8-2015

Domain Decomposition Methods for Discontinuous Galerkin Approximations of Elliptic Problems

Craig Dwain Collins
University of Tennessee - Knoxville, ccolli37@vols.utk.edu

To the Graduate Council:

I am submitting herewith a dissertation written by Craig Dwain Collins entitled "Domain Decomposition Methods for Discontinuous Galerkin Approximations of Elliptic Problems." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Mathematics.

Ohannes A. Karakashian, Major Professor

We have read this dissertation and recommend its acceptance:

Michael Berry, Xiaobing Feng, Clayton Webster, Steven Wise

Accepted for the Council:

Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

Domain Decomposition Methods for Discontinuous Galerkin Approximations of Elliptic Problems

A Dissertation Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Craig Dwain Collins

August 2015

Dedication

To my mother Maricha, my father Jake, my wife Robin, and my daughter Maisie.
Thank you.

Acknowledgements

I must first thank my advisor, Dr. Ohannes Karakashian, for his outstanding tutelage and his immeasurable patience. The opportunity to learn from someone with his level of expertise is a gift I had not imagined. His example is one I will strive to emulate throughout the rest of my career.

I must also thank Dr. Steven Wise for his mentorship and encouragement. His effortless mastery of bringing out the best in his students is nothing short of amazing.

I would also like to thank the remaining members of my dissertation committee, Dr. Xiaobing Feng, Dr. Clayton Webster, and Dr. Michael Berry for taking time from their incredibly busy schedules to offer invaluable comments and questions.

I would also like to thank Dr. Tan Zhang, Dr. Kelly Pearson, and Dr. Renee Fister for their training and encouragement. Without them, I would never have found this path.

I must also thank Ben Walker, Angela Woofter, and Tina Murr for their unending supply of help (and patience) during my time at UTK.

No acknowledgement would be properly complete without offering thanks to Pam Armentrout, without whom I doubt any graduate student at UTK would ever succeed. I have rarely met anyone *that* good at her job – I can only dream of that type of excellence.

Finally, I must thank my family – my parents Jake and Risha, who taught me to believe in myself; my wife Robin, whose love and support are without measure; and my daughter Maisie, who has brought so much joy to all our lives.

Abstract

The application of the techniques of domain decomposition to construct effective preconditioners for systems generated by standard methods such as finite difference or finite element methods has been well-researched in the past few decades. However, results concerning the application of these techniques to systems created by the discontinuous Galerkin method (DG) are much more rare.

This dissertation represents the effort to extend the study of two-level nonoverlapping and overlapping additive Schwarz methods for DG discretizations of second- and fourth-order elliptic partial differential equations. In particular, the general Schwarz framework is used to find theoretical bounds for the condition numbers of the preconditioned systems corresponding to both the nonoverlapping and overlapping additive Schwarz methods. In addition, the impact on the performance of the preconditioners caused by varying the penalty parameters inherent to DG methods is investigated. Another topic of investigation is the choice of course subspace made for the two-level Schwarz methods.

The results of in-depth computational experiments performed to validate and study various aspects of the theory are presented. In addition, the design and implementation of the methods are discussed.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The use of numerical methods to approximate solutions of partial differential equations (PDEs) is ubiquitous in many disciplines, including physics, chemistry, engineering, and biology. The approximation of a PDE solution involves, in general, two stages. The first is the discretization step, in which the continuous problem is replaced with a discrete formulation resulting in a system of equations involving a finite number of unknowns. One of the most widely used discretization methods is the *finite element method (FEM)*. The *discontinuous Galerkin (DG)* method is a particular finite element method which has gained widespread popularity due to its many beneficial attributes. Indeed, the discontinuous Galerkin method is especially well-suited to handle complex geometries and boundary conditions, facilitates adaptive implementation, and is eminently parallelizable. The second stage is the solution step, in which the discrete system is solved to produce the approximation. One drawback of the discontinuous Galerkin method is that the resulting systems are larger than those produced by other finite element methods. Given that a very large percentage of the CPU time is spent on the solution phase, the design of efficient solvers is of critical importance. *Domain decomposition (DD)* methods offer an elegant solution to this dilemma. While these techniques have been successfully applied to many other discretization methods, their application

to discontinuous Galerkin is relatively recent. The focus of this dissertation is to investigate the effectiveness of the techniques of domain decomposition in creating efficient preconditioners for systems generated by the discontinuous Galerkin method.

1.1 Background

The discontinuous Galerkin method was developed at Los Alamos Scientific Laboratory in the early 1970's and was first described by Reed and Hill in [36] as a method for approximating solutions for hyperbolic problems. Discontinuous Galerkin formulations for the second order elliptic problem was introduced by Douglas and Dupont [20], while Baker presented the DG formulation of the fourth order elliptic problem [8]. These formulations belong to a class of discontinuous Galerkin methods known as *interior penalty* methods (IP) due to the presence of terms penalizing the jumps of functions over interelement boundaries. Further contributions developing IP methods include works by Wheeler [44], Arnold [5], Baker, Jureidini, and Karakashian [9], and Karakashian and Jureidini [30].

The latter part of the twentieth century saw a flurry of development for other variants of discontinuous Galerkin methods. The DG method for the Navier-Stokes equations was presented by Bassi and Rebay [11] in 1997, while the local discontinuous Galerkin method (LDG) by Cockburn and Shu [18] was introduced in 1998. Cockburn [17] presented a DG method for convection-dominated problems in 1999. The nonsymmetric interior method was presented by Baumann and Oden [13] around this time as well. A good source of the history of the development of discontinuous Galerkin methods during this time is given by Arnold, et al. [6], while a unified presentation gathering all the variants under the same framework is given by Arnold, et al. in [7].

The techniques of domain decomposition presented in 1870 by Schwarz [39] have been highly influential in the development of effective preconditioners for iterative solution methods. In recent years, a comprehensive framework – often referred to

as the Schwarz framework – has been developed and codified for finite difference methods and conforming finite element methods. In particular, the texts by Toselli and Widlund [43], Smith, Bjørstad, and Gropp [42], and Quarteroni and Valli [35], along with the article by Xu [45] are widely accepted as containing the fundamental descriptions of the Schwarz framework for the classical methods mentioned previously. Applications of the Schwarz framework to nonconforming methods and discontinuous Galerkin formulations, while once quite rare in the literature, are becoming more frequent. Early examples for these types of problems include Bassi and Rebay, who used a Schwarz preconditioner for GMRES in a DG formulation for the Navier-Stokes equations [12]. The works of Feng and Karakashian [23] and [24] develop two-level Schwarz preconditioners for DG formulations of the second- and fourth-order elliptic problems, respectively. Lasser and Toselli [33] present Schwarz preconditioners for the advection-diffusion problem, while Antonietti and Ayuso examine apply Schwarz methods to a mixed formulation of a second-order elliptic problem [2]. Recent activity has included extending Schwarz methods to include *hp*-adaptive DG methods [4] and the weakly over-penalized symmetric interior penalty method (WOPSIP) [10], [3], as well as fully nonlinear Hamilton-Jacobi-Bellman equations [41].

1.2 Summary of Dissertation

This dissertation contains three essential chapters. The focus of Chapter 2 is the development of two-level additive Schwarz preconditioners, (both nonoverlapping and overlapping) for the prototypical second order elliptic PDE, Poisson's equation. Although these methods have been proposed earlier in [23], here the formulations are presented using sleeker notation and updated proofs. In addition, the effect of choosing a completely discontinuous coarse subspace will be compared with using a continuous coarse subspace. In particular, the goal is to determine under what conditions the definition of the coarse space bilinear form given in [23] may be recovered. These conditions will be referred to as the *penalty compatibility conditions*.

A second update will involve proving the dependence of the nonoverlapping Schwarz method on the penalty parameter γ while also showing that the overlapping preconditioner is independent of γ. Computational simulations will be presented to corroborate the theory. Finally, the numerical simulations will be greatly expanded to include a variety of tests, including the scalability of the preconditioners with respect to the number of subdomains, the effect of failure to enforce the compatibility conditions, and the effects of increasing the amount of overlap for the overlapping preconditioner.

Chapter 3 focuses on the development of two-level additive Schwarz preconditioners for the prototypical fourth order elliptic PDE, the biharmonic equation. This chapter represents an extension to the method presented in [24], focusing on the same topics as those found in Chaper 2. In addition, the analysis for the corresponding two-level overlapping method is presented for the first time, as well as the results of computational simulations for both preconditioners.

Chapter 4 will provide details of the implementation of the computer program designed to perform the numerical simulations supporting the theoretical results shown herein. All aspects of the design will be discussed, including the creation and representation of the mesh partitions, the data structures defined in support of the program, and the implementation of the techniques of domain decomposition utilized to create the additive Schwarz preconditioners.

The plans for future research arising from the work done in this dissertation will be discussed in Chapter 5.

Finally, Appendix A will discuss the reference element, corresponding affine transformations, and its role in the computations. Appendix B will describe the formulation and contstruction of the stiffness matrices, load vectors, and the edge off-diagonal matrix blocks used in the DG construction.

Chapter 2

Second Order Elliptic Problems

2.1 Preliminaries

Throughout the entirety of this dissertation, we consider an open and bounded domain $\Omega \subset \mathbb{R}^d$ with $d = 2, 3$. For a domain $D \subseteq \mathbb{R}^d$ and real number $m \geq 0$, we will use the standard notation $H^m(D) = W^{m,2}(D)$ to denote the (Hilbert) Sobolev space of order m with inner product

$$(u, v)_{m,D} \;=\; \sum_{|\alpha| \leq m} \int_D D^\alpha u D^\alpha v \; dx$$

and norm

$$\|u\|_{m,D} \;=\; (u, u)_{m,D}^{1/2}.$$

For simplicity, the m will not be included when its value is zero.

We will also use the Sobolev seminorms defined by

$$|u|_{m,D} \;=\; \left\{ \sum_{|\alpha| = m} \int_D |D^\alpha u|^2 \; dx \right\}^{1/2}.$$

2.1.1 Model Problem

Consider a bounded and open domain $\Omega \subset \mathbb{R}^d$ with $d = 2, 3$ whose boundary $\partial\Omega$ is the union of two disjoint sets Γ_D and Γ_N. Dirichlet boundary data is given on Γ_D while the Neumann data corresponds to Γ_N. Furthermore, assume that Γ_D has positive $(d-1)$-dimensional measure. In this section we will examine the following model problem:

$$-\Delta u = f \quad \text{in} \quad \Omega, \tag{2.1}$$

$$u = g_D \quad \text{on} \quad \Gamma_D, \tag{2.2}$$

$$\nabla u \cdot n = g_N \quad \text{on} \quad \Gamma_N, \tag{2.3}$$

where n is the unit normal vector exterior to Ω.

2.1.2 Partitions of Ω

In order to develop the discontinuous Galerkin formulation for the model problem, we first need to introduce some notation. Let $\mathcal{T}_h = \{K_i \ : \ i = 1, 2, \ldots, m_h\}$ be a family of simplicial partitions of the domain Ω parametrized by $0 < h \leq 1$. The members of the partition $\mathcal{T}_h$ are referred to as *cells*. We use the term *edges* to describe the sides of the cells, regardless of dimension. Denote $\mathcal{E}_h^I$ as the set of all edges strictly interior to the domain and $\mathcal{E}_h^B$ as the set of all boundary edges. Every interior edge $e \in \mathcal{E}_h^I$ is shared by two cells. We will designate one of these cells as K^+ and the other as K^-. (Note that we may choose either cell as K^+ – the assignment is made in a completely arbitrary fashion. We will soon see that this choice is not irrelevant, but this fact does not change the arbitrary nature of this assignment. In a similar fashion, for an edge $e \in \mathcal{E}_h^B$, we denote the cell containing that edge as K^+ by convention.

We will assume that the partition $\mathcal{T}_h$ satisfies the following conditions:

(P1) The partition $\mathcal{T}_h$ is *conforming*; i.e., no hanging nodes are allowed. This assumption is only included to simplify the analysis that follows and may be relaxed in actual computations.

(P2) The cells of $\mathcal{T}_h$ are *shape-regular*. In other words, for every $K \in \mathcal{T}_h$, the ratio of the radius R of the circumscribed ball to the radius ρ of the inscribed ball of K is bounded; i.e., there exist positive constants c_1, c_2 such that

$$c_1 \leq \frac{R}{\rho} \leq c_2.$$

This condition ensures that no cell K has a very "skinny" angle. An equivalent way of stating this requirement is that the cells of $\mathcal{T}_h$ satisfy the *minimum angle condition*.

(P3) $\mathcal{T}_h$ is locally *quasi-uniform*; i.e., if two cells K_i and K_j share an edge then $\operatorname{diam}(K_i) \approx \operatorname{diam}(K_j)$.

(P4) For every boundary edge $e \in \mathcal{E}_h^B$, either $e \in \Gamma_D$ or $e \in \Gamma_N$. Set $\mathcal{E}_h^D$ and $\mathcal{E}_h^N$ as the set of boundary edges on Γ_D and Γ_N, respectively. We then have $\mathcal{E}_h^B = \mathcal{E}_h^D \cup \mathcal{E}_h^N$ and, since Γ_D and γ_N are disjoint, $\mathcal{E}_h^D \cap \mathcal{E}_h^N = \emptyset$.

(P5) Boundary edges $e \in \mathcal{E}_h^B$ of a cell are either Dirichlet or Neumann type, but not both.

Given such a partition $\mathcal{T}_h$ of Ω, we will make use of the *broken* Sobolev spaces

$$H^m(\mathcal{T}_h) \;=\; \prod_{K \in \mathcal{T}_h} H^m(K).$$

In this context we will consider K to be open so that the elements of $H^m(\mathcal{T}_h)$ are single-valued.

We will frequently integrate along edges, so for $e \in \mathcal{E}_h$ define

$$\langle u, v \rangle_e = \int_e uv \, ds \quad \text{and} \quad |u|_e = \langle u, u \rangle_e^{1/2}.$$

It is necessary to define values of functions in $H^m(\mathcal{T}_h)$ on the edges, so for $v \in H^m(\mathcal{T}_h)$, $m \geq 1$, denote

$$v^+ = v|_{K^+} \quad \text{and} \quad v^- = v|_{K^-}.$$

Furthermore, for $e \in \mathcal{E}_h$ let v_e^+ denote the trace on e of v^+ and v_e^- denote the trace on e of v^-.

The discontinuous Galerkin formulation will also include the *jumps* and *averages* of such traces. We define these by

$$\begin{aligned} [v] &= v_e^+ - v_e^-, \quad \text{for} \quad e \in \mathcal{E}_h^I, & [v] &= v_e^+, \quad \text{for} \quad e \in \mathcal{E}_h^B, \\ \{v\} &= \tfrac{1}{2}(v_e^+ + v_e^-), \quad \text{for} \quad e \in \mathcal{E}_h^I, & \{v\} &= v_e^+, \quad \text{for} \quad e \in \mathcal{E}_h^B. \end{aligned}$$

2.1.3 Discontinuous Galerkin Formulation

For $v \in H^s(\mathcal{T}_h)$, $s > 3/2$ we define

$$\{\partial_n v\}_e = \frac{1}{2}(\nabla v^+ + \nabla v^-) \cdot \mathbf{n}^+, \quad \text{for} \quad e \in \mathcal{E}_h^I, \tag{2.4}$$

where $\mathbf{n}^+$ is the unit normal vector outward to K^+. On the "Energy Space" $E_h = H^s(\mathcal{T}_h)$, $s > 3/2$ we consider the bilinear form $a_h^{\gamma_h}(\cdot, \cdot)$ given by

$$\begin{aligned} a_h^{\gamma_h}(u, v) &= \sum_{K \in \mathcal{T}_h} (\nabla u, \nabla v)_K - \sum_{e_h \in \mathcal{E}_h} \Big(\langle \{\partial_n u\}, [v] \rangle_{e_h} + \langle \{\partial_n v\}, [u] \rangle_{e_h} \Big) \\ &\quad + \sum_{e_h \in \mathcal{E}_h} \frac{\gamma_h}{|e_h|} \langle [u], [v] \rangle_{e_h}, \end{aligned} \tag{2.5}$$

where $\mathcal{E}_h = \mathcal{E}_h^I \cup \mathcal{E}_h^D$ and $|e_h|$ is a 1-dimensional measure of the size of e_h. For $d = 2$ we take $|e_h|$ to be the length of e_h while for $d = 3$ we set $|e_h| = \mathrm{diam}(e_h)$. The *penalty parameters* γ_h are a family of dimensionless constants that are required to be larger than some threshold γ_0 in order to guarantee the coercivity of the above bilinear form.

The formulation of this bilinear form corresponds to what is called a *symmetric* interior penalty DG (SIPG) method [5]. The treatment of the averages of the normal derivatives on interior edges defined in (2.4) is referred to as Arnold's formulation. A similar form referred to as Baker's formulation consists in defining the averages of the normal derivatives on interior edges by

$$\{\partial_n v\}_e \;=\; \nabla v^+ \cdot \mathbf{n}^+, \quad \text{for} \quad e \in \mathcal{E}_h^I. \tag{2.6}$$

In the following analysis we will investigate both formulations.

In order to define a weak formulation for the model problem (2.1)-(2.3) and the corresponding discontinuous Galerkin approximation u_h to the solution u, we introduce the linear functional

$$F_h(v) \;=\; (f, v) + \sum_{e_h \in \mathcal{E}_h^N} \langle g_N, v \rangle_{e_h} - \sum_{e_h \in \mathcal{E}_h^D} \left\langle g_D, \partial_n v - \frac{\gamma_h}{|e_h|} v \right\rangle_{e_h}. \tag{2.7}$$

Assuming that $u \in H^2(\Omega)$ is a solution of the model problem (2.1)-(2.3), an integration by parts reveals that the bilinear form (2.5) is consistent with the model problem in the sense that

$$a_h^{\gamma_h}(u, v) \;=\; F_h(v), \quad \forall v \in E_h. \tag{2.8}$$

We then introduce the discontinuous Galerkin finite element spaces

$$V^h \;=\; \prod_{K \in \mathcal{T}_h} \mathcal{P}_q(K), \quad q \geq 1,$$

where $\mathcal{P}_q(K)$ is the space of polynomials of degree less than or equal to q on K. Define the discontinuous Galerkin approximation $u_h \in V^h$ of u to be solution of

$$a_h^{\gamma_h}(u_h, v) \;=\; F_h(v), \quad \forall v \in V^h. \tag{2.9}$$

The form $a_h^{\gamma_h}$ is clearly symmetric and enjoys continuity and coercivity properties which we present below. The proofs are well-known and are therefore omitted.

Introducing the norm $\|\cdot\|_{1,h} : E_h \to \mathbb{R}$ defined by

$$\|v\|_{1,h} \;=\; \left\{ \sum_{K\in\mathcal{T}_h} \|\nabla v\|_K^2 + \sum_{e_h\in\mathcal{E}_h} |e_h| |\{\partial_n v\}|_{e_h}^2 + \sum_{e_h\in\mathcal{E}_h} \frac{\gamma_h}{|e_h|} |[v]|_{e_h}^2 \right\}^{1/2}, \tag{2.10}$$

we have the following continuity and coercivity properties.

Lemma 2.1.1. *For both the Arnold and the Baker formulations,*

$$|a_h^{\gamma_h}(u, v)| \le \|u\|_{1,h} \|v\|_{1,h}, \quad \forall u, v \in E_h, \tag{2.11}$$

and there exist positive constants γ_0 and c_a (depending only on q and the shape regularity of the cells in $\mathcal{T}_h$) such that if $\gamma_h \ge \gamma_0$ then

$$a_h^{\gamma_h}(v, v) \ge c_a \|v\|_{1,h}^2, \quad \forall v \in V^h. \tag{2.12}$$

It follows from these properties that the problem (2.9) is well-posed.

With the help of the basis functions of V^h, equation (2.9) transforms into an $N \times N$ linear system

$$A\,\mathbf{x} = \mathbf{b}, \tag{2.13}$$

where N denotes the dimension of V^h and the coefficient matrix $A \in \mathbb{R}^{N\times N}$, called the stiffness matrix, is symmetric and positive definite.

It is not hard to show that the (2-norm) condition number of A is of the order $O(\underline{h}^{-2})$ where $\underline{h} = \min_{K\in\mathcal{T}_h} h_K$. So the system (2.13) becomes ill-conditioned for

small $\underline{h}$. In addition, the size of the linear system becomes large. Consequently, it is not efficient to solve it directly using the classical iterative methods. On the other hand, if one can find a symmetric positive definite $N \times N$ matrix B such that BA is well-conditioned, then any of the classical iterative methods (in particular, the Conjugate Gradient method) works effectively on the preconditioned system

$$BA\,\mathbf{x} = B\mathbf{b}. \tag{2.14}$$

2.2 Basic results

For the remainder of this chapter, D will denote a simply-connected, open, bounded domain in $\mathbb{R}^d$ with Lipschitz boundary ∂D. We shall assume that D is star-shaped with respect to some point $\mathbf{x}_0 \in D$, i.e. the line segment $[\mathbf{x}_0, \mathbf{x}]$ is contained in D for all $\mathbf{x}$ in D. We shall also assume that D is shape regular in the sense that

$$(\mathbf{x} - \mathbf{x}_0) \cdot \mathbf{n}_D \geq c_D \ \mathrm{diam}(D) \quad \text{for a.e. } \mathbf{x} \in \partial D, \tag{2.15}$$

where $c_D = O(1)$ and $\mathbf{n}_D$ denotes the unit outward normal vector to ∂D.

For such domains D, the following trace inequality holds (cf. [14], [30]):

$$|v|^2_{\partial D} \ \leq \ c_{tr} \left(h_D^{-1} \|v\|^2_D + h_D \|\nabla v\|^2_D \right) \quad \forall v \in H^1(D), \tag{2.16}$$

where $h_D = \mathrm{diam}(D)$. For a cell $K \in \mathcal{T}_h$, which is assumed to be shape regular, we have the following inverse inequality

$$|v|_{j,K} \ \leq \ c_{inv} h_K^{i-j} |v|_{i,K} \quad \forall v \in \mathcal{P}_q(D), \ 0 \leq i \leq j \leq q, \tag{2.17}$$

the constant c_{inv} depending only on q.

Furthermore, the following basic approximation result is assumed to hold. For $K \in \mathcal{T}_h$, let $u \in H^m(K), m \geq 1$ an integer. Then there exists $\chi \in \mathcal{P}_q(K), \ 0 \leq q \leq$

$m-1$, such that

$$|u-\chi|_{j,K} \leq ch_K^{q+1-j}|u|_{q+1,K}, \quad 0 \leq j \leq q+1. \tag{2.18}$$

We shall make essential use of the fact that elements of V^h, can be approximated by *continuous* piecewise polynomial functions, specifically by elements of $V^h \cap C(\Omega)$; the degree of approximation being controlled, not surprisingly, by the jumps of the discontinuous function. Here we quote the following local version of a result from [31] and [32].

Theorem 2.1. *Let $\mathcal{T}_h$ be a conforming simplicial partition of Ω. Then for any $v_h \in V^h$ and multi-index α with $|\alpha| \leq q$ the following approximation results hold:*

(i) *There exists $\chi \in V^h \cap C(\Omega)$ satisfying*

$$\|D^\alpha(v_h-\chi)\|_K^2 \leq c \sum_{e_h \in \omega_h(K)} |e_h|^{1-2|\alpha|}|[v_h]|_{e_h}^2, \quad \forall K \in \mathcal{T}_h, \tag{2.19}$$

where $\omega_h(K)$ is the (local) set of edges in $\mathcal{E}_h^I$ emanating from the vertices of K.

(ii) *Let Γ denote a subset of $\mathcal{E}_h^B$ such that no cell in $\mathcal{T}_h$ can contain edges from both Γ and its complement. Suppose g is the restriction to Γ of some function in $V^h \cap C(\Omega)$. Then, there exists $\chi \in V^h \cap C(\Omega)$ with $\chi\big|_\Gamma = g$ that satisfies*

$$\|D^\alpha(v_h-\chi)\|_K^2 \leq c \sum_{e_h \in \omega_h(K)} |e_h|^{1-2|\alpha|}|[v_h]|_{e_h}^2 + c \sum_{e_h \in \Gamma \cap \partial K} |e_h|^{1-2|\alpha|}|v_h-g|_{e_h}^2. \tag{2.20}$$

The constant c depends only on q and the shape regularity of the cells.

Remark 2.2.1. *In [31] and [32] conditions are given under which the above result holds for nonconforming meshes. Extensions to curved boundaries under reasonable assumptions appear to be straightforward.*

The next result consists of a piecewise version of the trace inequality (2.16). Its proof was given in [23] but we give it here for the sake of completeness using less cumbersome notation.

Lemma 2.1.1. *Let D be the union of some collection $\mathcal{T}_D$ of cells from $\mathcal{T}_h$ and assume that it is star-shaped with respect to some $\mathbf{x}_0 \in D$ and that (2.15) holds. Then, there is a constant c depending only on the shape regularity of the cells such that*

$$|v|^2_{\partial D} \leq c\big(H_D^{-1}\|v\|^2_D + H_D|v|^2_{1,h,D}\big) \quad \forall v \in H^1(\mathcal{T}_D), \tag{2.21}$$

where $H_D = \operatorname{diam}(D)$ and the seminorm $|v|_{1,h,D}$ is given by

$$|v|^2_{1,h,D} = \sum_{K\in\mathcal{T}_D} \|\nabla v\|^2_K + \sum_{e_h\in\mathcal{E}^I_{h,D}} |e_h|^{-1}|[v]|^2_{e_h}, \tag{2.22}$$

and $\mathcal{E}^I_{h,D}$ denotes all edges $e_h \in \mathcal{E}^I_h$ which are also in the interior of D.

Proof. For any $K \in \mathcal{T}_D$, the divergence theorem gives

$$\begin{aligned}\int_{\partial K} v^2(\mathbf{x}-\mathbf{x}_0)\cdot\mathbf{n}_K\, ds &= \int_K \operatorname{div}((\mathbf{x}-\mathbf{x}_0)v^2)\, dx \\ &= d\,\|v\|^2_K + 2\int_K u\nabla v\cdot(\mathbf{x}-\mathbf{x}_0)dx,\end{aligned} \tag{2.23}$$

where $\mathbf{n}_K$ is the unit outward normal to K. Summing (2.23) over all $K \in \mathcal{T}_D$ we get

$$\begin{aligned}\int_{\partial D} v^2(\mathbf{x}-\mathbf{x}_0)\cdot\mathbf{n}_D\, ds = &\sum_{K\in\mathcal{T}_D}\left(d\,\|v\|^2_K + 2\int_K v\nabla v\cdot(\mathbf{x}-\mathbf{x}_0)\, dx\right) \\ &- \sum_{e_h\in\mathcal{E}^I_{h,D}} \big\langle(\mathbf{x}-\mathbf{x}_0)\cdot\mathbf{n}_{K^+}, (v^+)^2-(v^-)^2\big\rangle_{e_h}.\end{aligned} \tag{2.24}$$

In view of (2.15), for the term on the left side of (2.24) we have

$$cH_D|v|^2_{\partial D} \leq \int_{\partial D} v^2(\mathbf{x}-\mathbf{x}_0)\cdot\mathbf{n}_D\, ds, \tag{2.25}$$

whereas using the Cauchy-Schwarz and arithmetic-geometric mean inequalities and the fact that $|\mathbf{x}-\mathbf{x}_0| \le h_D$, we obtain

$$\sum_{K\in\mathcal{T}_D}\left(d\,\|v\|_K^2 + 2\int_K v\nabla v\cdot(\mathbf{x}-\mathbf{x}_0)\,dx\right) \le c\|v\|_D^2 + cH_D^2\sum_{K\in\mathcal{T}_D}\|\nabla v\|_K^2. \tag{2.26}$$

Now writing $(v^+)^2-(v^-)^2=[v](v^++v^-)$, for $e_h\in\mathcal{E}_{h,D}^I$ there holds

$$\left|\left\langle(\mathbf{x}-\mathbf{x}_0)\cdot\mathbf{n}_{K^+},(v^+)^2-(v^-)^2\right\rangle_{e_h}\right| \le H_D^2|e_h|^{-1}|[v]|_{e_h}^2 + |e_h|\big(|v^+|_{e_h}^2+|v^-|_{e_h}^2\big). \tag{2.27}$$

Using the trace inequality (2.16) and the fact that $\mathcal{T}_h$ is locally quasi-uniform and shape regular ($|e_h|\approx h_{K^+}\approx h_{K^-}$), we have

$$|e_h|\big(|v^+|_{e_h}^2+|v^-|_{e_h}^2\big) \le c\sum_{K=K^+,K^-}\big(\|v\|_K^2+h_K^2\|\nabla v\|_K^2\big). \tag{2.28}$$

From the last two inequalities and using $h_K\le h_D$, we obtain

$$\begin{aligned}\sum_{e_h\in\mathcal{E}_{h,D}^I}&\left|\left\langle(\mathbf{x}-\mathbf{x}_0)\cdot\mathbf{n}_{K^+},(v^+)^2-(v^-)^2\right\rangle_{e_h}\right| \\ &\le c\|v\|_D^2 + cH_D^2\Big(\sum_{K\in\mathcal{T}_D}\|\nabla v\|_K^2+\sum_{e_h\in\mathcal{E}_{h,D}^I}|e_h|^{-1}|[v]|_{e_h}^2\Big).\end{aligned} \tag{2.29}$$

The conclusion of the lemma now follows from using (2.25), (2.26) and (2.29) in (2.24). □

The next result concerns the approximation of a piecewise H^1 function by a constant function. In [23] a proof was given under a convexity assumption which is dispensed with in the present treatment.

Lemma 2.1.2. *Let D be as in Lemma 2.1.1 and let $v\in\mathcal{T}_D$. Then there exists $\overline{v}\in\mathcal{P}_0(D)$ such that*

$$\|v-\overline{v}\|_D \le c\,H_D|v|_{1,h,D}. \tag{2.30}$$

Proof. The construction of $\overline{v}$ is done in two steps. If need be, we extend $\mathcal{T}_D$ into a conforming partition $\mathcal{T}_D^C$ and view v as an element of $H^1(\mathcal{T}_D^C)$.

Now let $\chi \in \mathcal{P}_q(\mathcal{T}_D^C) \cap C(D)$ be the continuous piecewise polynomial approximation of v as stipulated in Theorem 2.1 (i) to yield

$$\|v-\chi\|_D^2 \;=\; \sum_{K\in\mathcal{T}_D^C} \|v-\chi\|_K^2 \;\le\; c \sum_{e_h\in\mathcal{E}_{h,D}^{I,C}} |e_h|\,|[v]|_{e_h}^2 \;=\; c \sum_{e_h\in\mathcal{E}_{h,D}^{I}} |e_h|\,|[v]|_{e_h}^2. \tag{2.31}$$

Here, $\mathcal{E}_{h,D}^{I,C}$ are the edges in $\mathcal{E}_{h,D}^{I}$ as well as any new interior edges introduced by the conforming extension of $\mathcal{T}_D$. Also, the last equality is due to the fact that the jumps of v vanish on these new edges. Now since $|e_h| \le H_D$, this readily implies the bound

$$\|v-\chi\|_D \;\le\; cH_D|v|_{1,h,D}. \tag{2.32}$$

Note that χ belongs to $H^1(D)$. Hence, using (3.14) with $|\alpha| = 1$, we obtain

$$\begin{aligned} \|\nabla\chi\|_D^2 \;&=\; \sum_{K\in\mathcal{T}_D^C} \|\nabla\chi\|_K^2 \;\le\; 2\sum_{K\in\mathcal{T}_D^C} \|\nabla v\|_K^2 + 2\sum_{K\in\mathcal{T}_D^C} \|\nabla(v-\chi)\|_K^2 \\ &\le\; 2\sum_{K\in\mathcal{T}_D^C} \|\nabla v\|_K^2 + c\sum_{e_h\in\mathcal{E}_{h,D}^{I,C}} |e_h|^{-1}|[v]|_{e_h}^2 \;\le\; c\,|v|_{1,h,D}^2. \end{aligned} \tag{2.33}$$

In the second step, we let $\overline{v}$ be the constant function on D with value $\frac{1}{|D|}\int_D \chi\,dx$. We shall next see that the Poincaré type inequality

$$\|\overline{v}-\chi\|_D \;\le\; cH_D\|\nabla\chi\|_D, \tag{2.34}$$

holds. Indeed, since D is shape regular, there exist positive numbers r, R which are both $O(H_D)$ such that $Q(\mathbf{x}_0, 4r) \subset D \subset Q(\mathbf{x}_0, R)$ where $Q(\mathbf{x}_0, \rho)$ is the cube centered at $\mathbf{x}_0$ with side length ρ. Then (2.34) follows from Theorem 12.36 on page 370 of [34]. Finally, the estimate (2.30) now follows from (2.32), (2.33), (2.34) and the triangle inequality. □

Combining the previous two lemmas, we obtain the following useful result:

$$|\overline{v} - u|^2_{\partial D} \leq cH_D|u|^2_{1,h,D}. \tag{2.35}$$

To establish the convergence result for the overlapping Schwarz methods, we will need the help of the following generalized Poincaré inequality, which is extension to functions in $H^1(\mathcal{T}_h)$ of Lemma 3.10 of [43]. A proof can be found in [23].

Lemma 2.1.3. *Let D be as in Lemma 2.1.1 and let $0 < \rho < H_D D$. Then for any $v \in H^1(\mathcal{T}_D)$ there holds the following generalized Poincaré inequality:*

$$\|v\|^2_{B_\rho} \leq c\rho\Big(H_D^{-1}\|v\|^2_D + H_D|v|^2_{1,h,D}\Big), \quad \forall v \in H^1(\mathcal{T}_D), \tag{2.36}$$

where

$$B_\rho = \{\mathbf{x} \in D : \operatorname{dist}(\mathbf{x}, \partial D) \leq \rho\}$$

denotes the boundary layer of D of width ρ.

2.3 Schwarz Framework

We shall follow the powerful framework established by O. Widlund and coworkers [43]. It is of sufficient generality to cover a wide range of methods including nonoverlapping or overlapping, additive or multiplicative or other approaches e.g. hybrid and to the extent that the efficacy of a particular method can be gauged by verifying a set of assumptions.

In addition to the partition $\mathcal{T}_h$, we shall need a partition $\mathcal{T}_S$ of Ω into open *subdomains* $\{\Omega_i\}_{i=1}^p$ and a *coarse* partition $\mathcal{T}_H$ of Ω. At this point, we require the following alignments between the three partitions

$$\mathcal{T}_S \subseteq \mathcal{T}_h \quad \text{and} \quad \mathcal{T}_H \subseteq \mathcal{T}_h, \tag{2.37}$$

that is, each subdomain is the union of cells belonging to $\mathcal{T}_h$ and each cell in $\mathcal{T}_H$ is the union of cells belonging to $\mathcal{T}_h$. The partitions $\mathcal{T}_S$ and $\mathcal{T}_H$ are, just like $\mathcal{T}_h$ used to construct certain subspaces of V^h and corresponding bilinear forms on these subspaces.

2.3.1 Subdomain Spaces and Bilinear Forms

We define the subspaces $\{V_j^h\}_{j=1}^p$ associated with the subdomains $\{\Omega_j\}_{j=1}^p$ by

$$V_j^h = \{v \in V^h | v = 0 \quad \text{in} \quad \Omega \setminus \overline{\Omega}_j\}, \quad j = 1, 2, \cdots, p,$$

i.e. V_i^h is the restriction of V^h to Ω_i. We clearly have

$$V^h = V_1^h + \cdots + V_p^h. \tag{2.38}$$

To each subspace V_i^h we associate the bilinear form $a_i(\cdot, \cdot)$ defined as the restriction of $a_h^{\gamma_h}(\cdot, \cdot)$ to V_i^h:

$$a_i(u, v) \;=\; a_h^{\gamma_h}(u, v), \qquad \forall u, v \in V_i^h, \; i = 1, \ldots, p. \tag{2.39}$$

In order to give a concrete expression for these forms, we introduce the following sets:

1. $\mathcal{S}_i := \{e_h \in \mathcal{E}_h^I, e_h \in \partial\Omega_i\}$, $i = 1, \ldots, p$.
2. $\mathcal{T}_{h,i} := \{K \in \mathcal{T}_h, K \subset \Omega_i\}$.
3. $\mathcal{E}_{h,i}^I := \{e_h \in \mathcal{E}_h^I, e_h \subset \Omega_i\}$.
4. $\mathcal{E}_{h,i}^D := \{e_h \in \mathcal{E}_h^D, e_h \in \partial\Omega_i\}$.
5. $\mathcal{E}_{h,i} := \mathcal{E}_{h,i}^I \cup \mathcal{E}_{h,i}^D$.

For Arnold's formulation, the subdomain bilinear forms are give by

$$\begin{aligned} a_i(u,v) &= \sum_{K\in\mathcal{T}_{h,i}} (\nabla u, \nabla v)_K \\ &\quad - \sum_{e_h\in\mathcal{E}_{h,i}} \Big(\langle\{\partial_n u\},[v]\rangle_{e_h} + \langle\{\partial_n v\},[u]\rangle_{e_h} - \frac{\gamma_h}{|e_h|}\langle [u],[v]\rangle_{e_h} \Big) \\ &\quad - \sum_{e_h\in\mathcal{S}_i} \Big(\frac{1}{2}\langle \partial_n u, v\rangle_{e_h} + \frac{1}{2}\langle \partial_n v, u\rangle_{e_h} - \frac{\gamma_h}{|e_h|}\langle u, v\rangle_{e_h} \Big), \end{aligned} \tag{2.40}$$

for $u, v \in V_i^h$, $i = 1, \ldots, p$. In the above expression, the edge integrals on $\mathcal{S}_i$ must be interpreted as follows: The traces of u, v are taken from Ω_i and the normal derivatives are with respect to the unit outward normal vectors to Ω_i.

For Baker's formulation, the bilinear form $a_i(\cdot,\cdot)$ takes the form

$$\begin{aligned} a_i(u,v) &= \sum_{K\in\mathcal{T}_{h,i}} (\nabla u, \nabla v)_K \\ &\quad - \sum_{e_h\in\mathcal{E}_{h,i}} \Big(\langle\partial_n u,[v]\rangle_{e_h} + \langle\partial_n v,[u]\rangle_{e_h} - \frac{\gamma_h}{|e_h|}\langle [u],[v]\rangle_{e_h} \Big) \\ &\quad - \sum_{e_h\in\mathcal{S}_i} \Big(\langle \partial_n u, v\rangle_{e_h} + \langle \partial_n v, u\rangle_{e_h} - \frac{\gamma_h}{|e_h|}\langle u, v\rangle_{e_h} \Big), \end{aligned} \tag{2.41}$$

for $u, v \in V_i^h$, $i = 1, \ldots, p$. Note that for $e_h \in \mathcal{S}_i$, the terms $\langle \partial_n u, v\rangle_{e_h}$, $\langle \partial_n v, u\rangle_{e_h}$ will be present if and only if K^+ belongs to Ω_i.

2.3.2 The Coarse Subspace and Bilinear Form

In addition to the subspaces V_i^h, $i = 1, \ldots, p$, a coarse mesh subspace V_0^h corresponding to the partition $\mathcal{T}_H$ and the corresponding bilinear form $a_0(\cdot,\cdot) : V_0^h \times V_0^h \to R$ are presently introduced. $\mathcal{T}_H$ consists of simplicial cells D each being a union of cells from $\mathcal{T}_h$. We require $\mathcal{T}_H$ to satisfy the conditions (P1)-(P5) imposed on $\mathcal{T}_h$ and define $\mathcal{E}_H^I$, $\mathcal{E}_H^D$, $\mathcal{E}_H^N$, and $\mathcal{E}_H$ to be the analogues of the corresponding sets in $\mathcal{T}_h$.

It is well known that this construction is crucial in obtaining a good preconditioner. We shall consider two different possibilities for the choice of a coarse space. The first consists in using discontinuous piecewise polynomials which will be in keeping with the spirit of DG. However there are resulting compatibility issues arising from the penalty terms which must be investigated. The second consists in using continuous piecewise polynomials which has the advantages of resolving these compatibility issues in a single stroke.

2.3.3 The Discontinuous Coarse Space

Here we set

$$V_0^h := V^{H,D} = \mathcal{P}_{q_0}(\mathcal{T}_H), \quad 1 \leq q_0 \leq q. \tag{2.42}$$

It is clear that V_0^h is a subspace of V^h. Also, we define the coarse space bilinear form by

$$\begin{aligned} a_0(u,v) = a_H^{\gamma_H}(u,v) \;\; = \;\; & \sum_{K\in\mathcal{T}_H} (\nabla u, \nabla v)_K \\ & - \sum_{e_H\in\mathcal{E}_H} \Big(\langle\{\partial_n u\}, [v]\rangle_{e_H} + \langle\{\partial_n v\}, [u]\rangle_{e_H} \Big) \\ & + \sum_{e_H\in\mathcal{E}_H} \frac{\gamma_H}{|e_H|} \langle [u], [v]\rangle_{e_H}, \quad \forall u, v \in V^{H,D}. \end{aligned} \tag{2.43}$$

Here, $\mathcal{E}_H$ is the analog of $\mathcal{E}_h$.

In [23] $a_H^{\gamma_H}$ was defined as the restriction of $a_h^{\gamma_h}$ to $V^{H,D}$, i.e.

$$a_H^{\gamma_H}(u,v) \;=\; a_h^{\gamma_h}(u,v), \quad \forall u, v \in V^{H,D}. \tag{2.44}$$

Here, we have taken a more general approach. On the other hand, we would like to identify conditions that imply (2.44). Indeed, (2.44) may have beneficial analytical or practical consequences or may be even necessary in some contexts.

To begin, it easy to see that

$$\sum_{K\in\mathcal{T}_h}(\nabla u,\nabla v)_K \;=\; \sum_{K\in\mathcal{T}_H}(\nabla u,\nabla v)_K,\quad \forall u,v\in V^{H,D}. \tag{2.45}$$

Furthermore, for $u,v\in V^{H,D}$, the jumps across edges $e_h\in\mathcal{E}_h^I$ which are in the interior of some $K\in\mathcal{T}_H$ are zero. Also, for Arnold's formulation the values of $\langle\{\partial_n u\},[v]\rangle_{e_H}$ and $\langle\{\partial_n v\},[u]\rangle_{e_H}$ are independent of the sign convention used in the designation of K^+ vs. K^-. For $e_h\in\mathcal{E}_h^D$, there is no issue since we always use K^+ for the cell that contains it. Hence, we can combine these edge integrals to obtain

$$\begin{aligned}&\sum_{e_h\in\mathcal{E}_h}\Big(\langle\{\partial_n u\},[v]\rangle_{e_h}+\langle\{\partial_n v\},[u]\rangle_{e_h}\Big)\\ &\qquad=\sum_{e_H\in\mathcal{E}_H}\Big(\langle\{\partial_n u\},[v]\rangle_{e_H}+\langle\{\partial_n v\},[u]\rangle_{e_H}\Big),\quad \forall u,v\in V^{H,D}.\end{aligned} \tag{2.46}$$

For Baker's formulation (2.46) may not hold as is easily exhibited by the example in Figure 2.1. For this reason we consider the following *sign compatibility* assumption:

> For all edges $e_h\in\mathcal{E}_h^I$ that are part of an edge $e_H\in\mathcal{E}_H^I,\ e_H\subset\partial D$, $D\in\mathcal{T}_H$, the cells $K\in\mathcal{T}_h$ that contain e_h and belong to D have the same sign as D in relation to e_H. See e.g. Figure 2.2. (SC)

With this assumption, the analog of (2.46) also holds for Baker's formulation.

We now focus attention on the penalty jump terms. Note that these terms are identical for both the Arnold and Baker formulations and are independent of the $K^{+,-}$ convention. Furthermore, for $u,v\in V^{H,D}$, the jumps across edges $e_h\in\mathcal{E}_h^I$ which are in the interior of some $K\in\mathcal{T}_H$ are zero. On the other hand, we must deal with the attached weights. Using (2.45) and (2.46) (we assume that (SC) holds for Baker's

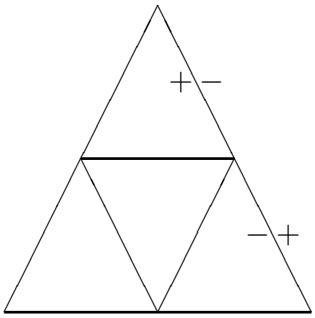


Figure 2.1: Violation of sign compatibility condition

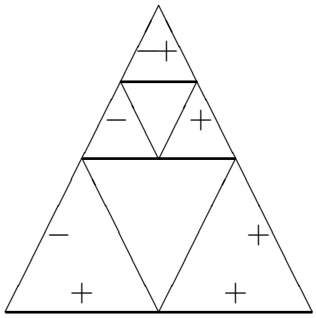


Figure 2.2: Compliance with sign compatibility condition

formulation) we obtain

$$a_h^{\gamma_h}(u,v) \;=\; a_0(u,v) + \sum_{e_H\in\mathcal{E}_H}\sum_{e_h\subset e_H}\left(\frac{\gamma_h}{|e_h|}-\frac{\gamma_H}{|e_H|}\right)\langle [u],[v]\rangle_{e_h}\,, \quad \forall u,v\in V^{H,D}. \tag{2.47}$$

This motivates the following *penalty compatibility* assumption:

$$\frac{\gamma_h}{|e_h|} \;=\; \frac{\gamma_H}{|e_H|}, \quad e_h\subset e_H\in\mathcal{E}_H. \tag{PC}$$

The following lemma encapsulates the above discussion.

Lemma 2.3.1.

(i) *Under assumption* (PC), (2.44) *holds for Arnold's formulation.*

(ii) *Under assumptions* (SC) *and* (PC), (2.44) *holds for Baker's formulation.*

(iii) *Under assumption* (PC) *restricted to* $e_H\in\mathcal{E}_H^D$ *we have* $F_h(v)=F_H(v), \quad \forall v\in V^{H,D}$.

Concerning the practicality of assumptions (SC) and (PC), we stress that they are easy to implement. Indeed, in most situations, one starts with the coarse mesh $\mathcal{T}_H$ and then obtains $\mathcal{T}_h$ via a process of refinement. In fact, if one starts with the

mesh $\mathcal{T}_h$ obtained say by a mesh generator, then it is not clear at all that one can identify a simplicial coarse mesh.

2.3.4 The Continuous Coarse Space

Here we set

$$V_0^h = V^{H,C} = \{v \in \mathcal{P}_{q_0}(\mathcal{T}_H) \cap C(\Omega),\ v\big|_{\Gamma_D} = 0\}. \tag{2.48}$$

Note that $V^{H,C}$ is a subspace of $V^{H,D}$. Hence, (2.47) holds for $u, v \in V^{H,C}$ also.

Since we are assuming that (SC) holds for Baker's formulation, it follows immediately from (2.47) that all the compatibilty issues considered above are resolved and that $a_0(u,v) = a_h^{\gamma_h}(u,v),\ \forall u, v \in V^{H,C}$. There are other advantages stemming from this choice. For one, the resulting stiffness matrix is smaller. Also, the analysis which will be carried out for the discontinuous coarse space will also cover this one, given that the crucial construct u_0 used in the proofs belongs to the continuous coarse space anyway.

2.4 Construction of the Schwarz Preconditioners

Define the projection operators $T_j : V^h \to V_j^h$ by

$$a_j(T_j u, v) = a_h^{\gamma_h}(u,v) \qquad \forall v \in V_j^h,\ j = 0, 1, \cdots, p. \tag{2.49}$$

That the operators T_j are well-defined follows from the easily provable facts that the bilinear forms are symmetric and coercive on their respective subspaces. For $j = 1, \ldots, p$, the coercivity of $a_j(\cdot,\cdot)$ is inherited from that of $a_h^{\gamma_h}(\cdot,\cdot)$ in view of their definition by restriction. For $a_0(\cdot,\cdot)$, coercivity will hold under a variety of scenarios, e.g. (i) taking $V_0^h = V^{H,C}$, or (ii) $V_0^h = V^{H,D}$ and enforcing (PC), or (iii) $V_0^h = V^{H,D}$ and $\gamma_H \geq \gamma_0$ in Lemma 2.1.1.

Following the framework given in [22, 42, 45], the general Schwarz approach consists in replacing the discrete problem (2.9) by the equation

$$\mathcal{P}(T_0, T_1, \ldots, T_p)u \;=\; \mathcal{P}(T_0, T_1, \ldots, T_p)u_h^{\gamma_h}, \tag{2.50}$$

where $\mathcal{P}$ is some polynomial in $T_0, T_1, \ldots, T_p$ with $\mathcal{P}(0, 0, \ldots, 0) = 0$. Note that the right side of (2.50) can be calculated without knowing $u_h^{\gamma_h}$.

Additive, multiplicative as well as other (e.g. hybrid) methods may be formulated as special cases of the polynomial $\mathcal{P}$ (see [42]). Whereas the analysis provided herein is sufficient to obtain condition number estimates for all these methods, we shall restrict ourselves to the additive nonoverlapping and overlapping cases.

For both the nonoverlapping and overlapping cases, the additive Schwarz method consists in the following choice of the polynomial $\mathcal{P}$

$$T \;=\; \mathcal{P}(T_0, T_1, \ldots, T_p) \;:=\; T_0 + T_1 + \cdots + T_p.$$

The operator T is the preconditioned form of the operator A induced by (2.9). In matrix notation, the additive Schwarz preconditioning corresponds to choosing the matrix B in (2.14) as

$$B = R_0^T A_0^{-1} R_0 + R_1^T A_1^{-1} R_1 + \cdots + R_p^T A_p^{-1} R_p, \tag{2.51}$$

where A_i is the stiffness matrix corresponding to $a_i(\cdot, \cdot)$ and R_i^T is the matrix representation of the embedding $V_i^h \to V^h$, $i = 0, 1, \ldots, p$.

We have the following simple but important result.

Lemma 2.4.1. *The operators T_i, $i = 0, 1, \ldots, p$, and hence T, are self-adjoint with respect to $a_h^{\gamma_h}(\cdot\cdot)$. Furthermore, T is invertible.*

Proof. The proof of the first assertion is in Lemma 2 of [42]. Now suppose $Tu = 0$ for some $u \in V^h$. We shall show that $u = 0$. Indeed,

$$\sum_{i=0}^{p} a_i(T_i u, T_i u) \;=\; \sum_{i=0}^{p} a_h^{\gamma_h}(u, T_i u) \;=\; a_h^{\gamma_h}(u, \big(\sum_{i=0}^{p} T_i\big)u) \;=\; a_h^{\gamma_h}(u, Tu) \;=\; 0.$$

Since the bilinear forms $a_i(\cdot,\cdot),\ i = 0,1,\ldots,p$ are coercive on their respective subspaces, it follows from the above that $T_i u = 0, i = 0,1,\ldots,p$. Now letting $u = u_0 + u_1 + \ldots, u_p,\ u_i \in V_i^h$, it follows that $0 = a_i(T_i u, u_i) = a_h^{\gamma_h}(u, u_i)$. Summing over i, we obtain $a_h^{\gamma_h}(u,u) = 0$, from which the desired result follows. □

Following the abstract analytical framework, we shall verify the following three assumptions and obtain bounds on the constants $C_0^2, \rho(\mathcal{E})$ and ω appearing in them.
Assumption 1: For any $u \in V^h$

$$\sum_{i=0}^{p} a_i(u_i, u_i) \ \leq\ C_0^2 a_h(u,u), \tag{2.52}$$

for *some* representation $u = \sum_{i=0}^{p} u_i$. Here, $1/C_0^2$ is a lower bound on the smallest eigenvalue of the preconditioned matrix.
Assumption 2: Let $0 \leq \mathcal{E}_{ij} \leq 1$ be the minimal values such that

$$|a_h^{\gamma_h}(u_i, u_j)| \ \leq\ \mathcal{E}_{ij}\, a_h^{\gamma_h}(u_i,u_i)^{\frac{1}{2}} a_h^{\gamma_h}(u_j,u_j)^{\frac{1}{2}}, \quad u_i \in V_i^h,\ u_j \in V_j^h,\ i,j = 1,\cdots,p. \tag{2.53}$$

That such values exist follows from the Cauchy-Schwarz inequality. Define $\rho(\mathcal{E})$ to be the spectral radius of $\mathcal{E}$.
Assumption 3: Let $\omega \geq 1$ be the smallest value such that

$$a_h^{\gamma_h}(u,u) \ \leq\ \omega\, a_i(u,u), \quad \forall u \in V_i^h,\ i = 0,1,\ldots,p. \tag{2.54}$$

The following result gives an estimate for the condition number of the preconditioner constructed using the additive Schwarz method. The theorem below is quoted from [42], Lemma 3. See also [43].

Theorem 2.4.1. *Suppose assumptions 1, 2, and 3 are satisfied. Then the abstract additive Schwarz method $T = T_0 + T_1 + \cdots + T_p$ satisfies*

$$\kappa(T) \ \leq\ \omega(1 + \rho(\mathcal{E}))C_0^2. \tag{2.55}$$

2.4.1 Nonoverlapping Additive Schwarz Preconditioner

In this case the subdomains Ω_i are disjoint and the sum (2.38) is direct. Also, in addition to the alignments shown in (2.37), we require $\mathcal{T}_S \subseteq \mathcal{T}_H$. It also turns out that the analysis will depend on the consideration of a certain *interface* bilinear form $\mathcal{I}(\cdot,\cdot) : V^h \times V^h \to \mathbb{R}$ that we now present.

For Arnold's formulation, it is given by

$$\begin{aligned}\mathcal{I}(u,v) \;=\; & \frac{1}{2}\sum_{e_h\in\mathcal{S}}\Big(\big\langle\partial_n u^+, v^-\big\rangle_{e_h} + \big\langle\partial_n v^+, u^-\big\rangle_{e_h} - \big\langle\partial_n u^-, v^+\big\rangle_{e_h} - \big\langle\partial_n v^-, u^+\big\rangle_{e_h}\Big) \\ & - \sum_{e_h\in\mathcal{S}}\frac{\gamma_h}{|e_h|}\Big(\big\langle u^+, v^-\big\rangle_{e_h} + \big\langle u^-, v^+\big\rangle_{e_h}\Big),\end{aligned} \tag{2.56}$$

where $\mathcal{S} := \cup_{i=1}^p \mathcal{S}_i$ is sometimes called the *skeleton* of the nonoverlapping partition.

For Baker's formulation, the interface form is given by

$$\mathcal{I}(u,v) = \sum_{e_h\in\mathcal{S}}\Big(\big\langle\partial_n u^+, v^-\big\rangle_{e_h} + \big\langle\partial_n v^+, u^-\big\rangle_{e_h} - \frac{\gamma_h}{|e_h|}\Big(\big\langle u^+, v^-\big\rangle_{e_h} + \big\langle u^-, v^+\big\rangle_{e_h}\Big)\Big). \tag{2.57}$$

It is easy to show that for both Arnold's and Baker's formulations, the following identity holds

$$a_h^{\gamma_h}(u,v) \;=\; \sum_{i=1}^p a_i(u_i,v_i) + \mathcal{I}(u,v), \quad \forall u,v\in V^h, \tag{2.58}$$

where

$$u \;=\; \sum_{i=1}^p u_i, \quad v \;=\; \sum_{i=1}^p v_i, \quad \text{and} \quad u_i, v_i \in V_i^h.$$

Lemma 2.4.2. *There exists a constant c independent of the number of subdomains, such that*

$$|\mathcal{I}(w,w)| \;\le\; c\overline{\gamma}\sum_{D\in\mathcal{T}_H}\underline{h}_D^{-1}\Big(H_D^{-1}\|w\|_D^2 + H_D|w|_{1,h,D}^2\Big), \quad \forall w\in V^h, \tag{2.59}$$

where

$$H_D = \mathrm{diam}(D), \quad \overline{\gamma} = \max_{e_h \in \mathcal{E}_h} \gamma_h, \quad \underline{h}_D = \min_{K \in \mathcal{T}_h, K \subset D} h_K.$$

Proof. We shall only treat Arnold's formulation, Baker's being similar. We have

$$\begin{aligned} \mathcal{I}(w,w) &= \sum_{e_h \in \mathcal{S}} \Big(\langle \partial_n w^+, w^- \rangle_{e_h} - \langle \partial_n w^-, w^+ \rangle_{e_h} \Big) - 2 \sum_{e_h \in \mathcal{S}} \frac{\gamma_h}{|e_h|} \langle w^+, w^- \rangle_{e_h} \qquad (2.60) \\ &:= A + B. \end{aligned}$$

Now from the arithmetic-geometric mean inequality we obtain

$$\begin{aligned} |A| &\leq \frac{1}{2} \sum_{e_h \in \mathcal{S}} |e_h| \Big(|\partial_n w^+|^2_{e_h} + |\partial_n w^-|^2_{e_h} \Big) + \frac{1}{2} \sum_{e_h \in \mathcal{S}} |e_h|^{-1} \Big(|w^+|^2_{e_h} + |w^-|^2_{e_h} \Big) \qquad (2.61) \\ &= A_1 + A_2. \end{aligned}$$

Using the trace and inverse inequalities (2.16), (2.17) it follows that

$$|A_1| \leq c \sum_{K \in \mathcal{T}_h} \|\nabla w\|^2_K. \qquad (2.62)$$

Now each $e_h \in \mathcal{S}$ belongs to the boundary ∂D of some cell in the coarse mesh $\mathcal{T}_H$. Hence, appealing to Lemma 2.1.1 we have

$$|A_2| \leq \sum_{D \in \mathcal{T}_H} \underline{h}_D^{-1} |w|^2_{\partial D} \leq c \sum_{D \in \mathcal{T}_H} \underline{h}_D^{-1} \Big(H_D^{-1} \|w\|^2_D + H_D |w|^2_{1,h,D} \Big). \qquad (2.63)$$

The estimation of B is identical to that of A_2. We have

$$|B| \leq c\overline{\gamma} \sum_{D \in \mathcal{T}_H} \underline{h}_D^{-1} \Big(H_D^{-1} \|w\|^2_D + H_D |w|^2_{1,h,D} \Big). \qquad (2.64)$$

The required result now follows from (2.61)-(2.64). □

Proposition 2.4.1. *For any $u \in V^h$, there exists a decomposition $u = \sum_{j=0}^{p} u_j,\ u_j \in V_j^h,\ j = 0, 1, \ldots, p$ such that*

$$\sum_{j=0}^{p} a_i(u_j, u_j) \ \leq \ c\overline{\gamma}\,[H:h]\,a_h^{\gamma_h}(u,u), \tag{2.65}$$

where the constant c is independent of p and

$$[H:h] \ = \ \max_{D \in \mathcal{T}_H} \frac{H_D}{\underline{h}_D}$$

is a measure of the fineness of $\mathcal{T}_h$ with respect to $\mathcal{T}_H$.

Proof. An important component of the proof consists in constructing an appropriate u_0 in the coarse space V_0^h. Let $\overline{v}$ be the piecewise constant approximation on $\mathcal{T}_H$ of u constructed in Lemma 2.1.2 and let u_0 be the continuous piecewise linear approximation of $\overline{v}$ constructed in Theorem 2.1. We also require u_0 to vanish on Γ_D. Thus u_0 belongs to the continuous coarse space $V^{H,C}$. According to the decomposition (2.38), which is direct for the nonoverlapping method, there exist uniquely determined functions $u_i \in V_i^h,\ i = 1, \ldots, p$ such that $u - u_0 = u_1 + \cdots + u_p$. We have

$$a_h^{\gamma_h}(u - u_0, u - u_0) = \sum_{j=1}^{p} a_j(u_j, u_j) + \mathcal{I}(u - u_0, u - u_0).$$

Adding $a_0(u_0, u_0)$ to both sides and using the Cauchy-Schwarz inequality on $a_h^{\gamma_h}$, we obtain

$$\begin{aligned} \sum_{j=0}^{p} a_j(u_j, u_j) \ &= \ a_h^{\gamma_h}(u - u_0, u - u_0) + a_0(u_0, u_0) - \mathcal{I}(u - u_0, u - u_0) \\ &\leq \ 2a_h^{\gamma_h}(u,u) + 2a_h^{\gamma_h}(u_0, u_0) + a_0(u_0, u_0) + |\mathcal{I}(u - u_0, u - u_0)|. \end{aligned} \tag{2.66}$$

As for $a_0(u_0, u_0)$, recall that u_0 is continuous and vanishes on Γ_D. Hence it follows from (2.47) that $a_0(u_0, u_0) = a_h^{\gamma_h}(u_0, u_0)$, leading to

$$\sum_{j=0}^{p} a_j(u_j, u_j) \;\leq\; 2a_h^{\gamma_h}(u, u) + 3a_h^{\gamma_h}(u_0, u_0) + |\mathcal{I}(u - u_0, u - u_0)|. \tag{2.67}$$

We next estimate $|\mathcal{I}(u - u_0, u - u_0)|$. From Lemma 2.4.2, it follows that

$$|\mathcal{I}(u - u_0, u - u_0)| \;\leq\; c\overline{\gamma} \sum_{D \in \mathcal{T}_H} \underline{h}_D^{-1} \Big(H_D^{-1} \|u - u_0\|_D^2 + H_D |u - u_0|_{1,h,D}^2 \Big). \tag{2.68}$$

We shall next estimate $\|u - u_0\|_D^2$. We have

$$\|u - u_0\|_D^2 \;\leq\; 2\|u - \overline{v}\|_D^2 + 2\|\overline{v} - u_0\|_D^2. \tag{2.69}$$

It follows from Lemma 2.1.2 that

$$\|u - \overline{v}\|_D \;\leq\; cH_D |u|_{1,h,D}. \tag{2.70}$$

Furthermore, it follows from the approximation result (2.20),

$$\|\overline{v} - u_0\|_D^2 \;\leq\; c \sum_{e_H \in \omega_e(D)} |e_H| \, |[\overline{v}]|_{e_H}^2 + c \sum_{e_H \in \mathcal{E}_H^D \cap \partial D} |e_H| \, |\overline{v}|_{e_H}^2, \tag{2.71}$$

where $\omega_e(D)$ is the (localized) set of edges e_H emanating from the vertices of D. We note that such edges are *interior*, i.e. belong to $\mathcal{E}_H^I$. We also note, as this is crucial to the analysis, that the second sum appears only if D has an edge on Γ_D and is due to the fact that u_0 vanishes on that part of $\partial\Omega$. Furthermore, since

$$\begin{aligned} |[\overline{v}]|_{e_H}^2 &\leq 3|\overline{v}^+ - u^+|_{e_H}^2 + 3|[u]|_{e_H}^2 + 3|\overline{v}^- - u^-|_{e_H}^2, \quad \text{for} \quad e_H \in \mathcal{E}_H^I, \\ |\overline{v}|_{e_H}^2 &\leq 2|\overline{v}^+ - u^+|_{e_H}^2 + 2|u|_{e_H}^2, \quad \text{for} \quad e_H \in \mathcal{E}_H^D, \end{aligned}$$

collecting terms in the sums in (2.71), we get

$$\begin{aligned}\|\overline{v}-u_0\|_D^2 &\leq c\sum_{D\in\omega(D)}|e_H|\,|\overline{v}-u|_{\partial D}^2+c\sum_{e_H\in\omega_e(D)}|e_H|\,|[u]|_{e_H}^2+c\sum_{e_H\in\mathcal{E}_H^D\cap\partial D}|e_H|\,|u|_{e_H}^2\\ &\leq c\sum_{D\in\omega(D)}|e_H|\,|\overline{v}-u|_{\partial D}^2+c\sum_{e_H\in\omega_e(D)}|e_H||\overline{e_h}|\sum_{e_h\subset e_H}|e_h|^{-1}|[u]|_{e_h}^2\\ &\quad +c\sum_{e_H\in\mathcal{E}_H^D\cap\partial D}|e_H||\overline{e_h}|\sum_{e_h\subset e_H}|e_h|^{-1}|u|_{e_h}^2,\end{aligned}\tag{2.72}$$

where $\omega(D)$ is the local patch of cells in $\mathcal{T}_H$ that share a vertex with D (contains the edges in $\omega_e(D)$) and

$$|\overline{e_h}|=\max_{e_h\subset e_H}|e_h|.$$

Given that $|\overline{e_h}|\leq|e_H|\leq cH_D$ and $\mathcal{T}_H$ is locally quasi uniform, we have from (2.35),

$$\|\overline{v}-u_0\|_D^2\leq cH_D^2|u|_{1,h,\omega(D)}^2+cH_D^2\sum_{e_h\in\partial D\cap\Gamma_D}|e_h|^{-1}|u|_{e_h}^2.\tag{2.73}$$

where $|u|_{1,h,\omega(D)}$ is the seminorm defined by (2.22) over the local patch $\omega(D)$. Combining this with (2.70), it follows from (2.69) that

$$\|u-u_0\|_D^2\leq cH_D^2|u|_{1,h,\omega(D)}^2+cH_D^2\sum_{e_h\in\partial D\cap\Gamma_D}|e_h|^{-1}|u|_{e_h}^2.\tag{2.74}$$

We next estimate the term $|u-u_0|_{1,h,D}^2$ in (2.68). Since u_0 is continuous, from the triangle inequality we have

$$\begin{aligned}|u-u_0|_{1,h,D}^2 &= \sum_{K\subset D}\|\nabla(u-u_0)\|_K^2+\sum_{e_h\in\mathcal{E}_{h,D}^I}|[u]|_{e_h}^2\\ &\leq 2\sum_{K\subset D}\Big(\|\nabla u\|_K^2+\|\nabla u_0\|_K^2\Big)+\sum_{e_h\in\mathcal{E}_{h,D}^I}|[u]|_{e_h}^2\\ &\leq 2|u|_{1,h,D}^2+2\|\nabla u_0\|_D^2.\end{aligned}\tag{2.75}$$

Now since $\overline{v}$ is constant on D, using the inverse inequality (2.17) and (2.73) we obtain

$$\begin{aligned} \|\nabla u_0\|_D^2 &= \|\nabla(u_0 - \overline{v})\|_D^2 \\ &\leq c|u|_{1,h,\omega(D)}^2 + c \sum_{e_h \in \partial D \cap \Gamma_D} |e_h|^{-1} |u|_{e_h}^2. \end{aligned} \tag{2.76}$$

Using this in (2.75), we obtain

$$|u - u_0|_{1,h,D}^2 \leq c|u|_{1,h,\omega(D)}^2 + c \sum_{e_h \in \partial D \cap \Gamma_D} |e_h|^{-1} |u|_{e_h}^2. \tag{2.77}$$

Using (2.74) and (2.77) in (2.68), there follows

$$\begin{aligned} |\mathcal{I}(u - u_0, u - u_0)| &\leq c\overline{\gamma} \sum_{D \in \mathcal{T}_H} \underline{h}_D^{-1} H_D \left(|u|_{1,h,\omega(D)}^2 + \sum_{e_h \in \partial D \cap \Gamma_D} |e_h|^{-1} |u|_{e_h}^2 \right) \\ &\leq c\,\overline{\gamma}\,[H : h]\, a_h^{\gamma_h}(u, u), \end{aligned} \tag{2.78}$$

in view of the locality of the patches $\omega(D)$.

We next estimate $a_h^{\gamma_h}(u_0, u_0)$. Since u_0 is continuous and vanishes on Γ_D, we obtain

$$a_h^{\gamma_h}(u_0, u_0) = \sum_{K \in \mathcal{T}_h} \|\nabla u_0\|_K^2 = \sum_{D \in \mathcal{T}_H} \|\nabla u_0\|_D^2. \tag{2.79}$$

We already encountered the term $\|\nabla u_0\|_D^2$ in (2.76) and have a bound for it in (2.77). Hence, summing over D in $\mathcal{T}_H$, we have

$$a_h^{\gamma_h}(u_0, u_0) \leq c\, a_h^{\gamma_h}(u, u). \tag{2.80}$$

The proof now follows from (2.78), (2.80), and (2.66). □

Having verified **Assumption 1** with $C_0^2 = c\,\overline{\gamma}\,[H : h]$, we undertake the verification of **Assumption 2**. It is easily seen that

$$a_h^{\gamma_h}(u_i, u_j) = 0 \quad \text{if} \quad \overline{\mathcal{S}}_i \cap \overline{\mathcal{S}}_j = \emptyset,\ i, j = 1, \ldots, p. \tag{2.81}$$

For such pairs (i,j), (2.53) holds with $\mathcal{E}_{ij} = 0$. For the remaining cases (2.53) holds with $\mathcal{E}_{ij} = 1$ in view of the Cauchy-Schwarz inequality. Hence, it follows at once that

$$\rho(\mathcal{E}) \;\leq\; \|\mathcal{E}\|_\infty \;\leq\; N_c + 1, \tag{2.82}$$

where

$$N_c = \max_i \{\text{number of } \Omega_j \text{ such that } \mathcal{S}_i \cap \mathcal{S}_j \neq \emptyset\}.$$

In practice N_c is small, e.g. 4 in the case of a conforming partition $\mathcal{T}_S$ into squares.

Concerning **Assumption 3**, we have

Lemma 2.4.3. *Let $\omega \geq 1$ be given satisfying*

$$\omega \;=\; \begin{cases} 1 & \text{if } V_0^h = V^{H,C}, \text{ or } V_0^h = V^{H,D} \text{ and assumption (PC) holds} \\ 1 + \frac{\mu}{c_a} & \text{otherwise,} \end{cases} \tag{2.83}$$

where c_a is the coercivity constant in (2.12) and μ is defined as

$$\mu \;=\; \max \left| \frac{\gamma_h}{|e_h|} \frac{|e_H|}{\gamma_H} - 1 \right|.$$

Then,

$$a_h^{\gamma_h}(u,u) \;\leq\; \omega\, a_i(u,u), \quad \forall u \in V_i^h, \;\; i = 0, \ldots, p. \tag{2.84}$$

Proof. For $i = 1, \ldots, p$, (2.84) holds with $\omega = 1$ in view of the definition of the subdomain bilinear forms by restriction. For $i = 0$, the situation is different. From (2.47) we have

$$a_h^{\gamma_h}(u,u) \;=\; a_0(u,u) + \sum_{e_H \in \mathcal{E}_H^I \cup \mathcal{E}_H^D} \sum_{e_h \subset e_H} \left(\frac{\gamma_h}{|e_h|} - \frac{\gamma_H}{|e_H|} \right) |[u]|_{e_h}^2. \tag{2.85}$$

Of course if $V_0^h = V^{H,C}$ or $V_0^h = V^{H,D}$ and assumption (PC) holds then (2.84) holds with $\omega = 1$ for $i = 0$ as well. In the general case

$$\begin{aligned}
\sum_{e_H \in \mathcal{E}_H^I \cup \mathcal{E}_H^D} \sum_{e_h \subset e_H} \left(\frac{\gamma_h}{|e_h|} - \frac{\gamma_H}{|e_H|} \right) |[u]|_{e_h}^2 &= \sum_{e_H \in \mathcal{E}_H^I \cup \mathcal{E}_H^D} \sum_{e_h \subset e_H} \left(\frac{\gamma_h}{|e_h|} \frac{|e_H|}{\gamma_H} - 1 \right) \frac{\gamma_H}{|e_H|} |[u]|_{e_h}^2 \\
&\leq \mu \sum_{e_H \in \mathcal{E}_H^I \cup \mathcal{E}_H^D} \frac{\gamma_H}{|e_H|} |[u]|_{e_H}^2 \\
&\leq \mu \|u\|_{2,H}^2 \\
&\leq \mu a_0(u,u),
\end{aligned}$$

in view of (2.12). Using this in (2.85) gives

$$a_h^{\gamma_h}(u,u) \leq \left(1 + \frac{\mu}{c_a}\right) a_0(u,u),$$

thus completing the proof. □

With the assumptions verified and the form of the constant in each assumption determined, we may now find the bound for the condition number $\kappa(T)$ of the of the operator T.

Theorem 2.2. *The condition number $\kappa(T)$ of the operator T of the nonoverlapping additive Schwarz method defined in this section, or equivalenty that of the matrix BA, satisfies*

$$\kappa(T) \leq c\overline{\gamma}\,\omega \left(N_c + 1\right)[H:h], \tag{2.86}$$

where c is independent of p and the constants

$$\overline{\gamma}, \quad \omega, \quad N_c, \quad \text{and} \quad [H:h]$$

have been defined.

Proof. This is an immediate consequence of the abstract estimate (2.55) and our estimates (2.65), (2.82) and (2.83). □

Remark 2.4.1. *In the spirit of full disclosure, if the penalty compatibility assumption (PC) is not enforced, then* (2.86) *must be replaced by*

$$\kappa(T) \leq c\overline{\gamma}\,\big(N_c+1\big)[H:h]^2$$

in view of the presence in ω *of the factor* $\frac{\gamma_h}{|e_h|}\frac{|e_H|}{\gamma_H}$ *which could be as large as* $[H:h]$. *This gives further incentive for the enforcement of compatibility between the fine and coarse mesh penalty parameters.*

2.4.2 Overlapping Additive Schwarz Preconditioner

Let $\mathcal{T}_S$ now denote an overlapping partition of Ω with overlaps characterized by a corresponding collection of parameters $\delta_i > 0$. The properties possessed by this partition will be made precise next. In particular, the association of the overlap parameter δ_i to a subdomain is designed to handle the possibility of having subdomains of varying size, as may occur in the presence of local refinements. We begin by introducing some preliminary assumptions and notation.

1. $\mathcal{T}_H \subset \mathcal{T}_h$ and $\mathcal{T}_S \subset \mathcal{T}_h$.

2. We assume that there exist nonnegative functions $\{\theta_i\}_{i=1}^p$, $\theta_i \in W^{1,\infty}(\Omega)$ such that

$$\sum_{i=1}^{p}\theta_i = 1 \quad \text{on} \quad \overline{\Omega}, \quad \theta_i = 0 \quad \text{on} \quad \Omega\setminus\Omega_i, \quad \|\nabla\theta_i\|_{L^\infty} \leq c\delta_i^{-1}.$$

   Here, δ_i represents the width of a boundary layer of Ω_i resulting from the overlap with other subdomains. It is assumed that the practical range of its values is $h \leq \delta_i \leq H$.

3. Let $N(x)$ denote the number of subdomains which contain x and $N_c \equiv \max_{x\in\Omega} N(x)$. It is reasonable to assume that N_c is a small number.

4. Define the sets Ω_i^I and $\Omega_i^{\delta_i}$ by

$$\Omega_i^I = \{x \in \Omega_i;\ x \notin \Omega_k \quad \text{for all} \quad k \neq i\}, \qquad \Omega_i^\delta = \Omega_i \setminus \Omega_i^I.$$

Remark 2.4.2. *1. One practical way, which we adopt here, of generating such partitions is (see e.g. [43]) to start with a nonoverlapping partition $\{\tilde{\Omega}_i\}_{i=1}^p$ which is aligned with $\mathcal{T}_H$ and then add layers of cells from $\mathcal{T}_h$.*

2. We may think of $\|\nabla\theta_i\|_{L^\infty} \leq c\delta_i^{-1}$ as characterizing the (local) overlaps by allowing θ_i to decrease in a controlled fashion.

3. It follows from (4) above that $\theta_i(x) = 1$ on Ω_i^I. It is also possible that $\Omega_i^I = \emptyset$, which may happen if δ_i is large or if the subdomains are small.

4. In view of the alignment $\mathcal{T}_S \subset \mathcal{T}_h$ it follows that if a cell $K \in \mathcal{T}_h$ belongs to Ω_i, then either $K \subset \Omega_i^I$ or $K \subset \Omega_i^{\delta_i}$.

The overall strategy in the analysis is similar to that of the nonoverlapping case. To verify **Assumption 1**, we shall use the same u_0 as in the nonoverlapping case but $u_1, \ldots, u_p$ are defined differently in the decomposition $u = u_0 + u_1 + \cdots + u_p$. In order to do so we shall employ the usual Lagrange interpolation operator $\Pi_h : C(\mathcal{T}_h) \to V^h$ which is defined locally on each cell of $\mathcal{T}_h$.

Lemma 2.2.1. *For $u_0 \in V_0^h$, let $u_i \in V_i^h$, $i = 1, \ldots, p$ be given by $u_i = \Pi_h\theta_i(u - u_0)$. Then, there exists a constant c is independent of p such that for $i = 1, \ldots, p$,*

$$a_i(u_i, u_i) \leq c\Big(\sum_{K\in\mathcal{T}_{h,i}} \|\nabla(u-u_0)\|_K^2 + \sum_{e_h\in\mathcal{E}_{h,i}^I\cup\mathcal{E}_{h,i}^D} \frac{\gamma_h}{|e_h|}|[u]|_{e_h}^2 + \delta_i^{-2}\|u-u_0\|_{\Omega_i^{\delta_i}}^2 \Big). \quad (2.87)$$

Proof. Since $\theta_i = 0$ on $\mathcal{S}_i$, u_i vanishes also on $\mathcal{S}_i$. Hence from (2.40) we obtain

$$a_i(u_i, u_i) = \sum_{K\in\mathcal{T}_{h,i}} \|\nabla u_i\|_K^2 - \sum_{e_h\in\mathcal{E}_{h,i}^I\cup\mathcal{E}_{h,i}^D} 2\langle\{\partial_n u_i\}, [u_i]\rangle_{e_h} + \sum_{e_h\in\mathcal{E}_{h,i}^I\cup\mathcal{E}_{h,i}^D} \frac{\gamma_h}{|e_h|}|[u_i]|_{e_h}^2. \quad (2.88)$$

Now using the Cauchy-Schwarz and the arithmetic-geometric mean inequalities on the second sum, followed by the trace and inverse inequalities, we obtain

$$\begin{aligned} a_i(u_i,u_i) &= \sum_{K\in\mathcal{T}_{h,i}} \|\nabla u_i\|_K^2 + \sum_{e_h\in\mathcal{E}_{h,i}^I\cup\mathcal{E}_{h,i}^D} |e_h|\,|\{\partial_n u_i\}|_{e_h}^2 + \sum_{e_h\in\mathcal{E}_{h,i}^I\cup\mathcal{E}_{h,i}^D} \frac{\gamma_h+1}{|e_h|}|[u_i]|_{e_h}^2 \\ &\le c\sum_{K\in\mathcal{T}_{h,i}} \|\nabla u_i\|_K^2 + c\sum_{e_h\in\mathcal{E}_{h,i}^I\cup\mathcal{E}_{h,i}^D} \frac{\gamma_h}{|e_h|}|[u_i]|_{e_h}^2. \end{aligned} \tag{2.89}$$

We next estimate the terms in (2.89). For this, let $\overline{\theta}_{i,K}$ be the average of θ_i over an element $K\in\mathcal{T}_{h,i}$. It can be shown that

$$\|\theta_i-\overline{\theta}_{i,K}\|_{L^\infty(K)} \le \begin{cases} 0 & \text{if} \quad K\subset\Omega_i^I, \\ ch_K{\delta_i}^{-1} & \text{if} \quad K\subset\Omega_i^{\delta_i}. \end{cases} \tag{2.90}$$

Indeed, the first part follows since $\theta_i\equiv 1$ on Ω_i^I. The second part follows from the approximation property (2.18) (which also holds in the L^∞-norm cf. e.g. [14]) and the fact that $\|\nabla\theta_i\|_{L^\infty}\le c\delta_i^{-1}$.

Let Π_K denote the restriction of the (global) interpolation operator Π_h to the cell K. By the inverse inequality (2.17) and the fact that $0\le\overline{\theta}_{i,K}\le 1$ we get with $w=u-u_0$

$$\begin{aligned} \|\nabla u_i\|_K^2 &= \|\nabla\Pi_K(\theta_i w)\|_K^2 \\ &\le 2\|\nabla\Pi_K(\overline{\theta}_{i,K}w)\|_K^2 + 2\|\nabla\Pi_K\big((\theta_i-\overline{\theta}_{i,K})w\big)\|_K^2 \\ &\le 2\|\nabla w\|_K^2 + ch_K^{-2}\|\Pi_K\big((\theta_i-\overline{\theta}_{i,K})w\big)\|_K^2, \end{aligned} \tag{2.91}$$

using the fact that $\Pi_K(\overline{\theta}_{i,K}w)=\overline{\theta}_{i,K}w$. Now Π_K is stable with respect to the L^∞ norm. Also, using the fact that w is a polynomial,

$$\begin{aligned} \|\Pi_K\big((\theta_i-\overline{\theta}_{i,K})w\big)\|_K^2 &\le ch_K^d\|\theta_i-\overline{\theta}_{i,K}\|_{L^\infty(K)}^2\|w\|_{L^\infty(K)}^2 \\ &\le c\|\theta_i-\overline{\theta}_{i,K}\|_{L^\infty(K)}^2\|w\|_K^2. \end{aligned} \tag{2.92}$$

Thus, using (2.90) and (2.92) in (2.91), it follows that

$$\|\nabla u_i\|_K^2 \leq 2\|\nabla(u-u_0)\|_K + \begin{cases} 0, & \text{if} \quad K \in \Omega_i^I, \\ c\delta_i^{-2}\|u-u_0\|_K^2, & \text{if} \quad K \in \Omega_i^{\delta_i}. \end{cases} \tag{2.93}$$

Let $e_h = \partial K^+ \cap \partial K^- \in \mathcal{E}_{h,i}^I$. Since the mesh $\mathcal{T}_h$ is conforming, e_h is a full edge of K^+ and K^-. Hence, given that θ_i is continuous, with Π_{e_h} denoting the Lagrange interpolation operator on e_h, we have

$$|[u_i]|_{e_h}^2 = |\Pi_{K^+}(\theta_i w^+) - \Pi_{K^-}(\theta_i w^-)|_{e_h}^2 = |\Pi_{e_h}\theta_i[w]|_{e_h}^2.$$

Therefore, using the stability of Π_{e_h} in the L^∞-norm, the inverse estimate (2.17) on e_h, and the fact that u_0 is continuous, we obtain

$$|[u_i]|_{e_h}^2 \leq c|[w]|_{e_h}^2 = c|[u]|_{e_h}^2, \quad e_h \in \mathcal{E}_{h,i}^I. \tag{2.94}$$

For $e_h \in \mathcal{E}_{h,i}^D$, we have $|u_i|_{e_h}^2 = |\Pi_{e_h}\theta_i w|_{e_h}^2 = |\Pi_{e_h}\theta_i u|_{e_h}^2$ since u_0 vanishes on Γ_D. Hence the argument used above yields

$$|u_i|_{e_h}^2 \leq c|u|_{e_h}^2, \quad e_h \in \mathcal{E}_{h,i}^D. \tag{2.95}$$

Finally, using (2.93), (2.94) and (2.95) concludes the proof. □

Proposition 2.4.2. *For any $u \in V^h$, let $u = u_0 + u_1 + \cdots + u_p$ where u_0 is as in Proposition 2.4.1 and let $u_i = \Pi_h\theta_i(u-u_0)$, $i = 1, \ldots, p$. Then, there exists a constant c which is independent of p such that*

$$\sum_{i=0}^{p} a_i(u_i, u_i) \leq cN_c\,[H:\delta]\,a_h^{\gamma_h}(u,u), \quad \textit{where} \quad [H:\delta] := \max_{1\leq i\leq p}\max_{D\subset\Omega_i}\frac{H_D}{\delta_i}. \tag{2.96}$$

Proof. The proof consists in estimating the terms in (2.87). For some of these we shall use bounds already obtained in the proof of Proposition 2.4.1. To begin, note

that since $\sum_{i=1}^{p}\theta_i = 1$ on Ω and Π_h is the identity operator when restricted to V_h, we have $u = \sum_{i=0}^{p} u_i$. Since at any point in Ω the number of overlapping subdomains is bounded by N_c and $\mathcal{T}_H \subseteq \mathcal{T}_h$ we have

$$\begin{aligned}
\sum_{i=1}^{p}\sum_{K\in\mathcal{T}_{h,i}} \|\nabla(u-u_0)\|_K^2 &\leq N_c \sum_{D\in\mathcal{T}_H}\sum_{K\subseteq D} \|\nabla(u-u_0)\|_K^2 \\
&\leq N_c \sum_{D\in\mathcal{T}_H} |u-u_0|_{1,h,D}^2 \\
&\leq cN_c \sum_{D\in\mathcal{T}_H} \Big(|u|_{1,h,\omega(D)}^2 + \sum_{e_h\subset\partial D\cap\Gamma_D} |e_h|^{-1}|u|_{e_h}^2 \Big) \\
&\leq cN_c\, a_h^{\gamma_h}(u,u),
\end{aligned} \tag{2.97}$$

where we have also used the bound (2.77) on $|u-u_0|_{1,h,D}^2$. As for the second sum in (2.87), it follows immediately that

$$\sum_{i=1}^{p}\sum_{e_h\in\mathcal{E}_{h,i}^I\cup\mathcal{E}_{h,i}^D} \frac{\gamma_h}{|e_h|}|[u]|_{e_h}^2 \leq a_h^{\gamma_h}(u,u). \tag{2.98}$$

We now estimate the last terms in (2.87). For this, let $\mathcal{T}_{H,i}$ be the collection of cells from $\mathcal{T}_H$ such that $\Omega_i \subset \cup_{D\in\mathcal{T}_{H,i}} D$. Using Lemma 2.1.3 with $\rho = \delta_i$, it follows from the already established bounds (2.74) and (2.77) that

$$\begin{aligned}
\|u-u_0\|_{\Omega_i^{\delta_i}}^2 &= \sum_{D\in\mathcal{T}_{H,i}} \|u-u_0\|_{D\cap\Omega_i^{\delta_i}}^2 \\
&\leq c\delta_i \sum_{D\in\mathcal{T}_{H,i}} \Big(H_D^{-1}\|u-u_0\|_D^2 + H_D|u-u_0|_{1,h,D}^2 \Big) \\
&\leq c\delta_i \sum_{D\in\mathcal{T}_{H,i}} H_D \Big(|u|_{1,h,\omega(D)}^2 + \sum_{e_h\subset\partial D\cap\Gamma_D} |e_h|^{-1}|u|_{e_h}^2 \Big).
\end{aligned} \tag{2.99}$$

A comment on the application of Lemma 2.1.3 is in order here. Recall that the partition $\mathcal{T}_S$ was obtained from a nonoverlapping partition $\mathcal{T}_{\tilde{S}} = \{\tilde{\Omega}_i,\, i = 1,\ldots,p\}$ aligned with $\mathcal{T}_H$ by adding layers of cells from $\mathcal{T}_h$. So it is indeed the case $D\cap\Omega_i^{\delta_i}$ is a boundary layer of D. Hence, the use of Lemma 2.1.3 is justified.

Multiplying with δ_i^{-2} and summing, it follows

$$\sum_{i=1}^{p} \delta_i^{-2} \|u - u_0\|^2_{\Omega_i^{\delta_i}} \;\leq\; cN_c\,[H:\delta]\,a_h^{\gamma_h}(u,u). \tag{2.100}$$

Finally, since $a_0(u_0,u_0) = a_h^{\gamma_h}(u_0,u_0) \leq c a_h^{\gamma_h}(u,u)$ according to (2.80), the required result follows. □

Theorem 2.3. *The condition number $\kappa(T)$ of the operator T of the overlapping additive Schwarz method defined in this section, or equivalenty that of the matrix BA, satisfies*

$$\kappa(T) \;\leq\; c\,\omega\,N_c^2\,[H:\delta], \tag{2.101}$$

where c is independent of p and where ω and $[H:\delta]$ are as in (2.83) and (2.96) respectively.

Proof. Proposition 2.4.2 asserts that **Assumption 1** holds with $C_0^2 = cN_c\,[H:\delta]$. The verification of **Assumption 3** is exactly as in the nonoverlapping case. As for **Assumption 2**, the argument used before can still be applied provided we replace the condition $\overline{\mathcal{S}}_i \cap \overline{\mathcal{S}}_j = \emptyset$ in (2.81) by $\overline{\Omega}_i \cap \overline{\Omega}_j = \emptyset$ to yield $\rho(\mathcal{E}) \leq N_c + 1$. The result now follows from the estimate (2.55) given in Theorem 2.4.1. □

Remark 2.4.3. *If the penalty compatibility condition (PC) is enforced, then $\omega = 1$ in the above estimate and hence $\kappa(T)$ becomes independent of the penalty parameters.*

2.5 Numerical Experiments

The results of the experiments presented in this section are designed to either provide computational verification of the theory presented herein or to investigate various performance aspects of the preconditioners. In particular, the following topics are examined:

- Comparison of the performance of the nonoverlapping and overlapping preconditioners.
- Dependence of the preconditioner on the penalty parameter γ.
- Comparison of the performance of both preconditioners with the penalty compatibility condition (PC) enforced versus not enforced.

All experiments are performed using the preconditioned Conjugate Gradient (PCG) method to solve the system generated by the discontinuous Galerkin discretization. The preconditioners used correspond to the matrix B given in (2.51), which has a different form for the nonoverlapping and overlapping methods. Three test problems are considered in order to examine the method in a variety of situations.

The first is a problem with a smooth solution

$$\begin{cases} -\Delta u = 2\pi^2 \sin(\pi x)\sin(\pi y), & \text{in } \Omega = [0,1]\times[0,1] \\ u = 0, & \text{on } \partial\Omega \end{cases} \tag{P1}$$

with exact solution $u = \sin(\pi x)\sin(\pi y)$.

The second test problem

$$\begin{cases} -\Delta u = 2048\pi^2 \sin(32\pi x)\sin(32\pi y), & \text{in } \Omega = [0,1]\times[0,1] \\ u = 0, & \text{on } \partial\Omega \end{cases} \tag{P2}$$

has the smooth but oscillatory solution $u = \sin(32\pi x)\sin(32\pi y)$.

The final test problem

$$\begin{cases} -\Delta u = 0, & \text{in } \Omega \\ u = g_D, & \text{on } \partial\Omega \end{cases} \tag{P3}$$

is a problem whose solution has a singularity; here, Ω is chosen as an L-shaped domain with vertices $(0,0), (0,\frac{1}{2}), (-\frac{1}{2},\frac{1}{2}), (-\frac{1}{2},-\frac{1}{2}), (\frac{1}{2},-\frac{1}{2})$, and $(\frac{1}{2},0)$ with the reentrant

corner at the origin. The boundary data g_D is defined so that the exact solution is $u = r^{2/3} \sin \frac{2\theta}{3}$ in polar coordinates.

The triangulations $\mathcal{T}_h$ and $\mathcal{T}_H$ are taken to be uniform, following the pattern shown in Figure 2.3. This choice is purely for simplicity; the methods work just as well for completely unstructured meshes. The subdomains in the nonoverlapping case are taken to be squares comprised of fine level cells. For the overlapping preconditioner, the construction of the subdomains follows the approach discussed previously in Remark 2.4.2. In this case, we begin with a nonoverlapping subdomain partition and extend the subdomains by appending a layer of fine level cells belonging to $\mathcal{T}_h$, as exhibited in Figure 4.14. This implementation corresponds to minimal overlap with $\delta \approx h$. To create subdomains with generous overlap, continue the process of appending layers of fine level cells in the same manner until the desired overlap is obtained. Unless otherwise indicated, all experiments subsequently reported will use 16 subdomains, with the subdomain partition corresponding to a grid or "checkerboard", as seen in Figure 4.13.

In addition, the discontinuous coarse space will be utilized in all experiments in order to study the effect of the penalty compatibility condition (PC).

The estimation of the condition number $\kappa(BA)$ will follow the approach of [23]. The estimation is derived from the error reduction property of the CG method:

$$(Ae_k, e_k) \leq 4 \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right)^{2k} (Ae_0, e_0), \qquad k = 0, \ldots. \tag{2.102}$$

Here $e_k = \mathbf{x} - \mathbf{x}_k$, where $\mathbf{x}$ is the exact solution (computed to machine precision beforehand) and $\mathbf{x}_k$ is the k-th iterate. Using (2.102) as a model, the average error reduction per iteration μ is

$$\mu = \left\{ \frac{(Ae_k, e_k)}{(Ae_0, e_0)} \right\}^{1/2k}. \tag{2.103}$$

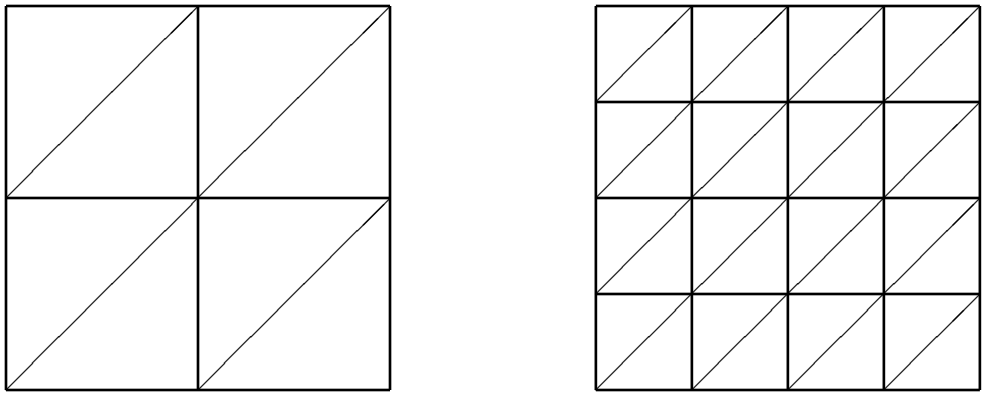


Figure 2.3: Example: Coarse Partition $\mathcal{T}_H$ and Fine Partition $\mathcal{T}_h$

From this, the estimate of $\kappa(A)$ or of $\kappa(BA)$ is calculated by

$$\kappa = \left(\frac{1+\mu}{1-\mu}\right)^2. \tag{2.104}$$

Note that these estimates are in actuality underestimates of $\kappa(BA)$; since the convergence of the method is actually dependent upon these numbers, they are reported instead of $\kappa(BA)$. The PCG iterations are stopped once the condition $\|e_k\| \leq 10^{-9}$ is met. This choice of stopping criterion takes advantage of the fact that the exact solution $\mathbf{x}$ is available (since it is needed to compute the condition number estimates anyway) to provide a equitable means of comparing solution methods.

The performance of the following three solvers will be compared throughout the experiments:

1. Preconditioned conjugate gradient (PCG) with the nonoverlapping additive Schwarz preconditioner,

2. PCG with the overlapping additive Schwarz preconditioner,

3. Traditional conjugate gradient (CG) with no preconditioning.

Comparisons will be made based on three metrics:

1. Condition number κ estimates of the corresponding stiffness matrix,

2. Iteration counts,

3. CPU runtime.

Another aspect of the theory we will seek to verify throughout all experiments is that the nonoverlapping preconditioned system has a condition number of order $O(\frac{H}{h})$, as predicted by Theorem 2.2. Similarly, we will observe if the overlapping preconditioned system has a condition number of order $O(\frac{H}{\delta})$, as predicted by Theorem 2.3.

Experiments are performed on a variety of combinations of coarse grids $\mathcal{T}_H$ and fine grids $\mathcal{T}_h$. The indexing indicates that the diameters of cells in $\mathcal{T}_H$ are of order H whereas the diameters of cells in $\mathcal{T}_h$ are of order h, where $h \leq H$. Since the cells involved in these experiments always have diameter smaller than 1, to maintain a cleaner presentation the notation

$$1/H \ : \ 1/h$$

to indicate the ratio between coarse and find grids will frequently be used.

2.5.1 Comparison of the Nonoverlapping and the Overlapping Preconditioners

The two preconditioners constructed in this chapter share much in common, yet it remains to discover which offers the better performance. The subdomain matrices corresponding to both methods have a sparse block structure, a feature which may be exploited for improved efficiency when solving the subdomain systems. However, the

Table 2.1: (P1) Comparison of solvers. κ estimates, (iterations) for $q = 1$

	NOV PCG			OV PCG			CG
$1/h$ \ $1/H$	4	8	16	4	8	16	
16	14.0 (36)	6.4 (23)	-	5.3 (21)	4.3 (19)	-	149.5 (115)
32	27.7 (49)	14.0 (34)	6.2 (22)	6.7 (24)	4.6 (19)	4.1 (18)	444.3 (196)
64	51.8 (65)	28.0 (49)	13.5 (33)	9.2 (28)	5.5 (21)	4.2 (18)	1362.1 (338)
128	106.7 (93)	55.8 (66)	26.4 (44)	13.1 (33)	8.0 (26)	5.0 (20)	4631.9 (647)

subdomain matrices corresponding to the overlapping method have greater dimension due the contributions from the overlap. One important question is whether the increased computational cost from this increased matrix dimension will be offset by gains made elsewhere.

In addition, we will seek to verify aspects of the theory predicted by Theorem 2.2 and Theorem 2.3; namely, that the nonoverlapping preconditioned system has a condition number of order $O(\frac{H}{h})$ and the overlapping preconditioned system has a condition number of order $O(\frac{H}{\delta})$.

Table 2.1 offers a comparison of the three solvers when applied to test problem (P1) using $\gamma_h = 10.0$, degree $q = 1$, and a variety of coarse and fine meshes. Note that the combination of $1/H = 1/h = 16$ is excluded. In this case the coarse grid solve provides the solution on the fine mesh as well, thus obviating the need for any further work. From the results it is clear that both Schwarz methods offer an immediate and considerable improvement over the standard conjugate gradient method, both in terms of condition numbers and iterations. In addition, it appears as though the overlapping method results in much smaller condition numbers relative to the nonoverlapping method, with similar comparisons to be made for the iteration counts.

The adherence to the $O(\frac{H}{h})$ law predicted by Theorem 2.2 for the nonoverlapping method is evident; each condition number roughly increases by a factor of two as

Table 2.2: (P2) Comparison of solvers. κ estimates, (iterations) for $q = 1$

	NOV PCG			OV PCG			CG
$1/h$ \ $1/H$	4	8	16	4	8	16	
16	14.3 (40)	6.7 (27)	-	4.9 (23)	4.3 (21)	-	177.6 (143)
32	26.0 (55)	15.2 (41)	7.3 (28)	5.5 (25)	4.3 (22)	3.9 (21)	571.1 (259)
64	48.7 (75)	27.3 (56)	15.0 (41)	6.3 (27)	4.8 (23)	4.1 (21)	1964.1 (489)
128	87.8 (102)	48.7 (73)	27.8 (56)	9.1 (32)	6.7 (27)	4.8 (23)	6341.6 (894)

$1/h$ doubles. On the other hand, the overlapping method is stubbornly refusing to follow the $O(\frac{H}{\delta})$ rule (recall that $\delta \approx h$ in these experiments). Instead, the condition numbers in this case grow by a factor much less than two for initial refinements, with the rate of growth that seems to increase as the number of refinements grows.

Table 2.2 displays the results for test problem (P2) again using $\gamma_h = 10.0$, degree $q = 1$, and the same selection of coarse and fine meshes. The performance in this case when compared with (P1) is quite similar for all three methods, even though this test problem is highly oscillatory when compared with the bubble function represented in (P1). Again, both preconditioning methods offer substantial improvement over the performance of the standard CG method. Note that the results from CG deteriorate somewhat when compared with (P1), while the performance of the preconditioned methods actually improve slightly. The adherence to the $O(\frac{H}{h})$ law for the nonoverlapping preconditioner is again quite strong. The rate of growth for the condition number in the overlapping case is still underperforming the $O(\frac{H}{\delta})$ law, although it does appear to be growing as h increases.

Table 2.3 exhibits the performance of the three methods for test problem (P2). For this problem we again choose $\gamma_h = 10.0$ and utilize the same array of coarse and fine meshes, but for variety we choose to use piecewise quadratic polynomials $q = 2$ to construct the DG approximation. Note that the subdomain partition in this

Table 2.3: (P3) Comparison of solvers. κ estimates, (iterations) for $q = 2$

	NOV PCG			OV PCG			CG
$1/h$ \ $1/H$	4	8	16	4	8	16	
16	11.6 (34)	5.8 (23)	-	4.3 (21)	3.7 (18)	-	514.9 (231)
32	22.1 (47)	11.0 (32)	5.5 (22)	4.9 (22)	3.9 (19)	3.6 (19)	1986.7 (461)
64	42.0 (65)	21.1 (45)	10.3 (31)	6.5 (26)	4.4 (21)	3.6 (18)	7707.6 (922)
128	79.5 (90)	40.1 (63)	19.7 (43)	9.7 (32)	5.9 (24)	4.1 (20)	29863.0 (1843)

case only contains 12 subdomains; the domain is L-shaped, but we follow the same "checkerboard" pattern for assigning subdomains. The results are much in keeping with the previous test problems – the performance of CG worsens even further while the preconditioned methods remain very consistent. The overlapping method again provides the best performance. The adherence to the $O(\frac{H}{h})$ law for the nonoverlapping preconditioner is again easily seen, while the overlapping case persists in falling short of the $O(\frac{H}{\delta})$ rule.

2.5.2 Dependence on the Penalty Parameter γ

The primary role of the parameter γ is to penalize the jump terms so that the bilinear form $a_h^{\gamma_h}$ is coercive. In theory, we only need the condition that γ is larger than some constant γ_0 (depending on the polynomial degree of the approximation space V^h) to guarantee that this is the case. However, if γ is taken too large, the condition number of the resulting stiffness matrix will suffer. There is no existing theory that dictates how to choose γ in a precise manner, so the choice is often made heuristically.

Previously in this section, we showed that the condition number of the nonoverlapping preconditioned system has an explicit dependence on γ (Theorem 2.2). Similarly, the condition number of the overlapping preconditioned system is independent of γ, assuming that the penalty compatibility condition (PC) is enforced (Theorem 2.3).

In the experiments that follow we will enforce (PC) and investigate the role of γ computationally.

Tables 2.4-2.5 display the performance of both of the PCG methods for (P1). Here we use $q = 1$, with a minimal overlap $\delta \approx h$ for the overlapping PCG experiments in this section. Results are shown for several combinations of coarse and fine meshes and for $\gamma = 10, 100, 1000$ and 10000. For comparison, Table 2.6 reports corresponding estimates of $\kappa(A)$, iteration counts, and runtimes when using the CG method without preconditioning.

As an example, Figures 2.4-2.6 illustrate the relative performance of each method for the particular meshes $h = 1/64$ and $H = 1/8$. In terms of improvement in condition number, iteration count, and runtime, the advantages gained by the preconditioning are immediately obvserved. The result of increasing γ is obviously quite detrimental to the results for CG. More importantly, the dependence upon γ for the nonoverlapping method predicted by Theorem 2.2 is evident, both in the illustration offered by Figures 2.4-2.6 and in the raw numbers reported in Tables 2.4-2.5. On the other hand, the independence of γ for the overlapping method predicted by Theorem 2.101 is exceptionally clear, as condition numbers, iteration counts, and runtimes remain remarkably stable as the penalty parameter is increased.

The performance of both of the PCG methods for (P2) is shown by Tables 2.7-2.8. Again we use $q = 1$ with minimal overlap $\delta \approx h$ for the overlapping PCG method. Table 2.9 offers the corresponding results for the CG method. Figures 2.7-2.9 offer a sample visualization of the reults for the particular meshes $h = 1/64$ and $H = 1/8$. Similarly, Tables 2.10-2.12 report the results for (P3) with $q = 2$ and minimal overlap, while Figures 2.10-2.12 give a sample illustration for $h = 1/64$ and $H = 1/8$. In all cases, the results are qualitatively identical. The performance of CG is drastically reduced as γ increases, while the dependence on γ for the nonoverlapping method is clearly evinced. Finally, the overlapping method is almost completely impervious to variations in γ. The theoretical predictions made by Theorems 2.2 and 2.101 are supported quite strongly by these experiments.

Table 2.4: Variations of γ (PCG). (P1) with $q = 1$, κ estimates (iterations)

	Nonoverlapping PCG: $\kappa(BA)$, (iterations)				Overlapping PCG: $\kappa(BA)$, (iterations)			
$\frac{1}{H} : \frac{1}{h}$	10	100	1000	10000	10	100	1000	10000
4 : 16	14.0 (36)	53.7 (68)	125.7 (107)	200.2 (133)	5.3 (21)	5.7 (22)	5.4 (21)	5.2 (21)
4 : 32	27.7 (49)	145.4 (110)	374.9 (179)	653.9 (236)	6.7 (24)	6.9 (24)	6.8 (24)	6.6 (24)
4 : 64	51.8 (65)	318.8 (159)	931.7 (272)	1671.6 (367)	9.2 (28)	9.8 (29)	9.6 (29)	9.5 (29)
8 : 16	6.4 (23)	22.9 (44)	44.0 (61)	79.4 (81)	4.3 (19)	4.5 (19)	4.5 (19)	4.3 (19)
8 : 32	14.0 (34)	67.9 (74)	181.8 (119)	325.5 (160)	4.6 (19)	4.8 (20)	4.7 (20)	4.5 (19)
8 : 64	28.0 (49)	165.6 (115)	483.4 (196)	931.2 (274)	5.5 (21)	5.6 (21)	5.5 (21)	5.4 (21)
16 : 32	6.2 (22)	20.4 (40)	46.4 (61)	73.4 (76)	4.1 (18)	3.9 (18)	3.8 (17)	3.8 (17)
16 : 64	13.5 (33)	74.0 (77)	208.8 (124)	350.6 (164)	4.2 (18)	4.2 (18)	4.1 (18)	3.9 (17)

Table 2.5: Variations of γ (PCG). (P1) with $q = 1$, CPU runtimes

	NOV PCG: CPU time				OV PCG: CPU time			
$\frac{1}{H} : \frac{1}{h}$	10	100	1000	10000	10	100	1000	10000
4 : 16	0.043	0.064	0.098	0.117	0.044	0.049	0.043	0.048
4 : 32	0.239	0.491	0.731	0.990	0.244	0.254	0.247	0.261
4 : 64	1.678	3.735	6.149	8.267	1.514	1.646	1.552	1.578
8 : 16	0.035	0.052	0.063	0.080	0.041	0.043	0.040	0.042
8 : 32	0.188	0.331	0.514	0.665	0.205	0.212	0.213	0.206
8 : 64	1.291	2.817	4.527	6.709	1.337	1.332	1.332	1.334
16 : 32	0.191	0.222	0.318	0.393	0.202	0.205	0.198	0.201
16 : 64	1.002	1.942	2.975	3.865	1.255	1.256	1.257	1.222

Table 2.6: Variations of γ (CG). (P1) with $q = 1$

	CG: $\kappa(A)$, (iterations), CPU runtime			
$\frac{1}{h}$	10	100	1000	10000
16	149.5 (115) 0.068	604.9 (229) 0.134	1199.5 (320) 0.187	1942.0 (424) 0.247
32	444.3 (196) 0.491	1962.8 (407) 1.122	4838.4 (662) 1.780	7234.5 (761) 1.869
64	1362.1 (338) 6.265	7162.9 (800) 14.197	15062.5 (1106) 19.665	27281.0 (1487) 26.419

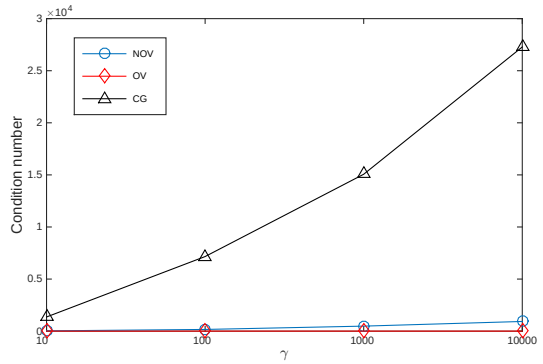


Figure 2.4: (P1) Variations of γ. κ estimates, $q = 1$

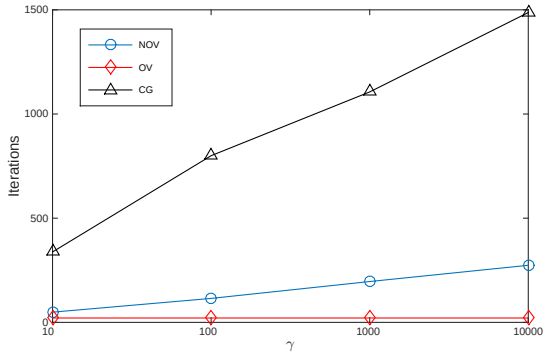


Figure 2.5: (P1) Variations of γ. Iteration counts, $q = 1$

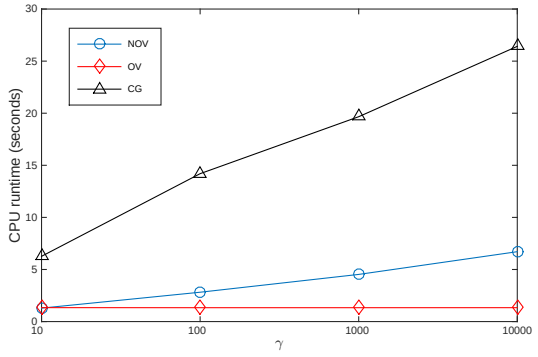


Figure 2.6: (P1) Variations of γ. CPU runtimes, $q = 1$

Table 2.7: Variations of γ (PCG). (P2) with $q = 1$, κ estimates (iterations)

	Nonoverlapping PCG: $\kappa(BA)$, (iterations)				Overlapping PCG: $\kappa(BA)$, (iterations)			
$\frac{1}{H} : \frac{1}{h}$	10	100	1000	10000	10	100	1000	10000
4 : 16	14.3 (40)	72.8 (90)	145.5 (129)	255.1 (163)	4.9 (23)	5.6 (24)	5.7 (24)	5.5 (23)
4 : 32	26.0 (55)	182.4 (145)	461.7 (229)	839.6 (309)	5.5 (25)	5.9 (27)	5.9 (26)	5.7 (26)
4 : 64	48.7 (75)	347.1 (199)	1026.9 (341)	1859.0 (461)	6.3 (27)	6.2 (27)	5.9 (26)	5.6 (25)
8 : 16	6.7 (27)	28.1 (53)	60.3 (76)	94.5 (96)	4.3 (21)	4.8 (22)	4.9 (22)	4.7 (21)
8 : 32	15.2 (41)	82.0 (94)	221.4 (156)	395.7 (206)	4.3 (22)	4.6 (22)	4.6 (22)	4.5 (22)
8 : 64	27.3 (56)	182.0 (140)	541.9 (241)	1025.7 (332)	4.8 (23)	4.6 (22)	4.5 (22)	4.5 (22)
16 : 32	7.3 (28)	32.5 (60)	72.7 (89)	121.1 (113)	3.9 (21)	4.1 (21)	4.0 (21)	3.9 (21)
16 : 64	15.0 (41)	93.6 (102)	239.5 (160)	456.6 (216)	4.1 (21)	3.9 (20)	3.9 (21)	3.8 (20)

Table 2.8: Variations of γ (PCG). (P2) with $q = 1$, CPU runtimes

	NOV PCG: CPU runtimes				OV PCG: CPU runtimes			
$\frac{1}{H} : \frac{1}{h}$	10	100	1000	10000	10	100	1000	10000
4 : 16	0.044	0.086	0.116	0.148	0.045	0.048	0.047	0.046
4 : 32	0.266	0.579	0.906	1.209	0.249	0.259	0.252	0.254
4 : 64	1.848	4.589	7.734	10.463	1.517	1.557	1.487	1.467
8 : 16	0.036	0.057	0.078	0.092	0.042	0.053	0.051	0.043
8 : 32	0.205	0.414	0.656	0.830	0.229	0.225	0.231	0.233
8 : 64	1.469	3.311	5.494	7.647	1.645	1.423	1.448	1.425
16 : 32	0.175	0.297	0.433	0.537	0.231	0.231	0.246	0.238
16 : 64	1.216	2.466	3.791	4.925	1.375	1.335	1.353	1.381

Table 2.9: Variations of γ (CG). (P2) with $q = 1$

	CG: $\kappa(A)$, (iterations), CPU runtime			
$\frac{1}{h}$	10	100	1000	10000
16	177.6 (143) 0.085	845.8 (302) 0.177	2012.9 (452) 0.263	3282.5 (562) 0.332
32	571.1 (259)	3353.1 (626) 1.507	7978.3 (948) 2.303	13216.9 (1269) 3.082
64	1964.1 (489) 8.835	11581.9 (1180) 20.807	27825.1 (1826) 32.305	41319.4 (2160) 41.664

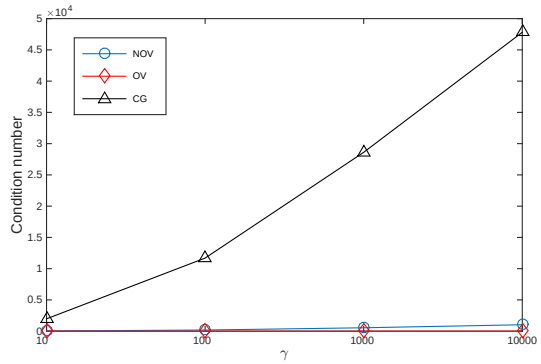


Figure 2.7: (P2) Variations of γ. κ estimates, $q = 1$

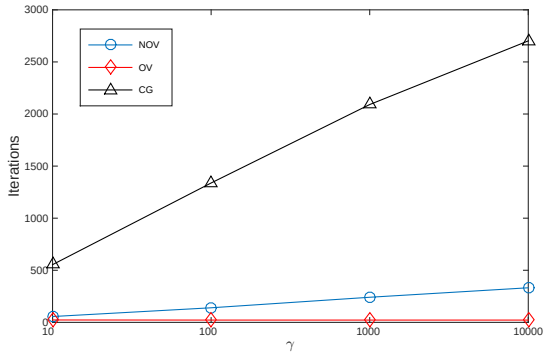


Figure 2.8: (P2) Variations of γ. Iteration counts, $q = 1$

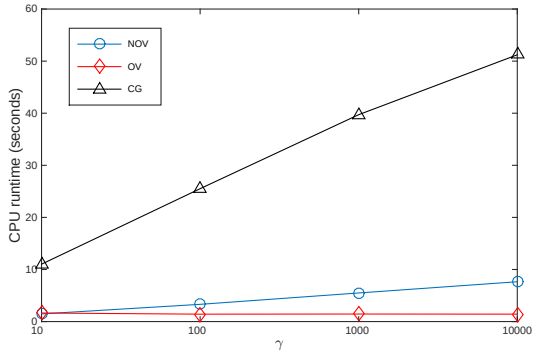


Figure 2.9: (P2) Variations of γ. CPU runtimes, $q = 1$

Table 2.10: Variations of γ (PCG). (P3) with $q = 2$, κ estimates (iterations)

	Nonoverlapping PCG: $\kappa(BA)$, (iterations)				Overlapping PCG: $\kappa(BA)$, (iterations)			
$\frac{1}{H} : \frac{1}{h}$	10	100	1000	10000	10	100	1000	10000
4 : 16	11.6 (34)	81.6 (93)	251.5 (171)	456.7 (246)	4.3 (21)	4.2 (21)	4.1 (22)	4.0 (23)
4 : 32	22.1 (47)	161.9 (132)	530.9 (252)	1011.5 (366)	4.9 (22)	5.0 (23)	4.8 (24)	4.6 (24)
4 : 64	42.0 (65)	311.4 (184)	1068.8 (360)	2056.5 (527)	6.5 (26)	6.4 (27)	6.1 (27)	5.9 (28)
8 : 16	5.8 (23)	30.1 (56)	97.2 (106)	159.0 (141)	3.7 (18)	3.5 (19)	3.4 (19)	3.3 (19)
8 : 32	11.0 (32)	85.6 (94)	303.2 (189)	564.3 (266)	3.9 (19)	3.7 (20)	3.6 (20)	3.6 (21)
8 : 64	21.1 (45)	174.6 (136)	656.6 (278)	1280.2 (405)	4.4 (21)	4.4 (22)	4.1 (22)	3.9 (22)
16 : 32	5.5 (22)	28.9 (54)	96.5 (105)	167.4 (144)	3.6 (19)	3.4 (19)	3.2 (19)	3.2 (20)
16 : 64	10.3 (31)	82.5 (92)	346.6 (195)	652.3 (285)	3.6 (18)	3.5 (19)	3.3 (19)	3.3 (20)

Table 2.11: Variations of γ (PCG). (P3) with $q = 2$, CPU runtimes

	NOV PCG: CPU runtimes				OV PCG: CPU runtimes			
$\frac{1}{H} : \frac{1}{h}$	10	100	1000	10000	10	100	1000	10000
4 : 16	0.071	0.150	0.262	0.359	0.077	0.079	0.081	0.083
4 : 32	0.455	1.077	1.989	2.799	0.457	0.466	0.479	0.481
4 : 64	3.146	7.626	14.241	20.512	3.951	4.043	4.038	4.099
8 : 16	0.055	0.104	0.175	0.228	0.075	0.077	0.077	0.079
8 : 32	0.340	0.804	1.499	2.075	0.429	0.440	0.439	0.444
8 : 64	2.483	5.812	11.106	15.956	3.661	3.723	3.712	3.704
16 : 32	0.285	0.548	0.952	1.277	0.431	0.432	0.433	0.454
16 : 64	1.863	4.460	8.233	11.909	3.475	3.563	3.547	3.605

Table 2.12: Variations of γ (CG). (P3) with $q = 2$

	CG: $\kappa(A)$, (iterations), CPU runtime			
$\frac{1}{h}$	10	100	1000	10000
16	514.9 (231) 0.351	2717.2 (560) 0.761	11056.5 (1186) 1.615	22816.4 (1796) 2.438
32	1986.7 (461) 2.874	10574.0 (1122) 6.918	45216.1 (2440) 15.692	95172.7 (3720) 23.254
64	7707.6 (922) 30.328	41245.2 (2247) 73.511	179087.8 (4925) 161.154	382748.6 (7556) 247.143

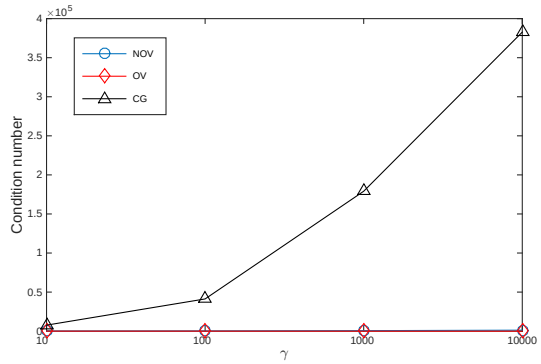


Figure 2.10: (P3) Variations of γ. κ estimates, $q = 2$

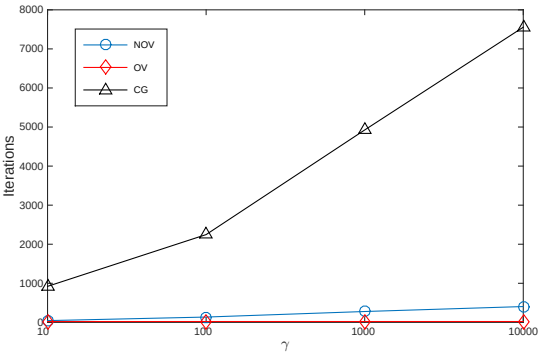


Figure 2.11: (P3) Variations of γ. Iteration counts, $q = 2$

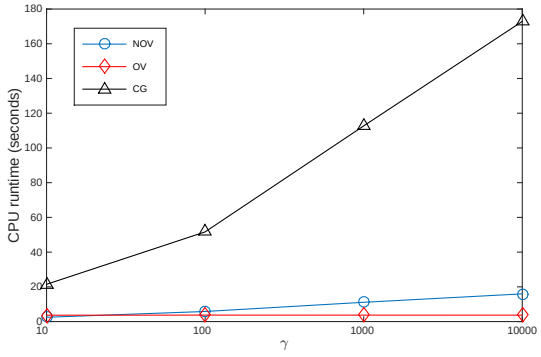


Figure 2.12: (P3) Variations of γ. CPU runtimes, $q = 2$

2.5.3 Penalty Compatibility for the Discontinuous Coarse Subspace

Recall from Section 2.3 that choosing to use a completely discontinuous coarse subspace $V^{H,D}$ when constructing the coarse mesh component of the Schwarz preconditioner necessitates the enforcement of (PC) in order for the direct definition of the coarse solver (2.43) to coincide with the definition by restriction (2.44). Here we will examine computationally the effects of failing to enforce the penalty compatibility condition.

For a nice variety of results, we investigate the effect of not enforcing (PC) over the same range of γ used in the previous section; i.e., $\gamma = 10, 100, 1000$, and 10000 for each test problem under identical conditions as the experiments reported previously.

The effects of not enforcing (PC) for (P1) are reported in Tables 2.13 and 2.14. These results should be compared to those in Tables 2.4 and 2.5, respectively. It is immediately clear that the performance of both preconditioners suffers from the lack of (PC), in terms of condition numbers, iteration counts, and runtimes. As an example, Figures 2.13 - 2.15 display comparisons for the nonoverlapping method between the results for $1/h = 64$ and $1/H = 8$ with (PC) both enforced and not enforced, across all three performance metrics. The corresponding comparison for the overlapping method is shown in Figures 2.16 - 2.18. It appears that the negative effect of failing to enforce (PC) is exacerbated in the nonoverlapping method as γ is allowed to increase, whereas in the overlapping case it seems there may be a limit to how drastically the results may be affected.

Similar comparisons for test problem (P2) are shown in Tables 2.15, 2.16, and Figures 2.13 - 2.18. Corresponding results for test problem (P3) are shown in Tables 2.17, 2.18, and Figures 2.25 - 2.30. All experiments corroborate the need to enforce (PC) when using the discontinuous coarse subspace $V^{H,D}$. It should be reiterated that the enforcement of this condition is negligible in implementation, as it consists simply of a rescaling by an integer factor at each level of refinement.

Table 2.13: (PC) not enforced. (P1), $q = 1$, κ estimates (iterations)

	Nonoverlapping PCG: $\kappa(BA)$, (iterations)				Overlapping PCG: $\kappa(BA)$, (iterations)			
$\frac{1}{H} : \frac{1}{h}$	10	100	1000	10000	10	100	1000	10000
4 : 16	23.7 (45)	101.0 (92)	205.4 (134)	304.8 (164)	8.6 (27)	8.9 (28)	8.6 (27)	8.2 (27)
4 : 32	70.2 (76)	311.2 (161)	758.6 (251)	1219.3 (320)	15.7 (38)	14.1 (35)	13.6 (34)	13.3 (34)
4 : 64	221.4 (135)	750.2 (246)	1912.6 (395)	3213.1 (517)	27.0 (49)	23.2 (46)	22.3 (45)	22.1 (44)
8 : 16	8.6 (27)	26.2 (47)	57.8 (68)	84.1 (84)	5.1 (21)	5.4 (21)	5.2 (21)	5.0 (20)
8 : 32	27.4 (47)	110.8 (94)	262.9 (143)	455.3 (189)	7.9 (26)	7.6 (25)	7.6 (25)	7.2 (25)
8 : 64	87.8 (82)	310.4 (155)	818.1 (256)	1528.8 (354)	15.3 (36)	12.1 (32)	11.0 (30)	10.8 (30)
16 : 32	8.4 (25)	26.6 (46)	61.2 (69)	95.0 (85)	4.8 (20)	4.6 (19)	4.4 (19)	4.4 (19)
16 : 64	26.5 (45)	114.2 (94)	298.8 (150)	568.3 (204)	7.1 (24)	6.7 (23)	6.2 (22)	6.0 (22)

Table 2.14: (PC) not enforced. (P1), $q = 1$, CPU runtimes

	NOV PCG: CPU runtimes				OV PCG: CPU runtimes			
$\frac{1}{H} : \frac{1}{h}$	10	100	1000	10000	10	100	1000	10000
4 : 16	0.051	0.083	0.121	0.141	0.049	0.048	0.047	0.048
4 : 32	0.322	0.627	0.933	1.275	0.317	0.291	0.283	0.284
4 : 64	3.123	5.307	8.578	11.306	2.092	1.996	1.958	1.951
8 : 16	0.034	0.054	0.067	0.082	0.044	0.044	0.041	0.042
8 : 32	0.215	0.444	0.589	0.794	0.237	0.237	0.238	0.237
8 : 64	1.954	3.507	5.533	7.679	1.725	1.646	1.550	1.554
16 : 32	0.152	0.226	0.375	0.411	0.214	0.208	0.205	0.207
16 : 64	1.283	2.258	3.494	4.702	1.393	1.359	1.325	1.343

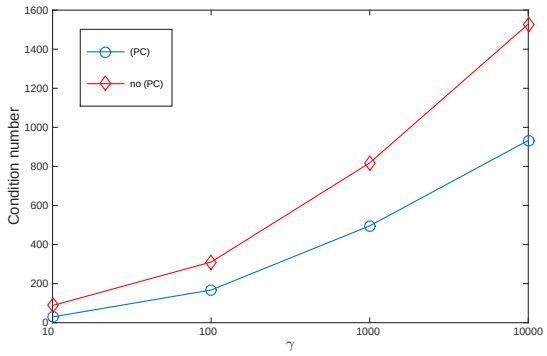


Figure 2.13: (P1), NOV. Effect of (PC) for κ, $q = 1$

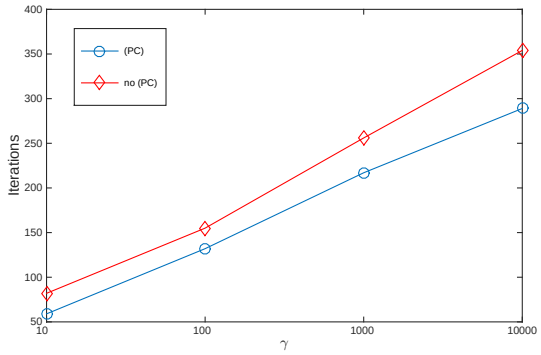


Figure 2.14: (P1), NOV. Effect of (PC) for iteration counts, $q = 1$

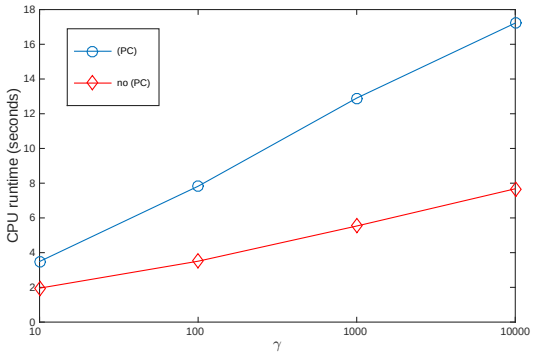


Figure 2.15: (P1), NOV. Effect of (PC) for CPU runtimes, $q = 1$

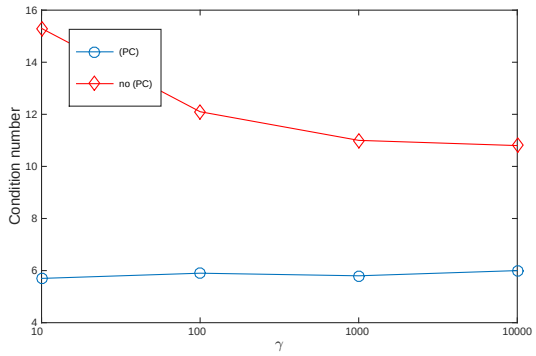


Figure 2.16: (P1), OV. Effect of (PC) for κ, $q = 1$

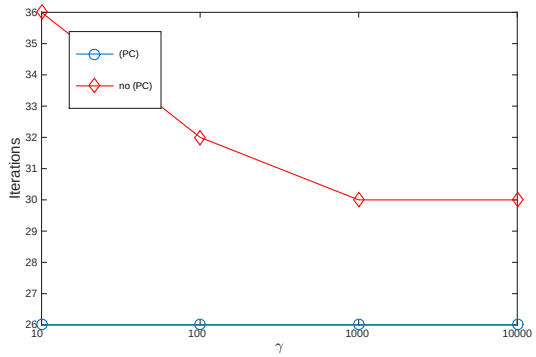


Figure 2.17: (P1), OV. Effect of (PC) for iteration counts, $q = 1$

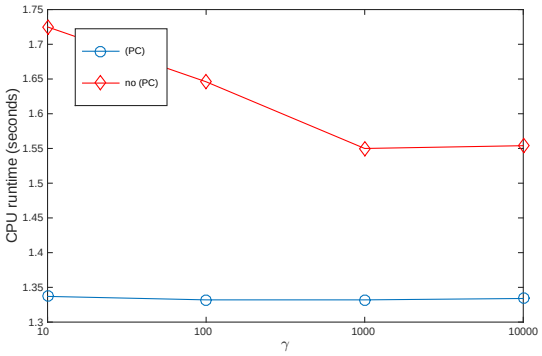


Figure 2.18: (P1), OV. Effect of (PC) for CPU runtimes, $q = 1$

Table 2.15: (PC) not enforced. (P2), $q = 1$, κ estimates (iterations)

	Nonoverlapping PCG: $\kappa(BA)$, (iterations)				Overlapping PCG: $\kappa(BA)$, (iterations)			
$\frac{1}{H} : \frac{1}{h}$	10	100	1000	10000	10	100	1000	10000
4 : 16	21.5 (50)	111.8 (108)	242.9 (157)	406.0 (204)	8.2 (30)	8.6 (30)	8.9 (30)	8.6 (30)
4 : 32	67.2 (87)	342.7 (195)	792.2 (297)	1443.0 (402)	12.8 (39)	11.5 (38)	11.3 (37)	11.0 (37)
4 : 64	200.6 (151)	781.1 (293)	2028.2 (470)	3430.9 (622)	18.9 (48)	14.4 (42)	13.4 (40)	13.0 (39)
8 : 16	8.8 (31)	33.2 (58)	67.4 (80)	121.1 (109)	5.4 (24)	5.6 (24)	5.6 (23)	5.6 (23)
8 : 32	28.7 (56)	133.3 (120)	319.1 (185)	607.8 (258)	7.2 (29)	7.3 (28)	7.1 (28)	6.9 (28)
8 : 64	84.8 (98)	354.4 (196)	979.7 (327)	1743.6 (434)	13.4 (39)	9.4 (32)	9.3 (33)	8.9 (32)
16 : 32	10.1 (33)	36.9 (63)	89.2 (101)	142.6 (125)	4.7 (23)	4.7 (23)	4.7 (23)	4.4 (22)
16 : 64	29.1 (56)	139.4 (121)	357.1 (193)	642.7 (258)	7.3 (28)	6.4 (25)	6.0 (26)	5.8 (26)

Table 2.16: (PC) not enforced. (P2), $q = 1$, CPU runtimes

	NOV PCG: CPU runtimes				OV PCG: CPU runtimes			
$\frac{1}{H} : \frac{1}{h}$	10	100	1000	10000	10	100	1000	10000
4 : 16	0.056	0.107	0.140	0.178	0.050	0.053	0.050	0.050
4 : 32	0.391	0.860	1.277	1.553	0.329	0.315	0.314	0.309
4 : 64	3.679	6.695	10.668	13.864	2.110	1.935	1.877	1.853
8 : 16	0.042	0.061	0.082	0.113	0.045	0.048	0.046	0.044
8 : 32	0.275	0.503	0.749	1.025	0.265	0.261	0.261	0.263
8 : 64	2.378	4.501	7.438	9.657	1.865	1.665	1.683	1.667
16 : 32	0.228	0.322	0.480	0.594	0.234	0.243	0.241	0.231
16 : 64	1.467	2.906	4.768	6.022	1.550	1.470	1.476	1.477

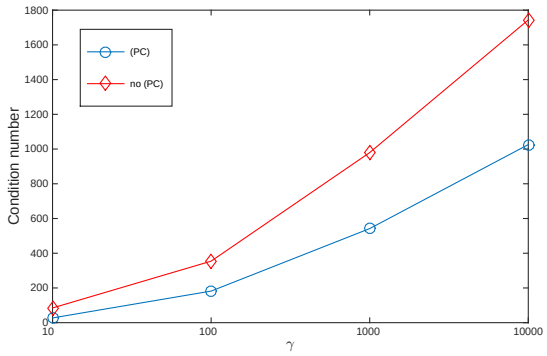


Figure 2.19: (P2), NOV. Effect of (PC) for κ, $q = 1$

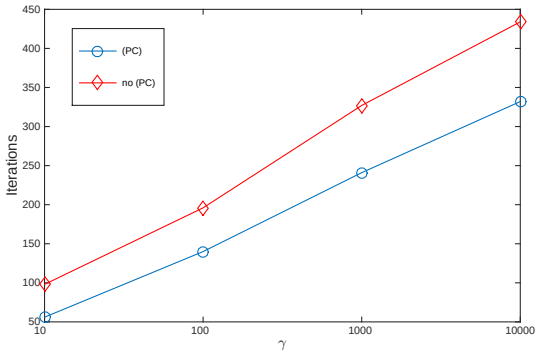


Figure 2.20: (P2), NOV. Effect of (PC) for iteration counts, $q = 1$

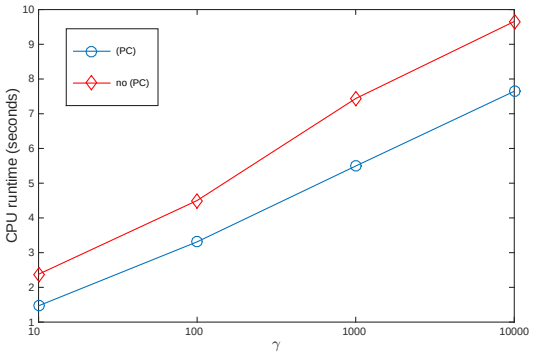


Figure 2.21: (P2), NOV. Effect of (PC) for CPU runtimes, $q = 1$

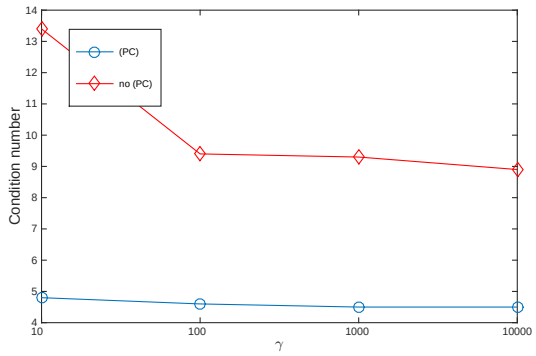


Figure 2.22: (P2), OV. Effect of (PC) for κ, $q = 1$

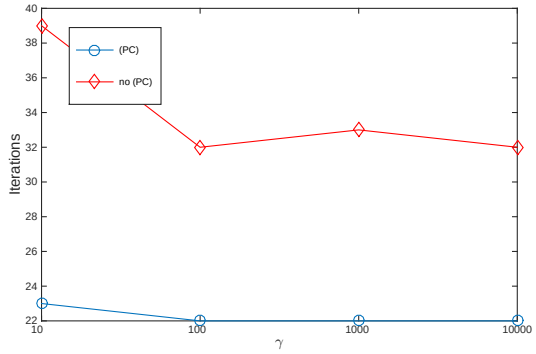


Figure 2.23: (P2), OV. Effect of (PC) for iteration counts, $q = 1$

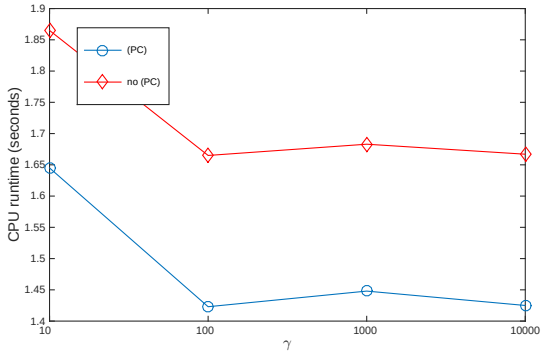


Figure 2.24: (P2), OV. Effect of (PC) for CPU runtimes, $q = 1$

Table 2.17: (PC) not enforced. (P3), $q = 1$, κ estimates (iterations)

	Nonoverlapping PCG: $\kappa(BA)$, (iterations)				Overlapping PCG: $\kappa(BA)$, (iterations)			
$\frac{1}{H} : \frac{1}{h}$	10	100	1000	10000	10	100	1000	10000
4 : 16	39.1 (62)	137.6 (122)	371.9 (210)	650.4 (289)	11.4 (34)	7.0 (28)	6.5 (28)	6.4 (29)
4 : 32	134.8 (115)	335.6 (189)	922.4 (332)	1670.6 (468)	25.1 (50)	10.9 (35)	9.6 (34)	9.1 (35)
4 : 64	396.9 (199)	788.3 (294)	2116.9 (507)	3697.8 (706)	59.7 (78)	18.5 (46)	14.5 (44)	13.6 (44)
8 : 16	11.0 (32)	36.0 (61)	113.9 (115)	196.1 (159)	5.8 (24)	4.4 (21)	4.1 (22)	3.9 (22)
8 : 32	41.6 (64)	130.6 (117)	433.7 (224)	813.1 (323)	11.7 (34)	6.2 (26)	5.7 (26)	5.5 (27)
8 : 64	144.1 (117)	332.4 (187)	1034.1 (345)	1890.7 (492)	26.9 (53)	9.8 (33)	8.4 (32)	7.8 (32)
16 : 32	10.5 (32)	33.8 (59)	109.7 (113)	208.9 (163)	5.6 (24)	4.1 (21)	3.8 (21)	3.7 (22)
16 : 64	41.3 (62)	120.8 (110)	452.8 (227)	869.9 (331)	11.5 (34)	5.8 (25)	5.4 (26)	5.1 (26)

Table 2.18: (PC) not enforced. (P3), $q = 1$, CPU runtimes

	NOV PCG: CPU runtimes				OV PCG: CPU runtimes			
$\frac{1}{H} : \frac{1}{h}$	10	100	1000	10000	10	100	1000	10000
4 : 16	0.108	0.191	0.309	0.411	0.103	0.091	0.096	0.096
4 : 32	0.956	1.504	2.536	3.609	0.748	0.593	0.583	0.598
4 : 64	8.172	11.792	19.811	27.289	7.305	5.243	5.110	5.118
8 : 16	0.072	0.113	0.188	0.254	0.090	0.081	0.082	0.082
8 : 32	0.590	0.979	1.783	2.507	0.587	0.507	0.501	0.511
8 : 64	5.099	7.726	13.616	19.540	5.702	4.421	4.360	4.348
16 : 32	0.360	0.599	1.034	1.435	0.489	0.460	0.465	0.469
16 : 64	3.059	4.891	9.387	13.363	4.512	3.923	3.994	3.987

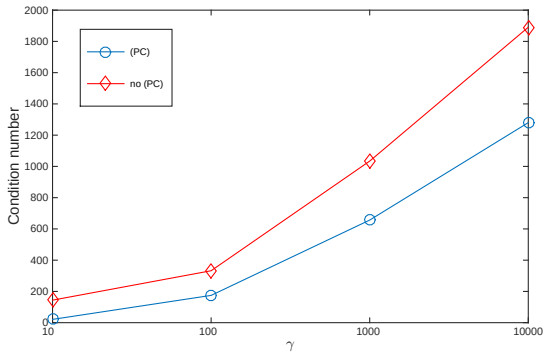


Figure 2.25: (P3), NOV. Effect of (PC) for κ, $q = 2$

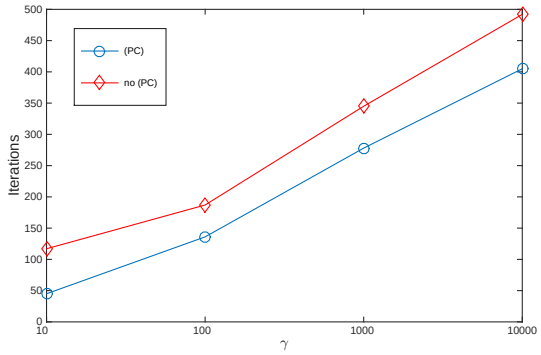


Figure 2.26: (P3), NOV. Effect of (PC) for iteration counts, $q = 2$

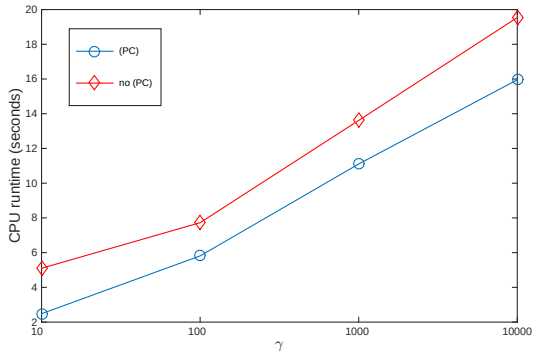


Figure 2.27: (P3), NOV. Effect of (PC) for CPU runtimes, $q = 2$

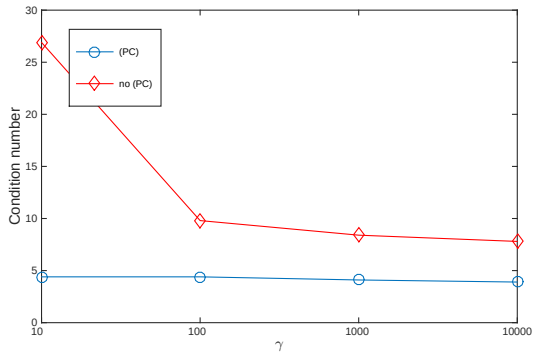


Figure 2.28: (P3), OV. Effect of (PC) for κ, $q = 2$

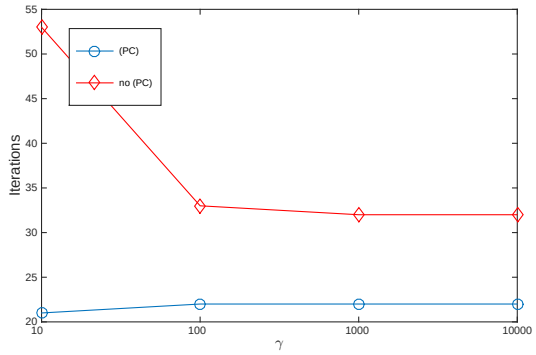


Figure 2.29: (P3), OV. Effect of (PC) for iteration counts, $q = 2$

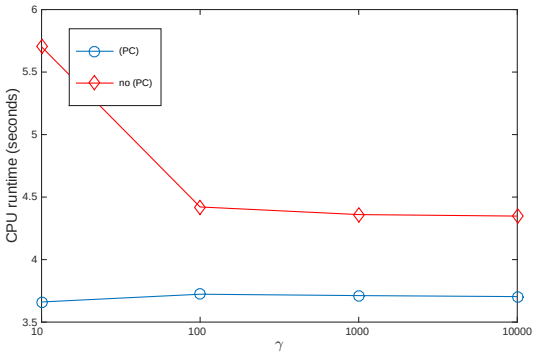


Figure 2.30: (P3), OV. Effect of (PC) for CPU runtimes, $q = 2$

Chapter 3

Fourth Order Elliptic Problems

3.1 Preliminaries

3.1.1 Model Problem

Let $\Omega \subset \mathbb{R}^d,\ d = 2, 3$ be a bounded polygonal domain with boundary $\partial\Omega$. Consider the following model problem:

$$\Delta^2 u = f \quad \text{in} \quad \Omega, \tag{3.1}$$

$$u = g_D \quad \text{on} \quad \partial\Omega \tag{3.2}$$

$$\frac{\partial u}{\partial n} = g_N \quad \text{on} \quad \partial\Omega. \tag{3.3}$$

In the case $d = 2$, the above problem models the bending of a clamped elastic plate subject to the external load f and u stands for the vertical displacement of the plate.

3.1.2 Partitions of Ω

The notation introduced in Chapter 2 will be used throughout this chapter as well, particularly mesh, cell, and edge designations. The assumptions made on $\mathcal{T}_h$, similar to those made in Section 2.1.2, are as follows for the fourth order problem.

(P1) The elements of $\mathcal{T}_h$ are *shape regular*.

(P2) $\mathcal{T}_h$ is locally *quasi-uniform*; i.e., if two cells K_i and K_j share an edge then $\text{diam}(K_i) \approx \text{diam}(K_j)$.

(P3) $\mathcal{T}_h$ is conforming, i.e. no hanging nodes are allowed. As before, this assumption is mainly imposed to simplify the proofs and may certainly be relaxed in actual computations.

The definitions and notations used for broken Sobolev spaces, edge integrals, and DG jumps and averages are precisely the same as in Section 2.1.2 as well.

3.1.3 Discontinuous Galerkin Formulation

On the energy space $E_h = H^s(\mathcal{T}_h)$, $s > 7/2$, define the bilinear form $a_h(\cdot, \cdot)$ as

$$\begin{aligned} a_h(u,v) \;=\; & \sum_{K\in\mathcal{T}_h} (\Delta u, \Delta v)_K \\ & + \sum_{e_h\in\mathcal{E}_h} \left(\langle\{\partial_n \Delta u\}, [v]\rangle_{e_h} + \langle\{\partial_n \Delta v\}, [u]\rangle_{e_h} + \frac{\sigma_h}{|e_h|^3} \langle [u], [v]\rangle_{e_h} \right) \\ & - \sum_{e_h\in\mathcal{E}_h} \left(\langle\{\Delta u\}, [\partial_n v]\rangle_{e_h} + \langle\{\Delta v\}, [\partial_n u]\rangle_{e_h} - \frac{\tau_h}{|e_h|} \langle [\partial_n u], [\partial_n v]\rangle_{e_h} \right). \end{aligned} \tag{3.4}$$

To contrast with the bilinear form used for the second order problem (2.5), the fourth order bilinear form contains penalty terms corresponding to both jumps and jumps of normal derivatives on edges. Here, the penalty parameters σ_h and τ_h must each be larger than some threshold in order to guarantee the stability of the method.

The bilinear form presented above corresponds to Arnold's formulation; Baker's formulation is given by

$$\begin{aligned} a_h(u,v) &= \sum_{K\in\mathcal{T}_h} (\Delta u, \Delta v)_K \\ &\quad + \sum_{e_h\in\mathcal{E}_h} \left(\langle \partial_n \Delta u, [v]\rangle_{e_h} + \langle \partial_n \Delta v, [u]\rangle_{e_h} + \frac{\sigma_h}{|e_h|^3} \langle [u], [v]\rangle_{e_h} \right) \\ &\quad - \sum_{e_h\in\mathcal{E}_h} \left(\langle \Delta u, [\partial_n v]\rangle_{e_h} + \langle \Delta v, [\partial_n u]\rangle_{e_h} - \frac{\tau_h}{|e_h|} \langle [\partial_n u], [\partial_n v]\rangle_{e_h} \right). \end{aligned} \tag{3.5}$$

In addition, define the functional F_h by

$$F_h(v) = (f,v) + \sum_{e_h\in\mathcal{E}_h^B} \left(\langle g_D, \partial_n \Delta v\rangle_{e_h} - \langle g_N, \Delta v\rangle_{e_h} + \frac{\sigma_h}{|e_h|^3} \langle g_D, v\rangle_{e_h} + \frac{\tau_h}{|e_h|} \langle g_N, \partial_n v\rangle_{e_h} \right). \tag{3.6}$$

The bilinear form a_h is consistent with the BVP (3.1)-(3.3). Indeed, suppose $u \in H^s(\mathcal{T}_h), s > 7/2$ is a solution. Then integration by parts reveals that

$$a_h(u,v) = F_h(v), \quad \forall v \in E_h. \tag{3.7}$$

Using this as motivation, we introduce the DG finite element spaces

$$V^h = \prod_{K\in\mathcal{T}_h} \mathcal{P}_q(K), \quad q \geq 2,$$

where $\mathcal{P}_q(K)$ is the space of polynomials of degree less than or equal to q on K. If necessary, the notation V_q^h will be used to indicate the degree of $\mathcal{P}_q$ explicitly. Define the DG approximation of u to be the element $u_h \in V^h$ that satisfies

$$a_h(u_h, v) = (f,v), \quad \forall v \in V^h. \tag{3.8}$$

Note that the form a_h is symmetric by construction. Introducing the norm $\|\cdot\|_{2,h} : E_h \to \mathbb{R}$ defined by

$$\begin{aligned}\|v\|_{2,h} \;=\; \Bigg\{ \sum_{K\in\mathcal{T}_h} \|\Delta v\|_K^2 + \sum_{e_h\in\mathcal{E}_h} \Bigg(& |e_h|\,|\{\Delta v\}|_{e_h}^2 + |e_h|^3\,|\{\partial_n \Delta v\}|_{e_h}^2 \\ & + \frac{\tau_h}{|e_h|}\,|[\partial_n v]|_{e_h}^2 + \frac{\sigma_h}{|e_h|^3}\,|[v]|_{e_h}^2 \Bigg) \Bigg\}^{1/2},\end{aligned} \tag{3.9}$$

we have the following continuity and coercivity properties of a_h.

Lemma 3.1.1. *For both the Arnold and Baker formulations,*

$$|a_h(u,v)| \;\le\; \|u\|_{2,h}\|v\|_{2,h}, \quad \forall u, v \in E^h \tag{3.10}$$

and there exist positive constants σ, τ and c_a depending only on q and the shape regularity of the cells in $\mathcal{T}_h$ such that if $\sigma_h \ge \sigma_0$ and $\tau_h \ge \tau_0$, then

$$|a_h(v,v)| \;\ge\; c_a\|v\|_{2,h}^2, \quad \forall v \in V^h. \tag{3.11}$$

These properties are sufficient to guarantee that the problem (3.8) is well-posed. The proofs are well-known and are therefore omitted.

The linear system corresponding to (3.8), again formed by choosing a basis for V^h, is

$$A\,\mathbf{x} = \mathbf{b}. \tag{3.12}$$

The stiffness matrix A is symmetric, positive definite; unfortunately, the (2-norm) condition number of A is of the order $O(\underline{h}^{-4})$ where $\underline{h} = \min_{K\in\mathcal{T}_h} h_K$. Thus, for small h, (3.12) is even more ill-conditioned than the second order system (2.13). The approach to solving this system is again to contruct the preconditioned system

$$BA\,\mathbf{x} = B\mathbf{b} \tag{3.13}$$

using the Schwarz framework, then solve this system iteratively with the Conjugate Gradient method.

3.2 Basic results

A few results from Chapter 2 will be needed in order to construct the Schwarz preconditioners for the biharmonic problem. As before, D will denote a simply-connected, open, bounded domain in $\mathbb{R}^d$ with Lipschitz boundary ∂D that is star-shaped and shape regular in the sense of (2.15). In addition, frequent use will be made of the trace and inverse inequalities on D ((2.16) and (2.17), respectively), as well as the basic approximation result (2.18). The broken trace lemma (Lemma (2.1.1)) and generalized Poincaré inequality (Lemma (2.1.3)) will also be necessary.

The following result of [26] will play a crucial role in the analysis. It will enable second order partial derivatives of $u \in V^h$ to be bound in terms of the norm $\|u\|_{2,h}$. Indeed, one of the difficulties of fourth order problems is that $\|\Delta u\|$ is not a seminorm, so it cannot play a role similar to that of $\|\nabla u\|$ in the context of second order problems.

Lemma 3.2.1. *There exists an operator $E : V^h \to V^h_{q+2} \cap H^2_0(\Omega)$ satisfying the bound*

$$|v_h - E(v_h)|^2_{m,K} \leq c \sum_{e_h \in \omega_h(K)} \left(|e_h|^{1-2m} |[v_h]|^2_{e_h} + |e_h|^{3-2m} |[\partial_n v_h]|^2_{e_h} \right), \quad \forall K \in \mathcal{T}_h, \tag{3.14}$$

where $m = 0, 1, 2$ and where $\omega_h(K)$ is the (local) set of edges in $\mathcal{T}_h$ emanating from the vertices of K.

In particular, this is the local version of the result given in [26]. In [26] the derivative jumps are given as $[\nabla u]$. These can be replaced by $[\partial_n u]$ upon noticing that one has $|[\partial_\tau u]|_{e_h} = |\partial_\tau [u]|_{e_h}$ for tangential derivatives along planar edges. Using the inverse inequality on e_h implies $|e_h|^{3-2m}|[\partial_\tau u]^2_{e_h} \leq c|e_h|^{1-2m}|[u]|^2_{e_h}$, a term that is already included in the right side of (3.14). The construction of $E(u)$ is done using a family of 2D macro elements introduced in [21].

Furthermore, the following estimates will be used frequently in the subsequent analysis.

Lemma 3.2.2. *For all $u \in V^h$, the following bounds hold:*

$$\|E(u)\|_{2,\Omega} \leq c(\Omega)\|E(u)\|_{\Omega} \tag{3.15}$$

$$\|E(u)\|_{2,\Omega}^2 \leq c \sum_{K\in\mathcal{T}_h} \|\Delta u\|_K^2 + c \sum_{e_h\in\mathcal{E}_h} \left(|e_h|^{-3} |[u]|_{e_h}^2 + |e_h|^{-1} |[\partial_n u]|_{e_h}^2 \right) \leq c\|u\|_{2,h}^2 \tag{3.16}$$

$$\sum_{K\in\mathcal{T}_h} |u|_{2,K}^2 \leq c \sum_{K\in\mathcal{T}_h} \|\Delta u\|_K^2 + c \sum_{e_h\in\mathcal{E}_h} \left(|e_h|^{-3} |[u]|_{e_h}^2 + |e_h|^{-1} |[\partial_n u]|_{e_h}^2 \right) \leq c\|u\|_{2,h}^2. \tag{3.17}$$

Proof. To begin, (3.15) is a well-known elliptic a priori bound (cf. e.g. [27]). Using it, the triangle inequality and (3.14) with $m = 2$,

$$\begin{aligned}
\|E(u)\|_{2,\Omega}^2 &\leq \|\Delta E(u)\|_{\Omega}^2 \\
&\leq c \sum_{K\in\mathcal{T}_h} \|\Delta u\|_{\Omega}^2 + c \sum_{K\in\mathcal{T}_h} \|\Delta(E(u) - u)\|_{\Omega}^2 \\
&\leq c \sum_{K\in\mathcal{T}_h} \|\Delta u\|_{\Omega}^2 + c \sum_{K\in\mathcal{T}_h} \sum_{e_h\in\omega_h(K)} \left(|e_h|^{-3} |[u]|_{e_h}^2 + |e_h|^{-1} |[\partial_n u]|_{e_h}^2 \right) \\
&\leq c \sum_{K\in\mathcal{T}_h} \|\Delta u\|_{\Omega}^2 + c \sum_{e_h\in\mathcal{E}_h} \left(|e_h|^{-3} |[u]|_{e_h}^2 + |e_h|^{-1} |[\partial_n u]|_{e_h}^2 \right) \\
&\leq c\|u\|_{2,h}^2.
\end{aligned} \tag{3.18}$$

This proves inequality (3.16). The third inequality (3.17) follows by the triangle inequality, (3.14) with $m = 2$, and (3.16) □

3.3 Schwarz Framework

The construction of the nonoverlapping and overlapping additive Schwarz preconditioners for the fourth order problem mirrors the development presented in Chapter 2. Much of the notation introduced previously will be used with identical definitions. In particular, the domain Ω is divided into a collection of subdomains $\{\Omega_j\}_{j=1}^p$ with corresponding partition $\mathcal{T}_S$, while $\mathcal{T}_H$ is a coarse partition of Ω.

As before, the following alignments will be required between the partitions $\mathcal{T}_h, \mathcal{T}_H$, and $\mathcal{T}_S$:

$$\mathcal{T}_S \subseteq \mathcal{T}_h \quad \text{and} \quad \mathcal{T}_H \subseteq \mathcal{T}_h. \tag{3.19}$$

3.3.1 Subdomain Spaces and Bilinear Forms

In order to define the subdomain bilinear forms for the current problem, it will be useful to recall a few definitions presented in Chapter 2. Define the subspaces $\{V_j^h\}_{j=1}^p$ associated with the subdomains $\{\Omega_j\}_{j=1}^p$ by

$$V_j^h = \{v \in V^h | v = 0 \quad \text{in} \quad \Omega \setminus \overline{\Omega}_j\}, \quad j = 1, 2, \cdots, p,$$

and the associated bilinear forms $a_i(\cdot, \cdot)$ by restriction:

$$a_i(u, v) \; = \; a_h(u, v), \qquad \forall u, v \in V_i^h, \; i = 1, \ldots, p.$$

As before, the decomposition

$$V^h \; = \; V_1^h \; + \; \cdots \; + \; V_0^h \tag{3.20}$$

holds. In addition, we will consider the following sets:

1. $\mathcal{S}_i := \{e_h \in \mathcal{E}_h^I, e_h \in \partial\Omega_i\}$, $i = 1, \ldots, p$.
2. $\mathcal{T}_{h,i} := \{K \in \mathcal{T}_h, K \subset \Omega_i\}$.

3. $\mathcal{E}_{h,i}^I := \{e_h \in \mathcal{E}_h^I, e_h \subset \Omega_i\}$.

4. $\mathcal{E}_{h,i}^B := \{e_h \in \mathcal{E}_h^B, e_h \in \partial\Omega_i\}$. Note that this set may be empty.

5. $\mathcal{E}_{h,i} := \mathcal{E}_{h,i}^I \cup \mathcal{E}_{h,i}^B$.

For Arnold's formulation, the subdomain bilinear forms are give by

$$\begin{aligned} a_i(u,v) \;=\; & \sum_{K\in\mathcal{T}_{h,i}} (\Delta u, \Delta v)_K \\ & + \sum_{e_h\in\mathcal{E}_{h,i}} \left(\langle\{\partial_n\Delta u\},[v]\rangle_{e_h} + \langle\{\partial_n\Delta v\},[u]\rangle_{e_h} + \frac{\sigma_h}{|e_h|^3}\langle[u],[v]\rangle_{e_h} \right) \\ & - \sum_{e_h\in\mathcal{E}_{h,i}} \left(\langle\{\Delta u\},[\partial_n v]\rangle_{e_h} + \langle\{\Delta v\},[\partial_n u]\rangle_{e_h} - \frac{\tau_h}{|e_h|}\langle[\partial_n u],[\partial_n v]\rangle_{e_h} \right) \\ & + \sum_{e_h\in\mathcal{S}_i} \left(\frac{1}{2}\langle\partial_n\Delta u, v\rangle_{e_h} + \frac{1}{2}\langle\partial_n\Delta v, u\rangle_{e_h} + \frac{\sigma_h}{|e_h|^3}\langle u,v\rangle_{e_h} \right) \\ & - \sum_{e_h\in\mathcal{S}_i} \left(\frac{1}{2}\langle\Delta u, \partial_n v\rangle_{e_h} + \frac{1}{2}\langle\Delta v, \partial_n u\rangle_{e_h} - \frac{\tau_h}{|e_h|}\langle\partial_n u,\partial_n v\rangle_{e_h} \right). \end{aligned} \tag{3.21}$$

In the above expression, the edge integrals on $\mathcal{S}_i$ must be interpreted as follows: The traces of u, v are taken from Ω_i and the normal derivatives are with respect to the unit outward normal vectors to Ω_i.

For Baker's formulation, the bilinear form $a_i(\cdot,\cdot)$ takes the form

$$\begin{aligned} a_i(u,v) \;=\; & \sum_{K\in\mathcal{T}_{h,i}} (\Delta u, \Delta v)_K \\ & + \sum_{e_h\in\mathcal{E}_{h,i}} \left(\langle\partial_n\Delta u,[v]\rangle_{e_h} + \langle\partial_n\Delta v,[u]\rangle_{e_h} + \frac{\sigma_h}{|e_h|^3}\langle[u],[v]\rangle_{e_h} \right) \\ & - \sum_{e_h\in\mathcal{E}_{h,i}} \left(\langle\Delta u,[\partial_n v]\rangle_{e_h} + \langle\Delta v,[\partial_n u]\rangle_{e_h} - \frac{\tau_h}{|e_h|}\langle[\partial_n u],[\partial_n v]\rangle_{e_h} \right) \\ & + \sum_{e_h\in\mathcal{S}_i} \left(\langle\partial_n\Delta u, v\rangle_{e_h} + \langle\partial_n\Delta v, u\rangle_{e_h} + \frac{\sigma_h}{|e_h|^3}\langle u,v\rangle_{e_h} \right) \\ & - \sum_{e_h\in\mathcal{S}_i} \left(\langle\Delta u, \partial_n v\rangle_{e_h} + \langle\Delta v, \partial_n u\rangle_{e_h} - \frac{\tau_h}{|e_h|}\langle\partial_n u,\partial_n v\rangle_{e_h} \right). \end{aligned} \tag{3.22}$$

Again, for $e_h \in \mathcal{S}_i$, the terms $\langle \partial_n u, v \rangle_{e_h}$, $\langle \partial_n v, u \rangle_{e_h}$ will be present if and only if K^+ belongs to Ω_i.

3.3.2 The Coarse Subspace and Bilinear Form

As in the second order problem, two choices will be investigated for the coarse mesh subspace V_0^h. The consideration of the continuous course space $V^{h,C}$ will be quite similar in this case, but the presence of the additional penalty terms in a_h will require a bit of extra attention when examining the discontinuous course space $V^{h,D}$.

3.3.3 The Discontinuous Coarse Space

Here we set

$$V_0^h := V^{H,D} = \mathcal{P}_{q_0}(\mathcal{T}_H), \quad 2 \leq q_0 \leq q. \tag{3.23}$$

Clearly V_0^h is a subspace of V^h. The coarse space bilinear form is defined by

$$\begin{aligned} a_0(u,v) \;=\; & \sum_{K \in \mathcal{T}_H} (\Delta u, \Delta v)_K \\ & + \sum_{e_H \in \mathcal{E}_H} \left(\langle \{\partial_n \Delta u\}, [v] \rangle_{e_H} + \langle \{\partial_n \Delta v\}, [u] \rangle_{e_H} + \frac{\sigma_H}{|e_H|^3} \langle [u], [v] \rangle_{e_H} \right) \\ & - \sum_{e_H \in \mathcal{E}_H} \left(\langle \{\Delta u\}, [\partial_n v] \rangle_{e_H} + \langle \{\Delta v\}, [\partial_n u] \rangle_{e_H} + \frac{\tau_H}{|e_H|} \langle [\partial_n u], [\partial_n v] \rangle_{e_H} \right). \end{aligned} \tag{3.24}$$

In [24], $a_0(\cdot,\cdot)$ was defined as the restriction of $a_h(\cdot,\cdot)$ to $V^{H,D}$, i.e.

$$a_0(u,v) = a_h(u,v), \quad \forall u, v \in V^{H,D}. \tag{3.25}$$

This definition requires the imposition of certain compatibility conditions relating the penalty paramaters σ_h to σ_H and τ_h to τ_H, in a similar fashion to the observations made in Chapter 2. For Baker's formulation, additonal conditions on the choice of K^+ and K^- on interior edges must also be enforced. Again, all the conditions that

are necessary for (3.25) to hold need to be identified. Based on such a study a rational choice as to whether to define $a_0(\cdot,\cdot)$ by (3.25) or not can be made.

To begin, it easy to see that

$$\sum_{K\in\mathcal{T}_h}(\Delta u,\Delta v)_K=\sum_{D\in\mathcal{T}_H}(\Delta u,\Delta v)_D,\quad \forall u,v\in V^{H,D}. \tag{3.26}$$

For $u,v\in V^{H,D}$ the jumps across edges $e_h\in\mathcal{E}_h^I$ which are in the interior of some $K\in\mathcal{T}_H$ are zero. Also, for Arnold's formulation the values of $\langle\{\partial_n\Delta u\},[v]\rangle_{e_H}$, $\langle\{\partial_n\Delta v\},[u]\rangle_{e_H}$, $\langle\{\Delta u\},[\partial_n v]\rangle_{e_H}$, and $\langle\{\Delta v\},[\partial_n u]\rangle_{e_H}$ are independent of the sign convention used in the designation of K^+ vs. K^-. For $e_h\in\mathcal{E}_h^B$, there is no issue since we always use K^+ for the cell that contains it. Hence, we can combine these edge integrals to obtain

$$\begin{aligned}&\sum_{e_h\in\mathcal{E}_h}\left(\langle\{\partial_n\Delta u\},[v]\rangle_{e_h}+\langle\{\partial_n\Delta v\},[u]\rangle_{e_h}-\langle\{\Delta u\},[\partial_n v]\rangle_{e_h}-\langle\{\Delta v\},[\partial_n u]\rangle_{e_h}\right)=\\&\sum_{e_H\in\mathcal{E}_H}\left(\langle\{\partial_n\Delta u\},[v]\rangle_{e_H}+\langle\{\partial_n\Delta v\},[u]\rangle_{e_H}-\langle\{\Delta u\},[\partial_n v]\rangle_{e_H}-\langle\{\Delta v\},[\partial_n u]\rangle_{e_H}\right),\end{aligned} \tag{3.27}$$

for all $u,v\in V^{H,D}$.

For Baker's formulation, (3.27) does not necessarily hold (refer to Figure 2.1 in Chapter 2). It will again be necessary to enforce the same sign compatibility assumption (SC) as in the second order case (Figure 2.2):

> For all edges $e_h\in\mathcal{E}_h^I$ that are part of an edge $e_H\in\mathcal{E}_H^I$, $e_H\subset\partial D$, $D\in\mathcal{T}_H$, the cells $K\in\mathcal{T}_h$ that contain e_h and belong to D have the same sign as D in relation to e_H. (SC)

Under this assumption, the fourth order version of (3.27) also holds for Baker's formulation.

The focus now turns to the penalty jump terms. Note that these terms are identical for both the Arnold and Baker formulations and are independent of the

$K^{+,-}$ convention. Furthermore, for $u, v \in V^{H,D}$, the jumps across edges $e_h \in \mathcal{E}_h^I$ which are in the interior of some $K \in \mathcal{T}_H$ are zero. On the other hand, the attached weights must still be considered. Using (3.26) and (3.27) and assuming that (SC) holds for Baker's formulation,

$$\begin{aligned} a_h(u,v) \;=\; a_0(u,v) \;&+\; \sum_{e_H \in \mathcal{E}_H} \sum_{e_h \subset e_H} \left(\frac{\sigma_h}{|e_h|^3} - \frac{\sigma_H}{|e_H|^3} \right) \langle [u], [v] \rangle_{e_h} \\ &+\; \sum_{e_H \in \mathcal{E}_H} \sum_{e_h \subset e_H} \left(\frac{\tau_h}{|e_h|} - \frac{\tau_H}{|e_H|} \right) \langle [\partial_n u], [\partial_n v] \rangle_{e_h} . \end{aligned} \tag{3.28}$$

This motivates introducing the following penalty compatibility conditions:

$$\frac{\sigma_h}{|e_h|^3} = \frac{\sigma_H}{|e_H|^3}, \qquad e_h \subset e_H \in \mathcal{E}_H, \tag{PC$_\sigma$}$$

$$\frac{\tau_h}{|e_h|} = \frac{\tau_H}{|e_H|}, \qquad e_h \subset e_H \in \mathcal{E}_H. \tag{PC$_\tau$}$$

Putting this together,

Lemma 3.3.1.

(i) *Under assumptions* (PC$_\sigma$) *and* (PC$_\tau$), (3.25) *holds for Arnold's formulation.*

(ii) *Under assumptions* (PC$_\sigma$), (PC$_\tau$), *and* (SC), (3.25) *holds for Baker's formulation.*

(iii) *Under assumptions* (PC$_\sigma$) *and* (PC$_\tau$) *restricted to* $e_H \in \mathcal{E}_H^B$ *we have* $F_h(v) = F_H(v)$, $\forall v \in V^{H,D}$.

As pointed out in Chapter 2, the enforcement of these conditions is trivial in practical implementation.

3.3.4 The Continuous Coarse Space

As before, set

$$V_0^h = V^{H,C} = \{v \in \mathcal{P}_{q_0}(\mathcal{T}_H) \cap C(\Omega),\, v\big|_{\partial\Omega} = 0\} \subset V^{H,D}. \tag{3.29}$$

Since we are assuming that (SC) holds for Baker's formulation, it follows from (3.28) that

$$a_h(u,v) = a_0(u,v) + \sum_{e_H \in \mathcal{E}_H} \sum_{e_h \subset e_H} \left(\frac{\tau_h}{|e_h|} - \frac{\tau_H}{|e_H|} \right) \langle [\partial_n u], [\partial_n [] v] \rangle_{e_h}, \quad \forall u, v \in V^{H,C}. \tag{3.30}$$

As mentioned previously, this option has the advantage of a smaller stiffness matrix corresponding to the coarse level, as well as eliminating the need to enforce the penalty compatibility conditions.

3.4 Construction of the Schwarz Preconditioners

The construction of the two-level additive Schwarz preconditioners for the fourth order problem will follow the same abstract framework detailed in Section 2.4. In particular, the current analysis will focus on verifying the following three assumptions (recall from Section 2.4):

Assumption 1: For any $u \in V^h$

$$\sum_{i=0}^{p} a_i(u_i, u_i) \leq C_0^2 a_h(u,u), \tag{3.31}$$

for *some* representation $u = \sum_{i=0}^{p} u_i$. Here, $1/C_0^2$ is a lower bound on the smallest eigenvalue of the preconditioned matrix.

Assumption 2: Let $0 \leq \mathcal{E}_{ij} \leq 1$ be the minimal values such that

$$|a_h(u_i, u_j)| \leq \mathcal{E}_{ij}\, a_h(u_i,u_i)^{\frac{1}{2}} a_h(u_j,u_j)^{\frac{1}{2}}, \quad u_i \in V_i^h,\, u_j \in V_j^h,\; i,j = 1, \cdots, p. \tag{3.32}$$

That such values exist follows from the Cauchy-Schwarz inequality. Define $\rho(\mathcal{E})$ to be the spectral radius of $\mathcal{E}$.

Assumption 3: Let $\omega \geq 1$ be the smallest value such that

$$a_h(u,u) \leq \omega\, a_i(u,u), \quad \forall u \in V_i^h,\ i = 0,1,\ldots,p. \tag{3.33}$$

Special attention will be given to the constants $C_0^2, \rho(\mathcal{E})$ and ω appearing in these assumptions, due to their role in determining the bounds for the condition number κ of the preconditioned system (3.13) in both the nonoverlapping and the overlapping case. The condition number estimates will again be derived using the abstract estimate (2.55).

3.4.1 Nonoverlapping Additive Schwarz Preconditioner

Recall that in this case the subdomains Ω_i are disjoint and the sum (2.38) is direct. The alignments (2.37) and $\mathcal{T}_S \subseteq \mathcal{T}_H$ are again required. The interface bilinear form $\mathcal{I}(\cdot,\cdot) : V^h \times V^h \to \mathbb{R}$ for the fourth order problem is again vital to the analysis. For Arnold's formulation, it is given by

$$\begin{aligned}
\mathcal{I}(u,v) \;=\; & -\frac{1}{2}\sum_{e_h \in \mathcal{S}} \Big(\big\langle \partial_n \Delta u^+, v^- \big\rangle_{e_h} + \big\langle \partial_n \Delta v^+, u^- \big\rangle_{e_h} \\
& \qquad - \big\langle \partial_n \Delta u^-, v^+ \big\rangle_{e_h} - \big\langle \partial_n \Delta v^-, u^+ \big\rangle_{e_h} \Big) \\
& +\frac{1}{2}\sum_{e_h \in \mathcal{S}} \Big(\big\langle \Delta u^+, \partial_n v^- \big\rangle_{e_h} + \big\langle \Delta v^+, \partial_n u^- \big\rangle_{e_h} \\
& \qquad - \big\langle \Delta u^-, \partial_n v^+ \big\rangle_{e_h} - \big\langle \Delta v^-, \partial_n u^+ \big\rangle_{e_h} \Big) \\
& -\sum_{e_h \in \mathcal{S}} \frac{\sigma_h}{|e_h|^3} \Big(\big\langle u^+, v^- \big\rangle_{e_h} + \big\langle u^-, v^+ \big\rangle_{e_h} \Big) \\
& -\sum_{e_h \in \mathcal{S}} \frac{\tau_h}{|e_h|} \Big(\big\langle \partial_n u^+, \partial_n v^- \big\rangle_{e_h} + \big\langle \partial_n u^-, \partial_n v^+ \big\rangle_{e_h} \Big).
\end{aligned} \tag{3.34}$$

Recall that $\mathcal{S} := \cup_{i=1}^{p} \mathcal{S}_i$ is sometimes called the *skeleton* of the nonoverlapping partition.

For Baker's formulation, the interface form is given by

$$\begin{aligned}
\mathcal{I}(u,v) \;=\; & -\sum_{e_h \in \mathcal{S}} \Big(\big\langle \partial_n \Delta u^+, v^- \big\rangle_{e_h} + \big\langle \partial_n \Delta v^+, u^- \big\rangle_{e_h} \Big) \\
& + \sum_{e_h \in \mathcal{S}} \Big(\big\langle \Delta u^+, \partial_n v^- \big\rangle_{e_h} + \big\langle \Delta v^+, \partial_n u^- \big\rangle_{e_h} \Big) \\
& - \sum_{e_h \in \mathcal{S}} \frac{\sigma_h}{|e_h|^3} \Big(\big\langle u^+, v^- \big\rangle_{e_h} + \big\langle u^-, v^+ \big\rangle_{e_h} \Big) \\
& - \sum_{e_h \in \mathcal{S}} \frac{\tau_h}{|e_h|} \Big(\big\langle \partial_n u^+, \partial_n v^- \big\rangle_{e_h} + \big\langle \partial_n u^-, \partial_n v^+ \big\rangle_{e_h} \Big).
\end{aligned} \tag{3.35}$$

For both Arnold's and Baker's formulations, the following identity holds

$$a_h(u,v) \;=\; \sum_{i=1}^{p} a_i(u_i, v_i) + \mathcal{I}(u,v), \quad \forall u, v \in V^h, \tag{3.36}$$

where

$$u \;=\; \sum_{i=1}^{p} u_i, \quad v \;=\; \sum_{i=1}^{p} v_i, \quad \text{and} \quad u_i, v_i \in V_i^h.$$

The first step in the analysis of the nonoverlapping method is to obtain a bound for $|\mathcal{I}(w,w)|$, for $w \in V^h$. In order to simplify the exposition, assume that the partition of each cell $D \in \mathcal{T}_H$ into cells from $\mathcal{T}_h$ is uniform; i.e., that there exists a representative h_D such that

$$h_D \approx \min_{K \in \mathcal{T}_h,\, K \subset D} h_K \quad \approx \max_{K \in \mathcal{T}_h,\, K \subset D} h_K. \tag{3.37}$$

The bound for $|\mathcal{I}(w,w)|$ is shown below.

Lemma 3.4.1. *There exists a constant c independent of the number of subdomains, such that*

$$|\mathcal{I}(w,w)| \;\leq\; c\sum_{K\in\mathcal{T}_h}\|\Delta w\|_K^2 \;+\; c(\overline{\sigma}+\overline{\tau})\sum_{D\in\mathcal{T}_H} h_D^{-3}\Big(H_D^{-1}\|w\|_D^2 + H_D|w|_{1,h,D}^2,\Big) \tag{3.38}$$

where

$$H_D = \operatorname{diam}(D), \quad \overline{\sigma} = \max_{e_h\subset\mathcal{E}_h}\sigma_h, \quad \overline{\tau} = \max_{e_h\subset\mathcal{E}_h}\tau_h.$$

Proof. The proof will be given for Arnold's formulation; Baker's is similar. First, note that

$$\begin{aligned}\mathcal{I}(w,w) \;=\; &-\sum_{e_h\in\mathcal{S}}\Big(\big\langle\partial_n\Delta w^+, w^-\big\rangle_{e_h} - \big\langle\partial_n\Delta w^-, w^+\big\rangle_{e_h} + \big\langle\Delta w^+, \partial_n w^-\big\rangle_{e_h} \\ &\qquad - \big\langle\Delta w^-, \partial_n w^+\big\rangle_{e_h}\Big) \\ &-2\sum_{e_h\in\mathcal{S}}\Big(\frac{\sigma_h}{|e_h|^3}\big\langle w^+, w^-\big\rangle_{e_h} + \frac{\tau_h}{|e_h|}\big\langle\partial_n w^+, \partial_n w^-\big\rangle_{e_h}\Big).\end{aligned} \tag{3.39}$$

Using the arithmetic-geometric mean inequality (and making the reasonable assumption that $\sigma_h, \tau_h \geq 1/2$),

$$\begin{aligned}|\mathcal{I}(w,w)| \;\leq\; &\frac{1}{2}\sum_{e_h\in\mathcal{S}}\Big(|e_h|^3\big(\,\big|\partial_n\Delta w^+\big|_{e_h}^2 + \big|\partial_n\Delta w^-\big|_{e_h}^2\,\big) + |e_h|\big(\,\big|\Delta w^+\big|_{e_h}^2 + \big|\Delta w^-\big|_{e_h}^2\,\big)\Big) \\ &+2\sum_{e_h\in\mathcal{S}}\frac{\sigma_h}{|e_h|^3}\Big(\,\big|w^+\big|_{e_h}^2 + \big|w^-\big|_{e_h}^2\,\Big) \\ &+2\sum_{e_h\in\mathcal{S}}\frac{\tau_h}{|e_h|}\Big(\,\big|\partial_n w^+\big|_{e_h}^2 + \big|\partial_n w^-\big|_{e_h}^2\,\Big) \\ =\; &A_1 + A_2 + A_3.\end{aligned} \tag{3.40}$$

Using the trace and inverse inequalities (2.16), (2.17),

$$|A_1| \;\leq\; c\sum_{K\in\mathcal{T}_h}\|\Delta w\|_K^2. \tag{3.41}$$

Now, note that each e_h belongs to the boundary ∂D of some cell in the coarse mesh $\mathcal{T}_H$. Appealing to Lemma (2.1.1),

$$\begin{aligned} |A_2| &\leq c\overline{\sigma} \sum_{D\in\mathcal{T}_H} h_D^{-3}|w|^2_{\partial D} \\ &\leq c\overline{\sigma} \sum_{D\in\mathcal{T}_H} h_D^{-3}\Big(H_D^{-1}\|w\|^2_D H_D|w|^2_{1,h,D}\Big). \end{aligned} \tag{3.42}$$

Finally, using the trace and inverse inequalities,

$$\begin{aligned} |A_3| &\leq c\sum_{e_h\in\mathcal{S}} \tau_h|e_h|^{-4}\big(\|w\|^2_{K^+} + \|w\|^2_{K^-}\big) \\ &\leq c\overline{\tau} \sum_{D\in\mathcal{T}_H} h_D^{-4}\|w\|^2_{D_h}, \end{aligned} \tag{3.43}$$

where D_h is a boundary layer of width h_D. Applying the generalized Poincaré inequality (2.1.3) with $\rho = h_D$,

$$|A_3| \leq c\overline{\tau} \sum_{D\in\mathcal{T}_H} h_D^{-3}\Big(H_D^{-1}\|w\|^2_D + H_D|w|^2_{1,h,D}\Big). \tag{3.44}$$

Combining (3.40) - (3.44) proves the estimate (3.38). □

With this lemma in place, the next step is the verification of **Assumption 1**.

Proposition 3.4.1. *For any $u \in V^h$, there exists a decomposition $u = \sum_{j=0}^{p} u_j$, $u_j \in V_j^h$, $j = 0, 1, \ldots, p$ such that*

$$\sum_{j=0}^{p} a_i(u_j, u_j) \leq c(\overline{\sigma} + \overline{\tau} + \overline{\tau}\,\overline{\gamma})\,[H : h]^3\, a_h(u, u), \tag{3.45}$$

where the constant c is independent of p, $\overline{\sigma}$ and $\overline{\tau}$ have been introduced earlier, $\overline{\gamma} = \max\left(\frac{\tau_H}{|e_H|}\frac{|e_h|}{\tau_h}\right)$ and $[H : h] = \max_{D\in\mathcal{T}_H} \frac{H_D}{h_D}$ is a measure of the fineness of $\mathcal{T}_h$ with respect to $\mathcal{T}_H$.

Proof. An important component of the proof consists in constructing an appropriate u_0 in the coarse space V_0^h. For $u \in V_h$, let $E(u)$ be the element in $V_{q+2}^h \cap H_0^2(\Omega)$ constructed in Lemma 3.2.1 and let $u_0 = I_H(E(u))$ be the continuous piecewise linear Lagrange interpolant of $E(u)$ on the coarse mesh $\mathcal{T}_H$. Thus u_0 belongs to the continuous coarse space $V^{H,C}$. Furthermore, the following approximation holds:

$$\|E(u) - u_0\|_D + H_D\|\nabla(E(u) - u_0)\|_D \leq cH_D^2|E(u)|_{2,D}, \quad \forall D \in \mathcal{T}_H. \tag{3.46}$$

The decomposition (3.20) indicates the existence of uniquely determined functions $u_i \in V_i^h,\ i = 1, \ldots, p$ such that $u - u_0 = u_1 + \cdots + u_p$. In view of (3.36),

$$a_h(u - u_0, u - u_0) = \sum_{j=1}^{p} a_j(u_j, u_j) + \mathcal{I}(u - u_0, u - u_0).$$

Adding $a_0(u_0, u_0)$ to both sides and applying the Cauchy-Schwarz inequality to the bilinear form $a_h(\cdot, \cdot)$,

$$\sum_{j=0}^{p} a_j(u_j, u_j) \leq 2a_h(u, u) + 2a_h(u_0, u_0) + a_0(u_0, u_0) + |\mathcal{I}(u - u_0, u - u_0)|. \tag{3.47}$$

The next parts of the proof consist of bounding terms on the right side of (3.47). The most difficult term to estimate will be $|\mathcal{I}(u - u_0, u - u_0)|$. From Lemma 3.4.1,

$$\begin{aligned} |\mathcal{I}(u - u_0, u - u_0)| \leq & \sum_{K \in \mathcal{T}_h} \|\Delta(u - u_0)\|_K^2 \\ & + c\gamma \sum_{D \in \mathcal{T}_H} h_D^{-3}\Big(H_D^{-1}\|u - u_0\|_D^2 + H_D|u - u_0|_{1,h,D}^2\Big). \end{aligned} \tag{3.48}$$

For the first sum in (3.48), since u_0 is piecewise linear it follows that

$$\sum_{K \in \mathcal{T}_h} \|\Delta(u - u_0)\|_K^2 = \sum_{K \in \mathcal{T}_h} \|\Delta u\|_K^2. \tag{3.49}$$

The second sum in (3.48) is broken apart by adding and subtracting $E(u)$ and applying the triangle inequality to obtain

$$\begin{aligned}\sum_{D\in\mathcal{T}_H} h_D^{-3}H_D^{-1}\|u-u_0\|_D^2 \;&\leq\; 2\sum_{D\in\mathcal{T}_H} h_D^{-3}H_D^{-1}\|u-E(u)\|_D^2 \\ &\quad + 2\sum_{D\in\mathcal{T}_H} h_D^{-3}H_D^{-1}\|E(u)-u_0\|_D^2.\end{aligned} \tag{3.50}$$

For the first of these sums, applying Lemma 3.2.1 with $m=0$ gives

$$\begin{aligned}\sum_{D\in\mathcal{T}_H} h_D^{-3}H_D^{-1}\|u-E(u)\|_D^2 \;&=\; \sum_{D\in\mathcal{T}_H} h_D^{-3}H_D^{-1}\sum_{K\subset D}\|u-E(u)\|_K^2 \\ &\leq\; c\sum_{D\in\mathcal{T}_H} h_D^{-3}H_D^{-1}\sum_{K\subset D}\sum_{e_h\in\omega(K)}\left(|e_h|\,|[u]|_{e_h}^2+|e_h|^3\,|[\partial_n u]|_{e_h}^2\right) \\ &\leq\; c\sum_{K\in\mathcal{T}_h}\left(|e_h|^{-3}\,|[u]|_{e_h}^2+|e_h|^{-1}\,|[\partial_n u]|_{e_h}^2\right) \\ &\leq\; c\|u\|_{2,h}^2.\end{aligned} \tag{3.51}$$

Note the use of the fact that $h_D\leq H_D$ and $|e_h|\approx h_D$ for $e_h\in\omega(K), K\subset D$. For the second sum in (3.50), using (3.46) and (3.16)

$$\begin{aligned}\sum_{D\in\mathcal{T}_H} h_D^{-3}H_D^{-1}\|E(u)-u_0\|_D^2 \;&\leq\; c\sum_{D\in\mathcal{T}_H} h_D^{-3}H_D^{3}|E(u)|_{2,D}^2 \\ &\leq\; c[H:h]^3|E(u)|_{2,\Omega}^2 \\ &\leq\; c[H:h]^3\|u\|_{2,h}^2.\end{aligned} \tag{3.52}$$

Combining the previous two estimates, it follows that the second sum in (3.48) may be bounded in the following way.

$$\sum_{D\in\mathcal{T}_H} h_D^{-3}H_D^{-1}\|u-u_0\| \;\leq\; c[H:h]^3\|u\|_{2,h}^2. \tag{3.53}$$

The final sum in (3.48) is estimated as follows. Using the definition of the $|\cdot|_{1,h,D}$ norm and the fact that u_0 is continuous gives

$$\begin{aligned}\sum_{D\in\mathcal{T}_H} h_D^{-3}H_D|u-u_0|^2_{1,h,D} &= \sum_{D\in\mathcal{T}_H} h_D^{-3}H_D \sum_{K\subset D}\|\nabla(u-u_0)\|^2_K \\ &\quad + \sum_{D\in\mathcal{T}_H} h_D^{-3}H_D \sum_{\mathcal{E}^I_{H,D}} |e_h|^{-1}\,|[u]|^2_{e_h}\,.\end{aligned} \tag{3.54}$$

Adding and subtracting $E(u)$ and applying the triangle inequality to the first sum on the right side of (3.54) gives the estimate

$$\begin{aligned}&\sum_{D\in\mathcal{T}_H} h_D^{-3}H_D \sum_{K\subset D}\|\nabla(u-u_0)\|^2_K \\ &\qquad \le 2\sum_{D\in\mathcal{T}_H} h_D^{-3}H_D \sum_{K\subset D}\Big(\|\nabla(u-E(u))\|^2_K + \|\nabla(E(u)-u_0)\|^2_K\Big)\end{aligned} \tag{3.55}$$

To bound the first sum in the above inequality, again use Lemma 3.2.1 with $m=1$ to get

$$\begin{aligned}&\sum_{D\in\mathcal{T}_H} h_D^{-3}H_D \sum_{K\subset D}\|\nabla(u-E(u))\|^2_K \\ &\qquad \le c\sum_{D\in\mathcal{T}_H} h_D^{-3}H_D \sum_{K\subset D}\sum_{e_h\in\omega(K)}\Big(|e_h|^{-1}\,|[u]|^2_{e_h} + |e_h|\,|[\partial_n u]|^2_{e_h}\Big) \\ &\qquad \le c[H:h]\sum_{K\in\mathcal{T}_h}\Big(|e_h|^{-3}\,|[u]|^2_{e_h} + |e_h|^{-1}\,|[\partial_n u]|^2_{e_h}\Big) \\ &\qquad \le c[H:h]\|u\|^2_{2,h}.\end{aligned} \tag{3.56}$$

Again note the use of the fact that $h_D \le H_D$ and $|e_h| \approx h_D$ for $e_h \in \omega(K), K \subset D$. An approach similar to (3.52) is used to estimate the second sum in (3.55). Using

(3.46) and (3.16),

$$\begin{aligned}\sum_{D\in\mathcal{T}_H} h_D^{-3}H_D\|\nabla(E(u)-u_0)\|_D^2 &\leq c\sum_{D\in\mathcal{T}_H} h_D^{-3}H_D^3|E(u)|_{2,D}^2\\ &\leq c[H:h]^3|E(u)|_{2,\Omega}^2\\ &\leq c[H:h]^3\|u\|_{2,h}^2.\end{aligned} \tag{3.57}$$

Since $[H:h]\leq[H:h]^3$, combining (3.56) and (3.57) gives the estimate

$$\sum_{D\in\mathcal{T}_H} h_D^{-3}H_D\sum_{K\subset D}\|\nabla(u-u_0)\|_K^2 \leq c[H:h]^3\|u\|_{2,h}^2. \tag{3.58}$$

For the second sum in (3.54), note that

$$\begin{aligned}\sum_{D\in\mathcal{T}_H} h_D^{-3}H_D\sum_{e_h\in\mathcal{E}_{H,D}^I}|e_h|^{-1}\,|[u]|_{e_h}^2 &\leq c[H:h]\sum_{e_h\in\mathcal{E}_h^I}|e_h|^{-3}\,|[u]|_{e_h}^2\\ &\leq c[H:h]\|u\|_{2,h}^2.\end{aligned} \tag{3.59}$$

Combining (3.58) and (3.59) with (3.54), the final term in (3.48) is estimated by

$$\sum_{D\in\mathcal{T}_H} h_D^{-3}H_D|u-u_0|_{1,h,D}^2 \leq c[H:h]^3\|u\|_{2,h}^2. \tag{3.60}$$

Finally, all the pieces of the puzzle needed to bound $\mathcal{I}|u-u_0,u-u_0|$ are in place. It follows from (3.49), (3.53), and (3.60) that

$$\begin{aligned}|\mathcal{I}(u-u_0,u-u_0)| &\leq c(\overline{\sigma}+\overline{\tau})[H:h]^3\|u\|_{2,h}^2\\ &\leq c(\overline{\sigma}+\overline{\tau})[H:h]^3a_h(u,u).\end{aligned} \tag{3.61}$$

Note the use of the coercivity ((3.11)) of $a_h(\cdot,\cdot)$ in the last step.

The second stage of the proof is deriving a bound for $a_h(u_0,u_0)$. Combining the knowledge that u_0 is continuous and piecewise linear on $\mathcal{T}_H$ and vanishes on $\partial\Omega$ with

the fact that $E(u) \in H_0^2(\Omega)$,

$$\begin{aligned} a_h(u_0, u_0) &= \sum_{e_h \in \mathcal{E}_h} \frac{\tau_h}{|e_h|} \left| [\partial_n u_0] \right|_{e_h}^2 \\ &= \sum_{e_h \in \mathcal{E}_h} \sum_{e_h \subset e_H} \frac{\tau_h}{|e_h|} \left| [\partial_n u_0] \right|_{e_h}^2 \\ &\leq \overline{\tau} \sum_{D \in \mathcal{T}_H} h_D^{-1} \left| [\partial_n (E(u) - u_0)] \right|_{\partial D}^2 . \end{aligned} \tag{3.62}$$

Now, applying the trace and inverse inequalities (2.16) and (2.17) on $\mathcal{T}_H$ gives

$$\begin{aligned} a_h(u_0, u_0) &\leq \overline{\tau} \sum_{D \in \mathcal{T}_H} h_D^{-1} \left| [\partial_n (E(u) - u_0)] \right|_{\partial D}^2 \\ &\leq \overline{\tau} \sum_{D \in \mathcal{T}_H} h_D^{-1} H_D^{-3} \| E(u) - u_0 \|_D^2 . \end{aligned} \tag{3.63}$$

Using the approximation and regularity results (3.46) and (3.16), as well as the coercivity of $a_h(\cdot, \cdot)$ given in (3.11),

$$\begin{aligned} a_h(u_0, u_0) &\leq \overline{\tau} \sum_{D \in \mathcal{T}_H} h_D^{-1} H_D^{-3} \| E(u) - u_0 \|_D^2 \\ &\leq c\overline{\tau} \sum_{D \in \mathcal{T}_H} h_D^{-1} H_D |E(u)|_{2,D}^2 \\ &\leq c\overline{\tau} [H : h] |E(u)|_{2,\Omega}^2 \\ &\leq c\overline{\tau} [H : h] \|u\|_{2,h}^2 \\ &\leq c\overline{\tau} [H : h] a_h(u, u). \end{aligned} \tag{3.64}$$

The final stage of the proof involves estimating $a_0(u_0, u_0)$. From (3.24) and (3.11),

$$\begin{aligned} a_0(u_0,u_0) &= \sum_{e_H\in\mathcal{E}_H} \frac{\tau_H}{|e_H|}\,|[\partial_n u_0]|^2_{e_H} \\ &= \sum_{e_H\in\mathcal{E}_H}\sum_{e_h\subset e_H} \left(\frac{\tau_H}{|e_H|}\frac{|e_h|}{\tau_h}\right)\frac{\tau_h}{|e_h|}\,|[\partial_n u_0]|^2_{e_h} \\ &\leq \overline{\gamma}\|u_0\|^2_{2,h} \\ &\leq c\overline{\gamma}a_h(u_0,u_0). \end{aligned} \tag{3.65}$$

Using the bound for $a_h(u_0, u_0)$ derived in (3.64), it follows that

$$a_0(u_0,u_0) \leq c\overline{\gamma}a_h(u_0,u_0) \leq c\overline{\gamma}\,\overline{\tau}[H:h]a_h(u,u). \tag{3.66}$$

Substitution of (3.61), (3.64), and (3.66) into (3.47) completes the proof, thus verifying **Assumption 1**. □

Remark 3.4.1. *Note that if the penalty compatibility condition* (PC_τ) *is enforced, then* $\overline{\gamma} = 1$.

The next step is to verify **Assumption 2**. This follows in precisely the same way as the second order problem. Again,

$$a_h(u_i,u_j) = 0 \quad \text{if} \quad S_i \cap S_j = \emptyset, \quad i,j = 1,\ldots,p.$$

Thus, for disjoint subdomains S_i and S_j, (3.32) holds with $\mathcal{E}_{ij} = 0$. All remaining subdomains are (pairwise) adjacent; in these cases, (3.32) holds with $\mathcal{E}_{ij} = 1$ by the Cauchy-Schwarz inequality. It follows that

$$\rho(\mathcal{E}) \leq \|\mathcal{E}\|_\infty \leq N_c, \tag{3.67}$$

where

$$N_c := \max_i |\{\Omega_j \,|\, S_i \cap S_j \neq \emptyset\}|.$$

In essence, $\rho(\mathcal{E})$ is bounded by the infinity norm of an adjacency matrix between the subdomains, which is typically small in most practical applications.

The verification of **Assumption 3** is presented as the following lemma.

Lemma 3.4.2. *Let $\omega \geq 1$ be given by*

$$\omega = \begin{cases} 1, & \text{if } V_0^h = V^{H,C} \text{ and assumption } (\mathrm{PC}_\tau) \text{ holds} \\ 1, & \text{if } V_0^h = V^{H,D} \text{ and assumptions } (\mathrm{PC}_\tau) \text{ and } (\mathrm{PC}_\sigma) \text{ hold} \\ 1 + \frac{\mu_\sigma + \mu_\tau}{c_a}, & \text{otherwise,} \end{cases} \tag{3.68}$$

where c_a is the coercivity constant in (3.11) and

$$\mu_\sigma = \max \left| \frac{\sigma_h}{|e_h|^3} \frac{|e_H|^3}{\sigma_H} - 1 \right| \quad \text{and} \quad \mu_\tau = \max \left| \frac{\tau_h}{|e_h|} \frac{|e_H|}{\tau_H} - 1 \right|.$$

Then there holds the bound

$$a_h(u,u) \leq \omega a_i(u,u), \qquad \forall u \in V_i^h, \quad i = 0, \ldots, p. \tag{3.69}$$

Proof. For $i = 1, \ldots, p$, the bound (3.69) holds with $\omega = 1$ by the definition of the subdomain bilinear forms by restriction. However, the more general definition of the coarse space bilinear form $a_0(\cdot,\cdot)$ necessitates a more careful analysis. From (3.28),

$$\begin{aligned} a_h(u,u) \;=\; a_0(u,u) \;&+\; \sum_{e_H \in \mathcal{E}_H} \sum_{e_h \subset e_H} \left(\frac{\sigma_h}{|e_h|^3} - \frac{\sigma_H}{|e_H|^3} \right) |[u]|_{e_h}^2 \\ &+\; \sum_{e_H \in \mathcal{E}_H} \sum_{e_h \subset e_H} \left(\frac{\tau_h}{|e_h|} - \frac{\tau_H}{|e_H|} \right) |[\partial_n u]|_{e_h}^2 . \end{aligned} \tag{3.70}$$

If $V_0^h = V^{H,C}$ and assumption (PC_τ) holds, or if $V_0^h = V^{H,D}$ and assumptions (PC_σ) and (PC_τ) hold, then the above sums vanish and (3.69) holds with $\omega = 1$ for $i = 0$

as well. In the general case,

$$\begin{aligned}
\sum_{e_H\in\mathcal{E}_H}\sum_{e_h\subset e_H}\left(\frac{\sigma_h}{|e_h|^3}-\frac{\sigma_H}{|e_H|^3}\right)|[u]|^2_{e_h} &+ \sum_{e_H\in\mathcal{E}_H}\sum_{e_h\subset e_H}\left(\frac{\tau_h}{|e_h|}-\frac{\tau_H}{|e_H|}\right)|[\partial_n u]|^2_{e_h} \\
&= \sum_{e_H\in\mathcal{E}_H}\sum_{e_h\subset e_H}\left(\frac{\sigma_h}{|e_h|^3}\frac{|e_H|^3}{\sigma_H}-1\right)\frac{\sigma_H}{|e_H|^3}|[u]|^2_{e_h} \\
&\quad + \sum_{e_H\in\mathcal{E}_H}\sum_{e_h\subset e_H}\left(\frac{\tau_h}{|e_h|}\frac{|e_H|}{\tau_H}-1\right)\frac{\tau_H}{|e_H|}|[\partial_n u]|^2_{e_h} \\
&\le \mu_\sigma\sum_{e_H\in\mathcal{E}_H}\frac{\sigma_H}{|e_H|^3}|[u]|^2_{e_H}+\mu_\tau\sum_{e_H\in\mathcal{E}_H}\frac{\tau_H}{|e_H|}|[\partial_n u]|^2_{e_H} \\
&\le (\mu_\sigma+\mu_\tau)\|u\|^2_{2,H} \\
&\le \left(\frac{\mu_\sigma+\mu_\tau}{c_a}\right)a_0(u,u),
\end{aligned} \tag{3.71}$$

where the coercivity (3.11) of $a_0(u,u)$ is used in the final step. Combining this inequality with (3.70) above gives

$$a_h(u,u) \le \left(1+\frac{\mu_\sigma+\mu_\tau}{c_a}\right)a_0(u,u), \tag{3.72}$$

which covers inequality (3.69) for the case $i=0$ and thus completes the proof. □

With all three assumptions verified and the forms of the constants involved determined, the main result for the nonoverlapping method is now presented.

Theorem 3.4.1. *The condition number $\kappa(T)$ of the operator T of the nonoverlapping additive Schwarz method defined in this section (or equivalently that of the matrix BA) satisfies*

$$\kappa(T) \le c(\overline{\sigma}+\overline{\tau}+\overline{\tau}\,\overline{\gamma})\omega(N_c+1)[H:h]^3 \tag{3.73}$$

where c is independent of p and the constants

$$\overline{\sigma}, \quad \overline{\tau}, \quad \overline{\gamma}, \quad \omega, \quad N_c, \quad \textit{and} \quad [H:h]$$

have been defined previously.

Proof. The proof is a direct application of the abstract estimate (2.55) to the estimates (3.45), (3.67), and (3.68) derived to verify the necessary assumptions. □

Remark 3.4.2. *If the various penalty compatibility assumptions are not enforced, then the parameter can be as large as powers of $[H:h]$ depending on the sizes of the parameters μ_σ, μ_τ shown in (3.68). This gives further incentive for the enforcement of these conditions.*

3.4.2 Overlapping Additive Schwarz Preconditioner

The focus of this section is to determine a corresponding condition number estimate for the overlapping case. The overlapping subdomain partition $\mathcal{T}_S$ has the same properties as assumed for the second order problem (see 2.4.2). In particular, recall that for each subdomain S_i, the overlap is characterized by some parameter $\delta_i > 0$. In addition, recall the following:

1. Require the alignments $\mathcal{T}_H \subset \mathcal{T}_h$ and $\mathcal{T}_S \subset \mathcal{T}_h$.

2. There exist nonnegative functions $\Omega_i \in W^{2,\infty} \cap C^1(\Omega)$, $i = 1, \ldots, p$, such that

$$\begin{aligned} \sum_{i=1}^{p} \theta_i &= 1, && \text{on} \quad \overline{\Omega}, \\ \theta_i &= 0 && \text{on} \quad \Omega \setminus \Omega_i \\ |\theta_i|_{\alpha,\infty} &\leq c\delta_i^{-|\alpha|} && |\alpha| = 0, 1, 2. \end{aligned}$$

   Here, δ_i represents the width of a boundary layer of Ω_i resulting from the overlap with other subdomains. It is assumed that the practical range of its values is $h \leq \delta_i \leq H$.

3. Let $N(x)$ denote the number of subdomains that contain x and $N_c \equiv \max_{x\in\Omega} N(x)$. It is reasonable to assume that N_c is a small number.

4. Define the sets Ω_i^I and $\Omega_i^{\delta_i}$ by

$$\Omega_i^I = \{x \in \Omega_i;\ x \notin \Omega_k \quad \text{for all} \quad k \neq i\}, \qquad \Omega_i^{\delta_i} = \Omega_i \setminus \Omega_i^I.$$

Finally, the contents of Remark 2.4.2 apply to the fourth order problem exactly as before. The approach to estimating the condition number for the overlapping Schwarz method again consists of verifying the three assumptions necessary to invoke the abstract estimate (2.55). To verify **Assumption 1**, use the same u_0 as the nonoverlapping case and define $u = u_0 + u_1 + \cdots + u_p$ as in the second order case by using the Lagrange interpolation operator $\Pi_h : C(\mathcal{T}_h) \to V^h$ defined locally on each cell $K \in \mathcal{T}_h$. Denote by Π_K the restriction of Π_h to K.

Proposition 3.4.2. *Let $u_0 \in V^{H,C}$ be as in (3.4.1), $w = u - u_0$, and $u_i = \Pi_h(\theta_i w) \in V_i^h$, $i = 1, \ldots, p$. Then there exists a constant c independent of p such that*

$$\begin{aligned} a_i(u_i, u_i) \ \leq\ & c\Bigg(\sum_{K\in\mathcal{T}_{h,i}} |u|^2_{2,K} + \sum_{K\in\Omega_i^{\delta_i}} \left(\delta_i^{-2}\|\nabla w\|^2_K + \delta_i^{-4}\|w\|^2_K\right) + \sum_{e_h\in\mathcal{E}_{h,i}} \frac{\sigma_h+\tau_h}{|e_h|^3}\,|[u]|^2_{e_h} \\ & + \sum_{e_h\in\mathcal{E}_{h,i}} \frac{\tau_h}{|e_h|}\,|[\partial_n u]|^2_{e_h} + \sum_{e_h\in\mathcal{E}_{h,i}} \frac{\tau_h}{|e_h|}\,|[\partial_n u_0]|^2_{e_h}\Bigg), \end{aligned} \tag{3.74}$$

for $i = 1, \ldots, p$.

Proof. Since Ω_i vanishes on $\mathcal{S}_i$, $u_i = 0$ on $\mathcal{S}_i$. By definition (3.21) of the subdomain bilinear form,

$$\begin{aligned} a_i(u_i, u_i) &= \sum_{K\in\mathcal{T}_{h,i}} \|\Delta u_i\|_K^2 + 2\sum_{e_h\in\mathcal{E}_{h,i}} \langle\{\partial_n \Delta u_i\}, [u_i]\rangle_{e_h} + \sum_{e_h\in\mathcal{E}_{h,i}} \frac{\sigma_h}{|e_h|^3} |[u_i]|_{e_h}^2 \\ &\quad - 2\sum_{e_h\in\mathcal{E}_{h,i}} \langle\{\Delta u_i\}, [\partial_n u_i]\rangle_{e_h} + \sum_{e_h\in\mathcal{E}_{h,i}} \frac{\tau_h}{|e_h|} |[\partial_n u_i]|_{e_h}^2 \\ &\leq c\Big(\sum_{K\in\mathcal{T}_{h,i}} \|\Delta u_i\|_K^2 + \sum_{e_h\in\mathcal{E}_{h,i}} \frac{\sigma_h}{|e_h|^3} |[u_i]|_{e_h}^2 + \sum_{e_h\in\mathcal{E}_{h,i}} \frac{\tau_h}{|e_h|} |[\partial_n u_i]|_{e_h}^2 \Big). \end{aligned} \tag{3.75}$$

In the last step the Cauchy-Schwarz inequality, trace and inverse inequalities, and the arithmetic-geometric mean inequality have all been used to absorb the two terms $\langle\{\partial_n \Delta u_i\}, [u_i]\rangle_{e_h}$ and $\langle\{\Delta u_i\}, [\partial_n u_i]\rangle_{e_h}$ into the sum $\sum_{K\in\mathcal{T}_{h,i}} \|\Delta u_i\|_K^2$.

The next step in the proof is to estimate the three terms on the right side of (3.75). Each estimate is rather tedious; for this reason, each term will be considered separately.

Estimation of $\sum_{K\in\mathcal{T}_{h,i}} \|\Delta u_i\|_K^2$.

Let $\overline{\theta}_{i,K}$ be the linear Lagrange interpolant of θ_i over $K \in \mathcal{T}_{h,i}$. It can be shown that

$$\begin{aligned} &\theta_i - \overline{\theta}_{i,K} = 0, \quad \text{if} \quad K \subset \Omega_i^I, \quad \text{and} \\ &\|\theta_i - \overline{\theta}_{i,K}\|_{L^\infty(K)} \leq c h_K^2 \delta_i^{-2} \quad \text{and} \quad \|\nabla(\theta_i - \overline{\theta}_{i,K})\|_{L^\infty(K)} \leq c h_K \delta_i^{-2}, \text{ if } K \subset \Omega_i^{\delta_i}. \end{aligned} \tag{3.76}$$

The above follows from properties of θ_i and the approximation properties of Lagrange interpolation which also hold in the L^∞-norm (cf. e.g. [14]).

The next few steps will be concerned with the term $\|\Delta u_i\|_K$. First, note that $\theta_i = 1$ for $K \in \Omega_i^I$, which gives

$$\|\Delta u_i\|_K = \|\Delta \Pi_K w\|_K = \|\Delta w\|_K = \|\Delta u\|_K, \quad \text{for} \quad K \in \Omega_i^I, \tag{3.77}$$

since u_0 is linear on K. As a result, only cells in $\Omega_i^{\delta_i}$ will need to be considered when bounding $\|\Delta u_i\|_K$. For $K \in \Omega_i^{\delta_i}$, using the triangle inequality gives

$$\|\Delta u_i\|_K^2 \;=\; \|\Delta\Pi_K(\theta_i w)\|_K^2 \;\leq\; 2\|\Delta\Pi_K((\theta_i - \overline{\theta}_{i,K})w)\|_K^2 + 2\|\Delta\Pi_K(\overline{\theta}_{i,K}w)\|_K^2. \quad (3.78)$$

For the first term on the right side of (3.77), using the inverse inequality (2.17), scaling, stability of Π_K in the L^∞ norm, the L^∞-L^2 inverse inequality together with (3.76) gives

$$\begin{aligned}
\|\Delta\Pi_K((\theta_i - \overline{\theta}_{i,K})w)\|_K^2 &\leq ch_K^{d-4}\|\Pi_K((\theta_i - \overline{\theta}_{i,K})w)\|_{L^\infty(K)}^2 \\
&\leq ch_K^{d-4}\|(\theta_i - \overline{\theta}_{i,K})w\|_{L^\infty(K)}^2 \\
&\leq ch_K^{-4}\|\theta_i - \overline{\theta}_{i,K}\|_{L^\infty(K)}^2\|w\|_K^2 \\
&\leq c\delta_i^{-4}\|w\|_K^2,
\end{aligned} \quad (3.79)$$

for $K \in \Omega_i^{\delta_i}$. For the second term on the right side of (3.77), using the triangle inequality, the approximation properties of Π_K, and the inverse inequality (2.17) gives

$$\begin{aligned}
\|\Delta\Pi_K(\overline{\theta}_{i,K}w)\|_K^2 &\leq 2\|\Delta(\Pi_K(\overline{\theta}_{i,K}w) - \overline{\theta}_{i,K}w)\|_K^2 + 2\|\Delta\overline{\theta}_{i,K}w\|_K^2 \\
&\leq ch_K^{2(q-1)}|\overline{\theta}_{i,K}w|_{q+1,K}^2 + |\overline{\theta}_{i,K}w|_{2,K}^2 \\
&\leq c|\overline{\theta}_{i,K}w|_{2,K}^2.
\end{aligned} \quad (3.80)$$

Using the Leibniz formula for partial derivatives and the fact the $\overline{\theta}_{i,K}$ is linear on K,

$$|\overline{\theta}_{i,K}w|_{2,K} \;\leq\; c\|\nabla\overline{\theta}_{i,K}\|_{L^\infty(K)}\|\nabla w\|_K + c\|\overline{\theta}_{i,K}\|_{L^\infty(K)}|w|_{2,K}. \quad (3.81)$$

Using (3.76) and the fact that $h_K \leq c\delta_i^{-1}$, we have the bounds

$$\|\nabla\overline{\theta}_{i,K}\|_{L^\infty(K)} \leq c\delta_i^{-1} \qquad \text{and} \qquad \|\overline{\theta}_{i,K}\|_{L^\infty(K)} \leq c.$$

From the last two inequalities and the fact that u_0 is linear it follows that

$$\begin{aligned} |\overline{\theta}_{i,K}w|_{2,K} &\leq c\delta_i^{-1}\|\nabla w\|_K + c|w|_{2,K} \\ &= c\delta_i^{-1}\|\nabla w\|_K + c|u|_{2,K}. \end{aligned} \tag{3.82}$$

Hence, using (3.82) and (3.79) in (3.78) (note $K \subset \Omega_i^{\delta_i}$ here) together with (3.77) (note $\|\Delta u\|_K \leq d|u|_{2,K}$) gives

$$\sum_{K\in\mathcal{T}_{h,i}} \|\Delta u_i\|_K^2 \leq c \sum_{K\in\mathcal{T}_{h,i}} |u|_{2,K}^2 + \sum_{K\in\Omega_i^{\delta_i}} \left(\delta_i^{-2}\|\nabla w\|_K^2 + \delta_i^{-4}\|w\|_K^2\right). \tag{3.83}$$

Estimation of $\sum_{e_h\in\mathcal{E}_{h,i}} \frac{\sigma_h}{|e_h|^3} |[u_i]|_{e_h}^2$.

Let $e_h \in \mathcal{E}_{h,i} = \mathcal{E}_{h,i}^I \cup \mathcal{E}_{h,i}^B$. The analysis in this part applies notationally to both $\mathcal{E}_{h,i}^I$ and $\mathcal{E}_{h,i}^B$. Since θ_i is smooth,

$$[u_i]|_{e_h} = \Pi_{K^+}(\theta_i w^+) - \Pi_{K^-}(\theta_i w^-).$$

The following two observations are crucial. Since the mesh $\mathcal{T}_h$ is conforming, the (nodal) Lagrange interpolation operator Π_{e_h} on e_h coincides with the restrictions of Π_{K^+} and Π_{K^-} to e_h. Also, as stated in Remark 2.4.2 (2), both Ω_i^I and $\Omega_i^{\delta_i}$ are unions of cells $K \in \mathcal{T}_h$. Hence, either $e_h \in \Omega_i^I$ or $e_h \in \Omega_i^{\delta_i}$.

If $e_h \in \Omega_i^I$, then $\theta_i|_{e_h} = 1$ and so

$$\begin{aligned} |[u_i]|_{e_h} &= |\Pi_{K^+} w^+ - \Pi_{K^-} w^-|_{e_h} \\ &= |[w]|_{e_h} \\ &= |[u]|_{e_h}, \end{aligned} \tag{3.84}$$

for $e_h \in \Omega_i^I$, given that u_0 is continuous and vanishes on $\mathcal{E}_{h,i}^B$.

If, on the other hand, $e_h \in \Omega_i^{\delta_i}$,

$$\begin{aligned} [u_i]_{e_h} &= \Pi_{K^+}(\theta_i w^+) - \Pi_{K^-}(\theta_i w^-) \\ &= \Pi_{e_h} \theta_i [w]_{e_h} \\ &= \Pi_{e_h} \theta_i [u]_{e_h}. \end{aligned}$$

Therefore, using scaling, stability of Π_{e_h} in the L^∞ norm, the L^∞-L^2 inverse inequality on e_h,

$$\begin{aligned} |[u_i]|_{e_h}^2 &\leq c|e_h|^{d-1} |\Pi_{e_h} \theta_i [u]|_{L^\infty(e_h)}^2 \\ &\leq c|e_h|^{d-1} |\theta_i [u]|_{L^\infty(e_h)}^2 \\ &\leq c|e_h|^{d-1} |\theta_i|_{L^\infty(e_h)}^2 |[u]|_{L^\infty(e_h)}^2 \\ &\leq c|[u]|_{e_h}^2, \end{aligned} \tag{3.85}$$

for $e_h \subset \Omega_i^{\delta_i}$.

Thus, from (3.84) and (3.85),

$$\sum_{e_h \in \mathcal{E}_{h,i}} \frac{\sigma_h}{|e_h|^3} |[u_i]|_{e_h}^2 \leq c \sum_{e_h \in \mathcal{E}_{h,i}} \frac{\sigma_h}{|e_h|^3} |[u]|_{e_h}^2 . \tag{3.86}$$

Estimation of $\sum_{e_h \in \mathcal{E}_{h,i}} \frac{\tau_h}{|e_h|} \left|[\partial_n u_i]\right|^2_{e_h}$.

It is easy to see that for $e_h \subset \overline{\Omega_i^I}$, for which $\theta_i = 1$,

$$\begin{aligned} \left|[\partial_n u_i]\right|_{e_h} &= \left|[\partial_n w]\right|_{e_h} \\ &\leq \left|[\partial_n u]\right|_{e_h} + \left|[\partial_n u_0]\right|_{e_h}. \end{aligned} \tag{3.87}$$

So in what follows, consider $e_h \in \Omega_i^{\delta_i}$. For such e_h, it is only necessary to consider the case $\theta_i \neq 1$. Hence it follows from the construction of Ω_i^I and $\Omega_i^{\delta_i}$ that both K^+ and K^- belong to $\Omega_i^{\delta_i}$. This eliminates the need to enlarge the boundary layer by an amount of $O(h)$, which could be done in any case at the cost of further complicating the notation, however. Then

$$\begin{aligned} \left|[\partial_n u_i]\right|_{e_h} &= \left|\nabla\big(\Pi_{K^+}(\theta_i w^+) - \Pi_{K^-}(\theta_i w^-)\big)\cdot \mathbf{n}^+\right|^2_{e_h} \\ &\leq 3\left|\nabla\big(\Pi_{K^+}(\theta_i w^+) - \theta_i w^+\big)\right|^2_{e_h} + 3\left|\nabla\big(\theta_i[w]\big)\cdot \mathbf{n}^+\right|^2_{e_h} \\ &\quad +3\left|\nabla\big(\Pi_{K^-}(\theta_i w^-) - \theta_i w^-\big)\right|^2_{e_h} \\ &:= 3(A^+ + B + A^-). \end{aligned} \tag{3.88}$$

The estimation of A^- is identical to that of A^+ so only the estimate for A^+ will be presented (dropping the $(^+)$ for simplicity).

$$\begin{aligned} A &\leq 3\left|\nabla\big(\Pi_K(\theta_i w) - \Pi_K(\overline{\theta}_{i,K} w)\big)\right|^2_{e_h} + 3\left|\nabla\big(\Pi_K(\overline{\theta}_{i,K} w) - \overline{\theta}_{i,K} w\big)\right|^2_{e_h} \\ &\quad +3\left|\nabla\big((\overline{\theta}_{i,K} - \theta_i) w\big)\right|^2_{e_h} \\ &:= 3(A_1 + A_2 + A_3). \end{aligned} \tag{3.89}$$

Using the trace and inverse inequalities, scaling, (3.76), and the L^∞-L^2 inverse inequality gives

$$\begin{aligned} A_1 &\leq ch_K^{-3}\|\Pi_K\big((\theta_i-\overline{\theta}_{i,K})w\big)\|_K^2 \\ &\leq ch_K^{d-3}\|\Pi_K\big((\theta_i-\overline{\theta}_{i,K})w\big)\|_{L^\infty(K)}^2 \\ &\leq ch_K^{d-3}\|(\theta_i-\overline{\theta}_{i,K})w\|_{L^\infty(K)}^2 \\ &\leq ch_K^{d-3}\|\theta_i-\overline{\theta}_{i,K}\|_{L^\infty(K)}^2\|w\|_{L^\infty(K)}^2 \\ &\leq ch_K\delta_i^{-4}\|w\|_K^2. \end{aligned} \tag{3.90}$$

Using the trace and inverse inequalities and the approximation properties of Π_K,

$$\begin{aligned} A_2 &\leq ch_K^{-3}\|\Pi_K(\overline{\theta}_{i,K}w)-\overline{\theta}_{i,K}w\|_K^2 \\ &\leq ch_K^{2(q+1)-3}|\overline{\theta}_{i,K}w|_{q+1,K}^2 \\ &\leq ch_K|\overline{\theta}_{i,K}w|_{2,K}^2 \\ &\leq ch_K(\delta_i^{-2}\|\nabla w\|_K^2+|u|_{2,K}^2). \end{aligned} \tag{3.91}$$

where the estimate (3.82) was used in the last step. As for A_3, from the trace inequality, Leibniz' formula and (3.76), it follows that

$$\begin{aligned} A_3 &\leq ch_K^{-1}|(\overline{\theta}_{i,K}-\theta_i)w|_{1,K}^2+ch_K|(\overline{\theta}_{i,K}-\theta_i)w|_{2,K}^2 \\ &\leq c\Big(h_K^{-1}|\overline{\theta}_{i,K}-\theta_i|_{1,\infty,K}^2+h_K|\theta_i|_{2,\infty,K}^2\Big)\|w\|_K^2 \\ &\quad +c\Big(h_K^{-1}\|\overline{\theta}_{i,K}-\theta_i\|_{L^\infty(K)}^2+h_K|\overline{\theta}_{i,K}-\theta_i|_{1,\infty,K}^2\Big)\|\nabla w\|_K^2 \\ &\quad +h_K\|\overline{\theta}_{i,K}-\theta_i\|_{L^\infty(K)}^2|w|_{2,K}^2 \\ &\leq ch_K(\delta_i^{-4}\|w\|_K^2+\delta_i^{-2}\|\nabla w\|_K^2+|u|_{2,K}^2), \end{aligned} \tag{3.92}$$

where we have also used the fact that $h_K\leq c\delta_i$ and $|w|_{2,K}=|u|_{2,K}$.

It remains to estimate B. In view of Leibniz' formula, (3.76), and the fact that $[u_0] = 0$ for $e_h \in \mathcal{E}_{h,i}$,

$$\begin{aligned} B &\leq 2\,|(\partial_n\theta_i)[w]|^2_{e_h} + 2\,|\theta_i[\partial_n w]|^2_{e_h} \\ &\leq c\|\nabla\theta_i\|^2_{L^\infty(e_h)}\,|[u]|^2_{e_h} + c\|\theta_i\|^2_{L^\infty(e_h)}\,|[\partial_n w]|^2_{e_h} \\ &\leq c\Big(\delta_i^{-2}\,|[u]|^2_{e_h} + |[\partial_n u]|^2_{e_h} + |[\partial_n u_0]|^2_{e_h}\Big) \\ &\leq c\Big(|e_h|^{-2}\,|[u]|^2_{e_h} + |[\partial_n u]|^2_{e_h} + |[\partial_n u_0]|^2_{e_h}\Big), \end{aligned} \tag{3.93}$$

where the fact that $|e_h| \leq c\delta_i$ was used in the last step.

Collecting the estimates (3.90)-(3.93) gives

$$\begin{aligned} \sum_{e_h\in\mathcal{E}_{h,i}} \frac{\tau_h}{|e_h|}\,|[\partial_n u_i]|^2_{e_h} \;\leq\; & \sum_{e_h\in\mathcal{E}_{h,i}} \frac{\tau_h}{|e_h|}\Big(\,|[\partial_n u]|^2_{e_h} + |[\partial_n u_0]|^2_{e_h}\,\Big) + \sum_{K\in\Omega_i^{\delta_i}} \frac{\tau_h}{|e_h|^3}\,|[u]|^2_{e_h} \\ & + \sum_{K\in\Omega_i^{\delta_i}} \Big(|u|^2_{2,K} + \delta_i^{-4}\|w\|^2_K + \delta_i^{-2}\|\nabla w\|^2_K\Big). \end{aligned} \tag{3.94}$$

The required estimate (3.74) now follows upon using (3.83), (3.86), and (3.94) in (3.75). □

Proposition 3.4.3. *For any $u \in V^h$, let $u = u_0 + u_1 + \cdots + u_p$ where u_0 is as in Proposition (3.4.1) and let $u_i = \Pi_h\theta_i(u - u_0)$, $i = 1, \ldots, p$. Then there exists a constant c which is independent of p such that*

$$\sum_{i=0}^{p} a_i(u_i, u_i) \;\leq\; c\Big(\overline{\tau}\,\overline{\gamma}[H:h] + N_c^2[H:\delta]^3\Big)a_h(u,u), \tag{3.95}$$

where

$$[H:\delta] \;:=\; \max_{1\leq i\leq p}\max_{D\subset\Omega_i}\frac{H_D}{\delta_i}.$$

Proof. The proof consists in estimating the terms in (3.74). To begin, note that since $\sum_{i=1}^p \theta_i = 1$ on Ω and Π_h is the identity operator when restricted to V_h, then $u = \sum_{i=0}^p u_i$.

Since at any point in Ω the number of overlapping subdomains is bounded by N_c and $\mathcal{T}_H \subseteq \mathcal{T}_h$, it follows from (3.17) and (3.11) that

$$\sum_{i=1}^{p} \sum_{e_h \in \mathcal{E}_{h,i}} |u|_{2,K}^2 \leq N_c \sum_{K \in \mathcal{T}_h} |u|_K^2 \leq c\|u\|_{2,h}^2 \leq c a_h(u,u). \tag{3.96}$$

Clearly

$$\sum_{i=1}^{p} \sum_{K \in \mathcal{T}_{h,i}} \left(\frac{\sigma_h + \tau_h}{|e_h|^3} |[u]|_{e_h}^2 + \frac{\tau_h}{|e_h|} |[\partial_n u]|_{e_h}^2 \right) \leq c a_h(u,u). \tag{3.97}$$

In the above it was assumed that $\tau_h \leq c\sigma_h$ for the sake of simplicity. The next step is estimating the last term in (3.74). Note that $E(u)$ belongs to $H_0^2(\Omega)$ and that $[\partial_n(E(u) - u_0)]$ vanishes on all edges e_h that are in the interior of a cell D in $\mathcal{T}_H$. It then follows from the inverse and trace inequalities, (3.46), (3.16) and (3.11) that

$$\begin{aligned}
\sum_{i=1}^{p} \sum_{e_h \in \mathcal{E}_{h,i}} \frac{\tau_h}{|e_h|} |[\partial_n u_0]|_{e_h}^2 &\leq c\overline{\tau} \sum_{e_H \in \mathcal{E}_H} h_D^{-1} |[\partial_n(E(u) - u_0)]|_{e_h}^2 \\
&\leq c\overline{\tau} \sum_{e_H \in \mathcal{E}_H} h_D^{-1} H_D^{-3} \|E(u) - u_0\|_D^2 \\
&\leq c\overline{\tau} \sum_{e_H \in \mathcal{E}_H} h_D^{-1} H_D |E(u)|_{2,D}^2 \\
&\leq c\overline{\tau}[H:h] |E(u)|_{2,\Omega}^2 \\
&\leq c\overline{\tau}[H:h] a_h(u,u).
\end{aligned} \tag{3.98}$$

Let $\tilde{\Omega}_i$ be the nonoverlapping part described in Remark 2.4.2 (1). From the triangle inequality,

$$\begin{aligned}
\sum_{K \subset \tilde{\Omega}_i \cap \Omega_i^{\delta_i}} \delta_i^{-4} \|u - u_0\|_K^2 &\leq 2 \sum_{K \subset \tilde{\Omega}_i \cap \Omega_i^{\delta_i}} \delta_i^{-4} \Big(\|u - E(u)\|_K^2 + \|E(u) - u_0\|_K^2 \Big) \\
&:= 2A_i + 2B_i.
\end{aligned} \tag{3.99}$$

Using Lemma 3.2.1 with $m = 0$,

$$\begin{aligned} A_i &\leq c \sum_{K \subset \tilde{\Omega}_i \cap \Omega_i^{\delta_i}} \delta_i^{-4} \sum_{e_h \in \omega(K)} \left(|e_h| \, |[u]|_{e_h}^2 + |e_h|^3 \, |[\partial_n u]|_{e_h}^2 \right) \\ &\leq c \sum_{K \subset \tilde{\Omega}_i \cap \Omega_i^{\delta_i}} \sum_{e_h \in \omega(K)} \left(|e_h|^{-3} \, |[u]|_{e_h}^2 + |e_h|^{-1} \, |[\partial_n u]|_{e_h}^2 \right) \\ &\leq c \sum_{e_h \in \Omega_i} \left(|e_h|^{-3} \, |[u]|_{e_h}^2 + |e_h|^{-1} \, |[\partial_n u]|_{e_h}^2 \right). \end{aligned} \tag{3.100}$$

Furthermore, using (3.46) and the generalized Poincaré inequality (2.36) with $\rho = \delta_i$ together with the fact that $|E(u) - u_0|_{1,h,D} = \|\nabla(E(u) - u_0)\|_D$ gives

$$\begin{aligned} B_i &\leq c \sum_{D \subset \tilde{\Omega}_i} \delta_i^{-4} \|E(u) - u_0\|_{D \cap \tilde{\Omega}_i \cap \Omega_i^{\delta_i}}^2 \\ &\leq c \sum_{D \subset \tilde{\Omega}_i} \delta_i^{-3} \Big(H_D^{-1} \|E(u) - u_0\|_D^2 + H_D \|\nabla(E(u) - u_0)\|_D^2 \Big) \\ &\leq c[H : \delta]^3 |E(u)|_{2,\tilde{\Omega}_i}^2. \end{aligned} \tag{3.101}$$

As in the second order problem, the partition $\mathcal{T}_S$ is obtained from a nonoverlapping partition $\mathcal{T}_{\tilde{S}} = \{\tilde{\Omega}_i, \, i = 1, \ldots, p\}$ aligned with $\mathcal{T}_H$ by adding layers of cells from $\mathcal{T}_h$. It follows that $D \cap \tilde{\Omega}_i \cap \Omega_i^{\delta_i}$ is indeed a boundary layer of D, which justifies the use of Lemma 2.1.3 above.

Combining (3.99)-(3.101) and (3.16) gives

$$\begin{aligned} \sum_{i=1}^{p} \sum_{K \in \Omega_i^{\delta_i}} \delta_i^{-4} \|u - u_0\|_K^2 &\leq c N_c \sum_{i=1}^{p} \sum_{K \subset \tilde{\Omega}_i \cap \Omega_i^{\delta_i}} \delta_i^{-4} \|u - u_0\|_K^2 \\ &\leq c N_c^2 \sum_{e_h \in \mathcal{E}_h} \left(|e_h|^{-3} \, |[u]|_{e_h}^2 + |e_h|^{-1} \, |[\partial_n u]|_{e_h}^2 \right) \\ &\quad + c N_c [H : \delta]^3 |E(u)|_{2,\Omega}^2 \\ &\leq c N_c^2 [H : \delta]^3 a_h(u, u). \end{aligned} \tag{3.102}$$

The treatment of the terms $\sum_{K\in\Omega_i^{\delta_i}} \delta_i^{-2}\|\nabla(u-u_0)\|_K^2$ proceeds exactly as above. One minor difference is that in this case the generalized Poincaré lemma is applied to $\nabla(E(u)-u_0)$, which is permitted since $E(u)$ and u_0 are smooth on $D\in\mathcal{T}_H$. Another difference is that $[H:\delta]$ appears instead of $[H:\delta]^3$. Here,

$$\sum_{i=0}^{p}\sum_{K\in\Omega_i^{\delta_i}} \delta_i^{-2}\|\nabla(u-u_0)\|_K^2 \;\leq\; cN_c^2[H:\delta]a_h(u,u). \tag{3.103}$$

The bound $a_0(u_0,u_0)\leq c\overline{\tau}\,\overline{\gamma}[H:h]a_h^{\gamma_h}(u,u)$ (cf. (3.65)) which was derived earlier is still valid since the same u_0 is used here. Combining this with the bounds (3.96)-(3.98), (3.102), (3.103) and keeping only the significant coefficients leads to the required estimate (3.95). □

Theorem 3.4.2. *The condition number $\kappa(T)$ of the operator T of the overlapping additive Schwarz method defined in this section (or equivalently, that of the matrix BA) satisfies*

$$\kappa(T)\;\leq\; c\omega\Big(\overline{\tau}\,\overline{\gamma}[H:h]+N_c^2[H:\delta]^3\Big)(1+2N_c), \tag{3.104}$$

where c is independent of P and the constants

$$\omega,\quad \overline{\tau},\quad \overline{\gamma},\quad N_c,\quad [H:h],\quad \textit{and}\quad [H:\delta]$$

have all been defined previously.

Proof. Proposition 3.4.3 asserts that **Assumption 1** holds with $C_0^2 = \overline{\tau}\,\overline{\gamma}[H:h]+N_c^2[H:\delta]^3$. The verification of **Assumption 3** is exactly the same as in the nonoverlapping case. For **Assumption 2**, the argument used in the second order overlapping case still applies: replace the condition $\overline{\mathcal{S}}_i\cap\overline{\mathcal{S}}_j=\emptyset$ in (2.81) by $\overline{\Omega}_i\cap\overline{\Omega}_j=\emptyset$. In this case, the bound $\rho(\mathcal{E})\leq 2N_c+1$ is a reasonable estimate. The result now follows from the abstract estimate (2.55) given in Theorem 2.4.1. □

3.5 Numerical Experiments

The experiments presented in this section mirror those conducted for the second order problem. Once again, the following topics are examined:

- Comparison of the performance of the nonoverlapping and overlapping preconditioners.
- Dependence of the preconditioner on the penalty parameters σ and τ.
- Comparison of the performance of both preconditioners with the penalty compatibility condition (PC) enforced versus not enforced.

All experiments are again performed using the preconditioned Conjugate Gradient (PCG) method. The three test problems considered in this case are standard for the biharmonic problem.

For the first problem, let $\Omega = [0,1] \times [0,1]$ and choose the right hand side function f and boundary data g_D and g_N so that the exact solution to (3.1)-(3.3) is given by

$$u(x,y) = \sin^2(\pi x)\sin^2(\pi y). \tag{P4}$$

For the second test problem, again let $\Omega = [0,1] \times [0,1]$ and choose choose the right hand side function f and boundary data g_D and g_N so that the exact solution to (3.1)-(3.3) is the oscillatory function is given by

$$u(x,y) = \sin^2(8\pi x)\sin^2(8\pi y). \tag{P5}$$

The final test problem is again a problem with a singularity. As before, Ω is chosen as an L-shaped domain with vertices $(0,0), (0,\frac{1}{2}), (-\frac{1}{2},\frac{1}{2}), (-\frac{1}{2},-\frac{1}{2}), (\frac{1}{2},-\frac{1}{2})$, and $(\frac{1}{2},0)$ with the reentrant corner at the origin. Choose the right hand side function $f = 0$ and the boundary data g_D and g_N so that the exact solution to (3.1)-(3.3) in

polar coordinates is given by

$$u = r^{5/3} \sin \tfrac{5\theta}{3}. \tag{P6}$$

All other aspects of the experiments correspond to those found in Chapter 2, with the exceptions of implementing $\|e_k\| \leq 10^{-7}$ as the stopping criterion for the CG and PCG iteration. As before, we will seek to verify the theory; in this case, that the nonoverlapping preconditioned system has a condition number of order $O([H : h]^3)$, as predicted by Theorem 3.4.1. Similarly, we will observe if the overlapping preconditioned system has a condition number of order $O([H : \delta]^3)$, as predicted by Theorem 3.4.2. Finally, we will enforce both penalty compatibility conditions (PC_σ) and (PC_τ) throughout this section unless otherwise noted.

3.5.1 Comparison of the Nonoverlapping and the Overlapping Preconditioners

In Chapter 2, the overlapping preconditioner exhibited superior performance to the nonoverlapping method in almost every aspect. In the current scenario, the higher regularity requirements of the fourth order problem necessitates the use of polynomial approximations of higher degree than those used for the second order. Naturally, the increased number of degrees of freedom will result in much larger stiffness matrices, which in turn increases the computational workload. The question at hand is whether or not the overlapping method will still maintain its outstanding performance in the face of greater computational overhead.

Table 3.1 offers a comparison of the three solvers when applied to test problem (P4) using $\sigma_h = \tau_h = 100.0$, degree $q = 3$, and a variety of coarse and fine meshes. Note that the combinations of $1/H = 1/h = 8$, $1/H = 1/h = 16$, and $1/H = 16, 1/h = 8$ are excluded. The first two cases have the same coarse and fine grids; solving the coarse grid problem eliminates the need for further work. The latter case violates the assumed alignment $\mathcal{T}_H \subseteq \mathcal{T}_h$.

Table 3.1: (P4) Comparison of solvers. κ estimates, (iterations) for $q = 3$

	NOV PCG			OV PCG			CG
$1/h$ \ $1/H$	4	8	16	4	8	16	
8	360.1 (119)	-	-	11.0 (21)	-	-	96419.5 (1706)
16	2279.2 (282)	307.7 (100)	-	26.6 (31)	8.2 (18)	-	?
32	18129.1 (779)	2006.9 (237)	182.3 (72)	76.6 (55)	16.1 (26)	6.4 (15)	?
64	145918.8 (2103)	13599.4 (612)	914.4 (143)	316.2 (114)	34.6 (37)	9.0 (18)	?

The results for conjugate gradient in the fourth order problem exhibit clearly the need for some type of improvement. We only report the performance of CG for the case $1/h = 8$ as the duration of subsequent tests are too lengthy for the current implementation. The performance of the preconditioned methods is significantly improved over that of CG. As in the second-order problem, the overlapping method exhibits condition numbers and iteration counts that are much smaller than those of the nonoverlapping counterpart.

The adherence to the $O([H : h]^3)$ law predicted by Theorem 3.4.1 for the nonoverlapping method is evident; each condition number increases roughly by a factor between five and ten as $1/h$ doubles. This is very close to the expected rate, where each condition number should grow at a factor of eight as $1/h$ doubles. As in the second order case, the rate of increase for κ with the overlapping method is again less than the $O(\frac{H}{\delta})$ prediction made by Theorem 3.4.2 (recall that $\delta \approx h$ in these experiments). Instead, the condition numbers in this case grow by a factor less than eight that seems to increase as the number of refinements grows.

Table 3.2 displays the results for test problem (P5) again using $\sigma_h = \tau_h = 100.0$, degree $q = 3$, and the same selection of coarse and fine meshes. We again only report the performance of CG for the case $1/h = 8$, which is much worse for this test problem. The performance of the preconditioned methods is quite similar to that observed for (P4), although it seems the nonoverlapping method exhibits some deterioration

Table 3.2: (P5) Comparison of solvers. κ estimates, (iterations) for $q = 3$

	NOV PCG			OV PCG			CG
$1/h$ \ $1/H$	4	8	16	4	8	16	
8	339.8 (144)	-	-	11.2 (27)	-	-	3163346.0 (25539)
16	2547.7 (416)	463.4 (172)	-	20.4 (37)	9.0 (25)	-	?
32	18374.3 (1083)	2483.7 (377)	351.3 (140)	48.6 (58)	13.4 (32)	7.5 (22)	?
64	123376.1 (2661)	14732.8 (891)	1382.3 (259)	188.0 (121)	29.9 (46)	11.2 (26)	?

while the overlapping method actually improves somewhat. The adherence to the $O([H : h]^3)$ rule for the nonoverlapping preconditioner is again evident, while the rate of growth for the condition number in the overlapping case is still underperforming the $O(\frac{H}{\delta})$ law.

Table 3.3 details the results for test problem (P6), using $\sigma_h = \tau_h = 100.0$, degree $q = 3$, and the same choice of coarse and fine meshes. As in the second-order problem, the subdomain partition in this case only contains 12 subdomains due to the L-shaped domain and "checkerboard" subdomain partition. The results are very similar to the previous test problems. The performance of the nonoverlapping preconditioner again follows the $O([H : h]^3)$ law, while the overlapping method is still behind the $O(\frac{H}{\delta})$ rule.

Table 3.3: (P6) Comparison of solvers. κ estimates, (iterations) for $q = 3$

	NOV PCG			OV PCG			CG
$1/h$ \ $1/H$	4	8	16	4	8	16	
8	417.9 (185)	-	-	9.5 (29)	-	-	172267.3 (3850)
16	2041.7 (427)	289.1 (156)	-	23.5 (50)	6.7 (25)	-	?
32	13918.5 (1158)	1644.8 (377)	252.0 (146)	78.5 (96)	13.1 (37)	6.2 (25)	?
64	114471.0 (3390)	14378.4 (1192)	1471.5 (358)	228.7 (163)	33.6 (63)	8.9 (32)	?

3.5.2 Dependence on the Penalty Parameters σ and τ

Previously in this chapter, we showed that the condition number of the nonoverlapping preconditioned system has an explicit dependence on both penalty parameters σ and τ (Theorem 3.4.1). In Theorem 3.4.2 it was shown that the overlapping method has a dependence only on the parameter τ. In a similar manner to the experiments performed for the second-order problem, we will investigate the effect of allowing both parameters σ and τ to increase at the same rate.

Tables 3.4-3.5 display the performance of both of the PCG methods for (P4). Here we use $q = 3$ and a minimal overlap $\delta \approx h$ for the overlapping PCG method. Results are shown for several combinations of coarse and fine meshes and for $\sigma = \tau = 100, 1000$ and 10000.

As an example, Figures 3.1-3.3 illustrate the relative performance of each preconditioned method for the particular meshes $h = 1/64$ and $H = 1/16$. The dependence on σ and τ for the nonoverlapping method is quite pronounced, as is reasonable give the dependence predicted by Theorem 3.4.1. The dependence on τ of the overlapping method predicted by Theorem 3.4.2 is slight, which is reasonable given that $\overline{\gamma} = 1$ when the condition (PC_τ) is enforced.

The corresponding reports for test problem (P5) are shown in Tables 3.6-3.7 and Figures 3.4-3.6. Similarly, the results for (P6) are displayed in 3.8-3.9 and Figures 3.7-3.9. In all experiments, the dependence of the nonoverlapping method on the penalty parameters σ and τ is clearly represented, while the dependence on τ for the overlapping method is again slight.

Table 3.4: Variations of σ, τ (PCG): (P4) with $q = 3$, κ estimates (iterations)

	Nonoverlapping PCG: $\kappa(BA)$, (iterations)			Overlapping PCG: $\kappa(BA)$, (iterations)		
$\frac{1}{H} : \frac{1}{h}$	100	1000	10000	100	1000	10000
4 : 8	360.1 (119)	1616.9 (254)	6015.2 (490)	11.0 (21)	17.9 (27)	19.3 (29)
4 : 16	2279.2 (282)	16724.3 (758)	105808.4 (1924)	26.6 (31)	37.3 (38)	59.2 (48)
8 : 16	307.7 (100)	1779.9 (240)	8889.6 (535)	8.2 (18)	14.2 (24)	21.3 (29)
8 : 32	2006.9 (237)	13443.3 (580)	97035.5 (1540)	16.1 (26)	31.4 (42)	65.1 (56)
16 : 32	182.3 (72)	1743.8 (258)	12420.9 (736)	6.4 (15)	15.9 (30)	37.8 (43)
16 : 64	914.4 (143)	12415.0 (646)	100552.4 (1726)	9.0 (18)	30.2 (33)	91.3 (53)

Table 3.5: Variations of σ, τ (PCG): (P4) with $q = 3$, CPU runtimes

	NOV PCG: CPU runtimes			OV PCG: CPU runtimes		
$\frac{1}{H} : \frac{1}{h}$	100	1000	10000	100	1000	10000
4 : 8	0.248	0.473	0.854	0.087	0.095	0.105
4 : 16	2.062	4.636	11.197	0.476	0.571	0.606
8 : 16	0.808	1.862	3.852	0.357	0.399	0.451
8 : 32	6.986	13.863	35.737	2.365	2.978	3.498
16 : 32	2.928	7.760	21.327	2.007	2.595	3.075
16 : 64	23.971	91.940	251.172	28.617	30.986	35.799

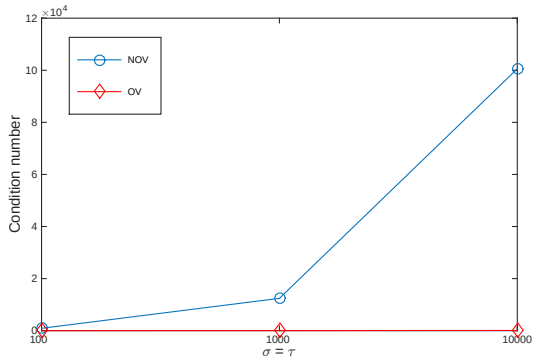


Figure 3.1: (P4) Variations of σ and τ. κ estimates, $q = 3$

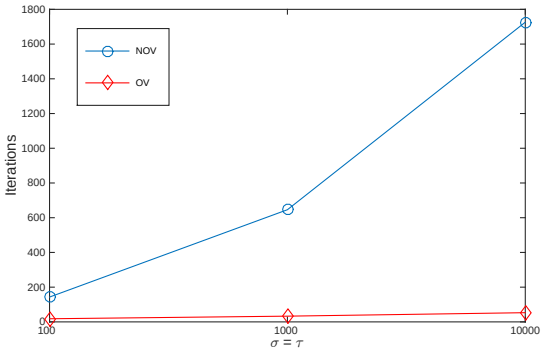


Figure 3.2: (P4) Variations of σ and τ. Iteration counts, $q = 3$

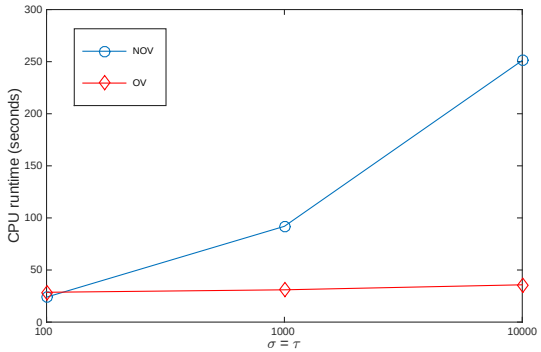


Figure 3.3: (P4) Variations of σ and τ. CPU runtimes, $q = 3$

Table 3.6: Variations of σ, τ (PCG): (P5) with $q = 3$, κ estimates (iterations)

	Nonoverlapping PCG: $\kappa(BA)$, (iterations)			Overlapping PCG: $\kappa(BA)$, (iterations)		
$\frac{1}{H} : \frac{1}{h}$	100	1000	10000	100	1000	10000
4 : 8	339.8 (144)	1629.3 (319)	5486.3 (575)	11.2 (27)	15.9 (31)	17.1 (33)
4 : 16	2547.7 (416)	18275.1 (1092)	121613.1 (2512)	20.4 (37)	29.8 (45)	36.6 (53)
8 : 16	463.4 (172)	2796.3 (398)	13456.4 (895)	9.0 (25)	15.0 (34)	22. 6()39
8 : 32	2483.7 (377)	19599.2 (1079)	165310.3 (3648)	13.4 (32)	17.9 (37)	78.6 (81)
16 : 32	351.3 (140)	2571.2 (364)	18552.7 (1216)	7.5 (22)	11.8 (28)	43.9 (60)
16 : 64	1382.3 (259)	19367.6 (1159)	142872.9 (2949)	11.2 (26)	39.4 (53)	94.9 (77)

Table 3.7: Variations of σ, τ (PCG): (P5) with $q = 3$, CPU runtimes

	NOV PCG: CPU runtimes			OV PCG: CPU runtimes		
$\frac{1}{H} : \frac{1}{h}$	100	1000	10000	100	1000	10000
4 : 8	0.212	0.329	0.575	0.089	0.096	0.109
4 : 16	2.045	5.155	11.880	0.496	0.543	0.606
8 : 16	1.065	2.175	4.788	0.405	0.476	0.520
8 : 32	9.005	24.842	87.048	2.543	2.726	4.472
16 : 32	4.323	10.344	33.774	2.224	2.463	3.967
16 : 64	38.921	169.3	430.841	30.080	36.956	42.781

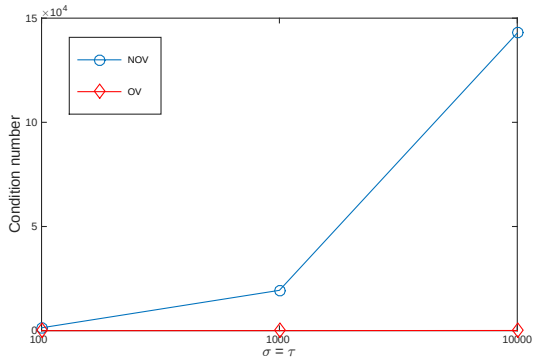


Figure 3.4: (P5) Variations of σ and τ. κ estimates, $q = 3$

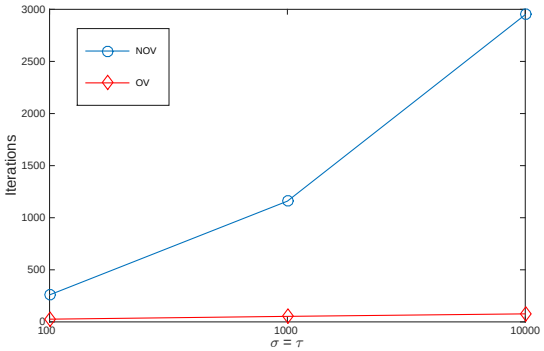


Figure 3.5: (P5) Variations of σ and τ. Iteration counts, $q = 3$

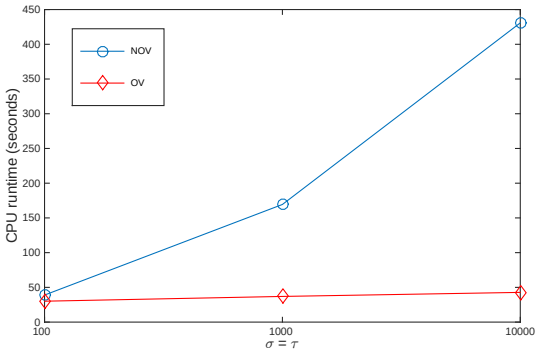


Figure 3.6: (P5) Variations of σ and τ. CPU runtimes, $q = 3$

Table 3.8: Variations of σ, τ (PCG): (P6) with $q = 3$, κ estimates (iterations)

	Nonoverlapping PCG: $\kappa(BA)$, (iterations)			Overlapping PCG: $\kappa(BA)$, (iterations)		
$\frac{1}{H} : \frac{1}{h}$	100	1000	10000	100	1000	10000
4 : 8	417.9 (185)	1960.2 (427)	5938.8 (763)	9.5 (29)	17.0 (41)	25.8 (53)
4 : 16	2041.7 (427)	12333.9 (1105)	51107.5 (2221)	23.5 (50)	34.3 (63)	62.1 (90)
8 : 16	289.1 (156)	1974.3 (428)	6748.3 (781)	6.7 (25)	13.1 (36)	29.5 (56)
8 : 32	1644.8 (377)	11675.0 (1063)	72119.9 (2803)	13.1 (37)	15.9 (43)	46.7 (76)
16 : 32	252.0 (146)	1832.6 (408)	9002.9 (960)	6.2 (25)	9.8 (32)	25.3 (53)
16 : 64	1471.5 (358)	11793.6 (1060)	84438.0 (2980)	8.9 (32)	11.3 (37)	53.3 (90)

Table 3.9: Variations of σ, τ (PCG): (P6) with $q = 3$, CPU runtimes

	NOV PCG: CPU runtimes			OV PCG: CPU runtimes		
$\frac{1}{H} : \frac{1}{h}$	100	1000	10000	100	1000	10000
4 : 8	0.171	0.348	0.631	0.062	0.086	0.101
4 : 16	1.489	3.947	7.833	0.401	0.508	0.619
8 : 16	0.635	1.706	3.094	0.276	0.335	0.464
8 : 32	6.726	18.161	47.690	1.812	1.954	2.874
16 : 32	3.181	8.273	18.861	1.549	1.752	2.366
16 : 64	38.856	112.733	347.690	20.015	20.793	30.065

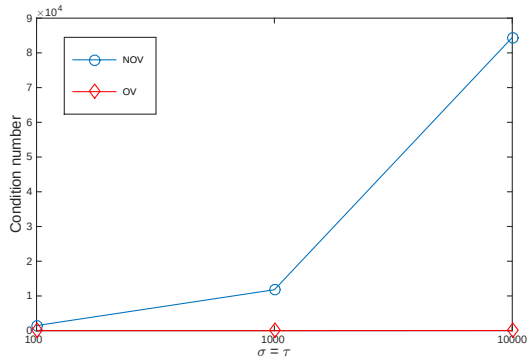


Figure 3.7: (P6) Variations of σ and τ. κ estimates, $q = 3$

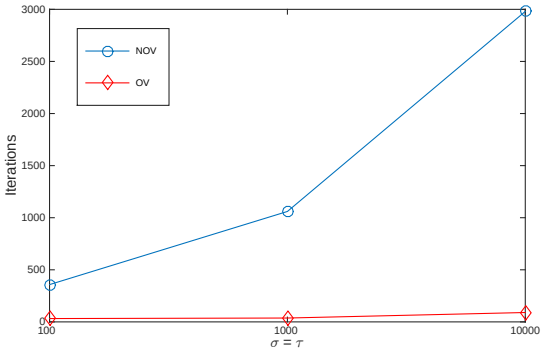


Figure 3.8: (P6) Variations of σ and τ. Iteration counts, $q = 3$

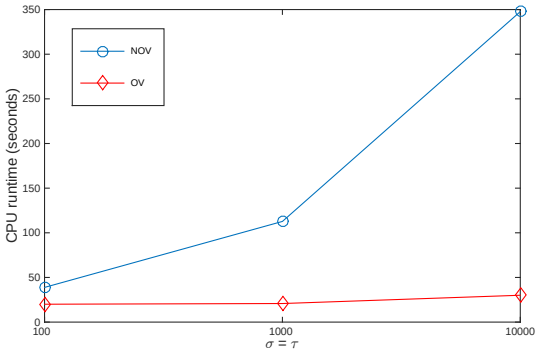


Figure 3.9: (P6) Variations of σ and τ. CPU runtimes, $q = 3$

3.5.3 Penalty Compatibility Conditions for the Discontinuous Coarse Subspace

As discussed previously, choosing to use a completely discontinuous coarse subspace $V^{H,D}$ when constructing the coarse mesh component of the Schwarz preconditioner necessitates the enforcement of (PC_σ) and (PC_τ) in order for the direct definition of the coarse solver (2.43) to coincide with the definition by restriction (2.44). Here we will examine computationally the effects of failing to enforce these penalty compatibility conditions.

We investigate the effect of not enforcing (PC_σ) and (PC_τ) over the same range of parameters used in the previous experiments; i.e., $\sigma = \tau = 100, 1000$, and 10000 for each test problem under identical conditions as the experiments reported previously.

The experimental results corresponding to the failure of enforcing (PC_σ) and (PC_τ) for (P4) are given in Tables 3.10 and 3.11. It is apparent that the effects are worse than in the second-order problem; here, both preconditioning methods show a marked decrease in performance (cf. Tables 2.13 and 2.14). As an example, Figures 3.10 - 3.12 display comparisons for the nonoverlapping method between the results with (PC_σ), (PC_τ) both enforced and both not enforced for $1/h = 64$ and $1/H = 16$, across all three performance metrics. The corresponding comparison for the overlapping method is shown in Figures 3.13 - 3.15.

Similar comparisons for test problem (P5) are shown in Tables 3.12, 3.13, and Figures 3.16 - 3.21. Corresponding results for test problem (P6) are shown in Tables 3.14, 3.15, and Figures 3.22 - 3.27. All experiments show that both methods suffer a similar decrease in performance, again emphasizing the importance of enforcing the compatibility conditions for the penalty paramaters σ and τ.

Table 3.10: (PC_σ), (PC_τ) not enforced. (P4), $q = 3$, κ estimates (iterations)

	Nonoverlapping PCG: $\kappa(BA)$, (iterations)			Overlapping PCG: $\kappa(BA)$, (iterations)		
$\frac{1}{H} : \frac{1}{h}$	100	1000	10000	100	1000	10000
4 : 8	986.0 (198)	4008.6 (389)	18893.2 (626)	25.4 (32)	41.7 (42)	47.2 (44)
4 : 16	25805.1 (887)	181115.9 (2201)	997600.9 (4411)	245.1 (93)	351.8 (115)	565.3 (146)
8 : 16	849.5 (168)	4889.5 (406)	24845.8 (901)	19.9 (27)	32.9 (35)	51.2 (45)
8 : 32	24680.0 (800)	192282.3 (2271)	1175577.1 (4615)	194.8 (87)	210.2 (94)	667.6 (176)
16 : 32	495.8 (116)	4755.1 (418)	25597.1 (887)	15.6 (25)	51.8 (54)	80.3 (62)
16 : 64	14318.7 (572)	135196.6 (1765)	835783.1 (4108)	114.8 (65)	248.6 (94)	952.9 (170)

Table 3.11: (PC_σ), (PC_τ) not enforced. (P4), $q = 3$, CPU runtimes

	NOV PCG: CPU runtimes			OV PCG: CPU runtimes		
$\frac{1}{H} : \frac{1}{h}$	100	1000	10000	100	1000	10000
4 : 8	0.220	0.435	0.692	0.122	0.132	0.135
4 : 16	4.571	11.273	22.289	0.977	1.197	1.414
8 : 16	1.042	2.339	5.214	0.417	0.478	0.570
8 : 32	18.597	52.698	113.222	4.634	4.933	8.162
16 : 32	3.802	11.847	24.674	2.317	3.545	3.918
16 : 64	81.605	255.922	739.614	38.797	46.129	64.606

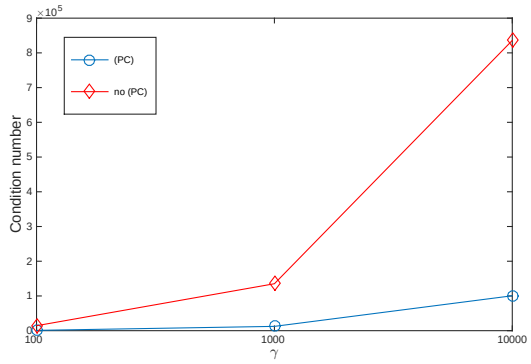


Figure 3.10: (P4), NOV. Effect of (PC_σ), (PC_τ) for κ, $q = 3$

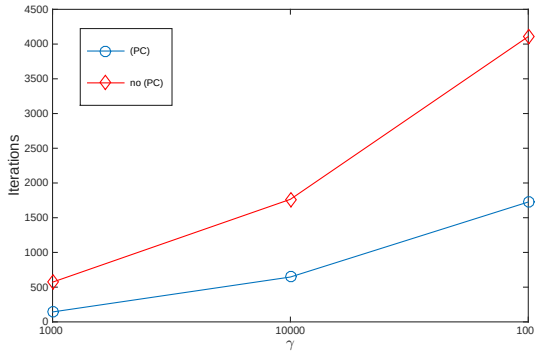


Figure 3.11: (P4), NOV. Effect of (PC_σ), (PC_τ) for iteration counts, $q = 3$

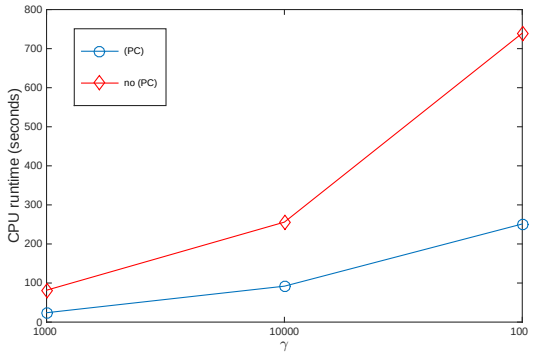


Figure 3.12: (P4), NOV. Effect of (PC_σ), (PC_τ) for CPU runtimes, $q = 3$

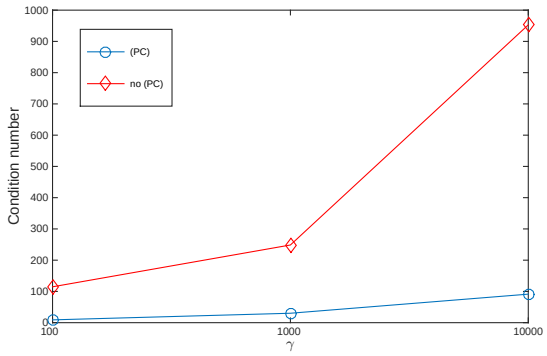


Figure 3.13: (P4), OV. Effect of (PC_σ), (PC_τ) for κ estimates, $q = 3$

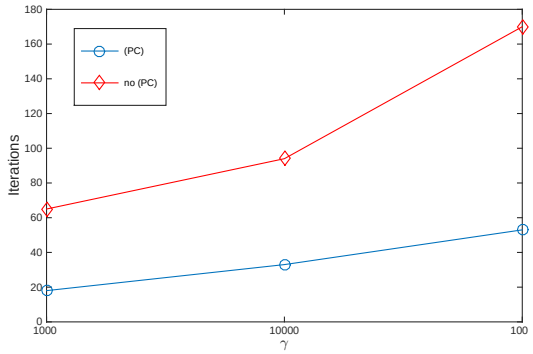


Figure 3.14: (P4), OV. Effect of (PC_σ), (PC_τ) for iteration counts, $q = 3$

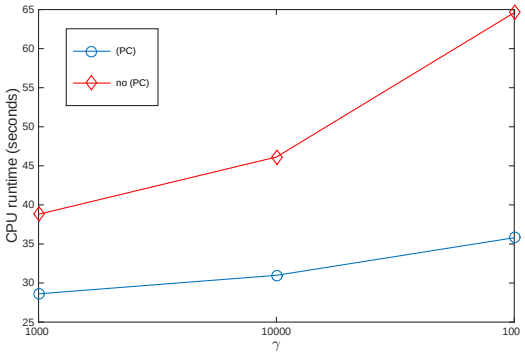


Figure 3.15: (P4), OV. Effect of (PC_σ), (PC_τ) for CPU runtimes, $q = 3$

Table 3.12: (PC_σ), (PC_τ) not enforced. (P5), $q = 3$, κ estimates (iterations)

	Nonoverlapping PCG: $\kappa(BA)$, (iterations)			Overlapping PCG: $\kappa(BA)$, (iterations)		
$\frac{1}{H} : \frac{1}{h}$	100	1000	10000	100	1000	10000
4 : 8	934.4 (238)	4426.0 (523)	15635.8 (829)	25.9 (41)	37.2 (48)	40.6 (50)
4 : 16	29858.6 (1375)	200943.1 (2901)	1279916.8 (7168)	181.9 (114)	272.6 (136)	350.4 (167)
8 : 16	1294.1 (279)	7685.1 (666)	358127 (1412)	21.3 (38)	35.8 (53)	52.0 (57)
8 : 32	36751.0 (1464)	226436.3 (2915)	1523513.7 (7185)	146.7 (102)	186.2 (121)	760.5 (246)
16 : 32	1016.5 (237)	7152.1 (596)	46286.0 (1918)	18.3 (35)	28.1 (41)	84.0 (83)
16 : 64	22477.7 (1012)	182209.9 (2758)	1608144.1 (8798)	154.8 (96)	345.45 (156)	890.0 (237)

Table 3.13: (PC_σ), (PC_τ) not enforced. (P5), $q = 3$, CPU runtimes

	NOV PCG: CPU runtimes			OV PCG: CPU runtimes		
$\frac{1}{H} : \frac{1}{h}$	100	1000	10000	100	1000	10000
4 : 8	0.256	0.514	0.841	0.115	0.128	0.132
4 : 16	6.509	13.618	33.244	1.186	1.306	1.558
8 : 16	1.596	3.712	7.539	0.514	0.650	0.706
8 : 32	33.761	68.446	180.873	5.411	6.067	10.997
16 : 32	7.038	16.642	52.824	2.776	3.076	4.920
16 : 64	147.642	403.967	1404.899	47.747	62.636	82.875

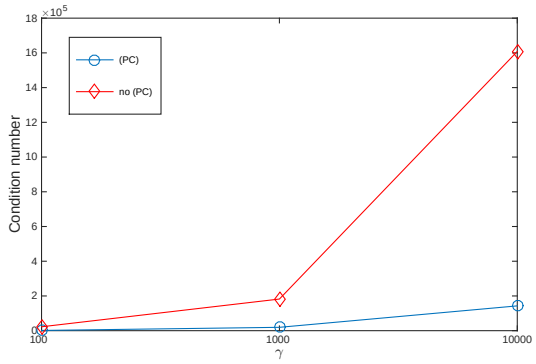


Figure 3.16: (P5), NOV. Effect of (PC_σ), (PC_τ) for κ, $q = 3$

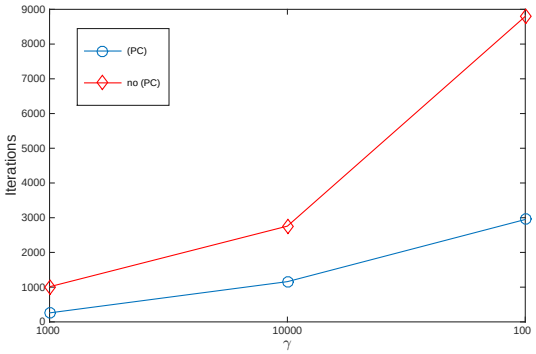


Figure 3.17: (P5), NOV. Effect of (PC_σ), (PC_τ) for iteration counts, $q = 3$

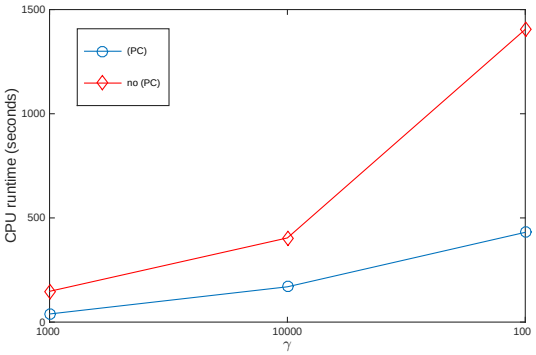


Figure 3.18: (P5), NOV. Effect of (PC_σ), (PC_τ) for CPU runtimes, $q = 3$

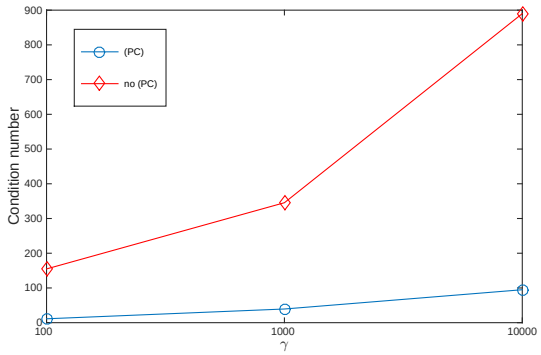


Figure 3.19: (P5), OV. Effect of (PC_σ), (PC_τ) for κ estimates, $q = 3$

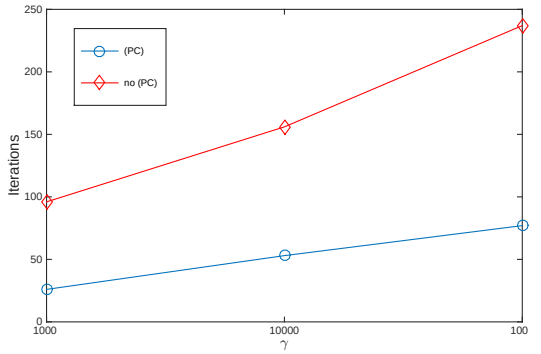


Figure 3.20: (P5), OV. Effect of (PC_σ), (PC_τ) for iteration counts, $q = 3$

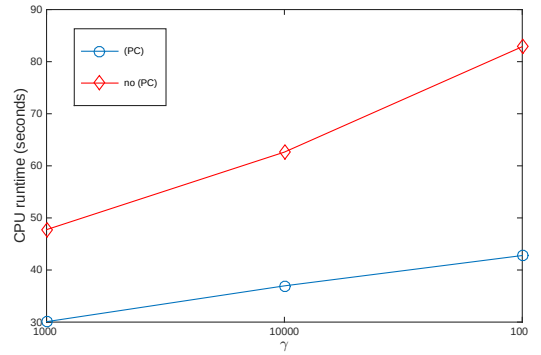


Figure 3.21: (P5), OV. Effect of (PC_σ), (PC_τ) for CPU runtimes, $q = 3$

Table 3.14: (PC_σ), (PC_τ) not enforced. (P6), $q = 3$, κ estimates (iterations)

	Nonoverlapping PCG: $\kappa(BA)$, (iterations)			Overlapping PCG: $\kappa(BA)$, (iterations)		
$\frac{1}{H} : \frac{1}{h}$	100	1000	10000	100	1000	10000
4 : 8	974.2 (284)	4942.7 (679)	13396.3 (1125)	21.5 (45)	39.8 (65)	59.7 (82)
4 : 16	20667.8 (1353)	106013.6 (2742)	379582.4 (4810)	198.9 (140)	349.8 (203)	601.1 (276)
8 : 16	737.7 (251)	5185.3 (701)	18897.9 (1398)	15.0 (38)	28.4 (53)	63.8 (81)
8 : 32	2085.8 (1361)	114302.2 (2939)	623610.8 (6952)	146.2 (124)	168.2 (140)	494.6 (238)
16 : 32	643.6 (230)	4699.4 (660)	21275.8 (1443)	14.0 (36)	21.4 (46)	54.5 (78)
16 : 64	21340.7 (1354)	159645.5 (3908)	823775.6 (8466)	99.8 (105)	117.5 (116)	523.8 (282)

Table 3.15: (PC_σ), (PC_τ) not enforced. (P6), $q = 3$, CPU runtimes

	NOV PCG: CPU runtimes			OV PCG: CPU runtimes		
$\frac{1}{H} : \frac{1}{h}$	100	1000	10000	100	1000	10000
4 : 8	0.254	0.590	0.947	0.083	0.115	0.151
4 : 16	4.855	9.590	16.582	0.974	1.320	1.795
8 : 16	1.094	2.732	5.704	0.370	0.467	0.596
8 : 32	23.292	49.692	122.698	4.194	4.660	7.357
16 : 32	4.875	13.080	28.165	1.843	2.135	3.104
16 : 64	144.640	426.047	932.504	32.468	34.453	62.891

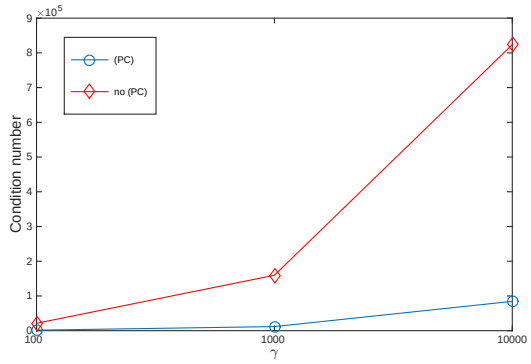


Figure 3.22: (P6), NOV. Effect of (PC_σ), (PC_τ) for κ, $q = 3$

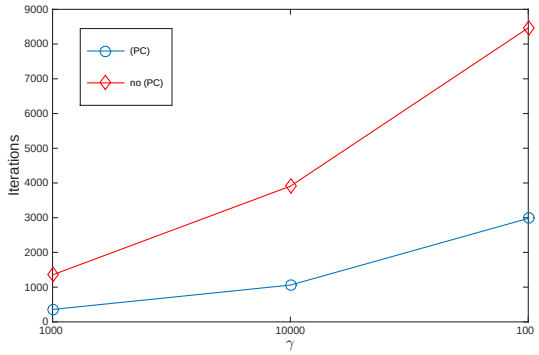


Figure 3.23: (P6), NOV. Effect of (PC_σ), (PC_τ) for iteration counts, $q = 3$

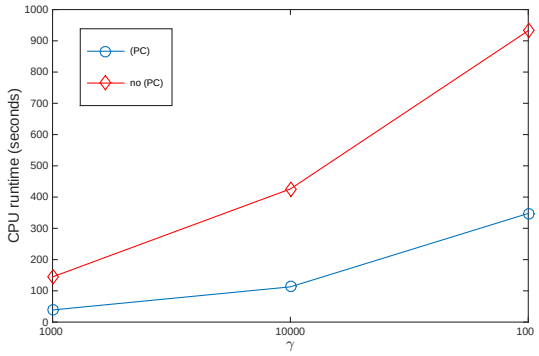


Figure 3.24: (P6), NOV. Effect of (PC_σ), (PC_τ) for CPU runtimes, $q = 3$

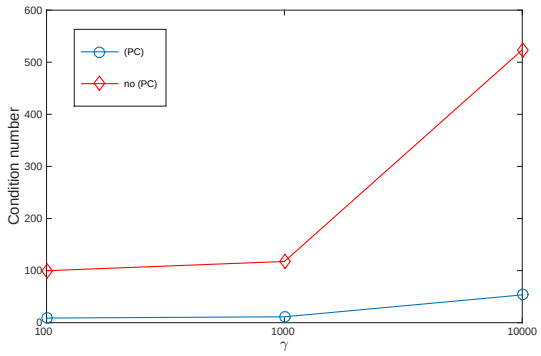


Figure 3.25: (P6), OV. Effect of (PC_σ), (PC_τ) for κ estimates, $q = 3$

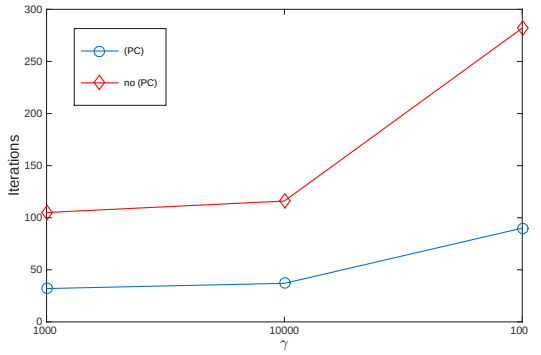


Figure 3.26: (P6), OV. Effect of (PC_σ), (PC_τ) for iteration counts, $q = 3$

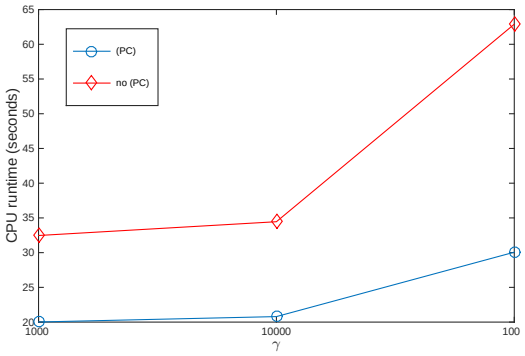


Figure 3.27: (P6), OV. Effect of (PC_σ), (PC_τ) for CPU runtimes, $q = 3$

Chapter 4

Computational Implementation

4.1 Introduction

This chapter is intended to provide details about the implementation of the methods described in previous chapters. The primary goal is to design programs capable of assembling and solving the linear system (2.13) corresponding to the discontinuous Galerkin formulation of model problems (2.1)-(2.3) and (3.1)-(3.3) using both the traditional conjugate gradient method as well as our preconditioned CG methods. The code supporting the experiments is written using the `C` programming language. One reason for this decision is the accessibility of dynamic memory allocation inherent in `C`. Another reason is that much of the foundational code from which this program is built was also written in `C`, making it a natural choice.

The implementation of these methods necessarily contains a lot of moving parts. We adhere as much as possible to a modular design, which has several benefits:

- Significant portions of the code may be reused for both the second- and fourth-order experiments
- Testing different aspects of the theory does not necessitate a complete rewrite of the code

- The myriad errors that invariably materialize are much easier to track down

Figure 4.1 illustrates the primary components of the main algorithm. The initialization phase is comprised of storing information about the initial mesh $\mathcal{T}_H$, storing various parameters used throughout the run, and loading quadrature data for subsequent use. The next phase involves the computation of local stiffness matrices and load vectors. The initial mesh is then refined a predetermined number of times, creating a hierarchy of meshes, with the same local information computed at each level. Once the mesh has been refined to the fine level mesh $\mathcal{T}_h$, the program proceeds to the solution phase. If the standard conjugate gradient method is the solver of choice, the solution is computed, the error and information about the run are displayed, and the process is finished. Otherwise, we construct a preconditioner using domain decomposition and then solve using PCG.

The remaining sections in this chapter will describe this process in greater detail, including the data objects and structures used, the refinement process, and various aspects of domain decomposition used to create the preconditioner.

4.2 Initialization

Only two aspects of this phase require further discussion: the quadrature template files and the creation of the initial mesh representation. The other tasks involve defining parameters used to control the run and allocating memory for certain data structures that will be discussed in a subsequent section.

4.2.1 Numerical Quadrature

Creating the discontinuous Galerkin formulation requires the computation of integrals over both cells and edges; in our case, two- and one-dimensional integrals, respectively. Rather than compute these integrals exactly, they are approximated using Gaussian quadrature. In general, Gaussian quadrature requires a collection of quadrature points

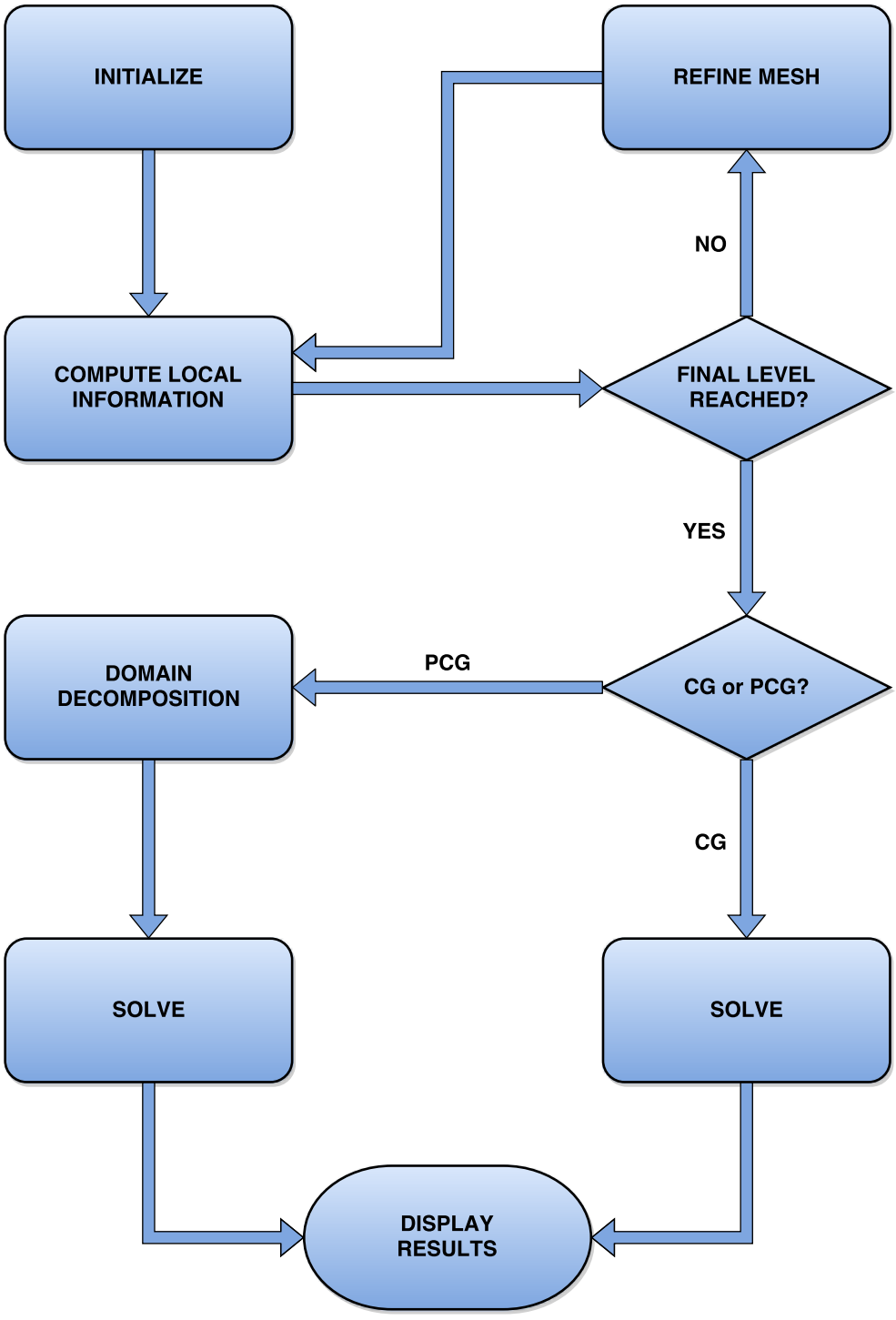


Figure 4.1: Diagram of Main Algorithm.

along with corresponding weights over the domain of integration. In our case, we use suitable points and weights on the reference element $\hat{K}$ along with an affine transformation to compute these approximations.

At the beginning of the run, we load quadrature templates that store these points and weights on $\hat{K}$, as well as the values of basis functions at each point. This simplifies the construction of local stiffness matrices and load vectors considerably.

4.2.2 Mesh Generation

The experiments conducted for the purpose of this thesis are performed with a "nice" two-dimensional domain (either the unit square or the L-shaped domain used in test problem (P3)) and a structured partition. This is purely for simplicity. The code is designed for compatibility with the package `Triangle`, a two-dimensional quality mesh generator and Delaunay triangulator created by Jonathan Shewchuk [40]. This allows the input of exotic domains and/or highly unstructured partitions, so long as the input is in a format compatuble with `Triangle`. All initial meshes used for the experiments described herein were input using this format.

4.3 Data Structures

Information about the initial mesh is given in terms of components of the cells; namely, *vertices, edges*, and *triangles*. Since keeping track of the geometry of these objects is essential, each was implemented in the code as a data structure. In particular, these objects were created as *structs*, a user-defined data type in `C` that is used to group related information of different data types. In addition to geometric information, each of these structures will also provide access to other information that will be used during the course of the run, such as PDE data or subdomain information. In general, information more substantive than an integer is accessed via a pointer in order to maintain a reasonable size for the structs.

```
typedef struct vertstruct {
  int IDnum;              // global ID number of node
  double x;               // x-coordinate of node
  double y;               // y-coordinate of node
  short bmark;            // boundary marker: 0-interior, 1-boundary
} VERT;
```

Figure 4.2: Vertex data structure.

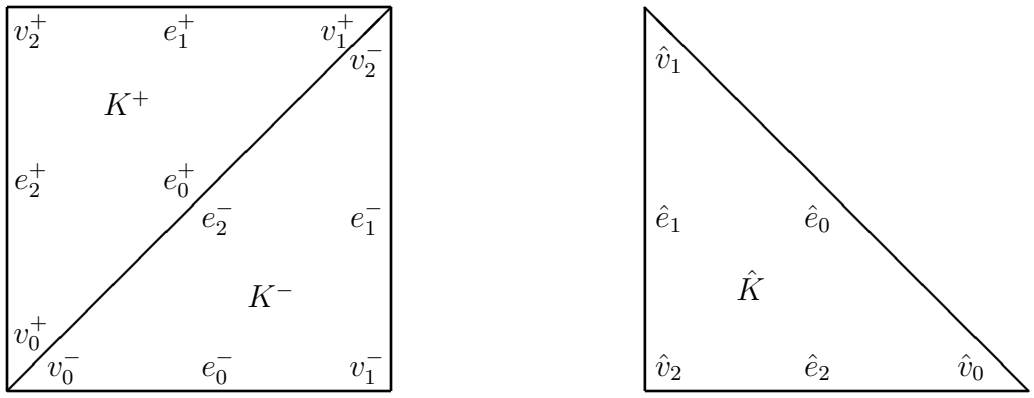
In order to keep track of the entire mesh and subsequent refinements, we will utilize *tree structures* to keep track of both the triangle and the edge hierarchies. The details of this implementation will be discussed in Section 4.4.

4.3.1 VERT structs

The struct `VERT` (Figure 4.2) is used to contain information about the vertices of cells in the mesh. In particular, the fields of this struct contain the x- and y-coordinates of the vertex, an identification number, and a marker to indicate if the vertex lies on the boundary. The identification number is used primarily for outputting a node file to `Triangle` in order to produce a quick visualization of a given mesh. This is done using the program `ShowMe`, which is included with the `Triangle` package.

4.3.2 EDGE structs

The struct `EDGE` (Figure 4.4) is a repository for all the necessary information about the edges of cells in the mesh. The purpose of many of the fields in this struct are self-explanatory; those that are not deserve some additional details. The field `meshlvl` is used to store the mesh level at which the edge is created, while the field `leaf` indicates if the edge is currently a leaf on the edge tree. During the refinement process, if a cell is marked for refinement, then its edges will also be refined; the field `refine_flag` indicates the edge is to be refined. The field `midpt` is initialized as a `NULL` pointer. If the edge is refined, then `midpt` will point to a `VERT` struct representing the midpoint of



(i) K^+, K^- edge enumeration

(ii) $\hat{K}$ edge enumeration

Figure 4.3: Local edge enumeration.

the edge. The field `enode` is a pointer from the `EDGE` struct back to the corresponding tree node. This pointer is used during the refinement process.

As mentioned previously, every interior edge e is shared by two triangles, K^+ and K^-, whose designation relative to the edge is arbitrary. The pointers `Kplus` and `Kminus` point to the `TRIANGLE` structs representing these cells. In addition, each edge e of a cell K is mapped to a corresponding edge $\hat{e}$ of the reference element $\hat{K}$. The field `Kploc` designates the edge number of $\hat{e}$ corresponding to the edge's position in the cell K^+. The field `Kmloc` plays a similar role for the edge's position in the cell K^-. See Figure 4.3 for a visual representation of this enumeration.

The fields `GData` and `EBlock` are pointers to structures containing geometric information about the edge and PDE data, respectively. In particular, `GData` points to the struct `EGDATA` (Figure 4.5), which contains the length of the edge and the components of the normal vector. The pointer `EBlock` points to the struct `EBLOCK` (Figure 4.6), which holds the off-diagonal edge matrix that will be discussed in Appendix B.

Finally, the integer array `sd_marks` is an array of flags used to indicate if the edge belongs to the boundary of a given subdomain. These flags are used both during the initial subdomain assignments and also reassigned when constructing overlapping

```
typedef struct edgestruct {
  int IDnum;                        // global ID
  unsigned meshlvl;                 // current mesh level
  unsigned bmark;                   // 0-int, 1-Dir, 2-Neu
  unsigned leaf;                    // 1: edge is a leaf
  unsigned isnew;                   // 1: edge is new
  unsigned refine_flag;             // 1: marked for refinement
  unsigned index;                   // used during initialization
  VERT *endpt[2];                   // Endpoints
  VERT *midpt;                      // Midpoint
  struct tree_edgenode *enode;      // pointer to tree node
  struct tristruct *Kplus;          // pointer to K+ element
  struct tristruct *Kminus;         // pointer to K- element
  unsigned Kploc;                   // local edge # (relative to K+)
  unsigned Kmloc;                   // local edge # (relative to K-)
  struct edge_geom_data *GData;     // storage for edge geometric info
  struct edge_storage *EBlock;      // off-diagonal edge matrix
  int *sd_marks;                    // bdry mark for each subdomain
} EDGE;
```

Figure 4.4: Edge data structure.

```
typedef struct edge_geom_data {
  double length;                    // length of edge
  double normal[2];                 // components of normal vector
} EGDATA;
```

Figure 4.5: Edge geometric data.

```
typedef struct edge_storage {
  double *off;                    // off-diagonal edge array
} EBLOCK;
```

Figure 4.6: Edge PDE data.

subdomains. This process will be described further in the sections detailing the contruction of the Schwarz preconditioners.

4.3.3 TRIANGLE structs

The struct `TRIANGLE` (Figure 4.7) is used to link all the important information about the cells in the mesh. As with the `EDGE` struct, only the fields which are not self-explanatory will be addressed.

The fields `meshlvl`, `leaf`, `refine_flag`, and `knode` all function in an way analogous to their `EDGE` counterparts. The field `nbors` is an array of pointers to those triangles that share an edge with the current triangle. This field is used for a particular type of subdomain designation that will be discussed later.

The fields `GData` and `KBlock` are pointers to structures containing geometric information about the triangle and PDE data, respectively. In particular, `GData` points to the struct `KGDATA` (Figure 4.8), which contains the area of the traingle, a `VERT` struct representing the barycenter of the triangle, and the affine transformation from the physical element K to the reference triangle $\hat{K}$. The pointer `KBlock` points to the struct `KBLOCK` (Figure 4.6), which holds the local stiffness matrix, the local load vector, and the local solution vector, all of which will be discussed in greater detail in Appendix **??**.

Finally, the fields `sd_list_orders` and `sd_incls` designate subdomain information about the triangle. `sd_list_orders` is an integer vector which indicates the triangle's numerical order within a given subdomain, while `sd_incls` is a vector of flags to indicate if the triangle belongs to a particular subdomain.

4.4 Mesh Refinement

The experiments conducted in this thesis all utilize *uniform* refinement; i.e., all triangles are refined in order to generate the next level in the mesh hierarchy. The code is designed in such a way that *adaptive* refinement is relatively straightforward to implement; however, this is a topic for future study. The refinement of a triangle is done by *quadrisection*. In other words, each triangle is subdivided into four smaller

```
typedef struct tristruct {
  int IDnum;                          // global ID
  unsigned meshlvl;                   // current mesh level
  unsigned leaf;                      // 1: element is a leaf
  unsigned isnew;                     // 1: element is new
  unsigned isrefined;                 // 1: element has been refined
  unsigned refine_flag;               // 1: marked for refinement
  struct vertstruct *verts[3];        // Vertices
  struct edgestruct *edges[3];        // Edges
  struct tristruct *nbors[3];         // Neighbors
  struct tree_elemnode *knode;        // pointer back to tree node
  struct elem_storage *KBlock;        // storage for arrays
  struct elem_geom_data *GData;       // storage for geometric info
  int *sd_list_orders;                // subdomain list order
  int *sd_incls;                      // subdomain inclusions
} TRIANGLE;
```

Figure 4.7: Triangle data structure.

```
typedef struct elem_geom_data {
  double area;                          // area of triangle
  struct vertstruct baryctr;            // coordinates of barycenter
  double aff_mat[2][2];                 // maps K to Khat
} KGDATA;
```

Figure 4.8: Triangle geometric data.

```
typedef struct elem_storage {
  double *sdb;               // lower triangular local stiff. matrix
  double *rhs;               // local RHS vector
  double *x;                 // local solution vector
} KBLOCK;
```

Figure 4.9: Triangle PDE data.

triangles by connecting the midpoints of opposite edges (see Figure 4.10). The corresponding edge refinement constitutes a *bisection*.

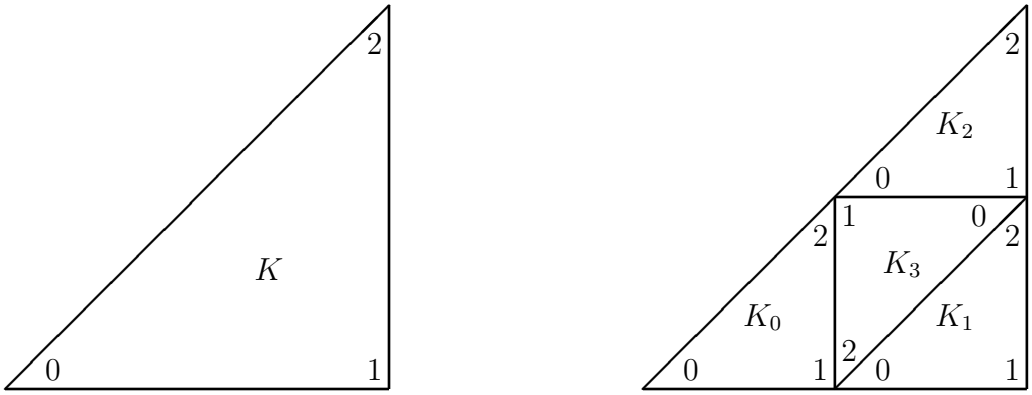


Figure 4.10: Refinement by quadrisection and subsequent vertex enumeration.

Recall that tree structures are used to maintain both the triangle and the edge hierarchies. The choice of refinement implemented in this program implies that the the triangle and edge trees will be *quadtrees* and *binary* trees, respectively. To be precise, each triangle in the initial mesh is the root node of a quadtree (see Figure 4.11) and each edge in the initial mesh is the root of a binary tree.

When constructing the local stiffness matrices and load vectors for a particular mesh, it is necessary to loop over the triangles and edges belonging to that mesh. In order to avoid performing a series of tree traversals, the collections of triangles and edges belonging to a particular mesh are collected into *doubly-linked lists* for easier access.

The mesh hierarchy is maintained by creating two pointer arrays for each refinement level. The array `klists` contains pointers for triangle doubly-linked lists, where each list identifies the triangles belonging to that particular mesh (see Figure 4.12). The array `elists` performs a similar function for the edges.

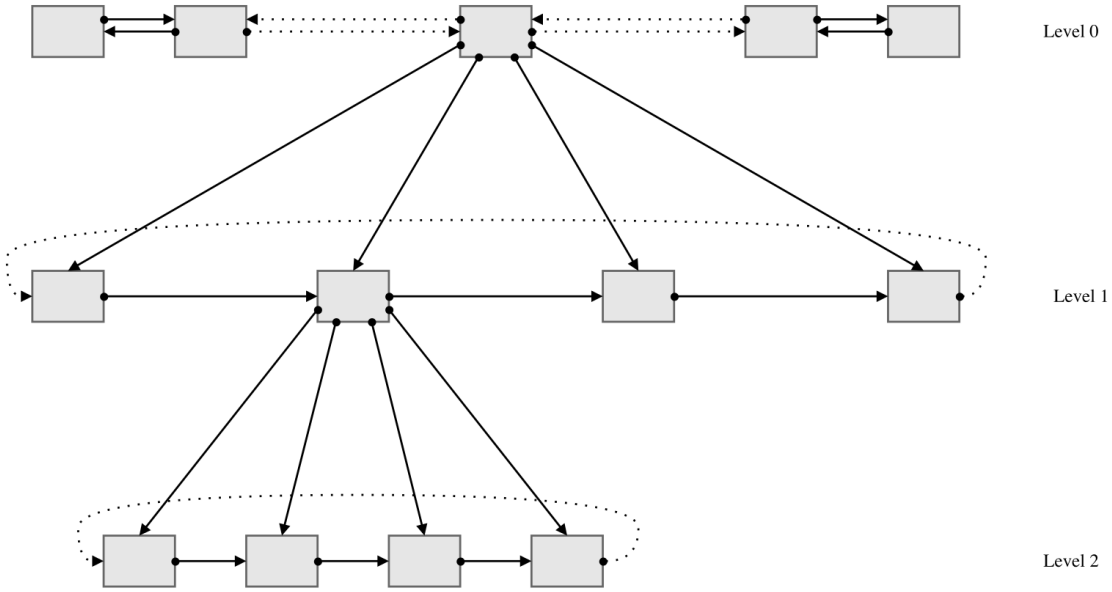


Figure 4.11: Triangle tree structure.

4.5 Computation of Local Information

At each level of refinement the local stiffness matrix and load vector is computed for each cell. In addition, the matrix representing interaction between adjacent cells – i.e., the "off-diagonal" matrix associated with each interior edge – is also computed. The stiffness matrix corresponding to each triangle is always symmetric, so we only store the lower triangular part. This array is stored in the struct `KBLOCK` in the field `sdb`. The local load vector is also stored in `KBLOCK`, in the field `rhs`. Finally, the off-diagonal block corresponding to each interior edge is stored in `EBLOCK` in the field `off`.

The details for the construction of these objects requires a rather lengthy exposition; therefore, they will be deferred to Appendix B, both for the second order problem and the fourth-order problem.

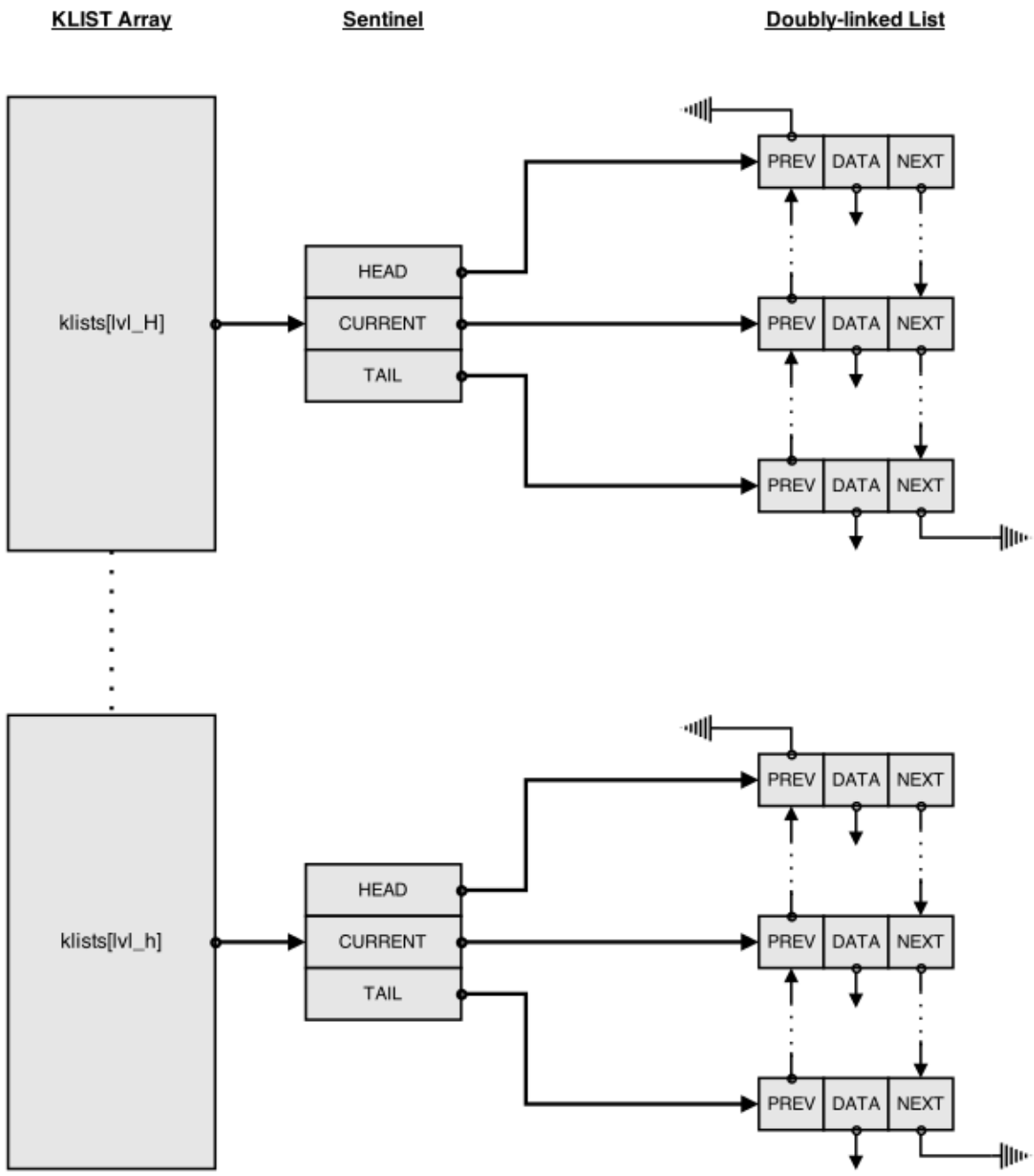


Figure 4.12: Diagram of KLIST array.

4.6 Solving the Linear System

In general, methods for solving the linear system (2.13) or the preconditioned system (2.14) fall into two categories: direct methods or iterative methods. Systems

generated by the discontinuous Galerkin method tend to be substantially larger than those arising from other methods due to the increased number of degrees of freedom; consequently, direct methods are generally much too costly. On the other hand, the SIPG formulation created here generates a system that is symmetric and positive definite (SPD). For problems of this type, the solver most often utilized is the Conjugate Gradient (CG) method. As highlighted in the current situation, if the condition number of the system is large, then convergence of CG can be too slow for practical application. In this scenario, the use of a preconditioner to improve the conditioning of the system is a common practice. The Preconditioned Conjugate Gradient (PCG) method can be substaintially more efficient than standard CG.

In many methods, the structure of the linear system generated by the discretization requires the formation of a global stiffness matrix and corresponding load vector. One very nice property of the discontinuous Galerkin method is the sparse block structure inherent in the stiffness matrix. This feature can be exploited to avoid the formation of a global stiffness matrix; for example, matrix-vector multiplications may be performed without creating the global matrix at all. Instead, such a product is calculated by incorporating contributions from local blocks, each corresponding to a geometric entity such as an interior edge or a cell of a particular mesh. These blocks correspond to the local stiffness matrices and load vectors described in 4.5.

4.6.1 Conjugate Gradient Method

The Conjugate Gradient method was introduced by Hestenes and Stiefel in 1952 [28]. The pseudocode for the CG method as implemented in these experiments is given by Algorithm 1. If all operations in the algorithm are performed using exact arithmetic, then convergence is guaranteed in a number of iterations at most equal to the dimension of the system. However, since all computations are perfomed using floating point arithmetic, the rate of convergence is determined by (2.102), as mentioned in Section 2.

Algorithm 1 Conjugate Gradient (CG)

Require: Tolerance tol, initial vector x_0, $itermax$, number of dof N

1. $r = b - Ax_0$
2. **for** $(i = 1; i \leq itermax, i{+}{+})$ **do**
3. $\quad \rho_{\text{new}} = r^T r$
4. $\quad err \ = (\rho_{\text{new}}/N)^{1/2}$
5. $\quad$ **if** $(err < tol)$ **then**
6. $\quad\quad$ `break`
7. $\quad$ **end if**
8. $\quad$ **if** $(i = 1)$ **then**
9. $\quad\quad d = r$
10. $\quad$ **else**
11. $\quad\quad \beta = \rho_{\text{new}}/\rho_{\text{old}}$
12. $\quad\quad d = r + \beta d$
13. $\quad$ **end if**
14. $\quad$ **if** $(d^T Ad \neq 0)$ **then**
15. $\quad\quad \alpha = \rho_{\text{new}}/(d^T Ad)$
16. $\quad$ **end if**
17. $\quad x = x + \alpha d$
18. $\quad r = r - \alpha Ad$
19. $\quad \rho_{\text{old}} = \rho_{\text{new}}$
20. **end for**

4.6.2 Preconditioned Conjugate Gradient Method

The preconditioners constructed for this work correspond to the two-level additive Schwarz methods developed in Chapters 2 and 3. The general pseudocode for the implementation of PCG used here is given by Algorithm 2. The application of the preconditioner during the process is handled by the routine `ASprecond`. This routine allows for the choice of either nonoverlapping or overlapping Schwarz preconditioning, as described in earlier chapters. The details of each implementation will be provided in the following two sections.

Algorithm 2 Preconditioned Conjugate Gradient (PCG)

Require: Tolerance tol, initial vector x_0, $itermax$, number of dof N
Require: Linked lists $klists$, $elists$, $SDcells$, $SDedges$
Require: Coarse level H, fine level h

1. $r = b - Ax_0$
2. **for** $(i = 1; i \leq itermax, i\texttt{++})$ **do**
3. $\quad z = 0$
4. $\quad z = \texttt{ASprecond}(klists, elists, SDcells, SDedges, x_0, H, h, r)$
5. $\quad \rho_{\text{new}} = r^T r$
6. $\quad \rho_z = z^T r$
7. $\quad err \ = \ (\rho_{\text{new}}/N)^{1/2}$
8. $\quad$ **if** $(err < tol)$ **then**
9. $\quad\quad$ `break`
10. $\quad$ **end if**
11. $\quad$ **if** $(i = 1)$ **then**
12. $\quad\quad d = z$
13. $\quad$ **else**
14. $\quad\quad \beta = \rho_z/\rho_{\text{old}}$
15. $\quad\quad d = z + \beta d$
16. $\quad$ **end if**
17. $\quad$ **if** $(d^T Ad \neq 0)$ **then**
18. $\quad\quad \alpha = \rho_z/(d^T Ad)$
19. $\quad$ **end if**
20. $\quad x = x + \alpha d$
21. $\quad r = r - \alpha Ad$
22. $\quad \rho_{\text{old}} = \rho_z$
23. **end for**

4.7 Two-Level Additive Schwarz Preconditioners

The creation of both the nonoverlapping and overlapping Schwarz preconditioners follows the design paradigm for maintaining the mesh hierarchy. As described previously, for each mesh level linked lists are created to represent all the cells and edges belonging to that particular level. A similar approach is adopted for representing the subdomain and coarse mesh information necessary to construct the Schwarz preconditioners. In particular, all cells and edges needed to create the

subdomain solvers are collected in linked lists for ease of access, as are the coarse mesh components.

Recall from (2.51) that the structure of the additive Schwarz preconditioner is given by

$$B = R_0^T A_0^{-1} R_0 + R_1^T A_1^{-1} R_1 + \cdots + R_p^T A_p^{-1} R_p,$$

where A_i is the stiffness matrix corresponding to $a_i(\cdot,\cdot)$ and R_i^T is the matrix representation of the embedding $V_i^h \to V^h$, $i = 0, 1, \ldots, p$. In the following sections, the details of constructing the stiffness matrices A_i for both the nonoverlapping and the overlapping methods will be presented. In addition, the details for constructing the embedding operators R_i^T will be shown.

4.7.1 Components of the Nonoverlapping Preconditioner

Once the local information has been computed and stored for each level, the primary task necessary to construct the preconditioner is essentially one of bookkeeping. Note that the information needed to contruct the coarse grid solver is collected and stored at the very beginning of the program – this is precisely the initial cell list and initial edge list. Thus, the remaining step in the construction of the nonoverlapping preconditioner is to assign a subdomain for each cell K and edge e_h of the fine mesh $\mathcal{T}_h$.

The primary domain used for the tests described herein is the unit square $\Omega = [0,1] \times [0,1]$, so a natural method for subdomain assignment is to create a $m \times m$ grid and designate subdomain inclusion for a particular triangle by the coordinates of its barycenter. In other words, if the (x, y)-coordinates of the barycenter of K fall within the bounds $(x_j, x_{j+1}) \times (y_k, y_{k+1})$ of a particular subdomain Ω_i, then K belongs to Ω_i (see Figure 4.13).

The routine `createInitSD(klist,SDcells)` assigns each fine level cell K in the list `klist` to a subdomain based on the criteria described above. The i^{th}-entry in the

vector `SDcells` is a pointer to the sentinel of a linked list that represents the cells in subdomain Ω_i.

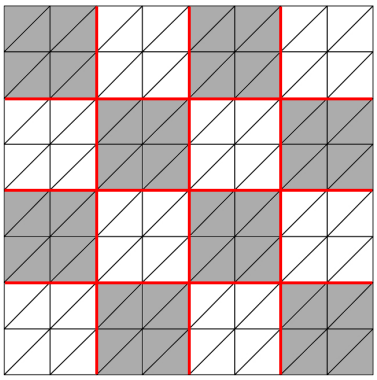


Figure 4.13: Initial subdomain partition.

The next step in the construction of the nonoverlapping preconditioner deals with interior edges. Note that boundary edges play no role in the creation of the local subdomain solvers; instead, they appear in the coarse grid solver. First, those interior edges belonging to the boundary of a subdomain $\mathcal{S}_i$ are grouped into linked lists, with each list corresponding to a subdomain Ω_i (recall the definition of the skeleton from Section 2.4.1 as the union of all such subdomain boundary edges). For a nonoverlapping subdomain partition (in which the subdomains are disjoint), such edges are straightforward to identify – those interior edges for which K^+ and K^- belong to different subdomains. If K^+ and K^- belong to the same subdomain, then that edge belongs to the interior of the corresponding subdomain.

The routine `assignEdges(SDcells,skel,SDedges)` loops over the subdomains and tests the edges of each cell. If the edge belongs to the boundary of Ω_i it is added to the linked list `skel[i]` (in Figure 4.13, note that the edges belonging to the skeleton are marked in red). If the edge does not belong to the skeleton, then it belongs to the current subdomain Ω_i and is added to the corresponding edge list `SDedges[i]`.

4.7.2 Components of the Overlapping Preconditioner

As mentioned in Remark 2.4.2 (1), a practical way to generate an overlapping subdomain partition is to begin with a nonoverlapping partition that is aligned with $\mathcal{T}_H$ and then add layers of cells from $\mathcal{T}_h$. This is the approach implemented in these experiments.

As in the nonoverlapping case, the process of constructing the overlapping preconditioner begins by assigning a subdomain for each cell $K \in \mathcal{T}_h$. This is again accomplished via the routine `createInitSD`, with the corresponding subdomain lists stored in `SDcells`.

The next phase involves creating the skeleton for the nonoverlapping partition. At this point, it is unnecessary to assign a subdomain to the remaining edges, so the routine `createSkeleton(SDcells,skel)` is called to store those edges belonging to the skeleton in the vector `skel` (note that each entry in `skel` is a pointer to a linked list). These subdomain boundaries $\mathcal{S}_i$ are used to determine which cells will be appended to a given subdomain, thus creating the extra "layer" described previously.

The routine `collectOverlap(SDcells,skel)` is used to create this layer of overlap. Recall that, at this point, the cells in a given subdomain Ω_i are stored in the list `SDcells[i]`. The concept behind the routine `collectOverlap` is to loop over all the cells and determine if a cell has a vertex that lies on the boundary $\mathcal{S}_i$ of a given subdomain Ω_i. If this is the case, then that cell is appended to the list corresponding to Ω_i. Figure 4.14 shows this process, beginning with a nonoverlapping partition, then the process of extending a given subdomain using the method described.

At this point the overlapping method creates a new difficulty. For the nonoverlapping case, a cell K can belong to at most a single subdomain Ω_i. In the overlapping case, this is no longer true. The field `sd_incls` found in struct `TRIANGLE` is used to keep track of the subdomains Ω_i to which a particular cell K belongs. `sd_incls` is simply a vector of integers with length equal to the number of subdomains whose entries are used as flags – 0 or 1 – to indicate subdomain inclusion. Note that

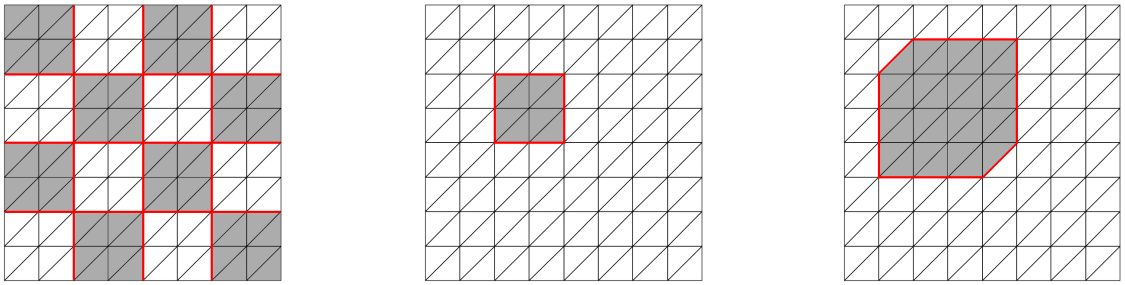


Figure 4.14: Construction of an overlapping partition.

the primary bookkeeping of subdomain inclusion is `SDcells` – each subdomain list contains all cells belonging to that subdomain. The use for the `sd_incls` arises during subsequent phases of the construction.

With the subdomain inclusions for each cell stored, the next stage of building the overlapping conditioner is determining the status of each interior edge. Contrasting with the nonoverlapping case, the issue is that an edge may belong to the interior of multiple subdomains while potentially also belonging to the boundary of another subdomain. As with the cells, this difficulty is dealt with by a vector of integer flags to indicate the status of an edge with respect to a particular subdomain. The field `sd_marks` located in the struct `EDGE` is utilized for this task. `sd_marks[i]` is either 0 to indicate the edge belongs to the interior of Ω_i, or 1 to indicate the edge belongs to $\mathcal{S}_i$. Again, `sd_marks` serves only an auxiliary purpose – the primary bookkeeping of subdomain inclusion for edges will be `SDedges`, the construction of which will be described below.

To designate the status of each interior edge, the routine `markSD_bdry(SDcells)` is called. The process works much as the routine `assignEdges`, in that if the cells K^+ and K^- corresponding to a given edge belong to different subdomains, then that edge will belong to the boundary of that subdomain. This routine loops over the edges of the cells belonging to each subdomain Ω_i and tests if `sd_incls[i]` for K^+ is different from `sd_incls[i]` for K^-. If so, `sd_marks[i]` is set to 1; otherwise, it is set to 0.

Finally, the routine `gatherSDedges(SDcells,SDedges)` collects the edges interior to each subdomain into a list, based on the status of `sd_marks`. Analagous to the nonoverlapping version, `SDedges` is a vector of pointers, each of which points to the list containing the interior edges relative to the corresponding subdomain.

4.7.3 Restriction and Prolongation Operators

The final pieces necessary to construct the additive Schwarz preconditioners are the prolongation and restriction operators. From (2.51), it is clear that global data must be restricted to each subspace V_i^h, solved locally, then prolongated (or embedded) back into the global space V^h, for $i = 0, 1, \ldots, p$ (note that this process for $i = 0$ corresponds to the coarse-grid correction).

Let $R_i : V^h \to V_i^h$, for $i = 0, 1, \ldots, p$ denote the matrix representation of the restriction operator from the global approximation space to the coarse subspace and the subspaces corresponding to the subdomains. Choose $R_i^T : V_i^h \to V^h$, for $i = 0, 1, \ldots, p$ as the corresponding prolongation operator. The construction of the prolongation operator will now be discussed.

Denote by $\hat{K}_0, \hat{K}_1, \hat{K}_2$, and $\hat{K}_3$ the four children of the reference element obtained by quadrisection. Let

$$\{\varphi_j^{\hat{K}}\}, \qquad j = 0, \ldots, n-1$$

be the n basis functions on the reference element, and let

$$\{\varphi_j^{\hat{K}_\ell}\}, \qquad j = 0, \ldots, n-1, \quad \ell = 0, 1, 2, 3$$

be the basis functions on the children of the reference element. Now, for $v \in \mathcal{P}_q(\hat{K})$, there exists the basis expansion

$$v(x) = \sum_{j=0}^{n-1} \alpha_j^{\hat{K}} \varphi_j^{\hat{K}}(x).$$

At the same time, v may be written as an expansion in terms of $\{\varphi_j^{\hat{K}_\ell}\}$:

$$v(x) \;=\; \sum_{j=0}^{n-1} \alpha_j^{\hat{K}} \varphi_j^{\hat{K}}(x) \;=\; \sum_{\ell=0}^{3} \sum_{j=0}^{n-1} \alpha_j^{\hat{K}_\ell} \varphi_j^{\hat{K}_\ell}(x). \tag{4.1}$$

Now, define

$$M_{i,j}^{\hat{K}_\ell} \;:=\; \varphi_j^{\hat{K}}(x_i^{\hat{K}_\ell}), \quad \text{for} \quad i,j = 0,\ldots,n-1 \quad \text{and} \quad \ell = 0,1,2,3. \tag{4.2}$$

Essentially, each $M_{i,j}^{\hat{K}_\ell}$ for $\ell = 0,1,2,3$ is a matrix corresponding to the basis functions $\varphi_j^{\hat{K}}$ evaluated at the degrees of freedom $x_i^{\hat{K}_\ell}$ of the children $\hat{K}_0, \hat{K}_1, \hat{K}_2$, and $\hat{K}_3$. Choosing $x = x_i^{\hat{K}_\ell}$ in (4.1) gives

$$\sum_{j=0}^{n-1} \alpha_j^{\hat{K}} \varphi_j^{\hat{K}}(x_i^{\hat{K}_\ell}) \;=\; \sum_{\ell=0}^{3} \sum_{j=0}^{n-1} \alpha_j^{\hat{K}_\ell} \varphi_j^{\hat{K}_\ell}(x_i^{\hat{K}_\ell}). \tag{4.3}$$

Since $\varphi_j^{\hat{K}_\ell}(x_i^{\hat{K}_\ell}) = \delta_{ji}$, using 4.1 gives

$$\sum_{j=0}^{n-1} M_{i,j}^{\hat{K}_\ell} \alpha_j^{\hat{K}} \;=\; \alpha_i^{\hat{K}_\ell}, \quad \text{for} \quad \ell = 0,1,2,3. \tag{4.4}$$

In block matrix form, this is

$$\begin{bmatrix} M^{\hat{K}_0} \\ \hline M^{\hat{K}_1} \\ \hline M^{\hat{K}_2} \\ \hline M^{\hat{K}_3} \end{bmatrix} \begin{bmatrix} \alpha_j^{\hat{K}} \end{bmatrix} \;=\; \begin{bmatrix} \alpha_i^{\hat{K}_0} \\ \hline \alpha_i^{\hat{K}_1} \\ \hline \alpha_i^{\hat{K}_2} \\ \hline \alpha_i^{\hat{K}_3} \end{bmatrix}. \tag{4.5}$$

The matrix $M = [M^{\hat{K}_0} M^{\hat{K}_1} M^{\hat{K}_2} M^{\hat{K}_3}]^T$ is precisely the matrix representation of the prolongation operator from the reference element to its children. As an example, consider the case $q = 1$, i.e., the approximation space consists of piecewise linear

polynomials. In this case, M has the form

$$\left[\begin{array}{ccc}
1 & 0 & 0 \\
\frac{1}{2} & \frac{1}{2} & 0 \\
\frac{1}{2} & 0 & \frac{1}{2} \\
\hline
\frac{1}{2} & \frac{1}{2} & 0 \\
0 & 1 & 0 \\
0 & \frac{1}{2} & \frac{1}{2} \\
\hline
\frac{1}{2} & 0 & \frac{1}{2} \\
0 & \frac{1}{2} & \frac{1}{2} \\
0 & 0 & 1 \\
\hline
0 & \frac{1}{2} & \frac{1}{2} \\
\frac{1}{2} & 0 & \frac{1}{2} \\
\frac{1}{2} & \frac{1}{2} & 0
\end{array}\right]$$

and maps the n-vector of coefficients on the parent cell to the $4n$-vector of coefficients on the children. The prolongation operator R_i^T is constructed by concatenating copies of this matrix M for each cell in the subdomain Ω_i.

4.7.4 Application of the Preconditioners

With the construction of the coarse grid components, subdomain components, and the restriction and prolongation operators complete, the application of the Schwarz preconditioner is relatively straightforward. Prior to beginning the PCG iterations, the stiffness matrices corresponding to the coarse subspace and the subdomains (each of which is sparse, symmetric, and positive definite) are factored using a Cholesky factorization $A_j = LL^T$, for $j = 0, 1, \ldots, p$. These factors are stored, so that during

each PCG iteration, the operation

$$R_j^T A_j^{-1} R_j x, \quad \text{for} \quad j = 0, 1, \ldots, p \tag{4.6}$$

consists of a restriction, an exact solve executed via a backward and a forward substitution, then a prolongation of the result. The Cholesky factorization is performed using the sparse matrix package `CSparse` created by Tim Davis [19], using the routine `cs_chol`. The corresponding backward and forward substitution solves are executed via the routines `cs_lsolve` and `cs_ltsolve`.

Remark 4.7.1. *It is important to note that choosing an exact solver for the coarse grid component and each of the subdomain solves is done purely for ensuring accurate estimates for the condition numbers $\kappa(BA)$. In order to develop a truly efficient implementation, the use of an exact solve in these situations would be abandoned and replaced with an iterative solver.*

Chapter 5

Future Directions

This chapter will contain a few of the plans for future research that will extend the ideas presented in this dissertation.

- *Investigate the creation of two-level Schwarz preconditioners applied to an adaptive DG scheme*

  The creation of *a posteriori* error estimates for the DG formulation of Poisson's equation was developed in [31]. The code developed for the experiments contained herein is already capable of implementing an adaptive process, making this a natural direction to investigate.

- *Analyze and construct Multilevel Schwarz preconditioners*

  The theoretical relationships between multilevel Schwarz methods and multigrid methods are explored in [45]. The comparison of the performance of the two methods applied to both second and fourth order elliptic problems is a potentially illuminating endeavor.

- *Construct and analyze other Schwarz preconditioners, including Multiplicative and Hybrid variants*

  A comparison of the performance of alternatives to the additive Schwarz methods constructed here could result in superior preconditioners. The questions to investigate are ones of design difficulty, computational efficiency, and performance.

- *Create a parallel implementation of the additive Schwarz preconditioners*

  The techniques of domain decomposition used to create the Schwarz preconditioners lend themselves readily to an implementation using a parallel architecture. In fact, the methods presented in this dissertation are designed specifically to reach peak performance in conjunction with a parallel implementation.

Bibliography

[1] R. A. Adams. *Sobolev Spaces*. Academic Press, New York, 1975.

[2] P. F. Antonietti and B. Ayuso. Schwarz domain decomposition preconditioners for discontinuous Galerkin approximations of elliptic problems: non-overlapping case. *ESAIM: Mathematical Modelling and Numerical Analysis*, 41(01):21–54, 2007. 3

[3] P. F. Antonietti, B. Ayuso de Dios, S. C. Brenner, and L. Y. Sung. Schwarz methods for a preconditioned wopsip method for elliptic problems. *Computational Methods in Applied Mathematics Comput. Methods Appl. Math.*, 12(3):241–272, 2012. 3

[4] P. F. Antonietti and P. Houston. A class of domain decomposition preconditioners for hp-discontinuous galerkin finite element methods. *Journal of Scientific Computing*, 46(1):124–149, 2011. 3

[5] D. Arnold. An interior penalty finite element method with discontinuous elements. *SIAM Journal on Numerical Analysis*, 19(4):742–760, 1982. 2, 9

[6] D. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini. Discontinuous galerkin methods for elliptic problems. In *Discontinuous Galerkin Methods*, pages 89–101. Springer, 2000. 2

[7] D. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini. Unified analysis of discontinuous galerkin methods for elliptic problems. *SIAM journal on numerical analysis*, 39(5):1749–1779, 2002. 2

[8] G. A. Baker. Finite element methods for elliptic equations using nonconforming elements. *Mathematics of Computation*, 31(137):45–59, 1977. 2

[9] G. A. Baker, W. N. Jureidini, and O. A. Karakashian. Piecewise solenoidal vector fields and the stokes problem. *SIAM journal on numerical analysis*, 27(6):1466–1485, 1990. 2

[10] A. T. Barker, S. C. Brenner, E. H. Park, and L. Y. Sung. Two-level additive schwarz preconditioners for a weakly over-penalized symmetric interior penalty method. *Journal of Scientific Computing*, 47(1):27–49, 2011. 3

[11] F. Bassi and S. Rebay. A high-order accurate discontinuous finite element method for the numerical solution of the compressible navier–stokes equations. *Journal of computational physics*, 131(2):267–279, 1997. 2

[12] F. Bassi and S. Rebay. Gmres discontinuous galerkin solution of the compressible navier-stokes equations. In *Discontinuous Galerkin Methods*, pages 197–208. Springer, 2000. 3

[13] C. E. Baumann and J. T. Oden. A discontinuous hp finite element method for convectiondiffusion problems. *Computer Methods in Applied Mechanics and Engineering*, 175(3):311–341, 1999. 2

[14] S. C. Brenner and R. Scott. *The Mathematical Theory of Finite Element Methods*. Springer-Verlag, 3rd edition, 2008. 11, 35, 89

[15] S. C. Brenner and K. Wang. Two-level additive schwarz preconditioners for c 0 interior penalty methods. *Numerische Mathematik*, 102(2):231–255, 2005.

[16] P. G. Ciarlet. *The Finite Element Method for Elliptic Problems*. North-Holland, Amsterdam, 1978.

[17] B. Cockburn. Discontinuous galerkin methods for convection-dominated problems. In *High-order methods for computational physics*, pages 69–224. Springer, 1999. 2

[18] B. Cockburn and C. W. Shu. The local discontinuous galerkin method for time-dependent convection-diffusion systems. *SIAM Journal on Numerical Analysis*, 35(6):2440–2463, 1998. 2

[19] T. A. Davis. *Direct Methods for Sparse Linear Systems (Fundamentals of Algorithms 2)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2006. 142

[20] J. Douglas and T. Dupont. Interior penalty procedures for elliptic and parabolic galerkin methods. In *Computing Methods in Applied Sciences*, pages 207–216. Springer, 1976. 2

[21] J. Douglas Jr., T. Dupont, P. Percell, and R. Scott. A family of C^1 finite elements with optimal approximation properties for various Galerkin methods for 2nd and 4th order problems. *RAIRO Analyse Numérique*, 13(3):227–255, 1979. 67

[22] M. Dryja and O. B. Widlund. Towards a unified theory of domain decomposition algorithms for elliptic problems. In T. Chan, editor, *Proceedings of Third International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 3–21. SIAM, 1990. 22

[23] X. Feng and O. A. Karakashian. Two-level additive Schwarz methods for a discontinuous Galerkin approximation of second order elliptic problems. *SIAM Journal on Numerical Analysis*, 39(4):1343–1365, 2001. 3, 13, 14, 16, 19, 40

[24] X. Feng and O. A. Karakashian. Two-level non-overlapping Schwarz preconditioners for a discontinuous Galerkin approximation of the biharmonic equation. *Journal of Scientific Computing*, 22(1-3):289–314, 2005. 3, 4, 71

[25] X. Feng and O. A. Karakashian. Fully discrete dynamic mesh discontinuous Galerkin methods for the Cahn-Hilliard equation of phase transition. *Mathematics of computation*, 76(259):1093–1117, 2007.

[26] E. H. Georgoulis, P. Houston, and J. Virtanen. An a posteriori error indicator for discontinuous Galerkin approximations of fourth-order elliptic problems. *IMA Journal of Numerical Analysis*, page drp023, 2009. 67

[27] P. Grisvard. *Singularities in Boundary Value Problems*, volume 22 of *Recherches en Mathématiques Appliquées [Research in Applied Mathematics]*. MASSON, Paris, 1992. 68

[28] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436, 1952. 132

[29] C. Johnson. *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Cambridge University Press, Cambridge, 1992.

[30] O. A. Karakashian and W. N. Jureidini. A nonconforming finite element method for the stationary Navier–Stokes equations. *SIAM Journal on Numerical Analysis*, 35(1):93–120, 1998. 2, 11

[31] O. A. Karakashian and F. Pascal. A posteriori error estimates for a discontinuous galerkin approximation of second-order elliptic problems. *SIAM Journal on Numerical Analysis*, 41(6):2374–2399, 2003. 12, 143

[32] O. A. Karakashian and F. Pascal. Convergence of adaptive discontinuous galerkin approximations of second-order elliptic problems. *SIAM Journal on Numerical Analysis*, 45(2):641–665, 2007. 12

[33] C. Lasser and A. Toselli. An overlapping domain decomposition preconditioner for a class of discontinuous Galerkin approximations of advection-diffusion problems. *Mathematics of Computation*, 72(243):1215–1238, 2003. 3

[34] G. Leoni. *A First Course in Sobolev Spaces*, volume 105. American Mathematical Society, 2009. 15

[35] A. Quarteroni and A. Valli. *Domain Decomposition Methods for Partial Differential Equations*. Oxford University Press, 1999. 3

[36] W. H. Reed and T. R. Hill. Triangular mesh methods for the neutron transport equation. Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory, 1973. 2

[37] B. Rivière. *Discontinuous Galerkin Methods for Solving Elliptic and Parabolic Equations: Theory and Implementation*. SIAM, 2008.

[38] M. A. Saum. *Adaptive Discontinuous Galerkin Finite Element Methods for Second and Fourth Order Elliptic Partial Differential Equations*. PhD thesis, University of Tennessee, August 2006.

[39] H. A. Schwarz. *Gesammelte Mathematische Abhandlungen*, volume 2. Springer Verlag, Berlin, 1890. First published in *Vierteljahrsschrift der Naturforschenden Gesellschaft* in Zürich, volume 15, 1870, pp.272-286. 2

[40] J. R. Shewchuk. Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. In Ming C. Lin and Dinesh Manocha, editors, *Applied Computational Geometry: Towards Geometric Engineering*, volume 1148 of *Lecture Notes in Computer Science*, pages 203–222. Springer-Verlag, May 1996. From the First ACM Workshop on Applied Computational Geometry. 123

[41] I. Smears. Nonoverlapping domain decomposition preconditioners for discontinuous galerkin finite element methods in H^2-type norms. *arXiv preprint arXiv:1409.4202*, 2014. 3

[42] B. E. Smith, P. Bjorstad, and W. Gropp. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press, New York, 1996. 3, 22, 23, 24

[43] A. Toselli and O. Widlund. *Domain Decomposition Methods – Algorithms and Theory*. Springer Series in Computational Mathematics, 2005. 3, 16, 24, 34

[44] M. F. Wheeler. An elliptic collocation-finite element method with interior penalties. *SIAM Journal on Numerical Analysis*, 15(1):152–161, 1978. 2

[45] J. Xu. Iterative methods by space decomposition and subspace correction. *SIAM review*, 34(4):581–613, 1992. 3, 22, 143

[46] H. You. *Adaptive Discontinuous Galerkin Finite Element Methods*. PhD thesis, University of Tennessee, August 2009.

Appendix

Appendix A

Affine Transformations and the Reference Element

The construction of the system corresponding to the DG formulation involves the computation of integrals over each element K and over edges e_h. Rather than compute these integrals directly, they are approximated using Gaussian quadrature. The computation of each local stiffness matrix requires the computation of several such approximations, which could prove costly if performed for each and every cell in a fine partition.

The common technique to avoid such inefficiency is to compute these approximations instead on a *reference element* $\hat{K}$, then use an affine transformation to map them back to the *physical element* K itself. The idea is that the necessary quadrature information to compute the integral approximations on the reference element $\hat{K}$ is computed and stored for easy access when computing local stiffness matrices and load vectors. Accessing template files containing this previously stored information is clearly much more efficient than performing a redundant computation over and over for each cell.

The implementation constructed for this thesis uses triangles for every cell K in a partition. The reference element $\hat{K}$ used is the triangle with vertices at $(1, 0)$, $(0, 1)$,

and $(0,0)$, enumerated as in Figure **??**. Given a triangle with vertices (x_0, y_0), (x_1, y_1), and (x_2, y_2), the affine transformation $\mathbf{F} : \hat{K} \to K$ is given (in matrix form) by

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x_0 - x_2 & x_1 - x_2 \\ y_0 - y_2 & y_1 - y_2 \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} + \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} \tag{A.1}$$

The inverse $\mathbf{F}^{-1} : K \to \hat{K}$ is given by

$$\begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} = \frac{1}{2|K|} \begin{bmatrix} y_1 - y_2 & x_2 - x_1 \\ y_2 - y_0 & x_0 - x_2 \end{bmatrix} \begin{bmatrix} x - x_2 \\ y - y_2 \end{bmatrix} \tag{A.2}$$

where $|K|$ is the area of the triangle K. Note that, componentwise, this is the system

$$\begin{aligned} \hat{x} &= \frac{1}{2|K|}\Big((y_1 - y_2)(x - x_2) + (x_2 - x_1)(y - y_2)\Big) \\ \hat{y} &= \frac{1}{2|K|}\Big((y_2 - y_0)(x - x_2) + (x_0 - x_2)(y - y_2)\Big) \end{aligned} \tag{A.3}$$

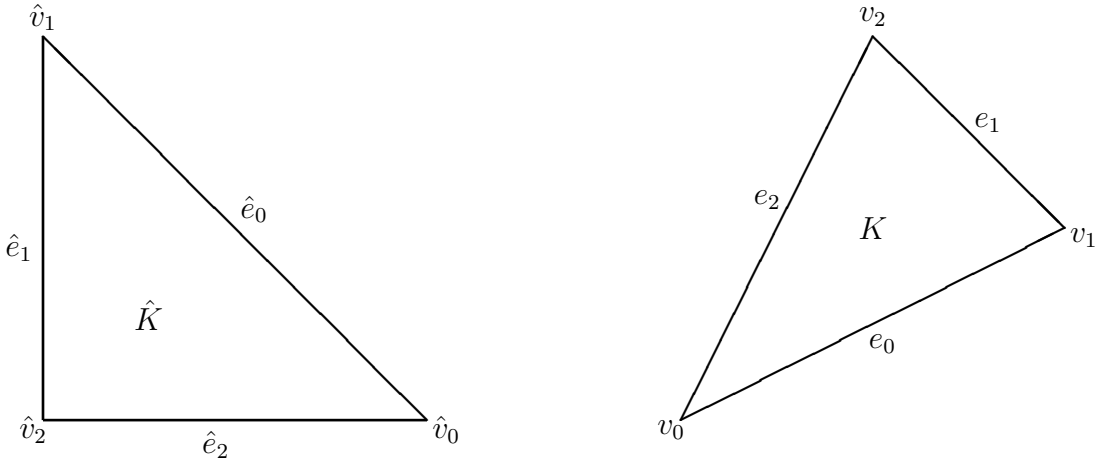


Figure A.1: Affine transformation between $\hat{K}$ and K.

It will be necessary to compute derivatives of basis functions ψ_j on each triangle K when computing local information. To avoid computing these directly, the derivatives of corresponding basis functions $\hat{\psi}_j$ on the reference triangle $\hat{K}$ will be calculated. The chain rule will be used to relate the derivatives on K to those on $\hat{K}$. In the following sections, the derivatives that appear in both the second order and the fourth order problems will be computed.

First Order Partial Derivatives

Suppose that $\psi(x, y)$ and $\hat{\psi}(\hat{x}, \hat{y})$ are corresponding basis function on K and $\hat{K}$, respectively. Using the chain rule,

$$\frac{\partial \psi}{\partial x} = \frac{\partial \hat{\psi}}{\partial \hat{x}} \frac{\partial \hat{x}}{\partial x} + \frac{\partial \hat{\psi}}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial x} \qquad \text{and} \qquad \frac{\partial \psi}{\partial y} = \frac{\partial \hat{\psi}}{\partial \hat{x}} \frac{\partial \hat{x}}{\partial y} + \frac{\partial \hat{\psi}}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial y}.$$

The derivatives

$$\frac{\partial \hat{x}}{\partial x}, \quad \frac{\partial \hat{x}}{\partial y}, \quad \frac{\partial \hat{y}}{\partial x}, \quad \text{and} \quad \frac{\partial \hat{y}}{\partial y}$$

can be computed using (A.3):

$$\begin{aligned} \frac{\partial \hat{x}}{\partial x} &= \frac{1}{2|K|}(y_1 - y_2) & \frac{\partial \hat{x}}{\partial y} &= \frac{1}{2|K|}(x_2 - x_1) \\ \frac{\partial \hat{y}}{\partial x} &= \frac{1}{2|K|}(y_2 - y_0) & \frac{\partial \hat{y}}{\partial y} &= \frac{1}{2|K|}(x_0 - x_2). \end{aligned} \tag{A.4}$$

For ease of notation, denote

$$\begin{aligned} a_{00} &:= y_1 - y_2 & a_{10} &:= x_2 - x_1 \\ a_{01} &:= y_2 - y_0 & a_{11} &:= x_0 - x_2. \end{aligned} \tag{A.5}$$

It follows that

$$\frac{\partial \psi}{\partial x} = \frac{1}{2|K|} \left(a_{00} \frac{\partial \hat{\psi}}{\partial \hat{x}} + a_{01} \frac{\partial \hat{\psi}}{\partial \hat{y}} \right) \tag{A.6}$$

and

$$\frac{\partial \psi}{\partial y} = \frac{1}{2|K|}\left(a_{10}\,\frac{\partial \hat{\psi}}{\partial \hat{x}} + a_{11}\,\frac{\partial \hat{\psi}}{\partial \hat{y}}\right) \tag{A.7}$$

Normal derivatives will be required as well. If $\mathbf{n} = [\mathbf{n}_x\ \mathbf{n}_y]^T$ is the outer unit normal for edge e_h, using the above gives

$$\begin{aligned}
\partial_n \psi &= \nabla \psi \cdot \mathbf{n} \\
&= \frac{\partial \psi}{\partial x}\,\mathbf{n}_x + \frac{\partial \psi}{\partial y}\,\mathbf{n}_y \\
&= \frac{1}{2|K|}\left[\left(a_{00}\,\frac{\partial \hat{\psi}}{\partial \hat{x}} + a_{01}\,\frac{\partial \hat{\psi}}{\partial \hat{y}}\right)\mathbf{n}_x + \left(a_{10}\,\frac{\partial \hat{\psi}}{\partial \hat{x}} + a_{11}\,\frac{\partial \hat{\psi}}{\partial \hat{y}}\right)\mathbf{n}_y\right].
\end{aligned} \tag{A.8}$$

Second Order Partial Derivatives

The second order derivatives needed occur in the fourth order problem, specifically in the Laplacian terms. Note that the first order derivatives (A.6) and (A.7) computed previously will be used here.

$$
\begin{aligned}
\frac{\partial^2\psi}{\partial x^2} &= \frac{1}{2|K|}\left[\, a_{00}\,\frac{\partial}{\partial x}\left(\frac{\partial\hat{\psi}}{\partial\hat{x}}\right) + \; a_{01}\,\frac{\partial}{\partial x}\left(\frac{\partial\hat{\psi}}{\partial\hat{y}}\right)\right] \\
&= \frac{1}{2|K|}\left[\, a_{00}\,\left(\frac{\partial^2\hat{\psi}}{\partial\hat{x}^2}\cdot\frac{\partial\hat{x}}{\partial x} + \frac{\partial^2\hat{\psi}}{\partial\hat{x}\partial\hat{y}}\cdot\frac{\partial\hat{y}}{\partial x}\right) + \; a_{01}\,\left(\frac{\partial^2\hat{\psi}}{\partial\hat{y}\partial\hat{x}}\cdot\frac{\partial\hat{x}}{\partial x} + \frac{\partial^2\hat{\psi}}{\partial\hat{y}^2}\cdot\frac{\partial\hat{y}}{\partial x}\right)\right] \\
&= \frac{1}{2|K|}\Bigg[\, a_{00}\,\left(\frac{1}{2|K|}\,a_{00}\,\frac{\partial^2\hat{\psi}}{\partial\hat{x}^2} + \frac{1}{2|K|}\,a_{01}\,\frac{\partial^2\hat{\psi}}{\partial\hat{x}\partial\hat{y}}\right) \\
&\qquad + \; a_{01}\,\left(\frac{1}{2|K|}\,a_{00}\,\frac{\partial^2\hat{\psi}}{\partial\hat{y}\partial\hat{x}} + \frac{1}{2|K|}\,a_{01}\,\frac{\partial^2\hat{\psi}}{\partial\hat{y}^2}\right)\Bigg] \\
&= \left(\frac{1}{2|K|}\right)^2\left[\, a_{00}^2\,\frac{\partial^2\hat{\psi}}{\partial\hat{x}^2} + 2a_{00}\,a_{01}\;\frac{\partial^2\hat{\psi}}{\partial\hat{x}\partial\hat{y}} + \; a_{01}^2\,\frac{\partial^2\hat{\psi}}{\partial\hat{y}^2}\right].
\end{aligned}
\tag{A.9}
$$

Similarly,

$$
\frac{\partial^2\psi}{\partial y^2} = \left(\frac{1}{2|K|}\right)^2\left[\, a_{10}^2\,\frac{\partial^2\hat{\psi}}{\partial\hat{x}^2} + 2a_{10}\,a_{11}\;\frac{\partial^2\hat{\psi}}{\partial\hat{x}\partial\hat{y}} + \; a_{11}^2\,\frac{\partial^2\hat{\psi}}{\partial\hat{y}^2}\right]. \tag{A.10}
$$

The Laplacian is then given by

$$
\Delta\psi = \left(\frac{1}{2|K|}\right)^2\left[\alpha_1\,\frac{\partial^2\hat{\psi}}{\partial\hat{x}^2} + \alpha_2\,\frac{\partial^2\hat{\psi}}{\partial\hat{x}\partial\hat{y}} + \alpha_3\,\frac{\partial^2\hat{\psi}}{\partial\hat{y}^2}\right], \tag{A.11}
$$

where

$$
\begin{aligned}
\alpha_1 &= \; a_{00}^2 + \; a_{10}^2 \\
\alpha_2 &= 2\; a_{00}\; a_{01} \; + 2\; a_{10}\; a_{11} \\
\alpha_3 &= \; a_{01}^2 + \; a_{11}^2.
\end{aligned}
\tag{A.12}
$$

In addition, the mixed second partial derivative is useful in the next section, so it is given here.

$$\frac{\partial^2\psi}{\partial x\partial y} = \left(\frac{1}{2|K|}\right)^2 \left[a_{00}a_{10}\ \frac{\partial^2\hat{\psi}}{\partial\hat{x}^2} + (a_{00}a_{11}+a_{01}a_{10})\ \frac{\partial^2\hat{\psi}}{\partial\hat{x}\partial\hat{y}} + a_{01}a_{11}\ \frac{\partial^2\hat{\psi}}{\partial\hat{y}^2}\right]. \tag{A.13}$$

Third Order Partial Derivatives

The third order derivatives needed also occur in the fourth order problem, specifically in the terms involving the normal derivative of the Laplacian.

$$\begin{aligned}
\frac{\partial^3\psi}{\partial x^3} &= \frac{\partial}{\partial x}\left(\frac{\partial^2\psi}{\partial x^2}\right) \\
&= \left(\frac{1}{2|K|}\right)^2 \left(a_{00}^2\ \frac{\partial}{\partial x}\left[\frac{\partial^2\hat{\psi}}{\partial\hat{x}^2}\right] + 2a_{00}a_{01}\ \frac{\partial}{\partial x}\left[\frac{\partial^2\hat{\psi}}{\partial\hat{x}\partial\hat{y}}\right] + a_{01}^2\ \frac{\partial}{\partial x}\left[\frac{\partial^2\hat{\psi}}{\partial\hat{y}^2}\right]\right) \\
&= \left(\frac{1}{2|K|}\right)^3 \left(a_{00}^3\ \frac{\partial^3\hat{\psi}}{\partial\hat{x}^3} + a_{00}^2a_{01}\ \frac{\partial^3\hat{\psi}}{\partial\hat{x}^2\partial\hat{y}} + 2a_{00}^2\ a_{01}\frac{\partial^3\hat{\psi}}{\partial\hat{x}^2\partial\hat{y}}\right. \\
&\qquad \left. + 2a_{00}\ a_{01}^2\ \frac{\partial^3\hat{\psi}}{\partial\hat{x}\partial\hat{y}^2} + a_{00}\ a_{01}^2\ \frac{\partial^3\hat{\psi}}{\partial\hat{x}\partial\hat{y}^2} + a_{01}^3\ \frac{\partial^3\hat{\psi}}{\partial\hat{y}^3}\right) \\
&= \left(\frac{1}{2|K|}\right)^3 \left(a_{00}^3\ \frac{\partial^3\hat{\psi}}{\partial\hat{x}^3} + 3a_{00}^2a_{01}\ \frac{\partial^3\hat{\psi}}{\partial\hat{x}^2\partial\hat{y}} + 3a_{00}\ a_{01}^2\ \frac{\partial^3\hat{\psi}}{\partial\hat{x}\partial\hat{y}^2} + a_{01}^3\ \frac{\partial^3\hat{\psi}}{\partial\hat{y}^3}\right).
\end{aligned} \tag{A.14}$$

Similarly,

$$\frac{\partial^3\psi}{\partial x^2\partial y} = \left(\frac{1}{2|K|}\right)^3 \left(a_{00}^2 a_{10}\ \frac{\partial^3\hat{\psi}}{\partial\hat{x}^3} + (2a_{00}\ a_{10}\ a_{01} + a_{00}^2 a_{11})\ \frac{\partial^3\hat{\psi}}{\partial\hat{x}^2\partial\hat{y}} + (2a_{00}\ a_{01}\ a_{11} + a_{01}^2 a_{10})\ \frac{\partial^3\hat{\psi}}{\partial\hat{x}\partial\hat{y}^2} + a_{01}^2 a_{11}\ \frac{\partial^3\hat{\psi}}{\partial\hat{y}^3} \right), \tag{A.15}$$

$$\frac{\partial^3\psi}{\partial x\partial y^2} = \left(\frac{1}{2|K|}\right)^3 \left(a_{00}\ a_{10}^2\ \frac{\partial^3\hat{\psi}}{\partial\hat{x}^3} + (2a_{00}\ a_{10}\ a_{11} + a_{01}\ a_{10}^2)\ \frac{\partial^3\hat{\psi}}{\partial\hat{x}^2\partial\hat{y}} + (2a_{01}\ a_{11}\ a_{10} + a_{00}\ a_{11}^2)\ \frac{\partial^3\hat{\psi}}{\partial\hat{x}\partial\hat{y}^2} + a_{01}\ a_{11}^2\ \frac{\partial^3\hat{\psi}}{\partial\hat{y}^3} \right), \tag{A.16}$$

and

$$\frac{\partial^3\psi}{\partial y^3} = \left(\frac{1}{2|K|}\right)^3 \left(a_{10}^3\ \frac{\partial^3\hat{\psi}}{\partial\hat{x}^3} + 3a_{10}^2 a_{11}\ \frac{\partial^3\hat{\psi}}{\partial\hat{x}^2\partial\hat{y}} + 3a_{10}\ a_{11}^2\ \frac{\partial^3\hat{\psi}}{\partial\hat{x}\partial\hat{y}^2} + a_{11}^3\ \frac{\partial^3\hat{\psi}}{\partial\hat{y}^3} \right). \tag{A.17}$$

Then, again with $\mathbf{n} = [\mathbf{n}_x\ \mathbf{n}_y]^T$,

$$\begin{aligned} \partial_n\Delta\psi &= \nabla(\Delta\psi)\cdot\mathbf{n} \\ &= \left(\frac{\partial^3\psi}{\partial x^3} + \frac{\partial^3\psi}{\partial x\partial y^2}\right)\mathbf{n}_x + \left(\frac{\partial^3\psi}{\partial x^2\partial y} + \frac{\partial^3\psi}{\partial y^3}\right)\mathbf{n}_y \\ &= \left(\frac{1}{2|K|}\right)^3 \left[\left(\beta_1\ \frac{\partial^3\hat{\psi}}{\partial\hat{x}^3} + \beta_2\ \frac{\partial^3\hat{\psi}}{\partial\hat{x}^2\partial\hat{y}} + \beta_3\ \frac{\partial^3\hat{\psi}}{\partial\hat{x}\partial\hat{y}^2} + \beta_4\ \frac{\partial^3\hat{\psi}}{\partial\hat{y}^3}\right)\mathbf{n}_x \right. \\ &\qquad \left. + \left(\eta_1\ \frac{\partial^3\hat{\psi}}{\partial\hat{x}^3} + \eta_2\ \frac{\partial^3\hat{\psi}}{\partial\hat{x}^2\partial\hat{y}} + \eta_3\ \frac{\partial^3\hat{\psi}}{\partial\hat{x}\partial\hat{y}^2} + \eta_4\ \frac{\partial^3\hat{\psi}}{\partial\hat{y}^3}\right)\mathbf{n}_y \right], \end{aligned} \tag{A.18}$$

where

$$\begin{aligned}
\beta_1 &= a_{00}^3 + a_{00}a_{10}^2 \\
\beta_2 &= 3a_{00}^2a_{01} + 2a_{00}a_{10}a_{11} + a_{01}a_{10}^2 \\
\beta_3 &= 3a_{00}a_{01}^2 + 2a_{01}a_{11}a_{10} + a_{00}a_{11}^2 \\
\beta_4 &= a_{01}^3 + a_{01}a_{11}^2
\end{aligned} \tag{A.19}$$

and

$$\begin{aligned}
\eta_1 &= a_{00}^2a_{10} + a_{10}^3 \\
\eta_2 &= 2a_{00}a_{10}a_{01} + a_{00}^2a_{11} + 3a_{10}^2a_{11} \\
\eta_3 &= 2a_{00}a_{01}a_{11} + a_{01}^2a_{10} + 3a_{10}a_{11}^2 \\
\eta_4 &= a_{01}^2a_{11} + a_{11}^3
\end{aligned} \tag{A.20}$$

Appendix B

Computation of PDE Data

B.1 Second Order Elliptic Problem

Derivation of the Bilinear Form

For each $K \in \mathcal{T}_h$, using integration by parts gives

$$\int_K -\Delta u v \, dx \;=\; \int_K \nabla u \cdot \nabla v \, dx \;-\; \int_{\partial K} (\nabla u \cdot \mathbf{n}) v \, ds \tag{B.1}$$

or, more compactly,

$$(-\Delta u, v)_K \;=\; (\nabla u, \nabla v)_K - \langle \nabla u \cdot \mathbf{n}, v \rangle_{\partial K} \,. \tag{B.2}$$

Summing over all $K \in \mathcal{T}_h$ gives

$$\begin{aligned}
(-\Delta u, v)_\Omega \;&=\; \sum_{K \in \mathcal{T}_h} \Big((\Delta u, \Delta v)_K - \langle \nabla u \cdot \mathbf{n}, v \rangle_{\partial K} \Big) \\
&=\; \sum_{K \in \mathcal{T}_h} (\nabla u, \nabla v)_K \;-\; \sum_{e_h \in \mathcal{E}_h^N} \langle \nabla u \cdot \mathbf{n}, v \rangle_{e_h} \;-\; \sum_{e_h \in \mathcal{E}_h^D} \langle \nabla u \cdot \mathbf{n}, v \rangle_{e_h} \\
&\quad - \sum_{e_h \in \mathcal{E}_h^I} \left\langle \nabla u^+ \cdot \mathbf{n}_{K^+}, v^+ \right\rangle_{e_h} \;-\; \sum_{e_h \in \mathcal{E}_h^I} \left\langle \nabla u^- \cdot \mathbf{n}_{K^-}, v^- \right\rangle_{e_h} .
\end{aligned} \tag{B.3}$$

The term involving Neumann edges is known data; therefore it is moved to the right side of the equation $(-\Delta u, v) = (f, v)$. Writing $n = n_{e_h} = \mathbf{n}_{K^+} = -\mathbf{n}_{K^-}$, and using the identity $a^+b^+ - a^-b^- = \{a\}[b] + [a]\{b\}$,

$$\begin{aligned}(-\Delta u, v)_\Omega &= \sum_{K\in\mathcal{T}_h} (\nabla u, \nabla v)_K - \sum_{e_h\in\mathcal{E}_h^D} \langle \partial_n u, v\rangle_{e_h} \\ &\quad - \sum_{e_h\in\mathcal{E}_h^I} \Big(\langle \partial_n u^+, v^+\rangle_{e_h} - \langle \partial_n u^-, v^-\rangle_{e_h} \Big) \\ &= \sum_{K\in\mathcal{T}_h} (\nabla u, \nabla v)_K - \sum_{e_h\in\mathcal{E}_h^D} \langle \partial_n u, v\rangle_{e_h} \\ &\quad - \sum_{e_h\in\mathcal{E}_h^I} \Big(\langle \{\partial_n u\}, [v]\rangle_{e_h} - \langle [\partial_n u], \{v\}\rangle_{e_h} \Big)\end{aligned} \tag{B.4}$$

Assuming u is sufficiently smooth, the term $\langle [\partial_n u], \{v\}\rangle_{e_h}$ is zero. For the SIP-DG method, it is replaced by the symmetrizing term $-\langle \{\partial_n v\}, [u]\rangle_{e_h}$. Combining this with the addition of the penalty terms and the symmetrizing term for the Dirichlet edges gives

$$\begin{aligned}(-\Delta u, v)_\Omega &= \sum_{K\in\mathcal{T}_h} (\nabla u, \nabla v)_K - \sum_{e_h\in\mathcal{E}_h^D} \Big(\langle \partial_n u, v\rangle_{e_h} + \langle \partial_n v, u\rangle_{e_h} \Big) \\ &\quad - \sum_{e_h\in\mathcal{E}_h^I} \Big(\langle \{\partial_n u\}, [v]\rangle_{e_h} + \langle \{\partial_n v\}, [u]\rangle_{e_h} \Big) \\ &\quad + \sum_{e_h\in\mathcal{E}_h^I\cup\mathcal{E}_h^D} \gamma_h |e_h|^{-1} \langle [u], [v]\rangle_{e_h} .\end{aligned} \tag{B.5}$$

Note that all nonzero terms added above are also added to the right hand side of $(-\Delta u, v) = (f, v)$. Putting this together, the bilinear form $a_h^{\gamma_h}(\cdot,\cdot)$ is given by

$$\begin{aligned} a_h^{\gamma_h}(u,v) &= \sum_{K\in\mathcal{T}_h} (\nabla u, \nabla v)_K - \sum_{e_h\in\mathcal{E}_h^D} \Big(\langle \partial_n u, v\rangle_{e_h} + \langle \partial_n v, u\rangle_{e_h} \Big) \\ &\quad - \sum_{e_h\in\mathcal{E}_h^I} \Big(\langle \{\partial_n u\}, [v]\rangle_{e_h} + \langle \{\partial_n v\}, [u]\rangle_{e_h} \Big) \\ &\quad + \sum_{e_h\in\mathcal{E}_h^I} \gamma_h |e_h|^{-1} \langle [u], [v]\rangle_{e_h} + \sum_{e_h\in\mathcal{E}_h^D} \gamma_h |e_h|^{-1} \langle u, v\rangle_{e_h} \end{aligned} \tag{B.6}$$

or, more concisely,

$$\begin{aligned} a_h^{\gamma_h}(u,v) &= \sum_{K\in\mathcal{T}_h} (\nabla u, \nabla v)_K - \sum_{e_h\in\mathcal{E}_h^I\cup\mathcal{E}_h^D} \Big(\langle \{\partial_n u\}, [v]\rangle_{e_h} + \langle \{\partial_n v\}, [u]\rangle_{e_h} \Big) \\ &\quad + \sum_{e_h\in\mathcal{E}_h^I\cup\mathcal{E}_h^D} \gamma_h |e_h|^{-1} \langle [u], [v]\rangle_{e_h} . \end{aligned} \tag{B.7}$$

The corresponding RHS linear functional is

$$\begin{aligned} \mathcal{L}(v) &:= \sum_{K\in\mathcal{T}_h} (f, v)_K + \sum_{e_h\in\mathcal{E}_h^N} \langle g_N, v\rangle_{e_h} - \sum_{e_h\in\mathcal{E}_h^D} \langle g_D, \partial_n v\rangle_{e_h} \\ &\quad + \sum_{e_h\in\mathcal{E}_h^D} \gamma_h |e_h|^{-1} \langle g_D, v\rangle_{e_h} . \end{aligned} \tag{B.8}$$

Formulation of Stiffness Matrix Terms

First, choose a basis for the DG space V^h. Let $\{\psi_j^K\}_{j=1}^p$ be a set of basis functions for each cell K and suppose that

$$u(\mathbf{x}) = \sum_{j=1}^n \xi_j^K \psi_j^K(\mathbf{x}) \quad \text{and} \quad v(\mathbf{x}) = \sum_{i=1}^p \psi_i^K(\mathbf{x}). \tag{B.9}$$

are the basis expansions of u and v, respectively. Here, n is the dimension of the basis; i.e., the number of degrees of freedom on K. Consider the terms from the bilinear

form one by one. Note that the superscript K will be suppressed except when needed in order to provide a less cumbersome notation.

Main Diagonal Block: Volume contributions from cell K

These terms represent the interaction of the basis functions of a cell K with each other and will appear in the main diagonal blocks. Using (B.9),

$$(\nabla u, \nabla v)_K \;=\; \sum_{i,j=1}^{n} \xi_j^K (\nabla \psi_j, \nabla \psi_i)_K.$$

All integrals will be computed on the reference element, so using the derivatives shown in Appendix 1,

$$\begin{aligned}(\nabla \psi_j, \nabla \psi_i)_K &= \int_K \nabla \psi_j \cdot \nabla \psi_i \; d\mathbf{x} \\ &= \left(\frac{1}{2|K|}\right) \int_{\hat{K}} \left[\left(a_{00} \frac{\partial \hat{\psi}_j}{\partial \hat{x}} + a_{01} \frac{\partial \hat{\psi}_j}{\partial \hat{y}} \right) \left(a_{00} \frac{\partial \hat{\psi}_i}{\partial \hat{x}} + a_{01} \frac{\partial \hat{\psi}_i}{\partial \hat{y}} \right) \right. \\ &\qquad \left. + \left(a_{10} \frac{\partial \hat{\psi}_j}{\partial \hat{x}} + a_{11} \frac{\partial \hat{\psi}_j}{\partial \hat{y}} \right) \left(a_{10} \frac{\partial \hat{\psi}_i}{\partial \hat{x}} + a_{11} \frac{\partial \hat{\psi}_i}{\partial \hat{y}} \right) \right] d\hat{\mathbf{x}}.\end{aligned} \tag{B.10}$$

Denoting for brevity

$$\begin{aligned}\hat{\varphi} \;:=\; &\left(a_{00} \frac{\partial \hat{\psi}_j}{\partial \hat{x}} + a_{01} \frac{\partial \hat{\psi}_j}{\partial \hat{y}} \right) \left(a_{00} \frac{\partial \hat{\psi}_i}{\partial \hat{x}} + a_{01} \frac{\partial \hat{\psi}_i}{\partial \hat{y}} \right) \\ &+ \left(a_{10} \frac{\partial \hat{\psi}_j}{\partial \hat{x}} + a_{11} \frac{\partial \hat{\psi}_j}{\partial \hat{y}} \right) \left(a_{10} \frac{\partial \hat{\psi}_i}{\partial \hat{x}} + a_{11} \frac{\partial \hat{\psi}_i}{\partial \hat{y}} \right),\end{aligned}$$

and approximating via Gaussian quadrature, we have

$$(\nabla \psi_j, \nabla \psi_i)_K \;\approx\; \frac{1}{2|K|} \sum_{q=1}^{n_q} \hat{w}_q \; \hat{\varphi}_j(\hat{x}_q), \tag{B.11}$$

where n_q is the number of Gaussian quadrature points used and $\hat{x}_q$ are the Gaussian quadrature points on the reference element with corresponding weights $\hat{w}_q$.

Contributions from edge e_h

Recall that each interior edge e_h is shared by two triangles, K^+ and K^-. In the derivations that follow, the definitions of jump and average will be used extensively. In addition, only the Arnold formulation will be shown, as Baker's follows similarly.

For each interior edge e_h, the edge terms from the bilinear form (B.7) may be expanded using the definition of jump and average. First, the flux terms:

$$\begin{aligned} \text{(i)} \quad -\langle\{\partial_n u\},[v]\rangle_{e_h} &= -\frac{1}{2}\left\langle \partial_n u^+, v^+\right\rangle_{e_h} + \frac{1}{2}\left\langle \partial_n u^+, v^-\right\rangle_{e_h} \\ &\quad -\frac{1}{2}\left\langle \partial_n u^-, v^+\right\rangle_{e_h} + \frac{1}{2}\left\langle \partial_n u^-, v^-\right\rangle_{e_h}, \end{aligned} \tag{B.12}$$

$$\begin{aligned} \text{(ii)} \quad -\langle\{\partial_n v\},[u]\rangle_{e_h} &= -\frac{1}{2}\left\langle \partial_n v^+, u^+\right\rangle_{e_h} + \frac{1}{2}\left\langle \partial_n v^+, u^-\right\rangle_{e_h} \\ &\quad -\frac{1}{2}\left\langle \partial_n v^-, u^+\right\rangle_{e_h} + \frac{1}{2}\left\langle \partial_n v^-, u^-\right\rangle_{e_h}. \end{aligned} \tag{B.13}$$

The penalty terms may be expanded the same way.

$$\begin{aligned} \text{(iii)} \quad \frac{\gamma_h}{|e_h|}\langle[u],[v]\rangle_{e_h} &= \frac{\gamma_h}{|e_h|}\left\langle u^+, v^+\right\rangle_{e_h} - \frac{\gamma_h}{|e_h|}\left\langle u^+, v^-\right\rangle_{e_h} \\ &\quad -\frac{\gamma_h}{|e_h|}\left\langle u^-, v^+\right\rangle_{e_h} + \frac{\gamma_h}{|e_h|}\left\langle u^-, v^-\right\rangle_{e_h}. \end{aligned} \tag{B.14}$$

Main Diagonal Block: Flux Terms

The details of these contributions will only be shown for the $\langle +,+\rangle_{e_h}$ terms as the $\langle -,-\rangle_{e_h}$ are similar. Using (B.12) – (B.14), these terms will be considered one at a time.

From (B.9),

$$\left\langle \partial_n u^+, v^+\right\rangle_{e_h} = \sum_{i,j=1}^{n} \xi_j^+ \left\langle \partial_n \psi_j^+, \psi_i^+\right\rangle_{e_h}. \tag{B.15}$$

Using the affine transformation (A.8), for brevity denote

$$\hat{\varphi}_j^+ \;=\; \left(a_{00}\,\frac{\partial\hat{\psi}}{\partial\hat{x}} \,+\, a_{01}\,\frac{\partial\hat{\psi}}{\partial\hat{y}}\right) n_x \,+\, \left(a_{10}\,\frac{\partial\hat{\psi}}{\partial\hat{x}} \,+\, a_{11}\,\frac{\partial\hat{\psi}}{\partial\hat{y}}\right) n_y \tag{B.16}$$

where $n = [n_x \;\; n_y]^T$. Then this main block diagonal term may be approximated via Gaussian quadrature by

$$\left\langle \partial_n \psi_j^+, \psi_i^+ \right\rangle_{e_h} \;\approx\; \frac{|e_h|}{2|K|} \sum_{q=1}^{n_q} \hat{w}_q \;\hat{\varphi}_j^+(\hat{x}_q)\;\psi_i^+(\hat{x}_q). \tag{B.17}$$

The main diagonal term $\langle \partial_n u^-, v^- \rangle_{e_h}$ is approximated similarly:

$$\left\langle \partial_n u^-, v^- \right\rangle_{e_h} \;=\; \sum_{i,j=1}^{n} \xi_j^- \left\langle \partial_n \psi_j^-, \psi_i^- \right\rangle_{e_h}, \tag{B.18}$$

where

$$\left\langle \partial_n \psi_j^-, \psi_i^- \right\rangle_{e_h} \;\approx\; \frac{|e_h|}{2|K|} \sum_{q=1}^{n_q} \hat{w}_q \;\hat{\varphi}_j^-(\hat{x}_q)\;\psi_i^-(\hat{x}_q). \tag{B.19}$$

where $\hat{\varphi}_j^-$ has an analagous definition to that of $\hat{\varphi}_j^+$.

The quadrature approximations of the corresponding symmetrizing terms that contribute to the main diagonal block, $\langle \partial_n v^+, u^+ \rangle_{e_h}$ and $\langle \partial_n v^-, u^- \rangle_{e_h}$ that appear in (B.13), are found exactly the same way as the previous two terms – simply with the i and j switched.

$$\left\langle \partial_n v^+, u^+ \right\rangle_{e_h} \;=\; \sum_{i,j=1}^{n} \xi_j^+ \left\langle \partial_n \psi_i^+, \psi_j^+ \right\rangle_{e_h} \tag{B.20}$$

where

$$\left\langle \partial_n \psi_i^+, \psi_j^+ \right\rangle_{e_h} \;\approx\; \frac{|e_h|}{2|K|} \sum_{q=1}^{n_q} \hat{w}_q \;\hat{\varphi}_i^+(\hat{x}_q)\;\psi_j^+(\hat{x}_q), \tag{B.21}$$

and

$$\left\langle \partial_n v^-, u^- \right\rangle_{e_h} \;=\; \sum_{i,j=1}^{n} \xi_j^- \left\langle \partial_n \psi_i^-, \psi_j^- \right\rangle_{e_h} \tag{B.22}$$

where

$$\left\langle \partial_n \psi_i^-, \psi_j^- \right\rangle_{e_h} \approx \frac{|e_h|}{2|K|} \sum_{q=1}^{n_q} \hat{w}_q \ \hat{\varphi}_i^-(\hat{x}_q) \ \psi_j^-(\hat{x}_q), \tag{B.23}$$

where the definitions of $\hat{\varphi}_i^{\pm}$ are analogous to the previous definition of $\hat{\varphi}_j^+$.

Main Diagonal Block: Penalty Terms

The contribution to the main diagonal blocks from the penalty terms are approximated as follows. The terms from (B.14) are easily dealt with. Again using the basis expansion (B.9),

$$\left\langle u^+, v^+ \right\rangle_{e_h} = \sum_{i,j=1}^{n} \xi_j^+ \left\langle \psi_j^+, \psi_i^+ \right\rangle_{e_h}, \tag{B.24}$$

where

$$\left\langle \psi_j^+, \psi_i^+ \right\rangle_{e_h} \approx |e_h| \sum_{q=1}^{n_q} \hat{w}_q \ \hat{\psi}_j^+(\hat{x}_q) \hat{\psi}_i^+(\hat{x}_q). \tag{B.25}$$

Also,

$$\left\langle u^-, v^- \right\rangle_{e_h} = \sum_{i,j=1}^{n} \xi_j^- \left\langle \psi_j^-, \psi_i^- \right\rangle_{e_h}, \tag{B.26}$$

where

$$\left\langle \psi_j^-, \psi_i^- \right\rangle_{e_h} \approx |e_h| \sum_{q=1}^{n_q} \hat{w}_q \ \hat{\psi}_j^-(\hat{x}_q) \hat{\psi}_i^-(\hat{x}_q). \tag{B.27}$$

Off-Diagonal Block: Flux Terms

The construction of the off-diagonal blocks follows very similarly to the derivations shown previously. Recall that these blocks represent the interaction of degrees of freedom from K^+ and K^- across an interior edge e_h. Since local enumeration follows a counterclockwise orientation, this means the quadrature points corresponding to the K^- cell must be traversed in reverse. The rest of the derivations follow exactly as before.

For brevity, again make use of the shorthand $\hat{\varphi}_j^+$ (B.16). The approximations from the off-diagonal flux terms are as follows.

From (B.12),

$$\left\langle \partial_n u^+, v^- \right\rangle_{e_h} = \sum_{i,j=1}^{n} \xi_j^+ \left\langle \partial_n \psi_j^+, \psi_i^- \right\rangle_{e_h} \tag{B.28}$$

where

$$\left\langle \partial_n \psi_j^+, \psi_i^- \right\rangle_{e_h} \approx \frac{|e_h|}{2|K^+|} \sum_{q=1}^{n_q} \hat{w}_q \; \hat{\varphi}_j^+(\hat{x}_q) \; \psi_i^-(\hat{x}_{\tilde{q}}), \tag{B.29}$$

where $\tilde{q} := n_q + 1 - q$ denotes the reverse traversal previously mentioned. Also,

$$\left\langle \partial_n u^-, v^+ \right\rangle_{e_h} = \sum_{i,j=1}^{n} \xi_j^- \left\langle \partial_n \psi_j^-, \psi_i^+ \right\rangle_{e_h} \tag{B.30}$$

where

$$\left\langle \partial_n \psi_j^-, \psi_i^+ \right\rangle_{e_h} \approx \frac{|e_h|}{2|K^-|} \sum_{q=1}^{n_q} \hat{w}_q \; \hat{\varphi}_j^-(\hat{x}_{\tilde{q}}) \; \psi_i^+(\hat{x}_q). \tag{B.31}$$

The approximations from (B.13) are

$$\left\langle \partial_n v^+, u^- \right\rangle_{e_h} = \sum_{i,j=1}^{n} \xi_j^- \left\langle \partial_n \psi_i^+, \psi_j^- \right\rangle_{e_h} \tag{B.32}$$

where

$$\left\langle \partial_n \psi_i^+, \psi_j^- \right\rangle_{e_h} \approx \frac{|e_h|}{2|K^+|} \sum_{q=1}^{n_q} \hat{w}_q \; \hat{\varphi}_i^+(\hat{x}_q) \; \psi_j^-(\hat{x}_{\tilde{q}}), \tag{B.33}$$

and

$$\left\langle \partial_n v^-, u^+ \right\rangle_{e_h} = \sum_{i,j=1}^{n} \xi_j^+ \left\langle \partial_n \psi_i^-, \psi_j^+ \right\rangle_{e_h} \tag{B.34}$$

where

$$\left\langle \partial_n \psi_i^-, \psi_j^+ \right\rangle_{e_h} \approx \frac{|e_h|}{2|K^-|} \sum_{q=1}^{n_q} \hat{w}_q \; \hat{\varphi}_i^-(\hat{x}_{\tilde{q}}) \; \psi_j^+(\hat{x}_q). \tag{B.35}$$

Off-Diagonal Block: Penalty Terms

The approximations from the off-diagonal penalty terms from (B.14) are as follows.

$$\left\langle u^+, v^- \right\rangle_{e_h} \;=\; \sum_{i,j=1}^{n} \xi_j^+ \left\langle \psi_j^+, \psi_i^- \right\rangle_{e_h}, \tag{B.36}$$

where

$$\left\langle \psi_j^+, \psi_i^- \right\rangle_{e_h} \;\approx\; |e_h| \sum_{q=1}^{n_q} \hat{w}_q \; \hat{\psi}_j^+(\hat{x}_q)\hat{\psi}_i^-(\hat{x}_{\tilde{q}}). \tag{B.37}$$

Also,

$$\left\langle u^-, v^+ \right\rangle_{e_h} \;=\; \sum_{i,j=1}^{n} \xi_j^- \left\langle \psi_j^-, \psi_i^+ \right\rangle_{e_h}, \tag{B.38}$$

where

$$\left\langle \psi_j^-, \psi_i^+ \right\rangle_{e_h} \;\approx\; |e_h| \sum_{q=1}^{n_q} \hat{w}_q \; \hat{\psi}_j^-(\hat{x}_{\tilde{q}})\hat{\psi}_i^+(\hat{x}_q). \tag{B.39}$$

B.2 Fourth Order Elliptic Problem

Derivation of the Bilinear Form

For each $K \in \mathcal{T}_h$, using integration by parts gives

$$\begin{aligned} \int_K \Delta^2 u v \; dx \;&=\; -\int_K \nabla(\Delta u) \cdot \nabla v \; dx \;+\; \int_{\partial K} v \; \nabla(\Delta u) \cdot \mathbf{n} \; ds \\ &=\; \int_K \Delta u \Delta v \; dx \;-\; \int_{\partial K} \Delta u \; \nabla v \cdot \mathbf{n} \; ds \\ &\quad +\; \int_{\partial K} v \; \nabla(\Delta u) \cdot \mathbf{n} \; ds \end{aligned} \tag{B.40}$$

or, more compactly,

$$(\Delta^2 u, v)_K \;=\; (\Delta u, \Delta v)_K \;-\; \langle \Delta u, \partial_n v \rangle_{\partial K} \;+\; \langle \partial_n \Delta u, v \rangle_{\partial K}. \tag{B.41}$$

Summing over all $K \in \mathcal{T}_h$, writing $n = n_{e_h} = \mathbf{n}_{K^+} = -\mathbf{n}_{K^-}$, and using the identity $a^+b^+ - a^-b^- = \{a\}[b] + [a]\{b\}$ gives

$$
\begin{aligned}
(\Delta^2 u, v)_\Omega &= \sum_{K\in\mathcal{T}_h} \Big[(\Delta u, \Delta v)_K \,-\, \langle \Delta u, \partial_n v\rangle_{\partial K} \,+\, \langle \partial_n \Delta u, v\rangle_{\partial K} \Big] \\
&= \sum_{K\in\mathcal{T}_h} (\Delta u, \Delta v)_K \,+\, \sum_{e_h\in\mathcal{E}_h^B} \Big[\langle \partial_n \Delta u, v\rangle_{e_h} \,-\, \langle \Delta u, \partial_n v\rangle_{e_h} \Big] \\
&\quad + \sum_{e_h\in\mathcal{E}_h^I} \Big[\langle \partial_n \Delta u^+, v^+\rangle_{e_h} \,-\, \langle \partial_n \Delta u^-, v^-\rangle_{e_h} \\
&\qquad\qquad - \langle \Delta u^+, \partial_n v^+\rangle_{e_h} \,+\, \langle \Delta u^-, \partial_n v^-\rangle_{e_h} \Big] \\
&= \sum_{K\in\mathcal{T}_h} (\Delta u, \Delta v)_K \,+\, \sum_{e_h\in\mathcal{E}_h^B} \Big[\langle \partial_n \Delta u, v\rangle_{e_h} \,-\, \langle \Delta u, \partial_n v\rangle_{e_h} \Big] \\
&\quad + \sum_{e_h\in\mathcal{E}_h^I} \Big[\langle \{\partial_n \Delta u\}, [v]\rangle_{e_h} \,+\, \langle [\partial_n \Delta u], \{v\}\rangle_{e_h} \\
&\qquad\qquad - \langle \{\Delta u\}, [\partial_n v]\rangle_{e_h} \,-\, \langle [\Delta u], \{\partial_n v\}\rangle_{e_h} \Big] \\
&= \sum_{K\in\mathcal{T}_h} (\Delta u, \Delta v)_K \,+\, \sum_{e_h\in\mathcal{E}_h^B} \Big[\langle \partial_n \Delta u, v\rangle_{e_h} \,-\, \langle \Delta u, \partial_n v\rangle_{e_h} \Big] \\
&\quad + \sum_{e_h\in\mathcal{E}_h^I} \Big[\langle \{\partial_n \Delta u\}, [v]\rangle_{e_h} \,-\, \langle \{\Delta u\}, [\partial_n v]\rangle_{e_h} \Big] \\
&= \sum_{K\in\mathcal{T}_h} (\Delta u, \Delta v)_K \,+\, \sum_{e_h\in\mathcal{E}_h^I\cup\mathcal{E}_h^B} \Big[\langle \{\partial_n \Delta u\}, [v]\rangle_{e_h} \,-\, \langle \{\Delta u\}, [\partial_n v]\rangle_{e_h} \Big] \\
&= \sum_{K\in\mathcal{T}_h} (\Delta u, \Delta v)_K \,+\, \sum_{e_h\in\mathcal{E}_h^I\cup\mathcal{E}_h^B} \Big[\langle \{\partial_n \Delta u\}, [v]\rangle_{e_h} \,-\, \langle \{\Delta u\}, [\partial_n v]\rangle_{e_h} \Big] \\
&\quad + \sum_{e_h\in\mathcal{E}_h^I\cup\mathcal{E}_h^B} \Big[\langle \{\partial_n \Delta v\}, [u]\rangle_{e_h} \,-\, \langle \{\Delta v\}, [\partial_n u]\rangle_{e_h} \Big]
\end{aligned}
\tag{B.42}
$$

Finally, adding the penalty terms

$$\sum_{e_h \in \mathcal{E}_h^I \cup \mathcal{E}_h^B} \left(\frac{\gamma_h}{|e_h|^3} \langle [u], [v] \rangle_{e_h} \; + \; \frac{\gamma_h}{|e_h|} \langle [\partial_n u], [\partial_n v] \rangle_{e_h} \right) \tag{B.43}$$

and rearranging, we have the bilinear form

$$\begin{aligned} a_h^{\gamma_h}(u,v) \;\; := \;\; & \sum_{K \in \mathcal{T}_h} (\Delta u, \Delta v)_K \; + \; \sum_{e_h \in \mathcal{E}_h^I \cup \mathcal{E}_h^B} \left(\frac{\gamma_h}{|e_h|^3} \langle [u], [v] \rangle_{e_h} \; + \; \frac{\gamma_h}{|e_h|} \langle [\partial_n u], [\partial_n v] \rangle_{e_h} \right) \\ & + \sum_{e_h \in \mathcal{E}_h^I \cup \mathcal{E}_h^B} \Big(\langle [u], \{\partial_n \Delta v\} \rangle_{e_h} \; + \; \langle [v], \{\partial_n \Delta u\} \rangle_{e_h} \\ & \qquad - \langle [\partial_n u], \{\Delta v\} \rangle_{e_h} \; - \; \langle [\partial_n v], \{\Delta u\} \rangle_{e_h} \Big) \end{aligned} \tag{B.44}$$

and the corresponding RHS linear functional

$$\begin{aligned} \mathcal{L}(v) \;\; := \;\; & \sum_{K \in \mathcal{T}_h} (f, v)_K \; + \; \sum_{e_h \in \mathcal{E}_h^B} \Big(\langle \partial_n \Delta v, g_D \rangle_{e_h} - \langle \Delta v, g_N \rangle_{e_h} \Big) \\ & + \sum_{e_h \in \mathcal{E}_h^B} \left(\frac{\sigma_h}{|e_h|^3} \langle g_D, v \rangle_{e_h} + \frac{\tau_h}{|e_h|} \langle g_N, \partial_n v \rangle_{e_h} \right). \end{aligned} \tag{B.45}$$

Formulation of Stiffness Matrix Terms

First, choose a basis for the DG space V^h. Let $\{\psi_j^K\}_{j=1}^p$ be a set of basis functions for each cell K and suppose that

$$u(\mathbf{x}) \; = \; \sum_{j=1}^{n} \xi_j^K \psi_j^K(\mathbf{x}) \quad \text{and} \quad v(\mathbf{x}) \; = \; \sum_{i=1}^{p} \psi_i^K(\mathbf{x}). \tag{B.46}$$

are the basis expansions of u and v, respectively. Here, n is the dimension of the basis; i.e., the number of degrees of freedom on K. Consider the terms from the bilinear form one by one. Note that the superscript K will be suppressed except when needed in order to provide a less cumbersome notation.

Main Diagonal Block: Volume contributions from cell K

These terms represent the interaction of the basis functions of a cell K with each other and will appear in the main diagonal blocks. Using (B.46),

$$(\Delta u, \Delta v)_K \;=\; \sum_{i,j=1}^{n} \xi_j^K (\Delta\psi_j, \Delta\psi_i)_K.$$

All integrals will be computed on the reference element, so using the derivatives shown in Appendix 1,

$$\begin{aligned}
(\Delta\psi_j.\Delta\psi_i)_K &= \int_K \Delta\psi_j \Delta\psi_i \; d\mathbf{x} \\
&= \left(\frac{1}{2|K|}\right)^4 \int_{\hat{K}} \left[\left(\alpha_1 \frac{\partial^2\hat{\psi}_j}{\partial\hat{x}^2} + \alpha_2 \frac{\partial^2\hat{\psi}_j}{\partial\hat{x}\partial\hat{y}} + \alpha_3 \frac{\partial^2\hat{\psi}_j}{\partial\hat{y}^2}\right)\right. \\
&\qquad \left. \cdot \left(\alpha_1 \frac{\partial^2\hat{\psi}_i}{\partial\hat{x}^2} + \alpha_2 \frac{\partial^2\hat{\psi}_i}{\partial\hat{x}\partial\hat{y}} + \alpha_3 \frac{\partial^2\hat{\psi}_i}{\partial\hat{y}^2}\right)\right] d\hat{\mathbf{x}}
\end{aligned} \tag{B.47}$$

Denoting

$$\begin{aligned}
\hat{\varphi}_j &:= \alpha_1 \frac{\partial^2\hat{\psi}_j}{\partial\hat{x}^2} + \alpha_2 \frac{\partial^2\hat{\psi}_j}{\partial\hat{x}\partial\hat{y}} + \alpha_3 \frac{\partial^2\hat{\psi}_j}{\partial\hat{y}^2} \\
\hat{\varphi}_i &:= \alpha_1 \frac{\partial^2\hat{\psi}_i}{\partial\hat{x}^2} + \alpha_2 \frac{\partial^2\hat{\psi}_i}{\partial\hat{x}\partial\hat{y}} + \alpha_3 \frac{\partial^2\hat{\psi}_i}{\partial\hat{y}^2},
\end{aligned}$$

and approximating via Gaussian quadrature, we have

$$(\Delta\psi_j, \Delta\psi_i)_K \;\approx\; \left(\frac{1}{2|K|}\right)^3 \sum_{q=1}^{n_q} \hat{w}_q \; \hat{\varphi}_j(\hat{x}_q) \; \hat{\varphi}_i(\hat{x}_q), \tag{B.48}$$

where n_q is the number of Gaussian quadrature points used and $\hat{x}_q$ are the Gaussian quadrature points on the reference element with corresponding weights $\hat{w}_q$.

Contributions from edge e_h

Recall that each interior edge e_h is shared by two triangles, K^+ and K^-. In the derivations that follow, the definitions of jump and average will be used extensively. In addition, only the Arnold formulation will be shown, as Baker's follows similarly.

For each interior edge e_h, the edge terms from the bilinear form (B.44) may be expanded using the definition of jump and average. First, the flux terms:

$$\begin{aligned}
\text{(i)}\quad \langle [u], \{\partial_n \Delta v\} \rangle_{e_h} &= \frac{1}{2}\langle u^+, \partial_n \Delta v^+ \rangle_{e_h} - \frac{1}{2}\langle u^-, \partial_n \Delta v^+ \rangle_{e_h} \\
&\quad + \frac{1}{2}\langle u^+, \partial_n \Delta v^- \rangle_{e_h} - \frac{1}{2}\langle u^-, \partial_n \Delta v^- \rangle_{e_h}.
\end{aligned} \tag{B.49}$$

$$\begin{aligned}
\text{(ii)}\quad \langle [v], \{\partial_n \Delta u\} \rangle_{e_h} &= \frac{1}{2}\langle v^+, \partial_n \Delta u^+ \rangle_{e_h} - \frac{1}{2}\langle v^-, \partial_n \Delta u^+ \rangle_{e_h} \\
&\quad + \frac{1}{2}\langle v^+, \partial_n \Delta u^- \rangle_{e_h} - \frac{1}{2}\langle v^-, \partial_n \Delta u^- \rangle_{e_h}.
\end{aligned} \tag{B.50}$$

$$\begin{aligned}
\text{(iii)}\quad -\langle [\partial_n u], \{\Delta v\} \rangle_{e_h} &= -\frac{1}{2}\langle \partial_n u^+, \Delta v^+ \rangle_{e_h} + \frac{1}{2}\langle \partial_n u^-, \Delta v^+ \rangle_{e_h} \\
&\quad - \frac{1}{2}\langle \partial_n u^+, \Delta v^- \rangle_{e_h} + \frac{1}{2}\langle \partial_n u^-, \Delta v^- \rangle_{e_h}
\end{aligned} \tag{B.51}$$

$$\begin{aligned}
\text{(iv)}\quad -\langle [\partial_n v], \{\Delta u\} \rangle_{e_h} &= -\frac{1}{2}\langle \partial_n v^+, \Delta u^+ \rangle_{e_h} + \frac{1}{2}\langle \partial_n v^-, \Delta u^+ \rangle_{e_h} \\
&\quad - \frac{1}{2}\langle \partial_n v^+, \Delta u^- \rangle_{e_h} + \frac{1}{2}\langle \partial_n v^-, \Delta u^- \rangle_{e_h}
\end{aligned} \tag{B.52}$$

The penalty terms may expanded in the same way.

$$\begin{aligned}(\text{v}) \quad \frac{\gamma_h}{|e_h|^3}\langle [u],[v]\rangle_{e_h} \;&=\; \frac{\gamma_h}{|e_h|^3}\langle u^+, v^+\rangle_{e_h} \;-\; \frac{\gamma_h}{|e_h|^3}\langle u^+, v^-\rangle_{e_h} \\ &\quad - \frac{\gamma_h}{|e_h|^3}\langle u^-, v^+\rangle_{e_h} \;+\; \frac{\gamma_h}{|e_h|^3}\langle u^-, v^-\rangle_{e_h}\end{aligned} \tag{B.53}$$

$$\begin{aligned}(\text{vi}) \quad \frac{\gamma_h}{|e_h|}\langle [\partial_n u],[\partial_n v]\rangle_{e_h} \;&=\; \frac{\gamma_h}{|e_h|}\langle \partial_n u^+, \partial_n v^+\rangle_{e_h} \;-\; \frac{\gamma_h}{|e_h|}\langle \partial_n u^+, \partial_n v^-\rangle_{e_h} \\ &\quad - \frac{\gamma_h}{|e_h|}\langle \partial_n u^-, \partial_n v^+\rangle_{e_h} \;+\; \frac{\gamma_h}{|e_h|}\langle \partial_n u^-, \partial_n v^-\rangle_{e_h}\end{aligned} \tag{B.54}$$

The terms above containing functions both on K^+ or both on K^- will contribute only to the main diagonal block (note that this includes terms corresponding to boundary edges). The mixed terms with one function from K^+ and the other from K^- will contribute to the off-diagonal block corresponding to the shared edge e_h.

Main Diagonal Block: Flux Terms

The details of these contributions will only be shown for the $\langle +,+\rangle_{e_h}$ terms as the $\langle -,-\rangle_{e_h}$ are similar. Using (B.49) – (B.54), these terms will be considered one at a time.

From (B.46),

$$\langle u^+, \partial_n \Delta v^+\rangle_{e_h} \;=\; \sum_{i,j=1}^{n} \xi_j^+ \left\langle \psi_j^+, \partial_n \Delta \psi_i^+ \right\rangle_{e_h} \tag{B.55}$$

Using the affine transformation (A.18), for brevity denote

$$\begin{aligned}\hat{\varphi}_i^+ \;=\; & \left(\beta_1^+ \frac{\partial^3\hat{\psi}_i^+}{\partial\hat{x}^3} + \beta_2^+ \frac{\partial^3\hat{\psi}_i^+}{\partial\hat{x}^2\partial\hat{y}} + \beta_3^+ \frac{\partial^3\hat{\psi}_i^+}{\partial\hat{x}\partial\hat{y}^2} + \beta_4^+ \frac{\partial^3\hat{\psi}_i^+}{\partial\hat{y}^3}\right)\mathbf{n}_x \\ & + \left(\eta_1^+ \frac{\partial^3\hat{\psi}_i^+}{\partial\hat{x}^3} + \eta_2^+ \frac{\partial^3\hat{\psi}_i^+}{\partial\hat{x}^2\partial\hat{y}} + \eta_3^+ \frac{\partial^3\hat{\psi}_i^+}{\partial\hat{x}\partial\hat{y}^2} + \eta_4^+ \frac{\partial^3\hat{\psi}_i^+}{\partial\hat{y}^3}\right)\mathbf{n}_y,\end{aligned} \tag{B.56}$$

where the coefficients are those given in (A.19) and (A.20). Putting this together, this main block diagonal term may be approximated via Gaussian quadrature by

$$\left\langle \psi_j^+, \partial_n\Delta\psi_i^+\right\rangle_{e_h} \;\approx\; |e_h|\left(\frac{1}{2|K|}\right)^3 \sum_{q=1}^{n_q} \hat{w}_q\; \hat{\psi}_j^+(\hat{x}_q)\; \hat{\varphi}_i^+(\hat{x}_q). \tag{B.57}$$

The main diagonal term $\langle u^-, \partial_n\Delta v^-\rangle_{e_h}$ is approximated similarly:

$$\langle u^-, \partial_n\Delta v^-\rangle_{e_h} \;=\; \sum_{i,j=1}^{n} \xi_j^- \left\langle \psi_j^-, \partial_n\Delta\psi_i^-\right\rangle_{e_h} \tag{B.58}$$

where

$$\left\langle \psi_j^-, \partial_n\Delta\psi_i^-\right\rangle_{e_h} \;\approx\; |e_h|\left(\frac{1}{2|K|}\right)^3 \sum_{q=1}^{n_q} \hat{w}_q\; \hat{\psi}_j^-(\hat{x}_q)\; \hat{\varphi}_i^-(\hat{x}_q)\,, \tag{B.59}$$

with $\hat{\varphi}_i^-$ defined analogously to the definition of $\hat{\varphi}_i^+$.

The quadrature approximations of the symmetrizing terms contributing to the main diagonal block, $\langle v^+, \partial_n\Delta u^+\rangle_{e_h}$ and $\langle v^-, \partial_n\Delta u^-\rangle_{e_h}$ found in (B.50), are found exactly the same way as the previous two terms – simply with the i and j switched.

$$\langle v^+, \partial_n\Delta u^+\rangle_{e_h} \;=\; \sum_{i,j=1}^{n} \xi_j^+ \left\langle \psi_i^+, \partial_n\Delta\psi_j^+\right\rangle_{e_h} \tag{B.60}$$

where

$$\left\langle \psi_i^+, \partial_n\Delta\psi_j^+\right\rangle_{e_h} \;\approx\; |e_h|\left(\frac{1}{2|K|}\right)^3 \sum_{q=1}^{n_q} \hat{w}_q\; \hat{\psi}_i^+(\hat{x}_q)\; \hat{\varphi}_j^+(\hat{x}_q)\,, \tag{B.61}$$

and

$$\langle v^-, \partial_n \Delta u^- \rangle_{e_h} \;=\; \sum_{i,j=1}^{n} \xi_j^- \left\langle \psi_i^-, \partial_n \Delta \psi_j^- \right\rangle_{e_h} \tag{B.62}$$

where

$$\langle \psi_i^-, \partial_n \Delta \psi_j^- \rangle_{e_h} \;\approx\; |e_h| \left(\frac{1}{2|K|} \right)^3 \sum_{q=1}^{n_q} \hat{w}_q \; \hat{\psi}_i^-(\hat{x}_q) \; \hat{\varphi}_j^-(\hat{x}_q)\,, \tag{B.63}$$

where the definitions of $\hat{\varphi}_j^{\pm}$ are analogous to the previous definition of $\hat{\varphi}_i^+$.

The next terms, found in (B.51), are $\langle \partial_n u^+, \Delta v^+ \rangle_{e_h}$ and $\langle \partial_n u^-, \Delta v^- \rangle_{e_h}$. As before,

$$\langle \partial_n u^+, \Delta v^+ \rangle_{e_h} \;=\; \sum_{i,j=1}^{n} \xi_j^+ \left\langle \partial_n \psi_j^+, \Delta \psi_i^+ \right\rangle_{e_h} \tag{B.64}$$

and

$$\langle \partial_n u^-, \Delta v^- \rangle_{e_h} \;=\; \sum_{i,j=1}^{n} \xi_j^- \left\langle \partial_n \psi_j^-, \Delta \psi_i^- \right\rangle_{e_h}. \tag{B.65}$$

Appealing to the results of the affine transformations (A.8) and (A.11), for brevity denote

$$\hat{\mu}_j^+ \;:=\; \left(a_{00} \, \frac{\partial \hat{\psi}_j^+}{\partial \hat{x}} \;+\; a_{01} \, \frac{\partial \hat{\psi}_j^+}{\partial \hat{y}} \right) \mathbf{n}_x \;+\; \left(a_{10} \, \frac{\partial \hat{\psi}_j^+}{\partial \hat{x}} \;+\; a_{11} \, \frac{\partial \hat{\psi}_j^+}{\partial \hat{y}} \right) \mathbf{n}_y \tag{B.66}$$

and

$$\hat{\nu}_i^+ \;:=\; \alpha_1 \, \frac{\partial^2 \hat{\psi}_i^+}{\partial \hat{x}^2} \;+\; \alpha_2 \, \frac{\partial^2 \hat{\psi}_i^+}{\partial \hat{x} \partial \hat{y}} \;+\; \alpha_3 \, \frac{\partial^2 \hat{\psi}_i^+}{\partial \hat{y}^2}, \tag{B.67}$$

with the coefficients defined as in (A.5) and (A.12). Then

$$\left\langle \partial_n \psi_j^+, \Delta \psi_i^+ \right\rangle_{e_h} \;\approx\; |e_h| \left(\frac{1}{2|K|} \right)^3 \sum_{q=1}^{n_q} \hat{w}_q \; \hat{\mu}_j^+(\hat{x}_q) \; \hat{\nu}_i^+(\hat{x}_q)\,, \tag{B.68}$$

and

$$\left\langle \partial_n \psi_j^-, \Delta \psi_i^- \right\rangle_{e_h} \;\approx\; |e_h| \left(\frac{1}{2|K|} \right)^3 \sum_{q=1}^{n_q} \hat{w}_q \; \hat{\mu}_j^-(\hat{x}_q) \; \hat{\nu}_i^-(\hat{x}_q)\,, \tag{B.69}$$

where the definitions of $\hat{\mu}_j^-$ and $\hat{\nu}_i^-$ are analogous to the definitions of $\hat{\mu}_j^+$ and $\hat{\nu}_i^+$, respectively.

The approximations of the next symmetrizing terms contributing to the main diagonal block, $\langle \partial_n v^+, \Delta u^+ \rangle_{e_h}$ and $\langle \partial_n v^-, \Delta u^- \rangle_{e_h}$ found in (B.52), are found exactly the same way as the previous two terms; again by symmetry, simply by switching i and j. Then

$$\langle \partial_n v^+, \Delta u^+ \rangle_{e_h} \;=\; \sum_{i,j=1}^{n} \xi_j^+ \left\langle \partial_n \psi_i^+, \Delta \psi_j^+ \right\rangle_{e_h}, \tag{B.70}$$

where

$$\left\langle \partial_n \psi_i^+, \Delta \psi_j^+ \right\rangle_{e_h} \;\approx\; |e_h| \left(\frac{1}{2|K|} \right)^3 \sum_{q=1}^{n_q} \hat{w}_q \; \hat{\mu}_i^+(\hat{x}_q) \; \hat{\nu}_j^+(\hat{x}_q), \tag{B.71}$$

and

$$\langle \partial_n v^-, \Delta u^- \rangle_{e_h} \;=\; \sum_{i,j=1}^{n} \xi_j^- \left\langle \partial_n \psi_i^-, \Delta \psi_j^- \right\rangle_{e_h}, \tag{B.72}$$

where

$$\left\langle \partial_n \psi_i^-, \Delta \psi_j^- \right\rangle_{e_h} \;\approx\; |e_h| \left(\frac{1}{2|K|} \right)^3 \sum_{q=1}^{n_q} \hat{w}_q \; \hat{\mu}_i^-(\hat{x}_q) \; \hat{\nu}_j^-(\hat{x}_q). \tag{B.73}$$

As usual, the definitions of $\hat{\mu}_i^\pm$ and $\hat{\nu}_j^\pm$ are analogous to the definitions of $\hat{\mu}_j^+$ and $\hat{\nu}_i^+$ given previously.

Main Diagonal Block: Penalty Terms

The contribution to the main diagonal blocks from the penalty terms are approximated as follows. The terms from (B.53) are easily dealt with. Again using the basis expansion (B.46),

$$\left\langle u^+, v^+ \right\rangle_{e_h} \;=\; \sum_{i,j=1}^{n} \xi_j^+ \left\langle \psi_j^+, \psi_i^+ \right\rangle_{e_h}, \tag{B.74}$$

where

$$\left\langle \psi_j^+, \psi_i^+ \right\rangle_{e_h} \;\approx\; |e_h| \sum_{q=1}^{n_q} \hat{w}_q \; \hat{\psi}_j^+(\hat{x}_q) \hat{\psi}_i^+(\hat{x}_q). \tag{B.75}$$

Also,

$$\left\langle u^-, v^- \right\rangle_{e_h} = \sum_{i,j=1}^{n} \xi_j^- \left\langle \psi_j^-, \psi_i^- \right\rangle_{e_h}, \tag{B.76}$$

where

$$\left\langle \psi_j^-, \psi_i^- \right\rangle_{e_h} \approx |e_h| \sum_{q=1}^{n_q} \hat{w}_q \, \hat{\psi}_j^-(\hat{x}_q)\hat{\psi}_i^-(\hat{x}_q). \tag{B.77}$$

The terms from (B.53) are also fairly simple. Again using the affine transformation (A.8) and the shorthand $\hat{\mu}_j^+$ (B.66),

$$\left\langle \partial_n u^+, \partial_n v^+ \right\rangle_{e_h} = \sum_{i,j=1}^{n} \xi_j^+ \left\langle \partial_n \psi_j^+, \partial_n \psi_i^+ \right\rangle_{e_h}, \tag{B.78}$$

where

$$\left\langle \partial_n \psi_j^+, \partial_n \psi_i^+ \right\rangle_{e_h} \approx |e_h| \left(\frac{1}{2|K^+|} \right)^2 \sum_{q=1}^{n_q} \hat{w}_q \, \hat{\mu}_j^+(\hat{x}_q)\hat{\mu}_i^+(\hat{x}_q). \tag{B.79}$$

The corresponding $\langle -, - \rangle_{e_h}$ term is

$$\left\langle \partial_n u^-, \partial_n v^- \right\rangle_{e_h} = \sum_{i,j=1}^{n} \xi_j^- \left\langle \partial_n \psi_j^-, \partial_n \psi_i^- \right\rangle_{e_h}, \tag{B.80}$$

where

$$\left\langle \partial_n \psi_j^-, \partial_n \psi_i^- \right\rangle_{e_h} \approx |e_h| \left(\frac{1}{2|K^-|} \right)^2 \sum_{q=1}^{n_q} \hat{w}_q \, \hat{\mu}_j^-(\hat{x}_q)\hat{\mu}_i^-(\hat{x}_q). \tag{B.81}$$

Off-Diagonal Block: Flux Terms

The construction of the off-diagonal blocks follows very similarly to the derivations shown previously. Recall that these blocks represent the interaction of degrees of freedom from K^+ and K^- across an interior edge e_h. Since local enumeration follows a counterclockwise orientation, this means the quadrature points corresponding to the K^- cell must be traversed in reverse. The rest of the derivations follow exactly as before. Again make use of the shorthands for $\hat{\varphi}_j^+$ (B.56), $\hat{\mu}_j^+$ (B.66), and $\hat{\nu}_j^+$ (B.67). With this in mind, the approximations from the off-diagonal flux terms are as follows.

From (B.49),

$$\langle u^+, \partial_n \Delta v^- \rangle_{e_h} \;=\; \sum_{i,j=1}^{n} \xi_j^+ \left\langle \psi_j^+, \partial_n \Delta \psi_i^- \right\rangle_{e_h} \tag{B.82}$$

where

$$\left\langle \psi_j^+, \partial_n \Delta \psi_i^- \right\rangle_{e_h} \;\approx\; |e_h| \left(\frac{1}{2|K^-|} \right)^3 \sum_{q=1}^{n_q} \hat{w}_q \; \hat{\psi}_j^+(\hat{x}_q) \; \hat{\varphi}_i^-(\hat{x}_{\tilde{q}}), \tag{B.83}$$

where $\tilde{q} := n_q + 1 - q$ denotes the reverse traversal previously mentioned. Also,

$$\langle u^-, \partial_n \Delta v^+ \rangle_{e_h} \;=\; \sum_{i,j=1}^{n} \xi_j^- \left\langle \psi_j^-, \partial_n \Delta \psi_i^+ \right\rangle_{e_h} \tag{B.84}$$

where

$$\left\langle \psi_j^-, \partial_n \Delta \psi_i^+ \right\rangle_{e_h} \;\approx\; |e_h| \left(\frac{1}{2|K^+|} \right)^3 \sum_{q=1}^{n_q} \hat{w}_q \; \hat{\psi}_j^-(\hat{x}_{\tilde{q}}) \; \hat{\varphi}_i^+(\hat{x}_q). \tag{B.85}$$

The approximations from (B.50) are

$$\langle v^+, \partial_n \Delta u^- \rangle_{e_h} \;=\; \sum_{i,j=1}^{n} \xi_j^- \left\langle \psi_i^+, \partial_n \Delta \psi_j^- \right\rangle_{e_h} \tag{B.86}$$

where

$$\langle \psi_i^+, \partial_n \Delta \psi_j^- \rangle_{e_h} \;\approx\; |e_h| \left(\frac{1}{2|K^-|} \right)^3 \sum_{q=1}^{n_q} \hat{w}_q \; \hat{\psi}_i^+(\hat{x}_q) \; \hat{\varphi}_j^-(\hat{x}_{\tilde{q}}) \,, \tag{B.87}$$

and

$$\langle v^-, \partial_n \Delta u^+ \rangle_{e_h} \;=\; \sum_{i,j=1}^{n} \xi_j^+ \left\langle \psi_i^-, \partial_n \Delta \psi_j^+ \right\rangle_{e_h} \tag{B.88}$$

where

$$\langle \psi_i^-, \partial_n \Delta \psi_j^+ \rangle_{e_h} \;\approx\; |e_h| \left(\frac{1}{2|K^+|} \right)^3 \sum_{q=1}^{n_q} \hat{w}_q \; \hat{\psi}_i^-(\hat{x}_{\tilde{q}}) \; \hat{\varphi}_j^+(\hat{x}_q). \tag{B.89}$$

The approximations from (B.51) are

$$\langle \partial_n u^+, \Delta v^- \rangle_{e_h} \;=\; \sum_{i,j=1}^{n} \xi_j^+ \left\langle \partial_n \psi_j^+, \Delta \psi_i^- \right\rangle_{e_h} \tag{B.90}$$

where

$$\left\langle \partial_n \psi_j^+, \Delta \psi_i^- \right\rangle_{e_h} \approx |e_h| \left(\frac{1}{2|K^+|} \right) \left(\frac{1}{2|K^-|} \right)^2 \sum_{q=1}^{n_q} \hat{w}_q \, \hat{\mu}_j^+(\hat{x}_q) \, \hat{\nu}_i^-(\hat{x}_{\tilde{q}}) \,, \tag{B.91}$$

and

$$\langle \partial_n u^-, \Delta v^+ \rangle_{e_h} = \sum_{i,j=1}^{n} \xi_j^- \left\langle \partial_n \psi_j^-, \Delta \psi_i^+ \right\rangle_{e_h} . \tag{B.92}$$

where

$$\left\langle \partial_n \psi_j^-, \Delta \psi_i^+ \right\rangle_{e_h} \approx |e_h| \left(\frac{1}{2|K^+|} \right)^2 \left(\frac{1}{2|K^-|} \right) \sum_{q=1}^{n_q} \hat{w}_q \, \hat{\mu}_j^-(\hat{x}_{\tilde{q}}) \, \hat{\nu}_i^+(\hat{x}_q) \,. \tag{B.93}$$

The approximations from (B.52) are

$$\langle \partial_n v^+, \Delta u^- \rangle_{e_h} = \sum_{i,j=1}^{n} \xi_j^- \left\langle \partial_n \psi_i^+, \Delta \psi_j^- \right\rangle_{e_h} , \tag{B.94}$$

where

$$\left\langle \partial_n \psi_i^+, \Delta \psi_j^- \right\rangle_{e_h} \approx |e_h| \left(\frac{1}{2|K^+|} \right) \left(\frac{1}{2|K^-|} \right)^2 \sum_{q=1}^{n_q} \hat{w}_q \, \hat{\mu}_i^+(\hat{x}_q) \, \hat{\nu}_j^-(\hat{x}_{\tilde{q}}) \,, \tag{B.95}$$

and

$$\langle \partial_n v^-, \Delta u^+ \rangle_{e_h} = \sum_{i,j=1}^{n} \xi_j^+ \left\langle \partial_n \psi_i^-, \Delta \psi_j^+ \right\rangle_{e_h} , \tag{B.96}$$

where

$$\left\langle \partial_n \psi_i^-, \Delta \psi_j^+ \right\rangle_{e_h} \approx |e_h| \left(\frac{1}{2|K^+|} \right)^2 \left(\frac{1}{2|K^-|} \right) \sum_{q=1}^{n_q} \hat{w}_q \, \hat{\mu}_i^-(\hat{x}_{\tilde{q}}) \, \hat{\nu}_j^+(\hat{x}_q) \,. \tag{B.97}$$

Off-Diagonal Block: Penalty Terms

Again using the shorthand $\hat{\mu}_j^+$ (B.66), the approximations from the off-diagonal penalty terms from (B.53) and (B.54) are as follows.

$$\left\langle u^+, v^- \right\rangle_{e_h} = \sum_{i,j=1}^{n} \xi_j^+ \left\langle \psi_j^+, \psi_i^- \right\rangle_{e_h}, \tag{B.98}$$

where

$$\left\langle \psi_j^+, \psi_i^- \right\rangle_{e_h} \approx |e_h| \sum_{q=1}^{n_q} \hat{w}_q \; \hat{\psi}_j^+(\hat{x}_q)\hat{\psi}_i^-(\hat{x}_{\tilde{q}}). \tag{B.99}$$

Also,

$$\left\langle u^-, v^+ \right\rangle_{e_h} = \sum_{i,j=1}^{n} \xi_j^- \left\langle \psi_j^-, \psi_i^+ \right\rangle_{e_h}, \tag{B.100}$$

where

$$\left\langle \psi_j^-, \psi_i^+ \right\rangle_{e_h} \approx |e_h| \sum_{q=1}^{n_q} \hat{w}_q \; \hat{\psi}_j^-(\hat{x}_{\tilde{q}})\hat{\psi}_i^+(\hat{x}_q). \tag{B.101}$$

Finally,

$$\left\langle \partial_n u^+, \partial_n v^- \right\rangle_{e_h} = \sum_{i,j=1}^{n} \xi_j^+ \left\langle \partial_n \psi_j^+, \partial_n \psi_i^- \right\rangle_{e_h}, \tag{B.102}$$

where

$$\left\langle \partial_n \psi_j^+, \partial_n \psi_i^- \right\rangle_{e_h} \approx |e_h| \left(\frac{1}{2|K^+|}\right)\left(\frac{1}{2|K^-|}\right) \sum_{q=1}^{n_q} \hat{w}_q \; \hat{\mu}_j^+(\hat{x}_q)\hat{\mu}_i^-(\hat{x}_{\tilde{q}}), \tag{B.103}$$

and

$$\left\langle \partial_n u^-, \partial_n v^+ \right\rangle_{e_h} = \sum_{i,j=1}^{n} \xi_j^- \left\langle \partial_n \psi_j^-, \partial_n \psi_i^+ \right\rangle_{e_h}, \tag{B.104}$$

where

$$\left\langle \partial_n \psi_j^-, \partial_n \psi_i^+ \right\rangle_{e_h} \approx |e_h| \left(\frac{1}{2|K^-|}\right)\left(\frac{1}{2|K^+|}\right) \sum_{q=1}^{n_q} \hat{w}_q \; \hat{\mu}_j^-(\hat{x}_{\tilde{q}})\hat{\mu}_i^+(\hat{x}_q). \tag{B.105}$$

Vita

Craig Dwain Collins was born in Murray, Kentucky in 1973. He attended Marshall County High School until his senior year, at which point he left to attend the Armand Hammer United World College of the American West in Montezuma, New Mexico. Leaving this international school in 1990, he returned to Kentucky and began taking classes at Murray State University. A scholastic career was not yet meant to be – he left school to pursue a career in music, successfully working as a professional guitarist for more than a decade.

In 2002 the siren song of academia became too strong to ignore and he returned to Murray State University to study mathematics. He earned a Bachelor of Science degree in Mathematics in the Spring of 2006. He stayed at Murray State as a graduate student, under the tutelage of Dr. Renee Fister. During the next two years he helped author several papers with Dr. Fister on the application of Optimal Control theory to tumor modeling. In the Spring of 2008, he earned a Master of Science degree in Mathematics. He accepted a position as a lecturer at Murray State for the 2008-09 school year.

In the fall of 2009, he was accepted into the Ph.D. program at the University of Tennessee, Knoxville. As a graduate student at UTK, he worked for Dr. Steven Wise as a research assistant on multigrid methods and unconditionally energy stable schemes for the Cahn-Hilliard-Brinkman system. This work led to a publication with Dr. Wise and Dr. Jie Shen of Purdue. He earned a Doctor of Philosophy degree in Mathematics in the Summer of 2015, working under Dr. Ohannes Karakashian.