

ADAPTIVE DISCONTINUOUS GALERKIN METHODS FOR NON-LINEAR
REACTIVE FLOWS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MURAT UZUNCA

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
MATHEMATICS

DECEMBER 2014

Approval of the thesis:

**ADAPTIVE DISCONTINUOUS GALERKIN METHODS FOR NON-LINEAR
REACTIVE FLOWS**

submitted by **MURAT UZUNCA** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Mathematics Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Mustafa Korkmaz
Head of Department, **Mathematics**

Prof. Dr. Bülent Karasözen
Supervisor, **Mathematics Department, METU**

Examining Committee Members:

Prof. Dr. Münevver Tezer
Mathematics Department, METU

Prof. Dr. Bülent Karasözen
Mathematics Department, METU

Assoc. Prof. Dr. Yusuf Uludağ
Chemical Engineering Department, METU

Assoc. Prof. Dr. Ömür Uğur
Institute of Applied Mathematics, METU

Assist. Prof. Dr. Fikriye Nuray Yılmaz
Mathematics Department, Gazi University

Date:

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: MURAT UZUNCA

Signature :

ABSTRACT

ADAPTIVE DISCONTINUOUS GALERKIN METHODS FOR NON-LINEAR REACTIVE FLOWS

Uzunca, Murat

Ph.D., Department of Mathematics

Supervisor : Prof. Dr. Bülent Karasözen

December 2014, 122 pages

The aim of this thesis is to solve the convection/reaction dominated non-stationary semi-linear diffusion-convection-reaction problems with internal/boundary layers in an accurate and efficient way using a time-space adaptive algorithm. We use for space discretization the symmetric interior penalty discontinuous Galerkin method, and backward Euler for time discretization. Our main interest is to derive robust residual-based a posteriori error estimators both in space and time. To derive the a posteriori bounds for the fully discrete system, we utilize the elliptic reconstruction technique. The use of elliptic reconstruction technique allows us to use the a posteriori error estimators derived for stationary models and to obtain optimal orders in $L^\infty(L^2)$ norms.

Keywords: Time-Space Adaptivity, Discontinuous Galerkin Methods, Non-linear Reaction, A Posteriori Error Estimation, Elliptic Reconstruction

ÖZ

DOĞRUSAL OLMAYAN REAKSİYON AKIŞLARI İÇİN UYARLAMALI SÜREKSİZ GALERKİN METODLARI

Uzunca, Murat

Doktora, Matematik Bölümü

Tez Yöneticisi : Prof. Dr. Bülent Karasözen

Aralık 2014 , 122 sayfa

Bu tezin amacı, konveksiyon veya reaksiyonun baskın olduğu zamana bağımlı yarı-doğrusal difüzyon-konveksiyon-reaksiyon denklemlerinin, uzayda simetrik süreksiz Galerkin ve zamanda geriye dönük Euler yöntemleriyle ayrıklaştırılarak zaman-uzay uyarlamalı ağlarla çözümüdür. Uzay ve zamanda oluşan tabakalar, çözüme bağlı hata kestiriciler aracılığıyla saptanarak yüksek kesinlikte sayısal çözümler elde edilmiştir. Söz konusu hata kestiricilerin ayrıklaştırılmış sisteme göre oluşturulması için "eliptik yeniden inşaa" adı verilen bir teknik kullanılmıştır. Bu teknik sayesinde, zamandan bağımsız modeller için geliştirilmiş hata kestiricilerinin kullanılabilmesi sağlanmış ve $L^\infty(L^2)$ -tipi normlarında optimal yakınsama katsayıları elde edilmiştir.

Anahtar Kelimeler: Zaman-Uzay Uyarlama, Süreksiz Galerkin Metodları, Doğrusal Olmayan Reaksiyon, Sonradan Hata Kestirici, Eliptik Yeniden İnşaa

To my family

ACKNOWLEDGMENTS

First of all, I would like to special thank to my supervisor Prof. Dr. Bülent Karasözen for valuable guidance, useful suggestions, penetrative advices, his kindness and patience. He have provided me the opportunity to get inside and understand the world of Numerical Analysis and Applied Mathematics. I would have not been able to succeed in completing my PhD studies without his golden support.

I acknowledge Dr. Hamdullah Yücel for discussions on discontinuous Galerkin methods.

I acknowledge Assoc. Prof. Dr. Murat Manguoğlu for the suggestions, help and useful discussions.

I acknowledge Dr. Martin Weiser and Lars Lubkoll for their kind hosting in Berlin when we collaborated to study on their own package program KASKADE.

I acknowledge Prof. Dr. Fredi Tröltzsch, Prof. Dr. Ronald W. Hoppe and Prof. Dr. Peter Deuffhard for their suggestions and valuable discussions.

I am thankful to my thesis defense committee members for their useful comments and discussions.

I would like to thank to Prof. Dr. Zafer Ercan and Assoc. Prof. Dr. S. Feza Arslan for their pre-guidance in the beginning of my PhD study at Middle East Technical University.

I special thank to Assis. Prof. Dr. İsmail Uğur Tiryaki and Assis. Prof. Dr. Tahire Özen Öztürk for their guidance and suggestions which make me decide to start PhD study at Middle East Technical University.

I would like to thank Ersin Kızıgut, Ayşegül Kıvılcım, Adalet Çengel, Sibel Doğru Akgöl and all other Friends, all Members of the Mathematics Department and all Members of the Institute of Applied Mathematics at Middle East Technical University for the pleasant and friendship atmosphere.

I acknowledge the financial support of Turkish Scientific and Technical Research Council (TÜBİTAK).

Finally, special thanks to Mukaddes Kaya and to my Family for their support and patience through the all my educational life.

TABLE OF CONTENTS

| | |
|---|------|
| ABSTRACT | v |
| ÖZ | vi |
| ACKNOWLEDGMENTS | viii |
| TABLE OF CONTENTS | ix |
| LIST OF TABLES | xiii |
| LIST OF FIGURES | xiv |
| CHAPTERS | |
| 1 INTRODUCTION | 1 |
| 2 DISCONTINUOUS GALERKIN METHODS | 7 |
| 2.1 Preliminaries | 9 |
| 2.1.1 Sobolev Spaces | 9 |
| 2.1.2 Trace Theorems | 11 |
| 2.1.3 Cauchy-Schwarz's and Young's Inequalities | 11 |
| 2.2 Construction of IPG Methods | 12 |
| 2.3 Computation Tools for Integral Terms | 15 |
| 2.3.1 Reference Element | 15 |

| | | |
|-------|---|----|
| 2.3.2 | Numerical Quadrature | 16 |
| 2.3.3 | Basis Functions | 17 |
| 2.4 | Effect of Penalty Parameter | 19 |
| 2.5 | Problems with Convection | 21 |
| 2.5.1 | Coercivity of Bilinear Form | 22 |
| 3 | ELLIPTIC PROBLEMS WITH ADAPTIVITY | 27 |
| 3.1 | Model Elliptic Problem | 28 |
| 3.1.1 | Discrete System in Matrix-Vector Form | 29 |
| 3.2 | Adaptivity | 32 |
| 3.2.1 | The Adaptive Procedure | 33 |
| 3.2.2 | A Posteriori Error Estimation | 36 |
| 3.3 | Solution of Linearized Systems | 42 |
| 3.3.1 | Matrix Reordering & Block LU Factorization | 43 |
| 3.4 | Comparison with Galerkin Least Squares FEM (GLSFEM) | 46 |
| 3.5 | Numerical Examples | 50 |
| 3.5.1 | Example with Polynomial Type Non-Linearity | 50 |
| 3.5.2 | Example with Monod Type Non-Linearity | 55 |
| 3.5.3 | Coupled Example with Arrhenius Type Non-Linearity | 58 |
| 3.5.4 | Coupled Example with Polynomial Type Non-Linearity | 59 |
| 4 | PARABOLIC PROBLEMS WITH TIME-SPACE ADAPTIVITY | 61 |
| 4.1 | Preliminaries and Model Equation | 62 |

| | | |
|-------|---|-----|
| 4.2 | Semi-Discrete and Fully Discrete Formulations | 64 |
| 4.2.1 | Semi-Discrete Formulation | 65 |
| 4.2.2 | Fully Discrete Formulation | 67 |
| 4.3 | Time-Space Adaptivity for Non-Stationary Problems | 67 |
| 4.3.1 | A Posteriori Error Bounds for Semi-Discrete System | 70 |
| 4.3.2 | A Posteriori Error Bounds for Fully Discrete System | 73 |
| 4.3.3 | Adaptive Algorithm | 75 |
| 4.4 | Solution of Fully Discrete System | 77 |
| 4.5 | Numerical Examples | 79 |
| 4.5.1 | Example with Polynomial Type Non-Linearity (Benchmark of the Algorithm) | 79 |
| 4.5.2 | Coupled Example with Polynomial Type Non-Linearity | 83 |
| 4.5.3 | A Non-linear ADR in Homogeneous Porous Media | 85 |
| 4.5.4 | A Non-linear ADR in Deterministic Heterogeneous Porous Media | 86 |
| 5 | CONCLUSIONS AND FUTURE WORK | 91 |
| | REFERENCES | 93 |
| | APPENDICES | |
| A | MATLAB TUTORIAL | 101 |
| A.1 | Linear Model Problem | 101 |
| A.2 | Description of MATLAB Codes | 102 |
| A.2.1 | Mesh Generation | 103 |

| | | |
|-------|--|-----|
| A.2.2 | User Defined Quantities | 105 |
| A.2.3 | Forming and Solving Linear Systems | 106 |
| A.2.4 | Plotting Solutions | 108 |
| A.3 | Models with Non-Linear Reaction Mechanisms | 108 |
| A.4 | Full Version of the Code <i>Main_Nonlinear</i> | 111 |
| B | A SIMPLE MATLAB CODE FOR MATRIX REORDERING | 117 |
| | CURRICULUM VITAE | 119 |

LIST OF TABLES

TABLES

| | |
|--|----|
| Table 3.1 Example 3.5.1: Average number of Newton iterations, average number of <i>BiCGStab</i> iterations, total computation time in seconds corresponding to the uniformly refined (adaptively refined) mesh | 52 |
|--|----|

LIST OF FIGURES

FIGURES

| | |
|---|----|
| Figure 2.1 Two adjacent elements sharing an edge (left); an element near to domain boundary (right) | 13 |
| Figure 2.2 Collapsed coordinate transform between reference triangle and reference square | 18 |
| Figure 2.3 Degree vs. condition number of the stiffness matrix: comparison for different type of basis functions for the Poisson problem (2.4) | 19 |
| Figure 2.4 Condition number of the stiffness matrix of the SIPG method as the functions of the penalty parameter σ (left) and the mesh-size h (right) with different polynomial degree k for the Poisson problem (2.4) | 20 |
| Figure 2.5 Maximum nodal errors of the SIPG approximation as a function of penalty parameter σ with different polynomial degree k for the Poisson problem (2.4) | 20 |
| Figure 3.1 Adaptive strategy | 33 |
| Figure 3.2 Bisection of a triangle | 35 |
| Figure 3.3 Solution profiles of linear convection dominated problem (3.38) | 49 |
| Figure 3.4 Comparison of the GLSFEM and DGFEM for linear convection dominated problem (3.38) | 50 |
| Figure 3.5 Example 3.5.1; Adaptive mesh | 51 |
| Figure 3.6 Example 3.5.1: Uniform (left) and adaptive (right) solutions, quadratic elements | 51 |
| Figure 3.7 Example 3.5.1: Global errors: comparison of the methods by quadratic elements (left), adaptive DG for polynomial degrees 1-4 (right) | 52 |
| Figure 3.8 Example 3.5.1: Condition number of the matrices J (unpermuted matrix), S (Schur complement matrix) and A (left top block of permuted matrix): Uniform refinement (left) and adaptive refinement (right) | 53 |

| | |
|--|----|
| Figure 3.9 Example 3.5.1: Sparsity patterns of the unpermuted (left), permuted (middle) and the Schur complement (right) matrices at the final refinement levels: Uniform refinement (top) with DoFs 196608 and adaptive refinement (bottom) with DoFs 70716 | 54 |
| Figure 3.10 Example 3.5.1: Computation time vs. DoFs: Uniform refinement (left) and adaptive refinement (right) | 55 |
| Figure 3.11 Example 3.5.1: # Average <i>BiCGStab</i> iterations vs. DoFs: Uniform refinement (left) and adaptive refinement (right) | 55 |
| Figure 3.12 Example 3.5.2: Adaptive meshes (left) and the cross-section plots (right) of the solutions at the left outflow boundary by quadratic elements (top) and quartic elements (bottom) | 56 |
| Figure 3.13 Example 3.5.3: Uniform(left) and adaptive(right) solutions to the temperature(top) and reactant(bottom), quadratic elements with DoFs 12288 for uniform refinement and with DoFs 6168 for adaptive refinement | 57 |
| Figure 3.14 Example 3.5.4: Adaptive mesh, quartic elements with DoFs 33690 | 58 |
| Figure 3.15 Example 3.5.4: Uniformly (left) and adaptively (right) obtained cross-section plots on the outflow boundary for the component u_2 , quartic elements with DoFs 61440 for uniform refinement and with DoFs 33690 for adaptive refinement | 60 |
| Figure 4.1 Adaptive algorithm chart on a single time step $(t_{k-1}, t_k]$ | 76 |
| Figure 4.2 Example 4.5.1: Decays of estimators and errors for uniform time-space | 80 |
| Figure 4.3 Example 4.5.1: Error vs. spatial (left) and temporal (right) estimators for $\epsilon = 10^{-6}$ | 80 |
| Figure 4.4 Example 4.5.1: Spatial effectivity indices (left) and estimators (right) | 81 |
| Figure 4.5 Example 4.5.1: Temporal effectivity indices (left) and estimators (right) | 81 |
| Figure 4.6 Example 4.5.1: Adaptive mesh | 82 |
| Figure 4.7 Example 4.5.1: Uniform (left) and adaptive (right) solutions at $T=0.5$ | 82 |
| Figure 4.8 Example 4.5.1: Evolution of DoFs (left) and time-steps Δt (right) . | 83 |

| | |
|---|-----|
| Figure 4.9 Example 4.5.2: Cross-section plots in the cross-wind direction at $t = 0.1$ (left) and $t = 1$ (right) | 84 |
| Figure 4.10 Example 4.5.2: Adaptive meshes at $t = 0$, $t = 0.1$ and $t = 1$ (from left to right) | 84 |
| Figure 4.11 Example 4.5.2: Evolution of DoFs (left) and time-steps Δt (right) . | 85 |
| Figure 4.12 Example 4.5.3: Adaptive meshes (top) and solution profiles (bottom) at $t = 0.2$, $t = 0.6$ and $t = 1$ (from left to right) | 86 |
| Figure 4.13 Example 4.5.3: Evolution of DoFs (left) and time-steps Δt (right) . | 87 |
| Figure 4.14 Example 4.5.4: Permeability field (left) and velocity streamlines (right) | 87 |
| Figure 4.15 Example 4.5.4: Evolution of DoFs (left) and time-steps Δt (right) . | 88 |
| Figure 4.16 Example 4.5.4: Adaptive mesh (left) and concentration (right) at $t = 0.3$ | 89 |
| Figure 4.17 Example 4.5.4: Adaptive mesh (left) and concentration (right) at $t = 1$ | 89 |
| Figure A.1 Initial mesh on the unit square $\Omega = [0, 1]^2$ with nodes n_i , triangles E_j and edges e_k | 104 |

CHAPTER 1

INTRODUCTION

Many engineering problems such as chemical reaction processes, heat conduction, nuclear reactors, population dynamics, porous medium in geosciences etc. are governed by coupled convection-diffusion partial differential equations (PDEs) with non-linear source or sink terms. It is a significant challenge to solve such PDEs numerically when they are convection/reaction-dominated. In this thesis, we consider the general convection/reaction dominated coupled semi-linear diffusion-convection-reaction equations of the form

$$\frac{\partial u_i}{\partial t} - \epsilon_i \Delta u_i + \vec{\beta}_i \cdot \nabla u_i + r_i(\vec{u}) = f_i \quad \text{in } \Omega_i \times (0, T], \quad (1.1a)$$

$$u_i(x, t) = g_i^D \quad \text{on } \Gamma_{D,i} \times (0, T], \quad (1.1b)$$

$$\epsilon_i \frac{\partial u_i}{\partial \vec{n}}(x, t) = g_i^N \quad \text{on } \Gamma_{N,i} \times (0, T], \quad (1.1c)$$

$$u_i(x, 0) = u_i^0 \quad \text{in } \Omega_i, \quad i = 1, \dots, m \quad (1.1d)$$

with Ω_i are bounded, open, convex domains in \mathbb{R}^2 with boundaries $\partial\Omega_i = \Gamma_{D,i} \cup \Gamma_{N,i}$, $\Gamma_{D,i} \cap \Gamma_{N,i} = \emptyset$, $0 < \epsilon_i \ll 1$ are the diffusivity constants, $f_i \in L^2(\Omega)$ are the source functions, $\vec{\beta}_i \in (W^{1,\infty}(\Omega))^2$ are the velocity fields, $g_i^D \in H^{3/2}(\Gamma_{D,i})$ are the Dirichlet boundary conditions, $g_i^N \in H^{1/2}(\Gamma_{N,i})$ are the Neumann boundary conditions, $u_i^0 \in L^2(\Omega)$ are the initial conditions and $\vec{u}(x) = (u_1, \dots, u_m)^T$ and \vec{n} denote the vector of unknowns and outward normal vector to the boundary, respectively.

The non-linear reaction terms $r_i(\vec{u})$ occur in chemical engineering usually in the form of products and rational functions of concentrations, or exponential functions of the temperature, expressed by the Arrhenius law. Such models describe chemical processes [15, 16] and they are strongly coupled as an inaccuracy in one unknown af-

fects all the others. The applications in geoscience mostly concern reactive transport with advection [13, 58, 78] and strong permeability contrasts such as layered reservoirs [83] or vanishing and varying diffusivity posing challenges in computations [68]. The permeability in heterogeneous porous and fractured media varies over orders of magnitude in space, which results in highly variable flow field, where the local transport is dominated by advection or diffusion [79]. Accurate and efficient numerical solution of the diffusion-convection-reaction equations to predict the macroscopic mixing, anomalous transport of the solutes and contaminants for a wide range of parameters like permeability and Péclet numbers, different flow velocities and reaction rates are challenging problems [79]. Moreover, being an important class of non-linear convection-diffusion equations, the model (1.1) is used as the pellet equations, which model the intra-particle mass and heat transport in porous catalyst pellets.

For the convection/reaction-dominated diffusion-convection-reaction models, the standard Galerkin finite element methods are known to produce spurious oscillations, especially in the presence of sharp fronts in the solution, on boundary and interior layers. In last two decades, several stabilization and shock/discontinuity capturing techniques were developed for linear and non-linear stationary and time dependent problems of type (1.1). In the linear convection or advection dominated case, the streamline upwind Petrov-Galerkin (SUPG) methods and discontinuous Galerkin (DG) methods are capable of handling the nonphysical oscillations due to the advection. Nevertheless, in the non-linear stationary case, the non-linear reaction term produces sharp layers in addition to the spurious oscillations due to the convection. It is a challenge to resolve such layers accurately and efficiently since the true location of the layers are not known a priori. For the non-linear stationary problems, SUPG is used with the anisotropic shock capturing technique as SUPG-SC for reactive transport problems [15, 16, 17]. It was shown that SUPG-SC is capable of reducing the unphysical oscillations in cross-wind direction. However, the parameters of the SUPG and SUPG-SC should be designed carefully for the efficient solution of the discretized equations. A comprehensive review of weighted residual methods, the orthogonal collocation, Galerkin, tau and least squares methods is given in [77] for solving the linear and non-linear pellet equations, where the methods are compared with respect to convergence of the residuals and computational efficiency. On

the other hands, in the non-stationary case, the resolution of spatial layers is more critical since the nature of the sharp layers may vary as time progresses, and it is mostly possible that there occur also temporal layers in addition to the spatial one.

In contrast to the stabilized continuous Galerkin finite element methods, DG methods produce stable discretizations without the need for extra stabilization strategies, and damp the unphysical oscillations for linear convection dominated problems. The DG combines the best properties of the finite volume and continuous finite elements methods. Finite volume methods can only use lower degree polynomials, and continuous finite elements methods require higher regularity due to the continuity requirements. In [90], several non-linear steady-state convection dominated problems are solved with DG-SC, discontinuous Galerkin method with the shock-capturing technique in [66]. Therein, the construction of the shock-capturing parameters are problem dependent and requires the solution of the system for several times, as in the case of SUPG-SC, and even there still occur some unphysical oscillations. The DG method is in particular suitable for non-matching grids and hp (space and order) adaptivity [52], detecting sharp layers and singularities. They are easily adapted locally for non-conforming finite elements requiring less regularity. Higher order DG approximation can be easily used by hierarchical bases [24], and the convergence rates are limited by the consistency error which makes the DG suitable for complex fluid flows. The DG methods are robust with respect to the variation of the physical parameters like diffusion constant and permeability. The stability of the DG approximation retained by adjusting the penalty parameter to penalize the jumps at the interface of the elements. A unified analysis of the interior penalty DG methods for elliptic PDEs are given in [5]. Other advantages of the DG methods are conservation of mass and fluxes and parallelization. Moreover, DG methods are better suited for adaptive strategies which are based on a posteriori error estimation.

The aim of this thesis is to solve the convection/reaction dominated problems of type (1.1) in an accurate and efficient way using DG method to discretize in space and backward Euler as the time integrator. Our main interest is to derive energy-norm adaptive algorithms based on residual-based a posteriori error estimation both in space and time to save both spatial degrees of freedom (DoFs) and the number of time-steps, yet, to the best of our knowledge, it has not been applied to the non-

stationary semi-linear problems of type (1.1). The aim of a posteriori estimates is to derive bounds to the difference between the computed solution and the unknown true solution, and then using the derived bounds the regions where the error is too large are located and refined adaptively. For the steady-state linear problems, there are a variety of well-understood a posteriori error estimation studies on pure diffusion problems using standard FEMs [3, 84] and DG methods [18, 51, 56], and for the steady-state linear diffusion-convection problems you can see the works [41, 52, 58, 59, 31, 73, 74, 86, 85] and references therein. In case of the non-stationary models, there are valuable studies only on the linear pure diffusion models [42, 45] and linear diffusion-convection equations [35] using DG in space. There are also some studies using interior penalty Galerkin (IPG) methods with efficient time integrators and with space and time adaptivity for a class of linear diffusion-convection equations [34, 33, 36] using energy-norm based constructions.

It is well-known that energy techniques to derive a posteriori estimates in non-stationary problems are challenging. First of all, the arising discrete residual usually does not make sense, and even leads to singular right hand sides. Moreover, it is known that the derived a posteriori estimates are optimal in $L^2(H^1)$ -type norms but suboptimal in $L^\infty(L^2)$ -type norms. In this thesis, we utilize the elliptic reconstruction technique [63] by which we derive a posteriori error estimates by energy techniques for the non-stationary models of the form (1.1) using a posteriori error estimates derived for the stationary (elliptic) case of (1.1), to the best of our knowledge, it has not been applied yet and it will be the major theoretical contribution of this thesis. The idea of the elliptic reconstruction technique is to construct an auxiliary solution whose difference to the numerical solution can be estimated by a known (elliptic) a posteriori estimate, and the constructed auxiliary solution satisfies a variant of the given problem with a right hand side which can be controlled in an optimal way. By this way, in contrast to the standard energy techniques, we do not need to try to adapt the estimates case by case in order to compare the exact solution with numerical solution directly, and we are able to obtain results being optimal order in both $L^2(H^1)$ and $L^\infty(L^2)$ -type norms, while the results obtained by the standard energy methods are only optimal order in $L^2(H^1)$ -type norms, but sub-optimal order in $L^\infty(L^2)$ -type norms.

In the sequel, this thesis is organized as follows. First, in Chapter 2 we give a detailed information about the interior penalty discontinuous Galerkin (IPG) methods which we use in this thesis to discretize the problems in space, and computational tools used in IPG methods such as finite element spaces, basis functions etc. For the construction of the scheme, since IPG schemes concerns with the diffusion part of the problem, we consider the general Poisson problem. We discuss on the effect of the penalty parameter, as well. In the end of Chapter 2, we give the IPG formulation for a general steady-state linear diffusion-convection-reaction equations with upwinding for convection, which forms the key part of the non-linear elliptic model to be introduced in Chapter 3.

In Chapter 3, we introduce the steady-state (elliptic) form of the non-stationary model 1.1. We give the existence and uniqueness results concerning with the elliptic semi-linear diffusion-convection-reaction equations using the coercivity results of the arising bilinear form shown in Chapter 2 and the assumptions on the non-linear reaction term, monotonicity, locally Lipschitz continuity, boundedness etc. Then, being the major contribution of this thesis, we study the adaptivity tool for elliptic semi-linear problems. We derive and prove the in diffusion parameter (or in Péclet number) robust a posteriori error bounds, which is one of the main theoretical contribution of this thesis. To derive a posteriori bounds, we utilize the robust a posteriori error estimation used in [74] for steady-state linear diffusion-convection equations. Because of the fact that the condition number of the arising stiffness matrix grows rapidly with the number of elements and with the penalty parameter in DG methods, efficient solution strategies such as preconditioning are required to solve the linear systems. While more robust compared to iterative solvers, direct solvers are usually more memory and time consuming due to fill-in. Furthermore, they are known to be less scalable on parallel architectures. Therefore, in Chapter 3, we introduce an iterative method called matrix reordering [82] which is robust and efficient. In numerical studies, we will demonstrate the efficiency of the matrix reordering iterative method in detailed by comparing the CPU times, number of iteration numbers etc. with the ones obtained by the direct solvers. Further, in Chapter 3, we will compare the adaptive DG approximations with the approximations by a famous stabilized FEM, Galerkin least squares FEM [22, 21, 54] which also owns the method SUPG. Chapter 3 ends with some nu-

merical studies on the stationary semi-linear models by which we demonstrate that the adaptive DG schemes are capable of resolving the solution of the elliptic problems at the layers, and they are more accurate comparing to the other methods such as SUPG-SC [15, 16] and DG-SC [90].

In Chapter 4, we begin by introducing the DG discretized semi-discrete formulation of the model (1.1), and we give the existence and uniqueness results. Then, we introduce the elliptic reconstruction technique [63] in details by discussing the usual energy techniques and why elliptic reconstruction technique is needed. As a result of elliptic reconstruction technique, we state a posteriori error bounds for the semi-discrete system using a posteriori error bounds driven and proven in Chapter 3 for stationary problems. Next, after stating the in time backward Euler discretized fully-discrete formulation of the model problem (1.1), we introduce a posteriori error bounds in space SIPG and in time backward Euler discretized fully-discrete system of (1.1) utilizing the related bounds for the semi-discrete system using a $L^\infty(L^2) + L^2(H^1)$ -type norm, and we give the modification of in diffusion parameter robust adaptive algorithm chart both in space and time [25] to the non-stationary semi-linear problems of type (1.1), and also we state the solution of linear systems arising from the Newton's method algebraically. Chapter 4 followed by the numerical studies demonstrating the effectiveness of the adaptive algorithm. We show that our adaptive algorithm is robust in diffusion parameter, as for the linear non-stationary diffusion-convection-reaction equations given in [25], by demonstrating both the spatial and temporal effectivity indices and rates of errors and estimators for various diffusion parameters. We also point out in some examples that our adaptive algorithm is capable of catching not only the spatial layers but also the temporal layers by decreasing the time-step size there. In addition, by some examples of flow transport, we figure out that besides the refinement procedure, the coarsening procedure of the adaptive algorithm works effectively, which is crucial to save time and DoFs efficiently. Finally, this thesis ends with conclusions and future works in Chapter 5. In Appendix A, you can find a MATLAB routine to solve the semi-linear diffusion-convection-reaction equations using DG methods to discretize in space.

CHAPTER 2

DISCONTINUOUS GALERKIN METHODS

The first occurrence of the discontinuous Galerkin (DG) method goes over to the 1973 when Reed and Hill [69] introduced it for steady-state neutron transport being a first-order hyperbolic problem. This study have set light to the development of DG methods to use in nearly hyperbolic problems. The most known studies are the work done by Bassi and Rebay [12] for the compressible Navier-Stokes equations, study of Cockburn and Shu [29] on the local discontinuous Galerkin (LDG) method for diffusion-convection equations, and Peraire and Persson [65] who have introduced the compact discontinuous Galerkin (CDG) method. Apart from the DG method, interior penalty (IP) methods have their own development route in the literature, which is first introduced by Douglas and Dupont [38] and have developed for elliptic and parabolic problems later on. Wheeler [88] have also contributed the development of DG methods for elliptic and parabolic problems in seventies. Then, in eighties, Arnold et. al. [6] proposed a unified analysis of several DG methods. In the sequel, DG methods have been also developed for elliptic problems in [8, 23, 72] and for the ones with advection in [7, 14, 47, 52].

In the last decade, DG methods have become so popular being mostly an alternate method to the finite volume method (FVM) which have been used as a major method to simulate the problems in industry and has a lack of ability to use higher order approximations. DG methods combine the best properties of the FVMs and continuous finite elements methods. FVMs can only use lower degree polynomials, and continuous finite elements methods require higher regularity due to the continuity requirements. Together with the nice properties of FVMs and continuous FEMs, DG

methods have a number of useful properties. The so called properties are the consistency, flexibility, stability, conservation of local quantities, robustness and compactness. First of all, DG methods are simply a type of FEMs. Hence, the consistency of DG methods can be easily interpreted using the Galerkin orthogonality property of FEMs. The flexibility of DG methods comes from the fact that the functions in DGFEM space are discontinuous along the inter-element boundaries, which is a key point in the generation of unstructured meshes with hanging nodes and on the construction of higher order basis functions. It allows to easily use different order of polynomials on different elements since the supports of the functions in DGFEM space are just a single element and there is no overlapping between the elements as in the classical FEMs. Thus, DG methods can be easily used in p -refinement schemes, where the order of the polynomials on the elements having steep gradient are arranged adaptively. The stability of DG methods are handled via the penalty term which penalizes the jumps of the solution on the element boundaries. In this way, the stability in DG methods are inherited and one need not to propose additional stabilization techniques which is the case in the classical FEMs such as streamline upwind Petrov-Galerkin (SUPG) method being the most popular method used in convection dominated problems. The local properties of DG methods allows to locally conserve several physical quantities such as mass and energy, which plays an important role in the flow and transport problems, and DG methods lead to mass matrices with block diagonal structure making DG methods useful in stiff ODEs arising from the semi-discretization of initial-boundary value problems. Further, by the local construction of DG methods, one can locally detect the sharp layers and the singularities singularities of the solution in the mesh via the fully discontinuous polynomial representation of the solution, which makes DG methods extremely convenient for the adaptive h -refinement and also hp -refinement schemes [76]. The DG methods are able to be designed so that the resulting numerical scheme is robust with respect to the problem parameters such as the case of the perturbation of the diffusion parameter. By this ability, at least it is expected that the perturbation of the so called parameter will affect the solution only locally (not globally) and it will keep unperturbed in the remaining field. In addition, the (Dirichlet) boundary conditions in DG methods are imposed in a weak manner. In this way, one not only have a scheme which is robust with respect to the boundary condition but also do not need to construct finite element spaces with

certain conditions on the boundary. All these mentioned properties make DG methods have a compact formulation which is a key ingredient in parallel computing where it is important to minimize the data exchange in parallel executions. Besides all the advantages, DG methods have some drawbacks. Compared to the continuous finite elements methods, DG methods produce systems with larger degrees of freedom and ill-conditioned matrices increasing linearly with the order of basis functions.

In this chapter, we give a detailed information about the interior penalty discontinuous Galerkin (IPG) methods which we use in this thesis to discretize the problems in space. As a starting point of the construction, we consider the general Poisson problem in Section 2.2 since the construction of IPG schemes concerns with the diffusion part of the problem. Then, in Section 2.3, we will discuss the computational tools used in IPG methods such as finite element spaces, basis functions etc. Being the key term of IPG methods, the effect of the penalty parameter will be analyzed in Section 2.4. In the end of the chapter, we give the arising scheme for a general steady-state linear diffusion-convection-reaction equations in Section 2.5, which forms the stationary linear part of the scalar form of the model problem (1.1) used in this thesis, and we show the coercivity of the arising bilinear form which is needed in the sequential chapters.

2.1 Preliminaries

In this section, we introduce some useful definitions and identities required in the construction of IPG schemes and to show the coercivity of the bilinear form arising from the IPG scheme.

2.1.1 Sobolev Spaces

On a polygonal domain Ω in \mathbb{R}^d , the spaces $L^p(\Omega)$ of p-integrable functions are defined by

$$L^p(\Omega) = \{v \text{ Lebesgue measurable} : \|v\|_{L^p(\Omega)}^2 < \infty\}, \quad 1 \leq p \leq \infty$$

equipped with the norms

$$\begin{aligned}\|v\|_{L^p(\Omega)} &= \left(\int_{\Omega} |v(x)|^p dx\right)^{1/p}, \quad 1 \leq p < \infty, \\ \|v\|_{L^\infty(\Omega)} &= \operatorname{ess\,sup}\{|v(x)| : x \in \Omega\}, \quad p = \infty.\end{aligned}$$

We mainly consider the space $L^2(\Omega)$ which is a Hilbert space equipped with the usual L^2 -inner product

$$(u, v)_\Omega = \int_{\Omega} u(x)v(x)dx, \quad \|v\|_{L^2(\Omega)} = \sqrt{(v, v)_\Omega}.$$

Let $\mathcal{D}(\Omega)$ denotes the subspace of the space C^∞ having compact support in Ω . For any multi-index $\alpha = (\alpha_1, \dots, \alpha_d) \in \mathbb{N}^d$ with $|\alpha| = \sum_{i=1}^d \alpha_i$, the distributional derivative $D^\alpha v$ is defined by

$$D^\alpha v(\psi) = (-1)^{|\alpha|} \int_{\Omega} v(x) \frac{\partial^{|\alpha|} \psi}{\partial x_1^{\alpha_1} \dots \partial x_d^{\alpha_d}}, \quad \forall \psi \in \mathcal{D}(\Omega).$$

Then, we introduce for an integer s the Sobolev spaces

$$H^s(\Omega) = \{v \in L^2(\Omega) : D^\alpha v \in L^2(\Omega), \forall 0 \leq |\alpha| \leq s\}$$

with the associated Sobolev norm

$$\|v\|_{H^s(\Omega)} = \left(\sum_{0 \leq |\alpha| \leq s} \|D^\alpha v\|_{L^2(\Omega)}^2 \right)^{1/2},$$

and the associated Sobolev seminorm

$$|v|_{H^s(\Omega)} = \|\nabla^s v\|_{L^2(\Omega)} = \left(\sum_{|\alpha|=s} \|D^\alpha v\|_{L^2(\Omega)}^2 \right)^{1/2}.$$

The Sobolev spaces with vanishing functions on the domain boundary are defined by

$$H_0^s(\Omega) = \{v \in H^s(\Omega) : v|_{\partial\Omega} = 0\}.$$

We are mainly interested in th case $s = 1$

$$H^1(\Omega) = \{v \in L^2(\Omega) : \nabla v \in (L^2(\Omega))^d\}.$$

Moreover, for a partition (most possibly triangles) ξ_h of Ω , we define the broken Sobolev spaces by

$$H^s(\xi_h) = \{v \in L^2(\Omega) : v|_K \in H^s(K), \forall K \in \xi_h\}$$

with the associated broken Sobolev norm

$$\|v\|_{H^s(\xi_h)} = \left(\sum_{K \in \xi_h} \|v\|_{H^s(K)}^2 \right)^{1/2},$$

and the associated broken gradient seminorm

$$|v|_{H^0(\xi_h)} = \left(\sum_{K \in \xi_h} \|\nabla v\|_{L^2(K)}^2 \right)^{1/2}.$$

2.1.2 Trace Theorems

Theorem 2.1.1. [70, Theorem 2.5] *For $s_0 > 1/2$ and $s_1 > 3/2$, there exist trace operators $\gamma_0 : H^{s_0}(\Omega) \mapsto H^{s_0-1/2}(\partial\Omega)$ and $\gamma_1 : H^{s_1}(\Omega) \mapsto H^{s_1-3/2}(\partial\Omega)$ being extensions of the boundary values and boundary normal derivatives, respectively, with polygonal boundary $\partial\Omega$, and for $v \in C^1(\bar{\Omega})$, we have*

$$\gamma_0 v = v|_{\partial\Omega}, \quad \gamma_1 v = \nabla v \cdot \vec{n}|_{\partial\Omega}.$$

In the above theorem, the space $H^{s-1/2}(\partial\Omega)$ ($H^{s-3/2}(\partial\Omega)$) is the space of completion of all functions in $H^s(\partial\Omega)$ ($H^{s-1}(\partial\Omega)$) with the property

$$H^s(\partial\Omega) \subset H^{s-1/2}(\partial\Omega) \subset H^{s-1}(\partial\Omega).$$

For instance, when $s = 2$, γ_0 (γ_1) belongs to the space $H^{3/2}(\partial\Omega)$ ($H^{1/2}(\partial\Omega)$) being the interpolated space between the spaces $H^2(\partial\Omega)$ ($H^1(\partial\Omega)$) and $H^1(\partial\Omega)$ ($L^2(\partial\Omega)$). As a consequence of the above theorem, we have the trace inequalities

$$\|v\|_{L^2(e)} \leq C_{Tr} h_K^{-1/2} \|v\|_{L^2(K)}, \quad (2.1a)$$

$$\|\nabla v \cdot \vec{n}\|_{L^2(e)} \leq C_{Tr} h_K^{-1/2} \|\nabla v\|_{L^2(K)}, \quad (2.1b)$$

where h_K denotes the diameter of an element K , and the positive constant C_{Tr} is independent of h_K .

2.1.3 Cauchy-Schwarz's and Young's Inequalities

The following inequalities are the most used identities in the analysis of FEMs, which we use in this thesis, as well.

- Cauchy-Schwarz's inequality: For any $u, v \in L^2(\Omega)$

$$|(u, v)_{L^2(\Omega)}| \leq \|u\|_{L^2(\Omega)} \|v\|_{L^2(\Omega)}. \quad (2.2)$$

- Young's inequality: For any $\delta > 0$ and for any $a, b \in \mathbb{R}$

$$ab \leq \frac{\delta}{2} a^2 + \frac{1}{2\delta} b^2. \quad (2.3)$$

2.2 Construction of IPG Methods

In this section, we give the detailed construction of the IPG methods [5, 70] applied to the general Poisson equation

$$-\epsilon \Delta u = f \quad \text{in } \Omega \subset \mathbb{R}^2, \quad (2.4a)$$

$$u = g^D \quad \text{on } \Gamma_D, \quad (2.4b)$$

$$\epsilon \nabla u \cdot \vec{n} = g^N \quad \text{on } \Gamma_N, \quad (2.4c)$$

with $\partial\Omega = \Gamma_D \cup \Gamma_N$ and $\Gamma_D \cap \Gamma_N = \emptyset$. Using the Poisson equation as a starting point is meaningful since the IPG methods are applied to the diffusion parts of the problems.

Let the mesh $\xi_h = \{K\}$ be a family of shape regular elements (triangles), i.e. for some positive constant h_0 there holds

$$\max_{K \in \xi_h} \frac{h_K^2}{|K|} \leq h_0,$$

where h_K and $|K|$ denote the diameter and the area of the element K , respectively. Let also that $\bar{\Omega} = \cup \bar{K}$ and $K_i \cap K_j = \emptyset$ for $K_i, K_j \in \xi_h$. Denote by Γ_h^0, Γ_h^D and Γ_h^N the set of interior, Dirichlet boundary and Neumann boundary edges, respectively, so that $\Gamma_h^0 \cup \Gamma_h^D \cup \Gamma_h^N$ forms the skeleton of the mesh. For any $K \in \xi_h$, let $\mathbb{P}_k(K)$ be the set of all polynomials of degree at most k on K . Then, set the finite dimensional solution and test function space by

$$V_h = \{v \in L^2(\Omega) : v|_K \in \mathbb{P}_k(K), \forall K \in \xi_h\} \not\subset H_0^1(\Omega).$$

Note that the trial and test function spaces are the same because the boundary conditions in discontinuous Galerkin methods are imposed in a weak manner (see the

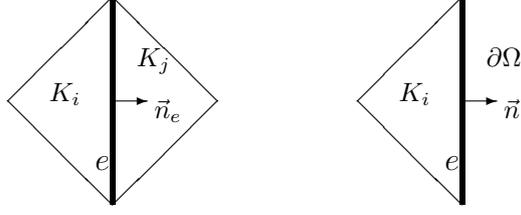


Figure 2.1: Two adjacent elements sharing an edge (left); an element near to domain boundary (right)

IPG construction below). The classical (continuous) FEM uses a conforming, finite-dimensional subspace $V_h \subset H_0^1(\Omega)$, which requires that the space V_h contains functions of particular smoothness (e.g. $V_h \subset \{v \in C(\bar{\Omega}) : v = 0 \text{ on } \partial\Omega\}$). On the other hands, discontinuous Galerkin methods make it easy to use the non-conforming spaces, in which case the functions in $V_h \not\subset H_0^1(\Omega)$ are allowed to be discontinuous on the inter-element boundaries.

Since the functions in V_h are discontinuous along the inter-element boundaries, along an interior edge, there are two different traces from the adjacent elements sharing that edge. In the light of this fact, let us first introduce some notations before starting the construction of IPG formulation. Let $K_i, K_j \in \xi_h$ ($i < j$) be two adjacent elements sharing an interior edge $e = K_i \cap K_j \subset \Gamma_h^0$ (see Fig.2.1). Denote the trace of a scalar function v from inside K_i by v_i and from inside K_j by v_j . Then, set the jump and average values of v on the edge e

$$[v] = v_i \vec{n}_e - v_j \vec{n}_e, \quad \{v\} = \frac{1}{2}(v_i + v_j),$$

where \vec{n}_e is the unit normal to the edge e oriented from K_i to K_j . Similarly, we set the jump and average values of a vector valued function \vec{q} on e

$$[\vec{q}] = \vec{q}_i \cdot \vec{n}_e - \vec{q}_j \cdot \vec{n}_e, \quad \{\vec{q}\} = \frac{1}{2}(\vec{q}_i + \vec{q}_j).$$

Observe that $[v]$ is a vector for a scalar function v , while, $[\vec{q}]$ is scalar for a vector valued function \vec{q} . On the other hands, along any boundary edge $e = K_i \cap \partial\Omega$, we set

$$[v] = v_i \vec{n}, \quad \{v\} = v_i, \quad [\vec{q}] = \vec{q}_i \cdot \vec{n}, \quad \{\vec{q}\} = \vec{q}_i,$$

where \vec{n} is the unit outward normal to the boundary at e .

Now, we are ready to construct the IPG discretization of the diffusion part of the problem. We multiply the continuous equation (2.4a) by a test function $v \in V_h$, we integrate over Ω and we split the integrals leading to

$$-\sum_{K \in \xi_h} \int_K \epsilon \Delta u v dx = \sum_{K \in \xi_h} \int_K f v dx.$$

Applying the divergence theorem on every element integral gives

$$\sum_{K \in \xi_h} \int_K \epsilon \nabla u \cdot \nabla v dx - \sum_{K \in \xi_h} \int_{\partial K} \epsilon (\nabla u \cdot \vec{n}) v ds = \sum_{K \in \xi_h} \int_K f v dx + \sum_{e \in \Gamma_h^N} \int_e g^N v ds.$$

Or using the jump definitions ($v \in V_h$ are element-wise discontinuous), we may have

$$\sum_{K \in \xi_h} \int_K \epsilon \nabla u \cdot \nabla v dx - \sum_{e \in \Gamma_h^0 \cup \Gamma_h^D} \int_e [\epsilon v \nabla u] ds = \sum_{K \in \xi_h} \int_K f v dx + \sum_{e \in \Gamma_h^N} \int_e g^N v ds.$$

One can easily verify that $[\epsilon v \nabla u] = \{\epsilon \nabla u\} \cdot [v] + [\epsilon \nabla u] \cdot \{v\}$. Then, using also the fact that $[\nabla u] = 0$ (u is assumed to be smooth enough so that ∇u is continuous), we get

$$\sum_{K \in \xi_h} \int_K \epsilon \nabla u \cdot \nabla v dx - \sum_{e \in \Gamma_h^0 \cup \Gamma_h^D} \int_e \{\epsilon \nabla u\} \cdot [v] ds = \sum_{K \in \xi_h} \int_K f v dx + \sum_{e \in \Gamma_h^N} \int_e g^N v ds.$$

Yet, the left hand side may not be coercive. To handle this and to penalize the solutions on the inter-element boundaries, noting that $[u] = 0$ along the interior edges (u is assumed to be continuous), we reach at

$$\begin{aligned} & \sum_{K \in \xi_h} \int_K \epsilon \nabla u \cdot \nabla v dx - \sum_{e \in \Gamma_h^0 \cup \Gamma_h^D} \int_e \{\epsilon \nabla u\} \cdot [v] ds + \kappa \sum_{e \in \Gamma_h^0} \int_e \{\epsilon \nabla v\} \cdot [u] ds \\ & + \sum_{e \in \Gamma_h^0} \frac{\sigma \epsilon}{h_e} \int_e [u] \cdot [v] ds = \sum_{K \in \xi_h} \int_K f v dx + \sum_{e \in \Gamma_h^N} \int_e g^N v ds, \end{aligned}$$

where h_e denote the length of the edge e and σ is the penalty parameter. Finally, we add to the both sides the edge integrals on the Dirichlet boundary edges (keeping unknown on the left hand side and imposing Dirichlet boundary condition on the right hand side)

$$\sum_{K \in \xi_h} \int_K \epsilon \nabla u \cdot \nabla v dx - \sum_{e \in \Gamma_h^0 \cup \Gamma_h^D} \int_e \{\epsilon \nabla u\} \cdot [v] ds + \kappa \sum_{e \in \Gamma_h^0 \cup \Gamma_h^D} \int_e \{\epsilon \nabla v\} \cdot [u] ds$$

$$\begin{aligned}
+ \sum_{e \in \Gamma_h^0 \cup \Gamma_h^D} \frac{\sigma \epsilon}{h_e} \int_e [u] \cdot [v] ds &= \sum_{K \in \xi_h} \int_K f v dx + \sum_{e \in \Gamma_h^D} \int_e g^D \left(\frac{\sigma \epsilon}{h_e} v + \kappa \epsilon \nabla v \cdot \vec{n} \right) ds \\
&+ \sum_{e \in \Gamma_h^N} \int_e g^N v ds, \tag{2.5}
\end{aligned}$$

which gives the IPG formulation. The parameter κ in the IPG formulation determines the type of the IPG method. It varies on the values $\kappa \in \{-1, 0, 1\}$ giving that

- $\kappa = -1$: Symmetric interior penalty Galerkin (SIPG) method,
- $\kappa = 1$: Non-symmetric interior penalty Galerkin (NIPG) method,
- $\kappa = 0$: Incomplete interior penalty Galerkin (IIPG) method.

In this thesis, we only consider the symmetric interior penalty Galerkin (SIPG) method by setting $\kappa = -1$.

Proposition 2.2.1. *The equivalence (consistency) of the model problem (2.4) and IPG variational problem (2.5) is obvious by the construction of the IPG formulation above if we assume that the solution u of the model problem (2.4) satisfies $u \in H^s(\Omega)$ for some $s > 3/2$, which provides a sufficient smoothness of the solution needed in the IPG construction.*

2.3 Computation Tools for Integral Terms

In this section, we give some useful tools used to compute integrals on physical elements. Firstly, we mention about the *reference element* approach which is a common tool among the whole finite elements method. Then, we discuss the basis functions used in IPG methods, and also the numerical quadrature concept.

2.3.1 Reference Element

It is well-known that computing the integrals on the physical elements is difficult and costly. The common technique in FEMs, instead, is to compute all the integrals on a reference element and moving them to the physical elements.

We use the unit triangle on the first quadrant as the reference triangle \hat{K} with vertices $\hat{A}_1 = (0, 0)$, $\hat{A}_2 = (1, 0)$, $\hat{A}_3 = (0, 1)$, while, a physical element K has the vertices

$A_i(x_i, y_i)$ for $i = 1, 2, 3$. To compute the integrals on the physical elements, we utilize the invertible affine mapping $F_K : \hat{K} \mapsto K$ defined by

$$F_K \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix}, \quad x = \sum_{i=1}^3 x_i \hat{\psi}_i(\hat{x}, \hat{y}), \quad y = \sum_{i=1}^3 y_i \hat{\psi}_i(\hat{x}, \hat{y})$$

with the shape functions

$$\hat{\psi}_1(\hat{x}, \hat{y}) = 1 - \hat{x} - \hat{y}, \quad \hat{\psi}_2(\hat{x}, \hat{y}) = \hat{x}, \quad \hat{\psi}_3(\hat{x}, \hat{y}) = \hat{y}.$$

One may rewrite the mapping as

$$\begin{pmatrix} x \\ y \end{pmatrix} = F_K \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = B_K \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} + b_K,$$

where B_K is a non-singular matrix and b_K is a translation vector given by

$$B_K = \begin{pmatrix} a_{11}^K & a_{12}^K \\ a_{21}^K & a_{22}^K \end{pmatrix} = \begin{pmatrix} x_2 - x_1 & x_3 - x_1 \\ y_2 - y_1 & y_3 - y_1 \end{pmatrix}, \quad b_K = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}.$$

Thus, the inverse of the affine map F_K can be defined explicitly by

$$F_K^{-1} : K \mapsto \hat{K} : \quad F_K^{-1}(x) = B_K^{-1}(x - b_K) = \hat{x},$$

where the inverse matrix B_K^{-1} is given by

$$B_K^{-1} = \frac{1}{\det B_K} \begin{pmatrix} a_{22}^K & -a_{12}^K \\ -a_{21}^K & a_{11}^K \end{pmatrix} = \frac{1}{2|K|} \begin{pmatrix} \hat{a}_{11}^K & \hat{a}_{12}^K \\ \hat{a}_{21}^K & \hat{a}_{22}^K \end{pmatrix} = \frac{1}{2|K|} \hat{B}_K.$$

All the above definitions yields the identities between the functions on the reference element and the functions on the physical elements as

$$\begin{aligned} \hat{v}(\hat{x}, \hat{y}) &= v(x, y), \\ \hat{\nabla} \hat{v}(\hat{x}, \hat{y}) &= B_K^T \nabla v(x, y). \end{aligned}$$

2.3.2 Numerical Quadrature

The flexibility of DG methods leads to use of high order polynomials, while, the explicit integral formulation is so complicated with high order polynomials. Thus, the use of high order quadrature rules in order to compute the integrals exactly is

necessary. We use the numerical quadrature rule in [39] to approximate the integrals on the reference element \hat{K} , having the form

$$\int_{\hat{K}} \hat{v} \approx \sum_{j=1}^{N_q} w_j \hat{v}(s_{x,j}, s_{y,j}),$$

where w_j 's denote the quadrature weights and $(s_{x,j}, s_{y,j}) \in \hat{K}$ are the quadrature nodes inside the reference element. The use of the affine map F_K with the quadrature formula above leads to the computation of the integrals on a physical element K as

$$\begin{aligned} \int_K v &= \int_{\hat{K}} v \circ F_K \det(B_K) = 2|K| \int_{\hat{K}} \hat{v} \approx 2|K| \sum_{j=1}^{N_q} w_j \hat{v}(s_{x,j}, s_{y,j}), \\ \int_K \nabla v \cdot w &\approx 2|K| \sum_{j=1}^{N_q} w_j (B_K^T)^{-1} \hat{\nabla} \hat{v}(s_{x,j}, s_{y,j}) \cdot \hat{w}(s_{x,j}, s_{y,j}), \\ \int_K \nabla v \cdot \nabla w &\approx 2|K| \sum_{j=1}^{N_q} w_j (B_K^T)^{-1} \hat{\nabla} \hat{v}(s_{x,j}, s_{y,j}) \cdot (B_K^T)^{-1} \hat{\nabla} \hat{w}(s_{x,j}, s_{y,j}). \end{aligned}$$

2.3.3 Basis Functions

The functions in DG solution space V_h do not require to be continuous along the inter-element boundaries. This property of DG methods provides a flexibility to choose and construct the basis functions since the basis functions in DG space V_h have a support just a single element K on which they are defined, and they vanish outside that element. Such a construction results in the setting

$$V_h = \text{span}\{\psi_i^K : 1 \leq i \leq N_{loc}, K \in \xi_h\}$$

with the global basis functions

$$\psi_i^K(x) = \begin{cases} \hat{\psi}_i \circ F_K(x), & \text{if } x \in K, \\ 0, & \text{if } x \notin K, \end{cases}$$

where $\{\hat{\psi}_i\}$'s are the local basis functions defined on the reference element \hat{K} and N_{loc} denotes the local dimension depending on the order k of the polynomial basis functions which is set $N_{loc} = (k+1)(k+2)/2$ in 2D.

There are a variety of basis functions such as Lagrange shape functions, monomial bases, Legendre polynomials etc. In this thesis, we use the orthogonal Dubiner basis

[30] defined on the reference triangle

$$\hat{K} = \{x = (x_1, x_2) | 0 \leq x_1, x_2 \leq 1\}.$$

The construction of such basis polynomials based on the collapsed coordinate transform between the reference triangle \hat{K} and the reference square $\hat{Q} = [-1, 1]^2$ (see Fig.2.2). First, the basis polynomials on the square \hat{Q} is formed by a generalized tensor product of the Jacobi polynomials on the interval $[-1, 1]$, and then, these basis

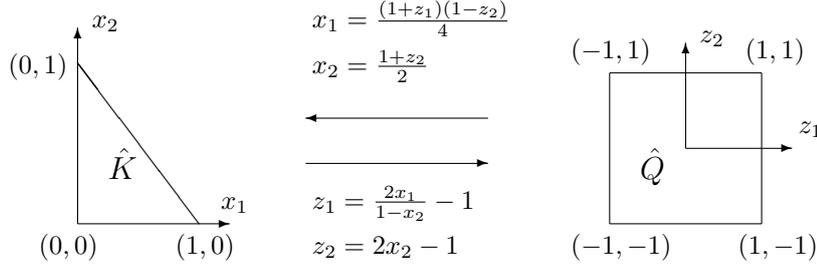


Figure 2.2: Collapsed coordinate transform between reference triangle and reference square

polynomials are transformed to the reference triangle \hat{K} using the collapsed coordinate transform in Fig.2.2. The explicit forms of Dubiner basis polynomials on the reference triangle \hat{K} are given by

$$\begin{aligned} \phi_{mn}(x_1, x_2) &= (1 - z_2)^m P_m^{0,0}(z_1) P_n^{2m+1,0}(z_2) \\ &= 2^m (1 - x_2)^m P_m^{0,0}\left(\frac{2x_1}{1-x_2} - 1\right) P_n^{2m+1,0}(2x_2 - 1), \end{aligned}$$

for $0 \leq m, n, m + n \leq Nloc$, where $P_n^{\alpha,\beta}(x)$'s denote the corresponding n -th order Jacobi polynomials on the interval $[-1, 1]$, which are orthogonal polynomials under the Jacobi weight $(1 - x)^\alpha(1 + x)^\beta$, i.e.

$$\int_{-1}^1 (1 - x)^\alpha (1 + x)^\beta P_m^{\alpha,\beta}(x) P_n^{\alpha,\beta}(x) dx = \delta_{mn}.$$

This property of the Jacobi polynomials yields the orthogonality of the Dubiner basis on the reference triangle \hat{T} as

$$\iint_{\hat{T}} \phi_{mn}(x_1, x_2) \phi_{ij}(x_1, x_2) dx_1 dx_2 = \frac{1}{8} \delta_{mi} \delta_{nj}.$$

The advantage of the Dubiner basis is that its orthogonality leads to diagonal mass matrix and better-conditioned stiffness matrix compared to the other basis polynomi-

als (see Fig.2.3), and for higher order polynomial basis, it provides high accuracy in the approximation of the integrals.

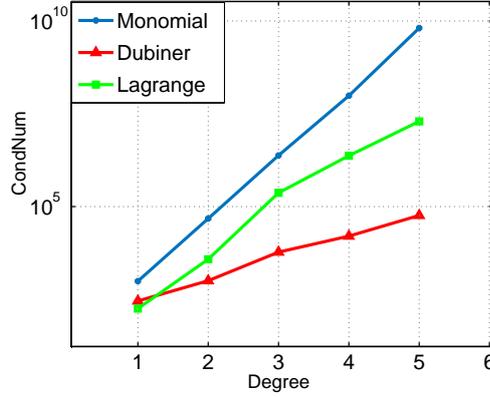


Figure 2.3: Degree vs. condition number of the stiffness matrix: comparison for different type of basis functions for the Poisson problem (2.4)

2.4 Effect of Penalty Parameter

The penalty parameter σ in the SIPG formulation should be selected sufficiently large to ensure the coercivity of the bilinear form [70, Sec. 27.1], which is needed for the stability of the convergence of the SIPG method. It ensures that the matrix arising from the SIPG discretization of the diffusion part is symmetric positive definite. At the same time it should not be too large since the conditioning of the matrix arising from the bilinear form increases linearly by the penalty parameter (see Fig.2.4, left). In the literature, several choices of the penalty parameter are suggested. In [40], computable lower bounds are derived, and in [32], the penalty parameter is chosen depending on the diffusion coefficient ϵ . The effect of the penalty parameter on the condition number was discussed in detail for the DG discretization of the Poisson equation in [26] and in [83] for layered reservoirs with strong permeability contrasts, e.g. ϵ varying between 10^{-1} and 10^{-7} .

To examine the effect of the penalty parameter, we study on the Poisson problem (2.4a) with the appropriate load function f , diffusion constant $\epsilon = 1$ and Dirichlet boundary conditions using the exact solution $u(x) = \sin(\pi x_1) \sin(\pi x_2)$. In Fig.2.5, we have plotted the maximum nodal errors depending on the penalty parameter to

show the instability bound of the scheme for different degrees of bases, where the triangular symbols indicate our choice $\sigma = 3k(k + 1)$.

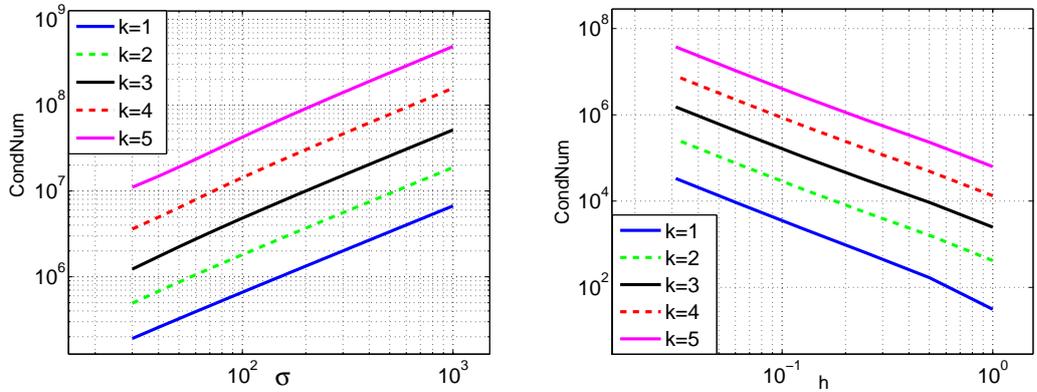


Figure 2.4: Condition number of the stiffness matrix of the SIPG method as the functions of the penalty parameter σ (left) and the mesh-size h (right) with different polynomial degree k for the Poisson problem (2.4)

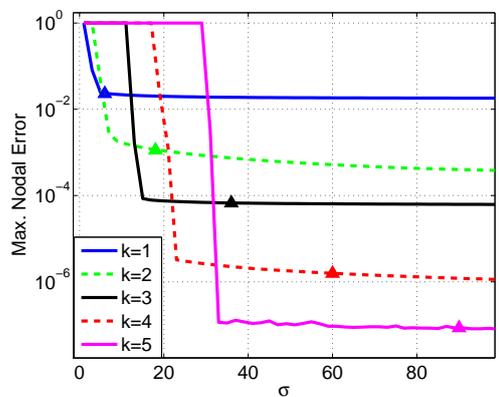


Figure 2.5: Maximum nodal errors of the SIPG approximation as a function of penalty parameter σ with different polynomial degree k for the Poisson problem (2.4)

Similarly, the condition number of the stiffness matrix increases with decreasing mesh-size and increasing order of the DG discretization (see Fig.2.4, right), which affects the efficiency of an iterative solver. Similar results can be found in [26].

2.5 Problems with Convection

In this Section, we consider the scalar diffusion-convection equation of the form

$$-\epsilon \Delta u + \vec{\beta} \cdot \nabla u + \alpha u = f \quad \text{in } \Omega, \quad (2.6a)$$

$$u = g^D \quad \text{on } \Gamma_D, \quad (2.6b)$$

$$\epsilon \frac{\partial u}{\partial \vec{n}} = g^N \quad \text{on } \Gamma_N \quad (2.6c)$$

with Ω is a bounded, open, convex domain in \mathbb{R}^2 with boundary $\partial\Omega = \Gamma_D \cup \Gamma_N$, $\Gamma_D \cap \Gamma_N = \emptyset$, $0 < \epsilon \ll 1$ is the diffusivity constant, $f \in L^2(\Omega)$ is the source function, $\vec{\beta} \in (W^{1,\infty}(\Omega))^2$ is the velocity field, $g^D \in H^{3/2}(\Gamma_D)$ and $g^N \in H^{1/2}(\Gamma_N)$ are the Dirichlet and Neumann boundary conditions, respectively. The linear reaction coefficient α is a positive number which mimics the temporal discretization parameter $1/\Delta t$, where Δt is the time-step size. For the well-posedness of the problem, we also assume for some non-negative constant α_0 that

$$\alpha - \frac{1}{2} \nabla \cdot \vec{\beta}(x) \geq \alpha_0 \geq 0. \quad (2.7)$$

In order to discretize the problem (2.6), we apply SIPG formulation to the diffusion part given in the previous section, and the original upwinding scheme [61, 69] to the convection part. With the aim of upwinding, let us decompose the boundary edges into the set Γ^- of inflow edges and the set Γ^+ of outflow edges defined by

$$\Gamma_h^- = \{x \in \partial\Omega : \vec{\beta} \cdot \vec{n} < 0\}, \quad \Gamma_h^+ = \partial\Omega \setminus \Gamma_h^-,$$

where \vec{n} is the unit outward normal to the boundary $\partial\Omega$. The set of inflow and outflow boundary edges of an element $K \in \xi_h$ is defined in a similar way by

$$\partial K^- = \{x \in \partial K : \vec{\beta} \cdot \vec{n}_K < 0\}, \quad \partial K^+ = \partial K \setminus \partial K^-,$$

where \vec{n}_K is the unit outward normal vector to the element boundary ∂K . Moreover, on an interior edge ∂K , we denote the trace of a function v from inside the element K by v^{in} and from outside the element K by v^{out} . Then, SIPG with upwinding discretized formulation of (2.6) reads as: find $u_h \in V_h$ such that

$$a_h(u_h, v_h) = l_h(v_h) \quad \forall v_h \in V_h, \quad (2.8)$$

with the bilinear form

$$a_h(u_h, v_h) = a_h^{DR}(u_h, v_h) + a_h^C(u_h, v_h), \quad (2.9)$$

where

$$\begin{aligned} a_h^{DR}(u_h, v_h) &= \sum_{K \in \xi_h} \int_K (\epsilon \nabla u_h \cdot \nabla v_h + \alpha u_h v_h) dx - \sum_{e \in \Gamma_h^0 \cup \Gamma_h^D} \int_e \{\epsilon \nabla u_h\} \cdot [v_h] ds \\ &\quad - \sum_{e \in \Gamma_h^0 \cup \Gamma_h^D} \int_e \{\epsilon \nabla v_h\} \cdot [u_h] ds + \sum_{e \in \Gamma_h^0 \cup \Gamma_h^D} \frac{\sigma \epsilon}{h_e} \int_e [u_h] \cdot [v_h] ds, \\ a_h^C(u_h, v_h) &= \sum_{K \in \xi_h} \int_K \vec{\beta} \cdot \nabla u_h v_h dx + \sum_{K \in \xi_h} \int_{\partial K^- \setminus \partial \Omega} \vec{\beta} \cdot \vec{n} (u_h^{out} - u_h^{in}) v_h ds \\ &\quad - \sum_{K \in \xi_h} \int_{\partial K^- \cap \Gamma_h^-} \vec{\beta} \cdot \vec{n} u_h^{in} v_h ds, \\ l_h(v_h) &= \sum_{K \in \xi_h} \int_K f v_h dx + \sum_{e \in \Gamma_h^D} \int_e g^D \left(\frac{\sigma \epsilon}{h_e} v_h - \epsilon \nabla v_h \cdot \vec{n} \right) ds \\ &\quad - \sum_{K \in \xi_h} \int_{\partial K^- \cap \Gamma_h^-} \vec{\beta} \cdot \vec{n} g^D v_h ds + \sum_{e \in \Gamma_h^N} \int_e g^N v_h ds. \end{aligned}$$

Remark 2.5.1. *Through the integration by parts, one may also have for the convective term that*

$$\begin{aligned} a_h^C(u_h, v_h) &= \sum_{K \in \xi_h} \int_K (-\vec{\beta} u_h \cdot \nabla v_h - \nabla \cdot \vec{\beta} u_h v_h) dx \\ &\quad + \sum_{K \in \xi_h} \int_{\partial K^+ \setminus \partial \Omega} \vec{\beta} \cdot \vec{n} u_h^{in} (v_h - v_h^{out}) ds + \sum_{K \in \xi_h} \int_{\partial K^+ \cap \Gamma_h^+} \vec{\beta} \cdot \vec{n} u_h^{in} v_h ds. \end{aligned}$$

2.5.1 Coercivity of Bilinear Form

We show the coercivity of the bilinear form $a_h(u, v) = a_h^{DR}(u, v) + a_h^C(u, v)$ by showing the coercivities of the bilinear forms $a_h^{DR}(u, v)$ and $a_h^C(u, v)$ separately.

To show the coercivity of the bilinear form $a_h^{DR}(u, v)$, which corresponds to the diffusion and linear reaction terms, we define the energy norm (or diffusion norm)

$$|||v|||^2 = \sum_{K \in \xi_h} (\|\epsilon \nabla v\|_{L^2(K)}^2 + \alpha_0 \|v\|_{L^2(K)}^2) + \sum_{e \in \Gamma_h^0 \cup \Gamma_h^D} \frac{\epsilon \sigma}{h_e} \|[v]\|_{L^2(e)}^2. \quad (2.10)$$

Lemma 2.5.2. *The bilinear form $a_h^{DR}(u, v)$ corresponding to the diffusion and linear reaction terms in the SIPG bilinear form (2.9) is coercive satisfying that*

$$a_h^{DR}(v, v) \geq \frac{1}{2} |||v|||^2, \quad \forall v \in V_h, \quad (2.11)$$

where the norm $||| \cdot |||$ is defined as in (2.10).

Proof. The bilinear form $a_h^{DR}(u, v)$ satisfies

$$\begin{aligned} a_h^{DR}(v, v) &= \sum_{K \in \xi_h} \int_K (\epsilon(\nabla v)^2 + \alpha v^2) dx - 2 \sum_{e \in \Gamma_h^0 \cup \Gamma_h^D} \int_e \{\epsilon \nabla v\} \cdot [v] ds \\ &\quad + \sum_{e \in \Gamma_h^0 \cup \Gamma_h^D} \frac{\sigma \epsilon}{h_e} \int_e [v]^2 ds. \end{aligned}$$

We need to find an upper bound to the term $\sum_{e \in \Gamma_h^0 \cup \Gamma_h^D} \int_e \{\epsilon \nabla v\} \cdot [v] ds$ (a lower bound to the negative of the term). Using the Cauchy-Schwarz's inequality, we obtain

$$\begin{aligned} \sum_{e \in \Gamma_h^0 \cup \Gamma_h^D} \int_e \{\epsilon \nabla v\} \cdot [v] ds &\leq \sum_{e \in \Gamma_h^0 \cup \Gamma_h^D} \|\{\epsilon \nabla v \cdot \vec{n}_e\}\|_{L^2(e)} \| [v] \|_{L^2(e)} \\ &\leq \sum_{e \in \Gamma_h^0 \cup \Gamma_h^D} \|\{\epsilon \nabla v \cdot \vec{n}_e\}\|_{L^2(e)} \underbrace{\left(\frac{1}{|e|} \right)^{\frac{1}{2} - \frac{1}{2}}}_1 \| [v] \|_{L^2(e)}. \end{aligned}$$

For the interior edges $e = K_i \cap K_j \in \Gamma_h^0$, using the definition of the average operator and the trace inequality (2.1b), we get

$$\begin{aligned} \|\{\epsilon \nabla v \cdot \vec{n}_e\}\|_{L^2(e)} &\leq \frac{1}{2} \|\epsilon \nabla v_i \cdot \vec{n}_e\|_{L^2(e)} + \frac{1}{2} \|\epsilon \nabla v_j \cdot \vec{n}_e\|_{L^2(e)} \\ &\leq \frac{C_{Tr} \epsilon}{2} h_{K_i}^{-1/2} \|\nabla v\|_{L^2(K_i)} + \frac{C_{Tr} \epsilon}{2} h_{K_j}^{-1/2} \|\nabla v\|_{L^2(K_j)}. \end{aligned}$$

Let us denote by h the maximum element size, i.e. $h = \max(h_K)$. Obviously there holds $|e| \leq h_K \leq h$ for 2D case, which of use leads to

$$\begin{aligned} \int_e \{\epsilon \nabla v\} \cdot [v] ds &\leq \frac{C_{Tr} \epsilon}{2} |e|^{1/2} \left(h_{K_i}^{-1/2} \|\nabla v\|_{L^2(K_i)} + h_{K_j}^{-1/2} \|\nabla v\|_{L^2(K_j)} \right) \\ &\quad \times \left(\frac{1}{|e|} \right)^{1/2} \| [v] \|_{L^2(e)} \\ &\leq C_{Tr} \epsilon \left(\|\nabla v\|_{L^2(K_i)}^2 + \|\nabla v\|_{L^2(K_j)}^2 \right)^{1/2} \left(\frac{1}{|e|} \right)^{1/2} \| [v] \|_{L^2(e)}. \end{aligned}$$

For the edges in the set of boundary edges Γ_h^D , similar bound can be proceeded. Summation on $e \in \Gamma_h^0 \cup \Gamma_h^D$ and noting that the maximum number of neighbors

elements (triangles) in a conforming mesh is 3 yields

$$\sum_{e \in \Gamma_h^0 \cup \Gamma_h^D} \int_e \{\epsilon \nabla v\} \cdot [v] ds \leq \sqrt{3} C_{Tr} \epsilon \left(\sum_{K \in \xi_h} \|\nabla v\|_{L^2(K)}^2 \right)^{\frac{1}{2}} \left(\sum_{e \in \Gamma_h^0 \cup \Gamma_h^D} \frac{1}{|e|} \|[v]\|_{L^2(e)}^2 \right)^{\frac{1}{2}}.$$

For a constant $\delta > 0$, applying the Young's inequality (2.3), we obtain

$$\sum_{e \in \Gamma_h^0 \cup \Gamma_h^D} \int_e \{\epsilon \nabla v\} \cdot [v] ds \leq \frac{\delta}{2} \sum_{K \in \xi_h} \|\epsilon^{1/2} \nabla v\|_{L^2(K)}^2 + \frac{3C_{Tr}^2 \epsilon}{2\delta} \sum_{e \in \Gamma_h^0 \cup \Gamma_h^D} \frac{1}{|e|} \|[v]\|_{L^2(e)}^2.$$

Now, assuming that the convection field $\vec{\beta}$ is divergence free and using the condition (2.7) gives

$$\begin{aligned} a_h^{DR}(v, v) &\geq \sum_{K \in \xi_h} \int_K (\epsilon (\nabla v)^2 + \alpha_0 v^2) dx - 2 \sum_{e \in \Gamma_h^0 \cup \Gamma_h^D} \int_e \{\epsilon \nabla v\} \cdot [v] ds \\ &\quad + \sum_{e \in \Gamma_h^0 \cup \Gamma_h^D} \frac{\sigma \epsilon}{h_e} \int_e [v]^2 ds \\ &\geq (1 - \delta) \sum_{K \in \xi_h} \|\epsilon^{1/2} \nabla v\|_{L^2(K)}^2 + \frac{1}{2} \sum_{K \in \xi_h} \alpha_0 \|v\|_{L^2(K)}^2 \\ &\quad + \sum_{e \in \Gamma_h^0 \cup \Gamma_h^D} \frac{\epsilon}{|e|} \left(\sigma - \frac{3C_{Tr}^2}{\delta} \right) \|[v]\|_{L^2(e)}^2. \end{aligned}$$

Finally, choosing $\delta = 1/2$ and the penalty parameter σ large enough ($\sigma \geq 6C_{Tr}^2$) yields

$$a_h^{DR}(v, v) \geq \frac{1}{2} \|v\|^2.$$

□

To show the coercivity of the bilinear form $a_h^C(u, v)$, which corresponds to the convection part, first note that the bilinear form $a_h^C(u, v)$ is equivalent to

$$a_h^C(u, v) = \sum_{K \in \xi_h} \int_K \vec{\beta} \cdot \nabla u v dx + \sum_{e \in \Gamma_h^0} \int_e |\vec{\beta} \cdot \vec{n}| (u^{in} - u^{out}) v ds + \sum_{e \in \Gamma_h^-} \int_e |\vec{\beta} \cdot \vec{n}| u^{in} v ds.$$

In [55], it is shown for the above definition of the bilinear form $a_h^C(u, v)$, there holds

$$\begin{aligned} a_h^C(v, v) &= \frac{1}{2} \sum_{e \in \Gamma_h^-} |\vec{\beta} \cdot \vec{n}| \|v^{in}\|_{L^2(e)}^2 + \frac{1}{2} \sum_{e \in \Gamma_h^+} |\vec{\beta} \cdot \vec{n}| \|v^{out}\|_{L^2(e)}^2 \\ &\quad + \frac{1}{2} \sum_{e \in \Gamma_h^0} |\vec{\beta} \cdot \vec{n}| \|v^{in} - v^{out}\|_{L^2(e)}^2. \end{aligned} \quad (2.12)$$

, which yields the following lemma.

Lemma 2.5.3. *The bilinear form $a_h(u, v)$ (2.9) is coercive satisfying*

$$a_h(v, v) \geq \frac{1}{2} \|v\|_{DG}^2, \quad \forall v \in V_h$$

with the DG norm

$$\begin{aligned} \|v\|_{DG} &= \|v\| + \frac{1}{2} \sum_{e \in \Gamma_h^-} |\vec{\beta} \cdot \vec{n}| \|v^{in}\|_{L^2(e)}^2 + \frac{1}{2} \sum_{e \in \Gamma_h^+} |\vec{\beta} \cdot \vec{n}| \|v^{out}\|_{L^2(e)}^2 \\ &\quad + \frac{1}{2} \sum_{e \in \Gamma_h^0} |\vec{\beta} \cdot \vec{n}| \|v^{in} - v^{out}\|_{L^2(e)}^2. \end{aligned} \quad (2.13)$$

Proof. Using the identities (2.11) and (2.12), for all $v \in V_h$, we immediately obtain the coercivity of the bilinear form $a_h(u, v)$

$$\begin{aligned} a_h(v, v) &= a_h^{DR}(v, v) + a_h^C(v, v) \\ &\geq \frac{1}{2} \|v\|^2 + \frac{1}{2} \sum_{e \in \Gamma_h^-} |\vec{\beta} \cdot \vec{n}| \|v^{in}\|_{L^2(e)}^2 + \frac{1}{2} \sum_{e \in \Gamma_h^+} |\vec{\beta} \cdot \vec{n}| \|v^{out}\|_{L^2(e)}^2 \\ &\quad + \frac{1}{2} \sum_{e \in \Gamma_h^0} |\vec{\beta} \cdot \vec{n}| \|v^{in} - v^{out}\|_{L^2(e)}^2 \\ &\geq \frac{1}{2} \|v\|_{DG}^2, \end{aligned}$$

where the DG norm $\|\cdot\|_{DG}$ is given by (2.13). □

CHAPTER 3

ELLIPTIC PROBLEMS WITH ADAPTIVITY

This chapter aims to consider and study on the stationary semi-linear diffusion - convection-reaction equations, the stationary form of the model (1.1). We give the existence and uniqueness results of the elliptic system. Then, being the main tool of this thesis, we discuss the adaptivity to solve the such models, which is a native tool for the convection dominated problems to handle the unphysical oscillations at the interior/boundary layers due to the convection. We derive and prove the residual-based robust a posteriori error estimates, which is one of the main theoretical contribution of this thesis. Using the driven a posteriori estimates, we introduce an adaptive algorithm chart for the stationary models. The results obtained for the stationary model will be a key ingredient in the non-stationary models since one of the major aim of this thesis is to use the ready a posteriori error estimates driven for elliptic models in the non-stationary systems instead of adapting the estimates for non-stationary models case by case in order to compare the exact solution with numerical solution directly.

Since the stiffness matrices obtained by DG methods become larger, ill-conditioned and dense for higher order DG elements, preconditioning of the arising systems are needed. For this reason, we introduce as an iterative method the matrix reordering and partitioning technique in [80]. We give the detailed construction of the matrix reordering and partitioning technique, and we demonstrate its efficiency numerically.

In the literature, there are various methods to solve the convection dominated diffusion-convection-reaction equations, especially for linear ones. Among them the most famous methods are the stabilized finite elements methods such as Galerkin least squares FEMs [22, 21, 54]. The most known method among this class of methods

is the streamline upwind Petrov-Galerkin (SUPG) method to solve the convection dominated diffusion-convection-reaction equations. In this chapter, we discuss on the Galerkin least squares FEMs, and we compare the adaptive DG approximations with the ones obtained by the Galerkin least squares FEMs in Section 3.4, and the ones obtained by SUPG through the numerical studies in Section (3.5).

3.1 Model Elliptic Problem

We consider as the elliptic model problem the convection dominated scalar stationary form of the model (1.1) given by

$$\alpha u - \epsilon \Delta u + \vec{\beta} \cdot \nabla u + r(u) = f \quad \text{in } \Omega, \quad (3.1a)$$

$$u = g^D \quad \text{on } \Gamma_D, \quad (3.1b)$$

$$\epsilon \frac{\partial u}{\partial \vec{n}} = g^N \quad \text{on } \Gamma_N \quad (3.1c)$$

with Ω is a bounded, open, convex domain in \mathbb{R}^2 with boundary $\partial\Omega = \Gamma_D \cup \Gamma_N$, $\Gamma_D \cap \Gamma_N = \emptyset$, $0 < \epsilon \ll 1$ is the diffusivity constant, $f \in L^2(\Omega)$ is the source function, $\vec{\beta} \in (W^{1,\infty}(\Omega))^2$ is the velocity field, $g^D \in H^{3/2}(\Gamma_D)$ and $g^N \in H^{1/2}(\Gamma_N)$ are the Dirichlet and Neumann boundary conditions, respectively. The linear reaction coefficient α is a positive number which mimics the temporal discretization parameter $1/\Delta t$, where Δt is the time-step size. Further, we assume that the non-linear reaction term is bounded, locally Lipschitz continuous and monotone, i.e. satisfy for any $s, s_1, s_2 \geq 0$, $s, s_1, s_2 \in \mathbb{R}$ the following conditions

$$|r_i(s)| \leq C_S, \quad C_S > 0, \quad s \in [-S, S], \quad (3.2a)$$

$$\|r_i(s_1) - r_i(s_2)\|_{L^2(\Omega)} \leq L(S) \|s_1 - s_2\|_{L^2(\Omega)}, \quad L(S) > 0, \quad (3.2b)$$

$$r_i \in C^1(\mathbb{R}_0^+), \quad r_i(0) = 0, \quad r'_i(s) \geq 0. \quad (3.2c)$$

Moreover, we assume that

$$\alpha - \frac{1}{2} \nabla \cdot \vec{\beta}(x) \geq \alpha_0 \geq 0, \quad (3.3a)$$

$$\|\alpha - \nabla \cdot \vec{\beta}(x)\|_{L^\infty(\Omega)} \leq c_* \alpha_0, \quad (3.3b)$$

for some non-negative constants α_0 and c_* . The identity (3.3a) is needed to have a coercive bilinear form (well-posedness of the linear part), while, we use the identity (3.3b) to prove the reliability of a posteriori error estimate.

In order to discretize the problem (3.1), we apply SIPG formulation to the diffusion part and the original upwinding scheme [61, 69] to the convection part. Using the notations and definitions from the previous chapter, the solution of (3.1) reads as: find $u_h \in V_h$ such that

$$a_h(u_h, v_h) + b_h(u_h, v_h) = l_h(v_h), \quad \forall v_h \in V_h \quad (3.4)$$

with

$$\begin{aligned} a_h(u_h, v_h) &= \sum_{K \in \xi_h} \int_K \epsilon \nabla u_h \cdot \nabla v_h dx + \sum_{K \in \xi_h} \int_K \alpha u_h v_h dx + \sum_{K \in \xi_h} \int_K \vec{\beta} \cdot \nabla u_h v_h dx \\ &\quad - \sum_{e \in \Gamma_h^0 \cup \Gamma_h^D} \int_e \{\epsilon \nabla v_h\} \cdot [u_h] ds - \sum_{e \in \Gamma_h^0 \cup \Gamma_h^D} \int_e \{\epsilon \nabla u_h\} \cdot [v_h] ds \\ &\quad + \sum_{K \in \xi_h} \int_{\partial K^- \setminus \partial \Omega} \vec{\beta} \cdot \vec{n} (u_h^{out} - u_h^{in}) v_h ds - \sum_{K \in \xi_h} \int_{\partial K^- \cap \Gamma_h^-} \vec{\beta} \cdot \vec{n} u_h^{in} v_h ds \\ &\quad + \sum_{e \in \Gamma_h^0 \cup \Gamma_h^D} \frac{\sigma \epsilon}{h_e} \int_e [u_h] \cdot [v_h] ds, \\ b_h(u_h, v_h) &= \sum_{K \in \xi_h} \int_K r(u_h) v_h dx, \\ l_h(v_h) &= \sum_{K \in \xi_h} \int_K f v_h dx + \sum_{e \in \Gamma_h^D} \int_e g^D \left(\frac{\sigma \epsilon}{h_e} v_h - \epsilon \nabla v_h \cdot \vec{n} \right) ds \\ &\quad - \sum_{K \in \xi_h} \int_{\partial K^- \cap \Gamma_h^-} \vec{\beta} \cdot \vec{n} g^D v_h ds + \sum_{e \in \Gamma_h^N} \int_e g^N v_h ds. \end{aligned}$$

The formulation (3.4) differs from the formulation (2.8) by the additional non-linear form $b_h(u, v)$ which is linear in the second argument. Thus, it is valid that the bilinear form $a_h(u, v)$ in (3.4) is coercive on V_h with the DG norm (2.13), which will be used to show the existence of the unique solution of the variational problem (3.4).

3.1.1 Discrete System in Matrix-Vector Form

The approximate solution to the discrete problem (3.4) has the form

$$u_h = \sum_{i=1}^{Nel} \sum_{l=1}^{Nloc} U_l^i \phi_l^i, \quad (3.5)$$

where ϕ_l^i 's are the basis polynomials spanning the DGFEM space V_h , U_l^i 's are the unknown coefficients to be found, Nel denotes the number of triangles and $Nloc$

is the number of local dimension. In DG methods, we choose the piecewise basis polynomials ϕ_l^i 's in such a way that each basis function has only one triangle as a support, i.e. we choose on a specific triangle K_e , $e \in \{1, 2, \dots, Nel\}$, the basis polynomials ϕ_l^e which are zero outside the triangle K_e , $l = 1, 2, \dots, Nloc$. By this construction, the stiffness matrix obtained by DG methods has a block structure, each of which related to a triangle or face integral (there is no overlapping as in continuous FEM case). The product $dof := Nel * Nloc$ gives the degree of freedom (DoFs) in DG methods. Inserting the linear combination of u_h in (3.4) and choosing the test functions as $v_h = \phi_l^i$, $l = 1, 2, \dots, Nloc$, $i = 1, 2, \dots, Nel$, we obtain the non-linear system of equations in matrix-vector form given by

$$S\mathbf{U} + \mathbf{b}(\mathbf{U}) = \mathbf{L}, \quad (3.6)$$

where $\mathbf{U} \in \mathbb{R}^{dof}$ is the vector of unknown coefficients U_l^i 's, $S \in \mathbb{R}^{dof \times dof}$ is the stiffness matrix corresponding to the bilinear form $a_h(u_h, v_h)$, $\mathbf{b}(\mathbf{U}) \in \mathbb{R}^{dof}$ is the vector function of U related to the non-linear form $b_h(u_h, v_h)$ and $\mathbf{L} \in \mathbb{R}^{dof}$ is the vector to the linear form $l_h(v_h)$. The explicit definitions are given by

$$S = \begin{bmatrix} S_{11} & S_{12} & \cdots & S_{1,Nel} \\ S_{21} & S_{22} & & \vdots \\ \vdots & & \ddots & \\ S_{Nel,1} & \cdots & & S_{Nel,Nel} \end{bmatrix}, \quad \mathbf{U} = \begin{bmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \\ \vdots \\ \mathbf{U}_{Nel} \end{bmatrix}$$

$$\mathbf{b}(\mathbf{U}) = \begin{bmatrix} \mathbf{b}_1(\mathbf{U}) \\ \mathbf{b}_2(\mathbf{U}) \\ \vdots \\ \mathbf{b}_{Nel}(\mathbf{U}) \end{bmatrix}, \quad \mathbf{L} = \begin{bmatrix} \mathbf{L}_1 \\ \mathbf{L}_2 \\ \vdots \\ \mathbf{L}_{Nel} \end{bmatrix}$$

where all the block matrices have dimension $Nloc$:

$$S_{ji} = \begin{bmatrix} a_h(\phi_1^i, \phi_1^j) & a_h(\phi_2^i, \phi_1^j) & \cdots & a_h(\phi_{Nloc}^i, \phi_1^j) \\ a_h(\phi_1^i, \phi_2^j) & a_h(\phi_2^i, \phi_2^j) & & \vdots \\ \vdots & & \ddots & \\ a_h(\phi_1^i, \phi_{Nloc}^j) & \cdots & & a_h(\phi_{Nloc}^i, \phi_{Nloc}^j) \end{bmatrix}, \quad \mathbf{U}_i = \begin{bmatrix} U_1^i \\ U_2^i \\ \vdots \\ U_{Nloc}^i \end{bmatrix}$$

$$\mathbf{b}_i = \begin{bmatrix} b_h(u_h, \phi_1^i) \\ b_h(u_h, \phi_2^i) \\ \vdots \\ b_h(u_h, \phi_{Nloc}^i) \end{bmatrix}, \quad \mathbf{L}_i = \begin{bmatrix} l_h(\phi_1^i) \\ l_h(\phi_2^i) \\ \vdots \\ l_h(\phi_{Nloc}^i) \end{bmatrix}$$

Proposition 3.1.1. *The non-linear vector $\mathbf{b}(\mathbf{U})$ in (3.6) is locally Lipschitz with respect to \mathbf{U} .*

Proof. For given functions u^1, u^2, v with u^1, u^2 , satisfying (3.5), we have by definition

$$b_h(u^1 - u^2, v) = \int_{\Omega} r(u^1 - u^2)v dx.$$

Applying Cauchy-Schwarz's inequality and using the locally Lipschitz continuity condition (3.2b), we get

$$\begin{aligned} b_h(u^1 - u^2, v) &\leq \|r(u^1 - u^2)\|_{L^2(\Omega)} \|v\|_{L^2(\Omega)} \\ &\leq L_S \|u^1 - u^2\|_{L^2(\Omega)} \|v\|_{L^2(\Omega)}, \end{aligned}$$

which means that the non-linear form $b_h(u, v)$ is locally Lipschitz continuous in the first argument. Since the components of the vector $\mathbf{b}(\mathbf{U}^1 - \mathbf{U}^2)$ are nothing but the non-linear forms $b_h(u^1 - u^2, \phi_l^i)$, $l = 1, 2, \dots, Nloc$, $i = 1, 2, \dots, Nel$, each component of the vector $\mathbf{b}(\mathbf{U})$ is locally Lipschitz continuous, which yields that the vector $\mathbf{b}(\mathbf{U})$ is locally Lipschitz with respect to \mathbf{U} . \square

Proposition 3.1.2 (Existence of Unique Solution). *The SIPG formulation (3.4) has a unique solution.*

Proof. The coercivity (Lemma 2.5.3) of the bilinear form $a_h(u, v)$, in the algebraic point of view, means that the matrix S in the system (3.6) arising from the bilinear form $a_h(u, v)$ is positive definite. This, combining with the locally Lipschitz condition (Lemma 3.1.1) of the non-linear vector $\mathbf{b}(\mathbf{U})$ in (3.6), means that the system (3.6), as a result, the SIPG formulation (3.4) has a unique solution. \square

3.2 Adaptivity

Most of the convection dominated problems lead to internal/boundary layers where the solution has large gradients. The standard FEMs are known to produce strong oscillations around the layers, and one has to find accurate approximations in order to handle the nonphysical oscillations using some certain techniques. A naive approach is to refine the mesh uniformly. But it is not desirable as it highly increase the degrees of freedom and refines the mesh unnecessarily in regions where the solutions are smooth. On the other hands, in the semi-linear diffusion-convection-reaction equations, in addition to the nonphysical oscillations due to convection, non-linear reaction leads to sharp fronts. The adaptive tools can overcome all the so-called nonphysical oscillations and shocks, in which the mesh is refined only locally using an adaptive strategy. In this section, we describe the adaptive strategy for elliptic semi-linear diffusion-convection-reaction equations.

In the adaptive finite elements, the elements in a triangulation are selected to be refined locally if their estimated local errors are large. Thus, the crucial part of the adaptive algorithms is to estimate the local errors. The major tool to estimate the local errors is a posteriori error estimation which uses the approximate solution and the given problem data. There are plenty many studies on a posteriori error estimation most of which mainly with respect to the energy norm induced by the weak formulation [3, 10, 86, 85, 84]. On the other hands, through the local properties being very suitable for adaptive schemes, there are several studies on a posteriori error estimation using DG discretization. The convergence analysis of a residual-based a posteriori error estimation using DG was firstly studied by Karakashian and Pascal [57]. Hoppe et al. [49] analyzed the convergence of a posteriori error estimates for the interior penalty discontinuous Galerkin method. In [2], a posteriori error estimator robust in any unknown constant has been derived by Ainsworth. Further, a posteriori error estimation using DG discretization are also studied by Rivière et al. [71], Houston et al. [52] and Ern et al. [41], and references therein.

In this section, we introduce the adaptive strategy for elliptic semi-linear diffusion-convection-reaction models using the residual-based in diffusion parameter (Péclet number) robust a posteriori error estimation used for elliptic linear diffusion-convection-

reaction models in [75]. We prove a posteriori bounds with respect to the energy norm induced by the SIPG formulation of the system.

3.2.1 The Adaptive Procedure

Our adaptive algorithm is based on the standard adaptive finite element (AFEM) iterative loop (Fig.3.1):

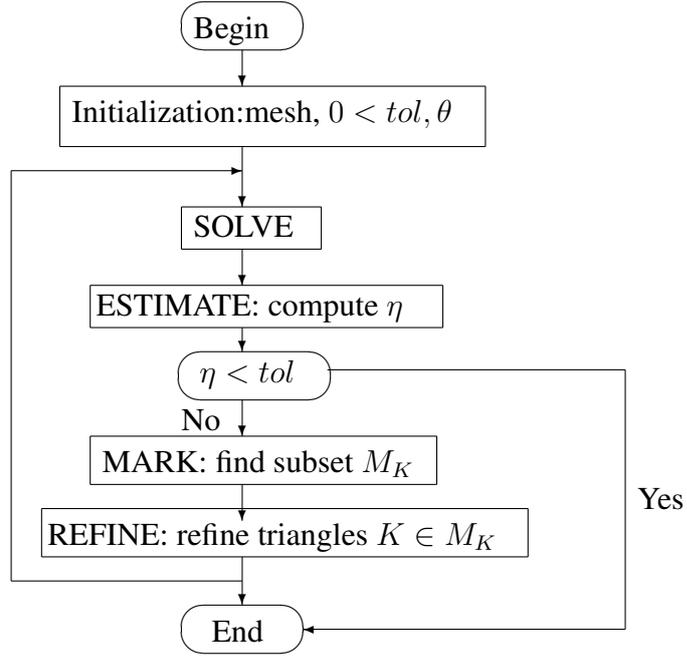


Figure 3.1: Adaptive strategy

The first step, SOLVE, is to solve the SIPG discretized system (3.4) on a given triangulation ξ_h . The ESTIMATE step is the key part of the adaptive procedure, by which we are able to determine the elements having large error to be refined using computed solution and given data (a posteriori). As an estimator, we use a residual based error indicator based on the modification of the error estimator given in Schötzau and Zhu [75] for a single linear convection dominated diffusion-convection-reaction equation to the diffusion-convection equation with non-linear reaction mechanism, which is robust in the Péclet number. To do this, we include in a posteriori error indicator the non-linear reaction term as local contributions to the cell residuals and not to the interior/boundary edge residuals [87, Chp. 5.1.4]. Let u_h be the solution to (3.4). Then,

for each element $K \in \xi_h$, we define the local error indicators η_K^2 as

$$\eta_K^2 = \eta_{R_K}^2 + \eta_{E_K^0}^2 + \eta_{E_K^D}^2 + \eta_{E_K^N}^2, \quad (3.7)$$

In (3.7), η_{R_K} denote the cell residuals

$$\eta_{R_K}^2 = \rho_K^2 \|f - \alpha u_h + \epsilon \Delta u_h - \vec{\beta} \cdot \nabla u_h - r(u_h)\|_{L^2(K)}^2,$$

while, $\eta_{E_K^0}$, $\eta_{E_K^D}$ and $\eta_{E_K^N}$ stand for the edge residuals coming from the jump of the numerical solution on the interior, Dirichlet boundary and Neumann boundary edges, respectively

$$\begin{aligned} \eta_{E_K^0}^2 &= \sum_{e \in \partial K \cap \Gamma_h^0} \left(\frac{1}{2} \epsilon^{-\frac{1}{2}} \rho_e \|[\epsilon \nabla u_h]\|_{L^2(e)}^2 + \frac{1}{2} \left(\frac{\epsilon \sigma}{h_e} + \alpha_0 h_e + \frac{h_e}{\epsilon} \right) \|[u_h]\|_{L^2(e)}^2 \right), \\ \eta_{E_K^D}^2 &= \sum_{e \in \partial K \cap \Gamma_h^D} \left(\frac{\epsilon \sigma}{h_e} + \alpha_0 h_e + \frac{h_e}{\epsilon} \right) \|g^D - u_h\|_{L^2(e)}^2, \\ \eta_{E_K^N}^2 &= \sum_{e \in \partial K \cap \Gamma_h^N} \epsilon^{-\frac{1}{2}} \rho_e \|g^N - \epsilon \nabla u_h \cdot \vec{n}\|_{L^2(e)}^2, \end{aligned}$$

where the weights ρ_K and ρ_e , on an element K and along an edge e , respectively, are defined by

$$\rho_K = \min\{h_K \epsilon^{-\frac{1}{2}}, \alpha_0^{-\frac{1}{2}}\}, \quad \rho_e = \min\{h_e \epsilon^{-\frac{1}{2}}, \alpha_0^{-\frac{1}{2}}\},$$

for $\alpha_0 \neq 0$. When $\alpha_0 = 0$, we take $\rho_K = h_K \epsilon^{-\frac{1}{2}}$ and $\rho_e = h_e \epsilon^{-\frac{1}{2}}$. Then, our a posteriori error indicator is given by

$$\eta = \left(\sum_{K \in \xi_h} \eta_K^2 \right)^{1/2}. \quad (3.8)$$

We also introduce the data approximation error,

$$\Theta^2 = \Theta^2(f) + \Theta^2(u^D) + \Theta^2(u^N) \quad (3.9)$$

with

$$\begin{aligned} \Theta^2(f) &= \sum_{K \in \xi} \rho_K^2 (\|f - f_h\|_{L^2(K)}^2 + \|(\vec{\beta} - \vec{\beta}_h) \cdot \nabla u_h\|_{L^2(K)}^2 + \|(\alpha - \alpha_h) u_h\|_{L^2(K)}^2), \\ \Theta^2(u^D) &= \sum_{e \in \Gamma_h^D} \left(\frac{\epsilon \sigma}{h_e} + \alpha_0 h_e + \frac{h_e}{\epsilon} \right) \|g^D - \hat{g}^D\|_{L^2(e)}^2, \\ \Theta^2(u^N) &= \sum_{e \in \Gamma_h^N} \epsilon^{-\frac{1}{2}} \rho_e \|g^N - \hat{g}^N\|_{L^2(e)}^2 \end{aligned}$$

with \hat{g}^D and \hat{g}^N denoting the mean integrals of g^D and g^N , respectively.

In the MARK step, if the given tolerance is not satisfied, we determine the set of elements $M_K \subset \xi_h$ to be refined using the error indicator defined in (3.7). To do this, we use the bulk criterion proposed by Döfler [37], by which the approximation error is decreased by a fixed factor for each loop. In the light of bulk criterion, we choose the set of elements $M_K \subset \xi_h$ satisfying

$$\sum_{K \in M_K} \eta_K^2 \geq \theta \sum_{K \in \xi_h} \eta_K^2$$

for a user defined parameter $0 < \theta < 1$. Here, bigger θ results in more refinement of triangles in a single loop, where, smaller θ causes more refinement loops.

Finally, REFINES step, we refine the marked elements $K \in M_K$ using the newest vertex bisection method [27]. This process can be summarized as (see Fig.3.2): for each element $K \in \xi_h$, we label one vertex of K as a newest vertex. The opposite edge of the newest vertex is called as the refinement edge. Then, a triangle is bisected to two new children triangles by connecting the newest vertex to the midpoint of the refinement edge, and this new vertex created at the midpoint of the refinement edge is assigned to be the newest vertex of the children. Following a similar rule, these two children triangles are bisected to obtain four children elements belonging to the father element (the refined triangle $K \in M_K$). After bisecting all $K \in M_K$, we also divide some elements $K \in \xi_h \setminus M_K$ to keep the conformity of the mesh, i.e. hanging nodes are not allowed.

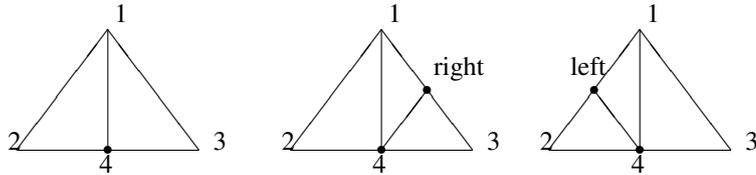


Figure 3.2: Bisection of a triangle

In the case of coupled problems, instead of a single component problem, we refine the elements being the union of the set of the elements to be refined for each component, i.e., let η_K^1 and η_K^2 be the computed local error indicators corresponding to

each unknown component of a two component system. Next, we determine the set of elements M_K^1 and M_K^2 satisfying

$$\sum_{K \in M_K^1} (\eta_K^1)^2 \geq \theta \sum_{K \in \xi_h} (\eta_K^1)^2, \quad \sum_{K \in M_K^2} (\eta_K^2)^2 \geq \theta \sum_{K \in \xi_h} (\eta_K^2)^2.$$

Then, we refine the marked elements $K \in M_K^1 \cup M_K^2$ using the newest vertex bisection method. The adaptive procedure ends after a sequence of mesh refinements up to attain a solution with an estimated error within a prescribed tolerance.

3.2.2 A Posteriori Error Estimation

In order to measure the error, we use the energy norm

$$|||v|||^2 = \sum_{K \in \xi_h} (\|\epsilon \nabla v\|_{L^2(K)}^2 + \alpha_0 \|v\|_{L^2(K)}^2) + \sum_{e \in \Gamma_h^0 \cup \Gamma_h^D} \frac{\epsilon \sigma}{h_e} \|[v]\|_{L^2(e)}^2, \quad (3.10)$$

and the semi-norm

$$|v|_C^2 = |\vec{\beta}v|_*^2 + \sum_{e \in \Gamma_h^0 \cup \Gamma_h^D} (\alpha_0 h_e + \frac{h_e}{\epsilon}) \|[v]\|_{L^2(e)}^2, \quad (3.11)$$

where

$$|u|_* = \sup_{w \in H_0^1(\Omega) \setminus \{0\}} \frac{\int_{\Omega} u \cdot \nabla w dx}{|||w|||}.$$

The terms $|\vec{\beta}v|_*^2$ and $\frac{h_e}{\epsilon} \|[v]\|_{L^2(e)}^2$ in (3.11) are used to bound the convection part, whereas, the term $\alpha_0 h_e \|[v]\|_{L^2(e)}^2$ is used to bound the linear reaction part of the discrete system.

Theorem 3.2.1. *Let u and u_h be the solutions of the continuous problem (3.1) and the discrete SIPG problem (3.4), respectively. Also assume that there hold the assumptions (3.2) and (3.3). Then, we have a posteriori error estimates*

$$|||u - u_h||| + |u - u_h|_C \lesssim \eta + \Theta \quad (\text{reliability}), \quad (3.12)$$

$$\eta \lesssim |||u - u_h||| + |u - u_h|_C + \Theta \quad (\text{efficiency}), \quad (3.13)$$

where the error indicator η and the data oscillation error Θ are defined in (3.7) and (3.9), respectively, and with the energy norm (3.10) and the semi-norm (3.11).

Before proceeding the proof of the Theorem 3.2.1, let us first give some key tools to be used in the proof. The proof of the Theorem 3.2.1 is analogous to the ones in [75] for the linear problems. Therefore, only the proofs in which the non-linear reaction term plays a role will be given. In the following, the symbols \lesssim and \gtrsim are stand for the bounds up to positive constants independent of the local mesh size h , the diffusion coefficient ϵ and the penalty parameter σ . Further, we use the DG norm defined by

$$\|v\|_{DG}^2 = |||v||| + |v|_C \quad (3.14)$$

with the definitions in (3.10) and (3.11).

First, note that for the solutions u and u_h of the continuous problem (3.1) and the discrete SIPG problem (3.4), respectively, the error $\|u - u_h\|_{DG}$ is not well-defined, since $u \in H_0^1(\Omega)$ and $u_h \in V_h \not\subset H_0^1(\Omega)$. Therefore, we split SIPG solution u_h as

$$u_h = u_h^c + u_h^r$$

with $u_h^c \in H_0^1(\Omega) \cap V_h$ being the conforming part of the solution and $u_h^r \in V_h$ is the remainder term. In this way, we have $u_h \in H_0^1(\Omega) + V_h$, and from the triangular inequality, we obtain

$$\|u - u_h\|_{DG} \leq \|u - u_h^c\|_{DG} + \|u_h^r\|_{DG}.$$

Now, both the terms on the right hand side are well-defined norms, and our aim is to find bounds for them. Next, using the Remark 2.5.1 for the terms related to the convection, we introduce the following auxiliary forms:

$$D_h(u, v) = \sum_{K \in \xi_h} \int_K \left(\epsilon \nabla u \cdot \nabla v + (\alpha - \nabla \cdot \vec{\beta}) uv \right) dx, \quad (3.15a)$$

$$\begin{aligned} O_h(u, v) = & - \sum_{K \in \xi_h} \int_K \vec{\beta} u \cdot \nabla v dx + \sum_{K \in \xi_h} \int_{\partial K^+ \cap \Gamma^+} \vec{\beta} \cdot \vec{n}_K uv ds \\ & + \sum_{K \in \xi_h} \int_{\partial K^+ \setminus \partial \Omega} \vec{\beta} \cdot \vec{n}_K u (v - v^{out}) ds, \end{aligned} \quad (3.15b)$$

$$K_h(u, v) = - \sum_{e \in \Gamma_h^0 \cup \Gamma_h^D} \int_e \{ \epsilon \nabla u \} \cdot [v] ds - \sum_{e \in \Gamma_h^0 \cup \Gamma_h^D} \int_e \{ \epsilon \nabla v \} \cdot [u] ds, \quad (3.15c)$$

$$J_h(u, v) = \sum_{e \in \Gamma_h^0 \cup \Gamma_h^D} \frac{\sigma \epsilon}{h_e} \int_e [u] \cdot [v] ds. \quad (3.15d)$$

Define the bilinear form $\tilde{a}_h(u, v)$ by

$$\tilde{a}_h(u, v) = D_h(u, v) + O_h(u, v) + J_h(u, v),$$

which is well-defined on $H_0^1(\Omega) + V_h$ and satisfies the coercivity property [75, Lemma 4.1]

$$\tilde{a}_h(v, v) \geq |||v|||^2, \quad \forall v \in H_0^1(\Omega).$$

Moreover, the SIPG bilinear form $a_h(u, v)$ in (3.4) satisfies

$$a_h(u, v) = \tilde{a}_h(u, v) + K_h(u, v), \quad \forall u, v \in V_h, \quad (3.16)$$

$$a_h(u, v) = \tilde{a}_h(u, v), \quad \forall u, v \in H_0^1(\Omega). \quad (3.17)$$

Further, the auxiliary forms are continuous [75, Lemma 4.2]:

$$|D_h(u, v)| \lesssim |||u||| |||v|||, \quad u, v \in H_0^1(\Omega) + V_h, \quad (3.18)$$

$$|O_h(u, v)| \lesssim |\vec{\beta}u|_* |||v|||, \quad u \in H_0^1(\Omega) + V_h, v \in H_0^1(\Omega), \quad (3.19)$$

$$|J_h(u, v)| \lesssim |||u||| |||v|||, \quad u, v \in H_0^1(\Omega) + V_h, \quad (3.20)$$

and for $u \in V_h, v \in V_h \cap H_0^1(\Omega)$, we have [75, Lemma 4.3]

$$|K_h(u, v)| \lesssim \sigma^{-1/2} \left(\sum_{e \in \Gamma_h^0 \cup \Gamma_h^D} \frac{\sigma_e}{h_e} \|[u]\|_{L^2(e)} \right)^{1/2} |||v|||. \quad (3.21)$$

We also have for the non-linear form $b_h(u, v)$, using the assumption (3.2a)

$$|b_h(u, v)| \lesssim |||v|||, \quad u, v \in H_0^1(\Omega) + V_h. \quad (3.22)$$

Now, we give some auxiliary results and conditions which are used in the proofs.

Lemma 3.2.2. [75, Lemma 4.4](Inf-sup condition) For all $u \in H_0^1(\Omega)$, we have

$$|||u||| + |\vec{\beta}u|_* \lesssim \sup_{v \in H_0^1(\Omega) \setminus \{0\}} \frac{\tilde{a}_h(t; u, v)}{|||v|||}. \quad (3.23)$$

Definition 3.2.3. (Approximation operator) Let $V_h^c = V_h \cap H_0^1(\Omega)$ be the conforming subspace of V_h . For any $u \in V_h$, there exists an approximation operator $A_h : V_h \mapsto V_h^c$ satisfying

$$\sum_{K \in \xi} \|u - A_h u\|_{L^2(K)}^2 \lesssim \sum_{e \in \Gamma_h^0 \cup \Gamma_h^D} \int_e h_e |[u]|^2 ds, \quad (3.24)$$

$$\sum_{K \in \xi} \|\nabla(u - A_h u)\|_{L^2(K)}^2 \lesssim \sum_{e \in \Gamma_h^0 \cup \Gamma_h^D} \int_e \frac{1}{h_e} |[u]|^2 ds. \quad (3.25)$$

Definition 3.2.4. (Interpolation operator) For any $u \in H_0^1(\Omega)$, there exists an interpolation operator

$$I_h : H_0^1(\Omega) \mapsto \{w \in C(\bar{\Omega}) : w|_K \in \mathbb{P}_1(K), \forall K \in \xi, w = 0 \text{ on } \Gamma\},$$

which satisfies

$$|||I_h u||| \lesssim |||u|||, \quad (3.26)$$

$$\left(\sum_{K \in \xi} \rho_K^{-2} \|u - I_h u\|_{L^2(K)}^2 \right)^{1/2} \lesssim |||u|||, \quad (3.27)$$

$$\left(\sum_{e \in \Gamma_0 \cup \Gamma^D} \epsilon^{1/2} \rho_e^{-1} \|u - I_h u\|_{L^2(K)}^2 \right)^{1/2} \lesssim |||u|||. \quad (3.28)$$

Now, consider the splitting of the discrete solution $u_h = u_h^c + u_h^r$ as $u_h^c = A_h u_h \in H_0^1(\Omega) \cap V_h$ with A_h is the approximation operator in Definition 3.2.3 and $u_h^r = u_h - u_h^c \in V_h$.

Lemma 3.2.5. [75, Lemma 4.7](Bound for remainder term) For the remainder term, we have the bound

$$\|u_h^r\|_{DG} \lesssim \eta, \quad (3.29)$$

where η is our a posteriori error indicator (3.8).

Lemma 3.2.6. For any $v \in H_0^1(\Omega)$, we have

$$\int_{\Omega} f(v - I_h v) dx - \tilde{a}_h(u_h, v - I_h v) - b_h(u_h, v - I_h v) \lesssim (\eta + \Theta) |||v|||, \quad (3.30)$$

where I_h is the interpolation operator in Definition 3.2.4.

Proof. Let

$$T = \int_{\Omega} f(v - I_h v) dx - \tilde{a}_h(u_h, v - I_h v) - b_h(u_h, v - I_h v).$$

Applying integration by parts, we get

$$\begin{aligned} T &= \sum_{K \in \xi_h} \int_K (f - \alpha u_h + \epsilon \Delta u_h - \vec{\beta} \cdot \nabla u_h - r(u_h))(v - I_h v) dx \\ &\quad - \sum_{K \in \xi_h} \int_{\partial K} \epsilon \nabla u_h \cdot \vec{n}_K (v - I_h v) ds \\ &\quad + \sum_{K \in \xi_h} \int_{\partial K^- \setminus \partial \Omega} \vec{\beta} \cdot \vec{n}_K (u_h - u_h^{out})(v - I_h v) ds \\ &= T_1 + T_2 + T_3. \end{aligned}$$

Adding and subtracting the data approximation terms into the term T_1 , we obtain

$$\begin{aligned} T_1 &= \sum_{K \in \xi_h} \int_K (f_h - \alpha u_h + \epsilon \Delta u_h - \vec{\beta}_h \cdot \nabla u_h - r(u_h))(v - I_h v) dx \\ &\quad + \sum_{K \in \xi_h} \int_K ((f - f_h) - (\alpha - \alpha_h) u_h - (\vec{\beta} - \vec{\beta}_h) \cdot \nabla u_h)(v - I_h v) dx. \end{aligned}$$

The use of the Cauchy-Schwarz inequality and interpolation operator identity (3.27) gives

$$\begin{aligned} T_1 &\lesssim \left(\sum_{K \in \xi_h} \eta_{R_K}^2 \right)^{1/2} \left(\sum_{K \in \xi_h} \rho_K^{-2} \|v - I_h v\|_{L^2(K)}^2 \right)^{1/2} \\ &\quad + \left(\sum_{K \in \xi_h} \Theta_K^2 \right)^{1/2} \left(\sum_{K \in \xi_h} \rho_K^{-2} \|v - I_h v\|_{L^2(K)}^2 \right)^{1/2} \\ &\lesssim \left(\sum_{K \in \xi_h} (\eta_{R_K}^2 + \Theta_K^2) \right)^{1/2} |||v|||. \end{aligned}$$

Moreover, for the terms T_2 and T_3 , we have [75, Lemma 4.8]

$$\begin{aligned} T_2 &\lesssim \left(\sum_{K \in \xi_h} \eta_{E_K}^2 \right)^{1/2} |||v|||, \\ T_3 &\lesssim \left(\sum_{K \in \xi_h} \eta_{J_K}^2 \right)^{1/2} |||v|||, \end{aligned}$$

which finishes the proof. \square

Lemma 3.2.7. *(Bound to the conforming part of the error) The conforming part of the error satisfies*

$$\|u - u_h^c\|_{DG} \lesssim \eta + \Theta. \quad (3.31)$$

Proof. Since $u - u_h^c \in H_0^1(\Omega)$, we have $|u - u_h^c|_C = |\vec{\beta}(u - u_h^c)|_*$. Then, from the inf-sup condition (3.23), we get

$$\|u - u_h^c\|_{DG} = |||u - u_h^c||| + |u - u_h^c|_C \lesssim \sup_{v \in H_0^1(\Omega) \setminus \{0\}} \frac{\tilde{a}_h(u - u_h^c, v)}{|||v|||}.$$

So, we need to bound the term $\tilde{a}_h(u - u_h^c, v)$. Using the fact that $u - u_h^c \in H_0^1(\Omega)$,

we have

$$\begin{aligned}
\tilde{a}_h(u - u_h^c, v) &= \tilde{a}_h(u, v) - \tilde{a}_h(u_h^c, v) \\
&= \int_{\Omega} f v dx - b_h(u, v) - \tilde{a}_h(u_h^c, v) \\
&= \int_{\Omega} f v dx - b_h(u, v) - D_h(u_h^c, v) - J_h(u_h^c, v) - O_h(u_h^c, v) \\
&= \int_{\Omega} f v dx - b_h(u_h, v) + b_h(u_h, v) - b_h(u, v) \\
&\quad - \tilde{a}_h(u_h, v) + D_h(u_h^r, v) + J_h(u_h^r, v) + O_h(u_h^r, v).
\end{aligned}$$

We also have from the SIPG scheme (3.4) that

$$\int_{\Omega} f I_h v dx = \tilde{a}_h(u_h, I_h v) + K_h(u_h, I_h v) + b_h(u_h, I_h v),$$

where I_h is the interpolation operator in Definition 3.2.4. Hence, we obtain that

$$\tilde{a}(u - u_h^c, v) = T_1 + T_2 + T_3 + T_4$$

with the terms

$$\begin{aligned}
T_1 &= \int_{\Omega} f(v - I_h v) dx - \tilde{a}_h(u_h, v - I_h v) - b_h(u_h, v - I_h v) \\
T_2 &= D_h(u_h^r, v) + J_h(u_h^r, v) + O_h(u_h^r, v) \\
T_3 &= K_h(u_h^s, I_h v) \\
T_4 &= b_h(u_h, v) - b_h(u, v).
\end{aligned}$$

From the identity (3.30), we have

$$T_1 \lesssim (\eta + \Theta) |||v|||.$$

The continuity results (3.18)-(3.20) and the bound to remainder term (3.29) yields

$$T_2 \lesssim (|||u_h^r||| + |\vec{\beta} u_h^r|_*) |||v||| \leq \eta |||v|||.$$

Moreover, using the identities (3.21) and (3.26), we get

$$T_3 \lesssim \sigma^{-1/2} \left(\sum_{K \in \xi} \eta_{J_K}^2 \right)^{1/2} |||I_h v||| \lesssim \sigma^{-1/2} \left(\sum_{K \in \xi} \eta_{J_K}^2 \right)^{1/2} |||v|||.$$

Finally, using the Cauchy-Schwarz's inequality and the identity 3.22, we obtain

$$\begin{aligned}
T_4 &= b_h(u_h, v) - b_h(u, v) = \int_{\Omega} r(u_h) v dx - \int_{\Omega} r(u) v dx \\
&\leq C_1 \|v\|_{L^2(\Omega)} - C_2 \|v\|_{L^2(\Omega)} \\
&\lesssim |||v|||,
\end{aligned}$$

which finishes the proof. \square

Now, we can give the proof of Theorem 3.2.1.

Proof. (Theorem 3.2.1) In case of the reliability condition (3.12), combining the bounds (3.29) and (3.31) to the remainder and the conforming parts of the error, respectively, we obtain

$$\begin{aligned} \|u - u_h\|_{DG} &\leq \|u - u_h^c\|_{DG} + \|u_h^r\|_{DG} \\ &\leq \eta + \Theta + \eta \\ &\lesssim \eta + \Theta. \end{aligned}$$

The proof to the efficiency condition (3.13) is similar to [75, Theorem 3.3]. We only use the boundedness property (3.2a) of the non-linear reaction term to bound the terms occurring in the procedure in [75].

□

3.3 Solution of Linearized Systems

Recall from Section 3.1.1 that the approximate solution u_h is of the form

$$u_h = \sum_{i=1}^{Nel} \sum_{l=1}^{Nloc} U_l^i \phi_l^i,$$

where ϕ_l^i 's are the basis polynomials spanning the DGFEM space V_h , U_l^i 's are the unknown coefficients to be found, Nel denotes the number of triangles and $Nloc$ is the number of local dimension depending on the degree of polynomials k , and the system dimension is $dof := Nel * Nloc$ giving the DoFs in DG methods. Inserting the linear combination of u_h in (3.4) and choosing the test functions as $v_h = \phi_l^i$, $l = 1, 2, \dots, Nloc$, $i = 1, 2, \dots, Nel$, the discrete residual of the system (3.4) in matrix vector form is given by

$$R(\mathbf{U}) = S\mathbf{U} + \mathbf{b}(\mathbf{U}) - \mathbf{L} = 0, \quad (3.32)$$

where $\mathbf{U} \in \mathbb{R}^{dof}$ is the vector of unknown coefficients U_l^i 's, and the stiffness matrix $S \in \mathbb{R}^{dof \times dof}$, the vector function $h \in \mathbb{R}^{dof}$ and the vector $\mathbf{L} \in \mathbb{R}^{dof}$ are defined as in Section 3.1.1. Being the usual approach, we solve the non-linear system of equations

(3.32) by Newton-Raphson method. So, we start with an initial guess $\mathbf{U}^{(0)}$ to the solution \mathbf{U} . Then, the Newton-Raphson process to solve the system (3.32) reads as: given initial guess $\mathbf{U}^{(0)}$, for $i = 0, 1, 2, \dots$, solve the system

$$\begin{aligned} Jw^{(i)} &= -R^{(i)}, \\ \mathbf{U}^{(i+1)} &= \mathbf{U}^{(i)} + w^{(i)} \end{aligned} \tag{3.33}$$

until a prescribed tolerance is satisfied. In (3.33), the sparse matrix $J = S + \mathbf{b}'(\mathbf{U}^{(0)})$ denotes the value of the Jacobian matrix of the residual function $R(\mathbf{U})$ at the initial iterate $\mathbf{U}^{(0)}$ and remains unchanged among the iteration steps, $w^{(i)} = \mathbf{U}^{(i+1)} - \mathbf{U}^{(i)}$ is the Newton correction, and $R^{(i)} = R(\mathbf{U}^{(i)})$ denotes the value of the residual function $R(\mathbf{U})$ at the current iterate \mathbf{U}^i .

The solution of the linear system (3.33), in case of a huge system, may be challenge since stiffness matrices obtained by DG schemes become ill-conditioned and dense for higher order DG elements [7]. Thus, it would be better to solve a preconditioned linear system of (3.33). There are several preconditioners developed for the efficient and accurate solution of the linear diffusion-convection equations [4, 46]. In this section, we introduce the matrix reordering and partitioning technique in [80] as a preconditioner.

3.3.1 Matrix Reordering & Block LU Factorization

To solve the linear systems (3.33), we apply the matrix reordering and partitioning technique in [80], which uses the largest eigenvalue and corresponding eigenvector of the Laplacian matrix. This reordering allows us to obtain a partitioning and a preconditioner based on this partitioning. Since our matrices arising from the SIPG formulation are non-symmetric (due to the convective terms), as the first step, we compute the symmetric structure by adding its transpose to itself. A symmetric, square and sparse matrix could be represented as a graph where same index rows and columns are mapped into vertices and nonzeros of the sparse matrix are mapped into the edges of the graph. Since the matrix is symmetric, the corresponding graph is undirected.

The Laplacian matrix (\mathcal{L}) is, then, defined as follows

$$\mathcal{L}(i, j) = \begin{cases} \text{deg}(v_i) & \text{if } i = j, \\ -1 & \text{if } i \neq j \end{cases}$$

in which the $\text{deg}(v_i)$ is the degree of the vertex i . The reordering we use is based on the unweighted Laplacian matrix given above. If the graph contains only one connected component, the eigenvalues of \mathcal{L} are $0 = \lambda_1 < \lambda_2 \leq \lambda_3 \leq \dots \leq \lambda_n$, otherwise there are as many zero eigenvalues as the number of connected components.

Certain eigenvalues and corresponding eigenvectors of the Laplacian matrix have been studied extensively. Most notably the second nontrivial eigenvalue of the Laplacian and the corresponding eigenvector known as the algebraic connectivity and the Fiedler vector of the graph [43]. Nodal domain theorem in [44] shows that the eigenvectors corresponding to the eigenvalues other than the first and the second smallest eigenvalue give us the connected components of the graph. In [11], the Fiedler vector for permuting the matrices to reduce the bandwidth is proposed. Reordering to obtain effective and scalable parallel banded preconditioners is proposed in [64]. We use a sparse matrix reordering for partitioning and solving linear systems using the largest eigenvalue and the corresponding eigenvector of the Laplacian matrix. Using this reordering, we show that one can reveal underlying structure of a sparse matrix. A simple Matlab code to find the reordered matrix and the permutation matrix can be found in Appendix B.

To form the preconditioned system of the linear system (3.33), we first construct a permutation matrix P using the matrix reordering technique described above, applied to the sparse Jacobian matrix J . Then, we apply the permutation matrix P to obtain the permuted system $Nw = d$ where $N = PJP^T$, $w = Pw^{(i)}$ and $d = -PR^{(i)}$. After solving the permuted system, the solution of the unpermuted linear system (3.33) can be obtained by applying the inverse permutation, $w^{(i)} = P^T w$.

Given a sparse linear system of equations $Nw = d$, after reordering, one way to solve this system is via block LU factorization. Suppose, the permuted matrix N , the right

hand side d and the solution w is partitioned as follows:

$$\begin{pmatrix} A & B \\ C^T & D \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}.$$

A block LU factorization of the coefficient matrix can be given as

$$\begin{pmatrix} A & B \\ C^T & D \end{pmatrix} = \begin{pmatrix} A & 0 \\ C^T & S \end{pmatrix} \begin{pmatrix} I & U \\ 0 & I \end{pmatrix},$$

where $U = A^{-1}B$ and $S = D - C^T A^{-1}B$, also known as the Schur complement matrix. If the cost can be amortized, one can form U and S once and use them for solving linear systems with the same coefficient matrix and different right hand sides. After this factorization, there are various approaches that one can take to solve the system. One way is to solve the system via block backward and forward substitution, by first solving the linear system $At = d_1$, and then solving the Schur complement system $Sw_2 = d_2 - C^T t$ and obtaining $w_1 = t - Uw_2$. This method is summarized in Algorithm 1.

Algorithm 1 Algorithm for solving the linear system after reordering

Input: The coefficient matrix: $\begin{pmatrix} A & B \\ C^T & D \end{pmatrix}$ and the right hand side: $\begin{pmatrix} d_1 \\ d_2 \end{pmatrix}$

Output: The solution vector: $\begin{pmatrix} w_1 \\ w_2 \end{pmatrix}$

- 1: solve $At = d_1$
 - 2: solve $Sw_2 = d_2 - C^T t$
 - 3: compute $w_1 = t - Uw_2$
-

We note that this approach involves solving two linear systems of equations with the coefficient matrix A and S . These linear systems can be solved directly or iteratively which requires effective preconditioners. Other approaches could involve solving the system $Nw = d$ iteratively where the preconditioner could take many forms. There are many other techniques for solving block partitioned and saddle point linear systems, we refer to [19] for a more detailed survey of some of these methods.

3.4 Comparison with Galerkin Least Squares FEM (GLSFEM)

Among the numerical methods to solve the convection dominated diffusion-convection equations, since one of the most popular one is the class of Galerkin Least Squares FEM (GLSFEM) such as SUPG, we discuss on this section the GLSFEMs. We compare the numerical solutions obtained by DGFEM and adaptive DGFEM (DGAFEM) formulations with the numerical solutions obtained by GLSFEMs, and we demonstrate by a numerical example that the SIPG approximation is rather accurate than the GLSFEM approximation.

For linear PDEs, the weak form in the standard Galerkin method is obtained by multiplying the differential equation with a test function v and integrating over a suitable function space V

$$(\mathcal{L}u, v) = (f, v), \quad \forall v \in V = H_0^1(\Omega),$$

where $\mathcal{L} = -\epsilon\Delta + \vec{\beta} \cdot \nabla + \alpha$ is the linear part of the stationary diffusion-convection-reaction equation (3.1). Defining the residual as $R(u) = f - \mathcal{L}u$, the standard Galerkin method can be interpreted in form of the residual orthogonality $(R(u), v) = 0$. In the case of non self-adjoint differential operators like the diffusion-convection-reaction operator \mathcal{L} , it can happen that $(\mathcal{L}u, v)$ is not coercive or symmetric on V , and the resulting FEM discretization may be unstable.

For transport problems, another popular approach is based on the least squares formulation of the Galerkin FEM. Let us write the simple form of the model problem (3.1) as

$$\mathcal{L}u + r(u) = f \quad \text{in } \Omega, \tag{3.34a}$$

$$u = g \quad \text{on } \partial\Omega. \tag{3.34b}$$

Define the least-squares functional

$$J(u) := \frac{1}{2} \|\mathcal{L}u + r(u) - f\|_{L^2(\Omega)}^2.$$

A minimizer of $J(u)$ is obtained by

$$\lim_{t \rightarrow 0} \frac{d}{dt} J(u + tv) = 0, \quad \forall v \in V,$$

which yields the least-squares term

$$\tilde{J}_\Omega(u, v) := (\mathcal{L}u + r(u) - f, \mathcal{L}v + r'(u)v)_{L^2(\Omega)}.$$

For linear problems with $r(u) = 0$, the least squares Galerkin method reduces to the minimization problem

$$F(u) = \min_{v \in V} F(v),$$

where the functional $F(\cdot)$ is defined by

$$F(v) = \frac{1}{2} \| \mathcal{L}v - f \|_{L^2(\Omega)}^2.$$

The first order optimality condition leads to the least squares Galerkin method

$$(\mathcal{L}u, \mathcal{L}v) = (f, \mathcal{L}v), \quad \forall v \in V.$$

The bilinear form $(\mathcal{L}u, \mathcal{L}v)$ is symmetric and coercive and has stronger stability properties compared the standard Galerkin method.

There are many publications on the Galerkin least squares finite element methods (GLSFEM). We mention here two books [22, 54] and the review article [21]. There are mainly two variants of the GLSFEMs; the stabilized and the direct versions.

Stabilized finite elements method [53]: The standard (continuous) Galerkin FEM for the problem (3.34) reads: find $u_h \in U_h \subset U$ (U : solution space) such that

$$a(u_h, v_h) + (r(u_h), v_h)_{L^2(\Omega)} = (f, v_h)_{L^2(\Omega)}, \quad \forall v_h \in V_h \subset V \quad (3.35)$$

where $a(u, v) = (\epsilon \nabla u + \vec{\beta} \cdot \nabla u + \alpha u, v)_{L^2(\Omega)}$ is the standard bilinear form to the linear part of (3.34). The general stabilized FEMs formulation reads as: for all $v_h \in V_h \subset V$, find $u_h \in U_h \subset U$ such that

$$a(u_h, v_h) + (r(u_h), v_h)_{L^2(\Omega)} + \sum_{\mathbf{K}} \tau_{\mathbf{K}} \mathbf{S}_{\mathbf{K}}(\mathbf{u}_h, \mathbf{v}_h) = (f, v_h)_{L^2(\Omega)} \quad (3.36)$$

where the stabilization parameter is defined on each element K as [48]

$$\tau_K = \frac{1}{\frac{4\epsilon}{h^2} + \frac{2|\vec{\beta}|}{h} + |\alpha|}.$$

One way to proceed GLSFEMs is then to use the least-squares term $\tilde{J}_K(u, v)$ as the stabilization term S_K in (3.36), i.e.: for all $v_h \in V_h$, find $u_h \in U_h$ such that

$$a(u_h, v_h) + (r(u_h), v_h)_{L^2(\Omega)} + \sum_K \tau_K \tilde{J}_K(u_h, v_h) = (f, v_h)_{L^2(\Omega)} \quad (3.37)$$

Note that, being another stabilized FEM, streamline upwind Petrov-Galerkin (SUPG) method is obtained by setting

$$S_K(u_h, v_h) = (\mathcal{L}u_h + r(u_h) - f, \vec{\beta} \cdot \nabla v_h)_{L^2(K)}$$

with different choices of the parameter τ_K .

The direct variant of GLSFEM: The second way to proceed the GLSFEMs is to consider and discretize just the least-squares term $\tilde{J}_\Omega(u, v)$. One may solve this problem in a straightforward manner: for all $v_h \in V_h \subset H^2(\Omega) \cap V$, find $u_h \in U_h \subset H^2(\Omega) \cap U$ such that $\tilde{J}_\Omega(u_h, v_h) = 0$, i.e.

$$\int_\Omega (\mathcal{L}u_h + r(u_h))(\mathcal{L}v_h + r'(u_h)v_h)dx = \int_\Omega f(\mathcal{L}v_h + r'(u_h)v_h)dx$$

which is not only a fourth order problem but also the solution and trial subspaces U_h and V_h need to consist of continuously differentiable functions making it complicated to construct bases functions (standard finite element spaces cannot be used anymore) and the assembly of the stiffness matrix. The condition number of the stiffness is order of $\mathcal{O}(h^{-4})$ instead of order $\mathcal{O}(h^{-2})$ for the standard Galerkin FEM. Hence, this approach is impractical. Instead, being the most common practical way, the problem (3.34) is converted into a first-order system as [50, 21].

$$\begin{aligned} p - \nabla u &= 0 & \text{in } \Omega \\ -\epsilon \nabla \cdot p + \vec{\beta} \cdot \nabla u + \alpha u + r(u) &= f & \text{in } \Omega \\ u &= g & \text{on } \Gamma \end{aligned}$$

Then, we define now the least-square functional for $z = (p, u)^T$ as

$$J(z) := \frac{1}{2} \|p - \nabla u\|_{L^2(\Omega)}^2 + \frac{1}{2} \| -\epsilon \nabla \cdot p + \vec{\beta} \cdot \nabla u + \alpha u + r(u) - f \|_{L^2(\Omega)}^2$$

A minimizer of $J(z)$ is obtained by the identity

$$\lim_{t \rightarrow 0} \frac{d}{dt} J(z + tv) = 0, \quad \forall v$$

which yields a least-squares term of order two. Using this approach, we solve the resulting least-squares term, which is a second-order problem now, using continuous (discontinuous) finite elements solution and trial spaces $S_h \subset H^1(\Omega, div) \times U$ ($S_h \subset H^1(\Omega, div) \times H^1(\Omega)$) and $T_h \subset H^1(\Omega, div) \times V$ ($T_h \subset H^1(\Omega, div) \times H^1(\Omega)$), respectively. The condition number of the stiffness matrix is retained as $\mathcal{O}(h^{-2})$ as in the standard Galerkin method [21]. For convection dominated problems, the resulting linear systems of equations are solved usually with preconditioned conjugate gradient method due to large condition numbers, as reported in [60] for GLSFEM solution of singularly perturbed diffusion-convection problems.

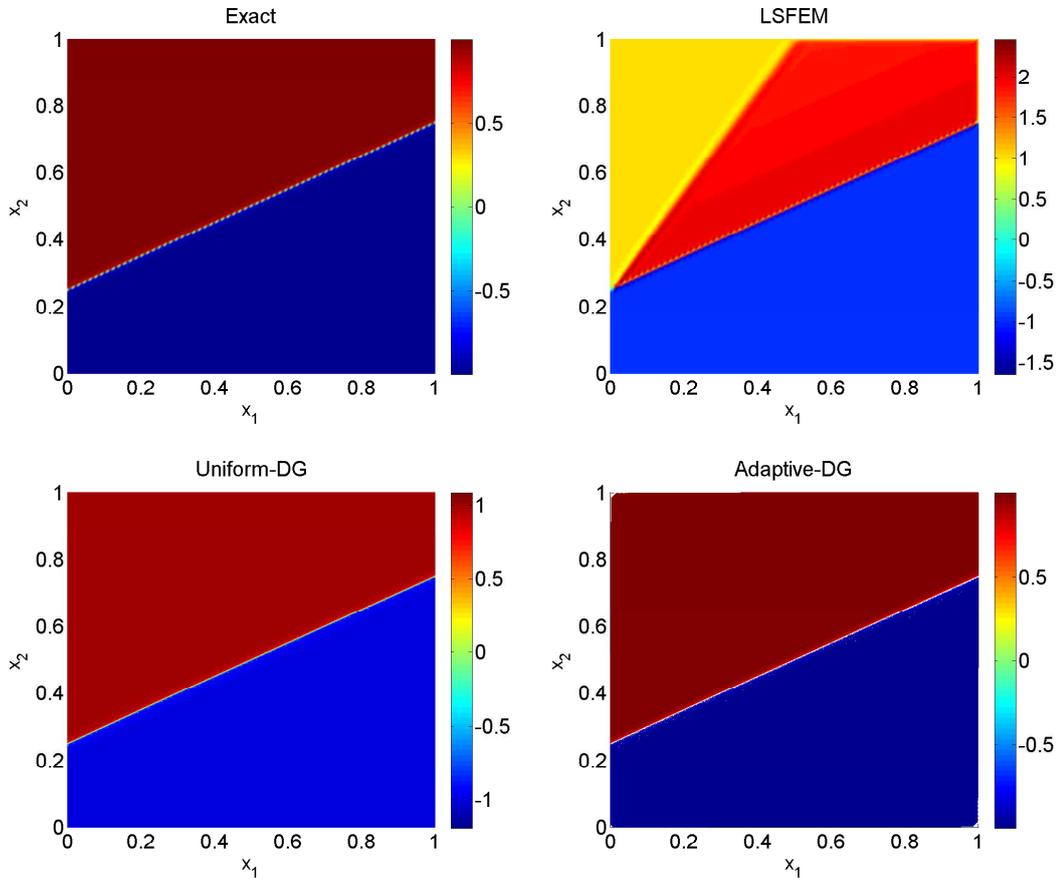


Figure 3.3: Solution profiles of linear convection dominated problem (3.38)

In order to compare the GLSFEM with the DGFEM and DGAFEM, we consider the linear problem [89]

$$-\epsilon \Delta u + \vec{\beta} \cdot \nabla u + \alpha u = f \quad \text{in } (0, 1)^2 \quad (3.38)$$

with $\epsilon = 10^{-6}$, $\vec{\beta} = (2, 3)^T$ and $\alpha = 1$. The load function f and Dirichlet boundary conditions are chosen so that the exact solution is

$$u(x_1, x_2) = \frac{\pi}{2} \arctan \left(\frac{1}{\sqrt{\epsilon}} (-0.5x_1 + x_2 - 0.25) \right)$$

Fig.3.3 shows that GLSFEM is not stable for the non-linear model (3.38), and even uniform DG scheme is capable of solving the problem. We can see from the Fig.3.4 the global L^2 -errors and the result that the solutions obtained by DGAFEM produces more accurate profiles than the DGFEM for the convection dominated problem (3.38).

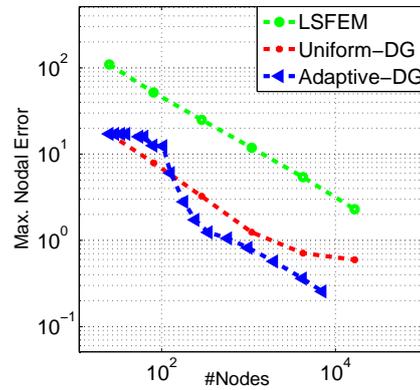


Figure 3.4: Comparison of the GLSFEM and DGFEM for linear convection dominated problem (3.38)

3.5 Numerical Examples

In this section, we give several numerical examples demonstrating the effectiveness and accuracy of the DGAFEM for convection dominated non-linear diffusion-convection-reaction equations of the form (3.1).

3.5.1 Example with Polynomial Type Non-Linearity

Our first example is taken from [15] with Dirichlet boundary condition on $\Omega = (0, 1)^2$ with $\epsilon = 10^{-6}$, $\vec{\beta} = \frac{1}{\sqrt{5}}(1, 2)^T$, $\alpha = 1$ and $r(u) = u^2$. The source function f and Dirichlet boundary condition are chosen so that $u(x_1, x_2) = \frac{1}{2} \left(1 - \tanh \frac{2x_1 - x_2 - 0.25}{\sqrt{5\epsilon}} \right)$

is the exact solution. The problem is characterized by an internal layer of thickness $\mathcal{O}(\sqrt{\epsilon} |\ln \epsilon|)$ around $2x_1 - x_2 = \frac{1}{4}$.

The mesh is locally refined by DGAFEM around the interior layer (Fig.3.5) and the spurious solutions are damped out in Fig.3.6, similar to the results as in [15] using SUPG-SC, in [90] with SIPG-SC. On adaptively and uniformly refined meshes, from the Fig.3.7, it can be clearly seen that the adaptive meshes reduce the substantial computing time. On uniform meshes, the SIPG is slightly more accurate as shown in [90] than the SUPG-SC in [15]. The error reduction by increasing degree of the polynomials is remarkable on finer adaptive meshes (Fig.3.7, right).

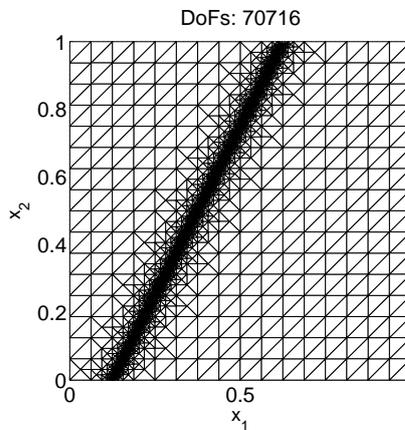


Figure 3.5: Example 3.5.1; Adaptive mesh

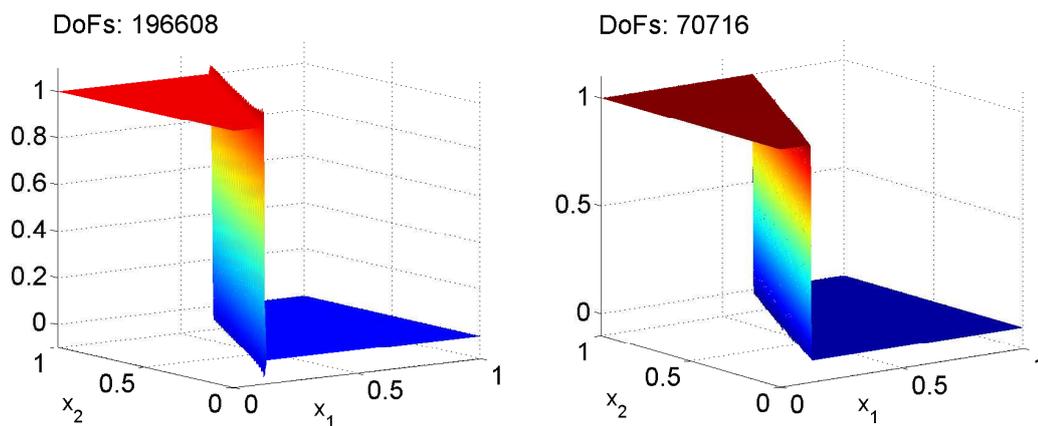


Figure 3.6: Example 3.5.1: Uniform (left) and adaptive (right) solutions, quadratic elements

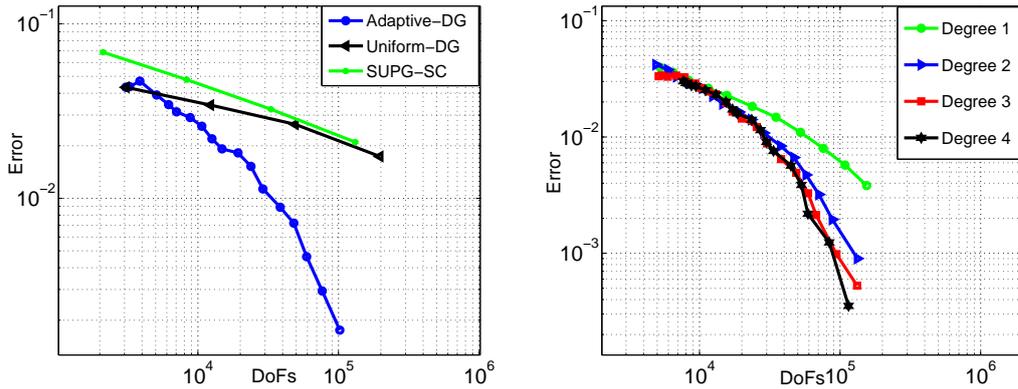


Figure 3.7: Example 3.5.1: Global errors: comparison of the methods by quadratic elements (left), adaptive DG for polynomial degrees 1-4 (right)

In Table 3.1, we give the results using the solution technique in Section 3.3.1 for the *BiCGStab* with the stopping criterion as $\|r_i\|_2/\|r_0\|_2 \leq tol$ for $tol = 10^{-7}$ (r_i is the residual of the corresponding system at the i^{th} iteration) applied to the unpermuted system and Schur complement system with and without preconditioning on the finest levels of uniformly (4^{th} refinement level with DoFs 196608 and 32768 triangular elements) and adaptively (17^{th} refinement level with DoFs 70716 and 11786 triangular elements) refined meshes. As a preconditioner, the incomplete LU factorization of the Schur complement matrix S ($ILU(S)$) is used for the linear system with the coefficient matrix S . The linear systems with the coefficient matrix A are solved directly. Table 3.1 shows that solving the problem via the block LU factorization using the Schur complement system with the preconditioner $ILU(S)$ is the fastest.

Table 3.1 Example 3.5.1: Average number of Newton iterations, average number of *BiCGStab* iterations, total computation time in seconds corresponding to the uniformly refined (adaptively refined) mesh

| Linear Solver | # Newton | # BiCGStab | Time |
|------------------------------------|-------------|---------------|----------------|
| BiCGStab w/o prec. (Unpermuted) | 10.8 (10.5) | 818 (757.5) | 1389.3 (773.3) |
| BiCGStab w/ prec. M_1 (Permuted) | 10.3 (10.3) | 1.5 (3) | 423.1 (374.2) |
| BiCGStab w/ prec. M_2 (Permuted) | 10.3 (10.3) | 1.5 (3) | 416.8 (375.9) |
| Block LU + (BiCGStab w/o prec.) | 10.3 (10.9) | 247.5 (315.5) | 270.9 (310.3) |
| Block LU + (BiCGStab w/ prec.) | 10.3 (10.9) | 19 (28.5) | 140.9 (114.7) |

The time for applying the permutation to obtain the reordered matrix and the permutation matrix P takes 9.9 seconds, whereas, it takes 0.13 seconds to form the Schur complement matrix S and 0.04 seconds to compute the $\text{ILU}(S)$ on a PC with Intel Core-i7 processor and 8GB RAM using the 64-bit version of Matlab-R2010a. We note that since the Jacobian matrix does not change during the non-linear iterations, the permutation, the Schur complement matrix and $\text{ILU}(S)$ is computed only once for each run.

In all of the following results and figures, the Jacobian matrix J is scaled by a left Jacobi preconditioner before reordering to obtain a well conditioned matrix. The reordering procedure is applied to the scaled Jacobian matrix. Reordering times, which are included in the total computation time, for the uniform and adaptive refinements are 102.1 seconds and 41.4 seconds, respectively.

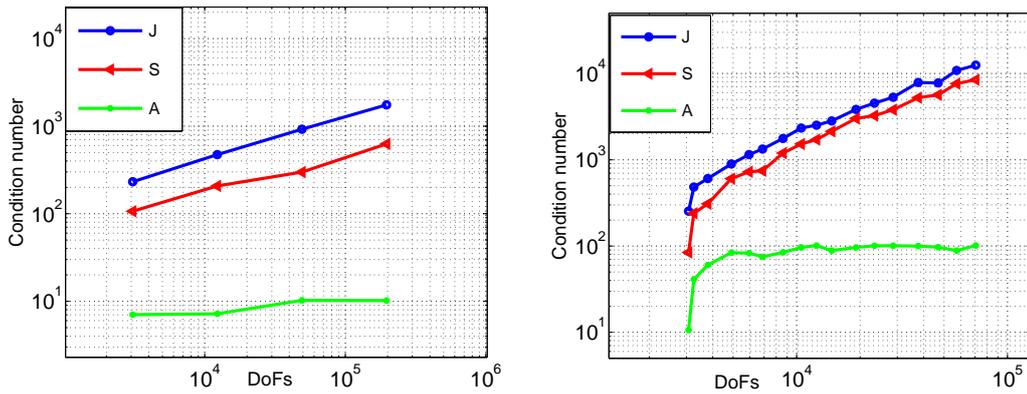


Figure 3.8: Example 3.5.1: Condition number of the matrices J (unpermuted matrix), S (Schur complement matrix) and A (left top block of permuted matrix): Uniform refinement (left) and adaptive refinement (right)

Fig.3.8 shows the condition numbers of the Jacobian matrices J of the original system, S and A of the block LU factorized system on the uniformly and adaptively refined meshes. The condition numbers of the coefficient matrix A are almost constant for uniform refinement by different orders of DG discretizations and around 10, whereas the condition number of S lower than of the Jacobian matrix J . This is due to the clustering of nonzero elements around the diagonal (Fig.3.9) due to the matrix reordering. For adaptive refinement, Fig.3.8, right, we observe the same behavior, whereas the conditions numbers are larger of order one than for the uniform

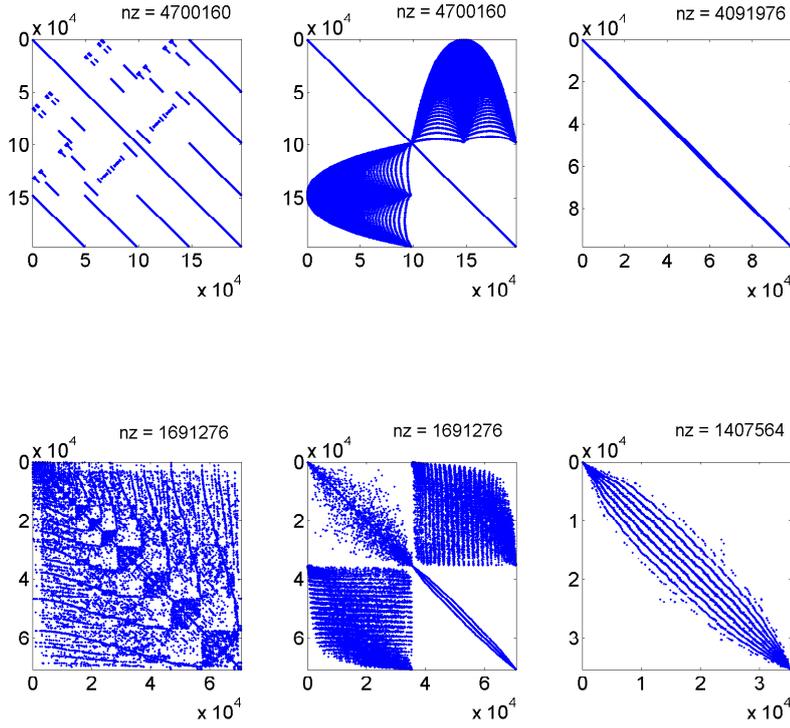


Figure 3.9: Example 3.5.1: Sparsity patterns of the unpermuted (left), permuted (middle) and the Schur complement (right) matrices at the final refinement levels: Uniform refinement (top) with DoFs 196608 and adaptive refinement (bottom) with DoFs 70716

refinement. For comparison, we provide results by using BiCGStab with two block preconditioners. The preconditioning matrices M_1 and M_2 for the permuted full systems are given as

$$M_1 = \begin{pmatrix} A & 0 \\ C^T & S \end{pmatrix}, \quad M_2 = \begin{pmatrix} A & B \\ 0 & S \end{pmatrix}.$$

Total number of iterations and time for different algorithms are given in Table 3.1. Our proposed method where we compute the block LU factorization of the partitioned matrix and solve the system involving the Schur complement iteratively via preconditioned BiCGStab is the best in terms of the total time compared to other methods for both uniform and adaptive refinement. In Fig.3.10 and Fig.3.11, we present the total time and the average number of linear solver iterations, respectively, for uniform

and adaptive refinements as the problem size has been increased. We observe that the proposed preconditioned linear solver has been the best in terms of time with a reasonable number of iterations for different problem sizes regardless of refinement type.

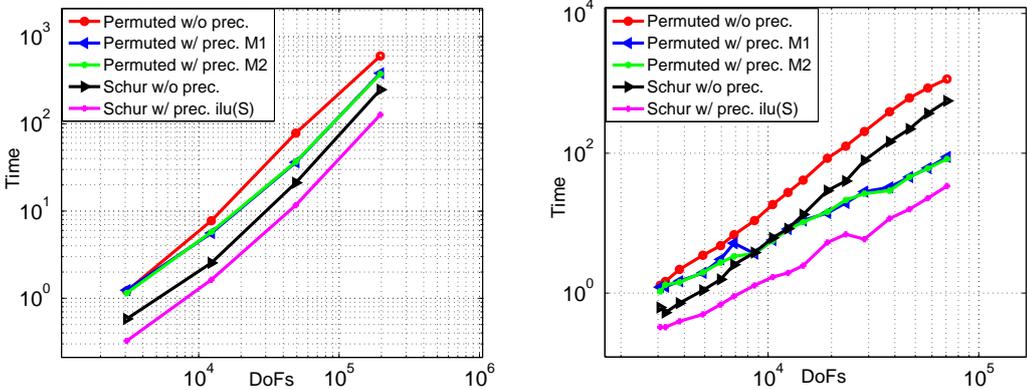


Figure 3.10: Example 3.5.1: Computation time vs. DoFs: Uniform refinement (left) and adaptive refinement (right)

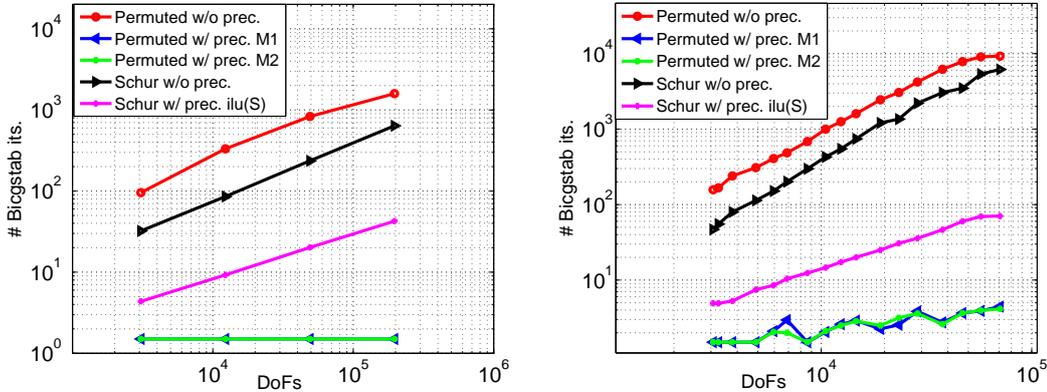


Figure 3.11: Example 3.5.1: # Average *BiCGStab* iterations vs. DoFs: Uniform refinement (left) and adaptive refinement (right)

3.5.2 Example with Monod Type Non-Linearity

We consider the problem in [15] of type (3.1) on $\Omega = (0, 1)^2$, and having a Monod type non-linearity $r(u) = -u/(1 + u)$ and homogeneous source function. The con-

vection field and the diffusion coefficient are given as $\vec{\beta}(x_1, x_2) = (-x_2, x_1)^T$ and $\epsilon = 10^{-6}$, respectively. The Dirichlet boundary condition is prescribed as

$$u(x_1, x_2) = \begin{cases} 1 & \text{for } 1/3 \leq x_1 \leq 2/3, x_2 = 0, \\ 0 & \text{for } x_1 < 1/3, x_1 > 2/3, x_2 = 0, \\ 0 & \text{for } x_1 = 1 \text{ or } x_2 = 1. \end{cases}$$

On the left boundary ($x_1 = 0, 0 \leq x_2 \leq 1$), no-flow condition is assumed.

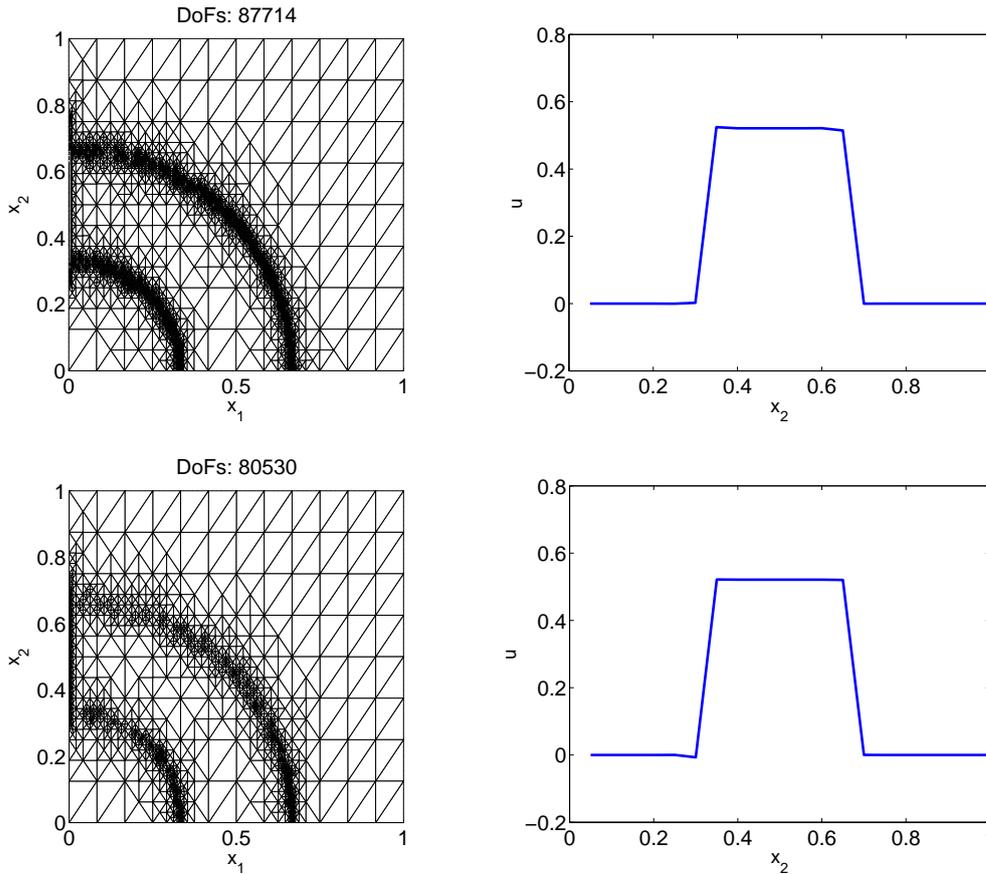


Figure 3.12: Example 3.5.2: Adaptive meshes (left) and the cross-section plots (right) of the solutions at the left outflow boundary by quadratic elements (top) and quartic elements (bottom)

There are both internal and boundary layers on the mesh (Fig.3.12, left), around them oscillations occur. Fig.3.12, right, shows that by DGAFEM, the oscillations are almost disappear, similar to the results in [15] for the SUPG-SC and in [90] for SIPG-SC. Fig.3.12, left, shows that the adaptive process leads to correctly refined meshes. Moreover, by increasing polynomial degree of the basis functions ($k = 4$), the oscil-

lations are completely eliminated on the outflow boundary (Fig.3.12, bottom) and the sharp front is preserved. This is not the case for SUPG-SC [15] and SIPG-SC [90], where still small oscillations are present.

As in case of polynomial non-linearity, Example 3.5.1, the block LU factorized system solved by BiCGStab with the preconditioner ILU(S) is the most efficient solver, with an average number of 7 Newton iterations. The computing times for the uniform refinement was 20.6 seconds, and 30.5 for the adaptive refinement.

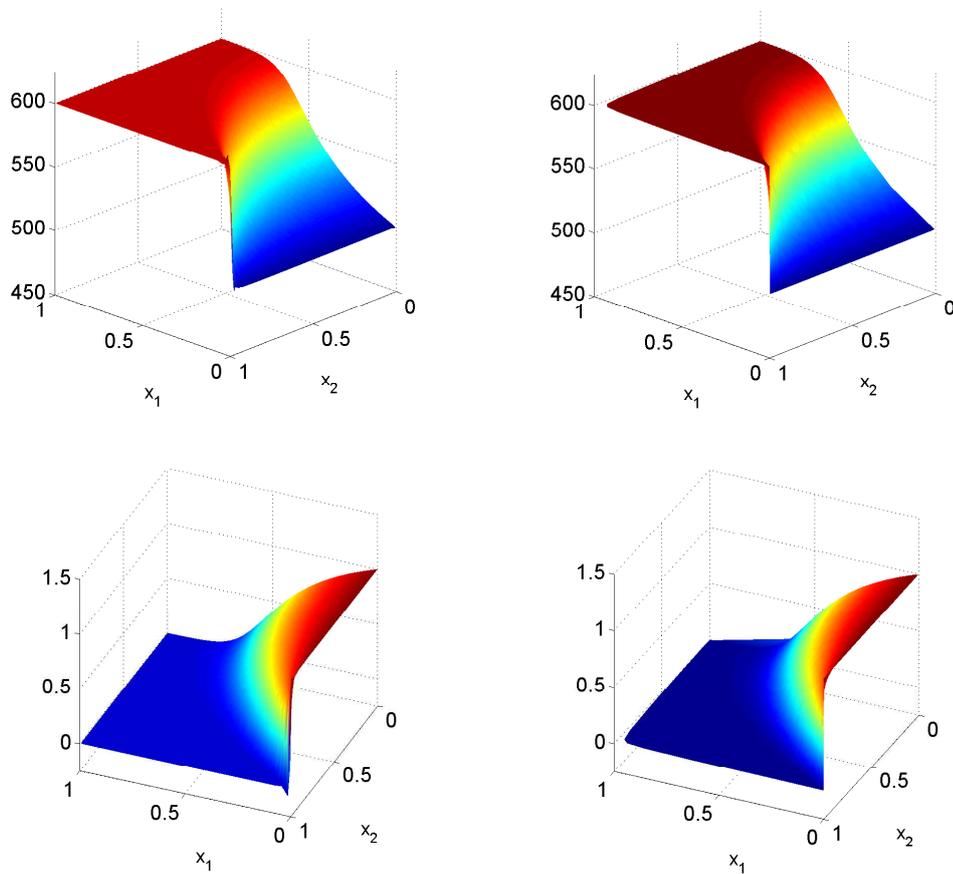


Figure 3.13: Example 3.5.3: Uniform(left) and adaptive(right) solutions to the temperature(top) and reactant(bottom), quadratic elements with DoFs 12288 for uniform refinement and with DoFs 6168 for adaptive refinement

3.5.3 Coupled Example with Arrhenius Type Non-Linearity

Next example is the non-linear reaction for a two-component system in [81]:

$$\begin{aligned} -\nabla \cdot (\epsilon \nabla u_1) + \vec{\beta} \cdot \nabla u_1 - 100k_0 u_2 e^{\frac{-E}{Ru_1}} &= 0, \\ -\nabla \cdot (\epsilon \nabla u_2) + \vec{\beta} \cdot \nabla u_2 + k_0 u_2 e^{\frac{-E}{Ru_1}} &= 0, \end{aligned}$$

on $\Omega = (0, 1)^2$ with the convection field $\vec{\beta} = (1 - x_2^2, 0)^T$, the diffusion constant $\epsilon = 10^{-6}$, the reaction rate coefficient $k_0 = 3 \times 10^8$ and the quotient of the activation energy to the gas constant $\frac{E}{R} = 10^4$. The unknowns u_1 and u_2 represent the temperature of the system and the concentration of the reactant, respectively.

There are oscillations (Fig.3.13, left) around the layers, even small, for the uniform refinement as for SIPG-SC in [90]. On the other hand, these oscillations are completely dumped out by DGAFEM with almost half of the DoFs used in the uniform refinement (Fig.3.13, right).

The block LU factorization based algorithm with the preconditioner ILU(S) requires 10.5 seconds for the uniform and 24.4 seconds for the adaptive refinements. Matrix reordering and permutation took 2.44 seconds for the uniform and 2.17 seconds for adaptive refinements, respectively.

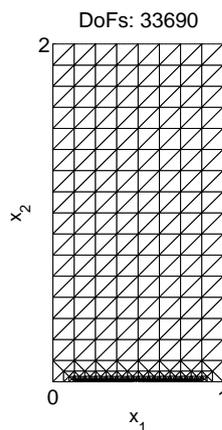


Figure 3.14: Example 3.5.4: Adaptive mesh, quartic elements with DoFs 33690

3.5.4 Coupled Example with Polynomial Type Non-Linearity

Our final problem is the modification of the non-stationary transport problem, Example 2, in [17]. The problem is stated as the following:

$$\begin{aligned}\alpha u_1 - \nabla \cdot (\epsilon \nabla u_1) + \vec{\beta} \cdot \nabla u_1 + 50u_1^2 u_2^2 &= 0, \\ \alpha u_2 - \nabla \cdot (\epsilon \nabla u_2) + \vec{\beta} \cdot \nabla u_2 + 50u_1^2 u_2^2 &= 0,\end{aligned}$$

on the rectangular domain $\Omega = (0, 1) \times (0, 2)$ with the convection field $\vec{\beta} = (0, -1)^T$, the diffusion constant $\epsilon = 10^{-10}$ and linear reaction constant $\alpha = 0.1$. On the left, right and lower parts of the boundary of the domain, Neumann boundary conditions are prescribed. On the remaining part of the boundary, Dirichlet boundary conditions are chosen as

$$u_1(\mathbf{x}) = \begin{cases} 8(x_1 - 0.375) & \text{for } 0.375 < x_1 \leq 0.5, \\ -8(x_1 - 0.625) & \text{for } 0.5 < x_1 \leq 0.625, \\ 0 & \text{otherwise} \end{cases}$$

$$u_2(\mathbf{x}) = \begin{cases} 8(x_1 - 0.125) & \text{for } 0.125 \leq x_1 \leq 0.25, \\ -8(x_1 - 0.375) & \text{for } 0.25 < x_1 \leq 0.375, \\ 8(x_1 - 0.625) & \text{for } 0.625 \leq x_1 \leq 0.75, \\ -8(x_1 - 0.875) & \text{for } 0.75 < x_1 \leq 0.875, \\ 0 & \text{otherwise.} \end{cases}$$

There is a boundary layer on the outflow boundary, Fig.3.14. Fig.3.15 shows that oscillations are almost damped using DGAFEM approximations, similar to those results in [17] using SUPG-SC. It can be seen from Fig.3.14 that the mesh is correctly refined by DGAFEM near the boundary layer.

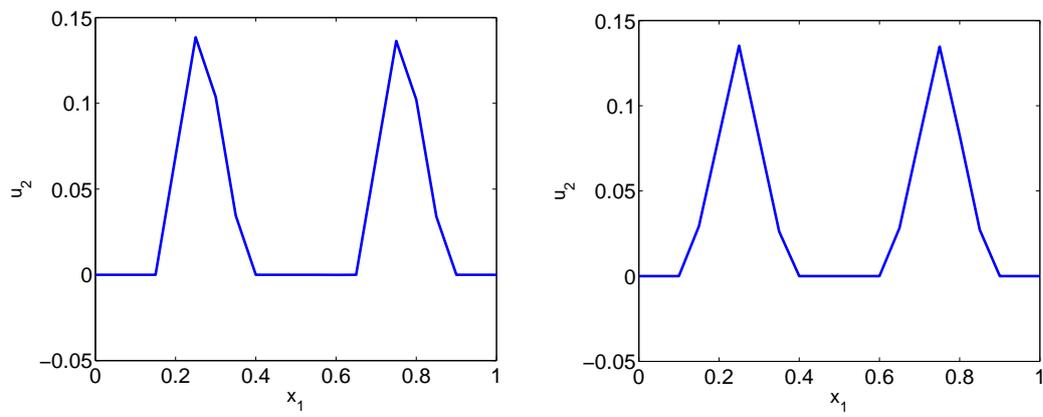


Figure 3.15: Example 3.5.4: Uniformly (left) and adaptively (right) obtained cross-section plots on the outflow boundary for the component u_2 , quartic elements with DoFs 61440 for uniform refinement and with DoFs 33690 for adaptive refinement

CHAPTER 4

PARABOLIC PROBLEMS WITH TIME-SPACE ADAPTIVITY

The aim of this chapter is to propose an adaptive algorithm both in space and time applied to the DG discretized (in space) systems using (residual-based) a posteriori error estimates. We derive a posteriori error estimates for semi-linear problems of the form (1.1) using the ready elliptic a posteriori estimates derived in Chapter 3, by which in contrast to the standard energy techniques we do not need to try to adapt the estimates case by case in order to compare the exact solution with numerical solution directly. For this reason, we use the *elliptic reconstruction* technique in [63] which allows us to utilize ready a posteriori estimates derived for elliptic models to bound the main part of the spatial error. The idea of the elliptic reconstruction technique is to construct an auxiliary solution whose difference to the numerical solution can be estimated by a known (elliptic) a posteriori estimate, and the constructed auxiliary solution satisfies a variant of the given problem with a right hand side which can be controlled in an optimal way. By this way, we are able to obtain results being optimal order in both $L^2(H^1)$ and $L^\infty(L^2)$ -type norms, while the results obtained by the standard energy methods are only optimal order in $L^2(H^1)$ -type norms, but sub-optimal order in $L^\infty(L^2)$ -type norms. In [25], a posteriori error estimates in the $L^\infty(L^2) + L^2(H^1)$ -type norm are derived for linear parabolic diffusion-convection-reaction equations using backward Euler in time and discontinuous Galerkin in space utilizing the elliptic reconstruction technique. In this chapter, we extend the study in [25] in a similar way by deriving and implementing a posteriori error estimates in the $L^\infty(L^2) + L^2(H^1)$ -type norm using backward Euler in time and SIPG in space for the convection dominated parabolic problems with non-linear reaction mechanisms.

Application of the adaptive discontinuous Galerkin methods and a posteriori error estimates to the problems in geoscience are reviewed recently in [31]. Most of the applications of DG methods in geoscience concern reactive transport with advection [13, 58, 78] and strong permeability contrasts such as layered reservoirs [83] or vanishing and varying diffusivity posing challenges in computations [68]. The permeability in heterogeneous porous and fractured media varies over orders of magnitude in space, which results in highly variable flow field, where the local transport is dominated by advection or diffusion [79]. Accurate and efficient numerical solution of the ADR equations to predict the macroscopic mixing, anomalous transport of the solutes and contaminants for a wide range of parameters like permeability and Péclet numbers, different flow velocities and reaction rates are challenging problems [79]. In order to resolve the complex flow patterns accurately, higher order time stepping methods like exponential time stepping methods are used [79]. We show here, using time-space adaptive first order backward Euler and DG in space, the same results can be obtained.

4.1 Preliminaries and Model Equation

Let $\Omega \subset \mathbb{R}^2$ be a bounded, open and convex domain with boundary $\partial\Omega$. For a Banach space X , define the spaces $L^p(0, T; X)$

$$\|v\|_{L^p(0,T;X)} = \left(\int_0^T \|v(t)\|_X^p dt \right)^{1/p} < \infty, \quad \text{for } 1 \leq p < +\infty,$$

$$\|v\|_{L^\infty(0,T;X)} = \operatorname{esssup}_{0 \leq t \leq T} \|v(t)\|_X < \infty, \quad \text{for } p = +\infty.$$

We also introduce the space

$$H^1(0, T; X) = \{v \in L^2(0, T; X) \mid v_t \in L^2(0, T; X)\}.$$

We denote by $C(0, T; X)$ and $C^{0,1}(0, T; X)$ the spaces of continuous and Lipschitz-continuous functions $v : [0, T] \mapsto X$, respectively, equipped with the norms

$$\|v\|_{C(0,T;X)} = \max_{0 \leq t \leq T} \|v(t)\|_X < \infty,$$

$$\|v\|_{C^{0,1}(0,T;X)} = \max \{ \|v\|_{C(0,T;X)}, \|v_t\|_{C(0,T;X)} \} < \infty.$$

The model equation we consider in this Chapter is the system of semi-linear diffusion-convection-reaction equations of type (1.1) for an indexed set $i = 1, 2, \dots, m$ given by

$$\frac{\partial u_i}{\partial t} - \nabla \cdot (\epsilon_i \nabla u_i) + \vec{\beta}_i \cdot \nabla u_i + r_i(\vec{u}) = f_i \quad (4.1)$$

in $\Omega \times (0, T]$ for the vector of unknowns $\vec{u} = \vec{u}(x, t) = (u_1, u_2, \dots, u_m)^T$ with appropriate boundary and initial conditions, and with $u_j = u_j(x, t)$, $j = 1, 2, \dots, m$. We assume that the source functions $f_i \in C(0, T; L^2(\Omega))$, and the velocity fields $\vec{\beta}_i \in C(0, T; W^{1, \infty}(\Omega))^2$ either given or computed. To be applicable to the models of flow in heterogeneous media, the symmetric dispersion tensors ϵ_i are taken of the form

$$\epsilon_i = \begin{bmatrix} D_i^1 & 0 \\ 0 & D_i^2 \end{bmatrix}$$

with $0 < D_i^1, D_i^2 \ll 1$. Moreover, we assume that the non-linear reaction terms are bounded, locally Lipschitz continuous and monotone, in other words, they satisfy for any $s, s_1, s_2 \geq 0$, $s, s_1, s_2 \in \mathbb{R}$ the following conditions

$$|r_i(s)| \leq C_S, \quad C_S > 0, \quad s \in [-S, S], \quad (4.2a)$$

$$\|r_i(s_1) - r_i(s_2)\|_{L^2(\Omega)} \leq L(S) \|s_1 - s_2\|_{L^2(\Omega)}, \quad L > 0, \quad (4.2b)$$

$$r_i \in C^1(\mathbb{R}_0^+), \quad r_i(0) = 0, \quad r_i'(s) \geq 0. \quad (4.2c)$$

We further assume that there are $\alpha_0, c_* \geq 0$ satisfying for $i = 1, 2, \dots, m$

$$\| -\nabla \cdot \vec{\beta}_i(x, t) \|_{C(0, T; L^\infty(\Omega))} \leq c_* \alpha_0. \quad (4.3)$$

In the sequel, for simplicity, we just consider a single equation of the system (4.1) ($m=1$) without any subscript to construct the SIPG discretization in space and a homogeneous dispersion tensor leading to a simple diffusivity constant $0 < \epsilon \ll 1$. We further take into account the homogeneous Dirichlet boundary condition on the whole boundary to simplify a posteriori error constructions. It can be proceeded with heterogeneous dispersion tensor and inhomogeneous Dirichlet or Neumann boundary condition in a standard way.

Thus, we consider the semi-linear system

$$\frac{\partial u}{\partial t} - \epsilon \Delta u + \vec{\beta} \cdot \nabla u + r(u) = f \quad \text{in } \Omega \times (0, T], \quad (4.4a)$$

$$u(\cdot, t) = 0 \quad \text{on } \partial\Omega \times (0, T], \quad (4.4b)$$

$$u(\cdot, 0) = u_0 \quad \text{in } \Omega. \quad (4.4c)$$

Then, the standard weak formulation of (4.4) reads as: for any $v \in H_0^1(\Omega)$, find $u \in L^2(0, T; H_0^1(\Omega)) \cap H^1(0, T; L^2(\Omega))$ satisfying for all $t \in (0, T]$

$$\int_{\Omega} \frac{\partial u}{\partial t} v dx + a(t; u, v) + b(t; u, v) = l(v), \quad (4.5)$$

$$a(t; u, v) = \int_{\Omega} (\epsilon \nabla u \cdot \nabla v + \vec{\beta} \cdot \nabla uv) dx, \quad (4.6a)$$

$$b(t; u, v) = \int_{\Omega} r(u) v dx, \quad (4.6b)$$

$$l(v) = \int_{\Omega} f v dx, \quad (4.6c)$$

which have a unique solution in the space $C(0, T; L^2(\Omega))$ under the given regularity assumptions and the conditions (4.2a)-(4.2c). Further, using the definitions, one can easily show that the bilinear form $a(t; u, v)$ is coercive and continuous on the space $H_0^1(\Omega)$ such that

$$a(t; v, v) \geq |||v|||^2, \quad \forall v \in H_0^1(\Omega), \quad (4.7)$$

$$a(t; u, v) \leq |||u|||_{DG} |||v|||, \quad \forall v \in H_0^1(\Omega), \forall u \in H_0^1(\Omega) + V_h. \quad (4.8)$$

4.2 Semi-Discrete and Fully Discrete Formulations

In this section, we introduce the semi-discrete and fully-discrete formulations of the model (4.4). Most of the notations here are analogue to the ones introduced in Chapter 3, but now time-dependent.

4.2.1 Semi-Discrete Formulation

Let the mesh $\xi = \{K\}$ be a family of shape regular elements (triangles). We set the mesh-dependent finite dimensional solution and test function space by

$$V_h = V_h(\xi) = \{v \in L^2(\Omega) : v|_K \in \mathbb{P}_k(K), \forall K \in \xi\} \not\subset H_0^1(\Omega).$$

For a given $t \in [0, T]$, we restate the sets of inflow and outflow edges by

$$\Gamma_t^- = \{x \in \partial\Omega : \vec{\beta}(x, t) \cdot \vec{n}(x) < 0\}, \quad \Gamma_t^+ = \partial\Omega \setminus \Gamma_t^-,$$

$$\partial K_t^- = \{x \in \partial K : \vec{\beta}(x, t) \cdot \vec{n}_K(x) < 0\}, \quad \partial K_t^+ = \partial K \setminus \partial K_t^-.$$

Moreover, we denote by Γ_h^0 and Γ_h^∂ the set of interior and boundary edges, respectively, so that the union set $\Gamma_h = \Gamma_h^0 \cup \Gamma_h^\partial$ forms the skeleton of the mesh. Then, utilizing the SIPG construction in Chapter 3, the semi-discrete formulation of (4.4) reads as: for $t = 0$ set $u_h(0) \in V_h(\xi)$ as the projection (orthogonal L^2 -projection) of u_0 onto $V_h(\xi)$, and for each $t \in (0, T]$, for all $v_h \in V_h(\xi)$, find $u_h \in C^{0,1}(0, T; V_h(\xi))$ such that

$$\int_{\Omega} \frac{\partial u_h}{\partial t} v_h dx + a_h(t; u_h, v_h) + K_h(u_h, v_h) + b_h(t; u_h, v_h) = l_h(v_h), \quad (4.9)$$

where the forms are given by

$$\begin{aligned} a_h(t; u_h, v_h) &= \sum_{K \in \xi} \int_K \epsilon \nabla u_h \cdot \nabla v_h dx + \sum_{K \in \xi} \int_K \vec{\beta} \cdot \nabla u_h v_h dx \\ &\quad + \sum_{e \in \Gamma_h} \frac{\sigma \epsilon}{h_e} \int_e [u_h] \cdot [v_h] ds - \sum_{K \in \xi} \int_{\partial K_t^- \cap \Gamma_t^-} \vec{\beta} \cdot \vec{n}_K u_h v_h ds \\ &\quad + \sum_{K \in \xi} \int_{\partial K_t^- \setminus \partial\Omega} \vec{\beta} \cdot \vec{n}_K (u_h^{out} - u_h) v_h ds, \end{aligned} \quad (4.10a)$$

$$K_h(u_h, v_h) = - \sum_{e \in \Gamma_h} \int_e (\{\epsilon \nabla v_h\} \cdot [u_h] + \{\epsilon \nabla u_h\} \cdot [v_h]) ds, \quad (4.10b)$$

$$b_h(t; u_h, v_h) = \sum_{K \in \xi} \int_K r(u_h) v_h dx, \quad (4.10c)$$

$$l_h(v_h) = \sum_{K \in \xi} \int_K f_h v_h dx. \quad (4.10d)$$

Upon integration by parts on the convective term, bilinear form (4.10a) will be

$$\begin{aligned}
a_h(t; u_h, v_h) &= \sum_{K \in \xi} \int_K \epsilon \nabla u_h \cdot \nabla v_h dx - \sum_{K \in \xi} \int_K (\vec{\beta} u_h \cdot \nabla v_h + \nabla \cdot \vec{\beta} u_h v_h) dx \\
&+ \sum_{e \in \Gamma_h} \frac{\sigma \epsilon}{h_e} \int_e [u_h] \cdot [v_h] ds + \sum_{K \in \xi} \int_{\partial K_t^+ \cap \Gamma_t^+} \vec{\beta} \cdot \vec{n}_K u_h v_h ds \\
&+ \sum_{K \in \xi} \int_{\partial K_t^+ \setminus \partial \Omega} \vec{\beta} \cdot \vec{n}_K u_h (v_h - v_h^{out}) ds. \tag{4.11a}
\end{aligned}$$

Note that the bilinear form $a_h(t; u, v)$ in (4.10a) is well-defined for the functions $u, v \in H_0^1(\Omega)$, and equal to

$$a_h(t; u, v) = \int_{\Omega} (\epsilon \nabla u \cdot \nabla v + \vec{\beta} \cdot \nabla uv) dx.$$

Thus, the continuous weak formulation (4.5) can be rewritten for any $t \in (0, T]$ as

$$\int_{\Omega} \frac{\partial u}{\partial t} v dx + a_h(t; u, v) + b(t; u, v) = l(v), \quad \forall v \in H_0^1(\Omega). \tag{4.12}$$

Proposition 4.2.1 (Existence of Unique Solution). *The SIPG semi-discrete system (4.9) has a unique solution in $C^{0,1}(0, T; V_h(\xi))$.*

Proof. Using the matrix-vector notations and construction introduced in Section 3.1.1, we reach the system of non-linear equations of (4.9) in matrix-vector form by

$$M\mathbf{U}_t + S\mathbf{U} = \mathbf{L} - \mathbf{b}(\mathbf{U}), \tag{4.13}$$

where \mathbf{U} is the vector of unknown coefficients, the matrix S is the stiffness matrix related to the coercive (Lemma 2.5.3) bilinear form $a_h(t; u_h, v_h) + K_h(u_h, v_h)$, and the Lipschitz (Lemma 3.1.1) vector function $\mathbf{b}(\mathbf{U})$ and the vector \mathbf{L} corresponds to the non-linear form $b_h(t; u_h, v_h)$ and the linear form $l_h(v_h)$, respectively. The matrix M is the usual mass matrix which by DG construction has a symmetric block diagonal structure, and therefore it is a symmetric positive definite matrix. As a consequent, in the algebraic point of view, what we have in the ODE system (4.13) is that M is an invertible matrix, S is a positive definite matrix and the right hand side is Lipschitz with respect to \mathbf{U} , which means by the theory of ordinary differential equations that the system (4.13), as a result, the semi-discrete problem (4.9) has a unique solution. \square

4.2.2 Fully Discrete Formulation

In this thesis, to integrate the non-stationary models in time, we use the backward Euler method which is an unconditionally stable integrator and convenient for stiff ODEs as it is the case in our model problem. Hence, we consider a subdivision of $[0, T]$ into n time intervals $I_k = (t^{k-1}, t^k]$ of length Δt_k , $k = 1, 2, \dots, n$. Set $t^0 = 0$ and for $n \geq 1$, $t^k = \Delta t_1 + \Delta t_2 + \dots + \Delta t_k$. Denote by ξ^0 an initial triangulation and by ξ^k the mesh associated to the k^{th} time step for $k > 0$, which is obtained from ξ^{k-1} possibly by locally refining/coarsening. Moreover, we assign the finite element space $V_h^k = V_h(\xi^k)$ to each mesh ξ^k . Then, backward Euler in time the fully discrete formulation of (4.9) reads as: for $t = 0$ set $u_h^0 \in V_h^0$ as the projection (orthogonal L^2 -projection) of u_0 onto V_h^0 and for $k = 1, 2, \dots, n$, find $u_h^k \in V_h^k$ satisfying for all $v_h^k \in V_h^k$

$$\int_{\Omega} \frac{u_h^k - u_h^{k-1}}{\Delta t_k} v_h^k dx + a_h(t^k; u_h^k, v_h^k) + K_h(u_h^k, v_h^k) + b_h(t^k; u_h^k, v_h^k) = \int_{\Omega} f^k v_h^k dx. \quad (4.14)$$

Remark 4.2.2. *A single system of (4.14) can be considered as a non-stationary semi-linear diffusion-convection-reaction equation of type (3.1) with the linear reaction coefficient $\alpha = 1/\Delta t_k > 0$ and with a modified right hand side. Hence, it is a direct consequent not only by the existence of unique solution result given for the semi-discrete system (4.9) but also by the ones given for stationary problems that for each $k = 1, 2, \dots, n$, the full-discrete system (4.14) has a unique solution.*

4.3 Time-Space Adaptivity for Non-Stationary Problems

The solution of the evolution problems modeled by the convection dominated diffusion-convection-reaction equations has a number of challenges. In one hand, one has to resolve the solution around the interior/boundary layers due to the convection domination. In the other hand, the nature of non-stationary model leads to the resolution of spatial layers to be more critical since the location of the layers may vary as time progresses, and it is mostly possible that there occur also temporal layers in addition to the spatial one. The native approach is the use of adaptive algorithms to resolve

the solution in an accurate and efficient way around the regions where the solution obeys large gradients. The key tool in adaptive algorithms, thus, is the way of locating the so-called regions, which is usually and naturally based on a posteriori error estimation. On the other hands, even for linear evolution problems, the results for a posteriori error estimation in evolution problems are limited. The analysis of existence studies are mostly based on energy techniques comparing the continuous and the discrete solution directly, and a posteriori bound driven there are optimal order in $L^2(H^1)$ -type norms. We refer to the studies in [1, 9, 20, 28, 67] and reference therein.

In this thesis, our approach is to derive energy norm based a posteriori estimates by not comparing the exact solution with the numerical solution directly, instead, by comparing the numerical solution with a constructed auxiliary solution whose difference to the numerical solution can be estimated by a posteriori estimate driven for stationary problem in Chapter 3. By the computational and theoretical point of view, this approach is more useful than the direct approaches which have a number of challenges. In order to be more clear, let consider a simple evolution problem

$$\begin{aligned} u'(t) + A(u(t)) &= f, & \text{in } \Omega \times (0, T], \\ u(\cdot, t) &= 0, & \text{on } \partial\Omega \times (0, T], \\ u(\cdot, 0) &= u_0, & \text{in } \Omega, \end{aligned}$$

leading to the weak formulations

$$\langle u'(t), v \rangle + a(u(t), v) = \langle f, v \rangle, \quad \forall v \in V, \text{ a.e. } t \in (0, T], \quad (4.15)$$

$$\langle U'(t), v \rangle + a(U(t), v) = \langle f, v \rangle, \quad \forall v \in V_h, \text{ a.e. } t \in (0, T] \quad (4.16)$$

with the exact and numerical solutions u and U , respectively, in the appropriate solution spaces. Then, the direct approach adaptive algorithms aim to derive an energy norm based a posteriori error indicator η to estimate the error between u and U by a suitable energy norm $\|\cdot\|_E$, i.e. find η satisfying

$$\|u - U\|_E \leq \eta(U).$$

The native way of deriving such a residual-based a posteriori error indicator is to estimate, using the Galerkin orthogonality property, the residual function R under the

given energy norm, and satisfying the variational formulation

$$\begin{aligned}\langle R_U, v \rangle &= \int_0^T (\langle f, v \rangle - \langle U'(t), v \rangle - a(U(t), v)) dt \\ &= \int_0^T (\langle f, v - I_h v \rangle - \langle U'(t), v - I_h v \rangle - a(U(t), v - I_h v)) dt.\end{aligned}$$

The above identity leads to the error equation

$$\frac{1}{2} \|u(T) - U(T)\|_{L^2(\Omega)}^2 + \int_0^T a(u - U, u - U) dt = \frac{1}{2} \|u(0) - U(0)\|_{L^2(\Omega)}^2 + \langle R_U, v \rangle$$

in which it is well-known that the integral term is of sub-optimal in $L^\infty(L^2)$ -type norms. On the other hands, the residual function R_U results in a number of challenges:

- It is possible that $\langle R_U, v \rangle$ is not well-defined,
- The numerical solution U does not satisfy point-wise structural conditions of the true solution u .

The approach to handle the above drawbacks is to construct a continuous auxiliary solution \tilde{U} which is easily computable from the numerical solution U through a suitable operator, and it satisfies the following conditions

- The difference $U - \tilde{U}$ is computable and can be estimated by a known (elliptic) a posteriori error indicator,
- $\langle R_{\tilde{U}}, v \rangle$ is well-defined and can be estimated by a computable a posteriori error indicator,
- $\langle R_{\tilde{U}}, v \rangle$ satisfy the original PDE with a modified right hand side.

In this thesis, the tool to obtain such an auxiliary solution \tilde{U} is the *elliptic reconstruction* technique introduced in [63] to derive a posteriori error estimates for the linear semi-discrete problems. In what follows, the *elliptic reconstruction* is an operator $\mathcal{R} : V_h \mapsto V$ with $\mathcal{R}U = \tilde{U}$, which is stated in the following definition.

Definition 4.3.1. [63, Definition 2.1] For each $t \in (0, T]$, let $U \in V_h$ be the solution of the discrete system (4.16). The elliptic reconstruction $\tilde{U} = \mathcal{R}U \in V$ of U is defined as the solution of the stationary problem

$$a(\tilde{U}(t), v) = \langle g_h(t), v \rangle, \quad \forall v \in V, \text{ a.e. } t \in (0, T]$$

with the modified right hand side term

$$g_h := A_h U - f_h + f,$$

where $A_h : V_h \mapsto V_h$ is the discrete version of A satisfying

$$a(U, v) = \langle A_h U, v \rangle \quad \forall v \in V_h,$$

and f_h denote the L^2 -projection of f onto the space V_h .

In the sequel, we will use the *elliptic reconstruction* technique to derive a posteriori error indicators for semi-discrete system (4.9) and the fully-discrete system (4.14) by utilizing a posteriori error estimate (3.8) driven for stationary model in Chapter 3. This approach is studied in [25] for the non-stationary linear diffusion-convection-reaction equations. Here, we will utilize the results in [25] to modify a posteriori error estimates for the non-stationary semi-linear diffusion-convection-reaction equations using a posteriori error estimates we have derived for stationary models in Chapter 3.

4.3.1 A Posteriori Error Bounds for Semi-Discrete System

In order to measure the error for the semi-discrete problem, we use the $L^\infty(L^2) + L^2(H^1)$ -type norm

$$\|v\|_*^2 = \|v\|_{L^\infty(0,T;L^2(\Omega))}^2 + \int_0^T \| |v| \|^2 dt,$$

and we also take into account the DG-norm

$$\|v\|_{DG} = \| |v| \| + |v|_C,$$

where the energy norm $\| |v| \|$ and the semi-norm $| \cdot |_C$ are defined as in (3.10) and (3.11) in Chapter 3, respectively.

We make use the elliptic reconstruction technique in [63]. In the view of the continuous semi-discrete problem (4.5), for each $t \in (0, T]$, the elliptic reconstruction $w \in H_0^1(\Omega)$ is the unique solution of the problem

$$a(t; w, v) + b(t; w, v) = \int_\Omega \left(f - \frac{\partial u_h}{\partial t} \right) v dx, \quad \forall v \in H_0^1(\Omega). \quad (4.17)$$

The SIPG discretization, on the other hand, of the above system reads as: for each $t \in (0, T]$, find $w_h \in C^{0,1}(0, T; V_h(\xi))$ satisfying for all $v_h \in V_h(\xi)$

$$a_h(t; w_h, v_h) + K_h(w_h, v_h) + b_h(t; w_h, v_h) = \int_{\Omega} \left(f - \frac{\partial u_h}{\partial t} \right) v_h dx,$$

which implies according to (4.9) that $w_h = u_h$. Hence, the error bound to the term $\|w - u_h\|_{DG}$ can be found using the ready a posteriori error bound (3.12) for non-linear stationary problem.

Most of the steps of the construction of a posteriori error bounds for the semi-discrete system is analogous to the ones in [25]. For the error $e(t) = u(t) - u_h(t)$ of the semi-discrete problem, we set the decomposition $e(t) = \mu(t) + \nu(t)$ with

$$\mu(t) = u(t) - w(t), \quad \nu = w(t) - u_h(t).$$

Further, as in the stationary case, we also decompose the SIPG solution $u_h(t) \in V_h$ for each $t \in (0, T]$ as

$$u_h(t) = u_h^c(t) + u_h^r(t)$$

with $u_h^c(t) \in H_0^1(\Omega) \cap V_h$ being the conforming part of the solution and $u_h^r \in V_h$ is the remainder term. By this constructions, we will have the following conforming error definitions

$$e^c(t) = u(t) - u_h^c(t), \quad \nu^c(t) = w(t) - u_h^c(t) \quad (4.18)$$

Theorem 4.3.2. *For the time-dependent error $e = u - u_h$ of the semi-discrete system (4.9), there holds*

$$\|e\|_* \lesssim \tilde{\eta}, \quad (4.19)$$

where the error estimator $\tilde{\eta}$ is defined by

$$\tilde{\eta}^2 = \|e(0)\|^2 + \int_0^T \tilde{\eta}_{S_1}^2 dt + \min \left\{ \left(\int_0^T \tilde{\eta}_{S_2}^2 dt \right)^2, \rho_T^2 \int_0^T \tilde{\eta}_{S_2}^2 dt \right\} + \max_{0 \leq t \leq T} \tilde{\eta}_{S_3}^2,$$

with

$$\begin{aligned}
\tilde{\eta}_{S_1}^2 &= \sum_{K \in \xi} \rho_K^2 \left\| f - \frac{\partial u_h}{\partial t} + \epsilon \Delta u_h - \vec{\beta} \cdot \nabla u_h - r(u_h) \right\|_{L^2(K)}^2 \\
&\quad + \sum_{e \in \Gamma_h^0} \epsilon^{-1/2} \rho_e \|\epsilon \nabla u_h\|_{L^2(e)}^2 + \sum_{e \in \Gamma_h} \left(\frac{\epsilon \sigma}{h_e} + \alpha_0 h_e + \frac{h_e}{\epsilon} \right) \|[u_h]\|_{L^2(e)}^2, \\
\tilde{\eta}_{S_2}^2 &= \sum_{e \in \Gamma_h} h_e \left\| \left[\frac{\partial u_h}{\partial t} \right] \right\|_{L^2(e)}^2, \\
\tilde{\eta}_{S_3}^2 &= \sum_{e \in \Gamma_h} h_e \|[u_h]\|_{L^2(e)}^2,
\end{aligned}$$

and the weight $\rho_T = \min(\epsilon^{-\frac{1}{2}}, \alpha_0^{-\frac{1}{2}})$. Note that the indicator $\tilde{\eta}_{S_1}$ is the same as the error indicator η in (3.7) for the elliptic model by the only differences that we have a modified right hand side and no linear reaction term.

Proof. For any $t \in (0, T]$, let $u = u(t)$ and $u_h = u_h(t)$ be the exact solution and SIPG semi-discrete approximation of (4.4), respectively. For any $v \in H_0^1(\Omega)$, the equations (4.12) and (4.17) reads

$$\int_{\Omega} \frac{\partial u}{\partial t} v dx + a_h(t; u, v) + \int_{\Omega} r(u) v dx = \int_{\Omega} f v dx, \quad (4.20)$$

$$a_h(t; w, v) + \int_{\Omega} r(w) v dx = \int_{\Omega} \left(f - \frac{\partial u_h}{\partial t} \right) v dx. \quad (4.21)$$

Subtracting (4.21) from (4.20), we obtain

$$\int_{\Omega} \frac{\partial e}{\partial t} v dx + a_h(t; \mu, v) + \int_{\Omega} (r(u) - r(w)) v dx = 0 \quad (4.22)$$

Choosing $v = e^c \in H_0^1(\Omega)$ in (4.22) and using the error definitions (4.18), we get

$$\int_{\Omega} \frac{\partial e^c}{\partial t} e^c dx + a_h(t; e^c, e^c) + \int_{\Omega} (r(u) - r(w)) e^c dx = \int_{\Omega} \frac{\partial u_h^c}{\partial t} e^c dx + a_h(t; \nu^c, e^c) \quad (4.23)$$

In (4.23), using the Young's inequality, and imposing the coercivity and continuity facts (4.7) and (4.8), respectively, we arrive at

$$\frac{d}{dt} \|e^c\|^2 + \|e^c\|^2 + \int_{\Omega} (r(u) - r(w)) e^c dx \lesssim \|\nu^c\|_{DG} + \left\| \frac{\partial u_h^c}{\partial t} \right\| \|e^c\|. \quad (4.24)$$

Adding and subtracting the term $r(u_h^c)$ into the integral term related to the non-linear term in (4.24), we obtain

$$\begin{aligned} \frac{d}{dt} \|e^c\|^2 + \|e^c\|^2 + \int_{\Omega} (r(u) - r(u_h^c)) e^c dx &\lesssim \|\nu^c\|_{DG} + \left\| \frac{\partial u_h^c}{\partial t} \right\| \|e^c\| \\ &+ \int_{\Omega} (r(w) - r(u_h^c)) e^c dx. \end{aligned} \quad (4.25)$$

Now, consider the integral terms related to the non-linear term in (4.25). In one hand, using the Cauchy-Schwarz's and Young's inequalities, and the local Lipschitz condition (4.2b), we have

$$\begin{aligned} \int_{\Omega} (r(w) - r(u_h^c)) e^c dx &\leq L(S) \|\nu^c\| \|e^c\| \\ &\lesssim \|\nu^c\|_{DG}^2. \end{aligned} \quad (4.26)$$

On the other hand, using the Cauchy-Schwarz's and Young's inequalities, and the conditions (4.2a) and (4.2c), we obtain

$$\int_{\Omega} (r(u) - r(u_h^c)) e^c dx \gtrsim \|e^c\|^2. \quad (4.27)$$

Thus, combining the identities (4.26) and (4.27), the inequality (4.25) become

$$\frac{d}{dt} \|e^c\|^2 + \|e^c\|^2 \lesssim \|\nu^c\|_{DG} + \left\| \frac{\partial u_h^c}{\partial t} \right\| \|e^c\|. \quad (4.28)$$

Finally, the error bound (4.19) easily follows from (4.28) and [25, Theorem 5.4]. \square

4.3.2 A Posteriori Error Bounds for Fully Discrete System

For the fully-discrete system (4.14), we consider the solutions at discrete time instances. For this reason, for any $v_h^k \in V_h^k$, let $A^k \in V_h^k$ be the unique solution of the stationary system

$$a_h(t^k; u_h^k, v_h^k) + K_h(u_h^k, v_h^k) + b_h(t^k; u_h^k, v_h^k) = \int_{\Omega} A^k v_h^k dx. \quad (4.29)$$

Note that for $k \geq 1$, $A^k = I_h^k f^k - (u_h^k - I_h^k u_h^{k-1}) / \Delta t_k$ with I_h^k being the L^2 -projection operator onto the space V_h^k . Then, the elliptic reconstruction $w^k \in H_0^1(\Omega)$ is defined as the unique solution of the stationary problem

$$a_h(t^k; w^k, v) + b(t^k; w^k, v) = \int_{\Omega} A^k v dx, \quad \forall v \in H_0^1(\Omega). \quad (4.30)$$

Next, we define the time-dependent solution $u_h(t) \in V_h^k \cup V_h^{k+1}$ as a piecewise continuous function so that on each interval $(t^{k-1}, t^k]$, $u_h(t)$ is the linear interpolation of the values u_h^{k-1} and u_h^k given by

$$u_h(t) = l_{k-1}(t)u_h^{k-1} + l_k(t)u_h^k \quad (4.31)$$

with the linear Lagrange interpolation basis l_{k-1} and l_k are defined on $[t^{k-1}, t^k]$. Further, we use the decomposition of each discrete solution $u_h^k = u_h^{k,c} + u_h^{k,r}$ with $u_h^{k,c} \in H_0^1(\Omega)$ and $u_h^{k,r} \in V_h^k$, as well as the error definitions $e^c = u - u_h^c$ and $\nu^k = w^k - u_h^k$, where, on an arbitrary interval $(t^{k-1}, t^k]$, we have

$$u_h^c(t) = l_{k-1}(t)u_h^{k-1,c} + l_k(t)u_h^{k,c}.$$

Theorem 4.3.3. *Through the definition (4.31), for the in time continuous error $e = u - u_h$ of the fully-discrete system (4.14), there holds*

$$\|e\|_*^2 \lesssim \eta_S^2 + \eta_T^2, \quad (4.32)$$

where the spatial estimator η_S is given by [25]

$$\begin{aligned} \eta_S^2 &= \|e(0)\|^2 + \frac{1}{3} \sum_{k=1}^n \Delta t_k (\eta_{S_{1,k-1}}^2 + \eta_{S_{1,k}}^2) + \sum_{k=1}^n \Delta t_k \eta_{S_{2,k}}^2 + \max_{0 \leq k \leq n} \eta_{S_{3,k}}^2 \\ &\quad + \min \left\{ \left(\sum_{k=1}^n \Delta t_k \eta_{S_{4,k}} \right)^2, \rho_T^2 \sum_{k=1}^n \Delta t_k \eta_{S_{4,k}}^2 \right\} \end{aligned} \quad (4.33)$$

with

$$\begin{aligned} \eta_{S_{1,k}}^2 &= \sum_{K \in \xi^{k-1} \cup \xi^k} \rho_K^2 \left\| A^k + \epsilon \Delta u_h^k - \vec{\beta}^k \cdot \nabla u_h^k - r(u_h^k) \right\|_{L^2(K)}^2 \\ &\quad + \sum_{e \in \Gamma_h^0} \epsilon^{-1/2} \rho_e \left\| [\epsilon \nabla u_h^k] \right\|_{L^2(e)}^2 + \sum_{e \in \Gamma_h} \left(\frac{\epsilon \sigma}{h_e} + \alpha_0 h_e + \frac{h_e}{\epsilon} \right) \left\| [u_h^k] \right\|_{L^2(e)}^2, \\ \eta_{S_{2,k}}^2 &= \sum_{K \in \xi^{k-1} \cup \xi^k} \rho_K^2 \left\| f^k - I_h^k f^k + \frac{u_h^{k-1} - I_h^k u_h^{k-1}}{\Delta t_k} \right\|_{L^2(K)}^2, \\ \eta_{S_{3,k}}^2 &= \sum_{e \in \Gamma_h} h_e \left\| [u_h^k] \right\|_{L^2(e)}^2, \\ \eta_{S_{4,k}}^2 &= \sum_{e \in \Gamma_h} h_e \left\| \left[\frac{u_h^k - u_h^{k-1}}{\Delta t_k} \right] \right\|_{L^2(e)}^2, \end{aligned}$$

and the temporal estimator η_T is given by [25]

$$\begin{aligned} \eta_T^2 &= \sum_{k=1}^n \int_{t_{k-1}}^{t_k} \eta_{T_{1,k}}^2 dt \\ &\quad + \min \left\{ \left(\sum_{k=1}^n \int_{t_{k-1}}^{t_k} \eta_{T_{2,k}} dt \right)^2, \rho_T^2 \sum_{k=1}^n \int_{t_{k-1}}^{t_k} \eta_{T_{2,k}}^2 dt \right\} \end{aligned} \quad (4.34)$$

with

$$\begin{aligned} \eta_{T_{1,k}}^2 &= \sum_{K \in \xi^{k-1} \cup \xi^k} \epsilon^{-1} \|l_k(\vec{\beta}^k - \vec{\beta})u_h^k + l_{k-1}(\vec{\beta}^{k-1} - \vec{\beta})u_h^{k-1}\|_{L^2(K)}^2, \\ \eta_{T_{2,k}}^2 &= \sum_{K \in \xi^{k-1} \cup \xi^k} \|f - f^k + l_{k-1}(A^k - A^{k-1}) + l_k(\nabla \cdot \vec{\beta}^k - \nabla \cdot \vec{\beta})u_h^k \\ &\quad + l_k(\nabla \cdot \vec{\beta}^{j-1} - \nabla \cdot \vec{\beta})u_h^{k-1}\|_{L^2(K)}^2. \end{aligned}$$

Proof. On an arbitrary interval $(t^{k-1}, t^k]$, let $e = e(t) = u(t) - u_h(t)$ represents the error of the fully-discrete system (4.14), where the approximate solution $u_h(t)$ is formulated as in (4.31). Then, the use of the systems (4.12) and (4.30) yields for all $v \in H_0^1(\Omega)$ the error equation

$$\begin{aligned} \int_{\Omega} \frac{\partial e}{\partial t} v dx + a_h(t; e, v) + \int_{\Omega} (r(u) - r(u_h)) v dx &= \int_{\Omega} (f - f^k) v dx \\ &\quad + \int_{\Omega} \left(f^k - \frac{\partial u_h}{\partial t} \right) v dx - a_h(t; u_h, v) - \int_{\Omega} r(u_h) v dx. \end{aligned} \quad (4.35)$$

After choosing $v = e^c$ in (4.35) and utilizing the conditions (4.2a)-(4.2c) related to the non-linear term, as it done in the proof of the semi-discrete case in Section 4.3.1, the error bound (4.32) easily follows from [25, Theorem 6.5]. \square

4.3.3 Adaptive Algorithm

Our time-space adaptive algorithm (see Fig. 4.1) for the non-stationary semi-linear model of type (1.1) based on the residual-based a posteriori estimations derived above, and utilizes the residual-based a posteriori error estimators derived in Chapter 3. The algorithm starts with an initial uniform mesh in space and with a given initial sufficiently large time step. At each time step, the space and time-step are adaptively arranged according to the user defined tolerances `ttol` for time-step refinement, and

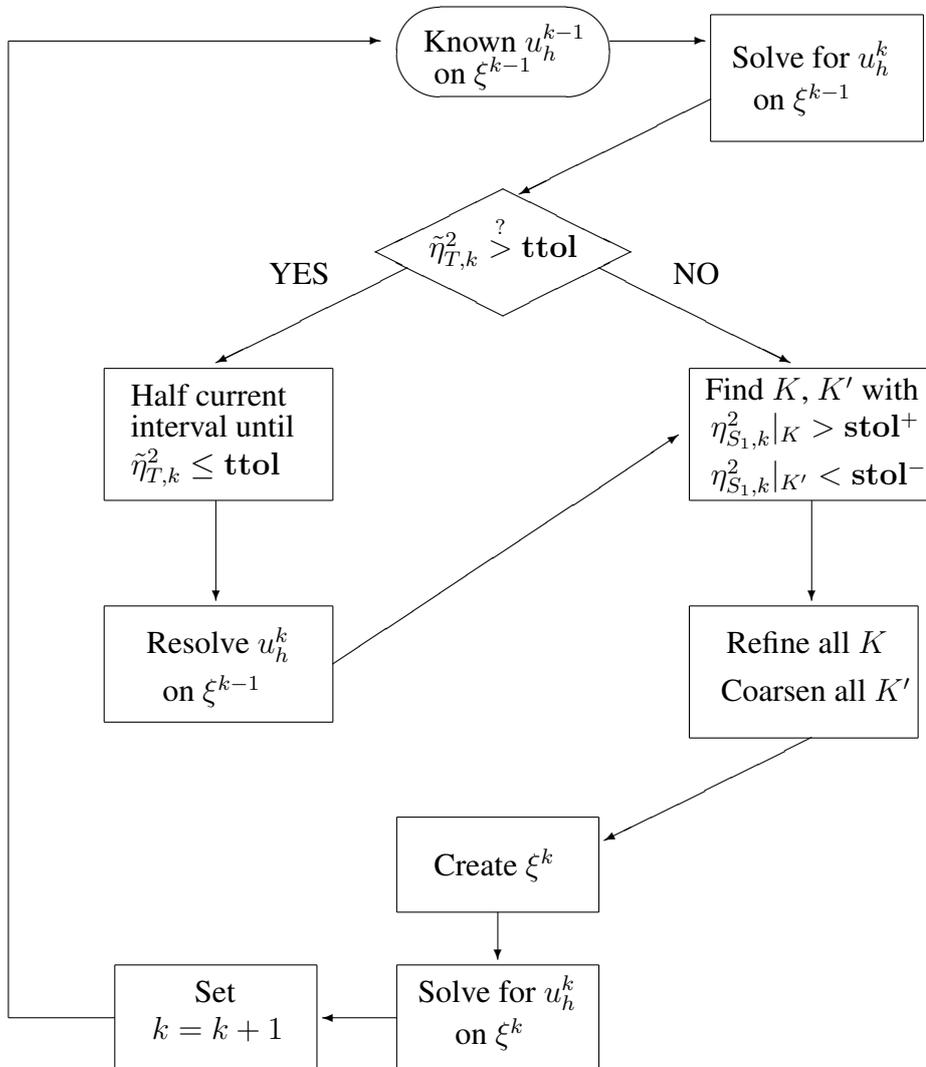


Figure 4.1: Adaptive algorithm chart on a single time step $(t_{k-1}, t_k]$

stol^+ and stol^- for spatial mesh, former corresponding to refinement and latter corresponding to the coarsening procedures in space. Note that we do not need a temporal tolerance corresponding to the time-step coarsening, since we start, in our problems, with a uniform equispaced distribution of $[0, T]$ having sufficiently large time-steps. Thus, it is enough just bisecting the time intervals not satisfying the temporal tolerance ttol , and keep unchanged the remaining. Both the refinement and coarsening processes in space are determined by the indicator $\eta_{S_{1,k}}$ appearing in the spatial estimator (4.33), which is indeed analogue to the elliptic indicator (3.7). Since the temporal estimator η_T (4.34) is not easy to compute, the adaptive refinement of the time-steps are driven by the modified temporal-indicator [25]

$$\tilde{\eta}_{T_k}^2 = \int_{t_{k-1}}^{t_k} \eta_{T_{1,k}}^2 dt + \min\{\rho_T, T\} \int_{t_{k-1}}^{t_k} \eta_{T_{2,k}}^2 dt$$

sum of which gives a bound for the temporal estimator η_T^2 .

Although the adaptive algorithm, Fig. 4.1, stands for a single equation of the system (4.4), it is not difficult to extend the algorithm to the coupled systems. For this, say we have a system of two equations for the unknowns u_1 and u_2 , the temporal-indicators $\tilde{\eta}_{T_k}^1, \tilde{\eta}_{T_k}^2$ and the spatial-indicators $\eta_{S_{1,k}}^1, \eta_{S_{1,k}}^2$ corresponding to the unknowns u_1 and u_2 , respectively, are computed. To adapt the time-step size, we ask the temporal condition for the both temporal-indicators, i.e. $\tilde{\eta}_{T_k}^1 \leq \text{ttol}$ and $\tilde{\eta}_{T_k}^2 \leq \text{ttol}$. On the other hand, to select the elements to be refined, we take the set of elements which is the union of the sets of the elements satisfying $\eta_{S_{1,k}}^1 > \text{stol}^+$ and $\eta_{S_{1,k}}^2 > \text{stol}^+$, and similar procedure to select the elements to be coarsened, but not including any elements which are selected to be refined. Numerical studies demonstrate that the adaptive algorithm is capable of resolving the layers in space as the time progresses.

4.4 Solution of Fully Discrete System

In this section, we discuss the solution of the fully-discrete system (4.14) on an arbitrary k^{th} time-step, which is solved for all $k = 1, 2, \dots, n$. In order to not be confused about the notations, let us consider the system (4.14) on an arbitrary k^{th}

time-step without the superscript for the time-step of the form

$$\int_{\Omega} \frac{u_h - w_h}{\Delta t} v_h dx + a_h(u_h, v_h) + K_h(u_h, v_h) + b_h(u_h, v_h) = \int_{\Omega} f v_h dx, \quad \forall v_h \in V_h \quad (4.36)$$

where we have set $u_h := u_h^k$, $w_h := u_h^{k-1}$, $v_h := v_h^k$, $f := f^k$, $\Delta t := \Delta t_k$, $a_h(u_h, v_h) := a_h(t^k; u_h^k, v_h^k)$, $K_h(u_h, v_h) := K_h(u_h^k, v_h^k)$, $b_h(u_h, v_h) := b_h(t^k; u_h^k, v_h^k)$ and $V_h := V_h^k$. The approximate solution u_h and the known solution (from the previous time-step) w_h of (4.36) have the form

$$u_h = \sum_{i=1}^{Nel} \sum_{l=1}^{Nloc} u_l^i \phi_l^i, \quad w_h = \sum_{i=1}^{Nel} \sum_{l=1}^{Nloc} w_l^i \phi_l^i, \quad (4.37)$$

where ϕ_l^i 's are the basis polynomials spanning the space V_h , $\mathbf{U} = \{u_l^i\}$ is the vector of unknown coefficients to be found and $\mathbf{W} = \{w_l^i\}$ is the vector of known coefficients. Using the notations and definitions in Section 3.1.1, the discrete residual of the system (4.36) in matrix vector form is given by

$$R(\mathbf{U}) = M\mathbf{U} - M\mathbf{W} + \Delta t(S\mathbf{U} + \mathbf{b}(\mathbf{U}) - \mathbf{L}) = 0, \quad (4.38)$$

where M is the mass matrix, S is the stiffness matrix corresponding to the bilinear form $a_h(u_h, v_h) + K_h(u_h, v_h)$, $\mathbf{b}(\mathbf{U})$ is the vector function of \mathbf{U} related to the non-linear form $b_h(u_h, v_h)$ and \mathbf{L} is the vector to the linear form $l_h(v_h)$.

Next, we solve the system (4.38) by Newton method. In the sequel, we start with an initial guess $\mathbf{U}^{(0)}$ (most possibly $\mathbf{U}^{(0)} = \mathbf{W}$, i.e. the known solution from the previous time-step) and for $i = 0, 1, 2, \dots$, we solve the system

$$\begin{aligned} J^i \delta \mathbf{U}^{(i)} &= -R(\mathbf{U}^{(i)}) \\ \mathbf{U}^{(i+1)} &= \mathbf{U}^{(i)} + \delta \mathbf{U}^{(i)} \end{aligned} \quad (4.39)$$

until a prescribed tolerance is satisfied. In (4.39), the sparse matrix $J^i = M + \Delta t(S + J_b^i)$ denotes the value of the Jacobian matrix of the residual function $R(\mathbf{U})$ at the current iterate $\mathbf{U}^{(i)}$, and J_b^s stands for the Jacobian matrix to the vector function $\mathbf{b}(\mathbf{U})$ at the current iterate $\mathbf{U}^{(i)}$.

4.5 Numerical Examples

In this section, we give the numerical studies demonstrating the performance of the time-space adaptive algorithm. All the computations are implemented on MATLAB-R2014a. In the problems, by the very coarse initial mesh, we mean an initial mesh which is formed, for instance on $\Omega = (0, 1)^2$, by dividing the region with $\Delta x_1 = \Delta x_2 = 0.5$ leading to 8 triangular elements and 48 DoFs for quadratic elements. As the first example, we give a test example with polynomial type non-linearity having a non-moving internal layer to figure out the benchmark of the algorithm by looking at the rates of error, spatial and temporal estimators, and effectivity indices (proportion of the estimator to the error) for different tolerances and diffusion parameters ϵ . We expect that the effectivity indices lie in a small band for different diffusion parameters meaning that our estimators are robust in the system Péclet number. Moreover, to demonstrate the mentioned properties, we use the average weighted DoFs

$$\text{Average Weighted DoFs} = \frac{1}{T} \sum_{k=1}^n \Delta t_k \lambda_k,$$

where λ_k denotes the total number of DoFs on the union mesh $\xi^{k-1} \cup \xi^k$. Since the first example has a non-moving internal layer, a monotonic increase in the DoFs is expected by the time progresses. Conversely, we give problems having moving layers by the time progresses in Example 4.5.2-4.5.3. In this case, we expect that the refinement and coarsening procedures in space work simultaneously leading to oscillations in time vs DoFs plots. By Example 4.5.2, we also test the performance of our algorithm for a coupled system. As the final example, Example 4.5.4, we consider a real geoscience problem representing a reaction in porous media having internal layers at different locations due to the high-permeability rocks.

4.5.1 Example with Polynomial Type Non-Linearity (Benchmark of the Algorithm)

The first example is taken from [16] with a polynomial non-linear term

$$u_t + \vec{\beta} \cdot \nabla u - \epsilon \Delta u + u^4 = f \quad \text{in } \Omega = (0, 1)^2$$

with the convection field $\vec{\beta} = (2, 3)^T$ and the diffusion coefficient $\epsilon = 10^{-6}$. The source function f and the Dirichlet boundary condition are chosen so that the exact solution is given by

$$u(\vec{x}, t) = 16 \sin(\pi t) x_1 (1 - x_1) x_2 (1 - x_2) [0.5 + \pi^{-1} \arctan(2\epsilon^{-1/2}(0.25^2 - (x_1 - 0.5)^2 - (x_2 - 0.5)^2))].$$

We start by demonstrating the decrease of the errors by uniform time-space refinement using linear DG elements. In Fig. 4.2, the expected first order convergence in space and time is shown.

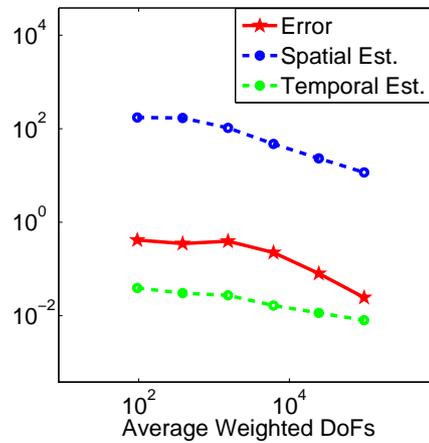


Figure 4.2: Example 4.5.1: Decays of estimators and errors for uniform time-space

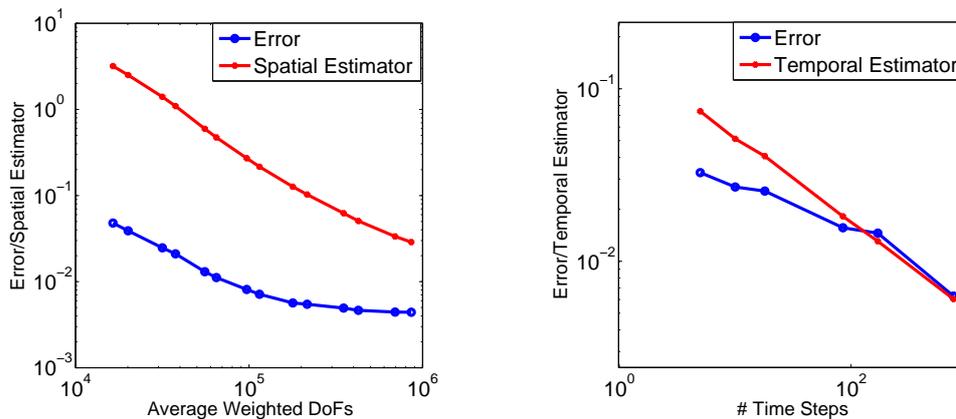


Figure 4.3: Example 4.5.1: Error vs. spatial (left) and temporal (right) estimators for $\epsilon = 10^{-6}$

In the sequel, we use quadratic DG elements. We investigate the performance of the spatial estimator by fixing the temporal time-step $\Delta t = 0.002$ so that the temporal error is dominated by the spatial error. Then, we reduce the spatial estimator tolerance stol^+ from 10^{-1} to 10^{-6} . The rate of the errors and the spatial estimators are similar as illustrated in Fig. 4.3, left, for $\epsilon = 10^{-6}$. Fig. 4.4 shows the spatial effectivity indices and the decrease of the spatial estimators for various diffusion constants ϵ . One can see that the effectivity indices converges asymptotically to a small band, as the results in [25] for linear problems, showing the robustness of the spatial estimator.

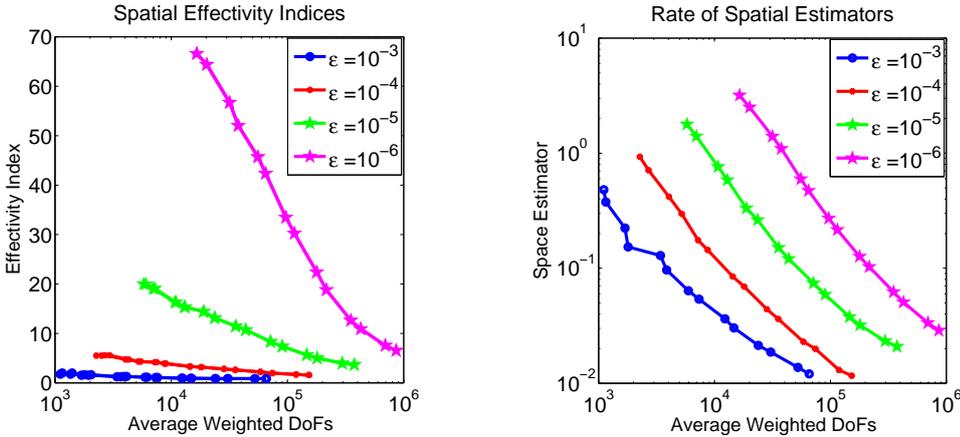


Figure 4.4: Example 4.5.1: Spatial effectivity indices (left) and estimators (right)

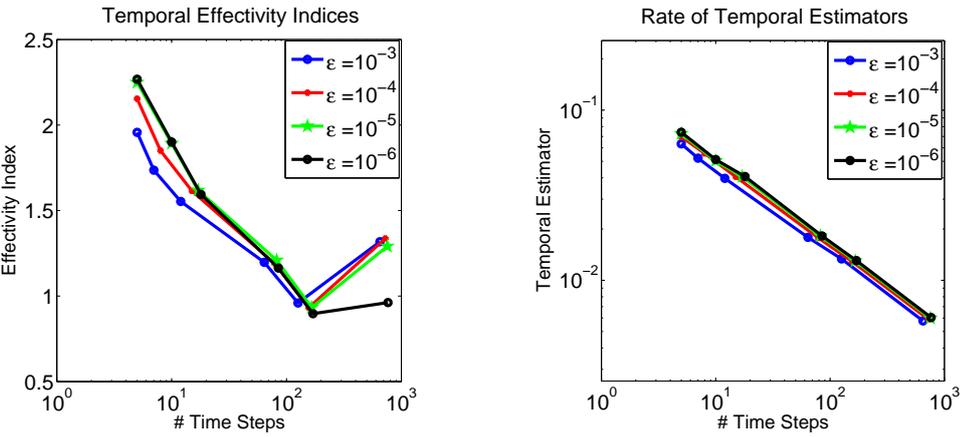


Figure 4.5: Example 4.5.1: Temporal effectivity indices (left) and estimators (right)

To investigate the performance of the temporal estimator, we fix a sufficiently fine spatial mesh so that the spatial error is dominated by the temporal error, and then we

reduce the temporal estimator tolerance \mathbf{ttol} in the range $10^{-1} - 10^{-6}$. In Fig. 4.5, the temporal effectivity indices and the decrease of the temporal estimators are not affected by ϵ , and effectivity indices are almost the same within the band 1-2, showing the robustness of the temporal estimator.

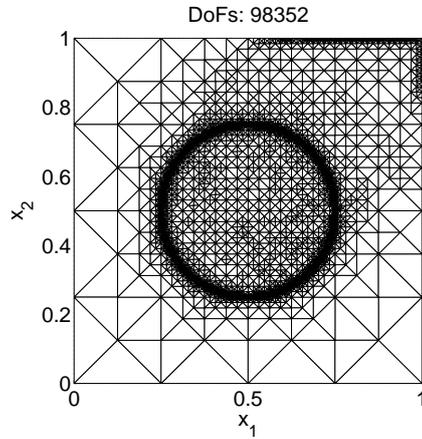


Figure 4.6: Example 4.5.1: Adaptive mesh

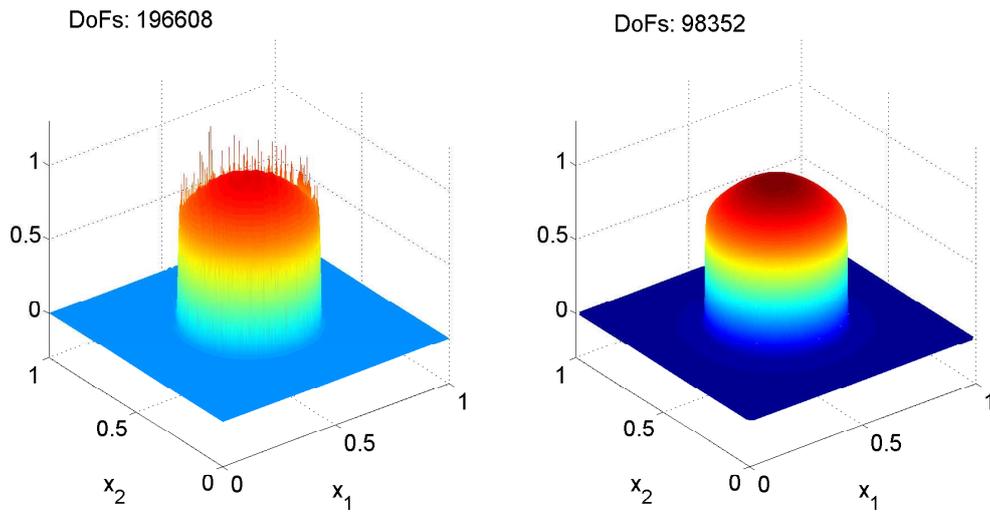


Figure 4.7: Example 4.5.1: Uniform (left) and adaptive (right) solutions at $T=0.5$

Finally, we apply the time-space adaptive algorithm with the tolerances $\mathbf{ttol} = 10^{-3}$, $\mathbf{stol}^+ = 3 \times 10^{-4}$ and $\mathbf{stol}^- = 3 \times 10^{-7}$. Firstly, we prepare an initial mesh starting from a very coarse spatial mesh and a uniform partition of the time interval $[0, 0.5]$ with the step-size $\Delta t = 0.25$ until the user defined tolerances \mathbf{ttol} and \mathbf{stol}^+ are satisfied. The adaptive mesh at the final time $T = 0.5$ is shown in Fig. 4.6. In Fig. 4.8

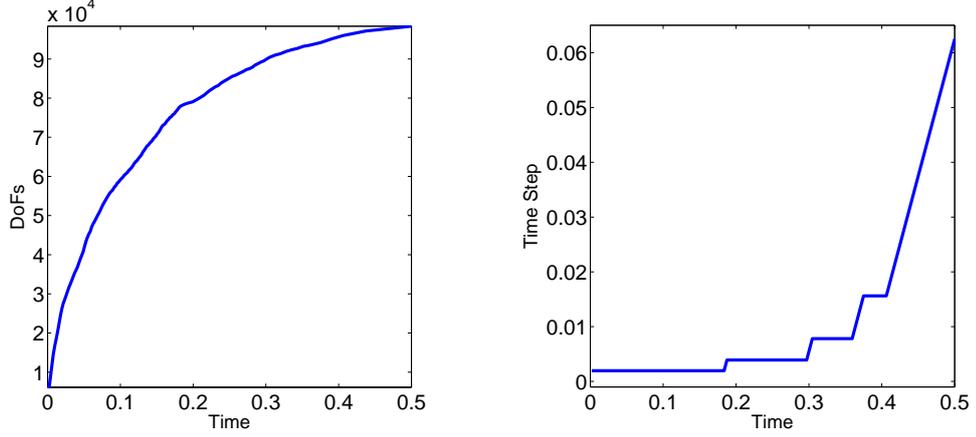


Figure 4.8: Example 4.5.1: Evolution of DoFs (left) and time-steps Δt (right)

on the right, the change of the time-steps is shown, whereas the change in the DoFs is illustrated in Fig. 4.8 on the left. Since the layers in the problem do not move as the time progresses, the number of DoFs increases monotonically by the spatial grid refinement. In Fig. 4.7, it is shown that all the oscillations are damped out by adaptive algorithm using less DoFs compared to the uniform one.

4.5.2 Coupled Example with Polynomial Type Non-Linearity

The next example is a coupled non-linear problem taken from [17].

$$\frac{\partial u_i}{\partial t} - \epsilon \Delta u_i + \vec{\beta}_i \cdot \nabla u_i + u_1 u_2 = f_i, \quad i = 1, 2$$

on $\Omega = (0, 1)^2$ with the convection fields $\vec{\beta}_1 = (1, 0)^T$ and $\vec{\beta}_2 = (-1, 0)^T$, and the diffusion constant $\epsilon = 10^{-5}$. The Dirichlet boundary conditions, initial conditions and the load functions f_i are chosen so that the exact solutions are

$$u_1(\vec{x}, t) = \frac{1}{2} \left(1 - \tanh \frac{2x_1 - 0.2t - 0.8}{\sqrt{5\epsilon}} \right),$$

$$u_2(\vec{x}, t) = \frac{1}{2} \left(1 + \tanh \frac{2x_1 + 0.2t - 0.9}{\sqrt{5\epsilon}} \right).$$

We use again quadratic DG elements. We prepare an initial mesh, Fig. 4.10 on the left, starting with a very coarse spatial mesh and a uniform partition of the time interval $[0, 1]$ with the step-size $\Delta t = 0.1$ until the user defined tolerances $\text{ttol} = 10^{-3}$

and $\text{stol}^+ = 10^{-1}$ are satisfied. Here, two sharp fronts move towards to each other and then mix directly after the time $t = 0.1$, Fig. 4.9. The movement of the fronts are also visible in Fig. 4.10 claiming that refinement/coarsening of the adaptive algorithm works well. We see that the sharp fronts in the cross-wind direction $x_2 = 0.5x_1 + 0.75$ are almost damped out. Moreover, Fig. 4.10-4.11 show that both the spatial and temporal estimators catch the time where the two sharp fronts mix.

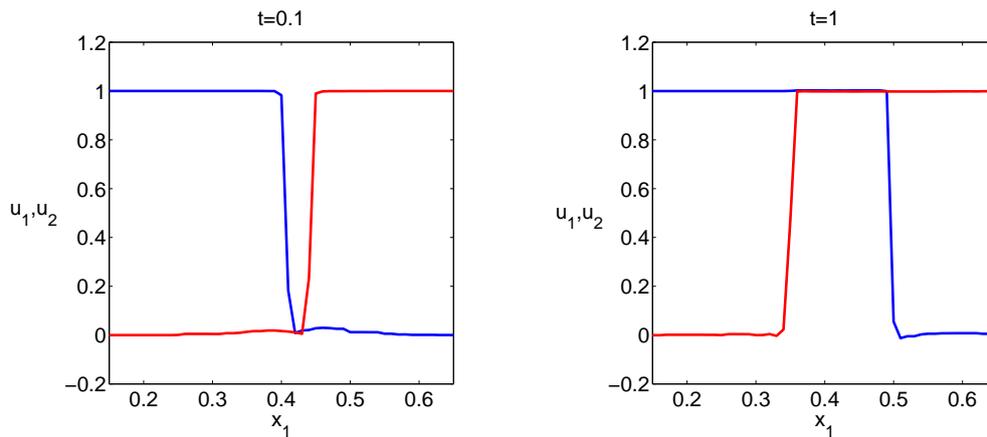


Figure 4.9: Example 4.5.2: Cross-section plots in the cross-wind direction at $t = 0.1$ (left) and $t = 1$ (right)

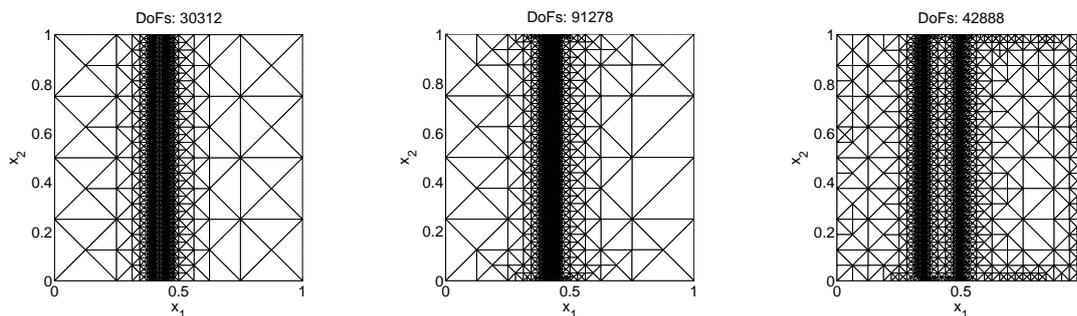


Figure 4.10: Example 4.5.2: Adaptive meshes at $t = 0$, $t = 0.1$ and $t = 1$ (from left to right)

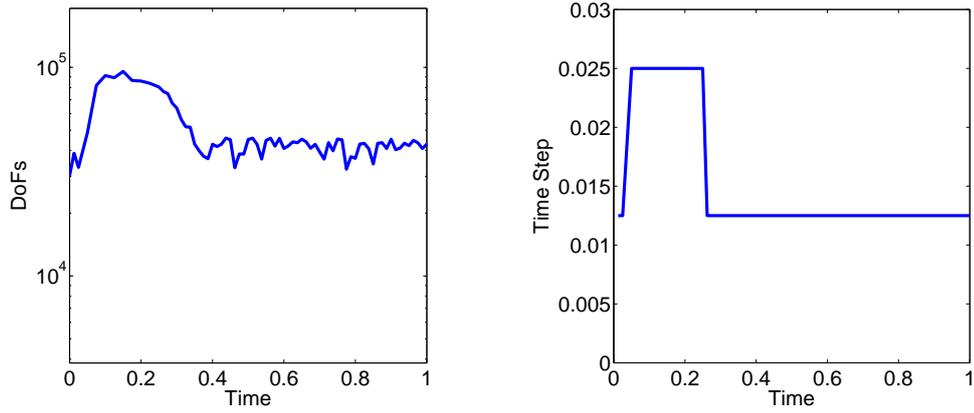


Figure 4.11: Example 4.5.2: Evolution of DoFs (left) and time-steps Δt (right)

4.5.3 A Non-linear ADR in Homogeneous Porous Media

We consider the advection-diffusion-reaction (ADR) equation in [79] with polynomial type non-linear reaction

$$\frac{\partial u}{\partial t} - \epsilon \Delta u + \vec{\beta} \cdot \nabla u + \gamma u^2(u - 1) = 0 \quad \text{in } \Omega \times (0, T]$$

with $\Omega = (0, 1)^2$. We take as in [79] the homogeneous dispersion tensor as $\epsilon = 10^{-4}$, the velocity field $\vec{\beta} = (-0.01, -0.01)^T$ and $\gamma = 100$. The initial and boundary conditions are chosen by the exact solution

$$u(\vec{x}, t) = [1 + \exp(a(x_1 + x_2 - bt) + a(b - 1))]^{-1},$$

where $a = \sqrt{\gamma/(4\epsilon)}$ and $b = -0.02 + \sqrt{\gamma\epsilon}$. The problem is a transport of a front in homogeneous porous media. We simulate the given problem for the final time $T = 1$, and with quadratic DG elements. We begin by preparing an initial mesh starting from a very coarse spatial mesh and a uniform partition of the time interval $[0, 1]$ with the step-size $\Delta t = 0.25$ until the user defined tolerances $\text{ttol} = 3 \times 10^{-3}$ and $\text{stol}^+ = 10^{-3}$ are satisfied. In Fig. 4.12, the adaptive meshes and solution profiles are shown at the times $t = \{0.2, 0.6, 1\}$, where the movement of the front can be seen. The time vs DoFs and time vs time step-size plots in Fig. 4.13 indicate clearly the oscillations in DoFs and time-steps due to the movement of the front.

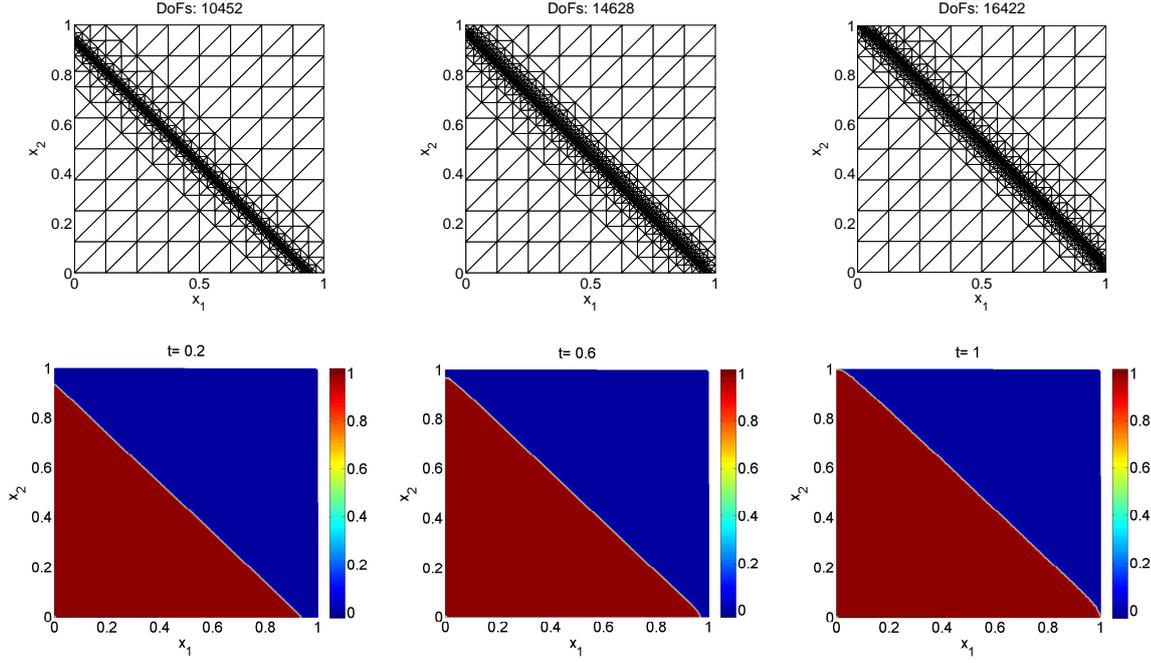


Figure 4.12: Example 4.5.3: Adaptive meshes (top) and solution profiles (bottom) at $t = 0.2$, $t = 0.6$ and $t = 1$ (from left to right)

4.5.4 A Non-linear ADR in Deterministic Heterogeneous Porous Media

As the final numeric, we consider the ADR equation in [79] with Monod or Langmuir isotherm type non-linear reaction

$$\begin{aligned}
 \frac{\partial u}{\partial t} - \nabla \cdot (\epsilon \nabla u) + \vec{\beta}(x) \cdot \nabla u + \frac{u}{1+u} &= 0 \quad \text{in } \Omega \times (0, T], \\
 u(x, t) &= 1 \quad \text{on } \Gamma^D \times [0, T], \\
 -\epsilon \nabla u(x, t) \cdot \vec{n} &= 0 \quad \text{on } (\partial\Omega \setminus \Gamma^D) \times [0, T], \\
 u(x, 0) &= 0 \quad \text{in } \Omega
 \end{aligned}$$

with $\Omega = (0, 3) \times (0, 2)$ and $\Gamma^D = \{0\} \times [0, 2]$. The problem represents a reaction in porous media, for instance, transport in a highly idealized fracture pattern. Here ϵ stands for the heterogeneous dispersion tensor given by

$$\epsilon = \begin{bmatrix} 10^{-3} & 0 \\ 0 & 10^{-4} \end{bmatrix}.$$

The velocity field $\vec{\beta}(x)$ is determined via the Darcy's law

$$\vec{\beta} = -\frac{k(x)}{\mu} \nabla p,$$

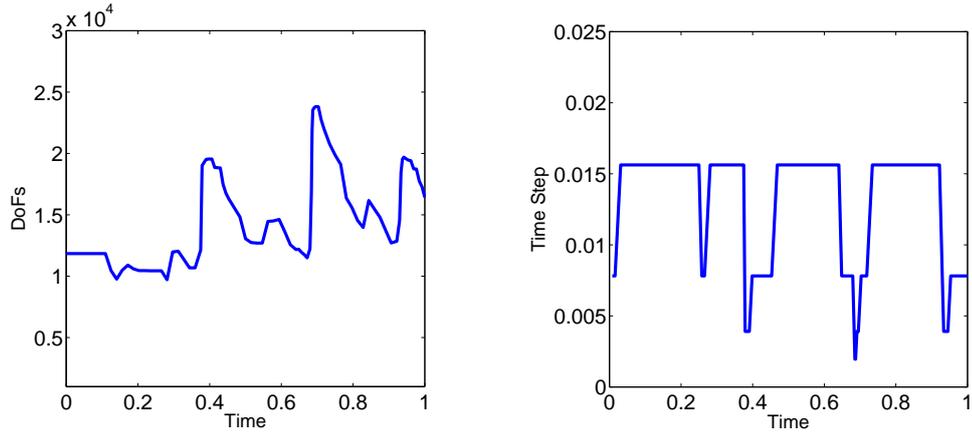


Figure 4.13: Example 4.5.3: Evolution of DoFs (left) and time-steps Δt (right)

where p is the fluid pressure, μ is the fluid viscosity and $k(x)$ is the permeability of the porous medium. Using the mass conservation property $\nabla \cdot \vec{\beta}(x) = 0$ under the assumption that rock and fluids are incompressible, the velocity field $\vec{\beta}(x)$ is computed by solving the system

$$\begin{aligned} \nabla \cdot \left(\frac{k(x)}{\mu} \nabla p \right) &= 0 \quad \text{in } \Omega, \\ p &= 1 \quad \text{on } \{0\} \times [0, 2], \\ p &= 0 \quad \text{on } \{3\} \times [0, 2], \\ -k(x) \nabla p \cdot \vec{n} &= 0 \quad \text{on } (0, 3) \times \{0, 2\}. \end{aligned}$$

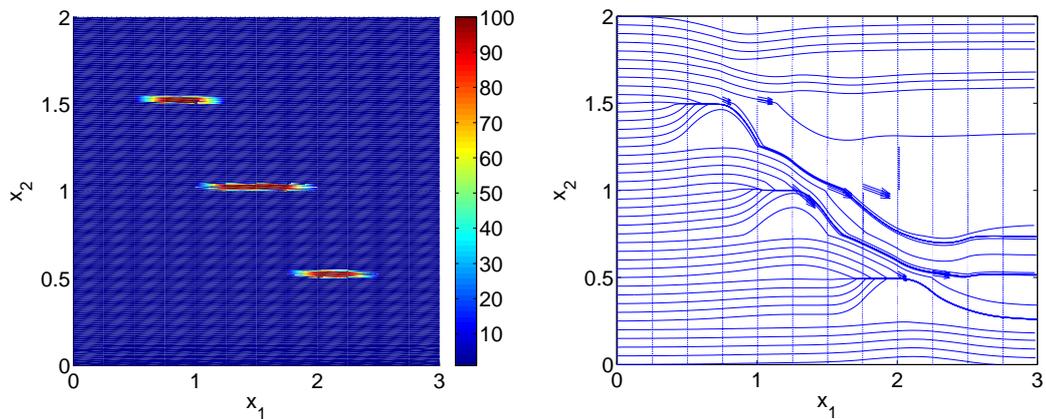


Figure 4.14: Example 4.5.4: Permeability field (left) and velocity streamlines (right)

We simulate the given problem for the final time $T = 1$ using linear DG elements. We take the fluid viscosity $\mu = 0.1$, and the permeability field as in [79] with three parallel streaks the permeability of which are 100 times greater than the permeability of the surrounding domain, see Fig. 4.14 on the left, by which the flow is canalized from the lower-permeability rocks into the high-permeability ones, Fig. 4.14 on the right. For the adaptive procedure, we prepare an initial mesh starting from a very coarse spatial mesh and a uniform partition of the time interval $[0, 1]$ with the step-size $\Delta t = 0.05$ until the user defined tolerances $\text{ttol} = 10^{-3}$ and $\text{stol}^+ = 3 \times 10^{-4}$ are satisfied. Fig. 4.16-4.17 show the adaptive meshes and concentrations at $t = 0.3$ and $t = 1$, where we can clearly see the flow-focusing due to the high-permeability.

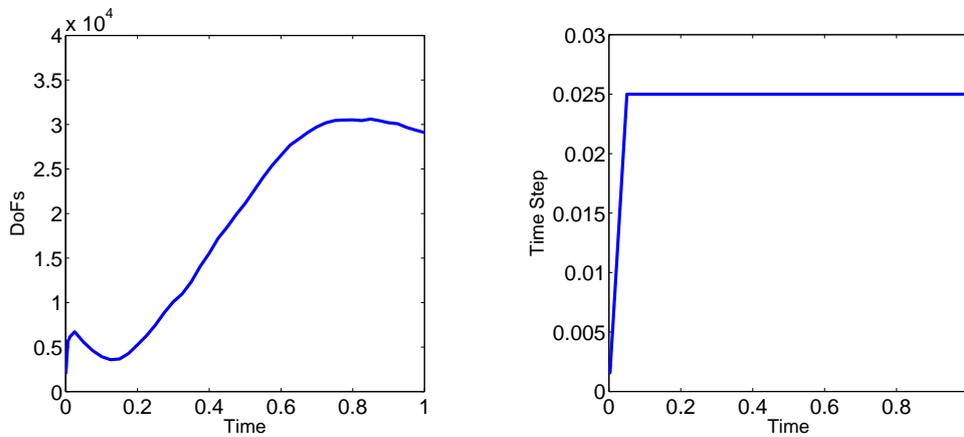


Figure 4.15: Example 4.5.4: Evolution of DoFs (left) and time-steps Δt (right)

Time vs DoFs and time vs time step-size plots are given in Fig. 4.15. We see that initially small time steps are used and then it reaches a steady time step, Fig. 4.15 on the right. The number of DoFs increases (refinement dominates coarsening) monotonically after the meet of first high-permeability rock until the meet of third high-permeability rock and then the increase stops, Fig. 4.15 on the left. This is meaningful since there is no sharp flow canalization after the third high-permeability rock.

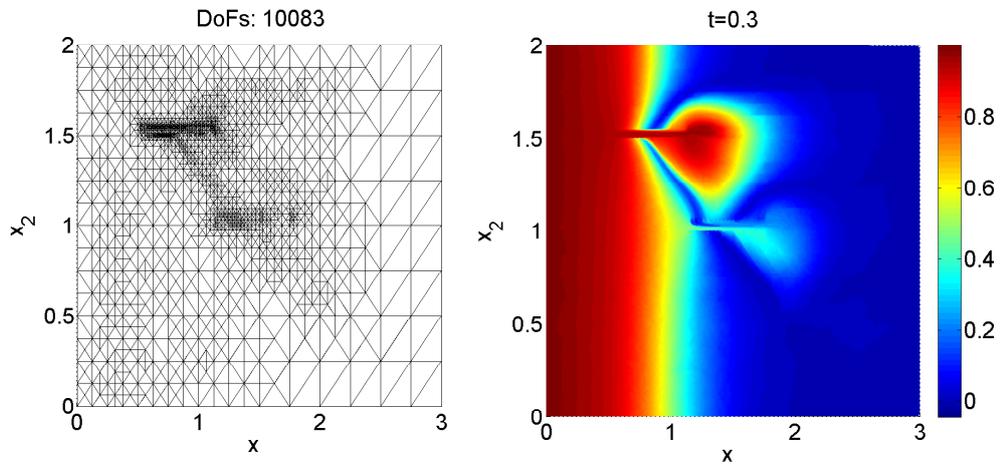


Figure 4.16: Example 4.5.4: Adaptive mesh (left) and concentration (right) at $t = 0.3$

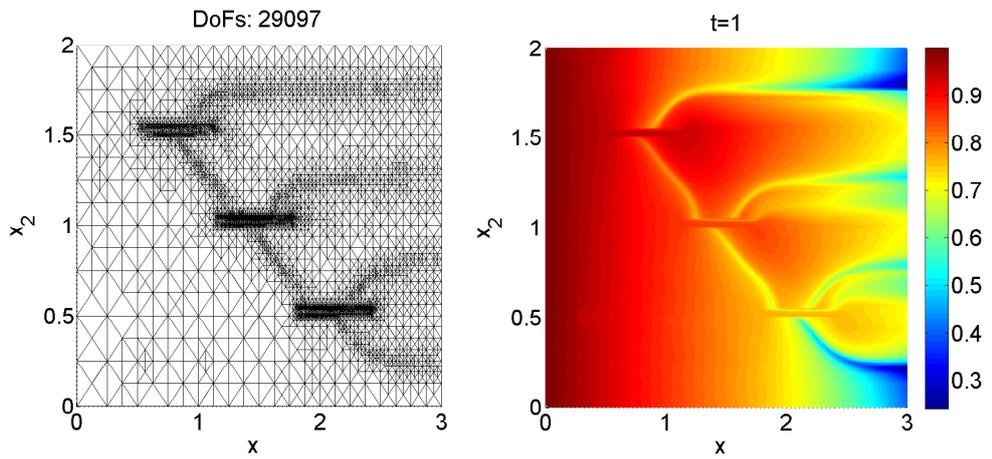


Figure 4.17: Example 4.5.4: Adaptive mesh (left) and concentration (right) at $t = 1$

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

In this thesis, we have studied convection/reaction dominated stationary and non-stationary semi-linear diffusion-convection-reaction equations. In order to handle the unphysical oscillations due to the convection, we have applied symmetric interior penalty Galerkin (SIPG) method, a type of discontinuous Galerkin methods, to discretize the problem in space, being an alternate to the well-known stabilized continuous Galerkin methods such as streamline upwind Petrov-Galerkin (SUPG) method. We have given the detailed construction of SIPG formulation on the general Poisson equation, and we have discussed the effect of penalty parameter in Chapter 2. In Chapter 3, we have studied stationary semi-linear diffusion-convection-reaction equations with existence and uniqueness results of the solution. We have figured out the adaptivity tool based on the residual-based in Péclet number robust a posteriori error estimation to handle not only the layers produced by convection but also the sharp fronts due to the non-linear reaction as an alternate to the shock/discontinuity capturing techniques in the literature, and we have shown that adaptive DG approximations are more accurate than the ones obtained by the solutions through Galerkin least squares FEMs and shock/discontinuity capturing techniques. We have proven a posteriori error bounds for stationary semi-linear diffusion-convection-reaction equations, which is the main theoretical contribution of this thesis. Moreover, we have introduced an efficient iterative method, matrix reordering technique, as a preconditioner to solve the linear systems arising from the Newton's method applied to the discrete system of stationary semi-linear diffusion-convection-reaction equations.

In Chapter 4, we have considered the non-stationary semi-linear diffusion-convection-reaction equations discretized in space by SIPG and in time by backward Euler method. We have driven and proved a posteriori error bounds for the semi-discrete and fully-discrete systems using a $L^\infty(L^2) + L^2(H^1)$ -type norm. We have stated in detailed the elliptic reconstruction technique which was a key tool in constructing a posteriori error bounds for the non-stationary model. To construct the energy-norm a posteriori error bounds for the non-stationary model, we have utilized the elliptic reconstruction technique to use a posteriori error bounds driven for stationary models in contrast to standard energy techniques which have to adapt the estimates case by case in order to compare the exact solution with numerical solution directly and are sub-optimal order in $L^\infty(L^2)$ -type norms. Using driven a posteriori error estimates, we have introduced an adaptive algorithm both in space and time, in space using both refinement and coarsening. Through the numerical studies, we have demonstrated that our adaptive algorithm allows to capture the interior and boundary layers very sharply without any significant oscillation, and also the temporal layers. In addition, we have shown the efficiency and in diffusion parameter robustness of the algorithm numerically by demonstrating the spatial and temporal effectivity indices, and the rates of the errors and estimators. We have noted that the results are similar to those ones for non-stationary linear diffusion-convection-reaction equations in [25].

In contrast to the residual-based energy techniques, there are more popular approaches such as goal oriented and hierarchical error estimates for the non-stationary problems. As a consequent, we will study on such approaches to solve the non-stationary semi-linear diffusion-convection-reaction equations as a future work. Further, in the real life non-stationary models, there occur huge systems of equations, and most of the time the perturbations of some parameters of the systems are considered. Such systems are needed to be reduced to smaller systems using certain existing techniques. Hence, again as a future work, we will study on the reduced order modeling based on proper orthogonal decomposition (POD) to solve the non-stationary semi-linear diffusion-convection-reaction equations using adaptive DG methods, which is an open problem in the literature.

REFERENCES

- [1] S. Adjerid, J. E. Flaherty, and I. Babuška. A posteriori error estimation for the finite element method-of-lines solution of parabolic problems. *Math. Models Methods Appl. Sci.*, 9(2):261–286, 1999.
- [2] M. Ainsworth. A posteriori error estimation for discontinuous Galerkin finite element approximation. *SIAM J. Numer. Anal.*, 45:1777–1798, 2007.
- [3] M. Ainsworth and J. T. Oden. *A posteriori error estimation in finite element analysis*. Wiley-Interscience, 1997.
- [4] P. F. Antonietti and E. Süli. Domain decomposition preconditioning for discontinuous Galerkin approximations of convection-diffusion problems. In *Domain Decomposition Methods in Science and Engineering XVIII*, volume 70 of *Lecture Notes in Computational Science and Engineering*. Springer Berlin Heidelberg, 2009.
- [5] D. Arnold, F. Brezzi, B. Cockburn, and L. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM J. Numer. Anal.*, 39:1749–1779, 2002.
- [6] D. N. Arnold. An interior penalty finite element method with discontinuous elements. *SIAM J. Numer. Anal.*, 19:724–760, 1982.
- [7] B. Ayuso and L. D. Marini. Discontinuous Galerkin methods for advection-diffusion-reaction problems. *SIAM J. Numer. Anal.*, 47:1391–1420, 2009.
- [8] I. Babuška, , and M. Zlámal. Nonconforming elements in the finite element method with penalty. *SIAM J. Numer. Anal.*, 10:45–59, 1973.
- [9] I. Babuška, M. Feistauer, and P. Šolín. On one approach to a posteriori error estimates for evolution problems solved by the method of lines. *Numer. Math.*, 89:225–256, 2001.
- [10] I. Babuška and T. Strouboulis. *The finite element method and its reliability*. Numerical mathematics and scientific computation. Clarendon Press, 2001.
- [11] S. T. Barnard, A. Pothen, and H. Simon. A spectral algorithm for envelope reduction of sparse matrices. *Numerical linear algebra with applications*, 4:317–334, 1995.

- [12] F. Bassi and S. Rebay. A high order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. *J. Comput. Phys.*, 131:267–279, 1997.
- [13] P. Bastian, C. Engwer, J. Fahlke, and O. Ippisch. An unfitted discontinuous Galerkin method for pore-scale simulations of solute transport. *Mathematics and Computers in Simulation*, 81(10):2051–2061, 2011.
- [14] C. E. Baumann and J. T. Oden. A discontinuous hp finite element method for convection-diffusion problems. *Comput. Methods Appl. Mech. Engrg.*, 175:311–341, 1999.
- [15] M. Bause. Stabilized finite element methods with shock-capturing for nonlinear convection-diffusion-reaction models. In *Numerical Mathematics and Advanced Applications 2009*, pages 125–133. Springer Berlin Heidelberg, 2010.
- [16] M. Bause and K. Schwegler. Analysis of stabilized higher-order finite element approximation of nonstationary and non-linear convection-diffusion-reaction equations. *Comput. Methods Appl. Mech. Engrg.*, 209–212:184–196, 2012.
- [17] M. Bause and K. Schwegler. Higher order finite element approximation of systems of convection-diffusion-reaction equations with small diffusion. *Journal of Computational and Applied Mathematics*, 246:52–64, 2013.
- [18] R. Becker, P. Hansbo, and M. G. Larson. Energy norm a posteriori error estimation for discontinuous Galerkin methods. *Comput. Methods Appl. Mech. Engrg.*, 192:723–733, 2003.
- [19] M. Benzi, G. H. Golub, and J. Liesen. Numerical solution of saddle point problems. *Acta numerica*, 14:1–137, 2005.
- [20] A. Bergam, C. Bernardi, and Z. Mghazli. A posteriori analysis of the finite element discretization of some parabolic equations. *Math. Comp.*, 74:1117–1138, 2005.
- [21] P. B. Bochev and M. D. Gunzburger. Finite element methods of least-squares type. *SIAM Review*, 40:789–837, 1998.
- [22] P. B. Bochev and M. D. Gunzburger. *Least-squares finite element methods*. Applied Mathematical Sciences, Vol. 166. Springer, 2009.
- [23] F. Brezzi, G. Manzini, D. Marini, P. Pietra, and A. Russo. Discontinuous Galerkin approximations for elliptic problems. *Numer. Methods Partial Differential Equations*, 16:365–378, 2000.
- [24] R. Bürger, M. Sepùlveda, and T. Voitovich. On the Prorjol-Koornwinder-dubiner hierarchical orthogonal polynomial basis for the DG-FEM, 2009.

- [25] A. Cangiani, E. H. Georgoulis, and S. Metcalfe. Adaptive discontinuous Galerkin methods for nonstationary convection-diffusion problems. *IMA Journal of Numerical Analysis*, pages 1–20, 2013.
- [26] P. Castillo. Performance of discontinuous Galerkin methods for elliptic PDEs. *SIAM J. Sci. Comput.*, 24:524–547, 2012.
- [27] L. Chen. *iFEM: an innovative finite element methods package in MATLAB*. Technical report, Department of Mathematics, University of California, Irvine, 2008.
- [28] Z. Chen and J. Feng. An adaptive finite element algorithm with reliable and efficient error control for linear parabolic problems. *Math. Comp.*, 73:1167–1193, 2004.
- [29] B. Cockburn and C. W. Shu. The local discontinuous Galerkin method for time-dependent convection-diffusion systems. *SIAM J. Numer. Anal.*, 35:2440–2463, 1999.
- [30] S. Deng and W. Cai. Analysis and application of an orthogonal nodal basis on triangles for discontinuous spectral element methods. *Appl. Num. Anal. Comp. Math*, 2:326–345, 2005.
- [31] D. A. Di Pietro and M. Vohralik. A review of recent advances in discretization methods, a posteriori error analysis, and adaptive algorithms for numerical modeling in geosciences. *Geosciences Numerical Methods*, 69:701–729, 2014.
- [32] V. A. Dobrev, R. D. Lazarov, and L. T. Zikatanov. Preconditioning of symmetric interior penalty discontinuous Galerkin FEM for elliptic problems. In *Domain Decomposition Methods in Science and Engineering XVII*, volume 60 of *Lecture Notes in Computer Science and Engineering*, pages 33–44. Springer, 2008.
- [33] V. Dolejší. Analysis and application of IIPG method to quasilinear nonstationary convection-diffusion problems. *J. Comp. Appl. Math.*, 222:251–273, 2008.
- [34] V. Dolejší. hp-DGFEM for non-linear convection-diffusion problems. *Mathematics and Computers in Simulation*, 87:87–118, 2013.
- [35] V. Dolejší, A. Ern, and M. Vohralík. A framework for robust a posteriori error control in unsteady nonlinear advection-diffusion problems. *SIAM J. Numer. Anal.*, 51:773–793, 2013.
- [36] V. Dolejší, M. Feistauer, and V. Sobotíková. Analysis of the discontinuous Galerkin method for nonlinear convection-diffusion problems. *Comput. Methods Appl. Mech. Engrg.*, 194:2709–2733, 2005.
- [37] W. Dörfler. A convergent adaptive algorithm for Poisson’s equations. *SIAM Journal on Numerical Analysis*, 33:1106–1124, 1996.

- [38] J. Douglas and T. Dupont. Interior penalty procedures for elliptic and parabolic Galerkin methods. In R. Glowinski and J. L. Lions, editors, *Computing Methods in Applied Sciences*, volume 58 of *Lecture Notes in Phys*, pages 207–216. Springer Berlin Heidelberg, 1976.
- [39] D. A. Dunavant. High degree efficient symmetrical Gaussian quadrature rules for the triangle. *Internat. J. Numer. Methods Engrg.*, 21:1129–1148, 1985.
- [40] Y. Epshteyn and B. Rivière. Estimation of penalty parameters for symmetric interior penalty Galerkin methods. *J. Comput. Appl. Math.*, 206:843–872, 2007.
- [41] A. Ern, A. F. Stephansen, and M. Vohralík. Guaranteed and robust discontinuous Galerkin a posteriori error estimates for convection-diffusion-reaction problems. *Journal of Computational and Applied Mathematics*, 234:114–130, 2010.
- [42] A. Ern and M. Vohralík. A posteriori error estimation based on potential and flux reconstruction for the heat equation. *SIAM J. Numer. Anal.*, 48:198–223, 2010.
- [43] M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23:298–305, 1973.
- [44] M. Fiedler. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czechoslovak Mathematical Journal*, 25:619–633, 1975.
- [45] E. Georgoulis, O. Lakkis, and J. Virtanen. A posteriori error control for discontinuous Galerkin methods for parabolic problems. *SIAM Journal on Numerical Analysis*, 49(2):427–458, 2011.
- [46] E. Georgoulis and D. Loghin. Norm preconditioners for discontinuous Galerkin *hp*-finite element methods. *SIAM Journal on Scientific Computing*, 30:2447–2465, 2008.
- [47] J. Gopalakrishnan and G. Kanschat. A multilevel discontinuous Galerkin method. *Numer. Math*, 95:527–550, 2003.
- [48] G. Hauke. A simple subgrid scale stabilized method for the advection-diffusion-reaction equation. *Comput. Methods. in Applied Mech. and Eng.*, 191:2925–2947, 2002.
- [49] R. H. W. Hoppe, G. Kanschat, and T. Warburton. Convergence analysis of an adaptive interior penalty discontinuous Galerkin method. *SIAM Journal on Numerical Analysis*, 47:534–550, 2008.
- [50] P. Houston, M. Jensen, and E. Süli. *hp*-discontinuous Galerkin finite element methods with least-squares stabilization. *J. Sci. Comput.*, 17:3–25, 2002.

- [51] P. Houston, D. Schötzau, and T. P. Wihler. Energy norm a posteriori error estimation of hp-adaptive discontinuous Galerkin methods for elliptic problems. *Mathematical Models and Methods in Applied Sciences*, 17:33–62, 2007.
- [52] P. Houston, C. Schwab, and E. Süli. Discontinuous hp-finite element methods for advection-diffusion-reaction problems. *SIAM J. Numer. Anal.*, 39:2133–2163, 2002.
- [53] T. J. R. Hughes, L. P. Franca, and G. M. Hulbert. A new finite element formulation for computational fluid dynamics: VIII. the Galerkin/least-squares method for advective-diffusive equations. *Comput. Methods Appl. Mech. Engrg.*, 73:173–189, 1989.
- [54] B. N. Jiang. *The Least-squares Finite Element Method*. Theory and Applications in Computational Fluid Dynamics and Electromagnetics. Springer, 1998.
- [55] C. Johnson and J. Pitkäranta. An analysis of the discontinuous Galerkin method for a scalar hyperbolic equation. *Math. Comp.*, 46:1–26, 1986.
- [56] 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*, 41:2374–2399, 2003.
- [57] O. A. Karakashian and F. Pascal. Convergence of adaptive discontinuous Galerkin approximations of second-order elliptic problems. *SIAM Journal on Numerical Analysis*, 45:641–665, 2007.
- [58] W. Klieber and B. Rivière. Adaptive simulations of two-phase flow by discontinuous Galerkin methods. *Computer Methods in Applied Mechanics and Engineering*, 196(1–3):404–419, 2006.
- [59] G. Kunert. A posteriori error estimation for convection dominated problems on anisotropic meshes. *Math. Methods Appl. Sci.*, 26:589–617, 2003.
- [60] R. D. Lazarov and P. S. Vassilevski. Least-squares streamline diffusion finite element approximations to singularly perturbed convection-diffusion problems. In L. G. Vulkov, J. J. H. Miller, and G. I. Shishkin, editors, *Analytical and Numerical Methods for Singularly Perturbed Problems*, pages 83–94. Nova Science Publishing House, 2000.
- [61] P. Lesaint and P. A. Raviert. On a finite element for solving the neutron transport equation, mathematical aspects of finite elements in partial differential equations. *Math. Res. Center, Univ. of Wisconsin-Madison, Academic Press, New York*, pages 89–123, 1974.
- [62] P. d. Leva. MULTIPROD TOOLBOX, multiple matrix multiplications, with array expansion enabled. Technical report, University of Rome Foro Italico, Rome, 2008.

- [63] C. Makridakis and R. H. Nochetto. Elliptic reconstruction and a posteriori error estimates for parabolic problems. *SIAM J. Numer. Anal.*, 41(4):1585–1594, 2003.
- [64] M. Manguoğlu, M. Koyutürk, A. H. Sameh, and A. Grama. Weighted matrix ordering and parallel banded preconditioners for iterative linear system solvers. *SIAM Journal on Scientific Computing*, 32:1201–1216, 2010.
- [65] J. Peraire and P. O. Persson. The compact discontinuous Galerkin (CDG) method for elliptic problems. *SIAM J. Sci. Comput.*, 30:1806–1824, 2008.
- [66] P. Persson and J. Peraire. Sub-cell shock capturing for discontinuous Galerkin methods, 2006.
- [67] M. Picasso. Adaptive finite elements for a linear parabolic problem. *Comput. Methods Appl. Mech. Engrg.*, 167:223–237, 1998.
- [68] J. Proft and B. Riviere. Discontinuous Galerkin methods for convection-diffusion equations for varying and vanishing diffusivity. *Int. J. Numer. Anal. Model*, 6(4):533–561, 2009.
- [69] 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, Los Alamos, NM, 1973.
- [70] B. Rivière. *Discontinuous Galerkin methods for solving elliptic and parabolic equations, Theory and implementation*. SIAM, 2008.
- [71] B. Rivière and M. F. Wheeler. A posteriori error estimates for a discontinuous Galerkin method applied to elliptic problems. *Computers Math. with Applications*, 46:141–163, 2003.
- [72] B. Rivière, M. F. Wheeler, and V. Girault. Improved energy estimates for interior penalty, constrained and discontinuous Galerkin methods for elliptic problems. *Comput. Geosci.*, 3:337–360, 1999.
- [73] G. Sangalli. Robust a-posteriori estimator for advection-diffusion-reaction problems. *Math. Comp.*, 77:41–70, 2008.
- [74] D. Schötzau and L. Zhu. A robust a-posteriori error estimator for discontinuous Galerkin methods for convection-diffusion equations. *Applied Numerical Mathematics*, 59:2236–2255, 2009.
- [75] D. Schötzau and L. Zhu. A robust a-posteriori error estimator for discontinuous Galerkin methods for convection-diffusion equations. *Applied Numerical Mathematics*, 59:2236–2255, 2009.
- [76] P. Solin, K. Segeth, and I. Dolezel. *Higher-Order Finite Element Methods*. Chapman & Hall/CRC Press, 2003.

- [77] J. Solsvik, S. Tangen, and H. A. Jakobsen. Evaluation of weighted residual methods for the solution of the pellet equations: The orthogonal collocation, Galerkin, tau and least-squares methods. *Computers and Chemical Engineering*, 58:223–259, 2013.
- [78] S. Sun and M. F. Wheeler. L₂(H₁) norm a posteriori error estimation for discontinuous Galerkin approximations of reactive transport problems. *Journal of Scientific Computing*, 22–23(1–3):501–530, 2005.
- [79] A. Tambue, G. J. Lord, and S. Geiger. An exponential integrator for advection-dominated reactive transport in heterogeneous porous media. *Journal of Computational Physics*, 229:3957–3969, 2010.
- [80] O. Tari and M. Manguoğlu. Revealing the saddle point structure using the largest eigenvector of the Laplacian. International Conference On Preconditioning Techniques For Scientific And Industrial Applications (19-21 June, 2013), Oxford, UK, 2013.
- [81] T. E. Tezduyar and Y. J. Park. Discontinuity capturing finite element formulations for nonlinear convection-diffusion-reaction equations. *Comp. Meth. Appl. Mech. Eng.*, 59:307–325, 1986.
- [82] M. Uzunca, B. Karasözen, and M. Manguoğlu. Adaptive discontinuous Galerkin methods for non-linear diffusion-convection-reaction equations. *Computers and Chemical Engineering*, 68:24–37, 2014.
- [83] P. Van Slingerland and C. Vuik. Fast linear solver for diffusion problems with applications to pressure computation in layered domains. *Computational Geosciences*, pages 1–14, 2014.
- [84] R. Verfürth. *A review of a posteriori error estimation and adaptive mesh refinement techniques*. Wiley-Teubner Series: Advances in Numerical Mathematics. Wiley-Teubner, 1996.
- [85] R. Verfürth. A posteriori error estimators for convection-diffusion equations. *Numer. Math.*, 80:641–663, 1998.
- [86] R. Verfürth. Robust a posteriori error estimates for stationary convection-diffusion equations. *SIAM Journal on Numerical Analysis*, 43:1766–1782, 2005.
- [87] R. Verfürth. *A posteriori Error Estimation Techniques for Finite Element Methods*. Oxford University Press, 2013.
- [88] M. F. Wheeler. An elliptic collocation-finite element method with interior penalties. *SIAM J. Numer. Anal.*, 15:152–161, 1978.

- [89] H. Yücel, M. Heinkenschloss, and B. Karasözen. Distributed optimal control of diffusion-convection-reaction equations using discontinuous Galerkin methods. In A. Cangiani, R. L. Davidchack, E. Georgoulis, A. N. Gorban, J. Levesley, and M. V. Tretyakov, editors, *Numerical Mathematics and Advanced Applications 2011*, pages 389–397. Springer Berlin Heidelberg, 2013.
- [90] H. Yücel, M. Stoll, and P. Benner. Discontinuous Galerkin finite element methods with shock-capturing for nonlinear convection dominated models. *Computers and Chemical Engineering*, 58:278–287, 2013.

APPENDIX A

MATLAB TUTORIAL

We discuss a collection of MATLAB routines using discontinuous Galerkin (DG) methods for solving and simulating steady-state diffusion-convection-reaction equations in 2D. The code employs the sparse matrix facilities of MATLAB with the coding style "vectorization" which replaces for loops by matrix operations. Moreover, we utilize multiple matrix multiplications "*MULTIPROD*" [62] to decrease the number of for loops in an efficient way.

A.1 Linear Model Problem

The general (linear) model problem used in the code is

$$\alpha u - \epsilon \Delta u + \vec{\beta} \cdot \nabla u = f \quad \text{in } \Omega, \quad (\text{A.1a})$$

$$u = g^D \quad \text{on } \Gamma^D, \quad (\text{A.1b})$$

$$\epsilon \nabla u \cdot \vec{n} = g^N \quad \text{on } \Gamma^N. \quad (\text{A.1c})$$

The domain Ω is bounded, open, convex in \mathbb{R}^2 with boundary $\partial\Omega = \Gamma^D \cup \Gamma^N$, $\Gamma^D \cap \Gamma^N = \emptyset$, $0 < \epsilon \ll 1$ is the diffusivity constant, $f \in L^2(\Omega)$ is the source function, $\vec{\beta} \in (W^{1,\infty}(\Omega))^2$ is the velocity field, $g^D \in H^{3/2}(\Gamma^D)$ is the Dirichlet boundary condition, $g^N \in H^{1/2}(\Gamma^N)$ is the Neumann boundary condition and \vec{n} denote the unit outward normal vector to the boundary.

Using the notations in Chapter 3, the DG discretized system to the problem (A.1) combining with the upwind discretization for the convection part reads as: find $u_h \in$

V_h such that

$$a_h(u_h, v_h) = l_h(v_h) \quad \forall v_h \in V_h, \quad (\text{A.2})$$

$$\begin{aligned} a_h(u_h, v_h) &= \sum_{K \in \xi_h} \int_K \epsilon \nabla u_h \cdot \nabla v_h dx + \sum_{K \in \xi_h} \int_K \alpha u_h v_h dx + \sum_{K \in \xi_h} \int_K \vec{\beta} \cdot \nabla u_h v_h dx \\ &\quad - \sum_{e \in \Gamma_h^0 \cup \Gamma_h^D} \int_e \{\epsilon \nabla u_h\} \cdot [v_h] ds + \kappa \sum_{e \in \Gamma_h^0 \cup \Gamma_h^D} \int_e \{\epsilon \nabla v_h\} \cdot [u_h] ds \\ &\quad + \sum_{K \in \xi_h} \int_{\partial K^- \setminus \partial \Omega} \vec{\beta} \cdot \vec{n} (u_h^{\text{out}} - u_h^{\text{in}}) v_h ds - \sum_{K \in \xi_h} \int_{\partial K^- \cap \Gamma_h^-} \vec{\beta} \cdot \vec{n} u_h^{\text{in}} v_h ds \\ &\quad + \sum_{e \in \Gamma_h^0 \cup \Gamma_h^D} \frac{\sigma \epsilon}{h_e} \int_e [u_h] \cdot [v_h] ds, \\ l_h(v_h) &= \sum_{K \in \xi_h} \int_K f v_h dx + \sum_{e \in \Gamma_h^D} \int_e g^D \left(\frac{\sigma \epsilon}{h_e} v_h + \kappa \epsilon \nabla v_h \cdot \vec{n} \right) ds \\ &\quad - \sum_{K \in \xi_h} \int_{\partial K^- \cap \Gamma_h^-} \vec{\beta} \cdot \vec{n} g^D v_h ds + \sum_{e \in \Gamma_h^N} \int_e g^N v_h ds. \end{aligned}$$

The parameter $\sigma \in \mathbb{R}_0^+$ should be sufficiently large independent of the mesh size h and the diffusion coefficient ϵ . In our code, we choose the penalty parameter σ on interior edges depending on the polynomial degree k as $\sigma = 3k(k+1)$ for the SIPG and IIPG methods, whereas, we take $\sigma = 1$ for the NIPG method. On boundary edges, we take the penalty parameter as twice of the penalty parameter on interior edges.

A.2 Description of MATLAB Codes

The given codes are mostly self-explanatory with comments to explain what each section of the code does. In this section, we give the line-by-line descriptions of our main code. The use of the code consists of three main parts

1. Mesh generation,
2. Entry of user defined quantities (boundary conditions, order of basis etc.),
3. Forming and solving the linear systems,
4. Plotting the solutions.

Except the last one, all the parts above, in the case of our code, take place in the m-file *Main_Linear.m* which is the main code to be used by the users for linear problems without need to entry to any other m-file. The last part, plotting the solutions, takes place in the m-file *dg_error.m*

A.2.1 Mesh Generation

In this section, we define the data structure of a triangular mesh on a polygonal domain in \mathbb{R}^2 . The data structure presented here is based on simple arrays [27] which are stored in a MATLAB "struct" that collects two or more data fields in one object that can then be passed to routines. To obtain an initial mesh, firstly, we define the nodes, elements, Dirichlet and Neumann conditions in the m-file *Main_Linear.m* through the lines 14–20, and we call the *getmesh* function to form the initial mesh structure *mesh*, line 22.

```

Nodes
Nodes = [0,0;0.5,0;1,0;0,0.5;0.5,0.5;1,0.5;0,1;0.5,1;1,1];
Elements
Elements = [4,1,5;1,2,5;5,2,6; 2,3,6;7,4,8;4,5,8;8,5,9;5,6,9];
Dirichlet bdry edges
Dirichlet = [1,2;2,3;1,4;3,6;4,7;6,9;7,8;8,9];
Neumann bdry edges
Neumann = [];
Initial mesh struct
mesh = getmesh(Nodes,Elements,Dirichlet,Neumann);

```

As it can be understood, each row in the **Nodes** array corresponds to a mesh node with the first column keeps the x-coordinate of the node and the second is for the y-coordinate, and the *i* – *th* row of the **Nodes** array is called the node having index *i*. In the **Elements** array, each row with 3 columns corresponds to a triangular element in the mesh containing the indices of the nodes forming the 3 vertices of the triangles in the counter-clockwise orientation. Finally, in the **Dirichlet** and **Neumann** arrays, each row with 2 columns corresponds to a Dirichlet and Neumann boundary edge containing the indices of the starting and ending nodes, respectively (see Fig.A.1).

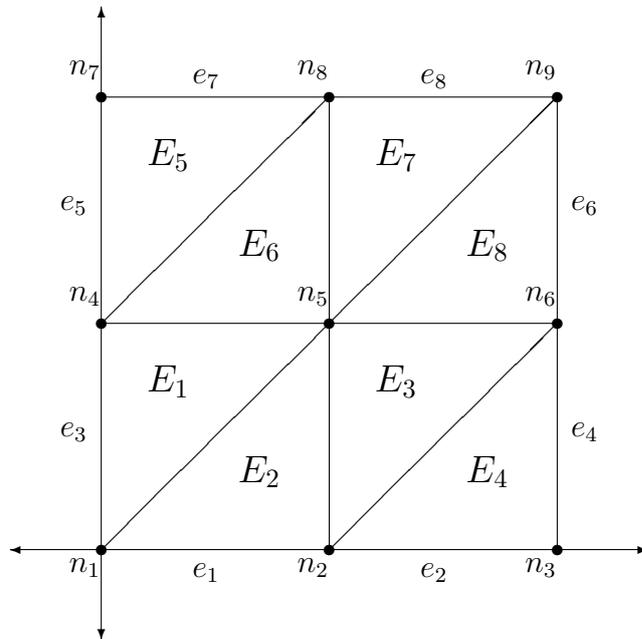


Figure A.1: Initial mesh on the unit square $\Omega = [0, 1]^2$ with nodes n_i , triangles E_j and edges e_k

The mesh "struct" in the code has the following fields:

- Nodes, Elements, Edges, intEdges, DbdEdges, NbdEdges, intEdges
- vertices1, vertices2, vertices3,
- Dirichlet, Neumann, EdgeEls, ElementsE.

which can be reached by `mesh.Nodes`, `mesh.Elements` and so on, and they are used by the other functions to form the DG construction. In line 24–26, the initial mesh is uniformly refined several times in a "for loop" by calling the function `uniformrefine`.

```

for jj=1:2
    mesh = uniformrefine(mesh);    Refine mesh
end

```

A.2.2 User Defined Quantities

There are certain input values that have to be supplied by the user. Here, we will describe that how one can define these quantities in the main code.

In lines 29–31, one determines the type of the DG method (SIPG, NIPG or IIPG) and the order of the polynomial basis to be used by the variables *method* and *degree*, respectively. According to these choices, the values of the penalty parameter and the parameter $\kappa \in \{-1, 1, 0\}$ defining DG method in (A.2) are set by calling the sub-function *set_parameter* in line 33.

```
method : NIPG=1, SIPG=2, IIPG=3
method = 2;
Degree of polynomials
degree = 1;
Set up the problem
[penalty, kappa] = set_parameter(method, degree);
```

The next step is to supply the problem parameters. In line 88–101, the diffusion constant ϵ , the advection vector $\vec{\beta}$ and the linear reaction term α are defined via the sub-functions *fdiff*, *fadv* and *freact*, respectively.

```
function diff = fdiff(x,y)
    diff = (1e-6).*ones(size(x));
end

Advection
function [adv1,adv2] = fadv(x,y)
    adv1 = (1/sqrt(5)).*ones(size(x));
    adv2 = (2/sqrt(5)).*ones(size(x));
end

Linear reaction
function react = freact(x,y)
    react = ones(size(x));
end
```

The exact solution (if exists) and the source function f are defined in lines 114–157 via the sub-functions $fexact$ and $fsource$, respectively. Finally, in lines 163–174, the boundary conditions are supplied via the sub-functions $DBCexact$ and $NBCexact$.

```
function DBC=DBCexact (fdiff, x, y)
    Evaluate the diffusion function
    diff = feval(fdiff, x, y);
    Dirichlet Boundary Condition
    DBC = 0.5*(1-tanh((2*x-y-0.25)./(sqrt(5*diff))));
end

Neumann Boundary Condition
function NC = NBCexact (mesh, fdiff, x, y)
    Neumann Boundary Condition
    NC = zeros(size(x));
end
```

A.2.3 Forming and Solving Linear Systems

To form the linear systems, firstly, let us rewrite the discrete DG scheme (A.2) as

$$a_h(u_h, v_h) := D_h(u_h, v_h) + C_h(u_h, v_h) + R_h(u_h, v_h) = l_h(v_h) \quad \forall v_h \in V_h, \quad (\text{A.3})$$

where the forms $D_h(u_h, v_h)$, $C_h(u_h, v_h)$ and $R_h(u_h, v_h)$ corresponding to the diffusion, convection and linear reaction parts of the problem, respectively, given by

$$\begin{aligned} D_h(u_h, v_h) = & \sum_{K \in \xi_h} \int_K \epsilon \nabla u_h \cdot \nabla v_h dx + \sum_{e \in \Gamma_0 \cup \Gamma_D} \frac{\sigma \epsilon}{h_e} \int_e [u_h] \cdot [v_h] ds \\ & - \sum_{e \in \Gamma_0 \cup \Gamma_D} \int_e \{\epsilon \nabla u_h\} \cdot [v_h] ds + \kappa \sum_{e \in \Gamma_0 \cup \Gamma_D} \int_e \{\epsilon \nabla v_h\} \cdot [u_h] ds \end{aligned}$$

$$\begin{aligned}
C_h(u_h, v_h) &= \sum_{K \in \xi_h} \int_K \vec{\beta} \cdot \nabla u_h v_h dx \\
&\quad + \sum_{K \in \xi_h} \int_{\partial K^- \setminus \partial \Omega} \vec{\beta} \cdot \vec{n} (u_h^{out} - u_h^{in}) v_h ds - \sum_{K \in \xi_h} \int_{\partial K^- \cap \Gamma^-} \vec{\beta} \cdot \vec{n} u_h^{in} v_h ds \\
R_h(u_h, v_h) &= \sum_{K \in \xi_h} \int_K \alpha u_h v_h dx \\
l_h(v_h) &= \sum_{K \in \xi_h} \int_K f v_h dx + \sum_{e \in \Gamma_D} \int_e g^D \left(\frac{\sigma \epsilon}{h_e} v_h + \kappa \epsilon \nabla v_h \cdot \mathbf{n} \right) ds \\
&\quad - \sum_{K \in \xi_h} \int_{\partial K^- \cap \Gamma^-} \vec{\beta} \cdot \vec{n} g^D v_h ds + \sum_{e \in \Gamma_N} \int_e g^N v_h ds,
\end{aligned}$$

For a set of basis functions $\{\phi_i\}_{i=1}^N$ spanning the space V_h , the discrete solution $u_h \in V_h$ is of the form

$$u_h = \sum_{j=1}^N v_j \phi_j \quad (\text{A.4})$$

where $v = (v_1, v_2, \dots, v_N)^T$ is the unknown coefficient vector. After substituting (A.4) into (A.3) and taking $v_h = \phi_i$, we get for $i = 1, 2, \dots, N$, the linear systems of equations

$$\sum_{j=1}^N v_j D_h(\phi_j, \phi_i) + \sum_{j=1}^N v_j C_h(\phi_j, \phi_i) + \sum_{j=1}^N v_j R_h(\phi_j, \phi_i) = l_h(\phi_i) \quad (\text{A.5})$$

Thus, for $i = 1, 2, \dots, N$, to form the linear system in matrix-vector form, we need the matrices $D, C, R \in \mathbb{R}^{N \times N}$ related to the terms including the forms D_h, C_h and R_h in (A.5), respectively, satisfying

$$Dv + Cv + Rv = F$$

with the unknown coefficient vector v and the vector $F \in \mathbb{R}^N$ related to the linear rhs functionals $l_h(\phi_i)$ such that $F_i = l_h(\phi_i)$, $i = 1, 2, \dots, N$. In the code *Main_Linear.m*, all the matrices D, C, R and the vector F are obtained by calling the function *global_system* in lines 36–37, in which the sub-functions introduced in the previous subsection are used. In line 39, we set the stiffness matrix, *Stiff*, as the sum of the obtained matrices and we solve the linear system for the unknown coefficient vector *coef* := v .

```

    Compute global matrices and rhs global vector
[D,C,R,F] = global_system(mesh,@fdiff,@fadv,@freact,[],...
    @fsource,@DBCexact,@NBCexact,penalty,kappa,degree);

Stiff = D + C + R;      Stiffness matrix

coef = Stiff\F;        Solve the linear system

```

A.2.4 Plotting Solutions

After solving the problem for the unknown coefficient vector, the solutions are plotted via the function `dg_error` in line 74, and also the L^2 -error between the exact and numerical solution is computed.

```

    Compute L2-error and plot the solution
[l2err,hmax] = dg_error(coef,mesh,@fexact,@fdiff,degree);

```

A.3 Models with Non-Linear Reaction Mechanisms

The model problem in this case is

$$\alpha u - \epsilon \Delta u + \vec{\beta} \cdot \nabla u + r(u) = f \quad \text{in } \Omega, \quad (\text{A.6a})$$

$$u = g^D \quad \text{on } \Gamma^D, \quad (\text{A.6b})$$

$$\epsilon \nabla u \cdot \vec{n} = g^N \quad \text{on } \Gamma^N. \quad (\text{A.6c})$$

which arises from the time discretization of the time-dependent non-linear diffusion-convection-reaction equations. Here, the coefficient of the linear reaction term, $\alpha > 0$, stand for the temporal discretization, corresponding to $1/\Delta t$, where Δt is the discrete time-step. The model (A.6) differs from the model (A.1) by the additional non-linear term $r(u)$.

To solve the non-linear problems, we use the m-file *Main_Nonlinear* which is similar to the m-file *Main_Linear*, but now we use Newton iteration to solve for $i =$

1, 2, ..., N the non-linear system of equations

$$\sum_{j=1}^N v_j D_h(\phi_j, \phi_i) + \sum_{j=1}^N v_j C_h(\phi_j, \phi_i) + \sum_{j=1}^N v_j R_h(\phi_j, \phi_i) + \int_{\Omega} r(u_h) \phi_i dx = l_h(\phi_i) \quad (\text{A.7})$$

Similar to the linear case, the above system leads to the matrix-vector form

$$Dv + Cv + Rv + H(v) = F$$

where, in addition to the matrices $D, C, R \in \mathbb{R}^{N \times N}$ and the vector $F \in \mathbb{R}^N$, we also need the vector $H \in \mathbb{R}^N$ related to the non-linear term such that

$$H_i(v) = \int_{\Omega} r \left(\sum_{j=1}^N v_j \phi_j \right) \phi_i dx, \quad i = 1, 2, \dots, N.$$

For an initial guess $v^0 = (v_1^0, v_2^0, \dots, v_N^0)^T$, we solve the system

$$\begin{aligned} J^k w^k &= -R^k \\ v^{k+1} &= w^k + v^k, \quad k = 0, 1, 2, \dots \end{aligned} \quad (\text{A.8})$$

until a user defined tolerance is satisfied. In (A.8), R^k and J^k denote the vector of system residual and its Jacobian matrix at the current iterate v^k , respectively, given by

$$\begin{aligned} R^k &= Sv^k + H(v^k) - F \\ J^k &= S + J_H(v^k) \end{aligned}$$

where $J_H(v^k)$ is the Jacobian matrix of the non-linear vector H at v^k

$$J_H(v^k) = \begin{bmatrix} \frac{\partial H_1(v^k)}{\partial v_1^k} & \frac{\partial H_1(v^k)}{\partial v_2^k} & \dots & \frac{\partial H_1(v^k)}{\partial v_N^k} \\ \vdots & \ddots & & \vdots \\ \frac{\partial H_N(v^k)}{\partial v_1^k} & \frac{\partial H_N(v^k)}{\partial v_2^k} & \dots & \frac{\partial H_N(v^k)}{\partial v_N^k} \end{bmatrix}$$

In the code *Main_Nonlinear*, obtaining the matrices D, C, R and the rhs vector F is similar to the linear case, but now we give the function handle *freact_nonlinear*, which is a sub-function in the main file *Main_Nonlinear*, lines 104–109, as an input to compute the source function.

```

Compute global matrices and rhs global vector
[D, C, R, F] = global_system(mesh, @fdiff, @fadv, @freact, @freact_nonlinear, ...
    @fsource, @DBCexact, @NBCexact, penalty, kappa, degree);

```

```

function [r,dr] = freact_nonlinear(u)
    Value of the non-linear reaction term
    r = u.^2;
    Value of the Jacobian of the non-linear reaction term
    dr = 2*u;
end

```

In line 42, we initialize the initial guess for Newton iteration, and we solve the non-linear system in lines 46–71. To obtain the non-linear vector H and its Jacobian J_H at the current iterate, we call the function *nonlinear_global* in line 51. It uses the function handle *freact_nonlinear* and has to be supplied by user for $r(u)$ and $r'(u)$.

```

    Newton iteration
noi=0;
for ii=1:50
    noi=noi+1;

    Compute the non-linear vector and its Jacobian matrix at
    the current iterate
    [H,JH] = nonlinear_global(coef,mesh,@freact_nonlinear,degree);

    Form the residual of the system
    Res = Stiff*coef + H - F;

    Form the Jacobian matrix of the system
    (w.r.t. unknown coefficients coef)
    J = Stiff + JH ;

    Solve the linear system for the correction "w"
    w = J \ (-Res);

    Update the iterate
    coef = coef + w;

    Check the accuracy
    if norm(J*w+Res) < 1e-20
        break;
    end
end

```

```
end
```

A.4 Full Version of the Code *Main_Nonlinear*

```
1   This routine solves the diffusion-convection-reaction equation
2
3    $\alpha u - \epsilon \Delta u + b \cdot \nabla u + r(u) = f$ 
4
5   using DG-FEM.
6
7   function Main_Nonlinear()
8   clear all
9   clc
10
11   Generate the mesh
12
13   Nodes
14   Nodes = [0,0;0.5,0;1,0;0,0.5;0.5,0.5;1,0.5;0,1;0.5,1;1,1];
15   Elements
16   Elements = [4,1,5;1,2,5;5,2,6; 2,3,6;7,4,8;4,5,8;8,5,9;5,6,9];
17   Dirichlet bdry edges
18   Dirichlet = [1,2;2,3;1,4;3,6;4,7;6,9;7,8;8,9];
19   Neumann bdry edges
20   Neumann = [];
21   Initial mesh struct
22   mesh = getmesh(Nodes,Elements,Dirichlet,Neumann);
23
24   for jj=1:2
25       mesh = uniformrefine(mesh);   Refine mesh
26   end
27
28   method : NIPG=1, SIPG=2, IIPG=3
29   method = 2;
30   Degree of polynomials
31   degree = 1;
32   Set up the problem
```

```

33 [penalty,kappa] = set_parameter(method,degree);
34
35     Compute global matrices and rhs global vector
36 [D,C,R,F] = global_system(mesh,@fdiff,@fadv,@freact,@freact_nonlinear,...
37     @fsource,@DBCexact,@NBCexact,penalty,kappa,degree);
38
39 Stiff = D + C + R;      Stiffness matrix
40
41     Initial guess for Newton iteration
42 coef = zeros(size(Stiff,1),1);
43
44     Newton iteration
45 noi=0;
46 for ii=1:50
47     noi=noi+1;
48
49     Compute the non-linear vector and its Jacobian matrix at
50     the current iterate
51 [H,JH] = nonlinear_global(coef,mesh,@freact_nonlinear,degree);
52
53     Form the residual of the system
54 Res = Stiff*coef + H - F;
55
56     Form the Jacobian matrix of the system
57     (w.r.t. unknown coefficients coef)
58 J = Stiff + JH ;
59
60     Solve the linear system for the correction "w"
61 w = J \ (-Res);
62
63     Update the iterate
64 coef = coef + w;
65
66     Check the accuracy
67 if norm(J*w+Res) < 1e-20
68     break;
69 end
70
71 end
72

```

```

73     Compute L2-error and plot the solution
74 [l2err,hmax] = dg_error(coef,mesh,@fexact,@fdiff,degree);
75
76     Degree of freedom
77 dof = size(mesh.Elements,1)*(degree+1)*(degree+2)*0.5;
78
79 fprintf('    DoFs      h_max      L2-error      it\n')
80
81 fprintf(' 7d      5.3f      5.3e      d\n',dof, hmax ,l2err,noi);
82
83 end
84
85     Define diffusion, advection, and reaction as subfunctions
86
87     Diffusion
88 function diff = fdiff(x,y)
89     diff = (1e-6).*ones(size(x));
90 end
91
92     Advection
93 function [adv1,adv2] = fadv(x,y)
94     adv1 = (1/sqrt(5))*ones(size(x));
95     adv2 = (2/sqrt(5))*ones(size(x));
96 end
97
98     Linear reaction
99 function react = freact(x,y)
100     react = ones(size(x));
101 end
102
103     Non-linear reaction
104 function [r,dr] = freact_nonlinear(u)
105     Value of the non-linear reaction term
106     r = u.^2;
107     Value of the Jacobian of the non-linear reaction term
108     dr = 2*u;
109 end
110
111     Define exact solution and force as subfunctions
112

```

```

113     Exact solution
114 function [yex,yex_x,yex_y] = fexact(fdiff,x,y)
115     Evaluate the diffusion function
116     diff = feval(fdiff,x,y);
117     Exact value
118     yex = 0.5*(1-tanh((2*x-y-0.25)./(sqrt(5*diff))));
119     First derivative value wrt x
120     yex_x = (-1./sqrt(5*diff)).*(sech((2*x-y-0.25)./(sqrt(5*diff)))).^2;
121     First derivative value wrt y
122     yex_y=((0.5)./(sqrt(5*diff))).*(sech((2*x-y-0.25)./(sqrt(5*diff)))).^2;
123 end
124
125 Force function
126 function source = fsource(fdiff,fadv,freact,freact_nonlinear,x,y)
127     Evaluate the diffusion function
128     diff = feval(fdiff,x,y);
129     Evaluate the advection function
130     [adv1,adv2] = feval(fadv,x,y);
131     Evaluate the reaction function
132     reac = feval(freact,x,y);
133
134     Exact value
135     yex = 0.5*(1-tanh((2*x-y-0.25)./(sqrt(5*diff))));
136     First derivative value wrt x
137     yex_x = (-1./sqrt(5*diff)).*(sech((2*x-y-0.25)./(sqrt(5*diff)))).^2;
138     First derivative value wrt y
139     yex_y=((0.5)./(sqrt(5*diff))).*(sech((2*x-y-0.25)./(sqrt(5*diff)))).^2;
140     Second derivative value wrt x
141     yex_xx = ((0.8)./diff).*tanh((2*x-y-0.25)./(sqrt(5*diff))).*...
142             (sech((2*x-y-0.25)./(sqrt(5*diff)))).^2;
143     Second derivative value wrt y
144     yex_yy = ((0.2)./diff).*tanh((2*x-y-0.25)./(sqrt(5*diff))).*...
145             (sech((2*x-y-0.25)./(sqrt(5*diff)))).^2;
146
147     Value of non-linear reaction (if exists)
148     if ~isempty(freact_nonlinear)
149         nonlin_reac = freact_nonlinear(yex);
150     else
151         nonlin_reac = 0;
152     end

```

```

153
154     Value of the source function
155     source = -diff.*(yex_xx+yex_yy)+(adv1.*yex_x+adv2.*yex_y)+...
156         reac.*yex+nonlin_reac;
157 end
158
159
160     Boundary Conditions
161
162     Drichlet Boundary Condition
163 function DBC=DBCexact(fdiff,x,y)
164     Evaluate the diffusion function
165     diff = feval(fdiff,x,y);
166     Drichlet Boundary Condition
167     DBC = 0.5*(1-tanh((2*x-y-0.25)./(sqrt(5*diff))));
168 end
169
170     Neumann Boundary Condition
171 function NC = NBCexact(mesh,fdiff,x,y)
172     Neumann Boundary Condition
173     NC = zeros(size(x));
174 end
175
176
177
178     Set-up parameters function for DG-FEM
179
180 function [penalty,kappa]=set_parameter(method,degree)
181
182     global Equation;
183
184     Equation.b0 = 1;      Superpenalization parameter (In standart b0=1)
185     Equation.base = 2;   Choose the basis ( 1:monomials, 2:Dubiner Basis)
186
187     switch method
188     case 1
189         NIPG
190         Equation.method = 1;
191         kappa = 1;        type of primal method
192         penalty = 1;      penalty parameter

```

```
193     case 2
194         SIPG
195         Equation.method = 2;
196         kappa = -1;           type of primal method
197         penalty = 3*degree*(degree+1); penalty parameter
198     case 3
199         IIPG
200         Equation.method = 3;
201         kappa = 0;           type of primal method
202         penalty = 3*degree*(degree+1); penalty parameter
203
204     end
205
206 end
```

APPENDIX B

A SIMPLE MATLAB CODE FOR MATRIX REORDERING

```
function [N, partitionPoint] = MatrixReorder(M)
    Authors:   Omer Tari, Murat Manguoglu
              June 2013

    This function reorders a sparse matrix using the largest
    eigenvector of the Laplacian of the graph corresponding to
    the matrix. Reordered matrix can be partitioned into 2x2 blocks

    Input:     M - sparse input matrix
    Output:    N - resulting reordered matrix
              partitionPoint - row and column index of the partition borders

    Reference: Tari O, Manguoglu M, ''Revealing the Saddle Point Structure
    Using the Largest Eigenvector of the Laplacian'', International
    Conference On Preconditioning Techniques For Scientific And Industrial
    Applications (19-21 June 2013), Oxford, UK

    Get size of the Matrix
    [x,y] = size(M);

    theRank = sprank(M);

    if( theRank ~= x || x ~= y )
        return;
    end
    permute = symrcm(M);
    M = M(permute,permute);
```

```

    If not symmetric make symmetric
isSym = M - M';
if( nnz(isSym) == 0 )
    isSym = abs(M);
else
    isSym = abs(M) + abs(M');
end

    Get diagonal
Diagonal = diag(diag(isSym));

    Get rest
Rest = isSym - Diagonal;

    Row , Col and Values of Sparse
[row,col,values] = find(Rest);
values = -1;

RestNew = sparse(row,col,values,x,y);
summationVector = sum(abs(RestNew'));
DiagonalNew = diag(summationVector);
Laplacian = DiagonalNew + RestNew;

opts.accuracy = 1e-12;
[eigenVectors,eigenValues] = eigs(Laplacian,5,'lm',opts);
theEigenVector = eigenVectors(1:x,1);

partitionPoint = 0;
absEigenVector(x) = 0;
for i = 1:x
    if ( theEigenVector(i) <= 0 )
        absEigenVector(i) = - theEigenVector(i);
        partitionPoint = partitionPoint+1;
    end
end

[t,p] = sort(absEigenVector,'descend');
N = M(p,p);
end

```

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Uzunca, Murat

Nationality: Turkish (TC)

Date and Place of Birth: 23.02.1980, Istanbul

Marital Status: Single

e-mail: uzunca@gmail.com

EDUCATION

| Degree | Institution | Year of Graduation |
|---------------|------------------------------|---------------------------|
| Ph.D. | Mathematics Department, METU | 2014 |
| B.S. | Mathematics Department, AIBU | 2008 |

COMPUTER SKILLS

Proficient: Matlab, Mathematica, Latex, Microsoft Office Packages,
Microsoft Windows OS, Linux/Unix

Intermediate: Java, HTML, SQL, C, C++

Basic: Web Development, Flash

LANGUAGE SKILLS

Turkish native

English fluently

German basic

AWARDS AND SCHOLARSHIPS

- 09/2008–08/2013** 2211-Doctorate Scholarship
Turkish Scientific and Technical Research Council (TÜBİTAK)
- 06/2008** B.S. High Honor Degree With CGPA 3.68/4.00
Mathematics Department, AIBU

RESEARCH VISITS

Place: Zuse Institute Berlin, Berlin, Germany

Time period: 01/2013–02/2013

Aim: Collaborated study in the implementation of discontinuous Galerkin method into the numerical tool package KASKADE7.2.

Supervised by: Dr. Martin Weiser

Supported by: Zuse Institute Berlin, Berlin

REVIEWING ACTIVITIES

Months/Year **Journal**

05, 09/2014 Journal of Chemical Engineering & Process Technology

PUBLICATIONS

Articles in Journals

Bülent Karasözen, Murat Uzunca: *Time-Space Adaptive Discontinuous Galerkin Method for Advection-Diffusion Equations with Non-Linear Reaction Mechanism.*

International Journal on Geomathematics (2014), **5**, 255-288.

DOI: <http://dx.doi.org/10.1007/s13137-014-0067-z>

Murat Uzunca, Bülent Karasözen, Murat Manguoğlu: *Adaptive discontinuous Galerkin methods for non-linear diffusion-convection-reaction equations.*

Computers and Chemical Engineering (2014), **68**, 24-37.

DOI: <http://dx.doi.org/10.1016/j.compchemeng.2014.05.002>

Papers Submitted to International Journals

Bülent Karasözen, Canan Akkoyunlu, Murat Uzunca: *Model Order Reduction for Nonlinear Schrödinger Equation*. Institute of Applied Mathematics, Middle East Technical University, number 2014-7, Preprint, 2014.

Ayşe Sariaydın, Murat Uzunca, Bülent Karasözen: *Average vector field method with discontinuous Galerkin finite element space discretization for Allen–Cahn equation*. Institute of Applied Mathematics, Middle East Technical University, number 2014-5, Preprint, 2014.

Conference Proceedings:

Bülent Karasözen, Murat Uzunca, Murat Manguoğlu: *Adaptive Discontinuous Galerkin Methods for Nonlinear Diffusion-Convection-Reaction Equations*.

A. Abdulle et al. (eds.), Numerical Mathematics and Advanced Applications - ENUMATH 2013, Lecture Notes in Computational Science and Engineering (2015), **103**, 85–93.

DOI: http://dx.doi.org/10.1007/978-3-319-10705-9_8

Canan Akkoyunlu, Murat Uzunca, Bülent Karasözen: *Model Order Reduction for Nonlinear Schrödinger Equation*. In: Chaotic Modeling and Simulation (CMSIM) (2013), **4**, 637-645.

SCIENTIFIC MEETINGS

Presentations in International Scientific Meetings

Murat Uzunca, Bülent Karasözen: *Space-Time Adaptive Discontinuous Galerkin Methods for Diffusion-Convection Equations with Nonlinear Reaction*. Conference on Computational Methods in Applied Mathematics CMAM6, St. Wolfgang/Strobl, Austria.

Murat Uzunca, Bülent Karasözen: *Adaptive Discontinuous Finite Elements For Nonlinear Diffusion-Convection-Reaction Equations*. 25th Biennial Numerical Analysis Conference, Glasgow, Scotland.

Murat Uzunca, Bülent Karasözen: *Adaptive Discontinuous Finite Elements For Nonlinear Reactive Flows*. Workshop on Adaptivity and Model Order Reduction in PDE Constrained Optimization, Hamburg, Germany.

Presentations in National Scientific Meetings

Murat Uzunca, Bülent Karasözen: *Doğrusal Olmayan Parabolik Difüzyon-Konveksiyon-Reaksiyon Denklemlerinin Süreksiz Galerkin Yöntemleriyle Zaman ve Uzayda Uyarlamalı Çözümü*. 27. Ulusal Matematik Sempozyumu UMS 2014, Istanbul, Turkey.

PROFESSIONAL EXPERIENCE

| Year | Place | Enrollment |
|-------------|------------------------------|--------------------|
| 2009-2014 | Mathematics Department, METU | Research Assistant |

TEACHING EXPERIENCE

MATH 119 - Calculus With Analytic Geometry (METU)

Teaching Assistant for Fall 2009, 2010, 2011, 2012, 2013, Spring 2012, 2013
(Assistant Coordinator for Fall 2011, 2012, Spring 2012, 2013).

MATH 120 - Calculus for Functions of Several Variables (METU)

Teaching Assistant for Spring 2010, 2011 (Assistant Coordinator for Spring 2011).

MATH 219 - Introduction to Differential Equations (METU)

Teaching Assistant and Assistant Coordinator for Fall 2013, 2014, Spring 2013.

MATH 480 - Numerical Methods for Differential Equations (METU)

Teaching Assistant and Assistant Coordinator for Spring 2012, 2013.