DOMAIN DECOMPOSITION METHODS FOR VARIATIONAL INEQUALITIES

by

SAMIA RIAZ

A thesis submitted to
The University of Birmingham
for the degree of
Doctor of Philosophy (Ph.D.)

School of Mathematics
The University of Birmingham
July 2013

Abstract

Variational inequalities have found many applications in applied science. A partial list includes obstacles problems, fluid flow in porous media, management science, traffic network, and financial equilibrium problems. However, solving variational inequalities remain a challenging task as they are often subject to some set of complex constraints, for example the obstacle problem.

Domain decomposition methods provide great flexibility to handle these types of problems. In our thesis we consider a general variational inequality, its finite element formulation and its equivalence with linear and quadratic programming. We will then present a non-overlapping domain decomposition formulation for variational inequalities. In our formulation, the original problem is reformulated into two subproblems such that the first problem is a variational inequality in subdomain Ω^i and the other is a variational equality in the complementary subdomain Ω^e. This new formulation will reduce the computational cost as the variational inequality is solved on a smaller region. However one of the main challenges here is to obtain the global solution of the problem, which is to be coupled through an interface problem. Finally, we validate our method on a two dimensional obstacle problem using quadratic programming.

Some part of this thesis has been accepted to publish in proceeding from the twenty first international conference on domain decomposition methods [94].

Acknowledgements

Alhamdulillah, all praises to Allah for the strengths and His blessing in completing this thesis. I would like to express my deepest gratitude to all those people who helped me to complete this thesis. I am very thankful to my Supervisor Dr. Daniel Loghin who was very supportive and helpful right from the first day and whose continuous encouragement, guidance and patience throughout the period enabled me to complete my research work and complete this thesis. I would also like to thank the School of Mathematics to make available all the facilities needed to do research work and complete this thesis.

As far as for the matter of financial support, I am thankful to the University of Engineering and Technology Lahore, Pakistan for awarding me the scholarship for my studies in UK. I cannot forget all the love, support, encouragement and guidance given to me from all of my family members in my home country and in UK. My (late) parents who encouraged, supported and made available everything possible in their means and always wished to get the highest education and knowledge and after completing this thesis I feel proud in myself in fulfilling their wishes. My special thanks are for my daughters Rafia and Esha who suffered and missed me a lot throughout this period and to my brothers and my sisters-in-law for supporting, looking after and providing everything possible to me and my family and made it possible for me to have peace of mind and complete my thesis.

Contents

Chapter 1

Introduction

Variational inequalities have gained great importance over the last few decades both from a theoretical and a practical point of view. These problems arise in many physical, engineering and financial phenomena and have applications in various fields. For example, we could find some applications in the study of obstacle problems [27] [71], American options pricing model [117], the Stefan problem, the filtration dam problem [95], flow problems [49] etc.

A famous variational inequality problem was introduced by Antonio Signorini in 1959, named as the Signorini problem; which was essentially an equilibrium problem. These problems were solved by Fichera 1963 [44] and 1964 [45]. Some more details on the historical review of variational inequality problems can be found in [[2], pp 282-284].

Variational inequalities can often be reduced to a complementarity problem. An LCP can be solved by optimization techniques such as linear programming (LP) or quadratic programming (QP) (constrained minimization problem) problems. For example Lion in [30] and Murty in [88] have given some results to show the equivalence of LCP to LP and QP problems. The LCP seems to be a bridge between variational inequalities and optimization problems.

Domain decomposition (DD) methods are one of the most powerful tools for developing parallel algorithms for the solutions of very large scale problems in the field of science and engineering. These methods have been extensively used for the solution of elliptic problems either for partial differential equations or partial differential inequalities. DD methods are based on the rule of divide, conquer and combine. Combining the solutions for different subdomains is a challenging task. The decomposition of a domain is also very important as in different subdomains the behavior of the original problem differs and each subproblem could be solved independently depending on the behavior of the subproblem. In this thesis we apply DD methods to reformulate our variational inequality problem and then find its solution using optimization techniques for minimization problem.

1.1 Elliptic variational inequalities (EVI)

Let $\Omega \in \mathbb{R}^d,\ d = 1, 2,$ be a bounded open domain with boundary $\partial\Omega$. Consider the problem

$$\begin{cases} \mathcal{L}u \geqslant f \quad \text{in } \Omega, \\ u \geqslant \psi \quad \text{in } \Omega, \\ (\mathcal{L}u - f)(u - \psi) = 0 \quad \text{in } \Omega, \\ u = 0 \quad \text{on } \partial\Omega, \end{cases} \tag{1.1.1}$$

where $\mathcal{L}$ is an elliptic operator defined as

$$\mathcal{L}u \;=\; -\operatorname{div}(\mathbf{a} \cdot \nabla u) + \mathbf{b} \cdot \nabla u + c\, u.$$

These types of problems arise when a problem is to be solved subject to some constraints, for example, in the above problem, we have to compute the solution with the restriction $u \geqslant \psi$, and thus it gives rise to a variational inequality problem

$$a(u, v - u) \geqslant \ell(v - u), \quad \forall v \in K,$$

where $K = \{v \in V : v \geqslant \psi \text{ in } \Omega,\ v = 0 \ \text{ in } \ \Omega\}$,

The discrete form of (1.1.1) is

$$VI(L, \mathbf{f}, \Psi) : \begin{cases} L\mathbf{u} \geqslant \mathbf{f}, \\ \mathbf{u} \geqslant \Psi, \\ (L\mathbf{u} - \mathbf{f})_i(\mathbf{u} - \Psi)_i = 0 \ , 1 \leqslant i \leqslant n, \end{cases} \tag{1.1.2}$$

where $L \in \mathbb{R}^{n \times n}, \mathbf{u}, \mathbf{f}, \Psi \in \mathbb{R}^n$ and the last product is understood in an entrywise sense. Problem (1.1.2) is related to the following LCP and QP problems.

Find $\mathbf{w}, \mathbf{z} \in \mathbb{R}^n$ such that

$$LCP(\tilde{M}, \mathbf{q}) : \quad \begin{cases} \mathbf{w} - \tilde{M}\mathbf{z} = \mathbf{q}, \\ \mathbf{w} \geqslant 0, \quad \mathbf{z} \geqslant 0, \\ \mathbf{w}^T\mathbf{z} = 0. \end{cases}$$

$$QP(A, \mathbf{b}, \mathbf{c}, D) : \quad \begin{cases} \text{Minimize} \quad Q(\mathbf{x}) = \mathbf{c}^T\mathbf{x} + \dfrac{1}{2}\mathbf{x}^T D\mathbf{x}, \\ \text{subject to} \quad A\mathbf{x} \geqslant \mathbf{b}, \\ \qquad\qquad\qquad \mathbf{x} \geqslant 0. \end{cases}$$

We will show in Chapter 3 that these variational inequality problems are equivalent to linear complementarity and quadratic programming problems.

$$VI(L, \mathbf{f}, \Psi) \equiv LCP(L, L\Psi - \mathbf{f}) \equiv QP(I, \Psi, -\mathbf{f}, L).$$

As a model example for variational inequality we consider the well known Obstacle Problem.

1.1.1 Obstacle problem

The obstacle problem seeks the equilibrium position of an elastic membrane (a string in the one dimensional case) in an open bounded domain Ω with closed boundary $\partial\Omega$, which lies above an obstacle ψ under a given vertical force f. The membrane is fixed along the boundary $\partial\Omega$, so $u = 0$ on $\partial\Omega$. This can be written as the system of equations

$$\begin{cases} -\Delta u - f \geqslant 0 \quad \text{in } \Omega, \\ u - \psi \geqslant 0 \quad \text{in } \Omega, \\ (u-\psi)(-\Delta u - f) = 0 \quad \text{in } \Omega. \end{cases} \tag{1.1.3}$$

Some examples are shown in figures (1.1)-(1.3)

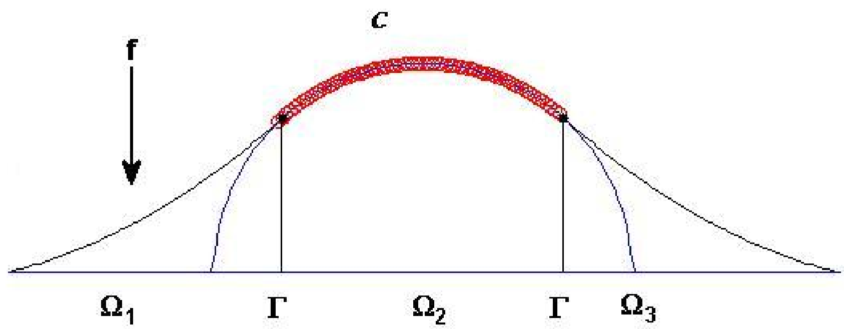


Figure 1.1: Obstacle problem in 1D

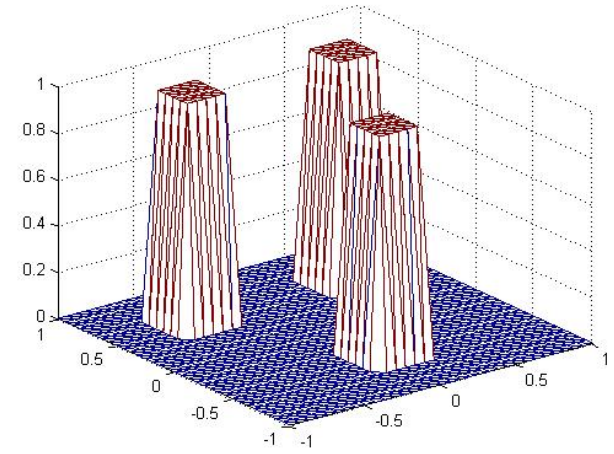


Figure 1.2: Obstacle functions

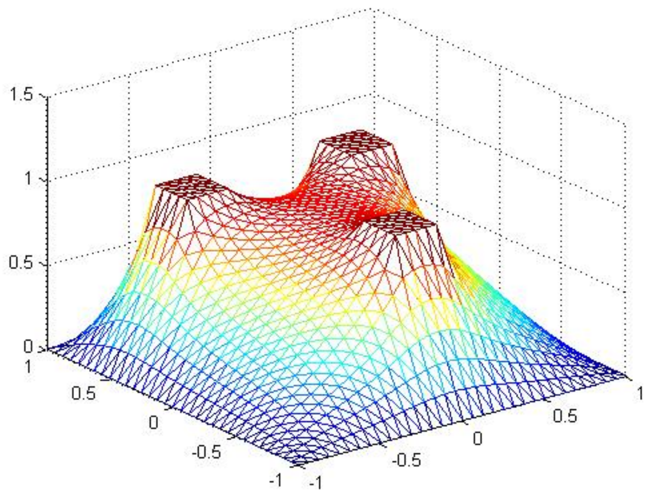


Figure 1.3: corresponding solution

Figure (1.1)-(1.3) correspond to the obstacle problem in which a membrane lies above three obstacles (cylinders), could be seen that the obstacles have support on a small area of the domain.

1.1.2 Convection diffusion problems

In Chapter 2, we give another example of variational inequalities, a convection diffusion problem, which was motivated by models in mathematical finance such as the Black-Scholes equation.

$$\begin{cases} -\mathrm{div}(\mathbf{a}\cdot\nabla u)+\mathbf{b}\cdot\nabla u \geqslant f \ \text{ in } \ \Omega, \\ u \ \geqslant \ \psi \text{ in } \ \Omega, \\ (u-\psi)(-\mathrm{div}(\mathbf{a}\cdot\nabla u)+\mathbf{b}\cdot\nabla u - f) = 0 \ \text{ in } \ \Omega, \end{cases} \tag{1.1.4}$$

$\mathbf{a} = \alpha I_d$, is a diffusion direction and $\mathbf{b}$ is a convection vector. Since these problems are non-symmetric in nature, they do not possess a minimization formulation and hence can not be solved by quadratic programming solvers. We describe a method in which by making some suitable substitution we convert the non symmetric problem into a symmetric reaction diffusion problem. This problem, can then be solved by domain decomposition method. The solution to the convection diffusion problem is obtained by converting back the variables.

1.2 Parabolic variational inequalities (PVI)

Parabolic variational inequalities usually arises in the theory of heat conduction, air conditioning heat flow, Black-Scholes, Stefan problem [40]. The dynamic obstacle problem, a kind of parabolic variational inequality, is of great importance in physics, mechanics and engineering applications. The parabolic variational inequality can be written as;

find $u(\mathbf{x},t)$ such that

$$\begin{cases} \dfrac{\partial u}{\partial t} - \mathcal{L}u \geqslant f, \\ u(\mathbf{x},t) \ \geqslant \ \psi(\mathbf{x},t) \ \text{ in } \Omega\times]0,T[, \\ (u-\psi)\left(\dfrac{\partial u}{\partial t} - \ \mathcal{L}u\right) \ = \ 0, \\ u(\mathbf{x},t) \ = \ 0 \ \text{ on } \Gamma \times [0,T], \\ u(\mathbf{x},0) \geqslant \psi(\mathbf{x},0) \quad \text{on } \Omega, \ \text{at } t=0. \end{cases} \tag{1.2.1}$$

The variational formulation for parabolic problem is; find $u(\mathbf{x},t)$ such that

$$\left(\frac{\partial u}{\partial t}, v-u\right) + a(u, v-u) \geqslant \ell(v-u), \quad \forall v \in K, \ t \in [0,T], \tag{1.2.2}$$

with initial condition

$$(u(\mathbf{x},0), v-u) \ \geqslant (\psi(\mathbf{x},0), v-u), \tag{1.2.3}$$

where

$$K = \{v | v \in H_0^1(\Omega) : \ v \geqslant \psi \text{ in } \Omega\}$$

is a non-empty convex subset of V, $u(\mathbf{x},0) \in K$ and $f \in L^2(0,T;L^2(\Omega))$.

1.2.1 Semi-discretization in space

The matrix form of the semi-discretised problem is

$$(\mathbf{v}-\mathbf{u})^T \ M\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} + (\mathbf{v}-\mathbf{u})^T L\mathbf{u} \ \geqslant \ (\mathbf{v}-\mathbf{u})^T\mathbf{f},$$

or equivalently

$$M\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{t}} + L\mathbf{u} \geqslant \mathbf{f}, \tag{1.2.4}$$

with initial condition

$$\mathbf{u}(\mathbf{x}, 0) \geqslant \Psi(\mathbf{x}, 0).$$

1.2.2 Discretization in time

To obtain the fully discrete form of the parabolic variational inequality we use two schemes, the backward Euler and Crank Nicolson. The matrix formulation for the fully discrete problem can be written as

$$\begin{cases} \widehat{A}_k \mathbf{u}^k \geqslant \widehat{\mathbf{f}}^k, \\ \mathbf{u}^k \geqslant \Psi^k, \\ (\mathbf{u}^k - \Psi^k)_i (\widehat{A}_k \mathbf{u}^k - \widehat{\mathbf{f}}^k)_i = 0 \quad 1 < k < i, \end{cases} \tag{1.2.5}$$

where $\widehat{A}_k$ and $\widehat{\mathbf{f}}_k$ depend on the time discretization scheme. These problems upon discretization yield the same type of discrete variational inequalities.

1.3 Domain decomposition Methods

Domain decompositions methods generally refers to the splitting of boundary value problem, into coupled subproblem on smaller subdomains, which are the partitioning of the original domain. In 1870 Schwarz [103] proposed an overlapping domain decomposition method to compute the numerical solutions of partial differential equations on an exotic domain combining a disc and a rectangle. The decomposition of domain is also very important as in different subdomains behavior of the original problem may differ and each subproblem can be solved on the subdomains independently and in parallel. Some more advantages are that, the domain decomposition algorithms are geometry free and can be implemented as fast iterative solver.

In this thesis we consider non-overlapping domain decomposition methods for (1.1.1) and (1.2.1). A detailed discussion is given in Chapters 5 and 6.

1.3.1 Domain decomposition Method for elliptic problem

For elliptic problems, we introduced a novel approach in which, Ω is partitioned into two subdomains and solve our variational inequalities in such a way that for each subdomain either we solve an equality problem or an inequality problem. Thus, we partition the domain Ω into two subdomains Ω^i and Ω^e with interface $\mathit{\Gamma}$ such that

$$\Omega = \Omega^i \cup \Omega^e \cup \mathit{\Gamma},$$

and in this way, we reformulate our problem into two subproblems one of which is a variational inequality in subdomain Ω^i and the other is a standard PDE in subdomain Ω^e. The domain decomposition formulation of (1.1.1) can be written as follows: find $z_1,\ z_2,\ \lambda,\ w$ such that

$$\begin{cases} \mathcal{L}z_1 = f \quad \text{in } \ \Omega^e, \\ z_1 \ = \ 0 \ \text{ on } \partial\Omega^e \setminus \mathit{\Gamma}, \\ z_1 \ = \ 0 \ \text{ on } \mathit{\Gamma}, \end{cases} \tag{1.3.1}$$

$$\begin{cases} \mathcal{S}^e\lambda = -\mathbf{n}\cdot\nabla z_1 - \mathbf{n}\cdot\nabla w, \quad \text{on } \ \mathit{\Gamma} \end{cases} \tag{1.3.2}$$

$$\begin{cases} \mathcal{L}w \geqslant f \ \text{ in } \ \Omega^i, \\ w \ \geqslant \ \psi \ \text{ in } \ \Omega^i, \\ (\mathcal{L}w - f)(w - \psi) = 0 \ \text{ in } \ \Omega^i, \\ w = 0 \ \text{ on } \partial\Omega^i \setminus \mathit{\Gamma}, \\ w \ = \ \lambda \ \text{ on } \mathit{\Gamma}, \end{cases} \tag{1.3.3}$$

$$\begin{cases} \mathcal{L}z_2 = 0 \quad \text{in } \ \Omega^e, \\ z_2 \ = \ 0 \ \text{ on } \partial\Omega^e \setminus \mathit{\Gamma}, \\ z_2 = \lambda \ \text{ on } \mathit{\Gamma}, \end{cases} \tag{1.3.4}$$

The resulting solution is then

$$u|_{\Omega^e} = z := z_1 + z_2,$$

where the solution for λ and w is approximated in an iterative manner. Note that (1.3.1) and (1.3.4) are standard linear problems which can be solved either directly or indirectly (using DDM, for example). We show in Chapter 5 that the discretization of problems (1.3.1)-(1.3.4) gives rise to an algebraic system

$$A\mathbf{u} = \begin{pmatrix} A^e_{II} & O & A^e_{I\Gamma} \\ O & A^i_{II} & A^i_{I\Gamma} \\ A^e_{\Gamma I} & A^i_{\Gamma I} & A_{\Gamma\Gamma} \end{pmatrix} \begin{pmatrix} \mathbf{u}^e_I \\ \mathbf{u}^i_I \\ \mathbf{u}_\Gamma \end{pmatrix} \geqslant \begin{pmatrix} \mathbf{f}^e_I \\ \mathbf{f}^i_I \\ \mathbf{f}_\Gamma \end{pmatrix} = \mathbf{f}. \tag{1.3.5}$$

The idea of applying the domain decomposition method here is to solve the variational inequality problem only in the domains containing the support of the obstacles. This approach leads us to a reduced variational inequality system, which could be seen as a Schur complement approach for the system (1.3.5) and is discussed in detail in Chapter 5. The reduced system for (1.3.5) is

$$\tilde{A}\tilde{\mathbf{u}} = \begin{pmatrix} A^i_{II} & A^i_{I\Gamma} \\ A^i_{\Gamma I} & S^e \end{pmatrix} \begin{pmatrix} \mathbf{u}^i_I \\ \mathbf{u}_\Gamma \end{pmatrix} \geqslant \begin{pmatrix} \mathbf{f}^i_I \\ \tilde{\mathbf{f}}_\Gamma \end{pmatrix} = \tilde{\mathbf{f}}, \tag{1.3.6}$$

where

$$S^e := A_{\Gamma\Gamma} - A^e_{\Gamma I}(A^e_{II})^{-1}A^e_{I\Gamma},$$

and the solution on Ω^e is given by

$$\mathbf{u}^e_I = (A^e_{II})^{-1}(\mathbf{f}^e_I - A^e_{I\Gamma}\mathbf{u}_\Gamma).$$

Thus, we try to achieve the task of solving a variational inequality in a smaller region rather than in whole domain with the aim to reduce the computational cost.

1.3.2 Domain decomposition method for PVI

The domain decomposition method for PVI at each each time step k, can be written as

$$\widehat{A}(\mathbf{u})^k = \begin{pmatrix} \widehat{A}^e_{II} & O & \widehat{A}^e_{I\Gamma} \\ O & \widehat{A}^i_{II} & \widehat{A}^i_{I\Gamma} \\ \widehat{A}^e_{\Gamma I} & \widehat{A}^i_{\Gamma I} & \widehat{A}_{\Gamma\Gamma} \end{pmatrix} \begin{pmatrix} (\mathbf{u}^e_I)^k \\ (\mathbf{u}^i_I)^k \\ (\mathbf{u}_\Gamma)^k \end{pmatrix} \geqslant \begin{pmatrix} (\widehat{\mathbf{f}}^e_I)^k \\ (\widehat{\mathbf{f}}^i_I)^k \\ (\widehat{\mathbf{f}}_\Gamma)^k \end{pmatrix} = \widehat{\mathbf{f}}^k, \tag{1.3.7}$$

In particular, we have

$$\widehat{A}^e_{II} = \widehat{L}^e_{II} + \widehat{M}^e_{II}, \quad \widehat{A}^i_{II} = \widehat{L}^i_{II} + \widehat{M}^i_{II},$$

$$\widehat{A}_{\Gamma\Gamma} = \widehat{L}_{\Gamma\Gamma} + \widehat{M}_{\Gamma\Gamma}, \quad \widehat{A}^e_{I\Gamma} = \widehat{L}^e_{I\Gamma} + \widehat{M}^e_{I\Gamma}, \quad \widehat{A}^i_{I\Gamma} = \widehat{L}^i_{I\Gamma} + \widehat{M}^i_{I\Gamma}.$$

Where $\widehat{L} = (1+\theta)L$ and $\widehat{M} = \dfrac{M}{\Delta t_k}$, $\quad \theta = 0$ or $1/2$, depending on the discretization scheme. By using this notation, we have the following matrix form at each time step k

$$\widehat{A}^e_{II}(\mathbf{u}^{e\{1\}}_I)^k = (\widehat{\mathbf{f}}^e_I)^k, \tag{1.3.8a}$$

$$\widehat{S}^e(\mathbf{u}_\Gamma)^k = (\widehat{\mathbf{f}}_\Gamma)^k - \widehat{A}^e_{\Gamma I}(\mathbf{u}^{e,1}_I)^k - \widehat{A}^i_{\Gamma I}(\mathbf{u}^i_I)^k, \tag{1.3.8b}$$

$$\widehat{A}^i_{II}(\mathbf{u}^i_I)^k \;\geqslant\; (\widehat{\mathbf{f}}^i_I)^k - \widehat{A}^i_{I\Gamma}(\mathbf{u}_\Gamma)^k, \tag{1.3.8c}$$

$$\widehat{A}^e_{II}(\mathbf{u}^{e\{2\}}_I)^k = -\widehat{A}^e_{I\Gamma}(\mathbf{u}_\Gamma)^k, \tag{1.3.8d}$$

subject to conditions $((\widehat{\mathbf{f}}^i_I)^k - \widehat{A}^i_{II}(\mathbf{u}^i_I)^k - \widehat{A}^i_{I\Gamma}(\mathbf{u}_\Gamma)^k)_j((\mathbf{u}^i_I)^k - (\boldsymbol{\psi}_I)^k)_j = 0$, which represent the complementarity conditions for (1.3.8c). The Partial Schur complement, $\widehat{S}^e$, is given by

$$\widehat{S}^e := \widehat{A}_{\Gamma\Gamma} - \widehat{A}^e_{\Gamma I}(\widehat{A}^e_{II})^{-1}\widehat{A}^e_{I\Gamma}.$$

The resulting solution at each time step, k, is then

$$[(\mathbf{u}_I^{e\{1\}} + \mathbf{u}_I^{e\{2\}})^k, (\mathbf{u}_I^i)^k, (\mathbf{u}_\Gamma)^k].$$

1.4 Definitions and Notations

In this section we review some basic concepts from the theory of partial differential equations. We will also discuss the theory of function spaces that have a key role in the theory of finite element methods. We will see in this chapter and the rest of the thesis that the study of finite element methods and the accuracy of the approximate solution which is obtained by this method requires classes of functions with specific differentiability and integrability properties. In this section we introduce some standard definitions, notations and simple results from the theory of functional analysis see [75] [91] [105].

We will also use the follwing notations for certain problems

PDE	partial differential equation
PDI	partial differential inequality
PVI	parabolic variational inequality
FEM	finite element method
DDM	domain decomposition method
KKT	Karush Kuhn Tucker
VI	variational inequality problem
MIN	minimization formulation
LCP	linear complementarity problem
LP	linear programming
QP	quadratic programming
RQP	reduced quadratic programming
$PRQP$	Picard reduced quadratic programming
$NRQP$	Newton reduced quadratic programming
$JFNG$	Jacobian free Newton-GMRES

1.4.1 Spaces of continuous functions

Let $\mathbb{N}$ be the set of non-negative integers. An n-tuple $\alpha = (\alpha_1, \alpha_2, ..., \alpha_n)$, is called a multi index of order $|\alpha| = \alpha_1 + \alpha_2 + ... + \alpha_n$. Let $D_j = \frac{\partial}{\partial x_j}$, we define a partial differential operator of order $|\alpha|$ as

$$D^\alpha = D_1^{\alpha_1} ... D_n^{\alpha_n} = \frac{\partial^{|\alpha|}}{\partial x_1^{\alpha_1} ... x_n^{\alpha_n}}.$$

Let Ω be a bounded open set in $\mathbb{R}^n$, $k \in \mathbb{N}$ and let $u : \Omega \to \mathbb{R}$, then the set of all continuous real valued functions with up to k continuous derivatives is denoted by $C^k(\Omega)$

$$C^k(\Omega) = \{u : D^\alpha u \text{ is continuous on } \Omega, \text{for all } \alpha \in \mathbb{N}^n \text{ with } |\alpha| \leqslant k\}.$$

$$C_k^0(\Omega) = \{u : u \in C^k(\Omega), \text{ and whose support is a bounded subset of } \Omega\}$$

$C_0^\infty(\Omega)$ is the set of all real-valued, continuous and infinitely differentiable functions defined and whose support is in Ω

$$C_0^\infty(\Omega) = \bigcap_{k \in N} C_0^k(\Omega).$$

1.4.2 L^p-Spaces

The L^p-Spaces are defined for $1 \leqslant p \leqslant \infty$, as

$$L^p(\Omega) = \{u(x) : \Omega \to \mathbb{R} : \int_\Omega |u(\mathbf{x})|^p \, d\mathbf{x} < \infty\} \text{ for} 1 \leqslant p \leqslant \infty,$$

where the integral is a Lebesgue integral. The spaces $L^p(\Omega)$ with norm

$$\|u\|_{L^p(\Omega)} = \left(\int_\Omega |u(\mathbf{x})|^p \, d\mathbf{x} \right)^{1/p}$$

are Banach spaces. The above L^p spaces can be extended to the case when $p = \infty$ i.e., the space consisting of functions u, and equipped with norm

$$\|u\|_{L^\infty(\Omega)} = \mathrm{ess}_{\sup - \{x \in \Omega\}} \sup u(\mathbf{x}).$$

A particularly important case of L^p spaces is the case when $p = 2$, the space $L^2(\Omega)$ is then a Hilbert space with norm defined by

$$\|u\|_{L^2(\Omega)} = \left(\int_\Omega |u(\mathbf{x})|^2 \, d\mathbf{x} \right)^{1/2},$$

and equipped with inner product

$$(u, v) = \int_\Omega u(\mathbf{x}) \; v(\mathbf{x}) d\Omega.$$

1.4.3 Sobolev Spaces

In this section we introduce an important class of spaces known as Sobolev spaces. These spaces act as a pillar in the theory of the finite element formulations of partial differential equations. Before introducing Sobolev spaces we define the weak derivative.

Suppose that $u \in C^k(\Omega)$, with Ω an open subset of $\mathbb{R}^n$, and let $v \in C_0^\infty(\Omega)$, then the integration by parts holds for all $v \in C_0^\infty(\Omega)$

$$\int_\Omega D^\alpha \; u(x) \; v(x) d\mathbf{x} \; = (-1)^{|\alpha|} \int_\Omega u(\mathbf{x}) D^\alpha v(\mathbf{x}) \; d\mathbf{x},$$

the boundary terms vanish because v and its derivative have support in Ω. Suppose that u is a locally integrable function on Ω i.e. $u \in L^1_{loc}(\Omega)$. Then $w^\alpha \in L^1_{loc}(\Omega)$ defined by

$$\int_\Omega w^\alpha(\mathbf{x}) \; D^\alpha v(\mathbf{x}) d\mathbf{x} = (-1)^{|\alpha|} \int_\Omega w(\mathbf{x}) D^\alpha v(\mathbf{x}) \; d\mathbf{x}, \qquad \forall \; v \in C_0^\infty(\Omega),$$

is called the weak derivative of u of order α. By using the above identities we could define the spaces of weakly differentiable functions, known as Sobolev spaces. Let α be multi-index of order $k > 0$, and let $p \in [1, \infty], D^\alpha$ denote the weak derivative of order $|\alpha|$. We define $W_k^p(\Omega)$ the set of L^p functions with derivatives of order up to and including k also in $L^p(\Omega)$

$$W_k^p(\Omega) = \{u \in L^p(\Omega) : D^\alpha u \in L^p(\Omega), |\alpha| \leqslant \ k\}.$$

The set $W_k^p(\Omega)$ is called the Sobolev space of order k, with norm given by

$$\|u\|_{W_k^p(\Omega)} = \ \|u\|_k^p = \left(\sum_{|\alpha|\leqslant k} \|D^\alpha u\|_{L^p(\Omega)}\right)^{1/p}, \qquad 1 \leqslant p < \infty.$$

When $p = \infty$ the corresponding norm is

$$\|u\|_{W_k^\infty(\Omega)} = \sum_{|\alpha|\leqslant \ k} \|D^\alpha u\|_{L^\infty(\Omega)}.$$

The case $p = 2$ has a special importance. The spaces $W_k^2(\Omega)$ corresponds to Hilbert spaces, equipped with the inner product

$$(u, v)_{W_2^k(\Omega)} = \sum_{|\alpha|\leqslant k} (D^\alpha u, D^\alpha v)_{L^2(\Omega)}$$

We will denote these spaces by $H^k(\Omega) = W_k^2(\Omega)$. Throughout this document, for the solution of partial differential equations as well as for the solution of partial differential inequalities we will frequently use the Hilbert spaces $H^1(\Omega)$ and $H^2(\Omega)$. The space $W_p^k(\Omega)$, its norm and semi-norm can be defined for $p = 2$ and $k = 0, 1$ as follows

$k = 0$

$$\|u\|_{H^0(\Omega)} \ = \ \|u\|_{L_2(\Omega)},$$

for $k = 1$

$$H^1(\Omega) \ = \ \{u \in L_2(\Omega) : D_j u \in L_2(\Omega), \ \ j = 1, ...n\},$$

$$\|u\|_{H^1(\Omega)} = \left\{ \|u\|^2_{L_2(\Omega)} + \sum_{j=1}^{n} \|D_j u\|^2_{L_2(\Omega)} \right\}^{1/2},$$

and semi-norm

$$|u|_{H^1(\Omega)} = \left\{ \sum_{j=1}^{n} \|D_j u\|^2_{L_2(\Omega)} \right\}^{1/2}.$$

We define $H^1_0(\Omega)$ as the closure of $C^\infty_0(\Omega)$ in the norm $\|.\|_{H^1(\Omega)}$. This is the Sobolev space that we will use in the next section to solve elliptic boundary value problems, with homogenous Dirichlet boundary conditions: $u = 0$ on $\partial\Omega$, the boundary of Ω. For sufficiently smooth boundary and $k = 1$ the space can be defined as

$$H^1_0(\Omega) = \{u \in H^1(\Omega) : u = 0 \text{ on } \partial\Omega\}.$$

Another space of interest is $H^{1/2}(\Gamma,)$ which is the interpolation space between $H^1(\Gamma)$ and $L^2(\Gamma)$

$$H^{1/2}(\Gamma) =: [H^1(\Gamma), L^2(\Gamma)]_{1/2},$$

The space $H^{1/2}_0(\Gamma)$ denotes the completion of $C^\infty_0(\Gamma)$ in $H^{1/2}(\Gamma)$. The space $H^{1/2}_{00}(\Gamma)$ is the subspace of $H^{1/2}_0(\Gamma)$ and can be defined as

$$H^{1/2}_{00}(\Gamma) =: [H^1_0(\Gamma), L^2(\Gamma)]_{1/2},$$

equipped with the norm $\| \cdot \|_{1/2,\Gamma}$.

1.5 Outline

An outline of the thesis is as follows:

In Chapter 2 we describe the classical formulation of partial differential equations and its equivalent formulations with variational and minimization problems. We shall also review some basic results of existence and uniqueness of solution for variational problems.

In section (2.2) of this chapter, we introduced the variational inequalities which will form the main focus of our thesis. Variational inequalities arise when a problem is to be solved with respect to some constraints. We discuss equivalence formulations, existence and uniqueness of the solution for the variational inequalities. Then we present the obstacle problem and convection diffusion problems as examples of the variational inequality problems.

In Chapter 3 we present the finite element formulations and the matrix formulation of variational problems. We also introduce generic linear complementary problem (LCP) and quadratic programming problems (QP) and show how variational inequalities can be reformulated as LCP and QP problems.

In Chapter 4, we discuss the parabolic variational inequality. Some spaces and functional setting are given to describe the PVI. We use finite element methods to obtain the semi-discretization in space and to discretize in time we use finite difference methods for backward Euler and Crank-Nicolson scheme. The LCP and QP formulation for parabolic variational inequalities are also given in this chapter.

In Chapter 5, we apply domain decomposition method to solve elliptic variational inequalities. We reformulate our problem into two subproblems: a variational inequality in some subdomains and variational equality in complementary subdomains. Thus, we try to solve a variational inequality locally, that is in a smaller region. To solve the variational inequality, we introduce the quadratic programming algorithms and then in remaining subdomains the variational equality is solved as a standard PDE. We also present algorithms to solve the obstacle problem. First algorithm is a direct procedure in which the problem is solved for PDE and PDI separately and then subdomains solutions are combine to produce the global solution. Whereas Picard reduced QP algorithms are iterative procedure in which the subproblems in subdomain Ω^i and on Γ are solved iteratively. Finally we, solve the nonlinear problem at the interface Γ, using algorithms which employ Newton's method, Newton-GMRES method and preconditioned Newton-GMRES method.

In Chapter 6, we describe the domain decomposition methods for parabolic variational inequality. At each time step k we solve two subproblems, a PDE and a variational inequality and then solution is combined together through a non linear interface problem. To implement our method we consider an example of parabolic obstacle problem, in which the obstacle changes its position with time. We apply all algorithms introduced in Chapter 5 to the parabolic variational inequality.

In Chapter 7 we present numerical results for the two dimensional obstacle problem for both elliptic and parabolic type a well as the for the obstacle problem with convection diffusion parameters as examples of variational inequalities and validate our method.

Chapter 2

Variational Formulation

In this chapter we consider general elliptic boundary value problems and their variational formulation. Variational methods provide a strong basis to study the existence theory of PDE, PDI and other applied problems. These methods have been extensively applied to solve partial differential equations see [14] [27] [64] [105] and to solve partial differential inequalities for example see [27] [50] [82]. In this chapter we consider some mathematical aspects of finite element approximation, including existence, uniqueness and convergence of the solution of partial differential equations and partial differential inequalities. We develop some theoretical tools to study the rest of the document. The basic ideas concepts and notations introduced here will be used throughout the document.

2.1 Classical formulation of elliptic problems

Let Ω be a bounded open set in $\mathbb{R}^d$ with boundary $\partial\Omega$. For a given function $f = f(\mathbf{x})$ we seek a function $u = u(\mathbf{x}) \in V \subseteq C^2(\Omega) \cap C(\overline{\Omega})$. such that

$$\mathcal{L}u = f, \tag{2.1.1}$$

where $\mathcal{L}$ is a linear operator defined to be

$$\mathcal{L}u \;=\; -\operatorname{div}(\mathbf{a}\cdot\nabla u)+\mathbf{b}\cdot\nabla u+c\,u, \tag{2.1.2}$$

where the coefficients $\mathbf{a},\mathbf{b}$ and c satisfy the following conditions

$$[\mathbf{a}]_{ij}=a_{ij}\in C^1(\overline{\Omega}), i,j=1...d, \quad \mathbf{b}\in[C(\overline{\Omega})]^d, \quad c\in C(\overline{\Omega}),$$

and

$$\xi^T\mathbf{a}\,\xi \;\geqslant\; \xi^T\lambda\,\xi, \;\; \forall\;\; \xi\in\mathbb{R}^d, \;\forall\mathbf{x}\in\mathbb{R}^d, \;c\geqslant\; 0, \tag{2.1.3}$$

with $\lambda>\; 0$ a positive constant independent of ξ and $\mathbf{x}$. The condition (2.1.3) is referred as the ellipticity condition and equation (2.1.1) with these conditions on constants is called an elliptic problem. The equation (2.1.1) is usually accompanied by one of the following boundary conditions:

(a) Dirichlet boundary condition:

$u=g$ on $\partial\Omega_D$, where g is a function defined on $\partial\Omega_D$

(b) Neumann boundary conditions:

$\dfrac{\partial u}{\partial n}=g$ on $\partial\Omega_N$, where n is the outward unit normal to $\partial\Omega_N$

(c) Robin boundary conditions:

$\dfrac{\partial u}{\partial n}+\mu u=g$ on $\mathit{\Gamma}$, where $\mu\geqslant 0$ on $\partial\Omega_R$.

In this chapter we consider the homogenous Dirichlet boundary value problem, defined as

$$\begin{cases} \mathcal{L}u=f \quad \text{in } \Omega, \\ u \;=\; 0 \;\; \text{on } \partial\Omega, \end{cases} \tag{2.1.4}$$

where $\mathcal{L}$ is a linear operator defined above.

The function $u=u(\mathbf{x})\in C^2(\Omega)\cap C(\overline{\Omega})$ satisfying (2.1.4) is called the classical solution of this problem. This solution is unique if the boundary is sufficiently smooth and depends also on $\mathbf{a}, \mathbf{b}$ and c. In general, when the situation occurs where the smoothness is not possible, and for such problem the classical theory is insufficient. This limitation can be overcome by weakening the differentiability requirement on u.

2.1.1 Variational or weak formulation

To define a weak solution of equation (2.1.4) take any v that satisfies the same essential boundary conditions as u and integrate by parts. The boundary term vanishes and we get the following formulation

$$\begin{cases} \text{find } u \ \in \ V \ \text{ such that } \ \forall v \in V, \\ a(u,v) = \ell(v), \end{cases} \tag{2.1.5}$$

where

$$a(u,v) = \int_\Omega \nabla u \cdot \mathbf{a} \cdot \nabla v \; d\Omega + \int_\Omega \mathbf{b} \cdot \nabla u \; v \; d\Omega + \int_\Omega cu \; v \; d\Omega,$$

$$\ell(v) = \int_\Omega fv \; d\Omega.$$

We assume that

$$\begin{cases} a_{ij}, b_k, c \in L^\infty(\Omega), \quad i,j,k = 1, ..., d \\ \text{and} \quad c - \frac{1}{2}\text{div}\mathbf{b} \geqslant 0, \quad \forall x \in \Omega. \end{cases} \tag{2.1.6}$$

In this formulation the second derivative term is eliminated. Thus, the solution of (2.1.5) requires less continuity than those of the classical solutions of (2.1.4). Therefore these solutions are called weak solutions and these formulations are called weak formulations. It is obvious that if u is the classical solution of (2.1.4) then it is also a weak solution of (2.1.5). As the derivative of v comes from the integration by parts, so v must be in a space where functions are more regular than L^2 functions. We will therefore choose the functions u and v from a Sobolev space. Also, since we require that u and v satisfy homogenous Dirichlet boundary conditions, it is natural to choose the space $H_0^1(\Omega)$, as we described in section (1.4.3). Thus, we could summarize above variational formulation as, find $u \in H_0^1(\Omega)$ such that

$$a(u,v) \;=\; \ell(v) \qquad \forall\; v \in H_0^1(\Omega).$$

2.1.2 The Lax-Milgram Lemma

We present a well known result the Lax-Milgram lemma, which plays an important role in the theory of the existence and uniqueness of the solution of PDE.

Theorem 2.1.7 ***Lax-Milgram:*** *Let V be a Hilbert space, ℓ a linear functional on V and $a(\cdot,\cdot)$ a bilinear form on $V \times V$ such that*
(i) $a(\cdot,\cdot)$ *is coercive, i.e., there exists a constant $\alpha > 0$ such that*

$$a(v,v) \geqslant \alpha\|v\|_V^2, \qquad \forall v \in V, \tag{2.1.8}$$

(ii) $a(\cdot,\cdot)$ *is continuous, i.e., there exists a constant $\beta > 0$ such that*

$$a(u,v) \leqslant \beta\|u\|_V\|v\|_V \qquad \forall\; u,v \;\in V,$$

(iii) $\ell(\cdot)$ *is a bounded linear functional on V, i.e., there exists a constant $\gamma > 0$ such that*

$$|\ell(v)| \;\leqslant\; \gamma\|v\|_V \;\; \forall\; v \;\in\; V.$$

Then there exist a unique function $u \in V$ such that $\forall\; v \in V$

$$a(u,v) = \ell(v).$$

Proof of this theorem can be found in [27].

Existence and uniqueness of weak solutions for elliptic problems

We apply the Lax-Milgram Lemma with $V = H_0^1(\Omega)$ and $\|\cdot\|_V = \|\cdot\|_{H^1(\Omega)}$ to prove the existence of a unique solution. The bilinear form in (2.3.2) satisfies the coercivity condition provided $c - \frac{1}{2}\text{div }\mathbf{b} \geqslant 0$ and the condition of continuity with

$$\beta = 2d \max\{\max_{1\leqslant i,j\leqslant d} \max_{x\in\overline{\Omega}} |a_{ij}(x)|, \max_{1\leqslant j\leqslant d} \max_{x\in\overline{\Omega}} |b_j(x)|, \max_{x\in\overline{\Omega}} |c(x)|\}$$

and ℓ is a bounded linear functional [[105],P-18-20], therefore there exist a unique solution of our problem.

The uniqueness is immediate: If we consider u_1 and u_2 are two solutions, then by taking $v = u_2$ (respectively $v = u_1$) relative to u_1 (respectively to u_2) in the variational formulation and by adding the result we obtain $a(u_1 - u_2, u_1 - u_2) \leqslant 0$, which is in contradiction to (2.1.8) unless $u_1 = u_2$.

2.1.3 Minimization Formulation

Let the bilinear form in (2.1.5) be symmetric, then we can define a minimization problem associated to the classical problem as follows. Let $V \subset H_0^1(\Omega)$ and define the quadratic functional $J : V \to \mathbb{R}$, by

$$J(w) = \frac{1}{2}a(w, w) - \ell(w). \tag{2.1.9}$$

Consider the following minimization problem

$$\begin{cases} \text{find } u \in V \text{ such that } \forall v \in V, \\ J(u) \leqslant J(v), \end{cases} \tag{2.1.10}$$

Lemma 2.1.11 *A function $u(\mathbf{x})$ is the solution of minimization problem* (2.1.10) *if and only if it satisfies the variational formulation* (2.1.5).

Proof

Let $u(x)$ be a function satisfying the variational problem in $H_0^1(\Omega)$, consider

$$\begin{aligned}J(u+w) &= \frac{1}{2}a(u+w,u+w) - \ell(u+w)\\ &= \frac{1}{2}a(u,u) - \ell(u) + a(u,w) - \ell(w) + \frac{1}{2}a(w,w)\end{aligned}$$

by using (2.1.5) *and coercivity of the bilinear form we get*

$$\begin{aligned}J(u+w) &= J(u) + \frac{1}{2}a(w,w)\\ &\geqslant J(u),\end{aligned}$$

hence $J(u) \leqslant J(u+w)$. for any $w \in V$.

Now suppose that $u(x)$ is the solution of minimization problem, then for any $v \in V$ and real number ε we have

$$J(u) \leqslant J(u+\varepsilon v), \quad \textit{where} \quad (u+\varepsilon v) \in V$$

Thus the differentiable function

$$g(\varepsilon) \equiv J(u+\varepsilon v) = \frac{1}{2}a(u,u) + \varepsilon a(u,v) + \frac{\varepsilon^2}{2}a(v,v) - (f,u) - \varepsilon(f,v)$$

has a minimum at $\varepsilon = 0$ and hence $g'(0) = 0$. But

$$g'(0) = a(u,v) - (f,v)$$

this implies that

$$a(u,v) = (f,v)$$

hence u is the solution of variational problem.

In this section we described the partial differential equations and their equivalent formulations such as variational formulation and minimization formulation. We also present some important results from literature such as Lax Milgram Lemma to prove the existence and uniqueness of the solution. The finite element formulation for PDE is given in Chapter 3. In next section we introduced the elliptic variational inequalities which is the main focus of our thesis.

2.2 Elliptic Variational Inequalities

Variational inequalities have found many applications in applied science. In mathematics, the variational inequality is an inequality involving a functional over a convex set. However, solving variational inequalities remain a challenging task as they are often subject to some set of complex constraints. Variational inequalities have gained importance in analysis, both from the theoretical and the practical point of view and extensive research has been done to construct and analyze the methods to solve variational inequalities see [27] [30] [31] [40] [50] [70] [82]. The mathematical theory of variational inequalities was initially developed to deal with equilibrium problems, specially the Signorini problems: in this problem the functional involved was obtained as the first variation of the potential energy [69] [101]. In this chapter we describe the partial differential inequality and its weak formulation and then introduce the well known obstacle problem as an example of a variational inequality. The obstacle problem gives rise to a variational inequality problem in the weak form due to the presence of some constraints [90]. This variational inequality problem is equivalent to a constrained minimization problem.

2.3 Classical formulation of Partial Differential Inequalities

Let $\psi : \Omega \to \mathbb{R}$ and let V be a suitable function space to be specified later. Let K be a closed and convex subset of V given by

$$K = \{v \in V : v \geqslant \psi \text{ in } \Omega,\ v = 0 \ \text{ on } \ \partial\Omega\},$$

Consider the problem find $u \in K$, such that

$$PDI : \begin{cases} \mathcal{L}u \geqslant f \ \text{ in } \Omega, \\ u \ \geqslant \ \psi \ \text{ in } \Omega, \\ (\mathcal{L}u - f)(u - \psi) = 0 \ \text{ in } \Omega, \\ u \ = \ 0 \ \text{ on } \partial\Omega, \end{cases} \tag{2.3.1}$$

where $\mathcal{L}$ is an elliptic operator defined as

$$\mathcal{L}u \ = \ -\operatorname{div}(\mathbf{a} \cdot \nabla u) + \mathbf{b} \cdot \nabla u + c\, u.$$

The function $u = u(x) \in C^2(\Omega) \cap C(\overline{\Omega})$ satisfying (2.3.1) is called the classical solution of this problem. This solution is unique if the boundary is sufficiently smooth and also depends on the coefficients $\mathbf{a}, \mathbf{b}$ and c.

2.3.1 Variational formulation

Let u be the classical solution of (2.3.1). To define a weak solution take any v that satisfies the same essential boundary conditions as of u, multiply by $v - u$ and integrate by parts, to get the formulation

$$\begin{cases} \text{find } u \ \in \ K \ \text{ such that } \ \forall v \in K, \\ a(u, v-u) \geqslant \ell(v-u), \end{cases} \tag{2.3.2}$$

where

$$a(u,v) = \int_\Omega \nabla u \cdot \mathbf{a} \cdot \nabla v \; d\Omega + \int_\Omega \mathbf{b} \cdot \nabla uv \; d\Omega + \int_\Omega cuv \; d\partial\Omega,$$

$$\ell v = \int_\Omega fvd\Omega.$$

As before, the second derivative term is eliminated. Thus, the solution of (2.3.2) requires less continuity than the classical solution of (2.3.1). It is obvious that if u is the classical solution of (2.3.1) then it is also a weak solution of (2.3.1). As in the weak formulation the derivative of v appeared in the result of integration by parts, so v must be in a space where functions are more regular than L^2. As we require that u and v satisfy zero Dirichlet boundary conditions, therefore we choose the space $H_0^1(\Omega)$, as described in section (1.4.3). Thus, we could summarize above variational formulation as, find $u \in K$ such that

$$a(u, v-u) \;\geqslant\; \ell(v-u) \qquad \forall\; v \in K.$$

2.3.2 Variational Inequalities and Lax-Milgram Lemma

In this section we review the existence and uniqueness of the problem. It is interesting to know that the existence and uniqueness of variational inequalities can also be proved by using the well known abstract result 'Lax-Milgram Lemma'.

Theorem 2.3.3 ***Lax-Milgram:*** *Let V be a Hilbert space, let $K \subset V$, is closed and convex, ℓ a linear functional on V and $a(\cdot,\cdot)$ a bilinear form on $V \times V$ such that* *(i)* $a(\cdot,\cdot)$ *is coercive with respect to* $\|v\|_V$*, i.e. there exist a constant* $C_1 > 0$ *such that*

$$a(v,v) \geqslant C_1\|v\|_V^2 \quad \forall v \in V, \tag{2.3.4}$$

(ii) $a(\cdot,\cdot)$ *is continuous, i.e. there exist a constant* $C_2 > 0$

$$a(u,v) \leqslant C_2\|u\|_V\|v\|_V \quad \forall u,v \ \in V,$$

(iii) $\ell(\cdot)$ *is bounded on* V *with respect to* $\|v\|_V$ *such that for* $C_3 \geqslant 0$

$$\ell(v) \leqslant C_3\|v\|_V \quad \forall v \in V.$$

Then the variational formulation (2.3.2) *has a unique solution.*

Proof of this theorem can be found in [71].

2.3.3 Existence and uniqueness of Solution

To apply Lax-Milgram lemma let $V = H_0^1(\Omega)$ and $\|\cdot\|_V = \|\cdot\|_{H^1(\Omega)}$. The bilinear form in (2.3.2) satisfies the coercivity condition provided $c - \frac{1}{2}\text{div}\,\mathbf{b} \geqslant 0$ and the condition of continuity with

$$\beta = 2d\max\{\max_{1\leqslant i,j\leqslant d}\max_{x\in\overline{\Omega}}|a_{ij}(x)|, \max_{1\leqslant j\leqslant d}\max_{x\in\overline{\Omega}}|b_j(x)|, \max_{x\in\overline{\Omega}}|c(x)|\},$$

and ℓ is a bounded linear functional [[105],P-18-20]. Therefore, there exist a unique solution of our problem. The uniqueness is immediate: If we consider u_1 and u_2 are two solutions, then by taking $v = u_2$ (respectively $v = u_1$) relative to u_1 (respectively to u_2) in the variational inequality and by adding the result we obtain $a(u_1 - u_2, u_1 - u_2) \leqslant 0$, which is contradiction to (2.3.4). Hence $u_1 = u_2$.

2.3.4 Minimization Formulation

Let the bilinear form in (2.3.2) is symmetric (*i.e.*, let $\mathbf{b} = 0$) and coercive. Then we can associate a minimization problem with the classical formulation of a elliptic problem. Define the quadratic functional $J : V \to \mathbb{R}, \quad V \subset H_0^1(\Omega)$ by

$$J(w) = \frac{1}{2}a(w, w) - \ell(w). \tag{2.3.5}$$

Let K be a closed and convex subset of V, then the minimization problem reads

$$\begin{cases} \text{find } u \ \in \ K \ \text{ such that } \ \forall v \in K, \\ J(u) \ \leqslant \ J(v), \end{cases} \tag{2.3.6}$$

Lemma 2.3.7 *A function $u(\mathbf{x})$ is the solution of the minimization problem* (2.3.6) *if and only if it satisfies the variational formulation* (2.3.2).

Proof

Let $u(x)$ be a function satisfying the variational problem in K then the variation about that function yields the minimization problem.

$$\begin{aligned} J(u+w) &= \frac{1}{2}a(u+w, u+w) - \ell(u+w) \\ &= \frac{1}{2}a(u,u) - \ell(u) + \frac{1}{2}a(u,w) - \ell(w) + \frac{1}{2}a(w,w) \end{aligned}$$

by using (2.3.2) *and symmetry of bilinear form we get*

$$\begin{aligned} J(u+w) &= J(u) + \frac{1}{2}a(w,w) \\ &\geqslant J(u) \end{aligned}$$

hence $J(u) \leqslant J(u+w)$.

Now suppose that $u(x)$ *is the solution of the minimization problem; then for any* $v \in V$ *and real number* ε, $g(\varepsilon) = J(u + \varepsilon(v-u))$ *is differentiable and has minimum at* $\varepsilon = 0$

$$
\begin{aligned}
J(u) &\leqslant J(u+\varepsilon(v-u)), \quad \textit{where} \quad (u+\varepsilon(v-u)) \in V, \\
&= \frac{1}{2}\, a(u,u) + \varepsilon a(u, v-u) + \frac{\varepsilon^2}{2}\, a(v-u, v-u) - \ell(u) - \varepsilon \ell(v-u) \\
&= \frac{1}{2}\, a(u,u) + \varepsilon a(u, v-u) + \frac{\varepsilon^2}{2}\, a(v-u, v-u) - \ell(u) - \varepsilon \ell(v-u) \\
&= J(u) + \varepsilon(a(u, v-u) - \ell(u, v-u)) + \frac{\varepsilon^2}{2}\, a(v-u, v-u)
\end{aligned}
$$

subtracting $J(u)$ *from both side and dividing by* ε *we get*

$$
0 \leqslant \frac{1}{\varepsilon}\,(J(u+\varepsilon(v-u)) - J(u)) = a(u, v-u) - \ell(u, v-u) + \frac{\varepsilon}{2}\, a(v-u, v-u),
$$

by taking limit $\varepsilon \to 0$

$$
a(u, v-u) \geqslant (f, v-u),
$$

hence u *is the solution of the variational inequality problem [68].*

2.4 Obstacle Problem

As a model example for a variational inequality we consider the well known obstacle problem, to illustrate how a problem can be formulated as a variational inequality. The obstacle problem is to determine the equilibrium position of a string (an elastic membrane, in two dimensional problem) in a domain Ω with closed boundary $\partial\Omega$, which lies above an obstacle ψ under the vertical force f. The classical solution u of this model problem is the vertical displacement of membrane. Since the membrane is fixed along the boundary

$\partial\Omega$, we have the boundary conditions $u = 0$ on $\partial\Omega$. This can be written as

$$\begin{cases} -\Delta u - f \geqslant 0 \ \text{ in } \ \Omega, \\ u - \psi \quad \geqslant \quad 0 \ \text{ in } \ \Omega, \\ (u-\psi)(-\Delta u - f) = 0 \ \text{ in } \ \Omega. \end{cases} \tag{2.4.1}$$

To describe the obstacle problem we consider the figure (2.1)

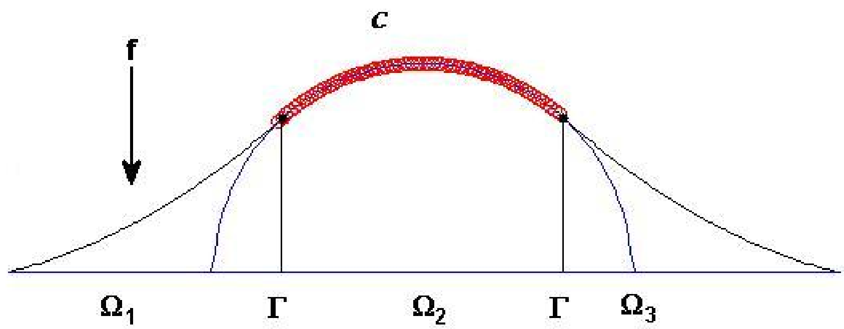


Figure 2.1: Obstacle problem in 1D

- red shaded area shows the coincidence set $\mathcal{C}$ for u and ψ,
- PDE holds in $\Omega \setminus \mathcal{C}$,
- $-\Delta u = f$, in Ω_1 and Ω_3, where f is the given vertical force,
- $u = \psi$, in Ω_2.

Weak or variational formulation

The weak formulation can be obtained by multiplying with variations $(v-u)$ and integrating by parts

$$\begin{cases} \text{find } u \;\in\; K \;\; \text{such that} \;\; \forall v \in K, \\ (\nabla u, \nabla(v-u)) \geqslant \; (f, (v-u)), \end{cases} \tag{2.4.2}$$

where $V \subseteq H_0^1(\Omega)$ and $K = \{v \in V | \, v \;\geqslant\; \psi \text{ in } \Omega\}$, with the obstacle function $\psi \,:\, \Omega \to \mathbb{R}$.

Minimization formulation

The minimization form of the obstacle problem is

$$\begin{cases} \text{find} \;\; u \in K, \text{ such that } \; \forall\, v \in K, \\ J(u) \;\leqslant J(v), \end{cases} \tag{2.4.3}$$

where

$$J(v) \;=\; \frac{1}{2}(\nabla v, \nabla v) - (f, v).$$

2.5 Convection diffusion problem

Convection-diffusion problems arise in the modeling of heat and mass transfer phenomena as well as processes of continuum mechanics. These problems are non-symmetric in nature. Various techniques could be found to derive and study the discretization of convection diffusion problem. For example, some finite difference techniques are given in [7], [58], [100], [104] etc. A finite difference scheme for one dimensional convection dominated problem is presented in [58], where an error of $O(h^2)$ is shown to hold in maximum norm. In [53], a finite difference scheme on triangular meshes is proposed, where it is shown that for self adjoint operators, this scheme can be seen to be similar to one obtained by the finite element method. A cell-centered finite difference scheme on triangular meshes is given in [115], where the error estimate for uniform triangular meshes is proven to be of $O(h^2)$ in H^1-norm. Another cell centered finite difference scheme for non symmetric problems is derived in [79] and proved an error of $O(h^{m-1}),\ 3/2 < m < 3$, which is the extension of the results presented in [42]. When the convected term in convection diffusion problems is approximated by central finite differences, the resulting scheme is second order and stable for sufficiently small mesh parameter, h. This conditional stability can be overcome by considering upwind schemes, which are obtained by adding the artificial diffusion in order to produced non oscillating solution. Several upwind methods have been developed and analyzed in [54], [65], [33], [15]. Some streamline methods are described in [63], [113], [41], [66]. Some upwind methods, to obtain unconditional stability and second order convergence are proposed in [7] [100]. Petrov-Galerkin methods, in which artificial diffusion is added in the stream line direction are presented in [16], [57], [56]. These methods are also referred to as stream line diffusion methods and are shown to be higher order accurate and have good stability properties independent of the mesh parameter. A rigorous analysis of these methods is given in [89] and [112]. [97] proposed a central upwind difference scheme for two-dimensional problem conduction and convection and proved a better rate of convergence. In a finite element framework, different strategies are considered to upwind the convective term. An example of an upwind finite element formulation can be found in [26], where the proposed scheme involved the approach of a modified weighting function for one dimensional problem for the upwind effect. An extension to two dimensional problem was studied later in [54]. A simple finite element upwind scheme is proposed in [55], where a modified quadrature rule is applied for the convective term to achieve the upwind effect. An upwind Petrov-Galerkin method is presented in [15]. In this formulation, artificial diffusion is added in the flow direction and to achieve a Petrov-Galerkin scheme, standard Galerkin functions are modified by adding stream line upwind perturbation in flow direction, resulting in a consistent weighted residual formulation.

As previously mentioned, the convection diffusion problems are non-symmetric in nature, so they do not possess a minimization formulation and hence can not be solved by QP solvers. We will present a method in which, by using an appropriate substitution, we can convert the non-symmetric convection-diffusion problem into a symmetric reaction diffusion problem. We can then apply any QP solver to find out the solution for it, which will be then used to obtain the solution of the original convection-diffusion problem.

2.5.1 Formulation of problem

An obstacle problem with convection-diffusion term is

$$\begin{cases} -\mathrm{div}(\mathbf{a} \cdot \nabla u) + \mathbf{b} \cdot \nabla u \geqslant f \quad \text{in} \quad \Omega, \\ u \ \geqslant \ \psi \text{ in } \ \Omega, \\ (u-\psi)(-\mathrm{div}(\mathbf{a} \cdot \nabla u) + \mathbf{b} \cdot \nabla u - f) = 0 \quad \text{in} \quad \Omega, \end{cases} \tag{2.5.1}$$

$\mathbf{a} = \alpha I_d$, is a diffusion vector and $\mathbf{b}$ is a convection vector.
Let $u = Ue^{px+qy}$ where $p = \dfrac{b_1}{\alpha}$ and $q = \dfrac{b_2}{\alpha}$.
By using these substitutions, the problem (2.5.2) becomes a symmetric reaction diffusion problem.

$$\begin{cases} -\alpha \Delta U + CU \geqslant \check{f} \quad \text{in} \quad \Omega, \\ U \geqslant \check{\psi} \quad \text{in} \quad \Omega, \\ (U-\check{\psi})(-\alpha \Delta U - \check{f}) = 0 \quad \text{in} \quad \Omega, \end{cases} \tag{2.5.2}$$

where

$$C = \frac{b_1^2 + b_2^2}{4\alpha}, \quad \check{\psi} = \psi e^{-\frac{b_1 x + b_2 y}{2\alpha}}, \quad \check{f} = f e^{-\frac{b_1 x + b_2 y}{2\alpha}}$$

This symmetric reaction diffusion problem can be solved by DDM, this solution is then used to determine the solution of the convection diffusion problem.

In this chapter we introduced the variational formulation of PDE and PDI and discussed its various aspects. As an example of a variational inequality problem, we presented an obstacle problem. We also show that the convection diffusion problem, which is non symmetric in nature can be converted into a symmetric reaction diffusion problem. In next chapter we present the finite element method, then obtain the matrix formulation of PDE and PDI, described in this chapter. We also introduce generic linear complementary problem (LCP) and quadratic programming problems (QP) and show how variational inequalities can be reformulated as LCP and QP problems.

Chapter 3

Finite element methods

The finite element method (FEM) is a valuable tool for approximating the solution of both PDE and PDI. It was initially proposed in [32], but was not fully appreciated at that time. Later on, this method was rediscovered by engineers, see [28]. The analysis of this method was started in [131] with the development of a number of important results in this field, in particular, an asymptotic estimate of the discretization error is derived. Nowadays the FEM has become an established method for the numerical approximation of PDE and PDI, typically used in engineering design and analysis problems.

Finite element methods are generally employed, when it is not possible to determine the analytical solution of the problem. This can be due to complex geometries, boundary conditions or boundary operators. Finite element methods convert the problem into system involving matrices by using spatial discretization. Solution to this system of matrices correspond to the solution of the original boundary value problem.

In this chapter, we describe the finite element method for both PDE, PDI together with some important results from the theory of finite element methods.

3.1 Formulation of finite element method

The formulation of finite element method is quite systematic.

The first step of formulation of the elliptic boundary value problem is to convert it into the weak formulation, which is to find $u \in V$ such that

$$a(u, v) = \ell(v) \quad \forall v \in V, \tag{3.1.1}$$

where $V \subset H_0^1(\Omega)$. The bilinear form for elliptic variational problems is symmetric, continuous, coercive and the functional ℓ is continuous.

The second step is to replace V in above equation by V_h, the finite dimensional subspace of V which consists of continuous piecewise polynomial functions with compact support in the domain. Then the discretization of weak formulation reads

$$\text{find} \;\; u_h \in V_h \;\; \text{such that} \;\; a(u_h, v_h) \;=\; \ell(v_h) \quad \forall v_h \in V_h. \tag{3.1.2}$$

The minimization problem reads

$$\begin{cases} \text{find } u_h \;\in\; K_h \;\; \text{such that} \;\; \forall v_h \in K_h, \\ J(u_h) \;\leqslant J(v_h), \end{cases}$$

where

$$J(w_h) = \frac{1}{2} a(w_h, w_h) - \ell(w_h), \quad w_h \in V_h.$$

The third step is to find the approximations for the functions u and v. These functions are typically piecewise continuous linear functions, usually polynomials, defined on the subdivision of the mesh. The functions u and v are replaced by approximations u_h, v_h, written as linear combinations of basis function $\{\phi_i(x,y)\}$.

$$u_h = \sum_{i=1}^{N} u_i \phi_i(x,y), \qquad v_h = \sum_{j=1}^{N} v_j \phi_j(x,y),$$

Then the discretized form can be written as the summation of contributions from each element

$$\sum_{i=1}^{N}\sum_{j=1}^{N} \ (v_j - u_j)\ a(\phi_i, \phi_j)\ u_i \ \geqslant \ \sum_{j=1}^{N}(v_j - u_j)\ell(\phi_j),$$

which in matrix form reads

$$(\mathbf{v} - \mathbf{u})^T L\mathbf{u} = (\mathbf{v} - \mathbf{u})^T \mathbf{f}. \tag{3.1.3}$$

3.1.1 Galerkin Orthogonality

We note that $V_h \subset \ H_0^1(\Omega)$, also bilinear form is symmetric, coercive and the functional ℓ is continuous, it follows from the Lax-Milgram lemma (3.1.2) has a unique solution $u_h \in V_h$.. Also (3.1.1) holds for any $v = v_h \in V_h \subset V$, so it can be written as

$$a(u, v_h) \ = \ \ell(v_h).$$

Subtracting (3.1.2) from the above identity we have

$$a(u - u_h, v_h) = 0 \quad \forall v_h \in V_h. \tag{3.1.4}$$

We see that the finite element error $e_h = u - u_h$ is orthogonal to V_h. The property (3.1.4) is called the Galerkin orthogonality property and plays an important role in the theory of error analysis for finite element methods [64] [105].

Lemma: 3.1.5. ***Cea's lemma:*** *Let the bilinear form $a(\cdot,\cdot)$ be continuous and coercive on $V \times V$ and let ℓ be a bounded linear functional on V. If u and u_h satisfy* (3.1.1) *and* (3.1.2) *then*

$$\|u - u_h\| \leqslant \frac{\beta}{\alpha} \inf_{v_h \in V_h} \|u - v_h\|.$$

Proof of this lemma can be found in [64].

3.2 Technical details

In this section we give some technical details to implement the finite element method to partial differential equations. The implementation of finite element methods requires some suitable approximation of solution and the domain. In the following we will show the assembly of the element matrices for the system of (3.1.3) using finite element methods.

3.2.1 Linear basis functions

To construct the finite element approximation to elliptic problem with homogenous boundary conditions we consider a subdivision of the domain $\Omega \subset \mathbb{R}^2$. Let $\mathcal{T}_h$ denote a subdivision of the domain Ω, into a set of simplices T_k such that $\bigcup_{k=1}^{N} T_k = \Omega_h$. Let us define the basis functions $\phi_i \in H^1(\Omega_h)$, corresponding to this subdivision of V_h such that

$$\phi_j(x_i, y_i) = \delta_{ij}. \tag{3.2.1}$$

The functions u_h and v_h can then be written as linear combinations of these basis functions

$$u_h(x,y) = \sum_{i=1}^{N} u_i \phi_i(x,y), \quad v_h(x,y) = \sum_{j=1}^{N} v_j \phi_j(x,y). \tag{3.2.2}$$

In the following we assume T_k to be a generic triangle with vertices 1,2 and 3. Let $u^k = u|_{T_k}$ be the restriction of u on the triangle element. We denote by $u^k(x_j, y_j) = u_j^k, \;\; j = 1,2,3.$

Where u_j^k is

$$u_j^k = a_1 + a_2 x_j + a_3 y_j \quad \text{for} \quad j = 1, 2, 3. \tag{3.2.3}$$

solving (3.2.3) we have

$$a_1 = \frac{1}{2A_k}(\alpha_1 u_1^k + \alpha_2 u_2^k + \alpha_3 u_3^k), \tag{3.2.4}$$

$$a_2 = \frac{1}{2A_k}(\beta_1 u_1^k + \beta_2 u_2^k + \beta_3 u_3^k), \tag{3.2.5}$$

$$a_3 = \frac{1}{2A_k}(\gamma_1 u_1^k + \gamma_2 u_2^k + \gamma_3 u_3^k), \tag{3.2.6}$$

where A_k is the area of the triangle T_k also

$$\alpha_i = x_j y_k - x_k y_j, \tag{3.2.7}$$

$$\beta_i = y_j - y_k, \tag{3.2.8}$$

$$\gamma_i = -(x_j - x_k), \tag{3.2.9}$$

where $i, j, k = 1, 2, 3$. Substituting (3.2.4)-(3.2.6) into (3.2.2) we get

$$u^k(x, y) = \sum_{i=1}^{3} u_i \phi_i(x, y), \tag{3.2.10}$$

where $\phi_i(x, y)$ are defined in (3.2.1), also

$$\phi_1 = \frac{1}{2A_k}(\alpha_1 + \beta_1 x + \gamma_1 y), \tag{3.2.11}$$

$$\phi_2 = \frac{1}{2A_k}(\alpha_2 + \beta_2 x + \gamma_2 y), \tag{3.2.12}$$

$$\phi_3 = \frac{1}{2A_k}(\alpha_3 + \beta_3 x + \gamma_3 y), \tag{3.2.13}$$

Transformation to the canonical triangle

Consider the transformation of each global triangle (in the (x, y) coordinates system) to the canonical triangle E, with vertex 1 at the origin(0,0), vertex 2 at (1,0), vertex 3 at (0,1) by using the following transformation

$$\begin{cases} x = a\xi + b\eta + c, \\ y = d\xi + e\eta + f. \end{cases} \tag{3.2.14}$$

Then the coordinates $x(\xi, \eta)$ and $y(\xi, \eta)$ in terms of canonical elements can be written as

$$\begin{cases} x(\xi, \eta) = (x_2 - x_1)\xi + (x_3 - x_1)\eta + x_1, \\ y(\xi, \eta) = (y_2 - y_1)\xi + (y_3 - y_1)\eta + y_1. \end{cases} \tag{3.2.15}$$

Note that the Jacobian of this affine transformation of an arbitrary T_k is given by

$$J_k = \frac{\partial(x, y)}{\partial(\xi, \eta)} = \begin{pmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_1 & y_3 - y_1 \end{pmatrix}, \tag{3.2.16}$$

with determinant

$$|J_k| = det \begin{pmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{pmatrix} = 2A_k. \tag{3.2.17}$$

Substituting (3.2.9) and (3.2.15) into (3.2.13), we obtain

$$\begin{aligned} \phi_1(\xi, \eta) =& 1 - \xi - \eta, \\ \phi_2(\xi, \eta) =& \xi, \\ \phi_1(\xi, \eta) =& \eta. \end{aligned}$$

Hence the transformation of (3.2.10) into canonical form can be written as

$$u^k(x(\xi,\eta),y(\xi,\eta)) = u^k(\xi,\eta) = \sum_{i=1}^{3} u_i\phi_i(\xi,\eta).$$

Assembly of stiffness matrices

The element stiffness matrix for elliptic problem is

$$(l^k)_{ij} = \int_K \nabla\phi_i(x,y)\nabla\phi_j(x,y)dxdy, \tag{3.2.18}$$
$$= \int_K \partial_x\phi_j(x,y)\partial_x\phi_i(x,y) + \partial_y\phi_i(x,y)\partial_y\phi_j(x,y)dxdy. \tag{3.2.19}$$

It can be shown that after transformations to the canonical element the element stiffness matrix has the form

$$L^k = \frac{1}{4A_k}\begin{pmatrix} |u_2^k - u_3^k|^2 & (u_2^k - u_3^k)(u_3^k - u_1^k) & (u_2^k - u_3^k)(u_1^k - u_2^k) \\ & |u_3^k - u_1^k|^2 & (u_3^k - u_1^k)(u_1^k - u_2^k) \\ symm. & & |u_1^k - u_2^k|^2 \end{pmatrix} \tag{3.2.20}$$

Assembly of mass matrix

The mass matrix is given by

$$(m^k)_{ij} = \int_K \phi_i(x,y)\phi_j(x,y)dxdy, \tag{3.2.21}$$

transforming to the canonical element we have

$$(m^k)_{ij} = \int_E \phi_i(\xi,\eta)\phi_j(\xi,\eta)|J_k|d\xi d\eta. \tag{3.2.22}$$

Approximation of right hand side

The right hand side in the finite element method can be approximated as

$$f(x,y) = \sum_i f_h(x_i, y_i)\phi_i(x,y),$$

then the elemental right hand side is given by

$$(f^k)_i = \int_K \sum_i f(x_i, y_i)\phi_i(x,y)\phi_j(x,y)dxdy, \tag{3.2.23}$$

transforming to the canonical elements we have

$$(f^k)_i = f(x_i, y_i)\int_E \sum_i \phi_i(\xi,\eta)\phi_j(\xi,\eta)|J_k|d\xi d\eta = (m^k)_{ij} f(x_i, y_j). \tag{3.2.24}$$

3.3 Finite Element Method for Variational Inequalities

As for variational equalities, the finite element method plays an important role in the theory of variational inequality problems. A lot of work has been done for the finite element formulation and analysis of variational inequalities see [50]. In this chapter we describe the matrix formulation of variational inequality problems. We show that variational inequalities problems can be transformed into linear complementarity problems. We will also present some results from the literature to show the equivalence of variational inequalities with optimization problems such as LP and QP problems [30], [60], [80], [88] [43]. The formulation of the finite element method for variational inequalities is based on the same procedure as that for the partial differential equations discussed in previous section.

3.3.1 Formulation of the method

Recalling the weak formulation of variational inequality from Chapter 2

$$\text{find } u \in K \text{such that} \quad a(u, v-u) \geqslant \ell(v-u) \quad \forall v \in K,$$

where $K \subset V \subset H_0^1(\Omega)$ for the case of homogenous Dirichlet boundary conditions. For the case when the bilinear form $a(\cdot,\cdot)$ is symmetric the minimization problem reads

$$\begin{cases} \text{find } u \ \in \ K \ \text{ such that } \ \forall v \in K, \\ J(u) \ \leqslant J(v), \end{cases} \tag{3.3.1}$$

where

$$J(w) = \frac{1}{2}a(w,w) - \ell(w), \quad w \subset H_0^1(\Omega). \tag{3.3.2}$$

The second step is to replace V in the above formulation by V_h, the finite dimensional subspace of V which consists of continuous piecewise polynomial functions with compact support in the domain. Then the discretization of the weak formulation reads

$$\text{find } u_h \in K_h \ \text{ such that } \ a(u_h, v_h - u_h) \ \geqslant \ \ell(v_h - u_h) \quad \forall v_h \in K_h, \tag{3.3.3}$$

where

$$K_h = \{v \in V_h | v \geqslant \psi_h \text{ in } \Omega\}.$$

The minimization problem reads

$$\begin{cases} \text{find } u_h \ \in \ K_h \ \text{ such that } \ \forall v_h \in K_h, \\ J(u_h) \ \leqslant J(v_h), \end{cases} \tag{3.3.4}$$

where

$$J(w_h) = \frac{1}{2}a(w_h, w_h) - \ell(w_h), \quad w_h \in V_h. \tag{3.3.5}$$

The functions u and v are replaced by approximations u_h, v_h, written as linear combinations of basis function $\{\phi_i(x, y)\}$.

$$u_h = \sum_{i=1}^{N} u_i \phi_i(x, y), \qquad v_h = \sum_{j=1}^{N} v_j \phi_j(x, y),$$

Then the discretized form can be written as the summation of contributions from each element

$$\sum_{i=1}^{N} \sum_{j=1}^{N} \ (v_j - u_j) \ a(\phi_i, \phi_j) \ u_i \ \geqslant \ \sum_{j=1}^{N} (v_j - u_j) \ell(\phi_j),$$

which in matrix form reads

$$(\mathbf{v} - \mathbf{u})^T L\mathbf{u} \geqslant (\mathbf{v} - \mathbf{u})^T \mathbf{f}. \tag{3.3.6}$$

3.4 Matrix formulation

The system of linear equations (3.3.6) can be expressed in matrix form as follows

$$VI(L, \mathbf{f}, \Psi) : \begin{cases} L\mathbf{u} \geqslant \mathbf{f}, \\ \mathbf{u} \ \geqslant \ \Psi, \\ (\mathbf{v} - \mathbf{u})_i (L\mathbf{u} - \mathbf{f})_i \ = 0 \quad (1 \leqslant i \leqslant n). \end{cases} \tag{3.4.1}$$

The matrix form of the minimization problem is

$$\mathrm{MIN}(L, \mathbf{f}, \Psi) : \begin{cases} \text{minimize} \quad \frac{1}{2}\mathbf{u}^T L\mathbf{u} - \mathbf{f}^T \mathbf{u}, \\ \text{subject to} \quad \mathbf{u} \ \geqslant \ \Psi, \end{cases} \tag{3.4.2}$$

where the matrix L is positive definite, symmetric and sparse and is given by

$$L_{ij} = a(\phi_i, \phi_j), \quad \mathbf{f}_i = \int_\Omega f\phi_i(x,y)d\Omega.$$

3.5 The linear complementarity problem

Complementarity problems are getting more attention in both applications and from mathematical point of views. The discretization of variational inequality problem also contains the linear complementarity problem as a special case. The Linear Complementarity Problem (LCP) is a general problem which can be posed as both linear and quadratic programs see [30] [43] [88].

Let $\tilde{M}$ be a square matrix of order n and $\mathbf{q}$ a column vector in $\mathbb{R}^n$. The LCP is defined as find $\mathbf{w} = (w_1, ..., w_n)^T$, $\mathbf{z} = (z_1, ..., z_n)^T$ satisfying

$$LCP(\tilde{M}, \mathbf{q}) : \quad \begin{cases} \mathbf{w} - \tilde{M}\mathbf{z} = \mathbf{q}, \\ \mathbf{w} \geqslant 0, \quad \mathbf{z} \geqslant 0, \\ \mathbf{w}_i \mathbf{z}_i = 0 \quad (1 \leqslant i \leqslant n). \end{cases} \tag{3.5.1}$$

3.5.1 LCP formulation of Variational Inequality

By choosing $\mathbf{v} = \mathbf{u} + \mathbf{e}_i$ and $\mathbf{v} = 2\mathbf{u} - \Psi$ the discrete variational inequality (3.4.1) can be expressed in the form of an LCP as,

$$LCP(L, L\Psi - \mathbf{f}) : \quad \begin{cases} \text{find } \mathbf{u} \ \in \mathbb{R}^n, \text{such that,} \\ (\mathbf{u} - \Psi)_i (L\mathbf{u} \ - \ \mathbf{f})_i \ = \ 0 \quad (1 \leqslant i \leqslant n), \\ \mathbf{u} - \Psi \ \geqslant \ 0, \\ L\mathbf{u} - \mathbf{f} \ \geqslant \ 0. \end{cases} \tag{3.5.2}$$

This has the form (3.5.1) if we set

$$\tilde{M}\mathbf{z} + \mathbf{q} \;=\; \mathbf{w} \;=\; L\mathbf{u} - \mathbf{f} \;=\; L(\mathbf{u} - \Psi) + (L\Psi - \mathbf{f}),$$

which implies that

$$\tilde{M} = L, \quad \mathbf{q} \;=\; L\Psi - \mathbf{f}, \quad \mathbf{z} \;=\; \mathbf{u} - \Psi,$$

and hence we conclude that

$$VI(L, \mathbf{f}, \Psi) \;\equiv\; LCP(L, L\Psi - \mathbf{f}).$$

The complementarity problem and discrete variational inequality problem have the same solution. Linear complementarity problems can be solved by Linear programming (LP) and Quadratic programming (QP). In the following two sections we will discuss how we can solve the variational inequality problem by using LP and QP programming problem by using some results and theorems from [88].

3.5.2 Linear Programming Problem

We consider now the following general linear programming problem

$$LP(A,b,c): \quad \begin{cases} \text{minimize} \quad \mathbf{c}^T\mathbf{x}, \\ \text{subject to} \\ A\mathbf{x} \geqslant \mathbf{b}, \quad \mathbf{x} \geqslant 0. \end{cases} \tag{3.5.3}$$

Here $A \in \mathbb{R}^{N\times m}$. If $\mathbf{x}$ is an optimum feasible solution of LP then there exist a dual vector $\mathbf{y} \in \mathbb{R}^m$, a primal slack vector $\mathbf{u} \in \mathbb{R}^N$ which together satisfy the LCP

$$\begin{cases} \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix} - \begin{pmatrix} 0 & -A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \begin{pmatrix} \mathbf{c} \\ -\mathbf{b} \end{pmatrix}, \\ \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix} \geqslant 0, \qquad \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \geqslant 0, \\ \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix}^T \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = 0. \end{cases} \tag{3.5.4}$$

Conversely, if $\mathbf{u}, \mathbf{v}, \mathbf{x}, \mathbf{y}$ satisfy (3.5.4) then $\mathbf{x}$ is an optimum solution of the LP problem [60] [88]. If $n = m + N$, and if we set

$$\mathbf{w} = \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix}, \quad \mathbf{z} = \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix}, \quad \tilde{M} = \begin{pmatrix} 0 & -A^T \\ A & 0 \end{pmatrix}, \quad \mathbf{q} = \begin{pmatrix} \mathbf{c} \\ -\mathbf{b} \end{pmatrix},$$

then (3.5.4) is seen to be an LCP of order n. Thus, we see that the solution of an LP problem (3.5.3) can equivalently be obtained by solving the LCP (3.5.4) [88].

3.5.3 Quadratic Programming Problem

Let us now consider a minimization problem in which a quadratic objective function is to be minimized subject to a linear inequality constraints

$$QP(A, \mathbf{b}, \mathbf{c}, D): \quad \begin{cases} \text{Minimize} \quad Q(\mathbf{x}) = \mathbf{c}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T D\mathbf{x}, \\ \text{subject to} \quad A\mathbf{x} \geqslant \mathbf{b}, \\ \qquad\qquad\quad \mathbf{x} \geqslant 0. \end{cases} \tag{3.5.5}$$

If $\overline{\mathbf{x}}$ is an optimum solution of QP, $\overline{\mathbf{x}}$ is also an optimum solution of the LP

$$\begin{cases} \text{Minimize} \quad (\mathbf{c}^T + \overline{\mathbf{x}}^T D)\mathbf{x}, \\ \text{subject to} \quad A\mathbf{x} \geqslant \mathbf{b}, \\ \qquad\qquad\qquad \mathbf{x} \geqslant 0. \end{cases} \tag{3.5.6}$$

Using the equivalence between (3.5.3) and (3.5.4) we have the following corollary

Corollary 3.5.7 *If $\overline{\mathbf{x}}$ is the optimum feasible solution of Quadratic programming problem, there exist vectors $\overline{\mathbf{y}} \in \mathbb{R}^n$ and slack vectors $\overline{\mathbf{u}} \in \mathbb{R}^n, \overline{\mathbf{v}} \in \mathbb{R}^n$ such that $\overline{\mathbf{x}}, \overline{\mathbf{y}}, \overline{\mathbf{u}}, \overline{\mathbf{v}}$ together satisfy*

$$\begin{cases} \begin{pmatrix} \overline{\mathbf{u}} \\ \overline{\mathbf{v}} \end{pmatrix} - \begin{pmatrix} D & -A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \overline{\mathbf{x}} \\ \overline{\mathbf{y}} \end{pmatrix} = \begin{pmatrix} \mathbf{c}^T \\ -\mathbf{b} \end{pmatrix}, \\ \begin{pmatrix} \overline{\mathbf{u}} \\ \overline{\mathbf{v}} \end{pmatrix} \geqslant 0, \qquad \begin{pmatrix} \overline{\mathbf{x}} \\ \overline{\mathbf{y}} \end{pmatrix} \geqslant 0, \\ \begin{pmatrix} \overline{\mathbf{u}} \\ \overline{\mathbf{v}} \end{pmatrix}^T \begin{pmatrix} \overline{\mathbf{x}} \\ \overline{\mathbf{y}} \end{pmatrix} = 0. \end{cases} \tag{3.5.8}$$

Letting

$$\mathbf{w} = \begin{pmatrix} \overline{\mathbf{u}} \\ \overline{\mathbf{v}} \end{pmatrix}, \quad \mathbf{z} = \begin{pmatrix} \overline{\mathbf{x}} \\ \overline{\mathbf{y}} \end{pmatrix},$$

and

$$\tilde{M} = \begin{pmatrix} D & -A^T \\ A & 0 \end{pmatrix}, \quad \mathbf{q} = \begin{pmatrix} \mathbf{c}^T \\ -b \end{pmatrix},$$

we can see that (3.5.8) is an LCP which can be solved by solving the quadratic programming problem (3.5.5)[30] [88].

Remark 3.5.9 (3.5.8) *is the KKT system associative with* (3.5.5) *and $\overline{\mathbf{x}}$ is a so-called*

KKT point for (3.5.5).

3.5.4 Quadratic Programming and LCPs

Theorem 3.5.10 *If* $\overline{\mathbf{x}}$ *is an optimum solution of* (3.5.5), $\overline{\mathbf{x}}$ *must be a KKT point for it. Conversely if* D *is symmetric and positive definite and* $\overline{\mathbf{x}}$ *is a KKT point of* (3.5.5), $\overline{\mathbf{x}}$ *is an optimum feasible solution of* (3.5.5).

The proof of this theorem can be found in [88].

If $\overline{\mathbf{x}}$ is an optimum solution of (3.5.5) then it must be a KKT point for it, also we have seen in corollary (3.5.7) that a quadratic programming problem can be transformed into an LCP. Solving (3.5.8) gives a KKT point for (3.5.5) and this KKT point is an optimum solution of (3.5.5), if D is symmetric and positive definite. Conversely consider a $LCP(\tilde{M}, \mathbf{q})$, where $\tilde{M}$ is symmetric and positive definite. The $LCP(\tilde{M}, \mathbf{q})$ can also be written as

$$\begin{cases} \text{minimize} \quad \mathbf{z}^T(\tilde{M}\mathbf{z} + \mathbf{q}), \\ \text{subject to} \quad \tilde{M}\mathbf{z} + \mathbf{q} \geqslant 0, \\ \qquad\qquad\qquad\qquad \mathbf{z} \geqslant 0. \end{cases} \tag{3.5.11}$$

This is a quadratic program, where the matrix $\tilde{M}$ is symmetric and positive definite. $LCP(\tilde{M}, \mathbf{q})$ has a solution $(\overline{\mathbf{w}}, \overline{\mathbf{z}})$ if the objective optimum value in this program is zero, and $\overline{\mathbf{z}}$ is a optimum solution for it. Conversely, if the optimum objective value in this quadratic programming problem is greater than zero, then the $LCP(\tilde{M}, \mathbf{q})$ has no solution. We conclude that every LCP, which is associated with a symmetric and positive definite matrix can be transformed into a quadratic program [88].

3.5.5 Quadratic programming formulation for minimization problem

The minimization problem can be formulated as the QP problem

$$QP(I,\Psi,\mathbf{f},L): \begin{cases} \text{minimize} \;\; J(\mathbf{u}) = \frac{1}{2}\mathbf{u}^T\mathbf{L}\mathbf{u} - \mathbf{f}^T\mathbf{u}, \\ \text{subject to} \;\; \mathbf{u} \;\; \geqslant \;\; \Psi. \end{cases} \tag{3.5.12}$$

If we compare it with (3.5.5) we see that

$$A = I, \quad \mathbf{b} = \Psi, \quad \mathbf{c} = -\mathbf{f}, \quad D = L.$$

Hence, we conclude that

$$\mathrm{MIN}(L,\Psi,\mathbf{f}) \equiv QP(I,\Psi,-\mathbf{f},L).$$

In this chapter we presented the matrix formulation of both the variational equality and variational inequality problems. We described its equivalence with LCP and QP problems using some results from the literature. In Chapter 5, we will apply a non overlapping domain decomposition method for variational inequality problems. We will also propose some iterative algorithms for computing the solution.

Chapter 4

Parabolic Variational Inequalities

4.1 Introduction

Parabolic variational inequalities are time dependent problems and usually arise in the theory of heat conduction, air conditioning heat flow, Stefan problem [40], [81], [96], American option problem [124] etc. The dynamic obstacle problem, a kind of parabolic variational inequality, is of great importance in physics, mechanics and engineering applications.

In this chapter we will discuss the formulation of parabolic variational inequalities and their weak formulation. We will first consider the semi-discretization, where we discretized in space using the finite element method. To convert into fully discrete problem we then discretized in time by using finite difference methods. The discretization in space give rise to an initial value problem for a system of ordinary differential inequalities. For discretization in time we will employ standard methods such as backward Euler and Crank-Nicolson. We derive its equivalence with LCP and QP to solve the parabolic variational inequalities in a similar fashion to the elliptic variational inequalities discussed in Chapter 2.2. An extension of domain decomposition method for the case of parabolic variational inequalities is given in Chapter 6.

4.2 Spaces and functional setting

We include here a brief overview of the spaces and some concepts of functional analysis used to study parabolic variational inequalities. The basic concepts are given in [40] and the new ideas are discussed in [83].

In the time dependent case we will use two Hilbert spaces instead of one space as we used for steady state case.

Let V and H be two Hilbert spaces with V a dense subset of H.

We denote the inner product for V by $((\cdot,\cdot))$ H by $(\cdot,\cdot)$, the norms for V and H are denoted by $\|\cdot\|$ and $|\cdot|$ respectively, such that

$$|v| \leqslant c\,\|v\| \qquad \forall\, v \in V.$$

Let V' is dual space of V and let H identify with its dual space, then we have

$$V \;\subset\; H \;\subset\; V'.$$

The dual space is equipped with the dual norm defined as

$$\|w\|_* = \sup_{v\in V}(w,v) \text{ with } \; \|v\| = 1.$$

For a given interval $[0,T] \;\subset\; \mathbb{R}$ and a Banach space X with norm $\|\cdot\|$, we denote the $L^p(0,T;X)$ the space of functions, $t \to f(t)$ that are measurable from $[0,t] \to X$. Let X is Hilbert space equipped with inner product $(\cdot,\cdot)_X$, the spaces $L^p(0,T;X)$ are the Banach spaces for $p \neq \infty$, with norm

$$\|f\|_{L^p(0,T;X)} \;=\; \left(\int_0^T \; \|f\|_X^p dt\right)^{1/p} \;\; < \infty,$$

and for $p = \infty$ $L^p(0,T;X)$ are the Banach spaces with norm

$$\|f\|_{L^\infty(0,T;X)} \;=\; \sup_{t\in(0,T)} \mathrm{ess}\|f\|_X \;<\infty.$$

The case $p = 2$ is the special case for which the space $L^2(0,T;X)$ is the Hilbert space with inner product

$$(f,g)_{L^2(0,T;X)} \;=\; \int_0^T \;(f,g)_X \; dt.$$

To set up our parabolic variational inequality we assume that $V = H_0^1(\Omega)$ and $H = L^2(\Omega)$, $K \subseteq V$ defined as $K = \{v \in V, v \geqslant \psi \text{ in } \Omega\}$.

4.3 Formulation of parabolic partial differential inequality

Let Ω be bounded open set in $\mathbb{R}^d$, with boundary $\partial\Omega$. We seek a function $u(\mathbf{x},t),\quad t \in [0,T],\ \mathbf{x} \in \Omega$ such that

$$\begin{cases} \dfrac{\partial u}{\partial t} - \mathcal{L}u \geqslant f, \\ u(\mathbf{x},t) \;\geqslant\; \psi(\mathbf{x},t) \;\text{ in } \Omega\times(0,T), \\ (u-\psi)\left(\dfrac{\partial u}{\partial t} - \;\mathcal{L}u\right) \;=\; 0, \\ u(\mathbf{x},t) \;=\; 0 \;\text{ on } \partial\Omega\times(0,T), \\ u(\mathbf{x},0) \geqslant \psi(\mathbf{x},0) \quad \text{in } \Omega,\ \text{at } t=0, \end{cases} \tag{4.3.1}$$

where $\mathcal{L}$ is a linear operator defined as

$$\mathcal{L}\mathbf{u} \;=\; -\operatorname{div}(\mathbf{a}\cdot\nabla\mathbf{u}) + \mathbf{b}\cdot\nabla\mathbf{u} + c\;\mathbf{u},$$

the coefficients $\mathbf{a}$, $\mathbf{b}$ and c satisfy the same conditions for ellipticity discussed in Chapter 2, in equation (2.1.3). The function $u = u(x,t)$ satisfying (4.3.1) is called the classical solution of this problem. As in the case of elliptic partial differential inequalities we discussed in Chapter 2, in many applications of parabolic variational inequalities we have to deal with the situation where the smoothness is not possible, and for such problem the classical theory is insufficient. This limitation of classical theory can be overcome by generalizing the notion of solution u of the partial differential equations with 'non-smooth' data, by weakening the differentiability requirement on u.

4.3.1 Variational formulation

Let us suppose that $u(\mathbf{x},t)$ is the classical solution of (4.3.1). To derive the variational formulation of the equation, let K be any closed convex subset of $V \subseteq H_0^1(\Omega)$ take any v, that satisfies the same essential boundary conditions as u, multiply by test function $v-u$ and integrate by parts, we have the following parabolic variational inequality: find $u(\mathbf{x},t) \in L^2(0,T;K)$ for almost all $t \in [0,T]$ with $u_t \in L^2(0,T;L^2(\Omega))$ such that

$$\left(\frac{\partial u}{\partial t}, v-u\right) + a(u,v-u) \geqslant \ell(v-u), \quad \forall v \in K,\ t \in [0,T] \tag{4.3.2}$$

with initial condition

$$(u(\mathbf{x},0), v-u) \ \geqslant (\psi(\mathbf{x},0), v-u), \tag{4.3.3}$$

where

$$K = \{v | v \in H_0^1(\Omega) :\ v \geqslant \psi \text{ in } \Omega\},$$

is a non-empty convex subset of V, $u_0 \in K$, $f \in L^2(0,T;L^2(\Omega))$ and

$$a(u,v) = \int_\Omega \nabla u \cdot \mathbf{a} \cdot \nabla v \, d\Omega + \int_\Omega \mathbf{b} \cdot \nabla uv \, d\Omega + \int_\Omega cuv \, d\partial\Omega,$$

$$\ell v = \int_{\Omega} f v \; d\Omega.$$

4.4 Existence and uniqueness of solution

The existence and uniqueness of variational inequalities can be checked by using the following theorem from [50] and [82].

Theorem 4.4.1 *Let V be a Hilbert space, ℓ is a linear functional on V, K, is a convex subset of V, $u_0 \in K$ and $a(\cdot,\cdot)$ a bilinear form such that*
(i) $a(\cdot,\cdot)$ is coercive, i.e. there exist a constant $\alpha \geqslant 0$ such that

$$a(v,v) \geqslant \alpha \|v\|_H^2 \qquad \forall v \in V, \tag{4.4.2}$$

(ii) $a(\cdot,\cdot)$ is continuous, i.e. there exist a constant $\beta > 0$ such that

$$a(u,v) \leqslant \beta \|u\|_H \|v\|_H \qquad \forall u, v \;\in V,$$

(iii) Let $f(\mathbf{x},t)$ and $u(\mathbf{x},0) = u_0$ satisfies the following conditions,

$$f \in L^2(0,T,V) \; and \; \frac{\partial f}{\partial t} \in L^2(0,T,V')$$

$$f(\mathbf{x},0) - \mathcal{L}u_0 \;\in\; K.$$

Then there exists a unique solution for parabolic variational inequality (7.4)*, also the solution satisfies*

$$u, \; \frac{\partial u}{\partial t} \;\in\; L^2(0,T;V) \cap L^\infty(0,T;H).$$

Proof of this theorem can be found in [82].

Remark 4.4.3 *The uniqueness is immediate: Consider u_1 and u_2 be two solutions, then by taking $v = u_2(t)$ (respectively $v = u_1(t)$) relative to $u_1(t)$ (respectively to $u_2(t)$) in the* *variational inequality* (4.3.2) *and add the results we obtain*

$$-(\frac{d\zeta(\mathbf{x},t)}{dt}, \zeta(\mathbf{x},t)) - a(\zeta(\mathbf{x},t), \zeta(\mathbf{x},t)) \;\geqslant\; 0$$

where $\zeta = u_1 - u_2$ *and* $\zeta(\mathbf{x}, 0) = 0$, *By using the coercivity condition* (4.4.2) *the above inequality can be written as*

$$\frac{1}{2}\frac{d}{dt}|\zeta(\mathbf{x},t)|^2 + \alpha\|\zeta(\mathbf{x},t)\|^2 \;\leqslant\; 0.$$

By using the initial condition $\zeta(0) = 0$ *we have*

$$|\zeta(\mathbf{x},t)|^2 \leqslant 0,$$

which implies that $\zeta = 0$ *and hence* $u_1 = u_2$.

4.5 Semi-discretization in space

Consider the variational formulation (4.3.2) and (4.3.3)

$$\left(\frac{\partial u}{\partial t}, v - u\right) + a(u, v - u) \geqslant \ell(v - u),$$

with initial conditions

$$(u(\mathbf{x}, 0), v - u) \;\geqslant (\psi(\mathbf{x}, 0), v - u),$$

where $V \subset H_0^1(\Omega)$ for the homogenous Dirichlet boundary conditions.

The second step is to replace V in above equation by V_h, the finite dimensional subspace of V which consist of continuous piecewise polynomial functions. Let $K_h = \{v_h \in V_h : v_h \geqslant \psi_h \text{ in } \Omega\}$. Then the semi-discretization of the weak formulation reads

$$\begin{cases} \text{find } u_h(\mathbf{x},t) \in K_h, \quad \forall t \in (0,T) \text{ such that} \\ \left(\dfrac{\partial u_h}{\partial t}, v_h - u_h\right) + a(u_h, v_h - u_h) \ \geqslant \ \ell(v_h - u_h) \quad \forall v_h \in V_h. \end{cases} \tag{4.5.1}$$

The third step is to find the approximations for the functions u_h and v_h. These functions are typically piecewise continuous linear functions, with respect to the subdivision of the domain. We choose a separation of variables ansatz to represent u_h and v_h :

$$u_h(\mathbf{x},t) = \sum_{i=1}^{N} u_i(t)\phi_i(\mathbf{x}), \quad v_h(\mathbf{x},t) = \sum_{j=1}^{N} v_j(t)\phi_j(\mathbf{x}). \tag{4.5.2}$$

Additionally, we also have initial condition

$$(u_h(\mathbf{x},0), v_h - u_h) \ \geqslant (\psi_h(\mathbf{x},0), v_h - u_h).$$

Then the semi-discretized form can be written as

$$\sum_{i=1}^{N}\sum_{j=1}^{N} \ (v_j - u_i)\frac{\partial u_i}{\partial t}(\phi_i, \phi_j) + \sum_{i=1}^{N}\sum_{j=1}^{N} \ (v_j - u_i) \ a(\phi_i, \phi_j) \ u_i \ \geqslant \ \sum_{i=1}^{N} \ \sum_{j=1}^{N}(v_j - u_i)\ell(\phi_j - \phi_i), \tag{4.5.3}$$

In matrix form it becomes

$$(\mathbf{v} - \mathbf{u})^T M \frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} + (\mathbf{v} - \mathbf{u})^T L\mathbf{u} \geqslant (\mathbf{v} - \mathbf{u})^T \mathbf{f}.$$

where matrix L is called the stiffness matrix, with

$$[L]_{ij} = a(\phi_j, \phi_i),$$

and M is called the mass matrix, with

$$[M]_{ij} = \int_\Omega \phi_i \phi_j \; d\Omega$$

$$[\mathbf{f}]_i \;=\; \ell(\phi_i),$$

and

$$[\Psi]_i \;=\; \psi_i(\mathbf{x}, t).$$

For this parabolic problem, the matrix L is positive definite, symmetric and sparse.

4.6 Discretization in time

We shall now consider time discretization methods to convert the semi-discrete problem (4.5.1) into a fully discrete problem. We will consider the backward Euler and the Crank-Nicolson methods [50]. Let $0 = t_0 < t_1 < ...t_m = T$ be the subdivision of the interval $[0, T]$, and let $\Delta t_k = t_k - t_{k-1}$ be the time step. Then at each time step the fully-discrete problem reads; find $u_h^k \in V_h$, such that for all $k = 0, ...t_m - 1$,

$$\begin{cases} \left(\dfrac{u_h^{k+1} - u_h^k}{\Delta t_k}, v_h - u_h^{k+1} \right) + a(u_h^{k+\theta}, v_h - u_h^{k+1}) - \ell(v_h - u_h^{k+1}) \;\geqslant\; 0, \\ u_h(\mathbf{x}, 0) \;\geqslant \psi_h(\mathbf{x}, 0), \\ u_h^{k+\theta} = u_h^k + \theta(u_h^{k+1} - u_h^k), \end{cases} \tag{4.6.1}$$

where $\theta \in [0, 1]$ is fixed. For $\theta = 1$, the scheme becomes the backward Euler method and for $\theta = 1/2$ the scheme is the Crank-Nicolson method.

4.6.1 Backward Euler scheme

The backward Euler method for the semi-discrete problem reads find $u_h^k \in V_h$, such that for all $k = 0, ...t_m - 1$,

$$\begin{cases} \left(\dfrac{u_h^{k+1} - u_h^k}{\Delta t_k}, v_h - u_h^{k+1}\right) + a(u_h^{k+1}, v_h - u_h^{k+1}) \geqslant \ell(v_h - u_h^{k+1}) \quad \forall v \in V_h, \\ u_h^{k+1} \geqslant \psi^{k+1}, \end{cases} \tag{4.6.2}$$

the initial condition is

$$(u_h(\mathbf{x}, 0), v_h - u_h) \geqslant (\psi_h(\mathbf{x}, 0), v_h - u_h). \tag{4.6.3}$$

4.6.2 Crank-Nicolson scheme

The Crank-Nicolson method for the semi-discrete problem can be described as find $u_h^k \in V_h,\ k = 0, ...t_m - 1$, such that

$$\begin{cases} \left(\dfrac{u_h^{k+1} - u_h^k}{\Delta t_k}, v_h - u_h^{k+1}\right) + a\left(\dfrac{u_h^{k+1} + u_h^k}{2}, v_h - u_h^{k+1}\right) \geqslant \ell(v_h - u_h^{k+1}) \quad \forall v \in V_h, \\ u_h^{k+1} \geqslant \psi^{k+1}, \end{cases} \tag{4.6.4}$$

with initial condition

$$(u(\mathbf{x}, 0), v - u) \geqslant (\psi(\mathbf{x}, 0), v - u). \tag{4.6.5}$$

To discretized in time we considered backward Euler and Crank-Nicolson methods [50]. We know from general theory of numerical analysis, that these methods are unconditionally stable and are first order and second order accurate respectively.

4.7 Minimization formulation

In this section we consider the minimization formulation for the Backward Euler and the Crank-Nicolson schemes given in (4.6.2)-(4.6.5).

Backward Euler scheme

The backward Euler method for the minimization problem reads

$$\begin{cases} \text{for} \quad k = 0, 1, ..., t_m, \\ \text{find } u_h^k \in K_h, \text{ such that } \forall v_h \in K_h, \\ J(u_h^k) \leqslant J(v_h), \end{cases} \tag{4.7.1}$$

with initial condition

$$u(\mathbf{x}, 0) \geqslant \psi(\mathbf{x}, 0),$$

where

$$J(u_h^k) = \frac{1}{2}\tilde{a}(u_h^k, u_h^k) - \tilde{\ell}(u_h^k),$$

$$\tilde{a}(u_h^k, u_h^k) = a(u_h^k, u_h^k) + \frac{1}{\Delta t_k}(u_h^k, u_h^k)$$

$$\tilde{\ell}(u_h^k) = \ell(u_h^k) + \frac{1}{\Delta t_k}(u_h^k, u_h^{k-1}).$$

Crank-Nicolson scheme

Crank-Nicolson method for Minimization problem can be written as

$$\begin{cases} \text{for} \quad k = 0, 1, ..., t_m, \text{find } u_h^k \in K_h, \text{such that } \forall v_h \in K_h, \\ J(u_h^k) \leqslant J(v_h), \end{cases} \tag{4.7.2}$$

with initial condition

$$u_h(\mathbf{x}, 0) \geqslant \psi_h(\mathbf{x}, 0),$$

where here

$$J(u_h^k) = \frac{1}{2}\tilde{a}(u_h^k, u_h^k) - \tilde{\ell}(u_h^k),$$

$$\tilde{a}(u_h^k, u_h^k) = a(u_h^k, u_h^k) + \frac{1}{\Delta t_k}(u_h^k, u_h^k),$$

$$\tilde{\ell}(u_h^k) = \ell(u_h^k) + \frac{1}{\Delta t_k}(u_h^k, u_h^{k-1}) - \frac{1}{2}a(u_h^k, u_h^{k-1}).$$

4.8 Matrix formulation of semi-discrete problem

The system of linear equations given in (4.5.3) can be expressed in the matrix form as

$$(\mathbf{v}-\mathbf{u})^T\ M\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{t}} + (\mathbf{v}-\mathbf{u})^T L\mathbf{u}\ \geqslant\ (\mathbf{v}-\mathbf{u})^T\mathbf{f},$$

or equivalently

$$M\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{t}} + L\mathbf{u} \geqslant \mathbf{f}, \tag{4.8.1}$$

with initial condition

$$\mathbf{u}(\mathbf{x}, 0) \geqslant \Psi(\mathbf{x}, 0),$$

where $L, M, \mathbf{f}, \Psi$, are defined previously.

4.8.1 Matrix formulation for fully discrete problem in time

The matrix formulation for the fully discrete problem can be written as

$$\begin{cases} \widehat{A}\mathbf{u}^k \geqslant \widehat{\mathbf{f}}^k, \\ \mathbf{u}^k \geqslant \Psi^k, \\ (\mathbf{u}^k - \Psi^k)_i(\widehat{A}\mathbf{u}^k - \widehat{\mathbf{f}}^k)_i = 0 \ \ \forall\ i, \end{cases} \tag{4.8.2}$$

where $\widehat{A}$ and $\widehat{\mathbf{f}}$ depends on the time discretization scheme.

Backward Euler scheme

For the backward Euler scheme we have $\widehat{A} = \dfrac{M}{\Delta t} + L$ and $\widehat{\mathbf{f}} = \dfrac{M}{\Delta t} + \mathbf{f}$. Thus we obtain the following system of equation at each time step $k = 0, 1..., t_m - 1$

$$VI\left(\frac{M}{\Delta t_k} + L, \mathbf{f}, \Psi\right) : \begin{cases} \left(\dfrac{M}{\Delta t_k} + L\right)\mathbf{u}^{k+1} \geqslant \dfrac{M}{\Delta t_k}\mathbf{u}^k + \mathbf{f}^{k+1} \\ \mathbf{u}^{k+1} \geqslant \Psi^{n+1} \\ \left(\left(\dfrac{M}{\Delta t_k} + L\right)\mathbf{u}^{k+1} - \dfrac{M}{\Delta t_k}\mathbf{u}^k - \mathbf{f}^{k+1}\right)(\mathbf{u}^{k+1} - \Psi^{k+1}) = 0, \end{cases} \tag{4.8.3}$$

with initial condition

$$\mathbf{u}^0 = \mathbf{u}(\mathbf{x}, 0) \geqslant \Psi(\mathbf{x}, 0),$$

Crank-Nicolson scheme

For the Crank Nicolson scheme we have $\widehat{A} = \dfrac{M}{\Delta t} + \dfrac{L}{2}$ and $\widehat{\mathbf{f}} = \dfrac{M}{\Delta t} - \dfrac{L}{2} + \mathbf{f}$. Thus we obtain the following inequalities at each step $k = 0, 1..., t_m - 1$

$$VI\left(\frac{M}{\Delta t_k} + \frac{L}{2}, \Psi, \widehat{\mathbf{f}}\right) : \begin{cases} \left(\dfrac{M}{\Delta t_k} + \dfrac{L}{2}\right)\mathbf{u}^{k+1} + \left(-\dfrac{M}{\Delta t_k} + \dfrac{L}{2}\right)\mathbf{u}^k \geqslant \widehat{\mathbf{f}}^{k+1} \\ \mathbf{u}^{k+1} \geqslant \Psi^{k+1} \\ \left(\left(\dfrac{M}{\Delta t_k} + \dfrac{L}{2}\right)\mathbf{u}^{k+1} + \left(-\dfrac{M}{\Delta t_k} + \dfrac{L}{2}\right)\mathbf{u}^k - \widehat{\mathbf{f}}^{k+1}\right)(\mathbf{u}^{k+1} - \Psi^{k+1}) = 0, \end{cases} \tag{4.8.4}$$

with initial condition

$$\mathbf{u}^0 = \mathbf{u}(\mathbf{x}, 0) \geqslant \Psi(\mathbf{x}, 0).$$

4.8.2 Minimization formulation

The matrix formulation of minimization problems (4.7.1) and (4.7.2) can be written as follows.

Backward Euler scheme

In this case, we obtain the following set of minimization problem at each time step k

$$\begin{cases} \text{for} \quad k = 0, 1, ...t_m - 1, \\ \text{minimize } J(\mathbf{u}^{k+1}) = \frac{1}{2}(\mathbf{u}^{k+1})^T \left(\frac{M}{\Delta t_k} + L \right) \mathbf{u}^{k+1} - \left(\frac{M}{\Delta t_k} \mathbf{u}^k + \mathbf{f}^{k+1} \right) \mathbf{u}^{k+1}, \\ \mathbf{u}^{k+1} \geqslant \Psi^{k+1}, \end{cases} \tag{4.8.5}$$

with initial condition

$$\mathbf{u}^0 = \mathbf{u}(\mathbf{x}, 0) \geqslant \Psi(\mathbf{x}, 0).$$

Crank-Nicolson scheme

In this case, we obtain the following set of minimization problem at each time step k

$$\begin{cases} \text{for} \quad k = 0, 1, ...t_m - 1 \\ \text{minimize } J(\mathbf{u}^{k+1}) = \frac{1}{2}(\mathbf{u}^{k+1})^T \left(\frac{M}{\Delta t_k} + \frac{L}{2} \right) \mathbf{u}^{k+1} - \left((\frac{M}{\Delta t_k} - \frac{L}{2}) \mathbf{u}^k - \mathbf{f}^{k+1} \right) \mathbf{u}^{k+1}, \\ \mathbf{u}^{k+1} \geqslant \Psi^{k+1}, \end{cases} \tag{4.8.6}$$

with initial condition

$$\mathbf{u}^0 = \mathbf{u}(\mathbf{x}, 0) \geqslant \Psi(\mathbf{x}, 0).$$

4.9 LCP formulation of parabolic variational Inequalities

In this section we shall describe the LCP formulation of the variational inequalities (4.8.3) and (4.8.4) constructed by backward Euler and Crank-Nicolson methods respectively.We recall here the generic LCP problem (3.5.1)

$$LCP(\tilde{M},\mathbf{q}): \begin{cases} \mathbf{w}-\tilde{M}\mathbf{z}=\mathbf{q}, \\ \mathbf{w}\geqslant 0, \quad \mathbf{z}\geqslant 0, \\ \mathbf{w}_i\mathbf{z}_i=0 \quad (1\leqslant i\leqslant n). \end{cases}$$

Backward Euler Scheme

Comparing (4.8.3) with (3.5.1), we obtain
$\tilde{M}=\dfrac{M}{\Delta t_k}+L,\ \ \mathbf{z}=\mathbf{u}^{k+1}-\Psi^{k+1},\quad q=\left(\dfrac{M}{\Delta t_k}+L\right)\Psi^{k+1}-\dfrac{M}{\Delta t_k}\mathbf{u}^k-\mathbf{f}^{k+1}.$
We conclude that

$$VI\left(\frac{M}{\Delta t_k}+L,\mathbf{f},\Psi\right)\equiv LCP\left(\frac{M}{\Delta t_k}+L,\left(\frac{M}{\Delta t_k}+L\right)\Psi^{k+1}-\frac{M}{\Delta t_k}\mathbf{u}^k-\mathbf{f}^{k+1}\right).$$

Hence, the above LCP formulation for the backward Euler method can be written as

$$\begin{cases} \text{find} \ \ \mathbf{u}^k\in\mathbb{R}^{m\times N} \\ \text{with} \ \ \mathbf{u}^0 \ \geqslant \ \Psi(\mathbf{x},0), \\ \text{such that for} \ \ k=0,1...,t_m-1, \\ \mathbf{u}^{k+1}=LCP\left(\dfrac{M}{\Delta t_k}+L,\left(\dfrac{M}{\Delta t_k}+L\right)\Psi^{k+1}-\dfrac{M}{\Delta t_k}\mathbf{u}^k-\mathbf{f}^{k+1}\right). \end{cases} \tag{4.9.1}$$

Crank-Nicolson scheme

For the Crank-Nicolson method we have
$M=\dfrac{M}{\Delta t_k}+\dfrac{L}{2},\ \ \mathbf{z}=\mathbf{u}^{k+1}-\Psi^{k+1},\quad \mathbf{q}=\left(\dfrac{M}{\Delta t_k}+\dfrac{L}{2}\right)\Psi^{k+1}+\left(-\dfrac{M}{\Delta t_k}+\dfrac{L}{2}\right)\mathbf{u}^k-\mathbf{f}^{k+1},$
and therefore

$$VI\left(\frac{M}{\Delta t_k}+\frac{L}{2},\Psi,\mathbf{f}\right)\equiv LCP\left(\left(\frac{M}{\Delta t_k}+\frac{L}{2}\right),\left(\frac{M}{\Delta t_k}+\frac{L}{2}\right)\Psi^{k+1}+\left(-\frac{M}{\Delta t_k}+\frac{L}{2}\right)\mathbf{u}^k-\mathbf{f}^{k+1}\right).$$

Hence, the above LCP formulation for the Crank-Nicolson method can be written as

$$\begin{cases} \text{find } \ \mathbf{u}^k \in \mathbb{R}^{m\times N} \\ \text{with } \mathbf{u}^0 \ \geqslant \ \Psi(\mathbf{x},0), \\ \text{such that for } \ k=0,1...,t_m-1, \\ \mathbf{u}^{k+1} = LCP\left(\left(\dfrac{M}{\Delta t_k}+\dfrac{L}{2}\right), \left(\dfrac{M}{\Delta t_k}+\dfrac{L}{2}\right)\Psi^{k+1} + \left(-\dfrac{M}{\Delta t_k}+\dfrac{L}{2}\right)\mathbf{u}^k + \mathbf{f}^{k+1}\right). \end{cases} \tag{4.9.2}$$

4.9.1 QP formulation of minimization problem

In this section, we describe the QP formulation of the minimization problem (4.8.5) and (4.8.2) constructed by backward Euler and Crank-Nicolson methods respectively.

Backward Euler scheme

Comparing (4.8.5) with (3.5.5) (see also (3.4.2)) we obtain the following equivalence
$D = \dfrac{M}{\Delta t_k} + L,\quad \mathbf{c} = -\dfrac{M}{\Delta t_k}\mathbf{u}^k - \mathbf{f}^{k+1},\ \ \mathbf{b} = \Psi^{k+1},\ \ \mathbf{A} = \mathbf{I}.$
We conclude that

$$\text{MIN}\left(\frac{M}{\Delta t_k}+L, \frac{M}{\Delta t_k}\mathbf{u}^k + \mathbf{f}^{k+1}, \Psi^{k+1}\right) \equiv QP\left(\mathbf{I}, \Psi^{k+1}, -\frac{M}{\Delta t_k}\mathbf{u}^k - \mathbf{f}^{k+1}, \frac{M}{\Delta t_k}+L\right).$$

Hence the above QP formulation for backward Euler method can be written as

$$\begin{cases} \text{find } \ \mathbf{u}^k \in \mathbb{R}^{m\times N} \\ \text{with } \mathbf{u}^0 \ \geqslant \ \Psi(\mathbf{x},0), \\ \text{such that for } \ k=0,1...,t_m-1, \\ \mathbf{u}^{k+1} = QP\left(\mathbf{I}, \Psi^{k+1}, (-\dfrac{M}{\Delta t_k}+L)\mathbf{u}^k - \mathbf{f}^{k+1}), \dfrac{M}{\Delta t_k}+L\right). \end{cases} \tag{4.9.3}$$

Crank-Nicolson scheme

Similarly for the Crank-Nicolson method, we obtain

$$D = \frac{M}{\Delta t_k} + L, \qquad \mathbf{c} = (-\frac{M}{\Delta t_k} + \frac{L}{2})\mathbf{u}^k - \mathbf{f}^{k+1}, \quad \mathbf{b} = \Psi^{k+1}, \quad \mathbf{A} = \mathbf{I}.$$

$$\text{MIN}\left(\frac{M}{\Delta t_k} + \frac{L}{2}, \Psi^{k+1}, (\frac{M}{\Delta t_k} - \frac{L}{2})\mathbf{u}^k + \mathbf{f}^{k+1}\right) \equiv QP\left(\mathbf{I}, \Psi^{k+1}, (-\frac{M}{\Delta t_k} + \frac{L}{2})\mathbf{u}^k - \mathbf{f}^{k+1}, \frac{M}{\Delta t_k} + \frac{L}{2}.\right)$$

Hence the above QP formulation for Crank-Nicolson method can be written as

$$\begin{cases} \text{find } \ \mathbf{u}^k \in \mathbb{R}^{m \times N} \\ \text{with } \mathbf{u}(\mathbf{x}, 0) \ \geqslant \ \Psi(\mathbf{x}, 0), \\ \text{such that for } \ k = 0, 1..., t_m - 1, \\ \mathbf{u}^{k+1} = QP\left(\mathbf{I}, \Psi^{k+1}, \left(-\frac{M}{\Delta t_k} + \frac{L}{2}\right)\mathbf{u}^k - \mathbf{f}^{k+1}, \frac{M}{\Delta t_k} + \frac{L}{2}\right). \end{cases} \tag{4.9.4}$$

In this chapter we introduced the parabolic variational inequalities. We applied the finite element method to discretize in space. To convert into fully discrete problem we used standard backward Euler and Crank-Nicolson methods. We described the variational and minimization formulations of parabolic variational inequalities and the corresponding LCP and QP problems.

Chapter 5

Domain Decomposition Methods For Elliptic Variational Inequalities

5.1 Solution methods

Domain decomposition methods (DDM) form an active research field in the area of iterative methods. They are typically used for large scale algebraic systems arising from the modeling of partial differential equations, for which parallel algorithms are desired. These methods are a powerful tool for devising parallel algorithms. Domain decomposition methods can be categorized into two branches namely: overlapping and non overlapping. Overlapping DDM are usually referred to as Schwarz alternating method and the additive Schwarz method [8], [20]. Non overlapping methods are usually called substructuring methods [118]. In both cases, the problem is formulated in each subdomain, yielding a family of reduced size subproblems that are coupled together through some suitable boundary conditions at the interface. These reduced problems are solved at the expense of an iterative procedure among the subdomains to impose the interface coupling. Overlapping domain decomposition methods are inefficient when the region of overlap is reduced [13] or in the case of elliptic problems with large jump coefficients [118]. A comparison between overlapping and non overlapping DDM can be found in [24] [11] [18], [19].

Another important class of solution techniques is that of multilevel and multi-grid methods for constrained minimization problems, introduced in [87] and [48]. Some variants of these method were studied in [73] and were analyzed in [74]. A challenging task for multi-grid method when applied to variational inequalities, is the representation of the coincidence set on a coarse grid, as shown in the review paper [52]. Some multi-grid and two level domain decomposition methods are given in [111] [107] in which it is shown that the overlapping DDM has linear convergence for the constrained obstacle problem if the obstacle and computed functions have been decomposed in a suitable manner. Some more variants of multi-grid method are discussed in [10], where the decomposition of the closed convex set for the minimization problem is introduced as a sum of closed convex level subsets; the convergence rate is shown to depend on the number of levels.

In the following, we give a literature review of domain decomposition methods for partial differential equations and variational inequalities.

5.1.1 DDM for partial differential equations

In 1870 Schwarz [103] proposed an overlapping domain decomposition method to compute the numerical solutions of partial differential equations on an exotic domain combining a disc and a rectangle. A detailed discussion for overlapping and non overlapping DDM for partial differential equations and related algorithm to solve these problems can be found in [23], [92], [114], [116]. A brief study of the Schwarz alternating method for a range of different problems including Laplace equations and Stokes equations can be found in [85], [84], [86]. These papers emphasize the excellent convergence properties of Schwarz methods, and why Schwarz methods can be applied and extended to a wide range of problems. Another class of Schwarz methods are the two level or multi level additive Schwarz methods, introduced in [35], [34], [37]. A two level method for symmetric and positive definite problem is given in [39]. In this paper, a good convergence rate is proved and the running time of the problem is shown to reduce for subdomains with a small region of overlap. An increasing number of conjugate gradient iterations are compensated by the fact that the local problems are smaller. A generalization of the two level methods is described in [126] [12], [38], [125]. In these papers, the family of domain decomposition methods are shown to be merged with multi-grid methods.

Several domain decomposition methods are proposed and analyzed for solving symmetric positive definite elliptic problems. For these problems, domain decomposition methods have been shown to yield a good condition number as an iterative method and can be implemented in parallel. Domain decomposition methods for solving convection diffusion problems are few as compared to elliptic problems. Some domain decomposition methods for such problems are described in [46], [59], [6], [77]. A DDM for convection dominated problem was proposed and analyzed in [93], where convergence is achieved without any condition on the macro element. An overlapping DDM is proposed in [67], in which iterative Schwarz method is used for a convected-dominated problem and convergence is proved for the algorithm. Numerical experiments presented there show that the number of iterations depends on the number of subdomains in the flow direction. A combination of sequential and parallel DDM for convection dominated problems is given in [127]. The sequential algorithm is applied in the down stream direction and the parallel algorithm is used in the crosswind direction. In each iteration, a local problem is solved by stream line diffusion finite element method with artificial boundary conditions. The convergence of the global approximated solution is shown to be of order $O(h^{3/2})$ in the L^2- norm.

5.1.2 DDM for variational inequalities

There are classical iterative methods like point projection and point over relaxation methods [50]. However, the convergence rate of these methods has been shown to substantially increase on fine meshes. Domain decomposition methods for variational inequalities were initially used in [85], and subsequently many algorithms were constructed and analyzed. For example, some Schwarz algorithms for variational inequalities are described in [86], [128], [129]. A DDM method for the Signorini problem is presented in [102], where a projection method is applied to solve the constrained minimization problem and an optimal preconditioner for the decomposed problem is derived. [110] proposed a DDM by using space decomposition techniques for variational inequalities and were able to prove good rate of convergence.

An additive Schwarz method for the obstacle problem is presented in [9] and shows nice recurrence for the error between two consecutive steps. Geometric convergence is proved by using projection operators onto a closed convex Hilbert space. An overlapping DDM with monotone operators for the obstacle problem is given in [76], where it is shown that the algorithm is monotonically convergent with a suitable choice of initial guess. A generalized Schwarz algorithm for the obstacle problem with two subdomains is given in [129], where the obstacle problem is solved with the Robin boundary condition $g_i(v) = \theta_i(v) + (1-\theta_i)\frac{\partial v}{\partial n_i}$, $0 < \theta \leqslant 1$. Some convergence results for variational inequalities can be found in [107], [108], [109] [122], [123]. For example [108], [109] proposed some general space minimization problems for convex functional over a convex constrained subsets, using space decomposition methods. This work is further extended in [106] , where an analysis of the rate of convergence of the method is provided. [123] proposes a monotonic convergent method for the two sided obstacle problem. The estimate of convergence is given there, is valid only for the two subdomain case and with uniform overlapping size. The systems with different governing problems in different subdomains can be solved by non overlapping (DD) methods by solving in parallel for the solution in each subdomain. A non overlapping DDM with two subdomains for the free boundary problem is given in [61]. Convergence analysis and numerical results are also given to support the algorithm. Another non-overlapping domain decomposition method for the obstacle problem is given in [62]. The global domain is converted into several non overlapping subdomains and then in each subdomain a variational inequality is solved. Numerical experiments presented there show that the algorithm is monotonically convergent, but the rate of convergence is mesh dependent.

5.2 A non-overlapping domain decomposition method

In this chapter we introduce a non overlapping domain decomposition method for elliptic variational inequalities. The purpose of using the domain decomposition method is to reformulate the original problem into two subproblems showing different behaviour in their respective subdomains, namely one subproblem in some subdomains governed by a partial differential equation and the other in the complementary subdomain satisfying a variational inequality. Each problem is then solved locally (in each subdomain), with Dirichlet boundary conditions on the interface, and the solutions are coupled together through an interface problem.

5.2.1 Formulation of the method

Let Ω be a smooth bounded open convex domain with boundary $\partial\Omega$. We recall here the variational inequality problem described in Chapter 2:

$$\begin{cases} \mathcal{L}u \geqslant f \;\; \text{in } \Omega, \\ u \;\geqslant\; \psi \;\; \text{in } \Omega, \\ (\mathcal{L}u - f)(u-\psi) = 0 \;\; \text{in } \Omega, \\ u \;=\; 0 \;\; \text{on } \partial\Omega. \end{cases} \tag{5.2.1}$$

The weak formulation of the above inequality is

$$\begin{cases} \text{find } u \ \in \ K \ \text{ such that } \ \forall v \in K, \\ a(u, v-u) \geqslant \ell(v-u), \end{cases} \tag{5.2.2}$$

where the bilinear form is given by

$$a(v,w) = \int_\Omega \nabla v \cdot \mathbf{a} \cdot \nabla w \ d\Omega + \int_\Omega \mathbf{b} \cdot \nabla vw \ d\Omega + \int_\Omega cvw \ d\Omega'$$

and

$$K = \{v \in V : v \geqslant \psi \text{ in } \Omega, \} \ \text{ with } V = H_0^1(\Omega).$$

Let Ω be partitioned into m non-overlapping disjoint open subdomains $\Omega_j, \ \ j = 1, ..., m$ and let $\mathit{\Gamma}_j := \partial\Omega_j \setminus \partial\Omega, \ \ $ with $\mathit{\Gamma} = \bigcup_{j=1}^m \mathit{\Gamma}_j$, the interface. Thus,

$$\overline{\Omega} = \bigcup_{j=1}^{m} \overline{\Omega}_j \text{ and } \Omega_i \cap \Omega_j = \emptyset.$$

If we denote by u_j, f_j, ψ_j the restrictions of u, f, ψ respectively to each subdomain Ω_j and by λ_j, the solution restricted to the interface $\mathit{\Gamma}_j$, then problem (5.2.1) can be defined in each subdomain Ω_j for $j = 1, ..., m$ as follows

$$\begin{cases} \text{find } \ u_j \in K \ \text{ in } \ \Omega_j \ \text{ such that} \\ \mathcal{L}u_j \geqslant f_j \ \text{ in } \ \Omega_j, \\ u_j \ \geqslant \ \psi_j \ \text{ in } \ \Omega_j, \\ (\mathcal{L}u_j - f)(u_j - \psi_j) = 0, \ \text{ in } \ \Omega_j, \\ u_j = \lambda_j, \ \text{ on } \ \partial\Omega_j \cap \mathit{\Gamma}, \\ u_j = 0, \ \text{ on } \ \partial\Omega \cap \partial\Omega_j. \end{cases} \tag{5.2.3}$$

To apply the idea of partitioning the domain in our problem, we decompose the domain into two multiply-connected subdomains Ω^e and Ω^i. In Ω^e we solve a PDE and in Ω^i we solve a PDI. This partitioning requires some mild assumption about the coincidence set $\mathcal{C}$ for u and ψ to locate Ω^i. We choose Ω^i, the domain containing the region corresponding to the coincidence set, then $\Omega^e = \Omega \setminus \Omega^i$.

To understand this partitioning of Ω, we consider the Figure (5.1) for $1D$ obstacle problem.

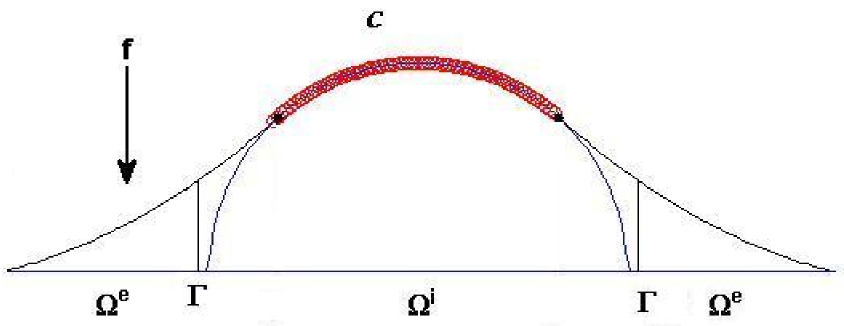


Figure 5.1: Domain decomposition for obstacle problem in 1D.

Since it is not possible to estimate the exact extreme points for the coincidence set $\mathcal{C}$, we assume that the interface Γ lies outside the support of the obstacle. This mild assumption allows us to pose the problem (5.2.1) as a PDE on Ω^e as well as on the interface Γ.

Thus for our problem the partitioning of Ω together with interface Γ can be defined as

$$\Omega = \Omega^e \cup \Omega^i \cup \Gamma, \quad \text{where} \quad \Gamma = \overline{\Omega^e} \cap \overline{\Omega^i}.$$

Let $z = u|_{\Omega^e}, w = u|_{\Omega^i}$, $f^e = f|_{\Omega^e}$ and $f^i = f|_{\Omega^i}$ be the restrictions of u and f on Ω^e and Ω^i respectively. We shall use superscript e for subdomain Ω^e and superscript i for subdomain Ω^i throughout this chapter. By using the above partitioning the problem (5.2.3) can equivalently be written as two subproblems PDE and PDI as follows

$$PDE : \begin{cases} \mathcal{L}z = f \quad \text{in} \;\; \Omega^e, \\ z \;=\; 0 \;\; \text{on} \;\; \partial\Omega^e \setminus \Gamma, \\ z \;=\; \lambda \;\; \text{on} \;\; \Gamma, \end{cases} \qquad PDI : \begin{cases} \mathcal{L}w \geqslant f \;\; \text{in} \;\; \Omega^i, \\ w \;\geqslant\; \psi \;\; \text{in} \;\; \Omega^i, \\ (\mathcal{L}w - f)(w - \psi) = 0 \;\; \text{in} \;\; \Omega^i, \\ w \;=\; 0 \;\; \text{on} \;\; \partial\Omega^i \setminus \Gamma, \\ w \;=\; \lambda \;\; \text{on} \;\; \Gamma, \end{cases} \tag{5.2.4}$$

where λ is the value of solutions z and w on Γ respectively.

The subproblem PDE can be decoupled into two sets of problems, as for the case of standard domain decomposition methods for PDE [92]

$$PDE_1 : \begin{cases} \mathcal{L}z_1 = f \quad \text{in} \;\; \Omega^e, \\ z_1 \;=\; 0 \;\; \text{on} \;\; \partial\Omega^e, \\ z_1 = 0 \;\; \text{on} \;\; \Gamma, \end{cases} \quad PDE_2 : \begin{cases} \mathcal{L}z_2 = 0 \quad \text{in} \;\; \Omega^e, \\ z_2 \;=\; 0 \;\; \text{on} \;\; \partial\Omega^e \setminus \Gamma, \\ z_2 = \lambda \;\; \text{on} \;\; \Gamma, \end{cases} \tag{5.2.5}$$

Note that z_2 is the $\mathcal{L}$-extension of λ to domain Ω^e and will be denoted by $E\lambda$, where E

is the $\mathcal{L}$-extension operator. By using above definitions, the solution in each subdomain can be written as

$$u|_{\Omega^e} = z = z_1 + z_2,$$

and

$$u|_{\Omega^i} = w.$$

5.2.2 Variational formulation

Note that by using the above splitting the weak formulation (5.2.2) can be written as

$$\begin{cases} \text{find } u \in K \quad \text{such that } \forall v \in K, \\ a^e(z, v-z) + a^i(w, v-w) \geqslant \ (f, v-z)_{\Omega^e} + (f, v-w)_{\Omega^i}, \end{cases} \tag{5.2.6}$$

where

$$a^e(z,v) = \int_{\Omega^e} \nabla z \cdot \mathbf{a} \cdot \nabla v \; d\Omega^e + \int_{\Omega^e} \mathbf{b} \cdot \nabla z v \; d\Omega^e + \int_{\Omega^e} czv \; d\Omega^e.$$

and

$$a^i(w,v) = \int_{\Omega^i} \nabla w \cdot \mathbf{a} \cdot \nabla v \; d\Omega^i + \int_{\Omega^i} \mathbf{b} \cdot \nabla w v \; d\Omega^i + \int_{\Omega^i} cwv \; d\Omega^i.$$

The variational formulations of (5.2.5) and PDI are

$$\begin{cases} \text{find } z_1 \in H_0^1(\Omega^e) \text{ such that } \ \forall v \in H^1(\Omega^e) \\ a^e(z_1, v-z) - \displaystyle\int_{\mathit{\Gamma}} \mathbf{n}_1 \cdot \nabla z_1 \cdot (v-z) \; d\mathit{\Gamma} = (f^e, v-z)_{\Omega^e}, \end{cases} \tag{5.2.7}$$

$$\begin{cases} \text{find } z_2 \in H^1(\Omega^e) \text{ such that } \ \forall v \in H_0^1(\Omega^e) \\ a^e(z_2, v-z) - \displaystyle\int_{\mathit{\Gamma}} \mathbf{n}_1 \cdot \nabla z_2 \cdot (v-z) \; d\mathit{\Gamma} = 0, \end{cases} \tag{5.2.8}$$

$$\begin{cases} \text{find } w \in K \text{ such that } \ \forall v \in K \\ a^i(w, v-w) - \displaystyle\int_{\mathit{\Gamma}} \mathbf{n}_2 \cdot \nabla w \cdot (v-w) d\mathit{\Gamma} \geqslant (f^i, v-w)_{\Omega^i}. \end{cases} \tag{5.2.9}$$

Adding above weak formulations we get

$$
\begin{aligned}
&a^e(z_1, v-z) - \int_\Gamma \mathbf{n}_1 \cdot \nabla z_1 \cdot (v-z) d\Gamma + a^e(z_2, v-z) - \int_\Gamma \mathbf{n}_1 \cdot \nabla z_2 \cdot (v-z) d\Gamma \\
&+ a^i(w, v-w) - \int_\Gamma \mathbf{n}_2 \cdot \nabla w \cdot (v-w) d\Gamma \geqslant (f, v-z)_{\Omega^e} + (f, v-w)_{\Omega^i}.
\end{aligned}
$$

Using $z_1 = 0$, $z_2 = \lambda = w$ on Γ and letting $v = F\mu$, where F is any extension operator we have

$$
\begin{aligned}
&a^e(z_1, \mu-\lambda) + a^i(w, \mu-\lambda) - \int_\Gamma \mathbf{n}_1 \cdot \nabla z_1 \cdot (\mu-\lambda) d\Gamma - \int_\Gamma \mathbf{n}_1 \cdot \nabla z_2 \cdot (\mu-\lambda) d\Gamma - \\
&\int_\Gamma \mathbf{n}_2 \cdot \nabla w \cdot (\mu-\lambda) d\Gamma \geqslant (f, \mu-\lambda)_{\Omega^e} + (f, \mu-\lambda)_{\Omega^i}.
\end{aligned}
$$

Using the weak formulation (5.2.6) yields a partial Steklov-Poincaré inequality for λ :

$$
-\int_\Gamma \mathbf{n}_1 \cdot \nabla E\lambda \cdot (\mu-\lambda) d\Gamma \geqslant \int_\Gamma \mathbf{n}_1 \cdot \nabla z_1 \cdot (\mu-\lambda) d\Gamma + \int_\Gamma \mathbf{n}_2 \cdot \nabla w \cdot (\mu-\lambda) d\Gamma.
$$

$$
\int_\Gamma \mathbf{n}_1 \cdot \nabla E\lambda \cdot (\mu-\lambda) d\Gamma \ \leqslant -\int_\Gamma (\mathbf{n}_1 \cdot \nabla z_1 + \mathbf{n}_2 \cdot \nabla w) \cdot (\mu-\lambda) d\Gamma. \tag{5.2.10}
$$

Equivalently

$$
(\mathcal{S}^e\lambda, \mu-\lambda) \ \leqslant \ (g(\lambda), \mu-\lambda). \tag{5.2.11}
$$

As discussed in the beginning of the chapter, we assume that the interface Γ lies outside the support of the obstacle and the problem (5.2.1) satisfies a PDE for λ on the interface Γ. Therefore (5.2.11) can be seen to be equivalent to the non-linear variational problem

$$
(\mathcal{S}^e\lambda, \mu) \ = \ (g(\lambda), \mu), \tag{5.2.12}
$$

The Steklov-Poincaré operator $\mathcal{S}^e : \Lambda \to \Lambda'$ (where $\Lambda = H^{1/2}(\mathit{\Gamma}),\ H_0^{1/2}(\mathit{\Gamma})$ or $H_{00}^{1/2}(\mathit{\Gamma})$ depending on the nature of the problem) is defined as

$$(\mathcal{S}^e \lambda, \mu) \ := \int_{\mathit{\Gamma}} (\mathbf{n}_1 \cdot \nabla(E\lambda))\mu \ d\mathit{\Gamma},$$

and

$$(g(\lambda), \mu) \ := -\int_{\mathit{\Gamma}} (\mathbf{n}_1 \cdot \nabla z_1 + \mathbf{n}_2 \cdot \nabla w)\, \mu \ d\mathit{\Gamma}.$$

Applying Green's formula, we get the alternative representation of $\mathcal{S}^e$

$$(\mathcal{S}^e \lambda, \mu) := a^e(E\lambda, F\mu) \quad \forall \lambda, \mu \in \Lambda,$$

where F denotes an arbitrary extension operator to Ω^e. By using the above definition of $\mathcal{S}^e$, our classical problem can be written as an ordered sequence of three decoupled problems involving elliptic problem on subdomain Ω^e together with a problem set on the interface $\mathit{\Gamma}$ which is coupled with the problem on Ω^i.

$$\begin{cases} \mathcal{L}z_1 = f \quad \text{in} \;\; \Omega^e, \\ z_1 \;\; = \;\; 0 \;\; \text{on} \;\; \partial\Omega^e \setminus \mathit{\Gamma}, \\ z_1 \;\; = \;\; 0 \;\; \text{on} \;\; \mathit{\Gamma}, \end{cases} \tag{5.2.13a}$$

$$\begin{cases} \mathcal{S}^e\lambda = -\mathbf{n}\cdot\nabla z_1 - \mathbf{n}\cdot\nabla w, \end{cases} \tag{5.2.13b}$$

$$\begin{cases} \mathcal{L}w \geqslant f \;\; \text{in} \;\; \Omega^i, \\ w \;\; \geqslant \;\; \psi \;\; \text{in} \;\; \Omega^i, \\ (\mathcal{L}w - f)(w - \psi) = 0 \;\; \text{in} \;\; \Omega^i, \\ w = 0 \;\; \text{on} \;\; \partial\Omega^i \setminus \mathit{\Gamma}, \\ w \;\; = \;\; \lambda \;\; \text{on} \;\; \mathit{\Gamma}, \end{cases} \tag{5.2.13c}$$

$$\begin{cases} \mathcal{L}z_2 = 0 \quad \text{in} \;\; \Omega^e, \\ z_2 \;\; = \;\; 0 \;\; \text{on} \;\; \partial\Omega^e \setminus \mathit{\Gamma}, \\ z_2 = \lambda \;\; \text{on} \;\; \mathit{\Gamma}. \end{cases} \tag{5.2.13d}$$

The resulting solution in Ω^e, is

$$u|_{\Omega^e} = z = z_1 + z_2,$$

where as solutions for interface $\mathit{\Gamma}$ and Ω^i i.e. λ and w could be approximated in an iterative manner. To write down the weak formulation of above problem we rewrite the sets of equations (5.2.13d) as

$$\begin{cases} \mathcal{L}\tilde{z}_2 = \mathcal{L}q^e \quad \text{in} \;\; \Omega^e, \\ \tilde{z}_2 \;\; = \;\; 0 \;\; \text{on} \;\; \partial\Omega^e \setminus \mathit{\Gamma}, \\ \tilde{z}_2 = 0 \;\; \text{on} \;\; \mathit{\Gamma}, \end{cases} \tag{5.2.14}$$

where $\tilde{z}_2 = z_2 - q^e, q^e$ chosen such that $q^e \in H_0^1(\Omega^e)$. By using these notations, the weak formulations of domain decomposition problem can be written as

$$\begin{cases} \text{find } z_1 \in H_0^1(\Omega^e) \\ \text{such that } \forall\ v \in H_0^1(\Omega^e), \\ a^e(z_1, v) = (f, v)_{\Omega^e}, \end{cases} \tag{5.2.15a}$$

$$\begin{cases} \text{find } \lambda \in H_{00}^{1/2}(\mathit{\Gamma}) \text{ such that } \forall\ \mu \in H_{00}^{1/2}(\mathit{\Gamma}), \\ (\mathcal{S}^e\lambda, \mu) = ((f^e, F^e\mu^e) - a^e(z_1, F^e\mu^e)) + ((f^i, F^i\mu^i) - a^i(w, F^i\mu^i)), \end{cases} \tag{5.2.15b}$$

$$\begin{cases} \text{find } w \in H^1(\Omega^i) \\ \text{such that } \forall\ v \in H^1(\Omega^i), \\ a^i(w, v - w)\ \geqslant (f, v - w)_{\Omega^i}, \end{cases} \tag{5.2.15c}$$

$$\begin{cases} \text{find } \tilde{z}_2 = z_2 - q^e \in H_0^1(\Omega^e) \\ \text{such that } \forall\ v\ \in H_0^1(\Omega^e), \\ a^e(\tilde{z}_2, v) = -a^e(q^e, v). \end{cases} \tag{5.2.15d}$$

5.2.3 Finite element discretization

Finite element approximation of the domain decomposition formulation can be described as follows

Let $\Omega \subset \mathbb{R}^d$ be a bounded open convex set and let $\mathfrak{T}_h$ be some subdivision of $\overline{\Omega}$ into simplices $\mathbf{t}$. Then $\overline{\Omega}$ can be written as

$$\overline{\Omega} = \bigcup_{\mathbf{t} \in \mathfrak{T}_h} \mathbf{t}.$$

Let $P_r(\mathbf{t})$ denote the space of polynomials defined on $\mathbf{t}$ in d variables of degree less than or equal to r.

We define the finite dimensional spaces of continuous piecewise polynomial functions on some subdivision $\mathfrak{T}_h$ of Ω^e and Ω^i into simplices $\mathbf{t}$ of maximum diameter h, as

$$K_h^e := \{v_h \in C^0(\overline{\Omega}^e) \mid v_h \in P(\mathbf{t}),\ \forall \mathbf{t} \in \mathfrak{T}_h,\ v_h|_{\partial\Omega^e \cap \partial\Omega} = 0\},$$

$$K_h^i := \{v_h \in C^0(\overline{\Omega}^i) \mid v_h \in P(\mathbf{t}),\ \forall \mathbf{t} \in \mathfrak{T}_h,\ v \geqslant \psi \text{ in } \Omega^i \text{ and } v|_{\partial\Omega^i \cap \partial\Omega} = 0\}.$$

Let $\mathcal{N}^e, \mathcal{N}^i$ denote the sets of nodes located in the subdomains Ω^e, Ω^i and $\mathcal{N}^\Gamma$ be the number of nodes lying on the interface Γ, with $|\mathcal{N}^e| = N,\ |\mathcal{N}^i| = N^i,\ |\mathcal{N}^\Gamma| = N^\Gamma$. Let now

$$K_h^{e_I} = \text{span}\{\Phi_k,\ k \in \mathcal{N}^e\},$$

$$K_h^{i_I} = \text{span}\{\Phi_k,\ k \in \mathcal{N}^i\},$$

$$K_h^\Gamma = \text{span}\{\Phi_k,\ k \in \mathcal{N}^\Gamma\},$$

$$S^h = \text{span}\{\gamma_0(\Gamma)\Phi_k,\ k \in \mathcal{N}_i^\Gamma\}.$$

This yields the decomposition

$$K^h = K_h^{e_I} \oplus K_h^{i_I} \oplus K_h^\Gamma \subset H_0^1(\Omega).$$

By using the above definitions, we have the following finite element discretization for the domain decomposition method:

$$\begin{cases} \text{find } \ z_1^h \in K_h^e \\ a^e(z_1^h, v) = (f, v)_{\Omega^e}, \quad \forall v_h \in K_h^e \end{cases} \tag{5.2.16a}$$

$$\begin{cases} \text{find } \lambda_h \in S^h \text{ such that } \forall \ \mu_h \in S^h, \\ \mathcal{S}^e(\lambda_h, \mu_h) = \big((f^e, F^e\mu_h^e) - a^e(z_1^h, F^e\mu_h^e)\big) + ((f^i, F^i\mu_h^i) - a^i(w_h, F^i\mu_h^i)) \,, \end{cases} \tag{5.2.16b}$$

$$\begin{cases} \text{find } \ w_h \in K_h^i \ \text{ such that } \ \forall v_h \in K_h^i, \\ a^i(w_h, v_h - w_h) \geqslant (f, v_h - w_h)_{\Omega^i}, \end{cases} \tag{5.2.16c}$$

$$\begin{cases} \text{find } \tilde{z}_2^h = z_2^h - q_h^e \in K_h^e \\ \text{such that } \forall \ v_h \ \in K_h^e, \\ a^e(\tilde{z}_h, v) = -a^e(q_h^e, v). \end{cases} \tag{5.2.16d}$$

5.2.4 Matrix formulation

The discrete formulation (5.2.16a)-(5.2.16c) leads to a Schur complement approach for the solution of the problem (5.2.1). To obtain the matrix formulation of the above discrete formulation of the domain decomposition problem let us denote the unknowns vectors by $\mathbf{u}^e, \mathbf{u}^i, \mathbf{u}_\Gamma$ and the right hand side vectors by $\mathbf{f}^e, \mathbf{f}^i, \mathbf{f}_\Gamma$ of lengths N^e, N^i, N^Γ respectively, such that $N = N^e + N^i + N^\Gamma$, with $A \in \mathbb{R}^{N\times N}$ and $\mathbf{f} \in \mathbb{R}^N$. Then the matrix representation of (5.2.1) can be written as

$$\begin{pmatrix} A_{II}^e & O & A_{I\Gamma}^e \\ O & A_{II}^i & A_{I\Gamma}^i \\ A_{\Gamma I}^e & A_{\Gamma I}^i & A_{\Gamma\Gamma} \end{pmatrix} \begin{pmatrix} \mathbf{u}_I^e \\ \mathbf{u}_I^i \\ \mathbf{u}_\Gamma \end{pmatrix} \geqslant \begin{pmatrix} \mathbf{f}_I^e \\ \mathbf{f}_I^i \\ \mathbf{f}_\Gamma \end{pmatrix}, \tag{5.2.17}$$

$$\mathbf{u} \geqslant \Psi,$$

$$(A\mathbf{u} - \mathbf{f})_i(\mathbf{u} - \Psi)_i = 0.$$

Where we have partitioned the degree of freedom into those internal to Ω^e and to Ω^i and those on the interface $\mathit{\Gamma}$. In particular, we have

$$(A^e_{II})_{kj} = a^e(\Phi_k, \Phi_j) \quad k, j \in \mathcal{N}^e,$$

$$(A^i_{II})_{kj} = a^i(\Phi_k, \Phi_j) \quad k, j \in \mathcal{N}^i,$$

$$(A_{I\mathit{\Gamma}})^e_{kj} = a^e(\Phi_k, \Phi_j), \quad k \in \mathcal{N}^e,\ j \in \mathcal{N}^{\mathit{\Gamma}},$$

$$(A_{I\mathit{\Gamma}})^i_{kj} = a^i(\Phi_k, \Phi_j) \quad k \in \mathcal{N}^i,\ j \in \mathcal{N}^{\mathit{\Gamma}}.$$

Proposition 5.2.18 *By using the above notation, the solution of problem* (5.2.15a)-(5.2.15c) *has the following algebraic form*

$$\left\{ A^e_{II}\mathbf{u}^{e\{1\}}_I = \mathbf{f}^e_I, \right. \tag{5.2.19a}$$

$$\left\{ S^e\mathbf{u}_{\mathit{\Gamma}} = \mathbf{f}_{\mathit{\Gamma}} - A^e_{\mathit{\Gamma} I}\mathbf{u}^{e\{1\}}_I - A^i_{\mathit{\Gamma} I}\mathbf{u}^i_I, \right. \tag{5.2.19b}$$

$$\left\{ A^i_{II}\mathbf{u}^i_I \ \geqslant \ \mathbf{f}^i_I - A^i_{I\mathit{\Gamma}}\mathbf{u}_{\mathit{\Gamma}}, \right. \tag{5.2.19c}$$

$$\left\{ \mathbf{u}^{e\{2\}}_I = -(A^e_{II})^{-1}A^e_{I\mathit{\Gamma}}\mathbf{u}_{\mathit{\Gamma}}. \right. \tag{5.2.19d}$$

Equations (5.2.19a)-(5.2.19d) could be seen as a partial Schur complement approach for the system (5.2.1) and the solution for $\mathbf{u}^i_I$ and $\mathbf{u}_{\mathit{\Gamma}}$ will be approximated in an iterative manner. The resulting solution is then $[\mathbf{u}^{e\{1\}}_I + \mathbf{u}^{e\{2\}}_I, \mathbf{u}^i_I, \mathbf{u}_{\mathit{\Gamma}}]$. The proof that the solution of a linear system on Ω^e is $\mathbf{u}^{e\{1\}}_I + \mathbf{u}^{e\{2\}}_I$ can be found in [3], [92].

5.3 The Obstacle problem

To demonstrate the implementation of the domain decomposition method we will consider a well known example: the obstacle problem. Recall from Chapter 3 that the finite element formulation of the obstacle problem yields the discrete problem; find $\mathbf{u}$, such that

$$\begin{aligned} A\mathbf{u} &\geqslant \mathbf{f}, \\ \mathbf{u} &\geqslant \Psi, \\ (\mathbf{u}-\Psi)_i(A\mathbf{u}-\mathbf{f})_i &= 0 \ \ \forall\, i, \end{aligned}$$

where $A_{ij} = \int_\Omega \nabla\phi_i.\nabla\phi_j \, d\Omega, \ \Psi_i = \psi(\mathbf{x}_i)$ and $\mathbf{f}_i = \int_\Omega f\phi_i \, d\Omega$. Equivalently, using the same decomposition as used in (5.2.17) the above system could be written as

$$\begin{pmatrix} A^e_{II} & O & A^e_{I\Gamma} \\ O & A^i_{II} & A^i_{I\Gamma} \\ A^e_{\Gamma I} & A^i_{\Gamma I} & A_{\Gamma\Gamma} \end{pmatrix} \begin{pmatrix} \mathbf{u}^e_I \\ \mathbf{u}^i_I \\ \mathbf{u}_\Gamma \end{pmatrix} \geqslant \begin{pmatrix} \mathbf{f}^e_I \\ \mathbf{f}^i_I \\ \mathbf{f}_\Gamma \end{pmatrix}. \tag{5.3.1}$$

The solution method described in (5.2.19a)-(5.2.19c) based on the Schur complement approach can be applied to the above system for the obstacle problem. We could see that the domain decomposition method described in this chapter leads to a PDE in Ω^e corresponding to the first row block in (5.3.1), the second row block is a reduced variational inequality in Ω^i and the third row block can be considered as either a PDE or a PDI on Γ. We can write the PDE corresponding to Ω^e as

$$A^e_{II}\mathbf{u}^e_I + A^e_{I\Gamma}\mathbf{u}_\Gamma = \mathbf{f}^e_I.$$

Solving for $\mathbf{u}^e$ we get

$$\mathbf{u}^e_I = (A^e_{II})^{-1}(\mathbf{f}^e_I - A^e_{I\Gamma}\mathbf{u}_\Gamma). \tag{5.3.2}$$

Replacing $\mathbf{u}^e$ in the remaining system we get the reduced system for obstacle problem, i.e., only for the sub-domains containing the support of the obstacle

$$\begin{pmatrix} A^i_{II} & A^i_{I\Gamma} \\ A^i_{\Gamma I} & S^e \end{pmatrix} \begin{pmatrix} \mathbf{u}^i_I \\ \mathbf{u}_\Gamma \end{pmatrix} \geqslant \begin{pmatrix} \mathbf{f}^i_I \\ \tilde{\mathbf{f}}_\Gamma \end{pmatrix}, \tag{5.3.3}$$

where

$$S^e := A_{\Gamma\Gamma} - A^e_{\Gamma I}(A^e_{II})^{-1}A^e_{I\Gamma} \quad \text{and} \quad \tilde{\mathbf{f}}_\Gamma := \mathbf{f}_\Gamma - A^e_{\Gamma I}(A^e_{II})^{-1}\mathbf{f}^e_I.$$

Hence, the reduced obstacle problem can be written as

$$\begin{aligned} \tilde{A}\tilde{\mathbf{u}} &\geqslant \tilde{\mathbf{f}}, \\ \tilde{\mathbf{u}} &\geqslant \tilde{\Psi}, \\ (\tilde{\mathbf{u}} - \tilde{\Psi})_i(\tilde{A}\tilde{\mathbf{u}} - \tilde{\mathbf{f}})_i &= 0, \end{aligned}$$

where

$$\tilde{A} = \begin{pmatrix} A^i_{II} & A^i_{I\Gamma} \\ A^i_{\Gamma I} & S^e \end{pmatrix}; \quad \tilde{\mathbf{u}} = \begin{pmatrix} \mathbf{u}^i_I \\ \mathbf{u}_\Gamma \end{pmatrix}; \quad \tilde{\mathbf{f}} = \begin{pmatrix} \mathbf{f}^i_I \\ \tilde{\mathbf{f}}_\Gamma \end{pmatrix}; \quad \text{and} \quad \tilde{\Psi} = \begin{pmatrix} \Psi^i_I \\ \Psi_\Gamma \end{pmatrix}.$$

We will see in algorithms (5.4.1.1) - (5.4.1.2) and (5.6.1.1)-(5.6.3.1) below how this reduced system can be solved more efficiently by splitting the system for A^i_{II} and S^e.

Minimization formulation of reduced obstacle problem

This reduced problem can be written as a minimization problem of the following form.

$$\begin{cases} \text{Find } \tilde{\mathbf{u}} \in \tilde{K} \text{ such that } \forall\ \mathbf{v} \in \tilde{K} = \{v \in \mathbb{R}^{N^i + N^\Gamma} \ : \ \mathbf{v} \geqslant \tilde{\Psi}\}, \\ \tilde{J}(\tilde{\mathbf{u}}) \leqslant \tilde{J}(\mathbf{v}). \end{cases} \tag{5.3.4}$$

where

$$\tilde{J}(\mathbf{v}) = \frac{1}{2}(\mathbf{v})^T \tilde{A}\mathbf{v} - (\mathbf{v})^T \tilde{\mathbf{f}}.$$

QP formulation of reduced obstacle problem

To solve this reduced obstacle problem we derive the QP formulation of the minimization problem (5.3.4), as described in Chapter 3. Thus, the quadratic programming formulation for the reduced minimization problem is equivalent to

$$QP(\tilde{A}, \tilde{\mathbf{f}}, I, \Psi) \begin{cases} \text{minimize} \quad \frac{1}{2}\tilde{\mathbf{u}}^T \tilde{A}\tilde{\mathbf{u}} - \tilde{\mathbf{f}}^T \tilde{\mathbf{u}}, \\ \text{subject to} \quad \tilde{\mathbf{u}} \ \geqslant \ \tilde{\Psi}, \end{cases} \tag{5.3.5}$$

where

$$\tilde{\mathbf{u}}^T = [(\mathbf{u}_I^i)^T, (\mathbf{u}_\Gamma)^T] \quad \text{and} \quad \tilde{\Psi}^T = [(\Psi_I)^T, (\Psi_\Gamma)^T].$$

5.4 Solution methods for the obstacle problem

We develop three algorithms to solve the obstacle problem. The first algorithm is a direct method in which we first calculate $\mathbf{u}_I^{e\{1\}}$ then solve the reduced system for the obstacle problem for $\mathbf{u}_I^i$ and $\mathbf{u}_\Gamma$ and in the third step we calculate $\mathbf{u}_I^{e\{2\}}$. The other algorithms we propose, are iterative methods to approximate the solutions $\mathbf{u}_I^i$ and $\mathbf{u}_\Gamma$. The algorithms are described as follows.

Algorithm 5.4.0.1 ***Reduced QP (RQP) direct algorithm***

1: ***step 1:*** *find* $\mathbf{u}_I^{e\{1\}} = (A_{II}^e)^{-1}\mathbf{f}_I^e$,

2: ***step 2:*** *find* $\tilde{\mathbf{u}} \in \tilde{K}$ *such that*

$$\tilde{J}(\tilde{\mathbf{u}}) \leqslant \tilde{J}(\mathbf{v}) \ \forall \ \mathbf{v} \in \tilde{K},$$

3: *where* $\tilde{\mathbf{u}}^T = [(\mathbf{u}_I^i)^T, (\mathbf{u}_\Gamma)^T]$, *and* $\tilde{J}(\mathbf{v}) = \frac{1}{2}(\mathbf{v})^T\tilde{A}\mathbf{v} - (\mathbf{v})^T\tilde{\mathbf{f}}$

4: ***step 3:*** *Compute*

$$\mathbf{u}_I^{e\{2\}} = -(A_{II}^e)^{-1}A_{I\Gamma}^e\mathbf{u}_\Gamma$$

5: *The resulting solution is then*

$$\mathbf{u} = [\mathbf{u}_I^{e\{1\}} + \mathbf{u}_I^{e\{2\}}, \mathbf{u}_I^i, \mathbf{u}_\Gamma].$$

In this algorithm, we see that the matrix for reduced inequality problem contains the partial Schur complement for the equality domain S^e as an element, which could make the algorithm computationally expensive. In the following we propose algorithms in which we approximated the reduced problem (5.3.4) in an iterative manner. In this algorithm we use two level refinement and the solution is calculated by using a coarse mesh to compute an initial guess.

5.4.1 Picard reduced QP algorithm

We proposed a Picard algorithm for solving the nonlinear problem corresponding to (5.2.19a) and (5.2.19b). In our Picard algorithm we will consider two cases, in the first case we solve for $\mathbf{u}_I^e$ on Ω^e, whereas we compute $\mathbf{u}_I^i$ on Ω^i and $\mathbf{u}_\Gamma$ on Γ by solving variational inequalities. In the second case we assume that solutions $\mathbf{u}_I^e$, $\mathbf{u}_\Gamma$ satisfy PDEs on Ω^e and Γ respectively. We solve a variational inequality only in the subdomain Ω^i. The proposed algorithms are included below;

Algorithm 5.4.1.1 ***Picard reduced QP algorithm I***

step 0: *Find an initial guess by using coarse mesh solution*

step 1: *find* $\mathbf{u}_I^{e\{1\}} = (A_{II}^e)^{-1}\mathbf{f}_I^e$,

step 2:

for $k = 0, 1, 2, ...,$ *till convergence* ***do***

Find $(\mathbf{u}_\Gamma)^{k+1} \in K_\Gamma$ *such that*

$$J((\mathbf{u}_\Gamma)^{k+1}) \leqslant J(\mathbf{v}) \quad \forall \mathbf{v} \in K_\Gamma$$

where

$$J(\mathbf{v}) := \frac{1}{2}\mathbf{v}^T S^e \mathbf{v} - \mathbf{v}^T\{\mathbf{f}_\Gamma - A_{\Gamma I}^e \mathbf{u}_I^{e\{1\}} - A_{\Gamma I}^i (\mathbf{u}_I^i)^k)\}$$

Find $(\mathbf{u}_I^i)^{k+1} \in K^i$ *such that*

$$J((\mathbf{u}_I^i)^{k+1}) \leqslant J(\mathbf{v}) \quad \forall \mathbf{v} \in K^i$$

where

$$J(\mathbf{v}) := \frac{1}{2}(\mathbf{v})^T A_{II}^i \mathbf{v} - (\mathbf{v})^T(\mathbf{f}_I^i - A_{I\Gamma}^i \mathbf{u}_\Gamma^{k+1})$$

If converged, set $\mathbf{u}_\Gamma = \mathbf{u}_\Gamma^{k+1}$ *and exit*

end for

step 3: *Compute*

$$\mathbf{u}_I^{e\{2\}} = -(A_{II}^e)^{-1} A_{I\Gamma}^e \mathbf{u}_\Gamma$$

The resulting solution is then

$$\mathbf{u} = [\mathbf{u}_I^{e\{1\}} + \mathbf{u}_I^{e\{2\}}, \mathbf{u}_I^i, \mathbf{u}_\Gamma].$$

Algorithm 5.4.1.2 ***Picard reduced QP algorithm II***

step 0: *Find an initial guess by using coarse mesh solution*

step 1: *find* $\mathbf{u}_I^{e\{1\}} = (A_{II}^e)^{-1}\mathbf{f}_I^e$,

step 2:

for $k = 0, 1, 2, ...,$ *till convergence* ***do***

  (i) Solve $S^e(\mathbf{u}_\Gamma)^{k+1} = (\mathbf{f}_\Gamma - A_{\Gamma I}^e \mathbf{u}_I^{e\{1\}} - A_{\Gamma I}^i (\mathbf{u}_I^i)^k)$

  (ii) Find $(\mathbf{u}_I^i)^{k+1} \in K^i$ *such that*

$$J((\mathbf{u}_I^i)^{k+1}) \leqslant J(\mathbf{v}) \quad \forall \mathbf{v} \in K^i$$

  where

$$J(\mathbf{v}) := \frac{1}{2}(\mathbf{v})^T A_{II}^i \mathbf{v} - (\mathbf{v})^T(\mathbf{f}_I^i - A_{I\Gamma}^i \mathbf{u}_\Gamma^{k+1})$$

  If converged, set $\mathbf{u}_\Gamma = \mathbf{u}_\Gamma^{k+1}$ *and exit*

end for

step 3: *Compute*

$$\mathbf{u}_I^{e\{2\}} = -(A_{II}^e)^{-1} A_{I\Gamma}^e \mathbf{u}_\Gamma$$

The resulting solution is then

$$\mathbf{u} = [\mathbf{u}_I^{e\{1\}} + \mathbf{u}_I^{e\{2\}}, \mathbf{u}_I^i, \mathbf{u}_\Gamma].$$

We described three algorithms to solve variational inequality problems. The first algorithm is a direct method which solves the system in one step. This method requires the construction of a Schur complement, as an element of a matrix to be solved. The algorithms (5.4.1.1) and (5.4.1.2) are iterative procedures. In these algorithms the solution on the interface and in the inequality region are obtained in an iterative manner.

5.5 Convergence analysis for Picard algorithm

In this section we give convergence analysis for Picard reduced QP algorithm.

Lemma 5.5.1 *Let* $(\tilde{\mathbf{u}})^{k+1}$ *is the solution of reduced minimization problem* (5.3.4) *at* $k+1$ *iteration, where*

$$\tilde{\mathbf{u}} = \begin{pmatrix} \mathbf{u}_I^i \\ \mathbf{u}_\Gamma \end{pmatrix}; \quad \tilde{\mathbf{e}} = \begin{pmatrix} \mathbf{e}_I^i \\ \mathbf{e}_\Gamma \end{pmatrix}; \quad \tilde{\mathbf{f}} = \begin{pmatrix} \mathbf{f}_I^i \\ \tilde{\mathbf{f}}_\Gamma \end{pmatrix};$$

Let $(\tilde{\mathbf{e}})^k = (\tilde{\mathbf{u}})^{k+1} - (\tilde{\mathbf{u}})^k,$ *Then*

$$J((\tilde{\mathbf{u}})^{k+1}) - J((\tilde{\mathbf{u}})^k) = \tilde{J}((\mathbf{e}_I^i)^{k+1}) + \tilde{J}((\mathbf{e}_\Gamma)^{k+1}),$$

where

$$\tilde{J}((\mathbf{e}_I^i)^{k+1}) = \frac{1}{2}((\mathbf{e}_I^i)^k)^T A_{II}^i (\mathbf{e}_I^i)^k - ((\mathbf{e}_I^i)^k)^T \{\mathbf{f}_I^i - A_{II}^i (\mathbf{u}_I^i)^k - A_{I\Gamma}^i (\mathbf{u}_\Gamma^{k+1})\}$$

and

$$\tilde{J}((\mathbf{e}_\Gamma)^{k+1}) = \frac{1}{2}((\mathbf{e}_\Gamma)^k)^T S^e (\mathbf{e}_\Gamma)^k - ((\mathbf{e}_\Gamma)^k)^T \{\tilde{\mathbf{f}}_\Gamma - S^e \mathbf{u}_\Gamma^k - A_{\Gamma I}^i (\mathbf{u}_I^i)^k\}.$$

Proof

Consider

$$
\begin{aligned}
J((\tilde{\mathbf{u}}^{k+1}) - J(\tilde{\mathbf{u}}^k) &= J(\mathbf{e}^k + \tilde{\mathbf{u}}^k) - J(\tilde{\mathbf{u}}^k) \\
&= \frac{1}{2}\{(\mathbf{e}^k + \tilde{\mathbf{u}}^k)^T \tilde{A}(\mathbf{e}^k + \tilde{\mathbf{u}}^k)\} - (\mathbf{e}^k + \tilde{\mathbf{u}}^k)^T \tilde{\mathbf{f}} \\
&- \frac{1}{2}(\tilde{\mathbf{u}}^k)^T \tilde{A}\tilde{\mathbf{u}}^k + (\tilde{\mathbf{u}}^k)^T \tilde{\mathbf{f}} \\
&= (\mathbf{e}^k)^T \tilde{A}(\frac{1}{2}\mathbf{e}^k + \tilde{\mathbf{u}}^k) - (\mathbf{e}^k)^T \tilde{\mathbf{f}} \\
&= \begin{pmatrix}((\mathbf{e}_I^i)^k)^T & ((\mathbf{e}_\Gamma)^k)^T\end{pmatrix} \begin{pmatrix} A_{II}^i & A_{I\Gamma}^i \\ A_{\Gamma I}^i & S^e \end{pmatrix} \left(\frac{1}{2}\mathbf{e}^k + \tilde{\mathbf{u}}^k\right) - (\mathbf{e}^k)^T \begin{pmatrix} \mathbf{f}_I^i \\ \tilde{\mathbf{f}}_\Gamma \end{pmatrix} \\
&= \frac{1}{2}((\mathbf{e}_I^i)^k)^T A_{II}^i((\mathbf{e}_I^i)^k - ((\mathbf{e}_I^i)^k)^T\{\mathbf{f}_I^i - A_{II}^i(\mathbf{u}_I^i)^k - A_{I\Gamma}^i((\mathbf{e}_\Gamma)^k + \mathbf{u}_\Gamma)^k\} \\
&+ \frac{1}{2}((\mathbf{e}_\Gamma)^k)^T S^e(\mathbf{e}_\Gamma)^k) - ((\mathbf{e}_\Gamma)^k)^T\{\tilde{\mathbf{f}}_\Gamma - S^e\mathbf{u}_\Gamma^k - A_{\Gamma I}^i(\mathbf{u}_I^i)^k\} \\
&= \frac{1}{2}((\mathbf{e}_I^i)^k)^T A_{II}^i((\mathbf{e}_I^i)^k - ((\mathbf{e}_I^i)^k)^T\{\mathbf{f}_I^i - A_{II}^i(\mathbf{u}_I^i)^k - A_{I\Gamma}^i(\mathbf{u}_\Gamma^{k+1})\} \\
&+ \frac{1}{2}(\mathbf{e}_\Gamma^k)^T S^e(\mathbf{e}_\Gamma)^k - ((\mathbf{e}_\Gamma)^k)^T\{\tilde{\mathbf{f}}_\Gamma - S^e\mathbf{u}_\Gamma^k - A_{\Gamma I}^i(\mathbf{u}_I^i)^k\} \\
&= \tilde{J}((\mathbf{e}_I^i)^{k+1}) + \tilde{J}((\mathbf{e}_\Gamma)^{k+1}).
\end{aligned}
$$

Theorem 5.5.2 *If* $\tilde{J}((\mathbf{e}_I^i)^{k+1})$ *and* $\tilde{J}((\mathbf{e}_\Gamma)^{k+1})$, *are defined as in above lemma then*

$$J((\tilde{\mathbf{u}}^{k+1}) \leqslant J(\tilde{\mathbf{u}}^k).$$

Therefore the sequence $(\tilde{\mathbf{u}}^k)$, $k \in \mathbb{N}$, *generated by the above algorithms converges to the solution of the reduced minimization problem* (5.3.4).

Proof

Consider

$$
\begin{aligned}
\tilde{J}((\mathbf{e}_I^i)^{k+1}) &= ((\mathbf{e}_I^i)^k)^T A_{II}^i (\mathbf{e}_I^i)^k - ((\mathbf{e}_I^i)^k)^T \{\mathbf{f}_I^i - A_{II}^i(\mathbf{u}_I^i)^k - A_{I\Gamma}^i(\mathbf{u}_\Gamma^{k+1})\} - \frac{1}{2}((\mathbf{e}_I^i)^k)^T A_{II}^i((\mathbf{e}_I^i)^k \\
&= ((\mathbf{u}_I^i)^{k+1} - (\mathbf{u}_I^i)^k)^T \ A_{II}^i \ \left((\mathbf{u}_I^i)^{k+1} - (\mathbf{u}_I^i)^k\right) - ((\mathbf{u}_I^i)^{k+1} - (\mathbf{u}_I^i)^k)^T \left(\tilde{\mathbf{f}}_I^i - A_{II}^i(\mathbf{u}_I^i)^k\right) \\
&- \frac{1}{2}((\mathbf{e}_I^i)^k)^T A_{II}^i((\mathbf{e}_I^i)^k \\
&= ((\mathbf{u}_I^i)^{k+1})^T \left(A_{II}^i((\mathbf{u}_I^i)^{k+1} - \tilde{\mathbf{f}}_I^i\right) - ((\mathbf{u}_I^i)^k)^T \left(A_{II}^i((\mathbf{u}_I^i)^{k+1} - \tilde{\mathbf{f}}_I^i\right) \\
&- \frac{1}{2}((\mathbf{e}_I^i)^k)^T A_{II}^i((\mathbf{e}_I^i)^k \\
&= ((\mathbf{u}_I^i)^{k+1})^T \left(A_{II}^i((\mathbf{u}_I^i)^{k+1} - \tilde{\mathbf{f}}_I^i\right) - ((\Psi_I^i))^T \left(A_{II}^i((\mathbf{u}_I^i)^{k+1} - \tilde{\mathbf{f}}_I^i\right) \\
&+ ((\Psi_I^i))^T \left(A_{II}^i((\mathbf{u}_I^i)^{k+1} - \tilde{\mathbf{f}}_I^i\right) - ((\mathbf{u}_I^i)^k)^T \left(A_{II}^i((\mathbf{u}_I^i)^{k+1} - \tilde{\mathbf{f}}_I^i\right) \\
&- \frac{1}{2}((\mathbf{e}_I^i)^k)^T A_{II}^i((\mathbf{e}_I^i)^k \\
&= ((\mathbf{u}_I^i)^{k+1} - (\Psi_I^i))^T \left(A_{II}^i((\mathbf{u}_I^i)^{k+1} - \tilde{\mathbf{f}}_I^i\right) - ((\mathbf{u}_I^i)^k - (\Psi_I^i))^T \left(A_{II}^i((\mathbf{u}_I^i)^{k+1} - \tilde{\mathbf{f}}_I^i\right) \\
&- \frac{1}{2}((\mathbf{e}_I^i)^k)^T A_{II}^i((\mathbf{e}_I^i)^k
\end{aligned}
$$

since $(\mathbf{u}_I^i)^{k+1}$ *is the solution of minimization problem in subdomain* Ω^i *so*

$$((\mathbf{u}_I^i)^{k+1} - (\Psi_I^i))^T \left(A_{II}^i((\mathbf{u}_I^i)^{k+1} - \tilde{\mathbf{f}}_I^i\right) = 0.$$

By using the symmetric and positive definite property of A_{II}^i*, we get*

$$\tilde{J}((\mathbf{e}_I^i)^{k+1}) = -((\mathbf{u}_I^i)^k - (\Psi_I^i))^T \left(A_{II}^i((\mathbf{u}_I^i)^{k+1} - \tilde{\mathbf{f}}_I^i\right) - \frac{1}{2}((\mathbf{e}_I^i)^k)^T A_{II}^i((\mathbf{e}_I^i)^k \leqslant 0.$$

Similarly, we can prove

$$\tilde{J}((\mathbf{e}_\Gamma)^{k+1}) = -((\mathbf{u}_\Gamma)^k - (\Psi_\Gamma))^T \left(S^e((\mathbf{u}_\Gamma)^{k+1} - \tilde{\mathbf{f}}_\Gamma\right) - \frac{1}{2}((\mathbf{e}_I^i)^k)^T S^e((\mathbf{e}_I^i)^k \leqslant 0.$$

Thus, we have,

$$J((\tilde{\mathbf{u}}^{k+1}) - J(\tilde{\mathbf{u}}^k) = -\frac{1}{2}((\mathbf{e}_I^i)^k)^T A_{II}^i((\mathbf{e}_I^i)^k - ((\mathbf{u}_I^i)^k - (\Psi_I^i))^T \left(A_{II}^i((\mathbf{u}_I^i)^{k+1} - \tilde{\mathbf{f}_I^i}\right)$$
$$- ((\mathbf{u}_\Gamma)^k - (\Psi_\Gamma))^T \left(S^e(\mathbf{u}_\Gamma)^{k+1} - \tilde{\mathbf{f}_\Gamma}\right),$$

which implies that

$$J((\tilde{\mathbf{u}}^{k+1}) - J(\tilde{\mathbf{u}}^k) \leqslant 0,$$

and hence

$$J((\tilde{\mathbf{u}}^{k+1}) \leqslant J(\tilde{\mathbf{u}}^k).$$

This implies that $\{J(\tilde{\mathbf{u}}^k)\}$ *converges, since* $\tilde{A}$ *is symmetric and positive definite and the functional* J *is continuous, the sequence* $\left(\tilde{\mathbf{u}}^k\right), k \in \mathbb{N}$ *is bounded and converges to* $\tilde{\mathbf{u}}$, *the solution of the minimization problem* (5.3.4).

Corollary 5.5.3 *Let* $(\mathbf{u})^{k+1}$ *be the solution of the global minimization* (5.3.1) *at the* $k+1$ *iteration. Let* $\mathbf{u} = \begin{pmatrix} \mathbf{u}_I^e \\ \mathbf{u}_I^i \\ \mathbf{u}_\Gamma \end{pmatrix}$, $\mathbf{e} = \begin{pmatrix} \mathbf{e}_I^e \\ \mathbf{e}_I^i \\ \mathbf{e}_\Gamma \end{pmatrix}$, $\mathbf{f} = \begin{pmatrix} \mathbf{f}_I^e \\ \mathbf{f}_I^i \\ \mathbf{f}_\Gamma \end{pmatrix}$ *and* $\mathbf{e}^k = (\mathbf{u})^{k+1} - (\mathbf{u})^k$, *then*

$$J(\mathbf{u}^{k+1}) \leqslant J(\mathbf{u}^k),$$

therefore the sequence $(\mathbf{u}^k)$, $\; k \in \mathbb{N}$ *converges to the solution of the global minimization problem* (5.3.1).

Proof

From lemma (5.5.1) *it is easy to show that*

$$J(\mathbf{u}^{k+1}) - J(\mathbf{u}^k) \leqslant \tilde{J}(\mathbf{e}_I^e)^k) + \tilde{J}(\mathbf{e}_I^i)^k) + \tilde{J}(\mathbf{e}_\Gamma)^k)$$

and we can prove

$$\tilde{J}(\mathbf{e}_I^e)^k) \leqslant 0.$$

Thus we have

$$J(\mathbf{u}^{k+1}) \leqslant J(\mathbf{u}^k),$$

and therefore $\{J(\mathbf{u}^k)\}$ *converges, since* A *is symmetric and positive definite and the functional* J *is continuous so* $(\mathbf{u}^k),\ k \in \mathbb{N}$ *is bounded and converges to* $\mathbf{u}$, *the solution of the global minimization problem* (5.3.1).

5.6 Newton's method for the nonlinear interface problem

In these methods, we solve the nonlinear interface problem in step 2(i) of the algorithm (5.4.1.2) by using Newton's method and then coupled with the PDI, in step 2(ii) of (5.4.1.2) iteratively.

Newton-Krylov methods and Krylov-Schwarz (domain decomposition) methods have been considered in many applications for the last few decades. The first method, is a Krylov method inside of Newton's method in a Jacobian-free manner, through directional differencing. The second one is an overlapping Schwarz domain decomposition to derive a preconditioner for the Krylov accelerator that relies primarily on local information, for data-parallel concurrency. These methods seem particularly well suited for solving nonlinear elliptic systems in high-latency, distributed-memory environments. Some Newton-Krylov and Krylov-Schwarz methods are discussed in [21].

The domain decomposition method we described in this chapter give rise to a non linear partial Steklov Poincaré operator which requires the solution of a nonlinear interface problem given by (5.2.19b). In the following algorithms we apply the Newton method and Newton-GMRES methods at interface, for non-linear elliptic problems. Recall the interface problem

$$S^e \mathbf{u}_\Gamma = \mathbf{f}_\Gamma - A^e_{\Gamma I}\mathbf{u}_I^{e\{1\}} - A^i_{\Gamma I}\mathbf{u}^i_I(\mathbf{u}_\Gamma).$$

Let

$$g(\mathbf{u}_\Gamma) = \mathbf{f}_\Gamma - A^e_{\Gamma I}\mathbf{u}_I^{e\{1\}} - A^i_{\Gamma I}\mathbf{u}^i_I(\mathbf{u}_\Gamma)$$

Then the above interface problem for $\mathbf{u}_\Gamma$ can be written as

$$F(\mathbf{u}_\Gamma) = S^e \mathbf{u}_\Gamma - g(\mathbf{u}_\Gamma) = 0 \tag{5.6.1}$$

Newton's method for $F(\mathbf{u}_\Gamma) = 0$ is

$$J\delta\mathbf{u}_\Gamma = -F(\mathbf{u}_\Gamma),$$

where J is the Jacobian given by

$$J = F_{\mathbf{u}_\Gamma}(\mathbf{u}_\Gamma) = S^e - \frac{\partial g(\mathbf{u}_\Gamma)}{\partial \mathbf{u}_\Gamma}. \tag{5.6.2}$$

For a small perturbation ε and an arbitrary direction $\mathbf{e}_j$ we have

$$\frac{\partial g(\mathbf{u}_\Gamma)}{\partial \mathbf{u}_\Gamma} = \frac{g(\mathbf{u}_\Gamma + \varepsilon\ \mathbf{e}_j) - g(\mathbf{u}_\Gamma)}{\varepsilon}$$

or

$$\frac{\partial g(\mathbf{u}_\Gamma)}{\partial \mathbf{u}_\Gamma} = -A^i_{\Gamma I}\frac{\mathbf{u}^i_I(\mathbf{u}_\Gamma + \varepsilon\ \mathbf{e}_j) - \mathbf{u}^i_I(\mathbf{u}_\Gamma)}{\varepsilon}$$

5.6.1 Newton reduced QP algorithm

Equations (5.2.19b) and (5.2.19c) form a coupled system which we solved previously, by using Picard iterations. In this algorithm we applied Newton's method for the solution of nonlinear interface problem. The proposed algorithm is included below;

Algorithm 5.6.1.1 ***Newton reduced QP algorithm***

step 0: *Find an initial guess by using coarse mesh solution*

step 1: *find* $\mathbf{u}_I^{e\{1\}} = (A_{II}^e)^{-1}\mathbf{f}_I^e$,

step 2:

for $k = 0, 1, 2, ...,$ *till convergence* ***do***

  compute Jacobian

  for $j = 1 : N_\Gamma$ *compute*

$$\frac{\partial g(\mathbf{u}_\Gamma^k(:,j))}{\partial \mathbf{u}_\Gamma^k} = -A_{\Gamma I}^i \frac{\mathbf{u}_I^i(\mathbf{u}_\Gamma^k + \varepsilon\ \mathbf{e}(:,j)) - \mathbf{u}_I^i(\mathbf{u}_\Gamma^k)}{\varepsilon}$$

  end

$$J^k = S^e + \frac{\partial g(\mathbf{u}_\Gamma^k)}{\partial \mathbf{u}_\Gamma^k}$$

  solve

$$J^k(\delta \mathbf{u}_\Gamma^k) = -F(\mathbf{u}_{\Gamma^k}).$$

  Set $(\mathbf{u}_\Gamma)^{k+1} = (\mathbf{u}_\Gamma)^k - \delta \mathbf{u}_\Gamma^k)$

  find $(\mathbf{u}_I^i)^{k+1} \in K^i$ *such that*

$$J((\mathbf{u}_I^i)^{k+1}) \leqslant J(\mathbf{v}), \quad \forall \mathbf{v} \in K^i$$

  where

$$J(\mathbf{v}) := \frac{1}{2}(\mathbf{v})^T A_{II}^i \mathbf{v} - (\mathbf{v})^T(\mathbf{f}_I^i - A_{I\Gamma}^i \mathbf{u}_\Gamma^{k+1})$$

  If converged, set $\mathbf{u}_\Gamma = \mathbf{u}_\Gamma^{k+1}$ *and exit*

end for

step 3: *Solve*

$$\mathbf{u}_I^{e\{2\}} = -(A_{II}^e)^{-1} A_{I\Gamma}^e \mathbf{u}_\Gamma$$

The resulting solution is then

$$\mathbf{u} = [\mathbf{u}_I^{e\{1\}} + \mathbf{u}_I^{e\{2\}}, \mathbf{u}_I^i, \mathbf{u}_\Gamma].$$

It is important to note that in Picard algorithm we solvethe interface problem through the direct inversion of the matrix S^e. On the other hand Newton's reduced QP algorithm required the construction of Jacobian matrix, which contains an extra term together with S^e, given by

$$J^k = S^e + \frac{\partial g(\mathbf{u}_\Gamma^k)}{\partial \mathbf{u}_\Gamma^k}.$$

Which increase the complexity of the Newton's algorithm. However, we will see in Chapter 7 that this algorithm exhibit rapid convergence. The complexity of this algorithm can be reduced by using GMRES. In this method the Jacobian is approximated by using matrix-vector products.

5.6.2 Newton-GMRES method with exact Jacobian

In this method we construct the Jacobian matrix in Newton step given by (5.6.2)

$$J = S^e + \frac{\partial g(\mathbf{u}_\Gamma)}{\partial \mathbf{u}_\Gamma}.$$

To find the solution of the non-linear interface problem

$$J^k \delta \mathbf{u}_\Gamma = -F(\mathbf{u}_\Gamma),$$

we employ GMRES [99]. The Newton-GMRES algorithm with exact Jacobian is given below

Algorithm 5.6.2.1 ***Newton-GMRES with exact Jacobian algorithm***

step 0: *Find an initial guess by using coarse mesh solution*

step 1: *find* $\mathbf{u}_I^{e\{1\}} = (A_{II}^e)^{-1}\mathbf{f}_I^e$,

step 2:

for $k = 0, 1, 2, ...,$ *till convergence* ***do***

  compute Jacobian

  for $j = 1 : N_\Gamma$

$$\frac{\partial g(\mathbf{u}_\Gamma^k(:,j))}{\partial \mathbf{u}_\Gamma^k} = -A_{\Gamma I}^i \frac{\mathbf{u}_I^i(\mathbf{u}_\Gamma^k + \varepsilon\ \mathbf{e}(:,j) - \mathbf{u}_I^i(\mathbf{u}_\Gamma^k)}{\varepsilon}$$

$$J^k = S^e + \frac{\partial g(\mathbf{u}_\Gamma^k)}{\partial \mathbf{u}_\Gamma^k}$$

  end

  apply GMRES to compute

$$\delta\mathbf{u}_\Gamma^k = GMRES(J^k, -F(\mathbf{u}_\Gamma^k))$$

  set $(\mathbf{u}_\Gamma)^{k+1} = (\mathbf{u}_\Gamma)^k + \delta\mathbf{u}_\Gamma^k$

  find $(\mathbf{u}_I^i)^{k+1} \in K^i$ *such that*

$$J((\mathbf{u}_I^i)^{k+1}) \leqslant J(\mathbf{v}), \quad \forall \mathbf{v} \in K^i$$

  where

$$J(\mathbf{v}) := \frac{1}{2}(\mathbf{v})^T A_{II}^i \mathbf{v} - (\mathbf{v})^T(\mathbf{f}_I^i - A_{I\Gamma}^i \mathbf{u}_\Gamma^{k+1})$$

  If converged, set $\mathbf{u}_\Gamma = \mathbf{u}_\Gamma^{k+1}$ *and exit*

end for

step 3: *Solve*

$$\mathbf{u}_I^{e\{2\}} = -(A_{II}^e)^{-1} A_{I\Gamma}^e \mathbf{u}_\Gamma$$

The resulting solution is then
$$\mathbf{u} = [\mathbf{u}_I^{e\{1\}} + \mathbf{u}_I^{e\{2\}}, \mathbf{u}_I^i, \mathbf{u}_\Gamma].$$

5.6.3 Jacobian free Newton-GMRES method

Jacobian free Newton Krylov methods are the combinations of Newton methods and Krylov methods for solving the Newton correction equations. In these methods the Jacobian is approximated by using matrix-vector product, without constructing and storing the Jacobian as in Newton method section (5.6). The Jacobian equation given in (5.6.2) is approximated by using the following approximate matrix-vector product in GMRES [17] [22]

$$J\mathbf{v} \approx S^e\mathbf{v} + [g(\mathbf{u}_\Gamma + \varepsilon\ \mathbf{v}) - g(\mathbf{u}_\Gamma)]/\varepsilon, \tag{5.6.3}$$

where ε is an arbitrary perturbation.

The Jacobian free Newton GMRES method does not require the construction of J, we instead form a vector that approximates these matrices multiplied by a vector $\mathbf{v}$. $g(\mathbf{u}_\Gamma + \varepsilon\ v)$ in $\tilde{J}v$ is approximated by first order Taylor series expansion about u [72]. We see that GMRES requires the action of Jacobian on a vector $\mathbf{v}$. Thus by using this method we solve the nonlinear interface problem without constructing, storing and inverting the Jacobian matrix. The method is described in the following algorithm;

Algorithm 5.6.3.1 ***Jacobian free Newton GMRES algorithm***

step 0: *Find an initial guess by using coarse mesh solution*

step 1: *find* $\mathbf{u}_I^{e\{1\}} = (A_{II}^e)^{-1}\mathbf{f}_I^e$,

step 2:

for $k = 0, 1, 2, ...,$ *till convergence* ***do***

apply GMRES to compute $\delta\mathbf{u}_\Gamma^k$, *using matrix vector product*

$$\delta\mathbf{u}_\Gamma^k = GMRES(J^k, -F(\mathbf{u}_\Gamma^k))$$

set $(\mathbf{u}_\Gamma)^{k+1} = (\mathbf{u}_\Gamma)^k + \delta\mathbf{u}_\Gamma^k$

find $(\mathbf{u}_I^i)^{k+1} \in K^i$ *such that*

$$J((\mathbf{u}_I^i)^{k+1}) \leqslant J(\mathbf{v}), \quad \forall \mathbf{v} \in K^i$$

where

$$J(\mathbf{v}) := \frac{1}{2}(\mathbf{v})^T A_{II}^i \mathbf{v} - (\mathbf{v})^T(\mathbf{f}_I^i - A_{I\Gamma}^i \mathbf{u}_\Gamma^{k+1})$$

If converged, set $\mathbf{u}_\Gamma = \mathbf{u}_\Gamma^{k+1}$ *and exit*

end for

step 3: *Solve*

$$\mathbf{u}_I^{e\{2\}} = -(A_{II}^e)^{-1} A_{I\Gamma}^e \mathbf{u}_\Gamma$$

The resulting solution is then

$$\mathbf{u} = [\mathbf{u}_I^{e\{1\}} + \mathbf{u}_I^{e\{2\}}, \mathbf{u}_I^i, \mathbf{u}_\Gamma].$$

5.6.4 Preconditioning of Newton-GMRES algorithm (JFNG)

One of the most effective methods for solving large sparse linear systems is the combination of Krylov method with some suitable preconditioning. Preconditioning an iterative solver is a procedure of transforming the linear system into one which has the same solution whilst aiming to improve the both robustness and efficiency of the iterative algorithm. The purpose of preconditioning the JFNG method is to reduced the number of GMRES iterations, by efficiently clustering the eigen values of iteration matrix [98].

A preconditioner can be applied on the right, on the left, or on both sides. In context of the Newton's method, the preconditioned residual serves as a useful estimate of the size of the Newton correction, itself, when the preconditioning is of high quality. Both left or right preconditioning strategies, may be employed in a Jacobian-free context. Applying right preconditioning in Newton's method we obtain

$$(JP^{-1})(P\delta\mathbf{u}) = -F(\mathbf{u}).$$

Where P represents the preconditioning matrix and P^{-1} the inverse of the preconditioning matrix. Thus, the right-preconditioning of (5.6.3) yields

$$JP^{-1}\mathbf{v} \approx S^e\mathbf{v} + [g(\mathbf{u}_\Gamma + \varepsilon P^{-1}\mathbf{v}) - g(\mathbf{u}_\Gamma)]/\varepsilon, \tag{5.6.4}$$

where ε is an arbitrary perturbation [72].

In Chapter 7, we will present some result using S^e as preconditioner in algorithm (5.6.3.1) to reduce the GMRES iterations.

In this chapter we described domain decomposition methods for elliptic variational inequalities. In our method, we decomposed the domain into two multiply connected domains Ω^e, in which we solve a PDE and Ω^i, which requires a mild assumption about the coincidence set $\mathcal{C}$, and in which we solve a PDI. We also presented some algorithms to solve the obstacle problem by using domain decomposition methods. The proposed algorithms include the reduced QP direct algorithm in which the problem is solved in three steps, which is a direct procedure. In comparison the Picard reduced QP algorithms are iterative procedure in which the subproblems in subdomain Ω^i and on $\varGamma$ are solved iteratively. Finally, we solve the nonlinear problem at the interface $\varGamma$, using algorithms which employ Newton's method, Newton-GMRES method and preconditioned Newton-GMRES method. In Chapter 7, we will present numerical results to validate and study the behavior of these algorithms.

Chapter 6

Domain Decomposition Method For Parabolic Variational Inequalities

Domain decomposition method is mostly devoted to the area of elliptic problems. A number of references could be found detailing the applications of domain decomposition methods to elliptic problems. In comparison, there are few references that considered the applications of domain decomposition methods to parabolic problems. In particular [119], [51], [86], [77], [78], [36] discussed the extension of domain decomposition methods to parabolic problems.

Some domain decomposition methods for bilateral obstacles, described in the form of parabolic variational inequalities, are considered in [130]. A domain decomposition method for a kind of parabolic variational inequality of fourth order is given in [25]. In this chapter, we extend the domain decomposition methods (DDM) described in Chapter 5 to parabolic variational inequalities. We present a non overlapping domain decomposition method for the system of algebraic equations resulting from the finite element and finite difference approximations corresponding to the space and time variables, respectively. For parabolic variational inequalities, we apply domain decomposition method such that, at each time level we convert our problem into two subproblems, one of which is a PDE in subdomain Ω^e

and other is a variational inequality in the complementary subdomain Ω^i. Each problem is then solved in each subdomain and solutions are coupled together through a non-linear interface problem. To implement our method we consider the parabolic obstacle problem.

6.1 Formulation of the problem

Let Ω be a smooth bounded open convex domain with boundary $\partial\Omega$. Recalling the parabolic variational inequality problem described in Chapter 4;

$$\begin{cases} \dfrac{\partial u}{\partial t} - \mathcal{L}u \geqslant f, \\ u(x,t) \ \geqslant \ \psi(x,t) \ \text{ in } \Omega\times]0,T[, \\ (u-\psi)\left(\dfrac{\partial u}{\partial t} - \ \mathcal{L}u\right) \ = \ 0, \\ u(x,t) \ = \ 0 \ \text{ on } \Gamma \times [0,T], \\ u(x,0) \geqslant \psi(x,0) \quad \text{on } \Omega, \text{ at } t=0, \end{cases}$$

Matrix formulation for fully discrete problem at each time level k can be written as

$$\begin{cases} \widehat{A}\mathbf{u}^k \geqslant \widehat{\mathbf{f}}^k, \\ \mathbf{u}^k \geqslant \Psi^k, \\ (\mathbf{u}^k - \Psi^k)_i(\widehat{A}\mathbf{u}^k - \widehat{\mathbf{f}})_i = 0 \ \ \forall\ i, \end{cases} \tag{6.1.1}$$

where $\widehat{A} = \dfrac{M}{\Delta t_k} + (1+\theta)L$ and $\widehat{\mathbf{f}} = \dfrac{M}{\Delta t_k} + \mathbf{f} - \theta L$. For $\theta = 0$, the scheme is backward Euler and for $\theta = 1/2$, the scheme is Crank-Nicolson.

6.1.1 Domain decomposition method

To obtain the domain decomposition method for the problem (6.1.1), let us denote the unknowns vectors by $\mathbf{u}^e, \mathbf{u}^i, \mathbf{u}^\Gamma$ and the right hand side vectors by $\mathbf{f}^e, \mathbf{f}^i, \mathbf{f}_\Gamma$ of lengths N^e, N^i, N^Γ in the corresponding subdomains Ω^e, Ω^i, and interface Γ, respectively, such that $N = N^e + N^i + N^\Gamma$, with $A \in \mathbb{R}^{N\times N}$ and $\mathbf{f} \in \mathbb{R}^N$. Let $t = t_0 < t_1 < ...t_m = T$ be the subdivision of the interval $[0, T]$, and let $\Delta t_k = t_k - t_{k-1}$ be the time step. Then the matrix representation of (5.2.1) at each each time level, $k = 0, ...t_m - 1$, can be written as

$$\widehat{A}\mathbf{u}^k = \begin{pmatrix} \widehat{A}^e_{II} & O & \widehat{A}^e_{I\varGamma} \\ O & \widehat{A}^i_{II} & \widehat{A}^i_{I\varGamma} \\ \widehat{A}^e_{\varGamma I} & \widehat{A}^i_{\varGamma I} & \widehat{A}_{\varGamma\varGamma} \end{pmatrix} \begin{pmatrix} (\mathbf{u}^e_I)^k \\ (\mathbf{u}^i_I)^k \\ (\mathbf{u}_\varGamma)^k \end{pmatrix} \geqslant \begin{pmatrix} (\widehat{\mathbf{f}}^e_I)^k \\ (\widehat{\mathbf{f}}^i_I)^k \\ (\widehat{\mathbf{f}}_\varGamma)^k \end{pmatrix} = \widehat{\mathbf{f}}^k, \tag{6.1.2}$$

where we have partitioned the degree of freedom into those internal to Ω^e and to Ω^i and those on the interface boundary $\varGamma$. In particular, we have

$$\widehat{A}^e_{II} = \widehat{L}^e_{II} + \widehat{M}^e_{II},$$

$$\widehat{A}^i_{II} = \widehat{L}^i_{II} + \widehat{M}^i_{II},$$

$$\widehat{A}_{\varGamma\varGamma} = \widehat{L}_{\varGamma\varGamma} + \widehat{M}_{\varGamma\varGamma},$$

$$\widehat{A}^e_{I\varGamma} = \widehat{L}^e_{I\varGamma} + \widehat{M}^e_{I\varGamma},$$

$$\widehat{A}^i_{I\varGamma} = \widehat{L}^i_{I\varGamma} + \widehat{M}^i_{I\varGamma}.$$

Where $\widehat{L} = (1+\theta)L$, $\widehat{M} = \dfrac{M}{\Delta t_k}$ and $\widehat{\mathbf{f}} = \mathbf{f} + \dfrac{M}{\Delta t_k} - \theta L$, $\theta = 0, 1/2$. By using these notations, at each time step k, we have the following matrix form

$$\widehat{A}^e_{II}(\mathbf{u}^{e\{1\}}_I)^k = (\widehat{\mathbf{f}}^e_I)^k, \tag{6.1.3a}$$

$$\widehat{S}^e(\mathbf{u}_\varGamma)^k = (\widehat{\mathbf{f}}_\varGamma)^k - \widehat{A}^e_{\varGamma I}(\mathbf{u}^{e,1}_I)^k - \widehat{A}^i_{\varGamma I}(\mathbf{u}^i_I)^k, \tag{6.1.3b}$$

$$\widehat{A}^i_{II}(\mathbf{u}^i_I)^k \geqslant (\widehat{\mathbf{f}}^i_I)^k - \widehat{A}^i_{I\varGamma}(\mathbf{u}_\varGamma)^k, \tag{6.1.3c}$$

$$\widehat{A}^e_{II}(\mathbf{u}^{e\{2\}}_I)^k = -\widehat{A}^e_{I\varGamma}(\mathbf{u}_\varGamma)^k, \tag{6.1.3d}$$

subject to conditions $((\widehat{\mathbf{f}}^i_I)^k - \widehat{A}^i_{II}(\mathbf{u}^i_I)^k - \widehat{A}^i_{I\Gamma}(\mathbf{u}_\Gamma)^k)_j((\mathbf{u}^i_I)^k - (\boldsymbol{\psi}_I)^k)_j = 0$, which represent the complementarity conditions for (6.1.3c). The Partial Schur complement, $\widehat{S}^e$, is given by

$$\widehat{S}^e := \widehat{A}_{\Gamma\Gamma} - \widehat{A}^e_{\Gamma I}(\widehat{A}^e_{II})^{-1}\widehat{A}^e_{I\Gamma}.$$

The set of equations (6.1.3a)-(6.1.3d) could be seen as a partial Schur complement approach for the system (6.1.1). At each time level solutions $\mathbf{u}^i_I$ and $\mathbf{u}_\Gamma$ will be approximated in an iterative manner. The resulting solution at each time step k, is then $[(\mathbf{u}^{e\{1\}}_I + \mathbf{u}^{e\{2\}}_I)^k, (\mathbf{u}^i_I)^k, (\mathbf{u}_\Gamma)^k]$.

6.2 The Parabolic Obstacle problem

To demonstrate the implementation of the domain decomposition method for parabolic variational inequalities we will consider a well known example: the obstacle problem. The parabolic obstacle problem is defined as; find u($\mathbf{x}$,t) such that

$$\begin{cases} \dfrac{\partial u(\mathbf{x},t)}{\partial t} - \Delta u(\mathbf{x},t) \geqslant f \ \text{ in } \ \Omega, \\ u(\mathbf{x},t) - \psi(\mathbf{x},t) \quad \geqslant \quad 0 \ \text{ in } \ \Omega, \\ (u(\mathbf{x},t) - \psi(\mathbf{x},t))(-\Delta u(\mathbf{x},t) - f(\mathbf{x},t)) = 0 \ \text{ in } \ \Omega, \end{cases} \tag{6.2.1}$$

In matrix form it can be written as

$$\begin{cases} \widehat{A}\mathbf{u} \geqslant \widehat{\mathbf{f}}, \\ \mathbf{u} \geqslant \Psi, \\ (\mathbf{u} - \Psi)_i(\widehat{A}\mathbf{u} - \widehat{\mathbf{f}})_i = 0 \ \ \forall\, i, \end{cases} \tag{6.2.2}$$

where $\widehat{A} = \dfrac{M}{\Delta t_k} + (1-\theta)L$, $\Psi_i = \psi(\mathbf{x}_i)$ and $\widehat{\mathbf{f}} = \mathbf{f} + \dfrac{M}{\Delta t_k} - \theta L$. The solution method for (6.1.3a)-(6.1.3d) described here is based on the Schur complement approach and can be applied to the above system for the obstacle problem. As in the elliptic case, the domain decomposition method for the parabolic problem, also leads at each time to a PDE in Ω^e corresponding to the first row block in (6.1.2), the second row block, is a reduced variational inequality in Ω^i and the third row block can be considered as either a PDE or a PDI on $\mathit{\Gamma}$. We can write the PDE corresponding to Ω^e as

$$\widehat{A}^e_{II}(\mathbf{u}^e_I)^k + \widehat{A}^e_{I\Gamma}(\mathbf{u}_\Gamma)^k = (\widehat{\mathbf{f}}^e_I)^k.$$

Solving for $(\mathbf{u}^e)^k$ we get

$$(\mathbf{u}^e_I)^k = (\widehat{A}^e_{II})^{-1}((\widehat{\mathbf{f}}^e_I)^k - \widehat{A}^e_{I\Gamma}(\mathbf{u}_\Gamma)^k). \tag{6.2.3}$$

Replacing $(\mathbf{u}^e)^k$ in the remaining system we get the reduced system for obstacle problems i.e only for the sub-domains contain the support of the obstacle.

$$\begin{pmatrix} \widehat{A}^i_{II} & \widehat{A}^i_{I\Gamma} \\ \widehat{A}^i_{\Gamma I} & \widehat{S}^e \end{pmatrix} \begin{pmatrix} (\mathbf{u}^i_I)^k \\ (\mathbf{u}_\Gamma)^k \end{pmatrix} \geqslant \begin{pmatrix} (\widehat{\mathbf{f}}^i_I)^k \\ (\tilde{\widehat{\mathbf{f}}}_\Gamma)^k \end{pmatrix},$$

where

$$\widehat{S}^e := \widehat{A}_{\Gamma\Gamma} - \widehat{A}^e_{\Gamma I}(\widehat{A}^e_{II})^{-1}\widehat{A}^e_{I\Gamma} \quad \text{and} \quad \tilde{\mathbf{f}}_\Gamma := \widehat{\mathbf{f}}_\Gamma - \widehat{A}^e_{\Gamma I}(\widehat{A}^e_{II})^{-1}\widehat{\mathbf{f}}^e_I.$$

Hence, the reduced obstacle problem can be written as

$$\begin{aligned} \tilde{\widehat{A}}(\tilde{\mathbf{u}})^k &\geqslant (\tilde{\widehat{\mathbf{f}}})^k, \\ (\tilde{\mathbf{u}})^k &\geqslant (\tilde{\Psi})^k, \\ ((\tilde{\mathbf{u}})^k - (\tilde{\Psi})^k)_i(\tilde{\widehat{A}}(\tilde{\mathbf{u}})^k - (\tilde{\widehat{\mathbf{f}}})^k)_i &= 0, \end{aligned}$$

where

$$\widetilde{\widehat{A}} = \begin{pmatrix} \widehat{A}^i_{II} & \widehat{A}^i_{I\Gamma} \\ \widehat{A}^i_{\Gamma I} & \widehat{S}^e \end{pmatrix}; \quad \tilde{\mathbf{u}} = \begin{pmatrix} \mathbf{u}^i_I \\ \mathbf{u}_\Gamma \end{pmatrix}; \quad \widetilde{\widehat{\mathbf{f}}} = \begin{pmatrix} \widehat{\mathbf{f}}^i_I \\ \widetilde{\widehat{\mathbf{f}}}_\Gamma \end{pmatrix}; \quad \text{and} \quad \tilde{\Psi} = \begin{pmatrix} \Psi_I \\ \Psi_\Gamma \end{pmatrix}.$$

We will see in algorithms (6.3.1.1) and (6.4.0.2)-(6.4.0.4) below how this reduced system can be solved more efficiently by splitting the system for $\mathbf{u}^i_I$ and $\mathbf{u}_\Gamma$, containing the partial Schur complement $\widehat{S}^e$.

Minimization formulation of the reduced obstacle problem

This reduced problem can be written as a minimization problem of the following form.

$$\begin{cases} \text{for} \quad k = 0, 1, ..., t_m - 1 \\ \text{Find } \tilde{\mathbf{u}} \in \tilde{K} \text{ such that } \forall\ \mathbf{v} \in \tilde{K} = \{v \in \mathbb{R}^{N^i + N^\Gamma} \ : \ \mathbf{v} \geqslant \tilde{\Psi}\} \\ \widehat{J}(\tilde{\mathbf{u}}) \leqslant \widehat{J}(\mathbf{v}). \end{cases} \tag{6.2.4}$$

where

$$\widehat{J}(\mathbf{v}) = \frac{1}{2}(\mathbf{v})^T \widetilde{\widehat{\mathbf{A}}}\mathbf{v} - (\mathbf{v})^T \widetilde{\widehat{\mathbf{f}}}.$$

QP formulation of reduced obstacle problem

To solve this reduced obstacle problem we derive the QP formulation of the minimization problem (6.2.4), as described in Chapter 4 Thus the quadratic programming formulation for the reduced minimization problem is equivalent to

$$QP(\widetilde{\widehat{A}}, \widetilde{\widehat{\mathbf{f}}}, I, \Psi) \begin{cases} \text{for} \quad k = 0, 1, ..., t_m - 1 \\ \text{minimize} \quad \frac{1}{2}\tilde{\mathbf{u}}^T \widetilde{\widehat{A}} \tilde{\mathbf{u}} - \widetilde{\widehat{\mathbf{f}}}^T \tilde{\mathbf{u}}, \\ \text{subject to} \quad \tilde{\mathbf{u}} \ \geqslant \ \tilde{\Psi}, \end{cases} \tag{6.2.5}$$

where

$$\tilde{\mathbf{u}}^T = [(\mathbf{u}_I^i)^T, (\mathbf{u}_\Gamma)^T] \quad \text{and} \quad \tilde{\Psi}^T = [(\Psi_I)^T, (\Psi_\Gamma)^T].$$

6.3 DD methods for the parabolic obstacle problem

We develop three algorithms to solve the obstacle problem. The first algorithm is a direct method in which we first calculate $(\mathbf{u}_I^{e\{1\}})^k$ then solve the reduced system for the obstacle problem for $(\mathbf{u}_I^i)^k$ and $(\mathbf{u}_\Gamma)^k$ and in third step we calculate $(\mathbf{u}_I^{e\{2\}})^k$. The other two algorithms we propose, are iterative methods to approximate the solutions $(\mathbf{u}_I^i)^k$ and $(\mathbf{u}_\Gamma)^k$. The algorithms are described as follows

Algorithm 6.3.0.1 ***Reduced QP (RQP) direct algorithm***

1: ***step 1:*** *set*

$$\widehat{A} = \frac{M}{\Delta t_k} + (1+\theta)L$$

$$\widehat{\mathbf{f}}^{k+1} = \frac{M}{\Delta t_k}\mathbf{u}(:,k) + \mathbf{f}(:,k+1) - \theta L \text{ and } \Psi^{k+1} = \Psi(:,k+1)$$

2: ***for*** $k = 0, 1, 2, ..., t_m - 1$ ***do***

3: ***step 2:*** *solve algorithm* (5.4.0.1) *to find* $\mathbf{u}(:,k+1)$

4: ***step 3:****set* $\mathbf{u}(:,k) = \mathbf{u}(:,k+1)$ *and repeat.*

5: ***end for***

Where $\theta = 0$, for backward Euler and $\theta = 1/2$, for Crank-Nicolson. In the following we propose algorithms in which we approximated the reduced problem (6.2.4) in an iterative manner at each time level. In this algorithm we use two level refinement and the solution is calculated by using a coarse mesh to provide an initial guess.

6.3.1 Picard reduced QP algorithm

In this algorithm we assume that solutions $\mathbf{u}_I^e$, $\mathbf{u}_\Gamma$ satisfy PDEs on Ω^e and Γ respectively. We solve the variational inequality only in the subdomain Ω^i. The proposed algorithm is included below.

Algorithm 6.3.1.1 ***Picard reduced QP algorithm***

1: ***step 1:*** *set*

$$\widehat{A} = \frac{M}{\Delta t_k} + (1+\theta)L$$

$$\widehat{\mathbf{f}} = \frac{M}{\Delta t_k}\mathbf{u}(:,k) + \mathbf{f}(:,k+1) - \theta L \textit{ and } \Psi = \Psi(:,k+1)$$

2: ***for*** $k = 0, 1, 2, ..., t_{m-1}$ ***do***

3: ***step 2:*** *solve algorithm* (5.4.1.2) *to find* $\mathbf{u}(:,k+1)$

4: ***step 3:*** *set* $\mathbf{u}(:,k) = \mathbf{u}(:,k+1)$ *and repeat.*

5: ***end for***

Where $\theta = 0$, for backward Euler and $\theta = 1/2$, for Crank-Nicolson. We described two algorithms to solve variational inequality problems. The first algorithm is a direct method which solves the system in one step. This method requires the construction of a Schur complement, as an element of a matrix to be solved. The algorithm (6.3.1.1) is an iterative procedure. In this algorithm the solution to interface boundary and inequality region is calculated in an iterative manner.

6.4 Newton's method for the nonlinear interface problem

The domain decomposition method we described in this chapter give rise to a non linear partial Steklov Poincare operator which requires the solution of a nonlinear interface problem given by (6.1.3b) at each time step. In the following algorithms we apply Newton's method and Newton-GMRES methods at interface boundary which make the number of iterations independent of mesh size and give faster convergence.

Algorithm 6.4.0.2 ***Newton reduced QP algorithm***

1: ***step 1:*** *set*

$$\widehat{A} = \frac{M}{\Delta t_k} + (1+\theta)L$$

$$\widehat{\mathbf{f}}^{k+1} = \frac{M}{\Delta t_k}\mathbf{u}(:,k) + \mathbf{f}(:,k+1) - \theta L \text{ and } \Psi^{k+1} = \Psi(:,k+1)$$

2: ***for*** $k = 0, 1, 2, ..., t_{m-1}$ ***do***

3: ***step 2:*** *solve algorithm* (5.6.1.1) *to find* $\mathbf{u}(:,k+1)$

4: ***step 3:*** *set* $\mathbf{u}(:,k) = \mathbf{u}(:,k+1)$ *and repeat.*

5: ***end for***

Algorithm 6.4.0.3 ***Newton-GMRES with exact Jacobian algorithm***

1: ***step 1:*** *set*

$$\widehat{A} = \frac{M}{\Delta t_k} + (1+\theta)L$$

$$\widehat{\mathbf{f}}^{k+1} = \frac{M}{\Delta t_k}\mathbf{u}(:,k) + \mathbf{f}(:,k+1) - \theta L \text{ and } \Psi^{k+1} = \Psi(:,k+1)$$

2: ***for*** $k = 0, 1, 2, ..., t_m - 1$ ***do***

3: ***step 2:*** *solve algorithm* (5.6.2.1) *to find* $\mathbf{u}(:,k+1)$

4: ***step 3:*** *set* $\mathbf{u}(:,k) = \mathbf{u}(:,k+1)$ *and repeat.*

5: ***end for***

Algorithm 6.4.0.4 ***Jacobian free Newton-GMRES algorithm***

1: ***step 1:*** *set*

$$\widehat{A} = \frac{M}{\Delta t_k} + (1+\theta)L$$

$$\widehat{\mathbf{f}}^{k+1} = \frac{M}{\Delta t_k}\mathbf{u}(:,k) + \mathbf{f}(:,k+1) - \theta L \text{ and } \Psi^{k+1} = \Psi(:,k+1)$$

2: ***for*** $k = 0, 1, 2, ..., t_m - 1$ ***do***

3: ***step 2:*** *solve algorithm* (5.6.3.1) *to find* $\mathbf{u}(:,k+1)$

4: ***step 3:*** *set* $\mathbf{u}(:,k) = \mathbf{u}(:,k+1)$ *and repeat.*

5: ***end for***

In this chapter we give the domain decomposition method for parabolic variational inequalities and proposed some algorithm to solve it. These algorithms are the extended version of the algorithms described for the elliptic variational inequalities. In the next chapter we will present numerical results to successfully validate these algorithms.

Chapter 7

Numerical Results

In this chapter we present a numerical investigation of the methods introduced in Chapter 5. Recall the obstacle problem from Chapter 2

$$\begin{cases} -\Delta u - f \geqslant 0 \;\; \text{in} \;\; \Omega, \\ u - \psi \quad \geqslant \quad 0 \;\; \text{in} \;\; \Omega, \\ (u-\psi)(-\Delta u - f) = 0 \;\; \text{in} \;\; \Omega, \end{cases}$$

where ψ is the obstacle function and f is a given vertical force. The matrix form is given by

$$\begin{aligned} A\mathbf{u} &\geqslant \mathbf{f}, \\ \mathbf{u} &\geqslant \Psi, \\ (\mathbf{u}-\Psi)_i(A\mathbf{u}-\mathbf{f})_i &= 0 \quad i = 1, 2, ..., n. \end{aligned}$$

where $A \in \mathbb{R}^{n\times n}$ and $\mathbf{f}, \Psi \in \mathbb{R}^n$. After applying the non overlapping domain decomposition methods presented in Chapter 6, we obtain the following system of inequalities

$$\begin{pmatrix} A^e_{II} & O & A^e_{I\mathit{\Gamma}} \\ O & A^i_{II} & A^i_{I\mathit{\Gamma}} \\ A^e_{\mathit{\Gamma} I} & A^i_{\mathit{\Gamma} I} & A_{\mathit{\Gamma\Gamma}} \end{pmatrix} \begin{pmatrix} \mathbf{u}^e_I \\ \mathbf{u}^i_I \\ \mathbf{u}_{\mathit{\Gamma}} \end{pmatrix} \geqslant \begin{pmatrix} \mathbf{f}^e_I \\ \mathbf{f}^i_I \\ \mathbf{f}_{\mathit{\Gamma}} \end{pmatrix}.$$

This partitioning yields two subproblems. One subproblem is the reduced variational inequality

$$\tilde{A}\tilde{\mathbf{u}} = \begin{pmatrix} A^i_{II} & A^i_{I\mathit{\Gamma}} \\ A^i_{\mathit{\Gamma} I} & S^e \end{pmatrix} \begin{pmatrix} \mathbf{u}^i_I \\ \mathbf{u}_{\mathit{\Gamma}} \end{pmatrix} \geqslant \begin{pmatrix} \mathbf{f}^i_I \\ \tilde{\mathbf{f}}_{\mathit{\Gamma}} \end{pmatrix} = \tilde{\mathbf{f}}, \tag{7.0.1}$$

where

$$S^e := A_{\mathit{\Gamma\Gamma}} - A^e_{\mathit{\Gamma} I}(A^e_{II})^{-1}A^e_{I\mathit{\Gamma}} \quad \text{and} \quad \tilde{\mathbf{f}}_{\mathit{\Gamma}} := \mathbf{f}_{\mathit{\Gamma}} - A^e_{\mathit{\Gamma} I}(A^e_{II})^{-1}\mathbf{f}^e_I.$$

and the other subproblem in the complementary subdomains is a standard PDE with solution given by

$$\mathbf{u}^e_I = (A^e_{II})^{-1}(\mathbf{f}^e - A^e_{I\mathit{\Gamma}}\mathbf{u}_{\mathit{\Gamma}}).$$

The system (7.0.1) is solved using two different approaches. One approach is a direct method in which we solve the matrix in the above system in one step, whereas in the second approach we approximate the solutions $\mathbf{u}^i_I$ and $\mathbf{u}_{\mathit{\Gamma}}$ in an iterative manner. In the domain decomposition method described in Chapter 5, we assume that the interface lies outside of the obstacle support. Therefore, we pose the interface problem as a standard PDE given by

$$S^e\mathbf{u}_{\mathit{\Gamma}} = \mathbf{f}_{\mathit{\Gamma}} - A^e_{\mathit{\Gamma} I}\mathbf{u}^{e\{1\}}_I - A^i_{\mathit{\Gamma} I}\mathbf{u}^i_I.$$

The only variational inequality in subdomain Ω^i is given by

$$A^i_{II}\mathbf{u}^i_I \geqslant \mathbf{f}^i_I - A^i_{I\Gamma}\mathbf{u}_\Gamma.$$

This can be solved as a minimization problem for the reduced obstacle problem (5.3.4)

$$\text{Find } \tilde{\mathbf{u}} \in \tilde{K} \text{ such that } \forall\ \mathbf{v} \in \tilde{K}$$

$$\tilde{J}(\tilde{\mathbf{u}}) \leqslant \tilde{J}(\mathbf{v}).$$

Where

$$\tilde{K} = \{v \in \mathbb{R}^{N^i+N^\Gamma}\ :\ \mathbf{v} \geqslant \tilde{\Psi} \text{ in } \Omega^i\}$$

and $(\tilde{\Psi})^T = [(\Psi^i_I)^T(\Psi_\Gamma)^T]$

7.1 Solution methods

The algorithms described in Chapter 5 require quadratic programming solvers. In the following we described some quadratic programming solvers, which can be used to determine the solutions for large scale sparse linear systems.

- Lemke method
- `quadprog` (a built-in matlab programme)
- interior ellipsoidal trust region and barrier function algorithm [121]
- Newton-KKT interior point method [1].

7.1.1 Lemke

Lemke method is an algorithm for solving the linear complementarity problems in optimization. It is a pivot-based method, and can efficiently find an appropriate pivot. Lemke algorithm find the solution in the following steps [120].

- introduce an auxiliary variable z_0 and modify the LCP (3.5.1)

$$\mathbf{w} = \mathcal{M} \begin{pmatrix} \mathbf{z} \\ \mathbf{z}_0 \end{pmatrix} + \mathbf{q} \tag{7.1.1}$$

  where

$$\mathcal{M} = \begin{pmatrix} \tilde{M} & \mathbf{d} \end{pmatrix} \qquad \mathbf{d} = (1\ 1\ 1...1)^T$$

find the solution of (7.1.1) as follows

- (i) if $\mathbf{q} \geqslant 0$, $\mathbf{w} = \mathbf{q}$, $\mathbf{z} = 0$ is the solution. If $\mathbf{q} \leqslant 0$, find $s =$ arg min $\mathbf{q}_i/\mathbf{d}_i$ and pivot $\mathbf{z}_0$ with $\mathbf{w}_s$. Compute $\mathcal{M}$ and $\mathbf{q}'$, let $\mathbf{t}_s = \mathbf{z}_s$ be driving variable.
- (ii) If $m'_s \geqslant 0$, where m'_s is the column vector of $\mathcal{M}$ corresponding to $\mathbf{t}_s$ then this algorithm does not give any solution to the LCP. Otherwise let $\mathbf{t}_p$ be the p-th element of the basic variables where $p =$ arg min $\{\mathbf{q}_i/m'_i : m'_i \leqslant 0\}$.
- (iii) Pivot $\mathbf{t}_p$ with $\mathbf{t}_s$ and Compute $\mathcal{M}$ and $\mathbf{q}'$. If $\mathbf{t}_p = \mathbf{z}_0$, then $\mathbf{q}'$ is the solution of LCP, otherwise set $\mathbf{t}_r = \mathbf{t}'_p$ and repeat (i)-(iii).

7.1.2 `Quadprog`

`quadprog` is a built in matlab programme included in the optimization toolbox to solve quadratic programming problems. This programme includes the algorithms for the following three methods

- trust region reflective
- interior point for convex problems

- active set.

Trust region reflective and interior point convex can be used for problems with large and sparse systems, whereas the active set method is a medium scale algorithm. For our problems, we used the trust region reflective algorithm. In this method, we need to supply a Hessian multiplier. Instead of computing the Hessian directly, we use the Hessian times vector product. In each iteration of this method, we approximate the solution of a dense linear system using the method of preconditioned conjugate gradient method(PCG) [29], [47].

7.1.3 Interior ellipsoidal trust region method (IET)

In this method we implemented, the interior ellipsoidal trust region and barrier function algorithm. We apply the dual solution updating technique in the standard QP form. In this method, the solution is calculated in two phases. In the first phase, an interior feasible point is found and in the second phase, a local optimal solution is determined [121].

7.1.4 Newton-KKT interior point method

Newton-KKT finds a local minimizer of an indefinite quadratic programming problem. This algorithm contains two methods

- affine scaling;

- barrier function method.

Both methods construct feasible primal iterates and require an initial feasible point. These algorithm do not involve the trust regions. In early iterations of this method, quasi Newton direction was used instead of the the Newton KKT direction. A matrix $W = A + E$ is substituted for the Hessian of the Lagrangian in order to obtain the Quasi-Newton direction, where E is a positive semi definite matrix which is constructed inside the algorithm. The quadratic programming structure allows a more efficient computation of W and such computation may often be skipped or W may be reused from the previous iteration. Affine scaling algorithms are simple and usually give faster convergence when barrier based interior point methods are used. The search direction generated by such algorithms is consisting of an affine scaling component and a centering component. When the barrier parameter is set to zero, the centering component vanishes, and the direction reduces to an affine scaling direction [1].

7.2 Numerical Experiments

Test 1: one obstacle. For our first test problem, we consider an elastic membrane which lies above an obstacle of height 1 centered at the origin with square cross-section with side length $\ell^o = 0.3$ under the forcing function $f = 1$ with $\Omega = (-1, 1)^2$. We choose Ω^i to be a square region with side-length ℓ^i which contains the support of the obstacle such that the interface Γ lies outside of the obstacle support.

Test 2: three obstacles

For the same domain Ω we consider the obstacle problem with three square obstacles of height 1 with centers located at $(0.5, 0.5), (-0.5, 0.5), (0, -0.5)$ and equal sides $\ell^o = 0.3$. We performed the same investigation, where we chose Ω^i to be a multiply-connected domain consisting of square regions of side-length ℓ^i (see Fig. 7.4).

We run both of these test problems for different sizes of Ω^i ℓ^i with different levels of refinement.

7.2.1 Global solution

To investigate our algorithms, we determine the global solution of the obstacle problem on the global domain Ω without the use of domain decomposition.

Test 1: one obstacle: global solution

methods=	`quadprog`	IET	Lemke
level = 4	33(0.26)	22(0.29)	218(1.38)
5	48(0.42)	27 (0.73)	946(7.30)
6	72 (0.85)	30 (3.55)	3938 (247.91)

Table 7.1: number of iterations(time)for solving one obstacle problem on.

Test 2: three obstacles: global solution

methods=	`quadprog`	IET	Lemke
level = 4	16(0.23)	25 (0.21)	202(1.27)
5	29 (0.35)	28 (0.89)	914(6.62)
6	98 (1.15)	39 (6.65)	3874(208.67)

Table 7.2: number of iterations (time) for solving three obstacle problem.

We see from the tables the numbers of iterations showing the dependence not only on the level of refinements but also on ℓ^i the size of Ω^i. It is also noted that the CPU time for all three methods depends on level of refinements. As we refine the mesh CPU time increased. We observed that Lemke required not only a high number of iterations but also took long time to converge than other two methods.

Reduced QP direct algorithm

This algorithm is a direct approach to solving the problem in three steps. In step 2 we solve the reduced system (7.0.1), which involves the partial Schur complement as an element of the matrix. To obtain a solution we used the Matlab function `quadprog`.

Test 1: one obstacle: RQP direct

ℓ^i=	0.4	0.5	0.6
level = 4	8(0.06)	9 (0.09)	9(0.16)
5	8(0.10)	9 (0.15)	9(0.18)
6	8 (0.24)	9 (0.35)	9(0.54)

Table 7.3: number of quadprog iterations $\ell^0 = 0.3$; $n_0 = 81$.

Test 2: three obstacles: RQP direct

ℓ^i=	0.4	0.5	0.6
n = 4	8 (0.06)	8 (0.07)	8(0.10)
5	8(0.17)	9(0.26)	9(0.29)
6	9 (0.88)	9(1.24)	9(1.48)

Table 7.4: number of quadprog iterations(time) for three obstacles problem $\ell^0 = 0.3$; $n_0 = 81$.

Comparing above tables, we see that solving the variational inequality problem on the global domain is substantially more computationally demanding, when compared to solving the reduced variational inequality on a smaller subdomain. Additionally it is noted that when solving the variational inequality problem on the global domain, the number of iterations increased with respect to the mesh size. In comparison, when solving the variational inequality problem on subdomain Ω^i, it can be seen in the above tables the number of iterations indicate the mesh independent performance. We could also see that the CPU time for solving the global variational inequality almost same for all three methods on coarse mesh but on finer mesh Lemke seems to be quiet expansive with respect to CPU time. Also it is noted that CPU time for solving reduced variational inequality is much low than the solving global variational inequality. However CPU time could be seen to depends on the levels of refinement for the reduced variational inequality.

It was mentioned in Chapter 5 that our proposed domain decomposition method requires a mild assumption of interface Γ. Therefore, if we have a reasonable intuition of the behavior of the interface, then the above algorithm is a better alternative to solve variational inequality in a smaller region than solving variational inequality on a global domain. Solving the reduced variational inequality on subdomain Ω^i using the direct algorithm requires the solution of a system containing S^e, the partial Schur complement as an element. This can be avoided at the expense of Picard iterations. These are iterative procedures, and solve the reduced system of matrix in two steps for $\mathbf{u}_\Gamma$ and $\mathbf{u}_I^i$.

Picard reduced QP algorithm

In the given algorithm, we solved the PDI in the step 2(ii) by using the Matlab function `quadprog`. The PDI is coupled together with the interface equality/inequality problem in step 2(i) in an iterative manner. Note that the variational inequality problem is now posed over a small subdomain, and hence has low complexity - we therefore decided not to report on it. The initial guess was computed on a fixed coarse mesh with n_0 nodes.

Test 1: one obstacle: Picard algorithm

$\ell^i=$	0.4	0.5	0.6
n = 4	1 (0.31)	6(0.36)	6 (0.36)
5	7(0.50)	10 (0.56)	10 (0.60)
6	11 (1.64)	15 (2.60)	16(3.13)

Table 7.5: Picard iterations(time) for square obstacle problem with $\ell^0 = 0.3$; $n_0 = 81$.

Table 7.5 displays the number of fixed point iterations required to solve the coupled equations (5.2.19b), (5.2.19c). The algorithm was stopped when the global complementarity conditions were brought below a tolerance of 10^{-3}; we found that this corresponds to a max norm of the algebraic error of the order of the tolerance. We see that the number of iterations grows logarithmically as we increase the level of refinement. On the other hand, reducing ℓ^i, the size of Ω^i leads to a smaller number of iterations, while preserving the dependence behavior on the refinement level.

Test 2: three obstacles: Picard algorithm

ℓ^i=	0.4	0.5	0.6
n = 4	1 (0.52)	8 (0.55)	8(0.57)
5	8(1.78)	12 (1.70)	12 (1.79)
6	12 (2.44)	17 (3.65)	20(3.87)

Table 7.6: Picard iterations for three obstacles problem with $\ell^0 = 0.3$; $n_0 = 81$.

The numerical results displayed in Table 7.6 show that for this harder problem, the number of iterations displays a logarithmic dependence for ℓ^i sufficiently small, but deteriorates for larger Ω^i. However, this is not the context we devised our algorithm for. Finally, we note that for this test problem, the variational inequality in step (ii) decouples into three independent variational inequalities. For this iterative method we see that the CPU time is also increased as compare to the CPU time for reduced QP direct method showing the dependence on level of refinement. We see from above results that in the Picard iterative procedure, the number of iterations shows a dependence on the refinement level. To overcome this dependence, we applied Newton's method to solve the nonlinear interface problem and obtained some improved results as shown in the following section.

7.2.2 Newton's methods for the nonlinear interface problem

In these methods, the nonlinear interface problem in step 2(i) of algorithm (6.3.1.1) is solved by using Newton's method coupled with the PDI in step 2(ii) using different iterative techniques.

Newton reduced QP algorithm

We rerun test 1 and test 2 with one Newton step at the interface Γ. We calculate the Jacobian and then find the solution by using Newton's method at the interface. The results are given in the following tables.

Test 1: one obstacle: Newton's method

ℓ^i=	0.4	0.5	0.6
levels = 4	1(0.69)	1 (1.65)	1(1.67)
5	1(1.50)	1 (2.12)	1(2.14)
6	1 (9.23)	1 (17.09)	1(21.27)

Table 7.7: Fixed point iterations(time) for square obstacle problem with $\ell^0 = 0.3$; $n_0 = 81$.

Test 2: three obstacles: Newton's method

ℓ^i=	0.4	0.5	0.6
levels = 4	1(3.09)	1 (3.12)	1(3.18)
5	1(18.42)	1 (19.64)	1(19.72)
6	2 (214.18)	1 (213.40)	1(219.04)

Table 7.8: Fixed point iterations(time) for three obstacles with $\ell^0 = 0.3$; $n_0 = 81$.

Tables (7.7) and (7.8) display the number of Newton iterations with one Newton step for the solution of the non linear problem at interface. These results are quite encouraging in the sense that only one fixed point iteration was required to determine the solution. When compared to the results in tables (7.5) and (7.6) it can be seen that by using this method, convergence is obtained in just one iteration. However CPU time for Newton method is more than the Picard method. CPU time increased almost 6 times to the level of refinement. Specially for coarse mesh at level 6 CPU time is very high for three obstacle problem.

Newton-GMRES exact Jacobian algorithm

In this method, we construct the Jacobian as given in (5.6.2). To determine the solution at the interface boundary Γ, we used the exact Jacobian for the Jacobian-vector product in GMRES. The following tables show the number of fixed point (GMRES) iterations for two test problems, using the exact Jacobian in GMRES.

Test 1: one obstacle: Exact Newton-GMRES

$\ell^i=$	0.4	0.5	0.6
level = 4	1(14)	1(27)	1(27)
5	1(34)	1(50)	1(50)
6	1(53)	1(96)	1(107)

Table 7.9: fixed point (GMRES) iterations for one obstacle problem.

Test 2: three obstacles: Exact Newton-GMRES

$\ell^i=$	0.4	0.5	0.6
level = 4	1(32)	1(44)	1(44)
5	1(41)	1(69)	1(69)
6	1(63)	1(123)	1(147)

Table 7.10: fixed point (GMRES) iterations for three obstacles problem.

In tables (7.9) and (7.10), we observed that the number of Picard iterations is still one for both test problems considered, but the complexity of Newton method is growing with the number of GMRES iterations. The number of GMRES iterations is essentially doubled as we increase the level of refinement.

Jacobian-free Newton-GMRES algorithm (JFNG)

In this method, solutions are obtained without constructing and storing the Jacobian. We approximate the Jacobian by Jacobian-vector product using GMRES.

Test 1: one obstacle: JFNG

ℓ^i=	0.4	0.5	0.6
level = 4	3(6.33) (0.10)	3(13.33) (1.07)	3(14.16) (1.12)
5	3(13.66)(1.43)	3(21.33)(2.12)	3(21.33)(2.17)
6	3(21.33)(3.26)	2(33.5) (5.83)	2(39)(7.50)

Table 7.11: fixed point(average numbers of (GMRES))(time) iterations for one obstacle problem.

Test 2: three obstacles: JFNG

ℓ^i=	0.4	0.5	0.6
level = 4	3(6.33)(2.40)	2(18.50)(13.33)	2(9.5)(13.48)
5	2(13)(4.93)	2(18.5) (21.33)	2(18.5)(21.47)
6	2(20.5)(17.78)	2(32) (21.39)	2(37)(21.57)

Table 7.12: fixed point(average numbers of GMRES) iterations for three obstacles problem.

The tables (7.11) and (7.12) show the number of fixed point (average numbers of GMRES) iterations for the two test problems: one obstacle and three obstacles, with coarse level set equal to 2. We observe that in Jacobian-free Newton-GMRES, the fixed point iterations are independent of the levels of refinement. Also, the number of GMRES iterations is reduced when directly compared to results obtained from the exact Jacobian Newton-GMRES algorithm. Also it is noted that the CPU time decreased as compare to the Newton method. However same for the Newton method the CPU time for level of refinement 6 is higher for test problem 2. In all above algorithms we observed that the CPU time increased with level of refinement. We see that the JFNG algorithm is less expensive in terms of both the number of iterations and the CPU time. Therefore, we conclude that for the obstacle problem, the Jacobian-free Newton-GMRES method is better choice in all of the cases considered above.

7.2.3 Preconditioning of Newton-GMRES algorithm (JFNG)

Our aim of preconditioning the JFNG method is not only to avoid the construction of Jacobian matrix J, but also to reduce the number of GMRES iterations. It will be shown in the following tables, that an effective preconditioner for JFNG can reduced the number of GMRES iterations. We rerun the test problems for the JFNG algorithm with S^e as preconditioner.

Test 1: one obstacle: preconditioned JFNG

ℓ^i=	0.4	0.5	0.6
level = 4	1(1.00)(0.36)	2(2.00) (0.51)	2(2.00)(1.98)
5	2(2.00)(0.56)	3(2.66)(0.79)	3(2.66)(0.84)
6	2(3.00)(1.34)	3(3.00) (1.90)	3(4.33)(2.16)

Table 7.13: fixed point(average numbers of (GMRES))(time) iterations for one obstacle problem.

Test 2: three obstacles: preconditioned JFNG

ℓ^i=	0.4	0.5	0.6
level = 4	1(1.00)(0.64)	3(2.00)(0.64)	3(2.00)(0.65)
5	3(2.00)(1.29)	3(3.33)(1.32)	3(3.33)(1.31)
6	3(3.00)(6.50)	3(4.66)(6.75)	3(5.33)(6.85)

Table 7.14: fixed point(average numbers of GMRES) iterations for three obstacles problem.

Tables (7.13) and (7.14) illustrate the number of fixed point iterations and average number of GMRES iterations for the preconditioned JFNG algorithm, where S^e is used as a preconditioner. The number of fixed points iterations are independent of levels of refinement same as the case of JFNG algorithm. It can be seen that the CPU time and number of GMRES iterations are reduced, with the logarithmic dependence on the levels of refinement. Results given in both tables show that S^e, is a good preconditioner for JFNG algorithm.

7.3 Convection diffusion problem

Recall the obstacle problem with convection diffusion parameter from chapter 2

$$\begin{cases} -\mathrm{div}(\mathbf{a}\cdot\nabla u)+\mathbf{b}\cdot\nabla u \geqslant f \ \text{ in } \ \Omega, \\ u \ \geqslant \ \psi \text{ in } \ \Omega, \\ (u-\psi)(-\mathrm{div}(\mathbf{a}\cdot\nabla u)+\mathbf{b}\cdot\nabla u - f) = 0 \ \text{ in } \ \Omega, \end{cases} \tag{7.3.1}$$

where $\mathbf{a} = \alpha I_d$, is a diffusion vector and $\mathbf{b}$ is a convection vector.

As for the case of elliptic obstacle problem, we run two test problems for convection diffusion problem i.e., for one obstacle and three obstacles. We run the following test problems 3 and 4, with various diffusion parameters and convection is fixed as 1.

Test 3: one obstacle

For this test problem, we consider an elastic membrane which lies above an obstacle of height 1 centered at the origin with square cross-section with side length $\ell^o = 0.3$ under the forcing function $f = 1$, with various diffusion parameters and $\mathbf{b} = (11)$ with $\Omega = (-1, 1)^2$. We choose Ω^i to be a square region with side-length ℓ^i which contains the support of the obstacle such that the interface Γ lies outside of the obstacle support.

Test 4: three obstacles

For the same domain Ω we consider the obstacle problem with three square obstacles of height 1 with centers located at $(-0.5, 0.2), (-0.5, 0.5), (0, -0.5)$ and equal sides $\ell^o = 0.3$. We performed the same investigation, where we chose Ω^i to be a multiply-connected domain consisting of square regions of side-length ℓ^i (see Fig. 7.5).

7.3.1 Global solution

We applied `Lemke`, to determine the solution of the convection diffusion problem on the global domain Ω. We determine the solution for different diffusion parameters whereas the convection is kept fixed as 1.

Test 3: one obstacle: global solution

$\alpha =$	10^{-2}	10^{-1}	1	10
level = 4	223 (0.42)	221(0.37)	218(0.35)	218(0.35)
5	975(7.72)	953(7.79)	946(7.59)	940(7.42)
6	4101(251.08)	3952(225.88)	3945(223.85)	3939(223.80)

Table 7.15: `Lemke` iterations for one obstacle convection diffusion problem.

Test 4: three obstacles: global solution

$\alpha =$	10^{-2}	10^{-1}	1	10
level = 4	213(0.34)	213(0.37)	204 (0.35)	204(0.34)
5	935(7.70)	937 (7.31)	916(7.23)	914(7.19)
6	3952(228.87)	3918 (222.20)	3876 (220.85)	3872(222.57)

Table 7.16: (`Lemke`) iterations (time)for three obstacles convection diffusion problem .

We observe that solving the convection diffusion problem on the global domain using `Lemke` requires a large number of iterations. In particular, the deterioration is of order $O(h^{-2})$. We observed that the CPU time is showing dependence on the level of refinements. CPU time for the coarse mesh is low but as we increased the level of refinement the CPU time grows rapidly. As we can see there is a large difference of time between level 5 and level 6. We will show in the next section that solving the convection diffusion problem, by using an appropriate substitution, we described in Chapter 2, provides a better alternative.

7.3.2 Convection diffusion problem converted into reaction diffusion problem

As mentioned previously, by using an appropriate substitution convection diffusion problems can be converted into symmetric reaction diffusion problems, and can be solved as minimization problems. Recalling the converted reaction diffusion problem from Chapter 2

$$\begin{cases} -\alpha\Delta U + CU \geqslant \check{f} \;\; \text{in} \;\; \Omega, \\ U \geqslant \check{\psi} \quad \text{in} \;\; \Omega, \\ (U-\check{\psi})(-\alpha\Delta U - \check{f}) = 0 \;\; \text{in} \;\; \Omega, \end{cases} \tag{7.3.2}$$

where

$$C = \frac{1}{4\alpha}, \quad \check{\psi} = \psi e^{-\frac{x+y}{2\alpha}}, \quad \check{f} = f e^{-\frac{x+y}{2\alpha}} \quad U = u e^{-\frac{x+y}{2\alpha}}$$

In the following, we present number of iterations, obtained by using the minimization formulation of the converted problem. Solutions were obtained by using Matlab program `quadprog`.

7.3.3 Picard reduced QP algorithm

The following tables display the number of Picard iterations required to obtain the solutions for various diffusion parameters α and convection parameter is set as 1, for test problems 3 and 4. In a similar manner to the elliptic obstacle problem, we also assumed here that the variational inequality problem is now posed over a small subdomain, and hence has low complexity - we therefore decided not to report on it.

Test 3: one obstacle: Picard algorithm

$\alpha =$	10^{-1}	1	10
level = 4	8 (0.19)	11 (0.23)	16(0.35)
5	14 (0.50)	18 (1.62)	24(1.78)
6	25(2.39)	30(3.59)	45(4.68)

Table 7.17: Picard iterations(time) for one obstacle converted problem.

Test 4: three obstacles: Picard algorithm

$\alpha =$	10^{-1}	1	10
level = 4	7 (0.15)	10(0.23)	14(0.32)
5	13(0.46)	14(0.57)	20(0.72)
6	23(2.43)	23(2.46)	31(3.39)

Table 7.18: Picard iterations for three obstacles converted problem.

We see that number of iterations is very small for Picard reduced QP algorithm when directly compared to the number of iterations required for the global solution of the associated convection diffusion problem. However, the number of iterations can be seen to depend on the levels of refinement in both cases. To avoid this dependence, we introduce the Newton reduced QP algorithm for the nonlinear interface problem. We can also see that the CPU time is sufficiently low than the time required by the global solution, still showing the dependence on the refinement levels.

7.3.4 Newton reduced QP algorithm

We rerun test 3 and 4 with one Newton step at the interface Γ. We calculate the Jacobian and then find the solution by using Newton's method at the interface. The results are given in the following tables.

Test 3: one obstacle: Newton's method

$\alpha =$	10^{-1}	1	10
level = 4	6 (8.22)	5(8.15)	5(8.10)
5	8 (50.57)	6(47.39)	7(47.51)
6	8 (90.57)	6 (89.13)	8(95.14)

Table 7.19: fixed point iterations(time) for one obstacle converted problem.

Test 4: three obstacles: Newton's method

$\alpha =$	10^{-1}	1	10
level = 4	2 (7.16)	2(7.15)	1(5.19)
5	2(7.26)	1(6.18)	1(6.01)
6	3(100.06)	1 (96.34)	2(97.57)

Table 7.20: fixed point iterations(time) for three obstacles converted problem.

The above tables show that the number of fixed points iterations is reduced when directly compared to both solutions obtaining globally and also through Picard algorithm. It is noted that the CPU time for Newton method increased as compare to the Picard algorithm. CPU time is almost doubled from level of refinement 5 to the level of refinement 6. We also observed that the number of fixed point iterations has been reduced with the use of Newton's method. Additionally results indicate mesh independent performance. Therefore, we can see that Newton's reduce QP algorithm is a suitable alternative when solving the convection diffusion problems that have been converted into appropriate reaction diffusion problems.

7.4 Parabolic variational inequality

We consider the parabolic variational inequality described in Chapter 4

$$(\frac{\partial u}{\partial t}, v-u) + a(u, v-u) \geqslant \ell(v-u), \quad \forall v \in K, \ t \in (0,T]$$

with initial condition

$$(u(\mathbf{x},0), v-u) \ \geqslant (\psi(\mathbf{x},0), v-u).$$

Applying discretization in both space and time variables and a non-overlapping domain decomposition method, the matrix representation of above problem at each time step, $k = 0, ...t_m - 1$, can be written as

$$\widehat{A} = \begin{pmatrix} \widehat{A}^e_{II} & O & \widehat{A}^e_{I\Gamma} \\ O & \widehat{A}^i_{II} & \widehat{A}^i_{I\Gamma} \\ \widehat{A}^e_{\Gamma I} & \widehat{A}^i_{\Gamma I} & \widehat{A}_{\Gamma\Gamma} \end{pmatrix} \begin{pmatrix} (\mathbf{u}^e_I)^k \\ (\mathbf{u}^i_I)^k \\ (\mathbf{u}_\Gamma)^k \end{pmatrix} \geqslant \begin{pmatrix} (\widehat{\mathbf{f}}^e_I)^k \\ (\widehat{\mathbf{f}}^i_I)^k \\ (\widehat{\mathbf{f}}_\Gamma)^k \end{pmatrix} = \widehat{f}$$

where

$$\widehat{A} = (1+\theta)L + \frac{M}{\Delta t_k}, \quad \widehat{\mathbf{f}} = \mathbf{f} - \theta L + \frac{M}{\Delta t_k} \quad \theta \ = \ 0 \text{ or } 1/2.$$

In the following algorithms at each time step, we solved a PDI, in step 2(ii) by using the Matlab function `quadprog`. The PDI is coupled with the interface equality problem in step 2(i) in an iterative manner.

Test 5: Moving obstacle For the test problem for parabolic variational inequality we consider an elastic membrane which lies above an obstacle of height 1 with square cross-section with side length $\ell^o = 0.3$, moving along x-axis, with time, under the forcing function $f = 1$ with $\Omega = (-1, 1)^2$. We choose Ω^i to be a square region with side-length ℓ^i which contains the support of the obstacle such that the interface Γ lies outside of the obstacle support.

7.4.1 Global solution

In a similar manner to both the elliptic obstacle problem and the obstacle problem with convection diffusion parameters, we will now solve the parabolic variational inequalities on global domain. The results will then be used to compare to those obtained using domain decomposition algorithms.

Test 5: moving obstacle: global solution

level \ Δt	0.1	0.01	0.001
4	26.25 (0.47)	15.62(0.36)	15.07(0.23)
5	55.72 (1.25)	20.65(1.18)	15.28(1.06)
6	106.12 (4.14)	36.26(3.60)	19.13(3.18)

Table 7.21: average number of quadprog iterations (time)for global solution $tol = 10^{-4}$.

In above tables We calculated the average number of iterations and CPU time per each time step. Number of iterations are showing dependence on the levels of refinement. From the table, we can see that the CPU time is inversely proportional to the time stepping parameter. In order to obtain a more accurate solution, smaller time step should be considered, which will lead to an increasing CPU time.

7.4.2 Picard reduced QP algorithm

We apply the Picard algorithm with global complementarity condition as a stopping criterion $\max\limits_{1\leqslant i\leqslant n} \mid (L\mathbf{u}-\mathbf{f})_i(\mathbf{u}-\Psi)_i \mid\leqslant tol$. The results are compiled for $tol = 10^{-3}, 10^{-4}, 10^{-5}$, for different h and Δt.

Test 5: moving obstacle: Picard algorithm

level \ Δt	0.1	0.01	0.001
4	5.44 (0.09)	2.80 (0.06)	1.60 (0.03)
5	9.22 (0.23)	6.48 (0.17)	3.00(0.12)
6	15.77(1.12)	10.65 (0.81)	5.88(0.64)

Table 7.22: average number of Picard iterations(time) for moving obstacle with $tol = 10^{-3}$.

level \ Δt	0.1	0.01	0.001
4	7.88(0.15)	3.42 (0.078)	2.17(0.093)
5	13.66 (0.28)(0.29)	8.50 (0.26)	4.00(0.15)
6	23.89 (1.53)	14.46(1.14)	7.32(1.04)

Table 7.23: average number of Picard iterations for moving obstacle with $tol = 10^{-4}$.

Δt / level	0.1	0.01	0.001
4	10.33 (0.12)	4.31(0.09)	2.60(0.08)
5	18.55 (0.42)	10.60 (0.25)	5.02(0.17)
6	32.44 (2.04)	18.17 (1.38)	9.00(1.23)

Table 7.24: average number of Picard iterations (time) for moving obstacle with $tol = 10^{-5}$.

These tables Tables (7.23)-(7.24) show the average number of iterations and the CPU time for Picard algorithm. We can see that the number of iterations is significantly higher when the solution is obtained without the use of domain decomposition method, in comparison to the number of iterations obtained using Picard iterative algorithm. The CPU time is also decreased when we use the Picard algorithm for domain decomposition method. However, it should be noted that, in both cases the number of iterations and CPU time could be seen to depend on the levels of refinement. To remove this dependence, we apply Newton's method to obtain the solution to the interface problem. The results are given in the following section.

7.4.3 Newton's method for the nonlinear interface problem

In these methods, the nonlinear interface problem in step 2(i) of algorithm (6.3.1.1) is solved by using Newton's method coupled with the PDI in step 2(ii) using different iterative techniques.

Test 5: moving obstacle: Newton's method

level \ Δt	0.1	0.01	0.001
4	1.77 (1.40)	1.59(0.23)	3.36(0.98)
5	2.44 (5.41)	2.90(1.29)	3.09(1.24)
6	3.18 (9.34)	3.01 (3.63)	2.00 (2.45)

Table 7.25: fixed point iterations (time)for moving obstacle with $tol = 10^{-3}$.

level \ Δt	0.1	0.01	0.001
4	2.66 (0.43)	2.00 (0.29)	4.24 (1.14)
5	3.33 (1.81)	2.17 (1.79)	4(1.74)
6	4.88 (7.75)	4.06(7.19)	3.02 (6.87)

Table 7.26: fixed point iterations(time) for moving obstacle with $tol = 10^{-4}$.

level \ Δt	0.1	0.01	0.001
4	3.33 (0.54)	3.34 (0.48)	5.62 (1.42)
5	4.22 (2.73)	4.36(2.17)	4.28 (2.67)
6	5.28(9.11)	5.19 (7.99)	3.15 (5.76)

Table 7.27: fixed point iterations (time)for moving obstacle problem with $tol = 10^{-5}$.

As was found for both the elliptic obstacle problem and the convection diffusion problem, the above tables show that, when Newton's method is used to determine the solution to the nonlinear interface problem, the number of iterations for parabolic problem also delivers results independent of mesh size. However CPU time not only showing the dependence on the levels of refinement but also increased in comparison to the Picard method. An issue with Newton method is that it requires the construction of the Jacobian, which can be computationally demanding. To avoid this, we apply the Newton-GMRES method to obtain a Jacobian-free algorithm, at the expense of GMRES iterations.

7.4.4 Newton-GMRES method for non linear interface problem

In this method, solutions are obtained without constructing and storing the Jacobian. We approximate the Jacobian by Jacobian-vector product using GMRES.

Test 5: moving obstacle: Newton-GMRES

level \ Δt	0.1	0.01	0.001
3	2.55 (7.66) (0.25)	2.41(4.65)(0.36)	4.17(4.82)(0.43)
4	5.33(14.27) (1.43)	3.63(6.21)(0.81)	3.76(2.74)(0.48)
5	9.22(27.11)(2.85)	5.02(11.34)(1.77)	3.04(3.87)(0.50)
6	14.22(42.33)(3.89)	6.72(23.17)(1.89)	4.16(6.58)(0.71)

Table 7.28: fixed point iterations (average numbers of GMRES) (time) $tol = 10^{-3}$.

level \ Δt	0.1	0.01	0.001
3	4.22(8.88)(0.89)	3.32(4.95)(0.32)	5.81(5.57)(0.24)
4	7.88(17.11)(2.94)	5.15(6.88)(1.15)	5.14(3.67)(1.32)
5	13.11(38.22) (3.22)	7.16(12.46)(2.91)	4.20(4.47)(1.51)
6	17.11(50.44)(5.27)	9.42(23.65)(3.12)	5.58(8.08)(2.08)

Table 7.29: fixed point iterations (average numbers of GMRES) (time) $tol = 10^{-4}$.

level \ Δt	0.1	0.01	0.001
3	5.77(10.33)(1.43)	4.31(5.04)(0.64)	7.17(5.94)(0.32)
4	10.01(17.55)(2.99)	6.40(7.59)(1.75)	6.02(3.86)(1.56)
5	16.66(31.88)(4.59)	9.13(14.30)(3.68)	4.60(5.04)(2.26)
6	20.32(54.77) (5.02)	12.33(26.50)(4.17)	7.12(9.33)(3.18)

Table 7.30: fixed point iterations (average numbers of GMRES):$tol = 10^{-5}$.

From tables (7.28)-(7.30), both the picards and GMRES iterations can be seen to exhibit a dependence on the levels of refinement. However, this dependence is reduced when the time step becomes smaller. CPU time is inversely proportional to the time step and is increased with level of refinement. It is additionally noted that the CPU time for the Newton-GMRES method is reduced as compare to the Newton method.

7.4.5 Preconditioned Newton-GMRES algorithm (JFNG)

In similar manner to elliptic obstacle problem, we apply preconditioning to JFNG method for parabolic problem to reduce the GMRES iterations. We rerun the test problem 5 for the JFNG algorithm with S^e as preconditioner.

Test 5: moving obstacle: Preconditioned Newton-GMRES

level \ Δt	0.1	0.01	0.001
3	2.55(2.11)(0.07)	2.44(1.00)(0.11)	4.54(1.00)(0.15)
4	5.22(3.22)(0.68)	3.49(1.86)(0.20)	3.48(1.00) (0.14)
5	8.11(5.44)(3.06)	4.97(2.90)(0.57)	3.05(1.00)(0.18)

Table 7.31: fixed point iterations(average numbers of preconditioned GMRES)(time):$tol = 10^{-3}$.

level \ Δt	0.1	0.01	0.001
3	4.11(2.00)(0.19)	3.31(1.13)(0.0.16)	5.64(1.00)(0.25)
4	8.11(3.66)(0.17)	5.09(2.00)(0.23)	5.16(1.08)(0.28)
5	13.44(6.33) (3.65)	4.14(3.00)(0.76)	4.22(1.43)(0.43)

Table 7.32: fixed point iterations(average number of preconditioned GMRES)9time0:$tol = 10^{-4}$.

Δt / level	0.1	0.01	0.001
3	5.55(2.33)(0.34)	9.10(3.11) (0.21)	7.08(1.04)(0.26)
4	10.11(3.66)(1.29)	6.37(2.15)(0.42)	6.04(1.13)(0.35)
5	17.22(6.55) (4.90)	4.31(1.22)(0.92)	4.64(1.54)(1.20)

Table 7.33: fixed point iterations(average numbers of preconditioned GMRES)(time):$tol = 10^{-5}$.

Tables (7.31)-(7.33) show that both the number of fixed point and GMRES iterations are significantly reduced in preconditioned JFNG algorithm. Also number of iterations are independent of mesh size for larger time step. However, for smaller time step, a logarithmic dependence could be seen. We can also see that the CPU time is also reduced for this method. However CPU time showing the dependence on both the levels of refinement as well as the tolerance. Overall the results in the tables illustrate that S^e is a good preconditioner for JFNG algorithm for the test problem 5.

For all three problems, the results in the tables indicate that solutions obtained using domain decomposition methods involve fewer iterations and less CPU time from those obtained globally without using DDM, highlighting the benefits of the use of the domain decomposition approach for our test problem. Additionally it was noted that our domain decomposition methods are all independent of the level of refinement when Newton's method is applied for the interface problem. These algorithms are further improved by introducing the Newton-GMRES method to convert them into Jacobian-free algorithms at the expense of GMRES iterations. For the elliptic obstacle problem, we see that the results are mesh independent. However, in the parabolic problem, iterations reduced and are independent of the levels of refinement for small time steps.

7.4.6 Numerical Analysis

In the following graph we try to validate the convergence of our global solution, obtained from the Picard reduced algorithm. We plot $\rho = \dfrac{|J(u^{k+1}) - J(u)|}{|J(u^k) - J(u)|}$.

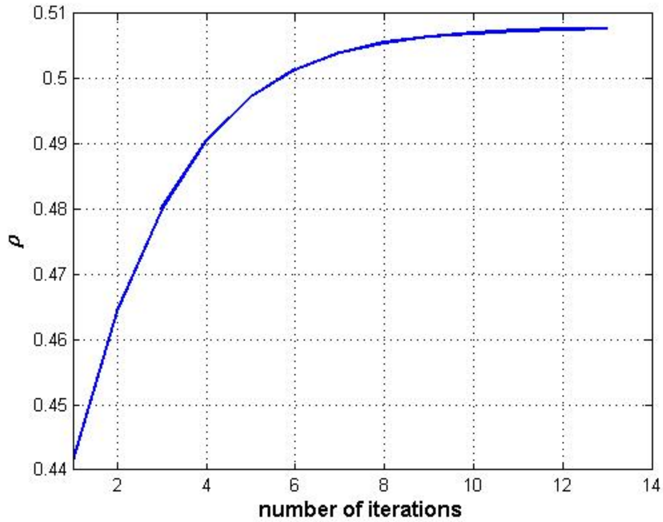


Figure 7.1: Linear rate of convergence

The graph in Figure (7.1) illustrate that the global solution of the minimization problem obtained from algorithm (5.4.1.2) converges both monotonically and linearly. We can see the values of ρ range between 0.4 and 0.51, and are always trivially less than 1. This indicates the good convergence rate for the Picard algorithm.

The graph in Figure (7.2) is plotted for the maximum value of ρ, corresponding to the different sizes of the subdomain Ω^i, obtained from the Picard algorithm.

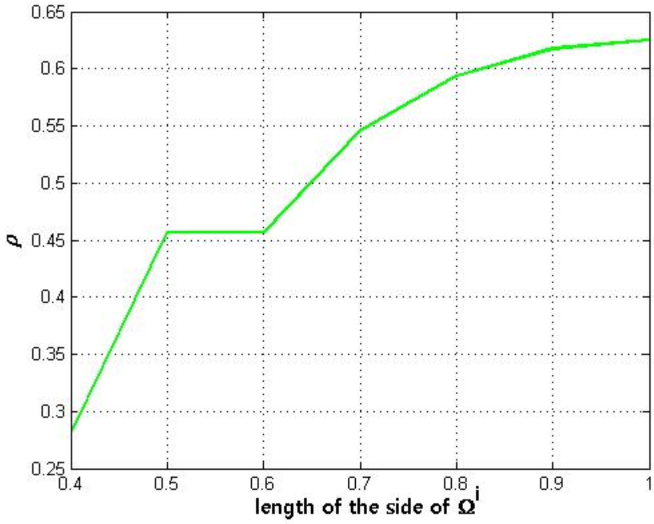


Figure 7.2: Linear rate of convergence

The values of ρ range between 0.25 and 0.66, and again less than 1, which indicates a good convergence rate for Picard algorithm for different values of the subdomain Ω^i. Additionally it is noted in the graph that the value of ρ is unchanged for $\ell^i = 0.5$ and 0.6.

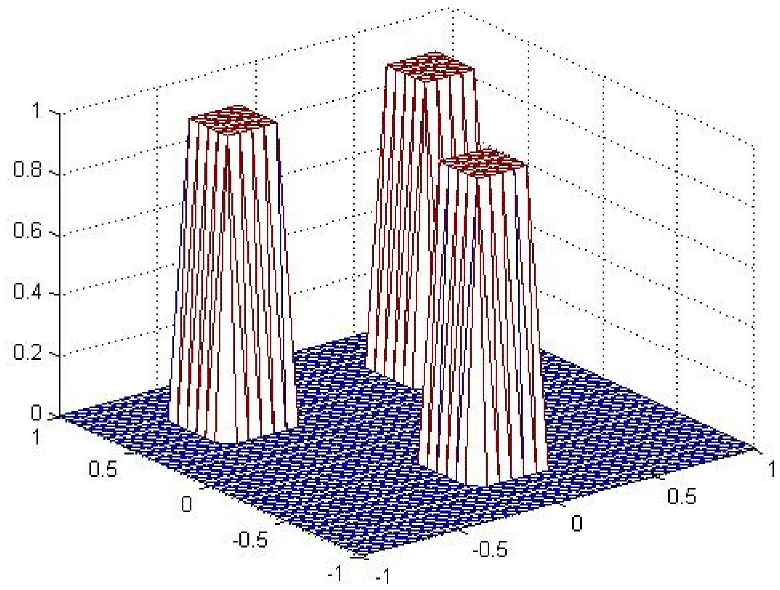


Figure 7.3: Obstacle functions

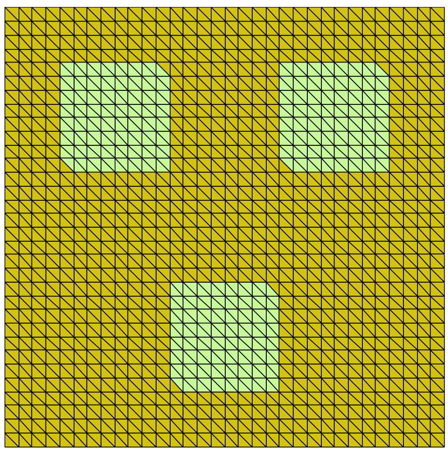
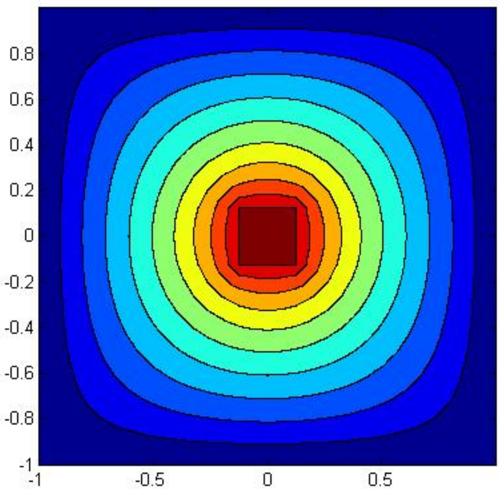


Figure 7.4: Corresponding choice of Ω^i



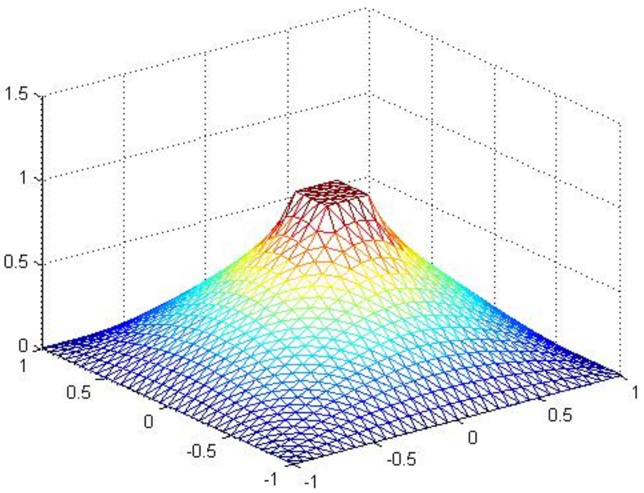


Figure 7.5: Test problem 1: solution for the choice of $\Omega^i : \ell^i = 0.4$.

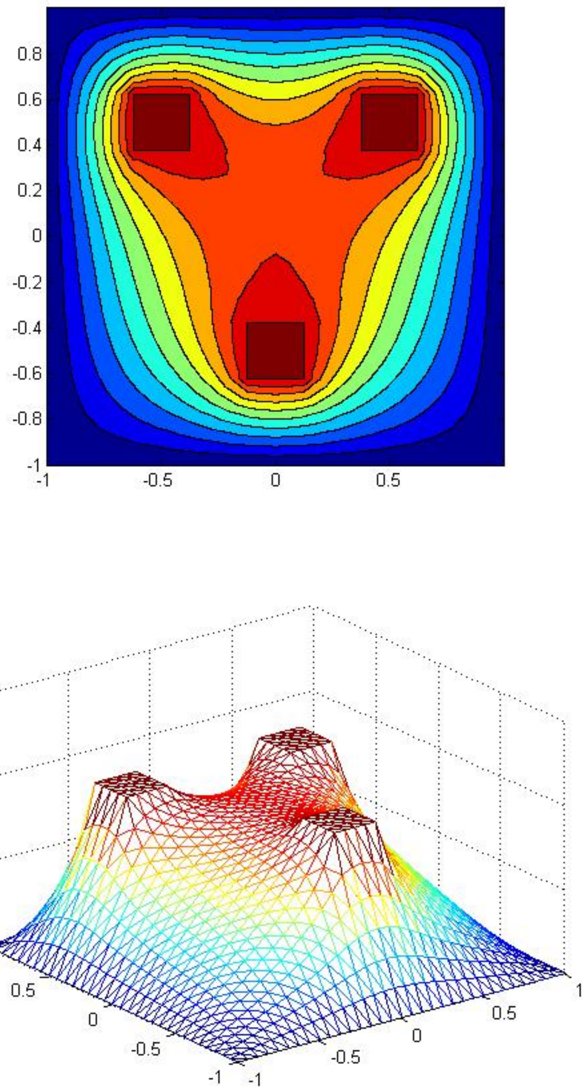


Figure 7.6: Test problem 2: solution for the choice of $\Omega^i : \ell^i = 0.4$.

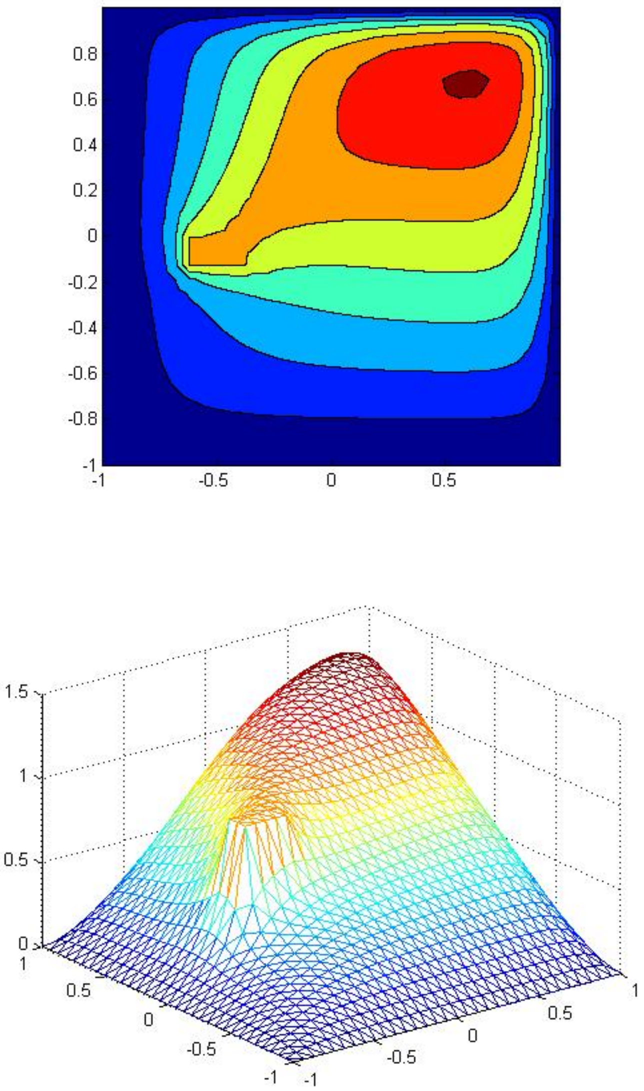


Figure 7.7: Test problem 3: convection diffusion problem $\alpha = 10^{-1}$ and $b_1 = b_2 = 1$.

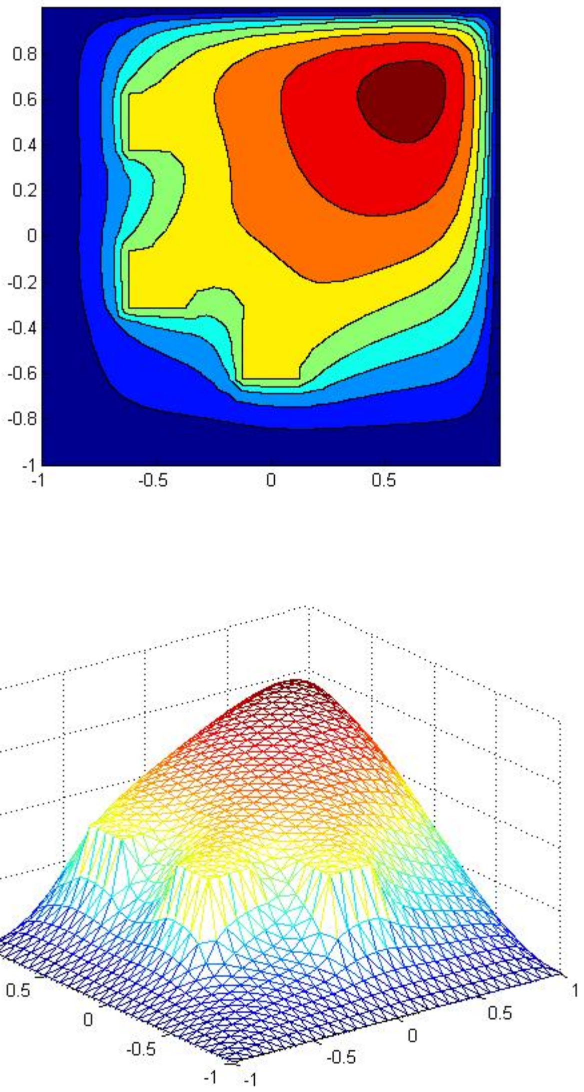


Figure 7.8: Test problem 4: convection diffusion problem $\alpha = 10^{-1}$ and $b_1 = b_2 = 1$.

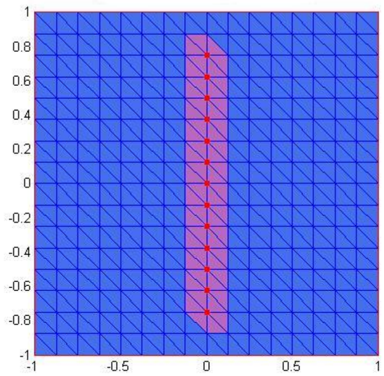


Figure 7.9: choice of Ω^i

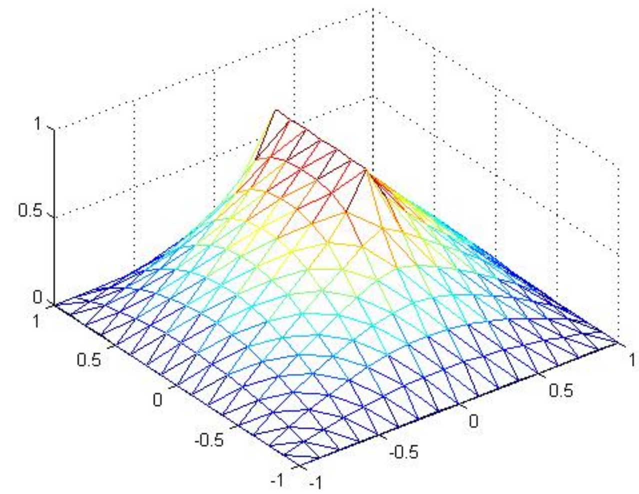


Figure 7.10: solution for the set measure 0.

Chapter 8

Conclusion And Future Work

8.1 Conclusion

In this thesis, we presented the variational inequalities (2.3.1) and provided relevant results in order to justify the existence and uniqueness of its solution. We have shown that these variational inequality problems can be expressed as LCP problems by using the results from [88]. A detailed discussion was given in Chapter 3. In this chapter, we have shown that these variational inequalities problems could be solved by using the optimization techniques such as LP or QP problems. We showed that solution of these types of problems can be achieved by solving a constrained minimization problem over a non empty closed convex set. In particular we have seen that LCP serves as a bridge between variational inequality problems and optimization problems.

To solve the variational inequality problem, we developed a non overlapping domain decomposition method in Chapter 5. The main advantage in using domain decomposition methods was that we were able to convert our problem into two subproblems one of which is variational inequality in subdomain Ω^i and another which is a PDE in complementary subdomain Ω^e. This approach was shown to yield a sequence of two decoupled problems (5.2.13a) and (5.2.13d), with Dirichlet boundary conditions on subdomain Ω^e together with the problem set on the interface Γ, and a variational inequality in a subdomain Ω^i. Therefore, we converted the variational inequality problem from the whole domain to a reduced variational inequality problem (5.3.3) in the subdomains containing the support of the obstacles. However, it has been found that many domain decomposition methods cannot solve convection diffusions problems very efficiently. This is because these algorithms often do not care about the hyperbolic nature of convection diffusion problems. Therefore, we chose to investigate domain decomposition algorithm to be applied to convection diffusion problems. We presented the standard convection diffusion problem in Chapter 2, where it was discussed that these problems do not possess a minimization formulation due to the fact that they are non-symmetric in nature. We also showed that by using an appropriate substitution, we were able to convert these problems into symmetric reaction diffusion problems. We can then apply any QP solver to determine a solution, which could then be used to obtain the solution of convection diffusion problem. A validation of domain decomposition methods for these problems was described in Chapter 7. In this thesis, we also described parabolic variational inequalities. We used finite element method to discretize in space and to convert into fully discrete problem we used backward Euler and the Crank Nicolson methods. We extended the domain decomposition methods given in Chapter 5 to parabolic variational inequalities.

We developed algorithms for the domain decomposition methods to solve the variational inequality problems of both elliptic and parabolic type as well as the variational inequality problems with convection diffusion parameters. The algorithm (5.4.0.1) is a direct three step procedure, requiring the solution of the reduced system (5.3.3). This reduced system contains the partial Schur complement as an element, therefore it can be expensive. To overcome this, we presented two algorithms (5.4.1.1) and (5.4.1.2) at the expense of iterative procedures. In these algorithms, the solutions on Ω^i and Γ are obtained in an iterative manner. We also described Newton's method, Newton-GMRES method and preconditioned Newton-GMRES method to solve the nonlinear interface problem at Γ. In Chapter 7 we presented some numerical results for the solution of two dimensional obstacle problems of both elliptic and parabolic type and also the obstacle problem with convection diffusion parameters in order to validate our domain decomposition algorithms. We observed that the global solution to these problems without the use of domain decomposition can be computationally demanding. Furthermore, the results given in Chapter 7 showed the dependence on the levels of refinement. When we solved the problem by using the direct reduced QP direct algorithm, the number of iterations was not only reduced, but also independent of levels of refinement. The Picard iterative algorithm (5.4.1.2) showed dependence with respect to the levels of refinement. In order to deal with this dependence, we applied Newton's method to determine the solution for the non-linear problem at the interface Γ, and found that the algorithm required only one iteration to achieve convergence, for the elliptic obstacle problem, which is a good achievement. We improved this algorithm further by introducing the Newton-GMRES and preconditioned Newton-GMRES methods at the interface in order to make the algorithm Jacobian free. Now, as opposed to constructing the Jacobian matrix, we instead made use of the Jacobian-vector products produced through the use of GMRES. This came at the expense of GMRES iterations, however was found to deliver promising results.

8.2 Future work

We are interested in the implementation of the inverse of the partial Schur complement using standard domain decomposition methods for PDE in (5.2.4). In particularly, we exploit the fact that the Steklov-Poincaré operators arising in a non-overlapping domain decomposition methods are coercive and continuous with respect to Sobolev norm of index 1/2 as given in [3], [4], [5]. Therefore, we would be interested to determine some suitable preconditioner for our partial Schur Steklov-Poincaré operator.

In this thesis, we presented a generic algorithm for domain decomposition method and validated our results for the obstacle problems of both elliptic and parabolic type as well as the obstacle problem with convection diffusion parameters. We are now interested to validate our domain decomposition algorithm for contact problems.
The main motivation behind any domain decomposition method is to test the algorithm on a parallel computer. So we would also be interested in testing our algorithms on a parallel computer.

List of References

[1] P.A. Absil and A.L. Tits. Newton-KKT interior-point methods for indefinite quadratic programming. *Computational Optimization and Applications*, 36(1):5–41, 2007.

[2] S.S. Antman. The influence of elasticity on analysis: Modern developments. *American Mathematical Society*, 9(3), 1983.

[3] M. Arioli, D. Kourounis, and D. Loghin. Discrete fractional Sobolev norms for domain decomposition preconditioning. Technical report, Technical report RAL-TR-2008-031, Rutherford Appleton Laboratory, Chilton, UK, 2008.

[4] M. Arioli and D. Loghin. Matrix square-root ˙ Technical report, Technical Report RAL-TR-2008-003, Rutherford Appleton Laboratory, Didcot, UK, 2008.

[5] M. Arioli and D. Loghin. Discrete interpolation norms with applications. *SIAM Journal on Numerical Analysis*, 47(4):2924–2951, 2009.

[6] A Auge, A Kapurkin, G Lube, and FC Otto. Domain decomposition methods for convection dominated problems.

[7] Owe Axelsson and Ivar Gustafsson. A modified upwind scheme for convective transport equations and the use of a conjugate gradient method for the solution of non-symmetric systems of equations. *IMA Journal of Applied Mathematics*, 23(3):321–337, 1979.

[8] L. Badea. On the Schwarz alternating method with more than two subdomains for nonlinear monotone problems. *SIAM Journal on Numerical Analysis*, pages 179–204, 1991.

[9] L. Badea and J. Wang. An additive Schwarz method for variational inequalities. *Mathematics of Computation*, 69(232):1341–1354, 2000.

[10] Lori Badea. Multigrid methods for some quasi-variational inequalities.

[11] Petter E Bjørstad and Olof B Widlund. To overlap or not to overlap: A note on a domain decomposition method for elliptic problems. *SIAM Journal on Scientific and Statistical Computing*, 10(5):1053–1061, 1989.

[12] James H Bramble, Joseph E Pasciak, and Jinchao Xu. Parallel multilevel preconditioners. *Mathematics of Computation*, 55(191):1–22, 1990.

[13] J.H. Bramble, J.E. Pasciak, and A.T. Vassilev. Analysis of non-overlapping domain decomposition algorithms with inexact solves. *Mathematics of Computation*, 67(221):1–19, 1998.

[14] S.C. Brenner and L.R. Scott. *The Mathematical Theory of Finite Element Methods*, volume 15. Springer Verlag, 2008.

[15] Alexander N Brooks and Thomas JR Hughes. Streamline upwind/petrov-galerkin formulations for convection dominated flows with particular emphasis on the incompressible navier-stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32(1):199–259, 1982.

[16] Alexander Nelson Brooks. *A Petrov-Galerkin finite element formulation for convection dominated flows*. PhD thesis, California Institute of Technology, 1981.

[17] P.N. Brown and Y. Saad. Hybrid krylov methods for nonlinear systems of equations. *SIAM Journal on Scientific and Statistical Computing*, 11(3):450–481, 1990.

[18] X.C. Cai, W.D. Gropp, and D.E. Keyes. A comparison of some domain decomposition algorithms for nonsymmetric elliptic problems. In *Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations, Philadelphia, PA*, 1992.

[19] X.C. Cai, W.D. Gropp, and D.E. Keyes. A comparison of some domain decomposition and ilu preconditioned iterative methods for nonsymmetric elliptic problems. *Numerical Linear Algebra with Applications*, 1(5):477–504, 1994.

[20] X.C. Cai, Y. Saad, and Army High Performance Computing Research Center. Overlapping domain decomposition algorithms for general sparse matrices. *Numerical Linear Algebra with Applications*, 3(3):221–237, 1996.

[21] Xiao-Chuan Cai, David E Keyes, and Venkatasubramanian Venkatakrishnan. Newton-krylov-schwarz: An implicit solver for cfd. Technical report, DTIC Document, 1995.

[22] T.F. Chan and K.R. Jackson. Nonlinearly preconditioned krylov subspace methods for discrete newton algorithms. *SIAM Journal on Scientific and Statistical Computing*, 5(3):533–542, 1984.

[23] T.F. Chan and T.P. Mathew. Domain decomposition algorithms. *Acta Numerica*, 3(1):61–143, 1994.

[24] Tony F Chan and Danny Goovaerts. On the relationship between overlapping and nonoverlapping domain decomposition methods. *SIAM Journal on Matrix Analysis and Applications*, 13(2):663–670, 1992.

[25] Juan Chen, Yiming Chen, Yufeng Liu, and Wenjuan Li. Domain decomposition method for a kind of parabolic variational inequality with fourth-order. 2012.

[26] I Christie, DF Griffiths, AR Mitchell, and OC Zienkiewicz. Finite element methods for second order differential equations with significant first derivatives. *International Journal for Numerical Methods in Engineering*, 10(6):1389–1396, 1976.

[27] P.G. Ciarlet. *The Finite Element Method for Elliptic Problems*, volume 4. North-Holland, 1978.

[28] Ray W Clough. The finite element method in plane stress analysis. 1960.

[29] Yuying Coleman, Thomas F Li. A reflective newton method for minimizing a quadratic function subject to bounds on some of the variables. *SIAM Journal on Optimization*, 6:1040–1058, 1996.

[30] R.W. Cottle, F. Giannessi, and J.L. Lions. *Variational Inequalities and Complementarity Problems: Theory and Applications.* J. Wiley, 1980.

[31] R. Courant. Variational methods for the solution of problems of equilibrium and vibrations. *Lecture Notes in Pure and Applied Mathematics*, pages 1–1, 1994.

[32] Richard Courant. Variational methods for the solution of problems of equilibrium and vibrations. *Bull. Amer. Math. Soc*, 49(1):23, 1943.

[33] Jean Donea and Antonio Huerta. *Finite Element Methods for Flow Problems.* Wiley, 2003.

[34] M Dryja and O Widlund. *An Additive Variant of the Schwarz Alternating Method for the Case of Many Subregions.* Courant Institute of Mathematical Sciences. Ultracomputer Research Laboratory, 1987.

[35] Maksymilian Dryja. An additive schwarz algorithm for two-and three-dimensional finite element elliptic problems. *Tony Chan, Roland Glowinski, Jacques P eriaux, and Olof Widlund, editors, Domain Decomposition Methods, Philadelphia, PA*, 1989.

[36] Maksymilian Dryja et al. Substructuring methods for parabolic problems. In *Proceedings of Fourth International Symposium on Domain Decomposition Methods for Partial Differential Equations, SIAM, Philadelphia*, pages 264–271, 1991.

[37] Maksymilian Dryja, Barry F Smith, and Olof B Widlund. Schwarz analysis of iterative substructuring algorithms for elliptic problems in three dimensions. *SIAM Journal on Numerical Analysis*, 31(6):1662–1694, 1994.

[38] Maksymilian Dryja and Olof B Widlund. *Multilevel Additive Methods for Elliptic Finite Element Problems.* New York University, Department of Computer Science, Courant Institute of Mathematical Sciences, 1990.

[39] Maksymilian Dryja and Olof B Widlund. Domain decomposition algorithms with small overlap. *SIAM Journal on Scientific Computing*, 15(3):604–620, 1994.

[40] G. Duvaut and J.L. Lions. *Inequalities in Mechanics and Physics*. Springer-Verlag Berlin, 1976.

[41] Kenneth Eriksson and Claes Johnson. Adaptive streamline diffusion finite element methods for stationary convection-diffusion problems. *Mathematics of Computation*, 60(201):167–188, 1993.

[42] Richard E Ewing, RD Lazarov, and PS Vassilevski. Local refinement techniques for elliptic problems on cell-centered grids. i. error analysis. *Mathematics of Computation*, 56(194):437–461, 1991.

[43] F. Facchinei and J.S. Pang. *Finite-Dimensional Variational Inequalities and Complementarity Problems*, volume 1. Springer Verlag, 2003.

[44] Gaetano Fichera. Sul problema elastostatico di Signorini con ambigue condizioni al contorno (on the elastostatic problem of Signorini with ambiguous boundary conditions). *Matematiche e Naturali*, page 8 34(2): 138:142, 1963.

[45] Gaetano Fichera. Elastostatic problems with unilateral constraints : the Signorini problem with ambiguous boundary conditions. page 613:679, 1964.

[46] M Garbey, Yu A Kuznetsov, and Yu V Vassilevski. A parallel schwarz method for a convection-diffusion problem. *SIAM Journal on Scientific Computing*, 22(3):891–916, 2000.

[47] A. Geletu. Solving optimization problems using the matlab optimization toolbox-a tutorial. *Mathematical Programming*, 2007.

[48] E Gelman and J Mandel. On multilevel iterative methods for optimization problems. *Mathematical Programming*, 48(1-3):1–17, 1990.

[49] R. Glowinski. *Numerical Methods for Nonlinear Vriational Problems*. Springer Verlag, 2008.

[50] R. Glowinski, J.L. Lions, and R. Trémolières. *Numerical Analysis of Variational Inequalities*, volume 8. North-Holland, 1981.

[51] Roland Glowinski. *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*. Number 30. SIAM, 1988.

[52] Carsten Gräser and Ralf Kornhuber. Multigrid methods for obstacle problems. 2008.

[53] Bernd Heinrich. *Finite Difference Methods on Irregular Networks: a generalized approach to second order elliptic problems*. Birkhäuser, 1987.

[54] JC Heinrich, PS Huyakorn, OC Zienkiewicz, and AR Mitchell. An upwindfinite element scheme for two-dimensional convective transport equation. *International Journal for Numerical Methods in Engineering*, 11(1):131–143, 1977.

[55] Thomas JR Hughes. A simple scheme for developing upwindfinite elements. *International Journal for Numerical Methods in Engineering*, 12(9):1359–1365, 1978.

[56] Thomas JR Hughes and Alec Brooks. A multidimensional upwind scheme with no crosswind diffusion. *Finite Element Methods for Convection Dominated Flows, AMD*, 34:19–35, 1979.

[57] TJR Hughes and A Brooks. A theoretical framework for petrov-galerkin methods with discontinuous weighting functions: Application to the streamline-upwind procedure. *Finite Elements in Fluids*, 4:47–65, 1982.

[58] Arlen Mikhailovich Il'in. Differencing scheme for a differential equation with a small parameter affecting the highest derivative. *Mathematical Notes*, 6(2):596–602, 1969.

[59] Caroline Japhet. Domain decomposition methods for convection-diffusion equation.

[60] R. Jeltch, T.T. LI, and I.H. Sloan. Some topics in industrial & applied mathematics (Series in Contemporary Applied Mathematics, vol. 8). 2007.

[61] B. Jiang, J.C. Bruch Jr, and J.M. Sloss. A nonoverlapping domain decomposition method for variational inequalities derived from free boundary problems. *Numerical Methods for Partial Differential Equations*, 22(1):1–17, 2006.

[62] Z. Jinping and W. Lieheng. Nonoverlapping domain decomposition method for solving obstacle variational inequalities. *Institute of Computational Mathematics and Scienctific/Engineering Computing, Chinese Academy of Science, Beijing*, 19(3):421–430, 1997.

[63] V John, JM Maubach, and L Tobiska. Nonconforming streamline-diffusion-finite-element-methods for convection-diffusion problems. *Numerische Mathematik*, 78(2):165–188, 1997.

[64] C. Johnson. *Numerical Solution of partial Differential Equations by the Finite Element Method*, volume 32. Cambridge University Press Cambridge, 1987.

[65] Claes Johnson, Uno Nävert, and Juhani Pitkäranta. Finite element methods for linear hyperbolic problems. *Computer Methods in Applied Mechanics and Engineering*, 45(1):285–312, 1984.

[66] Claes Johnson and Jukka Saranen. Streamline diffusion methods for the incompressible euler and navier-stokes equations. *Mathematics of Computation*, 47(175):1–18, 1986.

[67] Andrei Kapurkin and Gert Lube. A domain decomposition method for singularly perturbed elliptic problems. *Notes on Numerical Fluid Mechanics*, 49:151–151, 1995.

[68] J. Kepler. Uzawa-type methods for the obstacle problem.

[69] N. Kikuchi and J.T. Oden. *Contact Problems in Elasticity: A Study of Variational Inequalities and Finite Element Methods*, volume 8. Society for Industrial Mathematics, 1988.

[70] D. Kinderlehrer. Elliptic variational inequalities. In *Proc. Intl. Congress of Math. Vancouver*, pages 269–273, 1974.

[71] D. Kinderlehrer and G. Stampacchia. *An introduction to variational inequalities and their applications*, volume 31. Society for Industrial Mathematics, 2000.

[72] D.A. Knoll and D.E. Keyes. Jacobian-free newton–krylov methods: a survey of approaches and applications. *Journal of Computational Physics*, 193(2):357–397, 2004.

[73] Ralf Kornhuber. Monotone multigrid methods for elliptic variational inequalities I. *Numerische Mathematik*, 69:167–167, 1994.

[74] Ralf Kornhuber and Harry Yserentant. Multilevel methods for elliptic problems on domains not resolved by the coarse grid. *Contemporary Mathematics*, 180:49–49, 1994.

[75] A. Kufner and John. *Function Spaces*, volume 3. Springer, 1977.

[76] Y.A. Kuznetsov, P. Neittaanmäki, and P. Tarvainen. Overlapping domain decomposition methods for the obstacle problem. *Contemporary Mathematics*, 157:271–271, 1994.

[77] Yu A Kuznetsov. Domain decomposition methods for unsteady convection-diffusion problems. In *Proceedings of the Ninth International Conference in Computing Methods in Applied Sciences and Engineering*, pages 211–227, 1990.

[78] Yu M Laevskii and SV Gololobov. Explicit-implicit domain decomposition methods for solving parabolic equations. *Siberian Mathematical Journal*, 36(3):506–516, 1995.

[79] RD Lazarov, Ilya D Mishev, and Panayot S Vassilevski. Finite volume methods for convection-diffusion problems. *SIAM Journal on Numerical Analysis*, 33(1):31–55, 1996.

[80] C.E Lemke. Bimartix equilibrium points and mathematical programming. *Manag, Sci*, 11:681–689, 1965.

[81] J. Lions. On free surface problems: Methods of variational and quasi variational inequalities. *Computational Mechanics*, pages 129–148, 1975.

[82] JL Lions. Partial differential inequalities. *Russian Mathematical Surveys*, 27(2):91–159, 1972.

[83] J.L. Lions, E. Magenes, and P. Kenneth. *Non-Homogeneous Boundary Value Problems and Applications*, volume 1. Springer-Verlag Berlin, 1973.

[84] Pierre Louis Lions. On the Schwarz alternating method II, in T. Chan et al., eds.,. In *Domain Decomposition Methods*, pages 47–70. SIAM, Philadelphia, PA, 1989.

[85] P.L. Lions. On the Schwarz alternating method. I. In *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 1–42, 1988.

[86] P.L. Lions. On the Schwarz alternating method iii: a variant for nonoverlapping subdomains. In *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations*, volume 6, pages 202–223. SIAM: Philadelphia, PA, 1990.

[87] Jan Mandel. A multilevel iterative method for symmetric, positive definite linear complementarity problems. *Applied Mathematics and Optimization*, 11(1):77–95, 1984.

[88] K.G. Murty. *Linear Complementarity, Linear and Nonlinear Programming.* Heldermann Berlin, 1988.

[89] Uno Nävert. *A Finite Element Method for Convection-Diffusion Problems.* Chalmers University of Technology, Department of Computer Sciences, 1982.

[90] M.A. Noor and JR Whiteman. Error bounds for finite element solutions of mildly nonlinear elliptic boundary value problems. *Numerische Mathematik*, 26(1):107–116, 1976.

[91] J.T. Oden and J.N. Reddy. *An Introduction to the Mathematical Theory of Finite Elements.* Wiley, 1976.

[92] A. Quarteroni and A. Valli. *Domain Decomposition Methods for Partial Differential Equations.* Oxford University Press, USA, 1999.

[93] Rolf Rannacher and Guohui Zhou. *Analysis of a Domain Splitting Method for Nonstationary Convection Diffusion Problems.* IWR, 1994.

[94] Samia Riaz and D. Loghin. Domain decomposition method for obstacle problem. In *proceedings from the Twenty First International Conference on Dpmain Decomposition Methods, June 2012, Rennes, France.* Accepted.

[95] J.F. Rodrigues. *Obstacle Problems in Mathematical Physics.* Number 134. Elsevier Science Ltd, 1987.

[96] Matthew Rudd and Klaus Schmitt. Variational inequalities of elliptic and parabolic type. *Taiwanese Journal of Mathematics*, 6(3):pp–287, 2002.

[97] AK Runchal. Convergence and accuracy of three finite difference schemes for a two-dimensional conduction and convection problem. *International Journal for Numerical Methods in Engineering*, 4(4):541–550, 1972.

[98] Youcef Saad and Martin H Schultz. Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, 1986.

[99] Yousef Saad. *Iterative Methods for Sparse Linear Systems.* Siam, 2003.

[100] Samarskiui. *The Theory of Difference Schemes.*

[101] J. Schöberl. Solving the Signorini problem on the basis of domain decomposition techniques. *Computing*, 60(4):323–344, 1998.

[102] Joachim Schöberl. Solving the Signorini problem on the basis of domain decomposition techniques. *Computing*, 60(4):323–344, 1998.

[103] Hermann Schwarz. Uber einen grenzbergang durch alternierendes verfahren, ges, math. *Vierteljahrsschrift der Naturforschenden Gesellschaft in Zrich*, 15:Abhandlungen, Bd, 1:133–143, 1870.

[104] DB Spalding. A novel finite difference formulation for differential expressions involving both first and second derivatives. *International Journal for Numerical Methods in Engineering*, 4(4):551–559, 1972.

[105] E. Suli. *Finite Element Methods for Partial Differential Equations.* Oxford University Computing Laboratory, 2002.

[106] X-C Tai. Convergence rate analysis of domain decomposition methods for obstacle problems. *Journal of Numerical Mathematics*, 9(3):233–252, 2001.

[107] X.C. Tai. Rate of convergence for some constraint decomposition methods for nonlinear variational inequalities. *Numerische Mathematik*, 93(4):755–786, 2003.

[108] Xue-cheng Tai. Parallel function decomposition and space decomposition methods with applications to optimization, splitting and domain decomposition. In *Beijing Mathematics*. Citeseer, 1992.

[109] Xue-Cheng Tai. Parallel function decomposition and space decomposition methods Part I. function decomposition. In *Beijing Mathematics*. Citeseer, 1995.

[110] Xue-Cheng Tai. 1. some new domain decomposition and multigrid methods for variational inequalities. *Algorithms*, 16(6):7, 2002.

[111] Xue-Cheng Tai, Bjorn-ove Heimsund, and Jinchao Xu. Rate of convergence for parallel subspace correction methods for nonlinear variational inequalities. In *Thirteenth International Domain Decomposition Conference*, pages 127–138, 2001.

[112] Vidar Thomée. Some interior estimates for semidiscrete galerkin approximations for parabolic equations. *Mathematics of Computation*, 33(145):37–62, 1979.

[113] Lutz Tobiska and Rüdiger Verfürth. Analysis of a streamline diffusion finite element method for the stokes and navier-stokes equations. *SIAM Journal on Numerical Analysis*, 33(1):107–127, 1996.

[114] A. Toselli and O.B. Widlund. *Domain Decomposition Methods–Algorithms and Theory*, volume 34. Springer Verlag, 2005.

[115] PS Vassilevski, SI Petrova, and RD Lazarov. Finite difference schemes on triangular cell-centered grids with local refinement. *SIAM Journal on Scientific and Statistical Computing*, 13(6):1287–1313, 1992.

[116] O.B. Widlund. Some Schwarz methods for symmetric and nonsymmetric elliptic problems. In *Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations*, number 55, page 19. Society for Industrial & Applied, 1992.

[117] P. Wilmott, S. Howison, and J. Dewynne. *The Mathematics of Financial Derivatives: A Student Introduction*. Cambridge University Press, 1995.

[118] J. Xu and J. Zou. Some nonoverlapping domain decomposition methods. *SIAM review*, pages 857–914, 1998.

[119] Jinchao Xu. Iterative methods by space decomposition and subspace correction. *SIAM review*, 34(4):581–613, 1992.

[120] Katsu Yamane and Yoshihiko Nakamura. A numerically robust lcp solver for simulating articulated rigid bodies in contact. *Proceedings of Robotics: Science and Systems IV, Zurich, Switzerland*, 19:20, 2008.

[121] Y. Ye. On affine scaling algorithms for nonconvex quadratic programming. *Mathematical Programming*, 56(1):285–300, 1992.

[122] J. Zeng. Geometric convergence of overlapping Schwarz methods for obstacle problems. *Domain Decomposition Methods*, pages 237–241, 1998.

[123] J. Zeng and S. Zhou. On monotone and geometric convergence of Schwarz methods for two-sided obstacle problems. *SIAM Journal of Numerical Analysis*, pages 600–616, 1998.

[124] Xiao Lan Zhang. Numerical analysis of american option pricing in a jump-diffusion model. *Mathematics of Operations Research*, 22(3):668–690, 1997.

[125] Xuejun Zhang. Multilevel additive schwarz methods. Technical report, Citeseer, 1991.

[126] Xuejun Zhang. Multilevel schwarz methods. *Numerische Mathematik*, 63(1):521–539, 1992.

[127] Guohui Zhou. *A New Domain Decomposition Method for Convection Dominated Problems*. IWR, 1995.

[128] S. Zhou. An additive Schvvarz algorithm for a variational inequality. 1997.

[129] S. Zhou, J. Zeng, and X. Tang. Generalized Schwarz algorithm for obstacle problems. *Computers & Mathematics with Applications*, 38(7-8):263–271, 1999.

[130] Shuzi Zhou and Dehua Wang. Domain decomposition method for a parabolic variational inequality. *Applied Mathematics and Computation*, 166(1):213–223, 2005.

[131] Milovs Zlámal. On the finite element method. *Numerische Mathematik*, 12(5):394–409, 1968.