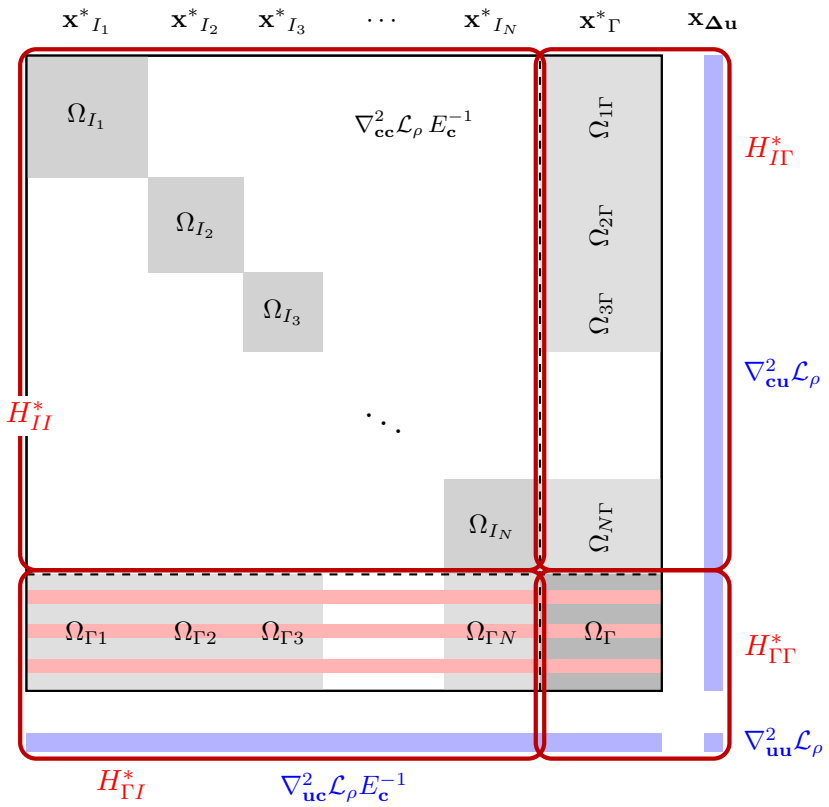



**Figure 5-4:** Formulation of the reordered $H^*_k$ Hessian matrix.

with

$$S_\Gamma = H^*_{\Gamma\Gamma} - H^*_{\Gamma I}\left(H^*_{II}\right)^{-1} H^*_{I\Gamma}, \quad \mathbf{b}_\Gamma = -\mathbf{g}_\Gamma + H^*_{\Gamma I}\left(H^*_{II}\right)^{-1}\mathbf{g}_I. \tag{5-26b}$$

The formulation of the Schur-complement equation will be accelerated up via the parallel computation of $(H^*_{II})^{-1}$ but the derived linear system will have to be solved via a standard numerical solver. After the computation of $\mathbf{x}^*_{\mathbf{\Gamma}}$, the interior variables $\mathbf{x}^*_{\mathbf{I}}$ can be computed in parallel via the formula

$$\mathbf{x}^*_{\mathbf{I}} = \left(H^*_{II}\right)^{-1}\left(-\mathbf{g}_I - H_{I\Gamma}\mathbf{x}^*_{\mathbf{\Gamma}}\right) \tag{5-26c}$$

### 5-2-9 Derivation of the Original State Variable and Computation of Newton-Step

Following the computation of $\mathbf{x}^*$ and its reformulation into $\mathbf{x}$ based on (5-14), the update step of the Newton optimisation method $\mathbf{x} = \begin{bmatrix}\Delta\mathbf{c}^\top & \Delta\mathbf{u}^\top\end{bmatrix}^T$ will arise. The state and control variables will be updated based on the formula:

$$\begin{bmatrix}\mathbf{c}^{k+1}\\ \mathbf{u}^{k+1}\end{bmatrix} = \begin{bmatrix}\mathbf{c}^{k}\\ \mathbf{u}^{k}\end{bmatrix} + \begin{bmatrix}\Delta\mathbf{c}\\ \Delta\mathbf{u}\end{bmatrix}. \tag{5-27}$$

### 5-2-10 Convergence Criteria of the Proposed Methodology

This inner iteration will be repeated until one or more of the selected convergence criteria is met, and $\arg\min_{\mathbf{c},\mathbf{u}} \mathcal{L}_\rho$ converges to an optimal value. These criteria can include lower bounds on the norm of the augmented Lagrangian's gradient or on the size of the Newton correction step as well as an upper bound on the total number of iterations. Then, the outer update of the Lagrange multipliers and of the penalty parameter will be performed. The iterations of the outer optimisation problem, as seen in Equation 5-10, will continue until the optimality conditions of the augmented Lagrangian are satisfied. These conditions can incorporate the acceptable limits of the equality and inequality constraints as well as the satisfaction of the stationarity condition.

### 5-2-11 Summary of the Proposed Methodology

To sum up, the complete proposed methodology is comprised by a series of consecutive steps. At first, the general optimisation problem is formulated with a PDE as the state equation, a cost functional and any additional equality or inequality constraints. Once the state and control variables are defined, the whole problem is discretised via an appropriate finite elements methodology and it is turned into a finite dimensional constrained optimisation problem. In order to reformulate it into an unconstrained optimisation problem and enforce the equalities and inequalities into the cost functional, the augmented Lagrangian methodology with quadratic and exponential penalties is implemented. For the minimisation of the current augmented Lagrangian function with respect to the state and control variables (inner optimisation), Newton iterations are selected. However, in order to efficiently solve the derived linear system, a transformation followed by a reordering of the selected variables is performed. By doing so, the transformed Hessian has a structure that is favourable to the implementation of substructuring domain decomposition methods. This allows the proposed methodology to solve the derived sparse and large linear system in parallel and reduce its computational costs. The update of the decision variables via the Newton step is repeated until the solution converges. Then the Lagrangian multipliers are revised and this whole process is repeated until the preselected optimisation criteria are met. Domain decomposition methodologies are utilised not only in the solution of the Newton-step linear system but also in the formulation of the Hessian as well as the transformation between the $\mathbf{x}^*$ and $\mathbf{x}$.

## Synopsis of Chapter 5

This chapter conducted a step by step analysis on the proposed methodology that incorporated the substructuring domain decomposition method into a PDE-constrained optimisation framework. Tackling the problem with a different perspective from already established methods, led to the formulation of an algorithm that could be used in a wide variety of problems associated but not limited to air pollution control. Contrary to established approaches, this method is capable of handling any type of constraints under mild assumptions. Emphasis was given to the various positive and disadvantageous characteristics as they emerged from the formulation of the algorithmic steps. In conclusion, since the proposed method is based

on the parallelisation of specific processes of the optimisation, it could potentially exhibit improved computational efficiency and scalability.

# Chapter 6

# Numerical Implementation and Performance Analysis

This chapter evaluates the proposed incorporation of domain decomposition methods in a PDE-constrained optimisation framework. In Section 6-1, all the aforementioned information is combined for the formulation of a numerical example that can be used as a benchmark test. It follows, step by step the algorithm that was derived in the previous chapter for a case of air-pollution control. Then Section 6-2 presents the optimum state that was reached after the convergence of the optimisation procedure. This state is then compared to similar examples with more degrees of freedom and with an unoptimised case. Finally, Section 6-3 offers insight on the numerical efficiency of the proposed methodology in comparison to the direct computation of the Newton step.

## 6-1 Implementation of the Proposed Methodology

Following the introduction of the proposed optimisation framework and the analysis of each one of its steps, a benchmark test is required in order to assess its performance and numerical efficiency. For that reason, a particular case of PDE-contained optimisation problem associated with the mitigation of air-pollution will be considered, as described in Chapter 2. The purpose of these tests will be to examine the computational costs of the proposed algorithm, to identify any limitations with respect to its implementation and to gain insight on how to improve the methodology.

### 6-1-1 Formulation of the Optimisation Problem

We decide that the objective of the presented optimisation problem will be to attempt keeping the concentration of sulphur dioxide ($SO_2$) below a curtain threshold level $c_t$, at a specific area of interest $\Omega_t$ of the domain. That area of interest could signify a town, an agricultural area, a historical site etc. The emission sources of the problem will be three industrial chimneys

placed at different areas of the domain $\Omega_{u_i}$ with a maximum output equal to $u_{max}$. They cannot be shut down completely (lower limit $u_{min}$) and their cumulative output should be above the threshold denoted as $u_{MinC}$. In addition, the chimneys stand at a distance from the ground equal to $H$. The speed of the wind is set as $V$ whereas as it direction $\theta$ is considered constant throughout the domain. The diffusion coefficient will be set to be isotropic and it will be computed as a function of the distance from the emission sources $r$, the air velocity $\mathbf{V}$ and the two-dimensional standard deviation $\sigma_{xy}$. From the theoretical analysis, this function should be equal to (2-7) but it can also be simplified to the formula:

$$\nu \coloneqq \frac{\sigma_{xy}^2}{2 \cdot r} \cdot \|\mathbf{V}\| \tag{6-1}$$

The model that will be used for the transport of the $SO_2$ in the atmosphere is the the steady state of advection-diffusion PDE as it was described in Section 2-4.

Then finally, the optimisation problem will take the following form:

$$\begin{aligned}
\min_{c,u} \quad & \mathcal{J}(c,u) = \frac{1}{2}\int_{\Omega_t} (c(\mathbf{x}) - c_t(\mathbf{x}))^2 \, d\Omega + \frac{\alpha}{2}\int_{\Omega_u} u(\mathbf{x})^2 \, d\Omega \\
\text{subject to} \quad & -\nu\nabla^2 c(\mathbf{x}) + \mathbf{V}\cdot\nabla c(\mathbf{x}) = \sum_{i=1}^{3} \chi_{\Omega_{u_i}}(\mathbf{x})\, u_i && \text{in } \Omega, \\
& c(\mathbf{x}) = 0 && \text{on } \Gamma_{\text{in}}, \\
& \nu\frac{\partial c}{\partial n} = 0 && \text{on } \Gamma_{\text{out}}, \\
& u_{\min} \le u_i \le u_{\max} && \forall \;\; i = 1,2,3 \\
& u_{MinC} \le u_1 + u_2 + u_3
\end{aligned} \tag{6-2}$$

The variables $\mathbf{x} = (x, y)^\top$ represent the spatial coordinates in the 2-D plane while the function $c(\mathbf{x})$ describes the concentration of the air-pollutant. The additional quadratic penalty term of the emission rate $u(\mathbf{x})$ in the cost functional is present to prevent the control from receiving too large values and to improve the conditioning of the derived linear system. Furthermore, the $\chi_{\Omega_{u_i}}$ denotes the characteristic function of the control regions $\Omega_{u_i}$:

$$\chi_{\Omega_{u_i}}(\mathbf{x}) = \begin{cases} 1 & \mathbf{x} \in \Omega_{u_i} \\ 0 & \mathbf{x} \notin \Omega_{u_i}. \end{cases}$$

### 6-1-2 Discretisation of the Optimisation Problem

The next step of the algorithm is the discretisation of the optimisation problem via the finite elements methodology. Based on the description in (5-6), the weak formulation of the PDE in (6-3) and the cost functional expressed in (6-2), the original optimisation problem can be reformulated as a finite-dimensional optimisation problem. For the simulation of the state variables, $P_2$ elements were utilised whereas for the control inputs $P_0$ elements are sufficient. The weak formulation of the state equation will be derived by multiplying both parts of the PDE with a test function $v$ with the same boundary conditions and properties as the state $c$:

$$\int_\Omega \nu\nabla c \cdot \nabla v \, d\Omega + \int_\Omega (\mathbf{V}\cdot\nabla c) v \, d\Omega = \sum_{i=1}^{3} u_i \int_{\Omega_i} v \, d\Omega. \tag{6-3}$$

The diffusion term will result in the diffusion matrix $K \in \mathbb{R}^{N_c \times N_c}$

$$\int_\Omega \nu \nabla c_h(\mathbf{x}) \cdot \nabla \phi_i \, d\Omega = \sum_{j=1}^{N_c} c_j \int_\Omega \nu \nabla \phi_j \cdot \nabla \phi_i \, d\Omega \implies K_{ij} = \int_\Omega \nu \nabla \phi_j \cdot \nabla \phi_i \, d\Omega \tag{6-4}$$

Similarly, the advection term will generate the advection matrix $C \in \mathbb{R}^{N_c \times N_c}$

$$\int_\Omega (\mathbf{V} \cdot \nabla c_h(\mathbf{x})) \phi_i \, d\Omega = \sum_{j=1}^{N_c} c_j \int_\Omega (\mathbf{V} \cdot \nabla \phi_j) \phi_i \, d\Omega \implies C_{ij} = \int_\Omega (\mathbf{V} \cdot \nabla \phi_j) \phi_i \, d\Omega. \tag{6-5}$$

Since both the diffusion and advection matrices act on the same state vector $\mathbf{c}$, they should be combined into a single matrix A, to help with the notation.

$$A := K + C \tag{6-6}$$

For the source term leads to the formulation of the control matrix $\mathbf{B} \in \mathbb{R}^{N_c \times 3}$

$$\sum_{j=1}^{3} u_j \int_\Omega \chi_{\Omega_{u_j}}(\mathbf{x}) \psi_j(\mathbf{x}) \, d\Omega = \sum_{j=1}^{3} u_j \int_{\Omega_{u_j}} \psi_i(\mathbf{x}) \, d\Omega \implies B_{ij} = \int_{\Omega_{u_j}} \psi_i(\mathbf{x}) \, d\Omega. \tag{6-7}$$

So, the discretised state equation is

$$\boxed{\mathcal{E}_h(\mathbf{c}, \mathbf{u}) := A\mathbf{c} - B\mathbf{u} = \mathbf{0}.} \tag{6-8}$$

It is important to mention that, the selected discretisation procedure, does not a priori enforce the pre-set Dirichlet conditions on the domain's boundary. The nodes of the mesh at this boundary are included amongst the degree of freedom of the state $\mathbf{c}$ and the have to be removed. This can be done either algebraically by the identification and the substitution of these variables with their value or numerically via their penalisation with sufficient large penalty parameters. From the discretisation of the objective functional two matrices arise. $M \in \mathbb{R}^{N_c \times N_c}$ is derived from the state related cost whereas $N \in \mathbb{R}^{3 \times 3}$ comes from the control related cost and they are both defines as:

$$\mathbf{M}_{ij} = \int_{\Omega_t} \phi_i \phi_j \, d\Omega, \qquad N_{ij} = \int_{\Omega_u} \chi_{\Omega_{u_i}}(\mathbf{x}) \cdot \chi_{\Omega_{u_j}}(\mathbf{x}) \, d\Omega. \tag{6-9}$$

Combining all these results together leads to the complete discretised PDE-constrained optimisation problem:

$$\boxed{\begin{aligned} \min_{\mathbf{c},\mathbf{u}} \quad & \frac{1}{2}(\mathbf{c} - \mathbf{c}_t)^T \mathbf{M}(\mathbf{c} - \mathbf{c}_t) + \frac{\alpha}{2}\mathbf{u}^T \mathbf{N}\mathbf{u} \\ \text{subject to} \quad & \mathbf{A}\mathbf{c} - \mathbf{B}\mathbf{u} = \mathbf{0}, \\ & \mathbf{u}_{\min} \leq \mathbf{u} \leq \mathbf{u}_{\max}, \\ & u_{\text{MinC}} \leq u_1 + u_2 + u_3. \end{aligned}} \tag{6-10}$$

It is important to mention that the $B$ matrix was simplified based on the assumption that the emission rate of each chimney cannot vary independently at every node of the mesh (input

is not distributed). On the contrary, it is localised at the areas of the chimneys. Thus the number of the variables of $\mathbf{u}$ can be considered equal to the degrees of freedom of the mesh at those areas and not of the whole domain ( everywhere else they would be a priori equal to zero). Furthermore, this idea was extended by assigning an average and uniform emission rate to the area of each chimney and as a result the number of control variables was set to three: $u_1,\ u_2,\ u_3$.

### 6-1-3 Coupling of Augmented Lagrangian and Domain Decomposition

Comparing the general form of the discrete optimisation problem (5-7) with this specific example (6-10) it is evident that:

$$\begin{aligned} \mathcal{J}_h(\mathbf{c},\mathbf{u}) =& \frac{1}{2}(\mathbf{c}-\mathbf{c}_d)^T\mathbf{M}(\mathbf{c}-\mathbf{c}_d)+\frac{\alpha}{2}\mathbf{u}^T\mathbf{N}\mathbf{u} \\ \mathcal{E}_h(\mathbf{c},\mathbf{u}) =& \mathbf{A}\mathbf{c}-\mathbf{B}\mathbf{u}=0 \\ Q_h(\mathbf{c},\mathbf{u}) =& \begin{bmatrix} \mathbf{u}-\mathbf{u}_{\max} \\ \mathbf{u}_{\min}-\mathbf{u} \\ u_{\mathrm{MinC}}-(u_1+u_2+u_3) \end{bmatrix} \leq 0. \end{aligned} \tag{6-11}$$

Having identified all the relative terms, it is now possible to derive the augmented Lagrangian function based on (5-8) as follows:

$$\begin{aligned} \mathcal{L}_\rho(\mathbf{c},\mathbf{u},\lambda,\mu) = \frac{1}{2}(\mathbf{c}-\mathbf{c}_d)^T\mathbf{M}(\mathbf{c}-\mathbf{c}_t)+\frac{\alpha}{2}\mathbf{u}^T\mathbf{N}\mathbf{u}\lambda^T(\mathbf{A}\mathbf{c}-\mathbf{B}\mathbf{u})+\frac{\rho}{2}(\mathbf{A}\mathbf{c}-\mathbf{B}\mathbf{u})^T(\mathbf{A}\mathbf{c}-\mathbf{B}\mathbf{u})+ \\ +\sum_{j=1}^{3}\frac{\mu_{U_j}}{\rho}\left[\exp\left(\rho\left(u_j-u_{\max}\right)\right)-1\right]+\sum_{j=1}^{3}\frac{\mu_{L_j}}{\rho}\left[\exp\left(\rho\left(u_{\min}-u_j\right)\right)-1\right]+ \\ +\frac{\mu_C}{\rho}\left[\exp\left(\rho\left(u_{\mathrm{MinC}}-u_1-u_2-u_3\right)\right)-1\right]. \end{aligned} \tag{6-12}$$

From now on, the optimisation algorithm begins an iterative process.

The next step is the minimisation of the $\mathcal{L}_\rho$ with respect to $\mathbf{c}$ and $\mathbf{u}$ (5-10). To reach the optimum a Newton method has been proposed and analysed. The gradient of the augmented Lagrangian (5-12) is equal to

$$\boxed{g_k = \begin{bmatrix} \mathbf{M}(\mathbf{c}^k-\mathbf{c}_d)+\mathbf{A}^T\left[\lambda^k+\rho\left(\mathbf{A}\mathbf{c}^k-\mathbf{B}\mathbf{u}^k\right)\right] \\ \alpha\mathbf{N}\mathbf{u}^k-\mathbf{B}^T\left[\lambda^k+\rho\left(\mathbf{A}\mathbf{c}^k-\mathbf{B}\mathbf{u}^k\right)\right]+\mathbf{s}_U^k-\mathbf{s}_L^k-s_C^k\cdot\mathbf{1}^{\mathbf{3}\times\mathbf{1}} \end{bmatrix}}. \tag{6-13}$$

since its two components are derived as follows:

$$\nabla_{\mathbf{c}}\mathcal{L}_\rho\left(\mathbf{c}^k,\mathbf{u}^k\right) = \underbrace{\mathbf{M}\left(\mathbf{c}^k-\mathbf{c}_d\right)}_{\nabla_{\mathbf{c}}\mathcal{J}_h}+\underbrace{\mathbf{A}^T\left[\lambda^k+\rho\left(\mathbf{A}\mathbf{c}^k-\mathbf{B}\mathbf{u}^k\right)\right]}_{E_c^T\left(\lambda^k+\rho\mathcal{E}_h\right)}+\underbrace{\mathbf{0}}_{\text{inequality contribution}}$$

$$\nabla_{\mathbf{u}}\mathcal{L}_\rho\left(\mathbf{c}^k,\mathbf{u}^k\right)=\underbrace{\alpha\mathbf{N}\mathbf{u}^k}_{\nabla_{\mathbf{u}}\mathcal{J}_h}-\underbrace{\mathbf{B}^T\left[\lambda^k+\rho\left(\mathbf{A}\mathbf{c}^k-\mathbf{B}\mathbf{u}^k\right)\right]}_{-E_u^T\left(\lambda^k+\rho\mathcal{E}_h\right)}+\underbrace{\mathbf{s}_U^k-\mathbf{s}_L^k-s_C^k\mathbf{1}}_{\text{inequality contribution}}$$

with

$$\begin{aligned}s_{U,i}^k&:=\mu_{U,i}^k\exp\left[\rho\left(u_j^k-u_{\max}\right)\right]\\ s_{L,i}^k&:=\mu_{L,i}^k\exp\left[\rho\left(u_{\min}-u_j^k\right)\right]\end{aligned}\qquad i=1,2,3. \tag{6-14}$$

and

$$s_C^k:=\mu_C^k\exp\left[\rho\left(u_{\text{MinC}}-u_1^k-u_2^k-u_3^k\right)\right]. \tag{6-15}$$

Following with the computation of the Hessian (5-13):

$$\boxed{H_k=\begin{bmatrix}\mathbf{M}+\rho\mathbf{A}^T\mathbf{A} & -\rho\mathbf{A}^T\mathbf{B}\\ -\rho\mathbf{B}^T\mathbf{A} & \alpha\mathbf{N}+\rho\mathbf{B}^T\mathbf{B}+\rho\,\text{diag}\left(\mathbf{s}_U^k+\mathbf{s}_L^k\right)+\rho s_C^k\cdot\mathbf{1}^{\mathbf{3\times3}}\end{bmatrix}} \tag{6-16}$$

from the individual terms

$$\nabla^2_{\mathbf{cc}}\mathcal{L}_\rho\left(\mathbf{c}^k,\mathbf{u}^k\right)=\underbrace{\mathbf{M}}_{\nabla^2_{\mathbf{cc}}\mathcal{J}_h}+\underbrace{\rho\mathbf{A}^T\mathbf{A}}_{\rho E_{\mathbf{c}}^T E_{\mathbf{c}}}+\underbrace{\mathbf{0}}_{+\sum_{i=1}^{n_c}\left[\left(\lambda^k+\rho\mathcal{E}_h\left(\mathbf{c}^k,\mathbf{u}^k\right)\right)\cancel{(E_{\mathbf{cc}})_i}\right]}+\underbrace{\mathbf{0}}_{s^k\cancel{Q_{\mathbf{cc}}}+\rho\cancel{Q_{\mathbf{c}}}^T D_s^k\cancel{Q_{\mathbf{c}}}}$$

$$\nabla^2_{\mathbf{cu}}\mathcal{L}_\rho\left(\mathbf{c}^k,\mathbf{u}^k\right)=\underbrace{\mathbf{0}}_{\cancel{\nabla^2_{\mathbf{cu}}\mathcal{J}_h}}+\underbrace{-\rho\mathbf{A}^T\mathbf{B}}_{\rho E_{\mathbf{c}}^T E_{\mathbf{u}}}+\underbrace{\mathbf{0}}_{+\sum_{i=1}^{3}\left[\left(\lambda^k+\rho\mathcal{E}_h\left(\mathbf{c}^k,\mathbf{u}^k\right)\right)\cancel{(E_{\mathbf{cu}})_i}\right]}+\underbrace{\mathbf{0}}_{s^k\cancel{Q_{\mathbf{cu}}}+\rho\cancel{Q_{\mathbf{c}}}^T D_s^k Q_{\mathbf{u}}}$$

$$\nabla^2_{\mathbf{uc}}\mathcal{L}_\rho\left(\mathbf{c}^k,\mathbf{u}^k\right)=\underbrace{\mathbf{0}}_{\cancel{\nabla^2_{\mathbf{uc}}\mathcal{J}_h}}+\underbrace{-\rho\mathbf{B}^T\mathbf{A}}_{\rho E_{\mathbf{u}}^T E_{\mathbf{c}}}+\underbrace{\mathbf{0}}_{+\sum_{i=1}^{n_c}\left[\left(\lambda^k+\rho\mathcal{E}_h\left(\mathbf{c}^k,\mathbf{u}^k\right)\right)\cancel{(E_{\mathbf{uc}})_i}\right]}+\underbrace{\mathbf{0}}_{s^k\cancel{Q_{\mathbf{uc}}}+\rho Q_{\mathbf{u}}^T D_s^k\cancel{Q_{\mathbf{c}}}}$$

$$\nabla^2_{\mathbf{uu}}\mathcal{L}_\rho\left(\mathbf{c}^k,\mathbf{u}^k\right)=\underbrace{\alpha\mathbf{N}}_{\nabla^2_{\mathbf{uu}}\mathcal{J}_h}+\underbrace{\rho\mathbf{B}^T\mathbf{B}}_{\rho E_{\mathbf{u}}^T E_{\mathbf{u}}}+\underbrace{\mathbf{0}}_{+\sum_{i=1}^{3}\left[\left(\lambda^k+\rho\mathcal{E}_h\left(\mathbf{c}^k,\mathbf{u}^k\right)\right)\cancel{(E_{\mathbf{uu}})_i}\right]}+\underbrace{\rho\,\text{diag}\left(\mathbf{s}_U^k+\mathbf{s}_L^k\right)+\rho s_C^k\cdot\mathbf{1}^{\mathbf{3\times3}}}_{s^k\cancel{Q_{\mathbf{uu}}}+\rho Q_{\mathbf{u}}^T D_s^k Q_{\mathbf{u}}}$$

The next step of the proposed methodology includes the decomposition of the system via the transformation of the state variable. The new set of variables $\mathbf{x}^*=\begin{bmatrix}E_c\cdot\Delta\mathbf{c} & \Delta\mathbf{u}\end{bmatrix}^T$ will result in a new Hessian that, based to (5-16), is equal to

$$H_k^*(\mathbf{c}^k,\mathbf{u}^k):=\begin{bmatrix}\underbrace{\nabla^2_{\mathbf{cc}}\mathcal{L}_\rho(\mathbf{c}^k,\mathbf{u}^k)\,E_{\mathbf{c}}^{-1}}_{(\mathbf{M}+\rho\mathbf{A}^T\mathbf{A})\mathbf{A}^{-1}} & \underbrace{\nabla^2_{\mathbf{cu}}\mathcal{L}_\rho(\mathbf{c}^k,\mathbf{u}^k)}_{-\rho\mathbf{A}^T\mathbf{B}}\\ \underbrace{\nabla^2_{\mathbf{uc}}\mathcal{L}_\rho(\mathbf{c}^k,\mathbf{u}^k)\,E_{\mathbf{c}}^{-1}}_{(-\rho\mathbf{B}^T\mathbf{A})\mathbf{A}^{-1}} & \underbrace{\nabla^2_{\mathbf{uu}}\mathcal{L}_\rho(\mathbf{c}^k,\mathbf{u}^k)}_{\alpha\mathbf{N}+\rho\mathbf{B}^T\mathbf{B}+\rho\,\text{diag}\left(\mathbf{s}_U^k+\mathbf{s}_L^k\right)+\rho s_C^k\cdot\mathbf{1}^{\mathbf{3\times3}}}\end{bmatrix}\Longrightarrow \tag{6-17}$$

$$H_k^*(\mathbf{c}^k,\mathbf{u}^k)=\begin{bmatrix}\mathbf{M}\mathbf{A}^{-1}+\rho\mathbf{A}^T & -\rho\mathbf{A}^T\mathbf{B}\\ -\rho\mathbf{B}^T & \alpha\mathbf{N}+\rho\mathbf{B}^T\mathbf{B}+\rho\,\text{diag}\left(\mathbf{s}_U^k+\mathbf{s}_L^k\right)+\rho s_C^k\cdot\mathbf{1}^{\mathbf{3\times3}}\end{bmatrix}.$$

The only remaining step before the computation of the Newton step is the calculation of the matrix $\mathbf{Z}$, Each one of its rows will be calculated through the solution of a linear system with the use of a typical domain decomposition method:

$$\mathbf{Z}_i = \mathbf{M}_i \mathbf{A}^{-1} \implies \mathbf{Z}_i \mathbf{A} = \mathbf{M}_i \implies \mathbf{A}^\top (\mathbf{Z}_i)^\top = (\mathbf{M}_i)^\top \tag{6-18}$$

Then the linear system that provides the Newton step of the transformed variables:

$$H_k^* \left( \mathbf{c}^k, \mathbf{u}^k \right) \cdot \begin{bmatrix} \mathbf{x}_\mathbf{c}^* \\ \Delta \mathbf{u} \end{bmatrix} = -g_k \left( \mathbf{c}^k, \mathbf{u}^k \right) \tag{6-19}$$

can be solved through the proposed substructuring domain decomposition methodology (5-25) and (5-26).

The final step of the inner optimisation problem is the computation of the original variable's Newton step via the solution of :

$$E_c \, \Delta c = \mathbf{x}_\mathbf{c}^* \tag{6-20}$$

with the use of the typical substructuring domain decomposition methodology

Having identified the optimiser of the augmented Lagrangian with respect to $\mathbf{c}$ and $\mathbf{u}$:

$$\left( \mathbf{c}^{k+1}, \mathbf{u}^{k+1} \right) = \arg\min_{\mathbf{c},\mathbf{u}} \mathcal{L}_\rho \left( \mathbf{c}^k, \mathbf{u}^k, \lambda^k, \mu^k \right) \implies \left( \mathbf{c}^{k+1}, \mathbf{u}^{k+1} \right) = \left( \mathbf{c}^k + \Delta c, \mathbf{u}^k + \Delta u \right) \tag{6-21}$$

the inner optimisation problem is complete and in the next phase, the Lagrangian multiplies and the penalty parameter will be updated:

$$\begin{aligned}
\boldsymbol{\lambda}^{k+1} &= \boldsymbol{\lambda}^k + \rho^k \left( \mathbf{A}\mathbf{c}^{k+1} - \mathbf{B}\mathbf{u}^{k+1} \right) \\
\mu_{U_i}^{k+1} &= \mu_{U_i}^k \exp \left[ \rho^k \left( u_i^{k+1} - u_{\max} \right) \right], \qquad i = 1, 2, 3, \\
\mu_{L_i}^{k+1} &= \mu_{L_i}^k \exp \left[ \rho^k \left( u_{\min} - u_i^{k+1} \right) \right], \qquad i = 1, 2, 3, \\
\mu_C^{k+1} &= \mu_C^k \exp \left[ \rho^k \left( u_{\mathrm{MinC}} - u_1^{k+1} - u_2^{k+1} - u_3^{k+1} \right) \right] \\
\rho^{k+1} &= \tau \rho^k.
\end{aligned} \tag{6-22}$$

## 6-2 Numerical Results of the Case Study - Air-Pollution Control

In the previous section all of the necessary steps of the designed algorithm are presented. The discretisation of the system with standard Galerkin finite elements was performed through the FreeFEM [71] open-source software whereas the matrix manipulation and the visual presentation of the results was done in MATLAB. Those simulation were performed in my personal laptop equipped with an Intel Core i7-10750H, 16 GB of RAM and 2.60 GHz based clock frequency. However, as the final step, the execution of the algorithm requires the specification of the problem's physical parameters and the appropriate tuning of the optimisation parameters.

### 6-2-1 Numerical Setup and Tuning

This case study of optimal air-pollution control, is to be used as a benchmark test that will provide insight on the designed methodology. This example is not drawn from an existing industrial plant at a specific location. Nevertheless, it is an objective of this thesis to present a real world example and not just a mathematical exercise. To that end, the physical parameters associated with the atmosphere and the emission sources (such as the wind velocity, the maximum emission rate, the exit gas velocity, etc.) will be estimated based on a medium scale coal plant.

However, this is not the only objective. The purpose of these numerical examples is also to evaluate both the numerical efficiency of the methodology and the quality of the optimum solution. Thus, the resulting optimisation problem should not be trivial. This issue can be addressed with the appropriate positioning of the three chimneys and the area of interest within the domain. The knowledge of the air velocity, the diffusion coefficient and the spatial dimensions of the domain can provide an approximation on region of influence for each of the chimneys. If they are positioned too close to each other then their regions of influence will be almost identical and the optimal control of all three will be almost the same. If they are too far apart then their area of influence will not overlap and as a result the mitigation of the air-pollution will be limited to the control of only one of the chimneys. This is why both the chimneys and the area of interest are selected so that, under the specified atmospheric parameters, they will lead to a coupled optimisation system. The area of influence of each chimney overlaps with the area of interest at a different extent, resulting in distinct optimal control inputs.

Moreover, another issue that arises, is associated the numerical simulation of the advection diffusion PDE with standard Galerkin finite elements. It is known that in advection dominated cases, these type of FEM will result in non-physical oscillations of the solution. Typical techniques that resolve this issue is the use of more advanced finite elements such as the SUPG [72]. In the presented example however, this problem was not addressed directly but rather avoided. Using the Péclet number as a reference, the selection of advection dominated examples was prevented through the appropriate choice of the associated physical parameters. The Péclet number is derived through the formula:

$$Pe = \frac{\|\mathbf{V}\| \cdot h}{2\nu} \tag{6-23}$$

and when it is lower that one, the resulting solution will not have any oscillations. In this formula, $\mathbf{V}$ denotes the wind speed, $\nu$ the diffusion coefficient and $h$ is the size of the mesh's elements.

Based on the aforementioned requirements, the values the physical parameters of the air pollution optimal control problem are specified in Table 6-1

**Table 6-1:** Air-Pollution Physical Parameters

| **Parameter** | **Symbol** | **Value** |
|---|---|---|
| *Domain configuration* | | |
| Domain Length | $L_x$ | 2000 $[m]$ |
| Domain Width | $L_y$ | 2000 $[m]$ |
| *Transport parameters* | | |
| Wind Velocity Magnitude | $\|\mathbf{V}\|$ | 2 $[m/s]$ |
| Wind Velocity Direction | $\theta$ | $0.85 \cdot \pi/2$ rad |
| Diffusion Coefficient | $\nu$ | 50 $[m^2/s]$ |
| *Emission Sources* | | |
| Number of Chimneys | $n_u$ | 3 [-] |
| Position of Chimney #1 | $x_{u_1}, y_{u_1}$ | $\begin{bmatrix}350 , & 350\end{bmatrix}$ [m × m] |
| Position of Chimney #2 | $x_{u_2}, y_{u_2}$ | $\begin{bmatrix}1000 , & 350\end{bmatrix}$ [m × m] |
| Position of Chimney #3 | $x_{u_3}, y_{u_3}$ | $\begin{bmatrix}1650 , & 350\end{bmatrix}$ [m × m] |
| Minimum Emission Rate | $u_{min}$ | 0 $[\mathrm{kg} \cdot \mathrm{s}^{-1}]$ |
| Maximum Emission Rate | $u_{max}$ | 1 $[\mathrm{kg} \cdot \mathrm{s}^{-1}]$ |
| Minimum Cumulative Emission Rate | $u_{MinC}$ | 1.8 $[\mathrm{kg} \cdot \mathrm{s}^{-1}]$ |
| Reference Volume of Air Emitted per Second | $V_R$ | 300 $[\mathrm{m}^3]$ |
| *Area of Interest* | | |
| Position of Area of Interest | $x_t, y_t$ | [1300 , 1700] [m × m] |
| Radius of Area of Interest | $R_t$ | 100 [m] |
| Target Concentration | $c_t$ | 0.005 $[kg/m^3]$ |

Based on the selected parameters, the optimisation problem is well-defined since all of its physical properties have been quantified. However, for the proposed optimisation algorithm, there are still some parameters that have to be defined as seen in Table 6-2.

First of all, for the discretisation of the optimisation problem, the size of the mesh within the specified domain must be selected. If the accuracy of the simulation is lacking then the mesh must be refined and the total number of DoFs will increase. Then, there are the values of the parameters that require initialisation for the optimisation algorithm to begin. These include the initial values for the decision variables and the multipliers. Finally, the last group of parameters is associated with the convergence of the algorithm which is assessed through several criteria . The convergence of the inner minimisation of the augmented

Lagrangian is evaluated based on the norm of the step along with the norm of the gradient computed at the suggested optimum. The outer optimisation is evaluated based on the equality constraint residual and the violation of the inequality constraints. The optimisation procedure is terminated when all preset tolerances are satisfied simultaneously or when the maximum permissible number of iterations has been reached.

**Table 6-2:** Tuning of Optimisation Parameters

| **Parameter** | **Symbol** | **Value** |
|---|---|---|
| *Domain decomposition* | | |
| Number of DoFs | $N_{DoFs}$ | 4163 |
| Number of subdomains | $N_s$ $(N_x \times N_y)$ | 4 $(2 \times 2)$ |
| *Augmented Lagrangian parameters* | | |
| Initial penalty parameter | $\rho^0$ | 1 |
| Initial equality multiplier parameter | $\lambda^0$ | $10^{-6}$ |
| Initial inequality multiplier parameter | $\mu^0$ | $10^{-6}$ |
| Initial state | $\mathbf{c}^0$ | 0 |
| Initial control | $\mathbf{u}^0$ | $\begin{bmatrix}0.6 & 0.6 & 0.6\end{bmatrix}^\top$ |
| Regularisation parameter | $\alpha$ | $10^{-6}$ |
| Equality tolerance | $\epsilon_E$ | $10^{-5}$ |
| Inequality tolerance | $\epsilon_I$ | $10^{-4}$ |
| Stationarity tolerance | $\epsilon_g$ | $10^{-5}$ |
| Step tolerance | $\epsilon_S$ | $10^{-6}$ |
| Maximum Number of Iterations | IterMax | 100 |

## 6-2-2 Numerical Results

Based on the described methodology and the parameters' selected value, the proposed algorithm was executed. The initial domain of $N_{DoFs}$ degrees of freedom was decomposed into four subdomains with an average of 1002 DoFs each. The number of interface variables was computed to be 154. It converged to a solution after 34 outer steps with an average of 2.5 inner iterations. The optimal control is :

$$u_1 \approx 72.6\%\ u_{max} \qquad u_2 \approx 23.1\%\ u_{max} \qquad u_3 \approx 84.3\%\ u_{max}$$

The cumulative emission rate is almost equal to 1.8 $kg \cdot s^{-1}$, within the acceptable margin ($\geq u_{MinC} - \epsilon_I$). The average concentration over the are of interest is computed at $5.046 g \cdot s^{-1}$ which is less than one percent away from the desired concentration. The final solution can be seen in Figure 6-1 and Figure 6-2

## Optimal Concentration Field

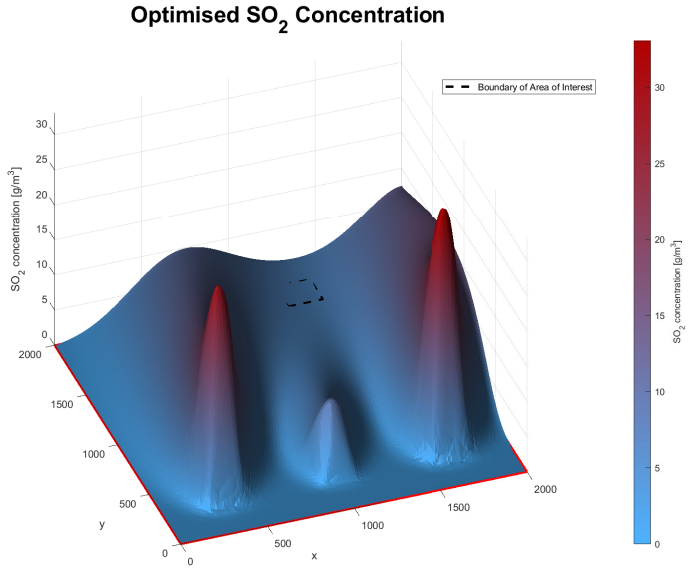



**Figure 6-1:** Optimised $SO_2$ concentration from the proposed algorithm.

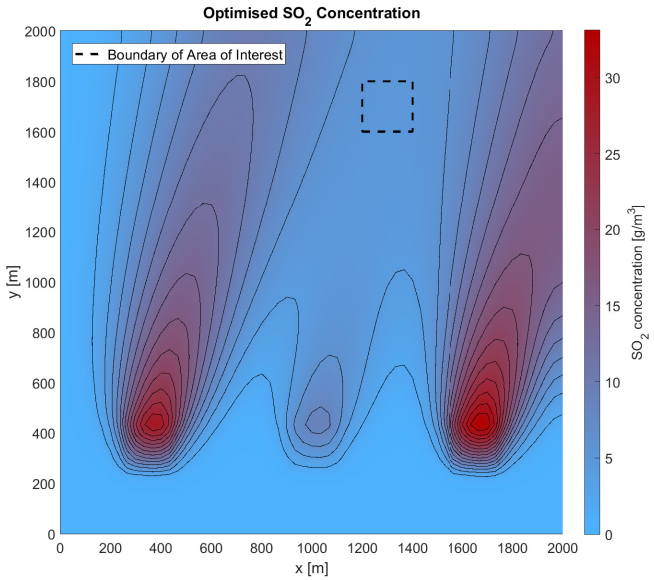



**Figure 6-2:** Optimised $SO_2$ concentration from the proposed algorithm.

### Optimal Concentration Field in Refined Mesh

As the case study is repeated with a denser mesh and more DoFs, the optimal control shows a minor fluctuation of $0.5\,‰ u_{max}$. For example, the doubling of the available DoFs, compared to the previous test, will result in the following optimal state:

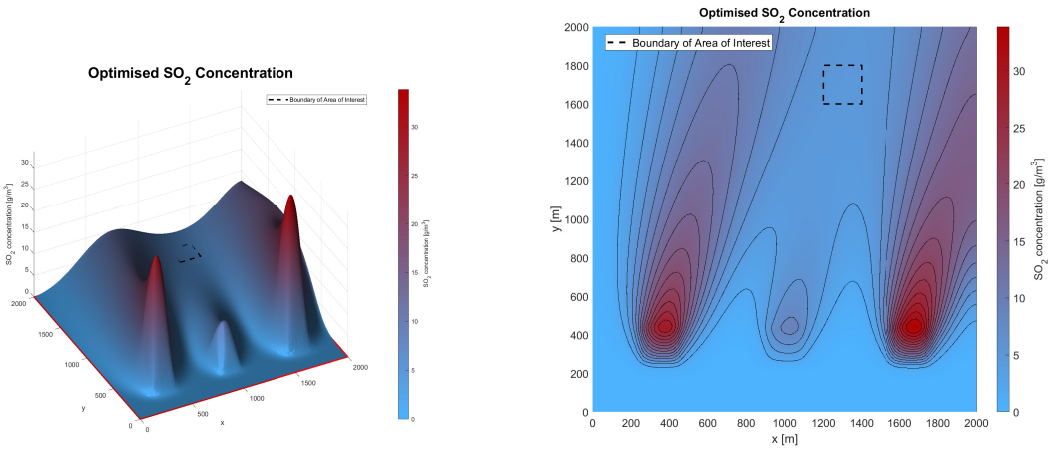



**Figure 6-3:** Optimised $SO_2$ concentration from the proposed algorithm with $7464$ DoFs.

### Uncontrolled Pollutant Dispersion

Assuming that the control of the emissions rates $\begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix}$ is not derived from an optimisation procedure but it is set to be uniform between the chimneys and equal to 60% of the maximum emission rate. Then, the derived dispersion of the pollution through the atmosphere would be as seen in Figure 6-4

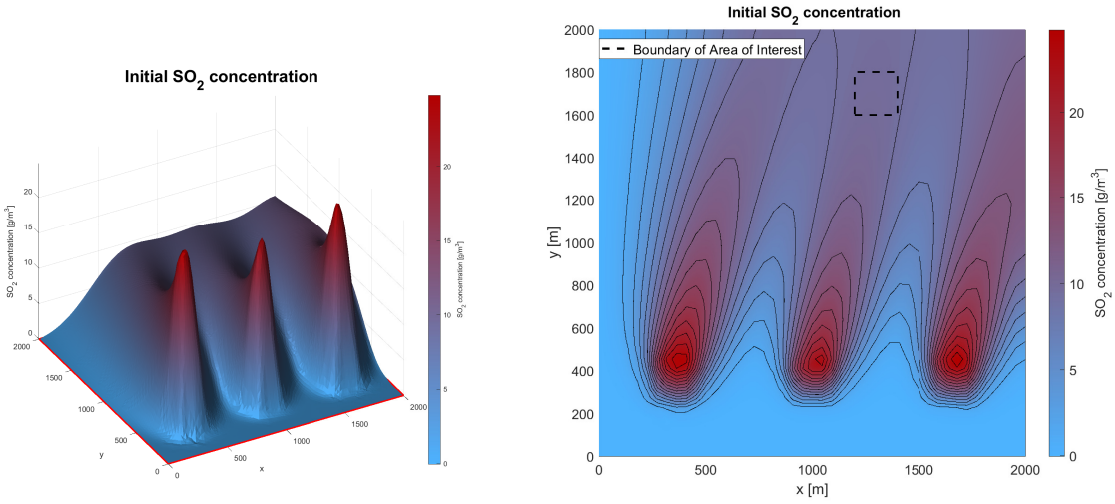



**Figure 6-4:** Initial $SO_2$ concentration with uniform emission rate.

In this case the average concentration at the are of interest is close to 9.6 $g \cdot m^{-3}$, which is almost double the targeted concentration.

These simulations demonstrate that the proposed algorithm can indeed reach an optimal state for both the concentration and the emission rates of the problem. However, until now, there has not been offered any indication regarding its numerical efficiency. This attribute is equally, if not more, important for a successfully designed optimisation framework with practical applicability.

## 6-3 Numerical Performance and Efficiency of the Proposed Method

In order to assess the numerical performance and efficiency of the proposed methodology a series of benchmark tests have been performed. At first, the presented example was run repeatedly with increasing number of total degrees of freedom while it was decomposed into four subdomains. The derived data will clarify on the effect of the mesh size to a selected decomposition. At the same time, the presented case study will be repeated for the same number of DoFs but an increasing number of subdomains. It is important to mention that the computer code for the implementation of the proposed method does not incorporate the communication and techniques required for the parallelisation. Instead it computes the results sequentially and by recording the necessary data, it presents the parallel version of the computations.

### 6-3-1 Linear Case Study

Based on the described case study, the designed technique will be compared to the direct computation of the linear system derived from the Newton step seen in Equation 5-11. A variety of tests will be conducted, to make sure that the solution of this linear system, is computed faster via the proposed decomposition in comparison typical numerical techniques. This is not a detrimental test since the proposed methodology does not just decomposes one matrix but a plethora of them along its steps for the computation of the optimum $(\mathbf{Z_i}, H_k^*, \mathbf{x_c^*})$. The computational efficiency of this strategy can be seen in Table 6-3.

**Table 6-3:** Computational Results in the Linear Case Study

| Subdomains | DOFs | $N_\Gamma$ | $\widetilde{N}_{Inter}$ | Outer Iterations | Avg. Inner Iterations | Efficiency |
|---|---|---|---|---|---|---|
| 4 $(2 \times 2)$ | 4163 | 154 | 1002 | 34 | 2.5 | 45.2 |
| 4 $(2 \times 2)$ | 11599 | 250 | 2837 | 61 | 3.0 | 73.1 |
| 4 $(2 \times 2)$ | 22799 | 350 | 5612 | 100 | 3.2 | 90.0 |
| 4 $(2 \times 2)$ | 37679 | 432 | 9312 | $\sim 100$ | $\sim 3.3$ | $\sim 137$ |
| 4 $(2 \times 2)$ | 11599 | 250 | 2837 | 61 | 3.0 | 73.1 |
| 9 $(3 \times 3)$ | 11599 | 514 | 1232 | 61 | 3.0 | 171 |
| 36 $(6 \times 6)$ | 11599 | 1246 | 287 | 61 | 3.0 | 348 |
| 81 $(9 \times 9)$ | 11599 | 1973 | 119 | 61 | 3.0 | 126 |

The numerical examples were conducted for a varying number of subdomains and degrees of freedom. The reported data include the total number of interface variables ($N_\Gamma$) together with the average interior variables per subdomain ($\widetilde{N}_{Inter}$). In addition, the number of outer iterations along with the average number of the inner iterations are presented. Finally, the efficiency of the proposed methodology is evaluated as a ratio of computational time of the direct case divided by the parallelised one. The computational costs are computed based on the inversion of the required matrices and not the assembly and the mesh generation since it comprises more than 85% of the total computation time [73].

However, this is not an very accurate representation of the method capabilities. In the previous methodology the inverse matrix are computed explicitly which increases the computational cost and memory requirements immensely. A much more standard practice is the derivation of the solution of the linear system without ever computing the inverse matrix. Based on this approach the numerical results are seen in Table 6-4.

**Table 6-4:** Practical Computational Results in the Linear Case Study

| Subdomains | DOFs | $\mathbf{N_\Gamma}$ | $\mathbf{\widetilde{N}_{Inter}}$ | Outer Iterations | Avg. Inner Iterations | Efficiency |
|---|---|---|---|---|---|---|
| 4 $(2 \times 2)$ | 4163 | 154 | 1002 | 27 | 2.7 | 6.8 |
| 4 $(2 \times 2)$ | 11599 | 250 | 2837 | 50 | 2.4 | 8.4 |
| 4 $(2 \times 2)$ | 22799 | 350 | 5612 | 100 | 2.7 | 7.8 |
| 4 $(2 \times 2)$ | 37679 | 432 | 9312 | 100 | 3.1 | 6.4 |
| 4 $(2 \times 2)$ | 22799 | 350 | 5612 | 100 | 2.7 | 7.8 |
| 9 $(3 \times 3)$ | 22799 | 667 | 2459 | 100 | 2.7 | 29 |
| 36 $(6 \times 6)$ | 22799 | 1728 | 585 | 100 | 2.7 | 47 |
| 81 $(9 \times 9)$ | 22799 | 2703 | 248 | 100 | 2.7 | 22 |

This is a much more accurate representation of the methodology's efficiency. But again, this is not end of the story. There are much more advanced numerical solvers that can speed up the computation for both the Newton step (5-11) and the linear system that is derived via the Schur complement (4-13). These approaches may involve approximations based on iterative solvers or the exploitation of the underlying matrix structure. However, the identification of the optimum generic strategies for the solution of large and sparse systems beyond the DDM approach is outside the scope of this work.

Nevertheless, the derivation of an more representative assessment of the proposed methodology's capabilities would require a comparison between the aforementioned approaches. For this purpose, the Matlab built-in capabilities for linear system solutions were utilised. Instead of enforcing specific method for all the linear systems, Matlab was allowed to choose whatever strategy it considered the best. The results of these tests are seen in Table 6-5

From these results it is evident that there are numerical techniques that speed up the direct approach by capitalising on the inner structure of the Hessian. However, the proposed approach is still beneficial for the right combination of $\mathbf{\widetilde{N}_{Inter}}$ and $\mathbf{N_\Gamma}$

**Table 6-5:** Advanced Computational Results in the Linear Case Study

| Subdomains | DOFs | $N_\Gamma$ | $\widetilde{N}_{Inter}$ | Outer Iterations | Avg. Inner Iterations | Efficiency |
|---|---|---|---|---|---|---|
| 4 $(2 \times 2)$ | 4163 | 154 | 1002 | 27 | 2.5 | 2.5 |
| 4 $(2 \times 2)$ | 11599 | 250 | 2837 | 50 | 2.4 | 1.4 |
| 4 $(2 \times 2)$ | 22799 | 350 | 5612 | 100 | 2.2 | 1.3 |
| 4 $(2 \times 2)$ | 37679 | 432 | 9312 | 100 | 2.2 | 1.1 |
| 4 $(2 \times 2)$ | 22799 | 350 | 5612 | 100 | 2.2 | 1.4 |
| 9 $(3 \times 3)$ | 22799 | 667 | 2459 | 100 | 2.2 | 2.5 |
| 36 $(6 \times 6)$ | 22799 | 1728 | 585 | 100 | 2.2 | 2.0 |
| 81 $(9 \times 9)$ | 22799 | 2703 | 248 | 100 | 2.2 | 0.9 |

### 6-3-2 General Case Study

The previous example can be used as an benchmark test for all the cases of linear PDEs. However, the situation changes dramatically when non-linearities are included in the algorithm. Every simplification that is seen in Equation 6-16 and Equation 6-17 is no longer valid. Those matrices will have to be recomputed at every iteration since, through the linearisation of the associated terms, they are dependent on the current values of $(\mathbf{c^k}, \mathbf{u^k})$. But the most important distinction is the computation of the $\mathbf{M_{cc}} \cdot \mathbf{E_c^{-1}}$ term. In the linear case, the $\mathbf{M_{cc}}$ is derived based on the cost functional and it is constant. As a result it will have to be computed only once in the beginning of the algorithm and the it can be reused at every iteration. Whenever an non-linear term is present that means that $\mathbf{M_{cc}}$ is dependent on the current state and thus, it will have to be computed at every inner iteration as well. This is a significant computational cost that in most cases dominates the whole procedure as seen in Table 6-6.

**Table 6-6:** Computational Results in the General Case Study

| Subdomains | DOFs | $N_\Gamma$ | $\widetilde{N}_{Inter}$ | Outer Iterations | Avg. Inner Iterations | Efficiency |
|---|---|---|---|---|---|---|
| 4 $(2 \times 2)$ | 4163 | 154 | 1002 | 27 | 2.7 | 0.24 |
| 4 $(2 \times 2)$ | 11599 | 250 | 2837 | 50 | 2.3 | 0.07 |
| 4 $(2 \times 2)$ | 22799 | 350 | 5612 | 100 | 2.2 | 0.03 |
| 4 $(2 \times 2)$ | 37679 | 432 | 9312 | $\sim 100$ | $\sim 2.3$ | 0.02 |
| 4 $(2 \times 2)$ | 22799 | 350 | 5612 | 100 | 2.2 | 0.03 |
| 9 $(3 \times 3)$ | 22799 | 667 | 2459 | 100 | 2.2 | 0.06 |
| 36 $(6 \times 6)$ | 22799 | 1728 | 585 | 100 | 2.2 | 0.05 |
| 81 $(9 \times 9)$ | 22799 | 2703 | 248 | 100 | 2.2 | 0.02 |

These results were to be expected. For the direct inversion of a large, sparse and almost symmetric matrices Matlab has developed optimised numerical methods that have been a product of many decades of work. These techniques take advantage of the sparsity of the

matrix and can produce a fast result without explicitly deriving the inverse of the Hessian. With that in mind there are numerical techniques and other tricks that could be used to improve the proposed methodology.

The "computational bottleneck" of the designed algorithm is the computation of the $\mathbf{Z}$ matrix. Not only it requires the solution of numerous (equal to the number of non-zero rows of $\mathbf{M_{cc}}$) linear systems that are equivalent to the solution of a PDE, but in the non-linear case they would have to be computed at every inner iteration of the optimisation algorithm. In an attempt to limit its effect on the computational performance, a set of options were identified.

First of all, a possible improvement is the introduction of additional processors that will be allocated towards the computation of the different rows of $\mathbf{Z}$ instead of the decomposition of the domain into more subdomains. In the present case, the processors are equal to the total number of the subdomains. If more sets of processors could be dedicated to the solution of the aforementioned linear systems then the efficiency of the methodology would increase. That approach however, could be easily hit a wall due to the limited hardware availability along with the polynomial growth in the number of the required processors.

Secondly, the simplification of the matrix $\mathbf{M_{cc}}$, as was suggested before, can reduce the total number of non-zero rows and thus lower the number of the linear systems that are required for the computation of $\mathbf{Z}$. In general, that simplification would mean the transition to a quasi-Newton method, since partial information of the Hessian will be lost.

In addition, the cost functional could be selected in a way that enhances the numerical efficiency of the proposed methodology without deviating from the optimal state. If the application can facilitate the choice of a much smaller area of interest then it is guaranteed that the number of non-zero rows of $\mathbf{M_{cc}}$ will reduce. In the specific case of air-pollution, the steady state of the advection-diffusion equation is studied. In addition, the area of interest is far away from the emission regions and as a result, the solution displays no abrupt changes. Thus, instead of comparing the values over all of the nodes of the mesh associated with the area of interest, the algorithm could evaluate the solution on a few point in the boundary of that region and still reach the same optimum state.

Finally, the presented options are ideas on how to improve the efficiency of the proposed methodology in comparison to the direct approach. Nevertheless, there is also the absolute computational efficiency, meaning the total time required for the algorithm to reach the optimum state. The computation of the matrix $\mathbf{Z}$ must be performed at every inner iteration of the Newton-step which increases the total time of the optimisation procedure even with the suggested parallelisation. To that end, the approach of inexact augmented Lagrangian is proposed. In this technique, the inner Newton-step will be performed only once before the update of the Lagrange multipliers. This will most probably increase the required number of the outer iterations but it is was seen that it reduced the total computations of the $\mathbf{Z}$ matrix since it reduced the total number of iterations.

The results of these changes can be seen in Table 6-7. We can see that the proposed methodology has regained its efficiency at the expense of an increased number of processors.

**Table 6-7:** Improved Computational Results in the General Case Study

| **Subdomains** | **DOFs** | $\mathbf{N_\Gamma}$ | $\widetilde{\mathbf{N}}_{\mathbf{Inter}}$ | **Outer Iter.** | **Size of Z** | **Efficiency** | **Max Efficiency (Processors )** |
|---|---|---|---|---|---|---|---|
| 4 $(2 \times 2)$ | 4163 | 154 | 1002 | 35 | 49 | 0.12 | 2.38 (196) |
| 4 $(2 \times 2)$ | 11599 | 250 | 2837 | 65 | 133 | 0.04 | 1.57 (532) |
| 4 $(2 \times 2)$ | 22799 | 350 | 5612 | 100 | 269 | 0.02 | 1.43 (1076) |
| 4 $(2 \times 2)$ | 37679 | 432 | 9312 | 100 | 437 | 0.01 | 0.98 (1748) |
| 4 $(2 \times 2)$ | 22799 | 350 | 5612 | 100 | 269 | 0.02 | 1.26 (1076) |
| 9 $(3 \times 3)$ | 22799 | 667 | 2459 | 100 | 269 | 0.02 | 1.46 (2421) |
| 36 $(6 \times 6)$ | 22799 | 1728 | 585 | 100 | 269 | 0.02 | 1.23 (9684) |
| 81 $(9 \times 9)$ | 22799 | 2703 | 248 | 100 | 269 | 0.01 | 0.86 (21789) |

## Synopsis of Chapter 5

In this chapter several numerical tests were conducted so as to evaluate the performance and efficiency of the proposed methodology. The simulated data and the comparison with other direct methods has provided an in-depth understanding of the major advantages and disadvantages of the designed algorithm. The optimisation framework is able to converge to an optimal state in the case study of air-pollution control under a variety of conditions. Its numerical efficiency varies depending on many factors but based on the total number of DoFs there appears to exist a critical number of subdomains for which the computational performance of the methodology is maximised.

# Chapter 7

# Conclusion

## 7-1 Concluding Remarks

The main goal of this thesis is to investigate the implementation of domain decomposition methods in PDE-constrained optimisation problems, that are associated with air-pollution control. To achieve this, an overview of the different types and characteristics of PDEs, that can be used to simulate the transport of pollutants in the atmosphere, was performed. Moreover, the principal requirements of an optimisation algorithm, designed to mitigate the concentration of pollutants in the air, were identified. This information eventually constituted the prerequisites of the proposed framework.

Following this, an examination of the various PDE-constrained optimisation methods was conducted. Emphasis was placed the techniques that are able to accommodate the necessary attributes of the different state equations, cost functionals and constraints that are crucial in air-pollution control. Contrary to most conventional approaches, the designed optimisation framework primarily focused on discretise-then-optimise strategies combined with the All-at-Once methodology. This was motivated by the simplicity and flexibility of a finite dimensional optimisation problem and by the fact that, keeping both the state and the control as decision variables, helps us preserves a structure that is similar the discretisation of the underlying PDE. In addition, emphasis was given to the analysis of the domain decomposition methods and specifically the substructuring approach as a solution to the increased computational costs of the derived large and sparse linear systems.

Then, this thesis developed and evaluated an optimisation algorithm based on the framework of PDE-constrained optimisation and domain decomposition methods. An augmented Lagrangian approach along with a Newton-step algorithm was implemented in order to deal with the constraints and ensure rapid convergence. Furthermore, to overcome the increased computational cost of the derived linear system, substructuring methods were utilised. The applicability of domain decomposition methodologies was dependent on the identification of a structure that, similar to the solution of a PDE, allows the decomposition of the linear systems into a set of smaller ones that can be solved in parallel. At places where that attribute was missing, it was artificially reconstructed to meet with the demands of the methodology.

The numerical experiments demonstrated that the proposed optimisation methodology manages to reduce the effects of air-pollution while at the same time satisfies the underlying PDE along with any other equality or inequality constraint. The impact of DDM in the optimisation procedure is also indicated in its comparison to conventional numerical techniques, as far as the computational cost is concerned. On the other hand, the restrictions of the designed algorithm were made clear through this rigorous testing. One clear example, is the convergence of the augmented Lagrangian which appears to be strongly dependent on both the initialisation, the stopping criteria and the update strategy of the Lagrangian and penalty parameters.

Another objective, of the thesis, was to design an optimisation algorithm that is capable of handling the complicated and non-linear governing equations and constraints. In theory, this goal was only partially achieved. The proposed algorithm can accommodate a broad class of equality or inequality constraint as long as it is localised is a subregion of the domain. As this portion grows larger, the number of the interface variables will become comparable to the number of general DoFs, thus leading to a substantial drop in the computational efficiency. Similarly, non-linearities in the PDE can be incorporated in the algorithm if they are local or if the Newton-step is substituted by a less restrictive approximation. Nevertheless, these assumptions are compatible with the application of air-pollution control and as a result the algorithm can be used in real-world scenarios without any practical limitation. Furthermore, the proposed method can facilitate the handling of constraints, an attribute that is not present in other established combinations of PDE constrained optimisation methods that are coupled with DDM.

Overall, the proposed framework manages to reach the optimal state of the problems in which it was tested. It is able to handle the prerequisites of a optimisation framework focused on air-pollution control and it can be extended to many other applications. Its numerical efficiency is shown to surpass methods with direct approaches in many cases and shows promise for additional improvement.

## 7-2 Recommendations for future work

The theoretical analysis of domain decomposition methods and PDE-constrained optimisation, along with an extensive numerical investigation, has been pivotal in the deeper understanding of this topic and the proposed methodology. For that reason there have been identified many areas for future research in which the proposing algorithm can be optimised with regard it numerical implementation and compete with the performance of other well-established methodologies.

First of all, the efficiency comparison in complicated cases has shown that the methodology requires improvement in this area and more specifically in the computation of the $\mathbf{M_{cc}E_c}^{-1}$ term. Some numerical tricks have been already been proposed that manage to surpass the performance of Matlab in this specific case study (additional processors, simplification of objective function, inexact Lagrangian). However, the designed algorithm can be improved further. Some ideas that warrant further investigation are:

- Small changes in the steps of the proposed algorithm to improve the condition of the resulting matrices (Section C-1).

- The implementation of Krylov solvers with preconditioners for the Schur complement linear system of the decomposed Hessian (Section C-2).
- The implementation of different approaches for the transformation of the inner structure of the Hessian that does not require the $\mathbf{M_{cc}E_c}^{-1}$ term (Section C-3).

Furthermore, for the specific application of air-pollution control, the implementation of finite element methods that more suitable than the standard Galerkin elements, such as the SUPG, is imperative for the accurate representation of advection dominated transport phenomena. It would also be of great value to assess the methodology against measured data and validate its convergence towards the optimal state. Beyond the air-pollution, the proposed methodology could also be investigated in other classes of PDE-constrained optimisation problems, including shape optimisation, since they share several structural and computational characteristics. Moreover, the algorithm requires a more robust convergence criteria, since the use of the global norm of the gradient or the step vector becomes more and more restrictive as the total number of DoFs increases which leads to unnecessary delays. This is also related to the need for a better balancing and scaling of the terms associated with the cost functional of the augmented Lagrangian.

Finally, it is suggested that the proposed algorithm is incorporated into the FreeFEM software environment. Although FreeFEM is primarily designed for finite element analysis, it can be coupled with many external libraries (PETSc,HPDDM,Scotch) that support the numerical solution of large linear systems and can handle the communication between the subdomains in domain decomposition methods. In this thesis, Matlab was utilised since it offers the level of customisation that is required for the proof of a concept and the benchmark testing of a designed algorithm as well as the visualisation of the results. The goal of the algorithm was to provide insight to the proposed methodology rather than optimise the numerical computing. To that end, the integration of already optimised advanced linear iterative solvers, preconditioners, and parallel frameworks could significantly improve the computational efficiency and scalability of the proposed method.

# Appendix A

# PDE-Constrained Optimisation & Domain Decomposition

## A-1 Implementation of DDM into PDE-Constrained Optimisation

The simple example presented in this appendix, aims to provide insight on the implementation of already established domain decomposition methods in a PDE constrained optimisation framework. Assume a linear-quadratic distributed optimal control problem, subject to only an elliptic state equation:

$$\begin{aligned} \min_{y,u} J(y,u) &= \frac{1}{2}\int_{\Omega}(y-y_t)^2\,\mathrm{d}\mathbf{x} + \frac{\alpha}{2}\int_{\Omega}u^2\,\mathrm{d}\mathbf{x} \\ \text{s.t. } &\begin{cases} -\Delta y = f+u & \text{in } \Omega \\ y = 0 & \text{on } \partial\Omega \end{cases} \end{aligned} \tag{A-1}$$

Here, $y$ denotes the state variable, $u$ the distributed control, $y_t$ is the target state, and $\alpha > 0$ is a regularization parameter. In order to define a solution constrained by this state equation $y$ would require to be twice differentiable. However, it is not easy to solve an optimisation problem under the assumption of a "strong" solution. It is much more advantageous to assume that only the first derivative of $y$ will exist and that it will be square-integrable. This "weak" solution will help the derivation of an optimal solution and will simplify the discretisation through finite element methods.

Assuming a test function $v$, the state equation can be reformulated by multiplying with it and then integrating, as follows:

$$\int_{\Omega}\nabla y \nabla v\, d\Omega = \int_{\Omega} u\, v\, d\Omega + \int_{\Omega} f\, v\, d\Omega. \tag{A-2}$$

Then the discretisation of the problem will transform this infinite dimensional OCP into a finite dimensional optimisation problem. The discrete state $V_h$ and control $U_h$ spaces are

derived from the basis functions $\phi(x)$ and $\psi(x)$ that are used to approximate the state $y$ and control $u$ respectively. By replacing the

$$y_h(x) = \sum_{i=1}^{n_y} \mathbf{y}\,\phi_i(x) \quad \text{and} \quad u_h(x) = \sum_{j=1}^{n_u} \mathbf{u}\,\psi_j(x) \tag{A-3}$$

into (A-1) the optimisation problem that arises is

$$\begin{aligned} \min_{\mathbf{y},\mathbf{u}} J_h(\mathbf{y},\mathbf{u}) &:= \min_{\mathbf{y},\mathbf{u}} \frac{1}{2}(\mathbf{y}-y_t)^\top M(\mathbf{y}-y_t) + \frac{\alpha}{2}\mathbf{u}^\top N\mathbf{u} \\ \text{s.t. } A\mathbf{y} &= B\mathbf{u} + F \end{aligned} \tag{A-4}$$

with

$$\begin{aligned} M_{ij} = \int_\Omega \phi_i(x)\phi_j(x)\,d\Omega \implies M \in \mathbb{R}^{n_y \times n_y} \qquad A_{ij} = \int_\Omega \nabla\phi_i(x)\nabla\phi_j(x)\,d\Omega \implies A \in \mathbb{R}^{n_y \times n_y} \\ N_{ij} = \int_\Omega \psi_i(x)\psi_j(x)\,d\Omega \implies N \in \mathbb{R}^{n_u \times n_u} \qquad B_{ij} = \int_\Omega \psi_j(x)\phi_i(x)\,d\Omega \implies B \in \mathbb{R}^{n_y \times n_u} \\ F_i = \int_\Omega f(x)\phi_i(x)\,d\Omega \quad F \in \mathbb{R}^{n_y} \qquad (y_t)_i = \int_\Omega y_t(x)\phi_i(x)\,d\Omega \quad y_t \in \mathbb{R}^{n_y} \end{aligned} \tag{A-5}$$

For the derivation of the optimality conditions in the continuous case, the Lagrangian can be introduced:

$$\mathcal{L}(y,u,p) = \frac{1}{2}\int_\Omega (y-y_t)^2\,d\Omega + \frac{\alpha}{2}\int_\Omega u^2\,d\Omega + \int_\Omega \nabla y \nabla p\,d\Omega - \int_\Omega u\,p\,d\Omega - \int_\Omega f\,p\,d\Omega \tag{A-6}$$

along with the Lagrangian multiplier or adjoint function $p$ that resides in the same space $V$ as the state. Then the KKT conditions can be expressed as follows:

$$\begin{aligned} \delta_p\mathcal{L} &= -\Delta y - u - f = 0 \quad &\Rightarrow \text{state equation,} \\ \delta_y\mathcal{L} &= -\Delta p + (y - y_t) = 0 \quad &\Rightarrow \text{adjoint equation,} \\ \delta_u\mathcal{L} &= \alpha u - p = 0 \quad &\Rightarrow \text{optimality condition for the control.} \end{aligned} \tag{A-7}$$

Following the described discretisation procedure, the Lagrangian will be equal to:

$$\mathcal{L}_h(\mathbf{y},\mathbf{p},\mathbf{u}) = \frac{1}{2}(\mathbf{y}-y_t)^\top M(\mathbf{y}-y_t) + \frac{\alpha}{2}\mathbf{u}^\top N\mathbf{u} + \mathbf{p}^\top(A\mathbf{y} - B\mathbf{u} - F) \tag{A-8}$$

and the KKT conditions, now a system of linear equations is formulated:

$$\begin{aligned} \partial_p\mathcal{L}_h = 0 \Rightarrow A\mathbf{y} - B\mathbf{u} - F = 0 \\ \partial_{\mathbf{y}}\mathcal{L}_h = 0 \Rightarrow M(\mathbf{y}-y_t) + A^\top\mathbf{p} = 0 \\ \partial_u\mathcal{L}_h = 0 \Rightarrow \alpha N u - B^\top\mathbf{p} = 0 \end{aligned} \implies \underbrace{\begin{pmatrix} A & 0 & -B \\ M & A^\top & 0 \\ 0 & -B^\top & \alpha N \end{pmatrix}}_{=:K} \underbrace{\begin{pmatrix} \mathbf{y} \\ \mathbf{p} \\ \mathbf{u} \end{pmatrix}}_{=:x} = \underbrace{\begin{pmatrix} F \\ My_t \\ 0 \end{pmatrix}}_{=:b}. \tag{A-9}$$

### A-1-1 Implementation of DDM into Reduced-Space Optimisation

There exists a natural way to take advantage of DDM in the case of the reduced-space optimisation framework. From the solution of the state equation, it is possible to compute an

explicit relationship between control $u$ and the state $y = y(u)$. As result, the state variable may be eliminated from the minimisation problem by introducing the reduced objective functional $\widehat{J}(u)$.

$$\min_{y,u} J(y,u) \implies \min_{u} \widehat{J}(u) := \min_{u} J(y(u),u) \tag{A-10}$$

For the described example, that means that after the discretisation the explicit relation between $\mathbf{u}$ and $\mathbf{y}$ is:

$$\mathbf{y} = A^{-1}\left(B\mathbf{u} + F\right) \tag{A-11}$$

So the finite dimensional optimisation problem that is equivalent to (A-10) arises from the substitution of (A-11) into (A-4):

$$\min_{\mathbf{u}} J_h(\mathbf{u}) := \min_{\mathbf{u}} \frac{1}{2}(A^{-1}\left(B\mathbf{u} + F\right) - y_t)^\top M(A^{-1}\left(B\mathbf{u} + F\right) - y_t) + \frac{\alpha}{2}\mathbf{u}^\top N\mathbf{u} \tag{A-12}$$

This reformulated optimisation problem has as decision variables the values of $\mathbf{u}$ and regardless of the selected optimiser it will require the evaluation of $J_h(\mathbf{u})$. So, the inversion of matrix $A$ is necessary at every iteration and it is equivalent with the solution of the governing PDE. At this point domain decomposition methods can be implemented to either solve the discretised PDE or formulate preconditioner matrices for Krylov solvers.

Furthermore, in this type of problems, it is common to apply adjoint-based techniques to compute the optimal response. From the formulation of the Lagrangian (A-8) and the KKT conditions (A-9), it is possible to derive the optimal control $\mathbf{u}$ from the adjoint $\mathbf{p}$:

$$\alpha N\mathbf{u} = B^\top \mathbf{p}. \tag{A-13}$$

However, from the same optimality conditions, the adjoint can be expressed with respect to the state $\mathbf{y}$ as:

$$\begin{aligned} A^\top \mathbf{p} &= -M\mathbf{y} + M\mathbf{y}_t \overset{\text{(A-11)}}{\implies} \\ \mathbf{p} = \left(A^\top\right)^{-1} M\mathbf{y}_t - \left(A^\top\right)^{-1} MA^{-1}\left(B\mathbf{u} + F\right) &\implies \\ \mathbf{p} = \left(A^\top\right)^{-1} M\left[\mathbf{y}_t - A^{-1}\left(B\mathbf{u} + F\right)\right]&. \end{aligned} \tag{A-14}$$

Finally, combining (A-13) and (A-14) the optimal control is computed via the formula:

$$\left[\alpha N + B^\top \left(A^\top\right)^{-1} MA^{-1}B\right]\mathbf{u} = B^\top \left(A^\top\right)^{-1} M\left[\mathbf{y}_t - A^{-1}F\right]. \tag{A-15}$$

To solve this linear system, the inverse of $A^\top$ must be computed which is equivalent to solving the adjoint PDE. Alternatively, domain decomposition methods can be implemented to formulate preconditioners of the global $K$ matrix in combination with Krylov solvers (A-9).

### A-1-2 Implementation of DDM into Coupled Optimality System

On the other hand, instead of reducing the space and expressing both $\mathbf{y}$ and $\mathbf{p}$ with respect to $\mathbf{u}$, there are cases where the opposite is possible. From the KKT conditions in (A-7), and

specifically from the optimality condition for the control, it is possible to eliminate the the control function by replacing it with:

$$u = \frac{1}{a}p. \tag{A-16}$$

As a result, the coupled system of PDEs that arise

$$\begin{aligned} -\Delta y &= f - \frac{1}{\alpha}p \quad \text{in } \Omega, \qquad y = 0 \text{ on } \partial\Omega, \\ -\Delta p &= y - y_t \quad \text{in } \Omega, \qquad p = 0 \text{ on } \partial\Omega \end{aligned} \tag{A-17}$$

can be decomposed, and their solution can be reconstructed from the solution of local subdomains. This implementation of the domains decomposition is similar to the use of DDM for the solution of any other PDE. The difference lies to the fact that this is a system of coupled PDEs and the decomposition characteristics should reflect that fact. The convergence rate of this method is directly related to the corresponding PDEs and the selected transmission conditions between the subdomains. Each of the subdomains will solve a local problem that is formulated as follows:

$$\begin{aligned} -\Delta y_i = f - \alpha^{-1}p_i \quad \text{in } \Omega_i, \quad y_i = 0 \quad \text{on } \partial\Omega_i \setminus \Gamma, \\ -\Delta p_i = y_i - y_t \quad \text{in } \Omega_i, \quad p_i = 0 \quad \text{on } \partial\Omega_i \setminus \Gamma \qquad i,j = 1,2,\cdots,n \\ T_1(y_i^{k+1}) + T_2(p_i^{k+1}) = T_1(y_j^k) + T_2(p_j^{k+1}) \text{ on } \Gamma_{ij} \end{aligned} \tag{A-18}$$

with $T_i$ denoting the selected interface conditions as described in (4-3).

### A-1-3 Implementation of DDM in Optimal Control Problems

It is possible to first decompose the optimal control problem before its optimisation and discretisation. By splitting the original domain into a set of subdomain, the local subproblems defined as:

$$\begin{aligned} -\Delta y_i = f + u_i \quad \text{in } \Omega_i, \qquad -\Delta y_j = f + u_j \quad \text{in } \Omega_j, \\ y_i = 0 \qquad \text{on } \partial\Omega_i \cap \partial\Omega, \qquad y_j = 0 \qquad \text{on } \partial\Omega_j \cap \partial\Omega, \\ y_i = y_j \qquad \text{on } \Gamma_{ij} := \partial\Omega_i \cap \partial\Omega_j, \\ \frac{\partial y_i}{\partial n_i} = -\frac{\partial y_j}{\partial n_j} \quad \text{on } \Gamma_{ij} := \partial\Omega_i \cap \partial\Omega_j. \end{aligned} \tag{A-19}$$

The derived solution of these problems will be equivalent to the global solution of the governing PDE from (A-1). However, these problems are over-defined in the attempt to guarantee the continuity of the whole solution. To independently solve the PDE in the subdomains, the only requirement is either the value of $y_{\Gamma_i}$ or the value of $\frac{\partial y_{\Gamma_i}}{\partial n_i}$ on the interface. As a result, the local solutions can be expressed based on the control $u_i$ and one of the aforementioned terms. For example, by setting $\frac{\partial y_{\Gamma_i}}{\partial n_i} = -g_i$ and ensuring that, at any adjacent interface of the $i$ subdomain, the expression:

$$g_j = \frac{\partial y_{\Gamma_j}}{\partial n_j} = -\frac{\partial y_{\Gamma_i}}{\partial n_i} = g_i \quad \text{on } \Gamma_{ij} := \partial\Omega_i \cap \partial\Omega_j \tag{A-20}$$

is a priori true, the whole optimisation problem can be expressed with respect to $u_i$ and $g_i$. However, the continuity of the solution is lost since there is no guarantee that the local solutions will adhere to $y_i = y_j$. This can be resolved if this term is added as a constraint on the reduced optimisation problem. In the aforementioned example, the form of the OCP will then be:

$$\begin{aligned} \min_{u,\partial y_\Gamma/\partial n} \quad & \sum_{i=1}^{n} \left( \frac{1}{2} \int_{\Omega_i} \left(y_i(\partial y_{\Gamma_i}/\partial n, u_i) - y_t\right)^2 \, dx + \frac{\alpha}{2} \int_{\Omega_i} u_i^2 \, dx \right) \\ \text{s.t.} \quad & y_i(\partial y_{\Gamma_i}/\partial n, u_i) = -y_j \left(\partial y_{\Gamma_j}/\partial n, u_j\right) \qquad \text{on } \Gamma_{ij} \quad \forall\, i, j = 1, 2, \cdots, n. \end{aligned} \tag{A-21}$$

This optimisation problem is equivalent to both A-1 and A-19.

# Appendix B

# Additional Case Studies

## B-1 Case Study of Quasi-3D State Equation

Implementing the quasi-3D model, that is described in Section 2-4, as the governing PDE it is possible to minimise the concentration of the $SO_2$ pollutant in a specified height. Selecting as reference the ground level ($z = 0\ m$) the same procedure that is specified in Section 6-1 is implemented. The only difference in this case study is the size of the domain that has increase to $8\ km \times 8\ km$ and the target concetration. The results of optimisation can be seen in Section B-1

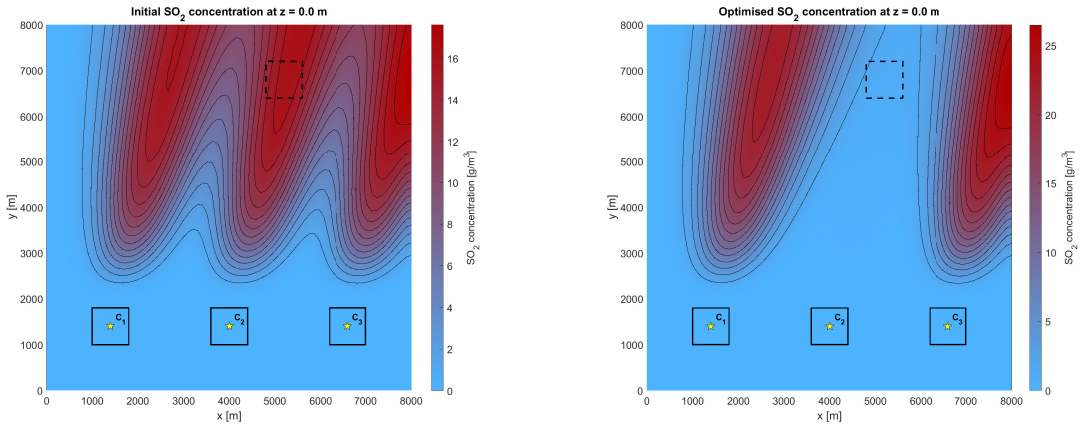



**Figure B-1:** Initial and Optimised $SO_2$ concentration in Quasi-3D.

# Appendix C

# Thoughts on the Proposed Method

## C-1 Proposed Changes on the Designed Algorithm

In the main body of this thesis, it has already been established the importance of computation of the $\mathbf{M_{cc}} \cdot \mathbf{E_c^{-1}}$ term as it is the reason for the efficiency drop of the proposed methodology. In order to tackle this issue a change to the steps of the algorithm is suggested.

The $\mathbf{M_{cc}} \cdot \mathbf{E_c^{-1}}$ is composed of the matrix $\mathbf{M_{cc}}$, which is associated with the $\nabla^2_{\mathbf{cc}}\mathcal{L}_\rho$ term of the Hessian, and the $\mathbf{E_c^{-1}}$ matrix. Several ideas that include the simplification of $\mathbf{M_{cc}}$ matrix have already been discussed in Chapter 6 and their main disadvantage is associated with the introduction of quasi-Newton methods or the limitation of the cost functionals. The $\mathbf{E_c^{-1}}$ term could not be changed since it is required for the reformulation of the inner structure of the Hessian. Nevertheless, it is mostly $\mathbf{E_c^{-1}}$ fault that the aforementioned term cannot be computed fast enough even with DDM and when it is decomposed it requires a lot processors.

To overcome this, it is suggested to change the order of the proposed steps. More specifically, in the described algorithm of Chapter 5, the proposed state variable transformation from $\mathbf{x} = \begin{bmatrix}\Delta\mathbf{c} & \Delta\mathbf{u}\end{bmatrix}^T$ into $\mathbf{x}^* = \begin{bmatrix}\mathbf{E_c} \cdot \Delta\mathbf{c} & \Delta\mathbf{u}\end{bmatrix}^T$ is performed before the reordering of the variables into interior and interface variables. By doing so, the presence of the ${\mathbf{E_c}}^{-1}$ is guaranteed as the only way to remove the disadvantageous structure of the $\mathbf{E_c^\top E_c}$ term. However, that would not be the case if those steps were performed in the reverse order.

By decomposing $\mathbf{x}$ into interior and interface variables the structure of the $\nabla^2_{\mathbf{cc}}\mathcal{L}_\rho$ is altered. The original formula :

$$\nabla^2_{\mathbf{cc}}\mathcal{L}_\rho\left(\mathbf{c}^k, \mathbf{u}^k\right) = \mathbf{M_{cc}}\left(\mathbf{c}^k, \mathbf{u}^k\right) + \rho E_{\mathbf{c}}^T E_{\mathbf{c}} \tag{C-1}$$

will be replaced by :

$$\nabla^2_{\mathbf{II}}\mathcal{L}_\rho\left(\mathbf{c}^k, \mathbf{u}^k\right) = \rho\left(E_{\mathbf{c}}^T\right)_R \cdot (E_{\mathbf{c}})_R\,. \tag{C-2}$$

This is possible since the degrees of freedom that are associated with the physical interface variables and the artificial interface variables of the localised $\mathbf{M_{cc}}$ are a priori known. Most

importantly, it has been shown that through this reordering the $(E_{\mathbf{c}})_R$ matrix will retain the inner structure of $\mathbf{E_c}$ that is crucial for the substructural domain decomposition method. Then, in order to reveal this structure into the upper left block of the reordered Hessian the term $\nabla^2_{\mathbf{II}}\mathcal{L}_\rho\left(\mathbf{c}^k, \mathbf{u}^k\right)$ will have to be multiplied via $(E_{\mathbf{c}})_R^{-1}$ instead of $E_{\mathbf{c}}^{-1}$.

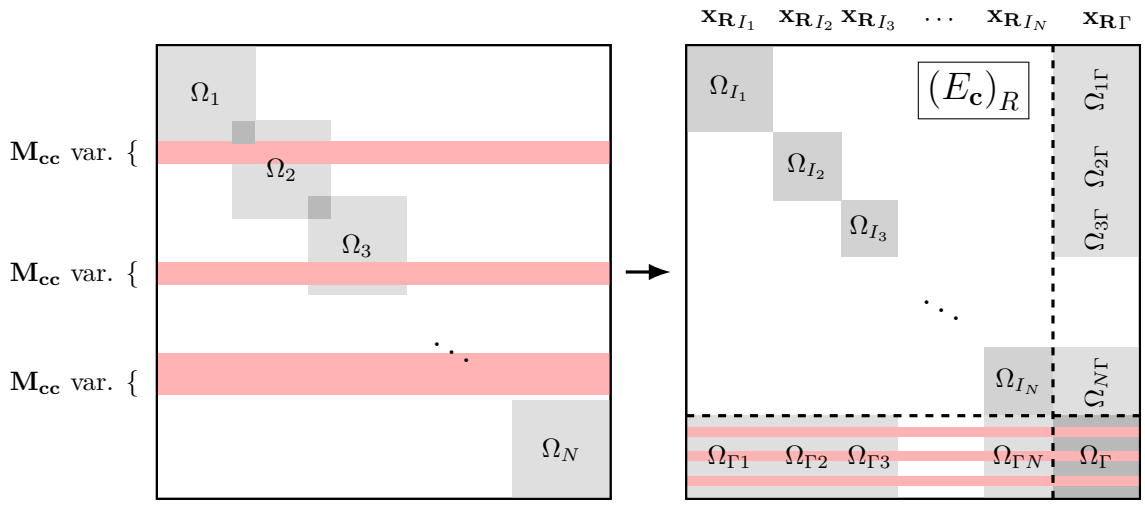
To summarise, after the discretisation of the optimisation problem and the formulation of the Hessian matrix and the gradient vector, the decision variables are reordered on the bases of interior and interface (physical and artificial) as seen in Equation C-3

$$\mathbf{x} = \begin{bmatrix} \Delta\mathbf{c} \\ \Delta\mathbf{u} \end{bmatrix} \implies \mathbf{x^R} = \begin{bmatrix} \mathbf{x_I^R} \\ \mathbf{x_\Gamma^R} \end{bmatrix}. \tag{C-3}$$

This leads to changes in the structure of the reordered Hessian ($\mathbf{H_R}$) and its upper left block that will be equal to:

$$\nabla^2_{\mathbf{II}}\mathcal{L}_\rho\left(\mathbf{c}^k, \mathbf{u}^k\right)_R = \rho\left(E_{\mathbf{c}}^T\right)_R \cdot (E_{\mathbf{c}})_R \,.$$

The reordered $(E_{\mathbf{c}})_R$ can be seen schematically in Figure C-1




**(a)** Original structure of $E_{\mathbf{c}}$ highlighting the variables associated with $\mathbf{M}_{cc}$

**(b)** Decomposed $E_{\mathbf{c}}$ with highlighted the $(E_c)_R$ upper left part

**Figure C-1:** Structure and proposed decomposition of the $E_{\mathbf{c}}$ so as to derive $(E_{\mathbf{c}})_R$.

The transformation of the state variables will now be done as described in Equation C-4.

$$\mathbf{x}^* = \begin{bmatrix} (E_{\mathbf{c}})_R \cdot \mathbf{x_I^R} \\ \mathbf{x_\Gamma^R} \end{bmatrix} = \begin{bmatrix} (E_{\mathbf{c}})_R & \varnothing \\ \varnothing & I \end{bmatrix} \begin{bmatrix} \mathbf{x_I^R} \\ \mathbf{x_\Gamma^R} \end{bmatrix} = \begin{bmatrix} (E_{\mathbf{c}})_R & \varnothing \\ \varnothing & I \end{bmatrix} \cdot \mathbf{x^R}. \tag{C-4}$$

As a result the Hessian will be equal to :

$$H_R^*\left(\mathbf{c}^k, \mathbf{u}^k\right) := \begin{bmatrix} \nabla^2_{\mathbf{II}}\mathcal{L}_\rho\left(\mathbf{c}^k, \mathbf{u}^k\right) \cdot (E_{\mathbf{c}})_R^{-1}, & \nabla^2_{\mathbf{I\Gamma}}\mathcal{L}_\rho\left(\mathbf{c}^k, \mathbf{u}^k\right) \\ \nabla^2_{\mathbf{\Gamma I}}\mathcal{L}_\rho\left(\mathbf{c}^k, \mathbf{u}^k\right) \cdot (E_{\mathbf{c}})_R^{-1}, & \nabla^2_{\mathbf{\Gamma\Gamma}}\mathcal{L}_\rho\left(\mathbf{c}^k, \mathbf{u}^k\right) \end{bmatrix}, \tag{C-5}$$

with

$$\nabla^2_{\mathbf{II}}\mathcal{L}_\rho\left(\mathbf{c}^k,\mathbf{u}^k\right)\cdot(E_\mathbf{c})_R^{-1}=\rho\,(E_\mathbf{c})_R^T\cdot(E_\mathbf{c})_R\cdot(E_\mathbf{c})_R^{-1}=\rho\left(E_\mathbf{c}^T\right)_R. \tag{C-6}$$

This part of the Hessian resulted in the exact same decomposition as seen in Figure 5-4. In addition because of the transformation of the reordered state, all the other terms of Equation C-1 that disappeared from Equation C-2 will end up in the other parts of the Hessian ($\mathbf{H_R}$), meaning $\nabla^2_{\mathbf{I\Gamma}}\mathcal{L}_\rho(\mathbf{c}^k,\mathbf{u}^k)_R$ $\nabla^2_{\mathbf{\Gamma I}}\mathcal{L}_\rho(\mathbf{c}^k,\mathbf{u}^k)_R$ and $\nabla^2_{\mathbf{\Gamma\Gamma}}\mathcal{L}_\rho(\mathbf{c}^k,\mathbf{u}^k)_R$.

The point of interest is $\nabla^2_{\mathbf{\Gamma I}}\mathcal{L}_\rho(\mathbf{c}^k,\mathbf{u}^k)_R$ that is multiplied with $(E_\mathbf{c})_R^{-1}$. In the previous methodology this term would be equal to :

$$\begin{aligned}\nabla^2_{\mathbf{uc}}\mathcal{L}_\rho\left(\mathbf{c}^k,\mathbf{u}^k\right)=\nabla^2_{\mathbf{uc}}\mathcal{J}_h\left(\mathbf{c}^k,\mathbf{u}^k\right)+\rho E_\mathbf{u}^T E_\mathbf{c}+\sum_{i=1}^{n}\left[\left(\lambda^k+\rho\mathcal{E}_h\left(\mathbf{c}^k,\mathbf{u}^k\right)\right)(E_\mathbf{uc})_i\right]+\\ +s^kQ_\mathbf{uc}+\rho Q_\mathbf{u}^T D_s^k Q_\mathbf{c}.\end{aligned} \tag{C-7}$$

In this case, due to the reordering, it is equal to:

$$\begin{aligned}\nabla^2_{\mathbf{\Gamma I}}\mathcal{L}_\rho\left(\mathbf{c}^k,\mathbf{u}^k\right)_R=&\nabla^2_{\mathbf{\Gamma I}}\mathcal{J}_h\left(\mathbf{c}^k,\mathbf{u}^k\right)+\rho E_\mathbf{u}^\top(E_\mathbf{c})_R+\sum_{i=1}^{n}\left[\left(\lambda^k+\rho\mathcal{E}_h\left(\mathbf{c}^k,\mathbf{u}^k\right)\right)[(E_\mathbf{uc})_R]_i\right]+\\ &+s^kQ_\mathbf{\Gamma I}+\rho Q_\mathbf{\Gamma}^T D_s^k Q_\mathbf{I}+(\mathbf{M_{cc}})_R+(E_\mathbf{c})_S\,.\end{aligned} \tag{C-8}$$

In this formula the term $(\mathbf{M_{cc}})_R$ corresponds to the red rows of the Figure C-1(**b**) and the $(E_\mathbf{c})_S$ denotes the rest grey rows from the bottom left region of the Figure C-1(**b**).

So in the end the total number of terms that must be multiplied via the $(E_\mathbf{c})_R^{-1}$ did not decrease in comparison to the previous algorithm. As a matter of fact due to $(E_\mathbf{c})_S$, the terms of the new expression in (C-8) are more than in the already proposed method seen in (C-7). However, an argument can be made that since $(E_\mathbf{c})_R$ is block diagonal then its inverse will be computed much faster than the original $E_\mathbf{c}^{-1}$ even with the implementation of DDM. This holds true for both the parts of the Hessian that are multiplied via $(E_\mathbf{c})_R^{-1}$ in order to produce the advantageous internal structure and foe the transformation to the original coordinates $\mathbf{x}$.

## C-2 Implementation of Krylov Solver

In the substructuring domain decomposition methodology the main advantage derives from the partition to a plethora of subdomains so as to parallelise the procedure and share the computational burden. However, this will lead to a larger and more dense Schur complement linear system. This is a problem since the interface variables increase proportionally with the number of the subdomains but the associated computational cost increases polynomially. To solve this issue, an iterative solver (Krylov) with preconditioners that are computed in parallel by the subdomains can be implemented. There is extensive literature around this topic for their implementation in the solution of PDEs and PDE-constrained optimisation problems without constraints. Many ideas can be drawn from this research and potentially increase the efficiency of the proposed methodology.

For the simplest case, the benefit in the efficiency and the scalability of the designed algorithm can be seen in Section B-1

**Table C-1:** Computational Results of Inexact Augmented Lagrangian in the General Case

| Subdomains | DOFs | $N_\Gamma$ | $\widetilde{N}_{Inter}$ | Outer Iter. | Size of Z | Average Computational Time [s] | | | | | | | Efficiency | Max. Efficiency |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | $N_{Int}$ | $N_{Schur}$ | $x_{Int}$ | $x_{Schur}$ | $Z_{Int}$ | $Z_{Schur}$ | $H^{-1}$ | | |
| 16 $(4 \times 4)$ | 186415 | 3114 | 11456 | 100 | 4 | 0.49 | 0.24 | 0.51 | 0.22 | 0.50 | 0.41 | 1.80 | 0.35 | 0.75 (64) |
| 144 $(12 \times 12)$ | 186415 | 11313 | 1216 | 100 | 4 | 0.065 | 0.296 | 0.035 | 0.282 | 0.040 | 0.257 | 1.861 | 1.00 | 1.94 (576) |
| 144 $(12 \times 12)$ | 296512 | 14380 | 1959 | 100 | 4 | 0.089 | 0.284 | 0.056 | 0.292 | 0.116 | 0.314 | 3.351 | 1.39 | 2.99 (576) |
| 144 $(12 \times 12)$ | 425684 | 17113 | 2837 | 100 | 4 | 0.139 | 0.432 | 0.087 | 0.423 | 0.094 | 0.361 | 4.726 | 1.61 | 3.02 (576) |
| 144 $(12 \times 12)$ | 751348 | 22376 | 5062 | 100 | 4 | 0.275 | 0.636 | 0.267 | 0.617 | 0.485 | 0.604 | 11.80 | 1.88 | 3.92 (576) |
| 256 $(16 \times 16)$ | 751348 | 31278 | 2809 | 100 | 4 | 0.159 | 0.772 | 0.154 | 0.762 | 0.163 | 0.871 | 11.63 | 1.92 | 3.95 (1024) |

In this table, the average time required for the solution of the Schur complement system into the Newton step ($N_{Schur}$), the computation of $\mathbf{x}$ ($\mathbf{x_{Schur}}$) and the computation of the $Z_i$ rows ($Z_{Schur}$) is presented. The same is done for the average time of the computation of the interior matrices of the subdomains denoted as $N_{Int}$, $\mathbf{x_{Int}}$ and $Z_{Int}$. The total time of the proposed algorithm is compared to the direct solution of the Hessian denoted as $H^{-1}$

## C-3 Attempted Direct Decomposition of the Hessian

From the structure of the Hessian in (5-13b), particular attention is given to the upper left block:

$$\nabla^2_{\mathbf{cc}}\mathcal{L}_\rho = \mathbf{M_{cc}} + \rho E_{\mathbf{c}}^T E_{\mathbf{c}}, \tag{C-9}$$

since it is the most dominant part of the matrix. In this case

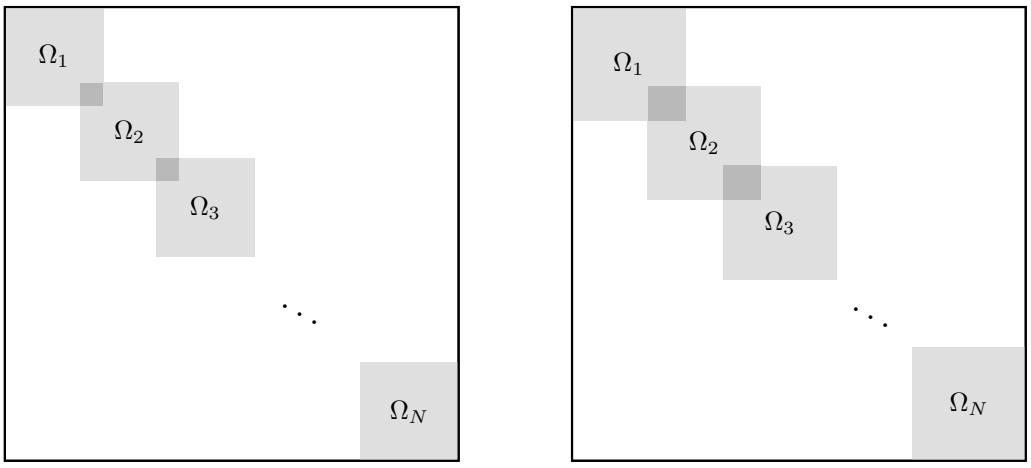
$$M_{\mathbf{cc}} := \nabla^2_{\mathbf{cc}}\mathcal{J}_h + \sum_{i=1}^{n}\left[\left(\lambda^k + \rho\mathcal{E}_h\right)(E_{\mathbf{cc}})_i\right] + s^k Q_{\mathbf{cc}} + \rho Q_{\mathbf{c}}^T D_s^k Q_{\mathbf{c}} \tag{C-10}$$

is introduced to represent all of the Hessian contributions except from $\rho E_{\mathbf{c}}^T E_{\mathbf{c}}$. The matrix $E_{\mathbf{c}}$ represents the gradient of the discretised underlying PDE with respect to the state variable. Due to the locality of the finite-element basis functions, $E_{\mathbf{c}}$ has a advantageous sparse structure that can be naturally combined with substructuring domain decomposition methods.

### C-3-1 Implementation of Additional Artificial Interface Variables

The efficiency of the substructuring method is directly related to the size of the Schur linear system and the size of the subdomains. In order to be successfully implemented, the choice of the interface variables is very important. In the solution of a PDE, this selection was dictated by the variables in the spatial or temporal interface of the domain. But, from an numerical point of view, this does not need to be upheld. Any choice that could lead to the formulation of an block diagonal matrix in the upper left part of the Hessian is capable of being decomposed via the proposed methodology. Identifying these variables in a random matrix is not an easy task and a standard interior-interface reordering based exclusively on the geometrical interface will generally not work. This can be even harder, especially if there are specific requirements with respect to the size of the subdomains and the Schur complement matrix that are associated with the computational performance of the methodology.

However $\rho E_{\mathbf{c}}^T E_{\mathbf{c}}$ and $\mathbf{M_{cc}}$ are not completely random. Due to the discretisation with finite elements methods, they exhibit a structure very similar to the $E_{\mathbf{c}}$ matrix. As a result, it was observed that it is possible to reformulate them into the desired Hessian via the selection of additional "artificial" interface variables.More specifically, for the $\mathbf{M_{cc}}$ and without any requirements or limitations on its terms (unlike before), the choice of the physical interface variables seems to lead in the desired reformulation of the matrix. Obviously, this is not the case for the $\rho E_{\mathbf{c}}^T E_{\mathbf{c}}$ elsewhere the $E_{\mathbf{c}}^{-1}$ term would not have been introduced in the main part of the thesis. However, by adding the degrees of freedom adjacent to the nodes that correspond to the actual physical interface of the domain, the reordered matrix will be block diagonal. As the total interface variables, the union of the aforementioned DoFs must be selected.




**(a)** Structure of $\mathbf{M_{cc}}$ **(b)** Structure of $\mathbf{E_c^T E_c}$

**Figure C-2:** Schematic structures of $\mathbf{M_{cc}}$ and $\mathbf{E_c^T E_c}$.

### C-3-2 Introduction of an Auxiliary State Variable

A different approach to prevent the formulation of the $\mathbf{E}_\mathbf{c}^T\mathbf{E}_\mathbf{c}$ term, could be the introduction of an auxiliary state variable denoted as $\xi$. The term :

$$\nabla^2_{\mathbf{cc}}\mathcal{L}_\rho\Delta\mathbf{c} = \mathbf{M_{cc}}\Delta\mathbf{c} + \rho E_\mathbf{c}^T E_\mathbf{c}\Delta\mathbf{c}$$

can be reformulated into

$$\nabla^2_{\mathbf{cc}}\mathcal{L}_\rho\Delta\mathbf{c} = \mathbf{M_{cc}}\Delta\mathbf{c} + \rho E_\mathbf{c}^T\xi \tag{C-11}$$

assuming that:

$$\xi = \rho E_\mathbf{c}\Delta\mathbf{c}. \tag{C-12}$$

This implementation will also affect the $\nabla^2_{\mathbf{uc}}\mathcal{L}_\rho$ term accordingly as seen:

$$\nabla^2_{\mathbf{uc}}\mathcal{L}_\rho\Delta\mathbf{c} = \mathbf{M_{uc}}\Delta\mathbf{c} + \rho E_\mathbf{u}^T\xi \tag{C-13}$$

Then combining (C-11),(C-13) and (C-12), the linear system that provides the Newton step will be reformulated as:

$$\underbrace{\begin{bmatrix} M_{\mathbf{cc}} & E_\mathbf{c}^T & M_{\mathbf{cu}} + E_\mathbf{c}^T E_\mathbf{u} \\ E_\mathbf{c} & -\rho^{-1}I & 0 \\ M_{\mathbf{uc}} & E_\mathbf{u}^T & M_{\mathbf{uu}} + E_\mathbf{u}^T E_\mathbf{u} \end{bmatrix}}_{K} \begin{bmatrix} \Delta\mathbf{c} \\ \boldsymbol{\xi} \\ \Delta\mathbf{u} \end{bmatrix} = - \begin{bmatrix} \mathbf{g_c} \\ \mathbf{0} \\ \mathbf{g_u} \end{bmatrix}. \tag{C-14}$$

The variables are subsequently reordered by subdomain. For each $\Omega_i$, define the local interior vector ${\mathbf{x}^*}_I$ and the total interface variables ${\mathbf{x}^*}_\Gamma$ as seen below:

$${\mathbf{x}^*}_{I_i} = \begin{bmatrix} \Delta\mathbf{c}_{I_i} \\ \xi_{I_i} \end{bmatrix}, \quad {\mathbf{x}^*}_\Gamma = \begin{bmatrix} \Delta\mathbf{c}_\Gamma \\ \xi_\Gamma \\ \Delta\mathbf{u} \end{bmatrix}. \tag{C-15}$$

With this ordering, the matrix $K$ takes the form:

$$K = \begin{bmatrix} K_1 & 0 & 0 & 0 & K_{1\Gamma} \\ 0 & K_2 & 0 & 0 & K_{2\Gamma} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & K_N & K_{N\Gamma} \\ K_{\Gamma 1} & K_{\Gamma 2} & \dots & K_{\Gamma N} & K_{\Gamma\Gamma} \end{bmatrix}, \tag{C-16}$$

in which the interior blocks are computed by the formula:

$$K_i = \begin{bmatrix} M_{ii} & E_{ii}^T \\ E_{ii} & -\rho^{-1}I_i \end{bmatrix}, \qquad i = 1, \dots, N_s. \tag{C-17}$$

As a result the $K$ matrix a structure similar to Figure 4-5 and the substructuring decomposition methods can be used for its solution. There were no assumption regarding the range of inequalities or spatial limitations of the non-linearities. However, the size of the system has been doubled.

# Bibliography


[1] Marilena Kampa and Elias Castanas. Human health effects of air pollution. *Environmental Pollution*, 151(2):362–367, 2008.

[2] Lu Bai, Jianzhou Wang, Xuejiao Ma, and Haiyan Lu. Air pollution forecasts: An overview. *International Journal of Environmental Research and Public Health*, 15, 4 2018.

[3] Federico Karagulian, Claudio A. Belis, Carlos F. C. Dora, Annette M. Prüss-Ustün, Sophie Bonjour, Heather Adair-Rohani, and Markus Amann. Contributions to cities' ambient particulate matter (PM): A systematic review of local source contributions at global level. *Atmospheric Environment*, 120:475–483, 2015.

[4] Alistair Hunt, Julia Ferguson, Fintan Hurley, and Alison Searl. Social costs of morbidity impacts of air pollution. Technical report, OECD, 2016.

[5] Zoi Vrontisi, Jan Abrell, Frederik Neuwahl, Bert Saveyn, and Fabian Wagner. Economic impacts of eu clean air policies assessed in a cge framework. *Environmental Science and Policy*, 55:54–64, 1 2016.

[6] William C. Skamarock, Joseph B. Klemp, Jimy Dudhia, David O. Gill, Zhiquan Liu, Judith Berner, Wei Wang, Jordan G. Powers, Michael G. Duda, Dale M. Barker, and Xiang-Yu Huang. A description of the advanced research WRF model version 4. NCAR Technical Note NCAR/TN-556+STR, National Center for Atmospheric Research, 2019.

[7] L. J. Álvarez-Vázquez, N. García-Chan, A. Martínez, and M. E. Vázquez-Méndez. An application of interactive multi-criteria optimization to air pollution control. *Optimization*, 64(6):1367–1380, 2015.

[8] Christos Vlachokostas, Charisios Achillas, Nicolas Moussiopoulos, E. Hourdakis, G. Tsilingiridis, L. Ntziachristos, G. Banias, N. Stavrakakis, and C. Sidiropoulos. Decision support system for the evaluation of urban air pollution control options: Application for particulate pollution in thessaloniki, greece. *Science of the Total Environment*, 407(23):5937–5948, 2009.

[9] Michael Hinze, René Pinnau, Michael Ulbrich, and Stefan Ulbrich. *Optimization with PDE Constraints*, volume 23 of *Mathematical Modelling: Theory and Applications*. Springer, 2009.

[10] Andrea Manzoni, Alfio Quarteroni, and Sandro Salsa. *Optimal Control of Partial Differential Equations*. Springer, 2021.

[11] Yang Zhang, Marc Bocquet, Vivien Mallet, Christian Seigneur, and Alexander Baklanov. Real-time air quality forecasting, part i: History, techniques, and current status. *Atmospheric Environment*, 60:632–655, 12 2012.

[12] Yang Zhang, Marc Bocquet, Vivien Mallet, Christian Seigneur, and Alexander Baklanov. Real-time air quality forecasting, part ii: State of the science, current research needs, and future prospects. *Atmospheric Environment*, 60:656–676, 2012.

[13] Andrea Toselli and Olof B. Widlund. *Domain Decomposition Methods: Algorithms and Theory*, volume 34 of *Springer Series in Computational Mathematics*. Springer, Berlin, Heidelberg, 2005.

[14] Andrew T. Barker and Martin Stoll. Domain decomposition in time for PDE-constrained optimization. *Computer Physics Communications*, 197:136–143, 2015.

[15] Günter Leugering. Domain decomposition in optimal control problems for partial differential equations revisited. In Oleg Imanuvilov, Günter Leugering, Roberto Triggiani, and Bing-Yu Zhang, editors, *Control Theory of Partial Differential Equations*, volume 242 of *Lecture Notes in Pure and Applied Mathematics*, pages 125–155. Chapman & Hall/CRC, 2005.

[16] Jean-David Benamou. A domain decomposition method with coupled transmission conditions for the optimal control of systems governed by elliptic partial differential equations. *SIAM Journal on Numerical Analysis*, 33(6):2401–2416, December 1996.

[17] Matthias Heinkenschloss and Hoang Nguyen. Domain decomposition preconditioners for linear–quadratic elliptic optimal control problems. Technical Report CAAM TR04-20, Department of Computational and Applied Mathematics, Rice University, Houston, Texas, November 2004.

[18] Matthias Heinkenschloss. A time-domain decomposition iterative method for the solution of distributed linear quadratic optimal control problems. *Journal of Computational and Applied Mathematics*, 173:169–198, 1 2005.

[19] Roscoe A. Bartlett, Matthias Heinkenschloss, Denis Ridzal, and Bart G. van Bloemen Waanders. Domain decomposition methods for advection dominated linear-quadratic elliptic optimal control problems. *Computer Methods in Applied Mechanics and Engineering*, 195(44–47):6428–6447, September 2006.

[20] L. S. Hou and Jangwoon Lee. A robin–robin non-overlapping domain decomposition method for an elliptic boundary control problem. *International Journal of Numerical Analysis and Modeling*, 8(3):443–465, 2011.

[21] Max D. Gunzburger, Matthias Heinkenschloss, and Hyesuk Kwon Lee. Solution of elliptic partial differential equations by an optimization-based domain decomposition method. *Applied Mathematics and Computation*, 113(2–3):111–139, 2000.

[22] Hoang Q. Nguyen. *Domain Decomposition Methods for Linear–Quadratic Elliptic Optimal Control Problems*. PhD thesis, Rice University, Houston, Texas, July 2004.

[23] Pierre-Henri Cocquet, Martin J. Gander, and Alexandre Vieira. Decompose-then-optimize: A new approach to design domain decomposition methods for optimal control problems. *Numerical Algorithms*, 101(2):1395–1423, 2026.

[24] World Health Organization. *WHO Global Air Quality Guidelines: Particulate Matter (PM2.5 and PM10), Ozone, Nitrogen Dioxide, Sulfur Dioxide and Carbon Monoxide*. World Health Organization, Geneva, 2021.

[25] Pablo Orellano, Julieta Reynoso, Nancy Quaranta, Ariel Bardach, and Agustin Ciapponi. Short-term exposure to particulate matter (pm10 and pm2.5), nitrogen dioxide (no2), and ozone (o3) and all-cause and cause-specific mortality: Systematic review and meta-analysis. *Environment International*, 142:105876, 2020.

[26] Maciej Strak et al. Long term exposure to low level air pollution and mortality in eight european cohorts within the elapse project: Pooled analysis. *The BMJ*, 374:n1904, 2021.

[27] European Union. Directive 2008/50/ec of the european parliament and of the council of 21 may 2008 on ambient air quality and cleaner air for europe, May 2008. Official Journal of the European Union, L 152, 11 June 2008, pp. 1–44.

[28] Ioannis Manisalidis, Elisavet Stavropoulou, Agathangelos Stavropoulos, and Eugenia Bezirtzoglou. Environmental and health impacts of air pollution: A review. *Frontiers in Public Health*, 8, 2 2020.

[29] Steven Herring and Pablo Huq. A review of methodology for evaluating the performance of atmospheric transport and dispersion models and suggested protocol for providing more informative results. *Fluids*, 3(1):20, 2018.

[30] Claudio Carnevale, Giovanna Finzi, Enrico Pisoni, and Marialuisa Volta. Identification of source-receptor models for secondary tropospheric pollution control. *IFAC Proceedings Volumes*, 39(1):762–767, 2006.

[31] Gregory M. Scott and Roseanne D. Diab. Forecasting air pollution potential: A synoptic climatological approach. *Journal of the Air & Waste Management Association*, 50(10):1831–1842, 2000. PMID: 28076226.

[32] Hélder Relvas, Diogo Lopes, and Jan Mateu Armengol. Empowering communities: Advancements in air quality monitoring and citizen engagement. *Urban Climate*, 60:102344, 2025.

[33] Meng Wang, Paul D. Sampson, Jianlin Hu, Michael Kleeman, Joshua P. Keller, Casey Olives, Adam A. Szpiro, Sverre Vedal, and Joel D. Kaufman. Combining land-use regression and chemical transport modeling in a spatiotemporal geostatistical model for ozone and pm2.5. *Environmental Science and Technology*, 50:5111–5118, 5 2016.

[34] Ihsane Gryech, Chaimae Asaad, Mounir Ghogho, and Abdellatif Kobbane. Applications of machine learning & internet of things for outdoor air pollution monitoring and prediction: A systematic literature review. *Engineering Applications of Artificial Intelligence*, 137, 11 2024.

[35] Jaroslaw Bernacki and Rafał Scherer. A comprehensive review of data-driven techniques for air pollution concentration forecasting. *Sensors*, 25(19):6044, 2025.

[36] Hui Liu, Guangxi Yan, Zhu Duan, and Chao Chen. Intelligent modeling strategies for forecasting air quality time series: A review. *Applied Soft Computing*, 102:106957, 2021.

[37] Sreeni Chadalavada, Oliver Faust, Massimo Salvi, Silvia Seoni, Nawin Raj, U. Raghavendra, Anjan Gudigar, Prabal Datta Barua, Filippo Molinari, and Rajendra Acharya. Application of artificial intelligence in air pollution monitoring and forecasting: A systematic review. *Environmental Modelling & Software*, 185:106312, 2025.

[38] Adil Masood and Kafeel Ahmad. A review on emerging artificial intelligence (ai) techniques for air pollution forecasting: Fundamentals, application and performance. *Journal of Cleaner Production*, 322:129072, 2021.

[39] John H. Seinfeld and Spyros N. Pandis. *Atmospheric Chemistry and Physics: From Air Pollution to Climate Change*. John Wiley & Sons, Hoboken, New Jersey, 3 edition, 2016.

[40] Sotiris Vardoulakis, Bernard E. A. Fisher, Koulis Pericleous, and Norbert Gonzalez-Flesca. Modelling air quality in street canyons: A review. *Atmospheric Environment*, 37(2):155–182, 2003.

[41] Yuqing Dai, Juncheng Qian, Jian Zhong, Xiaoming Cai, and A. Rob MacKenzie. Recent progress, bottlenecks, and outlook of multiscale air quality modelling: A review. *Atmospheric Environment: X*, 29:100435, 2026.

[42] N. S. Holmes and L. Morawska. A review of dispersion modelling and its application to the dispersion of particles: An overview of different dispersion models available. *Atmospheric Environment*, 40(30):5902–5928, 2006.

[43] J. Kukkonen, T. Olsson, D. M. Schultz, A. Baklanov, T. Klein, A. I. Miranda, A. Monteiro, M. Hirtl, V. Tarvainen, M. Boy, V. H. Peuch, A. Poupkou, I. Kioutsioukis, S. Finardi, M. Sofiev, R. Sokhi, K. E.J. Lehtinen, K. Karatzas, R. San José, M. Astitha, G. Kallos, M. Schaap, E. Reimer, H. Jakobs, and K. Eben. A review of operational, regional-scale, chemical weather forecasting models in europe. *Atmospheric Chemistry and Physics*, 12:1–87, 2012.

[44] Arthur C. Stern, Richard W. Boubel, D. Bruce Turner, and Donald L. Fox. *Fundamentals of Air Pollution*. Academic Press, London, 2 edition, 1984.

[45] John M. Stockie. The mathematics of atmospheric dispersion modeling. *SIAM Review*, 53(2):349–372, 2011.

[46] S. Pal Arya. *Air Pollution Meteorology and Dispersion*. Oxford University Press, New York, 1999.

[47] Daniel A. Vallero. *Fundamentals of Air Pollution*. Academic Press, Burlington, MA, 4 edition, 2008.

[48] Harbir Antil and Dmitriy Leykekhman. A brief introduction to pde-constrained optimization. In *Frontiers in PDE-Constrained Optimization*, volume 163 of *The IMA Volumes in Mathematics and its Applications*, pages 3–40. Springer, 2018.

[49] Néstor García-Chan, Lino J. Alvarez-Vázquez, Aurea Martínez, and Miguel E. Vázquez-Méndez. Bilevel optimal control of urban traffic-related air pollution by means of Stackelberg strategies. *Optimization and Engineering*, 23:1165–1188, 2022.

[50] Néstor García-Chan, Lino J. Alvarez-Vázquez, Aurea Martínez, and Miguel E. Vázquez-Méndez. Designing an ecologically optimized road corridor surrounding restricted urban areas: A mathematical methodology. *Mathematics and Computers in Simulation*, 190:745–759, 2021.

[51] Roland Herzog and Karl Kunisch. Algorithms for PDE-constrained optimization. *GAMM-Mitteilungen*, 33(2):163–176, 2010.

[52] Frédéric Hecht, Gontran Lance, and Emmanuel Trélat. *PDE-Constrained Optimization within FreeFEM*. SpringerBriefs on PDEs and Data Science. Springer Nature Singapore, 2026.

[53] Dimitri P. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods*. Athena Scientific, Belmont, MA, 1996.

[54] Andreas Wächter and Lorenz T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006.

[55] Dimitri P. Bertsekas. *Convex Optimization Algorithms*. Athena Scientific, Belmont, MA, 2015.

[56] Veronika Karl, Ira Neitzel, and Daniel Wachsmuth. A lagrange multiplier method for semilinear elliptic state constrained optimal control problems. *Computational Optimization and Applications*, 77(3):831–869, 2020.

[57] Alfio Quarteroni and Alberto Valli. *Numerical Approximation of Partial Differential Equations*, volume 23 of *Springer Series in Computational Mathematics*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1st ed. 1994, 2nd printing 2008 edition, 2008.

[58] Martin J Gander. Schwarz methods over the course of time *. *Electronic Transactions on Numerical Analysis*, 31:228–255, 2008.

[59] P.-L. Lions. On the schwarz alternating method. i. In *Proceedings of the 1st International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 1–42, Philadelphia, PA, 1988. SIAM.

[60] Tony F. Chan and Tarek P. Mathew. Domain decomposition algorithms. *Acta Numerica*, 3:61–143, 1994.

[61] Alfio Quarteroni and Alberto Valli. *Domain Decomposition Methods for Partial Differential Equations*. Oxford University Press, 1999.

[62] Pierre-Louis Lions. On the Schwarz alternating method iii: A variant for nonoverlapping subdomains. In Tony F. Chan, Roland Glowinski, Jacques Périaux, and Olof B. Widlund, editors, *Proceedings of the Third International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 202–223, Philadelphia, PA, 1990. SIAM.

[63] Shamsulhaq Basir and Inanc Senocak. A generalized schwarz-type non-overlapping domain decomposition method using physics-constrained neural networks, 2023.

[64] Victorita Dolean, Pierre Jolivet, and Frédéric Nataf. *An Introduction to Domain Decomposition Methods: Algorithms, Theory, and Parallel Implementation*. Other Titles in Applied Mathematics. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2015.

[65] M. Dryja and O. B. Widlund. Some domain decomposition algorithms for elliptic problems. In *Proceedings of the Conference on Iterative Methods for Large Linear Systems*, Austin, Texas, 10 1989. Academic Press.

[66] George Biros and Omar Ghattas. Parallel lagrange–newton–krylov–schur methods for PDE-constrained optimization. part II: The lagrange–newton solver and its application to optimal control of steady viscous flows. *SIAM Journal on Scientific Computing*, 27(2):714–739, 2005.

[67] Matthias Heinkenschloss and Hoang Nguyen. Neumann–neumann domain decomposition preconditioners for linear-quadratic elliptic optimal control problems. *SIAM Journal on Scientific Computing*, 28(3):1001–1028, 2006.

[68] Tyrone Rees, H. Sue Dollar, and Andrew J. Wathen. Optimal solvers for PDE-constrained optimization. *SIAM Journal on Scientific Computing*, 32(1):271–298, 2010.

[69] Volkan Akçelik, George Biros, Omar Ghattas, and Judith Hill. Parallel algorithms for PDE-constrained optimization. In *Parallel Processing for Scientific Computing*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2006.

[70] Martin J. Gander, Felix Kwok, and Bankim C. Mandal. Convergence of substructuring methods for elliptic optimal control problems. In *Domain Decomposition Methods in Science and Engineering XXIV*, volume 125 of *Lecture Notes in Computational Science and Engineering*, pages 291–300. Springer, 2018.

[71] Frédéric Hecht. *FreeFEM Documentation*, release 4.13 edition, January 2025. January 31, 2025.

[72] A. Segal. *Finite Element Methods for the Incompressible Navier–Stokes Equations*. Delft Institute of Applied Mathematics, Delft University of Technology, Delft, The Netherlands, 2017.

[73] Fred T. Tracy, Thomas C. Oppe, and Sharad Gavali. Testing parallel linear iterative solvers for finite element groundwater flow problems. Technical Report NAS-07-007, NASA Advanced Supercomputing Division, September 2007.

# Glossary

## List of Acronyms

| | |
|---|---|
| **AQF** | Air Quality Forecasting |
| **ARIMA** | Autoregressive Integrated Moving Average |
| **ASM** | Additive Schwarz Method |
| **BDD** | Balancing Domain Decomposition |
| **BDDC** | Balancing Domain Decomposition by Constraints |
| **BNN** | Balancing Neumann Neumann Domain Decomposition |
| **CTM** | Chemical Transport Model |
| **DDM** | Domain Decomposition Methods |
| **DOF** | Degree of Freedom |
| **DtN** | Dirichlet to Neumann Operator |
| **DtO** | Discretise-then-Optimise |
| **EEA** | European Environment Agency |
| **FETI** | Finite Element Tearing and Interconnect |
| **HPDDM** | High-Performance unified framework for Domain Decomposition Methods |
| **KKT** | Karush-Kuhn-Tucker Conditions |
| **KPIs** | Key Performance Indicators |
| **LUR** | Land-Use Regression |
| **NN** | Neumann-Neumann Method |
| **OCP** | Optimal Control Problem |
| **ORAS** | Optimised Restricted Additive Schwarz Method |
| **OtD** | Optimise-then-Discretise |
| **PDE** | Partial Differential Equation |
| **PETSc** | Portable, Extensible Toolkit for Scientific Computation |
| **RAS** | Restricted Additive Schwarz Method |

| | |
|---|---|
| **S-P** | Steklov Poincaré Operator |
| **SUPG** | Streamline-Upwind/Petrov–Galerkin FEM |
| **US EPA** | United States Environmental Protection Agency |
| **WHO** | World Health Organisation |

# List of Symbols

| | |
|---|---|
| $\alpha_{chem}$ | Chemical Reaction Order |
| $\epsilon_E$ | Equality tolerance |
| $\epsilon_g$ | Stationarity tolerance |
| $\epsilon_I$ | Inequality tolerance |
| $\epsilon_S$ | Step tolerance |
| $\lambda_{chem}$ | Chemical Reaction Parameter $[{}^{kg}/_{m^3}]^{1-a} \cdot s^{-1}$ |
| $\lambda_{dry/wet}$ | Dry or Wet Deposition Parameter $[{}^{1}/_{s}]$ |
| $\lambda_{grav}$ | Gravitational Settling Parameter $[{}^{m}/_{s}]$ |
| $\lambda_{phot}$ | Photolysis Parameter $[{}^{1}/_{s}]$ |
| $\nu$ | Diffusion Coefficient $[{}^{m^2}/_{s}]$ |
| $\theta$ | Wind Velocity Direction $[rads]$ |
| $\alpha$ | Regularisation parameter control penalisation |
| $\mathbf{K}$ | Diffusivity Tensor $[{}^{m^2}/_{s}]$ |
| $\mathbf{q}$ | Emission Source/Sink Term $[{}^{kg}/_{m^3 \cdot s}]$ |
| $\mathbf{r}$ | Distance from the Emission Source |
| $\mathbf{V}$ | Wind Velocity Field $[{}^{m}/_{s}]$ |
| $c$ | Concentration of the pollutant $[kg/m^3]$ |
| $c_t$ | Target Concentration $[kg/m^3]$ |
| $L_x$ | Domain Length [m] |
| $L_y$ | Domain Width [m] |
| $N_s$ | Number of Subdomains |
| $N_{DoFs}$ | Number of DoFs |
| $u_{max}$ | Maximum Emission Rate $[\mathrm{kg} \cdot \mathrm{s}^{-1}]$ |
| $u_{MinC}$ | Minimum Cumulative Emission Rate $[\mathrm{kg} \cdot \mathrm{s}^{-1}]$ |
| $u_{min}$ | Minimum Emission Rate $[\mathrm{kg} \cdot \mathrm{s}^{-1}]$ |
| $V_R$ | Reference Volume of Air Emitted Per Second $[\mathrm{m}^3]$ |

# PDE-Constrained Optimization for Air Pollutant Minimization

## Master Thesis



TU Delft Delft University of Technology

Delft Center for Systems and Control