

DISCONTINUOUS GALERKIN METHODS FOR HAMILTON-JACOBI EQUATIONS  
AND HIGH-DIMENSIONAL ELLIPTIC EQUATIONS

By

Zixuan Wang

A DISSERTATION

Submitted to  
Michigan State University  
in partial fulfillment of the requirements  
for the degree of

Applied Mathematics – Doctor of Philosophy

2015

## ABSTRACT

# DISCONTINUOUS GALERKIN METHODS FOR HAMILTON-JACOBI EQUATIONS AND HIGH-DIMENSIONAL ELLIPTIC EQUATIONS

By

Zixuan Wang

This thesis focuses on two related topics, which are to design efficient discontinuous Galerkin (DG) schemes for Hamilton-Jacobi (HJ) equations and high-dimensional elliptic equations.

In the first part, we propose a new DG method that solves for the viscosity solution of the general HJ equations. The new method is compact and easy to implement. We avoid the reconstruction of the solution across elements by utilizing the interfacial terms involving the Roe speed. A penalty term proportional to the jump of the normal derivative of the numerical solution is added to fix the entropy violation, which was inspired by the Harten and Hymans entropy fix [53] for Roe scheme for the conservation laws. Numerical experiments demonstrate good performance for general Hamiltonians, including nonconvex Hamiltonians.

In the second part, we develop an interior penalty DG method on sparse grids for efficient computations of high-dimensional second-order elliptic problems. Using a hierarchical basis representation, we construct a sparse finite element approximation space, reducing the degree of freedom from the standard  $O(h^{-d})$  to  $O(h^{-1}|\log_2 h|^{d-1})$  for  $d$ -dimensional problems, where  $h$  is the uniform mesh size in each dimension. Compared to the traditional full grid approaches, the accuracy of the numerical approximation of this method is only slightly deteriorated by a factor of  $|\log_2 h|^{d-1}$  in the energy norm. Error estimates are provided and confirmed by numerical tests in multi-dimensions.

## ACKNOWLEDGMENTS

First and foremost, I want to express my deepest appreciation to my advisor, Professor Yingda Cheng, for her invaluable encouragement, guidance and support during my graduate study. It is truly a great honor for me to be her first Ph.D. student. She has been a tremendous mentor and has always been there for me whenever I need her help. This dissertation would not have been possible without her expertise in numerical analysis and inspirational suggestions on the projects.

My sincere thanks also go to Professor Andrew Christlieb, for leading me into the field of scientific computing and offering insightful advices throughout these years. I would also like to thank the rest of my committee members, Professor Jianliang Qian for teaching me lots of knowledge, and Professor Chichia Chiu and Professor Zhengfang Zhou for their critical comments and valuable feedbacks. And I am indebted to my undergraduate advisor Professor Kangsheng Liu for his academic support and for giving me so many wonderful opportunities. I also want to acknowledge my dear friends Yuan Liu, Wei Guo, Xinghui Zhong, Xianfeng Hu, Liping Chen, Liangmin Zhou, Xin Yang and Qinfeng Gao for their friendships and helps.

Last but not the least, I am eternally grateful to my parents, Jiancheng Wang and Junyan Li, and my little sister Zirun Wang, whose persistent support and unconditional love made me who I am today. My deepest gratitude goes to my husband, Qi Tang, for loving me and always being there for me through this entire journey. Your love makes me feel at home wherever we go.

# TABLE OF CONTENTS

<b>LIST OF TABLES</b> . . . . .	<b>v</b>
<b>LIST OF FIGURES</b> . . . . .	<b>viii</b>
<b>Chapter 1 Introduction</b> . . . . .	<b>1</b>
1.1 Overview . . . . .	1
1.2 Numerical Methods for HJ Equations . . . . .	2
1.3 Review of DG Schemes for Elliptic Equations . . . . .	4
1.4 Sparse Grid Methods for High-dimensional PDEs . . . . .	5
<b>Chapter 2 DG Method for Directly Solving the Hamilton-Jacobi Equations</b>	<b>8</b>
2.1 Scheme in One Dimension . . . . .	9
2.1.1 One-dimensional Formulation . . . . .	9
2.1.2 Interpretation of the Scheme . . . . .	11
2.2 Scheme on Two-dimensional Cartesian Meshes . . . . .	16
2.3 Scheme on General Unstructured Meshes . . . . .	20
2.4 Numerical Results . . . . .	21
2.4.1 One-dimensional Results . . . . .	22
2.4.2 Two-dimensional Results . . . . .	34
<b>Chapter 3 Sparse Grid DG Methods for High-Dimensional Elliptic Equations</b>	<b>46</b>
3.1 DG Finite Element Spaces on Sparse Grids . . . . .	48
3.1.1 Hierarchical Decomposition of Piecewise Polynomial Spaces in One Dimension . . . . .	48
3.1.2 Sparse Discontinuous Finite Element Spaces in Multi-dimensions . . . . .	54
3.1.3 Approximation Results . . . . .	61
3.1.4 Approximations from Various Sparse Discontinuous Finite Element Spaces: A Numerical Investigation . . . . .	63
3.2 IP Method on Sparse Grids for Elliptic Equations . . . . .	74
3.2.1 Formulation of the Scheme . . . . .	74
3.2.2 Error Analysis . . . . .	77
3.3 Numerical Results . . . . .	81
3.3.1 Two-dimensional Results . . . . .	81
3.3.2 Three-dimensional Results . . . . .	84
3.3.3 Four-dimensional Results . . . . .	88
<b>Chapter 4 Conclusion</b> . . . . .	<b>92</b>
<b>BIBLIOGRAPHY</b> . . . . .	<b>94</b>

## LIST OF TABLES

Table 2.1:	Errors and numerical orders of accuracy for Example 2.4.1 when using $P^k$ , $k = 1, 2, 3$ , polynomials and third order Runge-Kutta time discretization on a uniform mesh of $M$ cells. Final time $t = 1$ . . . . .	23
Table 2.2:	Errors and numerical orders of accuracy for Example 2.4.2 when using $P^2$ polynomials and third order Runge-Kutta time discretization on a uniform mesh of $M$ cells. Final time $t = 1$ . $CFL = 0.1$ . . . . .	26
Table 2.3:	Errors and numerical orders of accuracy for Example 2.4.3 when using $P^2$ polynomials and third order Runge-Kutta time discretization on a uniform mesh of $M$ cells. Penalty constant $C = 0.25$ . Final time $t = 0.5$ . $CFL = 0.1$ . . . . .	27
Table 2.4:	Errors and numerical orders of accuracy for Example 2.4.3 when using $P^1$ and $P^2$ polynomials and third order Runge-Kutta time discretization on a random mesh with 40% perturbation of $M$ cells. Penalty constant $C = 0.25$ . Final time $t = 0.5$ . $CFL = 0.1$ . . . . .	27
Table 2.5:	Errors and numerical orders of accuracy for Example 2.4.5 when using $P^2$ polynomials and third order Runge-Kutta time discretization on a uniform mesh of $M$ cells. Penalty constant $C = 0.25$ . Final time $t = 1$ . $CFL = 0.1$ . . . . .	30
Table 2.6:	Errors and numerical orders of accuracy for Example 2.4.6 when using $P^2$ polynomials and third order Runge-Kutta time discretization on a uniform mesh of $M$ cells. Penalty constants $C = 0.25$ . Final time $t = 0.5/\pi^2$ . $CFL = 0.1$ . . . . .	31
Table 2.7:	Errors and numerical orders of accuracy for Example 2.4.7 when using $P^2$ polynomials and third order Runge-Kutta time discretization on a uniform mesh of $M$ cells. $CFL = 0.05$ . Penalty constant $C = 0.25$ . Final time $t = 1$ . A minmod limiter is used. . . . .	32
Table 2.8:	Errors and numerical orders of accuracy for Example 2.4.8 when using $P^2$ polynomials and third order Runge-Kutta time discretization on a uniform mesh of $M \times M$ cells. Final time $t = 1$ . $CFL = 0.1$ . . . . .	35
Table 2.9:	Errors and numerical orders of accuracy for Example 3.9 when using $P^2$ polynomials and third order Runge-Kutta time discretization on a uniform mesh of $M \times M$ cells. Final time $t = 1$ . $CFL = 0.1$ . . . . .	36

Table 2.10:	Errors and numerical orders of accuracy for Example 2.4.10 when using $P^2$ polynomials and third order Runge-Kutta time discretization on triangular meshes with characteristic length $h$ . Penalty constant $C = 0.25$ . Final time $t = 0.5/\pi^2$ . $CFL = 0.1$ . . . . .	37
Table 2.11:	Errors and numerical orders of accuracy for Example 2.4.11 when using $P^2$ polynomials and third order Runge-Kutta time discretization on a uniform mesh of $M \times M$ cells. Penalty constant $C = 0.25$ . Final time $t = 0.8$ . $CFL = 0.1$ . . . . .	38
Table 2.12:	Errors and numerical orders of accuracy for Example 2.4.13 when using $P^2$ polynomials and third order Runge-Kutta time discretization on triangular meshes with characteristic length $h$ . Penalty constant $C = 0.25$ . Final time $t = 0.5/\pi^2$ . $CFL = 0.1$ . . . . .	40
Table 3.1:	Orthonormal, vanishing-moment functions $f_1, \dots, f_{k+1}$ , for $k = 0, \dots, 4$ , for $x \in (0, 1)$ . The functions $f_i$ is extended to the interval $(-1, 1)$ as an odd or even function based on the rule $f_i(x) = (-1)^{i+k} f_i(-x)$ for $x \in (-1, 0)$ , and is zero elsewhere. . . . .	51
Table 3.2:	$L^2$ projection errors and orders of accuracy. FGDOF denotes the degrees of freedom of the full grid approximation space. $d = 2$ . $k = 1$ .	68
Table 3.3:	Projection errors and orders of accuracy. FGDOF denotes the degrees of freedom of the full grid approximation space. $d = 2$ . $k = 2$ . . . . .	69
Table 3.4:	Projection errors and orders of accuracy. FGDOF denotes the degrees of freedom of the full grid approximation space. $d = 2$ . $k = 3$ .	70
Table 3.5:	Projection errors and orders of accuracy. FGDOF denotes the degrees of freedom of the full grid approximation space. $d = 3$ . $k = 1$ .	71
Table 3.6:	Projection errors and orders of accuracy. FGDOF denotes the degrees of freedom of the full grid approximation space. $d = 3$ . $k = 2$ .	72
Table 3.7:	Projection errors and orders of accuracy. FGDOF denotes the degrees of freedom of the full grid approximation space. $d = 3$ . $k = 3$ .	73
Table 3.8:	Numerical errors and orders of accuracy for Example 3.3.1 computed by space $\hat{\mathbf{V}}_N^k$ with $k = 1, 2$ and indicated $N$ . . . . .	82
Table 3.9:	Sparsity and condition number of the stiffness matrix. Example 3.3.1 computed by the space $\hat{\mathbf{V}}_N^k$ with $k = 1, 2$ . SGDOF is the degree of freedom used for the sparse grid DG scheme. NNZ is the number of nonzero elements in the stiffness matrix. Order= $\log(\text{NNZ})/\log(\text{SGDOF})$ .	83

Table 3.10:	Numerical errors and orders of accuracy for Example 3.3.2 computed by the space $\hat{\mathbf{V}}_N^k$ with $k = 1, 2$ and indicated $N$ . . . . .	84
Table 3.11:	Numerical errors and orders of accuracy for Example 3.3.3 computed by the space $\hat{\mathbf{V}}_N^k$ with $k = 1, 2$ and indicated $N$ . . . . .	85
Table 3.12:	Numerical errors and orders of accuracy for Example 3.3.4 computed by the space $\hat{\mathbf{V}}_N^k$ with $k = 1, 2$ and indicated $N$ . . . . .	86
Table 3.13:	Sparsity and condition number of the stiffness matrix. Example 3.3.4 computed by the space $\hat{\mathbf{V}}_N^k$ with $k = 1, 2$ . SGDOF is the degree of freedom used for the sparse grid DG scheme. NNZ is the number of nonzero elements in the stiffness matrix. Order= $\log(\text{NNZ})/\log(\text{SGDOF})$ . . . . .	86
Table 3.14:	Numerical errors and orders of accuracy for Example 3.3.5 computed by the space $\hat{\mathbf{V}}_N^k$ with $k = 1, 2$ and indicated $N$ . . . . .	88
Table 3.15:	Numerical errors and orders of accuracy for Example 3.3.6 computed by the space $\hat{\mathbf{V}}_N^k$ with $k = 1, 2$ and indicated $N$ . . . . .	89
Table 3.16:	Sparsity and condition number of the stiffness matrix. Example 3.3.6 computed by the space $\hat{\mathbf{V}}_N^k$ with $k = 1, 2$ . SGDOF is the degree of freedom used for the sparse grid DG scheme. NNZ is the number of nonzero elements in the stiffness matrix. Order= $\log(\text{NNZ})/\log(\text{SGDOF})$ . . . . .	90
Table 3.17:	Numerical errors and orders of accuracy for Example 3.3.7 computed by the space $\hat{\mathbf{V}}_N^k$ with $k = 1, 2$ and indicated $N$ . . . . .	91

## LIST OF FIGURES

Figure 2.1:	Example 2.4.2. The numerical solution with various values of penalty constant $C$ : (a) $C = 0$ , (b) $C = 0.001$ , (c) $C = 0.25$ , and (d) $C = 1$ . Here $t = 1$ , $CFL = 0.1$ , $P^2$ polynomials, $M = 80$ . Solid line: the exact solution; circles: the numerical solution. <i>For interpretation of the references to color in this and all other figures, the reader is referred to the electronic version of this thesis.</i> . . . . .	25
Figure 2.2:	Example 2.4.3. Here $t = 1.5$ , $CFL = 0.1$ , $P^2$ polynomials, $M = 40$ . Penalty constant $C = 0.25$ . Solid line: the exact solution; circles: the numerical solution. . . . .	28
Figure 2.3:	Example 2.4.4. The numerical solution with various values of penalty constant $C$ : (a) $C = 0$ , (b) $C = 0.001$ , (c) $C = 0.25$ , and (d) $C = 1.0$ . Here $t = 1$ , $CFL = 0.1$ , $P^2$ polynomials, $M = 80$ . Solid line: the exact solution; circles: the numerical solution. . . . .	29
Figure 2.4:	Example 2.4.6. Here $t = 1.5/\pi^2$ , $CFL = 0.1$ , $P^2$ polynomials, $M = 80$ . Penalty constant $C = 0.25$ . Solid line: the exact solution; circles: the numerical solution. . . . .	31
Figure 2.5:	Example 2.4.7. Comparison of the numerical solution with and without the limiter: (a) $M = 80$ , without limiter; (b) $M = 80$ , with limiter; (c) $M = 81$ , without limiter; and (d) $M = 81$ , with limiter. Here $t = 1$ , $P^2$ polynomials, $CFL = 0.05$ . Penalty constant $C = 0.25$ . Solid line: the exact solution; circles: the numerical solution. . . . .	33
Figure 2.6:	Example 2.4.8. Here $t = 2\pi$ , $CFL = 0.1$ , $80 \times 80$ uniform mesh. (a) $P^1$ polynomials; (b) $P^2$ polynomials. One dimensional cut of $45^\circ$ with the $x$ axis. Solid line: the exact solution; circles: the numerical solution. . . . .	35
Figure 2.7:	Examples 2.4.10 and 2.4.13. The unstructured mesh used with characteristic length $h = 1/4$ . . . . .	37
Figure 2.8:	Example 2.4.10. Here $CFL = 0.1$ , $P^2$ polynomials. Triangular mesh with characteristic length $1/8$ . 2816 elements. Penalty constant $C = 0.25$ . (a) $t = 0.5/\pi^2$ ; (b) $t = 1.5/\pi^2$ . . . . .	38
Figure 2.9:	Example 2.4.11. Here $CFL = 0.1$ , $P^2$ polynomials on a $80 \times 80$ uniform mesh. Penalty constant: $C = 0.25$ . (a) $t = 0.8$ ; (b) $t = 1.5$ . . . . .	39

Figure 2.10:	Example 2.4.12. Here $t = 1$ , $CFL = 0.1$ , $P^2$ polynomials on a $40 \times 40$ uniform mesh. Penalty constant: $C = 0.25$ . (a) the numerical solution; (b) $\text{sign}(\varphi_y)$ . . . . .	40
Figure 2.11:	Example 2.4.13. Here $CFL = 0.1$ , $P^2$ polynomials. Triangular mesh with characteristic length $1/8$ . 2816 elements. Penalty constant: $C = 0.25$ . (a) $t = 0.5/\pi^2$ . (b) $t = 1.5/\pi^2$ . . . . .	41
Figure 2.12:	Example 2.4.14. Here $t = 1$ , $CFL = 0.1$ , $P^2$ polynomials on a $41 \times 41$ uniform mesh. Penalty constant: $C = 0.25$ . . . . .	41
Figure 2.13:	Example 2.4.15. The unstructured mesh used in the computation. Number of elements: 3480. . . . .	42
Figure 2.14:	Example 2.4.15. $CFL = 0.1$ . $P^2$ polynomials. Penalty constant: $C = 0.25$ . The numerical solution at the indicated times. . . . .	43
Figure 2.15:	Example 2.4.16. The unstructured mesh used in the computation. Number of elements: 5890. . . . .	44
Figure 2.16:	Example 2.4.16. Here $CFL = 0.1$ , $P^2$ polynomials. Penalty constant: $C = 0.25$ . The numerical solution at the indicated times. . . . .	45
Figure 3.1:	Example 3.3.1. The sparsity patterns of the matrices computed by the space $\hat{\mathbf{V}}_6^k$ with $k = 1, 2$ and $N = 6$ . Each dot represents a non-zero element in the stiffness matrix. (a): $k = 1$ , (b): $k = 2$ . . . . .	83
Figure 3.2:	Example 3.3.4. The sparsity patterns of the matrices computed by the space $\hat{\mathbf{V}}_6^k$ with (a) $k = 1$ and (b) $k = 2$ . $N = 6$ . . . . .	87
Figure 3.3:	Example 3.3.6. (a) $k = 1$ ; (b) $k = 2$ . $N = 4$ . Plotted along $x_1 = 0.4930, x_2 = 0.4930$ . . . . .	89
Figure 3.4:	Example 3.3.6. The sparsity patterns of the matrices computed by the space $\hat{\mathbf{V}}_4^k$ with (a) $k = 1$ and (b) $k = 2$ . . . . .	90

# Chapter 1

## Introduction

### 1.1 Overview

Ever since the discontinuous Galerkin (DG) finite element method was first developed by Reed and Hill to solve the conservation laws, the use of DG discretizations has become more widespread. The appeal of DG methods relates to their flexibility to choose bases, combined with compact stencils and favorable properties for arbitrarily unstructured meshes. Moreover, DG methods generally are able to capture the physically relevant discontinuities accurately for problems with rough solutions. DG methods also have excellent parallel efficiency and can easily accommodate the  $hp$ -adaptivity. Therefore, DG methods have attracted interest of many researchers and practitioners and have been proved useful in the real-world problems like meteorology, weather-forecasting, semiconductor device simulation, electrodynamics and plasma physics.

In this thesis, we will focus on the improvement of the DG methods for the general Hamilton-Jacobi (HJ) equations and the development of sparse grid DG methods for high-dimensional elliptic problems to make the numerical simulations more accurate, efficient and stable.

## 1.2 Numerical Methods for HJ Equations

In this section we will introduce the HJ equation and review numerical methods for solving this equation.

We consider the numerical solution of a general time-dependent HJ equation

$$\varphi_t + H(\nabla_{\mathbf{x}}\varphi, \mathbf{x}) = 0, \quad \varphi(\mathbf{x}, 0) = \varphi^0(\mathbf{x}), \quad \mathbf{x} \in \Omega \in \mathbb{R}^d \quad (1.1)$$

with suitable boundary conditions on  $\partial\Omega$ . The HJ equation arises in many applications, e.g., optimal control, differential games, crystal growth, image processing and calculus of variations. The solution of such equation is Lipschitz continuous but may develop discontinuous derivatives in finite time even when the initial data is smooth.

The viscosity solution [32, 33] was introduced as the unique physically relevant solution, and has been the focus of many numerical methods. Starting from [34, 78], finite difference methods such as essentially non-oscillatory (ENO) [67, 68] or weighted ENO (WENO) methods [58, 87] have been developed to solve the HJ equation. Those finite difference methods work quite efficiently for Cartesian meshes, however they lose the advantage of simplicity on unstructured meshes [1, 87].

Alternatively, the Runge-Kutta discontinuous Galerkin (RKDG) method, originally devised to solve the conservation laws [31], is more flexible for arbitrarily unstructured meshes. The first work of DG methods for HJ equations [56, 61] relies on solving the conservation law system satisfied by the derivatives of the solution. The methods work well numerically even on unstructured mesh, with provable stability results for certain special cases, and were later generalized in e.g. [50, 24]. Unfortunately, the procedure of recovering  $\varphi$  from its derivatives has made the algorithm indirect and complicated. In contrast, the design of DG methods for

directly solving the HJ equations is appealing but challenging, because the HJ equation is not written in the conservative form, for which the framework of DG methods could easily apply. In [25], a DG method for directly solving the HJ equation was developed, motivated by the idea that the one-dimensional linear HJ equation can be rewritten as a conservation law with a source term. Later, this algorithm was applied to solve front propagation problems [15] and front propagation with obstacles [16], in which simplified implementations of the entropy fix procedure were proposed. Meanwhile, central DG [62] and local DG [84] methods were recently developed for the HJ equation. Numerical experiments demonstrate that both methods work for nonconvex Hamiltonians. In addition, the first order version of the local DG method [84] reduces to the monotone schemes and thus has provable convergence properties. However, the central DG methods based on overlapping meshes are difficult to implement on unstructured meshes, and the local DG methods still need to resort to the information about the derivatives of  $\varphi$ , making the method less direct in computation.  $L^2$  error estimates for smooth solutions of the DG method [25] and local DG method [84] have been established in [83]. For recent developments of high order and DG methods for HJ equations, one can refer to the review papers [74, 75].

The direct method in [25] works well numerically for both linear and convex cases, but still needs to resort to the complicated procedure in [56, 61] for the entropy fix. Based on the observation that the method in [25] is closely related to Roe's linearization, we propose to use interfacial terms involving the Roe speed and develop a new entropy fix that was inspired by the Harten and Hyman's entropy fix [53] for Roe scheme for the conservation laws. In Chapter 2, we develop such a direct solver. Moreover, we will show that the new method has the following advantages. Firstly, the scheme works on unstructured meshes even for nonconvex Hamiltonian. Secondly, the method is simple to implement. The cumbersome

$L^2$  reconstruction of the solutions' derivative at the cell interface in [25] is avoided, and the entropy fix is automatically incorporated by the added jump terms in the derivatives of the numerical solution. Finally, the scheme is direct and compact, and the computation only needs the information about the current cell and its immediate neighbors.

### 1.3 Review of DG Schemes for Elliptic Equations

Elliptic equations have been used successfully in many application areas such as aeroacoustics, electro-magnetism, oil recovery simulation, weather forecasting, etc. There has been a lot of effort on the numerical methods for the elliptic equations, including the standard finite difference methods, collocation methods, Galerkin finite element methods, least squares methods, among the others. DG methods work well for purely hyperbolic problems by nature, yet these methods also prove to be useful for elliptic equations. As pointed out in the review paper [7], many DG discretization methods have been developed such as the method of Bassi and Rebay [12], the variations in [17], and the local DG methods introduced in [23, 27, 28, 30].

In the 1970s, a class of interior penalty (IP) DG finite element methods was proposed in [6, 10, 37, 82]. These IPDG methods arose from Nitsche's idea that, just as the Dirichlet boundary conditions can be imposed weakly by adding a penalty term instead of being restricted strongly in the approximation space, the continuity across interelement boundaries could be attained similarly. Therefore, it is possible to use the more flexible discontinuous finite element space in this penalty technique. In 1973, Babuska [9] proposed another penalization technique to weakly impose  $C^1$  continuity. In 1978, Wheeler [82] generalized Nitsche's method to second-order elliptic problems and later Alnold generalized this technique and an-

alyzed in further detail for linear and nonlinear elliptic and parabolic problems in his thesis [6]. A family of IPDG methods, which includes Oden-Baumann-Babuska [66], symmetric interior penalty Galerkin [82], nonsymmetric interior penalty Galerkin [70], incomplete interior penalty Galerkin [35], has been proposed to deal with elliptic problems during the last few decades. However since 1980s, less attention has been paid to IP methods mainly because these penalty methods were never proved to be more efficient than conforming finite element approximation space. In spite of this, IP methods have the advantage of flexibility in choosing the approximation space and are more suited for  $hp$ -adaptivity. For instance, in 1990, Baker [11] used the nonconforming finite element approximations to enforce the divergence-free condition point-wise in each triangle mesh for the Stokes system. The interior penalties were used to deal with the discontinuity in the velocity across interelement boundaries.

## 1.4 Sparse Grid Methods for High-dimensional PDEs

In this thesis, we are concerned with the numerical solution of high-dimensional PDEs. High-dimensional PDEs have been a major challenge in the scientific computing areas due to the storage requirements and computational complexity. How to conquer the *the curse of dimensionality* [14] is vital to the real-world problems as such high-dimensional problems often come from kinetic simulations, stochastic analysis, control, optimization and mathematical modeling in finance or statistics. Examples include high-dimensional Laplace problems and high-dimensional convection-diffusion problems which result from diffusion approximation techniques or the Fokker–Planck equation. The challenge of those problems typically cannot be simply solved by increase of computational resources, but requires the improvement of numerical techniques, better computational implementation and usage of parallel comput-

ing. Those different perspectives of treatment have attracted increasing attention over the last decades. Among them, the sparse grid techniques, introduced by Zenger [86], have been developed as a major tool to break the curse of dimensionality of grid-based approaches. Such approaches rely on a tensor product hierarchical basis representation, and can successfully reduce the degrees of freedom from the standard  $O(h^{-d})$  to  $O(h^{-1}|\log h|^{d-1})$  for  $d$ -dimensional problems, where  $h$  is the uniform mesh size. The underlying ideas of sparse grid techniques can be traced back to Smolyak [77] for numerical integration, and the methods are closely related to hyperbolic cross [8, 79], boolean method [36], discrete blending method [13], and splitting extrapolation method [63] in the literature. The trick is to balance the cost complexities and the accuracy of the scheme and find a proper truncation of the tensor product hierarchical bases, which could be formally derived by solving an optimization problem of cost/benefit ratios, as discussed in [45]. For recent developments of sparse grids method, we refer to the survey paper [21] and lecture notes [38].

When solving high-dimensional PDEs, besides the natural choice of traditional Galerkin finite element methods [86, 21, 71], sparse grids have been incorporated in finite difference methods [44, 49], finite volume methods [54], and spectral methods [46, 42, 72, 73]. However, the potential of sparse grids has not yet been fully realized under the DG framework, in which discontinuous basis function spaces are under consideration. The flexibility and the mature development of DG scheme make it an excellent candidate to be combined with the sparse grid approach to solve high-dimensional PDEs. In Chapter 3, we will develop DG methods on sparse grids for efficient computations of high-dimensional elliptic equations. Our study consists of a thorough investigation of the underlying sparse approximation space, its utilization in conjunction with the IPDG method, and the properties of the schemes implied. Numerical results validate our theoretical findings, which give promising outlook for the

extension of the scheme to wider applications.

# Chapter 2

## DG Method for Directly Solving the Hamilton-Jacobi Equations

In this chapter, we improve upon the DG method for HJ equation with convex Hamiltonians in [25] and develop a new DG method for directly solving the general HJ equations. The new method avoids the reconstruction of the solution across elements by utilizing the Roe speed at the cell interface. Besides, we propose an entropy fix by adding penalty terms proportional to the jump of the normal derivative of the numerical solution. The particular form of the entropy fix was inspired by the Harten and Hyman's entropy fix [53] for Roe scheme for the conservation laws.

The rest of this chapter is organized as follows: in Section 2.1, we introduce the numerical scheme for one-dimensional HJ equations. We generalize the scheme to compute on two-dimensional Cartesian meshes in Section 2.2 and on general unstructured meshes in Section 2.2. Section 2.4 is devoted to the discussion of the numerical results. Benchmark numerical experiments in one dimension and two dimensions are provided to validate the performance of the method on both structured and unstructured meshes.

## 2.1 Scheme in One Dimension

In this section, we will describe the numerical methods for the one-dimensional equation. We follow the method of lines approach, and below we will only describe the semi-discrete DG schemes. The resulting method of lines ODEs can be solved by the total variation diminishing (TVD) Runge-Kutta methods [76] or strong stability preserving (SSP) Runge-Kutta methods [41].

### 2.1.1 One-dimensional Formulation

In this subsection, we will start with the simple one-dimensional HJ equation. In this case, (1.1) becomes

$$\varphi_t + H(\varphi_x, x) = 0, \quad \varphi(x, 0) = \varphi^0(x). \quad (2.1)$$

Assume the computational domain is  $[a, b]$ , we will divide it into  $M$  cells as follows

$$a = x_{\frac{1}{2}} < x_{\frac{3}{2}} < \dots < x_{M+\frac{1}{2}} = b. \quad (2.2)$$

Now the cells and their centers are defined as

$$I_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}], \quad x_j = \frac{1}{2} \left( x_{j-\frac{1}{2}} + x_{j+\frac{1}{2}} \right), \quad j = 1, \dots, M \quad (2.3)$$

and the mesh sizes are

$$\Delta x_j = x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}}, \quad h = \max_j \Delta x_j. \quad (2.4)$$

The DG approximation space is

$$V_h^k = \{v : v|_{I_j} \in P^k(I_j), j = 1, \dots, M\} \quad (2.5)$$

where  $P^k(I_j)$  denotes all polynomials of degree at most  $k$  on  $I_j$ , and we let  $H_1 = \frac{\partial H}{\partial \varphi_x}$  be the partial derivative of the Hamiltonian with respect to  $\varphi_x$ .

To introduce the scheme, we need to define several quantities at the cell interface where the DG solution is discontinuous. If  $x_*$  is a point located at the cell interface, then  $\varphi_h \in V_h^k$  would be discontinuous at  $x_*$ . We can then define the Roe speed at  $x_*$  to be

$$\tilde{H}_{\varphi_h}(x_*) := \begin{cases} \frac{H((\varphi_h)_x(x_*^+), x_*^+) - H((\varphi_h)_x(x_*^-), x_*^-)}{(\varphi_h)_x(x_*^+) - (\varphi_h)_x(x_*^-)}, & \text{if } (\varphi_h)_x(x_*^+) \neq (\varphi_h)_x(x_*^-), \\ \frac{1}{2} (H_1((\varphi_h)_x(x_*^+), x_*^+) + H_1((\varphi_h)_x(x_*^-), x_*^-)), & \text{if } (\varphi_h)_x(x_*^+) = (\varphi_h)_x(x_*^-). \end{cases}$$

In the notations above, we use superscripts  $+$ ,  $-$  to denote the right, and left limits of a function. Notice that in order for the above definition to make sense, we restrict our attention to  $k \geq 1$  case, i.e. piecewise linear polynomials and above.

Similar to Harten and Hyman's entropy fix [53], to detect the entropy violating cells, we introduce

$$\delta_{\varphi_h}(x_*) := \max \left( 0, \tilde{H}_{\varphi_h}(x_*) - H_1((\varphi_h)_x(x_*^-), x_*^-), H_1((\varphi_h)_x(x_*^+), x_*^+) - \tilde{H}_{\varphi_h}(x_*) \right)$$

and

$$S_{\varphi_h}(x_*) := \max \left( \delta_{\varphi_h}(x_*), |\tilde{H}_{\varphi_h}(x_*)| \right).$$

We note that  $S_{\varphi_h}(x_*) \neq |\tilde{H}_{\varphi_h}(x_*)|$  only if  $|\tilde{H}_{\varphi_h}(x_*)| < \delta_{\varphi_h}(x_*)$ .

Now we are ready to formulate our DG scheme for (2.1). We look for  $\varphi_h(x, t) \in V_h^k$ , such

that

$$\begin{aligned}
& \int_{I_j} (\partial_t \varphi_h(x, t) + H(\partial_x \varphi_h(x, t), x)) v_h(x) dx \\
& + \min \left( \tilde{H}_{\varphi_h}(x_{j+\frac{1}{2}}), 0 \right) [\varphi_h]_{j+\frac{1}{2}} (v_h)_{j+\frac{1}{2}}^- + \max \left( \tilde{H}_{\varphi_h}(x_{j-\frac{1}{2}}), 0 \right) [\varphi_h]_{j-\frac{1}{2}} (v_h)_{j-\frac{1}{2}}^+ \\
& - C \Delta x_j \left( S_{\varphi_h}(x_{j+\frac{1}{2}}) - |\tilde{H}_{\varphi_h}(x_{j+\frac{1}{2}})| \right) [(\varphi_h)_x]_{j+\frac{1}{2}} (v_h)_{j+\frac{1}{2}}^- \\
& - C \Delta x_j \left( S_{\varphi_h}(x_{j-\frac{1}{2}}) - |\tilde{H}_{\varphi_h}(x_{j-\frac{1}{2}})| \right) [(\varphi_h)_x]_{j-\frac{1}{2}} (v_h)_{j-\frac{1}{2}}^+ \\
& = 0, \quad \forall j = 1, \dots, M
\end{aligned} \tag{2.6}$$

holds for any  $v_h \in V_h^k$  with  $k \geq 1$ . Here  $[u] = u^+ - u^-$  denotes the jump of a function at the cell interface,  $\Delta x_j$  serves as the scaling factor.  $C$  is a positive penalty parameter. The detailed discussion about the choice of  $C$  is contained in Section 2.4. In particular, we find that  $C = 0.25$  works well in practice.

### 2.1.2 Interpretation of the Scheme

In this subsection, we provide interpretation of the scheme (2.6) for the linear HJ equation with variable coefficient

$$\varphi_t + a(x)\varphi_x = 0$$

to illustrate the main ideas.

To better interpret our scheme and compare it with the old method, we first review the direct DG scheme in [25]. As we mentioned, Cheng and Shu in [25] used the idea that

$$\varphi_t + a(x)\varphi_x = 0 \tag{2.7}$$

with  $a'(x) \geq 0$ , is equivalent to a conservation law with a source term:

$$\varphi_t + (a(x)\varphi)_x = a'(x)\varphi. \quad (2.8)$$

The authors assumed  $a(x)$  is smooth and applied DG method to (2.8) with Roe flux which requires an upwinding flux to obtain:

$$\begin{aligned} & \int_{I_j} (\partial_t \varphi_h(x, t) + a(x) \partial_x \varphi_h(x, t)) v_h(x) dx \\ & + \frac{1}{2} \left( a(x_{j+\frac{1}{2}}) - |a(x_{j+\frac{1}{2}})| \right) [\varphi_h]_{j+\frac{1}{2}} (v)_{j+\frac{1}{2}}^+ + \frac{1}{2} \left( a(x_{j-\frac{1}{2}}) + |a(x_{j-\frac{1}{2}})| \right) [\varphi_h]_{j-\frac{1}{2}} (v)_{j-\frac{1}{2}}^+ \\ & = 0, \quad j = 1, \dots, M. \end{aligned} \quad (2.9)$$

This motivated them to propose the following are scheme for general HJ equations:

find  $\varphi_h \in V_h^k$ , such that

$$\begin{aligned} & \int_{I_j} (\partial_t \varphi_h(x, t) + H(\partial_x \varphi_h(x, t), x)) v_h(x) dx \\ & + \frac{1}{2} \left( \min_{x \in I_{j+\frac{1}{2}}} H_1(\partial_x \varphi_h, x_{j+\frac{1}{2}}) - \left| \min_{x \in I_{j+\frac{1}{2}}} H_1(\partial_x \varphi_h, x_{j+\frac{1}{2}}) \right| \right) [\varphi_h]_{j+\frac{1}{2}} (v_h)_{j+\frac{1}{2}}^- \\ & + \frac{1}{2} \left( \max_{x \in I_{j-\frac{1}{2}}} H_1(\partial_x \varphi_h, x_{j-\frac{1}{2}}) + \left| \max_{x \in I_{j-\frac{1}{2}}} H_1(\partial_x \varphi_h, x_{j-\frac{1}{2}}) \right| \right) [\varphi_h]_{j-\frac{1}{2}} (v_h)_{j-\frac{1}{2}}^- \\ & = 0, \quad j = 1, \dots, M \end{aligned} \quad (2.10)$$

The reconstructed information of  $\partial_x \varphi_h$  on the cells  $I_{j+\frac{1}{2}}$  and  $I_{j-\frac{1}{2}}$  was needed. Therefore,  $L^2$  type projection was used to reconstruct a continuous approximation of  $\varphi_h(x, t)$  in their

scheme. They defined a polynomial  $w_{j+\frac{1}{2}} \in P^{2k+1}$  on  $I_j \cup I_{j+1}$ , s.t.

$$\int_{I_j} \varphi_h v dx = \int_{I_j} w_{j+\frac{1}{2}} v dx$$

and

$$\int_{I_{j+1}} \varphi_h v dx = \int_{I_{j+1}} w_{j+\frac{1}{2}} v dx$$

then  $\partial_x \varphi_h = \partial_x w_{j+\frac{1}{2}}$  on  $I_{j+\frac{1}{2}}$ .

Their scheme has provable stability and error estimates for linear equations and demonstrates good convergence to the viscosity solutions for nonlinear equations. However, this scheme only works for equations with convex Hamiltonians. Moreover, in entropy violating cells, where it satisfies  $H_1(\varphi_x^-(x_{j-\frac{1}{2}})) < 0 < H_1(\varphi_x^+(x_{j-\frac{1}{2}}))$  or  $H_1(\varphi_x^-(x_{j+\frac{1}{2}})) < 0 < H_1(\varphi_x^+(x_{j+\frac{1}{2}}))$ , a correction based on the schemes in [56, 61] is necessary to guarantee stability of the method.

Now we provide an interpretation of the proposed scheme (2.6). Our scheme can be thought as an improvement upon the above direct DG method (2.10). Firstly, if  $a(x) \in C^1$ , we have

$$\tilde{H}_{\varphi_h}(x_{j+\frac{1}{2}}) = a(x_{j+\frac{1}{2}}), \quad \delta_{\varphi_h}(x_{j+\frac{1}{2}}) = 0, \quad S_{\varphi_h}(x_{j+\frac{1}{2}}) = |a(x_{j+\frac{1}{2}})|, \quad \forall j = 1, \dots, M,$$

therefore the scheme (2.6) reduces to

$$\begin{aligned}
& \int_{I_j} (\partial_t \varphi_h(x, t) + a(x) \partial_x \varphi_h(x, t)) v_h(x) dx \\
& + \min \left( a(x_{j+\frac{1}{2}}), 0 \right) [\varphi_h]_{j+\frac{1}{2}} (v_h)_{j+\frac{1}{2}}^- + \max \left( a(x_{j-\frac{1}{2}}), 0 \right) [\varphi_h]_{j-\frac{1}{2}} (v_h)_{j-\frac{1}{2}}^+, \\
& = 0, \quad \forall j = 1, \dots, M.
\end{aligned} \tag{2.11}$$

This is exactly the same as the direct DG method (2.9). Therefore, stability and error estimates of our scheme could be established the same as [25].

In addition, when  $a(x)$  is no longer smooth, especially when  $a(x)$  contains discontinuity at cell interfaces. The old scheme (2.9) will produce entropy violating shocks in the solutions' derivative [25]. In this case, the penalty terms in the proposed scheme (2.6) come into play, and the added viscosity enables us to capture the viscosity solution as demonstrated in Section 2.4. In particular, suppose  $a(x)$  is discontinuous at  $x_{j+\frac{1}{2}}$ , then

$$\tilde{H}_{\varphi_h}(x_{j+\frac{1}{2}}) = \frac{[a(x)\varphi_h]_{j+\frac{1}{2}}}{[\varphi_h]_{j+\frac{1}{2}}}.$$

If the entropy condition is violated at  $x_{j+\frac{1}{2}}$ , i.e.,  $a(x_{j+\frac{1}{2}}^-) < 0 < a(x_{j+\frac{1}{2}}^+)$ , then

$$\delta_{\varphi_h}(x_{j+\frac{1}{2}}) = \max \left( 0, \frac{[a(x)\varphi_h]_{j+\frac{1}{2}}}{[\varphi_h]_{j+\frac{1}{2}}} - a(x_{j+\frac{1}{2}}^-), a(x_{j+\frac{1}{2}}^+) - \frac{[a(x)\varphi_h]_{j+\frac{1}{2}}}{[\varphi_h]_{j+\frac{1}{2}}} \right) > 0,$$

and  $S_{\varphi_h}(x_{j+\frac{1}{2}}) > 0$ . When  $|\tilde{H}_{\varphi_h}(x_{j+\frac{1}{2}})| < \delta_{\varphi_h}(x_{j+\frac{1}{2}})$ , the penalty term

$$-C\Delta x_j \left( S_{\varphi_h}(x_{j+\frac{1}{2}}) - |\tilde{H}_{\varphi_h}(x_{j+\frac{1}{2}})| \right) [(\varphi_h)_x]_{j+\frac{1}{2}} (v_h)_{j+\frac{1}{2}}^-$$

will be nonzero. On the other hand, if  $a(x_{j+\frac{1}{2}}^-) > 0 > a(x_{j+\frac{1}{2}}^+)$ , corresponding to a shock in  $\varphi_x$ , we know the Roe scheme (2.11) could correctly capture this behavior. In fact, in this case

$$\delta_{\varphi_h}(x_{j+\frac{1}{2}}) = \max \left( 0, \frac{[a(x)\varphi_h]_{j+\frac{1}{2}}}{[\varphi_h]_{j+\frac{1}{2}}} - a(x_{j+\frac{1}{2}}^-), a(x_{j+\frac{1}{2}}^+) - \frac{[a(x)\varphi_h]_{j+\frac{1}{2}}}{[\varphi_h]_{j+\frac{1}{2}}} \right) = 0,$$

and  $S_{\varphi_h}(x_{j+\frac{1}{2}}) - |\tilde{H}_{\varphi_h}(x_{j+\frac{1}{2}})| = 0$ , the method (2.6) will reduce to (2.11). Similar arguments extend to the nonlinear case for sonic expanding cells for convex Hamiltonians,

$$H_1(\varphi_x^-(x_{j+\frac{1}{2}})) < 0 < H_1(\varphi_x^+(x_{j+\frac{1}{2}})).$$

The penalty term in (2.6) would be turned on automatically.

Finally we remark that the key differences of the scheme (2.6) compared to the method in [25] are: (1) the  $L^2$  reconstruction of  $\varphi_h$  across two elements is avoided and we use the Roe speed which could be easily computed, which is advantageous especially for multi-dimensional problems on unstructured meshes, as to be illustrated in Sections 2.2 and 2.3, (2) the added penalty terms automatically treat the sonic points, and the key idea is to add the viscosity based on the jump in  $(\varphi_h)_x$ , but not  $\varphi_h$  itself. This is natural considering the formation of monotone schemes such as the Lax-Friedrichs scheme for HJ equation. We will verify in Section 2.4 that the penalty terms do not degrade the order of the accuracy of the numerical scheme.

## 2.2 Scheme on Two-dimensional Cartesian Meshes

In this subsection, we generalize the scheme to compute HJ equations on two-dimensional Cartesian meshes. Now equation (1.1) is written as

$$\varphi_t + H(\varphi_x, \varphi_y, x, y) = 0, \quad \varphi(x, y, 0) = \varphi^0(x, y), \quad (x, y) \in [a, b] \times [c, d]. \quad (2.12)$$

The rectangular mesh is defined by

$$a = x_{\frac{1}{2}} < x_{\frac{3}{2}} < \dots < x_{N_x + \frac{1}{2}} = b, \quad c = y_{\frac{1}{2}} < y_{\frac{3}{2}} < \dots < y_{N_y + \frac{1}{2}} = d \quad (2.13)$$

and

$$I_{i,j} = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}], \quad J_i = [x_{i-1/2}, x_{i+1/2}], \quad K_j = [y_{j-1/2}, y_{j+1/2}] \\ i = 1, \dots, N_x, \quad j = 1, \dots, N_y. \quad (2.14)$$

Let

$$\Delta x_i = x_{i+1/2} - x_{i-1/2}, \quad \Delta y_j = y_{j+1/2} - y_{j-1/2}, \quad h = \max(\max_i \Delta x_i, \max_j \Delta y_j).$$

We define the approximation space as

$$V_h^k = \{v : v|_{I_{i,j}} \in P^k(I_{i,j}), \quad i = 1, \dots, N_x, \quad j = 1, \dots, N_y\} \quad (2.15)$$

where  $P^k(I_{i,j})$  denotes all polynomials of degree at most  $k$  on  $I_{i,j}$  with  $k \geq 1$ .

Let us denote  $H_1 = \frac{\partial H}{\partial \varphi_x}$  and  $H_2 = \frac{\partial H}{\partial \varphi_y}$ . Similar to the one-dimensional case, we need

to introduce several numerical quantities at the cell interface.

If  $x_*$  is located at the cell interface in the  $x$  direction, then  $\varphi_h \in V_h^k$  is discontinuous at  $(x_*, y)$  for any  $y$ , and we define the Roe speed in the  $x$  direction at  $(x_*, y)$  to be

$$\tilde{H}_{1,\varphi_h}(x_*, y) := \begin{cases} \frac{H((\varphi_h)_x(x_*^+, y), \overline{(\varphi_h)_y}, x_*^+, y) - H((\varphi_h)_x(x_*^-, y), \overline{(\varphi_h)_y}, x_*^-, y)}{(\varphi_h)_x(x_*^+, y) - (\varphi_h)_x(x_*^-, y)}, & \text{if } (\varphi_h)_x(x_*^+, y) \neq (\varphi_h)_x(x_*^-, y), \\ \frac{1}{2} \left( H_1((\varphi_h)_x(x_*^+, y), \overline{(\varphi_h)_y}, x_*^+, y) + H_1((\varphi_h)_x(x_*^-, y), \overline{(\varphi_h)_y}, x_*^-, y) \right), & \text{if } (\varphi_h)_x(x_*^+, y) = (\varphi_h)_x(x_*^-, y), \end{cases}$$

where

$$\overline{(\varphi_h)_y} = \frac{1}{2} \left( (\varphi_h)_y(x_*^+, y) + (\varphi_h)_y(x_*^-, y) \right)$$

is the average of the tangential derivative. Again, we define

$$\delta_{1,\varphi_h}(x_*, y) := \max \left( 0, \tilde{H}_{1,\varphi_h}(x_*, y) - H_1((\varphi_h)_x(x_*^-, y), \overline{(\varphi_h)_y}, x_*^-, y), \right. \\ \left. H_1((\varphi_h)_x(x_*^+, y), \overline{(\varphi_h)_y}, x_*^+, y) - \tilde{H}_{1,\varphi_h}(x_*, y) \right)$$

and

$$S_{1,\varphi_h}(x_*, y) := \max \left( \delta_{1,\varphi_h}(x_*, y), |\tilde{H}_{1,\varphi_h}(x_*, y)| \right).$$

Similarly, for  $y_*$  located at the cell interface in the  $y$  direction,  $\varphi_h \in V_h^k$  is discontinuous at

$(x, y_*)$  for any  $x$ , and we define the Roe speed in the  $y$  direction at  $(x, y_*)$  to be

$$\tilde{H}_{2,\varphi_h}(x, y_*) := \begin{cases} \frac{H(\overline{(\varphi_h)_x}, (\varphi_h)_y(x, y_*^+), x, y_*^+) - H(\overline{(\varphi_h)_x}, (\varphi_h)_y(x, y_*^-), x, y_*^-)}{(\varphi_h)_y(x, y_*^+) - (\varphi_h)_y(x, y_*^-)}, & \text{if } (\varphi_h)_y(x, y_*^+) \neq (\varphi_h)_y(x, y_*^-), \\ \frac{1}{2} \left( H_1(\overline{(\varphi_h)_x}, (\varphi_h)_y(x, y_*^+), x, y_*^+) + H_1(\overline{(\varphi_h)_x}, (\varphi_h)_y(x, y_*^-), x, y_*^-) \right), & \text{if } (\varphi_h)_y(x, y_*^+) = (\varphi_h)_y(x, y_*^-), \end{cases}$$

where

$$\overline{(\varphi_h)_x} = \frac{1}{2} \left( (\varphi_h)_x(x, y_*^+) + (\varphi_h)_x(x, y_*^-) \right)$$

is the average of the tangential derivative. Again, we define

$$\delta_{2,\varphi_h}(x, y_*) := \max \left( 0, \tilde{H}_{2,\varphi_h}(x, y_*) - H_2(\overline{(\varphi_h)_x}, (\varphi_h)_y(x, y_*^-), x, y_*^-), \right. \\ \left. H_2(\overline{(\varphi_h)_x}, (\varphi_h)_y(x, y_*^+), x, y_*^+) - \tilde{H}_{2,\varphi_h}(x, y_*) \right)$$

and

$$S_{2,\varphi_h}(x, y_*) := \max \left( \delta_{2,\varphi_h}(x, y_*), |\tilde{H}_{2,\varphi_h}(x, y_*)| \right).$$

We now formulate our scheme as: find  $\varphi_h(x, t) \in V_h^k$ , such that

$$\begin{aligned}
& \int_{I_{i,j}} (\partial_t \varphi_h(x, y, t) + H(\partial_x \varphi_h(x, y, t), \partial_y \varphi_h(x, y, t), x, y)) v_h(x, y) dx dy \\
& + \int_{K_j} \min \left( \tilde{H}_{1,\varphi_h}(x_{i+\frac{1}{2}}, y), 0 \right) [\varphi_h](x_{i+\frac{1}{2}}, y) v_h(x_{i+\frac{1}{2}}^-, y) dy \\
& + \int_{K_j} \max \left( \tilde{H}_{1,\varphi_h}(x_{i-\frac{1}{2}}, y), 0 \right) [\varphi_h](x_{i-\frac{1}{2}}, y) v_h(x_{i-\frac{1}{2}}^+, y) dy \\
& + \int_{J_i} \min \left( \tilde{H}_{2,\varphi_h}(x, y_{j+\frac{1}{2}}), 0 \right) [\varphi_h](x, y_{j+\frac{1}{2}}) v_h(x, y_{j+\frac{1}{2}}^-) dx \\
& + \int_{J_i} \max \left( \tilde{H}_{2,\varphi_h}(x, y_{j-\frac{1}{2}}), 0 \right) [\varphi_h](x, y_{j-\frac{1}{2}}) v_h(x, y_{j-\frac{1}{2}}^+) dx \tag{2.16} \\
& - C \Delta x_i \int_{K_j} \left( S_{1,\varphi_h}(x_{i+\frac{1}{2}}, y) - |\tilde{H}_{1,\varphi_h}(x_{i+\frac{1}{2}}, y)| \right) [(\varphi_h)_x](x_{i+\frac{1}{2}}, y) v_h(x_{i+\frac{1}{2}}^-, y) dy \\
& - C \Delta x_i \int_{K_j} \left( S_{1,\varphi_h}(x_{i-\frac{1}{2}}, y) - |\tilde{H}_{1,\varphi_h}(x_{i-\frac{1}{2}}, y)| \right) [(\varphi_h)_x](x_{i-\frac{1}{2}}, y) v_h(x_{i-\frac{1}{2}}^+, y) dy \\
& - C \Delta y_j \int_{J_i} \left( S_{2,\varphi_h}(x, y_{j+\frac{1}{2}}) - |\tilde{H}_{2,\varphi_h}(x, y_{j+\frac{1}{2}})| \right) [(\varphi_h)_y](x, y_{j+\frac{1}{2}}) v_h(x, y_{j+\frac{1}{2}}^-) dx \\
& - C \Delta y_j \int_{J_i} \left( S_{2,\varphi_h}(x, y_{j-\frac{1}{2}}) - |\tilde{H}_{2,\varphi_h}(x, y_{j-\frac{1}{2}})| \right) [(\varphi_h)_y](x, y_{j-\frac{1}{2}}) v_h(x, y_{j-\frac{1}{2}}^+) dx = 0
\end{aligned}$$

holds for any  $v_h \in V_h^k$  with  $k \geq 1$ . In practice, the volume and line integrals in (2.16) can be evaluated by Gauss quadrature formulas. The main idea in (2.16) is that in the normal direction of the interface, we apply the ideas from the one-dimensional case, but tangential to the interface, we use the average of the tangential derivatives from the two neighboring cells.

## 2.3 Scheme on General Unstructured Meshes

In this subsection, we describe the scheme on general unstructured meshes for (1.1). Let  $\mathcal{T}_h = \{K\}$  be a partition of  $\Omega$ , with  $K$  being simplices. We define the piecewise polynomial space

$$V_h^k = \left\{ v \in L^2(\Omega) : v|_K \in P^k(K), \forall K \in \mathcal{T}_h \right\},$$

where  $P^k(K)$  denotes the set of polynomials of total degree at most  $k$  on  $K$  with  $k \geq 1$ . For any element  $K$ , and edge in  $\partial K$ , we define  $\mathbf{n}_K$  to be the outward unit normal to the boundary of  $K$ , and  $\mathbf{t}_K$  is the unit tangential vector such that  $\mathbf{n}_K \cdot \mathbf{t}_K = 0$ . In higher dimensions, i.e.  $d > 2$ ,  $d - 1$  tangential vectors need to be defined. In addition, for any function  $u \in V_h^k$ , and  $\mathbf{x} \in \partial K$ , we define the traces of  $u_h$  from outside and inside of the element  $K$  to be

$$u_h^\pm(\mathbf{x}) = \lim_{\epsilon \downarrow 0} u_h(\mathbf{x} \pm \epsilon \mathbf{n}_K),$$

and  $[u_h](\mathbf{x}) = u_h^+(\mathbf{x}) - u_h^-(\mathbf{x})$ ,  $\overline{u_h}(\mathbf{x}) = \frac{1}{2}(u_h^+(\mathbf{x}) + u_h^-(\mathbf{x}))$ . We also let  $H_{\mathbf{n}_K} = \nabla_{\nabla\varphi} H \cdot \mathbf{n}_K$ .

Now following the Cartesian case, we define, for any  $\mathbf{x} \in \partial K$ ,

$$\tilde{H}_{\mathbf{n}_K, \varphi_h}(\mathbf{x}) := \begin{cases} \frac{H((\nabla_{\mathbf{x}}\varphi_h \cdot \mathbf{n}_K)^+, \overline{\nabla_{\mathbf{x}}\varphi_h \cdot \mathbf{t}_K}, \mathbf{x}^+) - H((\nabla_{\mathbf{x}}\varphi_h \cdot \mathbf{n}_K)^-, \overline{\nabla_{\mathbf{x}}\varphi_h \cdot \mathbf{t}_K}, \mathbf{x}^-)}{[\nabla_{\mathbf{x}}\varphi_h \cdot \mathbf{n}_K](\mathbf{x})}, & \text{if } [\nabla_{\mathbf{x}}\varphi_h \cdot \mathbf{n}_K](\mathbf{x}) \neq 0, \\ \frac{1}{2} \left( H_{\mathbf{n}_K}((\nabla_{\mathbf{x}}\varphi_h \cdot \mathbf{n}_K)^+, \overline{\nabla_{\mathbf{x}}\varphi_h \cdot \mathbf{t}_K}, \mathbf{x}^+) + H_{\mathbf{n}_K}((\nabla_{\mathbf{x}}\varphi_h \cdot \mathbf{n}_K)^-, \overline{\nabla_{\mathbf{x}}\varphi_h \cdot \mathbf{t}_K}, \mathbf{x}^-) \right), & \text{otherwise,} \end{cases}$$

$$\begin{aligned} \delta_{\mathbf{n}_K, \varphi_h}(\mathbf{x}) &:= \max \left( 0, \tilde{H}_{\mathbf{n}_K, \varphi_h}(\mathbf{x}) - H_{\mathbf{n}_K}((\nabla_{\mathbf{x}} \varphi_h \cdot \mathbf{n}_K)^-, \overline{\nabla_{\mathbf{x}} \varphi_h \cdot \mathbf{t}_K}, \mathbf{x}^-), \right. \\ &\quad \left. H_{\mathbf{n}_K}((\nabla_{\mathbf{x}} \varphi_h \cdot \mathbf{n}_K)^+, \overline{\nabla_{\mathbf{x}} \varphi_h \cdot \mathbf{t}_K}, \mathbf{x}^+) - \tilde{H}_{\mathbf{n}_K, \varphi_h}(\mathbf{x}) \right), \end{aligned}$$

and

$$S_{\mathbf{n}_K, \varphi_h}(\mathbf{x}) := \max \left( \delta_{\mathbf{n}_K, \varphi_h}(\mathbf{x}), |\tilde{H}_{\mathbf{n}_K, \varphi_h}(\mathbf{x})| \right).$$

Then we look for  $\varphi_h \in V_h^k$ , such that for each  $K$ ,

$$\begin{aligned} &\int_K ((\varphi_h)_t + H(\nabla_{\mathbf{x}} \varphi_h, \mathbf{x})) v_h \, d\mathbf{x} + \int_{\partial K} \min \left( \tilde{H}_{\mathbf{n}_K, \varphi_h}(\mathbf{x}), 0 \right) [\varphi_h](\mathbf{x}) v_h^-(\mathbf{x}) \, ds \\ &- C \frac{\Delta K}{\Delta S_K} \int_{\partial K} \left( S_{\mathbf{n}_K, \varphi_h}(\mathbf{x}) - |\tilde{H}_{\mathbf{n}_K, \varphi_h}(\mathbf{x})| \right) [\nabla_{\mathbf{x}} \varphi_h \cdot \mathbf{n}_K](\mathbf{x}) v_h^-(\mathbf{x}) \, ds = 0 \end{aligned}$$

for any test function  $v_h \in V_h^k$  with  $k \geq 1$ , where  $\Delta K$ ,  $\Delta S_K$  are size of the element  $K$  and edge  $S_K$  respectively. In practice, the volume and edge integrals need to be evaluated by quadrature rules with enough accuracy. For example, we use quadrature rules that are exact for  $(2k)$ -th order polynomial for the volume integral, and quadrature rules that are exact for  $(2k + 1)$ -th order polynomial for the edge integrals.

## 2.4 Numerical Results

In this section, we provide numerical results to demonstrate the high order accuracy and reliability of our schemes. In all numerical experiments, we use the third order TVD-RK method as the temporal discretization [76].

### 2.4.1 One-dimensional Results

In this subsection, we provide computational results for one-dimensional HJ equations.

**Example 2.4.1** We solve the following linear problem with smooth variable coefficient

$$\begin{cases} \varphi_t + \sin(x)\varphi_x = 0, & x \in [0, 2\pi] \\ \varphi(x, 0) = \sin(x), \\ \varphi(0, t) = \varphi(2\pi, t). \end{cases} \quad (2.17)$$

Since  $a(x)$  is smooth in this example, the penalty term automatically vanishes and the choice of  $C$  does not have an effect on the solution. We provide the numerical results for  $P^1, P^2$  and  $P^3$  polynomials in Table 2.1. The CFL numbers used are also listed in this table. For  $P^3$  polynomials, we set  $\Delta t = O(\Delta x^{\frac{4}{3}})$  in order to get comparable numerical errors in time as in space. From the results, we could clearly observe the optimal  $(k+1)$ -th order accuracy for  $P^k$  polynomials.

**Example 2.4.2** We solve the following linear problem with nonsmooth variable coefficient

$$\begin{cases} \varphi_t + \text{sign}(\cos(x))\varphi_x = 0, & x \in [0, 2\pi] \\ \varphi(x, 0) = \sin(x), \\ \varphi(0, t) = \varphi(2\pi, t). \end{cases} \quad (2.18)$$

The viscosity solution of this example has a shock forming in  $\varphi_x$  at  $x = \pi/2$ , and a rarefaction wave at  $x = 3\pi/2$ . We use this example to demonstrate the effect of the choice of  $C$  on the numerical solution. The solutions obtained with different values of  $C$  are provided

Table 2.1: Errors and numerical orders of accuracy for Example 2.4.1 when using  $P^k$ ,  $k = 1, 2, 3$ , polynomials and third order Runge-Kutta time discretization on a uniform mesh of  $M$  cells. Final time  $t = 1$ .

$P^1$ $CFL = 0.3$						
M	$L^1$ error	order	$L^2$ error	order	$L^\infty$ error	order
40	1.20E-03		2.55E-03		1.52E-02	
80	3.07E-04	1.96	6.83E-04	1.90	4.32E-03	1.81
160	7.84E-05	1.97	1.78E-04	1.94	1.14E-03	1.92
320	1.99E-05	1.98	4.56E-05	1.97	2.94E-04	1.96
640	5.03E-06	1.99	1.15E-05	1.98	7.43E-05	1.98
$P^2$ $CFL = 0.1$						
40	4.76E-05		9.97E-05		5.23E-04	
80	5.97E-06	2.99	1.36E-05	2.88	8.77E-05	2.58
160	7.48E-07	3.00	1.82E-06	2.90	1.35E-05	2.70
320	9.38E-08	2.99	2.38E-07	2.93	1.96E-06	2.78
640	1.18E-08	2.99	3.08E-08	2.95	2.72E-07	2.85
$P^3$ $CFL = 0.05$						
40	2.12E-06		5.13E-06		2.89E-05	
80	1.36E-07	3.97	3.49E-07	3.89	2.16E-06	3.75
160	8.71E-09	3.97	2.30E-08	3.93	1.57E-07	3.79
320	5.14E-10	4.09	1.35E-09	4.10	9.47E-09	4.06
640	4.83E-12	6.75	9.06E-12	7.24	4.52E-11	7.73
1280	2.03E-13	4.58	2.96E-13	4.94	1.42E-12	5.00

in Figure 2.1. If we take the penalty constant  $C = 0$ , that is, without entropy correction, the entropy condition is violated at the two cells neighboring  $x = 3\pi/2$ , and the numerical solution is not convergent towards the exact solution. As we slowly increasing the value of  $C$ , we could observe better and better convergence property. In particular, once  $C$  passes some threshold, its effect on the quality of the solution is minimum, and bigger values of  $C$  only cause slightly larger numerical errors. This is also demonstrated in Table 2.2. For this problem, the viscosity solution is not smooth, so we do not expect the full  $(k + 1)$ -th order accuracy for this example. However, for different values of  $C$  ranging from 0.125 to 1.0, the numerical errors listed in Table 2.2 are all of second order. Actually, for all of the simulations performed in this paper, we find that  $C = 0.25$  to be a good choice of the penalty constant. Unless otherwise noted, for the remaining of the paper, we will use  $C = 0.25$ .

**Example 2.4.3** One-dimensional Burgers' equation with smooth initial condition

$$\begin{cases} \varphi_t + \frac{1}{2}\varphi_x^2 = 0, & x \in [0, 2\pi] \\ \varphi(x, 0) = \sin(x), \\ \varphi(0, t) = \varphi(2\pi, t). \end{cases} \quad (2.19)$$

At  $t = 0.5$ , the solution is still smooth, and the optimal  $(k + 1)$ -th accuracy is obtained for  $P^k$  polynomials with both uniform and random meshes, see Tables 2.3 and 2.4. At  $t = 1$ , there will be a shock in  $\varphi_x$ , and our scheme could capture the kink sharply as shown in Figure 2.2.

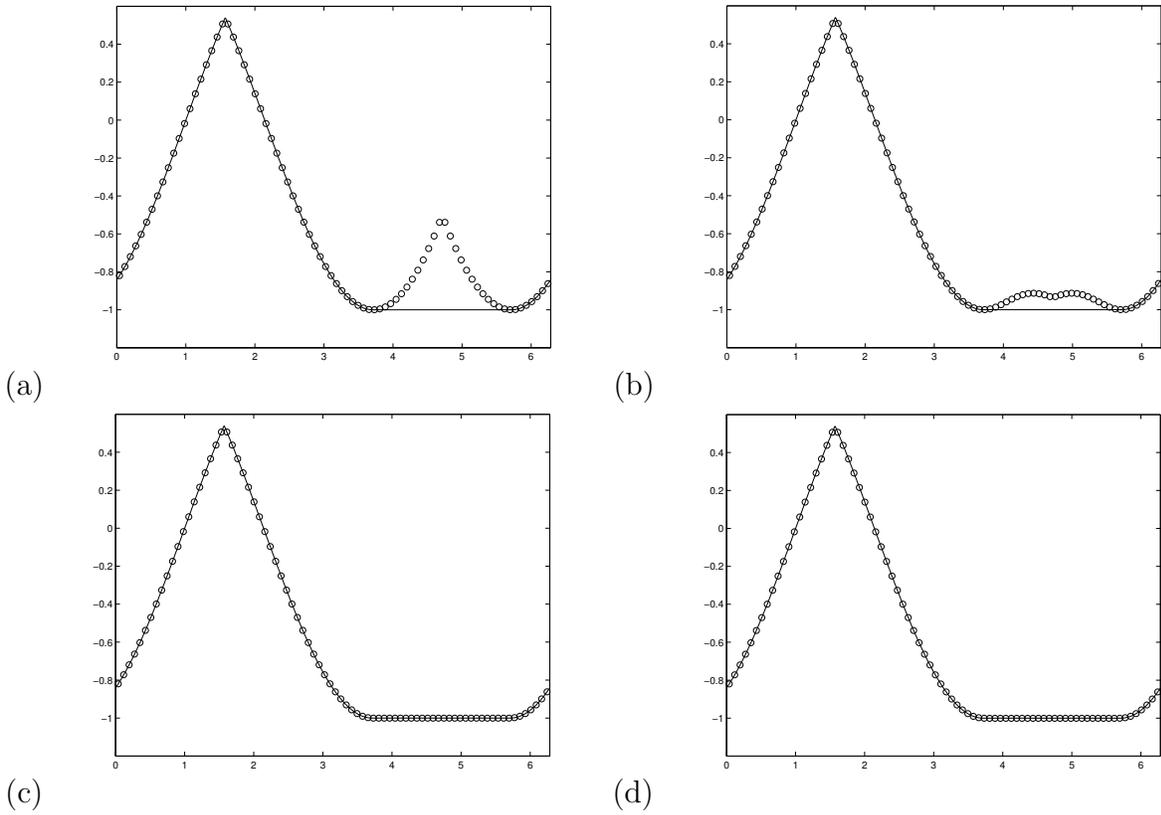


Figure 2.1: Example 2.4.2. The numerical solution with various values of penalty constant  $C$ : (a)  $C = 0$ , (b)  $C = 0.001$ , (c)  $C = 0.25$ , and (d)  $C = 1$ . Here  $t = 1$ ,  $CFL = 0.1$ ,  $P^2$  polynomials,  $M = 80$ . Solid line: the exact solution; circles: the numerical solution. *For interpretation of the references to color in this and all other figures, the reader is referred to the electronic version of this thesis.*

Table 2.2: Errors and numerical orders of accuracy for Example 2.4.2 when using  $P^2$  polynomials and third order Runge-Kutta time discretization on a uniform mesh of  $M$  cells. Final time  $t = 1$ .  $CFL = 0.1$ .

M	$L^1$ error	order	$L^2$ error	order	$L^\infty$ error	order
$C = 1.0$						
40	1.05E-03		1.85E-03		3.49E-03	
80	2.71E-04	1.95	4.78E-04	1.95	8.73E-04	2.00
160	6.89E-05	1.98	1.21E-04	1.98	2.18E-04	2.00
320	1.73E-05	1.99	3.06E-05	1.98	5.46E-05	2.00
640	4.34E-06	2.00	7.67E-06	2.00	1.37E-05	2.00
$C = 0.5$						
40	9.92E-04		1.74E-03		3.28E-03	
80	2.56E-04	1.95	4.50E-04	1.95	8.22E-04	2.00
160	6.49E-05	1.98	1.14E-04	1.98	2.06E-04	2.00
320	1.63E-05	1.99	2.88E-05	1.98	5.14E-05	2.00
640	4.09E-06	2.00	7.22E-06	2.00	1.29E-05	2.00
$C = 0.25$						
40	8.74E-04		1.53E-03		2.87E-03	
80	2.25E-04	1.96	3.95E-04	1.95	7.19E-04	2.00
160	5.69E-05	1.98	1.00E-04	1.98	1.80E-04	2.00
320	1.43E-05	1.99	2.52E-05	1.99	4.50E-05	2.00
640	3.58E-06	2.00	6.32E-06	1.99	1.13E-05	2.00
$C = 0.125$						
40	6.38E-04		1.10E-03		2.05E-03	
80	1.62E-04	1.97	2.84E-04	1.96	5.14E-04	2.00
160	4.09E-05	1.98	7.18E-05	1.98	1.29E-04	2.00
320	1.03E-05	1.99	1.81E-05	1.99	3.23E-05	2.00
640	2.57E-06	2.00	4.53E-06	2.00	8.57E-06	1.91

Table 2.3: Errors and numerical orders of accuracy for Example 2.4.3 when using  $P^2$  polynomials and third order Runge-Kutta time discretization on a uniform mesh of  $M$  cells. Penalty constant  $C = 0.25$ . Final time  $t = 0.5$ .  $CFL = 0.1$ .

M	$L^1$ error	order	$L^2$ error	order	$L^\infty$ error	order
$P^1$						
40	8.45E-04		1.23E-03		5.04E-03	
80	2.02E-04	2.07	2.99E-04	2.04	1.27E-03	1.99
160	4.93E-05	2.03	7.42E-05	2.01	3.42E-04	1.89
320	1.22E-05	2.01	1.86E-05	2.00	9.08E-05	1.91
640	3.04E-06	2.01	4.66E-06	2.00	2.36E-05	1.94
$P^2$						
40	1.27E-05		2.33E-05		1.28E-04	
80	1.53E-06	3.05	2.93E-06	2.99	2.10E-05	2.61
160	1.91E-07	3.00	3.73E-07	2.98	2.52E-06	3.06
320	2.39E-08	3.00	4.74E-08	2.98	3.56E-07	2.82
640	3.63E-09	2.72	6.23E-09	2.93	4.82E-08	2.88

Table 2.4: Errors and numerical orders of accuracy for Example 2.4.3 when using  $P^1$  and  $P^2$  polynomials and third order Runge-Kutta time discretization on a random mesh with 40% perturbation of  $M$  cells. Penalty constant  $C = 0.25$ . Final time  $t = 0.5$ .  $CFL = 0.1$ .

M	$L^1$ error	order	$L^2$ error	order	$L^\infty$ error	order
$P^1$						
40	1.23E-03		1.91E-03		1.01E-02	
80	2.70E-04	2.19	4.25E-04	2.17	2.59E-03	1.96
160	6.70E-05	2.01	1.05E-04	2.01	6.22E-04	2.06
320	1.62E-05	2.05	2.67E-05	1.97	2.03E-04	1.61
640	3.97E-06	2.03	6.69E-06	2.00	6.52E-05	1.64
$P^2$						
40	2.27E-05		4.52E-05		2.96E-04	
80	2.54E-06	3.16	5.84E-06	2.95	5.25E-05	2.50
160	3.19E-07	3.00	6.87E-07	3.09	5.82E-06	3.17
320	4.00E-08	3.00	9.34E-08	2.88	8.96E-07	2.70
640	5.38E-09	2.89	1.16E-08	3.01	1.32E-07	2.77

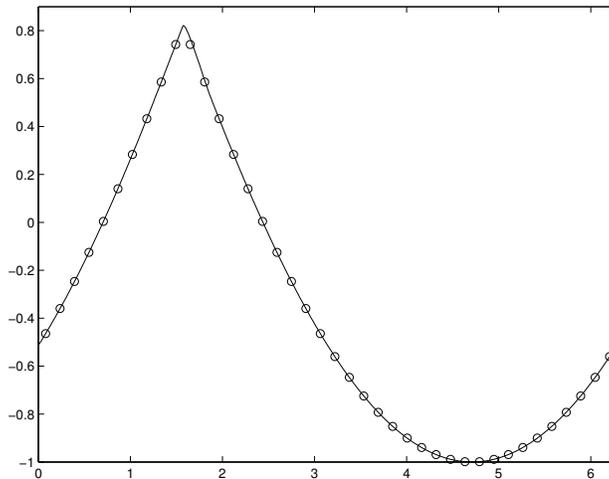


Figure 2.2: Example 2.4.3. Here  $t = 1.5$ ,  $CFL = 0.1$ ,  $P^2$  polynomials,  $M = 40$ . Penalty constant  $C = 0.25$ . Solid line: the exact solution; circles: the numerical solution.

**Example 2.4.4** One-dimensional Burgers' equation with a nonsmooth initial condition

$$\left\{ \begin{array}{l} \varphi_t + \frac{\varphi^2}{2} = 0, \quad x \in [0, 2\pi] \\ \varphi(x, 0) = \begin{cases} \pi - x, & \text{if } 0 \leq x \leq \pi, \\ x - \pi, & \text{if } \pi < x \leq 2\pi, \end{cases} \\ \varphi(0, t) = \varphi(2\pi, t). \end{array} \right. \quad (2.20)$$

The exact solution should have a rarefaction wave forming in its derivative, so the initial sharp corner at  $x = \pi$  should be smeared out at later times. Since the entropy condition is violated by the Roe type scheme, the entropy fix is necessary for convergence. Figure 2.3 shows the comparison of our schemes with various values of penalty constant  $C$  for this nonlinear problem. Again, we could see that  $C = 0.25$  is a good choice for this example.

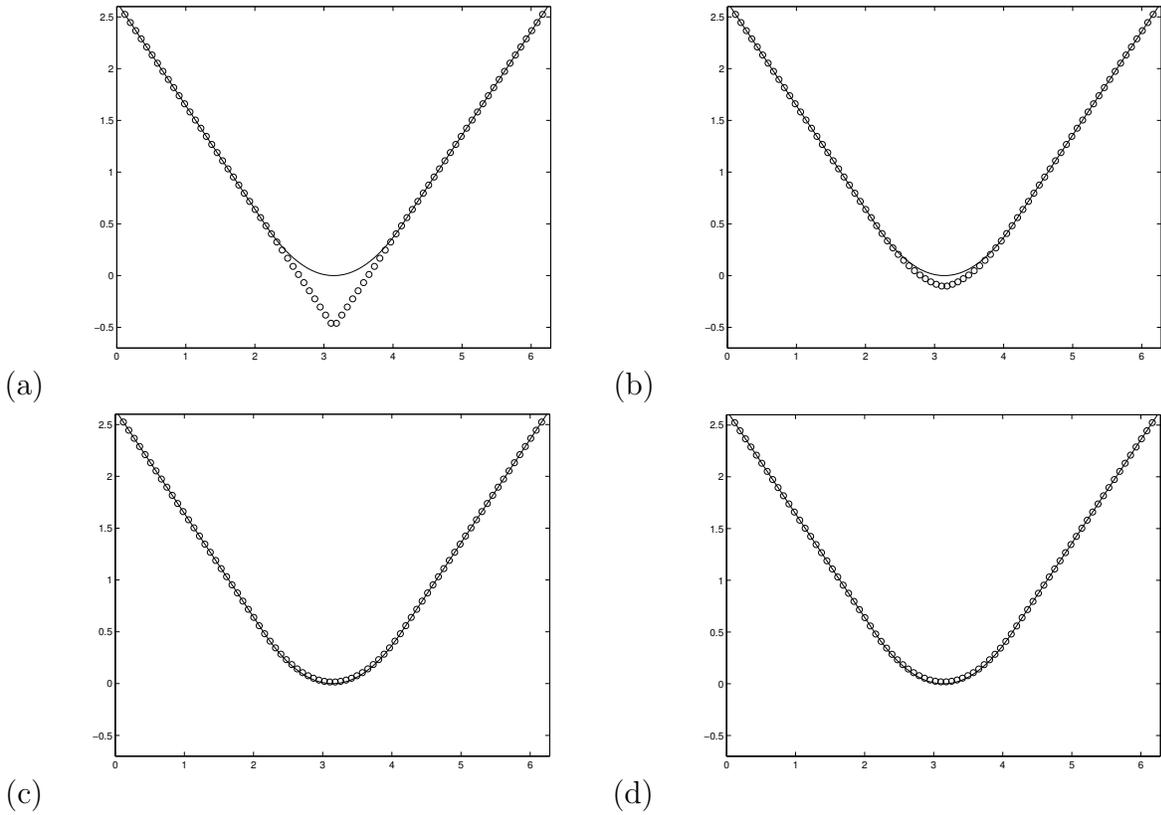


Figure 2.3: Example 2.4.4. The numerical solution with various values of penalty constant  $C$ : (a)  $C = 0$ , (b)  $C = 0.001$ , (c)  $C = 0.25$ , and (d)  $C = 1.0$ . Here  $t = 1$ ,  $CFL = 0.1$ ,  $P^2$  polynomials,  $M = 80$ . Solid line: the exact solution; circles: the numerical solution.

Table 2.5: Errors and numerical orders of accuracy for Example 2.4.5 when using  $P^2$  polynomials and third order Runge-Kutta time discretization on a uniform mesh of  $M$  cells. Penalty constant  $C = 0.25$ . Final time  $t = 1$ .  $CFL = 0.1$ .

M	$L^1$ error	order	$L^2$ error	order	$L^\infty$ error	order
40	6.24E-04		1.09E-03		2.13E-03	
80	1.69E-04	1.88	2.98E-04	1.87	5.54E-04	1.94
160	4.35E-05	1.96	7.67E-05	1.96	1.40E-04	1.98
320	1.10E-05	1.99	1.94E-05	1.99	3.51E-05	2.00
640	2.75E-06	2.00	4.88E-06	1.99	8.77E-06	2.00

**Example 2.4.5** One-dimensional Eikonal equation

$$\begin{cases} \varphi_t + |\varphi_x| = 0, & x \in [0, 2\pi] \\ \varphi(x, 0) = \sin(x), \\ \varphi(0, t) = \varphi(2\pi, t). \end{cases} \quad (2.21)$$

The exact solution is the same as the exact solution of Example 2.4.2. Our scheme could capture the viscosity solution of this nonsmooth Hamiltonian. The numerical errors and orders of accuracy using  $P^2$  polynomials are listed in Table 2.5. Since the solution is not smooth, we do not expect the optimal  $(k + 1)$ -th order accuracy for  $P^k$  polynomials.

**Example 2.4.6** One-dimensional equation with a nonconvex Hamiltonian

$$\begin{cases} \varphi_t - \cos(\varphi_x + 1) = 0, & x \in [-1, 1] \\ \varphi(x, 0) = -\cos(\pi x), \\ \varphi(-1, t) = \varphi(1, t). \end{cases} \quad (2.22)$$

This example involves a nonconvex Hamiltonian with smooth initial data. At  $t = 0.5/\pi^2$ , the exact solution is still smooth, and numerical results are presented in Table 2.6, demon-

Table 2.6: Errors and numerical orders of accuracy for Example 2.4.6 when using  $P^2$  polynomials and third order Runge-Kutta time discretization on a uniform mesh of  $M$  cells. Penalty constants  $C = 0.25$ . Final time  $t = 0.5/\pi^2$ .  $CFL = 0.1$ .

M	$L^1$ error	order	$L^2$ error	order	$L^\infty$ error	order
40	1.46E-05		2.16E-05		9.89E-05	
80	1.79E-06	3.02	2.87E-06	2.91	1.59E-05	2.64
160	2.22E-07	3.01	3.73E-07	2.95	2.39E-06	2.74
320	2.76E-08	3.01	4.79E-08	2.96	3.39E-07	2.82
640	3.51E-09	2.98	6.13E-09	2.97	4.53E-08	2.90

strating the optimal order of accuracy of the scheme. By the time  $t = 1.5/\pi^2$ , nonsmooth features would develop in  $\varphi$ , which are reliably captured in Figure 2.4.

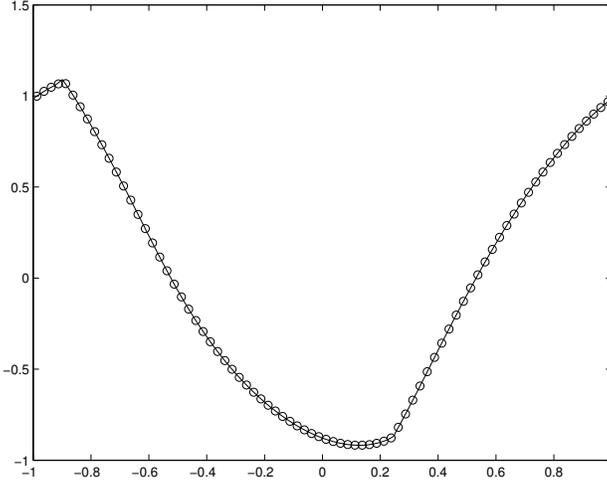


Figure 2.4: Example 2.4.6. Here  $t = 1.5/\pi^2$ ,  $CFL = 0.1$ ,  $P^2$  polynomials,  $M = 80$ . Penalty constant  $C = 0.25$ . Solid line: the exact solution; circles: the numerical solution.

**Example 2.4.7** One-dimensional Riemann problem with a nonconvex Hamiltonian

$$\begin{cases} \varphi_t + \frac{1}{4}(\varphi_x^2 - 1)(\varphi_x^2 - 4) = 0, & x \in [-1, 1]. \\ \varphi(x, 0) = -2|x|. \end{cases} \quad (2.23)$$

Table 2.7: Errors and numerical orders of accuracy for Example 2.4.7 when using  $P^2$  polynomials and third order Runge-Kutta time discretization on a uniform mesh of  $M$  cells.  $CFL = 0.05$ . Penalty constant  $C = 0.25$ . Final time  $t = 1$ . A minmod limiter is used.

M	$L^1$ error	order	$L^2$ error	order	$L^\infty$ error	order
Even M						
40	9.49E-03		2.21E-02		5.96E-02	
80	4.64E-03	1.03	1.10E-02	1.00	3.17E-02	0.91
160	2.28E-03	1.03	5.48E-03	1.00	1.64E-02	0.95
320	1.12E-03	1.02	2.73E-03	1.01	8.40E-03	0.97
640	5.60E-04	1.00	1.36E-03	1.00	4.27E-03	0.98
Odd M						
41	2.81E-03		6.74E-03		2.94E-02	
81	1.34E-03	1.09	3.35E-03	1.03	2.38E-02	0.31
161	6.41E-04	1.07	1.61E-03	1.06	9.88E-03	1.28
321	3.17E-04	1.02	7.99E-04	1.01	4.36E-03	1.19
641	1.56E-04	1.02	3.96E-04	1.02	3.12E-03	0.49

For this problem, the initial condition has a singularity at  $x = 0$ . Similar to [62, 84], a nonlinear limiter is needed in order to capture the viscosity solution. We use the standard minmod limiter [31]. This example and Example 2.4.14 are the only examples needing nonlinear limiting in this paper.

The numerical solutions with and without the limiter are listed in Figure 2.5 for odd and even values of  $M$ . Those different behaviors are due to the fact that the singular point  $x = 0$  would be exactly located at the cell interface for even  $M$  but not odd  $M$  at  $t = 0$ . We note that the method with limiter can correctly capture the viscosity solution for both even and odd  $M$ . The numerical errors and orders of accuracy using  $P^2$  polynomials with limiters are listed in Table 2.7. We could see that both methods converge, while the odd  $M$  giving slightly smaller errors. However, similar to [62], the method is only first order accurate for this nonsmooth problem.

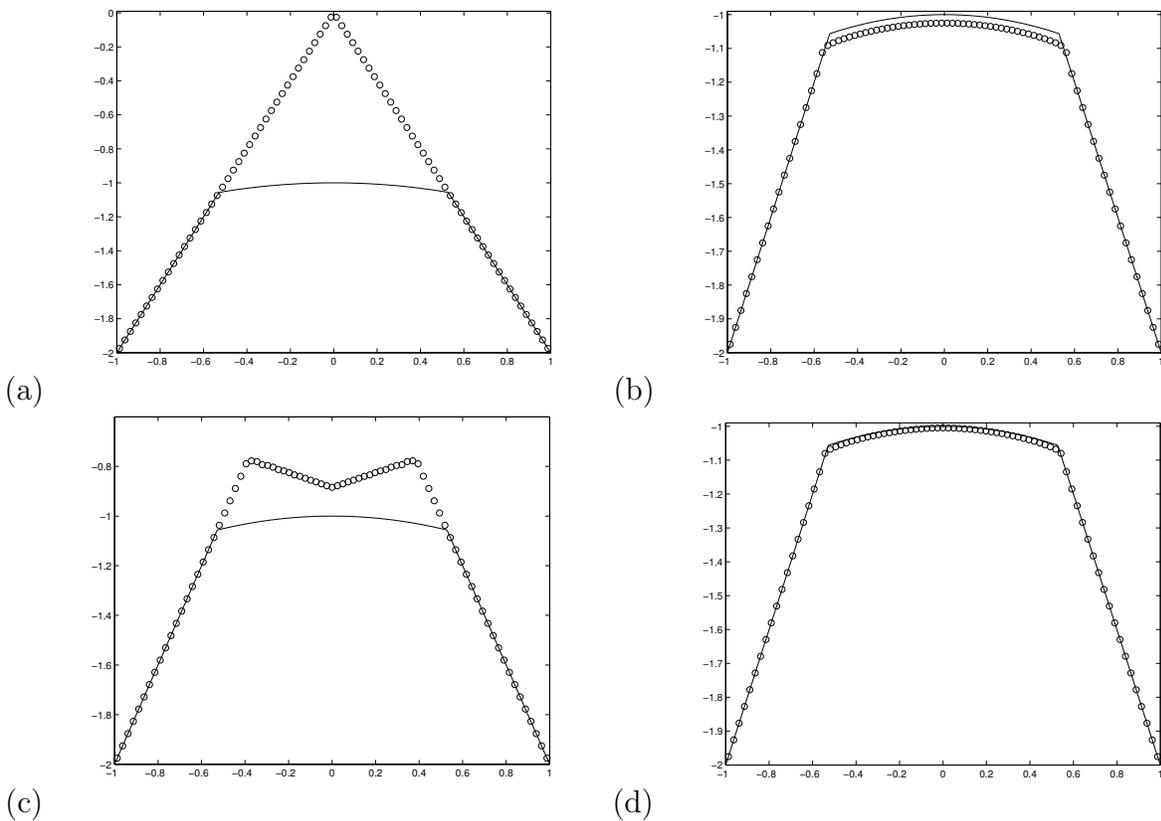


Figure 2.5: Example 2.4.7. Comparison of the numerical solution with and without the limiter: (a)  $M = 80$ , without limiter; (b)  $M = 80$ , with limiter; (c)  $M = 81$ , without limiter; and (d)  $M = 81$ , with limiter. Here  $t = 1$ ,  $P^2$  polynomials,  $CFL = 0.05$ . Penalty constant  $C = 0.25$ . Solid line: the exact solution; circles: the numerical solution.

## 2.4.2 Two-dimensional Results

In this subsection, we provide computational results for two-dimensional HJ equations on both Cartesian and unstructured meshes.

**Example 2.4.8** Two-dimensional linear problem with smooth variable coefficients

$$\varphi_t - y\varphi_x + x\varphi_y = 0. \quad (2.24)$$

The computational domain is  $[-1, 1]^2$ . The initial condition is given by

$$\varphi_0(x, y) = \begin{cases} 0, & 0.3 \leq r, \\ 0.3 - r, & 0.1 < r < 0.3, \\ 0.2, & r \leq 0.1, \end{cases} \quad (2.25)$$

where  $r = \sqrt{(x - 0.4)^2 + (y - 0.4)^2}$ . We impose periodic boundary condition on the domain.

This is a solid body rotation around the origin. The exact solution can be expressed as

$$\varphi(x, y, t) = \varphi_0(x \cos(t) + y \sin(t), -x \sin(t) + y \cos(t)). \quad (2.26)$$

For this problem, same as the argument in Example 2.4.1, the choice of  $C$  does not have an effect on the scheme. We list the numerical errors and orders in Table 2.8. With this nonsmooth initial condition, we do not expect to obtain  $(k + 1)$ -th order of accuracy. At  $t = 2\pi$ , i.e. one period of rotation, we take a snapshot at the line  $y = x$  in Figure 2.6. It can be clearly seen that a higher order scheme can yield better results for this nonsmooth initial condition.

Table 2.8: Errors and numerical orders of accuracy for Example 2.4.8 when using  $P^2$  polynomials and third order Runge-Kutta time discretization on a uniform mesh of  $M \times M$  cells. Final time  $t = 1$ .  $CFL = 0.1$ .

M	$L^1$ error	order	$L^2$ error	order	$L^\infty$ error	order
10	1.21E-03		3.10E-03		2.21E-02	
20	4.13E-04	1.55	1.32E-03	1.23	1.14E-02	0.95
40	1.38E-04	1.58	5.51E-04	1.26	6.49E-03	0.81
80	4.74E-05	1.54	2.36E-04	1.22	3.62E-03	0.84
160	1.54E-05	1.62	1.01E-04	1.23	2.07E-03	0.81

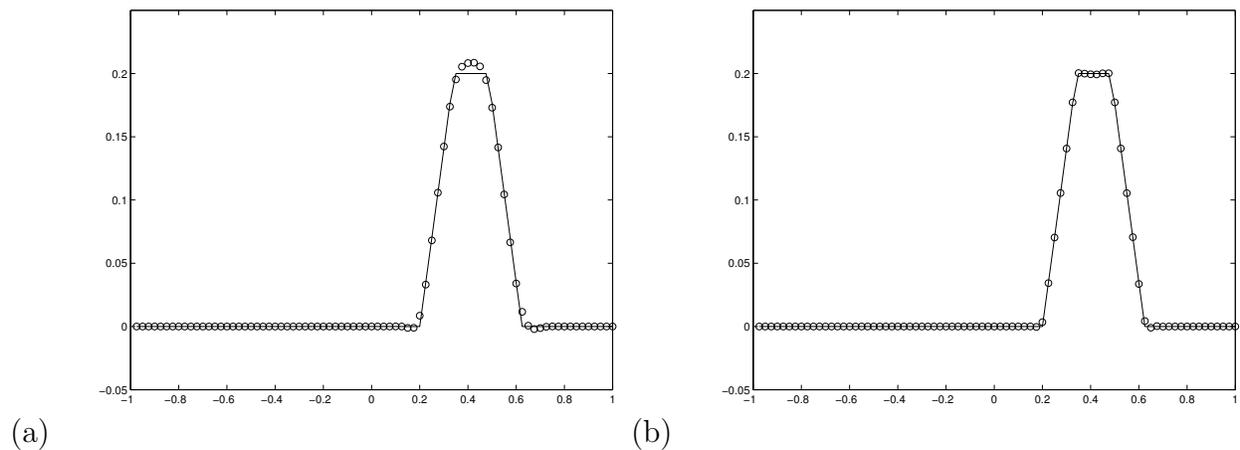


Figure 2.6: Example 2.4.8. Here  $t = 2\pi$ ,  $CFL = 0.1$ ,  $80 \times 80$  uniform mesh. (a)  $P^1$  polynomials; (b)  $P^2$  polynomials. One dimensional cut of  $45^\circ$  with the  $x$  axis. Solid line: the exact solution; circles: the numerical solution.

Table 2.9: Errors and numerical orders of accuracy for Example 3.9 when using  $P^2$  polynomials and third order Runge-Kutta time discretization on a uniform mesh of  $M \times M$  cells. Final time  $t = 1$ .  $CFL = 0.1$ .

M	$L^1$ error	order	$L^2$ error	order	$L^\infty$ error	order
20	1.42E-03		1.03E-02		2.79E-01	
40	1.54E-04	3.20	1.47E-03	2.81	5.25E-02	2.41
80	1.10E-05	3.81	1.10E-04	3.73	5.77E-03	3.19
160	1.12E-06	3.30	1.15E-05	3.26	8.96E-04	2.69

**Example 2.4.9** We solve the same equation (2.24) as in Example 2.4.8, but with a smooth initial condition as

$$\varphi_0(x, y) = \exp\left(-\frac{(x - 0.4)^2 + (y - 0.4)^2}{2\sigma^2}\right). \quad (2.27)$$

The constant  $\sigma = 0.05$  is chosen such that at the domain boundary,  $\varphi$  is very small, hence imposing the periodic boundary condition will lead to small errors. We then could observe the optimal order of accuracy in Table 2.9.

**Example 2.4.10** Two-dimensional Burgers' equation

$$\begin{cases} \varphi_t + \frac{(\varphi_x + \varphi_y + 1)^2}{2} = 0, \\ \varphi(x, y, 0) = -\cos\left(\frac{\pi(x + y)}{2}\right), \end{cases} \quad (2.28)$$

with periodic boundary condition on the domain  $[-2, 2]^2$ .

In this example, we test the performance of our method on unstructured meshes. A sample mesh used with characteristic length  $h = 1/4$  is given in Figure 2.7. At  $t = 0.5/\pi^2$ , the solution is still smooth. Numerical errors and order of accuracy using  $P^2$  polynomials are listed in Table 2.10, demonstrating the optimal order of accuracy. At  $t = 1.5/\pi^2$ , the solution is no longer smooth. Our scheme could capture the viscosity solution as shown in Figure 2.8.

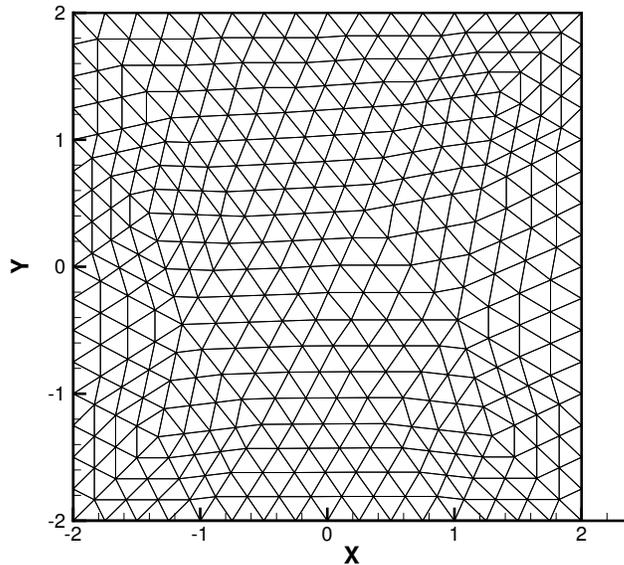


Figure 2.7: Examples 2.4.10 and 2.4.13. The unstructured mesh used with characteristic length  $h = 1/4$ .

Table 2.10: Errors and numerical orders of accuracy for Example 2.4.10 when using  $P^2$  polynomials and third order Runge-Kutta time discretization on triangular meshes with characteristic length  $h$ . Penalty constant  $C = 0.25$ . Final time  $t = 0.5/\pi^2$ .  $CFL = 0.1$ .

$h$	$L^1$ error	order	$L^2$ error	order	$L^\infty$ error	order
1	1.36E-02		2.31E-02		2.22E-01	
1/2	1.77E-03	2.93	3.23E-03	2.84	5.14E-02	2.11
1/4	2.25E-04	2.98	4.50E-04	2.84	8.95E-03	2.52
1/8	2.74E-05	3.04	5.82E-05	2.95	1.30E-03	2.78
1/16	3.40E-06	3.01	7.53E-06	2.95	1.84E-04	2.83

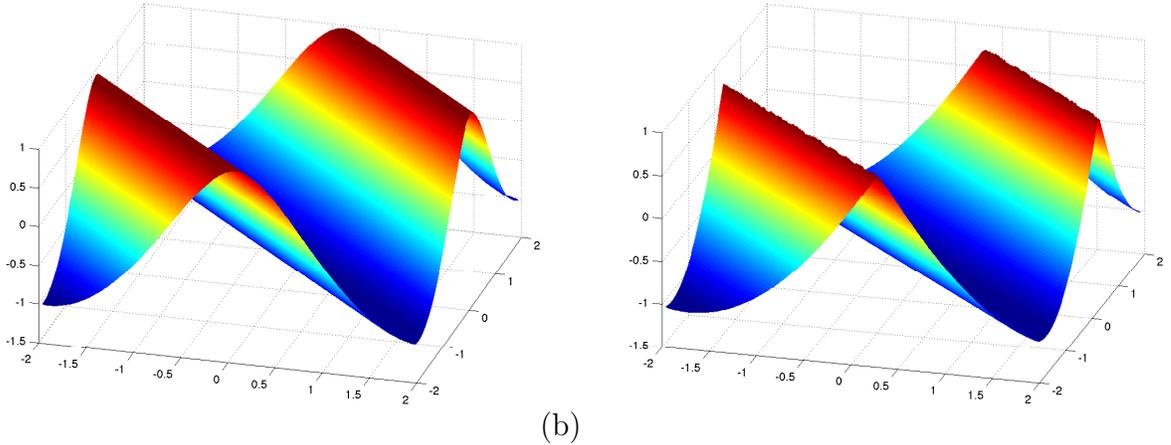


Figure 2.8: Example 2.4.10. Here  $CFL = 0.1$ ,  $P^2$  polynomials. Triangular mesh with characteristic length  $1/8$ . 2816 elements. Penalty constant  $C = 0.25$ . (a)  $t = 0.5/\pi^2$ ; (b)  $t = 1.5/\pi^2$ .

Table 2.11: Errors and numerical orders of accuracy for Example 2.4.11 when using  $P^2$  polynomials and third order Runge-Kutta time discretization on a uniform mesh of  $M \times M$  cells. Penalty constant  $C = 0.25$ . Final time  $t = 0.8$ .  $CFL = 0.1$ .

M	$L^1$ error	order	$L^2$ error	order	$L^\infty$ error	order
10	2.22E-03		3.95E-03		4.78E-02	
20	2.75E-04	2.98	4.50E-04	2.98	7.79E-03	2.62
40	3.70E-05	2.89	7.33E-05	2.77	1.50E-03	2.38
80	4.80E-06	2.95	9.83E-06	2.90	2.40E-04	2.64

**Example 2.4.11** Two-dimensional nonlinear equation from [62]

$$\begin{cases} \varphi_t + \varphi_x \varphi_y = 0, \\ \varphi(x, y, 0) = \sin(x) + \cos(y), \end{cases} \quad (2.29)$$

with periodic boundary condition on the domain  $[-\pi, \pi]^2$ .

At  $t = 0.8$ , the solution is still smooth, as shown in the left figure of Figure 2.9. Numerical errors and order of accuracy using  $P^2$  polynomials are listed in Table 2.11, demonstrating the optimal order of accuracy. At  $t = 1.5$ , singular features would form in the solution, as shown in the right figure of Figure 2.9.

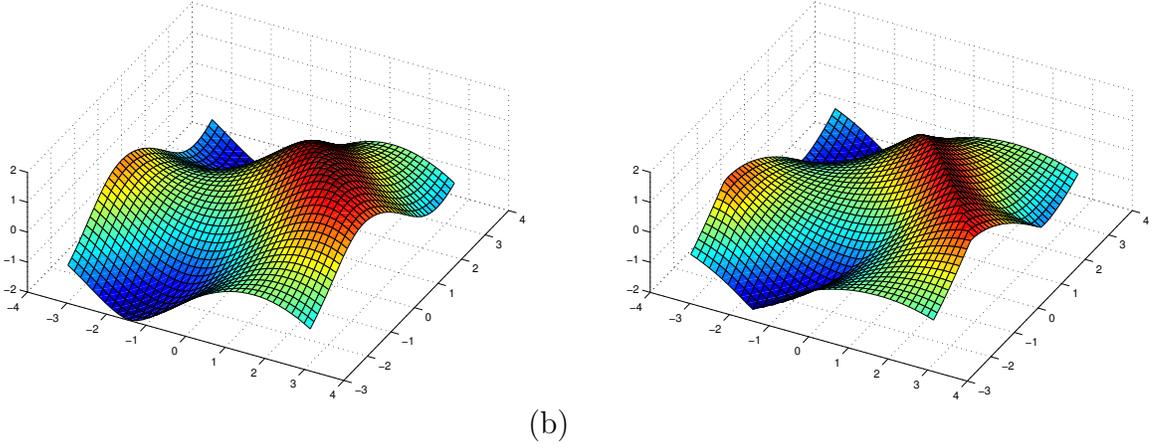


Figure 2.9: Example 2.4.11. Here  $CFL = 0.1$ ,  $P^2$  polynomials on a  $80 \times 80$  uniform mesh. Penalty constant:  $C = 0.25$ . (a)  $t = 0.8$ ; (b)  $t = 1.5$ .

**Example 2.4.12** An example related to controlling optimal cost determination from [68]

$$\begin{cases} \varphi_t + \sin(y)\varphi_x + (\sin(x) + \text{sign}(\varphi_y))\varphi_y - \frac{1}{2}\sin^2(y) + \cos(x) - 1 = 0, \\ \varphi(x, y, 0) = 0, \end{cases} \quad (2.30)$$

with periodic boundary condition on the domain  $[-\pi, \pi]^2$ .

The Hamiltonian is not smooth in this example. Our scheme can capture the features of the viscosity solution well. The numerical solution (left) and the optimal control term  $\text{sign}(\varphi_y)$  (right) at  $t = 1$  are shown in Figure 2.10.

**Example 2.4.13** Two-dimensional equation with a nonconvex Hamiltonian

$$\begin{cases} \varphi_t - \cos(\varphi_x + \varphi_y + 1) = 0, \\ \varphi(x, y, 0) = -\cos\left(\frac{\pi}{2}(x + y)\right), \end{cases} \quad (2.31)$$

with periodic boundary condition on the domain  $[-2, 2]^2$ .

We use the same unstructured mesh as in Example 2.4.10, see for example Figure 2.7.

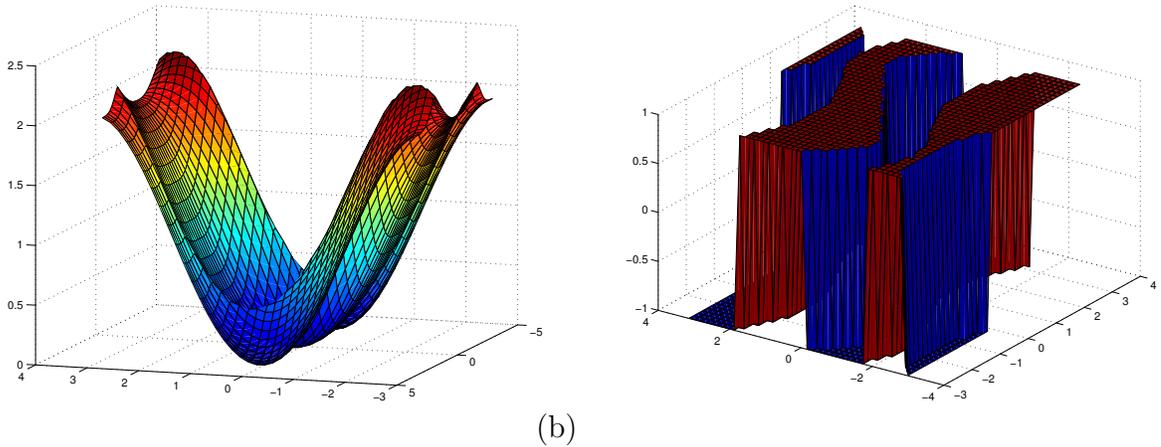


Figure 2.10: Example 2.4.12. Here  $t = 1$ ,  $CFL = 0.1$ ,  $P^2$  polynomials on a  $40 \times 40$  uniform mesh. Penalty constant:  $C = 0.25$ . (a) the numerical solution; (b)  $\text{sign}(\varphi_y)$ .

At  $t = 0.5/\pi^2$ , the solution is still smooth, see Table 2.12 for numerical errors and order of accuracy using  $P^2$  polynomials. At  $t = 1.5/\pi^2$ , singular features would develop in the solution, as shown in Figure 2.11.

Table 2.12: Errors and numerical orders of accuracy for Example 2.4.13 when using  $P^2$  polynomials and third order Runge-Kutta time discretization on triangular meshes with characteristic length  $h$ . Penalty constant  $C = 0.25$ . Final time  $t = 0.5/\pi^2$ .  $CFL = 0.1$ .

$h$	$L^2$ error	order	$L^2$ error	order	$L^\infty$ error	order
1	1.05E-02		1.65E-02		1.48E-01	
1/2	1.59E-03	2.71	2.49E-03	2.73	3.11E-02	2.25
1/4	2.42E-04	2.71	4.02E-04	2.63	6.35E-03	2.29
1/8	3.28E-05	2.89	5.84E-05	2.78	1.03E-03	2.62
1/16	3.96E-06	3.05	7.45E-06	2.97	1.72E-04	2.58

**Example 2.4.14** Two-dimensional Riemann problem

$$\begin{cases} \varphi_t + \sin(\varphi_x + \varphi_y) = 0, \\ \varphi(x, y, 0) = \pi(|y| - |x|), \end{cases} \quad (2.32)$$

on the domain  $[-1, 1]^2$ .

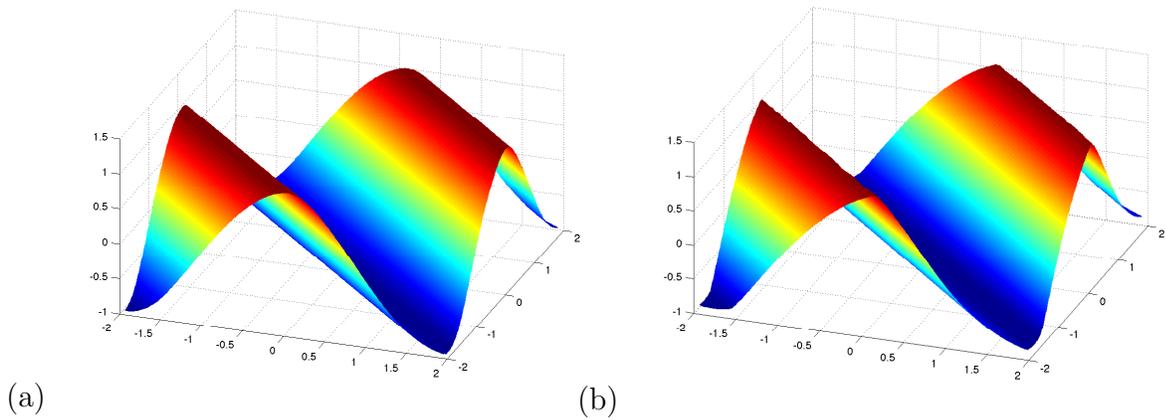


Figure 2.11: Example 2.4.13. Here  $CFL = 0.1$ ,  $P^2$  polynomials. Triangular mesh with characteristic length  $1/8$ . 2816 elements. Penalty constant:  $C = 0.25$ . (a)  $t = 0.5/\pi^2$ . (b)  $t = 1.5/\pi^2$ .

Similar to [56, 84], a nonlinear limiter is needed for convergence in this example. We use the moment limiter [60] and the numerical solution obtained by  $P^2$  polynomial at  $t = 1$  is provided in Figure 2.12.

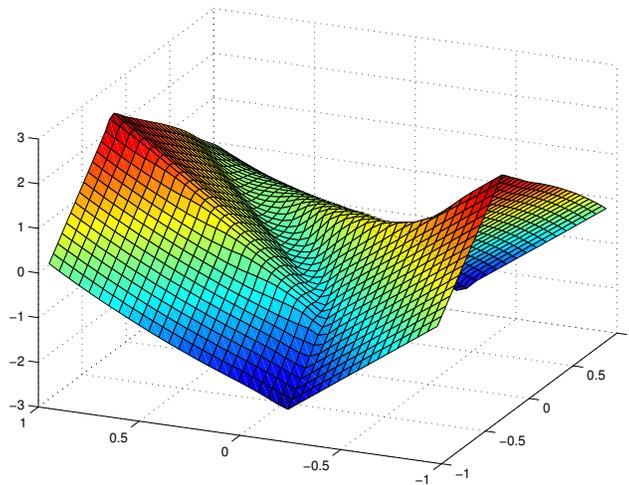


Figure 2.12: Example 2.4.14. Here  $t = 1$ ,  $CFL = 0.1$ ,  $P^2$  polynomials on a  $41 \times 41$  uniform mesh. Penalty constant:  $C = 0.25$ .

**Example 2.4.15** The problem of a propagating surface

$$\begin{cases} \varphi_t - \sqrt{\varphi_x^2 + \varphi_y^2} + 1 = 0, \\ \varphi(x, y, 0) = 1 - \frac{1}{4}(\cos(2\pi x) - 1), (\cos(2\pi y) - 1) \end{cases} \quad (2.33)$$

with periodic boundary condition on the domain  $[0, 1]^2$ . This is a special case of the example used in [67], and is also the two-dimensional Eikonal equation arising from geometric optics [59]. We use an unstructured mesh shown in Figure 2.13 with refinements near the center of the domain where the solution develops singularity. The numerical solutions at different times are displayed in Figure 2.14. Notice that the solution at  $t = 0$  is shifted downward by 0.35 to show the detail of the solutions at later times.

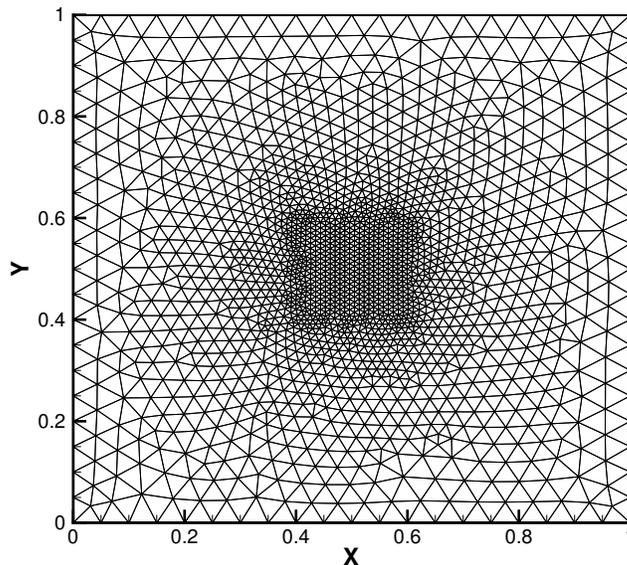


Figure 2.13: Example 2.4.15. The unstructured mesh used in the computation. Number of elements: 3480.

**Example 2.4.16** The problem of a propagating surface on the unit disk  $\{(x, y) : x^2 + y^2 \leq 1\}$

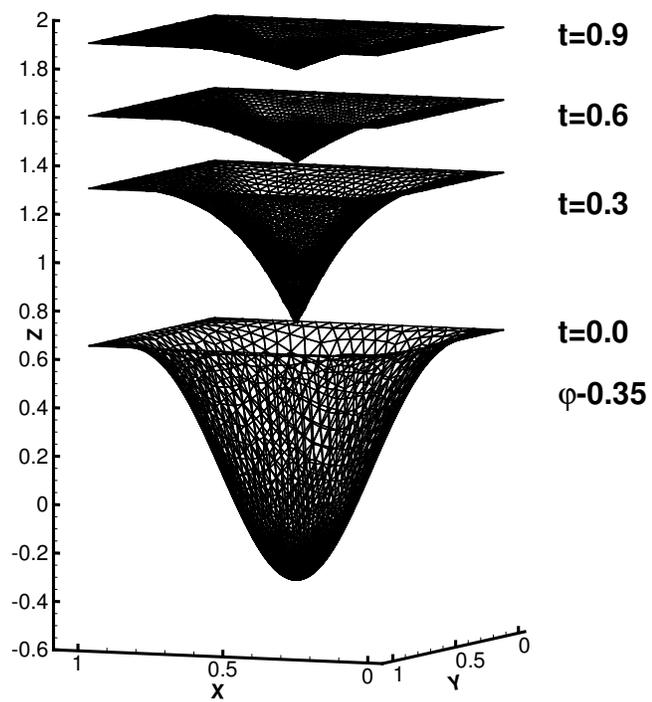


Figure 2.14: Example 2.4.15.  $CFL = 0.1$ .  $P^2$  polynomials. Penalty constant:  $C = 0.25$ . The numerical solution at the indicated times.

1}

$$\begin{cases} \varphi_t - \sqrt{\varphi_x^2 + \varphi_y^2} + 1 = 0, \\ \varphi(x, y, 0) = -\sin\left(\frac{\pi(x^2 + y^2)}{2}\right). \end{cases} \quad (2.34)$$

We use an unstructured mesh as depicted in Figure 2.15 with refinements near the origin where solution develops singularity. The numerical solutions at different times are displayed in Figure 2.16. Notice that the solution at  $t = 0$  is shifted downward by 0.2 to show the detail of the solutions at later times.

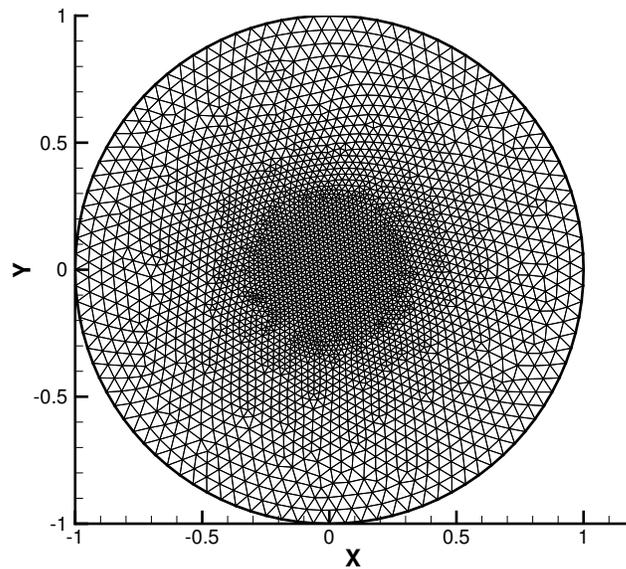


Figure 2.15: Example 2.4.16. The unstructured mesh used in the computation. Number of elements: 5890.

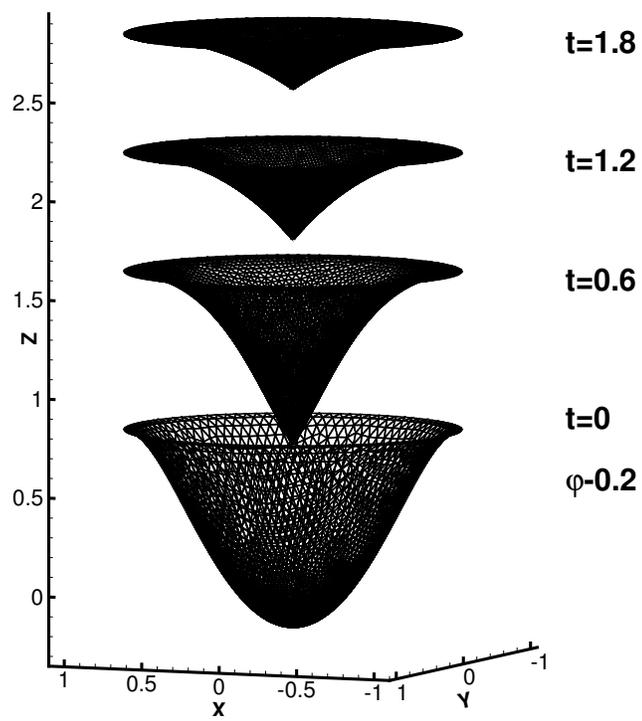


Figure 2.16: Example 2.4.16. Here  $CFL = 0.1$ ,  $P^2$  polynomials. Penalty constant:  $C = 0.25$ . The numerical solution at the indicated times.

# Chapter 3

## Sparse Grid DG Methods for High-Dimensional Elliptic Equations

In this chapter, we will develop sparse grid DG methods to treat high-dimensional PDEs. As an initial effort in our investigation, we focus on elliptic problems and use the symmetric IP method for the discretization of the following  $d$ -dimensional model equation,

$$-\nabla \cdot (\mathbf{K} \nabla u) = f \quad \text{in } \Omega = [0, 1]^d, \quad (3.1)$$

with Dirichlet boundary condition

$$u = g \quad \text{on } \partial\Omega, \quad (3.2)$$

where  $u(\mathbf{x}) : \Omega \rightarrow \mathbb{R}$  is the unknown function and the matrix-valued coefficient function  $\mathbf{K} = (k_{i,j})_{1 \leq i,j \leq d}$  is symmetric positive definite and bounded below and above uniformly, i.e. there exist two positive constant  $K_0, K_1$ , such that

$$\forall \mathbf{x} \in \Omega, \quad K_0 \mathbf{x} \cdot \mathbf{x} \leq \mathbf{K} \mathbf{x} \cdot \mathbf{x} \leq K_1 \mathbf{x} \cdot \mathbf{x}.$$

As shown in the previous chapter, the DG method has two essential ingredients in its design: (1) a (weak) formulation of the method based on the underlying PDEs, (2) an under-

lying approximation space. The appropriate choice of the formulation of the scheme as well as approximation properties of the space can guarantee the methods with desirable properties such as stability and convergence. One of the main advantages of the DG methods compared to continuous finite element methods is their freedom in choosing the approximation space due to the lack of continuity requirement. In fact, DG methods based on traditional piecewise polynomial space have been extensively studied for the last few decades, and the methods are rather mature for applications such as elliptic, parabolic and hyperbolic problems. Theoretical foundations are laid out using tools from finite element analysis. In recent years, the idea of using non-polynomial spaces [85, 81], or polynomial spaces with specific properties such as locally divergence-free properties [29] has been explored. They are mainly driven by the needs to mimic particular properties of the exact solution. Another interesting and more relevant development, which was inspired by early success in the finite volume framework [52], is to use multiresolution analysis with DG methods for adaptivity [22, 5, 57, 55, 39] and trouble-cell indicator [80] for hyperbolic conservation laws.

In this chapter, we will develop sparse grid IPDG methods for the model elliptic equation, starting with the construction and approximation properties of the sparse finite element space, and then apply it to elliptic problems in conjunction with IP method, with focus on error estimates, implementation and numerical simulations. The rest of this chapter is organized as follows: in Section 3.1, we construct and analyze the DG approximation space on sparse grids. In Section 3.2, we formulate the scheme and perform error analysis of the sparse grid IP method. Numerical examples in multi-dimensions are given to validate the accuracy and performance of the scheme in Section 3.3.

Notice that our method is restricted to problems on a box shaped domain. For more general domains, either a coordinate transformation or a more complex sparse finite element

space based on hierarchical decompositions on unstructured meshes will be required. We do not pursue such cases in this thesis and leave them to future study.

## 3.1 DG Finite Element Spaces on Sparse Grids

In this section, we will introduce the main ingredient of our algorithm: the DG finite element space on sparse grids. We proceed in several steps. First, we will review the standard piecewise polynomial spaces in one dimension, and introduce a hierarchical decomposition with a set of orthonormal bases. Second, we will construct the sparse finite element space from multi-dimensional multiwavelet bases based on the 1D construction. Finally, we will discuss some key features and approximation properties of the sparse finite element space. Numerical tests are given to compare several definitions of sparse discontinuous finite element spaces.

### 3.1.1 Hierarchical Decomposition of Piecewise Polynomial Spaces in One Dimension

In this subsection, we will introduce the hierarchical representation of piecewise polynomial spaces in one dimension. Without loss of generality, consider the interval  $\Omega = [0, 1]$ , we define the  $n$ -th level grid  $\Omega_n$ , consisting of  $I_n^j = (2^{-n}j, 2^{-n}(j+1)]$ ,  $j = 0, \dots, 2^n - 1$  with uniform cell size  $h = 2^{-n}$ . On this grid, we use  $\|\cdot\|_{H^s(\Omega_n)}$  to denote the broken Sobolev norm, i.e.  $\|v\|_{H^s(\Omega_n)}^2 = \sum_{j=0}^{2^n-1} \|v\|_{H^s(I_n^j)}^2$ , where  $\|v\|_{H^s(I_n^j)}$  is the standard Sobolev norm on  $I_n^j$ . Similarly,  $|\cdot|_{H^s(\Omega_n)}$  denotes the broken Sobolev semi-norm, i.e.  $|v|_{H^s(\Omega_n)}^2 = \sum_{j=0}^{2^n-1} |v|_{H^s(I_n^j)}^2$ .

We can define

$$V_n^k = \{v : v \in P^k(I_n^j), \forall j = 0, \dots, 2^n - 1\}$$

to be the usual piecewise polynomials of degree at most  $k$  on this grid.  $V_n^k$  has degrees of freedom  $2^n(k+1)$ , and we notice that they have the nested structure for different values of  $n$ ,

$$V_0^k \subset V_1^k \subset V_2^k \subset V_3^k \subset \dots$$

Due to the nested structure, we can define the multiwavelet subspace  $W_n^k$ ,  $n = 1, 2, \dots$  as the orthogonal complement of  $V_{n-1}^k$  in  $V_n^k$  with respect to the  $L^2$  inner product on  $\Omega$ , i.e.

$$V_{n-1}^k \oplus W_n^k = V_n^k, \quad W_n^k \perp V_{n-1}^k.$$

For notational convenience, we also denote the base space  $W_0^k := V_0^k$ , which consists of all polynomials of up to degree  $k$  on  $[0, 1]$ . Now, we have found a hierarchical representation of the standard piecewise polynomial space on grid  $\{I_n^j, j = 0, \dots, 2^n - 1\}$  as  $V_N^k = \bigoplus_{0 \leq n \leq N} W_n^k$ . We remark that the space  $W_n^k$ , for  $n > 1$ , represents the finer level details when the mesh becomes refined and this is the key to the reduction in degrees of freedom in higher dimensions. The dimension of  $W_n^k$  is  $2^{n-1}(k+1)$  when  $n \geq 1$ , and  $k+1$  when  $n = 0$ .

For implementation purpose, we need to introduce basis functions for space  $W_n^k$ . The case of  $n = 0$  is trivial. It suffices to use a standard polynomial basis on  $[0, 1]$ . For example, by using the scaled Legendre polynomials, we can easily obtain a set of orthonormal bases in  $W_0^k$ . When  $n > 1$ , we will use the orthonormal bases introduced in [4]. In particular, the bases for  $W_1^k$  are defined as

$$h_i(x) = 2^{1/2} f_i(2x - 1), \quad i = 1, \dots, k + 1$$

where  $\{f_i(x), i = 1, \dots, k+1\}$  are functions supported on  $[-1, 1]$  and depend on  $k$ , with the following properties.

1. The restriction of  $f_i$  to the interval  $(0, 1)$  is a polynomial of degree  $k$ .
2. The function  $f_i$  is extended to  $(-1, 0)$  as an even or odd function according to the parity of  $i + k$ :

$$f_i(x) = (-1)^{i+k} f_i(-x).$$

3. The bases have vanishing moments:

$$\int_{-1}^1 f_j(x) x^i dx = 0 \quad i = 0, 1, \dots, j + k - 1.$$

4. The bases have the orthogonality and normality properties:

$$\int_{-1}^1 f_i(x) f_j(x) dx = \langle f_i, f_j \rangle = \delta_{ij}, \quad i, j = 1, \dots, k.$$

Functions  $f_i$  can be computed by a repeated Gram-Schmidt algorithm. The particular form of  $f_i$  for up to  $k = 4$  are listed in Table 1 in [4]. For completeness, we provide them in Table 3.1.

Multiwavelet bases in [4] retain the orthonormal properties of wavelet bases for different hierarchical levels. More generally, the basis for  $W_n^k$ ,  $n \geq 1$  are defined as

$$v_{i,n}^j(x) = 2^{(n-1)/2} h_i(2^{n-1}x - j), \quad i = 1, \dots, k+1, j = 0, \dots, 2^{n-1} - 1.$$

To make the notations consistent and compact, when  $n = 0$ , the bases  $v_{i,0}^0(x)$ ,  $i = 1, \dots, k+1$  are defined as the rescaled Legendre polynomials on  $[0, 1]$  with orthonormal

Table 3.1: Orthonormal, vanishing-moment functions  $f_1, \dots, f_{k+1}$ , for  $k = 0, \dots, 4$ , for  $x \in (0, 1)$ . The functions  $f_i$  is extended to the interval  $(-1, 1)$  as an odd or even function based on the rule  $f_i(x) = (-1)^{i+k} f_i(-x)$  for  $x \in (-1, 0)$ , and is zero elsewhere.

$k = 0$	
$f_1(x) =$	$\sqrt{\frac{1}{2}}$
$k = 1$	
$f_1(x) =$	$\sqrt{\frac{3}{2}} (-1 + 2x)$
$f_2(x) =$	$\sqrt{\frac{1}{2}} (-2 + 3x)$
$k = 2$	
$f_1(x) =$	$\frac{1}{3} \sqrt{\frac{1}{2}} (1 - 24x + 30x^2)$
$f_2(x) =$	$\frac{1}{2} \sqrt{\frac{3}{2}} (3 - 16x + 15x^2)$
$f_3(x) =$	$\frac{1}{3} \sqrt{\frac{5}{2}} (4 - 15x + 12x^2)$
$k = 3$	
$f_1(x) =$	$\sqrt{\frac{15}{34}} (1 + 4x - 30x^2 + 28x^3)$
$f_2(x) =$	$\sqrt{\frac{1}{42}} (-4 + 105x - 300x^2 + 210x^3)$
$f_3(x) =$	$\frac{1}{2} \sqrt{\frac{35}{34}} (-5 + 48x - 105x^2 + 64x^3)$
$f_4(x) =$	$\frac{1}{2} \sqrt{\frac{5}{42}} (-16 + 105x - 192x^2 + 105x^3)$
$k = 4$	
$f_1(x) =$	$\sqrt{\frac{1}{186}} (1 + 30x + 210x^2 - 840x^3 + 630x^4)$
$f_2(x) =$	$\frac{1}{2} \sqrt{\frac{1}{38}} (-5 - 144x + 1155x^2 - 2240x^3 + 1260x^4)$
$f_3(x) =$	$\sqrt{\frac{35}{14694}} (22 - 735x + 3504x^2 - 5460x^3 + 2700x^4)$
$f_4(x) =$	$\frac{1}{8} \sqrt{\frac{21}{38}} (35 - 512x + 1890x^2 - 2560x^3 + 1155x^4)$
$f_5(x) =$	$\frac{1}{2} \sqrt{\frac{7}{158}} (32 - 315x + 960x^2 - 1155x^3 + 480x^4)$

property. In summary, we have [4, 3],

$$\int_0^1 v_{i,n}^j(x) v_{i',n'}^{j'}(x) dx = \delta_{ii'} \delta_{nn'} \delta_{jj'},$$

where  $\delta_{ii'}, \delta_{nn'}, \delta_{jj'}$  are the Kronecker delta symbols.

We remark that the multiwavelet bases under consideration in this section are closely related to the original Haar wavelet [51] and the multiwavelet bases constructed in [4, 3], and used for adaptive computations of DG methods [22, 5, 57, 55, 39].

The one-dimensional construction of piecewise polynomial space is complete, as can be seen from the following property.

**Property 3.1.1 (Completeness of the One-dimensional Construction [4].)** *We define*

$$V^k := \lim_{N \rightarrow \infty} V_N^k = \bigoplus_{0 \leq n \leq \infty} W_n^k$$

*which is actually the union of all  $V_N^k$ , and observe that*

$$\overline{V^k} = L^2(0, 1).$$

*By the orthogonality of the multiwavelet bases, the completeness of the one-dimensional construction can be proved.*

Next, we will discuss about the projection operators which are essential in the finite element analysis. We define  $P_n^k$  as the standard  $L^2$  projection operator from  $L^2[0, 1]$  to  $V_n^k$ .

From there, we can introduce the increment projector,

$$Q_n^k := \begin{cases} P_n^k - P_{n-1}^k, & \text{if } n \geq 1 \\ P_0^k, & \text{if } n = 0, \end{cases}$$

then we can see that

$$W_n^k = Q_n^k L^2(0, 1).$$

In summary, we arrive at the following identities for the hierarchical decomposition of the piecewise polynomial space and the projection operator:

$$V_N^k = \bigoplus_{0 \leq n \leq N} W_n^k, \quad P_N^k = \sum_{0 \leq n \leq N} Q_n^k.$$

The properties of  $Q_n^k$  naturally relies on  $P_n^k$ . For the  $L^2$  projection, we recall the following approximation results.

**Property 3.1.2 (Convergence Property of the Projection Operator [26])** *For a function  $v \in H^{p+1}(0, 1)$ , we have the convergence property of the  $L^2$  projection  $P_n^k$  as follows: for any integer  $t$  with  $1 \leq t \leq \min\{p, k\}$ ,*

$$\|P_n^k v - v\|_{H^s(\Omega_n)} \leq c_{k,s,t} 2^{-n(t+1-s)} \|v\|_{H^{t+1}(\Omega)}, \quad (3.3)$$

where  $c_{k,s,t}$  is a constant that depends on  $k, s, t$ , but not on  $n$ .

From this property, using basic algebra, we can deduce that for  $n \geq 1$ ,

$$\|Q_n^k v\|_{H^s(\Omega_n)} \leq \tilde{c}_{k,s,t} 2^{-n(t+1-s)} \|v\|_{H^{t+1}(\Omega)},$$

with

$$\tilde{c}_{k,s,t} = c_{k,s,t} (1 + 2^{-(t+1-s)}). \quad (3.4)$$

Finally, we want to remark on a subtle issue concerning the hierarchical decomposition and the multiwavelet space. If we use a projection other than  $P_n^k$ , say  $\underline{P}_n^k$  is another projection from  $L^2(\Omega)$  to  $V_n^k$ . Then the increment projector  $\underline{Q}_n^k \neq Q_n^k$ . The space  $\underline{W}_n^k = \underline{Q}_n^k L^2(0, 1)$  is different from  $W_n^k$ , and likewise for the multiwavelet bases. However, the hierarchical structure  $V_{n-1}^k \oplus \underline{W}_n^k = V_n^k$  still holds. In our discussion in the next subsection, we will highlight the implication of this statement and show that using different definitions of the projector and the increment space will not affect the definition of the sparse discontinuous finite element space.

### 3.1.2 Sparse Discontinuous Finite Element Spaces in Multi-dimensions

In this subsection, we introduce the sparse finite element space constructed from multi-dimensional multiwavelet bases based on the one-dimensional bases in Section 3.1.1.

For a  $d$ -dimensional problem, we consider the domain  $\mathbf{x} = (x_1, \dots, x_d) \in \Omega = [0, 1]^d$ . To facilitate the discussion, we first introduce some notations on the norms and operations of multi-indices in  $\mathbb{N}_0^d$ , where  $\mathbb{N}_0$  denotes the set of nonnegative integers. For  $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_d) \in \mathbb{N}_0^d$ , we define the  $l^1$  and  $l^\infty$  norms

$$|\boldsymbol{\alpha}|_1 := \sum_{j=1}^d \alpha_j, \quad |\boldsymbol{\alpha}|_\infty := \max_{1 \leq j \leq d} \alpha_j,$$

the component-wise arithmetic operations

$$\boldsymbol{\alpha} \cdot \boldsymbol{\beta} := (\alpha_1 \beta_1, \dots, \alpha_d \beta_d), \quad c \cdot \boldsymbol{\alpha} := (c \alpha_1, \dots, c \alpha_d), \quad 2^{\boldsymbol{\alpha}} := (2^{\alpha_1}, \dots, 2^{\alpha_d}),$$

and the relational operators

$$\boldsymbol{\alpha} \leq \boldsymbol{\beta} \Leftrightarrow \alpha_j \leq \beta_j, \forall j$$

$$\boldsymbol{\alpha} < \boldsymbol{\beta} \Leftrightarrow \boldsymbol{\alpha} \leq \boldsymbol{\beta} \text{ and } \boldsymbol{\alpha} \neq \boldsymbol{\beta}.$$

Now, for a multi-index  $\mathbf{l} = (l_1, \dots, l_d) \in \mathbb{N}_0^d$ , which indicates the level of the mesh in a multivariate sense, we consider the standard rectangular grid  $\Omega_{\mathbf{l}}$  with mesh size  $h_{\mathbf{l}} := (2^{-l_1}, \dots, 2^{-l_d})$ . We define the smallest size among all dimensions to be  $h = \min\{2^{-l_1}, \dots, 2^{-l_d}\}$ , an elementary cell  $I_{\mathbf{l}}^{\mathbf{j}} = \{\mathbf{x} : x_i \in (2^{-l_i} j_i, 2^{-l_i} (j_i + 1))\}$ , and

$$\mathbf{V}_{\mathbf{l}}^k = \{\mathbf{v} : \mathbf{v}(\mathbf{x}) \in P^k(I_{\mathbf{l}}^{\mathbf{j}}), \mathbf{0} \leq \mathbf{j} \leq 2^{\mathbf{l}} - \mathbf{1}\},$$

where  $P^k(I_{\mathbf{l}}^{\mathbf{j}})$  denotes polynomials of degree up to  $k$  in each dimension on cell  $I_{\mathbf{l}}^{\mathbf{j}}$ . The space  $\mathbf{V}_{\mathbf{l}}^k$  contains the traditional tensor-product piecewise polynomials used in the DG discretizations. Moreover, if we use equal refinement of size  $2^{-N}$  in each direction, we denote the space to be  $\mathbf{V}_N^k$ , and it consists of  $(2^N(k+1))^d$  degrees of freedom.

The foundation of sparse grid is to use the tensor product of the one-dimensional hierarchical decomposition, and only chooses enough bases to guarantee suitable approximation properties. To illustrate the main ideas, we introduce

$$\mathbf{W}_{\mathbf{l}}^k = W_{l_1, x_1}^k \times W_{l_2, x_2}^k \cdots \times W_{l_d, x_d}^k,$$

where  $W_{l_i, x_i}^k$  corresponds to the space  $W_{l_i}^k$  defined in the  $i$ -th dimension as defined in the

previous subsection. The number of degrees of freedom associated with  $\mathbf{W}_1^k$  is

$$\dim(\mathbf{W}_1^k) = \prod_{i=1}^d \dim(W_{l_i}^k).$$

Based on the one-dimensional hierarchical decomposition, it is easy to see

$$\mathbf{V}_1^k = V_{l_1, x_1}^k \times V_{l_2, x_2}^k \cdots \times V_{l_d, x_d}^k = \bigoplus_{j_1 \leq l_1, \dots, j_d \leq l_d} \mathbf{W}_j^k,$$

and

$$\mathbf{V}_N^k = V_{N, x_1}^k \times V_{N, x_2}^k \cdots \times V_{N, x_d}^k = \bigoplus_{|\mathbf{l}|_\infty \leq N} \mathbf{W}_1^k.$$

The basis functions for  $\mathbf{W}_1^k$  can be defined by a tensor product

$$v_{\mathbf{i}, \mathbf{l}}^{\mathbf{j}}(\mathbf{x}) := \prod_{t=1}^d v_{i_t, l_t}^{j_t}(x_t), \quad j_t = 0, \dots, \max(0, 2^{l_t-1} - 1); \quad i_t = 1, \dots, k + 1.$$

They form a set of orthonormal basis due to the property of the one-dimensional bases.

Now, we are ready to introduce the sparse finite element approximation space. In particular, we define

$$\hat{\mathbf{V}}_N^k := \bigoplus_{|\mathbf{l}|_1 \leq N} \mathbf{W}_1^k.$$

This definition is motivated by continuous finite element space on sparse grid [21]. Instead of considering the standard piecewise polynomial space  $\mathbf{V}_N^k$ , which has exponential dependence on the dimension  $d$ , we shall use its subspace  $\hat{\mathbf{V}}_N^k$ . This space has good approximation results (see Section 3.1.3 for details) with significantly reduced degrees of freedom. We remark that the key in the construction lies in the choice of condition for  $\mathbf{l}$ . Here we have taken it to be

$|\mathbf{l}|_1 \leq N$ . As demonstrated in [21], other conditions are also possible to realize dimension reduction with aims on optimizing various types of approximation results. We defer the numerical study for the comparison of different spaces to Section 3.1.4.

**Remark 3.1.3** *We mentioned in the previous subsection that there are different ways to construct the increment space  $W_n^k$ . However, the choice won't affect the definition of  $\hat{\mathbf{V}}_N^k$ .*

*This is because*

$$\bigoplus_{|\mathbf{l}|_1 \leq N} \mathbf{W}_1^k = \bigoplus_{|\mathbf{l}|_1 \leq N} \mathbf{V}_1^k = \bigoplus_{|\mathbf{l}|_1 \leq N} \underline{\mathbf{W}}_1^k,$$

for any space  $\underline{\mathbf{W}}_1^k$  constructed from one dimensional increment space  $\underline{W}_n^k$  satisfying  $V_{n-1}^k \oplus \underline{W}_n^k = V_n^k$ .

The next lemma will give a count of dimensions for the space  $\hat{\mathbf{V}}_N^k$ .

**Lemma 3.1.4** *The dimension of  $\hat{\mathbf{V}}_N^k$  is given by*

$$\dim(\hat{\mathbf{V}}_N^k) = (k+1)^d \left\{ \sum_{m=0}^{d-1} \binom{d}{m} \left( (-1)^{d-m} + 2^{N+m-d+1} \sum_{i=0}^{d-m-1} \binom{N}{i} \cdot (-2)^{d-m-1-i} \right) + 1 \right\}.$$

*Suppose there is an upper bound on the dimension  $d \leq d_0$ , then there exist constants  $c_{d_0}, C_{d_0}$  depending only on  $d_0$ , such that*

$$c_{d_0} (k+1)^d 2^N N^{d-1} \leq \dim(\hat{\mathbf{V}}_N^k) \leq C_{d_0} (k+1)^d 2^N N^{d-1}.$$

*Proof:* Due to the distinctiveness of the zero-th level in a mesh, we need to distinguish the zero-th level from other levels in the proof. Therefore, for each multi-index  $\mathbf{l}$ , we define the

set of the dimensions with mesh level 0 as  $\mathbf{l}^0 = \{i | l_i = 0, i = 1, \dots, d\}$  and the number of such dimensions as

$$|\mathbf{l}|_0 := \# \text{ in } \mathbf{l}^0.$$

Then

$$\begin{aligned} \dim(\hat{\mathbf{V}}_N^k) &= \dim\left(\bigoplus_{|\mathbf{l}|_1 \leq N} \mathbf{W}_{\mathbf{l}}^k\right) \\ &= \sum_{m=0}^d \dim\left(\bigoplus_{|\mathbf{l}|_0=m, |\mathbf{l}|_1 \leq N} \mathbf{W}_{\mathbf{l}}^k\right). \end{aligned}$$

We will discuss two cases based on the value of  $|\mathbf{l}|_0$ .

**Case 1.**  $0 \leq |\mathbf{l}|_0 = m \leq d - 1$ , i.e., there are  $m$  dimensions of level zero from the  $d$  dimensions of the multi-index  $\mathbf{l}$ . Clearly,

$$\dim\left(\bigoplus_{|\mathbf{l}|_0=m, |\mathbf{l}|_1 \leq N} \mathbf{W}_{\mathbf{l}}^k\right) = \binom{d}{m} \dim\left(\bigoplus_{\substack{l_t=0, 1 \leq t \leq m; \\ l_t > 0, t > m \\ |\mathbf{l}|_1 \leq N}} \mathbf{W}_{\mathbf{l}}^k\right).$$

Since there are  $d - m$  dimensions that have meshes of level no less than one, we always have  $|\mathbf{l}|_1 \geq d - m$ . Define  $e_m = (0, \dots, 0, 1, \dots, 1)$  to be a vector in  $\mathbb{N}_0^d$  whose first  $m$

dimensions are 0, and the rest  $d - m$  dimensions are 1. Then, by Lemma 3.6 in [21],

$$\begin{aligned}
\dim\left(\bigoplus_{\substack{l_t=0, 1 \leq t \leq m; l_t > 0, t > m \\ |\mathbf{l}_1| \leq N}} \mathbf{W}_1^k\right) &= (k+1)^d \sum_{\substack{l_t=0, 1 \leq t \leq m; l_t > 0, t > m \\ |\mathbf{l}_1| \leq N}} 2^{|\mathbf{l}-e\mathbf{m}|_1} \\
&= (k+1)^d \sum_{i=d-m}^N 2^{i-(d-m)} \sum_{\substack{l_t=0, 1 \leq t \leq m; l_t > 0, t > m \\ |\mathbf{l}_1|=i}} 1 \\
&= (k+1)^d \sum_{i=d-m}^N 2^{i-(d-m)} \binom{i-1}{(d-m)-1} \\
&= (k+1)^d \sum_{i=0}^{N+m-d} 2^i \binom{i+(d-m)-1}{(d-m)-1} \\
&= (k+1)^d \left( (-1)^{d-m} + 2^{N+m-d+1} \sum_{i=0}^{d-m-1} \binom{N}{i} \cdot (-2)^{d-m-1-i} \right).
\end{aligned}$$

**Case 2.**  $|\mathbf{l}_0| = d$  means  $\mathbf{l} = \mathbf{0}$ . Therefore,

$$\dim\left(\bigoplus_{|\mathbf{l}_0|=d, |\mathbf{l}_1| \leq N} \mathbf{W}_1^k\right) = \dim(\mathbf{W}_0^k) = (k+1)^d.$$

Finally, we combine both cases, and arrive at

$$\begin{aligned}
\dim(\hat{\mathbf{V}}_N^k) &= \sum_{m=0}^d \dim\left(\bigoplus_{|\mathbf{l}_0|=m, |\mathbf{l}_1| \leq N} \mathbf{W}_1^k\right) \\
&= (k+1)^d \left\{ \sum_{m=0}^{d-1} \binom{d}{m} \left( (-1)^{d-m} + 2^{N+m-d+1} \sum_{i=0}^{d-m-1} \binom{N}{i} \cdot (-2)^{d-m-1-i} \right) + 1 \right\}.
\end{aligned}$$

We notice that

$$\begin{aligned}
& \sum_{m=0}^{d-1} \binom{d}{m} 2^{N+m-d+1} \sum_{i=0}^{d-m-1} \binom{N}{i} \cdot (-2)^{d-m-1-i} \\
& \leq \sum_{m=1}^{d-1} \binom{d}{m} 2^{N+m-d+1} C_{d_0} N^{d-m-1} \\
& \leq C_{d_0} 2^{N-d+1} N^{d-1} \sum_{m=0}^d \binom{d}{m} 2^m N^{-m} \\
& = C_{d_0} 2^N N^{d-1} \left(1 + \frac{2}{N}\right)^d \leq C_{d_0} 2^N N^{d-1},
\end{aligned}$$

where we use  $C_{d_0}, c_{d_0}$  to denote generic constants that depend only on  $d_0$ , not on  $d, k, N$ .

They may have different values in each occurrence throughout the proof.

Similarly,

$$\begin{aligned}
\dim(\hat{\mathbf{V}}_N^k) & \geq \dim\left(\bigoplus_{|\mathbf{l}|_0=0, |\mathbf{l}|_1 \leq N} \mathbf{W}_1^k\right) = (k+1)^d \left( (-1)^d + 2^{N-d+1} \sum_{i=0}^{d-1} \binom{N}{i} \cdot (-2)^{d-1-i} \right) \\
& \geq (k+1)^d (-1 + c_{d_0} 2^{N-d+1} N^{d-1}) \geq c_{d_0} (k+1)^d 2^N N^{d-1}.
\end{aligned}$$

In summary, there exist constants  $c_{d_0}, C_{d_0}$ , such that

$$c_{d_0} (k+1)^d 2^N N^{d-1} \leq \dim(\hat{\mathbf{V}}_N^k) \leq C_{d_0} (k+1)^d 2^N N^{d-1},$$

and we are done. ■

This lemma implies that, upon mesh refinement, the degrees of freedom for the sparse finite element space will grow in the order of  $h^{-1} |\log_2(h)|^{d-1}$  instead of  $h^{-d}$  for the traditional piecewise polynomial space. This translates into a significant reduction in computational cost when  $d$  is large. However, we still need to verify the approximation properties of the reduced

space, and we will discuss it in the next subsection.

### 3.1.3 Approximation Results

Given any one-dimensional projection operator  $\underline{P}_n^k$  and its increment operator  $\underline{Q}_n^k$ , we can naturally define a projection from  $H^1(\Omega)$  to  $\hat{\mathbf{V}}_N^k$  as,

$$\hat{\mathbf{P}}_N^k := \sum_{|\mathbf{l}|_1 \leq N} \underline{Q}_{l_1, x_1}^k \otimes \cdots \otimes \underline{Q}_{l_d, x_d}^k, \quad (3.5)$$

where  $\underline{Q}_{l_i, x_i}^k$  denotes the operator  $\underline{Q}_{l_i}^k$  in the  $i$ -th dimension. In [71], continuous sparse finite element space is used with streamline diffusion method to solve transport-dominated diffusion problem. The finite element space considered is  $\hat{\mathbf{V}}_N^{c,k} := \hat{\mathbf{V}}_N^k \cap C^0(\Omega)$  for  $k \geq 1$ . The approximation properties for projector defined in (3.5) in  $L^2(\Omega)$ ,  $H^1(\Omega)$ ,  $H^2(\Omega_N)$  norms are obtained when  $\underline{P}_n^k$  is a univariate projector onto  $V_n^k \cap C^0([0, 1])$  satisfying (3.3). Because  $\hat{\mathbf{V}}_N^{c,k}$  is a subset of  $\hat{\mathbf{V}}_N^k$ , we can directly use the results in [71] and obtain an estimate of the projection error for  $\hat{\mathbf{P}}_N^k$ .

We introduce some notations about the seminorm of a function's mixed derivatives. For any set  $I = \{i_1, \dots, i_k\} \subset \{1, \dots, d\}$ , we define  $I^c$  to be the complement set of  $I$  in  $\{1, \dots, d\}$ . For non-negative integers  $\alpha, \beta$  and set  $I$ , we define the seminorm

$$|v|_{H^{\alpha, \beta, I}(\Omega)} := \sum_{0 \leq \alpha_1 \leq \alpha} \cdots \sum_{0 \leq \alpha_k \leq \alpha} \sum_{0 \leq \beta_1 \leq \beta} \cdots \sum_{0 \leq \beta_{d-k} \leq \beta} \left\| \left( \frac{\partial^{\alpha_1}}{\partial x_{i_1}^{\alpha_1}} \cdots \frac{\partial^{\alpha_k}}{\partial x_{i_k}^{\alpha_k}} \right) \left( \frac{\partial^{\beta_1}}{\partial x_{j_1}^{\beta_1}} \cdots \frac{\partial^{\beta_{d-k}}}{\partial x_{j_{d-k}}^{\beta_{d-k}}} \right) v \right\|_{L^2(\Omega)},$$

and

$$|v|_{\mathcal{H}^{t+1}(\Omega)} := \max_{s \in \{0,1\}} \max_{1 \leq k \leq d} \left( \max |v|_{H^{t+1,s,I}(\Omega)} \right).$$

We recall the following results in [71] about the approximation properties in  $L^2(\Omega)$ ,  $H^1(\Omega)$  and  $H^2(\Omega_N)$  norm.

**Theorem 3.1.5** *Let  $\hat{\mathbf{P}}_N^k$  defined in (3.5) be constructed from  $\underline{P}_n^k$  which is a univariate projector onto  $V_n^k \cap C^0([0,1])$  satisfying (3.3), then for  $k \geq 1$ , any  $1 \leq t \leq \min\{p, k\}$ , there exist constant  $\underline{c}_{k,t}$ ,  $\kappa_0(k, t, N)$ ,  $\kappa_1(k, t, N) > 0$ , such that for any  $v \in \mathcal{H}^{p+1}(\Omega)$ ,  $N \geq 1$ ,  $d \geq 2$ , we have*

$$|\hat{\mathbf{P}}_N^k v - v|_{H^s(\Omega_N)} = |\mathbf{P}_N^k v - v|_{H^s(\Omega)} \leq \underline{c}_{k,t} d^{1+s/2} \kappa_s(k, t, N)^{d-1+s} 2^{-N(t+1-s)} |v|_{\mathcal{H}^{t+1}(\Omega)},$$

for  $s = 0, 1$  ( $L^2$  norm and  $H^1$  seminorm), and

$$|\hat{\mathbf{P}}_N^k v - v|_{H^2(\Omega_N)} \leq \begin{cases} \underline{c}_{k,t} \left( d^{3/2} \kappa_1(k, t, N)^d + d^2 \kappa_0(k, t, N)^{d-1} 2^{-N} \right) 2^{-N(t-1)} |v|_{\mathcal{H}^{t+1}(\Omega)}, & k \geq 2 \\ \underline{c}_{k,t} \left( d^{1/2} + d^2 \kappa_0(k, t, N)^{d-1} 2^{-N} \right) |v|_{\mathcal{H}^2(\Omega)}, & k = 1, \end{cases}$$

where

$$\kappa_s(k, t, N) = \begin{cases} \tilde{c}_{k,0,t} (N+1) e^{1/(N+1)} + c_{k,0,0}, & s = 0 \\ 2\tilde{c}_{k,0,t} + c_{k,0,0}, & s = 1, \end{cases}$$

and  $\tilde{c}_{k,0,t}$ ,  $c_{k,0,0}$  are defined in (3.3) and (3.4). Moreover, for  $s = 0$ , if

$$\tilde{c}_{k,0,t} 2^{t+1} / (2^{t+1} - 1) + c_{k,0,0} < 1,$$

there exists a positive constant  $c_{t,k}$ , such that  $\kappa(k, t, s, N) < 1$  for all  $N \geq 1, d \geq 2$  with  $N + 1 \leq c_{t,k}(d - 1)$ .

This theorem implies a convergence rate of  $O(h^{k+1})$  up to the polylogarithmic term  $|\log_2 h|^{d-1}$  in the  $L^2$  norm, which is comparable with the traditional full grid approach when the function  $v$  retains enough smoothness. This estimate serves as the foundation in the error estimates of IP method in the energy norm. We remark that when switching the projection operator (say  $L^2$  projection or other projectors on to  $\hat{\mathbf{V}}_N^k$ ) the error bound may change. However, our estimates in Section 3.2.2 does not require any particular property from the projection. Therefore, it suffices for us to use this theorem directly from [71]. In future work, we will establish convergence property of projections onto  $\hat{\mathbf{V}}_N^k$ , for  $k \geq 0$ . This will be essential for error estimates for broader definitions of DG methods for other types of equations.

### 3.1.4 Approximations from Various Sparse Discontinuous Finite Element Spaces: A Numerical Investigation

In this subsection, we numerically investigate the approximation properties of several sparse finite element spaces, including  $\hat{\mathbf{V}}_N^k$  defined in the previous subsections, and

$$\tilde{\mathbf{V}}_N^k := \bigoplus_{\substack{|\mathbf{l}|_1 \leq N+d-1 \\ |\mathbf{l}|_\infty \leq N}} \mathbf{W}_1^k,$$

which is a slightly larger space.  $\tilde{\mathbf{V}}_N^k$  contains the sparse continuous piecewise linear space in [21] when  $k \geq 1$ . Moreover, by a similar argument as in the proof of Lemma 3.1.4, its degrees of freedom is also of the order  $h^{-1}|\log_2(h)|^{d-1}$ .

The other space we consider here retains not only sparsity with respect to  $h$ , but also with respect to the degree of polynomials (we call this as  $p$ -sparsity). If we introduce

$$\hat{\mathbf{W}}_1^k = \bigoplus_{|\mathbf{k}|_1 \leq k} W_{l_1, x_1}^{k_1} \times W_{l_2, x_2}^{k_2} \cdots \times W_{l_d, x_d}^{k_d},$$

with  $\mathbf{k} = (k_1, \dots, k_d)$ , the new sparse space can be defined as

$$\hat{\mathbf{V}}_N^k := \bigoplus_{|\mathbf{l}|_1 \leq N} \hat{\mathbf{W}}_1^k.$$

It is clear that

$$\hat{\mathbf{V}}_N^k \subset \hat{\mathbf{V}}_N^k \subset \tilde{\mathbf{V}}_N^k.$$

A count of dimensions for the space  $\hat{\mathbf{V}}_N^k$  is given by the following lemma.

**Lemma 3.1.6** *The dimension of  $\hat{\mathbf{V}}_N^k$  is*

$$\dim(\hat{\mathbf{V}}_N^k) = \binom{k+d}{d} \left\{ \sum_{m=0}^{d-1} \binom{d}{m} \left( (-1)^{d-m} + 2^{N+m-d+1} \sum_{i=0}^{d-m-1} \binom{N}{i} \cdot (-2)^{d-m-1-i} \right) + 1 \right\}.$$

Suppose there is an upper bound on the dimension  $d \leq d_0$ , then there exist constants  $c_{d_0}, C_{d_0}$  depending only on  $d_0$ , such that

$$c_{d_0} \binom{k+d}{d} 2^N N^{d-1} \leq \dim(\hat{\mathbf{V}}_N^k) \leq C_{d_0} \binom{k+d}{d} 2^N N^{d-1}.$$

Note that, when  $k$  and  $d$  are large,  $\dim(\hat{\mathbf{V}}_N^k)$  can be significantly smaller than  $\dim(\hat{\mathbf{V}}_N^k)$  and  $\dim(\tilde{\mathbf{V}}_N^k)$ .

As for the implementation of the space  $\hat{\mathbf{V}}_N^k$ , we can no longer use the orthogonal hierarchical bases in Section 3.1.1 to represent functions in  $\hat{\mathbf{V}}_N^k$ , since the vanishing-moment functions  $f_i$ , which are used to construct the orthogonal hierarchical bases, have the same degrees, see Table 3.1. Instead, we can adopt another set of non-orthogonal hierarchical basis functions defined as follows. We first construct the bases for  $W_n^k$ . Again, the case of  $n = 0$  is trivial. Similar to the orthogonal case, the bases for  $W_1^k$  are defined as

$$\hat{h}_i(x) = 2^{1/2} \hat{f}_i(2x - 1), \quad i = 1, \dots, k + 1,$$

where  $\hat{f}_i$ ,  $i = 1, \dots, k + 1$  are functions supported on  $[-1, 1]$ . Rather than choosing  $\hat{f}_i$  from Table 3.1, we let

$$\hat{f}_i(x) = x^{i-1}, \quad x \in (-1, 0)$$

and

$$\hat{f}_i(x) = -x^{i-1}, \quad x \in (0, 1).$$

The bases for  $W_n^k$ ,  $n \geq 1$  can be defined accordingly as

$$\hat{v}_{i,n}^j(x) = 2^{(n-1)/2} \hat{h}_i(2^{n-1}x - j), \quad i = 1, \dots, k + 1, j = 0, \dots, 2^{n-1} - 1.$$

Therefore, the basis functions in  $\hat{\mathbf{W}}_1^k$  (and  $\hat{\mathbf{V}}_N^k$ ) are defined as

$$\hat{v}_{\mathbf{i},1}^{\mathbf{j}}(\mathbf{x}) := \prod_{\substack{t=1 \\ |\mathbf{i}|_1 \leq k}}^d v_{i_t, l_t}^{j_t}(x_t), \quad j_t = 0, \dots, \max(0, 2^{l_t-1} - 1).$$

The purpose of our study in this subsection is to provide a numerical comparison of the

three types of sparse approximation spaces. In particular, we consider a smooth function

$$u(\mathbf{x}) = \exp\left(\prod_{i=1}^d x_i\right), \quad \mathbf{x} \in [0, 1]^d,$$

where  $d = 2, 3$  and the standard  $L^2$  projections:  $\hat{\mathbf{P}}_n^k$ ,  $\tilde{\mathbf{P}}_n^k$ , and  $\hat{\mathbf{P}}_n^k$  from  $H^1(\Omega)$  to  $\hat{\mathbf{V}}_N^k$ ,  $\tilde{\mathbf{V}}_N^k$ , and  $\hat{\mathbf{V}}_N^k$ , respectively. We measure various norms of the projection errors and the degrees of freedom for each space to find the best balance between accuracy and efficiency.

In Tables 3.2-3.7, we report the projection errors  $e_1 = \hat{\mathbf{P}}_n^k u - u$ ,  $e_2 = \tilde{\mathbf{P}}_n^k u - u$  and  $e_3 = \hat{\mathbf{P}}_n^k u - u$  in  $L^1$ ,  $L^2$ ,  $L^\infty$ , and  $H^1$  norms and the associated orders of accuracy for  $k = 1, 2, 3$ . In the implementation, the projections are obtained via the Gaussian quadrature based on sparse grids to save computational cost. When computing the error norms, the domain is uniformly divided into  $2^{dN}$  cell. Note that on each cell, the projection functions are smooth, and hence we adopt the  $6^d$ -point gaussian quadrature with 12-th order accuracy to evaluate the errors in  $L^1$ ,  $L^2$ ,  $L^\infty$ , and  $H^1$  norms.

For all three spaces, the degrees of freedom are all significantly less than  $(k+1)^d 2^{Nd}$ , which is the degree of freedom for the full grid approximation. For space  $\tilde{\mathbf{V}}_N^k$ ,  $(k+1)$ -th order of accuracy is clearly observed for the  $L^1$  and  $L^2$  errors, and  $k$ -th order of accuracy is observed for the  $H^1$  error, while slight reduction of accuracy is observed for the  $L^\infty$  error. For space  $\hat{\mathbf{V}}_N^k$ ,  $k$ -th order of accuracy is observed for the  $H^1$  error and slight reduction of accuracy is observed for  $L^1$  and  $L^2$  errors. However, we do observe about half-order reduction of accuracy for the  $L^\infty$  error. We remark that, on one hand, the magnitude of errors computed by space  $\hat{\mathbf{V}}_N^k$  is larger than that by space  $\tilde{\mathbf{V}}_N^k$  for a fixed  $N$ . The apparent reason is that the degrees of freedom of  $\hat{\mathbf{V}}_N^k$  is smaller than  $\tilde{\mathbf{V}}_N^k$ . On the other hand, when the degrees of freedom are comparable for the two spaces, the magnitude of errors computed by

space  $\hat{\mathbf{V}}_N^k$  is smaller than  $\tilde{\mathbf{V}}_N^k$ . Therefore,  $\hat{\mathbf{V}}_N^k$  is preferred if the computational efficiency is concerned. For  $\hat{\mathbf{V}}_N^k$ , which has the least degrees of freedom among the three sparse spaces, severe reduction of accuracy is observed for all error norms. Moreover, the magnitude of errors is significantly larger than that by space  $\hat{\mathbf{V}}_N^k$  or  $\tilde{\mathbf{V}}_N^k$  with comparable degrees of freedom.

In summary, based on the discussion, we will adopt space  $\hat{\mathbf{V}}_N^k$  in the computation for elliptic equations in the next section. A further reduction in computational cost is possible by finding the optimal subset of  $\hat{\mathbf{V}}_N^k$  by adaptive algorithms. We do not pursue this direction in this thesis, but leave it to future study.

Table 3.2:  $L^2$  projection errors and orders of accuracy. FGDOF denotes the degrees of freedom of the full grid approximation space.  $d = 2$ .  $k = 1$ .

	N		$\dim \tilde{\mathbf{V}}_N^k$	$\dim \hat{\mathbf{V}}_N^k$	$\dim \hat{\hat{\mathbf{V}}}_N^k$	FGDOF			
	2		48	32	24	64			
	3		192	80	60	256			
	4		320	192	144	1024			
	5		768	448	336	4096			
	6		1792	1024	768	16384			
N	$L^1$ error	order	$L^2$ error	order	$L^\infty$ error	order	$H^1$ error	order	
$\tilde{\mathbf{V}}_N^k$									
2	1.58E-03		2.41E-03		2.16E-02		7.45E-02		
3	4.05E-04	1.97	6.14E-04	1.97	6.62E-03	1.71	3.73E-02	1.00	
4	1.03E-04	1.97	1.56E-04	1.98	1.98E-03	1.74	1.86E-02	1.00	
5	2.62E-05	1.98	3.96E-05	1.98	5.81E-04	1.77	9.32E-03	1.00	
6	6.64E-06	1.98	1.00E-05	1.98	1.68E-04	1.79	4.66E-03	1.00	
$\hat{\mathbf{V}}_N^k$									
2	2.28E-03		3.16E-03		3.21E-02		8.10E-02		
3	6.30E-04	1.86	8.98E-04	1.82	1.17E-02	1.46	4.09E-02	0.99	
4	1.72E-04	1.87	2.50E-04	1.85	4.01E-03	1.54	2.05E-02	1.00	
5	4.68E-05	1.88	6.82E-05	1.87	1.31E-03	1.61	1.03E-02	1.00	
6	1.26E-05	1.90	1.84E-05	1.89	4.13E-04	1.67	5.13E-03	1.00	
$\hat{\hat{\mathbf{V}}}_N^k$									
2	1.95E-02		2.59E-02		1.94E-01		3.60E-01		
3	8.47E-03	1.21	1.14E-02	1.18	1.07E-01	0.86	2.72E-01	0.40	
4	3.77E-03	1.17	5.13E-03	1.16	5.69E-02	0.91	2.18E-01	0.32	
5	1.72E-03	1.13	2.34E-03	1.13	2.94E-02	0.95	1.81E-01	0.27	
6	7.94E-04	1.11	1.08E-03	1.11	1.49E-02	0.98	1.55E-01	0.23	

Table 3.3: Projection errors and orders of accuracy. FGDOF denotes the degrees of freedom of the full grid approximation space.  $d = 2$ .  $k = 2$ .

	N		$\dim \tilde{\mathbf{V}}_N^k$	$\dim \hat{\mathbf{V}}_N^k$	$\dim \hat{\hat{\mathbf{V}}}_N^k$	FGDOF			
	2		108	72	48	144			
	3		432	180	120	576			
	4		720	432	288	2304			
	5		1728	1008	672	9216			
	6		4032	2304	1536	36864			
N	$L^1$ error	order	$L^2$ error	order	$L^\infty$ error	order	$H^1$ error	order	
$\tilde{\mathbf{V}}_N^k$									
2	2.82E-05		4.36E-05		4.05E-04		2.26E-03		
3	3.50E-06	3.01	5.47E-06	2.99	5.94E-05	2.77	5.66E-04	2.00	
4	4.39E-07	3.00	6.86E-07	2.99	8.54E-06	2.80	1.42E-04	2.00	
5	5.51E-08	3.00	8.61E-08	3.00	1.21E-06	2.81	3.54E-05	2.00	
6	6.90E-09	3.00	1.08E-08	3.00	1.71E-07	2.83	8.84E-06	2.00	
$\hat{\mathbf{V}}_N^k$									
2	3.51E-05		5.23E-05		6.48E-04		2.41E-03		
3	4.84E-06	2.86	7.26E-06	2.85	1.23E-04	2.40	6.08E-04	1.99	
4	6.56E-07	2.88	9.96E-07	2.87	2.21E-05	2.47	1.53E-04	1.99	
5	8.81E-08	2.90	1.35E-07	2.88	3.77E-06	2.55	3.82E-05	2.00	
6	1.17E-08	2.91	1.81E-08	2.90	6.13E-07	2.62	9.55E-06	2.00	
$\hat{\hat{\mathbf{V}}}_N^k$									
2	1.73E-03		2.51E-03		2.78E-02		5.62E-02		
3	4.86E-04	1.83	7.22E-04	1.80	1.09E-02	1.35	2.80E-02	1.01	
4	1.35E-04	1.85	2.02E-04	1.84	3.99E-03	1.46	1.39E-02	1.01	
5	3.68E-05	1.88	5.53E-05	1.87	1.37E-03	1.55	6.95E-03	1.00	
6	9.90E-06	1.89	1.49E-05	1.89	4.45E-04	1.62	3.47E-03	1.00	

Table 3.4: Projection errors and orders of accuracy. FGDOF denotes the degrees of freedom of the full grid approximation space.  $d = 2$ .  $k = 3$ .

	N		$\dim \tilde{\mathbf{V}}_N^k$	$\dim \hat{\mathbf{V}}_N^k$	$\dim \hat{\hat{\mathbf{V}}}_N^k$	FGDOF			
	2		192	128	80	256			
	3		768	320	200	1024			
	4		1280	768	480	4096			
	5		3072	1792	1120	16384			
	6		7168	4096	2560	65536			
N	$L^1$ error	order	$L^2$ error	order	$L^\infty$ error	order	$H^1$ error	order	
$\tilde{\mathbf{V}}_N^k$									
2	3.72E-07		6.12E-07		4.67E-06		4.66E-05		
3	2.32E-08	4.00	3.84E-08	4.00	3.32E-07	3.82	5.83E-06	3.00	
4	1.45E-09	4.00	2.40E-09	4.00	2.31E-08	3.85	7.29E-07	3.00	
5	9.09E-11	4.00	1.50E-10	4.00	1.59E-09	3.86	9.11E-08	3.00	
6	5.68E-12	4.00	9.39E-12	4.00	1.09E-10	3.86	1.14E-08	3.00	
$\hat{\mathbf{V}}_N^k$									
2	4.37E-07		6.77E-07		7.54E-06		4.82E-05		
3	2.90E-08	3.91	4.55E-08	3.90	7.47E-07	3.34	6.07E-06	2.99	
4	1.92E-09	3.92	3.05E-09	3.90	7.08E-08	3.40	7.61E-07	3.00	
5	1.26E-10	3.93	2.03E-10	3.91	6.33E-09	3.48	9.52E-08	3.00	
6	8.28E-12	3.93	1.35E-11	3.92	5.35E-10	3.57	1.19E-08	3.00	
$\hat{\hat{\mathbf{V}}}_N^k$									
2	1.95E-04		2.88E-04		3.57E-03		8.44E-03		
3	4.20E-05	2.22	6.36E-05	2.18	1.10E-03	1.70	3.16E-03	1.42	
4	9.26E-06	2.18	1.42E-05	2.17	3.19E-04	1.79	1.23E-03	1.35	
5	2.09E-06	2.14	3.19E-06	2.15	8.77E-05	1.86	5.03E-04	1.29	
6	4.81E-07	2.12	7.31E-07	2.13	2.32E-05	1.92	2.12E-04	1.24	

Table 3.5: Projection errors and orders of accuracy. FGDOF denotes the degrees of freedom of the full grid approximation space.  $d = 3$ .  $k = 1$ .

	N		$\dim \tilde{\mathbf{V}}_N^k$	$\dim \hat{\mathbf{V}}_N^k$	$\dim \hat{\hat{\mathbf{V}}}_N^k$	FGDOF			
	2		352	104	52	512			
	3		1216	304	152	4096			
	4		3584	832	416	32768			
	5		9728	2176	1088	262144			
	6		25088	5504	2752	2097152			
N	$L^1$ error	order	$L^2$ error	order	$L^\infty$ error	order	$H^1$ error	order	
$\tilde{\mathbf{V}}_N^k$									
2	6.23E-04		1.20E-03		2.98E-02		3.76E-02		
3	1.57E-04	1.99	3.02E-04	1.99	9.30E-03	1.68	1.87E-02	1.01	
4	3.95E-05	1.99	7.60E-05	1.99	2.91E-03	1.67	9.34E-03	1.00	
5	9.99E-06	1.98	1.91E-05	1.99	9.20E-04	1.66	4.67E-03	1.00	
6	2.52E-06	1.98	4.82E-06	1.99	2.91E-04	1.66	2.33E-03	1.00	
$\hat{\mathbf{V}}_N^k$									
2	1.18E-03		1.91E-03		6.13E-02		4.41E-02		
3	3.65E-04	1.69	5.91E-04	1.69	2.71E-02	1.18	2.24E-02	0.98	
4	1.09E-04	1.74	1.79E-04	1.72	1.13E-02	1.26	1.13E-02	0.99	
5	3.21E-05	1.76	5.34E-05	1.75	4.45E-03	1.34	5.67E-03	1.00	
6	9.28E-06	1.79	1.56E-05	1.77	1.68E-03	1.40	2.84E-03	1.00	
$\hat{\hat{\mathbf{V}}}_N^k$									
2	2.16E-02		3.01E-02		3.76E-01		3.22E-01		
3	1.10E-02	0.97	1.56E-02	0.95	2.80E-01	0.43	2.53E-01	0.35	
4	5.61E-03	0.98	7.99E-03	0.96	1.94E-01	0.53	2.07E-01	0.30	
5	2.83E-03	0.99	4.07E-03	0.97	1.27E-01	0.61	1.74E-01	0.25	
6	1.42E-03	0.99	2.06E-03	0.98	7.97E-02	0.67	1.49E-01	0.22	

Table 3.6: Projection errors and orders of accuracy. FGDOF denotes the degrees of freedom of the full grid approximation space.  $d = 3$ .  $k = 2$ .

	N		$\dim \tilde{\mathbf{V}}_N^k$	$\dim \hat{\mathbf{V}}_N^k$	$\dim \hat{\hat{\mathbf{V}}}_N^k$	FGDOF			
	2		1188	351	130	1728			
	3		4104	1026	380	13824			
	4		12096	2808	1040	110592			
	5		32832	7344	2720	884736			
	6		84672	18576	6880	7077888			
N	$L^1$ error	order	$L^2$ error	order	$L^\infty$ error	order	$H^1$ error	order	
$\tilde{\mathbf{V}}_N^k$									
2	8.65E-06		1.88E-05		5.64E-04		9.83E-04		
3	1.08E-06	3.00	2.36E-06	3.00	8.11E-05	2.80	2.45E-04	2.00	
4	1.35E-07	3.00	2.95E-07	3.00	1.15E-05	2.82	6.12E-05	2.00	
5	1.69E-08	3.00	3.69E-08	3.00	1.65E-06	2.80	1.53E-05	2.00	
6	2.12E-09	3.00	4.62E-09	3.00	2.40E-07	2.78	3.83E-06	2.00	
$\hat{\mathbf{V}}_N^k$									
2	1.41E-05		2.58E-05		1.33E-03		1.10E-03		
3	2.12E-06	2.73	3.86E-06	2.74	3.16E-04	2.08	2.80E-04	1.98	
4	3.15E-07	2.75	5.76E-07	2.74	7.07E-05	2.16	7.07E-05	1.99	
5	4.62E-08	2.77	8.56E-08	2.75	1.50E-05	2.24	1.77E-05	1.99	
6	6.66E-09	2.79	1.26E-08	2.76	3.01E-06	2.31	4.44E-06	2.00	
$\hat{\hat{\mathbf{V}}}_N^k$									
2	4.50E-03		6.59E-03		1.13E-01		9.32E-02		
3	1.82E-03	1.30	2.71E-03	1.28	7.36E-02	0.62	5.93E-02	0.65	
4	7.51E-04	1.28	1.11E-03	1.29	4.30E-02	0.78	3.91E-02	0.60	
5	3.10E-04	1.28	4.53E-04	1.29	2.32E-02	0.89	2.70E-02	0.54	
6	1.30E-04	1.25	1.88E-04	1.26	1.18E-02	0.97	1.95E-02	0.47	

Table 3.7: Projection errors and orders of accuracy. FGDOF denotes the degrees of freedom of the full grid approximation space.  $d = 3$ .  $k = 3$ .

	N		$\dim \tilde{\mathbf{V}}_N^k$	$\dim \hat{\mathbf{V}}_N^k$	$\dim \hat{\hat{\mathbf{V}}}_N^k$	FGDOF		
	2		2816	832	260	4096		
	3		9728	2432	760	32768		
	4		28672	6656	2080	262144		
	5		77824	17408	5440	2097152		
	6		200704	44032	13760	16777216		
N	$L^1$ error	order	$L^2$ error	order	$L^\infty$ error	order	$H_1$ error	order
$\tilde{\mathbf{V}}_N^k$								
2	9.39E-08		2.37E-07		6.61E-06		1.80E-085	
3	5.96E-09	3.98	1.49E-08	3.99	5.33E-07	3.63	2.25E-06	3.00
4	3.66E-10	4.03	9.27E-10	4.01	3.13E-08	4.09	2.81E-07	3.00
5	2.30E-11	3.99	5.81E-11	4.00	2.47E-09	3.66	3.52E-08	3.00
6	2.09E-12	3.46	5.26E-12	3.46	6.07E-10	2.03	4.41E-09	2.99
$\hat{\mathbf{V}}_N^k$								
2	1.32E-07		2.82E-07		1.55E-05		1.92E-05	
3	9.66E-09	3.78	2.01E-08	3.81	2.01E-06	2.95	2.43E-06	2.98
4	6.94E-10	3.80	1.44E-09	3.80	2.35E-07	3.10	3.05E-07	2.99
5	4.99E-11	3.80	1.04E-10	3.78	2.66E-08	3.15	3.83E-08	3.00
6	3.92E-12	3.67	8.49E-12	3.62	3.13E-09	3.08	4.80E-09	2.99
$\hat{\hat{\mathbf{V}}}_N^k$								
2	7.49E-04		1.18E-03		2.52E-02		2.24E-02	
3	2.38E-04	1.65	3.86E-04	1.61	1.48E-02	0.77	1.11E-02	1.02
4	7.31E-05	1.70	1.22E-04	1.67	7.48E-03	0.98	5.46E-03	1.02
5	2.18E-05	1.75	3.71E-05	1.71	3.41E-03	1.13	2.69E-03	1.02
6	6.35E-06	1.78	1.10E-05	1.75	1.43E-03	1.25	1.76E-03	0.62

## 3.2 IP Method on Sparse Grids for Elliptic Equations

In this section, we solve the second order linear elliptic boundary value problems (3.1) by IP methods with the sparse finite element space  $\hat{\mathbf{V}}_N^k$ . We will first formulate the scheme and discuss its implementation issues, and then perform error analysis in the energy norm.

### 3.2.1 Formulation of the Scheme

First, we will introduce some basic notations about jumps and averages for piecewise functions defined on a grid  $\Omega_N$ . Let  $\Gamma := \bigcup_{T \in \Omega_N} \partial T$  be the union of the boundaries for all the elements in  $\Omega_N$  and  $S(\Gamma) := \Pi_{T \in \Omega_N} L^2(\partial T)$  be the set of  $L^2$  functions defined on  $\Gamma$ . For any  $q \in S(\Gamma)$  and  $\mathbf{q} \in [S(\Gamma)]^d$ , we define their averages  $\{q\}, \{\mathbf{q}\}$  and jumps  $[q], [\mathbf{q}]$  on the interior edges as follows. Suppose  $e$  is an interior edge shared by elements  $T_+$  and  $T_-$ , we define the unit normal vectors  $\mathbf{n}^+$  and  $\mathbf{n}^-$  on  $e$  pointing exterior to  $T_+$  and  $T_-$ , then

$$\begin{aligned} [q] &= q^- \mathbf{n}^- + q^+ \mathbf{n}^+, & \{q\} &= \frac{1}{2}(q^- + q^+), \\ [\mathbf{q}] &= \mathbf{q}^- \cdot \mathbf{n}^- + \mathbf{q}^+ \cdot \mathbf{n}^+, & \{\mathbf{q}\} &= \frac{1}{2}(\mathbf{q}^- + \mathbf{q}^+). \end{aligned}$$

If  $e$  is a boundary edge, then we let

$$[q] = q\mathbf{n}, \quad \{\mathbf{q}\} = \mathbf{q},$$

where  $\mathbf{n}$  is the outward unit normal.

Now we are ready to formulate the IPDG scheme for (3.1). We look for  $u_h \in \hat{\mathbf{V}}_N^k$ , such that

$$B(u_h, v) = L(v), \quad \forall v \in \hat{\mathbf{V}}_N^k \tag{3.6}$$

where

$$\begin{aligned}
B(w, v) &= \int_{\Omega} \mathbf{K} \nabla w \cdot \nabla v \, d\mathbf{x} - \sum_{e \in \Gamma} \int_e \{\mathbf{K} \nabla w\} \cdot [v] \, ds \\
&\quad - \sum_{e \in \Gamma} \int_e \{\mathbf{K} \nabla v\} \cdot [w] \, ds + \sum_{e \in \Gamma} \frac{\sigma}{h} \int_e [w] \cdot [v] \, ds,
\end{aligned} \tag{3.7}$$

and

$$L(v) = \int_{\Omega} f v \, d\mathbf{x} - \int_{\partial\Omega} \left( \mathbf{K} \nabla v \cdot \mathbf{n} + \frac{\sigma}{h} v \right) g \, ds, \tag{3.8}$$

where  $\sigma$  is a positive penalty parameter,  $h = 2^{-N}$  is the uniform mesh size in each dimension. The IP methods developed in [82, 6] are commonly used to solve elliptic problems, but usually with standard piecewise polynomial space. Here, by using the sparse finite element space, we can achieve a significant reduction in the size of the linear algebraic system (3.6) especially when the dimension is large.

We notice that there are still several important factors in ensuring real computational gains in implementation. First of all, it's obvious that we need efficient algorithms to evaluate  $L(v)$  when  $v$  is taken to be the basis functions  $v_{\mathbf{i}, \mathbf{l}}^{\mathbf{j}}(\mathbf{x})$  in  $\hat{\mathbf{V}}_N^k$ . A standard integration scheme will incur computational complexity with exponential dependence on  $dN$ . To avoid this, in our algorithm, we take full advantage of numerical integration on sparse grids developed in the literature [77, 40]. In particular, for each basis  $v_{\mathbf{i}, \mathbf{l}}^{\mathbf{j}}(\mathbf{x})$ , its domain of dependence consists of  $2^d$  smooth patches. Therefore, when evaluating  $\int_{\Omega} f v_{\mathbf{i}, \mathbf{l}}^{\mathbf{j}} \, d\mathbf{x}$ , we first divide this integral into  $2^d$  parts accordingly, and then for each patch, we implement a sparse grid integration with Gauss quadrature, and the number of levels employed in this calculation is taken as  $(I_{quad} - |\mathbf{l}|_1/2)$ , where  $I_{quad}$  is a fixed integer chosen large enough to guarantee sufficient

accuracy. The evaluation of  $\int_{\partial\Omega} (\mathbf{K}\nabla v \cdot \mathbf{n} + \frac{\sigma}{h}v) g ds$  can be performed in a similar fashion.

Another important factor when we assemble the linear system is the so-called *unidirectional principle* [19]. To demonstrate the main ideas, we can see that evaluating  $\int_{\Omega} \phi(\mathbf{x}) d\mathbf{x}$  with  $\phi(\mathbf{x}) = \phi_1(x_1) \dots \phi_d(x_d)$  is equivalent to multiplication of one-dimensional integrals  $\int_{[0,1]} \phi_1 dx_1 \dots \int_{[0,1]} \phi_d dx_d$ . Therefore, computing  $\int_{\Omega} f v_{\mathbf{i},1}^{\mathbf{j}} d\mathbf{x}$  when  $f(\mathbf{x})$  is separable, i.e.  $f(\mathbf{x}) = f_1(x_1) \dots f_d(x_d)$  or when  $f(\mathbf{x})$  is a sum of separable functions is straightforward, because we only need some small overhead to compute 1D integrals and assemble them to obtain the multi-dimensional integrations.

The same discussion holds true for the computation of the bilinear term  $B(u_h, v)$ . For example, if we use a direct method, we need to evaluate  $B(w, v)$  for  $(w, v)$  being all possible basis functions in  $\hat{\mathbf{V}}_N^k$ . From the definition of  $B(w, v)$ , each matrix element will involve four multi-dimensional integrations. If  $\mathbf{K}$  is separable (in particular, when  $\mathbf{K}$  is a constant function), due to the unidirectional principle, the matrices can be assembled fast. When  $\mathbf{K}$  is a general function, we need to compute true high dimensional integrals. This difficulty is identified as one of the main challenging tasks for computing PDEs on sparse grids, see e.g. [20, 2]. In this case, we can either use the sparse grid integration procedure mentioned above or by a computational procedure outlined as follows. Assume  $\mathbf{K}$  to be a smooth function, then we can find  $\mathbf{K}_h = \hat{P}_N^{2k} \mathbf{K}$  as the  $L^2$  projection of  $\mathbf{K}$  onto the sparse finite element space  $\hat{\mathbf{V}}_N^{2k}$ , and use  $\mathbf{K}_h$  in place of  $\mathbf{K}$  in the scheme. Notice that  $\mathbf{K}_h$  is a sum of separable functions, therefore the computation of the bilinear term is accelerated as the unidirectional case. The reason we use a higher order sparse finite element for projection of  $\mathbf{K}_h$  is to obtain exact evaluation of the volume integral  $\int_{\Omega} \mathbf{K}\nabla u \cdot \nabla v d\mathbf{x}$ . However, this process does change the values of three other terms up to approximation error of  $\mathbf{K}_h - \mathbf{K}$ . Another thing we did not explore is the efficient solver for the linear algebraic system. In the literature, iterative

methods have been proposed based on the semi-coarsening approach and its sparse grid extensions [64, 65, 43, 48, 47, 69, 18]. We leave those interesting implementation aspects to future study.

### 3.2.2 Error Analysis

This section contains error estimates of the IPDG method on sparse grids. Following [6], we define the energy norm of a function  $v \in H^2(\Omega_N)$  by

$$|||v|||^2 := \sum_{T \in \Omega_N} \int_T |\nabla v|^2 dx + \sum_{e \in \Gamma} h \int_e \left\{ \frac{\partial v}{\partial \mathbf{n}} \right\}^2 ds + \sum_{e \in \Gamma} \frac{1}{h} \int_e [v]^2 ds.$$

We review some basic properties of the bilinear operator  $B(\cdot, \cdot)$ .

**Lemma 3.2.1 (Orthogonality)** *Let  $u$  be the exact solution to (3.1), and  $u_h$  be the numerical solution to (3.6), then*

$$B(u - u_h, v) = 0, \quad \forall v \in \hat{\mathbf{V}}_N^k.$$

*Proof:* Using integration by parts, we can easily show  $B(u, v) = 0, \forall v \in \hat{\mathbf{V}}_N^k$ . The Galerkin orthogonality immediately follows. ■

**Lemma 3.2.2 (Boundedness[6, 7])** *There exists a positive constant  $C_b$ , depending only on  $K_1, \sigma$ , such that*

$$B(w, v) \leq C_b |||w||| \cdot |||v|||, \quad \forall w, v \in H^2(\Omega_N).$$

**Lemma 3.2.3 (Stability[6, 7])** *When  $\sigma$  is taken large enough, there exists a positive constant  $C_s$  depending only on  $K_0$ , such that*

$$B(v, v) \geq C_s |||v|||^2, \quad \forall v \in \hat{\mathbf{V}}_N^k.$$

**Theorem 3.2.4** *Let  $u$  be the exact solution to (3.1), and  $u_h$  be the numerical solution to (3.6). For  $k \geq 1$ ,  $u \in \mathcal{H}^{p+1}(\Omega)$ ,  $1 \leq t \leq \min\{p, k\}$ ,  $N \geq 1$ ,  $d \geq 2$ , we have*

$$|||u - u_h||| \leq \left(1 + \frac{C_b}{C_s}\right) \sqrt{C_d \underline{c}_{k,t} C^*} 2^{-Nt} |u|_{\mathcal{H}^{t+1}(\Omega)}.$$

where  $C_d$  is a constant that depends on  $d$  linearly.  $C_b, C_s$  are defined in Lemmas 3.2.2 and 3.2.3.

$$C^* = \max \left( \sqrt{d^2 \kappa_0^{2d-2} + 3d^3 \kappa_1^{2d} + 2d^4 \kappa_0^{2d-2} 2^{-2N}}, \right. \\ \left. \sqrt{d^2 \kappa_0^{2d-2} + d^3 \kappa_1^{2d} + 2d + 2d^4 \kappa_0^{2d-2} 2^{-2N}} \right),$$

where  $\kappa_s = \kappa_s(k, t, N)$ ,  $s = 0, 1$  and  $\underline{c}_{k,t}$  are defined in Theorem 3.1.5.

*Proof:* Choose any function  $u_I \in \hat{\mathbf{V}}_N^k$ , then we decompose the error into  $e = u - u_h = (u - u_I) + (u_I - u_h)$ . By Céa's lemma, using Lemma 3.2.1, 3.2.2 and 3.2.3,

$$C_s |||u_I - u_h|||^2 \leq B_h(u_I - u_h, u_I - u_h) = B_h(u_I - u, u_I - u_h) \leq C_b |||u - u_I||| \cdot |||u_I - u_h|||.$$

Therefore,  $|||u_I - u_h||| \leq \frac{C_b}{C_s} |||u - u_I|||$ , and

$$|||e||| \leq |||u_I - u_h||| + |||u - u_I||| \leq \left(1 + \frac{C_b}{C_s}\right) |||u - u_I|||.$$

We have

$$\|e\| \leq \left(1 + \frac{C_b}{C_s}\right) \inf_{u_I \in \hat{\mathbf{V}}_N^k} \|u - u_I\| \leq \left(1 + \frac{C_b}{C_s}\right) \|u - \hat{\mathbf{P}}_N^k u\|,$$

where the projection operator  $\hat{\mathbf{P}}_N^k$  has been specified in Theorem 3.1.5.

Next, we need to bound the energy norm of  $\eta := u - \hat{\mathbf{P}}_N^k u$ . Recall that

$$\|\eta\|^2 = \sum_{T \in \Omega_N} \int_T |\nabla \eta|^2 d\mathbf{x} + \sum_{e \in \Gamma} h \int_e \left\{ \frac{\partial \eta}{\partial \mathbf{n}} \right\}^2 ds + \sum_{e \in \Gamma} \frac{1}{h} \int_e [\eta]^2 ds.$$

The first term in the summation is

$$\sum_{T \in \Omega_N} \int_T |\nabla \eta|^2 d\mathbf{x} = |\eta|_{H^1(\Omega_N)}^2.$$

To bound the second and third terms, we use the trace inequalities [6]:

$$\begin{aligned} \|\phi\|_{L^2(\partial T)}^2 &\leq C_d \left( \frac{1}{h} \|\phi\|_{L^2(T)}^2 + h \|\phi\|_{H^1(T)}^2 \right), \quad \forall \phi \in H^1(T) \\ \left\| \frac{\partial \phi}{\partial \mathbf{n}} \right\|_{L^2(\partial T)}^2 &\leq C_d \left( \frac{1}{h} \|\phi\|_{H^1(T)}^2 + h \|\phi\|_{H^2(T)}^2 \right), \quad \forall \phi \in H^2(T) \end{aligned}$$

where  $C_d$  is a generic constant that depends on  $d$  linearly. It may have different values in each occurrence in the proof. Sum over all the elements, we get

$$\begin{aligned} \sum_{e \in \Gamma} h \int_e \left\{ \frac{\partial \eta}{\partial \mathbf{n}} \right\}^2 ds &\leq C_d \left( |\eta|_{H^1(\Omega_N)}^2 + h^2 |\eta|_{H^2(\Omega_N)}^2 \right) \\ \sum_{e \in \Gamma} \frac{1}{h} \int_e [\eta]^2 ds &\leq C_d \left( \frac{1}{h^2} \|\eta\|_{L^2(\Omega_N)}^2 + |\eta|_{H^1(\Omega_N)}^2 \right). \end{aligned}$$

In summary,

$$|||\eta|||^2 \leq C_d \left( \frac{1}{h^2} \|\eta\|_{L^2(\Omega_N)}^2 + |\eta|_{H^1(\Omega_N)}^2 + h^2 |\eta|_{H^2(\Omega_N)}^2 \right).$$

By Theorem 3.1.5,

$$\begin{aligned} |||\eta|||^2 &\leq C_d \underline{c}_{k,t}^2 \left( d^2 \kappa_0^{2d-2} + 3d^3 \kappa_1^{2d} + 2d^4 \kappa_0^{2d-2} 2^{-2N} \right) 2^{-2Nt} |u|_{\mathcal{H}^{t+1}(\Omega)}^2, & k \geq 2 \\ |||\eta|||^2 &\leq C_d \underline{c}_{k,t}^2 \left( d^2 \kappa_0^{2d-2} + d^3 \kappa_1^{2d} + 2d + 2d^4 \kappa_0^{2d-2} 2^{-2N} \right) 2^{-2Nt} |u|_{\mathcal{H}^{t+1}(\Omega)}^2, & k = 1 \end{aligned}$$

where we have used the shorthand notation  $\kappa_s = \kappa_s(k, t, N)$ ,  $s = 0, 1$ . Let's define

$$C^* = \max \left( \sqrt{d^2 \kappa_0^{2d-2} + 3d^3 \kappa_1^{2d} + 2d^4 \kappa_0^{2d-2} 2^{-2N}}, \sqrt{d^2 \kappa_0^{2d-2} + d^3 \kappa_1^{2d} + 2d + 2d^4 \kappa_0^{2d-2} 2^{-2N}} \right).$$

Then,

$$|||\eta||| \leq \sqrt{C_d \underline{c}_{k,t}} C^* 2^{-Nt} |u|_{\mathcal{H}^{t+1}(\Omega)}.$$

Therefore, we have proved the error estimate in the energy norm

$$|||e||| \leq \left( 1 + \frac{C_b}{C_s} \right) \sqrt{C_d \underline{c}_{k,t}} C^* 2^{-Nt} |u|_{\mathcal{H}^{t+1}(\Omega)}.$$

■

This theorem implies a convergence rate of  $O(h^k)$  up to the polylogarithmic term  $|\log_2 h|^{d-1}$  in the energy norm when  $u$  is smooth enough. However, we do require more regularity of  $u$  compared with IP methods using traditional piecewise polynomial space.

**Remark 3.2.5** *Proving convergence in  $L^2$  norm with the standard duality argument will encounter some difficulties in this framework. For example, let  $\varphi$  be the solution to the adjoint problem*

$$-\nabla \cdot (\mathbf{K}\nabla\varphi) = u - u_h \quad \text{on } \Omega, \quad \varphi = 0 \quad \text{on } \partial\Omega.$$

*Since  $\Omega$  is convex, we have  $\varphi \in H^2(\Omega)$ , and  $\|\varphi\|_{H^2(\Omega)} \leq C\|u - u_h\|_{L^2(\Omega)}$ . From the adjoint consistency of the IP method, we get*

$$\|u - u_h\|_{L^2(\Omega)}^2 = B(u - u_h, \varphi).$$

*However, to proceed from here, we will need to define an interpolant of  $\varphi$ :  $\varphi_I \in \hat{\mathbf{V}}_N^1$ , and bound  $\|\varphi - \varphi_I\|$ . From our previous discussion, this will require a bound in  $\|\varphi\|_{\mathcal{H}^2(\Omega)}$ . This is a stronger norm than the classical  $H^2$  norm, and cannot be bounded by  $\|u - u_h\|_{L^2(\Omega)}^2$ .*

### 3.3 Numerical Results

In this section, we provide multi-dimensional numerical results to demonstrate the performance of our sparse grid IPDG scheme.

#### 3.3.1 Two-dimensional Results

In this subsection, we gather computational results for two-dimensional cases. The penalty constant in this subsection is chosen to be  $\sigma = 10$  for  $k = 1$ , and  $\sigma = 20$  for  $k = 2$ .

**Example 3.3.1** We solve the following two-dimensional problem with constant coefficient

Table 3.8: Numerical errors and orders of accuracy for Example 3.3.1 computed by space  $\hat{\mathbf{V}}_N^k$  with  $k = 1, 2$  and indicated  $N$ .

N	$L^1$ error	order	$L^2$ error	order	$L^\infty$ error	order	$H^1$ error	order
$k = 1$								
3	4.49E-03		6.97E-03		3.26E-02		1.77E-01	
4	1.18E-03	1.93	1.93E-03	1.85	9.71E-03	1.75	8.80E-02	1.01
5	3.03E-04	1.96	5.09E-04	1.92	3.19E-03	1.60	4.36E-02	1.01
6	7.68E-05	1.98	1.32E-04	1.98	9.68E-04	1.72	2.16E-02	1.01
$k = 2$								
3	9.52E-05		1.33E-04		5.74E-04		7.61E-03	
4	1.42E-05	2.75	2.03E-05	2.71	9.65E-05	2.57	1.91E-03	1.99
5	2.05E-06	2.79	3.02E-06	2.75	1.59E-05	2.60	4.78E-04	2.00
6	2.89E-07	2.83	4.36E-07	2.79	2.66E-06	2.58	1.19E-04	2.00

on  $\Omega = [0, 1]^2$ .

$$\begin{cases} -\Delta u = 0, & \mathbf{x} \in \Omega, \\ u = \sin(\pi x_1) \frac{\sinh(\pi x_2)}{\sinh(\pi)}, & \mathbf{x} \in \partial\Omega, \end{cases} \quad (3.9)$$

where the exact solution is  $u = \sin(\pi x_1) \frac{\sinh(\pi x_2)}{\sinh(\pi)}$ . We test scheme with  $k = 1$  and  $k = 2$  on different levels of meshes. Numerical errors and orders of accuracy measured in  $L^1, L^2, L^\infty$  and  $H^1$  norms are listed in Table 3.8. We observe  $k$ -th order of accuracy for  $H^1$  norm, close to  $(k+1)$ -th order accuracy for  $L^1, L^2$  norms and slightly less than  $(k+1)$ -th order accuracy for the  $L^\infty$  norm. The results agree with the error estimates performed in the previous section.

In addition, we provide the sparsity patterns of the stiffness matrices for  $k = 1$  and  $k = 2$  in Figure 3.1 and Table 3.9. From Table 3.9, the number of nonzero elements scales like  $O(SGDOF^{1.5})$ , where  $SGDOF$  is the degree of freedom of the space used. This is a denser matrix than the one generated from traditional piecewise polynomial space. However, this is natural considering that the basis functions in  $\hat{\mathbf{V}}_N^k$  are no longer locally defined.

Table 3.9: Sparsity and condition number of the stiffness matrix. Example 3.3.1 computed by the space  $\hat{\mathbf{V}}_N^k$  with  $k = 1, 2$ . SGDOF is the degree of freedom used for the sparse grid DG scheme. NNZ is the number of nonzero elements in the stiffness matrix. Order= $\log(\text{NNZ})/\log(\text{SGDOF})$ .

N	SGDOF	NNZ	Order	Condition Number
$k = 1$				
3	80	992	1.57	3.58E+02
4	192	3216	1.54	1.43E+03
5	448	9168	1.49	5.68E+03
6	1024	24144	1.45	2.26E+04
$k = 2$				
3	180	3456	1.57	1.40E+03
4	432	11124	1.54	5.49E+03
5	1008	31596	1.50	2.16E+04
6	2304	83028	1.46	8.58E+04

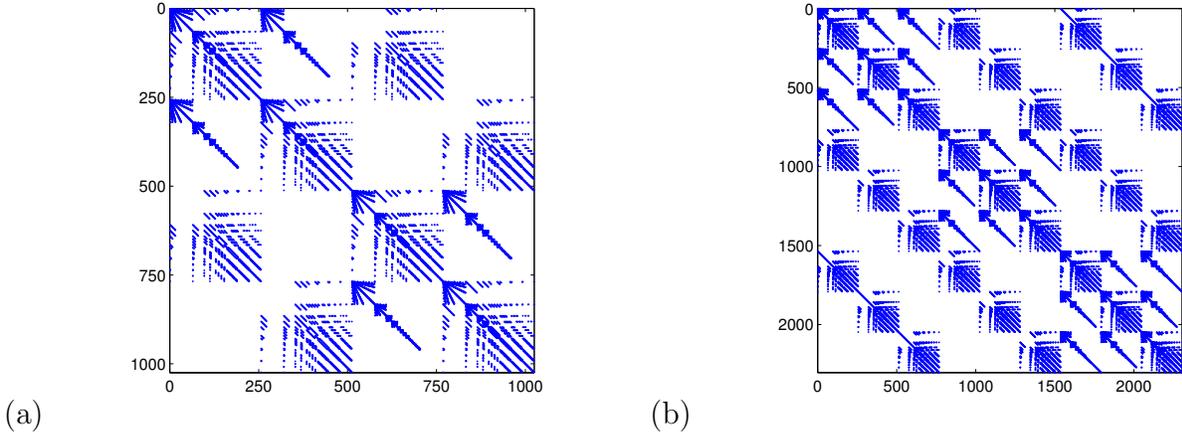


Figure 3.1: Example 3.3.1. The sparsity patterns of the matrices computed by the space  $\hat{\mathbf{V}}_6^k$  with  $k = 1, 2$  and  $N = 6$ . Each dot represents a non-zero element in the stiffness matrix. (a):  $k = 1$ , (b):  $k = 2$ .

**Example 3.3.2** We solve the following two-dimensional problem with smooth variable coefficient on  $\Omega = [0, 1]^2$ .

$$\begin{cases} -\nabla \cdot ((\sin(x_1 x_2) + 1) \nabla u) = f, & \mathbf{x} \in \Omega, \\ u = 0, & \mathbf{x} \in \partial\Omega. \end{cases} \quad (3.10)$$

Table 3.10: Numerical errors and orders of accuracy for Example 3.3.2 computed by the space  $\hat{\mathbf{V}}_N^k$  with  $k = 1, 2$  and indicated  $N$ .

N	$L^1$ error	order	$L^2$ error	order	$L^\infty$ error	order	$H^1$ error	order
$k = 1$								
3	1.30E-02		1.65E-03		4.50E-02		3.37E-01	
4	3.18E-03	2.03	4.08E-03	2.01	1.34E-02	1.74	1.66E-01	1.02
5	7.81E-04	2.02	1.01E-03	2.01	4.43E-03	1.60	8.26E-02	1.01
6	1.94E-04	2.01	2.55E-04	1.99	1.48E-03	1.58	4.11E-02	1.00
$k = 2$								
3	1.77E-04		2.17E-04		5.74E-04		1.35E-02	
4	2.71E-05	2.70	3.37E-05	2.69	1.01E-04	2.51	3.37E-03	2.00
5	3.99E-06	2.76	5.08E-06	2.73	1.91E-05	2.40	8.41E-04	2.00
6	5.67E-07	2.82	7.37E-07	2.78	2.99E-06	2.67	2.10E-04	2.00

$f$  is a given function such that the exact solution is  $u = \sin(\pi x_1) \sin(\pi x_2)$ . The numerical results are provided in Table 3.10. The conclusion is very similar to Example 3.3.1.

**Example 3.3.3** We solve the following two-dimensional problem with discontinuous coefficient on  $\Omega = [0, 1]^2$ .

$$\begin{cases} -\nabla \cdot ((\text{sign}((x_1 - 0.5)(x_2 - 0.5)) + 2) \nabla u) = f, & \mathbf{x} \in \Omega, \\ u = 0, & \mathbf{x} \in \partial\Omega. \end{cases} \quad (3.11)$$

$f$  is a given function such that the exact solution is  $u = \sin(\pi x_1) \sin(\pi x_2)$ . We provide the numerical results with  $k = 1$  and  $k = 2$  in Table 3.11. For this problem with smooth solution but discontinuous coefficient, the convergence rates are maintained.

### 3.3.2 Three-dimensional Results

In this subsection, we gather computational results for three-dimensional elliptic equations. The penalty constant in this subsection is chosen to be  $\sigma = 15$  for  $k = 1$ , and  $\sigma = 30$  for  $k = 2$ .

Table 3.11: Numerical errors and orders of accuracy for Example 3.3.3 computed by the space  $\hat{\mathbf{V}}_N^k$  with  $k = 1, 2$  and indicated  $N$ .

N	$L^1$ error	order	$L^2$ error	order	$L^\infty$ error	order	$H^1$ error	order
$k = 1$								
3	1.24E-02		1.57E-02		4.55E-02		3.33E-01	
4	3.07E-03	2.02	3.94E-03	2.00	1.36E-03	1.75	1.66E-02	1.00
5	7.58E-04	2.02	9.78E-04	2.01	4.50E-03	1.59	8.32E-02	1.00
6	1.89E-04	2.00	2.46E-04	1.99	1.50E-03	1.58	4.16E-02	1.00
$k = 2$								
3	1.96E-04		2.59E-04		1.21E-03		1.56E-02	
4	2.72E-05	2.85	3.50E-05	2.89	1.55E-04	2.96	3.70E-03	2.08
5	3.85E-06	2.82	4.94E-06	2.82	2.05E-05	2.92	8.93E-04	2.05
6	5.36E-07	2.84	7.02E-07	2.82	3.01E-06	2.82	2.19E-04	2.03

**Example 3.3.4** We solve the following three-dimensional problem with constant coefficient on  $\Omega = [0, 1]^3$ .

$$\begin{cases} -\Delta u = 0, & \mathbf{x} \in \Omega, \\ u = \sin(\pi x_1) \sin(\pi x_2) \frac{\sinh(\pi x_3)}{\sinh(\pi)}, & \mathbf{x} \in \partial\Omega, \end{cases} \quad (3.12)$$

where the exact solution is  $u = \sin(\pi x_1) \sin(\pi x_2) \frac{\sinh(\pi x_3)}{\sinh(\pi)}$ . We provide the numerical results with  $k = 1$  and  $k = 2$  in Table 3.12. For this three-dimensional example, we obtain  $k$ -th order for the  $H^1$  norm, and close to  $(k + 1)$ -th order for the  $L^1$  and  $L^2$  norm. However,  $L^\infty$  error seems to degrade to  $(k + \frac{1}{2})$ -th order. However, upon closer examination, similar behaviors have been observed in Section 3.1.4 for the  $L^2$  projection error of a smooth function onto  $\hat{\mathbf{V}}_N^k$ . The sparsity patterns of the stiffness matrices for  $k = 1$  and  $k = 2$  are reported in Figure 3.2. From Table 3.13, we can see the stiffness matrices scale less than the two-dimensional examples, i.e. the number of nonzero elements scales like  $O(SGDOF^{1.4})$ , where  $SGDOF$  is the degree of freedom of the space used.

Table 3.12: Numerical errors and orders of accuracy for Example 3.3.4 computed by the space  $\hat{\mathbf{V}}_N^k$  with  $k = 1, 2$  and indicated  $N$ .

N	$L^1$ error	order	$L^2$ error	order	$L^\infty$ error	order	$H^1$ error	order
$k = 1$								
3	1.29E-02		2.19E-02		1.09E-01		2.85E-01	
4	4.05E-03	1.67	6.98E-03	1.65	4.75E-02	1.20	1.44E-01	0.98
5	1.07E-03	1.92	1.94E-03	1.85	2.34E-02	1.02	7.02E-02	1.04
6	2.76E-04	1.96	5.22E-04	1.89	8.44E-04	1.47	3.39E-02	1.05
$k = 2$								
3	1.41E-04		2.06E-04		1.26E-03		1.05E-02	
4	2.51E-05	2.49	3.80E-05	2.44	3.35E-04	1.91	2.72E-02	1.95
5	4.18E-06	2.59	6.49E-06	2.55	6.51E-05	2.36	6.87E-04	1.98
6	6.69E-07	2.64	1.06E-06	2.62	1.09E-05	2.58	1.72E-04	2.00

Table 3.13: Sparsity and condition number of the stiffness matrix. Example 3.3.4 computed by the space  $\hat{\mathbf{V}}_N^k$  with  $k = 1, 2$ . SGDOF is the degree of freedom used for the sparse grid DG scheme. NNZ is the number of nonzero elements in the stiffness matrix. Order= $\log(\text{NNZ})/\log(\text{SGDOF})$ .

N	SGDOF	NNZ	Order	Condition Number
$k = 1$				
3	304	3760	1.43	3.73E+02
4	832	14080	1.42	1.51E+03
5	2176	45760	1.39	5.97E+03
6	5504	135872	1.37	2.36E+04
$k = 2$				
3	1026	20250	1.43	1.58E+03
4	2808	74628	1.41	5.98E+03
5	7344	240516	1.39	2.32E+04
6	18576	710532	1.37	9.15E+04

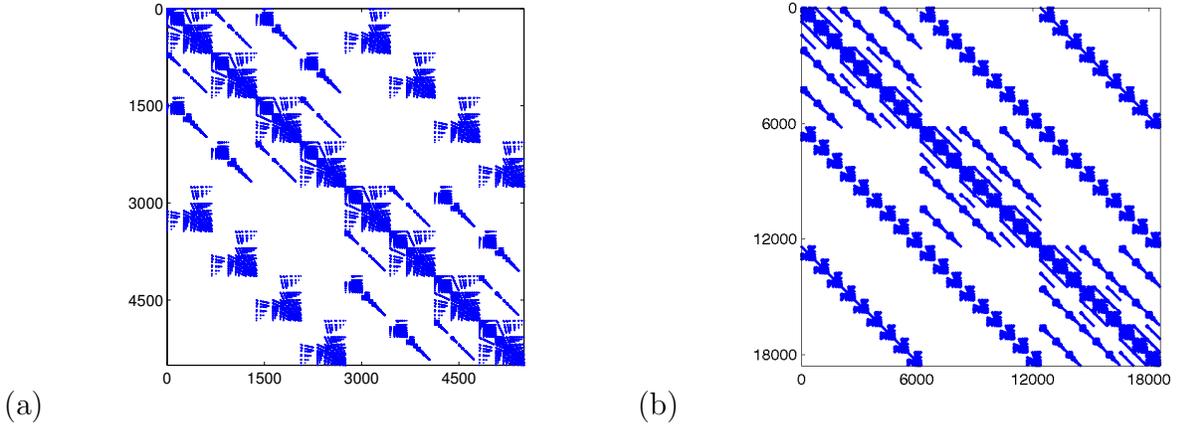


Figure 3.2: Example 3.3.4. The sparsity patterns of the matrices computed by the space  $\hat{\mathbf{V}}_6^k$  with (a)  $k = 1$  and (b)  $k = 2$ .  $N = 6$ .

**Example 3.3.5** We solve the following three-dimensional problem with smooth variable coefficient on  $\Omega = [0, 1]^3$ .

$$\begin{cases} -\nabla \cdot ((\sin(x_1 x_2 x_3) + 1) \nabla u) = f, & \mathbf{x} \in \Omega, \\ u = 0, & \mathbf{x} \in \partial\Omega, \end{cases} \quad (3.13)$$

$f$  is a given function such that the exact solution is  $u = \sin(\pi x_1) \sin(\pi x_2) \sin(\pi x_3)$ . We provide the numerical results with  $k = 1$  and  $k = 2$  in Table 3.14. The conclusion is similar to Example 3.3.4.

Table 3.14: Numerical errors and orders of accuracy for Example 3.3.5 computed by the space  $\hat{\mathbf{V}}_N^k$  with  $k = 1, 2$  and indicated  $N$ .

N	$L^1$ error	order	$L^2$ error	order	$L^\infty$ error	order	$H^1$ error	order
$k = 1$								
3	2.64E-02		3.40E-02		1.55E-01		4.32E-01	
4	6.23E-03	2.08	8.58E-03	1.98	3.54E-02	2.13	2.04E-01	1.08
5	1.49E-03	2.06	2.10E-04	2.03	2.07E-02	0.77	9.82E-02	1.06
6	3.68E-04	2.02	5.32E-04	1.98	7.58E-03	1.45	4.80E-02	1.03
$k = 2$								
3	1.63E-04		2.05E-04		8.24E-04		1.19E-02	
4	2.88E-05	2.50	3.66E-05	2.48	1.63E-04	2.34	3.00E-03	1.98
5	4.72E-06	2.61	6.06E-06	2.60	2.73E-05	2.58	7.54E-04	2.00
6	7.42E-07	2.67	9.58E-07	2.66	5.80E-06	2.23	1.88E-04	2.00

### 3.3.3 Four-dimensional Results

In this subsection, we gather computational results for four-dimensional cases. The penalty constant in this subsection is chosen to be  $\sigma = 30$  for  $k = 1$ , and  $\sigma = 60$  for  $k = 2$ .

**Example 3.3.6** We solve the following four-dimensional problem with constant variable coefficient on  $\Omega = [0, 1]^4$ .

$$\begin{cases} -\Delta u = 0 & \mathbf{x} \in \Omega, \\ u = \sin(\pi x_1) \sin(\pi x_2) \sin(\pi x_3) \frac{\sinh(\pi x_4)}{\sinh(\pi)} & \mathbf{x} \in \partial\Omega \end{cases} \quad (3.14)$$

where the exact solution is  $u = \sin(\pi x_1) \sin(\pi x_2) \sin(\pi x_3) \frac{\sinh(\pi x_4)}{\sinh(\pi)}$ .

We plot the numerical results for  $k = 1$  and  $k = 2$  in Figure 3.3 and Table 3.15. For this four-dimensional example, we obtain  $k$ -th order for the  $H^1$  norm, and close to  $(k + \frac{1}{2})$ -th order for the  $L^1$  and  $L^2$  norm. The  $L^\infty$  error seems to fluctuate. However, we expect the order of accuracy in  $L^\infty$  norm to grow upon refinement. The sparsity patterns of the stiffness matrices for  $k = 1$  and  $k = 2$  are reported in Figure 3.4. From Table 3.16, we can see the

Table 3.15: Numerical errors and orders of accuracy for Example 3.3.6 computed by the space  $\hat{\mathbf{V}}_N^k$  with  $k = 1, 2$  and indicated  $N$ .

N	$L^1$ error	order	$L^2$ error	order	$L^\infty$ error	order	$H^1$ error	order
$k = 1$								
3	2.44E-02		4.22E-02		3.31E-01		3.91E-01	
4	1.08E-02	1.18	2.08E-02	1.02	1.16E-01	1.51	2.37E-01	0.73
5	3.68E-03	1.54	7.15E-03	1.54	9.33E-02	0.31	1.22E-01	0.96
$k = 2$								
2	8.21E-04		1.34E-03		1.11E-02		4.20E-02	
3	1.76E-04	2.22	2.79E-04	2.27	2.76E-03	2.00	1.20E-02	1.81
4	3.32E-05	2.40	5.39E-05	2.37	8.76E-04	1.66	3.18E-03	1.91

stiffness matrices scale less than the two-dimensional and three-dimensional examples, i.e. the number of nonzero elements scales like  $O(SGDOF^{1.35})$ , where  $SGDOF$  is the degree of freedom of the space used.

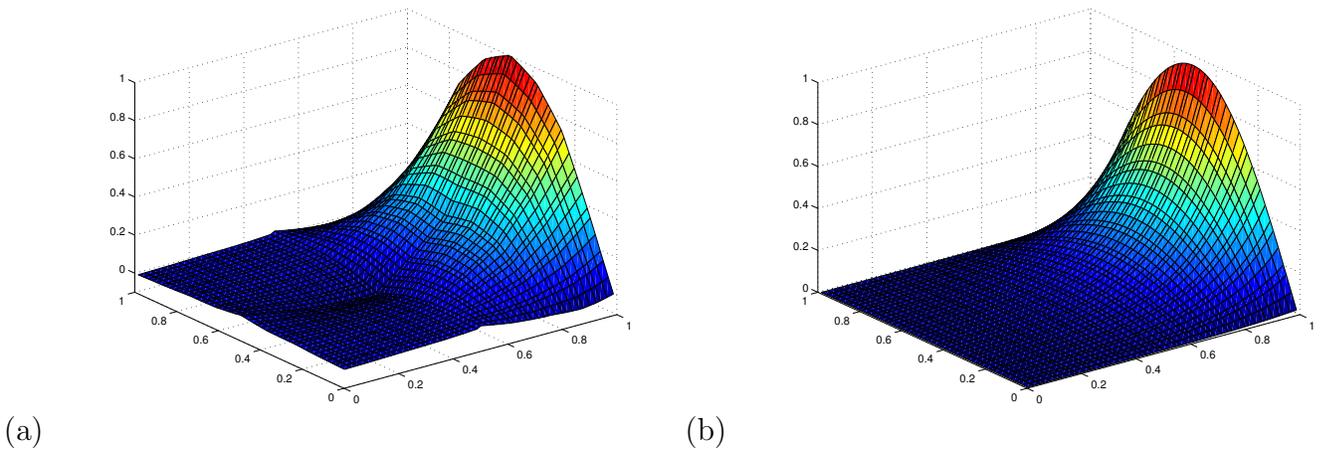


Figure 3.3: Example 3.3.6. (a)  $k = 1$  ; (b)  $k = 2$ .  $N = 4$ . Plotted along  $x_1 = 0.4930, x_2 = 0.4930$ .

Table 3.16: Sparsity and condition number of the stiffness matrix. Example 3.3.6 computed by the space  $\hat{\mathbf{V}}_N^k$  with  $k = 1, 2$ . SGDOF is the degree of freedom used for the sparse grid DG scheme. NNZ is the number of nonzero elements in the stiffness matrix. Order= $\log(\text{NNZ})/\log(\text{SGDOF})$ .

N	SGDOF	NNZ	Order	Condition Number
$k = 1$				
3	1008	12272	1.36	4.27E+02
4	3072	51712	1.35	2.26E+03
5	8832	187008	1.34	9.27E+03
$k = 2$				
2	1539	19683	1.35	7.40E+02
3	5103	102303	1.35	2.62E+03
4	15552	420336	1.34	9.72E+03

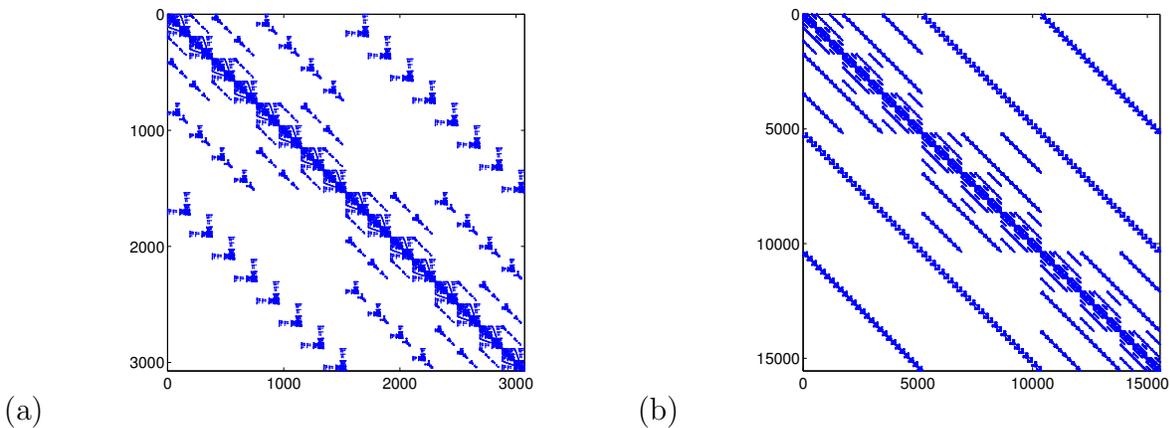


Figure 3.4: Example 3.3.6. The sparsity patterns of the matrices computed by the space  $\hat{\mathbf{V}}_4^k$  with (a)  $k = 1$  and (b)  $k = 2$ .

**Example 3.3.7** We solve the following four-dimensional problem with smooth variable coefficient on  $\Omega = [0, 1]^4$ .

$$\begin{cases} -\nabla \cdot ((\sin(x_1 x_2 x_3 x_4) + 1) \nabla u) = f, & \mathbf{x} \in \Omega, \\ u = 0, & \mathbf{x} \in \partial\Omega, \end{cases} \quad (3.15)$$

$f$  is a given function such that the exact solution is  $u = \sin(\pi x_1) \sin(\pi x_2) \sin(\pi x_3) \sin(\pi x_4)$ .

We provide the numerical results with  $k = 1$  and  $k = 2$  in Table 3.17. The conclusion is similar to Example 3.3.6.

Table 3.17: Numerical errors and orders of accuracy for Example 3.3.7 computed by the space  $\hat{\mathbf{V}}_N^k$  with  $k = 1, 2$  and indicated  $N$ .

N	$L^1$ error	order	$L^2$ error	order	$L^\infty$ error	order	$H^1$ error	order
$k = 1$								
3	6.15E-02		8.97E-02		2.94E-01		6.67E-01	
4	1.89E-02	1.70	2.63E-02	1.77	2.54E-01	0.21	3.20E-01	1.06
5	4.51E-03	2.07	6.80E-03	1.95	7.15E-02	1.83	1.45E-01	1.14
$k = 2$								
2	8.38E-04		1.09E-03		3.49E-03		3.74E-02	
3	1.62E-04	2.37	2.13E-04	2.36	1.34E-03	1.38	1.01E-02	1.90
4	2.97E-05	2.44	3.91E-05	2.45	3.80E-04	1.82	2.57E-03	1.97

# Chapter 4

## Conclusion

In this thesis, we proposed efficient DG schemes for HJ equations and high-dimensional elliptic equations to save computation and storage costs while maintaining accuracy of the numerical solution.

In Chapter 2, we proposed a new entropy fix DG method to directly solve for the viscosity solution of the general HJ equations. The Harten and Hyman entropy fix term, which is proportional to the jump of the normal derivative of the numerical solution, is added to correct the entropy violation automatically when needed. One and two dimensional numerical experiments on both structured and unstructured meshes were provided to demonstrate good performance of this method.

In Chapter 3, we developed a sparse grid IPDG method for efficient computations of high-dimensional second-order elliptic problems. The orthonormal hierarchical tensor product basis representation allowed us to utilize the sparse grid technique to reduce the degrees of freedom from the standard exponential dependence  $O(h^{-d})$  to  $O(h^{-1}|\log_2 h|^{d-1})$  for  $d$ -dimensional problems. Error estimate in the energy norm has been proved for the IP formulation of the elliptic equations. The good performance of this method is validated by the numerical results in multi-dimensions. Compared to conventional full grid methods, the new sparse grid DG methods can save storage and computation cost as the size of approximation spaces are significantly reduced. Future work includes the development of adaptive sparse grid DG methods based on hierarchical surplus and approximation theory

for various types of sparse finite element space.

# BIBLIOGRAPHY

# BIBLIOGRAPHY

- [1] R. Abgrall. Numerical discretization of the first-order Hamilton-Jacobi equation on triangular meshes. *Comm. Pure Appl. Math.*, 49(12):1339–1373, 1996.
- [2] S Achatz. Higher order sparse grid methods for elliptic partial differential equations with variable coefficients. *Computing*, 71(1):1–15, 2003.
- [3] B. Alpert, G. Beylkin, D. Gines, and L. Vozovoi. Adaptive solution of partial differential equations in multiwavelet bases. *J. Comput. Phys.*, 182(1):149–190, 2002.
- [4] B.K. Alpert. A class of bases in  $L^2$  for the sparse representation of integral operators. *SIAM J. Math. Anal.*, 24(1):246–262, 1993.
- [5] R. Archibald, G. Fann, and W. Shelton. Adaptive discontinuous Galerkin methods in multiwavelets bases. *Appl. Numer. Math.*, 61(7):879–890, 2011.
- [6] D.A. Arnold. An interior penalty finite element method with discontinuous elements. *SIAM J. Numer. Anal.*, 19(4):742–760, 1982.
- [7] D.A. Arnold, F. Brezzi, B. Cockburn, and L. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM J. Numer. Anal.*, 39:1749–1779, 2002.
- [8] K.I. Babenko. Approximation of periodic functions of many variables by trigonometric polynomials. *Doklady Akademii Nauk SSSR*, 132(2):247–250, 1960.
- [9] I. Babuška and M. Zlámal. Nonconforming elements in the finite element method with penalty. *SIAM J. Numer. Anal.*, 10(5):863–875, 1973.
- [10] G.A. Baker. Finite element methods for elliptic equations using nonconforming elements. *Math. Comp.*, 31(137):45–59, 1977.
- [11] G.A. Baker, W.N. Jureidini, and O.A. Karakashian. Piecewise solenoidal vector fields and the Stokes problem. *SIAM J. Numer. Anal.*, 27(6):1466–1485, 1990.
- [12] F. Bassi and S. Rebay. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. *J. Comput. Phys.*, 131:267–279, 1997.

- [13] G. Baszenski, F.-J. Delves, and S. Jester. Blending approximations with sine functions. In *Numerical Methods in Approximation Theory*, volume 9, pages 1–19. Springer, 1992.
- [14] R.E. Bellman. *Adaptive control processes: a guided tour*, volume 4. Princeton University Press Princeton, 1961.
- [15] O. Bokanowski, Y. Cheng, and C.-W. Shu. A discontinuous Galerkin solver for front propagation. *SIAM J. Sci. Comput.*, 33:923–938, 2011.
- [16] O. Bokanowski, Y. Cheng, and C.-W. Shu. A discontinuous Galerkin scheme for front propagation with obstacles. *Numer. Math.*, 126(1):1–31, 2014.
- [17] F. Brezzi, G. Manzini, D. Marini, P. Pietra, and A. Russo. Discontinuous Galerkin approximations for elliptic problems. *Numerical Methods for Partial Differential Equations*, 16(4):365–378, 2000.
- [18] H.-J. Bungartz. A multigrid algorithm for higher order finite elements on sparse grids. *Electron T. Numer. Ana.*, 6:63–77, 1997.
- [19] H.-J. Bungartz. *Finite elements of higher order on sparse grids*. Shaker, 1998.
- [20] H.-J. Bungartz and T. Dornseifer. Sparse grids: Recent developments for elliptic partial differential equations. In *Multigrid Methods V*, pages 45–70. Springer, 1998.
- [21] H.-J. Bungartz and M. Griebel. Sparse grids. *Acta numer.*, 13:147–269, 2004.
- [22] J.L.D. Calle, P.R.B. Devloo, and S.M. Gomes. Wavelets and adaptive grids for the discontinuous Galerkin method. *Numerical Algorithms*, 39(1-3):143–154, 2005.
- [23] P. Castillo, B. Cockburn, I. Perugia, and D. Schötzau. An a priori error analysis of the local discontinuous Galerkin method for elliptic problems. *SIAM J. Numer. Anal.*, 38(5):1676–1706, 2000.
- [24] Y. Chen and B. Cockburn. An adaptive high-order discontinuous Galerkin method with error control for the Hamilton-Jacobi equations. Part I: The one-dimensional steady state case. *J. Comput. Phys.*, 226(1):1027–1058, 2007.
- [25] Y. Cheng and C.-W. Shu. A discontinuous Galerkin finite element method for directly solving the Hamilton-Jacobi equations. *J. Comput. Phys.*, 223:398–415, 2007.

- [26] P. Ciarlet. *The finite element method for elliptic problems*. North-Holland, Amsterdam, 1975.
- [27] B. Cockburn and C. Dawson. Some extensions of the local discontinuous Galerkin method for convection-diffusion equations in multidimensions. In *The Proceedings of the Conference on the Mathematics of Finite Elements and Applications: MAFELAP X*. Elsevier, pages 225–238, 2000.
- [28] B. Cockburn, G. Kanschat, I. Perugia, and D. Schötzau. Superconvergence of the local discontinuous Galerkin method for elliptic problems on cartesian grids. *SIAM J. Numer. Anal.*, 39(1):264–285, 2001.
- [29] B. Cockburn, F. Li, and C.-W. Shu. Locally divergence-free discontinuous Galerkin methods for the Maxwell equations. *J. Comput. Phys.*, 194:588–610, 2004.
- [30] B. Cockburn and C.-W. Shu. The local discontinuous Galerkin method for time-dependent convection-diffusion systems. *SIAM J. Numer. Anal.*, 35(6):2440–2463, 1998.
- [31] B. Cockburn and C.-W. Shu. Runge-Kutta discontinuous Galerkin methods for convection-dominated problems. *J. Sci. Comput.*, 16:173–261, 2001.
- [32] M.G. Crandall, L.C. Evans, and P.-L. Lions. Some properties of viscosity solutions of Hamilton-Jacobi equations. *Trans. Amer. Math. Soc.*, 282(2):487–502, 1984.
- [33] M.G. Crandall and P.-L. Lions. Viscosity solutions of Hamilton-Jacobi equations. *Trans. Amer. Math. Soc.*, 277:1–42, 1983.
- [34] M.G. Crandall and P.-L. Lions. Two approximations of solutions of Hamilton-Jacobi equations. *Math. Comp.*, 43:1–19, 1984.
- [35] C. Dawson, S. Sun, and M.F. Wheeler. Compatible algorithms for coupled flow and transport. *Comput. Methods Appl. Mech. Eng.*, 193(23):2565–2580, 2004.
- [36] F.-J. Delvos.  $d$ -variate Boolean interpolation. *J. Approx. Theory.*, 34(2):99–114, 1982.
- [37] J. Douglas and T. Dupont. Interior penalty procedures for elliptic and parabolic galerkin methods. In *Computing methods in applied sciences*, pages 207–216. Springer, 1976.
- [38] J. Garcke and M. Griebel. *Sparse grids and applications*. Springer, 2013.

- [39] N. Gerhard and S. Müller. Adaptive multiresolution discontinuous Galerkin schemes for conservation laws: multi-dimensional case. *Comput. Appl. Math.*, pages 1–29, 2013.
- [40] T. Gerstner and M. Griebel. Numerical integration using sparse grids. *Numerical algorithms*, 18(3-4):209–232, 1998.
- [41] S. Gottlieb, C.-W. Shu, and E. Tadmor. Strong stability preserving high order time discretization methods. *SIAM Rev*, 43:89–112, 2001.
- [42] V. Gradinaru. Fourier transform on sparse grids: code design and the time dependent Schrödinger equation. *Computing*, 80(1):1–22, 2007.
- [43] M. Griebel. A parallelizable and vectorizable multi-level algorithm on sparse grids. In W. Hackbusch, editor, *Parallel algorithms for partial differential equations*, volume 31 of *Notes on numerical fluid mechanics*, pages 94–100. 1991.
- [44] M. Griebel. Adaptive sparse grid multilevel methods for elliptic PDEs based on finite differences. *Computing*, 61(2):151–179, 1998.
- [45] M. Griebel. Sparse grids and related approximation schemes for higher dimensional problems. In *Proceedings of the conference on foundations of computational mathematics*, Santander, Spain, 2005.
- [46] M. Griebel and J. Hamaekers. Sparse grids for the Schrödinger equation. *ESAIM: Mathematical Modelling and Numerical Analysis*, 41(02):215–247, 2007.
- [47] M. Griebel and P. Oswald. Tensor product type subspace splittings and multilevel iterative methods for anisotropic problems. *Adv. Comput. Math.*, 4(1):171–206, 1995.
- [48] M. Griebel, C. Zenger, and S. Zimmer. Multilevel Gauss-Seidel-algorithms for full and sparse grid problems. *Computing*, 50(2):127–148, 1993.
- [49] M. Griebel and G. Zumbusch. Adaptive sparse grids for hyperbolic conservation laws. In *Hyperbolic problems: theory, numerics, applications*, pages 411–422. Springer, 1999.
- [50] W. Guo, F. Li, and J. Qiu. Local-structure-preserving discontinuous Galerkin methods with Lax-Wendroff type time discretizations for Hamilton-Jacobi equations. *J. Sci. Comput.*, 47(2):239–257, 2011.
- [51] A. Haar. Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen*, 69(3):331–371, 1910.

- [52] A. Harten. Multiresolution algorithms for the numerical solution of hyperbolic conservation laws. *Comm. Pure Appl. Math.*, 48(12):1305–1342, 1995.
- [53] A. Harten and J.M. Hyman. Self adjusting grid methods for one-dimensional hyperbolic conservation laws. *J. Comput. Phys.*, 50(2):235–269, 1983.
- [54] P.W. Hemker. Sparse-grid finite-volume multigrid for 3D-problems. *Adv. Comput. Math.*, 4(1):83–110, 1995.
- [55] N. Hovhannisyanyan, S. Müller, and R. Schäfer. Adaptive multiresolution discontinuous Galerkin schemes for conservation laws. *Math. Comp.*, 83(285):113–151, 2014.
- [56] C. Hu and C.-W. Shu. A discontinuous Galerkin finite element method for Hamilton-Jacobi equations. *SIAM J. Sci. Comput.*, 21:666–690, 1999.
- [57] F. Iacono, G. May, S. Müller, and R. Schäfer. A high-order discontinuous Galerkin discretization with multiwavelet-based grid adaptation for compressible flows. *AICES Preprint AICES-2011/08-1, RWTH Aachen*, 2011.
- [58] G. Jiang and D. Peng. Weighted ENO schemes for Hamilton-Jacobi equations. *SIAM J. Sci. Comput.*, 21:2126–2143, 1999.
- [59] S. Jin and Z. Xin. Numerical passage from systems of conservation laws to Hamilton-Jacobi equations, and relaxation schemes. *SIAM J. Numer. Anal.*, 35(6):2385–2404, 1998.
- [60] L. Krivodonova. Limiters for high-order discontinuous Galerkin methods. *J. Comput. Phys.*, 226(1):879–896, 2007.
- [61] F. Li and C.-W. Shu. Reinterpretation and simplified implementation of a discontinuous Galerkin method for Hamilton-Jacobi equations. *Appl. Math. Lett.*, 18:1204–1209, 2005.
- [62] F. Li and S. Yakovlev. A central discontinuous Galerkin method for Hamilton-Jacobi equations. *J. Sci. Comput.*, 45(1-3):404–428, 2010.
- [63] C.B. Liem, T. Lü, and T.M. Shih. *The splitting extrapolation method: a new technique in numerical solution of multidimensional problems*, volume 7. World Scientific, 1995.
- [64] W.A. Mulder. A new multigrid approach to convection problems. *J. Comput. Phys.*, 83(2):303–323, 1989.

- [65] N.H. Naik and J. Van Rosendale. The improved robustness of multigrid elliptic solvers based on multiple semicoarsened grids. *SIAM J. Numer. Anal.*, 30(1):215–229, 1993.
- [66] J.T. Oden, I. Babuška, and C.E. Baumann. A discontinuous  $hp$  finite element method for diffusion problems. *J. Comput. Phys.*, 146(2):491–519, 1998.
- [67] S. Osher and J.A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.*, 79(1):12–49, 1988.
- [68] S. Osher and C.-W. Shu. High order essentially non-oscillatory schemes for Hamilton-Jacobi equations. *SIAM J. Numer. Anal.*, 28:907–922, 1991.
- [69] C. Pflaum. A multilevel algorithm for the solution of second order elliptic differential equations on sparse grids. *Numer. Math.*, 79(1):141–155, 1998.
- [70] B. Rivière, M.F. Wheeler, and V. Girault. A priori error estimates for finite element methods based on discontinuous approximation spaces for elliptic problems. *SIAM J. Numer. Anal.*, 39(3):902–931, 2001.
- [71] C. Schwab, E. Süli, and R.A. Todor. Sparse finite element approximation of high-dimensional transport-dominated diffusion problems. *ESAIM: Mathematical Modelling and Numerical Analysis*, 42(05):777–819, 2008.
- [72] J. Shen and L.-L. Wang. Sparse spectral approximations of high-dimensional problems based on hyperbolic cross. *SIAM J. Numer. Anal.*, 48(3):1087–1109, 2010.
- [73] J. Shen and H. Yu. Efficient spectral sparse grid methods and applications to high-dimensional elliptic problems. *SIAM J. Sci. Comput.*, 32(6):3228–3250, 2010.
- [74] C.-W. Shu. High order numerical methods for time dependent Hamilton-Jacobi equations. *Mathematics and Computation in Imaging Science and Information Processing, Lect. Notes Ser. Inst. Math. Sci. Natl. Univ. Singap*, 11:47–91, 2007.
- [75] C.-W. Shu. Survey on discontinuous Galerkin methods for Hamilton-Jacobi equations. *Recent Advances in Scientific Computing and Applications*, 586:323, 2013.
- [76] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *J. Comput. Phys.*, 77:439–471, 1988.
- [77] S.A. Smolyak. Quadrature and interpolation formulas for tensor products of certain classes of functions. In *Dokl. Akad. Nauk SSSR*, volume 4, pages 240–243, 1963.

- [78] P.E. Souganidis. Approximation schemes for viscosity solutions of Hamilton-Jacobi equations. *J. Differ. Equations*, 59(1):1–43, 1985.
- [79] V.N. Temlyakov. Approximations of functions with bounded mixed derivative. *Trudy Matematicheskogo Instituta im. VA Steklova*, 178:3–113, 1986.
- [80] M.J. Vuik and J.K. Ryan. Multiwavelet troubled-cell indicator for discontinuity detection of discontinuous Galerkin schemes. *J. Comput. Phys.*, 270:138–160, 2014.
- [81] W. Wang and C.-W. Shu. The WKB local discontinuous Galerkin method for the simulation of Schrödinger equation in a resonant tunneling diode. *J. Sci. Comput.*, 40(1-3):360–374, 2009.
- [82] M.F. Wheeler. An elliptic collocation-finite element method with interior penalties. *SIAM J. Numer. Anal.*, 15(1):152–161, 1978.
- [83] T. Xiong, C.-W. Shu, and M. Zhang. A priori error estimates for semi-discrete discontinuous Galerkin methods solving nonlinear Hamilton-Jacobi equations with smooth solutions. *Int. J. Numer. Anal. Mod.*, 10:154–177, 2013.
- [84] J. Yan and S. Osher. A local discontinuous Galerkin method for directly solving Hamilton-Jacobi equations. *J. Comput. Phys.*, 230(1):232–244, 2011.
- [85] L. Yuan and C.-W. Shu. Discontinuous Galerkin method based on non-polynomial approximation spaces. *J. Comput. Phys.*, 218(1):295–323, 2006.
- [86] C. Zenger. Sparse grids. In *Parallel Algorithms for Partial Differential Equations, Proceedings of the Sixth GAMM-Seminar*, volume 31, 1990.
- [87] Y.-T. Zhang and C.-W. Shu. High order WENO schemes for Hamilton-Jacobi equations on triangular meshes. *SIAM J. Sci. Comput.*, 24:1005–1030, 2003.