Optimal Approximation Spaces for Discontinuous Petrov-Galerkin Schemes with Optimal Test Functions

Optimale Approximationsräume für diskontinuierliche Petrov-Galerkin Verfahren mit optimalen Testfunktionen

Von der Fakultät für Maschinenwesen der Rheinisch–Westfälischen Technischen Hochschule Aachen zur Erlangung des akademischen Grades eines Doktors der Ingenieurwissenschaften genehmigte Dissertation

vorgelegt von

Ankit Chakraborty

Berichter: Assoc. Prof. Georg May, Ph.D.
Univ.-Prof. Marek Behr, Ph.D.

Tag der mündlichen Prüfung: 11.08.2022

Diese Dissertation ist auf den Internetseiten der Universitätsbibliothek online verfügbar.

Optimal Approximation Spaces for Discontinuous Petrov-Galerkin Schemes with Optimal Test Functions

Optimale Approximationsräume für diskontinuierliche Petrov-Galerkin Verfahren mit optimalen Testfunktionen

Von der Fakultät für Maschinenwesen der Rheinisch–Westfälischen Technischen Hochschule Aachen zur Erlangung des akademischen Grades eines Doktors der Ingenieurwissenschaften genehmigte Dissertation

vorgelegt von

Ankit Chakraborty

Berichter: Assoc. Prof. Georg May, Ph.D.
Univ.-Prof. Marek Behr, Ph.D.

Tag der mündlichen Prüfung: 11.08.2022

Diese Dissertation ist auf den Internetseiten der Universitätsbibliothek online verfügbar.

Abstract

Certain Petrov-Galerkin methods deliver inherently stable formulations of variational problems for a given mesh by selecting appropriate pairs of trial and test spaces. Furthermore, these schemes are especially suited for adaptation due to their inherent ability to yield robust a posteriori error estimates. These numerical methods are also known as Petrov-Galerkin schemes (PG schemes) with optimal test functions. On the other hand, metric-based continuous mesh models have previously been formulated with the aim to build (near) optimal anisotropic meshes with respect to interpolation error models associated with piecewise polynomial approximation spaces. These are compatible with any numerical scheme using such spaces. The primary goal of this dissertation is to formulate the correct continuous-mesh error models for the optimal Petrov-Galerkin methodology, thus pairing the ability to produce near-optimal anisotropic meshes with a numerical scheme which in turn produces optimal stability and approximation properties on these meshes.

Employing discontinuous test and trial spaces with optimal test function framework leads to the well-known discontinuous Petrov-Galerkin (DPG) schemes with optimal test functions. This dissertation presents an anisotropic metric-based mesh adaptation strategy utilizing the inbuilt a posteriori error estimator of the DPG schemes. Thus, we end up coupling DPG schemes with metric-based adaptations to generate (near) optimal approximation spaces. Moreover, employing continuous mesh models provides the flexibility of controlling the computational cost in terms of the degrees of freedom. Furthermore, an $hp-$adaptation technique based on a continuous mesh model has been proposed in this dissertation.

Solution variables do not hold one's prime interest in many engineering applications. In such cases, certain output functionals that depend upon the solution variables are of much more importance. Examples of such output functionals are aerodynamic drag and flux across a boundary. Hence, making goal-oriented adaptation indispensable for generating optimal meshes for such problems. Thus, the proposed adaptation strategies have been extended to perform goal-oriented mesh adaptations. The goal-oriented adaptation strategy utilizes an error estimator from the DPG-star method. Finally, the construction of near-optimal meshes is demonstrated using problems containing boundary layers and singularities for solution-driven refinements and goal-oriented refinements.

This page is intentionally left blank.

Zusammenfassung

Bestimmte Petrov-Galerkin-Methoden liefern inhärent stabile Formulierungen von Variationsproblemen für ein gegebenes Gitter, indem sie geeignete Paare von Versuchs- und Testräumen auswählen. Darüber hinaus eignen sich diese Verfahren aufgrund ihrer inhärenten Fähigkeit, robuste a posteriori-Fehlerschätzungen zu liefern, besonders zur Gitteranpassung. Diese numerischen Methoden sind auch als Petrov-Galerkin-Methoden (PG-Methoden) mit optimalem Testfunktionsrahmen bekannt. Es existieren bereits metrisch basierte kontinuierliche Gittermodelle mit dem Ziel, (nahezu) optimale anisotrope Gitter im Hinblick auf Interpolationsfehlermodelle zu erstellen, die mit stückweisen polynomialen Approximationsräumen verbunden sind. Diese sind mit jedem numerischen Verfahren, das solche Räume verwendet, kompatibel. Das primäre Ziel dieser Dissertation ist es, die richtigen Fehlermodelle für kontinuierliche Gitter für die optimale Petrov-Galerkin-Methode zu formulieren und so die Fähigkeit, nahezu optimale anisotrope Gitter zu erzeugen, mit einem numerischen Verfahren zu verbinden, das seinerseits optimale Stabilitäts- und Approximationseigenschaften auf diesen Gittern erzeugt.

Die Verwendung diskontinuierlicher Test- und Löesungsräume mit optimalen Testfunktionen führt zu den bekannten diskontinuierlichen Petrov-Galerkin (DPG)-Verfahren mit optimalen Testfunktionen. In dieser Dissertation wird eine anisotrope metrikbasierte Gitteranpassungsstrategie vorgestellt, die den eingebauten a-posteriori-Fehlerschätzer der DPG-Methode nutzt. Auf diese Weise werden DPG-Methoden mit metrikbasierter Gitter-Adaptierung gekoppelt, um (nahezu) optimale Approximationsräume zu erzeugen.

Die Verwendung von Modellen mit kontinuierlichen Gittern bietet die Flexibilität, die Rechenkosten in Bezug auf die Freiheitsgrade zu kontrollieren. Außerdem wird in dieser Dissertation eine $hp-$Anpassungstechnik auf der Grundlage eines kontinuierlichen Gittermodells vorgeschlagen.

Bei einer Vielzahl technischer Anwendungen sind die Lösungsvariablen nicht von vorrangigem Interesse. In diesen Fällen sind zielfunktionale, die von den Lösungsvariablen abhängen, von viel größerer Bedeutung. Einige Beispiele für solche zielfunktionale sind der Luftwiderstand oder ein Fluss über eine Grenze des Rechengebiets. Dies macht eine zielgerichtete Anpassung für die Erzeugung optimaler Gitter erforderlich. Daher wurden die vorgeschlagenen Anpassungsstrategien in dieser Dissertation erweitert, um zielgerichtete Gitteranpassungen durchzuführen. Die zielorientierte Anpassungsstrategie verwendet einen Fehlerschätzer der DPG-Star-Methode. Die Konstruktion nahezu optimaler Gitter wird anhand von Problemen mit Grenzschichten und Singularitäten für lösungsgesteuerte und zielorientierte Verfeinerungen demonstriert.

This page is intentionally left blank.

Acknowledgments

I want to thank everyone who has helped me through this Ph.D. First, I would like to thank my advisor, Prof. Georg May, for giving me an opportunity to work at AICES and IRTG. Working with him has been an incredible experience. His expertise in numerical methods, his ability to simplify complex ideas, and his work ethic were few of the many things I had the opportunity to learn from. In addition, he always encouraged me to be an explorer in terms of trying new ideas and exploring new topics. This has been the most enjoyable experience in this Ph.D. for me. I am also very appreciative of the patience with which he explained various fundamental concepts and also, provided me the opportunity to attend various conferences throughout my Ph.D.

I would also like to thank Prof. Marek Behr for the support over the years and for being part of my doctoral committee. I want to thank Prof. Julia Kowalski for taking time off from her schedule to chair my doctoral committee. I thank Prof. Mikhail Itskov for being part of my doctoral committee.

I thank Prof. Demkowicz for explaining the intricacies of DPG schemes and the associated mathematical machinery. His insights helped me to develop a robust understanding of the DPG finite element schemes, which is fundamental to the work presented in this thesis. I would also like to thank Dr. Brendan Keith for explaining his work on adjoint methods associated with DPG schemes. Then, I would also like to extend my thanks to Dr. Stefan Henneking for explaining his work on development of high-performance DPG finite element solvers.

Next, I would like to thank my colleague and close friend Dr. Ajay Rangarajan for his constant support and patience while answering questions regarding mesh adaptivity and the code, for innumerable discussion spanning from research to various aspects of life which made this Ph.D. a lot more fun.

The support I received from the AICES service team has made working here an enjoyable experience. In particular, I would like to thank Dr. Faber who has ensured that doctoral candidates get the best environment possible to conduct their research.

A special thanks to the closest group of people I have in Aachen: Eshwar Jagadeesh, Chew Junxian, Michal Rajski, and Karina Ruzaeva. I cannot comprehend their contribution to me and my growth as a person in a few words. But to summarize, I would like to thank them for being patient with me, making me a better individual, and being next to me during every high and low point in the last few years.

Among my IRTG colleagues, I am fortunate to meet one of the sharpest people I have ever encountered, which motivated me to be a better version of myself as a researcher

and person. In addition, Karsten, Stefan, and Konstantin have been great lunch-mates with whom I had many intriguing discussions. In short, IRTG has been an incredible experience.

Finally, I would like to express my sincere gratitude to my parents, Anita Chakraborty and Barun Chakraborty. Without your support and sacrifices, nothing would have been possible. So I owe you guys a lot!

Author's Contributions

Publications

The thesis is based on the following publications:

Journal Publications

- Ankit Chakraborty, Ajay Mandyam Rangarajan, Georg May, An anisotropic h-adaptive strategy for discontinuous Petrov-Galerkin schemes using a continuous mesh model, Computers & Mathematics with Applications, Vol 106 (2022),pp1-17.

Conference Proceedings

- Ankit Chakraborty, Ajay Mandyam Rangarajan, Georg May, A Residual-Based HP-Mesh Optimization Technique for Petrov-Galerkin Schemes with Optimal Test Functions, in: WCCM-ECCOMAS2020, doi:10.23967/wccm-eccomas.2020.014.

Contributions have been made in the following publication, but results have not directly influenced this work:

- Ajay Mandyam Rangarajan, Ankit Chakraborty, Georg May, A Goal Oriented Optimization Technique for Tetrahedral Grids using a Continuous-Mesh Model, AIAA Scitech 2019 Forum,AIAA 2019-0349.

Conferences

- 15^{th} U.S. National Congress on Computational Mechanics (2019): Optimal Approximation Spaces for Petrov-Galerkin Schemes.

- 14^{th} WCCM-ECCOMAS (2021): A Discontinuous Petrov-Galerkin Scheme Using Optimized Approximation Spaces.
- SIAM CSE (2021): Optimal Approximation Spaces for Discontinuous Petrov-Galerkin Schemes via Goal-Oriented Mesh Adaptations.
- 16th U.S. National Congress on Computational Mechanics (2021): A Goal Oriented HP Adaptive DPG Method.

Contents

1 Introduction **8**

1.1 Singular perturbation problems 10

1.2 Mesh-adaptivity 12

1.3 Motivation: combining metric-based mesh adaptation with DPG methodology 14

2 Discontinuous Petrov-Galerkin Schemes **18**

2.1 Discretization for linear problems 20

2.2 Discretization for non-linear problems 22

2.2.1 Relaxation 23

2.2.2 Static condensation (normal equations) 24

2.3 Brief introduction to DPG-star method 25

2.4 Discrete Least Squares finite element methods 27

2.4.1 Relation between DPG and DLS finite element methods. 28

2.4.2 Static condensation (DLS) 29

3 Mesh Metric Duality **32**

3.1 Metric space 33

3.2 Metric-ellipse relation 34

3.3 Ellipse-simplex relation 36

3.4 Riemannian metric space 37

3.5 Continuous mesh model 38

4 Continuous model of energy error estimate for $h-$ and $hp-$adaptation 42

4.1 Anisotropy computation 43

4.2 Mesh density computation 47

4.2.1 Continuous model of energy error estimate for h adaptation 47

4.2.1.1 Goal oriented adaptation 51

4.2.1.2 Regularization 52

4.2.2 Continuous model of energy error estimate for hp adaptation 53

4.2.2.1 Polynomial selection 56

4.2.2.2 Mesh density computation 57

4.2.2.3 Goal oriented adaptation 59

4.2.2.4 Polynomial order transfer between meshes 59

5 Numerical Results: h adaptation 62

5.1 Test case I: boundary layer 62

5.1.1 Goal-oriented adaptation: opposing boundary layer 66

5.1.2 Goal-oriented adaptation: Gaussian peak 68

5.2 Test case II: inverse tangent - flux target 71

5.3 Test case III: interior line singularity 72

5.4 Test case IV: L-shaped domain 77

6 Numerical Results: hp adaptation 82

6.1 Test case I: boundary layer 82

6.1.1 Goal oriented adaptation: Gaussian peak 86

6.2 Test case II: inverse tangent - flux target 88

6.3 Test case III: L-shaped domain 89

7 Conclusion 94

7.1 Summary 94

7.2 Future Work 96

Bibliography **ii**

List of Figures

1.1.1 Oscillation in approximate solution of the 1D convection dominated convection-diffusion equation with decreasing diffusivity constant ϵ. 12

3.3.1 Ellipse circumscribing a triangle with (a) $C = 3$ and (b) $C = 1$. 37

3.4.1 Contour plot showing distance from the origin $(0.5, 0.5)$ of the domain for different metric fields. 38

3.5.1 Comparison of the Mesh and the metric field that produced the mesh. . . . 39

4.1.1 Bound on an arbitrary fourth order polynomial. 44

4.2.1 Numerical justification of assumption 4.2.1 using $P = 1$ 48

4.2.2 Numerical justification of assumption 4.2.1 using $P = 2$ 49

4.2.3 Numerical justification of assumption 4.2.1 using $P = 3$ 49

4.2.4 Stencil representing the first level neighbors for patchwise reconstruction. . 52

4.2.5 Patch around an element k over which local problems are solved. Edges marked by red represents the boundaries of patch with Dirichlet boundary conditions 56

5.1.1 Boundary layer: solution contour for two values of ϵ. 63

5.1.2 Convergence plots of (a) L^2 error in u_h and (b) energy error using standard V-norm ($\epsilon = 0.005$, expected order of convergence $(p + 1)$ is represented by the dashed lines). 64

5.1.3 Convergence plots of (a) L^2 error in u_h and (b) energy error using scaled V-norm ($\epsilon = 0.005$, expected order of convergence $(p + 1)$ is represented by the dashed lines). 65

5.1.4 Convergence plots of (a) L^2 error in u_h and (b) energy error with different initial meshes using scaled V-norm ($\epsilon = 0.005, p = 3$). 65

5.1.5 Solution and mesh after 10 adaptation cycles ($\epsilon = 0.005, Ne = 484, p = 3$) using scaled V-norm. 66

5.1.6 Comparison of $||U - U_h||_{E(\Omega)}$ and $E_c^\star$ for (a) $p = 1$, (b) $p = 2$ and (c) $p = 3$ at $\epsilon = 0.005$ using scaled V-norm. 67

5.1.7 Boundary layer test case: contour plot of the dual solution. $\bar{\eta}$ 68

5.1.8 Convergence plots for (a) error in target functional and (b) dual weighted residual using scaled V-norm ($\epsilon = 0.005$, expected order of convergence $(2p + 1)$ is represented by the dashed lines). 68

5.1.9 Adapated mesh for (a) $p = 2$ and (b) $p = 3$ after 15^{th} adaptation cycle using scaled V-norm ($\epsilon = 0.005$). 69

5.1.10 Boundary layer test case: contour plot for dual solution $\bar{\eta}$ 70

5.1.11 Convergence plots for (a) error in target functional and (b) dual weighted residual using scaled V-norm ($\epsilon = 0.005$, expected order of convergence $(2p + 1)$ is represented by the dashed lines). 70

5.1.12 Adapated mesh for (a) $p = 2$ and (b) $p = 3$ after 15^{th} adaptation cycle using scaled V-norm. 71

5.2.1 Convergence plots for (a) error in target functional and (b) dual weighted residual with regularization (section 4.2.1.2) using scaled V-norm (expected order of convergence $(2p + 1)$ is represented by the dashed lines). 73

5.2.2 Adapated mesh for (a) $p = 2$ and (b) $p = 3$ after 15^{th} adaptation cycle using scaled V-norm. 73

5.2.3 Oscillation in mesh during adaptation cycles without regularization using scaled V-norm ($p = 2$). 74

5.2.4 Adaptation cycles with regularization using scaled V-norm ($p = 2$). 74

5.2.5 Comparison of convergence plots for error in target functional between regularized and non regularized adaptation cycles using scaled V-norm and $p = 2$. 74

5.3.1 Convergence plots of (a) L^2 error in u_h and (b) energy error at $\gamma = 3.5$ using scaled V-norm (expected order of convergence is represented by the dashed lines). 75

5.3.2 Convergence plots for optimal γ_1 and suboptimal γ_2 of L^2 error in u_h using scaled V-norm (expected order of convergence is represented by the dashed lines). 76

5.3.3 Convergence plots for optimal γ_1 and suboptimal γ_2 of energy error using scaled V-norm. 76

5.3.4 Solution and mesh after 15 adaptation cycles ($\gamma = 2.0, Ne = 856, p = 2$) using scaled V-norm. 76

5.3.5 Solution and mesh after 15 adaptation cycles ($\gamma = 3.3, Ne = 925, p = 2$) using scaled V-norm. 77

5.4.1 Convergence plots of (a) L^2 error in u_h and (b) energy norm using scaled V-norm (expected order of convergence $(p + 1)$ is represented by the dashed lines). 78

5.4.2 Adapted mesh for (a) $p = 2$ and (b) $p = 3$. 79

5.4.2 Grading of meshes in L-shaped domain for (a) $p = 1$, (b) $p = 2$, (c) $p = 3$ and (d) convergence in σ (expected order of convergence $(p + 1)$ is represented by the dashed lines). 80

5.4.3 Comparison of condition number between DLS and DPG (normal equations) for (a) $p = 2$ and (b) $p = 3$ (expected order of growth is represented by the dashed lines). 80

6.1.1 Boundary layer: (a) solution contour on an adapted mesh and (b) polynomial distribution on the same adapted mesh with $\epsilon = 0.005$ with 6980 degrees of freedom. 83

6.1.2 Convergence plots of (a) L^2 error in u_h and (b) energy norm using scaled V-norm. 83

6.1.3 Evolution of (a) number of mesh elements and (b) average polynomial order with adaptations using scaled V-norm at a fixed cost $\mathscr{N} = 3072$ for different boundary layer strengths ϵ. 84

6.1.4 Convergence plots of (a) L^2 error in u_h and (b) energy norm using scaled V-norm with different initial polynomial distributions. 84

6.1.5 Convergence plots of (a) L^2 error in u_h and (b) energy norm using scaled V-norm with different initial mesh. 85

6.1.6 Contour showing $j_\Omega(\mathbf{x})$. 86

6.1.7 Convergence plots for (a) error in target functional and (b) dual weighted residual using scaled V-norm. 87

6.1.8 Polynomial distribution on an adapted mesh. 88

6.2.1 Polynomial distribution on an adapted mesh. 90

6.2.2 Convergence plots for (a) error in target functional and (b) dual weighted residual using scaled V-norm. 90

6.3.1 Convergence plots of (a) L^2 error in u_h and (b) energy norm using scaled V-norm. 91

6.3.2 Convergence plots of (a) L^∞ error, (b) H^1 seminorm error and (c) H^1 norm using scaled V-norm. 91

6.3.3 Zoomed-in view showing polynomial order with (a) Level 0, (b) Level 1, (c) Level 2 and (d) Level 3 magnification. 92

List of Tables

5.1 Adaptation Vs. error for constant complexity using scaled V-norm ($\epsilon = 0.005$, $p = 3$, $\mathscr{N} = \int_\Omega d(\mathbf{x})\, d\mathbf{x} = 512.0 \times \frac{3\sqrt{3}}{4}$). 64

5.2 Adaptation Vs. error for constant complexity using scaled V-norm ($p = 2$, $\mathscr{N} = \int_\Omega d(\mathbf{x})\, d\mathbf{x} = 512.0 \times \frac{3\sqrt{3}}{4}$). 70

6.1 Adaptation Vs. error for constant complexity using scaled V-norm ($\mathscr{N} = \int_\Omega w(\mathbf{x}) d(\mathbf{x})\, d\mathbf{x} = 3072.0$). 88

6.2 Adaptation Vs. error for constant complexity using scaled V-norm ($\mathscr{N} = \int_\Omega w(\mathbf{x}) d(\mathbf{x})\, d\mathbf{x} = 3072.0$). 93

This page is intentionally left blank.

Chapter 1

Introduction

Numerical discretization of partial differential equations (PDEs) in an efficient and stable manner has been essential in many industrial and academic applications. A broad range of physical processes in the real world, across various fields ranging from fluid mechanics to population dynamics, can be modeled with PDEs. In most cases, these PDEs do not have a closed-form or analytical solution, and thus, one has to implement methods to solve the PDE numerically. As such, there arises the need for a stable numerical method that can approximate the PDE's solution accurately.

In terms of stability and accuracy, over the last decade, Petrov-Galerkin finite-element schemes with optimal test functions (optimal PG schemes) have been a critical development [21, 24, 25, 28, 32, 70, 94, 100]. For a given weak formulation of the underlying PDE and an approximation space, this framework aims to automatically compute the space of test functions such that the scheme is stable. Thus, it can be perceived as adapting the test functions for a given weak formulation to achieve stability and accuracy, contrary to the traditional perspective of finite elements where trial and test functions are selected a priori. Due to its mathematical robustness, optimal PG schemes have already been utilized to solve numerous challenging problems, especially those pertaining to loss of numerical stability due to variations in equation parameters such as convection-dominated convection-diffusion problems [14, 15, 16, 20, 21, 29, 84] and other singularly perturbed problems [48, 49, 64, 83]. Moreover, the framework has inherent capacity for easy coupling with other numerical methods such as the boundary element method and various other finite element techniques [44, 46, 47, 50, 63, 65]. The robustness of the optimal PG framework has also been displayed by applying it to a broad range of problems such as

wave-propagation [31, 55, 88, 100]), elasticity [13, 17, 44, 45, 71] and transmissions through optical fibres [33, 61, 62] to name a few.

In this work, we will concern ourselves with discontinuous Petrov-Galerkin (DPG) methodology, which utilizes discontinuous test and trial spaces. Demkowicz and Gopalakrishnan first introduced the DPG methodology in [25, 28]. The method begins with the variational formulation of a linear PDE, which has an infinite-dimensional trial and test space. The formulation is assumed to be well-posed, i.e., bilinear form satisfies the inf-sup condition [5]. The core idea of this methodology is to find a discrete test space such that the best possible discrete inf-sup constant is achieved, thus ensuring numerical stability. It can also be seen as minimization of the operator residual of the variational formulation. The computation of the so-called optimal test space involves the inversion of the Riesz map [1]. However, this is generally impractical due to the infinite-dimensional nature of the test space. In practice, to achieve this inversion, an enriched finite-dimensional test space is employed over which the Riesz map is approximated and then inverted. This step is followed by an approximation of the optimal test space. These methods are also known as minimum-residual methods in the literature. Typically, the inversion of the Riesz map is a global process. However, on utilizing broken test spaces, this inversion can be done locally and thus, making the overall method computationally cheaper and practical in terms of implementation. Nevertheless, the introduction of discontinuous test spaces requires auxiliary variables (traces and fluxes) that live on the mesh skeleton.

Hence, the DPG methodology can generate numerical stability by decoupling the trial and test space. An enriched test space is proposed for any discretized trial space, which is then utilized to compute the near-optimal test space. This approach of achieving numerical stability is very flexible in terms of various choices of trial spaces compared to other finite-element methods where one has to perform rigorous analysis to show numerical stability, especially in mixed methods. Moreover, since the whole framework is automatic

[1] For a real Hilbert space, Riesz map represents the bijection between the Hilbert space and its dual space.

in achieving numerical stability, there is no need to introduce stabilization terms as many other stabilized finite-element methods do. Apart from this, due to the discontinuous nature of the trial and test spaces, higher-order piecewise polynomials as basis functions for trial and test spaces are more of a norm than an exception. Also, DPG provides a rich algebraic structure. The associated stiffness matrix is always symmetric positive definite due to which techniques such as conjugate-gradient method can be easily utilized for solving the associated linear system.

Many finite element techniques face loss of robustness while dealing with singularly perturbed problems. By loss of robustness, we mean the degradation in the quality of the solution with respect to problem parameters. In the context of fluid dynamics, this parameter is generally the Reynolds number (Re). For many relevant problems in fluid dynamics, Re is in the range of $10^6 - 10^7$, leading to vast differences in scales for the inviscid part (the convective flux) and the viscous part (diffusive flux). The viscous contribution scales with the inverse of Re. Although Navier-Stokes equations are a complex set of equations and their complete mathematical understanding is still a topic of ongoing research, this behavior can be well understood by looking at a very simple convection-dominated diffusion of a scalar quantity. We present a brief demonstration of this issue in the next section.

1.1 Singular perturbation problems

Standard numerical methods, in general, tend to perform poorly for a broad class of problems known as singular perturbation problems. These problems are characterized by either very large or very small equation parameters. Typically, these parameters tend to change the nature of the governing PDE in the limiting case (e.g., elliptic to hyperbolic). A well-studied example of this phenomenon is the convection-diffusion problem given by

$$-\epsilon\nabla^2 u(\mathbf{x}) - \beta \cdot \nabla u(\mathbf{x}) = s(\mathbf{x}) \qquad \forall \mathbf{x} \in \Omega,$$

where $\Omega \subset \mathbb{R}^d$ represents the computational domain in d-dimensions. The above PDE models the distribution of a scalar quantity u under convective and diffusive influences. ϵ denotes the diffusivity constant and $\beta \in \mathbb{R}^d$ represents the convection velocity. In the limit $\epsilon \to 0$, the elliptic part of the equation $\epsilon \nabla^2 u$ tends to disappear, and the equation is very close to changing its nature from a second-order elliptic PDE to a first-order hyperbolic PDE. This change of nature is visible while approximating the solution on a fixed mesh. As the nature of the PDE tends towards hyperbolicity, oscillation starts appearing in the solution. When applied to convection-dominated diffusion, standard finite element methods such as the Bubonov-Galerkin method tend to perform poorly with the degrading elliptic nature of the PDE. This degradation can be seen through the bound on the error in the finite element solution for the Bubonov-Galerkin method with $u \in H_0^1$:

$$\|u - u_h\|_{H^1(\Omega)} = C(\epsilon) \inf_{w_h \in H_0^1} \|u - w_h\|_{H^1(\Omega)}$$

where $C(\epsilon)$ grows with the decrease in ϵ. The dependence of C on ϵ is generally known as loss of robustness, i.e., as the singular perturbation parameter decreases, the finite element error becomes increasingly loosely bound by the best approximation error. We have demonstrated this by solving a one-dimensional convection-diffusion equation given by

$$\begin{aligned} -\epsilon u''(x) + u'(x) &= s(x) \qquad && \forall\, x \in [0,1], \\ u(0) = u(1) &= 0, \end{aligned}$$

where the source $s(x)$ is computed in such a way that the exact solution u is given by $x + \frac{e^{\frac{x}{\epsilon}} - 1}{1 - e^{\frac{1}{\epsilon}}}$. In fig. 1.1.1, we can observe the oscillation in the solution as the value of ϵ decreases. These oscillation will keep on growing as $\epsilon \to 0$ until they completely pollute the solution.

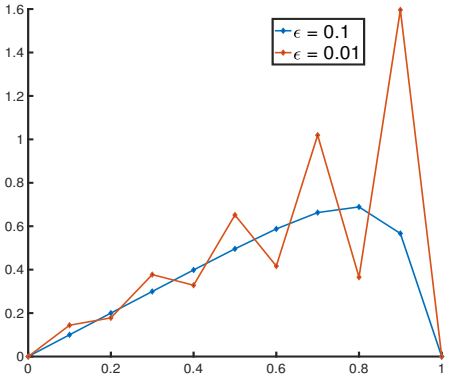


Figure 1.1.1: Oscillation in approximate solution of the 1D convection dominated convection-diffusion equation with decreasing diffusivity constant ϵ.

1.2 Mesh-adaptivity

As mentioned earlier, the DPG methodology separates the trial and test spaces while achieving numerical stability. Nevertheless, the quality of the approximate solution depends significantly on the approximation properties of basis function employed for the trial space. In most applications, polynomials are preferred for the basis functions such as Legendre polynomials or Jacobi polynomials. These polynomials are generated on reference finite elements and then mapped using affine or higher-order physical space transformation to the corresponding finite elements in the computational domain. This mapping depends upon the geometrical features of the finite element and plays an essential role in deciding the local approximation properties of the trial space for a given element. Another parameter that can be chosen while generating the basis function for the trial space is the order of the polynomials employed. It is directly connected to the capability of the trial space to capture the regularity of the solution to the continuous problem. Finally, the number of elements and their distribution in the computational domain play a dominant part in

deciding the associated computational cost and the resolution of the approximate solution.

There are multiple ways to enhance the capacity of the mesh and thus, in turn, the associated trial space to resolve the solution of the discretized PDE better. One naive way to achieve this is to perform uniform refinement. In uniform refinement, every element is split into multiple smaller elements without using any information from the computed solution. It results in a needless increase in the computational cost as elements might be added to sub-domains where the solution is smooth enough and does not require additional resolution. Another strategy is to refine the mesh using a refinement indicator isotropically. In most isotropic refinement strategies, the refinement indicator is computed for each element, then sorted. This is followed by refining a certain fraction of the elements according to the sorted values of the refinement indicator. This process of refining the element size is known as $h-$adaptation.

Furthermore, $h-$adaptation can be accompanied by varying the polynomial order of approximation used in each element. If the element size is kept constant throughout the mesh and only the order of approximation is allowed to vary, it is known as $p-$adaptation. When both adaptive procedures are performed simultaneously on the mesh, it is known as hp-adaptation. For any mesh-adaptation strategy, one must need a refinement indicator. Typically, refinement indicators are computed as a posteriori quantity using the approximate solution. Depending upon the goal of the computation, refinement indicators can measure the contribution of each element either in error in the solution or error in a specific target functional (such as drag in aerodynamic flows). However, even though isotropic refinements can reduce the error, as the name suggests, they do not take account of the directional features of the solution relative to the element during refinements. Hence, an essential question that arises here is how the shape of an element affects its capacity to resolve the solution? This question has motivated the development of anisotropic mesh adaptation algorithms.

In recent times, metric-based mesh optimization techniques have emerged as a promising paradigm for generating meshes composed of anisotropic simplexes [40, 76, 77, 89].

A significant advantage of these mesh optimization methods over conventional adaptive procedures, such as element subdivision and edge swapping, is the capability to globally optimize and regenerate the mesh using possibly very anisotropic elements rather than applying local modification. It has been long known that anisotropic meshes have substantial advantages, enabling the resolution of anisotropic solution features, such as thin boundary layers and shocks, due to skewed elements aligned along with the solution features [19, 74, 75, 79, 96, 99].

An essential aspect that many adaptation strategies lack is the capability to prescribe a predetermined number of elements in the mesh and then produce a near-optimal mesh with this constraint. Also, many adaptation strategies depend strongly on the initial mesh's quality. On the other hand, metric-based adaptation provides the capability to control the number of elements in the mesh by scaling the metric-field that drives the adaptation and, due to its re-meshing nature, its dependence on the initial mesh is minimal. Thus, providing a solution to both the issues of predetermined computational cost and dependence on the initial mesh.

1.3 Motivation: combining metric-based mesh adaptation with DPG methodology

One significant advantage of the DPG methodology is the presence of a residual-based inbuilt error estimator. Due to the presence of this inbuilt error estimator, DPG methodology with optimal test functions is tailor-made for mesh optimization [3, 32, 73], such that the optimal test space can be complemented with an optimal approximation space. This residual-based estimator is approximated by computing the norm of the dual of the residual. The dual of the residual is known as the error-representation function. Due to the discretization of the Riesz operator, this dual can be approximated by using the basis function of the same enriched test space that one utilizes to compute the approximate optimal test space. Since the norm of the error-representation function over

the element is an integral, the element's shape and its orientation with respect to this error-representation function are the parameters that control this integral. This interplay between the anisotropic features of the error-estimator and the anisotropy of the mesh elements has motivated the work presented in this dissertation. Precisely, the objective is to devise an adaptation strategy that utilizes the information about the anisotropy of the error-representation function to prescribe near-optimal shape and combine it with the isotropic strategy for element size computation under constraints.

Thus, in the present work, we propose and analyze a novel anisotropic mesh adaptation/optimization scheme based on the inbuilt error estimator of optimal PG schemes with discontinuous test and approximation spaces [28]. Our optimization framework uses the idea of analytic optimization [39, 40, 89] and local anisotropy computation [41]. In [89], the authors have proposed an analytic optimization technique using a L^q-norm error model. In the current work, we extend the idea of analytic optimization to the inbuilt error estimator accompanying discontinuous Petrov-Galerkin (DPG) schemes rather than using an extrinsic error model.

Typically, the problem of discrete mesh optimization can be formulated as

$$\min_{\forall T_h} E_{disc} \quad \text{such that} \quad N = const. \tag{1.1}$$

where E_{disc} is the error obtained from a suitable model, and N is the total degrees of freedom (DOFs) in the mesh. This optimization problem cannot easily be solved using analytic or semi-analytic tools. The question arises whether one can present a suitable reformulation of the above problem, which is continuous in nature. If so, then there are various analytic variational optimization techniques at our disposal. The continuous mesh approach allows us to formulate a suitable problem using a continuous analog of the discrete error estimator, which in turn transforms the discrete problem in problem 4.2.2 into a continuous one [76, 89]. While a mesh generated from such a continuous mesh model may not be provably optimal, it has been shown to be quite effective in terms of good approximation properties for higher-order methods such as hybridized discontinuous Galerkin schemes (HDG schemes) [89], Embedded Discontinuous Galerkin schemes (EDG

schemes) [42], and other high order finite element schemes [43].

In this work, the proposed continuous error model is based on the residual-based error estimate that accompanies the DPG framework i.e $E_{disc} = \|U - U_h\|_{E,\Omega}$ for solution-driven adaptations and a duality based error indicator using DPG-Star scheme [34, 69] for goal-oriented adaptations. Though we have focused on optimal PG schemes with discontinuous trial functions in this work, it can be extended to the optimal PG schemes with continuous trial functions as long as the error estimate can be localized and has a polynomial representation.

The thesis is organized as follows. In chapter 2, we provide a brief introduction to DPG schemes, ultra-weak formulation of a scalar convection-diffusion equation, and DPG-Star scheme. Next, chapter 3 recalls the mesh-metric duality; in chapter 4, we introduce the local optimization problem for computing the anisotropy and introduce the continuous optimization problem for the element size distribution. Finally, in chapter 5 and chapter 6, we provide numerical results that show the fidelity of the proposed method, followed by the conclusion and future prospects.

This page is intentionally left blank.

Chapter 2

Discontinuous Petrov-Galerkin Schemes

This chapter briefly describes Petrov-Galerkin schemes with optimal test functions and the associated inbuilt error estimator. We will restrict ourselves to a two-dimensional formulation. First, we present the framework that computes the optimal test functions (optimal test function framework) briefly and abstractly. We consider any inf-sup stable variational problem [5] :

$$u \in (\mathbb{X}, \|\cdot\|_{\mathbb{X}}): \qquad \langle \mathscr{L}u, v\rangle = \langle f, v\rangle, \qquad \forall\, v \in (\mathbb{V}, \|\cdot\|_{\mathbb{V}}),$$

where $\mathscr{L} : \mathbb{X} \to \mathbb{V}'$ is a continuous linear operator, $f \in \mathbb{V}'$, $\mathbb{X}$ and $\mathbb{V}$ are Hilbert spaces with $\mathbb{X}'$ and $\mathbb{V}'$ being the respective dual spaces. One way to characterize Petrov-Galerkin Schemes with optimal test functions [28] is the following: Given $\|\cdot\|_{\mathbb{V}}$ and $\|\cdot\|_{\mathbb{X}}$, they satisfy

$$\|\mathscr{L}u\|_{\mathbb{V}'} = \|u\|_E = \sup_{v\in\mathbb{V}} \frac{|\langle \mathscr{L}u, v\rangle|}{\|v\|_{\mathbb{V}}} \Rightarrow |\langle \mathscr{L}u, v\rangle| \leq \|u\|_E \|v\|_{\mathbb{V}}.$$

In a finite dimensional setting, given $\mathbb{X}_h \subset \mathbb{X}$, optimal PG schemes construct $\mathbb{V}_h^{opt} = \mathscr{R}_{\mathbb{V}}{}^{-1}\mathscr{L}(\mathbb{X}_h)$ such that

$$u_h \in \mathbb{X}_h: \qquad \langle \mathscr{L}u_h, v_h\rangle = \langle f, v_h\rangle \qquad \forall\, v_h \in \mathbb{V}_h^{opt},$$

inherits the well-posedness, and furthermore

$$\|u - u_h\|_E = \min_{v\in\mathbb{X}_h} \|u - v\|_E = \|\mathscr{L}u_h - f\|_{\mathbb{V}'} = \|\mathscr{R}_{\mathbb{V}}{}^{-1}(\mathscr{L}u_h - f)\|_{\mathbb{V}}, \tag{2.1}$$

where $\mathscr{R}_{\mathbb{V}} : \mathbb{V} \to \mathbb{V}'$ is the Riesz map. The Reisz map is defined by duality as $\langle \mathscr{R}_{\mathbb{V}} v, v'\rangle = (v, v')_{\mathbb{V}}\, \forall\, v' \in \mathbb{V}$. Since $\mathbb{V}$ is infinite-dimensional, it is not possible to explicitly compute $\mathscr{R}_{\mathbb{V}}{}^{-1}$. In order to deal with this issue, a choice of enriched space

$\mathbb{V}_r \subset \mathbb{V}\,(M = dim\,(\mathbb{V}_r) \geq dim\,(\mathbb{X}_h) = N)$ is employed over which the Riesz map is discretized and inverted [11, 12]. Since we are approximating the inverse of $\mathscr{R}_{\mathbb{V}}$, we have $\mathbb{V}_h$ which approximates $\mathbb{V}_h^{opt}$. It also implies $\mathbb{V}_h \subset \mathbb{V}_r \subset \mathbb{V}$ as $dim(\mathbb{V}_h) = N$. The discretization of the Riesz map is represented by a Gram matrix $\mathbb{G} \in \mathbb{R}^{M\times M}$ which is induced by the inner product on $\mathbb{V}$:

$$\mathbb{G}_{i,j} = (\psi_i, \psi_j)_{\mathbb{V}}. \tag{2.2}$$

Here ψ_i and ψ_j represents the basis of enriched space $\mathbb{V}_r$. The resulting discrete system on employing a finite dimensional basis for $\mathbb{X}_h$ and $\mathbb{V}_r$ (which we call the test search space) can be shown to be equivalent to the following linear system (normal equations) [94] :

$$B^T\mathbb{G}^{-1}B\hat{x}_h = B^T\mathbb{G}^{-1}l, \tag{2.3}$$

where $u_h = \sum_{i=1}^{N}(\hat{x}_h)_i\Phi_i$ is the discrete solution with Φ_i being the basis of the finite dimensional trial space $\mathbb{X}_h$ and $\hat{x}_h \in \mathbb{R}^N$. $B \in \mathbb{R}^{M\times N}$ is the enriched stiffness matrix where $B_{ij} = \langle \mathscr{L}\Phi_j, \psi_i\rangle$ and $l \in \mathbb{R}^M$ is the enriched load vector with $l_i = \langle f, \psi_i\rangle$. $\mathbb{G}$ might need a global assembly, if the test functions are continuous across the element boundaries. In the present work, we have employed an ultra-weak formulation with broken Sobolev spaces, which results in block diagonal structure of $\mathbb{G}$ and thus can be inverted locally for each element. However, this results in introduction of additional skeleton variables. Using static condensation, one can solve globally for the skeleton variables, similar to methods such as hybridisable discontinuous Galerkin schemes [97]. The residual based error estimator in eq. (2.1) can be approximated due to discretization of $\mathscr{R}_{\mathbb{V}}$ and is computed as follows:

$$\hat{y} = \mathbb{G}^{-1}\left(B\hat{x}_h - l\right), \tag{2.4}$$

where the dual of residual is approximated as $\varphi = \sum_{i=1}^{M}\hat{y}_i\psi_i$, and hence the error in energy norm is approximated as

$$\|u - u_h\|_E \approx \|\varphi\|_{\mathbb{V}}. \tag{2.5}$$

On employing polynomial basis functions for $\mathbb{V}_r$, we obtain a polynomial representation of φ (dual of the residual), which is also known as the error representation function.

Later, in section 4.1, we will show how this polynomial representation is paramount to the anisotropy computation required for the mesh adaptations.

2.1 Discretization for linear problems

In this section, we present the ultra-weak formulation of a linear scalar convection-diffusion problem. Let $\Omega \subset \mathbb{R}^2$ be a bounded connected Lipschitz domain and T_h be the triangulation of the domain Ω. ε_h^0 denotes the set of all internal edges and ε_h denotes the set of all edges. Consider the following function spaces

$$
\begin{aligned}
V_h &:= \{v \in L^2(\Omega) : v|_k \in P^p(k), \forall k \in T_h\},\\
\Sigma_h &:= \{\boldsymbol{\tau} \in (L^2(\Omega))^d : \boldsymbol{\tau}|_k \in (P^p(k))^d, \forall k \in T_h\},\\
\Lambda_h &:= \{\mu \in H^{\frac{1}{2}}(\varepsilon_h^0) : \mu|_e \in P^p(e), \forall e \in \varepsilon_h^0\},\\
\omega_h &:= \{\mu \in L^2(\varepsilon_h) : \mu|_e \in P^p(e), \forall e \in \varepsilon_h\},\\
V_{h,r} &:= \{v \in L^2(\Omega) : v|_k \in P^{p+\delta p}(k), \forall k \in T_h\},\\
\Sigma_{h,r} &:= \{\boldsymbol{\tau} \in (L^2(\Omega))^d : \boldsymbol{\tau}|_k \in (P^{p+\delta p}(k))^d, \forall k \in T_h\},
\end{aligned}
$$

where k represents an element, e represents an edge, $P^p(k)$ and $P^p(e)$ represents polynomials defined on elements and edges. δp represents the enrichment used for computing the approximate optimal test space. Furthermore, we define n_f as the number of faces in the triangulation. Next, consider a scalar second order convection-diffusion equation with Dirichlet boundary condition, constant diffusivity ϵ and constant convection along the direction $\boldsymbol{\beta} \in \mathbb{R}^2$ given by,

$$
\begin{aligned}
\nabla \cdot (\boldsymbol{\beta} u) - \epsilon \nabla^2 u &= s(\mathbf{x}), && \mathbf{x} \in \Omega \subset \mathbb{R}^d,\\
u &= g(\mathbf{x}), && \mathbf{x} \in \partial\Omega.
\end{aligned}
$$

Rewriting above system of equations as first order system:

$$\boldsymbol{\sigma} = \epsilon \nabla u, \qquad \mathbf{x} \in \Omega, \tag{2.6}$$

$$\nabla \cdot (\boldsymbol{\beta} u) - \nabla \cdot \boldsymbol{\sigma} = s(\mathbf{x}), \qquad \mathbf{x} \in \Omega, \tag{2.7}$$

$$u = g(\mathbf{x}), \qquad \mathbf{x} \in \partial\Omega. \tag{2.8}$$

We define the space $\mathbb{X}_h = \Sigma_h \times V_h \times \Lambda_h \times \omega_h$ and formulate the DPG method as a discontinuous Galerkin discretization of eq. (2.6) - eq. (2.8). Find $U_h = (\boldsymbol{\sigma}_h, u_h, \lambda_h, \hat{\sigma}_h) \in \mathbb{X}_h$ such that,

$$\frac{1}{\epsilon}\int_{T_h} \boldsymbol{\sigma}_h \cdot \boldsymbol{\tau}\, d\mathbf{x} + \int_{T_h} u_h \nabla \cdot \boldsymbol{\tau}\, d\mathbf{x} - \int_{\varepsilon_h^0} \lambda_h [\![\boldsymbol{\tau} \cdot \boldsymbol{n}_e]\!]\, ds + \int_{T_h} \boldsymbol{\sigma}_h \cdot \nabla v\, d\mathbf{x} - \int_{T_h} \boldsymbol{\beta} \cdot \nabla v\, u_h\, d\mathbf{x} -$$

$$\int_{\varepsilon_h} \hat{\sigma}_h [\![v]\!]\, ds + \int_{\varepsilon_h^0} |\boldsymbol{\beta} \cdot \boldsymbol{n}_e| \lambda_h [\![v]\!]\, ds = \int_{T_h} s(\mathbf{x}) v\, d\mathbf{x} - \int_{\partial T_h} \boldsymbol{\beta} \cdot \boldsymbol{n}_e g(\mathbf{x})\, d\mathbf{x} + \int_{\partial T_h} \boldsymbol{\tau} \cdot \boldsymbol{n}_e g(\mathbf{x})\, d\mathbf{x}$$

for all $(\boldsymbol{\tau}, v) \in \mathbb{V}_h$. $\mathbb{V}_h$ is the approximate optimal test space computed using the enriched test search space $\mathbb{V}_r = \Sigma_{h,r} \times V_{h,r}$ as mentioned previously. Also to reflect element by element computations, we use

$$\int_{T_h} \boldsymbol{\sigma}_h \cdot \boldsymbol{\tau}\, d\mathbf{x} = \sum_{k \in T_h} \int_k \boldsymbol{\sigma}_h \cdot \boldsymbol{\tau}\, d\mathbf{x}, \tag{2.9}$$

$$\int_{\varepsilon_h} \hat{\sigma}_h [\![v]\!]\, ds = \sum_{k \in T_h} \int_{\partial k} \hat{\sigma}_h\, sgn(\boldsymbol{n}_k)\, v\, ds, \tag{2.10}$$

$$\int_{\varepsilon_h^0} \lambda_h [\![\boldsymbol{\tau} \cdot \boldsymbol{n}_e]\!]\, ds = \sum_{k \in T_h} \int_{\partial k \setminus \partial\Omega} \lambda_h \boldsymbol{\tau} \cdot \boldsymbol{n}_k\, ds, \tag{2.11}$$

$$\int_{\varepsilon_h^0} |\boldsymbol{\beta} \cdot \boldsymbol{n}_e|\, \lambda_h [\![v]\!]\, ds = \sum_{k \in T_h} \int_{\partial k \setminus \partial\Omega} |\beta \cdot \boldsymbol{n}_k|\, sgn(\boldsymbol{n}_k)\, \lambda_h v\, ds. \tag{2.12}$$

In above variational formulation, each edge of the triangulation has a particular normal $\boldsymbol{n}_e$ associated with it. Hence

$$sgn(\boldsymbol{n}_k) = \begin{cases} 1 & \boldsymbol{n_k} = \boldsymbol{n}_e \\ -1 & \boldsymbol{n_k} = -\boldsymbol{n}_e \end{cases}, \tag{2.13}$$

where $\boldsymbol{n}_k$ is facing outward for an element $k \in T_h$. In case of ultra-weak variational formulation of diffusion and convection-diffusion equations, we have two components in the error representation function [28, 32, 94]. Thus, we represent the error representation function for convection-diffusion problem and diffusion problem as $(\psi_v, \boldsymbol{\psi_\tau})$. The algebraic structure of the linear system arising from eq. (2.12) has been already shown in eq. (2.3).

2.2 Discretization for non-linear problems

Next, we consider general non-linear (also system) convection-diffusion equations. In an abstract way, these equation can be represented as following:

$$\nabla \cdot (f_c(\boldsymbol{u}) - f_v(\boldsymbol{u}, \nabla \boldsymbol{u})) = s(\boldsymbol{u}, \nabla \boldsymbol{u}), \qquad \mathbf{x} \in \Omega, \tag{2.14}$$

with given diffusive and convective fluxes:

$$f_c : \mathbb{R}^m \to \mathbb{R}^{m \times d}, \qquad f_v : \mathbb{R}^m \times \mathbb{R}^{m \times d} \to \mathbb{R}^{m \times d}, \tag{2.15}$$

and a state dependent source term

$$s : \mathbb{R}^m \times \mathbb{R}^{m \times d} \to \mathbb{R}^m. \tag{2.16}$$

In above expression, d represents the spatial dimensions and the number of conservative variables is represented by m. Similar to the linear formulation, we will introduce an auxiliary variable representing the gradient of u as required by ultra-weak formulation. Thus, resulting in a first order system given by

$$\boldsymbol{\sigma} = \nabla \boldsymbol{u}, \qquad \mathbf{x} \in \Omega, \tag{2.17}$$

$$\nabla \cdot (f_c(\boldsymbol{u}) - f_v(\boldsymbol{u}, \boldsymbol{\sigma})) = s(\boldsymbol{u}, \boldsymbol{\sigma}), \qquad \mathbf{x} \in \Omega. \tag{2.18}$$

Next, we introduce few function spaces which will be used to compute the finite dimensional approximation of the solution and the approximate optimal test space. Here also, let $\Omega \subset \mathbb{R}^2$ be a bounded connected Lipschitz domain and T_h be the triangulation of the domain Ω. ε_h^0 denotes the set of all internal edges and ε_h denotes the set of all edges.

$$\begin{aligned}
V_h :=& \{v \in (L^2(\Omega))^m : v|_k \in (P^p(k))^m, \forall k \in T_h\},\\
\Sigma_h :=& \{\boldsymbol{\tau} \in (L^2(\Omega))^{m \times d} : \boldsymbol{\tau}|_k \in (P^p(k))^{m \times d}, \forall k \in T_h\},\\
\Lambda_h :=& \{\mu \in (H^{\frac{1}{2}}(\varepsilon_h^0))^m : \mu|_e \in (P^p(e))^m, \forall e \in \varepsilon_h^0\},\\
\omega_h :=& \{\mu \in L^2(\varepsilon_h)^m : \mu|_e \in (P^p(e))^m, \forall e \in \varepsilon_h\},\\
V_{h,r} :=& \{v \in (L^2(\Omega))^m : v|_k \in (P^{p+\delta p}(k))^m, \forall k \in T_h\},\\
\Sigma_{h,r} :=& \{\boldsymbol{\tau} \in (L^2(\Omega))^{m \times d} : \boldsymbol{\tau}|_k \in (P^{p+\delta p}(k))^{m \times d}, \forall k \in T_h\}.
\end{aligned}$$

In case of non-linear formulation, we will be using the classical discontinuous Galerkin discretization. Thus, on testing the strong form of the equations mentioned in eq. (2.17) and eq. (2.18) against $(\boldsymbol{\tau}, \boldsymbol{v}) \in \Sigma_{h,r} \times V_{h,r}$ and introducing auxiliary variables $\boldsymbol{f}_{n,h}$ and $\boldsymbol{\lambda}_h$ which approximates the non-linear normal flux and trace of u at each edge of the mesh, one obtains

$$\begin{aligned} 0 \;= \int_{T_h} \boldsymbol{\sigma}_h : \boldsymbol{\tau}\, d\mathbf{x} + \int_{T_h} \boldsymbol{u}_h \cdot \nabla \boldsymbol{\tau}\, d\mathbf{x} - \int_{T_h} \nabla \boldsymbol{v} : (f_c(\boldsymbol{u}_h) - f_v(\boldsymbol{u}_h, \boldsymbol{\sigma}_h))\, d\mathbf{x} - \int_{T_h} \boldsymbol{v} \cdot s(\boldsymbol{u}_h, \boldsymbol{\sigma}_h)\, d\mathbf{x} \\ + \int_{\epsilon_h} \boldsymbol{f}_{n,h} \cdot [\![v]\!]\, d\mathbf{x} - \int_{\epsilon_h^0} \boldsymbol{\lambda}_h \cdot [\![\boldsymbol{\tau} \cdot \boldsymbol{n}_e]\!]\, d\mathbf{x} - \mathscr{N}_{h,\delta\Omega}(\boldsymbol{\sigma}_h, \boldsymbol{u}_h, \boldsymbol{v}, \boldsymbol{\tau}), \end{aligned} \tag{2.19}$$

where $\mathscr{N}_{h,\delta\Omega}(\boldsymbol{\lambda}_h, \boldsymbol{f}_{n,h}, \boldsymbol{v}, \boldsymbol{\tau})$ represents the terms involving the boundary conditions for the non-linear problem.

2.2.1 Relaxation

For the sake of brevity, we define vectors

$$\boldsymbol{x}_h = (\boldsymbol{\sigma}_h, \boldsymbol{u}_h, \boldsymbol{\lambda}_h, \boldsymbol{f}_{n,h}), \quad \boldsymbol{y}_h = (\boldsymbol{v}, \boldsymbol{\tau}), \tag{2.20}$$

and abbreviate eq. (2.19) by

$$\mathscr{N}_{h,en}(\boldsymbol{x}_h, \boldsymbol{y}_h) = 0, \quad \forall \boldsymbol{y}_h \in \mathbb{V}_r = \Sigma_{h,r} \times V_{h,r}. \tag{2.21}$$

$\mathscr{N}_{h,en}$ represents the non-linear variational form for the problem obtained by testing the strong form against the test search space $\mathbb{V}_r$. In order to solve this non-linear problem, we have implemented damped Newton-Raphson method. To this end, starting from an initial guess $\boldsymbol{x}_h^0$, we linearize the problem to obtain the enriched stiffness matrix as shown below.

$$\int_{T_h} \boldsymbol{v} \frac{\delta \boldsymbol{u}_h}{\triangle\, t_k^n}\, d\mathbf{x} + \mathscr{N}_h^{'}[\boldsymbol{x}_h^n](\delta \boldsymbol{x}_h^n, \boldsymbol{y}_h) = -\mathscr{N}(\boldsymbol{x}_h^n, \boldsymbol{y}_h) \quad \forall \boldsymbol{y}_h \in \mathbb{V}_r, \tag{2.22}$$

for $\boldsymbol{x}_h^n \in \mathbb{X}_h = \Sigma_h \times V_h \times \Lambda_h \times \omega_h$. Once the enriched stiffness matrix is computed via the linear equation mentioned in eq. (2.22), we can compute the Gram matrix using the endowed inner product on $(\boldsymbol{v}, \boldsymbol{\tau})$ to produce the square linear system,

$$B^T \mathbb{G}^{-1} B \delta \boldsymbol{X}_h^n = B^T \mathbb{G}^{-1} l, \tag{2.23}$$

where B represents the enriched stiffness matrix, l represents the enriched load vector as computed using eq. (2.22) and $\delta \boldsymbol{X}_h^n$ represents the coefficients corresponding to $\delta \boldsymbol{x}_h^n$. The solution is updated in the usual way as

$$\boldsymbol{x}_h^{n+1} = \boldsymbol{x}_h^n + \delta \boldsymbol{x}_h^n, \tag{2.24}$$

where $\delta \boldsymbol{x}_h^n$ is computed using the coefficients $\delta \boldsymbol{X}_h^n$.

2.2.2 Static condensation (normal equations)

The linear system computed using eq. (2.23) constitutes a large system of equation. Since δu_h^n and $\delta \sigma_h^n$ are implicitly coupled via $\delta \lambda_h^n$ and $\delta f_{n,h}$ over the element boundaries, this coupling can be exploited to reduce the dimensionality of the linear system. We assume that $\delta \boldsymbol{X}_h^n = (\delta Q, \delta W, \delta \Lambda)$ are the basis coefficients corresponding to $\delta \boldsymbol{x}_h^n$. To simplify the notation, we represent the coefficients corresponding to $\delta \lambda_h^n$ and $\delta f_{n,h}^n$ by a single variable $\delta \Lambda$. Then, it can be implied that eq. (2.23) is equivalent to

$$\begin{bmatrix} A & B & R \\ C & D & S \\ L & M & N \end{bmatrix} \begin{bmatrix} \delta Q \\ \delta W \\ \delta \Lambda \end{bmatrix} = \begin{bmatrix} F \\ G \\ H \end{bmatrix}, \tag{2.25}$$

which can be split into

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \delta Q \\ \delta W \end{bmatrix} = \begin{bmatrix} F \\ G \end{bmatrix} - \begin{bmatrix} R \\ S \end{bmatrix} \delta \Lambda \tag{2.26}$$

and

$$\begin{bmatrix} L & M \end{bmatrix} \begin{bmatrix} \delta Q \\ \delta W \end{bmatrix} + N \delta \Lambda = H. \tag{2.27}$$

Equation (2.26) refers to local solution process to compute $\delta \sigma_h^n$ and δu_h^n. Since A, B, C and D are block diagonal and can be easily inverted, one can explicitly solve eq. (2.26) and substitute expression for $\begin{bmatrix} \delta Q & \delta W \end{bmatrix}^T$ in eq. (2.27) to solve for $\delta \lambda_h^n$ and $\delta f_{n,h}^n$ as shown in eq. (2.28):

$$\left(N - \begin{bmatrix} L & M \end{bmatrix} \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} \begin{bmatrix} R \\ S \end{bmatrix} \right) \delta \Lambda = H - \begin{bmatrix} L & M \end{bmatrix} \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} \begin{bmatrix} F \\ G \end{bmatrix}. \tag{2.28}$$

The matrix on the left hand side of eq. (2.28) is sparse $n_f \times n_f$ block matrix. The non-zero entries in each block is of the order $\mathcal{O}((2mp^{d-1})^2)$. Any block row will contain a dense diagonal block and $2d$ off-diagonal block entries (since, for a d-dimensional simplexes, each facet has $2d$ neighbouring facet). Hence, static condensation reduces the computational complexity of the linear solve by reducing the memory requirement and floating point operations by only utilizing the skeletal variables for the global solve. The static condensation is also implemented for the linear problems mentioned in section 2.1 to reduce the computational cost.

In order to solve the square system mentioned in eq. (2.28), we have utilized GMRES with a ILU(n) preconditioning available through PETSC library [7]. Also, we have utilized the finite element library called Netgen/Ngsolve [1] which offers geometry handling, mesh generation, quadrature rules and evaluation of basis functions for various finite elements.

2.3 Brief introduction to DPG-star method

In this section, we will provide a brief description of the DPG-Star method (DPG*) [34, 73] using the abstract problem defined at the begining of chapter 2. We have implemented DPG* for solving the dual problem to drive goal-oriented adaptations. DPG* has already been utilized to drive isotropic goal-oriented adaptations and an in-depth exposition can be found in [73]. Let the target functional be:

$$J(u) = (j_\Omega, u) + (j_{\partial\Omega}, Cu)_{\partial\Omega} = \int_\Omega j_\Omega u \, d\mathbf{x} + \int_{\partial\Omega} j_{\partial\Omega} Cu \, ds, \tag{2.29}$$

where $j_\Omega \in L^2(\Omega)$ and $j_{\partial\Omega} \in L^2(\partial\Omega)$ and C is a boundary operator on $\partial\Omega$. Let the strong form of the variational problem be given by the following linear problem:

$$Lu = s \quad \text{in}\,\Omega, \qquad Au = g \quad \text{on}\,\partial\Omega. \tag{2.30}$$

L is the linear differential operator on Ω, A is a linear differential boundary operator on $\partial\Omega$, $s \in L^2(\Omega)$ and $g \in L^2(\Omega)$. The continuous linear operator $\mathscr{L}$ and its dual $\mathscr{L}^\star$ is

generated by the bilinear form $b(u,v) = (Lu, v)$:

$$\langle \mathscr{L}u, v\rangle = \langle u, \mathscr{L}^{\star}v\rangle = b(u,v). \tag{2.31}$$

For our purpose, we will assume that the target functional in eq. (2.29) is compatible with the primal problem, i.e. we assume that dual problem satisfies the compatibility condition. The compatibility condition [58] for linear problems and functionals is shown in the following equation:

$$(Lu, z) + (Au, C^{\star}z)_{\partial\Omega} = (u, L^{\star}z) + (Cu, A^{\star}z)_{\partial\Omega}, \tag{2.32}$$

where $L^{\star}$, $A^{\star}$ and $C^{\star}$ are called the adjoint operators of L, A and C. With this compatibility condition, it can be noticed that for given L, A and C, only certain target functional are compatible. On assuming that the compatibility condition holds, the adjoint problem associated with the primal problem and the target functional mentioned in eq. (2.29) is given by:

$$L^{\star}z = j_{\Omega} \qquad \forall\, \mathbf{x} \in \Omega, \tag{2.33}$$

$$A^{\star}z = j_{\partial\Omega} \qquad \forall\, \mathbf{x} \in \partial\Omega. \tag{2.34}$$

The discrete DPG* can be stated as follows: Find $\hat{z}_h = \mathbb{G}^{-1}B\xi_h \in \mathbb{R}^M$ and $\xi_h \in \mathbb{R}^N$ such that

$$B^T\mathbb{G}^{-1}B\xi_h = J(w_h) \qquad \forall w_h \in \mathbb{X}_h. \tag{2.35}$$

The discrete dual solution is given by $z_h = \sum_{i=1}^{M}(\hat{z}_h)_i\psi_i$ where ψ_i represents the basis of $\mathbb{V}_r$. The dual solution z of the operator equation and associated boundary condition in eq. (2.33) and eq. (2.34) is approximated by z_h. For diffusion and convection-diffusion problems with ultra-weak formulation, z_h has two components as $z_h \in \mathbb{V}_h$ i.e. $z_h = (v_{h,z}, \boldsymbol{\tau}_{h,z})$. The error in target functional J is bounded by [73]:

$$|J(u-u_h)| \leq \|\mathscr{L}\|^{-1}\|\mathscr{L}u_h - l\|_{\mathbb{V}'}\|\mathscr{L}^{\star}z_h - J\|_{U'}. \tag{2.36}$$

Thus, a valid refinement indicator as already shown in [73] will be an approximation of $\|\mathscr{L}u_h - l\|_{\mathbb{V}'}\|\mathscr{L}^{\star}z_h - J\|_{U'}$ for each element. In eq. (2.36), one can replace the error in

residual of the dual problem ($\|\mathscr{L}^{\star} z_h - J\|_{U'}$) with an equivalent error estimator for DPG* method mentioned in Theorem 4.2 of [34] which we will reiterate next in eq. (2.37). Since, we are using DPG* to solve the dual problem mentioned in eq. (2.33) and eq. (2.34), we replace L with $L^{\star}$ in error estimator mentioned in [34].

$$(\eta^{\star})^2 = \sum_{k \in T_h} \left(\|L^{\star}(v_{h,z}, \boldsymbol{\tau}_{h,z}) - j_{\Omega}\|^2_{L^2(k)} \right) + \sum_{e \in \varepsilon_h^0} h_e \|[\![\boldsymbol{\tau}_{h,z} \cdot \boldsymbol{n}]\!]\|^2_{L^2(e)} + \sum_{e \in \varepsilon_h} {h_e}^{-1} \|[\![v_{h,z}]\!]\|^2_{L^2(e)} \tag{2.37}$$

Using the above error estimator, the authors in section 5.2 of [34] defined local refinement indicator $\eta_k^{\star}$ for each element $k \in T_h$ as shown in eq. (2.38).

$$\eta_k^{\star} = \left(\|L^{\star}(v_{h,z}, \boldsymbol{\tau}_{h,z}) - j_{\Omega}\|^2_{L^2(k)} + \sum_{e \in \partial k \setminus \partial\Omega} h_e \|[\![\boldsymbol{\tau}_{h,z} \cdot \boldsymbol{n}]\!]\|^2_{L^2(e)} + \sum_{e \in \partial k} {h_e}^{-1} \|[\![v_{h,z}]\!]\|^2_{L^2(e)} \right)^{\frac{1}{2}} \tag{2.38}$$

In eq. (2.37) and eq. (2.38), $L^{\star}$ is the adjoint differential operator, h_e is measure of size for edges in 2D and faces in 3D, $[\![v]\!] = v_{k^+}\boldsymbol{n}_{k^+} + v_{k^-}\boldsymbol{n}_{k^-}$ for internal edges and $[\![v]\!] = v_k \boldsymbol{n}_k$ for boundary edges. One crucial point to be noticed here is that the global stiffness matrix stays same for both the primal and dual problem. Since the adjoint solution exists in the approximate optimal test space ($\mathbb{V}_h$), to compute the dual weighted residual (DWR) [8] for error correction, one needs to project z_h to a richer space in order to overcome the Galerkin orthogonality. We achieve this by performing a patch-wise reconstruction using $\mathbb{V}_r$.

2.4 Discrete Least Squares finite element methods

DPG is widely implemented using the aforementioned normal equations (eq. (2.3)), but in some instances, the normal equations may lead to a very ill-conditioned linear system. For first-order formulations, the normal equations lead to a stiffness matrix whose condition number scales with $\mathscr{O}(h^{-2})$ where h is the element size. It is a major problem, especially for anisotropic adaptations, as the element size might degrade with subsequent adaptations. In order to cope with this situation, a discrete least square formulation (DLS) has been

proposed in [72]. In this section, we will be providing the details of the DLS Framework. These methods are based on the rich algebraic structure that DPG discretization possesses. In these methods, the normal equations obtained from DPG discretization are posed as an overdetermined system of equations that can be solved using QR-based algorithms. The significant advantage of these methods is having the condition number reduced by order of square root. This is very useful while solving ill-conditioned problems. Although ill-conditioning can arise due to various factors, in this work, we will be focusing on the ill-conditioning of the algebraic system due to significant variations in element size resulting from anisotropic mesh adaptations. DLS can be seen as discrete least-square minimization of a well-posed variational formulation. It is contrary to classical finite element formulations such as Bubonov-Galerkin, where one always solves a square linear system. Since DLS results in a linear system that has a condition number which is the square root of the condition number of the associated square linear system, it is possible to obtain a growth of the order of $\mathscr{O}(h^{-1})$ in condition number of the linear system.

2.4.1 Relation between DPG and DLS finite element methods.

We start with the normal equations (see eq. (2.3)) obtained from the classical DPG discretization:

$$B^T\mathbb{G}^{-1}B\hat{x}_h = B^T\mathbb{G}^{-1}l \Leftrightarrow A\hat{x}_h = F, \tag{2.39}$$

where $\mathbb{G} \in \mathbb{R}^{M\times M}$ is the gram matrix, $B \in \mathbb{R}^{M\times N}$ is rectangular as $M \geq N$, $A = B^T\mathbb{G}^{-1}B$ and $F = B^T\mathbb{G}^{-1}l$. Next, consider a factorization of $\mathbb{G}$ as $\mathbb{G} = LL^T$ for some $L \in \mathbb{R}^{M\times M}$. For $M \geq 1$, there can be infinite options for L. On introducing this factorization of $\mathbb{G}$ in eq. (2.39), one will obtain

$$\begin{aligned} B^TL^{-T}L^{-1}B\hat{x}_h = B^TL^{-T}L^{-1}l \Leftrightarrow &(L^{-1}B)^T(L^{-1}B)\hat{x}_h = (L^{-1}B)^TL^{-1}l \\ \Leftrightarrow \hat{x}_h = \underset{\hat{x}\in\mathbb{R}^N}{\operatorname{argmin}} &\|L^{-1}(B\hat{x} - l)\|_2^2, \end{aligned} \tag{2.40}$$

where minimization is done in euclidean norm in $\mathbb{R}^N$. Hence, it can be easily inferred that eq. (2.40) is equivalent to the problem mentioned in eq. (2.39). One way to chose

L is to perform Cholesky decomposition of $\mathbb{G}$. In this case, L will be a lower triangular matrix. Thus,

$$\hat{x}_h = \underset{\hat{x}\in\mathbb{R}^N}{\operatorname{argmin}} \|\tilde{B}\hat{x} - \tilde{l}\|_2^2, \quad \text{where} \quad \tilde{B} = L^{-1}B, \quad \tilde{l} = L^{-1}l. \tag{2.41}$$

The most common approach to solve the least-square problem is to implement QR-factorization algorithms such as Householder or Givens rotations as mentioned in [53].

Already shown in literature [93], the relative error in solution from a least-square QR solve is controlled by $cond(\tilde{B}) + \rho(\tilde{B}, \tilde{l}) \cdot cond(\tilde{B})^2$, where

$$\rho(\tilde{B}, \tilde{l}) = \frac{\|\tilde{B}\hat{x} - \tilde{l}\|_2}{\|\tilde{B}\|_2 \|\hat{x}\|_2}. \tag{2.42}$$

Since, it is assumed that the variational problem is well-posed, the residual in the numerator of the above expression is expected to converge to zero asymptotically as the mesh is refined. It is important to notice that $cond(\tilde{B}) = cond(A)^{1/2}$ as $A = \tilde{B}^T\tilde{B}$. It has already been proved that for many DPG discretization and least-squares method [9, 54], $cond(A)$ grows as $\mathcal{O}(h^{-2})$. Hence, it is very pragmatic to assume that $cond(\tilde{B})$ will grow as $\mathcal{O}(h^{-1})$ and if the residual converges to zero as mentioned above, then the term $\rho(\tilde{B}, \tilde{l}) \cdot cond(\tilde{B})^2$ will have an insignificant contribution. In such a scenario, the numerical stability of the least-square solution only depend upon $cond(\tilde{B}) \approx)\mathcal{O}(h^{-1})$. It will result in more accurate solution than using normal equations and thus, justifying the necessity to employ the discrete least-square formulation for ill-conditioned problems.

2.4.2 Static condensation (DLS)

Analogous to normal equations, the DLS framework also needs static condensation to reduce the computational complexity of solving the over-determined problem. Hence, in this section, we present the details of the static condensation for the DLS framework. Similar to the normal equations, the internal variables are eliminated from the linear system via a process similar to Schur-complement but in a framework of discrete least squares. The first step is to separate the contributions of internal variables and interface

variables. We separate the solution vector $\hat{x}_h$ as $\hat{x}_h = \begin{bmatrix} \hat{x}_i \\ \hat{x}_f \end{bmatrix}$ where $\hat{x}_i$ represents the internal variable (u_h and σ_h) and $\hat{x}_f$ represents the interface variables (λ_h and $\hat{\sigma}_h$). The stiffness matrix $L^{-1}B = \tilde{B} = \begin{bmatrix} \tilde{B}_i & \tilde{B}_f \end{bmatrix}$ is also separated into contributions due to internal variables and interface variables. On substituting these expressions into normal equations (see eq. (2.3)), we have

$$\begin{bmatrix} \tilde{B}_i^T \tilde{B}_i & \tilde{B}_i^T \tilde{B}_f \\ \tilde{B}_f^T \tilde{B}_i & \tilde{B}_f^T \tilde{B}_f \end{bmatrix} \begin{bmatrix} \hat{x}_i \\ \hat{x}_f \end{bmatrix} = \tilde{B}^T \tilde{B} \hat{x}_h = \tilde{B}^T \tilde{l} = \begin{bmatrix} \tilde{B}_i^T \tilde{l} \\ \tilde{B}_f^T \tilde{l} \end{bmatrix}. \tag{2.43}$$

Next, the Schur complement of $\hat{x}_f$ is given by

$$\tilde{B}_f^T (\boldsymbol{I} - P_i) \tilde{B}_f \hat{x}_f = (\tilde{B}_f^T \tilde{B}_f - \tilde{B}_f^T \tilde{B}_i (\tilde{B}_i^T \tilde{B}_i)^{-1} \tilde{B}_i^T \tilde{B}_f) \hat{x}_f \tag{2.44}$$

$$= \tilde{B}_f^T \tilde{l} - \tilde{B}_f^T \tilde{B}_i (\tilde{B}_i^T \tilde{B}_i)^{-1} \tilde{B}_i^T \tilde{l} = \tilde{B}_f^T (\boldsymbol{I} - P_i) \tilde{l}, \tag{2.45}$$

where $P_i = \tilde{B}_i (\tilde{B}_i^T \tilde{B}_i)^{-1} \tilde{B}_i^T$ and $\boldsymbol{I}$ is an identity matrix. It can be easily observed that $P_i^T = P_i$ and P_i is an orthogonal projection as $P_i^2 = P_i$. Thus, one can verify that $(\boldsymbol{I} - P_i) = (\boldsymbol{I} - P_i)^2$ and

$$\Big((\boldsymbol{I} - P_i) \tilde{B}_f \Big)^T (\boldsymbol{I} - P_i) \tilde{B}_f \hat{x}_f = \Big((\boldsymbol{I} - P_i) \tilde{B}_f \Big)^T (\boldsymbol{I} - P_i) \tilde{l}. \tag{2.46}$$

In eq. (2.46), the equivalence between a square system and an overdetermined system can be noted and expressed as a discrete least-squares problem:

$$\hat{x}_f = \underset{\hat{x} \in \mathbb{R}^{N_f}}{\operatorname{argmin}} \| (\boldsymbol{I} - P_i)(\tilde{B}_f \hat{x} - \tilde{l}) \|_2^2, \tag{2.47}$$

where N_f is the dimension of the finite dimensional space in which the interface variables lies. Once, the interface components are computed, the interior variables can be computed via local computations for each element using

$$\hat{x}_i = (\tilde{B}_i^T \tilde{B}_i)^{-1} \tilde{B}_i^T (\tilde{l} - \tilde{B}_f \hat{x}_f). \tag{2.48}$$

One significant issue that arises while computing $\hat{x}_i$ is the computation of $(\tilde{B}_i^T \tilde{B}_i)^{-1}$. If this quantity is not computed carefully, it might affect the global conditioning of QR

based methods and the conditioning would not be controlled as predicted by eq. (2.42). To solve this issue, the authors in [72] utilises full QR decomposition of $\tilde{B}_i$, which can be computed in a stable manner as:

$$\tilde{B}_i = \begin{bmatrix} Q_i & Q_f \end{bmatrix} \begin{bmatrix} R_i \\ 0 \end{bmatrix}, \tag{2.49}$$

where $\begin{bmatrix} Q_i & Q_f \end{bmatrix}$ is a unitary matrix and R_i is a upper triangular matrix. From this, it follows that

$$P_i = Q_i Q_i^T \qquad \text{and} \qquad (\tilde{B}_i^T \tilde{B}_i)^{-1} \tilde{B}_i^T = R_i^{-1} Q_i^T. \tag{2.50}$$

As mentioned earlier, these computations can be done locally for each element in the triangulation.

Chapter 3

Mesh Metric Duality

In the previous chapter, we discussed the DPG discretization technique for linear and non-linear problems. DPG schemes compute an optimal test space for a chosen finite-dimensional approximation space. The accuracy of the approximate solution depends not only on the stabilization provided by the DPG methodology but also on the quality of the approximation space that has been chosen. Typically, the regularity of the chosen basis for the approximation space depends upon the chosen discretization scheme and the governing PDE. However, there is still much opportunity to improve the quality of the solution by improving the quality of the mesh and order of approximation $\boldsymbol{p}$. Although a mesh can be composed of several types of elements, we will only focus on meshes composed of simplices in this work.

The key ingredient in generating an optimized approximation space is to formulate a mapping from the error in the solution or quantity of interest (q.o.i) to the geometrical properties of the mesh element. If we can formulate such a mapping, it is reasonable to expect that this mapping will empower us to minimize the total approximation error by modifying the size and shape of the elements ($h-$ adaptivity). Furthermore, variations in local order of approximation $\boldsymbol{p}_k$ can accompany the geometrical modifications leading to $hp-$adaptivity. Adpatations will lead to a sequence of meshes composed of anisotropic elements. [78, 86, 87] contains some of the initial work in this direction.

The initial work incorporating skewed elements while performing Delaunay triangulation was proposed in [80]. In [80], the length and orientation of each edge of an element is prescribed via a metric tensor introduced in [51]. This idea has been successfully exploited for various low order methods in the past few decades [18, 57, 60]. Although for low-order

methods, a large number of error models have been developed to incorporate metric-based anisotropic adaptation over the past few decades, efforts towards developing error models for higher-order methods have been relatively recent [35, 36, 38, 40, 42, 89, 98]. Many of these techniques utilize an extrinsic error model to minimize the interpolation error in certain norms (E.g., L^q or H^1). In this work, we solely focus on generating a map between the inbuilt residual-based error estimator that accompanies the PG schemes with optimal test functions and geometric properties of mesh elements to drive the mesh adaptations. Thus, eliminating any dependence on extrinsic error models.

In this chapter, we first present the basics of metric spaces. Then, we present the relationship between a continuous metric space and its discrete analog and a few essential properties of the metric conforming mesh generators.

3.1 Metric space

. In this section, we will briefly provide an overview of metric spaces. First, we consider the measure of length in canonical Euclidean space. The dot product in Euclidean space is given by

$$(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{D} x_i y_i, \quad \mathbf{x} \in \mathbb{R}^D, \mathbf{y} \in \mathbb{R}^D. \tag{3.1}$$

This notion of dot product can be generalized by introducing a symmetric positive-definite matrix $\mathscr{M} \in \mathbb{R}^{D\times D}$, known as the metric tensor. This generalized notion of dot product can be written as

$$(\mathbf{x}, \mathbf{y})_{\mathscr{M}} = (\mathbf{x}, \mathscr{M}\mathbf{y}) \,. \tag{3.2}$$

Furthermore, using the definition of the dot product, a suitable norm can be defined as

$$\|\mathbf{x}\|_{\mathscr{M}} = \sqrt{(\mathbf{x}, \mathscr{M}\mathbf{x})}. \tag{3.3}$$

In the case of D-dimensional Euclidean space, $\mathscr{M}$ is an identity matrix in $\mathbb{R}^{D\times D}$. In a broader sense, a metric space is defined as a non-empty set with a metric defined on it. This metric is a function that defines the concept of distance between any two members in

the set. Keeping this in mind, if the canonical Euclidean space is equipped with the dot product and norm mentioned in eq. (3.2) and eq. (3.3), it will still be a metric space [1].

Equivalently, the measure of the area or volume also differs from the one measured in canonical Euclidean space. For any closed bounded domain $\Omega \in \mathbb{R}^D$, the volume measured using the Euclidean norm and norm induced by $\mathscr{M}$ are related by

$$|\Omega|_{\mathscr{M}} = \sqrt{det(\mathscr{M})}|\Omega|_{\mathbf{I}}.$$

$|\Omega|_{\mathbf{I}}$ represents the volume of Ω in canonical Euclidean space. The measure of the angle between two vectors is also redefined, but this modification is not very pertinent for our purpose of developing adaptation algorithms. Related in-depth exposition can be found in [2, 76, 77].

3.2 Metric-ellipse relation

Next, we briefly show how a metric tensor $\mathscr{M}$ can represent an ellipsoid. This relation between the metric tensor and the ellipsoid is fundamental to the proposed adaptive algorithms. To begin with, let us consider a unit ball $\mathbb{B}$ centered at origin represented by

$$\mathbb{B} = \{\mathbf{x} \in \mathbb{R}^D : \|\mathbf{x}\|_{\mathbf{I}} \leq 1\}.$$

Every point that lies within B has a Euclidean distance ($\|\mathbf{x}\|_{\mathbf{I}}$) less than 1 from the center. The Euclidean measure can be replaced by the norm induced by $\mathscr{M}$ and will lead to a bounded domain given by

$$\mathbb{E} = \{\mathbf{x} \in \mathbb{R}^D : \|\mathbf{x}\|_{\mathscr{M}} \leq 1\}.$$

Next, lets consider the spectral decomposition of $\mathscr{M} = \mathbf{Q}^T\Gamma\mathbf{Q}$. $\mathbf{Q}$ and Γ represents $\mathbb{R}^{D\times D}$ matrices containing eigenvectors and eigenvalues. If $\mathbf{r}_i$ represents the i^{th} eigenvector

[1] In the present work, we will always refer to the Euclidean space equipped with metric tensor as metric space

and α_i represents i^{th} eigenvalue, then $\mathbf{Q}$ and Γ are given by

$$\mathbf{Q} = [\mathbf{r}_1\mathbf{r}_2\dots\mathbf{r}_D] \quad \Gamma = \begin{bmatrix} \alpha_1 & & \\ & \ddots & \\ & & \alpha_D \end{bmatrix}.$$

It is assumed that the eigenvalues α_i are arranged in an increasing order i.e. $\alpha_1 \leq \alpha_2 \leq \dots \alpha_D$. Since $\mathscr{M}$ is a SPD matrix, we have $\alpha_i > 0 \, \forall i \in [1, 2, \dots D]$.

On Substituting the spectral decomposition of $\mathscr{M}$ in the definition of the norm induced by $\mathscr{M}$(eq. (3.3)), one will have

$$\|\mathbf{x}\|_{\mathscr{M}}{}^2 = \mathbf{x}.\mathscr{M}\mathbf{x} = \mathbf{x}^T\mathbf{Q}^T\Gamma\mathbf{Q}\mathbf{x} = \mathbf{w}^T\Gamma\mathbf{w} = \sum_{i=1}^{D} w_i^2\alpha_i \leq 1. \tag{3.4}$$

In above expression, we have used $\mathbf{w} = Q\mathbf{x}$. Since, $\alpha_i > 0$, we can express α_i as $\frac{1}{h_i^2}$ where $h_i > 0$. On substituting $\alpha_i = \frac{1}{h_i^2}$ in eq. (3.4), we can rewrite eq. (3.4) as

$$\sum_{i=1}^{D} \left(\frac{w_i}{h_i}\right)^2 \leq 1. \tag{3.5}$$

Equation (3.5) represents the canonical ellipsoid in $\mathbb{R}^D$ which is centered at the origin, the length of principal axes are given by h_i, and the directions of principal axes are represented by the eigenvectors $\mathbf{r}_i$. Since α_is are assumed to be ordered in increasing order, h_is are ordered in decreasing order. Instead of using the axis lengths, the metric tensor can be parameterized using $D-1$ aspect ratios defined as

$$\beta_i = \frac{h_1}{h_{i+1}} \quad \forall i = 1, 2, \dots, (D-1),$$

and a quantity called density d, given by

$$d = \frac{1}{\prod_{i=1}^{D} h_i}.$$

In the case of two-dimensional ellipsoid, i.e., ellipse, the spectral decomposition is given by

$$\mathscr{M} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}^T \begin{bmatrix} \alpha_1 & 0 \\ 0 & \alpha_2 \end{bmatrix} \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix},$$

the aspect ratio is given by

$$\beta = \frac{h_1}{h_2},$$

and the density is given by

$$d = \frac{1}{h_1 h_2}. \tag{3.6}$$

3.3 Ellipse-simplex relation

The previous section demonstrated the relation between the ellipse and the metric tensor. This bijective relation allows one to control the ellipse via the metric tensor. Since we will be employing meshes comprising of triangles, we need a relationship between the triangle and the ellipse to manipulate the triangulation.

This section will introduce the relationship between the ellipse and the mesh element. Let us consider a triangle whose vertices lie on the ellipse's boundary and let $e_k \, \forall \, k = 1, 2, 3$ denote the edges of the triangle k, specified as vectors. It can be easily inferred that infinitely many triangles exist whose vertices lie on the ellipse. Thus, in order to reduce the number of possible candidates, we will only concern ourselves with a subset of triangles that are equilateral under the norm induced by the metric tensor $\mathscr{M}$ i.e.

$$e_k^T \mathscr{M} e_k = C, \quad k = 1, 2, 3 \quad \text{and} \quad C \in \mathbb{R}$$

under the condition of maximizing the area of the element ($|k|$). These mesh elements are also known as *unit* elements with respect to the metric. Applying this constraint, the number of candidate triangles gets reduced to four possible choices in 2D. In three dimensions, there will be 24 choices. Still, it is a non-trivial selection procedure among these candidates. Criterion such as preferring acute-angled elements over obtuse angle elements or measure of size or perimeter can be used as a differentiating factor. Acute angle elements are generally preferred over obtuse angle elements due to the former's ability to provide super-accuracy for low-order interpolation, and finite elements [92].

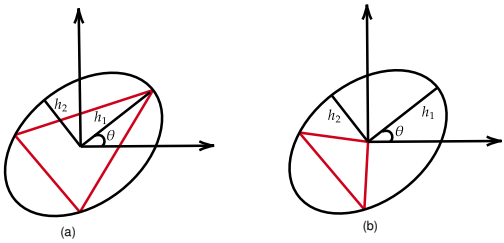


Figure 3.3.1: Ellipse circumscribing a triangle with (a) $C = 3$ and (b) $C = 1$.

In fig. 3.3.1, two different elements inscribed in ellipse are shown, with both being acute-angled elements with different values of C. The following relation relates the area of the triangle and its circumscribing ellipse:

$$|k| = \alpha h_1 h_2 = \frac{\alpha}{d}. \tag{3.7}$$

In eq. (3.7), α and C are related as follows

$$\alpha = \frac{\sqrt{3}}{4} C. \tag{3.8}$$

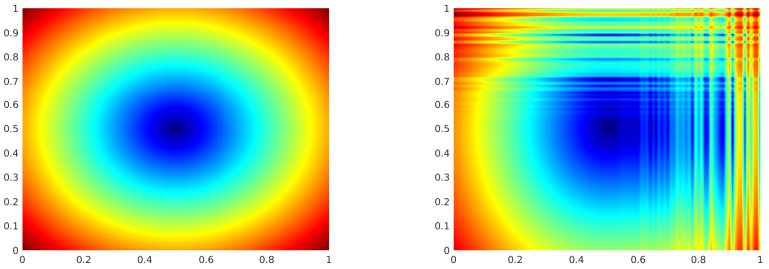
3.4 Riemannian metric space

In previous sections, $\mathscr{M}$ is considered to be a constant throughout the domain. Nevertheless, the metric can be a varying function. In the case of Riemannian metric space, the metric tensor varies smoothly over the whole domain. Every parameter that metric tensor depends upon and vice-versa also varies across the domain. Thus, the definition of distance and area under the variable metric tensor needs to be redefined. For continuous metric tensor field, the length of an edge $e = \mathbf{x}_2 - \mathbf{x}_1$ is redefined as

$$\|e\|_{\mathscr{M}} = \int_0^1 \sqrt{\boldsymbol{e}^T \mathscr{M}(\mathbf{x}_1 + t\boldsymbol{e})\boldsymbol{e}}\, dt. \tag{3.9}$$

This redefinition of length can be found in [85]. The proposed adaptive algorithms in this work aim to generate an optimal metric field $\mathscr{M}$. In fig. 3.4.1, we present the variation

of distance measured from the center of the domain located at $(0.5, 0.5)$ with respect to different metric tensors.



(a) $\mathscr{M} = \mathbf{I}$ **(b)** $\mathscr{M} = \mathscr{M}(\mathbf{x})$

Figure 3.4.1: Contour plot showing distance from the origin $(0.5, 0.5)$ of the domain for different metric fields.

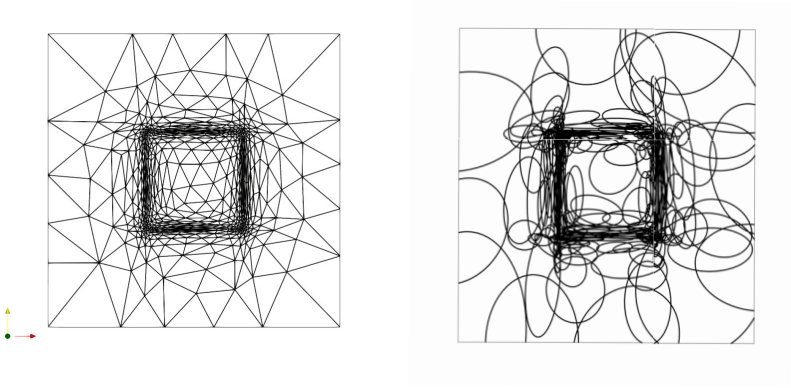
3.5 Continuous mesh model

In previous sections, we provided an overview of Euclidean metric spaces and Riemannian metric spaces without emphasizing their utility. This section presents the concepts that guide defining a mesh via a metric field and pave the way for developing optimizing strategies that employ analytical methods. The target for optimizing strategies is to minimize the error estimate E at a fixed total cost. The cost can be measured in terms of DOFs associated with the mesh. The error estimate E can be a measure of the error in solution variables in an appropriate norm or error in a target functional. This problem can be framed as

$$\min_{\forall T_h} E \quad \text{such that} \quad N = KNe, \tag{3.10}$$

where N represents the number of DOFs, Ne is the number of elements in the mesh, and K is a proportionality constant connecting N and Ne. In a discrete setting, the optimization problem presented in eq. (3.10) is intractable by analytic tools apart from very few trivial cases. Since the bijective relation between the mesh element k and the metric

tensor $\mathscr{M}$ only holds locally, it is a valid assumption to treat the metric tensor associated with each element in the mesh as a local snapshot of a continuous metric field. This equivalence allows us to construct optimal meshes via manipulating the continuous metric field $\mathscr{M}(\mathbf{x})$. The continuous nature of the equivalent problem allows us to use analytical tools such as the calculus of variations to obtain the optimal metric field. In fig. 3.5.1, we present an adapted mesh and the metric field that drove the adaptation.



(a) Mesh T_h **(b)** Metric field $\mathscr{M} = \mathscr{M}(x)$

Figure 3.5.1: Comparison of the Mesh and the metric field that produced the mesh.

Next, let us assume that we have a procedure at our disposal through which we can generate the optimal metric field $\mathscr{M}(\mathbf{x})$. We could use this metric field as an input for a metric-conforming mesh generator. While generating a single element that conforms to a metric tensor is possible, constructing a mesh completely comprising of conforming elements cannot be guaranteed [76, 77]. Thus, a metric conforming mesh generator attempts to produce a mesh conforming to a continuous metric field ($\mathscr{M}_{int}(\mathbf{x})$) in a least-squares sense. $\mathscr{M}_{int}(\mathbf{x})$ represents the suitable continuous interpolation of the discrete metric field. Typically, due to the limitations towards producing a space-filling metric-conforming mesh, metric conformity is always lost. The loss of metric-conformity would not allow us to construct provably optimal meshes. Even though the uniqueness of a metric conforming

mesh cannot be guaranteed, the approximation properties of such meshes are similar [76].

As mentioned earlier, optimizing mesh via a continuous metric field enables us to utilize tools like the calculus of variations. It motivates us to connect the approximation properties of the solution space to the continuous metric field $\mathscr{M}(\mathbf{x})$ instead of a discrete mesh. In order to achieve this connection, we need to define a continuous analogue of d in (eq. (3.6)) as $d(\mathbf{x})$. We refer $d(\mathbf{x})$ as continuous density based on continuous metric field $\mathscr{M}(\mathbf{x})$ and can be viewed as a continuous analog of the inverse of the mesh size. Next, we define a quantity known as mesh-complexity $\mathscr{N}$ and is given by

$$\mathscr{N} = \int_{\Omega} \sqrt{det(\mathscr{M}(\mathbf{x}))}\, d\mathbf{x}. \tag{3.11}$$

$\mathscr{N}$ can be viewed as a continuous analog of total DOFs in eq. (3.10). Now, if we replace the discrete error estimate with a continuous analog, we will have all the ingredients required to formulate a continuous analog of the optimization problem presented in eq. (3.10). In chapter 4, we will introduce the continuous analogue(E_c) of the discrete error estimate. The continuous equivalent of eq. (3.10) is given by

$$\min_{\forall \mathscr{M}(\mathbf{x})} E_c \quad \text{such that} \quad \mathscr{N} = const. \tag{3.12}$$

This page is intentionally left blank.

Chapter 4

Continuous model of energy error estimate for $h-$ and $hp-$adaptation

From chapter 2, it can be inferred that the accuracy of the discrete solution u_h also depends upon the trial basis we chose, whose approximation properties also depend upon the mesh T_h. In chapter 3, we provide a brief overview of continuous mesh models, Riemannian metric fields, and how a mesh and metric field can be seen as counterparts. This chapter provides the methodology for computing the optimal parameters for the metric field to generate meshes that will induce near-optimal approximation spaces using the inbuilt error estimator of DPG discretizations. Thus, we couple DPG schemes with a continuous mesh model, achieving this thesis's primary goal. As already mentioned in chapter 3, a triangle can be represented by three parameters, namely aspect ratio β, orientation θ, and mesh density d. The proposed adaptation methodology is a two-step process:

(a) computing the anisotropy parameters (θ, β).

(b) computing the mesh density (d).

A polynomial order selection procedure (see section 4.2.2.1) accompanies the computation of mesh density for $hp-$adaptations. The anisotropy computations are done for each element in the mesh, whereas the density computation is done under the constraint of constant complexity (see section 4.2.1 and section 4.2.2) over the whole domain. The density computation is analytical for $h-$adaptation and semi-analytical for $hp-$adaptation. Since the complexity of a mesh is the continuous analog of the degrees of freedom associated with the mesh, having a constraint on it constraints the computational cost associated with the discretization.

4.1 Anisotropy computation

In this section, we provide the details of the anisotropy computation. As mentioned in chapter 2, DPG methods with optimal test functions have an inbuilt residual-based error estimator:

$$\|U - U_h\|^2_{E,T_h} \approx \|(\psi_v, \boldsymbol{\psi_\tau})\|^2_{\mathbb{V},T_h} := \sum_{k \in T_h} \|(\psi_v, \boldsymbol{\psi_\tau})\|^2_{\mathbb{V},k},$$

where $U_h \in \mathbb{X}_h$ represents the solution of the convection-diffusion or diffusion problem and $(\psi_v, \boldsymbol{\psi_\tau})$ are the associated error representation functions. It has been already shown in [20, 26, 29] that one can control the equivalence of the energy norm with the norm of their choice by varying the norm associated with the test space. In this section, we employ the natural inner product associated with $H^1(T_h) \times H(div; T_h)$ which we call the standard V-norm and a scaled version of the V-norm. For a scalar convection-diffusion problem and diffusion problem with ultra-weak formulation, $\psi_v \in H^1(T_h)$ and $\boldsymbol{\psi_\tau} \in H(div; T_h)$, the standard V-norm for $k \in T_h$ is given by:

$$\begin{aligned}\|(\psi_v, \boldsymbol{\psi_\tau})\|^2_{\mathbb{V},k} &= \|\psi_v\|^2 + \|\nabla\psi_v\|^2 + \|\boldsymbol{\psi_\tau}\|^2 + \|\nabla \cdot \boldsymbol{\psi_\tau}\|^2 \\ &= \int_k \underbrace{(\psi_v(\mathbf{x}))^2 + \boldsymbol{\psi_\tau}(\mathbf{x}) \cdot \boldsymbol{\psi_\tau}(\mathbf{x}) + (\nabla\psi_v(\mathbf{x}) \cdot \nabla\psi_v(\mathbf{x}) + (\nabla \cdot \boldsymbol{\psi_\tau}(\mathbf{x}))^2))}_{e_k(\mathbf{x})}\, d\mathbf{x}.\end{aligned} \tag{4.1}$$

The scaled version of the V-norm [32] is given by

$$\begin{aligned}\|(\psi_v, \boldsymbol{\psi_\tau})\|^2_{\mathbb{V},k} &= \|\psi_v\|^2 + \sqrt{|k|}\,\|\nabla\psi_v\|^2 + \|\boldsymbol{\psi_\tau}\|^2 + \sqrt{|k|}\,\|\nabla \cdot \boldsymbol{\psi_\tau}\|^2 \\ &= \int_k \underbrace{(\psi_v(\mathbf{x}))^2 + \boldsymbol{\psi_\tau}(\mathbf{x}) \cdot \boldsymbol{\psi_\tau}(\mathbf{x}) + \sqrt{|k|}(\nabla\psi_v(\mathbf{x}) \cdot \nabla\psi_v(\mathbf{x}) + (\nabla \cdot \boldsymbol{\psi_\tau}(\mathbf{x}))^2))}_{e_k(\mathbf{x})}\, d\mathbf{x}.\end{aligned} \tag{4.2}$$

Here $|k|$ represents the volume of the element $k \in T_h$.

Definition 1. *Let $\bar{\mathbf{x}} = (\bar{x}, \bar{y}) \in \Omega$ and $i \in \mathbb{N}$. We say that a polynomial $P_{i,\bar{\mathbf{x}}}(\mathbf{x}) : \Omega \to \mathbb{R}$ is a homogeneous polynomial of order i centered at $\bar{\mathbf{x}}$ if*

$$P_{i,\bar{\mathbf{x}}}(\mathbf{x}) = \sum_{l=0}^{i} c_l (x - \bar{x})^l (y - \bar{y})^{i-l}, \tag{4.3}$$

where $c_l \in \mathbb{R}$ and $\mathbf{x} = (x, y)$. [[35], Lemma 3.12] presents an important result on the boundedness of a homogeneous polynomial. We mention this in the next lemma as it is one of the building blocks of the presented methodology.

Lemma 1. *Let $P_{i,\bar{\mathbf{x}}}(\mathbf{x}) : \Omega \to \mathbb{R}$ be a i^{th} order homogeneous polynomial centered at $\bar{\mathbf{x}}$, $i \geq 2$. Then there exist values $A_i \in \mathbb{R}^+, \rho_i \in \mathbb{R}^+$ and $\phi_i \in [0, 2\pi)$ such that*

$$|P_{i,\bar{\mathbf{x}}}(\mathbf{x})| \lesssim A_i \left((\mathbf{x} - \bar{\mathbf{x}})^T Q_{\phi_i} D_{\rho_i} {Q_{\phi_i}}^T (\mathbf{x} - \bar{\mathbf{x}})\right)^{\frac{i}{2}}. \tag{4.4}$$

In particular, one can choose

$$Q_{\phi_i} = \begin{bmatrix} \cos(\phi_i) & -\sin(\phi_i) \\ \sin(\phi_i) & \cos(\phi_i) \end{bmatrix}, \qquad D_{\rho_i} = \begin{bmatrix} 1 & 0 \\ 0 & \rho_i^{\frac{-2}{i}} \end{bmatrix},$$

where

$$\phi_i = \underset{\phi}{\operatorname{argmax}} \, |P_{i,\bar{\mathbf{x}}}(\xi(\phi))|,$$

$$A_i = |P_{i,\bar{\mathbf{x}}}(\xi(\phi_i))|, \qquad A_{i,\perp} = |P_{i,\bar{\mathbf{x}}}(\xi(\phi_i - \frac{\pi}{2}))|,$$

$$\rho_i = \frac{A_i}{A_{i.\perp}}.$$

Here $\xi(\phi_i)$ represents a vector at an angle ϕ_i (anti-clockwise) with x-axis on boundary of unit ball B.

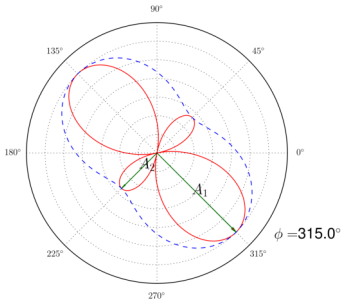


Figure 4.1.1: Bound on an arbitrary fourth order polynomial.

In fig. 4.1.1, the graph denoted by the red continuous line represents $|P_{i,\bar{\mathbf{x}}}(\xi(\phi))|$ for a fourth order polynomial and the graph denoted by the blue dotted line represents the bound mentioned in lemma 1. $|P_{i,\bar{\mathbf{x}}}(\xi(\phi))|$ and its bound are both evaluated on unit circle. In order to compute the anisotropy, we consider the polynomial representation of the error representation function which is computed locally due to broken test spaces as mentioned in chapter 2. Let the test space and the trial space be chosen as mentioned in chapter 2, then error representation functions ψ_v and $(\boldsymbol{\psi}_\tau)_{i,j} \forall i,j \in 1,2$ are polynomials of order $p+\delta p$. Thus, in eq. (4.1) and eq. (4.2) $e_k(\mathbf{x})$ is a polynomial of $2(p+\delta p)$. δp is the enrichment added to the order of $\mathbb{X}_h$ to obtain $\mathbb{V}_r$. We can write

$$e_k(\mathbf{x}) = \sum_{i=0}^{2(P+\delta P)} P_{i,\bar{\mathbf{x}}_k}(\mathbf{x}). \tag{4.5}$$

Equation (4.5) represents the decomposition of $e_k(\mathbf{x})$ into homogeneous polynomials centered at the centroid $\bar{\mathbf{x}}_k$ of $k \in T_h$. Let E_k denote the circumscribing ellipse of the element k as shown in fig. 3.3.1. Since integrals of odd monomials (whose cumulative power is odd) over a symmetric interval about the origin is zero, half of the monomials in eq. (4.5) integrate to zero over the circumscribing ellipse. Utilizing the bound introduced in lemma 1 for each homogeneous polynomial with even order in the decomposition of $e_k(\mathbf{x})$, we have:

$$\int_k e_k(\mathbf{x})\, d\mathbf{x} \leq \int_{E_k} e_k(\mathbf{x})\, d\mathbf{x} \lesssim \sum_{i=2, i\in Z_E^+}^{2(P+\delta P)} \int_{E_k} A_i \left((\mathbf{x}-\bar{\mathbf{x}}_k)^T Q_{\phi_i} D_{\rho_i} {Q_{\phi_i}}^T (\mathbf{x}-\bar{\mathbf{x}}_k) \right)^{\frac{i}{2}} d\mathbf{x}. \tag{4.6}$$

In above expression, we have ignored $i=0$ as it represents a constant which acts as an offset to the bound and has no anisotropy. Z_E^+ represents the set of positive even integers. The semi-axes of E_k, aspect ratio of E_k and the angle between major axis of E_k and the x-axis are denoted by h_1, $h_2 = \frac{h_1}{\beta}$ and ϕ respectively. Let $\hat{E} := \{\hat{\mathbf{x}} \in \mathbb{R}^2; |\hat{\mathbf{x}}| \leq 1\}$ be a closed unit ball centered at $\bar{\mathbf{x}}_k$. Then a mapping $F : \hat{E} \to \mathbb{R}^2$ may be defined such that $\mathbf{x} \in E_k$ is obtained as

$$\mathbf{x} = F(\hat{\mathbf{x}}) = Q_e S_e \hat{\mathbf{x}} + \bar{\mathbf{x}}_k, \tag{4.7}$$

where Q_e is the rotation matrix of angle ϕ and

$$S_e = \begin{bmatrix} h_1 & 0 \\ 0 & h_2 \end{bmatrix} = h_1 \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{\beta} \end{bmatrix}.$$

If we treat $\bar{\mathbf{x}}_k$ as the origin and introduce polar coordinates $\hat{\mathbf{x}} = \begin{bmatrix} cos(\theta) & sin(\theta) \end{bmatrix}^T$, we have $\mathbf{x} - \bar{\mathbf{x}}_k = Q_e S_e \hat{\mathbf{x}} = Q_e S_e \begin{bmatrix} cos(\theta) & sin(\theta) \end{bmatrix}^T$. On substituting $\mathbf{x} - \bar{\mathbf{x}}_k$ in eq. (4.6), we have

$$\int_k e_k(\mathbf{x})\, d\mathbf{x} \leq \int_{E_k} e_k(\mathbf{x})\, d\mathbf{x} \lesssim \sum_{i=2, i\in Z_E^+}^{2(P+\delta P)} \frac{A_i \lambda_k^{\frac{i+2}{2}}}{i+2} \int_0^{2\pi} (g_i(\theta; \beta, \phi - \phi_i))^{\frac{i}{2}}\, d\theta,$$

where

$$\begin{aligned} g_i(\theta, \beta, \phi - \phi_i) &= G_{11,i} cos^2(\theta) + G_{22,i} sin^2(\theta) + 2G_{12,i} sin(\theta) cos(\theta) \\ with \quad G_{11,i} &= \beta(cos^2(\phi - \phi_i) + \rho_i^{\frac{-2}{i}} sin^2(\phi - \phi_i)), \\ G_{12,i} &= -sin(\phi - \phi_i) cos(\phi - \phi_i)(1 - \rho_i^{\frac{-2}{i}}), \\ G_{22,i} &= \beta^{-1}(sin^2(\phi - \phi_i) + \rho_i^{\frac{-2}{i}} cos^2(\phi - \phi_i)), \end{aligned} \tag{4.8}$$

and $\lambda_k = \frac{1}{d_k}$ where d_k is computed using the implied metric (computed using the vertices of $k \in T_h$ where T_h is the current triangulation). The optimal anisotropic properties $\{\beta^\star, \phi^\star\}$ are set by minimizing the error bound for $k \in T_h$ (see eq. (4.9)):

$$\beta^\star, \phi^\star = \underset{\beta,\phi}{\operatorname{argmin}} \underbrace{\sum_{i=2, i\in Z_E^+}^{2(P+\delta P)} \frac{A_i \lambda_k^{\frac{i+2}{2}}}{i+2} \int_0^{2\pi} (g_i(\theta; \beta, \phi - \phi_i))^{\frac{i}{2}}\, d\theta}_{\bar{G}}. \tag{4.9}$$

Next, we will be briefly mentioning the procedure implemented for solving the minimization problem stated by eq. (4.9). From eq. (4.8), it can inferred that $g_i(\theta, \beta, \phi - \phi_i)$ (consequently $\bar{G}$) is π periodic in ϕ and $\bar{G} \to \infty$ for $\beta \to \infty$ as $g_i \to \infty$ for $\beta \to \infty$. Hence, the continuity of $\bar{G}$ with respect to β and ϕ implies the existence of atleast one minima. The minimum of eq. (4.9) is sought iteratively by performing an alternate search in β and ϕ. More details can be found in [section 7.2 [41]].

4.2 Mesh density computation

After computing the anisotropy parameters $(\beta^\star, \phi^\star)$, we still need to compute the density to identify the metric uniquely. In this section, we provide the details of the computation of the mesh density (d) using a continuous error model for driving $h-$ and $hp-$adaptations. As previously mentioned in chapter 3, the continuous representation of mesh allows us to employ analytical methods to compute the mesh density.

When $U_h = (\boldsymbol{\sigma}_h, u_h, \lambda_h, \hat{\sigma}_h)$ is approximated with $\mathbb{X}_h = \Sigma_h \times V_h \times \Lambda_h \times \omega_h$ using the DPG framework with optimal test functions, it has already been shown that $\|U - U_h\|_E$ is equivalent to the best approximation error under various test norms. In order to achieve this equivalence, the primary ingredient is the equivalence of the prescribed test norm with the optimal test norm. These salient results regarding equivalence of standard V-norm and the optimal test norm are rigorously proved in [27]. In the present work, we have prescribed discontinuous trace approximations. It has already been shown in [66] that this is a valid functional setting and is quasi-optimal when compared to conforming trace approximations. Hence, we assume $\|U - U_h\|_E \approx O(h^{p+1})$ which is essential for the proposed continuous error estimate. As mentioned in section 4.1, we have also used a scaled version of the standard V-norm. The reason behind this scaling is the fact that it helps in mitigating the increasing condition number of the Gram matrix $\mathbb{G}$ as the size of the elements starts reducing locally, which in turn keeps the condition number of the global stiffness matrix tractable. This ill-conditioning of the gram matrix is already mentioned in [20]. Another option to overcome this issue is to implement the DLS framework (see section 2.4.2), but it comes with the added cost of solving a least-squares problem.

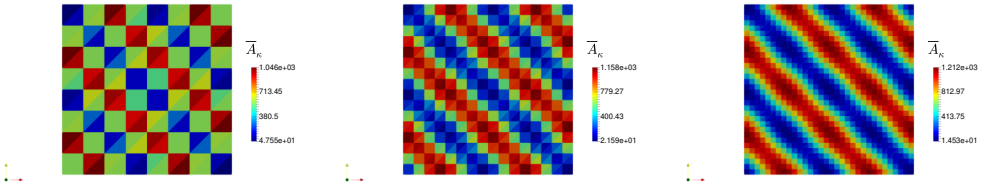
4.2.1 Continuous model of energy error estimate for h adaptation

Assumption 4.2.1: Let T_h be a triangulation and η_k be a local error estimate such that $\eta^2 = \sum_{k \in T_h} \eta_k^2$ and $\eta = O(h^s)$ i.e error estimate converges at s^{th} order [39]. We assume that

local error estimate η_k scales [1] as $\eta_k^2 = \overline{A}_k |k|^{(s+1)}$ where $\overline{A}_k$ depends upon the anisotropy of the element and order of the method s, but is independent of $|k|$. Furthermore, the order of the method s directly depends upon the polynomial degree of approximation p.

Verification of assumption 4.2.1 can be found in [section 8.1, [39]] where the authors have demonstrated this with a Laplace problem and in [section 3.2, [95]] where similar properties are studied for compressible flow problems.

In the present work, we have shown a similar demonstration solving the same Laplace problem as in [section 8.1, [39]] where the exact solution u has the form $u = sin(2\pi x)sin(2\pi y)$ in fig. 4.2.1, fig. 4.2.2 and fig. 4.2.3 for $p = 1, 2, 3$ using the standard V-norm. Similar to the observation made in [39], there is a very small dependence of the maximal value of $\overline{A}_k$ on element size, whereas dependence of minimal value is not. However, as already mentioned in [39], these elements with the smallest error usually do not contribute too much to the total error.



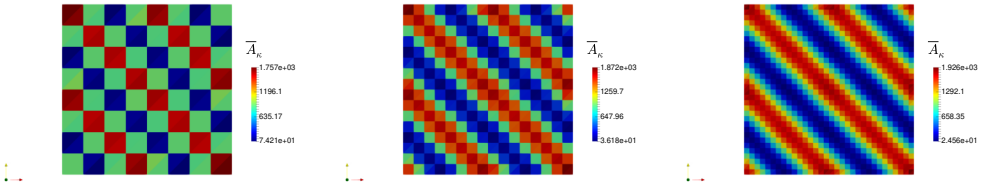
$ne = 128$ $ne = 512$ $ne = 2048$

Figure 4.2.1: Numerical justification of assumption 4.2.1 using $P = 1$

Let $\{T_h\}_n$ be the sequence of triangulation employed. We construct an error estimate which asymptotically achieves global equality with the inbuilt energy error estimate. Next, we define the error density function as

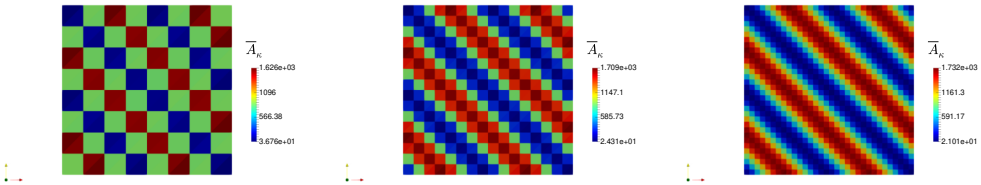
$$e_d(\mathbf{x}) = \overline{A}(\mathbf{x}) d(\mathbf{x})^{-s} \alpha^s, \tag{4.10}$$

[1] The scaling $\eta_k^2 \propto |k|^{(s+1)}$, follows from the global estimate $\eta = \mathcal{O}(h^s)$, using $h^{2s} \propto |k|^s$ and noting that the contribution η_k from each individual element scales with an additional factor of $|k|$, due to the local domain of integration.



$ne = 128$ $ne = 512$ $ne = 2048$

Figure 4.2.2: Numerical justification of assumption 4.2.1 using $P = 2$



$ne = 128$ $ne = 512$ $ne = 2048$

Figure 4.2.3: Numerical justification of assumption 4.2.1 using $P = 3$

Using assumption 4.2.1 and the a priori established order of convergence of the energy estimate [27], we can set $s = p + 1$ (p is the order of approximation). Using eq. (4.10), we can reformulate assumption 4.2.1, where we set $|k| = \frac{\alpha}{d}$, $\eta_k = \|U - U_h\|_{E,k}$ and $\alpha = \frac{3\sqrt{3}}{4}$, to obtain

$$\|U - U_h\|^2_{E,k} \approx e_d(\mathbf{x}_k)|k| \quad \text{for } \mathbf{x}_k \in k,\ k \in T_h. \tag{4.11}$$

Thus, asymptotically ($h \to 0$), we have

$$\|U - U_h\|^2_{E,\Omega} = \sum_{k \in T_h} e_d(\mathbf{x}_k)|k| \to \int_\Omega e_d(\mathbf{x})\, d\mathbf{x}.$$

Hence, we introduce the global continuous error estimate $E_c = \int_\Omega e_d(\mathbf{x})\, d\mathbf{x}$ which will be the primary target of the optimization problem 4.2.1. We treat $\overline{A}(\mathbf{x})$ as a continuous function though it can only be computed discretely on a triangulation. Next, we would

like to recall the notion of mesh complexity. We define the mesh complexity as

$$\mathscr{N} = \int_{\Omega} d(\mathbf{x})\, d\mathbf{x}.$$

In the case of triangulation, we will have a discrete density field i.e. $d(\mathbf{x})$ is piecewise constant. Hence, we have

$$\mathscr{N} = \int_{\Omega} d(\mathbf{x})\, d\mathbf{x} \approx \sum_{k \in T_h} d(\mathbf{x}_k)|k| = \frac{3\sqrt{3}}{4} Ne \quad \text{where } \mathbf{x}_k \in k. \tag{4.12}$$

In the above relation, we have used eq. (3.7) and Ne is the total number of elements in the mesh. Thus, we formulate a problem of minimizing continuous error at a fixed cost, i.e. the desired number of elements in the mesh, by prescribing an appropriate value of $\mathscr{N}$.

Problem 4.2.1: Let $\mathscr{N}$ be the desired complexity and $e_d(\mathbf{x})$ be the error density function. We seek a mesh density distribution $d(\mathbf{x}) : \Omega \to \mathbb{R}^+$ such that:

(a) $\mathscr{N} = \int_{\Omega} d(\mathbf{x})\, d\mathbf{x}$.

(b) $E_c = \int_{\Omega} e_d(\mathbf{x})\, d\mathbf{x}$ is minimized.

To compute the minimum of the continuous error estimate under the constraint of constant complexity $\mathscr{N}$, we use the calculus of variations. Taking variation of the constraint in problem 4.2.1 with respect to $d(\mathbf{x})$ yields:

$$\int_{\Omega} \delta d(\mathbf{x})\, d\mathbf{x} = 0. \tag{4.13}$$

Next, the error density function is substituted into the global continuous error estimate, and this is followed by taking its variation with respect to $d(\mathbf{x})$, after which we obtain

$$\delta E_c = -(p+1)\alpha^{(p+1)} \int_{\Omega} \overline{A}(\mathbf{x}) d(\mathbf{x})^{-(p+2)} \delta d(\mathbf{x})\, d\mathbf{x}. \tag{4.14}$$

From eq. (4.13) and eq. (4.14), it is implied that

$$\overline{A}(\mathbf{x}) d(\mathbf{x})^{-(p+2)} = const. \tag{4.15}$$

Hence again using the constraint, we obtain

$$d^{\star}(\mathbf{x}) = (const)\overline{A}(\mathbf{x})^{\frac{1}{p+2}}, \qquad const = \frac{\mathscr{N}}{\int_{\Omega} \overline{A}(\mathbf{x})^{\frac{1}{(p+2)}}\, d\mathbf{x}}. \tag{4.16}$$

On substituting the expression for *const*, we obtain the expression for the optimal density distribution (the superscript $\star$ represents the optimal solution):

$$d^{\star}(\mathbf{x}) = \frac{\mathscr{N}\overline{A}(\mathbf{x})^{\frac{1}{p+2}}}{\int_{\Omega}\overline{A}(\mathbf{x})^{\frac{1}{(p+2)}}\,d\mathbf{x}}. \tag{4.17}$$

Since we are in a discrete setting in terms of triangulation, we compute this desired density for each element $k \in T_h$. This allows us to replace $\overline{A}(\mathbf{x})$ with $\overline{A}_k$ and thus invoking assumption 4.2.1, to obtain

$$\overline{A}_k = \frac{\|U - U_h\|_{E,k}^2}{|k|^{p+2}} \tag{4.18}$$

and

$$d^{\star}(\mathbf{x}_k) = \frac{\mathscr{N}\|U - U_h\|_{E,k}^{\frac{2}{p+2}}}{|k|\left(\sum\limits_{k\in T_h}\|U - U_h\|_{E,k}^{\frac{2}{(p+2)}}\right)}. \tag{4.19}$$

If the computed density is substituted in the error density function shown in eq. (4.10), we obtain an estimate for error in an element constructed with density computed as mentioned in eq. (4.19):

$$\sqrt{e(\mathbf{x}_k, d^{\star})|k^{\star}|} = \alpha^{\frac{(p+2)}{2}}\mathscr{N}^{-\frac{(p+2)}{2}}\left(\sum_{k\in T_h}\|U - U_h\|_{E,k}^{\frac{2}{(p+2)}}\right)^{\frac{(p+2)}{2}}, \tag{4.20}$$

where $|k^{\star}| = \frac{\alpha}{d^{\star}}$. On summing up the expression mentioned in eq. (4.20) for every element in the triangulation, we obtain an estimate of the error on a mesh which is constructed using the optimal density distribution:

$$E_c^{\star} = \sqrt{\sum_{k\in T_h} e(\mathbf{x}_k, d^{\star})|k^{\star}|} = \left(\frac{\alpha}{\mathscr{N}}\right)^{\frac{p+1}{2}}\left(\sum_{k\in T_h}\|U - U_h\|_{E,k}^{\frac{2}{(p+2)}}\right)^{\frac{(p+2)}{2}}, \tag{4.21}$$

and is known as a continuous error estimate. Finally, after computing $(\beta^{\star}, \phi^{\star}, d^{\star})$ for every element in T_h, the discrete metric field can be generated and passed on to the metric-based mesh generator to produce the desired triangulation.

4.2.1.1 Goal oriented adaptation

For goal-oriented adaptation, we use $\eta_k = \eta_k^{\star}\tilde{\eta}_k$ where $\tilde{\eta}_k = \|U - U_h\|_{E,k}$ and $\eta_k^{\star}$ from eq. (2.38) in place of just $\|U - U_h\|_{E,k}$ for the global size computation. The anisotropy computation

stays the same. We kept the value of $s = p + 1$. In assumption 4.2.1, we require sufficient regularity, but in many cases, the regularity of both the dual and primal solution is questionable. Hence, we prefer to use $s = p + 1$ which is a pessimistic approach in place of $s = 2p + 1$ as the a priori convergence rate for target functionals. Using $s = p + 1$ results in overestimating the optimal area, and hence, meshes might not be sufficiently refined initially. However, this improves with subsequent adaptations. In [39], this has already been highlighted for non-regular solutions where the authors have used $s = p + 1$, which can be higher than the expected regularity of the solution. In that case, the optimal area will be underestimated, and with subsequent adaptation computed area converges to the optimal one. In order to obtain the global density distribution, we employ the product of the energy error estimator and the DPG* based error estimator from [34], but this can be replaced with DWR error estimator using the reconstructed adjoint solution. While reconstructing adjoint, we employ patchwise reconstruction. The details are documented in previous works [6, 37].

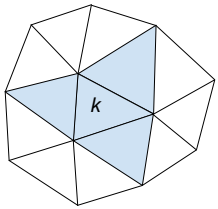


Figure 4.2.4: Stencil representing the first level neighbors for patchwise reconstruction.

4.2.1.2 Regularization

In eq. (4.18), the error density $\overline{A}$ can become very small if the a posteriori error estimate or indicator has a very small value. From eq. (4.17), it can be easily inferred that this may result in a minimal value of density. Even though mesh generators might still

produce a valid adapted mesh, they may excessively coarsen certain parts of the domain. Consequently, large values of a posteriori estimates on this adapted mesh will lead to refinement in the same region, which was coarsened in the previous adaptation cycle. The result is a cycle of alternate refinement and coarsening. One way to overcome this issue is a simple regularization of the parameter $\overline{A}$. The idea is to determine a new local mesh size so that the local error contribution is approximately the same as that of the optimized mesh described this far. We attain this by employing the local optimal error$(\sqrt{e(\mathbf{x}_k, d^\star)|k^\star|})$ of eq. (4.20) in place of the local error estimator or indicator (η_k). This is only done if $\eta_k < (\sqrt{e(\mathbf{x}_k, d^\star)|k^\star|})$ for a given element. An in-depth exposition to the idea of regularization can be found in [90]. Another way to mollify the gradient of the metric field is by controlling it locally. It can be achieved by using a metric-smoothing algorithm such as $Hv - correction$ that is provided within the software BAMG [59], and an in-depth exposition of the algorithm can be found in [10].

4.2.2 Continuous model of energy error estimate for hp adaptation

In the finite element community, it has been known over the years that $p-$adaptation should be preferred over $h-$adaptation in regions where the solution has high regularity, whereas $h-$adaptation is advantageous in the presence of discontinuities, singularities, and very high gradients. During the initial days, many $hp-$adaptation algorithms generated appropriate mesh using the a priori information about the solution and followed it up with $p-$refinement. However, this can lead to an intractable mesh generation process when there is not much a priori information about the solution. Moreover, performing $p-$refinement on a poorly constructed initial mesh can result in solution quality worse than $h-$adaptive meshes [30]. In order to overcome this arduous situation, automatic adaptation strategies are indispensable. An automatic mesh adaptation strategy would try to produce near-optimal $hp-$meshes without the user's much prior information. However, most automatic mesh adaptation strategies face the dilemma of choosing between $h-$refinement

or $p-$refinement. This issue has been addressed in the past through many strategies. Some of the proposed strategies are:

- [67] where the decision to prioritize h or p refinement depends upon the local regularity of the dual and the primal solution. The regularity is measured using local Sobolev indices.

- [52, 68] utilizes the smoothness of the solution variable as a deciding factor to choose between h or p refinement. The smoothness is computed using a truncated Legendre expansion of the solution.

- [91] uses the derivatives of error with respect to metric and polynomial order to construct optimal $hp-$meshes.

[23, 40, 90, 91] contains some of the recent development in context of metric-based adaptation for HDG and DG discretizations. As previously mentioned in chapter 3, the goal is to generate meshes with minimal error at a given computational cost (total DOFs).

In section 4.2.1, the $h-$adaptation strategy modified the approximation space by targeting the error-representation function associated with each element. The optimum properties that define the metric are computed by splitting them into anisotropic components (β, ϕ) and density d. The anisotropy computations are local and entirely isolated for each element. Thus, we can vary the polynomial order for each element in the mesh. However, the density computation is a non-local process and would require some modification. The selection of appropriate polynomial order would require an additional step on top of density computation.

Next, we would like to recall the concept of $hp-$mesh. If we have a triangulation T_h, then we could associate a integer p_k with every element $k \in T_h$. These integers could be brought together to form a vector called polynomial distribution vector $\mathbf{p} = \{p_k : k \in T_h\}$. The triangulation T_h and the associated polynomial distribution vector $\mathbf{p}$ forms the pair, $T_{h,p} := \{T_h, \mathbf{p}\}$ called the $hp-$mesh.

With the definition of $hp-$mesh, we would also like to modify the optimization goal stated in eq. (3.10). Rather than having the number of elements as a constraint, we would like to have the total number of DOFs as a constraint (in case $hp-$mesh, the number of DOFs and the number of elements in the mesh are not directly proportional). The new problem definition is given by

$$\min_{\forall T_h} E \quad \text{such that} \quad N(Ne, \mathbf{p}) = const. \tag{4.22}$$

Similar to $h-$adaptation, the problem mentioned above cannot be solved using analytical methods, and hence, we would require a continuous error model. Previously, we defined $\bar{A}(\mathbf{x})$ as continuous variable for a polynomial p at the point $\mathbf{x}$ and replaced it with $\bar{A}_k$ in discrete setting. Similarly, we can treat the polynomial distribution vector $\mathbf{p}$ and introduce a continuous analog $p(\mathbf{x})$ for the continuous mesh model. The polynomial distribution vector $\mathbf{p}$ can be seen as a vector created from the snapshots of $p(\mathbf{x})$ in each element. In section 3.5, we stated that the metric tensor associated with each element could be seen as a local snapshot of the continuous metric field. The relation between the polynomial distribution function and the polynomial distribution vector is similar. Now, we can define the continuous equivalent of the problem defined in eq. (4.22) as

$$\min_{\forall T_h} E_c \quad \text{such that} \quad \mathscr{N}_{h,p} = const. \tag{4.23}$$

We require to define the definition of mesh complexity $\mathscr{N}_{h,p}$ for a $hp-$mesh as the previous definition of complexity ($\mathscr{N}$) is directly proportional to the number of elements in the mesh and is useful for $h-$adaptation only. For $hp-$adaptation, we create a constraint representing the mesh's total DOFs. The continuous constraint is defined as follows:

$$\mathscr{N}_{h,p} = \int_{\Omega} w(\mathbf{x}) d(\mathbf{x})\, d\mathbf{x}, \tag{4.24}$$

where

$$w(\mathbf{x}) = \frac{(p(\mathbf{x}) + 1)(p(\mathbf{x}) + 2)}{2}. \tag{4.25}$$

The relation between the continuous mesh complexity and its discrete counterpart for $hp-$adaptation can be computed as follows:

$$\mathscr{N}_{h,p} = \int_{\Omega} w(\mathbf{x}) d(\mathbf{x})\, d\mathbf{x} \approx \sum_{k \in T_h} d(\mathbf{x}_k) w(\mathbf{x}_k) |k| = \sum_{k \in T_h} \alpha w(\mathbf{x}_k) = N(Ne, \mathbf{p}). \tag{4.26}$$

In eq. (4.26), we have used eq. (3.7) and $\alpha = \frac{3\sqrt{3}}{4}$. As mentioned in beginning of this section, the density computation is a two step process:

- Polynomial selection.
- Mesh density computation for the variable polynomial order.

4.2.2.1 Polynomial selection

In order to choose the proper polynomial order of approximation for an element, we solve the governing PDE locally over a patch surrounding the element. The boundary conditions for these local problems are obtained using trace ($\hat{u}_h$) in case of Dirichlet boundary conditions or normal flux ($\hat{\sigma}_h$) for Neumann boundary condition computed at the current polynomial order, but can be generalized for more complex cases. In fig. 4.2.5, we have shown an internal patch where we have used the Dirichlet boundary condition for the scalar convection-diffusion problem. We solve these local problems at three different

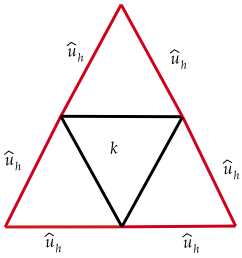


Figure 4.2.5: Patch around an element k over which local problems are solved. Edges marked by red represents the boundaries of patch with Dirichlet boundary conditions

polynomial orders: p_k and $p_k \pm 1$ where p_k represents the current polynomial order in

element k. Once the solution is computed, it is followed by computing the local energy error estimate $\|U - U_h\|_{E,k}$ for each polynomial order. We denote the energy error computed at the polynomial order $p_k + i$ by E_{p_k+i} and the number of degrees of freedom for each polynomial order by N_{p_k+i} where $i = -1, 0, 1$. Also, we introduce a new parameter m_{p_k+i} which corresponds to the amount of uniform refinement or coarsening required to achieve a level error and computed as follows:

$$m_{p_k+i} = \left(\frac{E_{p_k+i}}{E_{p_k}} \right)^{\frac{2}{s_i+1}} N_{p_k+i}, \tag{4.27}$$

where s_i is the a priori rate of convergence for order $p_k + i$. For energy norm, we have used $s_i = p_k + i$. In order to choose a better suited polynomial order, we compute m_{p_k+i} for $p_k - 1$ and $p_k + 1$ to achieve the same error as p_k and optimal order ($p_{k,opt}$) is the one which achieves E_{p_k} with least degrees of freedom.

$$p_{k,opt} = \underset{i=-1,0,1}{\operatorname{argmin}}\, m_{p_k+i} \tag{4.28}$$

The idea behind this is to find out the polynomial order, which is more efficient in terms of the number of degrees of freedom to achieve the same error level as E_{p_k}. For $i = 0$, the above computation is trivial. Once the polynomial order is selected, we still need to compute the mesh density distribution. For this purpose, we will target the global continuous estimate $E_c = \int_\Omega e_d(\mathbf{x})\, d\mathbf{x}$ during minimization. The enrichment δp of the test search space while performing $hp-$adaptations is more than the enrichment while performing $h-$adaptations by order of 1. An increase in the order of the approximation space increases the order of the test search space so that the solution at $p_k + 1$ has the same fidelity as p_k and p_{k-1}.

4.2.2.2 Mesh density computation

In order to generate the optimal density distribution at fixed cost, $\mathscr{N}_{h,p}$ will be used in the complexity constraint.

Problem 4.2.2: Let $\mathcal{N}_{h,p}$ be the desired complexity in terms of degrees of freedom and $e_d(\mathbf{x}, p(\mathbf{x}))$ be the error density. We seek a mesh density distribution $d(\mathbf{x}) : \Omega \to \mathbb{R}^+$ for a given polynomial distribution $p(\mathbf{x}) : \Omega \to Z^+$ such that:

(a) $\mathcal{N}_{h,p} = \int_\Omega w(\mathbf{x}) d(\mathbf{x})\, d\mathbf{x} \qquad with \quad w(\mathbf{x}) = \frac{2(p(\mathbf{x})+1)(p(\mathbf{x})+2)}{3\sqrt{3}}$.

(b) $E_c = \int_\Omega e_d(\mathbf{x}, p(\mathbf{x}))\, d\mathbf{x}$ is minimized.

For density computation, we have implemented the analytic optimization for $hp-$ continuous mesh model mentioned in [40]. Using eq. (4.10), we state the global continuous error as follows:

$$E_c = \int_\Omega \alpha^{(p(\mathbf{x})+1)} \overline{A}(\mathbf{x}, p(\mathbf{x})) d(\mathbf{x})^{-(p(\mathbf{x})+1)}\, d\mathbf{x}. \tag{4.29}$$

In case of $hp-$adaptation, we use $s = p(\mathbf{x}) + 1$ as local order of convergence in assumption 4.2.1. Taking variation of complexity constraint with respect to $d(\mathbf{x})$ yields:

$$\delta \mathcal{N}_{h,p} = \int_\Omega w(\mathbf{x}) \delta d(\mathbf{x}) d\mathbf{x} = 0. \tag{4.30}$$

Next, on taking the variation of global continuous error with respect to density, we obtain

$$\delta E_c = \int_\Omega -(p(\mathbf{x}) + 1) \alpha^{(p(\mathbf{x})+1)} \overline{A}(\mathbf{x}, p(\mathbf{x})) d^{-(p(\mathbf{x})+2)} \delta d\, d\mathbf{x}. \tag{4.31}$$

Using the complexity constraint, one can observe that we can obtain an admissible solution if we set

$$\frac{(p(\mathbf{x}) + 1)\overline{A}(\mathbf{x}, p(\mathbf{x}))}{w(\mathbf{x})} \alpha^{(p(\mathbf{x})+1)} d(\mathbf{x})^{-(p(\mathbf{x})+2)} = const. \tag{4.32}$$

Solving this for $d(\mathbf{x})$ yields

$$d^\star(\mathbf{x}) = \left(\frac{(p(\mathbf{x}) + 1)\overline{A}(\mathbf{x}, p(\mathbf{x}))\alpha^{(p(\mathbf{x})+1)}}{w(\mathbf{x})} \right)^{\frac{1}{(p(\mathbf{x})+2)}} const^{-\frac{1}{(p(\mathbf{x})+2)}}. \tag{4.33}$$

In contrast to $h-$adaptation, here *const* doesn't have a closed form solution. On substituting the expression for optimal density from eq. (4.33) into complexity constraint, we obtain

$$\mathcal{N}_{h,p} = \int_\Omega w(\mathbf{x}) \left(\frac{(p(\mathbf{x}) + 1)\overline{A}(\mathbf{x}, p(\mathbf{x}))\alpha^{(p(\mathbf{x})+1)}}{w(\mathbf{x})} \right)^{\frac{1}{(p(\mathbf{x})+2)}} const^{-\frac{1}{(p(\mathbf{x})+2)}}. \tag{4.34}$$

Equation (4.34) is a non-linear problem in *const* and has varying exponent. Thus, we need to employ numerical techniques to solve for *const*. We employ a bisection method to compute this constant in our current work. Once *const* is computed, we can substitute its value into eq. (4.33) to obtain the optimum density distribution $d^{\star}(\mathbf{x})$.

The proposed $hp-$adaptive algorithm is a generalization of the $h-$adaptation. If the variable polynomial distribution is substituted with a $p(\mathbf{x}) = p$ throughout the domain, then $w(\mathbf{x}) \equiv const$ and proposed iterative algorithm reverts back to $h-$only adaptation method. Since, we are in a discrete setting in terms of triangulation, the quantities $\overline{A}(\mathbf{x}, p(\mathbf{x}))$, $p(\mathbf{x})$ and optimal density $d^{\star}(\mathbf{x})$ are computed in a piecewise constant sense. Thus, if $\mathbf{x}_k \in k$ and $k \in T_h$, using assumption 4.2.1, we have

$$\overline{A}_{k,p_{k,opt}} = \frac{\|U - U_h\|^2_{E,k,p_{k,opt}}}{|k|^{p_{k,opt}+2}} \tag{4.35}$$

and

$$d^{\star}(\mathbf{x}_k) = \left(\frac{(p_{k,opt}+1)\overline{A}_{k,p_{k,opt}} \alpha^{(p_{k,opt}+1)}}{w(\mathbf{x}_k, p_{k,opt})} \right)^{\frac{1}{(p_{k,opt}+2)}} const^{-\frac{1}{(p_{k,opt}+2)}}, \tag{4.36}$$

where $\|U - U_h\|_{E,k,p_{k,opt}}$ represents the energy error in element k for the optimal polynomial order ($p_{k,opt}$) chosen via the process mentioned in section 4.2.2.1.

4.2.2.3 Goal oriented adaptation

For goal-oriented adaptations, a similar approach as mentioned in section 4.2.1.1 is followed. We replace the energy estimate in both polynomial selection procedure and the mesh density computation with the product of energy norm and $\eta_k^{\star}$ as we did for $h-$adaptations. Only nuance here is that we compute the dual error estimate ($\eta_k^{\star}$) at p_k and vary the energy estimate for p_{k+i} with $i = \pm 1$. The rationale behind computing $\eta_k^{\star}$ at current polynomial order is to avoid the amount of additional computational cost involved in solving adjoint system at different polynomial orders.

4.2.2.4 Polynomial order transfer between meshes

The methodology provided in the previous section computes the optimum density distribution and polynomial distribution using the current triangulation. However, passing

the metric field to a metric-based mesh generator will produce a new mesh with elements not trivially related to the current mesh. Hence, the problem arises of transferring the computed polynomial distribution from the current mesh to the newly generated one. To solve this issue, the current element-wise polynomial distribution is transferred to the nodes using a weighted average.

$$p_v = \frac{1}{\Omega_v} \sum_{k \in T(v)} |k| p_k, \tag{4.37}$$

where $T(v)$ represents the set of triangles that share v as a vertex and $\Omega_v = \sum_{k \in T(v)} |k|$. Once we have the discrete polynomial distribution over the vertices of the current mesh, a simple interpolation between two meshes can be done to achieve the polynomial distribution at new mesh nodes. The mesh-generators generally handle this interpolation. In the new mesh, we perform a similar interpolation from the mesh nodes to the elements for the polynomial order. We prescribe the average polynomial order for the element using the order at each vertex of the element i.e.

$$p_k = \frac{1}{3} \sum_{v \in V(k)} p_v, \tag{4.38}$$

where $V(k)$ is the set of vertices of the triangle $k \in T_h$.

This page is intentionally left blank.

Chapter 5

Numerical Results: h adaptation

This chapter presents the numerical results to show the fidelity of the proposed adaptation methodology. In this chapter, we will only focus on $h-$adaptation[1]. The results are obtained by employing our in-house discontinuous Petrov-Galerkin code developed as a part of this thesis. The in-house code also contains support for DLS (see section 2.4.2) and DPG-Star (see section 2.3), which are required for specific test cases.

5.1 Test case I: boundary layer

Sharp boundary layers are among the most encountered features in flow fields. This test problem is selected in order to validate the fidelity of the proposed algorithm in the presence of strong boundary layers. In particular, we solve,

$$\begin{aligned} \beta \cdot \nabla u - \epsilon \nabla^2 u &= s(\mathbf{x}) \qquad \text{in } \Omega = (0,1)^2, \\ u &= 0 \qquad \text{on } \partial\Omega, \end{aligned} \tag{5.1}$$

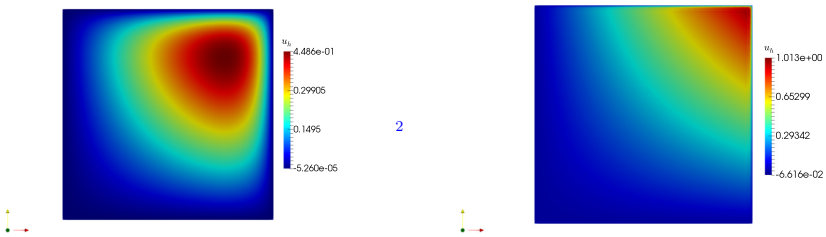
where $\epsilon > 0$ and $\beta = [1,1]^T$. The source term $s(\mathbf{x}) = s(x,y)$ is selected in such a way that the exact solution is given by

$$u(\mathbf{x}) = \left(x + \frac{e^{\frac{x}{\epsilon}} - 1}{1 - e^{\frac{1}{\epsilon}}}\right)\left(y + \frac{e^{\frac{y}{\epsilon}} - 1}{1 - e^{\frac{1}{\epsilon}}}\right). \tag{5.2}$$

The solution exhibits a sharp boundary layer near $x \approx 1$, and $y \approx 1$. Its sharpness is determined by ϵ. The smaller the value of ϵ, the sharper will be the boundary layer (see fig. 5.1.1).

In fig. 5.1.2 and fig. 5.1.3, we present the convergence results for the proposed $h-$adaptation algorithm for the standard V-norm and scaled V-norm associated with

[1]The comparison between anisotropic h, isotropic and uniform adaptation can be found in previous works such as [4, 98] showcasing the efficiency of anisotropic adaptation techniques.



(a) $\epsilon = 0.1$ **(b)** $\epsilon = 0.005$

Figure 5.1.1: Boundary layer: solution contour for two values of ϵ.

the test space. We start with an initial isotropic mesh of 32 elements. Between each adaptation cycle, $\mathscr{N}$ is increased by 30%. The growth in $\mathscr{N}$ is arbitrarily chosen. A different choice in growth may result in different pre-asymptotic behavior but should produce similar asymptotic results. Figure 5.1.2 and fig. 5.1.3 exhibits an initial sharp drop in the error, at a rate higher than the asymptotic rate. This pre-asymptotic drop is due to the fact that redistribution of DOFs contributes significantly to the error reduction. The algorithm identifies the anisotropic features present in the solution field during these initial adaptations. Once this identification is made, we reach convergence to the asymptotic rate. The extent of this pre-asymptotic region also depends upon the resolution of the initial mesh. From fig. 5.1.4, it can be inferred that an increase in the resolution of the initial mesh results in fewer adaptations required to reach the asymptotic convergence (Ne_{h_0} denotes the number of elements in the initial mesh). It happens because the information for local anisotropy computation and analytic optimization is sampled at a higher number of points, enhancing the algorithm's performance. Although in the pre-asymptotic region, the convergence may depend on the initial mesh, asymptotically, the convergence curves in fig. 5.1.4 are very similar, consistent with the expectation that each curve should represent a near-optimal sequence of meshes.

A significant advantage of the proposed adaptation algorithm is that the degrees of freedom can be redistributed at a fixed mesh complexity. In this sense, it is similar to

r-adaptation [81], although re-meshing makes the process more flexible. In table 5.1, we present the results showing the drop in error in both L^2 and energy norm. One can think of this drop in error as an effect of reallocating the degrees of freedom from regions with the smoother solution to regions with anisotropic features in the solution. It only takes a few adaptations to reach saturation in terms of the drop in error. The saturation naturally results from the fact that we keep the number of DOFs fixed during this numerical experiment. In fig. 5.1.5, it can be seen that the algorithm prescribes larger elements away from the boundary layer, with most of the degrees of freedom being allocated to the boundary layer.

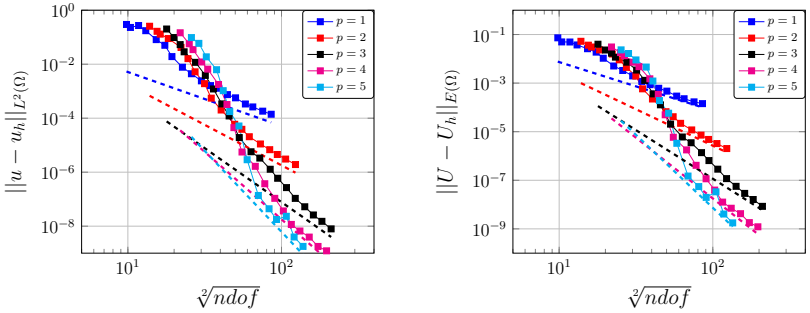


Figure 5.1.2: Convergence plots of (a) L^2 error in u_h and (b) energy error using standard V-norm ($\epsilon = 0.005$, expected order of convergence $(p + 1)$ is represented by the dashed lines).

Adaptation	Ne	$\|u - u_h\|_{2,\Omega}$	$\|U - U_h\|_{E,\Omega}$
0	512	0.0262192	0.0514999
2	437	0.000289966	0.00218748
4	463	6.05824e-06	4.77721e-05
6	484	3.65609e-06	3.09974e-05
8	489	2.02129e-06	1.86422e-05

Table 5.1: Adaptation Vs. error for constant complexity using scaled V-norm ($\epsilon = 0.005$, $p = 3$, $\mathcal{N} = \int_\Omega d(\mathbf{x})\, d\mathbf{x} = 512.0 \times \frac{3\sqrt{3}}{4}$).

Finally, in fig. 5.1.6 we present the comparison of the continuous error estimate in eq. (4.21) with the in-built error estimate of the DPG scheme for various polynomial

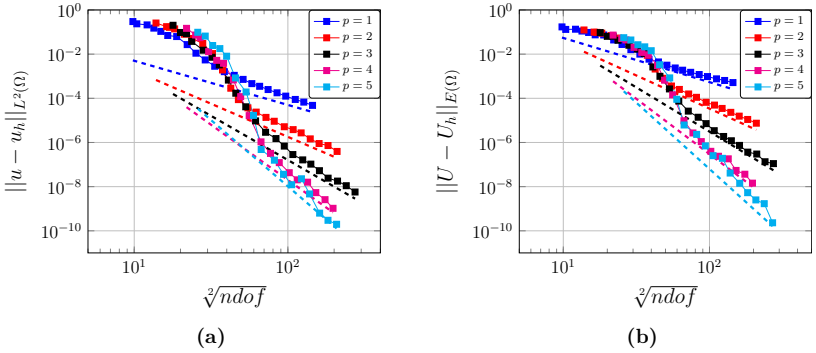


Figure 5.1.3: Convergence plots of (a) L^2 error in u_h and (b) energy error using scaled V-norm ($\epsilon = 0.005$, expected order of convergence $(p+1)$ is represented by the dashed lines).

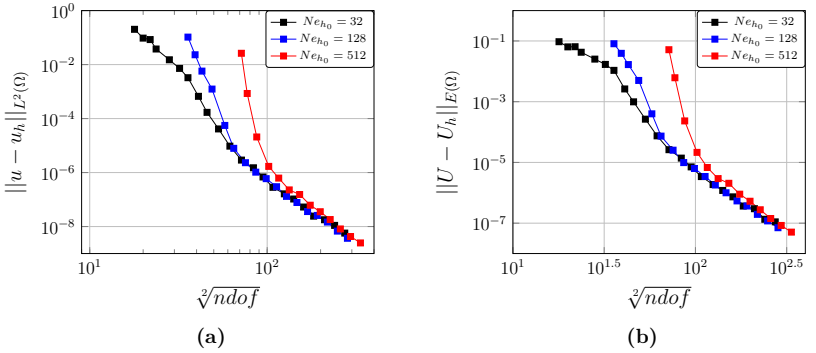


Figure 5.1.4: Convergence plots of (a) L^2 error in u_h and (b) energy error with different initial meshes using scaled V-norm ($\epsilon = 0.005, p = 3$).

orders for $h-$adaptations. The purpose of this comparison is to show that the continuous estimate which drives the adaptation follows the inbuilt error estimate, which is at the core of the presented adaptation algorithm. It is worth pointing out that the value of the continuous mesh estimate $E_c^\star$, computed on a given mesh, is a prediction of the error obtained on the next mesh, which is generated from the optimized metric. It is reflected in fig. 5.1.6 by shifting the data points of the curve corresponding to $E_c^\star$ to the right: we compare the predicted error with the error computed on that mesh.

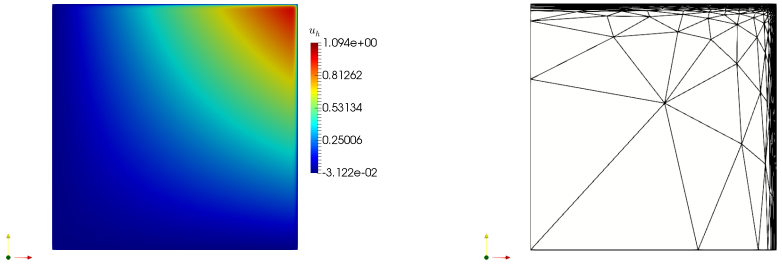


Figure 5.1.5: Solution and mesh after 10 adaptation cycles ($\epsilon = 0.005, Ne = 484, p = 3$) using scaled V-norm.

5.1.1 Goal-oriented adaptation: opposing boundary layer

Similar to the primal problem given by eq. (5.1), the dual problem is formulated by suitably choosing the target J such that the exact solution $\bar{\eta}$ to the dual problem is known. The target functional chosen here is a weighted volume integral of $u(\mathbf{x})$ over the domain:

$$J(u) = \int_{\Omega} j_{\Omega}(\mathbf{x}) u(\mathbf{x})\, d\mathbf{x}. \tag{5.3}$$

In the above target, we can change the functional by prescribing j_{Ω}. For the current linear set-up, we can write the corresponding dual problem as

$$-\beta \cdot \nabla \bar{\eta} - \epsilon \nabla^2 \bar{\eta} = j_{\Omega}(\mathbf{x}) \qquad \text{in } \Omega = (0,1)^2, \tag{5.4}$$

$$\bar{\eta} = 0 \qquad \text{on } \partial\Omega, \tag{5.5}$$

where $\beta = [1,\, 1]^T$. With appropriately chosen $j_{\Omega}(\mathbf{x})$, we can have

$$\bar{\eta}(\mathbf{x}) = \left(-x + 1 + \frac{e^{\frac{1-x}{\epsilon}} - 1}{1 - e^{\frac{1}{\epsilon}}}\right)\left(-y + 1 + \frac{e^{\frac{1-y}{\epsilon}} - 1}{1 - e^{\frac{1}{\epsilon}}}\right), \tag{5.6}$$

which solves eq. (5.4) and eq. (5.5). A contour plot of the dual solution is shown in fig. 5.1.7 for $\epsilon = 0.005$. The anisotropy and the global size distribution are computed as mentioned in section 4.1 and section 4.2.1. In each adaptation cycle, we prescribe an increase of 30% in the mesh complexity ($\mathcal{N}$). In fig. 5.1.8(a), we show the convergence of the target

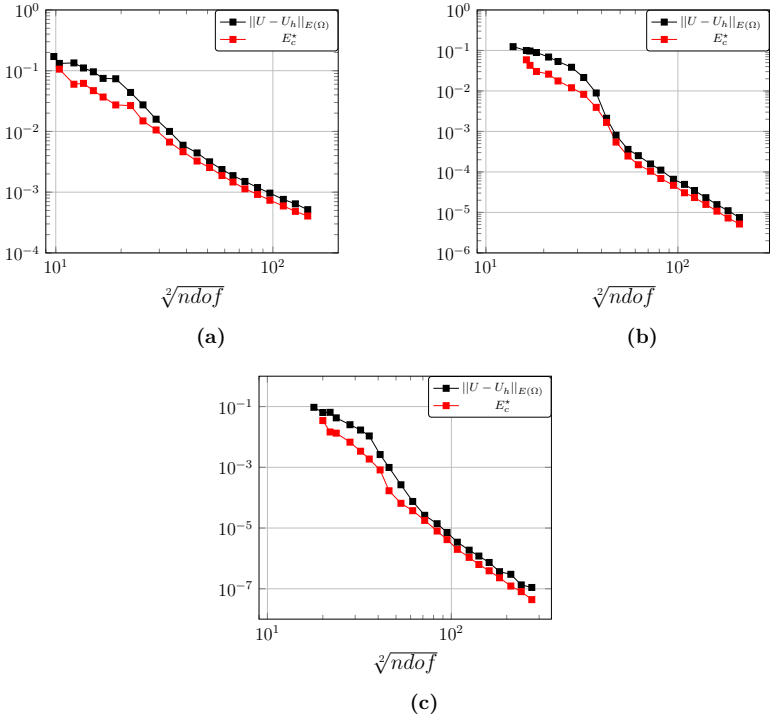


Figure 5.1.6: Comparison of $||U - U_h||_{E(\Omega)}$ and $E_c^\star$ for (a) $p = 1$, (b) $p = 2$ and (c) $p = 3$ at $\epsilon = 0.005$ using scaled V-norm.

functional. In fig. 5.1.8(b), we show the convergence of the dual weighted residual (DWR) error estimator for the target functional. The convergence rate for both error and DWR estimator is $2p + 1$, where p is the polynomial order of approximation for the primal u.

This superconvergence is in accordance with the previous results in [90]. In fig. 5.1.9, we have shown the adapted mesh for $p = 2$ and $p = 3$. One important point to notice in these meshes is the presence of smaller elements near the bottom left corner. Since the dual variable has a boundary layer, the lower value of $\|U - U_h\|_{E,k}$ is being compensated by a larger value of $\eta_k^\star$ mentioned in eq. (2.37) for the elements present near the bottom left corner. One can notice this difference by comparing the meshes in fig. 5.1.9 and the

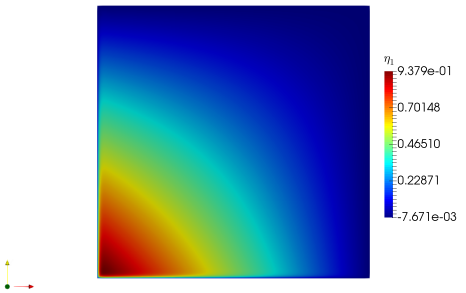
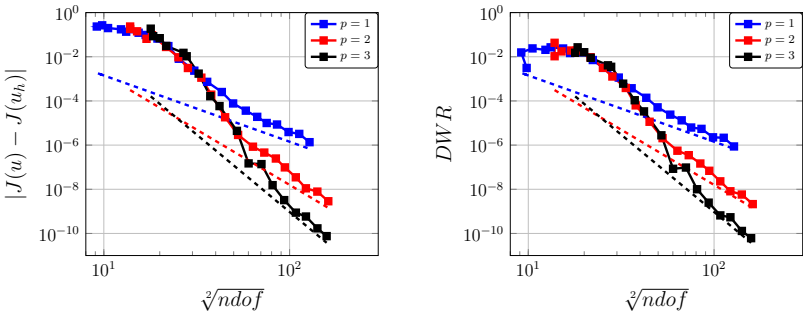


Figure 5.1.7: Boundary layer test case: contour plot of the dual solution. $\bar{\eta}$



(a) **(b)**

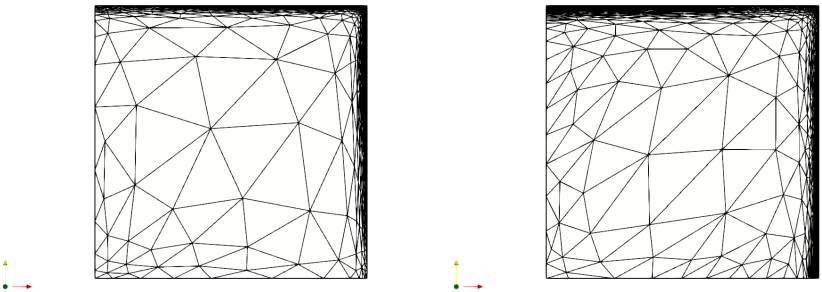
Figure 5.1.8: Convergence plots for (a) error in target functional and (b) dual weighted residual using scaled V-norm ($\epsilon = 0.005$, expected order of convergence $(2p + 1)$ is represented by the dashed lines).

solution adapted mesh shown in fig. 5.1.5.

5.1.2 Goal-oriented adaptation: Gaussian peak

Next, we consider the same primal problem with the target given by eq. (5.3), where

$$j_\Omega(\mathbf{x}) = e^{-\alpha\left((x-x_c)^2+(y-y_c)^2\right)}. \tag{5.7}$$



(a) $Ne = 1133$ **(b)** $Ne = 1177$

Figure 5.1.9: Adapated mesh for (a) $p = 2$ and (b) $p = 3$ after 15^{th} adaptation cycle using scaled V-norm ($\epsilon = 0.005$).

Here $(x_c, y_c) \in \Omega$ determines the location of the Gaussian peak, and α controls the steepness of the Gaussian peak. A similar problem has been considered in [82]. In the present iteration of the test case, we have chosen $(x_c, y_c) = (0.99, 0.5)$, $\epsilon = 0.005$ and $\alpha = 1000$. The peak is located in the middle of one of the boundary layers. In fig. 5.1.10, we present the contour plot of the dual solution. In fig. 5.1.11(a) and fig. 5.1.11(b), we show the convergence of the error in the target functional and dual weighted residual (DWR). We observe that the convergence rate for both error and DWR estimator is $2p+1$, where p is the polynomial order of approximation for the primal u. In this test case, we again prescribe an increase of 30% in the mesh complexity between each adaptation cycle.

In fig. 6.1.8, we present the adapted mesh for resolving the Gaussian peak. It can be easily inferred that only the right boundary layer near $x = 1.0$ is resolved as the center of the peak lies in this boundary layer. Next, we repeat the numerical experiment where we adapt the mesh while keeping the mesh complexity ($\mathscr{N}$) constant. Here also, one can notice that it takes only a few adaptation cycles to achieve a drop of $5 - 6$ orders of magnitude in error and DWR estimator. In table 5.2, we tabulate the error and the DWR

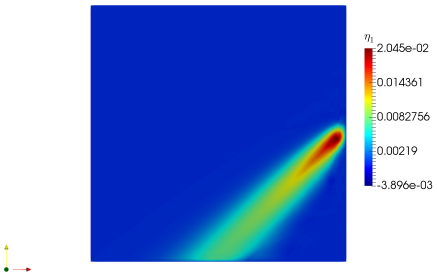
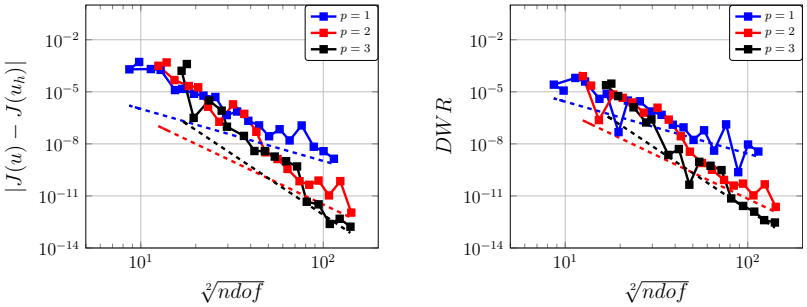


Figure 5.1.10: Boundary layer test case: contour plot for dual solution $\bar{\eta}$



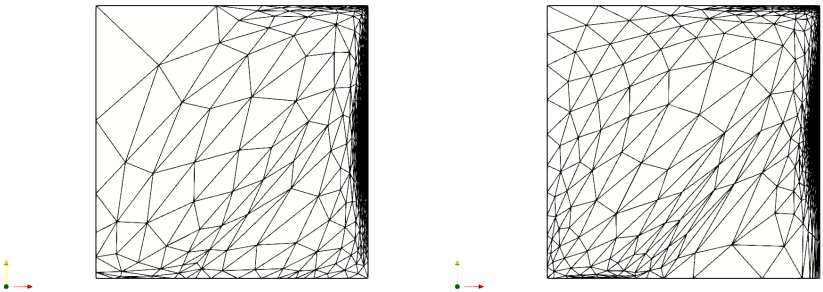
(a) **(b)**

Figure 5.1.11: Convergence plots for (a) error in target functional and (b) dual weighted residual using scaled V-norm ($\epsilon = 0.005$, expected order of convergence $(2p + 1)$ is represented by the dashed lines).

with subsequent adaptations.

Adaptation	Ne	$\lvert J(u) - J(u_h)\rvert$	DWR
0	512	0.000136149	3.55914e-05
2	436	7.47054e-09	1.03836e-08
4	469	9.99534e-10	1.75062e-09
6	470	3.56406e-10	5.92587e-10
8	478	2.35767e-10	8.58174e-10

Table 5.2: Adaptation Vs. error for constant complexity using scaled V-norm ($p = 2$, $\mathscr{N} = \int_\Omega d(\mathbf{x})\, d\mathbf{x} = 512.0 \times \frac{3\sqrt{3}}{4}$).



(a) $Ne = 1169$ **(b)** $Ne = 1178$

Figure 5.1.12: Adapated mesh for (a) $p = 2$ and (b) $p = 3$ after 15^{th} adaptation cycle using scaled V-norm.

5.2 Test case II: inverse tangent - flux target

Here, the differential operators in the governing primal PDE stays the same as in previous test case.

$$\begin{aligned} \beta \cdot \nabla u - \epsilon \nabla^2 u &= s(\mathbf{x}) \qquad \text{in } \Omega = (0,1)^2, \\ u &= g_D \qquad \text{on } \partial\Omega, \end{aligned} \tag{5.8}$$

where $\beta = [1,1]^T$. As in [91], the source term $s(\mathbf{x})$ is selected in such a way that the exact solution is given by

$$u(\mathbf{x}) = \left(tan^{-1}(\alpha(x - x_1)) + tan^{-1}(\alpha(x_2 - x))\right)\left(tan^{-1}(\alpha(y - y_1)) + tan^{-1}(\alpha(y_2 - y))\right),$$

where $x_1 = y_1 = \frac{1.0}{3.0}$, $x_2 = y_2 = \frac{2.0}{3.0}$, $\alpha = 50.0$ and $\epsilon = 0.01$. The boundary condition g_D is set to be the exact solution. Here, we focus on a flux target. The target for this problem is

$$J(u) = \int_{\partial\Omega} j_{\partial\Omega}(\mathbf{x}) \nabla u \cdot \mathbf{n} ds, \tag{5.9}$$

where we have

$$j_{\partial\Omega} = \begin{cases} 1 & \mathbf{n} = (1,0) \\ 0 & otherwise \end{cases}. \tag{5.10}$$

In this case, the dual problem has no volumetric source, while the weighting function $j_{\partial\Omega}(\mathbf{x})$ appears as a boundary condition of the dual problem. Hence, the dual problem is as follows:

$$-\beta \cdot \nabla\bar{\eta} - \epsilon\nabla^2\bar{\eta} = 0 \qquad \text{in } \Omega = (0,1)^2, \tag{5.11}$$

$$\bar{\eta} = j_{\partial\Omega} \qquad \text{on } \partial\Omega. \tag{5.12}$$

In fig. 5.2.1, we present the convergence in the target functional and DWR estimator, while the adapted mesh is shown in fig. 5.2.2. This numerical example demonstrates the need for regularization mentioned in section 4.2.1.2. In fig. 5.2.3, we present meshes obtained from three subsequent adaptations. It can be quite easily noticed that alternate refinement and coarsening of mesh elements occur at the top left corner of the domain and lead to spurious oscillation in the convergence of the target functional, even in the asymptotic regime. If regularization is employed, these oscillations tend to disappear. It can be seen in fig. 5.2.4. In fig. 5.2.5, we compare the convergence in target functional with and without regularization employed. The convergence with regularization is much smoother compared to the oscillatory convergence without regularization. While in the current setup, oscillations are not overwhelming, this is expected to be highly problem-dependent. Hence, the implementation of regularization is subjective and needs to be treated in a case-by-case manner.

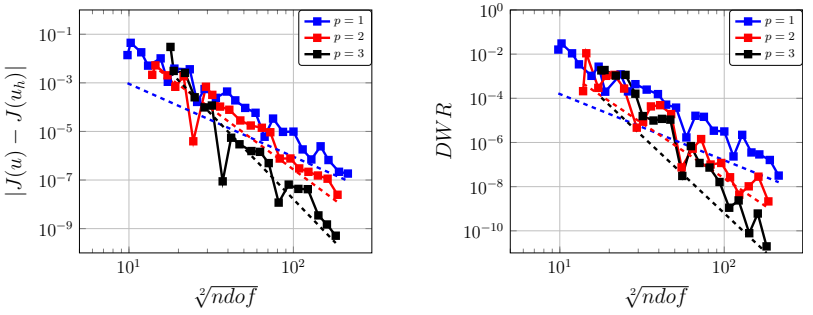
5.3 Test case III: interior line singularity

Consider the Poisson problem:

$$\begin{aligned} -\nabla^2 u &= s(\mathbf{x}) \qquad \text{in } \Omega = (0,1)^2, \\ u &= g_D \qquad \text{on } \partial\Omega, \end{aligned} \tag{5.13}$$

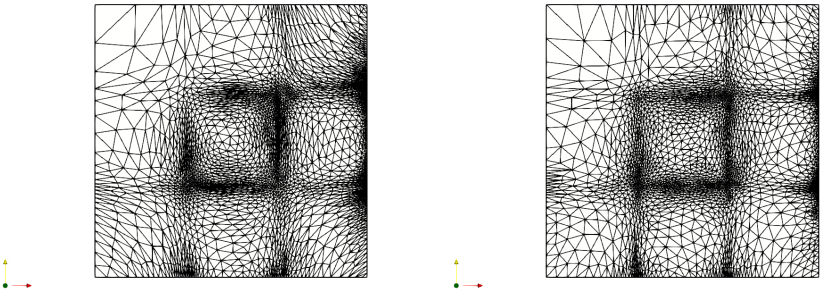
where the source term $s(\mathbf{x})$ is chosen in such a way that the exact solution is given by:

$$u(\mathbf{x}) = \begin{cases} \cos(\pi(y-0.5)), & x \leq \Theta y + 0.5, \\ \cos(\pi(y-0.5)) + (x - \Theta y - 0.5)^\gamma 2^\gamma, & x > \Theta y + 0.5. \end{cases} \tag{5.14}$$



(a) **(b)**

Figure 5.2.1: Convergence plots for (a) error in target functional and (b) dual weighted residual with regularization (section 4.2.1.2) using scaled V-norm (expected order of convergence $(2p+1)$ is represented by the dashed lines).

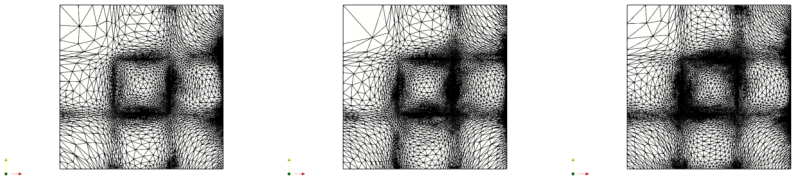


(a) $Ne = 4344$ **(b)** $Ne = 4276$

Figure 5.2.2: Adapated mesh for (a) $p = 2$ and (b) $p = 3$ after 15^{th} adaptation cycle using scaled V-norm.

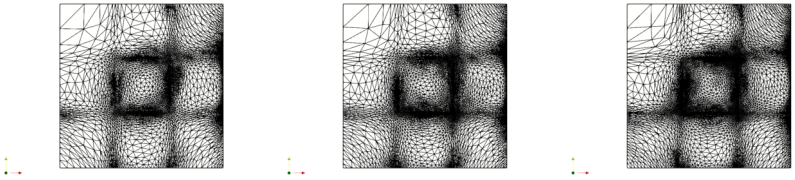
The boundary condition g_D is set to be the exact solution and $\Theta = 0.6$. A similar problem is presented in [82]. The regularity of the solution depends upon the parameter γ as the solution lies in fractional Sobolev space $H^{(\gamma+0.5-\epsilon)}\,\forall\epsilon > 0$. Consequently, the optimal order of convergence of error in u in L^2 norm is

$$min(p+1, \gamma+0.5-\epsilon), \tag{5.15}$$



(a) $Ne = 5785$ **(b)** $Ne = 7386$ **(c)** $Ne = 9742$

Figure 5.2.3: Oscillation in mesh during adaptation cycles without regularization using scaled V-norm ($p = 2$).



(a) $Ne = 5737$ **(b)** $Ne = 7412$ **(c)** $Ne = 9751$

Figure 5.2.4: Adaptation cycles with regularization using scaled V-norm ($p = 2$).

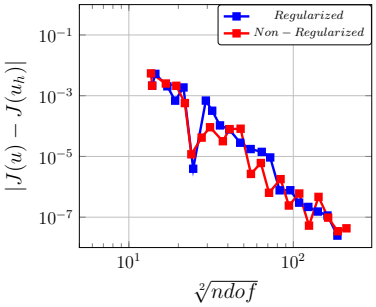


Figure 5.2.5: Comparison of convergence plots for error in target functional between regularized and non regularized adaptation cycles using scaled V-norm and $p = 2$.

where p is the order of approximation of u. Figure 5.3.1 shows the convergence plots in L^2 norm of the primary field variable u and the energy norm induced by the scaled V-norm. The primary objective of performing this numerical experiment is to test the optimality of the meshes being produced by the proposed methodology (using solution

based refinement). Since, the proposed methodology directly extracts information from the inbuilt estimator, optimality means that the mesh adapted for a given parameter γ should be sub-optimal for a different value of γ. We demonstrate this with two different values of γ ($\gamma = 2.0$ and $\gamma = 3.3$). To perform this test, we start with an isotropic mesh containing 32 elements, and we increase $\mathcal{N}$ by 30% in each adaptation cycle. The meshes are initially adapted for γ_1, and all the meshes in this adaptation run are saved. We then solve the Poisson problem on these saved meshes with $\gamma = \gamma_2$. In fig. 5.3.2, it can be observed that the optimal convergence order for primary field variable u as given by eq. (5.15) is only achieved in case of γ_1, whereas it is sub-optimal for $\gamma = \gamma_2$. This sub-optimal behavior can also be seen in the convergence plots of the energy error (fig. 5.3.3). It demonstrates the strong dependence of mesh on the problem at hand, which is achieved by directly utilizing the inbuilt error estimator. In fig. 5.3.4 and fig. 5.3.5, we present the solution for different γ and corresponding adapted mesh.

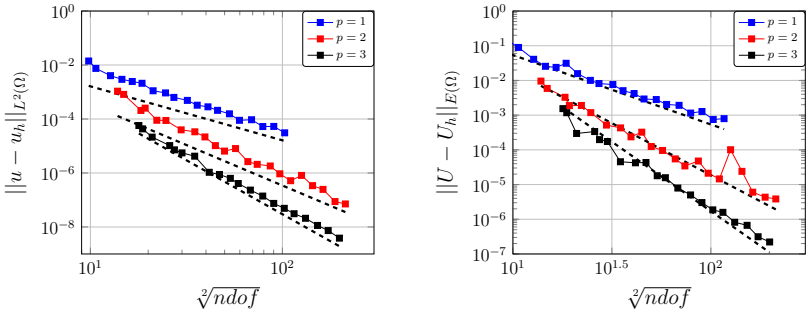


Figure 5.3.1: Convergence plots of (a) L^2 error in u_h and (b) energy error at $\gamma = 3.5$ using scaled V-norm (expected order of convergence is represented by the dashed lines).

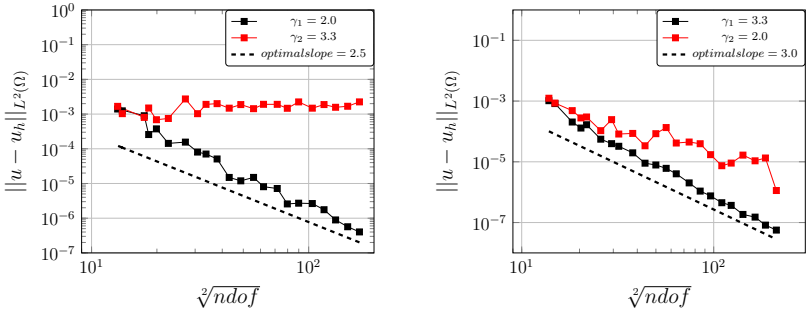


Figure 5.3.2: Convergence plots for optimal γ_1 and suboptimal γ_2 of L^2 error in u_h using scaled V-norm (expected order of convergence is represented by the dashed lines).

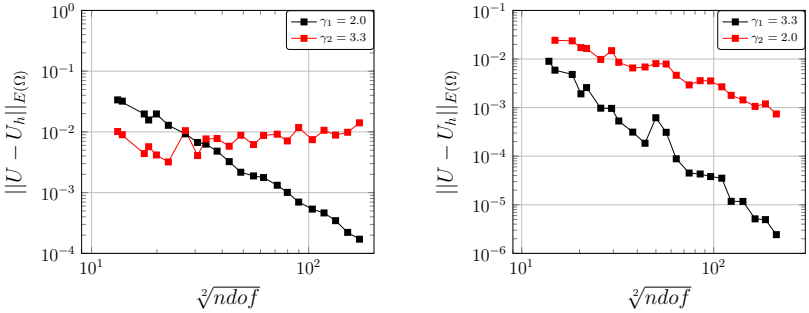


Figure 5.3.3: Convergence plots for optimal γ_1 and suboptimal γ_2 of energy error using scaled V-norm.

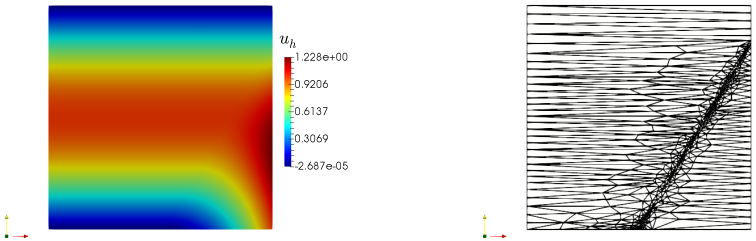


Figure 5.3.4: Solution and mesh after 15 adaptation cycles ($\gamma = 2.0, Ne = 856, p = 2$) using scaled V-norm.

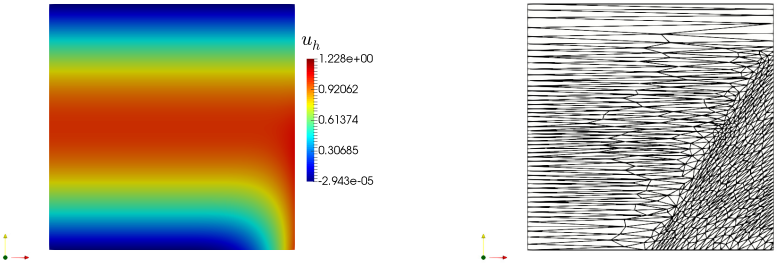


Figure 5.3.5: Solution and mesh after 15 adaptation cycles ($\gamma = 3.3, Ne = 925, p = 2$) using scaled V-norm.

5.4 Test case IV: L-shaped domain

Next, we consider the classical L-shaped domain Poisson problem:

$$\begin{aligned} -\nabla^2 u &= s(\mathbf{x}) \qquad \text{in } \Omega = (-1,1)^2 \setminus [0,1] \times [-1,0], \\ u &= g_D \qquad \text{on } \partial\Omega, \end{aligned} \tag{5.16}$$

where the source term $s(\mathbf{x})$ is chosen in such a way that the exact solution is given by:

$$u(\mathbf{x}) = r^{\frac{2}{3}} sin\left(\frac{2}{3}\theta\right) \quad where \quad \theta = tan^{-1}\left(\frac{y}{x}\right) \quad \text{and} \quad r = \sqrt{x^2 + y^2}. \tag{5.17}$$

The boundary condition g_D is the exact solution. The L-shaped domain with a r^α type solution exhibits corner singularity. The strength of the singularity is inversely proportional to α. For the present L-shaped domain, $\alpha = \frac{2.0}{3.0}$ and the singularity lies at the origin. This numerical experiment demonstrates the behavior of the proposed methodology in the presence of a singularity. Despite having a singular solution, it can be noticed that a convergence of $\mathcal{O}(h^{p+1})$ is achieved for $h-$adaptivity. We present the convergence in L^2 and energy norm in fig. 5.4.1. The exponential grading of meshes is qualitatively visible from fig. 5.4.2. In [98], Theorem 4.4 provides an expression for the optimal grading of mesh elements for degree p polynomial approximation to minimize the L^2 error in u. The same analysis can be done with a focus on ∇u such that the optimal mesh can minimize

L^2 error in ∇u. In this case, the mesh grading comes out to be,

$$h(r) = Cr^{\beta} \tag{5.18}$$

where C is a constant independent of r and

$$\beta = 1 - \frac{\alpha}{p+2}. \tag{5.19}$$

Since the energy error is equivalent to the composite L^2 error in the field variables, which is dominated by the error in ∇u, the mesh's grading is driven by ∇u rather than u. In fig. 5.4.2, we present the variation of $\sqrt{|k|}$ which is a measure of the size of an element with respect to its distance from the singularity for different polynomial orders using a log-log axis. To obtain the approximation of the exponent β, we use a linear fit to the data. The slopes of the linear fit agree with the theoretical exponent mentioned in eq. (5.19). In fig. 5.4.2, β_{obs} denotes the linear fit and β_{ref} denotes the reference values computed using eq. (5.19).

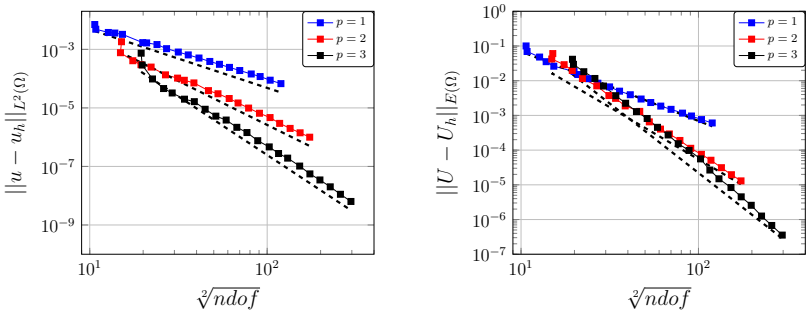
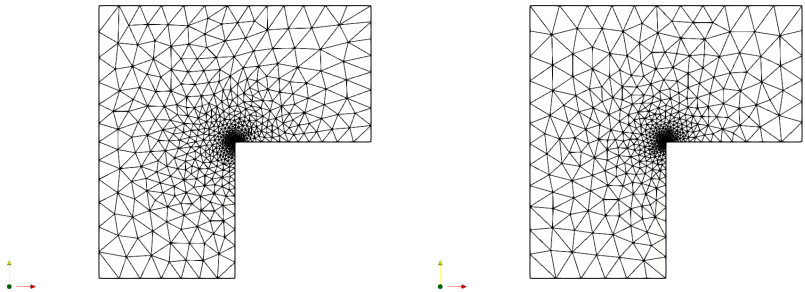


Figure 5.4.1: Convergence plots of (a) L^2 error in u_h and (b) energy norm using scaled V-norm (expected order of convergence $(p+1)$ is represented by the dashed lines).

In chapter 2, we have mentioned the importance of the discrete least square implementation. Here, in fig. 5.4.3, we present quantitative results showing the growth of condition number for adapted meshes using normal equations and DLS implementation for two different polynomial orders. As mentioned earlier, the condition number for the



(a) $Ne = 1364$ **(b)** $Ne = 1344$

Figure 5.4.2: Adapted mesh for (a) $p = 2$ and (b) $p = 3$.

normal equations has a quadratic growth rate, whereas DLS results in linear growth. It has significant implications, as using normal equations might limit the extent to which one can adapt the mesh, especially in the presence of strong features in the solution, which might require very small mesh elements and strong grading in size.

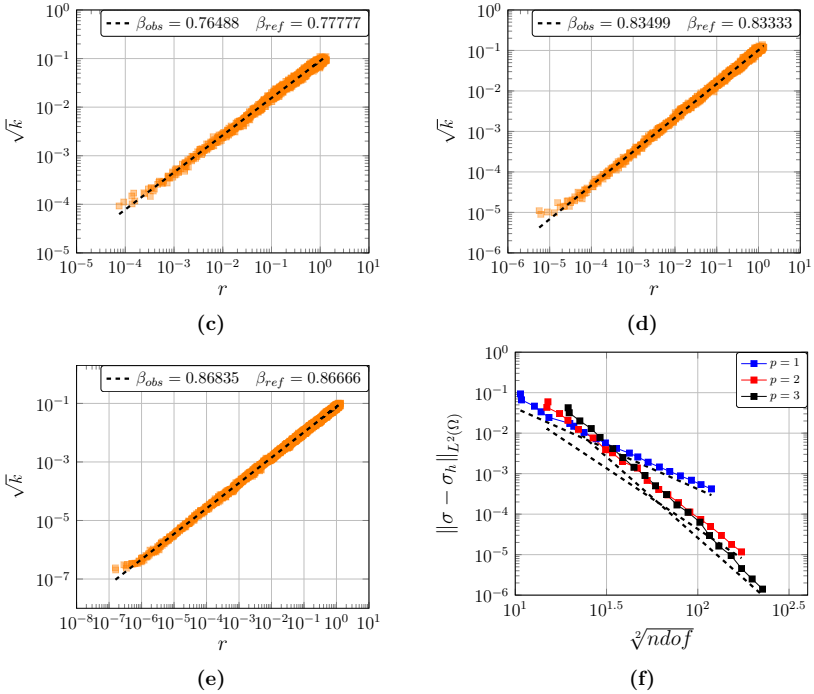


Figure 5.4.2: Grading of meshes in L-shaped domain for (a) $p = 1$, (b) $p = 2$, (c) $p = 3$ and (d) convergence in σ (expected order of convergence $(p + 1)$ is represented by the dashed lines).

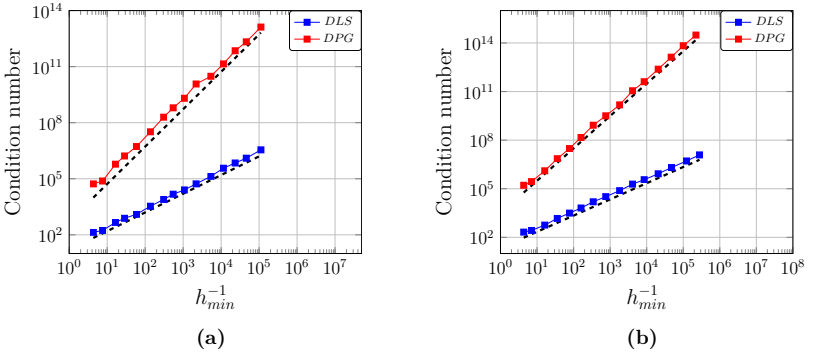


Figure 5.4.3: Comparison of condition number between DLS and DPG (normal equations) for (a) $p = 2$ and (b) $p = 3$ (expected order of growth is represented by the dashed lines).

This page is intentionally left blank.

Chapter 6

Numerical Results: hp adaptation

In the previous chapter, we presented the results showing the fidelity of the proposed $h-$adaptation algorithm. This chapter presents results for the $hp-$adaptation algorithm, which is obtained by augmenting a polynomial-order selection procedure and modifying the mesh density computation of the $h-$adaptation algorithm.

6.1 Test case I: boundary layer

For comparison of the proposed $hp-$adaptation algorithm with $h-$adaptation, we consider the same convection-diffusion problem (boundary layer) from chapter 5. For the sake of brevity for the readers, we reiterate the problem next.

$$\begin{aligned} \beta \cdot \nabla u - \epsilon \nabla^2 u &= s(\mathbf{x}) \qquad \text{in } \Omega = (0,1)^2, \\ u &= 0 \qquad \text{on } \partial\Omega, \end{aligned} \tag{6.1}$$

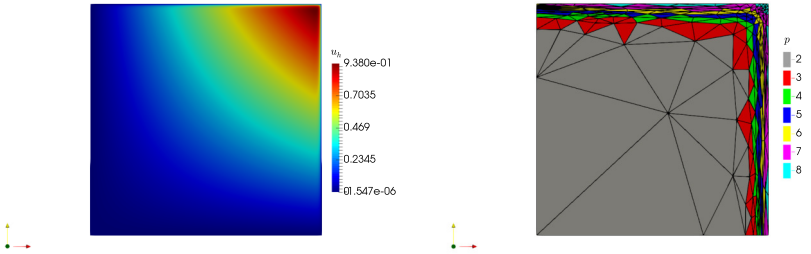
where $\beta = [1,1]^T$. The source term $s(\mathbf{x})$ is selected in such a way that the exact solution is given by

$$u(\mathbf{x}) = \left(x + \frac{e^{\frac{x}{\epsilon}} - 1}{1 - e^{\frac{1}{\epsilon}}}\right)\left(y + \frac{e^{\frac{y}{\epsilon}} - 1}{1 - e^{\frac{1}{\epsilon}}}\right). \tag{6.2}$$

In fig. 6.1.2, we present the convergence results comparing the $h-$adaptation algorithm (see section 4.2.1) and proposed $hp-$adaptation algorithm (see section 4.2.2). In case of $hp-$adaptations, we begin with the same mesh as we did in $h-$adaptation but with constant initial polynomial order of $p_{initial} = 2$. In $hp-$adaptation, $\mathscr{N}$ for the first adaptation cycle is computed as

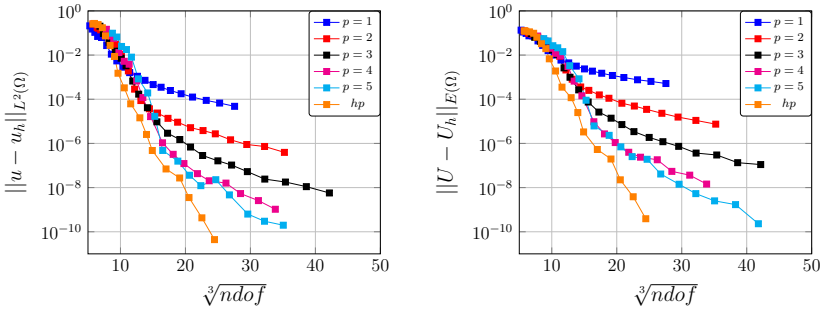
$$\mathscr{N} = Ne \times \frac{(p_{initial} + 1)\,(p_{initial} + 2)}{2} \times \frac{3\sqrt{3}}{4}, \tag{6.3}$$

where Ne represents the number of elements in the mesh. Between each adaptation cycle, $\mathscr{N}$ is increased by 30%. Similar to the case of $h-$adaptation, the choice of growth in $\mathscr{N}$ is arbitrary and a different choice in growth may result in different pre-asymptotic behavior but should produce similar asymptotic result.



(a) **(b)**

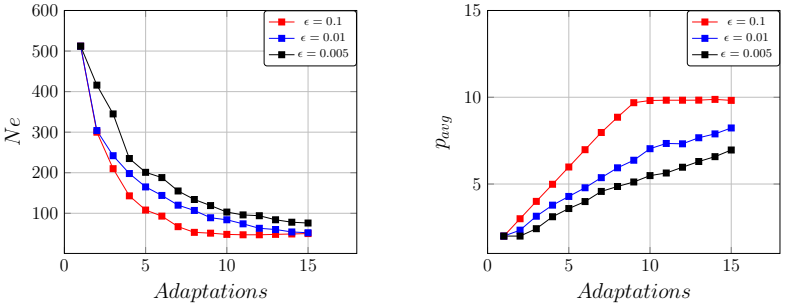
Figure 6.1.1: Boundary layer: (a) solution contour on an adapted mesh and (b) polynomial distribution on the same adapted mesh with $\epsilon = 0.005$ with 6980 degrees of freedom.



(a) **(b)**

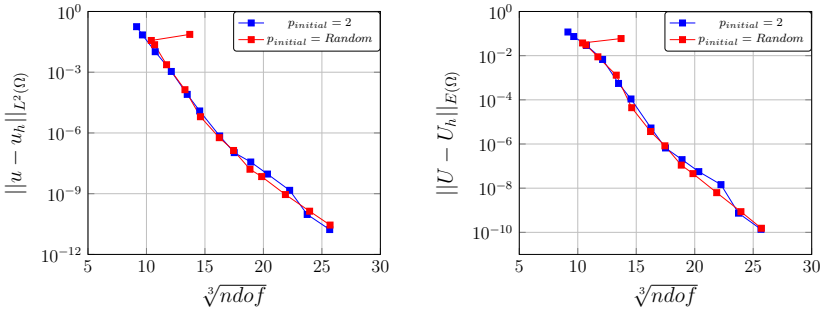
Figure 6.1.2: Convergence plots of (a) L^2 error in u_h and (b) energy norm using scaled V-norm.

On continuously increasing $\mathscr{N}$, the $hp-$adaptation algorithm produces a better convergence compared to the $h-$adaptation done at constant polynomial orders $p = 1 - 5$ as shown in fig. 6.1.2. In this section, we have replaced $\sqrt[2]{ndof}$ with $\sqrt[3]{ndof}$ on the x-axis of convergence plots to verify that we obtain exponential convergence ($\|e\|_X \approx Ce^{-bN^{1/3}}$). In



(a) **(b)**

Figure 6.1.3: Evolution of (a) number of mesh elements and (b) average polynomial order with adaptations using scaled V-norm at a fixed cost $\mathscr{N} = 3072$ for different boundary layer strengths ϵ.



(a) **(b)**

Figure 6.1.4: Convergence plots of (a) L^2 error in u_h and (b) energy norm using scaled V-norm with different initial polynomial distributions.

$hp-$adaptation, on increasing $\mathscr{N}$, initially the adaptation is dominated by $h-$refinement. Consequently, once the boundary layer is resolved, $p-$refinement starts taking place in the boundary layers along with $h-$refinements. The $hp-$algorithm prefers higher-order polynomials in elements in the boundary layer, whereas it prescribes $p = 2$ away from the boundary layer. The nature of the analytical solution is approximately quadratic away from the boundary layer, at least up to machine precision, i.e. $u(x) \approx xy$ [4]. Since with the assumption of quasi-regular elements, one can show the equivalence of scaled V-norm with L^2 norm of the field variables. We can expect to obtain the polynomial distribution

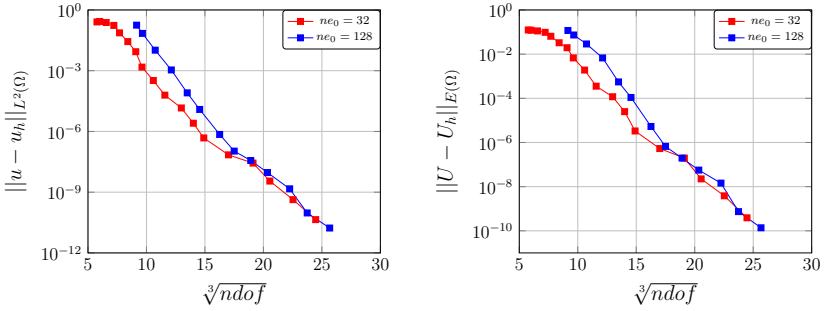


Figure 6.1.5: Convergence plots of (a) L^2 error in u_h and (b) energy norm using scaled V-norm with different initial mesh.

reflecting the local regularity of the solution variables.

Previously, while showing the convergence behavior, we kept increasing the $\mathscr{N}$ by 30%. In the next numerical experiment, we keep $\mathscr{N}$ constant and then perform the adaptation. Here, we capped the p_{max} (maximum polynomial order in the hp mesh) at 10. Since, $u \in C^\infty$, if we do not cap p_{max}, then p_{max} will keep on increasing indefinitely. Through this numerical experiment, we intend to observe the interplay between $h-$ and $p-$refinements, which is measured by computing the average polynomial order (p_{avg}) in the mesh. The evolution of p_{avg} with subsequent adaptations is shown in fig. 6.1.3 for different values of ϵ. $h-$refinements dominate initial adaptations, and it is only after some initial $h-$refinement in the boundary layers that $p-$refinements starts taking place along with $h-$refinements. This interplay between two adaptive processes is visible from the gradual decrease in the number of elements and increase in p_{avg}. In many adaptation strategies, the initial conditions such as initial mesh or distribution of the polynomial order affect the performance. In the proposed algorithms (both h and hp), arbitrary coarsening and refinements can be done due to re-meshing. Thus, it allows the meshes to be least-dependent on previous iterations. Here, we present two instances of variations in initial conditions:(a) different initial meshes but same starting polynomial order of approximation (fig. 6.1.5) (b) same initial mesh but one mesh with a random distribution of initial order

of approximation (fig. 6.1.4). We again observe (similar to the $h-$adaptation) that there is no deviation in the asymptotic behavior in the convergence plots. Thus, showing the fidelity of the proposed methodology and its robustness towards these variations.

6.1.1 Goal oriented adaptation: Gaussian peak

Next, we present results showing the fidelity of the proposed $hp-$adaptation algorithm for goal-oriented adaptations. We consider the problem of Gaussian peak with boundary layer (see section 6.1) being the primal problem. We reiterate the dual problem and the target functional. The dual problem is given by

$$-\beta \cdot \nabla \bar{\eta} - \epsilon \nabla^2 \bar{\eta} = j_\Omega(\mathbf{x}) \qquad \text{in } \Omega = (0,1)^2, \tag{6.4}$$

$$\bar{\eta} = 0 \qquad \text{on } \partial\Omega, \tag{6.5}$$

where $\beta = [1,\, 1]^T$. The target functional is given by:

$$J(u) = \int_\Omega j_\Omega(\mathbf{x}) u(\mathbf{x})\, d\mathbf{x}, \tag{6.6}$$

where

$$j_\Omega(\mathbf{x}) = e^{-\alpha\left((x-x_c)^2+(y-y_c)^2\right)}. \tag{6.7}$$

The location of the Gaussian peak is represented by $(x_c, y_c) = (0.99, 0.5)$ and $\alpha = 1000$ represents the strength of the Gaussian peak. In fig. 6.1.6, we present a three-dimensional illustration of the Gaussian peak.

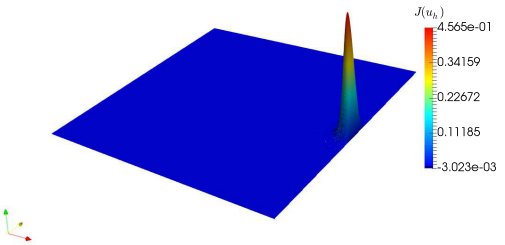


Figure 6.1.6: Contour showing $j_\Omega(\mathbf{x})$.

Next, in fig. 6.1.7(a) and fig. 6.1.7(b), we present the convergence plots comparing $h-$ and $hp-$adaptations. We obtain an exponential convergence for the $hp-$adaptations and thus, outperform the $h-$adaptations. The presence of geometrically graded meshes can account for this superior performance. We present a snapshot of one such mesh in fig. 6.1.8. On observing fig. 6.1.8, one can notice that both $h-$ and $p-$refinements mainly take place in the right boundary layer. The behavior of $h-$refinements here is very similar to the one we saw in section 5.1.2. As we previously saw in $hp-$refinements for boundary layer in section 6.1, the $h-$refinements precedes $p-$refinements, a similar trajectory is followed here. The difference here is that only one boundary layer gets $hp-$refinement, which the algorithm deems necessary for resolving the target functional. We can notice the saturation of the convergence curves in fig. 6.1.7(a) and fig. 6.1.7(b) which is due to the fact that we are nearly at machine precision and numerical round-off errors start to show up.

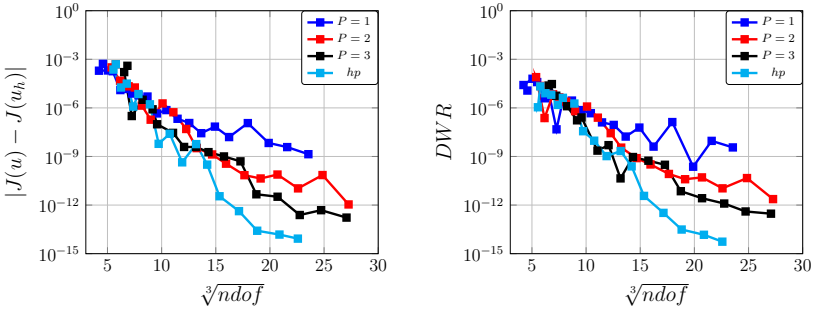


Figure 6.1.7: Convergence plots for (a) error in target functional and (b) dual weighted residual using scaled V-norm.

Next, we perform a numerical experiment where we keep $\mathcal{N}$ fixed and then run the adaptations. The purpose of this numerical experiment is same as before: to examine the capacity of the algorithm to distribute the Dofs optimally. Again, it takes only a few adaptations to reach nearly machine precision.

Adaptation	Ndof	$\lvert J(u) - J(u_h)\rvert$	DWR
0	3072	1.3599e-04	2.55956e-05
2	2698	2.62413e-09	1.16551e-09
4	3271	1.2955e-11	1.69234e-11
6	3045	2.92165e-12	2.71505e-12
8	3276	3.98067e-14	1.31114e-13

Table 6.1: Adaptation Vs. error for constant complexity using scaled V-norm ($\mathcal{N} = \int_\Omega w(\mathbf{x}) d(\mathbf{x})\, d\mathbf{x} = 3072.0$).

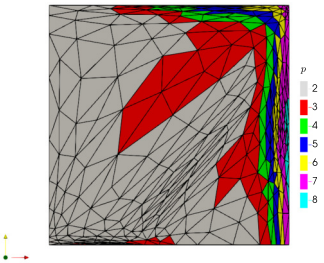


Figure 6.1.8: Polynomial distribution on an adapted mesh.

6.2 Test case II: inverse tangent - flux target

This numerical experiment compares $h-$ and $hp-$adaptations for a target functional that is not volumetric but rather a boundary integral. The differential operators in the governing PDE stay the same as the boundary layer test case, with only changes in the source term. We reiterate the PDE along with boundary conditions next.

$$\begin{aligned} \beta \cdot \nabla u - \epsilon \nabla^2 u &= s(\mathbf{x}) \qquad \text{in } \Omega = (0,1)^2, \\ u &= g_D \qquad \text{on } \partial\Omega, \end{aligned} \tag{6.8}$$

where $\beta = [1,1]^T$. The source term $s(\mathbf{x})$ is chosen in such a way that the exact solution is given by

$$u(\mathbf{x}) = \left(tan^{-1}(\alpha(x - x_1)) + tan^{-1}(\alpha(x_2 - x))\right)\left(tan^{-1}(\alpha(y - y_1)) + tan^{-1}(\alpha(y_2 - y))\right),$$

where $x_1 = y_1 = \frac{1.0}{3.0}$, $x_2 = y_2 = \frac{2.0}{3.0}$, $\alpha = 50.0$ and $\epsilon = 0.01$. The boundary condition g_D is obtained from the exact solution. The target functional is given by:

$$J(u) = \int_{\partial\Omega} j_{\partial\Omega}(\mathbf{x}) \nabla u \cdot \mathbf{n} ds, \tag{6.9}$$

where

$$j_{\partial\Omega} = \begin{cases} 1 & \mathbf{n} = (1,0) \\ 0 & otherwise \end{cases}. \tag{6.10}$$

The primal problem has no volumetric source, and the weighting function $j_{\partial\Omega}(\mathbf{x})$ shows up as the boundary condition for the dual problem:

$$-\beta \cdot \nabla \bar{\eta} - \epsilon \nabla^2 \bar{\eta} = 0 \qquad \text{in } \Omega = (0,1)^2, \tag{6.11}$$

$$\bar{\eta} = j_{\partial\Omega} \qquad \text{on } \partial\Omega. \tag{6.12}$$

In fig. 6.2.1, we present the polynomial distribution of a $hp-$adapted mesh. Since we are adapting for the flux on the right boundary and the convection is along with $[1,1]^T$, one can see that the majority of the $hp-$refinement takes place in the right lower diagonal half of the domain. It happens because the flux on the right boundary depends upon the inlet conditions of the bottom edge at $y = 0$, and hence, the dual error estimate gives more weighting to the elements in this half. In fig. 6.2.2(a) and fig. 6.2.2(b), we present the convergence results. Here also, we observe the exponential convergence in the case of $hp-$adaptations.

6.3 Test case III: L-shaped domain

Next, we present the classical L-shaped domain Poisson problem. We reiterate the problem as

$$\begin{aligned} -\nabla^2 u &= s(\mathbf{x}) \qquad \text{in } \Omega = (-1,1)^2 \setminus [0,1] \times [-1,0], \\ u &= g_D \qquad \text{on } \partial\Omega. \end{aligned} \tag{6.13}$$

Source term $s(\mathbf{x})$ is chosen in such a way that the exact solution is given by:

$$u(\mathbf{x}) = r^{\frac{2}{3}} sin\left(\frac{2}{3}\theta\right) \quad where \quad \theta = tan^{-1}\left(\frac{y}{x}\right) \quad \text{and} \quad r = \sqrt{x^2 + y^2}. \tag{6.14}$$

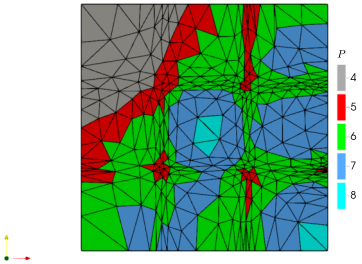


Figure 6.2.1: Polynomial distribution on an adapted mesh.

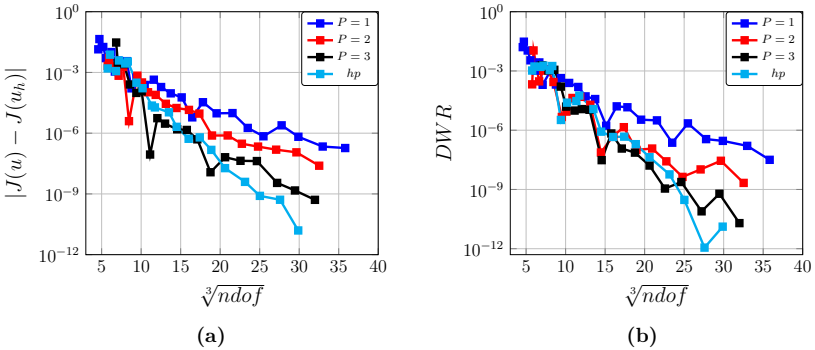
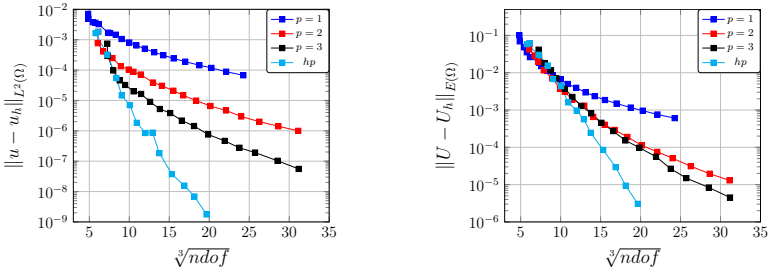


Figure 6.2.2: Convergence plots for (a) error in target functional and (b) dual weighted residual using scaled V-norm.

With this test case, we would like to demonstrate the performance of $hp-$adaptivity in the presence of a singularity. The additional flexibility of varying the local order of approximation, especially when there is a singularity, can lead to better results than a $h-$adapted mesh. In case of uniform refinement, it is expected to achieve a rate of convergence of $O(h^{\frac{4}{3}})$ in L^2 sense in u. In [98], it has already been shown that higher-order convergence (for $h-$adaptation) can be achieved using exponentially graded meshes. We achieve a similar result in section 4.2.1. However, the question arises whether we can do better than achieving a $p+1$ order in convergence?

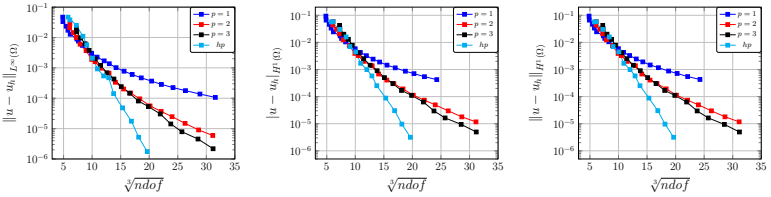
In [56] and [22], it has been shown that in order to get exponential convergence($\|e\|_X \approx Ce^{-bN^{1/3}}$), we need a geometrically graded mesh with prescribed polynomial distribution which depends upon the grading parameter. These meshes are generally hand-crafted in terms of the polynomial and mesh size distribution. Thus, requiring a priori information for mesh generation contrary to the proposed methodology where the size distribution and the polynomial order computations are automatic.

In fig. 6.3.1 and fig. 6.3.2, we observe exponential convergence in L^2 norm, energy norm, L^∞, H^1 norm and semi-norm. On comparing the $hp-$adaptation with $h-$adaptation results, one can observe that $hp-$adaptation completely outperforms $h-$adaptation for polynomial orders $p = 1 - 3$. Convergence in L^∞ becomes significant during the presence of a singularity as solution tends to blow up as singularity is approached. Hence, we provide the convergence in L^∞ in fig. 6.3.2.



(a) **(b)**

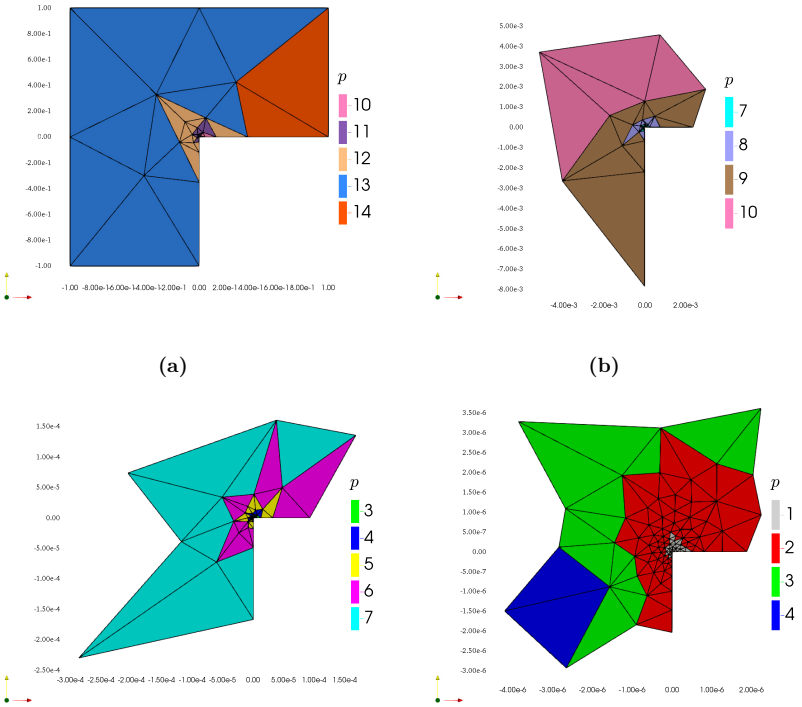
Figure 6.3.1: Convergence plots of (a) L^2 error in u_h and (b) energy norm using scaled V-norm.



(a) **(b)** **(c)**

Figure 6.3.2: Convergence plots of (a) L^∞ error, (b) H^1 seminorm error and (c) H^1 norm using scaled V-norm.

In fig. 6.3.3, we present the polynomial distribution on an adapted mesh. Each subplot represents a magnified view, with Level 0 denoting the least magnification and Level 3 denoting the most magnification. In fig. 6.3.3, it can be observed that near the singularity, we have the lowest polynomial of approximation. As we move away from the singularity, the algorithm chooses higher polynomial orders due to the smoothness of primal variables. Thus, showing the capability of the proposed $hp-$adaptation algorithm to wisely distribute the degrees of freedom utilizing the local information from the inbuilt error estimator.



(a) **(b)**

(c) **(d)**

Figure 6.3.3: Zoomed-in view showing polynomial order with (a) Level 0, (b) Level 1, (c) Level 2 and (d) Level 3 magnification.

Finally, we perform the numerical experiment where we keep the required DOFs fixed

and adapt the mesh. This numerical experiment aims to demonstrate the reduction in error at a fixed cost. In table 6.2, it can be observed that there is a reduction in both L^2 error in u_h and the energy error by three orders in magnitude.

Adap no:	0	3	5	7	10
$\|u - u_h\|_{L^2(\Omega)}$	0.000157763	$9.87e-07$	$1.34e-07$	$1.13e-07$	$1.89e-07$
$\|U - U_h\|_{E(\Omega)}$	0.0251805	$8.05e-04$	$1.24e-04$	$3.66e-05$	$1.98e-05$
p_{avg}	2	2.7	4.01	5.57	6.81
Ndof	3072	2848	3185	3238	3308

Table 6.2: Adaptation Vs. error for constant complexity using scaled V-norm ($\mathscr{N} = \int_\Omega w(\mathbf{x}) d(\mathbf{x})\, d\mathbf{x} = 3072.0$).

Chapter 7

Conclusion

7.1 Summary

The goal of this work has been to couple the metric-based adaptation and the discontinuous Petrov-Galerkin schemes (with optimal test functions). In previous works, DPG schemes with the optimal test functions have already been identified as a stable discretization method for a wide range of problems ranging from fluid flows to high-frequency wave propagation. The optimal test function framework works out-of-the-box for any approximation space. Thus, providing the flexibility of choosing the basis functions of the approximation space. An inbuilt residual-based error estimate accompanies DPG schemes with optimal test functions. This error estimate forms the core of this work along with mesh-metric duality.

Typically, constructing meshes that can resolve solution features at minimum cost is an intractable problem. Metric-based mesh generation has emerged as a promising solution to this problem. The goal of metric-based mesh generation and adaptation is to generate meshes that provide the best approximation of the discrete solution with the least cost possible. Many existing metric-based adaptation algorithms use extrinsic interpolation-based estimates to model the error in the discrete solution. These error models might not be tailor-made for a given numerical scheme that already has an inbuilt error estimator. DPG discretization with optimal test function is one such method. In this work, we combine the ideas of metric-based adaptation with the inbuilt error estimator that accompanies DPG schemes with optimal test functions. Thus, we end up complementing the so-called optimal test functions with appropriate trial functions.

Many mesh-adaptation algorithms are heuristic in nature, employing parameters that

control the algorithm's behavior. Due to this, there is always a need for an experienced practitioner to choose these parameters. In this work, metric construction is done by analytical means, and the only user-defined parameter is the user-prescribed degrees of freedom associated with the mesh. To achieve this, we employ a continuous error model, which has been previously employed for interpolation-based error models. The proposed methodology has been presented in chapter 4.

The results showing the fidelity of the $h-$adaptation algorithm are shown in chapter 5. Here, the anisotropic meshes are constructed using the energy norm for the solution-based adaptation and DPG-star based error estimator for goal-oriented adaptation. The error model has two components: (a) Anisotropy computations and (b) mesh-density computations. We have shown that the proposed $h-$adaptation algorithm gains a considerable advantage in terms of degrees of freedom associated with the mesh. We showed the algorithm's performance in the presence of boundary layers and singularity.

In chapter 6, we employ the $hp-$adaptive strategy to adapt the mesh in the presence of boundary layers and singularities as well as goal-oriented adaptations. Here, we can see that the proposed methodology results in geometrically graded meshes that produce exponential convergence both for the solution-based mesh refinement and goal-oriented mesh refinements. In the $hp-$adaptive procedure, we modified the density computation procedure to handle variable polynomial distribution and introduced a polynomial selection procedure. The polynomial selection procedure is based on the minimum DOFs required to reach the error of the current order of approximation. In $hp-$adaptations, we observed the trend that $h-$refinements generally precedes $p-$refinements. Once the region of difficulties such as boundary layer or singularity is refined spatially by $h-$refinements, then $p-$refinements starts taking place along with $h-$refinements in an intertwined way. Also, for both $h-$ and $hp-$ adaptations, we have observed that the algorithms are very robust towards initial conditions such as starting mesh or initial polynomial distribution.

7.2 Future Work

One of the major advantages that DPG provides is the automatic procedure of computing test spaces that generate discretizations that are stable and accurate. The capacity to automatically compute the optimal test space becomes indispensable while solving non-linear systems of PDE. This has been shown in multiple previous works [13, 17, 31, 33, 44, 45, 55, 61, 62, 71, 88, 100] ranging from elasticity to high frequency electro-magnetic transmission problems. Since our group would like to focus on fluid flows, a major avenue for further research will be to extend these adaptive capabilities to Euler and Navier-Stokes equations. While solving local problems during $hp-$adaptations, the boundary conditions selection for Euler equations can be done using the Eigenvectors of the Jacobian of the fluxes. For Navier-Stokes, this problem is not very clear, and hence, it will need further in-depth investigation.

The in-house code that has been used in this work runs on a single compute-core. A significant step would be to develop the existing code to harness the benefits of distributed computing. Currently, we are extending the capabilities of the code to perform distributed computing and also employ shared memory parallelism within each compute node. It will also allow us to solve large-scale problems focusing mainly on turbulent flows. For example, it would be very insightful to see the response of the automatic stabilization framework of DPG towards compressible turbulent flows and high-Mach number flows in the presence of multiple intersecting shocks. Also, distributed computing capabilities open a significant pathway for solving larger multi-physics systems such as chemically reacting and plasma flows.

In conclusion, the methodology presented in this work couples the so-called optimal test spaces with optimal approximation spaces, closing the loop for the DPG schemes. However, the presented work can be extended to other least-squares finite element methods where finite-dimensional polynomial basis functions can approximate the error estimate. Most mesh adaptation algorithms always aim to construct the best possible mesh for the

problem at hand, and this dissertation contributes to that aim.

Funding

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 333849990/GRK2379 (IRTG Modern Inverse Problems).

Bibliography

[1] NETGEN An advancing front 2D/3D-mesh generator based on abstract rules. *Computing and Visualization in Science*, 1(1):41–52, 1997. ISSN 1432-9360. doi: 10.1007/s007910050004.

[2] *Metrics, Definitions and Properties*, chapter 1, pages 1–51. John Wiley & Sons, Ltd, 2019. ISBN 9781119384380. doi: 10.1002/9781119384380.ch1.

[3] The double adaptivity paradigm : (how to circumvent the discrete inf–sup conditions of babuska and brezzi). *Computers & Mathematics with Applications*, 2020. ISSN 0898-1221. doi: 10.1016/j.camwa.2020.10.002.

[4] R. Ajay. Metric based hp−adaptation using a continuous mesh model for higher order schemes (PhD thesis). *RWTH Aachen*, 2021.

[5] I. Babuska. Error-bounds for finite element method. *Numerische Mathematik*, 16: 322–333, 1971.

[6] A. Balan, M. Woopen, and G. May. Hp-adaptivity on anisotropic meshes for hybridized discontinuous Galerkin schemes. *AIAA Paper 2015-2606*, 2015.

[7] S. Balay, J. Brown, K. Buschelman, V.Eijkhout, W.D.Gropp, D.Kaushik, M.G.Knepley, L. adn B.F.Smith, and H.Zhang. PETSc users manual. *Technical Report ANL-95/11 - Revision 3.3, Aragonne National Laboratory*, 2012.

[8] R. Becker and R. Rannacher. An optimal control approach to a posteriori error estimation in finite element methods. *Acta Numerica*, 10:1–102, 2001. doi: 10.1017/S0962492901000010.

[9] P. Bochev and M. Gunzburger. *Least-Squares Finite Element Methods*, volume 166. 01 2006. ISBN 978-0-387-30888-3. doi: 10.1007/b13382.

[10] H. Borouchaki, F. Hecht, and P. J. Frey. Mesh gradation control. *International Journal for Numerical Methods in Engineering*, 43(6):1143–1165, 1998. ISSN 00295981. doi: 10.1002/(SICI)1097-0207(19981130)43:6<1143::AID-NME470>3.0.CO;2-I.

[11] J. Bramble, R. Lazarov, and J. Pasciak. A least-squares approach based on a discrete minus one inner product for first order systems. *Mathematics of Computation*, 66: 935–955, 07 1997.

[12] J. H. Bramble and J. E. Pasciak. A new approximation technique for div-curl systems. *Mathematics of Computation*, 73(248):1739–1762, 2004.

[13] J. Bramwell, L. Demkowicz, J. Gopalakrishnan, and W. Qiu. A locking-free hp DPG method for linear elasticity with symmetric stresses. *Numerische Mathematik*, 122 (4):671–707, 2012. ISSN 0945-3245. doi: 10.1007/s00211-012-0476-6.

[14] D. Broersen and R. Stevenson. A robust petrov–galerkin discretisation of convection–diffusion equations. *Computers & Mathematics with Applications*, 68(11): 1605–1618, 2014. ISSN 0898-1221. doi: 10.1016/j.camwa.2014.06.019.

[15] D. Broersen and R. Stevenson. A petrov–galerkin discretization with optimal test space of a mild-weak formulation of convection–diffusion equations in mixed form. *IMA Journal of Numerical Analysis*, 35(1):39–73, 02 2014. ISSN 0272-4979. doi: 10.1093/imanum/dru003.

[16] D. Broersen, W. Dahmen, and R. Stevenson. On the stability of dpg formulations of transport equations. *Mathematics of Computation*, 87, 10 2015. doi: 10.1090/mcom/3242.

[17] C. Carstensen and F. Hellwig. Low-order discontinuous petrov–galerkin finite element methods for linear elasticity. *SIAM Journal on Numerical Analysis*, 54(6):3388–3410, 2016. doi: 10.1137/15M1032582.

[18] M. Castro-Diaz, F. Hecht, B. Mohammadi, and O. Pironneau. Anisotropic unstructured mesh adaption for flow simulations. *International Journal for Numerical Methods in Fluids*, 25:475–491, 1997.

[19] M. Ceze and K. J. Fidkowski. Anisotropic hp-adaptation framework for functional prediction. *AIAA Journal*, 51(2):492–509, 2013. doi: 10.2514/1.J051845.

[20] J. Chan, N. Heuer, T. Bui-Thanh, and L. Demkowicz. A robust dpg method for convection-dominated diffusion problems ii: Adjoint boundary conditions and mesh-dependent test norms. *Computers & Mathematics with Applications*, 67:771–795, 2014.

[21] A. Cohen, W. Dahmen, and G. Welper. Adaptivity and variational stabilization for convection-diffusion equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 46(5):1247–1273, 2012. ISSN 0764583X. doi: 10.1051/m2an/2012003.

[22] L. B. da Veiga, A. Chernov, L. Mascotto, and A. Russo. Exponential convergence of the hp virtual element method in presence of corner singularities. *Numerische Mathematik*, 138(3):581–613, 2018. ISSN 0945-3245. doi: 10.1007/s00211-017-0921-7.

[23] J. Dahm. Toward Accurate, Efficient, and Robust Hybridized Discontinuous Galerkin Methods (PhD thesis). *The University of Michigan, Ann Arbor*, 2017.

[24] W. Dahmen, C. Huang, C. Schwab, and G. Welper. Adaptive Petrov–Galerkin methods for first order transport equations. *SIAM J. Num. Anal.*, 50(5):2420–2445, 2012.

[25] L. Demkowicz and J. Gopalakrishnan. A class of discontinuous Petrov-Galerkin methods. Part I: The transport equation. *Computer Methods in Applied Mechanics and Engineering*, 199(23-24):1558–1572, 2010. ISSN 00457825. doi: 10.1016/j.cma.2010.01.003.

[26] L. Demkowicz and J. Gopalakrishnan. Analysis of the DPG Method for the Poisson Equation. *SIAM Journal on Numerical Analysis*, 49(5):1788–1809, 2011. ISSN 0036-1429. doi: 10.1137/100809799.

[27] L. Demkowicz and J. Gopalakrishnan. Analysis of the dpg method for the poisson equation. *SIAM Journal on Numerical Analysis*, 49:1788–1809, 2011.

[28] L. Demkowicz and J. Gopalakrishnan. A class of discontinuous petrov–galerkin methods. ii. optimal test functions. *Numerical Methods for Partial Differential Equations*, 27(1):70–105, 2011. doi: 10.1002/num.20640.

[29] L. Demkowicz and N. Heuer. Robust DPG method for convection-dominated diffusion problems. *SIAM Journal on Numerical Analysis*, 51(5):2514–2537, 2013. ISSN 00361429. doi: 10.1137/120862065.

[30] L. Demkowicz, W. Rachowicz, and P. Devloo. A fully automatic hp-adaptivity. *Journal of Scientific Computing*, 17, 12 2002. doi: 10.1023/A:1015192312705.

[31] L. Demkowicz, J. Gopalakrishnan, I. Muga, and J. Zitelli. Wavenumber explicit analysis of a dpg method for the multidimensional helmholtz equation. *Computer Methods in Applied Mechanics and Engineering*, 213-216:126–138, 2012. ISSN 0045-7825. doi: 10.1016/j.cma.2011.11.024.

[32] L. Demkowicz, J. Gopalakrishnan, and A. Niemi. A class of discontinuous Petrov-Galerkin methods. Part III: Adaptivity. *Applied Numerical Mathematics*, 62:396–427, 2013.

[33] L. Demkowicz, J. Gopalakrishnan, S. Nagaraj, and P. Sepulveda. A spacetime dpg method for the schrödinger equation. *SIAM Journal on Numerical Analysis*, 55, 10 2016. doi: 10.1137/16M1099765.

[34] L. Demkowicz, J. Gopalakrishnan, and B. Keith. The dpg-star method. *Computers & Mathematics with Applications*, 79(11):3092 – 3116, 2020. ISSN 0898-1221. doi: 10.1016/j.camwa.2020.01.012.

[35] V. Dolej. Anisotropic hp adaptive method based on interpolation error estimates in the Lq norm. 82(13):80–114, 2014. doi: 10.1016/j.apnum.2014.03.003.

[36] V. Dolejsi. A design of residual error estimates for a high order bdf-dgfe method applied to compressible flows. *International Journal for Numerical Methods in Fluids*, 73, 10 2013. doi: 10.1002/fld.3811.

[37] V. Dolejsi. Anisotropic hp -adaptive method based on interpolation error estimates in the L^q -norm. 82(13):80–114, 2014. doi: 10.1016/j.apnum.2014.03.003.

[38] V. Dolejsi. Anisotropic hp-adaptive method based on interpolation error estimates in the h1-seminorm. *Applications of Mathematics*, 60(6):597–616, 2015. ISSN 0862-7940. doi: 10.1007/s10492-015-0113-7.

[39] V. Dolejsi, G. May, F. Roskovec, and P. Solin. Anisotropic hp-mesh optimization technique based on the continuous mesh and error models. *Computers & Mathematics with Applications*, 74(1):45–63, 2017. ISSN 08981221. doi: 10.1016/j.camwa.2016.12.015.

[40] V. Dolejší, G. May, and A. Rangarajan. A continuous hp-mesh model for adaptive discontinuous Galerkin schemes. *Applied Numerical Mathematics*, 124(September): 1–21, 2018. ISSN 01689274. doi: 10.1016/j.apnum.2017.09.015.

[41] V. Dolejší, G. May, A. Rangarajan, and F. Roskovec. A Goal-Oriented High-Order Anisotropic Mesh Adaptation Using Discontinuous Galerkin Method for Linear Convection-Diffusion-Reaction Problems. *SIAM Journal on Scientific Computing*, 41(3):A1899–A1922, 2019. ISSN 1064-8275.

[42] K. J. Fidkowski. Comparison of hybrid and standard discontinuous galerkin methods in a mesh-optimisation setting. *International Journal of Computational Fluid Dynamics*, 33(1-2):34–42, 2019. doi: 10.1080/10618562.2019.1588962.

[43] L. Formaggia and S. Perotto. New anisotropic a priori error estimates. *Numerische Mathematik*, 89(4):641–667, 2001. ISSN 0945-3245. doi: 10.1007/s002110100273.

[44] F. Fuentes, B. Keith, L. Demkowicz, and P. Tallec. Coupled variational formulations of linear elasticity and the dpg methodology. *Journal of Computational Physics*, 348, 09 2016. doi: 10.1016/j.jcp.2017.07.051.

[45] F. Fuentes, L. Demkowicz, and A. Wilder. Using a dpg method to validate dma experimental calibration of viscoelastic materials. *Computer Methods in Applied Mechanics and Engineering*, 325:748–765, 2017. ISSN 0045-7825. doi: 10.1016/j.cma.2017.07.012.

[46] T. Führer and N. Heuer. Robust coupling of dpg and bem for a singularly perturbed transmission problem. *Computers & Mathematics with Applications*, 74(8):1940–1954, 2017. ISSN 0898-1221. doi: 10.1016/j.camwa.2016.09.016.

[47] T. Führer, N. Heuer, and M. Karkulik. On the coupling of dpg and bem. *Mathematics of Computation*, 86, 08 2015. doi: 10.1090/mcom/3170.

[48] T. Führer, N. Heuer, and E. Stephan. On the dpg method for signorini problems. *IMA Journal of Numerical Analysis*, 38, 09 2016. doi: 10.1093/imanum/drx048.

[49] T. Führer, N. Heuer, and J. S. Gupta. A time-stepping dpg scheme for the heat equation. *Computational Methods in Applied Mathematics*, 17(2):237–252, 2017. doi: doi:10.1515/cmam-2016-0037.

[50] T. Führer, N. Heuer, M. Karkulik, and R. Rodríguez. Combining the dpg method with finite elements. *Computational Methods in Applied Mathematics*, 18(4):639–652, 2018. doi: doi:10.1515/cmam-2017-0041.

[51] P. George, F. Hecht, and M. Vallet. Creation of internal points in voronoi's type method. control adaptation. *Advances in Engineering Software and Workstations*, 13(5):303–312, 1991. ISSN 0961-3552. doi: 10.1016/0961-3552(91)90034-2.

[52] S. Giani and P. Houston. Anisotropic hp-adaptive discontinuous galerkin finite element methods for compressible fluid flows. *International journal of numerical analysis and modeling.*, 9(4):928–949, January 2012.

[53] G. Golub. Numerical methods for solving linear least squares problems. *Numerische Mathematik*, 7(3):206–216, 1965. ISSN 0945-3245. doi: 10.1007/BF01436075.

[54] J. GOPALAKRISHNAN and W. QIU. An analysis of the practical dpg method. *Mathematics of Computation*, 83(286):537–552, 2014.

[55] J. Gopalakrishnan, I. Muga, and N. Olivares. Dispersive and dissipative errors in the dpg method with scaled norms for helmholtz equation. *SIAM Journal on Scientific Computing*, 36(1):A20–A39, 2014. doi: 10.1137/130918186.

[56] B. Guo and I. Babuška. The h-p version of the finite element method. *Computational Mechanics*, 1(3):203–220, 1986. ISSN 1432-0924. doi: 10.1007/BF00272624.

[57] W. Habashi, J. Dompierre, Y. Bourgault, D. Ait-Ali, M. Fortin, and M.-G. Vallet. Anisotropic mesh adaptation: Towards user-independent, mesh-independent and solver-independent cfd. part i: General principles. *International Journal for Numerical Methods in Fluids*, 32:725 – 744, 03 2000. doi: 10.1002/(SICI)1097-0363(20000330)32:6<725::AID-FLD935>3.0.CO;2-4.

[58] R. Hartmann. Adjoint consistency analysis of discontinuous galerkin discretizations. *SIAM J. Numer. Anal.*, 45:2671–2696, 01 2007. doi: 10.1137/060665117.

[59] F. Hecht. BAMG: Bidimensional anisotropic mesh generator. 2006. URL https://www.ljll.math.upmc.fr/hecht/ftp/bamg.

[60] F. Hecht, B. Mohammadi, F. Hecht, and B. Mohammadi. Mesh adaption by metric control for multi-scale phenomena and turbulence. doi: 10.2514/6.1997-859.

[61] S. Henneking and L. Demkowicz. A numerical study of the pollution error and dpg adaptivity for long waveguide simulations. *Computers & Mathematics with Applications*, 95, 04 2020. doi: 10.1016/j.camwa.2020.03.024.

[62] S. Henneking, J. Grosek, and L. Demkowicz. Model and computational advancements to full vectorial maxwell model for studying fiber amplifiers. *Computers & Mathematics with Applications*, 85:30–41, 03 2021. doi: 10.1016/j.camwa.2021.01.006.

[63] N. Heuer and M. Karkulik. Dpg method with optimal test functions for a transmission problem. *Computers & Mathematics with Applications*, 70(5):1070–1081, 2015. ISSN 0898-1221. doi: 10.1016/j.camwa.2015.06.032.

[64] N. Heuer and M. Karkulik. A robust dpg method for singularly perturbed reaction-diffusion problems. *SIAM J. Numer. Anal.*, 55:1218–1242, 2017.

[65] N. Heuer and M. Karkulik. Discontinuous Petrov–Galerkin boundary elements. *Numerische Mathematik*, 135(4):1011–1043, 2017. ISSN 0945-3245. doi: 10.1007/s00211-016-0824-z.

[66] N. Heuer, M. Karkulik, and F. J. Sayas. Note on discontinuous trace approximation in the practical dpg method. *Computers & Mathematics with Applications*, 68: 1562–1568, 2014.

[67] P. Houston. hp-adaptive discontinuous galerkin finite element methods for first-order hyperbolic problems. *SIAM Journal on Scientific Computing*, 23, 10 2000. doi: 10.1137/S1064827500378799.

[68] P. Houston. hp-adaptive discontinuous galerkin finite element methods for first-order hyperbolic problems. *SIAM Journal on Scientific Computing*, 23, 10 2000. doi: 10.1137/S1064827500378799.

[69] B. Keith. New ideas in adjoint methods for PDEs : A saddle-point paradigm for finite element analysis and its role in the DPG methodology (PhD thesis). *The University of Texas at Austin*, 2018.

[70] B. Keith, F. Fuentes, and L. Demkowicz. The DPG methodology applied to different variational formulations of linear elasticity. *Computer Methods in Applied Mechanics and Engineering*, 309:579–609, 2016. ISSN 00457825. doi: 10.1016/j.cma.2016.05.034.

[71] B. Keith, F. Fuentes, and L. Demkowicz. The dpg methodology applied to different variational formulations of linear elasticity. *Computer Methods in Applied Mechanics and Engineering*, 309, 01 2016. doi: 10.1016/j.cma.2016.05.034.

[72] B. Keith, S. Petrides, F. Fuentes, and L. Demkowicz. Discrete least-squares finite element methods. *Computer Methods in Applied Mechanics and Engineering*, 327: 226 – 255, 2017. ISSN 0045-7825. doi: 10.1016/j.cma.2017.08.043. Advances in Computational Mechanics and Scientific Computation—the Cutting Edge.

[73] B. Keith, A. V. Astaneh, and L. F. Demkowicz. Goal-oriented adaptive mesh refinement for discontinuous petrov–galerkin methods. *SIAM Journal on Numerical Analysis*, 57(4):1649–1676, 2019. doi: 10.1137/18M1181754.

[74] T. Leicht and R. Hartmann. Anisotropic mesh refinement for discontinuous galerkin methods in two-dimensional aerodynamic flow simulations. *International Journal for Numerical Methods in Fluids*, 56(11):2111–2138, 2008. doi: 10.1002/fld.1608.

[75] T. Leicht and R. Hartmann. Error estimation and anisotropic mesh refinement for 3d laminar aerodynamic flow simulations. *Journal of Computational Physics*, 229 (19):7344 – 7360, 2010. ISSN 0021-9991. doi: 10.1016/j.jcp.2010.06.019.

[76] A. Loseille and F. Alauzet. Continuous mesh framework part i: Well-posed continuous interpolation error. *SIAM Journal on Numerical Analysis*, 49(1):38–60, 2011. doi: 10.1137/090754078.

[77] A. Loseille and F. Alauzet. Continuous mesh framework part ii: Validations and applications. *SIAM Journal on Numerical Analysis*, 49(1):61–86, 2011. doi: 10.1137/10078654X.

[78] R. Löhner. Adaptive remeshing for transient problems. *Computer Methods in Applied Mechanics and Engineering*, 75(1):195–214, 1989. ISSN 0045-7825. doi: 10.1016/0045-7825(89)90024-8.

[79] D. Mavriplis. Multigrid strategies for viscous flow solvers on anisotropic unstructured meshes. *Journal of Computational Physics*, 145(1):141 – 165, 1998. ISSN 0021-9991. doi: 10.1006/jcph.1998.6036.

[80] D. J. Mavriplis. Adaptive mesh generation for viscous flows using delaunay triangulation. *Journal of Computational Physics*, 90:271–291, 1990.

[81] D. McRae. r-refinement grid adaptation algorithms and issues. *Computer Methods in Applied Mechanics and Engineering*, 189(4):1161–1182, 2000. ISSN 0045-7825. doi: 10.1016/S0045-7825(99)00372-2. Adaptive Methods for Compressible CFD.

[82] W. F. Mitchell. A collection of 2d elliptic problems for testing adaptive grid refinement algorithms. *Applied Mathematics and Computation*, 220:350 – 364, 2013. ISSN 0096-3003. doi: 10.1016/j.amc.2013.05.068.

[83] A. H. Niemi, J. A. Bramwell, and L. F. Demkowicz. Discontinuous petrov–galerkin method with optimal test functions for thin-body problems in solid mechanics. *Computer Methods in Applied Mechanics and Engineering*, 200(9):1291–1300, 2011. ISSN 0045-7825. doi: 10.1016/j.cma.2010.10.018.

[84] A. H. Niemi, N. Collier, and V. Calo. Automatically stable discontinuous petrov-galerkin methods for stationary transport problems: Quasi-optimal test space norm. *Computers & Mathematics with Applications*, 66, 01 2012. doi: 10.1016/j.camwa.2013.07.016.

[85] X. Pennec, P. Fillard, N. Ayache, and P. Epidaure. A riemannian framework for tensor computing. *International Journal of Computer Vision*, 66, 08 2004. doi: 10.1007/s11263-005-3222-z.

[86] J. Peraire, M. Vahdati, K. Morgan, and O. Zienkiewicz. Adaptive remeshing for compressible flow computations. *Journal of Computational Physics*, 72(2):449–466, 1987. ISSN 0021-9991. doi: 10.1016/0021-9991(87)90093-3.

[87] J. Peraire, J. Peiró, and K. Morgan. Adaptive remeshing for three-dimensional compressible flow computations. *Journal of Computational Physics*, 103(2):269–285, 1992. ISSN 0021-9991. doi: 10.1016/0021-9991(92)90401-J.

[88] S. Petrides and L. Demkowicz. An adaptive dpg method for high frequency time-harmonic wave propagation problems. *Computers & Mathematics with Applications*, 74, 07 2017. doi: 10.1016/j.camwa.2017.06.044.

[89] A. Rangarajan, A. Balan, and G. May. Mesh optimization for discontinuous Galerkin methods using a continuous mesh model. *AIAA Journal*, 56(10):4060–4073, 2018. ISSN 00011452. doi: 10.2514/1.J056965.

[90] A. Rangarajan, G. May, and V. Dolejsi. Adjoint-based anisotropic hp-adaptation for discontinuous galerkin methods using a continuous mesh model. *Journal of Computational Physics*, 409:109321, 2020. ISSN 0021-9991. doi: 10.1016/j.jcp.2020.109321.

[91] N. Ringue and S. Nadarajah. An optimization-based framework for anisotropic hp-adaptation of high-order discretizations. *Journal of Computational Physics*, 375: 589–618, 2018. ISSN 0021-9991. doi: 10.1016/j.jcp.2018.09.005.

[92] J. R. Shewchuk. What is a good linear finite element? - interpolation, conditioning, anisotropy, and quality measures. Technical report, In Proc. of the 11th International Meshing Roundtable, 2002.

[93] L. N. Trefethen and D. Bau. *Numerical Linear Algebra*. SIAM, 1997. ISBN 0898713617.

[94] A. Vaziri Astaneh, F. Fuentes, J. Mora, and L. Demkowicz. High-order polygonal discontinuous Petrov–Galerkin (PolyDPG) methods using ultraweak formulations. *Computer Methods in Applied Mechanics and Engineering*, 332(July):686–711, 2018. ISSN 00457825. doi: 10.1016/j.cma.2017.12.011.

[95] D. A. Venditti and D. L. Darmofal. Grid adaptation for functional outputs: Application to two-dimensional inviscid flows. *Journal of Computational Physics*, 176(1): 40 – 69, 2002. ISSN 0021-9991. doi: 10.1006/jcph.2001.6967.

[96] D. A. Venditti and D. L. Darmofal. Anisotropic grid adaptation for functional outputs: application to two-dimensional viscous flows. *Journal of Computational Physics*, 187(1):22 – 46, 2003. ISSN 0021-9991.

[97] M. Woopen, G. May, and J. Schütz. Adjoint-based error estimation and mesh adaptation for hybridized discontinuous Galerkin methods. *International Journal for Numerical Methods in Fluids*, 76(11):811–834, 2014. ISSN 10970363. doi: 10.1002/fld.3959.

[98] M. Yano. An Optimization Framework for Adaptive Higher-Order Discretizations of Partial Differential Equations on Anisotropic Simplex Meshes (PhD thesis). *Massachusetts Institute of Technology*, 2012.

[99] M. Yano and D. L. Darmofal. An optimization-based framework for anisotropic simplex mesh adaptation. *Journal of Computational Physics*, 231(22):7626–7649, 2012.

[100] J. Zitelli, I. Muga, L. Demkowicz, J. Gopalakrishnan, D. Pardo, and V. M. Calo. A class of discontinuous Petrov-Galerkin methods. Part IV: The optimal test norm and time-harmonic wave propagation in 1D. *Journal of Computational Physics*, 230 (7):2406–2432, 2011. ISSN 10902716. doi: 10.1016/j.jcp.2010.12.001.