

Summer 2024

Accelerating the Efficiency of Multiscale Hybridizable Discontinuous Galerkin Methods for Flows in Heterogeneous Media

Tony Charles Haines
Old Dominion University, thainesc49@gmail.com

Follow this and additional works at: https://digitalcommons.odu.edu/mathstat_etds



Part of the [Applied Mathematics Commons](#), [Computer Sciences Commons](#), and the [Other Physics Commons](#)

Recommended Citation

Haines, Tony C.. "Accelerating the Efficiency of Multiscale Hybridizable Discontinuous Galerkin Methods for Flows in Heterogeneous Media" (2024). Doctor of Philosophy (PhD), Dissertation, Mathematics & Statistics, Old Dominion University, DOI: 10.25777/e475-5362
https://digitalcommons.odu.edu/mathstat_etds/132

This Dissertation is brought to you for free and open access by the Mathematics & Statistics at ODU Digital Commons. It has been accepted for inclusion in Mathematics & Statistics Theses & Dissertations by an authorized administrator of ODU Digital Commons. For more information, please contact digitalcommons@odu.edu.

**ACCELERATING THE EFFICIENCY OF MULTISCALE HYBRIDIZABLE
DISCONTINUOUS GALERKIN METHODS FOR FLOWS IN HETEROGENEOUS
MEDIA**

by

Tony Charles Haines

B.S. Chemistry, August 2014, Virginia Commonwealth University

M.S. Mathematics, December 2017, Virginia State University

M.S. Mathematics, August 2024, Old Dominion University

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

COMPUTATIONAL AND APPLIED MATHEMATICS

OLD DOMINION UNIVERSITY

August 2024

Approved by:

Ke Shi (Director)

Yan Peng (Member)

Ruhai Zhou (Member)

Yongjin Lu (Member)

ABSTRACT

ACCELERATING THE EFFICIENCY OF MULTISCALE HYBRIDIZABLE DISCONTINUOUS GALERKIN METHODS FOR FLOWS IN HETEROGENEOUS MEDIA

Tony Charles Haines
Old Dominion University, 2024
Director: Dr. Ke Shi

A plethora of scientific and engineering problems encountered are multiscale in nature. This multiscale feature often influences simulation efforts wherever large disparities in spatial scales are experienced. Notable examples include composite materials, fluid flow through porous media and turbulent transport in high Reynolds number flow. Although there are promising results from the advancement of modern supercomputer, obtaining direct numerical solution of multiscale problems is very laborious. This difficulty stems from the tremendous amount of computer memory and CPU time required. Parallel computing may be one obvious choice in remedying this issue. However, the complexity and size of the discrete problem is not reduced. The goal of this dissertation is to design a multiscale model reduction framework within the hybridizable discontinuous Galerkin (HDG) finite element method. We utilize local snapshots that incorporate some local features of the solution space in constructing a lower dimensional trace space. This approach affords us the opportunity to avoid high dimensional representation of the trace spaces. Furthermore, we leverage the advantages of localized multiscale basis functions to capture the multiscale structure of the solution rather than the standard polynomial basis. These basis functions contain essential multiscale information embedded in the solution. They allow us to obtain better approximations through the coarse space enrichment. Moreover, orthogonality and its sparse representation are preserved. With these tools, we can construct coarse scale solutions accurately and efficiently

without solving a global fine scale system. We employ the use of neural network to further improve the efficiency of our method by training the network to learn the solution map of our model. Such training is done on a single coarse block instead of the entire domain. A significant advantage of this approach is that, once trained, this network can be used for any geometry and parameter distribution without retraining. We can avoid the time-consuming global assembly due to the local solvers.

Copyright, 2024, by Tony Charles Haines, All Rights Reserved.

I dedicate my dissertation to my father, Emmanuel M. Haines and my mother, Louise B. Haines.

ACKNOWLEDGMENTS

I would like to express my utmost gratitude to God Almighty for his never failing grace, blessings and unmerited favor upon my life. He has giving me the strength and patience to pursue all my endeavors. I express sincere gratitude to my advisor, Dr. Ke Shi for his immeasurable guidance, encouragement and pertinent discussions. I appreciate every opportunity given to me to work with him on this research project. I am thankful to Dr. Yan Peng, Dr. Ruhai Zhou and Dr. Yongjin Lu for serving as committee members and giving thoughtful advice.

Last but not the least, I would like to thank my family, parents and friends for their love and support throughout my graduate studies.

TABLE OF CONTENTS

	Page
LIST OF TABLES	viii
LIST OF FIGURES	ix
 Chapter	
1. INTRODUCTION	1
1.1 MOTIVATION	1
1.2 SCOPE	2
1.3 THESIS ORGANIZATION	2
2. BACKGROUND	4
2.1 LITERATURE REVIEW	4
2.2 STANDARD FINITE ELEMENT METHOD	6
2.3 PRINCIPLE OF HYBRIDIZATION	15
3. PROBLEM STATEMENT	19
3.1 DERIVATION OF THE HDG SCHEME	20
4. MULTISCALE HDG	23
4.1 FINE AND COARSE GRIDS	23
4.2 GLOBAL FORMULATION	24
5. MULTISCALE ORTHOGONAL BASES FUNCTIONS	28
5.1 MULTISCALE FUNTIONS ON THE UNIT INTERVAL	28
5.2 CONSTRUCTION OF MULTISCALE BASIS	30
5.3 EXAMPLES OF MULTISCALE FUNCTIONS	32
6. NUMERIAL RESULTS	37
6.1 MSHDG RESULTS	37
6.2 ARTIFICIAL NEURAL NETWORK	42
7. CONCLUSION	70
REFERENCES	72

LIST OF TABLES

Table	Page
1. Error and processor time for different scale levels on M_H	41
2. Elapsed time (seconds) to achieve an approximate solution in both methods at scale level L_0	47
3. Errors between the MsHDG and Artificial Neural Network (ANN) solutions for different test domains.....	47

LIST OF FIGURES

Figure	Page
1. Functions in finite dimensional space.....	8
2. Triangulation \mathcal{T}_h with mesh size h	12
3. Some common finite elements.....	15
4. Partition into two nonoverlapping subdomains.....	17
5. Fine and coarse grids.	24
6. Application of the operator \mathcal{T}_ε to a function.	29
7. Orthonormal basis on \mathbb{X}_0	32
8. Orthonormal basis for \mathbb{W}_1	33
9. Orthonormal basis for \mathbb{W}_2	34
10. Orthonormal basis for \mathbb{W}_3	35
11. Orthonormal basis for \mathbb{W}_3	36
12. Distribution of kappa.	38
13. Comparison of MsHDG solutions for M_H	39
14. Comparison of MsHDG solutions for M_H	40
15. Fine-scale solution.	41
16. Coarse block learning.....	43
17. Random samples of kappa κ	44
18. Training and testing errors for neural network.	45
19. Neural network architecture.....	48
20. Test domain 1.....	49
21. L_0 neural network solution for Test domain 1.	50
22. L_0 MsHDG solution for Test domain 1.....	51

Figure	Page
23. Test domain 2.....	52
24. L_0 neural network solution for Test domain 2.	53
25. L_0 MsHDG solution for Test domain 2.....	54
26. Test domain 3.....	55
27. L_0 neural network solution for Test domain 3.	56
28. L_0 MsHDG solution for Test domain 3.....	57
29. Test domain 4.....	58
30. L_0 neural network solution for Test domain 4.	59
31. L_0 MsHDG solution for Test domain 4.....	60
32. Test domain 5.....	61
33. L_0 neural network solution for Test domain 5.	62
34. L_0 MsHDG solution for Test domain 5.....	63
35. Test domain 6.....	64
36. L_1 neural network solution for Test domain 6.	65
37. L_1 MsHDG solution for Test domain 6.....	66
38. Test domain 7.....	67
39. L_1 neural network solution for Test domain 7.	68
40. L_1 MsHDG solution for Test domain 7.....	69

CHAPTER 1

INTRODUCTION

1.1 MOTIVATION

Flows in porous media appear in many scientific, industrial, engineering and environmental applications. One of the many characteristics of these diverse areas is that porous media are intrinsically multiscale and particularly display heterogeneities over a wide range of length-scales. Such heterogeneity is often represented by the multiscale fluctuations in the permeability of the media. For composite or naturally occurring materials, e.g., soil or rock, the permeability is small in granite formations (say, 10^{-15} cm²), medium in oil reservoirs, (say, 10^{-7} cm² to 10^{-9} cm²), and large in highly fractured materials (say, 10^{-15} cm²) [25]. The dispersed phases (particles or fibers), which may be randomly distributed in the matrix, may also give rise to discontinuity in electrical conductivity. Furthermore, in turbulent transport problems, the convective velocity field fluctuates randomly and contains many scales depending on the Reynolds number of flow [30].

Numerical solution of such problems poses a great challenge even with modern supercomputers and has attracted substantial attention in the scientific and engineering society. In groundwater simulations, it is quite common to have millions of grid blocks involved, with each block having a dimension of tens of meters, whereas the permeability measured from cores is at a scale of several centimeters [30]. This results in more than 10^5 degrees of freedom per spatial dimension in computation. Therefore, a tremendous amount of computer memory and CPU time are required, which may easily exceed the limit of today's computing resources. The situation is somewhat re-

lieved by parallel computing; however, the size of the discrete problem is not reduced. It is merely distributed between processors with more memory. Whenever it is affordable to resolve all the small scale features of a physical problem, direct solutions provide quantitative information of the physical processes at all scales. However, from an engineering perspective, it is often sufficient to predict the macroscopic properties of the multiscale systems, such as the effective conductivity, elastic moduli, permeability and eddy diffusivity. This necessitates the use of reduced-order methods or specialized techniques which are capable of capturing the fine scale effects on the larger scale, without ever fully resolving all the fine scale features.

1.2 SCOPE

The methods presented in this work analyse the case of scale separation, and attempt to capture the multiscale structure of the solution via basis functions within localized regions (Coarse-scale grid blocks). These basis functions contain essential multiscale information and are coupled through a global formulation to provide an approximate solution. Furthermore, the use of these basis functions allows parallel computing, which distributes the computation and reduces CPU time as more processors are used. And though the operation count of the multiscale method is comparable with that of conventional finite element method (FEM), this work aims to demonstrate the application of parallel computing and artificial neural network to computational time reduction.

1.3 THESIS ORGANIZATION

The thesis consists of seven chapters. **Chapter 2** gives a background to reduced-order methods and subsequently, multiscale methods. A brief literature review and discussion of multiscale methods are proposed. **Chapter 3** describes the physical problem under consideration and mathematical

details within the frame work of HDG finite element method. **Chapter 4** provides an introduction to Multiscale Finite Element Methods (MsFEM) and Hybridizable Discontinuous Galerkin scheme (HDG). **Chapter 5** presents multiscale basis functions. A brief description and detailed construction are provided. In **Chapter 6**, we presents the numerical results which includes solutions from coarse-scale enrichment and the neural network approximation of the multiscale HDG solution. **Chapter 7** provides a summary or conclusion of this study.

CHAPTER 2

BACKGROUND

Multiscale FEM (MsFEM) is a procedure of numerical upscaling that extends the capabilities of the mathematical theory of homogenization to more general cases including materials with nonperiodic properties, nonseparable scales, and/or random coefficients [25]. The main idea of this method is to obtain the large scale solution accurately and efficiently without resolving the small scale information. This is made possible by the construction of basis functions which capture the small scale information within each coarse blocks. The small scale information is then brought to the large scales through the coupling of the global stiffness matrix. Thus, the effect of the small scales on the large scales is captured correctly. As a result, methods that can flexibly couple the local multiscale solutions are much desirable.

2.1 LITERATURE REVIEW

Many multiscale numerical techniques have been developed and studied in literature. One of the earlier and efficient mixed finite element approximations on multiblock grids was introduced by Arbogast et al. in [6], Mary Wheeler and co-authors in [7]. The multiblock method proposed here appeared to be very flexible in constructing finite element approximations independently on each block. Possible discontinuities along the interblock faces were treated using mortar spaces, a tool borrowed from domain decomposition. This can be seen in the works of [12]. These techniques were introduced to accommodate methods that could be defined in separate subdomains and meshed independently. Mortar methods [13] introduces an auxiliary space for the Lagrange

multiplier associated with the continuity constraint on the global approximation solution. Classical mortaring techniques that have been extended as multiscale finite element methods can be found in [4], [8], [10], [28]. In a two scale (two-grid, fine and coarse) method, the aim is to resolve the local heterogeneities on the fine grid introduced on each coarse block and then glue these approximations together via mortar spaces. However, to have a well stable method, the mortar spaces had to satisfy a proper inf-sup condition. This method was shown to be well suited for problems with heterogeneous media, and a number of efficient methods and implementations have been proposed.

More recently, new classes of reduced-order methods have been presented. These include Galerkin multiscale finite element method (e.g., [3], [17], [22], [26], [27], [29]), mixed multiscale finite element methods (e.g., [1], [2], [5], [32]), multiscale finite volume method [33], mortar multiscale methods (e.g., [9], [36]), and variational multiscale methods [31]. In [24], the concept of Generalized Multiscale Finite Element proposed by [23] was used to construct a local reduced-order approximation for the solution space. The local snapshot functions and spectral decomposition of the snapshot space were constructed to approximate the solution in each coarse patch. This facilitated the transfer of the local properties of the solution into a global coarse-grid problem in a systematic way that substantially reduced the number of coupled degrees of freedom in the algebraic system. Since the solutions are discontinuous along the coarse grid interfaces, the local snapshot solutions were computed separately on each coarse block.

In this thesis, we analyze multiscale model reduction techniques within the framework of the Hybridizable Discontinuous Galerkin method (HDG). This method, as outlined in [19], allows sufficient prospects of "gluing" various finite element approximations together. This mechanism is made possible due to the notion of numerical trace and numerical flux. Numerical trace is a single valued function on the finite element interfaces and belongs to a certain Lagrange multiplier

space which is used to solve the global problem. The well-posedness and accuracy are ensured by a proper choice of the numerical flux, that involves a stabilization parameter τ [19]. Standard approaches for selection numerical traces involve the use of piecewise polynomials. Our approach relies on multiscale basis functions that are constructed independently (and in an offline stage) on each coarse-grid cell. Then, the coupling of these local functions is accomplished by the HDG method.

2.2 STANDARD FINITE ELEMENT METHOD

Finite element method (FEM) is a general technique for numerical solution of differential and integral equations in science and engineering. It was introduced by engineers in the late 50's and early 60's for the numerical solution of partial differential equations in structural engineering (elasticity equations, plate equations, etc). The workflow in FEM is as follows

1. Variational formulation of the given problem
2. Discretization using FEM: Construction of finite dimensional space V_h
3. Solution of the discrete problem
4. Implementation of the method on a computer: programming.

The idea of forming a variational formulation is deeply rooted in mathematical physics or to be more specific, calculus of variation – seeking the minimization point of a functional. Similar to calculus of one variable functions where the minimization problem is solved by looking for the critical point, the point where the derivative of the function is zero, the solution of minimization of a functional is given by the function in the underlying function space that satisfies exactly

the variational formulation. We illustrate this and the process of FEM using the following one dimensional example:

$$\begin{cases} -u''(x) = f(x) & \text{for } 0 < x < 1 \\ u(0) = u(1) = 0 \end{cases}$$

where $v' = \frac{dv}{dx}$ and f is a given continuous function. By integrating the equation $-u''(x) = f(x)$ twice, it is easy to see that this problem has a unique solution u . And to show that the solution u is also the solution of a minimization problem (M) and a variational problem (V), we will introduce the following notation

$$(v, w) = \int_0^1 v(x)w(x)dx$$

for real-valued piecewise continuous bounded functions. We also introduce the linear space $V = \{v: v \text{ is a continuous function on } [0,1], v' \text{ is piecewise continuous and bounded on } [0,1], \text{ and } v(0) = v(1)=0 \}$, and the linear functional $F : V \rightarrow \mathbb{R}$ is given by

$$F(v) = \frac{1}{2}(v', v') - (f, v).$$

The minimization (M) and variational (V) problems are as follows

$$(M) \quad \text{Find } u \in V \text{ such that } F(u) \leq F(v) \quad \forall v \in V,$$

$$(V) \quad \text{Find } u \in V \text{ such that } (u', v') = (f, v) \quad \forall v \in V,$$

By multiplying $-u''(x) = f(x)$ by an arbitrary test function $v \in V$, and integrating by parts over the interval $(0,1)$, we have

$$-(u'', v) = - \int_0^1 u'' v dx = -[u' v]_0^1 - \int_0^1 u' v' dx = u' v|_0^1 + \int_0^1 u' v' dx$$

using the fact that $v(0) = v(1) = 0$,

$$= -u'(1)v(1) + u'(0)v(0) + (u', v') = (u', v')$$

Thus, we can conclude that $(u', v') = (f, v) \quad \forall v \in V$,

We shall now construct a finite-dimensional subspace V_h of the space V defined above consisting of piecewise linear functions. To this end let $0 = x_0 < x_1 \dots < x_M < x_{M+1} = 1$, be a partition of the interval $(0, 1)$ into subintervals $I_j = (x_{j-1}, x_j)$ of length $h_j = x_j - x_{j-1}$, $j = 1, \dots, M+1$, where the quantity h is then a measure of how fine the partition is. Let V_h be defined as the set of functions v such that v is a linear function on each I_j , v is continuous on $[0, 1]$ and $v(0) = v(1) = 0$. We observe that $V_h \subset V$ and the basis functions ψ_i of V_h can be defined as follows

$$\psi_i(x_j) = \begin{cases} 1, & i = j \\ 0, & i \neq j, \quad i, j = 1 \dots M \end{cases}$$

i.e., ψ_i is the continuous piecewise linear function that takes the value 1 at node point x_j and the value 0 at other node points. See Figure 1

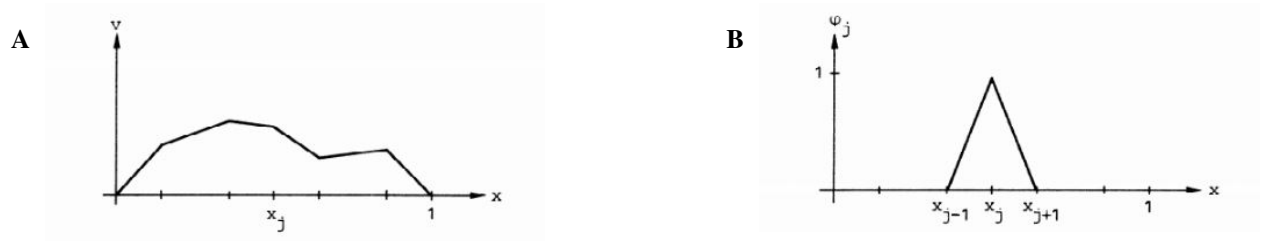


Figure 1. Functions in finite dimensional space. (A) Representation of function $v \in V_h$ by unit basis functions. (B) Piecewise linear function ψ_i . (Reproduced from [14])

A function $v \in V_h$ can then be represented as

$$v(x) = \sum_{i=1}^{n-1} \xi_i \psi_i(x), \quad x \in [0, 1]$$

where $\xi_i = v(x_i)$, ie, each $v \in V_h$ can be written in a unique way as a linear combination of the basis functions ψ_i . In particular, V_h is a linear space of dimension M with basis $\{\psi_i\}_{i=1}^M$. The finite element method for the boundary value problem can now be formulated as follows. Using the variational form (V) and the finite dimensional space V_h , we find $u_h \in V_h$ such that

$$(u'_h, v') = (f, v) \quad \forall v \in V_h$$

If this is true for all $v \in V_h$, then it's also true for the basis function $\psi_i \in V_h$.

$$(u'_h, \psi'_j) = (f, \psi_j) \quad j = 1, \dots, n-1$$

Then by taking a linear combination of the basis functions, and substituting into the above equation, we have

$$\begin{aligned} \left(\left(\sum_{i=1}^{n-1} \xi^u \psi_i(x) \right)', \psi'_j \right) &= (f, \psi_j) \\ \left(\left(\sum_{i=1}^{n-1} \xi^u \psi_i(x) \right)', \psi'_j \right) &= \int_0^1 \left(\sum_{i=1}^{n-1} \xi^u \psi_i(x) \right)' \psi'_j dx \\ &= \int_0^1 \sum_{i=1}^{n-1} (\xi^u \psi_i(x))' \psi'_j dx \\ &= \int_0^1 \sum_{i=1}^{n-1} \xi^u (\psi_i(x))' \psi'_j dx \\ &= \sum_{i=1}^{n-1} \xi^u \int_0^1 (\psi_i(x))' \psi'_j dx \\ &= \sum_{i=1}^{n-1} \xi^u (\psi'_i, \psi'_j) \\ \sum_{i=1}^{n-1} \xi^u (\psi'_i, \psi'_j) &= (f, \psi_j) \end{aligned}$$

where

$$u_h(x) = \sum_{i=1}^{n-1} \xi^u \psi_i(x)$$

This forms a linear system with M equations in M unknowns ξ_1, \dots, ξ_M which can be written as

$$A\xi = B$$

where A is the $M \times M$ matrix with elements $(a_{ij}) = (\psi'_i, \psi'_j)$, $\xi = (\xi_1, \dots, \xi_M)$ and $b = (b_1, \dots, b_M)$

with $b_i = (f, \psi_i)$ are M-vectors:

$$A = \begin{bmatrix} a_{11} & \cdot & \cdot & a_{1M} \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ a_{M1} & \cdot & \cdot & a_{MM} \end{bmatrix}, \xi = \begin{bmatrix} \xi_1 \\ \cdot \\ \cdot \\ \xi_M \end{bmatrix}, b = \begin{bmatrix} b_1 \\ \cdot \\ \cdot \\ b_M \end{bmatrix}$$

The matrix A is called the *stiffness matrix* and b the *load vector*.

For a two dimensional space, consider the Poisson's equation

$$\begin{aligned} -\nabla \cdot (\kappa \nabla u) &= f & \text{in } \Omega, \\ u &= 0 & \text{on } \partial\Omega. \end{aligned}$$

where $\Omega = [0, 1] \times [0, 1]$, $f \in L^2(\Omega)$. κ is a positive definite tensor. The weak formulation is to seek $u \in V = H_0^1(\Omega)$, such that

$$a(u, v) = l(v), \quad \forall v \in H_0^1(\Omega),$$

here

$$a(u, v) = \int_{\Omega} \kappa \nabla u \nabla v dx, \quad l(v) = \int_{\Omega} f v dx.$$

Then, by continuous Galerkin FEM method, we seek $u_h \in V_h \subset V$ such that

$$a(u_h, v) = l(v), \quad \forall v \in V_h.$$

where the finite dimensional function space V_h is define as

$$V_h := \{v \in C^0(\Omega), v|_{\partial\Omega} = 0, v|_K \in P_1(K), \text{ for all } K \in \mathcal{T}_h\}.$$

2.2.1 Finite Element Spaces

We shall now present some commonly used finite element spaces V_h . These spaces will consist of piecewise polynomial functions on subdivisions or "triangulations" $\mathcal{T}_h = \{K\}$ of a bounded domain $\Omega \subset \mathbb{R}^d, d = 1, 2, 3$, into elements K . For $d = 1$, the elements K will be intervals, for $d = 2$, triangles or quadrilaterals (see Figure 2) and for $d = 3$ tetrahedrons for instance. We will need to satisfy either $V_h \subset H^1(\Omega)$ or $V_h \subset H^2(\Omega)$, corresponding to second order or fourth order boundary value problems, respectively. Since the space V_h consists of piecewise polynomials, we have

$$V_h \subset H^1(\Omega) \iff V_h \subset C^0(\bar{\Omega})$$

$$V_h \subset H^2(\Omega) \iff V_h \subset C^1(\bar{\Omega})$$

where $\bar{\Omega} = \Omega \cup \Gamma$ and

$$C^0(\bar{\Omega}) = \{v : v \text{ is a continuous function defined on } \bar{\Omega}\}$$

$$C^1(\bar{\Omega}) = \{v \in C^0(\bar{\Omega}) : D^\alpha v \in C^0(\bar{\Omega}), \quad |\alpha| = 1\}$$

Thus, $V_h \subset H^1(\Omega)$ if and only if the functions $v \in V_h$ are continuous, and $V_h \subset H^2(\Omega)$ if and only if the functions $v \in V_h$ and their first derivatives are continuous. The equivalence depends on the fact that the functions v in V_h are polynomials on each element K so that if v is continuous across

the common boundary of adjoining elements, then the first derivatives $D^\alpha v$, $|\alpha| = 1$, exist and are piecewise continuous so that $v \in H^1(\Omega)$. On the other hand, if v is not continuous across a certain inter-element boundary, i.e., $v \notin C^0(\bar{\Omega})$, then the derivatives $D^\alpha v$, $|\alpha| = 1$, do not exist as functions in $L_2(\Omega)$ and thus $v \notin H^1(\Omega)$ (if v is discontinuous across an element side S , then $D^\alpha v$, $|\alpha| = 1$, would be a σ -function supported by S which is not a square-integrable function). To define a finite element space V_h , we will have to specify:

1. the triangulation $T_h = \{K\}$ of the domain Ω ,
2. the nature of the functions $v \in V_h$ on each K (eg linear, quadratic, cubic, etc),
3. the parameters to be used to describe the functions in V_h .

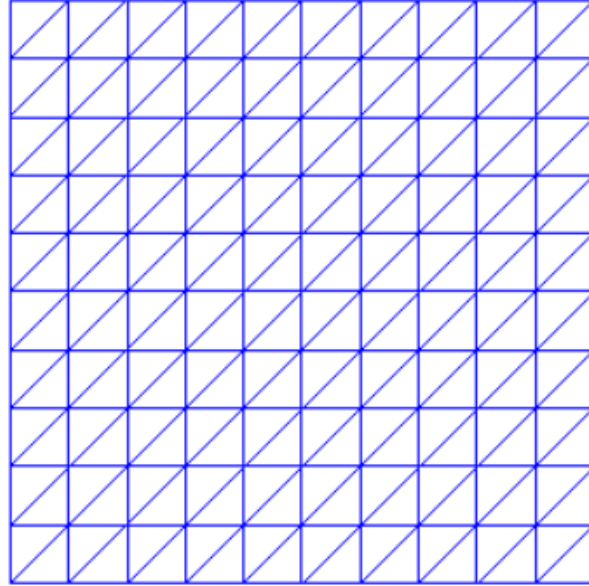


Figure 2. Triangulation \mathcal{T}_h with mesh size h comprising of finite elements.

2.2.2 Examples of Finite Elements

Let us consider some examples of finite elements. First, we examine the case when Ω is a domain in the plane \mathbb{R}^2 with polygonal boundary Γ . Let $T_h = \{K\}$ be a given triangulation of Ω into triangles K . We shall use the following notation for $r = 0, 1, 2, \dots$,

$$P_r(K) = \{v : v \text{ is a polynomial of degree } \leq r \text{ on } K\}$$

Thus, $P_1(K)$ is the space of linear functions defined on K , i.e., functions of the form

$$v(x) = a_{00} + a_{10}x_1 + a_{01}x_2, \quad x \in K$$

where $a_{ij} \in \mathbb{R}$. We see that $\{\psi_1, \psi_2, \psi_3\}$, where

$$\psi_1(x) \equiv 1, \psi_2(x) \equiv x_1, \psi_3(x) \equiv x_2,$$

is a basis for $P_1(K)$, and that $\dim P_1(K) = 3$, where $\dim W$ denotes the dimension of the linear space W . Further, $P_2(K)$ is the space of quadratic functions on K , i.e., functions of the form

$$v(x) = a_{00} + a_{10}x_1 + a_{01}x_2 + a_{20}x_1^2 + a_{11}x_1x_2 + a_{02}x_2^2, \quad x \in K$$

where $a_{ij} \in \mathbb{R}$. We see that $\{1, x_1, x_2, x_1^2, x_1x_2, x_2^2\}$ is a basis for $P_2(K)$ and that $\dim P_2(K) = 6$. In general, we have

$$P_r(K) = \{v : v(x) = \sum_{0 \leq i+j \leq r} a_{ij}x_1^i x_2^j \text{ for } x \in K, \text{ where } a_{ij} \in \mathbb{R}\},$$

and

$$\dim P_r(K) = \frac{(r+1)(r+2)}{2}$$

Example 1. Let

$$V_h = \{v \in C^0(\bar{\Omega}) : v|_K \in P_1(K), \quad \forall K \in T_h\}$$

i.e., V_h is the space of continuous piecewise linear functions. As parameters, or *global degrees of freedom*, to describe the functions in V_h , we choose the values at the node points of T_h (including the node points on Γ). Notice that if $K \in T_h$ is a triangle with vertices $a^i, i = 1, 2, 3$, then the degrees of freedom for K corresponds to the values at the vertices $a^i, i = 1, 2, 3$.

Example 2. Our next example is the following space

$$V_h = \{v \in C^0(\bar{\Omega}) : v|_K \in P_2(K), \quad \forall K \in T_h\}$$

i.e., V_h is the space of continuous piecewise quadratic functions where the *global degrees of freedom* of the functions $v \in V_h$ can be chosen as follows:

1. the values of v at the nodes of T_h
2. the values of v at the mid points of all the sides of the triangles in T_h

Example 3.

$$V_h = \{v \in C^0(\bar{\Omega}) : v|_K \in P_3(K), \quad \forall K \in T_h\}$$

i.e., V_h is the space of continuous piecewise cubic functions with *global degrees of freedom*

1. the values of v at the nodes of T_h
2. the values of v at the mid points a^{ij} on the sides of T_h
3. the values of v at the center of gravity for all $K \in T_h$.

We define a finite element to mean a triple (K, P_K, Σ) , where

K is a geometric object, for example a triangle,

P_K is a finite-dimensional linear space of functions defined on K

Σ is a set of degrees of freedom

such that a function $v \in P_K$ is uniquely determined by the degrees of freedom Σ . Below are some of the most common finite elements with various degrees of freedom.

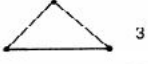

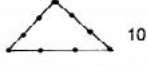
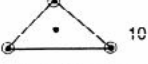
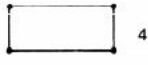

Degrees of freedom Σ Geometry	Function space P_K	Degree of continuity of corresponding FEM-space V_h
 3	$P_1(K)$	C^0
 6	$P_2(K)$	C^0
 10	$P_3(K)$	C^0
 10	$P_3(K)$	C^0
 4	$Q_1(K)$	C^0
 9	$Q_2(K)$	C^0

Figure 3. Examples of some common finite elements.(Reproduced from [14])

2.3 PRINCIPLE OF HYBRIDIZATION

Discontinuous Galerkin (DG) methods have been investigated and applied to a wide variety of problems. Initially introduced in Reed and Hill [35], these techniques have become popular beyond their original applications in fluid dynamics or electromagnetic problems. They provide a

natural stabilization to the solution due to inter-element fluxes. However, in recent years, hybridizable discontinuous Galerkin (HDG) methods have become more popular. As noted by Arnold and Brezzi [11], hybridization of DG methods derives from the mixed methods of Raviart and Thomas [34], where the continuity constrain is eliminated from the finite element space and imposed by means of Lagrange multipliers on the inter-element boundaries. This idea was further exploited by Cockburn and Gopalakrishnan [18] and Cockburn et al. [20] to formally develop the HDG method for second-order elliptic problems. In the HDG method, a numerical trace \hat{u}_h is introduced to approximate the trace of a solution besides u_h , which is the new unknown and may be called the hybrid unknown. Eliminating the unknown u_h by the hybrid unknown \hat{u}_h , we obtain a discretized equation in terms of \hat{u}_h only. As a result, the number of DOF of the HDG method can be considerably reduced, an advantage of HDG method over the DG method. Furthermore, HDG method has amazing features such as superconvergence properties and various connections with other numerical methods (mixed and nonconforming finite element methods, etc). The basic ideas of domain decomposition can be carried out in the following fashion.

Consider the Poisson equation on a region Ω , in two or three dimensions, with zero Dirichlet data given on the boundary $\partial\Omega$ of the region Ω . Also, suppose that Ω is partitioned into two nonoverlapping subdomains Ω_i (see Figure 4) such that

$$\overline{\Omega} = \overline{\Omega_1 \cup \Omega_2}, \quad \Omega_1 \cap \Omega_2 = \emptyset, \quad \Gamma = \partial\Omega_1 \cap \partial\Omega_2$$

We further assume that the boundaries of the subdomains are Lipschitz continuous.

$$\begin{aligned} -\nabla \cdot (\kappa \nabla u) &= f && \text{in } \Omega, \\ u &= 0 && \text{on } \partial\Omega. \end{aligned} \tag{1}$$

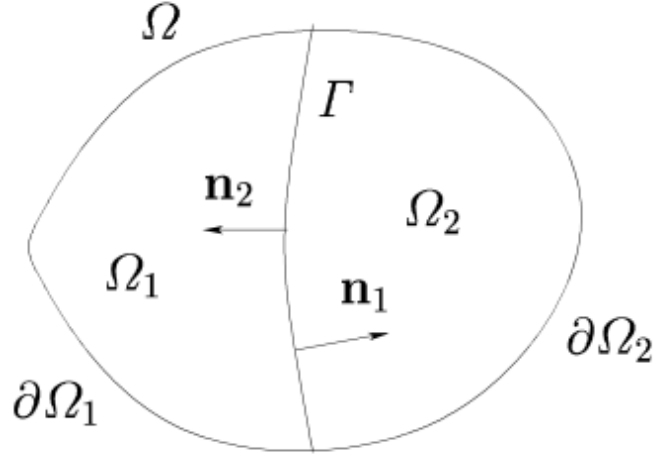


Figure 4. Partition of Ω into two nonoverlapping subdomains Ω_1 and Ω_2 . (Reproduced from [15])

Under suitable regularity assumptions on f (square-summable) and the boundaries of the subdomains, equation 1 is equivalent to the following coupled problem:

$$\begin{aligned}
 -\nabla \cdot (\kappa \nabla u_1) &= f && \text{in } \Omega_1, \\
 u_1 &= 0 && \text{on } \partial\Omega_1 \setminus \Gamma. \\
 -\nabla \cdot (\kappa \nabla u_2) &= f && \text{in } \Omega_2, \\
 u_2 &= 0 && \text{on } \partial\Omega_2 \setminus \Gamma. \\
 u_1 &= u_2 && \text{on } \Gamma, \\
 \frac{\partial u_1}{\partial \mathbf{n}_1} + \frac{\partial u_2}{\partial \mathbf{n}_2} &= 0 && \text{on } \Gamma.
 \end{aligned} \tag{2}$$

where u_i is the restriction of u to Ω_i and \mathbf{n}_i the unit outward normal to Ω_i . We refer to the normal derivatives as the flux. The conditions imposed on the interface Γ are called transmission condi-

tions. Equivalently, the above problem can be reformulated as finding \hat{u}_h on Γ such that

$$\begin{aligned} -\nabla \cdot (\kappa \nabla u) &= f && \text{in } \Omega_i, \\ u &= \hat{u} && \text{on } \partial\Omega_i, \\ \left[\frac{\partial u}{\partial \mathbf{n}}\right] &= 0 && \text{on } \Gamma. \end{aligned} \tag{3}$$

where

$$\left[\frac{\partial u}{\partial \mathbf{n}}\right] := \frac{\partial u^+}{\partial \mathbf{n}} + \frac{\partial u^-}{\partial \mathbf{n}} = 0 \quad \text{on } \Gamma.$$

CHAPTER 3

PROBLEM STATEMENT

In this thesis, we consider the following second-order elliptic differential equation defined on a bounded polyhedral domain Ω in \mathbb{R}^n , $n = 2, 3$.

$$-\nabla \cdot (\kappa(x) \nabla u) = f(x), \quad x \in \Omega \quad (4)$$

where $\kappa(x) \geq \kappa_0 > 0$ represents the permeability coefficient of a highly heterogeneous porous media with multiple scales. Application of equation 4 can be seen in different areas such as flows in porous media, diffusion and transport of passive chemicals or heat transfer in heterogeneous media. We shall now present the HDG method for equation 4 on a fine-grid \mathcal{T}_h . Equation 4 can be rewritten as a first order PDE with homogeneous Dirichlet boundary condition by introducing a new variable \mathbf{q} .

$$\alpha \mathbf{q} + \nabla u = 0 \quad \text{in } \Omega, \quad (5a)$$

$$\nabla \cdot \mathbf{q} = f \quad \text{in } \Omega, \quad (5b)$$

$$u = 0 \quad \text{on } \partial\Omega. \quad (5c)$$

where $\alpha(x) = \kappa(x)^{-1}$, $\Omega \subset \mathbb{R}^n$ ($n = 2, 3$) is a bounded polyhedral domain and $f \in L^2(\Omega)$. Consider a partitioning of the domain Ω into elements K forming a mesh \mathcal{T}_h that satisfy the standard finite element conditions. Denote also, the faces of K by F . Then, the method yields a scalar approximation u_h to u , a vector approximation \mathbf{q}_h to \mathbf{q} and a scalar approximation \hat{u}_h to the trace of u on the boundaries of each element. We also have the following finite dimension spaces of the

form

$$W_h := \{w \in L^2(\mathcal{T}_h) : w|_K \in W(K), K \in \mathcal{T}_h\},$$

$$\mathbf{V}_h := \{\mathbf{r} \in \mathbf{L}^2(\mathcal{T}_h) : \mathbf{r}|_K \in \mathbf{V}(K), K \in \mathcal{T}_h\},$$

$$M_h := \{\mu \in L^2(\mathcal{T}_h) : \mu|_F \in M_h(F)\},$$

where the transmission condition or the numerical trace is defined as

$$\hat{\mathbf{q}}_h \cdot \mathbf{n} = \mathbf{q}_h \cdot \mathbf{n} + \tau(u_h - \hat{u}_h) \quad \text{on } \partial\mathcal{T}_h. \quad (6)$$

3.1 DERIVATION OF THE HDG SCHEME

We multiply 5a by $\mathbf{r} \in \mathbf{V}_h$ and equation 5b by $w \in W_h$ and perform integration-by-parts. Then, the HDG method reads as follows: find $(u_h, \mathbf{q}_h, \hat{u}_h)$ in the space $W_h \times \mathbf{V}_h \times M_h$ such that the following weak problem is satisfied.

$$(\alpha \mathbf{q}_h, \mathbf{r})_{\mathcal{T}_h} - (u_h, \nabla \cdot \mathbf{r})_{\mathcal{T}_h} + \langle \hat{u}_h, \mathbf{r} \cdot \mathbf{n} \rangle_{\partial\mathcal{T}_h} = 0 \quad \forall \mathbf{r} \in \mathbf{V}_h, \quad (7a)$$

$$-(\mathbf{q}_h, \nabla w)_{\mathcal{T}_h} + \langle \hat{\mathbf{q}}_h \cdot \mathbf{n}, w \rangle_{\partial\mathcal{T}_h} = (f, w)_{\mathcal{T}_h} \quad \forall w \in W_h, \quad (7b)$$

$$\langle \hat{\mathbf{q}}_h \cdot \mathbf{n}, \mu \rangle_{\partial\mathcal{T}_h} = 0 \quad \forall \mu \in M_h, \quad (7c)$$

$$\hat{u}_h = 0 \quad \text{on } \partial\Omega. \quad (7d)$$

By substitute 6 into 7b

$$(\alpha \mathbf{q}_h, \mathbf{r})_{\mathcal{T}_h} - (u_h, \nabla \cdot \mathbf{r})_{\mathcal{T}_h} + \langle \hat{u}_h, \mathbf{r} \cdot \mathbf{n} \rangle_{\partial\mathcal{T}_h} = 0 \quad \forall \mathbf{r} \in \mathbf{V}_h, \quad (8a)$$

$$-(\mathbf{q}_h, \nabla w)_{\mathcal{T}_h} + \langle \mathbf{q}_h \cdot \mathbf{n} + \tau(u_h - \hat{u}_h), w \rangle_{\partial\mathcal{T}_h} = (f, w)_{\mathcal{T}_h} \quad \forall w \in W_h, \quad (8b)$$

$$\langle \hat{\mathbf{q}}_h \cdot \mathbf{n}, \mu \rangle_{\partial\mathcal{T}_h} = 0 \quad \forall \mu \in M_h, \quad (8c)$$

$$\hat{u}_h = 0 \quad \text{on } \partial\Omega. \quad (8d)$$

From 8b we observe that

$$-(\mathbf{q}_h, \nabla w)_{\mathcal{T}_h} + \langle \mathbf{q}_h \cdot \mathbf{n}, w \rangle_{\partial \mathcal{T}_h} = (\nabla \cdot \mathbf{q}_h, w)_{\mathcal{T}_h}.$$

Then, 8 can be reformulated as follows

$$(\alpha \mathbf{q}_h, \mathbf{r})_{\mathcal{T}_h} - (u_h, \nabla \cdot \mathbf{r})_{\mathcal{T}_h} = -\langle \hat{u}_h, \mathbf{r} \cdot \mathbf{n} \rangle_{\partial \mathcal{T}_h} \quad \forall \mathbf{r} \in \mathbf{V}_h, \quad (9a)$$

$$(\nabla \cdot \mathbf{q}_h, w)_{\mathcal{T}_h} + \langle \tau u_h, w \rangle_{\partial \mathcal{T}_h} = -\langle \tau \hat{u}_h, w \rangle_{\partial \mathcal{T}_h} + (f, w)_{\mathcal{T}_h} \quad \forall w \in W_h, \quad (9b)$$

$$\langle \hat{\mathbf{q}}_h \cdot \mathbf{n}, \mu \rangle_{\partial \mathcal{T}_h} = 0 \quad \forall \mu \in M_h, \quad (9c)$$

$$\hat{u}_h = 0 \quad \text{on } \partial \Omega. \quad (9d)$$

Since the spaces $W_h \times \mathbf{V}_h \times M_h$ are finite dimensional, we can express every function $(u_h, \mathbf{q}_h, \hat{u}_h)$ in $W_h \times \mathbf{V}_h \times M_h$ as a linear combination of polynomial shape functions of order p .

$$\mathbf{q}_h = \sum_{i=1} \mathbf{q}_i \phi_i \quad \in \mathbf{V}_h, \quad (10a)$$

$$u_h = \sum_{i=1} u_i \phi_i \quad \in W_h, \quad (10b)$$

$$\hat{u}_h = \sum_{i=1} \hat{u}_i \phi_i \quad \in M_h. \quad (10c)$$

where \mathbf{q}_i , u_i and \hat{u}_i are nodal values on each element, ϕ_i are polynomial shape functions of order p in each element. Given the element-by-element formulation, the vector \hat{u}_i is defined globally over the entire mesh skeleton (face/edges). This allows the trace to be defined in such a way that both \mathbf{q}_i , u_i can be eliminated from equation 9 to give rise to a single equation for \hat{u}_i . By substituting equation 10 into equation 9, we can obtain the following matrices.

$$\mathbb{A}_1 \begin{bmatrix} \mathbf{q}_h^i \\ u_h^i \end{bmatrix} = \mathbb{A}_f - \mathbb{A}_2 \hat{u}_h^i. \quad (11)$$

where \mathbb{A}_1 is the mass matrix within each element, \mathbb{A}_f the vector containing the forcing term and \mathbb{A}_2 the edge integrals. Using the transmission condition equation 6 and equation 10, we can eliminate \mathbf{q}_h^i, u_h^i in equation 11 to obtain a global system. From 11,

$$\begin{aligned}
 \langle \hat{\mathbf{q}}_h \cdot \mathbf{n}, w \rangle_{\partial \mathcal{T}_h} &= \langle \mathbf{q}_h \cdot \mathbf{n} + \tau(u_h - \hat{u}_h), w \rangle_{\partial \mathcal{T}_h} = 0, \\
 &= \mathbb{A}_3 \begin{bmatrix} \mathbf{q}_h^i \\ u_h^i \end{bmatrix} - \hat{\mathbb{D}} \hat{u}_h^i = 0. \\
 \mathbb{A}_3 \begin{bmatrix} \mathbf{q}_h^i \\ u_h^i \end{bmatrix} &= \hat{\mathbb{D}} \hat{u}_h^i. \tag{12}
 \end{aligned}$$

Substitute equation 11 into equation 12 for $\begin{bmatrix} \mathbf{q}_h^i \\ u_h^i \end{bmatrix}$,

$$\begin{aligned}
 \mathbb{A}_3(\mathbb{A}_1^{-1} \mathbb{A}_f - \mathbb{A}_1^{-1} \mathbb{A}_2 \hat{u}_h^i) &= \hat{\mathbb{D}} \hat{u}_h^i, \\
 \mathbb{A}_3 \mathbb{A}_1^{-1} \mathbb{A}_f - \mathbb{A}_3 \mathbb{A}_1^{-1} \mathbb{A}_2 \hat{u}_h^i &= \hat{\mathbb{D}} \hat{u}_h^i, \\
 \mathbb{A}_3 \mathbb{A}_1^{-1} \mathbb{A}_f &= (\mathbb{A}_3 \mathbb{A}_1^{-1} \mathbb{A}_2 + \hat{\mathbb{D}}) \hat{u}_h^i, \\
 \mathbb{C}_f &= \mathbb{M} \hat{u}_h^i.
 \end{aligned}$$

$$\hat{u}_h^i = \mathbb{M}^{-1} \mathbb{C}_f. \tag{13}$$

This gives us a global system for determining \hat{u}_h^i on the edges of the elements. Once this is obtained, we perform a back substitution into equation 11 to determine $\begin{bmatrix} \mathbf{q}_h^i \\ u_h^i \end{bmatrix}$ on the interior of each element. We may view the values of \hat{u}_h^i as the Dirichlet boundary data. Together, they form local solvers within the triangulation \mathcal{T}_h .

CHAPTER 4

MULTISCALE HDG

In this thesis, we consider a multiscale model reduction technique within the framework of hybridizable discontinuous Galerkin finite element method. This approach uses local snapshot spaces derived from two-grid approximation to avoid high dimensional representation of trace spaces. For this purpose, we shall consider as before, equation 5.

$$\begin{aligned}\alpha \mathbf{q} + \nabla u &= 0 && \text{in } \Omega, \\ \nabla \cdot \mathbf{q} &= f && \text{in } \Omega, \\ u &= 0 && \text{on } \partial\Omega.\end{aligned}$$

where $\alpha(x) = \kappa(x)^{-1}$, $\Omega \subset \mathbb{R}^n$ ($n = 2, 3$) is a bounded polyhedral domain and $f \in L^2(\Omega)$.

4.1 FINE AND COARSE GRIDS

We partition the domain Ω , as proposed by [24], into disjoint polygonal subdomains $\{\Omega_i\}_{i=1}^N$ with maximum diameter H_i , see e.g Figure 5. Denote by F_H , the face or edge of the subdomain Ω_i , if F_H is either shared by Ω_i or its neighboring subdomain Ω_j , i.e. $F_H = \overline{\Omega_i} \cap \overline{\Omega_j}$ or $F_H = \overline{\Omega_i} \cap \partial\Omega$. Then we shall denote the set of all coarse edges of a subdomain Ω_i as ξ_H , where $\xi_H = \cup_{i=1}^N \xi_H(\Omega_i)$. Furthermore, let $\mathcal{T}_h(\Omega_i)$ be a shape regular triangulations with triangular elements of maximum mesh-size h_i . Edges (faces) of this triangulation will be denoted by F_h . Let $\mathcal{T}_h = \cup_{i=1}^N \mathcal{T}_h(\Omega_i)$, $\xi_h(\Omega_i)$, be the set of all faces of the triangulation $\mathcal{T}_h(\Omega_i)$ and $\xi_h^0(\Omega_i)$ be the set of all interior edges of the triangulation $\mathcal{T}_h(\Omega_i)$, and set $\xi_h = \cup_{i=1}^N \xi_h(\Omega_i)$.

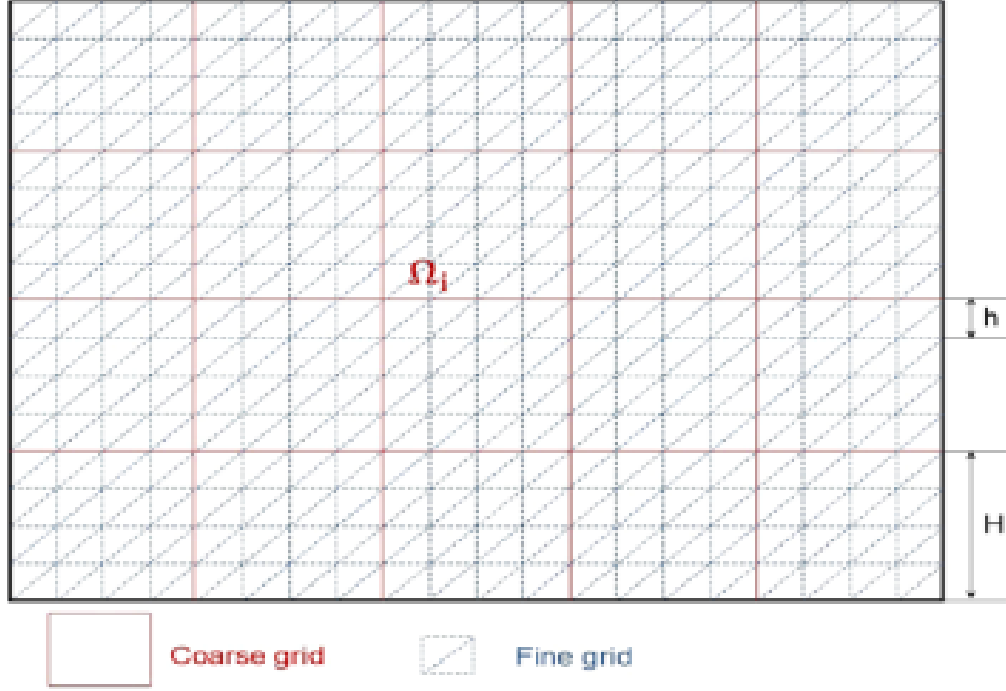


Figure 5. Fine (interior of red squares) and coarse grids.

Associated with this method are the following finite element spaces consisting of piece-wise polynomial functions.

$$W_h := \{w \in L^2(\mathcal{T}_h) : w|_K \in W(K), K \in \mathcal{T}_h\},$$

$$\mathbf{V}_h := \{\mathbf{r} \in \mathbf{L}^2(\mathcal{T}_h) : \mathbf{r}|_K \in \mathbf{V}(K), K \in \mathcal{T}_h\},$$

$$M_{h,H} := M_h^0 \oplus M_H,$$

where the spaces M_h^0, M_H are defined as

$$M_h^0 := \{\mu \in L^2(\mathcal{E}_{h,H}) : \text{for } F \in \mathcal{E}_h^0 \mu|_F \in M_h(F), \text{ and } \mu|_{\mathcal{E}_H} = 0\},$$

$$M_H := \{\mu \in L^2(\mathcal{E}_{h,H}) : \text{for } F \in \mathcal{E}_H \mu|_F \in M_H(F), \text{ and } \mu|_{\mathcal{E}_h^0 \cup \partial\Omega} = 0\}.$$

4.2 GLOBAL FORMULATION

The multiscale HDG (MsHDG) method reads as follows: find $(u_h, \mathbf{q}_h, \hat{u}_{h,H})$ in the space $W_h \times \mathbf{V}_h \times M_{h,H}$ such that the following weak problem is satisfied.

$$(\alpha \mathbf{q}_h, \mathbf{r})_{\mathcal{T}_h} - (u_h, \nabla \cdot \mathbf{r})_{\mathcal{T}_h} + \langle \hat{u}_{h,H}, \mathbf{r} \cdot \mathbf{n} \rangle_{\partial \mathcal{T}_h} = 0 \quad \forall \mathbf{r} \in \mathbf{V}_h, \quad (14a)$$

$$-(\mathbf{q}_h, \nabla w)_{\mathcal{T}_h} + \langle \hat{\mathbf{q}}_{h,H} \cdot \mathbf{n}, w \rangle_{\partial \mathcal{T}_h} = (f, w)_{\mathcal{T}_h} \quad \forall w \in W_h, \quad (14b)$$

$$\langle \hat{\mathbf{q}}_{h,H} \cdot \mathbf{n}, \mu \rangle_{\partial \mathcal{T}_h} = 0 \quad \forall \mu \in M_{h,H}, \quad (14c)$$

$$\hat{u}_{h,H} = 0 \quad \text{on } \partial \Omega. \quad (14d)$$

For triangulation $\mathcal{T}_h(\Omega_i)$, we write $(\eta, \zeta)_{\mathcal{T}} := \sum_{K \in \mathcal{T}} (\eta, \zeta)_K$, where $(\eta, \zeta)_D$ denotes the integral of $\eta \zeta$ over the domain $D \subset \mathbb{R}^n$. We also write $\langle \eta, \zeta \rangle_{\partial \mathcal{T}} := \sum_{K \in \mathcal{T}} \langle \eta, \zeta \rangle_{\partial K}$, where $\langle \eta, \zeta \rangle_{\partial D}$ denotes the integral of $\eta \zeta$ over the boundary of the domain $D \subset \mathbb{R}^{n-1}$. We now complete the method with the definition of the transmission condition or the normal component of the numerical trace:

$$\hat{\mathbf{q}}_{h,H} \cdot \mathbf{n} = \mathbf{q}_h \cdot \mathbf{n} + \tau(u_h - \hat{u}_{h,H}) \quad \text{on } \partial \mathcal{T}_h. \quad (15)$$

where τ is non-negative and known as the stabilization parameter. To allow us to solve the global system on just the coarse mesh, we split the third equation of 14 by testing separately with $\mu \in M_h^0$ and $\mu \in M_H$ such that

$$\langle \hat{\mathbf{q}}_{h,H} \cdot \mathbf{n}, \mu \rangle_{\partial \mathcal{T}_h} = 0 \quad \forall \mu \in M_h^0 \quad \text{and} \quad \langle \hat{\mathbf{q}}_{h,H} \cdot \mathbf{n}, \mu \rangle_{\partial \mathcal{T}_h} = \langle \hat{\mathbf{q}}_{h,H} \cdot \mathbf{n}, \mu \rangle_{\partial \Omega_i} = 0 \quad \forall \mu \in M_H. \quad (16)$$

On any subdomain Ω_i , given the boundary data of $\hat{u}_{h,H} = \xi_H$ for $\xi_H \in M_H(F), F \in \mathcal{E}_H(\Omega)$, we can

solve for $(\mathbf{q}_h, u_h, \widehat{u}_{h,H})|_{\Omega_i}$ by restricting the equation 14 on this particular Ω_i :

$$(\alpha \mathbf{q}_h, \mathbf{r})_{\mathcal{T}_h(\Omega_i)} - (u_h, \nabla \cdot \mathbf{r})_{\mathcal{T}_h(\Omega_i)} + \langle \widehat{u}_{h,H}, \mathbf{r} \cdot \mathbf{n} \rangle_{\partial \mathcal{T}_h(\Omega_i)} = 0, \quad (17a)$$

$$-(\mathbf{v}_h, \nabla w)_{\mathcal{T}_h(\Omega_i)} + \langle \widehat{\mathbf{q}}_{h,H} \cdot \mathbf{n}, w \rangle_{\partial \mathcal{T}_h(\Omega_i)} = (f, w)_{\mathcal{T}_h(\Omega_i)}, \quad (17b)$$

$$\langle \widehat{\mathbf{q}}_{h,H} \cdot \mathbf{n}, \mu \rangle_{\partial \mathcal{T}_h(\Omega_i)} = 0, \quad (17c)$$

$$\widehat{u}_{h,H} = \xi_H \quad \text{on } \partial \Omega_i, \quad (17d)$$

for all $(w, \mathbf{r}, \mu) \in W_h|_{\Omega_i} \times \mathbf{V}_h|_{\Omega_i} \times M_h^0|_{\mathcal{E}_h^0(\Omega_i)}$. For a single coarse block, the above local system is the regular HDG methods defined on Ω_i . Moreover, from [19] we already know that this system is stable. Using the principle of superposition, equation 17 can be further split into two parts, namely,

$$(\mathbf{q}_h, u_h, \widehat{u}_{h,H}) = (\mathbf{q}_h(f), u_h(f), \widehat{u}_{h,H}(f)) + (\mathbf{q}_h(\xi_H), u_h(\xi_H), \widehat{u}_{h,H}(\xi_H)).$$

where $(\mathbf{q}_h(f), u_h(f), \widehat{u}_{h,H}(f))$ satisfies

$$\begin{aligned} (\alpha \mathbf{q}_h(f), \mathbf{r})_{\mathcal{T}_h(\Omega_i)} - (u_h(f), \nabla \cdot \mathbf{r})_{\mathcal{T}_h(\Omega_i)} + \langle \widehat{u}_{h,H}(f), \mathbf{r} \cdot \mathbf{n} \rangle_{\partial \mathcal{T}_h(\Omega_i)} &= 0, \\ -(\mathbf{q}_h(f), \nabla w)_{\mathcal{T}_h(\Omega_i)} + \langle \widehat{\mathbf{q}}_{h,H}(f) \cdot \mathbf{n}, w \rangle_{\partial \mathcal{T}_h(\Omega_i)} &= (f, w)_{\mathcal{T}_h(\Omega_i)}, \\ \langle \widehat{\mathbf{q}}_{h,H}(f) \cdot \mathbf{n}, \mu \rangle_{\partial \mathcal{T}_h(\Omega_i)} &= 0, \\ \widehat{u}_{h,H} &= 0 \quad \text{on } \partial T, \end{aligned}$$

for all $(w, \mathbf{r}, \mu) \in W_h|_{\Omega_i} \times \mathbf{V}_h|_{\Omega_i} \times M_h^0|_{\mathcal{E}_h^0(\Omega_i)}$ and $(\mathbf{q}_h(\xi_H), u_h(\xi_H), \widehat{u}_{h,H}(\xi_H))$ satisfies

$$\begin{aligned} (\alpha \mathbf{q}_h(\xi_H), \mathbf{r})_{\mathcal{T}_h(\Omega_i)} - (u_h(\xi_H), \nabla \cdot \mathbf{r})_{\mathcal{T}_h(\Omega_i)} + \langle \widehat{u}_{h,H}(\xi_H), \mathbf{r} \cdot \mathbf{n} \rangle_{\partial \mathcal{T}_h(\Omega_i)} &= 0, \\ -(\mathbf{q}_h(\xi_H), \nabla w)_{\mathcal{T}_h(\Omega_i)} + \langle \widehat{\mathbf{q}}_{h,H}(\xi_H) \cdot \mathbf{n}, w \rangle_{\partial \mathcal{T}_h(\Omega_i)} &= 0, \\ \langle \widehat{\mathbf{q}}_{h,H}(\xi_H) \cdot \mathbf{n}, \mu \rangle_{\partial \mathcal{T}_h(\Omega_i)} &= 0, \\ \widehat{u}_{h,H}(\xi_H) &= \xi_H \quad \text{on } \partial T, \end{aligned}$$

for all $(w, \mathbf{r}, \mu) \in W_h|_{\Omega_i} \times \mathbf{V}_h|_{\Omega_i} \times M_h^0|_{\mathcal{E}_h^0(\Omega_i)}$.

Given equation 16, we can follow a similar process outlined in equation 12 - 13 to obtain a global system for $\widehat{u}_{h,H}$. This implies that we only need to solve the global system on the coarse mesh, a process known as upscaling using the following equation:

$$a(\xi_H, \mu) = l(\mu) \quad \text{for all } \mu \in M_H, \quad (18)$$

where the bilinear form $a(\xi_H, \mu) : M_H \times M_H \rightarrow R$ and the linear form $l(\mu) : M_H \rightarrow R$ are defined as

$$a(\xi_H, \mu) := \langle \widehat{\mathbf{q}}_{h,H}(\xi_H) \cdot \mathbf{n}, \mu \rangle_{\partial\Omega_i} \quad \text{and} \quad l(\mu) := a(f, \mu) = \langle \widehat{\mathbf{q}}_{h,H}(f) \cdot \mathbf{n}, \mu \rangle_{\partial\Omega_i}. \quad (19)$$

In the next chapter, we shall construct orthogonal basis functions used in the space M_H .

CHAPTER 5

MULTISCALE ORTHOGONAL BASES FUNCTIONS

Multiscale basis functions are a very important component used to represent the solution at different scales within the computational domain. These basis functions are usually designed to capture the behavior of the problem at various scales, allowing for an accurate approximation of the solution across the entire domain. In MsFEM, they are constructed by combining local fine-scale solutions with global coarse-scale information. This integration enables the accurate modeling of features at different scales and facilitates the efficient computation of the overall solution. Furthermore, they are often chosen to be orthogonal to ensure that they capture essential characteristics of the problem while minimizing computational costs. They provide a flexible framework for representing the solutions in a multiscale fashion, making it possible to efficiently analyze complex systems with varying scales of behavior. Therefore, by using these basis functions, MsFEM can effectively handle problems with heterogeneous material properties, discontinuities, or other multiscale phenomena. In this chapter, we consider the construction of multiscale basis functions for the coarse space M_H

5.1 MULTISCALE FUNTIONS ON THE UNIT INTERVAL

Consider the simplest invariant set $\Omega = [0, 1]$. The advantage of the construction on this set is the recursive generation of partitions of Ω and multiscale bases based on partitions. The following procedures have been proposed and discussed extensively in [16]. Let $\Omega = [0, 1]$, be an invariant

set with the associated contractive mapping

$$\phi_\varepsilon(t) = \frac{\varepsilon + t}{2} \quad t \in \Omega, \quad \varepsilon \in \mathbb{Z}_2, \quad (20)$$

where

$$\Omega = \phi_0(\Omega) \cup \phi_1(\Omega) \quad \text{and} \quad \text{meas}(\phi_0(\Omega) \cap \phi_1(\Omega)) = 0.$$

and $\text{meas}(A)$ denotes the Lebesgue measure of the set A . Then, for each $\varepsilon \in \mathbb{Z}_2$ we set $\Omega_\varepsilon := [\frac{\varepsilon}{2}, \frac{\varepsilon+1}{2}]$ and define the isometry \mathcal{T}_ε such that for $f \in L^2(\Omega)$:

$$\mathcal{T}_\varepsilon f := \sqrt{2}(f \circ \phi_\varepsilon^{-1}) \chi_{\Omega_\varepsilon} = \begin{cases} \sqrt{2}f(2t - \varepsilon) & \text{for } t \in \Omega_\varepsilon, \\ 0 & \text{for } t \notin \Omega_\varepsilon. \end{cases} \quad (21)$$

where χ_A is the characteristic function of the set A . The application of the operator \mathcal{T}_ε to a function can be seen in Figure 6.

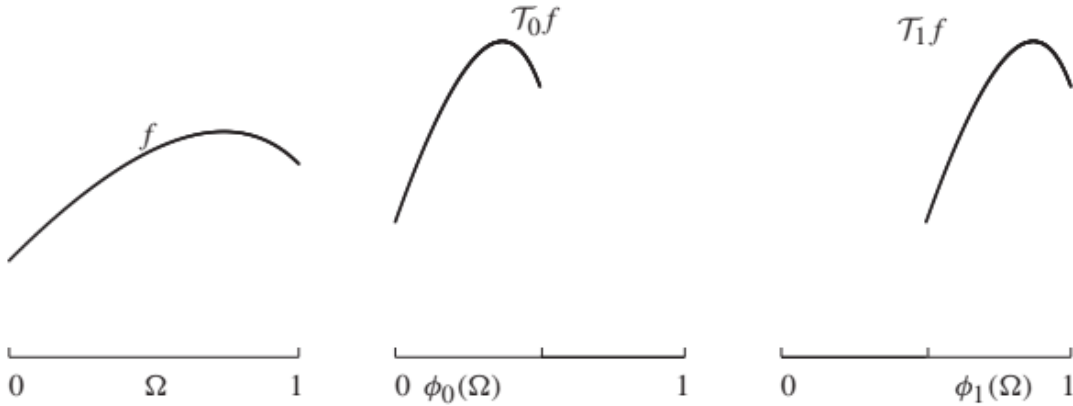


Figure 6. Application of the operator \mathcal{T}_ε to a function producing $\mathcal{T}_0 f$ and $\mathcal{T}_1 f$. (Reproduced from [16])

Moreover, given the space $\mathbb{X}_n = \mathbb{S}_{2^n}^k$, $n \in \mathbb{N}_0$, where $\mathbb{S}_1 \subseteq \mathbb{S}_2 \subseteq \dots \subseteq \mathbb{S}_{2^n}^k$, if X_0 is an orthonormal basis for \mathbb{S}_1^k , then

$$X_n := \bigcup_{\varepsilon \in \mathbb{Z}_2}^{\perp} \mathcal{T}_{\varepsilon} X_{n-1}, \quad n \in \mathbb{N} \quad (22)$$

is an orthonormal basis for $\mathbb{S}_{2^n}^k$. We refer the reader to [16] for a detailed proof.

5.2 CONSTRUCTION OF MULTISCALE BASIS

Let \mathbb{X}_n be a space such that $\mathbb{X}_{n-1} \subseteq \mathbb{X}_n$. By the theorem orthogonal decomposition, we have

$$\mathbb{X}_n = \mathbb{X}_{n-1} \oplus^{\perp} \mathbb{W}_n,$$

where \mathbb{W}_n is the orthogonal complement of \mathbb{X}_{n-1} in \mathbb{X}_n . This gives the decomposition for the space \mathbb{X}_n .

$$\mathbb{X}_n = \mathbb{X}_0 \oplus^{\perp} \mathbb{W}_1 \oplus^{\perp} \mathbb{W}_2 \oplus^{\perp} \dots \oplus^{\perp} \mathbb{W}_n,$$

with dimension $\mathbb{X}_n = k2^n$ and dimension $\mathbb{W}_n = k2^{n-1}$ for $k \geq 1$.

In order to construct multiscale orthonormal basis, we begin with the establishment of orthonormal basis W_j for the space \mathbb{W}_j for each $j \in \mathbb{N}_n$. First, we choose the Legendre polynomials of degree $\leq k-1$ on Ω as an orthonormal basis for $\mathbb{X}_0 = \mathbb{S}_1^k$ and denote these basis by X_0 . Then,

- (1) Use equation 22 to construct an orthonormal basis X_1 for the space \mathbb{X}_1 .

$$X_1 := \bigcup_{\varepsilon \in \mathbb{Z}_2}^{\perp} \mathcal{T}_{\varepsilon} X_0, \quad n \in \mathbb{N}$$

- (2) Form a linear combination of the basis functions in X_1 and require it to be orthogonal to all elements of X_0 through the process of Gram-Schmidt. This gives k linearly independent elements which are orthogonal to \mathbb{X}_0 .

(3) Orthonormalize these k functions and have them serve as an orthonormal basis for \mathbb{W}_1

For the construction of basis W_j , $j \geq 2$, we use the following results proposed in [16]. If W_1 is given as an orthonormal basis for \mathbb{W}_1 , then

$$W_{n+1} := \bigcup_{\varepsilon \in \mathbb{Z}_2}^{\perp} \mathcal{T}_{\varepsilon} W_n, \quad n \in \mathbb{N} \quad (23)$$

is an orthonormal basis for \mathbb{W}_{n+1} .

5.3 EXAMPLES OF MULTISCALE FUNCTIONS

We consider basis functions of the form $W_i := \{w_{ij} : j \in \mathbb{Z}_{w(i)}\}$ with double subscripts. The first subscript represents the level of the scale of the subspace and the second indicating the location of its support. In this work, we shall construct basis functions up to the third level. Since the decomposition for the space \mathbb{X}_n is given by

$$\mathbb{X}_n = \mathbb{X}_0 \oplus^\perp \mathbb{W}_1 \oplus^\perp \mathbb{W}_2 \oplus^\perp \dots \oplus^\perp \mathbb{W}_n,$$

for $n = 3$, we have,

$$\mathbb{X}_3 = \mathbb{X}_0 \oplus^\perp \mathbb{W}_1 \oplus^\perp \mathbb{W}_2 \oplus^\perp \mathbb{W}_3.$$

For the space \mathbb{X}_0 , we have the following orthonormal basis functions. See Figure 7.

$$\mathbb{X}_0 : \quad w_{00}(t) = 1, \quad w_{01}(t) = \sqrt{3}(2t - 1), \quad t \in [0, 1]$$

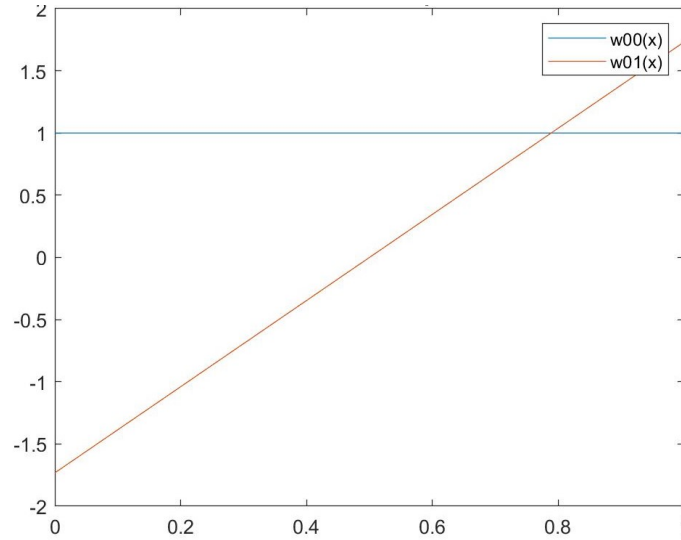


Figure 7. Orthonormal basis on \mathbb{X}_0 spanned by $w_{00}(t)$ and $w_{01}(t)$.

For the space \mathbb{W}_1 , we obtain following orthonormal basis functions through the process outlined in section 5.2. See Figure 8

$$w_{10}(t) = \begin{cases} 1 - 6t & \text{for } t \in [0, \frac{1}{2}], \\ 5 - 6t & \text{for } t \in (\frac{1}{2}, 1]. \end{cases}$$

$$w_{11}(t) = \begin{cases} \sqrt{3}(1 - 4t) & \text{for } t \in [0, \frac{1}{2}], \\ \sqrt{3}(4t - 3) & \text{for } t \in (\frac{1}{2}, 1]. \end{cases}$$

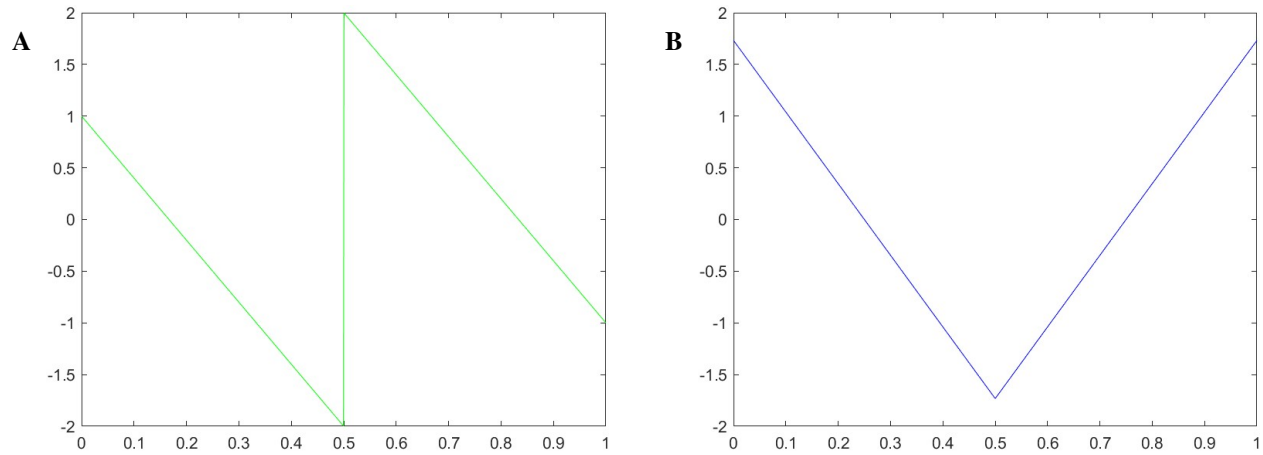


Figure 8. Orthonormal basis for \mathbb{W}_1 . (A) basis w_{10} . (B) basis w_{11} . (Reproduced from [16]).

Since we are furnished with the basis of \mathbb{W}_1 , we apply the operators \mathcal{T}_0 and \mathcal{T}_1 to \mathbb{W}_1 to obtain the basis functions for the space \mathbb{W}_2 . This can be seen in Figure 9.

$$\mathbb{W}_2 : \quad \mathcal{T}_0\{w_{10}, w_{11}\} \cup \mathcal{T}_1\{w_{10}, w_{11}\} := \text{span}\{w_{20}, w_{21}, w_{22}, w_{23}\}.$$

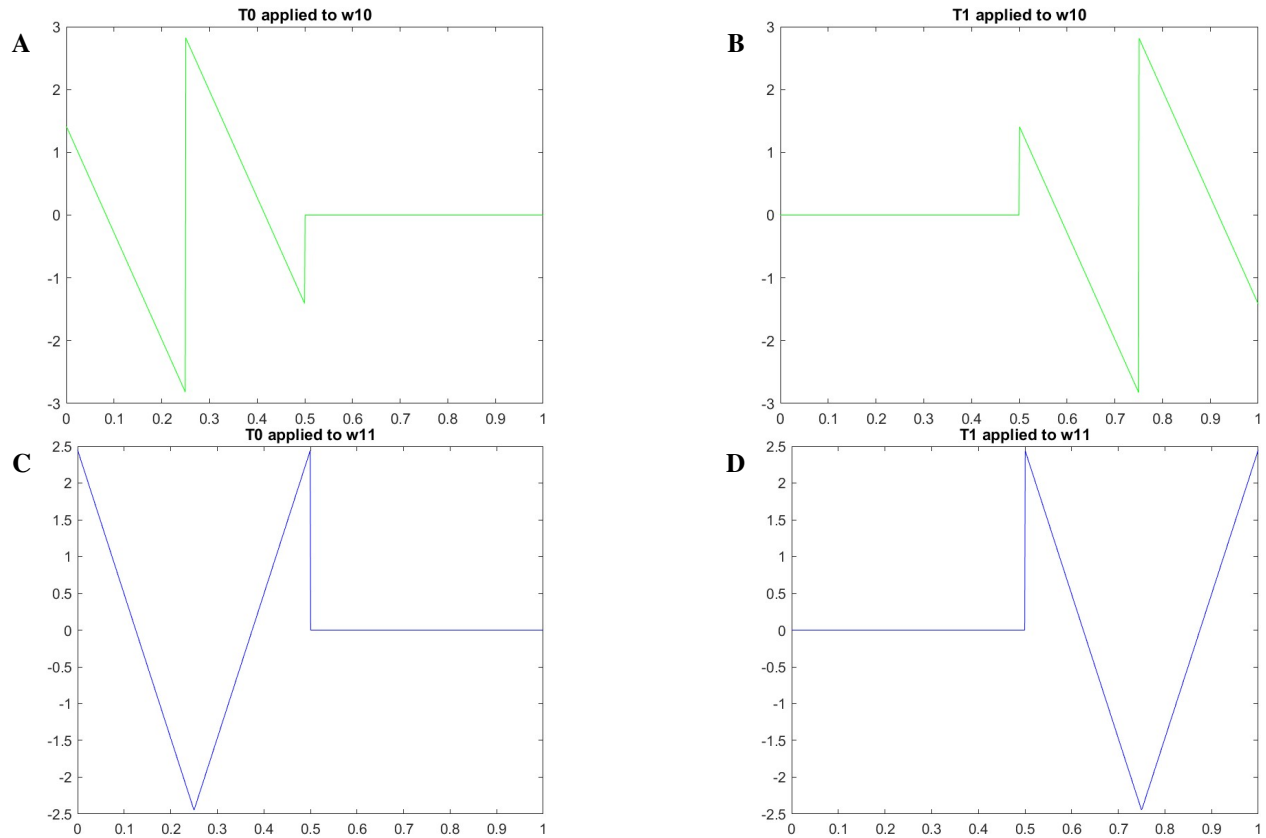


Figure 9. Orthonormal basis for \mathbb{W}_2 . (A) \mathcal{T}_0 applied to w_{10} . (B) \mathcal{T}_1 applied to w_{10} . (C) \mathcal{T}_0 applied to w_{11} . (D) \mathcal{T}_1 applied to w_{11} .

Finally, we apply the operators 21 to \mathbb{W}_2 to obtain the basis functions for the space \mathbb{W}_3 . This can be seen in Figure 10 and Figure 11

$$\mathbb{W}_3 : \mathcal{T}_0 \{w_{20}, w_{21}, w_{22}, w_{23}\} \cup \mathcal{T}_1 \{w_{20}, w_{21}, w_{22}, w_{23}\} := \text{span}\{w_{30}, w_{31}, w_{32}, w_{33}, w_{34}, w_{35}, w_{36}, w_{37}\}.$$

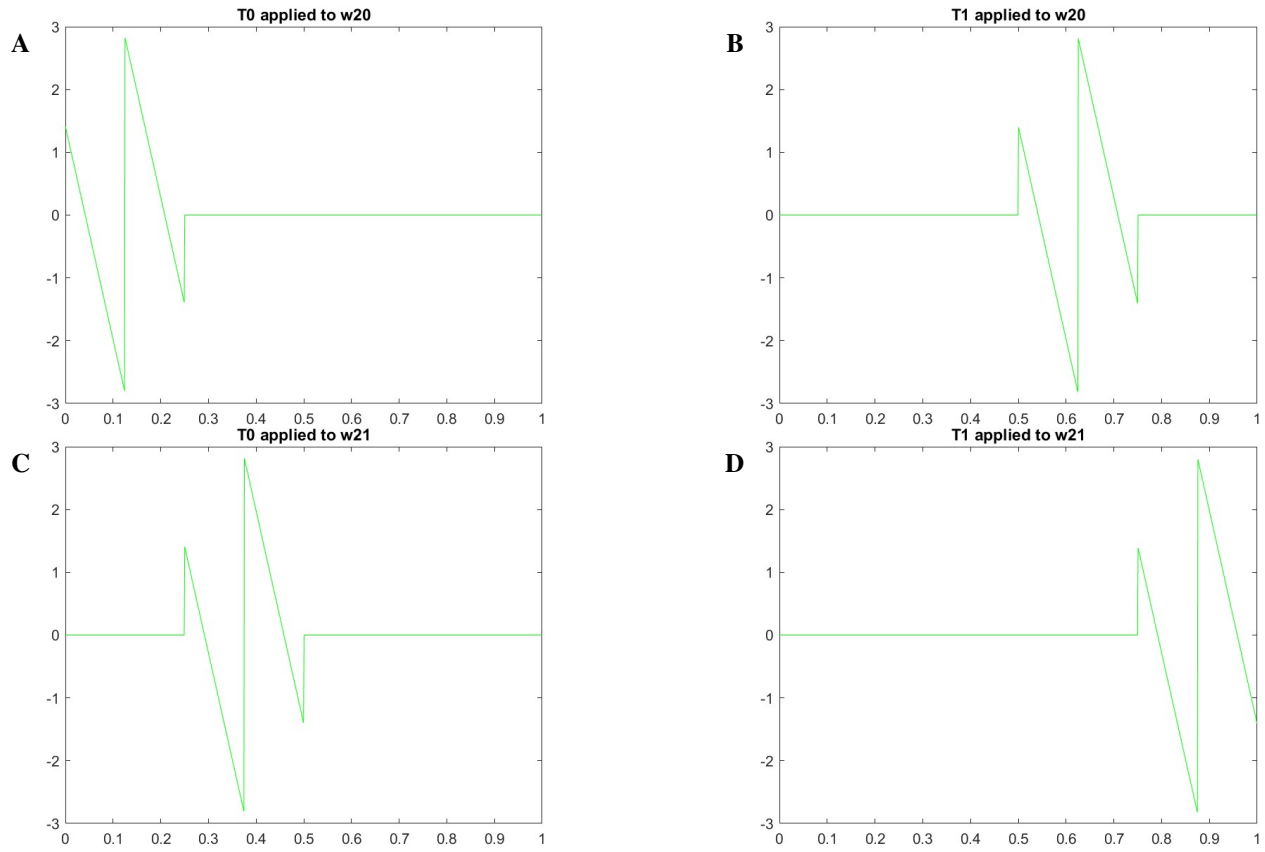


Figure 10. Orthonormal basis for \mathbb{W}_3 . (A) \mathcal{T}_0 applied to w_{20} . (B) \mathcal{T}_1 applied to w_{20} . (C) \mathcal{T}_0 applied to w_{21} . (D) \mathcal{T}_1 applied to w_{21} .

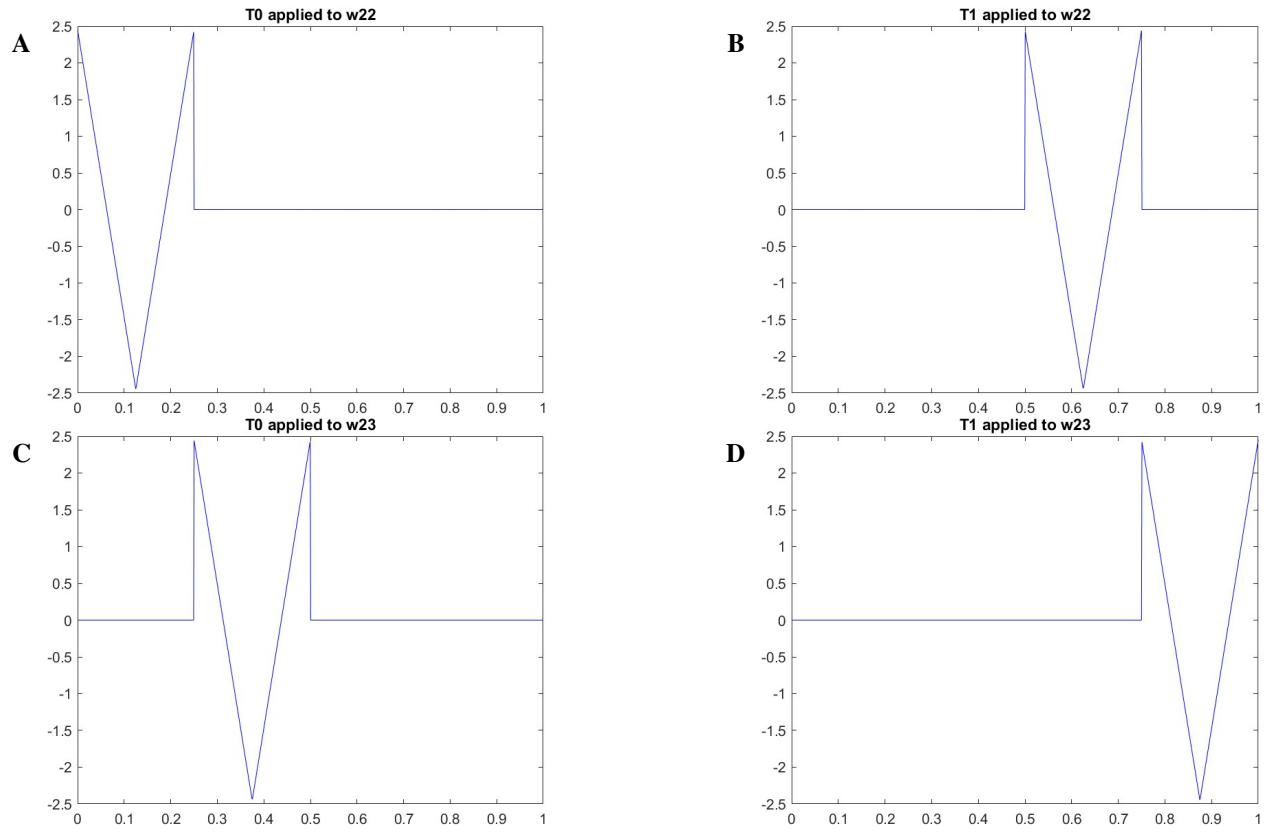


Figure 11. Orthonormal basis for \mathbb{W}_3 . (A) \mathcal{T}_0 applied to w_{22} . (B) \mathcal{T}_1 applied to w_{22} . (C) \mathcal{T}_0 applied to w_{23} . (D) \mathcal{T}_1 applied to w_{23} .

CHAPTER 6

NUMERIAL RESULTS

In this chapter, we develop numerical experiments by harnessing the results from previous chapters to establish our goals in two folds. The first, involves comparing our coarse mesh solution with that of the fine-scale. The efficiency of computation was improved through the implementing of parallel computing, and the enrichment of the coarse-scale solution through the addition of multiscale basis on the coarse edges. In this case, going from one level to another.

The second, entails the application of Artificial Neural Network (ANN) in determining a solution map (Dirichlet-to-Neumann) between the coarse blocks. Training is done on a single coarse block instead of the entire domain. A similar case of element learning was also established in [21]

6.1 MSHDG RESULTS

We consider the domain $\Omega = [0, 1] \times [0, 1]$ and divide Ω into $N = M \times M$ coarse blocks where $H = 1/M$ is the size of each coarse blocks. Then, within each subdomain Ω_i , we establish a structured triangulation (fine-scale) with m subintervals on each axis. This gives us a fine-mesh size of $h = 1/(Mm)$. We test our method with the following PDE parameters: $f = 1$, $\tau = 1$ representing the stabilization parameter on each edge (fine and coarse edge), and a multiscale coefficient $\kappa = 1$ in white regions and $\kappa = 10^4$ in the gray regions. See Figure 12. To fully implement the multiscale HDG method established in Chapter 4, we consider W_h and V_h to be spaces of piece-wise linear, discontinuous functions on \mathcal{T}_h , M_h a space of piece-wise linear, discontinuous functions on ξ_h and M_H is the multiscale space consisting of multiscale functions constructed in Chapter 5.

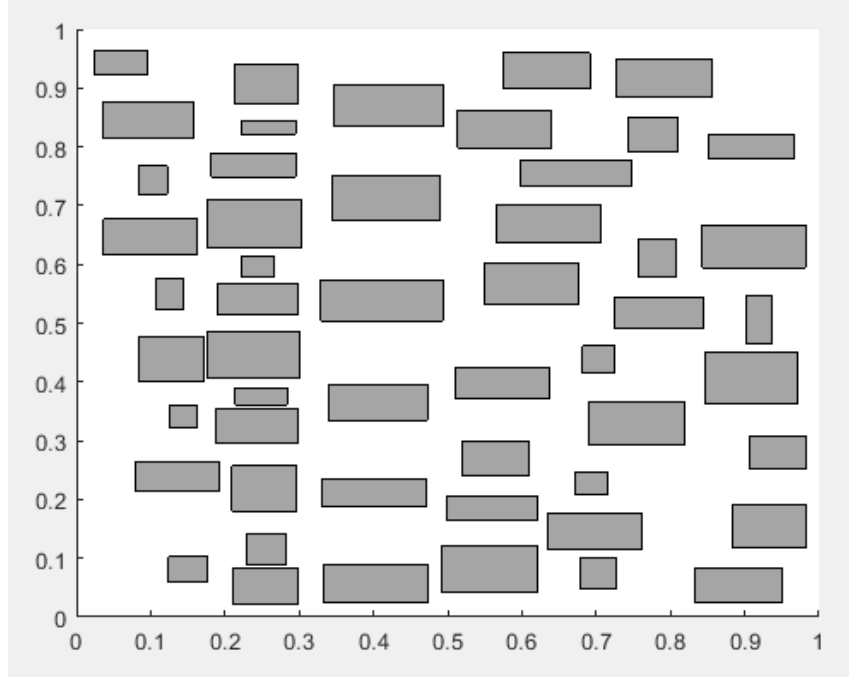


Figure 12. Distribution of kappa. $\kappa = 1$ in white regions and $\kappa = 10^4$ in dark or gray regions.

We can view the results of this implementation in Figure 13 and Figure 14. We choose a 10×10 coarse block with size $H = 1/10$. Then in each coarse block, we choose 8×8 smaller squares of size $h = H/8$. This give us 128 elements in each coarse block. We find approximations to the fine-scale solution by adding more multiscale basis functions on the coarse edges. This is accomplished by going from one level to the other, where the dimension of each level is given by $L_i = 2 \times 2^i$, for $i = 1, 2, 3$. Figure 15 shows the fine-scale solution with the given parameters. For error analysis, we use the weighted L^2 -norm for the solution $\|\hat{u}_h - \hat{u}_H\|_{L^2(\Omega)}$ between the fine-scale solutions \hat{u}_h and the coarse-scale \hat{u}_H on the coarse scale edges. The results can be seen in Table1.

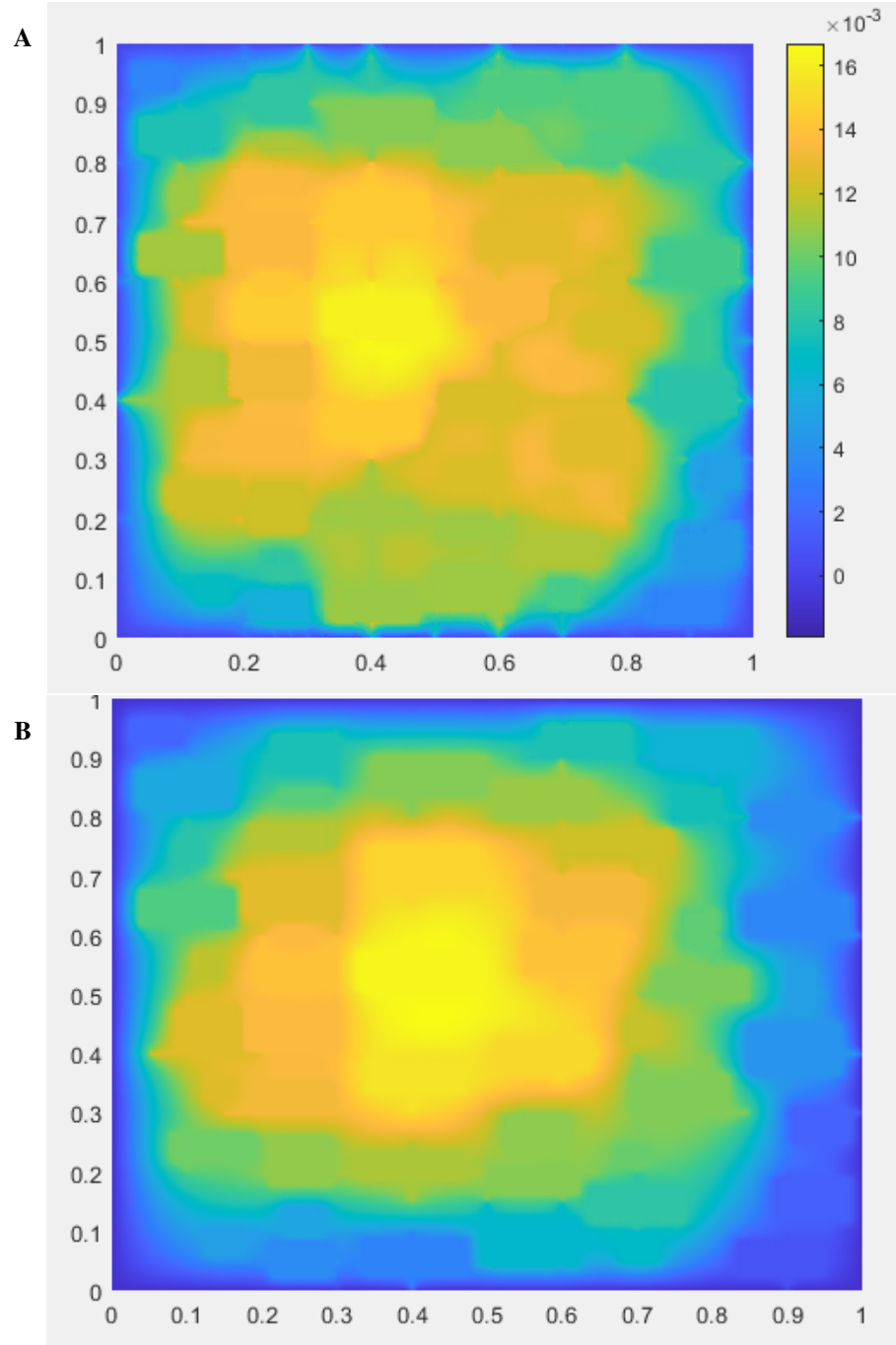


Figure 13. Comparison of MsHDG solutions for M_H . (A) L_0 , $\text{Dim}=360$. (B) L_1 , $\text{Dim}=720$

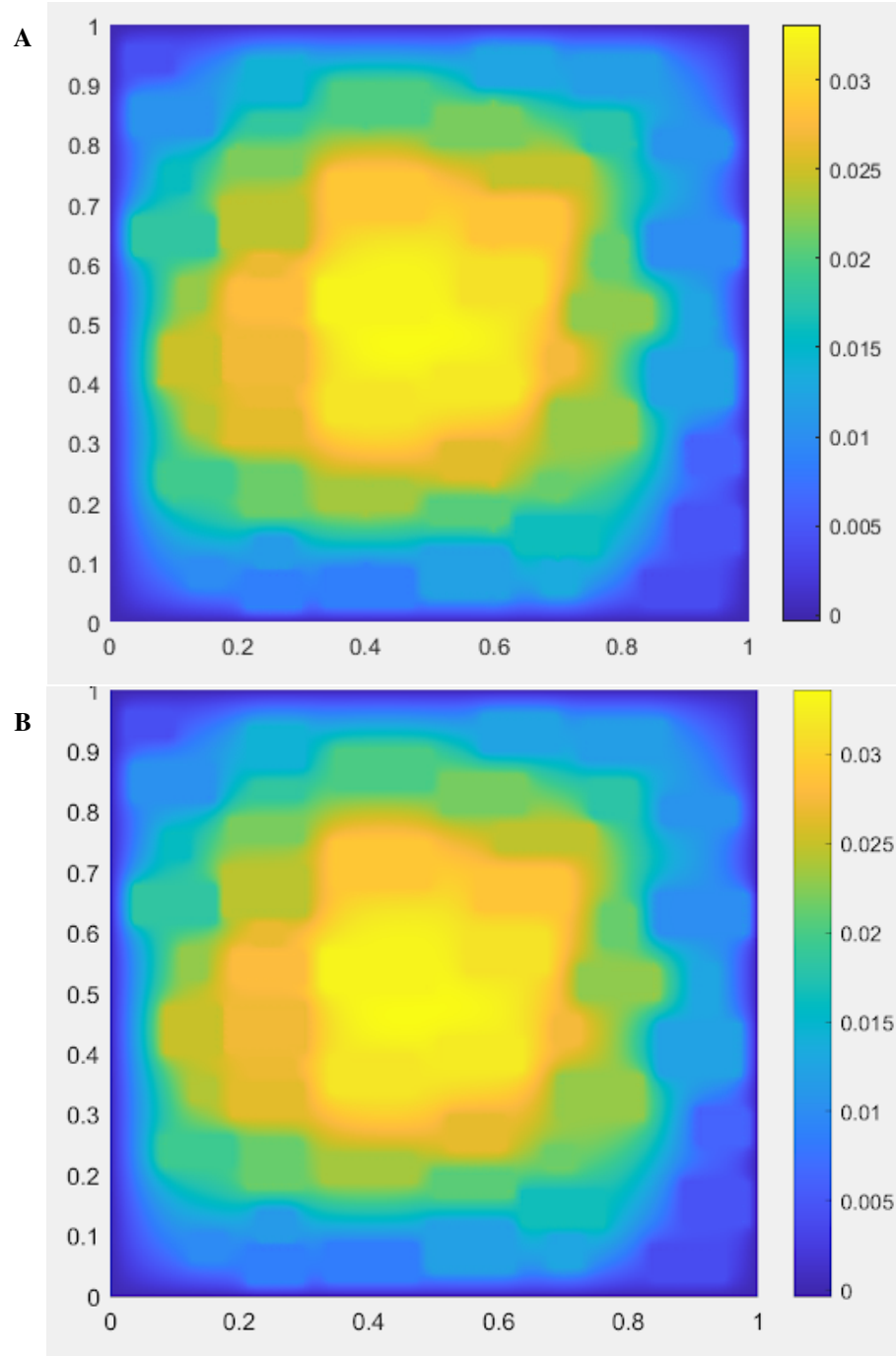


Figure 14. Comparison of MsHDG solutions for M_H . (A) L_2 , $\text{Dim}=1440$. (B) L_3 , $\text{Dim}=2880$.

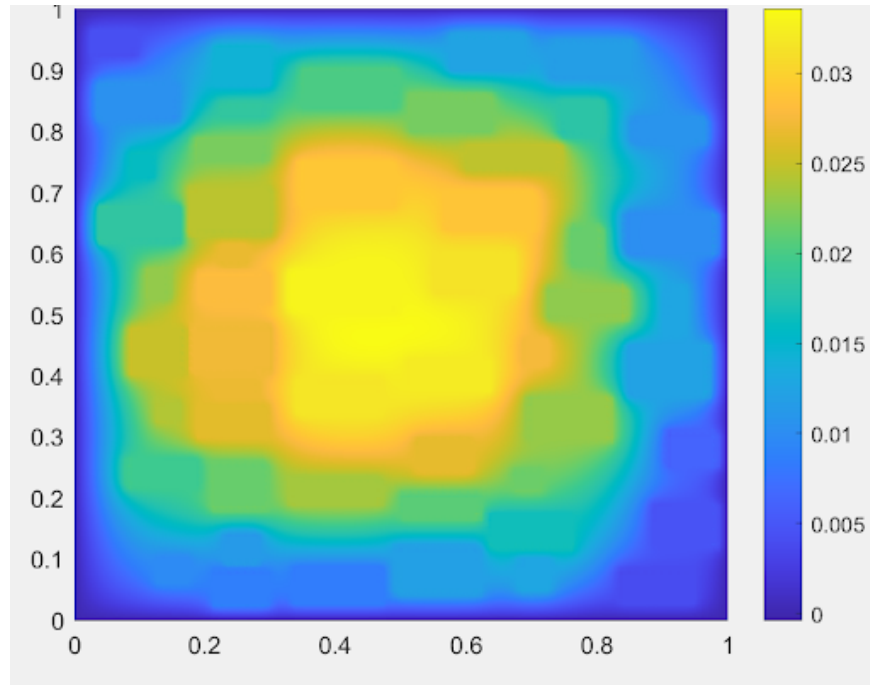


Figure 15. Fine-scale solution with dimension (Dim) = 152960.

Table 1. Error and processor time for different scale levels on M_H .

Level	Dim.	$\ \hat{u}_h - \hat{u}_H\ _{L^2(\Omega)}$	Elapsed time(seconds)
0	360	0.0041	3.9516
1	720	0.0023	6.3237
2	1440	0.00047	12.0186
3	2880	0.000042	28.5727

The fourth column (Elapsed time), in Table1 shows the results from parallel computation. Parallel processing was done using MATLAB parallel computing toolbox on an Intel(R) Core(TM) i7-14700F process with twenty cores and twenty-eight Logical processors. MATLAB used a maximum of eight workers.

6.2 ARTIFICIAL NEURAL NETWORK

The goal of this section is to improve the efficiency of our multiscale HDG method using machine learning. Previous numerical results established in section 6.1 were obtained through the use of local solvers for each coarse block over the entire domain. And although the dimension of the global system is reduced, the process does not eliminate the disadvantage of performing tedious operations over every coarse block. As such, a significant advantage of using neural network is that we can train the network on a single coarse block (reference coarse block) to learn the solution map, or more specifically, the Dirichlet-to-Neumann map between coarse block. And once the network is trained, it can be used to determine solution map for other geometries (coarse blocks) using a simple linear transformation.

We establish the training process of our neural network in the following manner. Consider the domain $\Omega = [0, 1] \times [0, 1]$ and divide Ω into $N = M \times M$ smaller squares, where $M = 6$. Then on Ω , we generate a structured triangulation (fine-scale) containing 128 finite elements. This domain will serve as our reference coarse block. We perform a random sampling of kappa (κ) such that within each small square, we have $\kappa = 1$ in the white regions and $\kappa = 10^4$ in the dark/gray regions. See Figure 17.

6.2.1 Data Generation

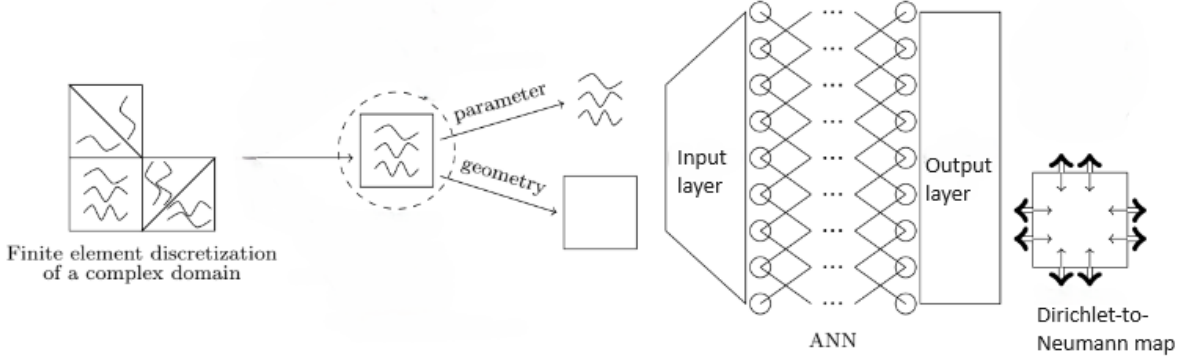


Figure 16. Coarse block learning. A domain is decompose into several subdomains. For each subdomain, the neural network takes the PDE parameters as inputs, and returns the Dirichlet-to-Neuman map. (Reproduced from [21])

We evaluate kappa within each element and store the following two matrices. The first, is the coordinates of all vertices of all elements in Ω . This will have a dimension $N_{elts} \times 3$. The second, is the matrix containing the barycentric coordinates of all the quadrature nodes, with dimension $3 \times N_{nd}$. Then, the product of these two matrices gives us the x and y-coordinates of all quadrature nodes in all triangles with dimension $N_{elts} \times N_{nd}$. We reshape this matrix into a $1 \times N_{elts} \times N_{nd}$ vector for each sample of kappa generated. The value of the entries of this matrix will depend on whether the x,y-coordinate is within the white or dark/gray regions. This will serve as the data for the input layer of our neuron network.

Next, we determine the output layer of the neural network.

$$\text{Output layer} := ([a(\xi_H, \mu)], [l(\mu)]).$$

Our reference domain consists of four coarse edges on which we assign multiscale basis function to

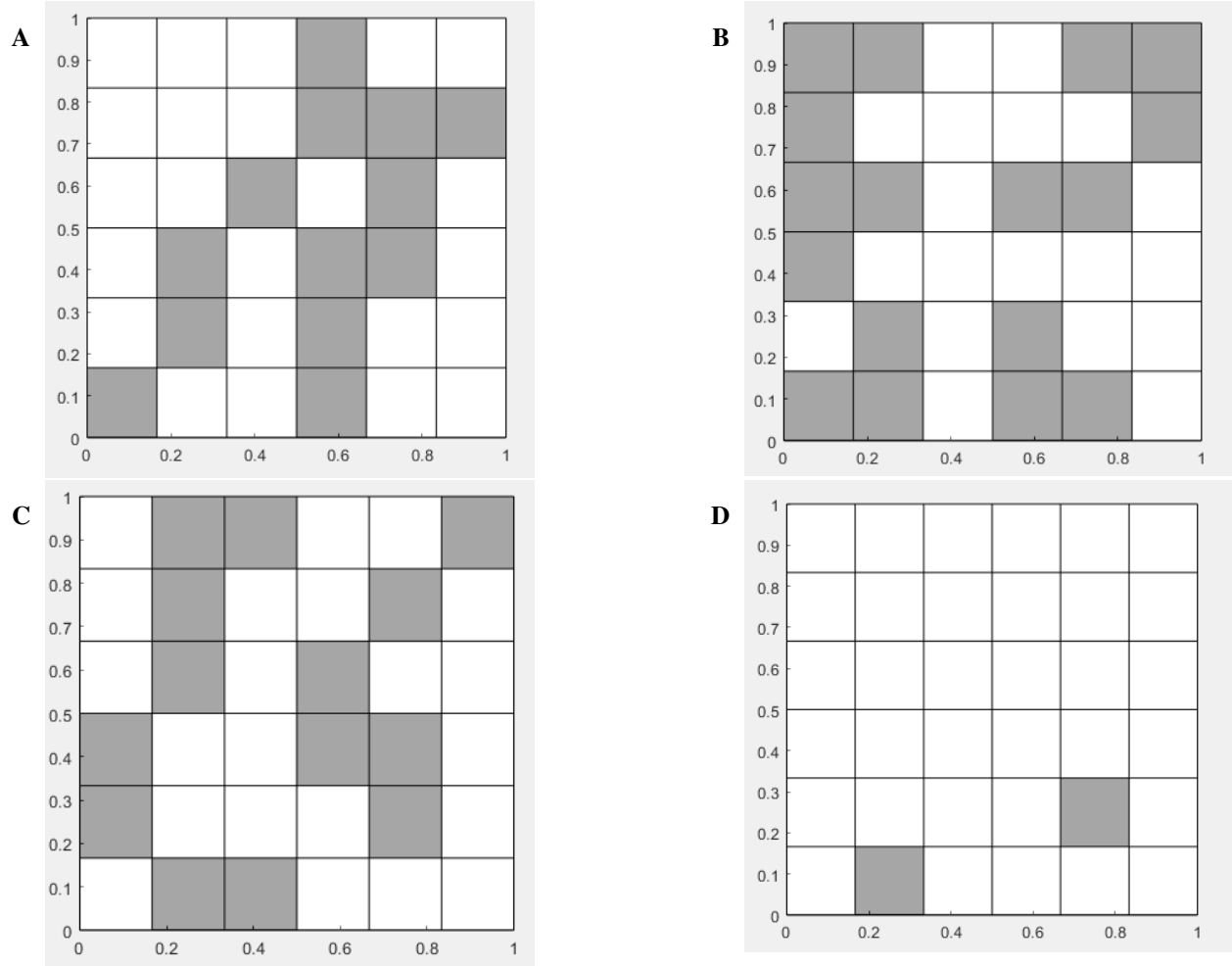


Figure 17. Random samples of kappa κ . (A) Sample 1 . (B) Sample 2 . (C) Sample 3. (D)

Sample 4. $\kappa = 1$ in the white regions and $\kappa = 10^4$ in the dark/gray regions

determine the flux response (Dirichlet-to-Neumann map). This is a symmetric matrix of dimension

$4 N_{basis} \times 4 N_{basis}$, where

$$N_{basis} = 2 \cdot 2^{level},$$

is the number of basis functions per coarse edge and *level* is the level of the scale. We form an upper

triangle of this matrix and use the values as the output layer with dimension $1 \times \frac{4N_{basis}(4N_{basis}+1)}{2}$.

We also form a $1 \times 4 N_{basis}$ vector for the right-hand-side of Equation 18. Then, the output layer

of our neural network is a vector with total dimension $1 \times \frac{4N_{basis}(4N_{basis}+1)}{2} \times 4 N_{basis}$.

For the rest of the neural network structure, we follow a similar design outlined in [21]. Among the $N_{samp} = 1000$ random samples generated for the distribution of κ , 800 of them constitute the training set and the remaining 200 samples, the testing set. We use a fully connected neural network with four hidden layers, where the size of the hidden layer is four times the input layer. The activation function is chosen to be the Rectified Linear Units (ReLU) along with Adam optimization and L1-norm error as the loss function (see Figure 19). We choose a batch size of 100 with an initial learning rate 10^{-3} for the first 3000 epochs. For the next 3000 epochs, we reduce the learning rate to 10^{-4} and then to 10^{-5} for the last 3000 epochs. All training was done on the PyTorch platform using 8.0 GB NVIDIA GeForce RTX 4060 dedicated GPU memory. Figure 18 shows the L1-norm errors of the neural network for both training and testing using two multiscale basis functions.

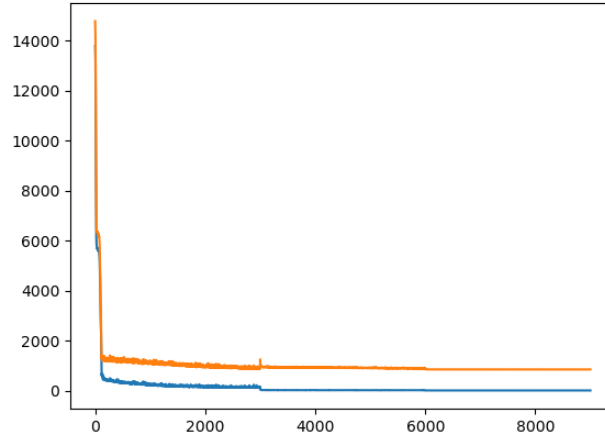


Figure 18. Training (blue) and testing(orange) errors for neural network. L1-norm errors (y-axis) vs. number of epochs(x-axis).

We tested the performance of our neural network on a two-by-two (2x2) and three-by-three (3x3) coarse blocks with random distribution of kappa κ for different levels (L_0, L_1) of multiscales basis functions. Given the reference domain (coarse block) $\hat{K} = [0, 1]^2$, the parameterization of any geometry K (physical domain) is determined by the following push-forward map:

$$F_K : \hat{K} \longrightarrow K,$$

$$(\hat{x}, \hat{y}) \longrightarrow \begin{pmatrix} a_{11}^K & a_{12}^K \\ a_{21}^K & a_{22}^K \end{pmatrix} \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} + \begin{pmatrix} b_1^K \\ b_2^K \end{pmatrix},$$

where the geometry of K is parameterized by the parameters a_{ij}^K and b_i^K . Since we are using square coarse blocks, this reduces to

$$(\hat{x}, \hat{y}) \longrightarrow \begin{pmatrix} h\hat{x} \\ h\hat{y} \end{pmatrix} + \begin{pmatrix} b_1^K \\ b_2^K \end{pmatrix},$$

where h represents the size of the coarse block K . Then, solving Equation 17 on K is equivalent to solving the same equation on \hat{K} with a re-scaled coefficient h .

The L2-norm error between the solution of the multiscale HDG method proposed in Chapter 4, and that of the neural network was also determined within an error tolerance of 10^{-3} . Figure 21 to Figure 34 uses two multiscale basis functions (L_0) per coarse edge, whereas Figure 36 and Figure 40 uses four multiscale basis functions (L_1). Table 2 shows the amount of time(seconds) it takes each method to achieve an approximate solution.

Table 3 shows the errors between the MsHDG and Artificial Neural Network (ANN) solutions for different test domains.

Table 2. Elapsed time (seconds) to achieve an approximate solution in both methods at scale level L_0 .

Method	2×2 coarse block	3×3 coarse block
ANN	0.5402	0.6547
MsHDG	2.0610	5.1105

Table 3. Errors between the MsHDG and Artificial Neural Network (ANN) solutions for different test domains.

Test domain	$\ \hat{u}_h^{MSHDG} - \hat{u}_h^{ANN}\ _{L^2(\Omega)}$
1	3.2609×10^{-4}
2	6.3265×10^{-4}
3	2.6397×10^{-4}
4	9.2869×10^{-4}
5	7.1630×10^{-4}
6	9.1246×10^{-4}
7	7.9571×10^{-4}

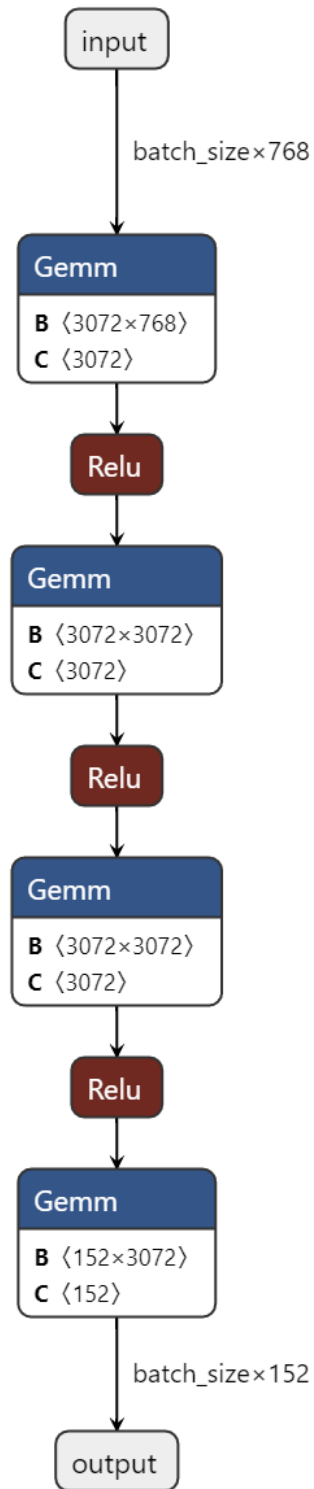


Figure 19. Neural network architecture comprising input layer, hidden layer with non-linear activation functions (ReLU), and an output layer.

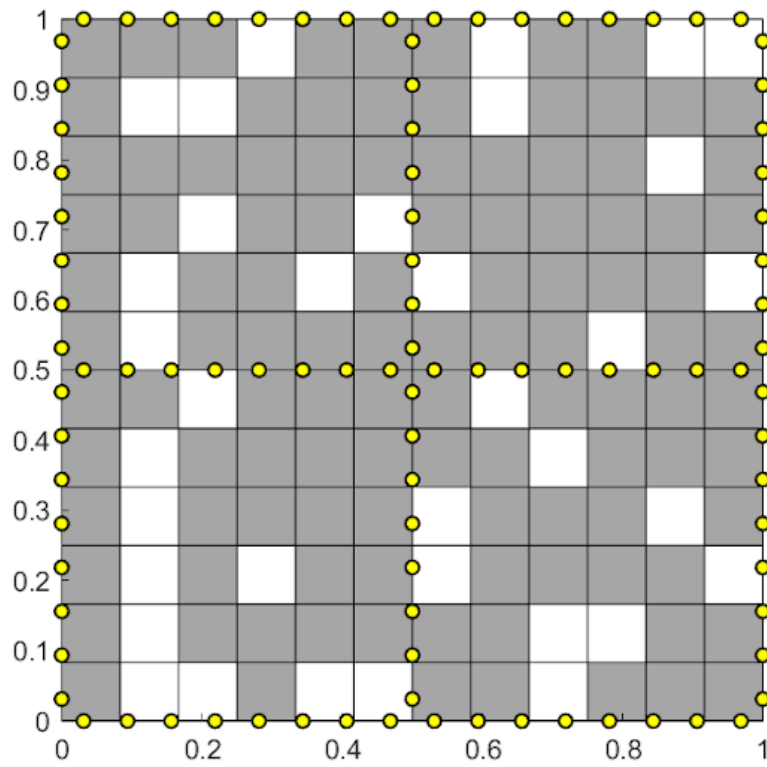


Figure 20. Test domain 1: $\kappa = 1$ in the white regions and $\kappa = 10^4$ in the dark/gray regions

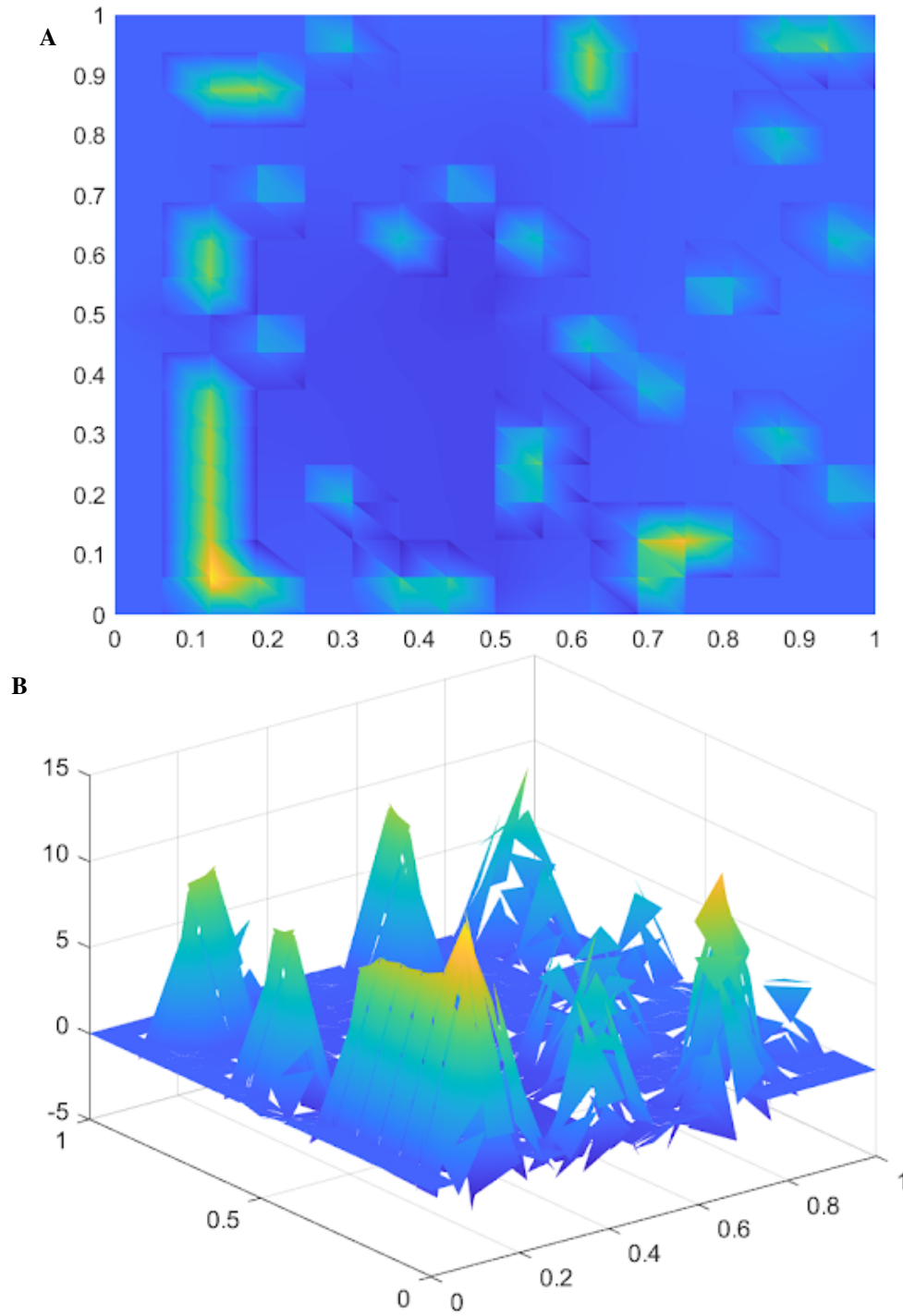


Figure 21. L_0 neural network solutions for Test domain 1. (A) 2-D view of neural network approximation . (B) 3-D view of neural network approximation ($\times 10^{-3}$).

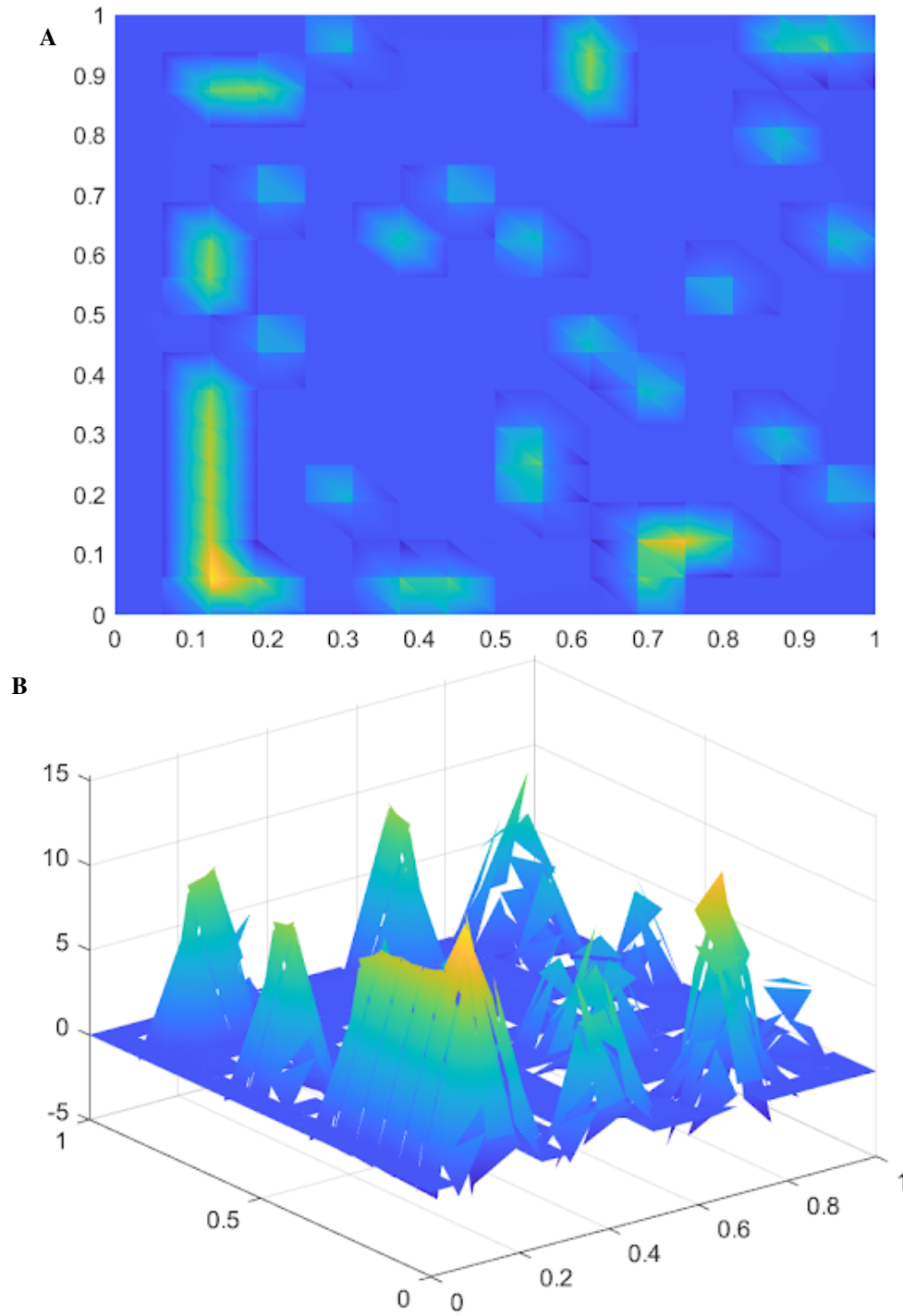


Figure 22. L_0 MsHDG solution for Test domain 1. (A) 2-D view of MsHDG approximation . (B) 3-D view of MsHDG approximation ($\times 10^{-3}$).

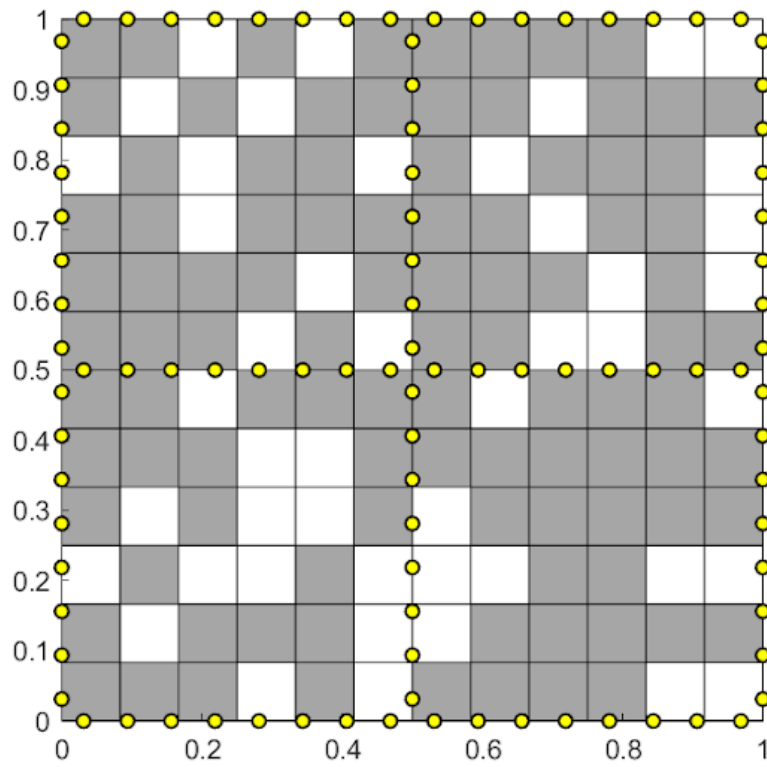


Figure 23. Test domain 2: $\kappa = 1$ in the white regions and $\kappa = 10^4$ in the dark/gray regions

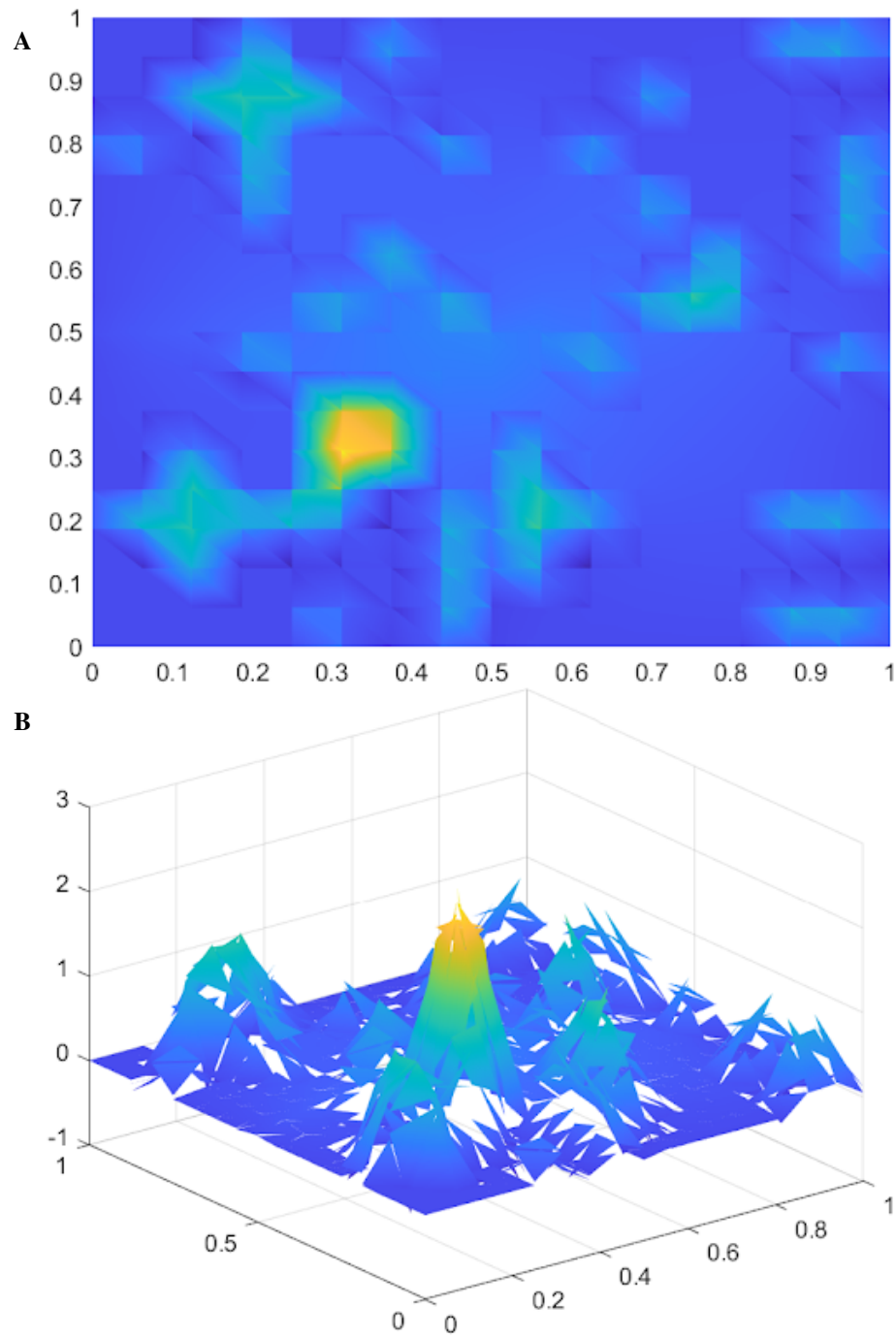


Figure 24. L_0 neural network solutions for Test domain 2. (A) 2-D view of neural network approximation . (B) 3-D view of neural network approximation ($\times 10^{-3}$).

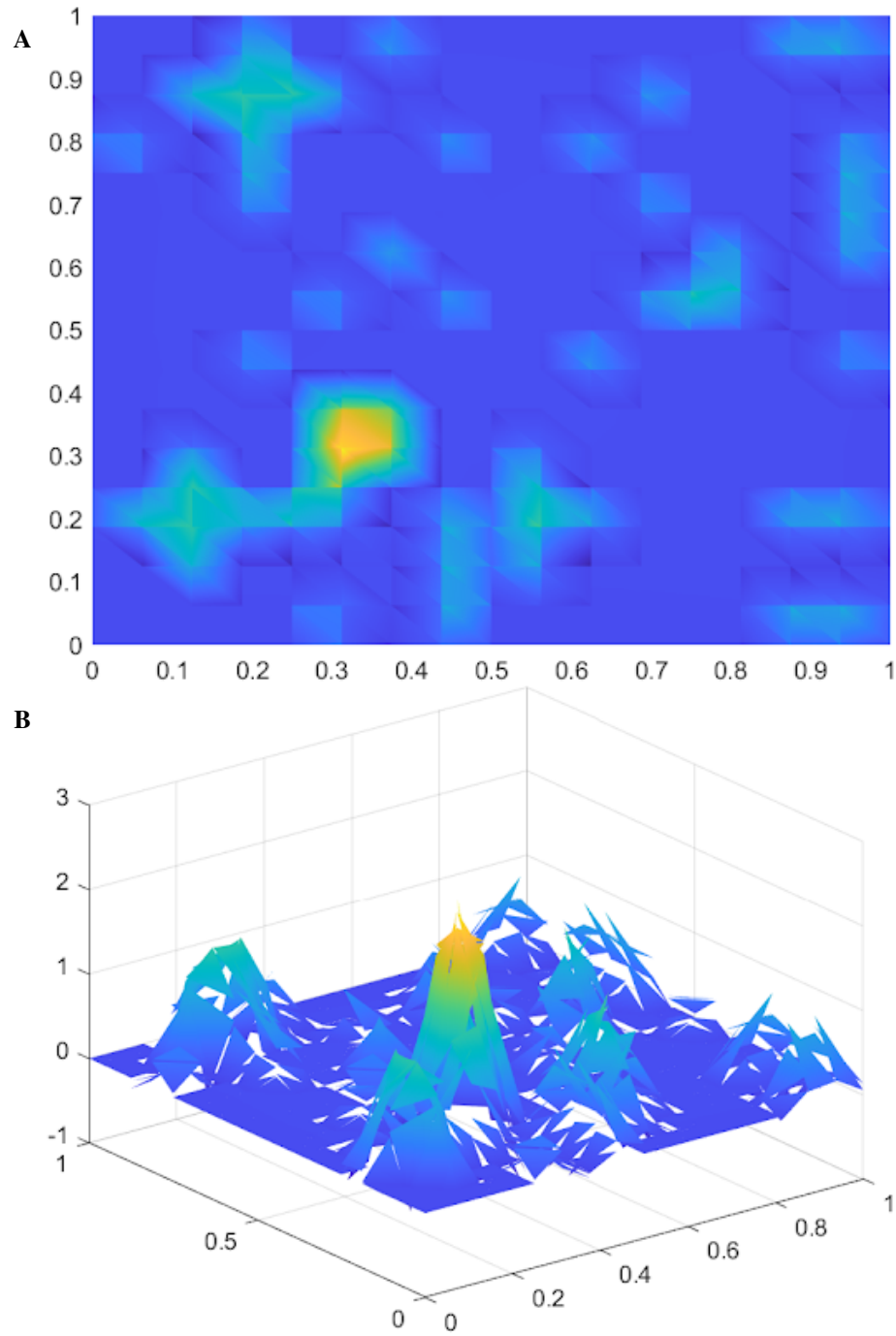


Figure 25. L_0 MsHDG solution for Test domain 2. (A) 2-D view of MsHDG approximation . (B) 3-D view of MsHDG approximation ($\times 10^{-3}$).

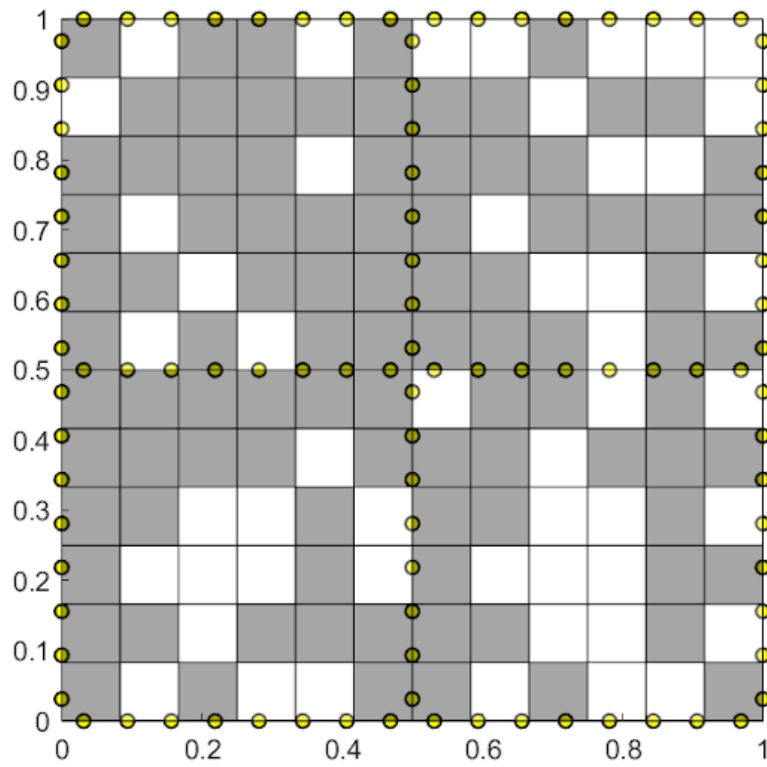


Figure 26. Test domain 3: $\kappa = 1$ in the white regions and $\kappa = 10^4$ in the dark/gray regions

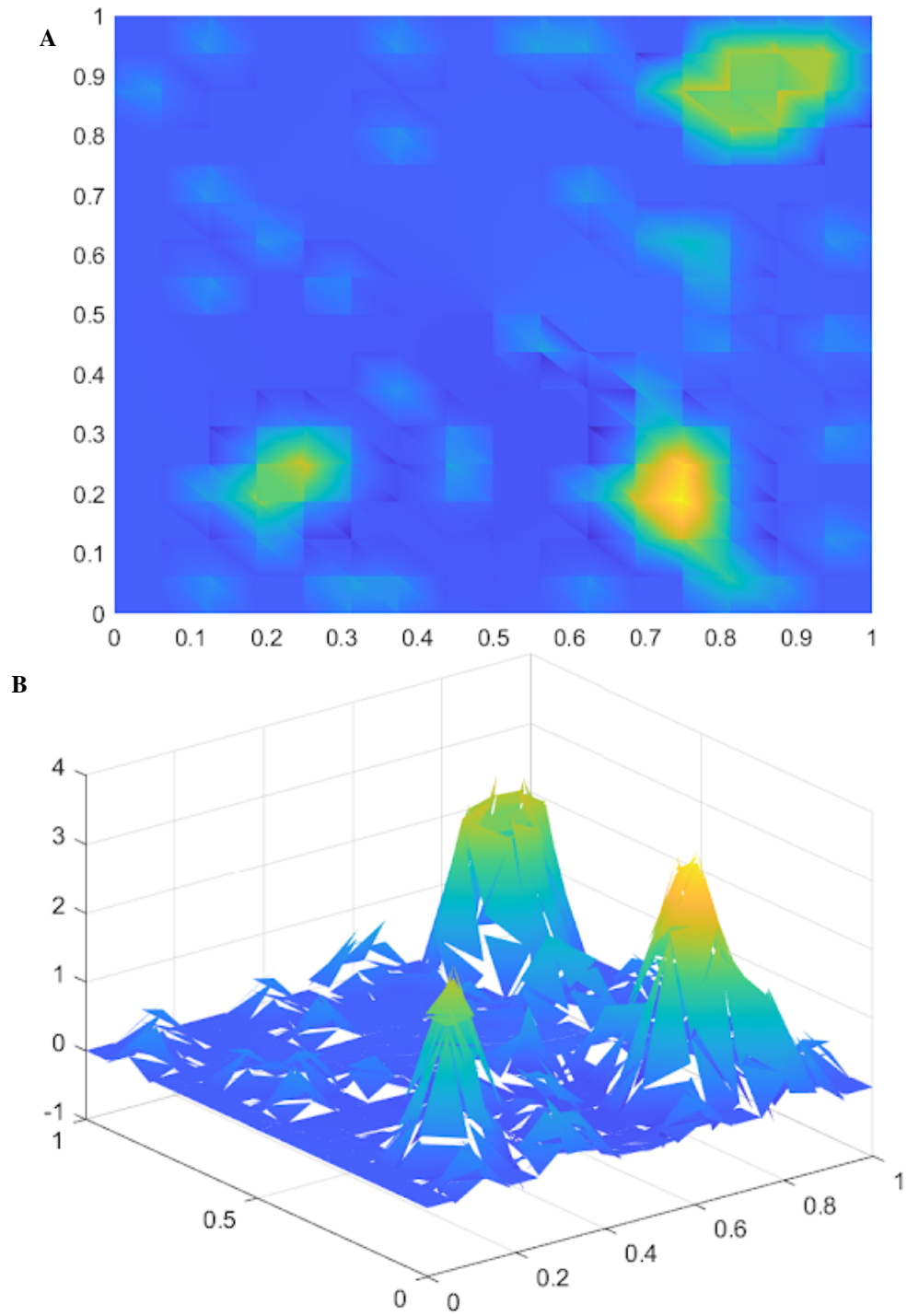


Figure 27. L_0 neural network solution for Test domain 3. (A) 2-D view of neural network approximation . (B) 3-D view of neural network approximation ($\times 10^{-3}$).

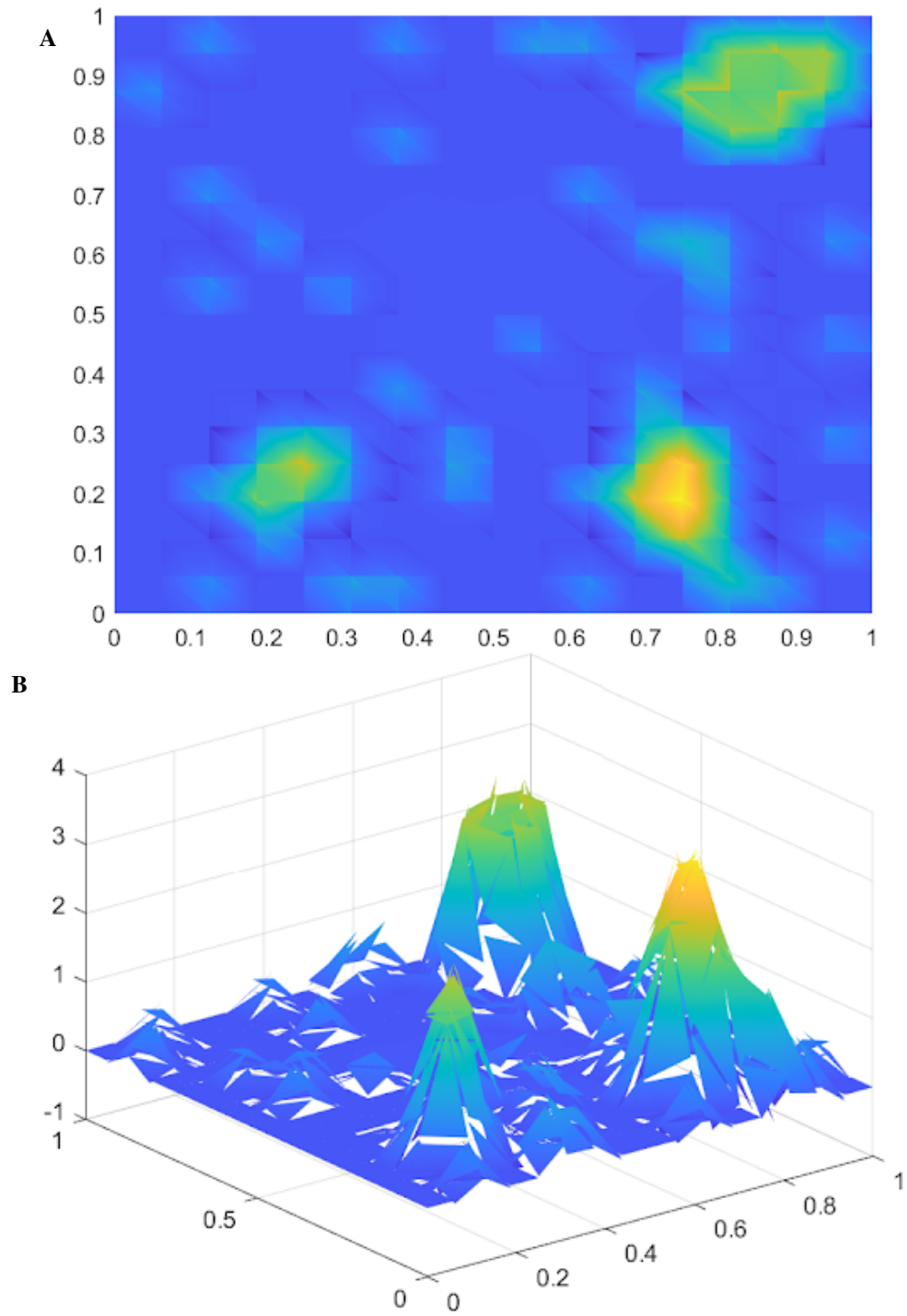


Figure 28. L_0 MsHDG solution for Test domain 3. (A) 2-D view of MsHDG approximation . (B) 3-D view of MsHDG approximation ($\times 10^{-3}$).

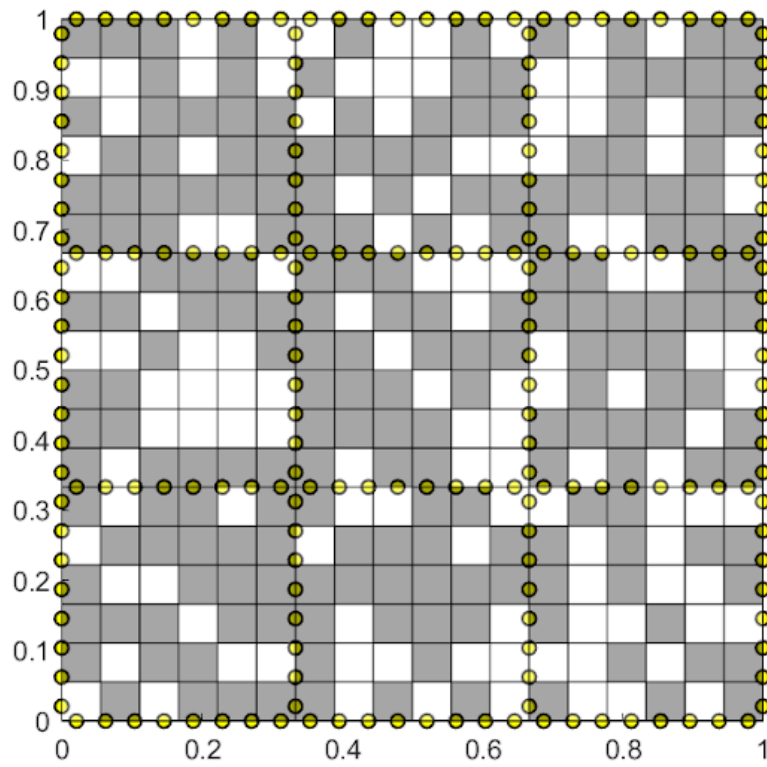


Figure 29. Test domain 4: $\kappa = 1$ in the white regions and $\kappa = 10^4$ in the dark/gray regions

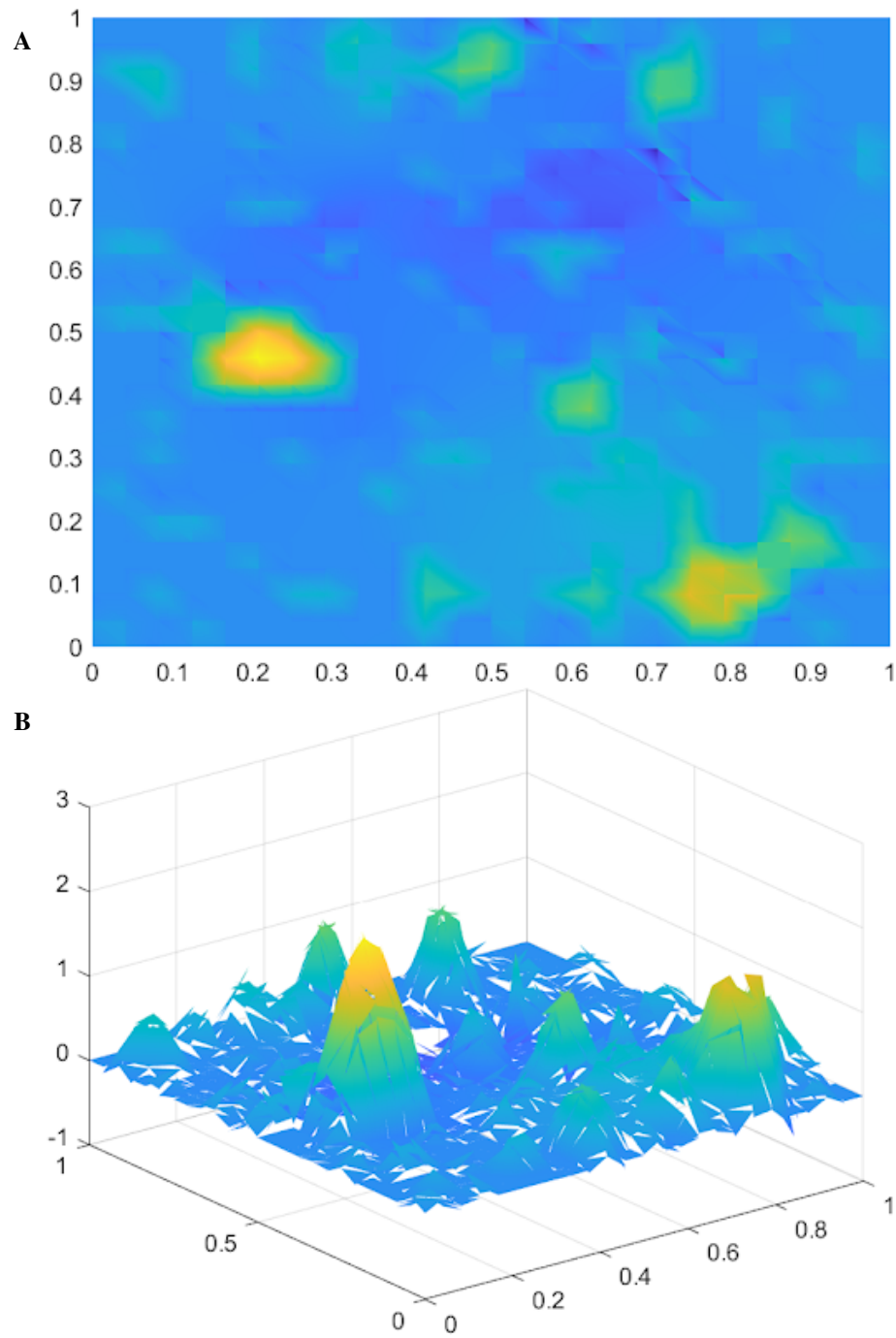


Figure 30. L_0 neural network solution for Test domain 4. (A) 2-D view of neural network approximation . (B) 3-D view of neural network approximation ($\times 10^{-3}$).

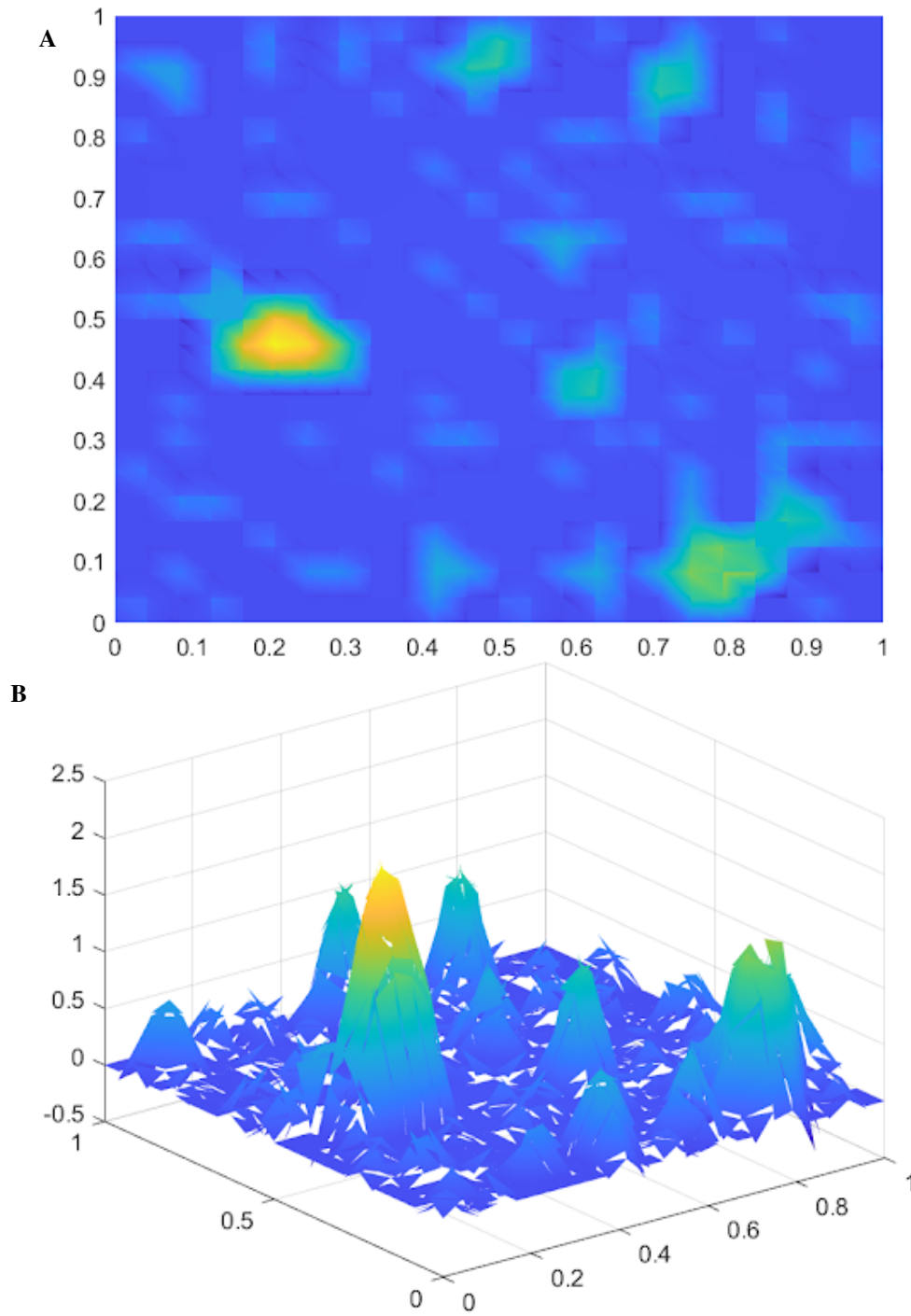


Figure 31. L_0 MsHDG solution for Test domain 4. (A) 2-D view of MsHDG approximation . (B) 3-D view of MsHDG approximation ($\times 10^{-3}$).

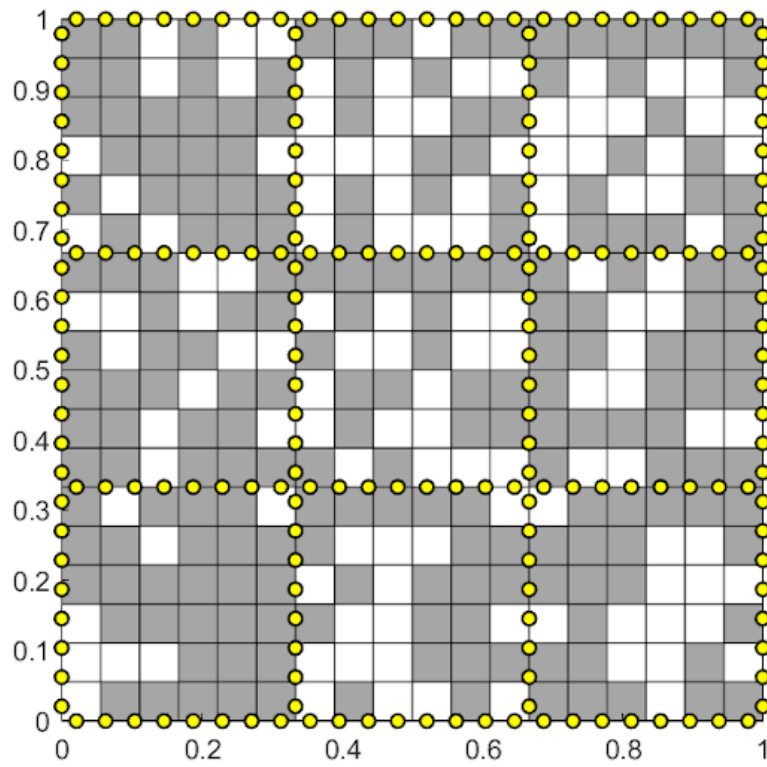


Figure 32. Test domain 5: $\kappa = 1$ in the white regions and $\kappa = 10^4$ in the dark/gray regions

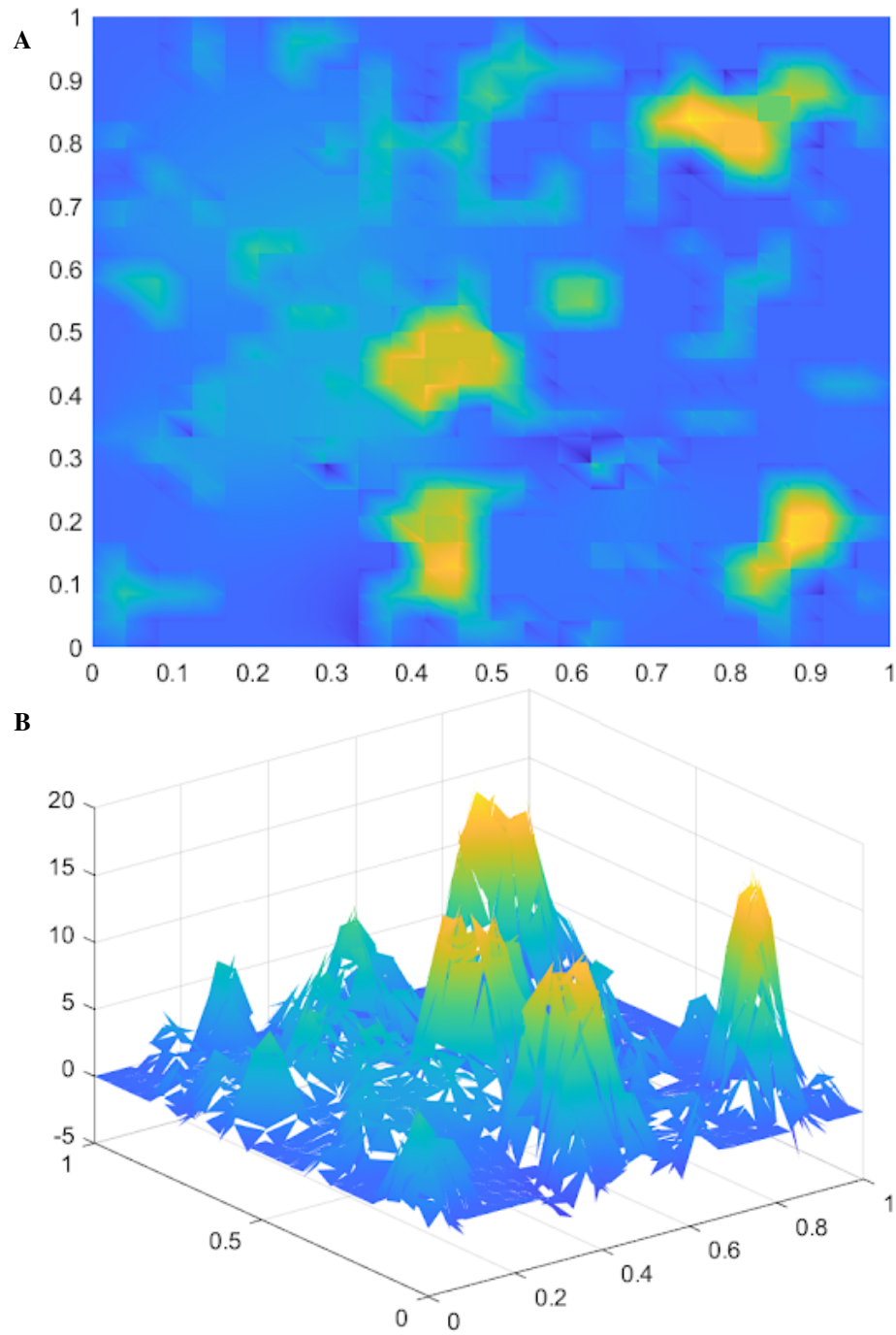


Figure 33. L_0 neural network solution for Test domain 5. (A) 2-D view of neural network approximation . (B) 3-D view of neural network approximation ($\times 10^{-4}$).

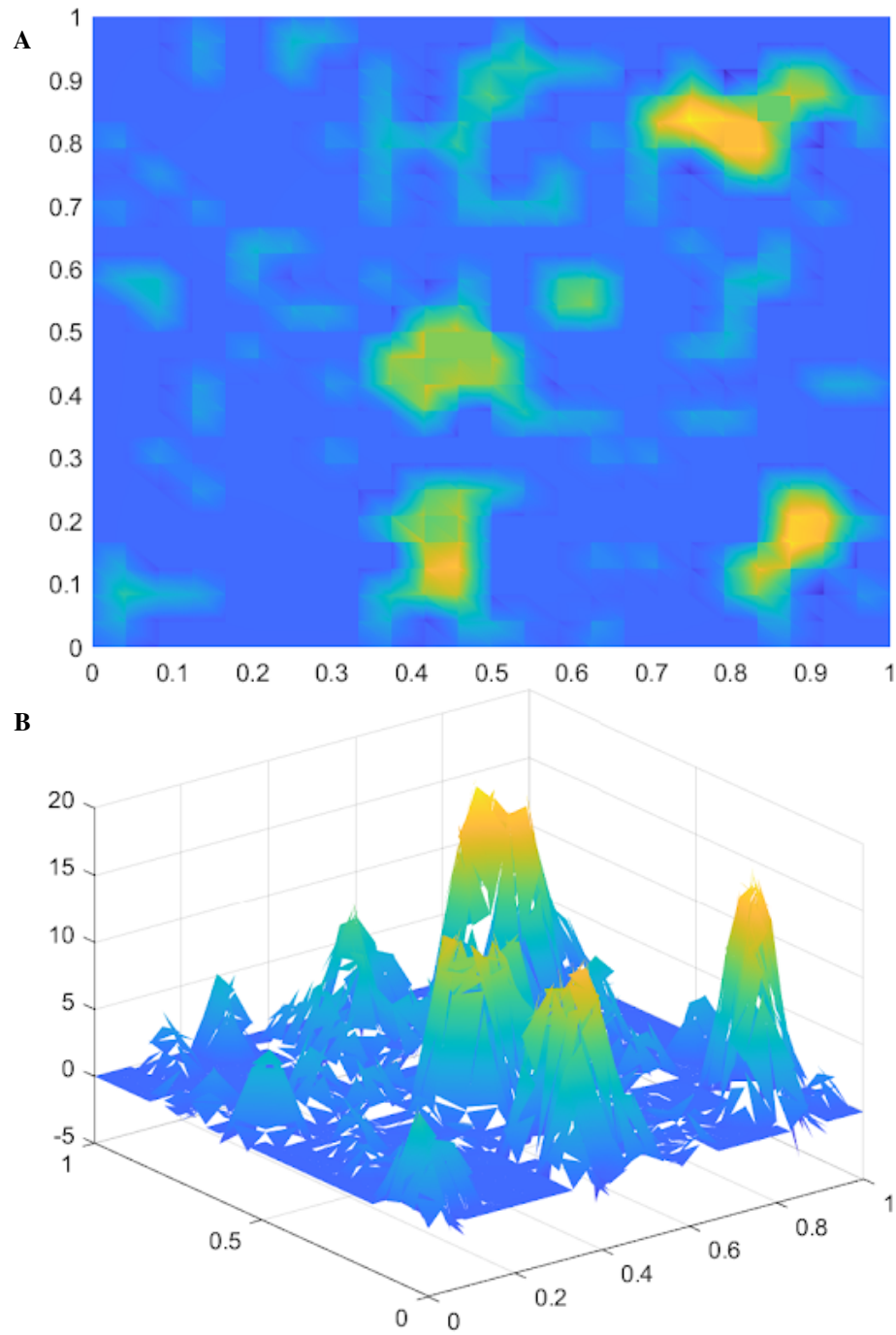


Figure 34. L_0 MsHDG solution for Test domain 5. (A) 2-D view of MsHDG approximation . (B) 3-D view of MsHDG approximation ($\times 10^{-4}$).

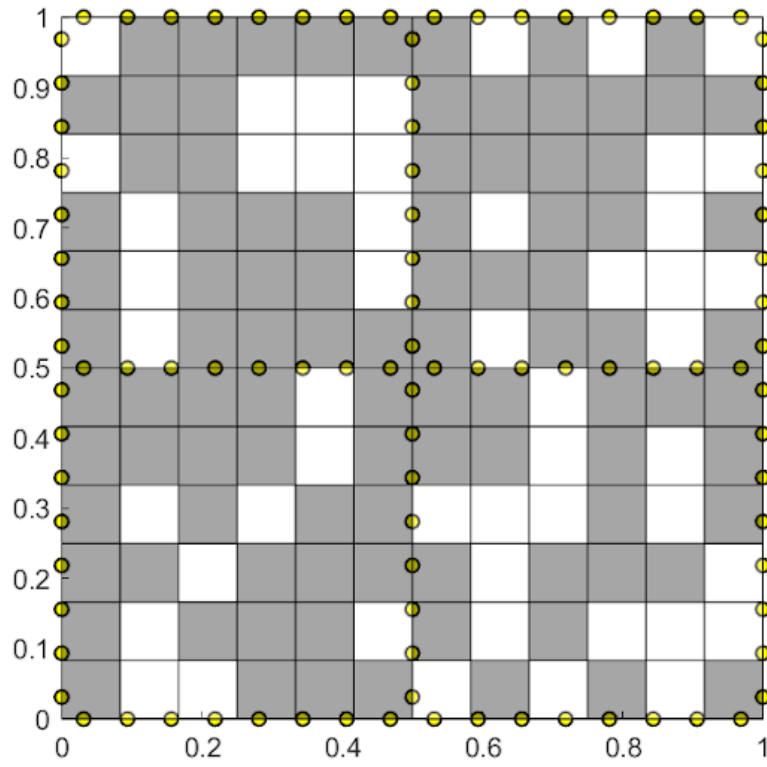


Figure 35. Test domain 6: $\kappa = 1$ in the white regions and $\kappa = 10^4$ in the dark/gray regions

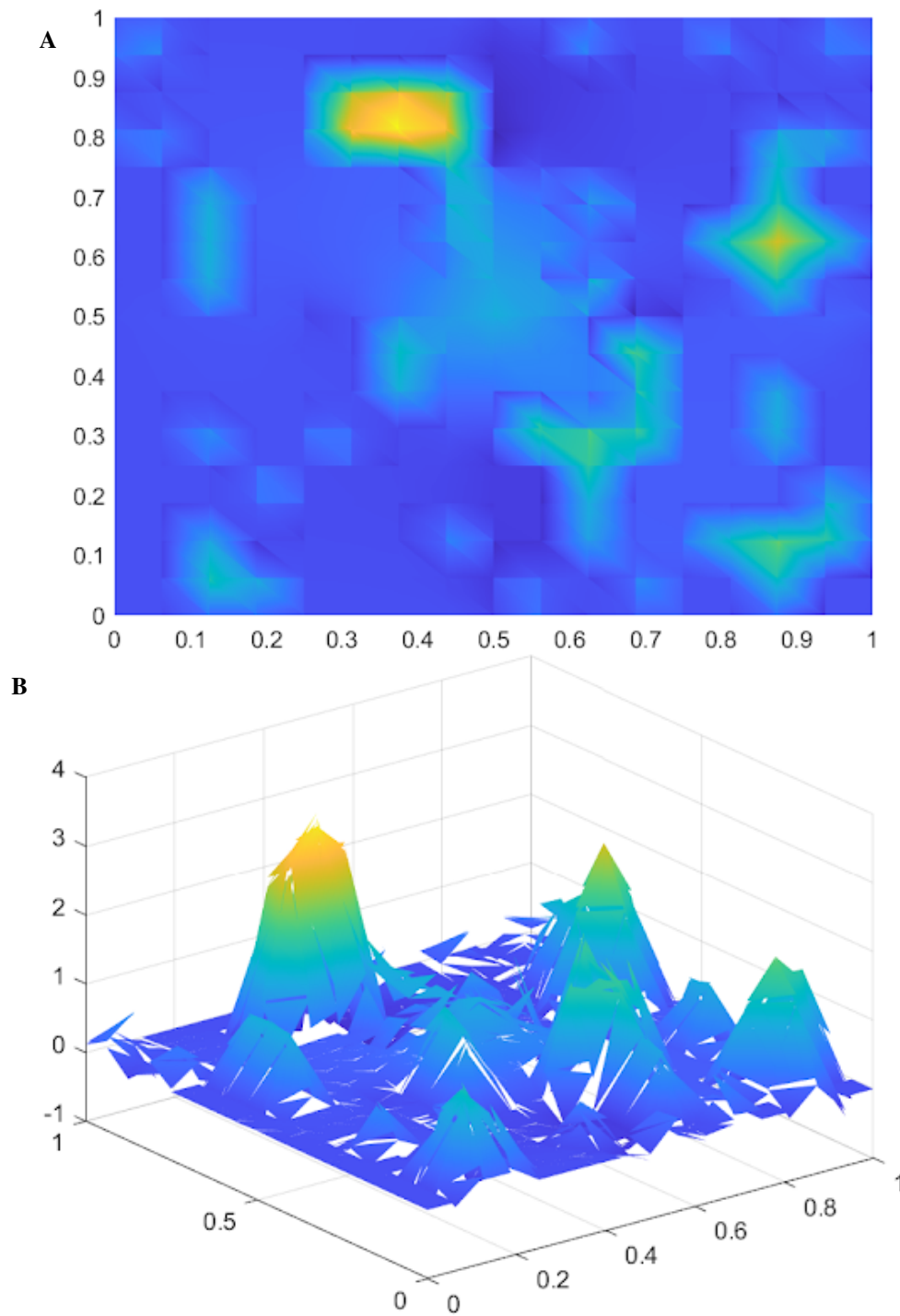


Figure 36. L_1 neural network solution for Test domain 6. (A) 2-D view of neural network approximation. (B) 3-D view of neural network approximation ($\times 10^{-3}$).

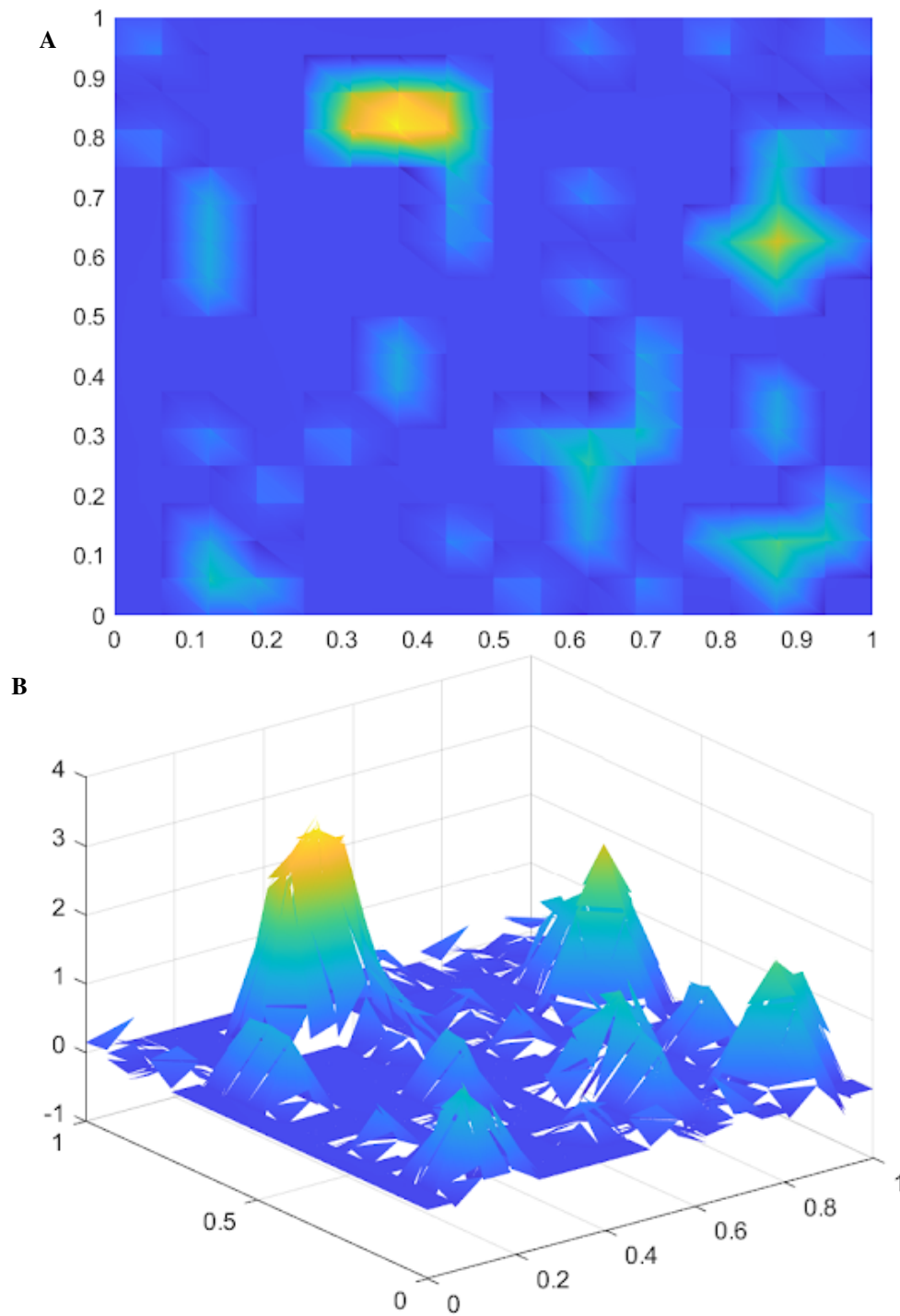


Figure 37. L_1 MsHDG solution for Test domain 6. (A) 2-D view of MsHDG approximation . (B) 3-D view of MsHDG approximation ($\times 10^{-3}$).

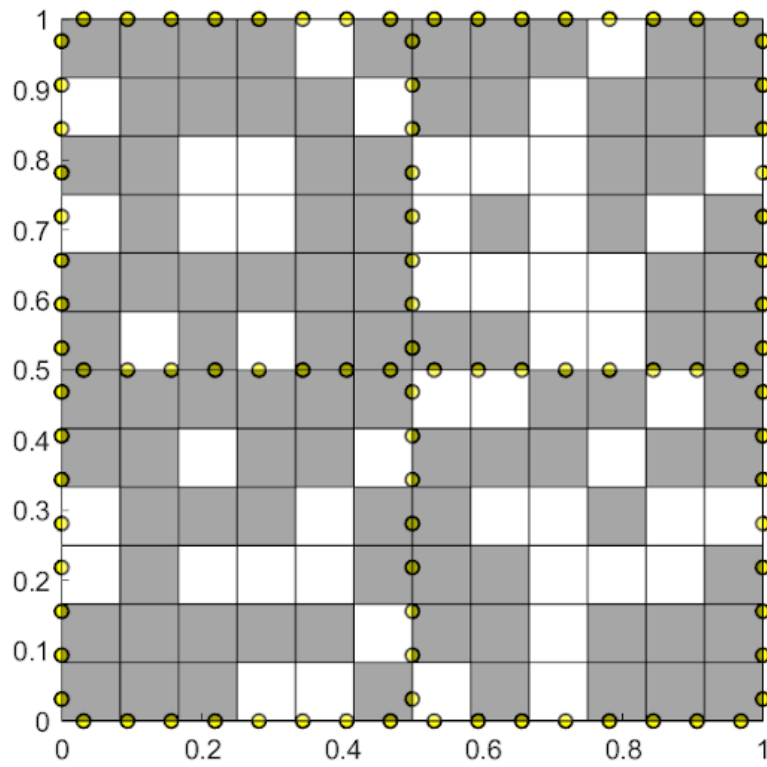


Figure 38. Test domain 7: $\kappa = 1$ in the white regions and $\kappa = 10^4$ in the dark/gray regions

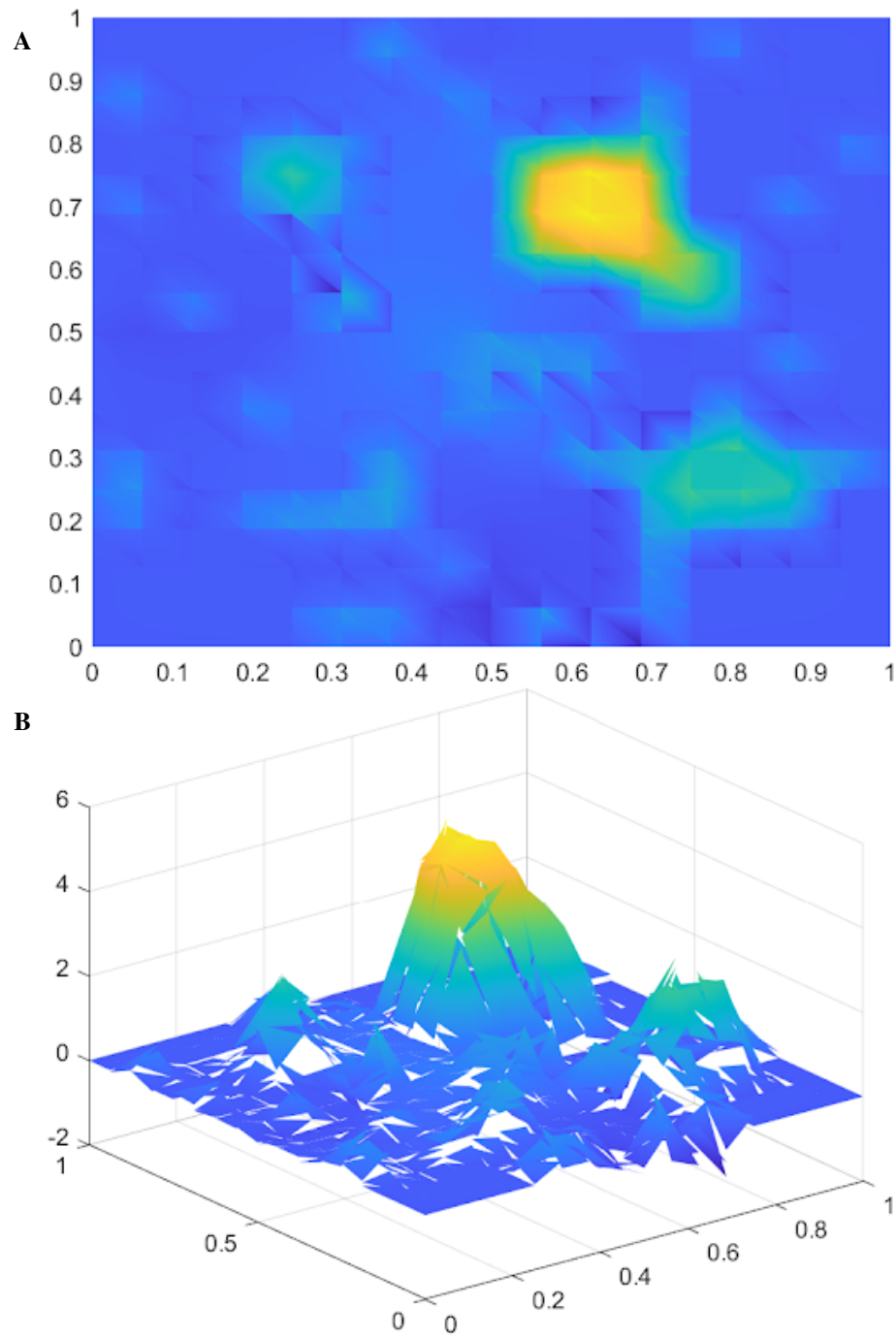


Figure 39. L_1 neural network solution for Test domain 7. (A) 2-D view of neural network approximation . (B) 3-D view of neural network approximation ($\times 10^{-3}$).

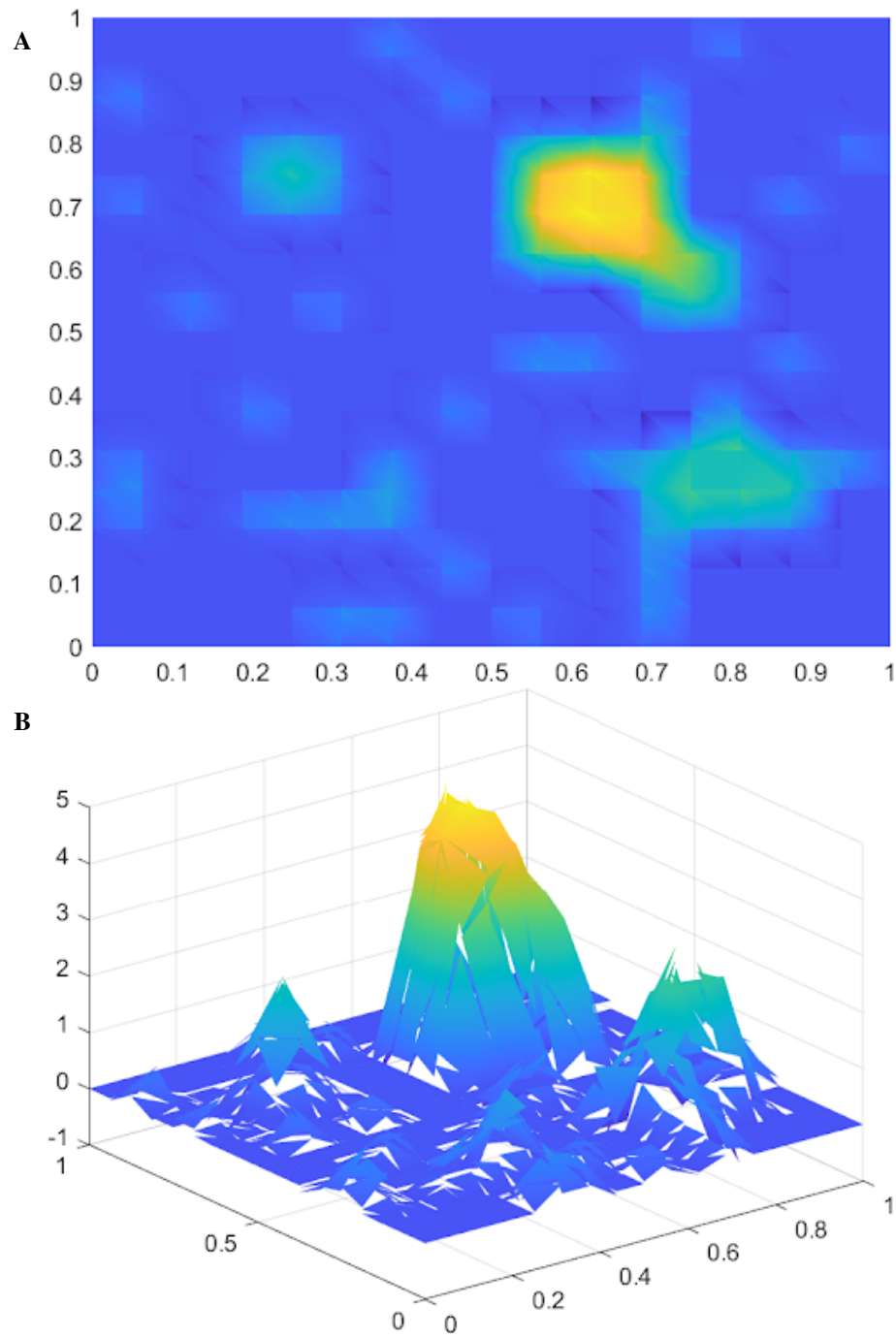


Figure 40. L_1 MsHDG solution for Test domain 7. (A) 2-D view of MsHDG approximation . (B) 3-D view of MsHDG approximation ($\times 10^{-3}$).

CHAPTER 7

CONCLUSION

The objective of this thesis was to solve the second-order elliptic differential equation, equation 4, defined on a bounded polyhedral domain with heterogeneous permeability coefficient κ . Numerical solution of such equation on a fine-scale may require a tremendous amount of memory and CPU time. Whenever it is affordable to resolve all the small scale features of a physical problem, direct solutions provide quantitative information of the physical processes at all scales. However, it is often sufficient to predict the macroscopic properties of the multiscale systems. This necessitates the use of reduced-order methods or specialized techniques which are capable of capturing the fine scale effects on the larger scale, without ever fully resolving all the fine scale features. We utilized local snapshots that incorporated local features of the solution space and employed multiscale basis functions to obtain better approximations through the process of coarse scale enrichment. Furthermore, neural network was incorporated to improve the efficiency of our method by training the network to learn the solution map between each coarse blocks. We state, that the size of the fine-scale domain (2x2) and (3x3) in the neural network section is chosen solely for demonstration purposes. It is expected that this method can be applied to a larger domain and higher scale levels (L_2 and L_3) of multiscale basis functions. Figure 21 - Figure 40 shows that we can obtain good approximations to the multiscale HDG solutions once the network parameters are fine-tuned to minimize the loss function. In Table 1, we can observe that as we enrich the coarse space, i.e., going from one level to the other, the error between the MsHDG approximations and the fine-scale solutions on the coarse edges decreases. It also shows the possibility of implementing

these methods in parallel. Table 2 shows that we can improve on the efficiency of our method by using neural network even as we increase the coarse scale dimensions.

REFERENCES

- [1] J. E. Aarnes, “On the use of a mixed multiscale finite element method for greater flexibility and increased speed or improved accuracy in reservoir simulation,” *Multiscale Model. Simul.*, vol. 2, pp. 421–439, 2004. [Online]. Available: <https://api.semanticscholar.org/CorpusID:39835273>.
- [2] J. E. Aarnes and Y. Efendiev, “Mixed multiscale finite element methods for stochastic porous media flows,” *SIAM J. Sci. Comput.*, vol. 30, pp. 2319–2339, 2008. [Online]. Available: <https://api.semanticscholar.org/CorpusID:19679668>.
- [3] T. Arbogast, “Analysis of a two-scale, locally conservative subgrid upscaling for elliptic problems,” *SIAM J. Numer. Anal.*, vol. 42, pp. 576–598, 2004. [Online]. Available: <https://api.semanticscholar.org/CorpusID:14268979>.
- [4] T. Arbogast, “Homogenization-based mixed multiscale finite elements for problems with anisotropy,” *Multiscale Modeling & Simulation*, vol. 9, no. 2, pp. 624–653, 2011. DOI: 10.1137/100788677.
- [5] T. Arbogast and K. J. Boyd, “Subgrid upscaling and mixed multiscale finite elements,” *SIAM J. Numer. Anal.*, vol. 44, pp. 1150–1171, 2006. [Online]. Available: <https://api.semanticscholar.org/CorpusID:2742225>.
- [6] T. Arbogast, L. C. Cowsar, M. F. Wheeler, and I. Yotov, “Mixed finite element methods on nonmatching multiblock grids,” *SIAM Journal on Numerical Analysis*, vol. 37, no. 4, pp. 1295–1315, 2000. DOI: 10.1137/S0036142996308447.

- [7] T. Arbogast, L. C. Cowsar, M. F. Wheeler, and I. Yotov, “Mixed finite element methods on nonmatching multiblock grids,” *SIAM J. Numer. Anal.*, vol. 37, pp. 1295–1315, 2000. [Online]. Available: <https://api.semanticscholar.org/CorpusID:14134911>.
- [8] T. Arbogast, G. Pencheva, M. F. Wheeler, and I. Yotov, “A multiscale mortar mixed finite element method,” *Multiscale Modeling & Simulation*, vol. 6, no. 1, pp. 319–346, 2007. DOI: 10.1137/060662587.
- [9] T. Arbogast, G. Pencheva, M. F. Wheeler, and I. Yotov, “A multiscale mortar mixed finite element method,” *Multiscale Modeling & Simulation*, vol. 6, no. 1, pp. 319–346, 2007. DOI: 10.1137/060662587.
- [10] T. Arbogast and H. Xiao, “A multiscale mortar mixed space based on homogenization for heterogeneous elliptic problems,” *SIAM Journal on Numerical Analysis*, vol. 51, no. 1, pp. 377–399, 2013. DOI: 10.1137/120874928.
- [11] D. Arnold and F. Brezzi, “Mixed and nonconforming finite element methods: Implementation, postprocessing and error estimates,” *RAIRO Modél. Math. Anal. Numér.*, vol. 19, pp. 7–32, Jan. 1985. DOI: 10.1051/m2an/1985190100071.
- [12] C. Bernardi, Y. Maday, and A. T. Patera, “Domain decomposition by the mortar element method,” in *Asymptotic and Numerical Methods for Partial Differential Equations with Critical Parameters*, H. G. Kaper, M. Garbey, and G. W. Pieper, Eds. Dordrecht: Springer Netherlands, 1993, pp. 269–286, ISBN: 978-94-011-1810-1. DOI: 10.1007/978-94-011-1810-1_17. [Online]. Available: https://doi.org/10.1007/978-94-011-1810-1_17.
- [13] C. Bernardi, Y. Maday, and A. T. Patera, “Domain decomposition by the mortar element method,” in *Asymptotic and Numerical Methods for Partial Differential Equations with*

- Critical Parameters*, H. G. Kaper, M. Garbey, and G. W. Pieper, Eds. Dordrecht: Springer Netherlands, 1993, pp. 269–286.
- [14] S. C. Brenner and L. R. Scott, *The Mathematical Theory of Finite Element Methods* (Texts in Applied Mathematics). Springer, 2008, vol. 15, ISBN: 9780387759333. DOI: 10.1007/978-0-387-75934-0. [Online]. Available: <http://dx.doi.org/10.1007/978-0-387-75934-0>.
- [15] F. Brezzi and M. Fortin, *Mixed and hybrid finite element methods*. Berlin, Heidelberg: Springer-Verlag, 1991, ISBN: 0387975829.
- [16] Z. Chen, C. A. Micchelli, and Y. Xu, “Multiscale basis functions,” in *Multiscale Methods for Fredholm Integral Equations* (Cambridge Monographs on Applied and Computational Mathematics), Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2015, pp. 144–198.
- [17] C.-C. Chu, I. G. Graham, and T. Y. Hou, “A new multiscale finite element method for high-contrast elliptic interface problems,” *Math. Comput.*, vol. 79, pp. 1915–1955, 2010. [Online]. Available: <https://api.semanticscholar.org/CorpusID:2155764>.
- [18] B. Cockburn and J. Gopalakrishnan, “New hybridization techniques,” *GAMM-Mitteilungen*, vol. 28, Nov. 2005. DOI: 10.1002/gamm.201490017.
- [19] B. Cockburn, J. Gopalakrishnan, and R. Lazarov, “Unified hybridization of discontinuous galerkin, mixed, and continuous galerkin methods for second order elliptic problems,” *SIAM Journal on Numerical Analysis*, vol. 47, no. 2, pp. 1319–1365, 2009. DOI: 10.1137/070706616.

- [20] B. Cockburn, J. Gopalakrishnan, and R. Lazarov, “Unified hybridization of discontinuous galerkin, mixed, and continuous galerkin methods for second order elliptic problems,” *SIAM J. Numer. Anal.*, vol. 47, pp. 1319–1365, Aug. 2009. DOI: 10.1137/070706616.
- [21] S. Du and S. N. Stechmann, *Element learning: A systematic approach of accelerating finite element-type methods via machine learning, with applications to radiative transfer*, 2023. arXiv: 2308.02467.
- [22] W. E and B. Engquist, “The heterogenous multiscale methods,” *Communications in Mathematical Sciences*, vol. 1, pp. 87–132, 2003. [Online]. Available: <https://api.semanticscholar.org/CorpusID:119805366>.
- [23] Y. Efendiev, J. Galvis, and T. Y. Hou, “Generalized multiscale finite element methods (gms-fem),” *Journal of Computational Physics*, vol. 251, pp. 116–135, 2013, ISSN: 0021-9991. DOI: <https://doi.org/10.1016/j.jcp.2013.04.045>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0021999113003392>.
- [24] Y. Efendiev, R. Lazarov, M. Moon, and K. Shi, “A spectral multiscale hybridizable discontinuous galerkin method for second order elliptic problems,” *Computer Methods in Applied Mechanics and Engineering*, vol. 292, pp. 243–256, 2015, Special Issue on Advances in Simulations of Subsurface Flow and Transport (Honoring Professor Mary F. Wheeler), ISSN: 0045-7825. DOI: <https://doi.org/10.1016/j.cma.2014.09.036>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0045782514003594>.
- [25] Y. Efendiev, R. Lazarov, and K. Shi, “A multiscale hdg method for second order elliptic equations. part i. polynomial and homogenization-based multiscale spaces,” *SIAM Journal on Numerical Analysis*, vol. 53, no. 1, pp. 342–369, 2015. DOI: 10.1137/13094089X.

- [26] Y. R. Efendiev, J. Galvis, and X.-h. Wu, “Multiscale finite element methods for high-contrast problems using local spectral basis functions,” *J. Comput. Phys.*, vol. 230, pp. 937–955, 2011. [Online]. Available: <https://api.semanticscholar.org/CorpusID:3234440>.
- [27] Y. R. Efendiev, T. Y. Hou, and V. Ginting, “Multiscale finite element methods for non-linear problems and their applications,” *Communications in Mathematical Sciences*, vol. 2, pp. 553–589, 2004. [Online]. Available: <https://api.semanticscholar.org/CorpusID:555240>.
- [28] B. Ganis and I. Yotov, “Implementation of a mortar mixed finite element method using a multiscale flux basis,” *Computer Methods in Applied Mechanics and Engineering*, vol. 198, no. 49, pp. 3989–3998, 2009, ISSN: 0045-7825. DOI: <https://doi.org/10.1016/j.cma.2009.09.009>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0045782509003077>.
- [29] T. Y. Hou and Y. R. Efendiev, “Multiscale finite element methods: Theory and applications,” 2009. [Online]. Available: <https://api.semanticscholar.org/CorpusID:60298776>.
- [30] T. Y. Hou and X.-H. Wu, “A multiscale finite element method for elliptic problems in composite materials and porous media,” *Journal of Computational Physics*, vol. 134, no. 1, pp. 169–189, 1997, ISSN: 0021-9991. DOI: <https://doi.org/10.1006/jcph.1997.5682>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0021999197956825>.
- [31] T. J. Hughes, G. R. Feijóo, L. Mazzei, and J.-B. Quincy, “The variational multiscale method—a paradigm for computational mechanics,” *Computer Methods in Applied Mechanics and Engineering*, vol. 166, no. 1, pp. 3–24, 1998, Advances in Stabilized Methods in Computa-

- tional Mechanics, ISSN: 0045-7825. DOI: [https://doi.org/10.1016/S0045-7825\(98\)00079-6](https://doi.org/10.1016/S0045-7825(98)00079-6). [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0045782598000796>.
- [32] O. P. Iliev, R. D. Lazarov, and J. Willems, “Variational multiscale finite element method for flows in highly porous media,” *Multiscale Model. Simul.*, vol. 9, pp. 1350–1372, 2011. [Online]. Available: <https://api.semanticscholar.org/CorpusID:15104313>.
- [33] P. Jenny, S. Lee, and H. Tchelepi, “Multi-scale finite-volume method for elliptic problems in subsurface flow simulation,” *Journal of Computational Physics*, vol. 187, no. 1, pp. 47–67, 2003, ISSN: 0021-9991. DOI: [https://doi.org/10.1016/S0021-9991\(03\)00075-5](https://doi.org/10.1016/S0021-9991(03)00075-5). [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0021999103000755>.
- [34] P. Raviart and J. Thomas, “A mixed finite element method for second order elliptic problems,” vol. 606, Jan. 1977.
- [35] W. Reed and T. Hill, “Triangular mesh methods for the neutron transport equation,” [*No source information available*], Jan. 1973.
- [36] Wheeler, Mary Fanett, Xue, Guangri, and Yotov, Ivan, “A multiscale mortar multipoint flux mixed finite element method,” *ESAIM: M2AN*, vol. 46, no. 4, pp. 759–796, 2012. DOI: [10.1051/m2an/2011064](https://doi.org/10.1051/m2an/2011064). [Online]. Available: <https://doi.org/10.1051/m2an/2011064>.