Discontinuous Galerkin Methods for Wave Equations: the MLP Estimator for the TVB Constant in Limiters and Local DG Methods for a Carpet Cloak Model

by

Xinyue Yu

B.Sc., UCLA, Los Angeles, United States, 2017

A dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy
in the Division of Applied Mathematics at Brown University

PROVIDENCE, RHODE ISLAND

May 2022

This dissertation by Xinyue Yu is accepted in its present form
by the Division of Applied Mathematics as satisfying the
dissertation requirement for the degree of Doctor of Philosophy.

Date

Chi-Wang Shu, Ph.D., Advisor

Recommended to the Graduate Council

Date

Mark Ainsworth, Ph.D., Reader

Date

Johnny Guzmán, Ph.D., Reader

Approved by the Graduate Council

Date

Andrew G. Campbell, Dean of the Graduate School

Vita

Education

- Brown University, Providence, RI, USA
  - Ph.D. candidate in Applied Mathematics Estimated 2022
    Advisor: Chi-Wang Shu
  - M.Sc. in Applied Mathematics 2018
- UCLA, Los Angeles, CA, USA
  - B.Sc. in Applied Mathematics 2017

Publications

1. **X. Yu** and C.-W. Shu, Multi-layer perceptron estimator for the total variation bounded constant in limiters for discontinuous Galerkin methods, La Matematica: Official Journal of the Association for Women in Mathematics, v1 (2022), pp.53-84. DOI: 10.1007/s44007-021-00004-9.

2. **X. Yu**, J. Li and C.-W. Shu, Local discontinuous Galerkin methods for a time-domain carpet cloak model, Annals of Mathematical Sciences and Applications, v7 (2022), pp.97-137. DOI: 10.4310/AMSA.2022.v7.n1.a4.

Teaching Experience

- Teaching Assistant
  - Statistical Inference II, Brown University Spring 2019
  - Applied Ordinary Differential Equations, Brown University Fall 2018

Acknowledgments

First, it is a genuine pleasure to express my deepest sense of thanks and gratitude to my thesis advisor, Professor Chi-Wang Shu. Professor Shu gives me the opportunity to start my research, and offers me invaluable guidance during my graduate studies. The knowledge he taught me widens and deepens my understanding of the field, and his rigorous attitude towards the research greatly influences me and helps me build up my own methodology and criterion in research. Moreover, his optimism always inspires and encourages me to face the obstacles in the life. I cannot imagine a better advisor in my life.

I'm glad to have a chance to work with Professor Jichun Li. I have learnt a lot from Professor Li about the electromagnetics in metamaterials, and I'd like to thank him for his professional suggestions and guidance during our collaboration.

I'm honored to have Professor Mark Ainsworth and Professor Johnny Guzmán in my committee. Professor Ainsworth and Professor Guzmán taught me the spectral methods and the finite element methods respectively, helping me build a strong background in the field of numerical analysis. Many thanks to them for spending time reading through my thesis, and providing invaluable feedback.

I would like to thank my teachers, fellow students, and friends, including Professor Bjorn Sandstede, Professor Jerome Darbon, Professor Hongjie Dong, Sun Zheng, Tianheng Chen, Kunrui Wang, Xuefei Cao, Zongyuan Li, Yinting Liao and many others, for their support and help during my Ph.D period. Especially, I want to thank my best friends Yixiang Deng, Yue Li and Tingwei Meng, for their company and encouragement during my hard time.

Last but not the least, I would like to thank my parents for their love and support in my life. My father is my life mentor, and his integrity and dedication towards work always inspire me, while my mom is always behind me, giving me courage to fight against the troubles. It would not have been possible for me to achieve this dissertation without the support from them!

Abstract of “Discontinuous Galerkin Methods for Wave Equations: the MLP Estimator for the TVB Constant in Limiters and Local DG Methods for a Carpet Cloak Model”, by Xinyue Yu, Ph.D., Brown University, May 2022

This thesis contains two parts, including the development of a modified total variation bounded (TVB) limiter applied to the discontinuous Galerkin (DG) methods, and the application of the local DG (LDG) method to solve the carpet cloak model.

The DG method was initially proposed by Reed and Hill to solve the neutron transport problem. Later, Cockburn and Shu introduced the Runge-Kutta DG (RKDG) methods for solving the linear and nonlinear hyperbolic partial differential equations (PDEs), and the LDG methods for solving the time-dependent convection-diffusion systems, which stimulated the rapid development and application of the DG methods. The DG method is widely used in numerical solution of partial differential equations because of its nice features, such as the flexible h-p adaptivity, easy handling of the complicated geometry, easy handling of hanging nodes and adaptivity, and high parallel efficiency.

Although the DG method has many good properties, for problems containing strong shocks, the DG method often needs to be supplemented by a limiter to control spurious oscillations and to ensure nonlinear stability. The TVB limiter is a popular choice and can maintain the original high order accuracy of the DG scheme in smooth regions and keep a sharp and non-oscillatory discontinuity transition, when a certain TVB constant M is chosen adequately. For scalar conservation laws, suitable choice of this constant M can be based on solid mathematical analysis. However, for nonlinear hyperbolic systems, there is no rigorous mathematical guiding principle for the determination of this constant, and numerical experiments often use *ad hoc* choices based on experience and through trial and

error. Our first topic is to develop a TVB constant artificial neural network (ANN) based estimator by constructing a multi-layer perceptron (MLP) model. We generate the training data set by constructing piecewise smooth functions containing local maxima, local minima, and discontinuities. By using the supervised learning strategy, the MLP model is trained offline. The proposed method gives the TVB constant M with robust performance to capture sharp and non-oscillatory shock transitions while maintaining the original high order accuracy in smooth regions. Numerical results using this new estimator in the TVB limiter for DG methods in one and two dimensions are given, and its performance is compared with the classical *ad hoc* choices of this TVB constant.

In the second part, we introduce the leap-frog LDG methods to solve the carpet cloak model. We prove the stability of the semi-discrete scheme, the suboptimal error estimate for unstructured meshes, and the optimal error estimate for tensor-product meshes. Then, the fully discrete scheme is stated and the stability is proved. Finally, the numerical accuracy tests on rectangular and triangular meshes are given respectively, and the results of numerical simulations of the wave propagation in the carpet cloak model using the DG scheme are presented.

Contents

Chapter One

Introduction

The discontinuous Galerkin (DG) method was initially proposed by Reed and Hill [71] to solve the neutron transport problem. Later, Cockburn and Shu introduced the Runge-Kutta DG (RKDG) methods for solving the linear and nonlinear hyperbolic partial differential equations (PDEs) [16, 17], and the local DG (LDG) methods for solving the time-dependent convection-diffusion systems [18], which stimulated the rapid development and application of the DG methods [19, 85]. The DG method shares the advantages of the continuous finite element methods, including flexible h-p adaptivity and easy handling of the complicated geometry. Additionally, it has unique nice features, such as it has the local mass matrix because of the discontinuous basis, it allows easy handling of hanging nodes and adaptivity, and it has high parallel efficiency.

In this dissertation, we present two topics concerning the development of the limiters applied on the DG methods, and the application of the LDG methods on the carpet cloak model.

As is well known, the solution of nonlinear conservation laws often generates discontinuities, even with smooth initial and boundary conditions. Although the DG method can be proved to be L^2 and entropy stable for nonlinear hyperbolic scalar equations and systems [36, 35, 8, 9], this does not prevent the numerical solution from generating spurious oscillations near discontinuities. These oscillations are unpleasant in visualization, and, more seriously, they may lead to nonlinear instability for hyperbolic systems since hyperbolicity may be lost when such oscillations bring the numerical solution outside of the physical constraints (e.g. the appearance of negative density or pressure for compressible gas dynamics). To control these oscillations, nonlinear limiters are often used. They might be applied in specific cells using shock detectors (also called troubled cell indicators), such as the KXRCF shock detector developed by Krivodonova et al. [41], the troubled cell indicator of Fu and Shu [24], and the artificial neural network (ANN) based troubled cell indicator [69]. They may also be applied everywhere, with a careful design attempting to retain the original high order accuracy in smooth regions. Examples include the minmod-based total variation diminishing (TVD) limiters [31, 62], the minmod-based total variation bounded (TVB) limiter [75], the moment limiter [3], the monotonicity-preserving limiter [79], and the weighted essentially non-oscillatory (WENO) limiter [66]. A summary and comparison of limiters can be found in [89].

One drawback of many of the limiters, including the popular minmod-based TVD limiters [31, 62], is that they may degenerate to first order accuracy near smooth extrema, even though they could retain the original high order accuracy in smooth and monotone regions [63]. To overcome this difficulty, Shu [75] designed a minmod-based TVB limiter, which can retain the original high order accuracy in smooth regions, including regions near smooth extrema. The adaptation and application of this TVB limiter to DG methods for solving scalar one-dimensional hyperbolic conservation laws were carried out in [16], and this limiter was further extended to DG methods solving one-dimensional systems and multidimensional cases in [15, 14, 17]. Comparing with the minmod-based TVD limiters, this TVB limiter significantly improves accuracy in smooth regions near solution extrema. However, it involves a TVB parameter M, which must be determined in a problem-dependent fashion. In the two extremes, $M = 0$ returns to the TVD limiter, and $M = +\infty$ returns to the original scheme without any limiter. If M is chosen too small, accuracy near smooth extrema might be affected; while if M is chosen too large, noticeable spurious oscillations may reappear near discontinuities. For scalar nonlinear conservation laws, there exists rigorous mathematical guidance on the choice of M to guarantee that accuracy is maintained in smooth regions

[75, 16]. However, for nonlinear systems, no such mathematical guidance exists, and hence in practice, M is usually chosen in an *ad hoc* fashion based on experience and through trial and error. With proper choices of the TVB constant M, DG schemes with the TVB limiter can give excellent resolution in the computational fluid dynamics simulations. Besides the examples for compressible gas dynamics in [15, 17], we could also mention the application in [45], combined with a wet-dry moving boundary treatment, for solving shallow water equations. Also for solving shallow water equations, it works well on unstructured triangular meshes [84]. The TVB limiter is used to indicate the troubled cells in the application of special relativistic hydrodynamics [88]. Effort has also been made to provide guidance for an automated choice of the TVB constant M. A unified approach for the determination of this constant in mixed type meshes was studied and applied by Kontzialis et al. [40] and by Panourgias et al. [64], where M was chosen according to the variation of the derivatives of the numerical solution. In [80], Vuik and Ryan proposed an automatic parameter selection strategy for this TVB constant M based on Tukey's boxplot method of outlier-detection, and its application with compact-WENO finite element method is shown in [25].

Our first topic is to introduce an artificial neural network (ANN) based estimator for this TVB constant M by constructing a multi-layer perceptron (MLP) model. ANNs have the ability to approximate mappings with high-level complexity and nonlinearity, and thus they have undergone rapid developments and applications in numerical computation in recent years. For example, the ANNs are studied to solve ordinary and partial differential equations [42, 26, 73]. The multi-layer perceptron (MLP) is one of the most widely-used ANN models. It consists of an input layer, an output layer, and functional hidden layers. In [69, 70], Ray and Hesthaven constructed a troubled-cell indicator based on the MLP model, and

Wen et al. [82] applied it in finite difference WENO methods. A well trained MLP model is free of problem-dependent parameter and hence suitable to be used as a unified approach for determining the TVB constant M in the TVB limiter applied to DG methods solving general conservation laws. In Chapter 2, we will briefly introduce the DG methods and the TVB limiter, and in Chapter 3 we propose the design of our MLP based estimator for the TVB constant. Addtionally, the good performance of the MLP-based TVB limiter in comparison with the *ad hoc* choice of the TVB constant M will be provided in Chapter 3.

The second topic is about applying the LDG methods to numerically solve the carpet cloak model. Since Leonhardt [46] and Pendry *et al.* [65] firstly demonstrated the idea of invisibility cloak design with metamaterials in 2006, much study has been done in both theoretical and numerical analysis. There are plenty of excellent works on the mathematical analysis of the cloaking phenomenon [1, 39, 27, 28], and on the numerical simulations of the cloaking models with the finite different (FD) methods [30, 34, 56], the finite element (FE) methods [5, 45, 50, 60], and the spectral methods [86, 87]. For more details, readers can consult the review papers [2, 7, 33], and the monographs [21, 32, 49, 59] as references. In 2014, Li *et al.* [50] proposed the mathematical analysis for the time-domain carpet cloak model.

Attracted by the good properties of the DG methods, mathematicians have also developed the DG methods to solve the Maxwell equations in the metamaterials. There are published works on the DG methods to solve the Drude models [47, 48, 53, 55, 74], the Maxwell equations in nonlinear optical media [4], and the wave propagation in media with dielectrics and metamaterials [11]. In [52], the DG method was first carried out to solve the carpet cloak model, and it gave a good performance in numerical simulations. However, the stability analysis and the error estimate of the method were left to be done. As a follow-up work of [52], we prove the stability for the DG methods solving the carpet cloak model, and we also give the proof of optimal convergence rates on rectangular meshes, and sub-optimal convergence rates on triangular meshes. The introduction of the carpet cloak model and the theoretical analysis for the semi-discrete LDG methods to solve the carpet cloak model will be provided in the Chapter 4. In Chapter 5, we will show the stability analysis and the numerical simulations of the fully discrete LDG methods solving the carpet cloak model respectively.

Chapter Two

Introduction to the discontinuous Galerkin (DG) methods and the total variation bounded (TVB) limiters

2.1 The DG methods

We consider the following conservation law:

$$\begin{cases} u_t + \nabla \cdot F(u) = 0, \quad \text{on } \Omega \subset \mathbb{R}^d, \quad d = 1, 2, \\ u(\cdot, 0) = u_0(\cdot), \end{cases} \tag{2.1}$$

where F is a linear or nonlinear flux function and Ω is a bounded domain in $\mathbb{R}^d$. In the one dimensional case, the conservation law is

$$\begin{cases} u_t + f(u)_x = 0, \quad \text{on } \Omega \subset \mathbb{R}, \\ u(x, 0) = u_0(x), \end{cases} \tag{2.2}$$

where $\Omega = [a, b]$. We discretize the domain by the partition $a = x_{1/2} < x_{3/2} < \cdots < x_{N+1/2} = b$. The cell I_i is denoted as $I_i = \{x : x_{i-1/2} < x < x_{i+1/2}\}$, for $1 \leq i \leq N$, and the mesh sizes are $h_i = x_{i+1/2} - x_{i-1/2}$. We define a piecewise continuous polynomial space

$$V_h^k = \{p \in L_2(\Omega) : p|_{I_i} \in P^k(I_i)\},$$

where $P^k(I_i)$ is the space of polynomials of degree $\leq k$ in I_i. Then the one-dimensional DG method is stated as follows: Find $u_h(\cdot, t) \in V_h^k$, such that for all $v_h \in V_h^k$, u_h satisfies:

$$\frac{d}{dt}\int_{I_i} u_h(x,t)v_h(x,t)\,dx - \int_{I_i} f(u_h(x,t))(v_h(x,t))_x\,dx + \hat{f}_{i+\frac{1}{2}}\,v_h(x^-_{i+\frac{1}{2}}, t) - \hat{f}_{i-\frac{1}{2}}\,v_h(x^+_{i-\frac{1}{2}}, t) = 0, \tag{2.3}$$

where $\hat{f}_{i+\frac{1}{2}} = \hat{f}(u_h(x^-_{i+\frac{1}{2}}, t), u_h(x^+_{i+\frac{1}{2}}, t))$ is a monotone numerical flux in the scalar case and an exact or approximate Riemann-solver based numerical flux in the system case, see [16, 15].

To implement the DG method, one can use a local basis over I_i: $\boldsymbol{v}_i = (v_i^0, \dots, v_i^k)^T$, and the numerical solution is expressed as

$$u_h(x,t) = \sum_{\ell=0}^{k} u_i^\ell(t) v_i^\ell(x), \quad \text{for} \quad x \in I_i. \tag{2.4}$$

The time dependent coefficients $\boldsymbol{u}_i(t) = (u_i^0(t), \dots, u_i^k(t))^T$ are the computational variables to be evolved in time. If we take the test functions as $v_h = v_i^l$, $l = 0, \dots, k$, the scheme can be written as

$$\sum_{\ell=0}^{k} \frac{du_i^\ell}{dt} \int_{I_i} v_i^l v_i^\ell dx = \int_{I_i} f\left(\sum_{\ell=0}^{k} u_i^\ell v_i^\ell\right) (v_i^l)_x \, dx - \hat{f}_{i+\frac{1}{2}} \, v_i^l(x_{i+\frac{1}{2}}) + \hat{f}_{i-\frac{1}{2}} \, v_i^l(x_{i-\frac{1}{2}}), \quad l = 0, \dots, k. \tag{2.5}$$

The integrals in (2.5) can be computed either exactly or via suitable quadratures. The coefficients $\boldsymbol{u}_i$ can be obtained by using a proper time discretization to solve the ordinary differential equation (ODE) (2.5). In this and the next Chapters, we will use the third order Runge-Kutta scheme (RK3) [76] in the computation. Denote $\boldsymbol{U}(t) = (\boldsymbol{u}_1(t), \dots, \boldsymbol{u}_N(t))^T$, the equation (2.5) can be written as

$$\frac{d}{dt}\boldsymbol{U}(t) = L(\boldsymbol{U(t)}),$$

where L is the spatial discretization operator. With $\boldsymbol{U}^n = \boldsymbol{U}(t_n)$, where t_n is n-th time step, the third order Runge-Kutta scheme is stated as follows:

$$\begin{aligned} \boldsymbol{U}^{(1)} &= \boldsymbol{U}^n + \Delta t L(\boldsymbol{U}^n), \\ \boldsymbol{U}^{(2)} &= \tfrac{3}{4}\boldsymbol{U}^n + \tfrac{1}{4}(\boldsymbol{U}^{(1)} + \Delta t L(\boldsymbol{U}^{(1)})), \\ \boldsymbol{U}^{n+1} &= \tfrac{2}{3}\boldsymbol{U}^n + \tfrac{1}{3}(\boldsymbol{U}^{(2)} + \Delta t L(\boldsymbol{U}^{(2)})). \end{aligned} \tag{2.6}$$

In the two dimensional case, the conservation law becomes

$$u_t + f(u)_x + g(u)_y = 0, \quad \text{on } \Omega \subset \mathbb{R}^2. \tag{2.7}$$

We consider the simple box geometry, and let $\Omega = [a_x, b_x] \times [a_y, b_y]$. Likewise, for simplicity of presentation, we use a rectangular mesh to cover the domain, consisting of the cells $I_{ij} = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}]$ for $1 \leq i \leq N_x$ and $1 \leq j \leq N_y$. Similar to the 1D case, we define

$$V_h^k = \{p \in L_2(\Omega) : p|_{I_{ij}} \in P^k(I_{ij})\},$$

where $P^k(I_{ij})$ is the set of polynomials of degree $\leq k$ over the cell I_{ij}. Recall the notation in (2.1) that $F(u) = (f(u), g(u))$. The 2D DG method is stated as follows: Find $u_h(\cdot, t) \in V_h^k$, such that for all $v_h \in V_h^k$, u_h satisfies:

$$\begin{aligned} \frac{d}{dt}\int_{I_{ij}} u_h(x,y,t)v_h(x,y)dxdy - \int_{I_{ij}} F(u_h(x,y,t)) \cdot \nabla v_h(x,y)dxdy \\ + \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \hat{g}_{i,j+\frac{1}{2}}\, v_h(x, y^-_{j+\frac{1}{2}})dx - \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \hat{g}_{i,j-\frac{1}{2}}\, v_h(x, y^+_{j-\frac{1}{2}})dx \\ + \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \hat{f}_{i+\frac{1}{2},j}\, v_h(x^-_{i+\frac{1}{2}}, y)dy - \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \hat{f}_{i-\frac{1}{2},j}\, v_h(x^+_{i-\frac{1}{2}}, y)dy = 0, \end{aligned} \tag{2.8}$$

where $\hat{f}_{i+\frac{1}{2},j} = \hat{f}(u_h(x^-_{i+\frac{1}{2}}, y, t), u_h(x^+_{i+\frac{1}{2}}, y, t))$ is a one-dimensional numerical flux as defined before, likewise for $\hat{g}_{i,j+\frac{1}{2}}$. Consider a proper local basis over I_{ij}: $\boldsymbol{v}_{ij} = (v^0_{ij}, \ldots, v^K_{ij})$ where $K = (k+1)(k+2)/2$, then the numerical solution is expressed as

$$u_h(x,y,t) = \sum_{\ell=0}^{K} u^\ell_{ij}(t)v^\ell_{ij}(x,y), \quad \text{for} \quad (x,y) \in I_{ij}. \tag{2.9}$$

Define the coefficients as $\boldsymbol{u}_{ij} = (u^0_{ij}, \ldots, u^K_{ij})$, and take the test functions as $v_h =$

v_i^l, $l = 0, \dots, K$, then the scheme can be written as

$$\begin{aligned}\sum_{\ell=0}^{k} \frac{du_{ij}^{\ell}}{dt} \int_{I_{ij}} v_{ij}^l v_{ij}^{\ell} dx = &\int_{I_{ij}} F(\sum_{\ell=0}^{K} u_{ij}^{\ell}(t) v_{ij}^{\ell}(x,y)) \cdot \nabla v_{ij}^l dxdy \\ &- \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \hat{g}_{i,j+\frac{1}{2}} v_{ij}^l (x, y_{j+\frac{1}{2}}^-) dx + \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \hat{g}_{i,j-\frac{1}{2}} v_{ij}^l (x, y_{j-\frac{1}{2}}^+) dx \\ &- \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \hat{f}_{i+\frac{1}{2},j} v_{ij}^l (x_{i+\frac{1}{2}}^-, y) dy + \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \hat{f}_{i-\frac{1}{2},j} v_{ij}^l (x_{i-\frac{1}{2}}^+, y) dy.\end{aligned} \tag{2.10}$$

Again, the coefficients $\boldsymbol{u}_{ij}(t)$ can be obtained by solving the ODE (2.10) by the third order Runge-Kutta time discretization (2.6).

2.2 The minmod-based TVB limiter

As mentioned in the introduction, the DG scheme provides high order accurate simulation of smooth solutions, and maintains L^2 and entropy stability for discontinuous solutions. However, this does not prevent the DG solution from showing spurious Gibbs oscillations near discontinuities, which may lead to nonlinear instability for solving nonlinear hyperbolic systems. Various nonlinear limiters are designed in the literature to control those spurious oscillations, while attempting to retain the original high order accuracy in smooth regions. In this section we will introduce the minmod-based TVB limiter [75, 16].

In the one dimensional case, we denote the cell average of u_h in each cell I_i as:

$$\bar{u}_i = \frac{1}{h_i} \int_{I_i} u_h(x) dx.$$

We further denote by $\tilde{u}_i$ and $\tilde{\tilde{u}}_i$ the differences between the point values of the numerical solution at the cell boundaries and the cell average, and by $\Delta^+\bar{u}_i$ and $\Delta^-\bar{u}_i$ the differences between the cell average of I_i and that of its neighboring cells:

$$\tilde{u}_i = u_h(x^-_{i+\frac{1}{2}}) - \bar{u}_i, \quad \tilde{\tilde{u}}_i = \bar{u}_i - u_h(x^+_{i-\frac{1}{2}}), \quad \Delta^+\bar{u}_i = \bar{u}_{i+1} - \bar{u}_i, \quad \Delta^-\bar{u}_i = \bar{u}_i - \bar{u}_{i-1}. \quad (2.11)$$

A nonlinear limiter changes the polynomial solution u_h in the cell I_i, while keeping the cell average $\bar{u}_i$ unchanged to maintain conservation. The purpose of the nonlinear limiter is to control spurious oscillations near discontinuities, while attempting to retain the original high order accuracy in smooth regions. The minmod-based TVD limiter [31, 62] modifies $\tilde{u}_i$ and $\tilde{\tilde{u}}_i$ by a limiter function:

$$\tilde{u}_i^{mod} = m(\tilde{u}_i, \Delta^+\bar{u}_i, \Delta^-\bar{u}_i), \qquad \tilde{\tilde{u}}_i^{mod} = m(\tilde{\tilde{u}}_i, \Delta^+\bar{u}_i, \Delta^-\bar{u}_i). \quad (2.12)$$

Once the modified values $\tilde{u}_i^{(mod)}$ and $\tilde{\tilde{u}}_i^{(mod)}$ are obtained, we can obtain the modified point values of the numerical solution at the cell boundaries:

$$u_h^{(mod)}(x^-_{i+\frac{1}{2}}) = \bar{u}_i + \tilde{u}_i^{(mod)}, \qquad u_h^{(mod)}(x^+_{i-\frac{1}{2}}) = \bar{u}_i - \tilde{\tilde{u}}_i^{(mod)}.$$

With the two modified point values $u_h^{(mod)}(x^-_{i+\frac{1}{2}})$, $u_h^{(mod)}(x^+_{i-\frac{1}{2}})$ and the original cell average $\bar{u}_i$, we can recover a unique p^k polynomial with $k \leq 2$ as the limited solution $u_h^{(mod)}$. For $k > 2$, we still recover a quadratic polynomial if the limiter is enacted (that is, if the limiter function m in (2.12) returns other than the first argument), since accuracy is not expected to be maintained in this case.

We now turn to the specific choices of the limiter function m in (2.12).

For the minmod-based TVD limiter [31, 62], m is defined as the minmod function

$$m(a_1, a_2, a_3) = \begin{cases} s\min(|a_1|, |a_2|, |a_3|), & \text{if } s = \text{sign}(a_1) = \text{sign}(a_2) = \text{sign}(a_3), \\ 0, & \text{otherwise.} \end{cases} \tag{2.13}$$

In words, the minmod function m returns the smallest argument (in magnitude), if all arguments have the same sign; otherwise it returns zero.

It can be proved [16] that, when the minmod limiter (2.13) is used and if the time discretization is via a TVD Runge-Kutta method such as (2.6), then the limited DG solution is total variation diminishing in the means (TVDM). This is a rather strong nonlinear stability property and prevents completely any spurious oscillations in the means near discontinuities. However, the drawback is that, as any TVD schemes, the method will suffer from accuracy degeneracy to first order near smooth extrema [63], hence the global accuracy in L^1 is at most second order for generic smooth solutions with finitely many smooth extrema.

For the minmod-based TVB limiter [75], m is defined as

$$m^{tvb}(a_1, a_2, a_3, h, M) = \begin{cases} a_1, & \text{if } |a_1| \leq Mh^2, \\ m(a_1, a_2, a_3), & \text{otherwise,} \end{cases} \tag{2.14}$$

where h is a local mesh size, $M \geq 0$ is a TVB constant, and m is the minmod function defined in (2.13). It can be shown [16] that, when the TVB limiter (2.14) is used and if the time discretization is via a TVD Runge-Kutta method such as (2.6), then the limited DG solution is total variation bounded in the means (TVBM). This is again a rather strong nonlinear stability property.

It is expected that the performance of the TVB limiter depends strongly on the choice of the TVB constant M. If M is chosen too large, noticeable spurious oscillations may reappear near discontinuities. After all, for $M = +\infty$, the limiter m^{tvb} in (2.14) will always return the first argument, namely we will obtain the unlimited solution. On the other hand, if M is chosen too small, the scheme may lose the original high-order accuracy near smooth extrema, just like the TVD minmod limiter. After all, for $M = 0$, we recover the TVD minmod limiter defined in (2.13). On the approximation level, given a smooth function u, the following result is proved in [16].

Lemma 2.2.1. If u is a smooth function, and $M_2 = \max_x |u_{xx}|$. Then, if M is taken as

$$M \geq \frac{2}{3} M_2, \tag{2.15}$$

the limiter (2.14) will not affect accuracy. That is, it will always return the first argument.

In fact, M_2 can be taken as a upper bound for the magnitude of the second derivative near the smooth extrema, rather than over the whole range of x.

The approximation result in the lemma above is also valid for linear or nonlinear scalar conservation laws. For one dimensional scalar conservation laws (2.2), we have the following lemma.

Lemma 2.2.2. If u is the solution of the one dimensional scalar conservation law (2.2), the initial condition $u_0(x)$ is smooth near $x = x_0$, and $u_0'(x_0) = 0$, then along the forward characteristic line

$$x(t) = x_0 + f'(u_0(x_0))t,$$

we have

$$u(x,t) = u(x_0), \qquad u_x(x,t) = 0, \qquad u_{xx}(x,t) = u_0''(x_0).$$

That is, along a smooth local extremum, the second derivative u_{xx} is invariant (constant in time).

Lemma 2.2.2 can be easily proved by solving the ODEs involving the evolution of u, u_x and u_{xx} along the forward characteristic line. Based on Lemmas 2.2.1 and 2.2.2, we conclude that the choice of M by (2.15), where $M_2 = \max_x |u_0''(x)|$, ensures that the limiter (2.14) will not affect accuracy. That is, it will always return the first argument. Thus, the choice of M to ensure high order accuracy in smooth regions for scalar conservation laws can be given with solid mathematical justification. In practice, because of the numerical errors near smooth extrema, we often take a slightly larger value of M than that given by (2.15), e.g. by $M = cM_2$ with $c > \frac{2}{3}$.

However, for nonlinear hyperbolic systems, there is no such mathematical guidance for the choice of the TVB constant M. This is because the value of u_{xx} at a smooth extremum is no longer invariant in time, hence cannot be determined based solely on the initial condition. The choice of M in such cases is then often given in an *ad hoc* fashion, based on experience and through trial and error. In the next Chapter, we would like to develop a constant estimator for M, based on an artificial neural network (ANN) based model, so that the TVB constant M can be chosen automatically .

Chapter Three

The multi-layer perceptron (MLP) limiter

3.1 The construction of the multi-layer perceptron (MLP) limiter

Our work on the construction of an ANN-based constant estimator is enlightened by the MLP troubled cell indicator developed by Hesthaven and Ray [69], which detects the location of discontinuities according to the function values at the cell boundaries and the local cell averages. Inspired by [69], we aim at constructing a constant estimator using the ANN model, which is able to (1) distinguish the cells near local extrema, discontinuities and in smooth monotone regions by the point values at the cell interface and the cell averages; (2) directly return the TVB limiter constant M accordingly, that maintains high order accuracy in smooth regions and non-oscillatory transaction at discontinuities. The multi-layer perceptron (MLP) model is one of the most commonly used artificial neural network model. The idea of developing a hypothetical nervous system (called as a perceptron) and imitating learning curves from neurological variables is introduced by Rosenblatt in 1958 [72]. In [61], Novikoff proved the perceptron convergence theorem, i.e., if the training data set is linearly seperable, the convergence of the perceptron is guaranteed. The capability of approximating continuous functions of MLP is studied in [20, 29]. The MLP model is well-known for its ability to estimate the relationship with high degree of complexity and nonlinearity. The other advantage of the model is that, the main computational cost of the model comes from the offline training procedure, and the online computational procedure involves simple matrix multiplications with negligible extra cost over the original DG scheme. As shown in Figure 3.1, the MLP model consists of an input layer, an output layer, and several hidden layers, including a normalization layer and fully connected layers.

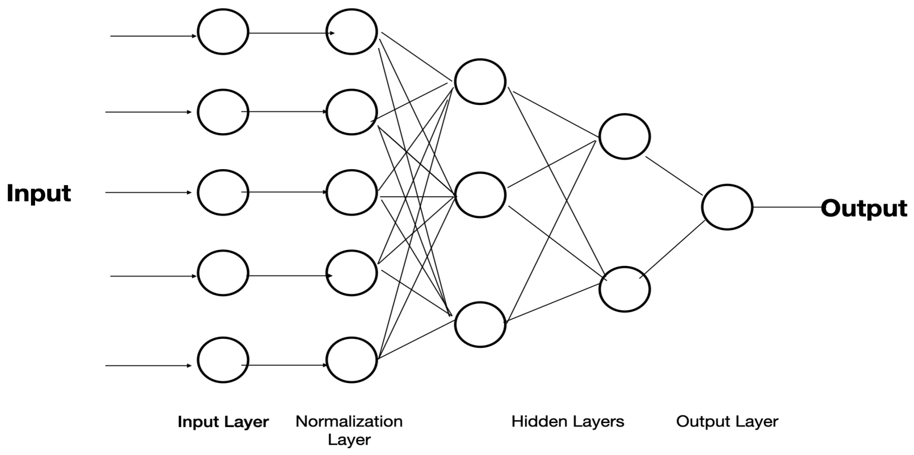


Figure 3.1: An MLP model with an input layer, a normalization layer, hidden layers, and an output layer.

This model can be viewed as an approximation map from the input layer to the output layer,

$$\boldsymbol{F}: \quad \mathbb{R}^{N_1} \mapsto \mathbb{R}^{N_o}, \qquad y = f(x|(\boldsymbol{w}, \boldsymbol{b})), \tag{3.1}$$

where the weights $\boldsymbol{w}$, the bias $\boldsymbol{b}$ and the activation function contained in the hidden layer determine the value of the predicted outputs. The cost function is then applied to measure the error between the network predicted output and the true output value given in the training data set. During the training process, proper training strategy, like the supervised learning [38] we used, is utilized to minimize the error by adjusting the weight and the bias. A well-trained model is capable of precisely predicting the outputs according to the input data, even when the input is not included in the training set.

3.1.1 Construction of the training data

Now we will introduce the design of the MLP-based estimator. In our case, the input data are function values in and near the cell I_i, i.e

$$v = (\bar{u}_{i-1}, \bar{u}_i, \bar{u}_{i+1}, u_h(x^-_{i+1/2}), u_h(x^+_{i-1/2}))^T \in \mathbb{R}^5.$$

The output would be the corresponding TVB limiter constant M_i for the cell I_i.

The input and output training data sets are denoted as $\mathbb{V}_x$ and $\mathbb{V}_y$ respectively, which are generated via the following two ways. Firstly, due to the fact that the DG solutions are piecewise polynomial functions approximating the real PDE solution, the type I data are function values from the L^2 projection of designed functions into suitable piecewise polynomial spaces. Secondly, inspired by the work of Sun et al. [78], we consider the effect of the numerical method on the solution's structure, such as the Gibbs oscillations near discontinuities or the smearing caused by the numerical dissipation. To enable the model to learn the feature of the numerical solutions, the data from numerical solutions of the DG method solving the advection equation $u_t + au_x = 0$ with discontinuous initial conditions are added. The detailed procedure is listed below.

Type I. Data from piecewise polynomial functions.

(a) In the interval $[a, b]$, choose piecewise smooth functions $u(x)$ containing one or more features listed below:

- Containing smooth monotone regions;

- Containing discontinuity points;
- Containing local smooth maxima and/or local smooth minima.

(b) Pick a point x and a mesh size h randomly, such that $a < x-\frac{3}{2}h < x+\frac{3}{2}h < b$, and construct a three-cell stencil containing $I_{i-1} = (x-\frac{3}{2}h, x-\frac{1}{2}h), I_i = (x-\frac{1}{2}h, x+\frac{1}{2}h)$, and $I_{i+1} = (x + \frac{1}{2}h, x + \frac{3}{2}h)$.

(c) Use the standard L^2 projection to project $u(x)$ onto the piecewise polynomial space with different degrees of freedom within each cell, and denote the obtained polynomials in each cell of the three-cell stencil as $u_{i-1}(x)$, $u_i(x)$, and $u_{i+1}(x)$.

(d) Collect the input data, i.e, $v = (\bar{u}_{i-1}, \bar{u}_i, \bar{u}_{i+1}, u_i(x + \frac{1}{2}h), u_i(x - \frac{1}{2}h))^T$.

(e) Determine corresponding output value $y = M \in \mathbb{V}_y$ by the following strategy:

- If the interval $I = (x - \frac{3}{2}h, x + \frac{3}{2}h)$ contains a discontinuity point, the standard minmod limiter should be applied to control spurious oscillations, i.e. $y = M = 0$;
- If the interval I contains a local maximum or a local minimum, we define $M = \frac{2}{3}c \max_{x\in I} |u''(x)|$. Here c is a constant greater than 1, to make M a safer upper bound according to Lemmas 2.2.1 and 2.2.1 for maintaining the original high order accuracy. In our numerical computation, we have taken the value $c = 5$.
- If $u(x)$ in the interval I is smooth and monotone, we choose M big enough so that the minmod limiter is not enacted (i.e. it returns the first argument). In our numerical computation, we have taken the value $M = 1000$ in this case.

Type II. Data from the numerical solution.

(a) We generate the piecewise smooth initial condition u_0 by the following procedure:

- Select the number of discontinuities contained in the initial condition: $1 \leq N_d \leq 6$;
- Randomly select N_d locations for the discontinuities in the domain $[-1, 1]$, and divide the domain into $N_d + 1$ subdomains;
- Within each subdomain, create random Fourier series $a_0 + \sum_{n=1}^{N_f}(a_n \cos(nx) + b_n \sin(nx))$ with different $1 \leq N_f \leq 6$, and i.i.d random variables a_0, a_n, and b_n.

(b) Use different mesh sizes $h = \frac{1}{30}, \frac{1}{60}, \frac{1}{90}, \frac{1}{180}$ to generate uniform meshes with N_x cells.

(c) With a random advection coefficient $a \in [-1, 1]$, apply the Runge-Kutta DG (RKDG) scheme with the degree of freedom k to compute the solution for N_t time steps, where $N_t = 1, 2, 3$ and $k = 1, 2, 3, 4$. The time step size is chosen as $\Delta t = \mathrm{C}\frac{h}{a}$, where the CFL constant is chosen as $\mathrm{C} < \frac{1}{2k+1}$. The obtained numerical solution in cell I_i is denoted as u_i.

(d) Collect the data from the numerical solution, i.e,

$$v = (\bar{u}_{i-1}, \bar{u}_{hi}, \bar{u}_{i+1}, u_i(x_{i+1/2}^-), u_i(x_{i-1/2}^+))^T.$$

(e) The cell is considered to contain a discontinuity or a local smooth extremum if the exact solution $u(x, t) = u_0(x - aN_t\Delta t)$ has discontinuity or a local extremum within the cell or its left or right neighbour cell, and $y = M \in \mathbb{V}_y$

is determined using the same strategy of step 5 in Type I. In general there could exist differences in the locations of discontinuities between the exact and the numerical solutions. In our case, only a few time steps are computed, therefore the difference can be neglected. It enables us to use the location of discontinuities in the exact solution to determine M.

Based on the above guideline, the training data set is constructed, and the details of this data set can be viewed in Table 3.1. For the Type I data, the mesh size h and the degrees of freedom of the projected polynomial space $k \in \{1, 2, 3, 4\}$ are varied.

Table 3.1: Rows 2-6 are the functions used to generate the Type I data. The last three columns are the numbers of cells containing discontinuities, local extrema, and total cell numbers. The second last row is the number of different types of cells in the data generated by the numerical solution of the DG scheme. The last row is the total data number in the data set, which is obtained by adding the data above within each column.

u(x)	domain	varied parameters	discontinuities	local extrema	total
$a\|x\|$	[-0.5,0.5]	$a \in [1, 10]$	1000	0	1000
$u_l I_{x<a} + u_r I_{x>a}$	[-1,1]	$(u_l, u_r) \in [-4, 4]^2$ $a \in [-0.56, 0.56]$	3200	0	3200
$\sin(k\pi x)$	$[0, \frac{k}{4}]$	k=1,. . . ,25	0	720	6480
$\sin(2\pi x)\cos(3\pi x)\sin(4\pi x)$	[0,1]		0	504	1400
$\sin^4(\pi x)$	[0,1]		0	144	1400
Type II data			950	1695	8451
Total			**5150**	**3063**	**21931**

3.1.2 The MLP model

We now briefly introduce the MLP training model. The input is a 5-dimensional vector $\boldsymbol{v}$. Before feeding the data into the hidden layers, we firstly add a normalization layer to normalize the data as follows. Denote the l-th element of the input vector $\boldsymbol{v}$ as v^l, $l = 1\ldots5$. The normalized function value would be $\tilde{\boldsymbol{v}}$, with the l-th element $\tilde{v}^l$ given by

$$\tilde{v}^l = \frac{v^l - \mu}{\sigma}, \tag{3.2}$$

where μ and σ are the mean and the standard deviation of the elements of all $\boldsymbol{v}$ in $\mathbb{V}_x$.

We apply five hidden layers containing 128, 64, 32, and 16 neurons respectively. Within each hidden layer, the weights and bias are randomly initialized using a normal distribution, and Leaky rectified linear unit (Leaky ReLU) is chosen as the activation [58]. The output layer has one neuron, as the output is the value of the limiter TVB constant M. The cost function is given by the mean squared error (MSE) function. The data set is split into two subsets, with 80% data used for training and the remaining 20% data for validation. The model is trained using the Adam optimization [37] with the batch size $S_b = 500$, and with 2000 iterations. Keras API is used for the model training (https://keras.io/).

3.1.3 Implementation of the estimator

After obtaining the well-trained model, it is simple to implement the estimator. The algorithm in the one-dimensional scalar case is described as follows:

(a) Apply the DG method for the spatial discretization, and proceed one Euler forward step in the third order Runge-Kutta time discretization.

(b) Generate $\boldsymbol{v}_i = (\bar{u}_{i-1}, \bar{u}_i, \bar{u}_{i+1}, u_h(x^-_{i+1/2}), u_h(x^+_{i-1/2}))^T$ within each cell I_i.

(c) Feed the data into the estimator, and obtain the corresponding M_i for each cell.

(d) Apply M_i in the minmod-based TVB limiter, and obtain the limited solution.

(e) Repeat Steps 1-4 twice for the next two Runge-Kutta inner stages, and finish the computation of the current time step.

There is no need to change the structure of the original DG code to implement the estimator. Since $\boldsymbol{v}_i$ in Step 2 is also needed in the minmod-based TVB limiter, the only extra work is adding Step 3 to predict the value of M, and in practice it is an one-line addition in the code.

In the two dimensional scalar case, we need to generate in the x direction and in the y direction:

$$\begin{aligned} \boldsymbol{v}_{ij}^x &= (\bar{u}_{i-1,j}, \bar{u}_{i,j}, \bar{u}_{i+1,j}, u_h(x_{i+\frac{1}{2}}^-, y_j), u_h(x_{i-\frac{1}{2}}^+, y_j))^T, \\ \boldsymbol{v}_{ij}^y &= (\bar{u}_{i,j-1}, \bar{u}_{i,j}, \bar{u}_{i,j+1}, u_h(x_i . y_{j+\frac{1}{2}}^-), u_h(x_i, y_{j-\frac{1}{2}}^+))^T, \end{aligned} \tag{3.3}$$

and we feed them into the estimator to obtain the predicted limiter TVB constants M_{ij}^x and M_{ij}^y respectively, and apply them in the limiter. It is clear that there is a low coding cost for the implementation of the estimator in the 2D case as well.

For hyperbolic systems, the estimator and the limiter could be applied component by component, but they are more effective if they are applied in local characteristic fields, which is the procedure that we adopt in our numerical tests. We refer to [15, 17] for more details.

3.2 Numerical tests

In this section, we will perform several standard numerical tests in one-dimension and two-dimension. For the scalar case, we will solve the linear advection equation and the nonlinear Burgers equation, and in the case of systems, the Euler equation of compressible gas dynamics will be approximated. Within each subsection, accuracy tests will be given for the DG scheme with MLP limiter for the degrees of freedom $k = 1, 2, 3$, when the exact solution is smooth. The results will be compared against DG schemes without the limiter. In the case that exact solutions are discontinuous, the performance of the MLP limiter will be presented and compared to that of the TVB limiter with the TVB constant M chosen in an *ad hoc* fashion through trial and error as given in the literature. In general, the MLP limiter has outstanding performance when applied to the DG method of different degrees of freedom. In all accuracy tests, periodic boundary condition is applied, and the simulations runs until $t = 0.3$. The CFL conditions are set to be CFL = 0.3 for $k = 1$, CFL = 0.18 for $k = 2$, and CFL = 0.1 for $k = 3$, according to the linear stability analysis [19].

3.2.1 Linear advection Equation

We firstly consider the one-dimensional linear advection equation with sine wave initial condition:

$$\begin{cases} u_t + u_x = 0, \\ u(x,0) = \sin(x), \quad x \in [0, 2\pi]. \end{cases} \tag{3.1}$$

Table 3.1 demonstrates the error and order of accuracy for the DG scheme with and without the MLP limiter. The MLP limiter method obtains the desired second, third and fourth order accuracy respectively, when applied to the DG scheme with degrees of freedom $k = 1, 2, 3$. The error and order are very close to that of the DG method without the limiter, indicating that the MLP limiter has the correct estimate for the TVB constant M and can maintain the original high order of accuracy.

Table 3.1: Accuracy test for 1D linear advection equation

	k=1 DG MLP-limiter				k=1 DG no limiter			
# cells	L^1 error	order	L^∞ error	order	L^1 error	order	L^∞ error	order
16	4.83 E-03		3.10 E-03		4.39 E-03		2.27 E-03	
32	1.29 E-03	1.90	6.51 E-03	2.25	1.22 E-03	1.84	6.25 E-03	1.86
64	3.15 E-04	2.03	1.60 E-03	2.02	3.41 E-04	1.95	1.60 E-03	1.96
128	7.86 E-05	2.00	4.01 E-04	2.00	7.86 E-05	2.00	4.01 E-04	2.00
256	1.96 E-05	2.00	1.00 E-04	2.00	1.96 E-05	2.00	1.09 E-04	2.00
	k=2 DG MLP-limiter				k=2 DG no limiter			
# cells	L^1 error	order	L^∞ error	order	L^1 error	order	L^∞ error	order
16	1.67 E-04		7.68 E-04		1.78 E-04		7.69 E-04	
32	2.18 E-05	2.94	1.35 E-04	2.50	2.28 E-05	2.96	1.46 E-04	2.39
64	2.47 E-06	3.13	1.55 E-05	3.11	2.46 E-06	3.21	1.51 E-05	3.27
128	3.12 E-07	2.98	1.97 E-06	2.94	3.12 E-07	2.98	1.97 E-06	2.94
256	3.90 E-08	2.97	2.46 E-07	3.00	3.89 E-08	2.97	2.46 E-07	3.00
	k=3 DG MLP-limiter				k=3 DG no limiter			
# cells	L^1 error	order	L^∞ error	order	L^1 error	order	L^∞ error	order
16	4.02 E-06		2.07 E-05		4.02 E-06		2.07 E-05	
32	2.67 E-07	3.91	1.69 E-06	3.62	2.67 E-07	3.91	1.69 E-06	3.62
64	1.34 E-08	4.31	1.09 E-07	3.95	1.34 E-08	4.31	1.09 E-07	3.95
128	8.75 E-10	3.94	6.51 E-09	4.07	8.75 E-10	3.94	6.51 E-09	4.07
256	5.67 E-11	3.95	4.09 E-10	3.99	5.67 E-11	3.95	4.09 E-10	3.99

To check the behavior of the limiter under discontinuous situation, we consider the multi-wave problem, with the initial condition given by

$$u_0(x) = \begin{cases} 10(x-0.2), & 0.2 < x < 0.3, \\ 10(0.4-x), & 0.3 < x < 0.4, \\ 1, & 0.6 < x < 0.8, \\ 100(x-1)(1.2-x), & 1.0 < x < 1.2, \\ 0, & \text{otherwise.} \end{cases} \tag{3.2}$$

The domain is $[0, 1.4]$, and periodic boundary condition is applied. The solution is evaluated at $t = 1.4$ using $N = 100$ cells. In this case, we use the randomly perturbed meshes, which is constructed based on a uniform mesh:

$$x_{i+\frac{1}{2}} \to x_{i+\frac{1}{2}} + \theta h_{i+\frac{1}{2}} \omega_{i+\frac{1}{2}}, \quad \omega_{i+\frac{1}{2}} \in \mathbb{U}([-0.5, 0.5]) \quad i = 1, \ldots, N-1,$$

where we choose $\theta = 0.15$. For all simulations As shown in Figure 3.1, the performance of the TVB limiter with different TVB constants $M = 0, 10, 100, 1000$ and the MLP limiter are compared. The choices of $M = 0, 10$ smear significantly at the two local maxima, and $M = 100, 1000$ fail to control oscillations near the discontinuities. However, the MLP limiter can precisely catch the local extrema without causing oscillation near the discontinuities. Figure 3.2 depicts the temporal history of TVB constant M chosen by MLP model. The MLP model precisely captures the discontinuous points and local extrema, and returns corresponded M.

In the two-dimensional linear case

$$\begin{cases} u_t + u_x + u_y = 0, \\ u(x, y, 0) = \sin(x+y), \quad (x, y) \in [0, 2\pi] \times [0, 2\pi], \end{cases} \tag{3.3}$$

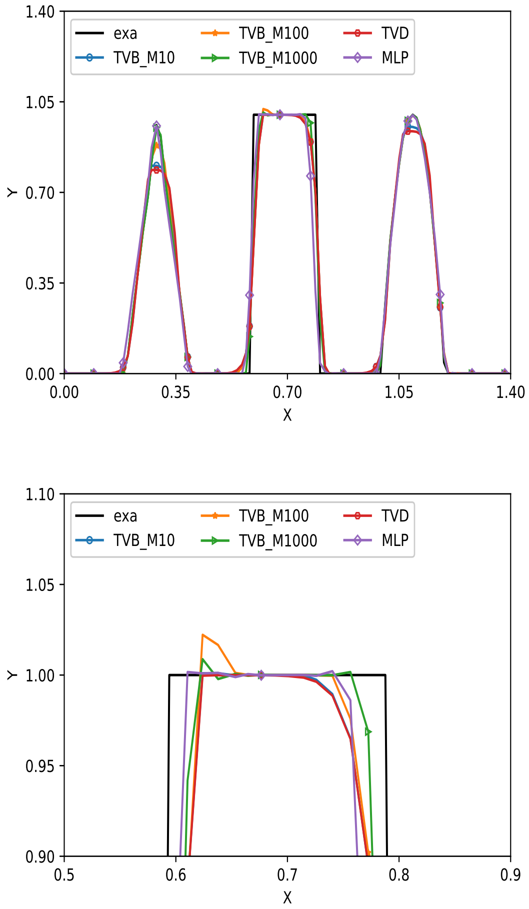


Figure 3.1: Solution for the multi-wave problem using the fourth order DG method, at the final time $t = 1.4$. The right figure is zoomed near $x = 0.7$.

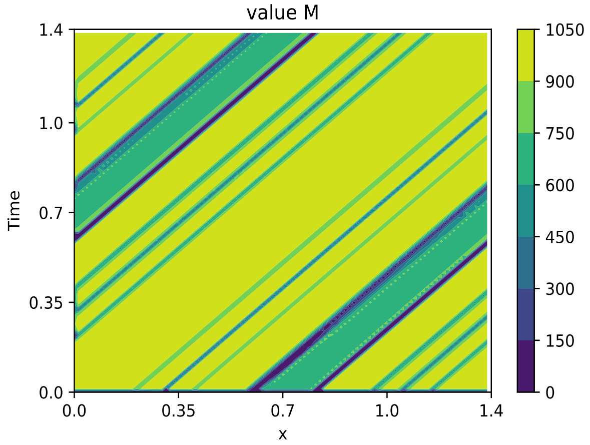


Figure 3.2: Temporal history of constant M chosen by MLP model of the Multiwave problem, k=2

Table 3.2: Accuracy test for 2D linear advection equation

	k=1 DG MLP-limiter				k=1 DG no limiter			
# cells	L^1 error	order	L^∞ error	order	L^1 error	order	L^∞ error	order
16×16	1.03 E-02		9.54 E-02		1.03 E-02		9.54 E-02	
32×32	2.60 E-03	1.98	2.52 E-03	1.91	2.60 E-03	1.98	2.52 E-03	1.91
64×64	6.52 E-04	2.00	6.40 E-03	1.98	6.52 E-04	2.00	6.40 E-03	1.98
128×128	1.62 E-04	2.00	1.60 E-03	2.00	1.62 E-04	2.00	1.60 E-03	2.00
256×256	4.06 E-05	2.00	4.01 E-04	2.00	4.06 E-05	2.00	4.01 E-04	2.00

	k=2 DG MLP-limiter				k=2 DG no limiter			
# cells	L^1 error	order	L^∞ error	order	L^1 error	order	L^∞ error	order
16×16	9.48 E-04		5.84 E-03		9.48 E-04		5.84 E-03	
32×32	9.89 E-05	3.26	1.21 E-03	2.26	9.89 E-05	3.26	1.21 E-03	2.26
64×64	1.14 E-05	3.11	1.46 E-04	3.05	1.14 E-05	3.11	1.46 E-04	3.05
128×128	1.42 E-06	3.00	1.87 E-05	2.97	1.42 E-06	3.00	1.87 E-05	2.97
256×256	1.78 E-07	3.00	2.34 E-06	3.00	1.78 E-07	3.00	2.34 E-06	3.00

	k=3 DG MLP-limiter				k=3 DG no limiter			
# cells	L^1 error	order	L^∞ error	order	L^1 error	order	L^∞ error	order
16×16	5.11 E-05		9.79 E-04		5.11 E-05		9.79 E-04	
32×32	3.20 E-06	3.99	6.09 E-05	4.00	3.20 E-06	3.99	6.09 E-05	4.00
64×64	2.01 E-07	3.99	3.74 E-06	4.02	2.01 E-07	3.99	3.74 E-06	4.02
128×128	1.27 E-08	3.98	2.05 E-07	4.05	1.27 E-08	3.98	2.05 E-07	4.05
256×256	8.24 E-10	3.95	1.27 E-08	4.13	8.24 E-10	3.95	1.27 E-08	4.13

the error and orders of the DG method with the MLP limiter and without the limiter are listed in Table 3.2. The MLP limiter again preserves high order accuracy in this 2D example.

3.2.2 Burgers equation

We consider the nonlinear Burgers equation in $1D$:

$$\begin{cases} u_t + \left(\frac{u^2}{2}\right)_x = 0, \\ u(x,0) = \frac{1}{4} + \sin(x), \quad x \in [0, 2\pi]. \end{cases} \tag{3.4}$$

Before $t = 1$, the solution is smooth, and we can compare the accuracy of the

DG scheme with and without the MLP limiter. From Table 3.3, we observe that applying the limiter does not affect accuracy also in this nonlinear case.

Table 3.3: Accuracy test for 1D Burgers equation

	k=1 DG MLP-limiter				k=1 DG no limiter			
# cells	L^1 error	order	L^∞ error	order	L^1 error	order	L^∞ error	order
16	4.53 E-03		2.67 E-02		4.53 E-03		2.67 E-02	
32	1.05 E-03	2.10	6.41 E-03	2.05	1.05 E-03	2.10	6.41 E-03	2.05
64	2.62 E-04	2.00	1.63 E-03	1.97	2.62 E-04	2.00	1.63 E-03	1.97
128	6.56 E-05	2.00	4.11 E-04	1.99	6.56 E-05	2.00	4.11 E-04	1.99
256	1.63 E-05	2.00	1.03 E-04	1.99	1.63 E-05	2.00	1.03 E-04	1.99
	k=2 DG MLP-limiter				k=2 DG no limiter			
# cells	L^1 error	order	L^∞ error	order	L^1 error	order	L^∞ error	order
16	1.17 E-04		4.96 E-04		1.17 E-04		4.96 E-04	
32	1.45 E-05	3.00	6.28 E-05	2.98	1.45 E-05	3.00	6.28 E-05	2.98
64	1.82 E-06	3.00	7.87 E-06	3.00	1.82 E-06	3.00	7.87 E-06	3.00
128	2.28 E-07	3.00	9.85 E-07	3.00	2.28 E-07	3.00	9.85 E-07	3.00
256	2.84 E-08	3.00	1.23 E-07	3.00	2.84 E-08	3.00	1.23 E-07	3.00
	k=3 DG MLP-limiter				k=3 DG no limiter			
# cells	L^1 error	order	L^∞ error	order	L^1 error	order	L^∞ error	order
16	9.88 E-06		1.51 E-04		9.88 E-06		1.51 E-04	
32	5.84 E-07	4.08	1.07 E-05	3.81	5.84 E-07	4.08	1.07 E-05	3.81
64	3.63 E-08	4.00	7.05 E-07	3.92	3.63 E-08	4.00	7.05 E-07	3.92
128	2.26 E-09	4.00	4.46 E-08	3.94	2.26 E-09	4.00	4.46 E-08	3.94
256	1.41 E-10	4.00	2.95 E-09	3.97	1.41 E-10	4.00	2.95 E-09	3.97

Next we test the compound wave problem, with a discontinuous initial condition:

$$u_0(x) = \begin{cases} l\sin(\pi x), & |x| \geq 1, \\ 3, & -1 < x \leq -0.5, \\ 1, & -0.5 < x \leq 0, \\ 3, & 0 < x \leq 0.5, \\ 2, & 0.5 < x \leq 1, \end{cases} \tag{3.5}$$

The domain is $[-4, 4]$ with perturbed mesh, and the periodic boundary condition is applied. We see the numerical result at $t = 0.4$ in Figure 3.3. The MLP limiter gives good performance on capturing the discontinuities without spurious oscillations.

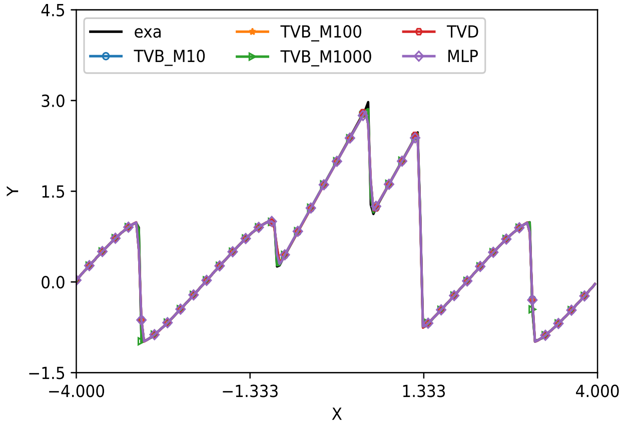


Figure 3.3: Comparison of solutions on a randomly perturbed mesh for the compound wave problem using the fourth order DG method with the TVB limiter with $M = 0, 10, 100, 1000$ and the MLP limiter. Here $T = 0.4$ and cell of number $N = 200$.

The two dimensional Burgers equation is stated as:

$$\begin{cases} u_t + \left(\frac{u^2}{2}\right)_x + \left(\frac{u^2}{2}\right)_y = 0, \\ u(x,y,0) = \frac{1}{4} + \sin(x+y), \quad (x,y) \in [0,2\pi] \times [0,2\pi]. \end{cases} \tag{3.6}$$

The error and order of accuracy of the solution at $t = 0.1$ are in Table 3.4. Similar to the one-dimensional case, the MLP-limiter does not affect the accuracy. When the time reaches $t = 1.2$, there is a shock in the exact solution, and as we can see in Figure 3.4, compared to the DG scheme without limiter, the MLP-limiter effectively controls the oscillation near the shock.

Table 3.4: Accuracy test for 2D Burgers equation

	k=1 DG MLP-limiter				k=1 DG no limiter			
# cells	L^1 error	order	L^∞ error	order	L^1 error	order	L^∞ error	order
16×16	1.32 E-02		8.80 E-02		1.32 E-02		8.80 E-02	
32×32	3.40 E-03	1.95	2.26 E-02	1.96	3.40 E-03	1.95	2.26 E-02	1.96
64×64	8.67 E-04	1.97	5.73 E-03	1.98	8.67 E-04	1.97	5.73 E-03	1.98
128×128	2.18 E-04	1.99	1.43 E-03	1.99	2.18 E-04	1.99	1.43 E-03	1.99
256×256	5.47 E-05	2.00	3.60 E-04	1.99	5.47 E-05	2.00	3.60 E-04	1.99
	k=2 DG MLP-limiter				k=2 DG no limiter			
# cells	L^1 error	order	L^∞ error	order	L^1 error	order	L^∞ error	order
16×16	1.27 E-03		1.43 E-02		1.27 E-03		1.43 E-02	
32×32	1.61 E-04	2.98	1.72 E-03	3.06	1.61 E-04	2.98	1.72 E-03	3.06
64×64	4.48 E-05	1.84	6.24 E-04	1.85	2.04 E-05	3.00	2.17 E-04	3.01
128×128	2.48 E-06	4.17	2.92 E-05	7.73	2.48 E-06	3.00	2.92 E-05	3.01
256×256	3.11 E-07	3.00	3.69 E-06	2.98	3.11 E-07	3.00	3.96 E-06	3.00
	k=3 DG MLP-limiter				k=3 DG no limiter			
# cells	L^1 error	order	L^∞ error	order	L^1 error	order	L^∞ error	order
16×16	9.53 E-05		1.06 E-04		9.53 E-05		1.06 E-04	
32×32	5.93 E-06	4.00	6.74 E-05	3.99	5.93 E-06	4.00	6.74 E-05	3.99
64×64	3.67 E-07	4.01	4.88 E-06	3.79	3.67 E-07	4.01	4.88 E-06	3.79
128×128	2.26 E-08	4.02	3.09 E-07	3.99	2.26 E-08	4.02	3.09 E-07	3.99
256×256	1.47 E-09	3.95	1.43 E-08	3.97	1.47 E-09	3.95	1.43 E-08	3.97

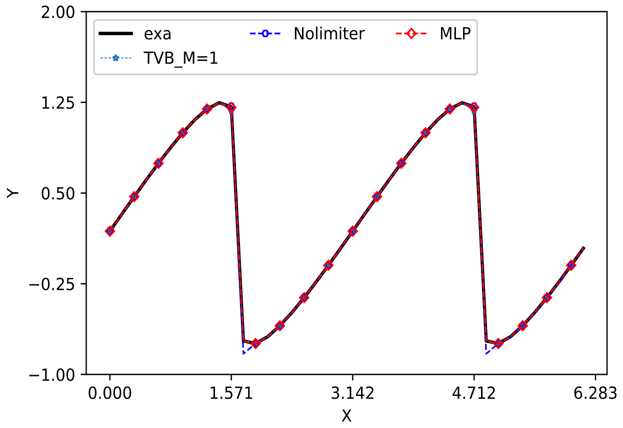


Figure 3.4: Comparison of solutions of the 2D Burgers equation with the initial condition $u_0(x, y) = \frac{1}{4} + \sin(x + y)$ using the fourth order DG method without limiter, with the TVB limiter with $M = 1$, and with the MLP limiter. Final time is $t = 1.2$ and the number of cells corresponds to $N_x = N_y = 40$.

3.2.3 Euler equation

In this subsection we apply the MLP limiter to solve nonlinear systems. We firstly consider the compressible Euler equation in one dimension:

$$\frac{\partial}{\partial t}\begin{pmatrix} \rho \\ \rho\mu \\ E \end{pmatrix} + \frac{\partial}{\partial x}\begin{pmatrix} \rho\mu \\ \rho\mu^2 + p \\ \mu(E+p) \end{pmatrix} = 0, \ \ 0 < x < 2\pi, \tag{3.7}$$

where ρ, μ, and p denote the density, velocity and pressure of the fluids, respectively. The total energy $E = \frac{p}{\gamma-1} + \frac{1}{2}\rho\mu^2$, with $\gamma = 1.4$ for air. For the system case, we choose to use the limiter in the local characteristic fields. That is, we firstly project the conserved variable $\boldsymbol{U} = (\rho, \rho\mu, E)^T$ into the local characteristic fields, and then apply the TVB or the MLP limiter in each characteristic field. Finally we project the limited numerical solution back to the conserved variable space. More details can be found in [15]. We will compare the performance of the MLP-limiter with the TVB-limiter with *ad hoc* choices of the TVB constant M through trial and error as adopted in the literature. In all the test cases, we present the results for the density ρ as representations.

Example 4.3.1: Artificial accuracy test.

We firstly consider the accuracy test in [24]. We set the initial condition as:

$$\rho(x,0) = \frac{1 + 0.2\sin(x)}{2\sqrt{3}}, \quad \mu(x,0) = \sqrt{\gamma}\rho(x,0), \quad p(x,0) = \rho(x,0)^{\gamma}. \tag{3.8}$$

The computational domain is set to be $[0, 2\pi]$, and periodic boundary condition is imposed. We take $\gamma = 3$, which allows us to verify that $2\sqrt{3}\rho(x,t)$ is the exact solution of the Burgers equation:

$$u_t + \left(\frac{u^2}{2}\right)_x = 0, \qquad u(x,0) = 1 + 0.2\sin(x), \tag{3.9}$$

and

$$\mu(x,t) = \sqrt{\gamma}\rho(x,t), \qquad p(x,t) = \rho(x,t)^{\gamma}. \tag{3.10}$$

At $t = 0.3$, the solution is smooth, and the error and order of accuracy of density are listed in Table 3.5. It is clear that the MLP limiter does not affect the accuracy in this 1D nonlinear system example.

Table 3.5: Accuracy test for 1D Euler equation

	k=1 DG MLP-limiter				k=1 DG no limiter			
# cells	L^1 error	order	L^∞ error	order	L^1 error	order	L^∞ error	order
16	4.96 E-03		8.86 E-03		4.96 E-03		8.86 E-03	
32	1.10 E-03	2.16	1.31 E-03	2.75	1.10 E-03	2.16	1.31 E-03	2.75
64	2.76 E-04	2.00	3.28 E-04	1.97	2.76 E-04	2.00	3.28 E-04	1.97
128	6.90 E-05	2.00	8.22 E-05	1.99	6.90 E-05	2.00	8.22 E-05	1.99
256	1.72 E-05	2.00	2.06 E-05	1.99	1.72 E-05	2.00	2.06 E-05	1.99
	k=2 DG MLP-limiter				k=2 DG no limiter			
# cells	L^1 error	order	L^∞ error	order	L^1 error	order	L^∞ error	order
16	1.93 E-04		3.76 E-04		1.93 E-04		3.76 E-04	
32	2.49 E-05	2.96	6.13 E-05	2.61	2.49 E-05	2.96	6.13 E-05	2.61
64	3.07 E-06	3.02	7.99 E-06	2.94	3.07 E-06	3.02	7.99 E-06	2.94
128	3.28 E-07	3.00	1.02 E-06	2.97	3.28 E-07	3.00	1.02 E-06	2.97
256	4.77 E-08	3.00	1.27 E-07	3.00	4.77 E-08	3.00	1.27 E-07	3.00
	k=3 DG MLP-limiter				k=3 DG no limiter			
# cells	L^1 error	order	L^∞ error	order	L^1 error	order	L^∞ error	order
16	7.27 E-06		1.21 E-05		7.27 E-06		1.21 E-05	
32	4.91 E-07	3.88	5.00 E-07	4.49	4.91 E-07	3.88	5.00 E-07	4.49
64	3.04 E-08	4.01	3.12 E-08	4.00	3.04 E-08	4.01	3.12 E-08	4.00
128	1.88 E-09	4.00	2.10 E-09	3.88	1.88 E-09	4.00	2.10 E-09	3.88
256	1.18 E-10	3.99	1.30 E-10	4.01	1.18 E-10	3.99	1.30 E-10	4.01

Example 4.3.2: The Sod problem.

This problem is a classic Riemann problem test, whose initial condition is

$$(\rho, \mu, p) = \begin{cases} (1, 0, 1), & x \leq 0, \\ (0.125, 0, 0.1), & x > 0. \end{cases} \tag{3.11}$$

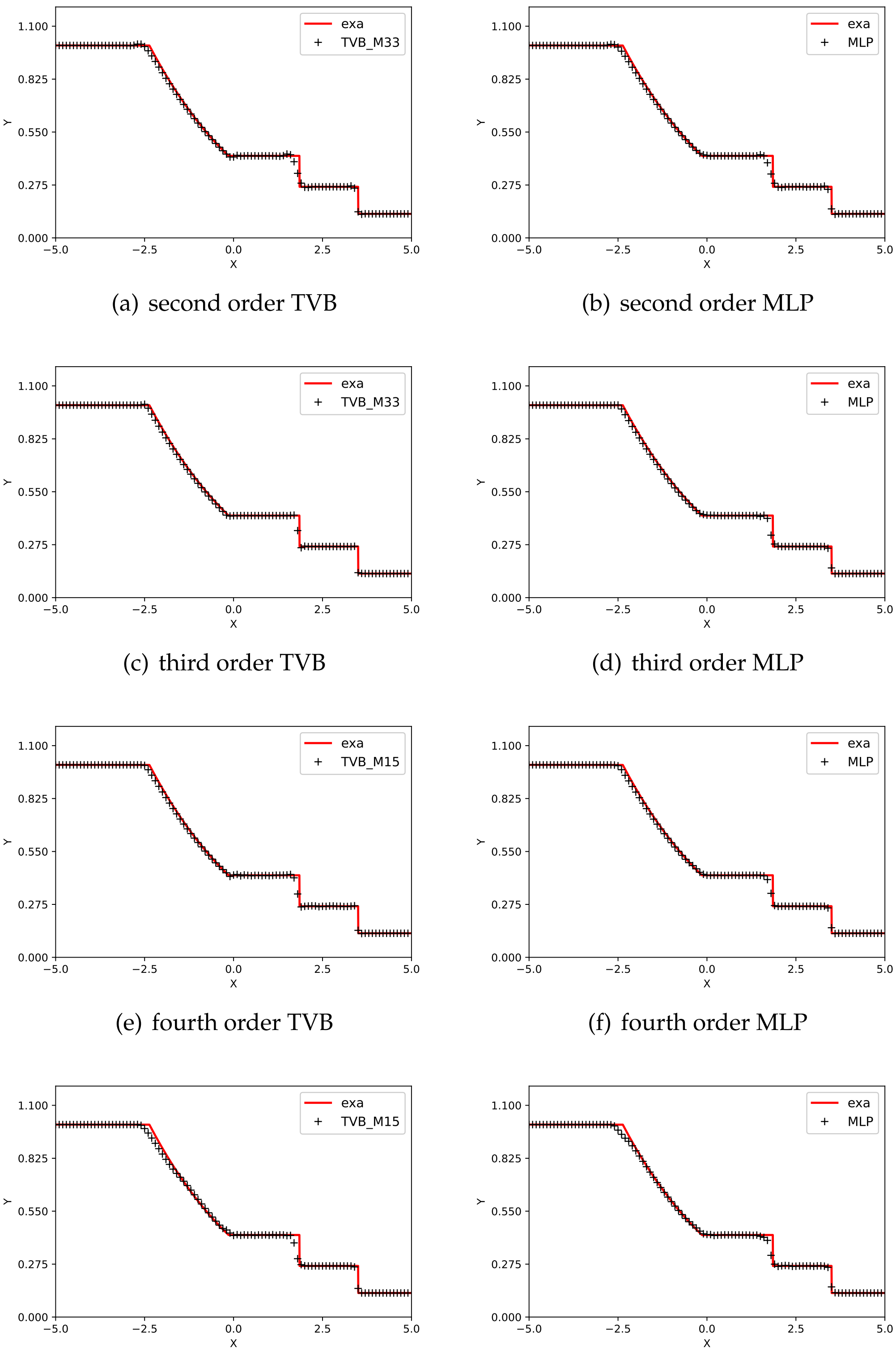
The domain is $x \in [-5, 5]$, and the simulation runs until $t = 2.0$ with the mesh size $N = 100$. We test the DG scheme with different orders of accuracy. If the TVB constant $M = 33$ or larger, the TVB limiter simulation fails with fourth or higher order DG schemes, due to the appearance of negative density. With $M = 33$, the TVB limiter gives good performance for the DG scheme with second and third order. On the other hand, while the solutions of TVB limiter with $M = 15$ smear a lot at discontinuities in lower order cases, it gives satisfying non-oscillatory result with fourth and fifth order DG schemes. Meanwhile, the MLP limiter gives good simulation in all cases, with results comparable to the $M = 33$ case in second and third order cases, and $M = 15$ in fourth and fifth order cases. The details are shown in Figure 3.5.

Example 4.3.3: The Lax problem.

Another famous Riemann problem test is the Lax problem, with the initial condition

$$(\rho, \mu, p) = \begin{cases} (0.445, 0.698, 0, 3.528), & x \leq 0, \\ (0.5, 0, 0, 0.571), & x > 0. \end{cases} \tag{3.12}$$

The domain is $x \in [-5, 5]$ and the number of cells is $N = 100$. We compute the solution until $t = 1.3$. In this case, we use $M = 33$ [15] which gives the best (sharpest) performance at discontinuities (especially at the contact discontinuity) for the third order DG scheme. As shown in Figure 3.6, although the solution of the TVB limiter with $M = 70$ has huge oscillations at the discontinuity in lower order



(a) second order TVB

(b) second order MLP

(c) third order TVB

(d) third order MLP

(e) fourth order TVB

(f) fourth order MLP

(g) fifth order TVB

(h) fifth order MLP

Figure 3.5: Comparison of solutions for the Sod problem using DG method of degree of freedom $k = 1, 2, 3, 4$ with the TVB limiter (left) and the MLP (right) limiter. Final time $t = 2.0$ and the number of cells $N = 100$.

cases, it gives best performance for the fifth order DG scheme. On the other hand, the MLP limiter works well for DG schemes with different orders of accuracy. The performance of the MLP limiter is as good as that of $M = 33$ TVB limiter for the third order scheme, and of $M = 70$ TVB limiter for the fifth order scheme. For the second order scheme, the MLP limiter describes the edge of the discontinuity better than that of the TVB limiters.

Example 4.3.4: The blast wave problem.

We now consider the interaction of two blast waves, with the initial condition:

$$(\rho, \mu, p) = \begin{cases} (1, 0, 1000), & 0 < x < 0.1, \\ (1, 0, 0.01), & 0.1 < x < 0.9, \\ (1, 0, 100), & 0.9 < x < 1. \end{cases} \tag{3.13}$$

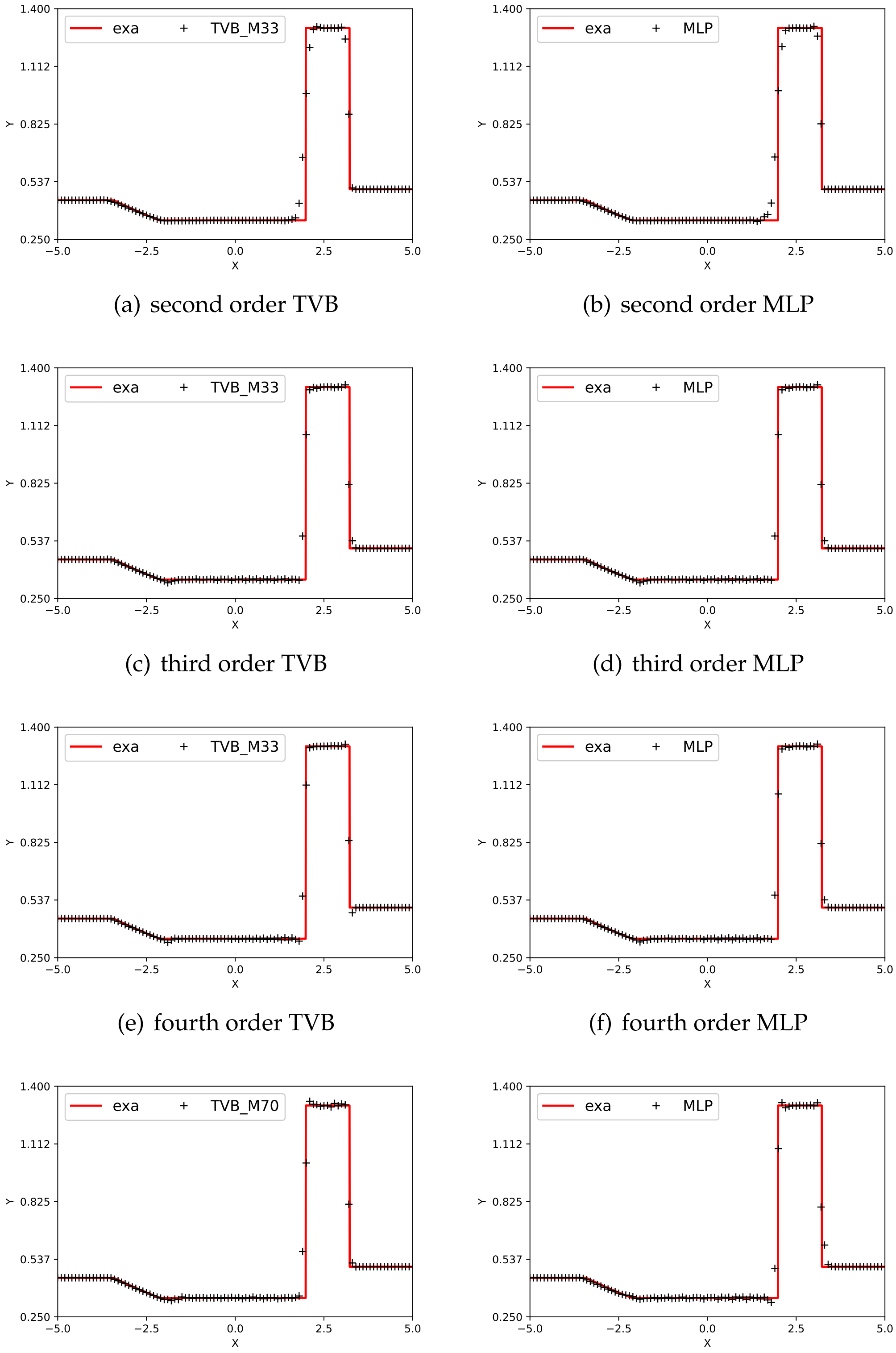
The domain is $x \in [0, 1]$ and reflective boundary condition is applied. We present the numerical density of the TVB limiter DG method with the TVB constant $M = 33$ [15] and the MLP limiter DG method at the time $t = 0.038$ in Figure 3.7. The solutions of the two methods are comparable.

Example 4.3.5: The Shu-Osher problem.

This example is introduced in [77], and it describes shock-turbulence interactions. It's initial condition is given by:

$$(\rho, \mu, p) = \begin{cases} (3.857143, 2.629369, 10.33333), & -5 \leq x < -4, \\ (1 + 0.2\sin(5x), 0, 1), & -4 \leq x \leq 5, \end{cases} \tag{3.14}$$

The domain is $x \in [-5, 5]$ and reflective boundary condition is applied. We present



(a) second order TVB

(b) second order MLP

(c) third order TVB

(d) third order MLP

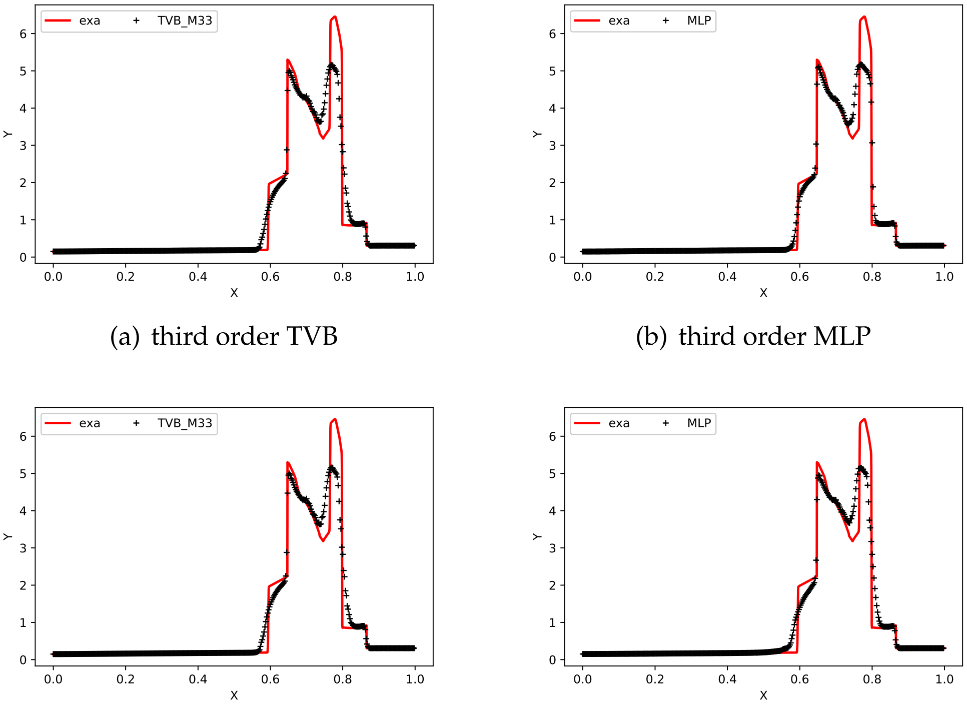
(e) fourth order TVB

(f) fourth order MLP

(g) fifth order TVB

(h) fifth order MLP

Figure 3.6: Comparison of solutions for the Lax problem using DG method of degree of freedom $k = 1, 2, 3, 4$ with the TVB limiter (left) and the MLP (right) limiter. Final time $t = 1.3$ and the number of cells $N = 100$.



(a) third order TVB

(b) third order MLP

(c) fifth order TVB

(d) fifth order MLP

Figure 3.7: Solution of the blast wave problem using the third order and fifth order DG schemes with the $M = 33$ TVB limiter (left), and the MLP limiter (right). Final time $T = 0.038$ and the number of cells $N = 400$.

the numerical density of the TVB limiter DG method with the TVB and the MLP limiter DG method at the time $t = 0.038$ in Figure 3.8. To achieve the best performance, the TVB constant is chosen as $M = 300$ [15] for $k = 1, 2, 3$ and $M = 550$ for $k = 4$. The overall performance are increased when higher order method are applied. The MLP model shows the performance similar to the TVB model at the oscillatory area.

Now we consider the two-dimensional Euler equation:

$$\frac{\partial}{\partial t}\begin{pmatrix} \rho \\ \rho\mu \\ \rho\nu \\ E \end{pmatrix} + \frac{\partial}{\partial x}\begin{pmatrix} \rho\mu \\ \rho\mu^2 + p \\ \rho\mu\nu \\ \mu(E+p) \end{pmatrix} + \frac{\partial}{\partial y}\begin{pmatrix} \rho\nu \\ \rho\mu\nu \\ \rho\nu^2 + p \\ \nu(E+p) \end{pmatrix} = 0, \tag{3.15}$$

where ρ is the density, μ and ν are the velocities in the x and y directions, respectively, and p is the fluid pressure. The total energy $E = \frac{p}{\gamma-1} + \frac{1}{2}\rho(\mu^2 + \nu^2)$, with $\gamma = 1.4$ for air.

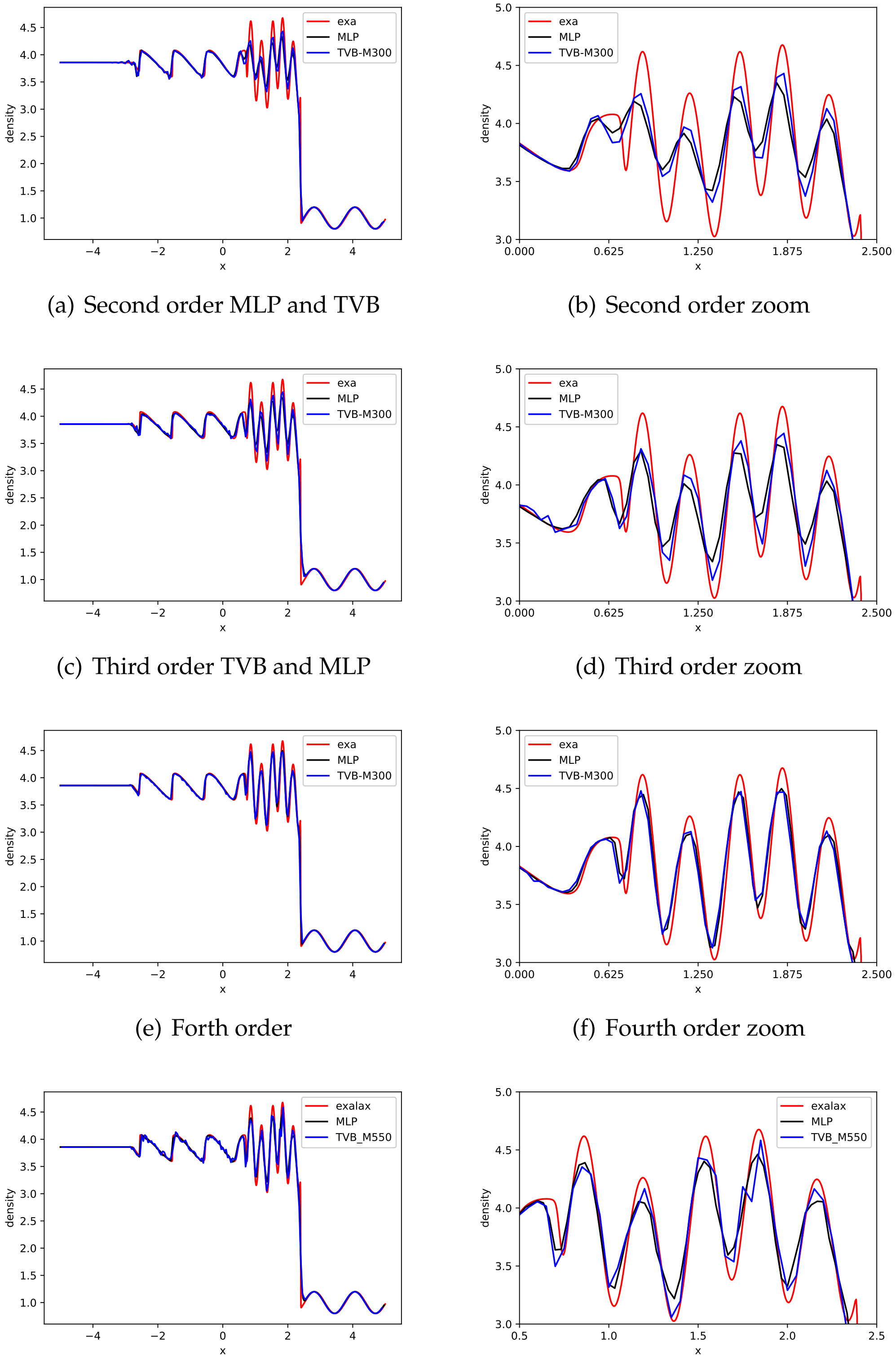
Example 4.3.5: Artificial accuracy test for the 2D Euler equation.

We conduct an accuracy test for the 2D Euler equation. The initial condition is:

$$\rho(x, y, 0) = \frac{1 + 0.2\sin(\frac{x+y}{2})}{\sqrt{6}}, \mu(x, y, 0) = \nu(x, y, 0) = \sqrt{\frac{\gamma}{2}}\rho(x, y, 0), p(x, y, 0) = \rho(x, y, 0)^{\gamma}. \tag{3.16}$$

The computational domain is $[0, 4\pi] \times [0, 4\pi]$. We set $\gamma = 3$, and it could be easily verified that $\sqrt{6}\rho(x, y, t)$ is the exact solution of the Burgers equation:

$$u_t + \left(\frac{u^2}{2}\right)_x + \left(\frac{u^2}{2}\right)_y = 0, \qquad u(x, y, 0) = 1 + 0.2\sin(\frac{x+y}{2}), \tag{3.17}$$



(a) Second order MLP and TVB

(b) Second order zoom

(c) Third order TVB and MLP

(d) Third order zoom

(e) Forth order

(f) Fourth order zoom

(g) Fifth order

(h) Fifth order zoom

Figure 3.8: Numerical Solution of the Shu-Osher problem (Left). Zoom close to the fluctuations(Right). Final time $T = 1.8$ and the number of cells $N = 200$.

and μ, ν and p satisfy:

$$\mu(x, y, t) = \mu(x, y, t) = \sqrt{\frac{\gamma}{2}}\rho(x, y, t), \qquad p(x, y, t) = \rho(x, y, t)^{\gamma}. \tag{3.18}$$

At $t = 0.3$, the solution is smooth. The error and order of accuracy of density are shown in Table 3.6. It can be observed that the MLP limiter does not affect the high order accuracy of the scheme for this 2D nonlinear system test case. In Table 3.7, the cpu time of simulations on $100x100$ mesh are analyzed and reported. The simulations have been run on Jupyter Notebook using a 2 GHz Quad-Core Intel Core i5 processor. The execution time of a single timestep(Tsp) increases when higher order scheme is used. It can be observed that the gap between the cost of the TVB and the MLP limter narrows when k increases. When $k = 3, 4$ the additional cost of applying MLP model in TVB DG scheme is negligible.

Example 4.3.6: The double Mach reflection problem.

This problem was introduced by Woodward and Colella [83]. We use the same setup as in [83], which describes a Mach 10 shock moving right into the undisturbed air, making a 60° angle with a reflecting wall. The density and pressure of the undisturbed air are 1.4 and 1 respectively. The computational domain is $[0, 4] \times [0, 1]$. We use the exact flow values of the Mach 10 shock at each time step as the top boundary condition. For the bottom boundary, we apply the post-shock condition for $x \in [0, \frac{1}{6}]$, and reflecting wall condition for $x \in [\frac{1}{6}, 4]$. The numerical simulation is generated up to $t = 0.2$. The simulations on uniformed meshes with 480×120 and 960×240 cells are shown in Figures 3.9 and 3.11, with the zoomed version near the Mach stem shown in Figures 3.10 and 3.12. For the TVB limiter, the TVB constant is chosen as $M = 50$ for the second and third order DG

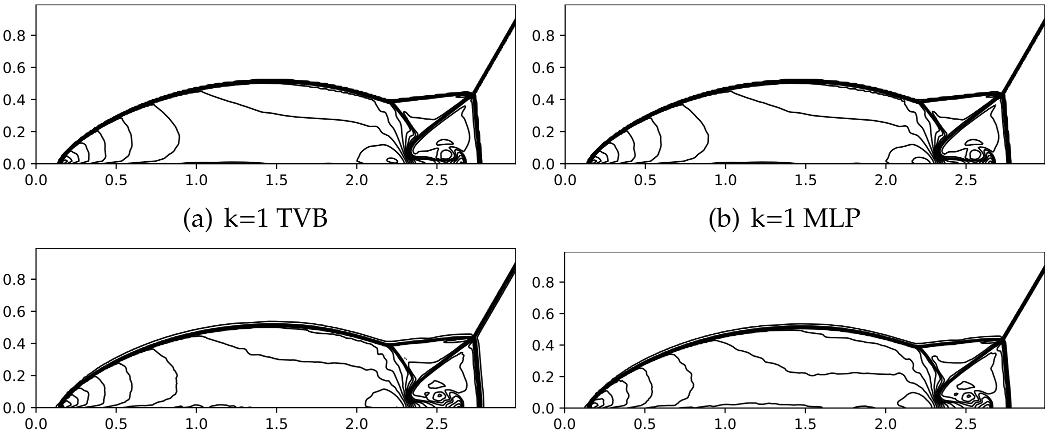
Table 3.6: 2D Euler equation accuracy test.

	k=1 DG MLP-limiter				k=1 DG no limiter			
# cells	L^1 error	order	L^∞ error	order	L^1 error	order	L^∞ error	order
16×16	1.00 E-3		7.24 E-02		1.00 E-3		7.24 E-02	
32×32	2.52 E-04	1.99	1.94 E-03	1.90	2.52 E-04	1.99	1.94 E-03	1.90
64×64	6.37 E-05	1.99	1.59 E-04	1.96	6.37 E-05	1.99	1.59 E-04	1.96
128×128	1.59 E-05	2.00	1.25 E-04	1.99	1.59 E-05	2.00	1.25 E-04	1.99
256×256	3.98 E-06	2.00	3.14 E-05	1.99	3.98 E-06	2.00	3.14 E-05	1.99
	k=2 DG MLP-limiter				k=2 DG no limiter			
# cells	L^1 error	order	L^∞ error	order	L^1 error	order	L^∞ error	order
16×16	1.17 E-04		4.96 E-04		1.27 E-03		1.43 E-02	
32×32	1.45 E-05	3.00	1.61 E-04	2.98	1.72 E-03	2.98	6.28 E-05	3.06
64×64	1.82 E-06	3.00	2.04 E-05	3.00	2.17 E-04	3.01	7.87 E-06	2.98
128×128	2.28 E-07	3.00	2.48 E-06	3.00	2.92 E-05	3.01	9.85 E-07	2.90
256×256	2.84 E-08	3.00	3.11 E-07	3.00	3.96 E-06	3.00	1.23 E-07	3.00
	k=3 DG MLP-limiter				k=3 DG no limiter			
# cells	L^1 error	order	L^∞ error	order	L^1 error	order	L^∞ error	order
16×16	9.53 E-05		1.06 E-04		9.53 E-05		1.06 E-04	
32×32	5.93 E-06	4.00	6.74 E-05	3.99	5.93 E-06	4.00	6.74 E-05	3.99
64×64	3.67 E-07	4.01	4.88 E-06	3.79	3.67 E-07	4.01	4.88 E-06	3.79
128×128	2.26 E-08	4.02	3.09 E-07	3.99	2.26 E-08	4.02	3.09 E-07	3.99
256×256	1.47 E-09	3.95	1.43 E-08	3.97	1.47 E-09	3.95	1.43 E-08	3.97

Table 3.7: Computational times, number of timesteps and execution time of a single timestep (TpS) for the 2D Euler problem. The total time and the time per timestep are expressed in seconds

	k=1			k=2			k=3			k=4		
Limiters	time	Steps	Tps	time	Steps	Tps	time	Steps	Tps	time	Steps	Tps
TVB	26.99	29	0.93	61.76	47	1.31	132.12	66	2.00	308.14	85	3.62
MLP	39.73	29	1.37	74.91	47	1.59	137.75	66	2.08	317.06	85	3.72

schemes [17]. Compared to the traditional TVB limiter with empirically chosen M through trial and error, the MLP limiter provides equally satisfying results.



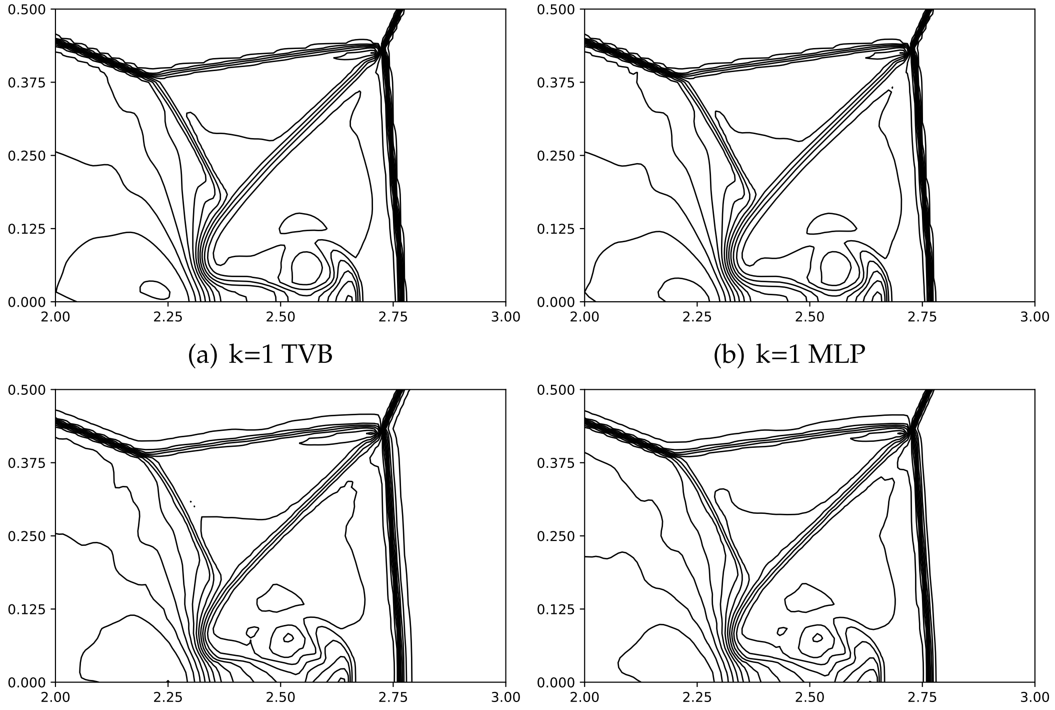
(a) k=1 TVB

(b) k=1 MLP

(c) k=2 TVB

(d) k=2 MLP

Figure 3.9: Double Mach reflection problem. DG method with $k = 1, 2$. Left: results with the TVB limiter. Right: results with the MLP limiter. Density ρ. 30 equally spaced contour lines from $\rho = 1.5$ to $\rho = 22.7$. Mesh grid: 480 × 120.



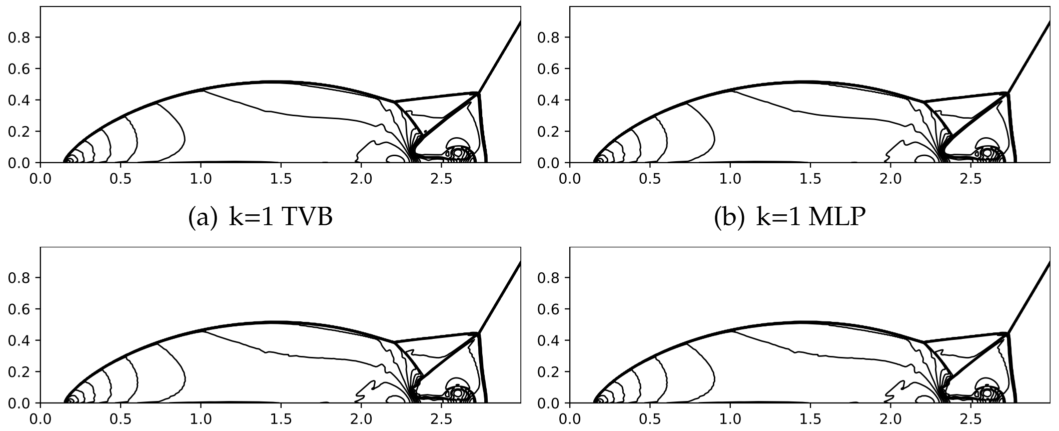
(a) k=1 TVB

(b) k=1 MLP

(c) k=2 TVB

(d) k=2 MLP

Figure 3.10: Double Mach reflection problem. DG method with $k = 1, 2$. Blown-up region around the double Mach stem. Left: results with the TVB limiter. Right: results with the MLP limiter. Density ρ. 30 equally spaced contour lines from $\rho = 1.5$ to $\rho = 22.7$. Mesh grid: 480 × 120.



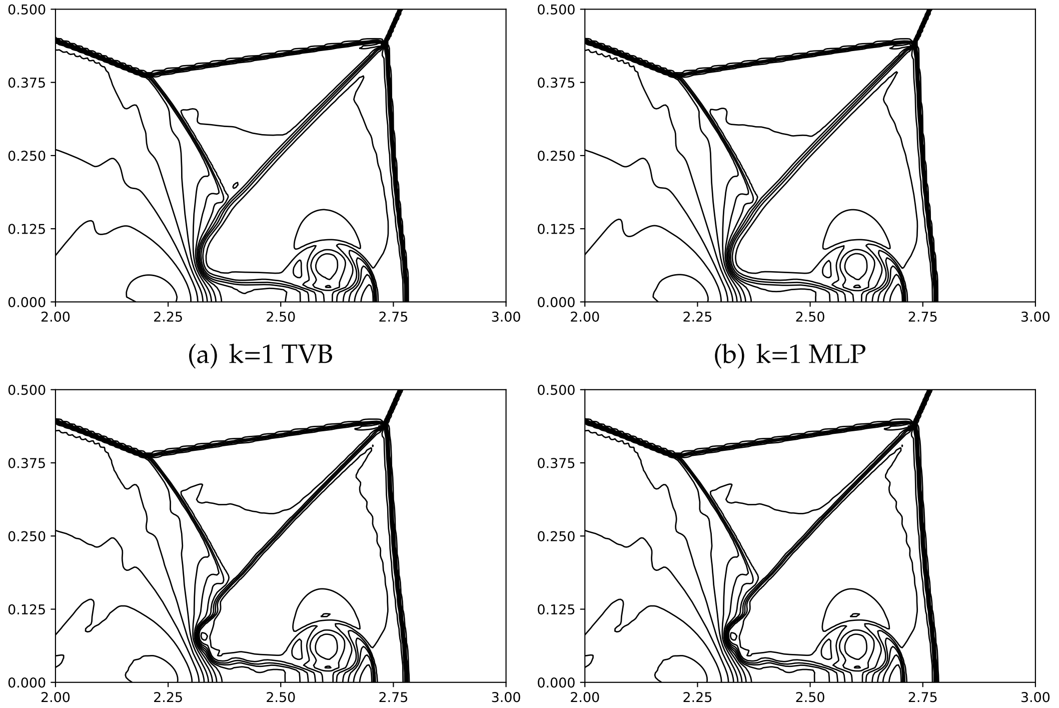
(a) k=1 TVB

(b) k=1 MLP

(c) k=2 TVB

(d) k=2 MLP

Figure 3.11: Double Mach reflection problem. DG method with $k = 1, 2, 3$. Left: results with the TVB limiter. Right: results with the MLP limiter. Mesh grid: 960×240.



(a) k=1 TVB

(b) k=1 MLP

(c) k=2 TVB

(d) k=2 MLP

Figure 3.12: Double Mach reflection problem. DG method with $k = 1, 2, 3$. Blown-up region around the double Mach stem. Left: results with the TVB limiter. Right: results with the MLP limiter. Density ρ. 30 equally spaced contour lines from $\rho = 1.5$ to $\rho = 22.7$. Mesh grid: 960×240.

Chapter Four

The semi-discrete LDG methods for the carpet cloak model

4.1 Introduction

Since Leonhardt [46] and Pendry *et al.* [65] firstly demonstrated the idea of invisibility cloak design with metamaterials in 2006, much study has been done in both theoretical and numerical analysis. There are a plenty of excellent works on the mathematical analysis of the cloaking phenomenon [1, 39, 27, 28], and on the numerical simulations of the cloaking models with the finite difference (FD) methods [30, 34, 56], the finite element (FE) methods [5, 44, 50, 60], and the spectral methods [86, 87]. For more details, readers can consult the review papers [2, 7, 33], and the monographs [21, 32, 49, 59] as references. In 2014, Li *et al.* proposed the mathematical analysis for the time-domain carpet cloak model [50]. In [52], a revised finite difference method for the carpet cloak model was developed, and the corresponding stability analysis was performed with the time step constraint $\tau = O(h^2)$, where τ and h are the time step size and spatial mesh size respectively. In order to relax the time step constraint to $\tau = O(h)$, the usual requirement for the FD or the FE methods to solve the time-dependent Maxwell equations, a new energy was introduced in [54]; moreover, the finite element method coupled with two time discretization methods to solve the carpet cloak model was developed therein.

The discontinuous Galerkin (DG) method was initially proposed by Reed and Hill [71] to solve the neutron transport problem. Later, Cockburn and Shu introduced the Runge-Kutta DG (RKDG) methods for solving the linear and nonlinear hyperbolic partial differential equations (PDEs) [16, 17], and the local DG (LDG) methods for solving the time-dependent convection-diffusion systems [18], which stimulated the rapid development and application of the DG methods [19, 85]. The DG method shares the advantages of the continuous finite element methods, including flexible *h*-*p* adaptivity and easy handling of the complicated geometry. Additionally, it has unique nice features, such as it has the local mass matrix because of the discontinuous basis, it allows easy handling of hanging nodes and adaptivity, and it has high parallel efficiency. Attracted by the good properties of the DG methods, mathematicians have developed the DG methods to solve the Maxwell equations in free space [6, 10, 13, 23, 81], and in dispersive media [43, 57, 81]. For the Maxwell equations in the metamaterials, there are published works on the DG methods to solve the Drude models [47, 48, 53, 55, 74], the Maxwell equations in nonlinear optical media [4], and the wave propagation in media with dielectrics and metamaterials [11].

In [52], the DG method was carried out to solve the carpet cloak model, and it gave a good performance in numerical simulations. However, the stability analysis and the error estimate of the method were left to be done. In this section, we will propose the semi-discrete DG method for the model, and prove its stability. Next, we provide a sub-optimal error estimate in the L^2 norm on unstructured meshes, and an optimal error estimate on tensor-product rectangular meshes.

4.2 The governing equations of the carpet cloak model

The governing equations for modeling the wave propagation in the carpet cloak are derived in [51] and given as follows (cf. [51, (2.3)-(2.5)]):

$$\partial_t D_x = \frac{\partial H}{\partial y}, \tag{4.1}$$

$$\partial_t D_y = -\frac{\partial H}{\partial x}, \tag{4.2}$$

$$\varepsilon_0 \lambda_2 \left(M_A^{-1} \partial_{t^2} \boldsymbol{E} + \omega_p^2 M_A^{-1} \boldsymbol{E} \right) = \partial_{t^2} \boldsymbol{D} + M_C \boldsymbol{D}, \tag{4.3}$$

$$\mu_0 \mu \partial_t H = -\nabla \times \boldsymbol{E}, \tag{4.4}$$

where the 2D electric displacement is denoted as $\boldsymbol{D} := (D_x, D_y)'$, the 2D electric field as $\boldsymbol{E} := (E_x, E_y)'$, and the magnetic field as H. Furthermore, $\partial_{t^k} u$ denotes the k-th derivative $\partial^k u / \partial t^k$ of a function u. For any $k \geq 1$, we adopt the 2D vector and scalar curl operators:

$$\nabla \times H = (\frac{\partial H}{\partial y}, -\frac{\partial H}{\partial x})', \quad \nabla \times \boldsymbol{E} = \frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y}, \;\; \forall\, \boldsymbol{E} = (E_x, E_y)'.$$

We note that (4.3) is revised from [51, (2.4)] by left-multiplying both sides with M_A^{-1} and by denoting the matrix M_C as $M_A^{-1} M_B$. Here M_A^{-1} denotes the inverse of the matrix M_A, which is proved to be symmetric positive definite [51, Lemma 2.1]. As shown in Fig. 4.1, the governing equations (4.1)-(4.4) hold true in the cloaking region formed by the quadrilateral with vertices $(-d, 0), (0, H_1), (d, 0)$ and $(0, H_2)$, where d, H_1 and H_2 are positive constants and $H_2 > H_1 > 0$. The cloaked region, where the hiding objects can be placed, is formed by the triangle with vertices $(0, H_1), (-d, 0)$ and $(d, 0)$.

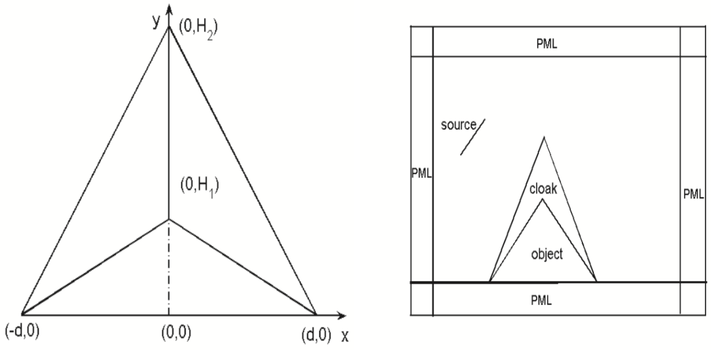


Figure 4.1: Left: The structure of the carpet cloak. Right: The setup of the carpet cloak simulation.

In order to make those objects inside the cloaked region invisible, the permittivity and permeability in the cloaking region need to be specially designed and are given by [51]:

$$\varepsilon = \begin{bmatrix} a & b \\ b & c \end{bmatrix} := \begin{bmatrix} \frac{H_2}{H_2-H_1} & -\frac{H_1H_2}{(H_2-H_1)d}\mathrm{sgn}(x) \\ -\frac{H_1H_2}{(H_2-H_1)d}\mathrm{sgn}(x) & \frac{H_2-H_1}{H_2} + \frac{H_2}{H_2-H_1}(\frac{H_1}{d})^2 \end{bmatrix}, \quad \mu = a,$$

where sgn(x) denotes the sign function. Furthermore, in (4.1)-(4.4), ε_0 and μ_0 denote the permittivity and permeability in free space, respectively; the matrices M_A and M_B are given as [51, page 1138]:

$$M_A = \begin{pmatrix} p_1^2\lambda_2 + p_2^2 & p_2p_4 + p_1p_3\lambda_2 \\ p_2p_4 + p_1p_3\lambda_2 & p_3^2\lambda_2 + p_4^2 \end{pmatrix}, \quad M_B = \begin{pmatrix} p_2^2 & p_2p_4 \\ p_2p_4 & p_4^2 \end{pmatrix}\omega_p^2,$$

where the positive constant ω_p is the plasma frequency resulting from the Drude dispersion model [51, page 1138], elements $p_i, i = 1, 2, 3, 4$, are

$$p_1 = \sqrt{\frac{\lambda_2 - a}{\lambda_2 - \lambda_1}}, \qquad p_2 = -\sqrt{\frac{a - \lambda_1}{\lambda_2 - \lambda_1}} \cdot \mathrm{sgn}(x),$$

$$p_3 = \sqrt{\frac{\lambda_2 - c}{\lambda_2 - \lambda_1}} \cdot \mathrm{sgn}(x), \qquad p_4 = \sqrt{\frac{c - \lambda_1}{\lambda_2 - \lambda_1}},$$

and λ_1 and λ_2 are the eigenvalues of the matrix ε given as:

$$\lambda_1 = \frac{a + c - \sqrt{(a-c)^2 + 4b^2}}{2}, \quad \lambda_2 = \frac{a + c + \sqrt{(a-c)^2 + 4b^2}}{2}.$$

To complete the carpet cloak model (4.1)-(4.4), we assume that (4.1)-(4.4) satisfy the initial conditions

$$\boldsymbol{D}(\boldsymbol{x}, 0) = \boldsymbol{D}_0(\boldsymbol{x}), \;\; \boldsymbol{E}(\boldsymbol{x}, 0) = \boldsymbol{E}_0(\boldsymbol{x}), \;\; H(\boldsymbol{x}, 0) = H_0(\boldsymbol{x})$$

$$\partial_t \boldsymbol{D}(\boldsymbol{x}, 0) = \boldsymbol{D}_1(\boldsymbol{x}), \;\; \partial_t \boldsymbol{E}(\boldsymbol{x}, 0) = \boldsymbol{E}_1(\boldsymbol{x}), \quad \forall\, \boldsymbol{x} \in \Omega,$$

and the perfect conducting boundary condition (PEC):

$$\boldsymbol{n} \times \boldsymbol{E} = \boldsymbol{0} \quad \text{on } \partial\Omega, \tag{4.5}$$

where $\boldsymbol{D}_0, \boldsymbol{D}_1, \boldsymbol{E}_0, \boldsymbol{E}_1$ and H_0 are some properly given functions, $\boldsymbol{n}$ is the unit outward normal vector to $\partial\Omega$, and Ω denotes a polygonal domain in R^2.

Using the stability obtained in [54, Theorem 2.1] and replacing both $\nabla \times \boldsymbol{E}$ and $\nabla \times \partial_t \boldsymbol{E}$ by (4.4), we can rewrite Theorem 2.1 of [54] as below, which is totally different from those established in [51, 52].

Theorem 4.2.1. For the solution $(\boldsymbol{D}, H, \boldsymbol{E})$ of (4.1)-(4.4), let the energy be defined as

$$\begin{aligned} ENG(t) := \Big[& \varepsilon_0 \lambda_2 ||M_A^{-\frac{1}{2}} \partial_{t^2} \boldsymbol{E}||^2 + 2\varepsilon_0 \lambda_2 \omega_p^2 ||M_A^{-\frac{1}{2}} \partial_t \boldsymbol{E}||^2 + \varepsilon_0 \lambda_2 \omega_p^4 ||M_A^{-\frac{1}{2}} \boldsymbol{E}||^2 \\ & + \mu_0 \mu \left(\omega_p^2 ||\partial_t H||^2 + ||\partial_{t^2} H||^2 \right) + ||\partial_t \boldsymbol{D}||^2 + ||M_C^{\frac{1}{2}} \boldsymbol{D}||^2 \Big] (t). \end{aligned} \tag{4.6}$$

Here and below the square root of a matrix M_C is denoted as $M_C^{\frac{1}{2}}$, and $||\cdot||^2 := ||\cdot||^2_{L^2(\Omega)}$.

Then we have the following energy identity:

$$ENG(t) - ENG(0) = 2\int_0^t \Big[\varepsilon_0 \lambda_2 (M_A^{-1}\partial_{t^2}\boldsymbol{E} + \omega_p^2 M_A^{-1}\boldsymbol{E}, \partial_t \boldsymbol{D}) \\ +(M_C \partial_t \boldsymbol{D}, \partial_{t^2}\boldsymbol{E}) + \omega_p^2 (M_C \boldsymbol{D}, \partial_t \boldsymbol{E})\Big](s)ds. \tag{4.7}$$

Furthermore, this leads to the stability:

$$ENG(t) \leq ENG(0)\cdot \exp(C_* t), \quad \forall\ t \in [0,T], \tag{4.8}$$

where the constant $C_* > 0$ depends on the physical parameters $\varepsilon_0, \mu_0, d, H_1, H_2$ and ω_p.

4.3 The semi-discrete LDG method

In this subsection, we introduce the LDG method for the carpet cloak model. We consider a rectangular physical domain $\Omega = [a,b]\times[c,d]$ to solve (4.1)-(4.4) for simplicity, and the domain is partitioned by a regular triangular mesh, $\Omega = \cup_{e\in\mathcal{T}_h} e$. Here $\mathcal{T}_h$ is a triangulation on Ω, and h is the mesh size, representing the largest diameter of all triangles. Tensor-product rectangular meshes will also be considered later. The time domain $[0,T]$ is discretized into N_t+1 uniform intervals by discrete times $0 = t_0 < t_1 < \cdots < t_{N_t+1} = T$, where $t_n = n\cdot\tau$, and the time step size $\tau = \frac{T}{N_t+1}$.

V_h^k denotes the finite element space of piecewise polynomials, i.e.,

$$V_h^k = \{v:\ v|_e \in P_k(e),\ \forall\ e\in\mathcal{T}_h\}, \tag{4.9}$$

where P_k is the space of polynomials of degree less or equal to k.

We use u_h to denote the corresponding numerical solution of the variable u, which is in the finite element space V_h^k. Note that functions contained in V_h^k can have discontinuities across the element interfaces. In the line integral over the boundary of a cell, $u_h^{(in)}$ denotes the value of u_h taken from inside of that cell, and $u_h^{(out)}$ denotes the value of u_h taken from the neighboring cell sharing that boundary. Furthermore, we use $(\cdot)$ and $\|\cdot\|$ to denote the inner product and the L^2 norm over the domain Ω respectively.

Then, the semi-discrete LDG method for (4.1)-(4.4) is generated as follows: Find $E_{xh}, E_{yh}, H_h, D_{xh}, D_{yh} \in C^1([0,T]; V_h^k)$ such that

$$\int_e \partial_t D_{xh}\phi_x + \int_e H_h \partial_y \phi_x - \int_{\partial e} \hat{H}_h \phi_x^{(\text{in})} n_y^{(in)} = 0, \tag{4.10}$$

$$\int_e \partial_t D_{yh}\phi_y - \int_e H_h \partial_x \phi_y + \int_{\partial e} \hat{H}_h \phi_y^{(\text{in})} n_x^{(in)} = 0, \tag{4.11}$$

$$\varepsilon_0 \lambda_2 \int_e \left(M_A^{-1} \partial_{t^2} \boldsymbol{E}_h + \omega_p^2 M_A^{-1} \boldsymbol{E}_h \right) \cdot \boldsymbol{u} = \int_e (\partial_{t^2} \boldsymbol{D}_h + M_C \boldsymbol{D}_h) \cdot \boldsymbol{u}, \tag{4.12}$$

$$\mu_0 \mu \int_e \partial_t H_h \psi - \int_e E_{yh} \partial_x \psi + \int_e E_{xh} \partial_y \psi + \int_{\partial e} (\hat{E}_{yh} n_x^{(in)} - \hat{E}_{xh} n_y^{(in)}) \psi^{(\text{in})} = 0, \tag{4.13}$$

for all test functions $\phi_x, \phi_y, \psi, u_1, u_2 \in V_h^k$ and all cells $e \in \mathcal{T}_h$, where $\boldsymbol{u} = (u_1, u_2)'$. $\hat{H}_h, \hat{E}_{yh}, \hat{E}_{xh}$ are the cell boundary terms obtained from integration by parts, and they are the so-called numerical fluxes. On the cell boundary ∂e, $\boldsymbol{n}^{(in)} = (n_x^{(in)}, n_y^{(in)})$ represents the unit normal vector pointing towards the outside of the element e.

To define the numerical fluxes in a triangulation, we first pick a fixed direction $\boldsymbol{\beta}$ not parallel to any triangle boundary edge. On each boundary edge of an element, there is an outward normal direction, $\boldsymbol{n}$, orthogonal to that edge. We call a side as the "right" side if $\boldsymbol{n} \cdot \boldsymbol{\beta} < 0$, and the "left" side if vice versa. We apply the commonly used alternating fluxes in LDG methods into our scheme, which are defined as choosing E_{xh} and E_{yh} on the "right" side and H_h on the "left" side:

$$\hat{E}_{xh} = E_{xh}^R, \tag{4.14}$$

$$\hat{E}_{yh} = E_{yh}^R, \tag{4.15}$$

$$\hat{H}_h = H_h^L. \tag{4.16}$$

A more detailed explanation of alternating fluxes for triangulations can be found in [85]. It is easy to check that in a rectangular mesh, when $\boldsymbol{\beta} = (1,1)$, the definitions of the "left" and "right" sides are consistent with the exact left (bottom) and right (top) sides on a vertical (horizontal) boundary. The above definition of alternating fluxes is enough when applying the periodic boundary condition. However, to satisfy the PEC boundary condition in (4.5), we take

$$\hat{E}_{xh} = 0, \text{ on } y = c, d, \tag{4.17}$$

$$\hat{E}_{yh} = 0, \text{ on } x = a, b, \tag{4.18}$$

$$\hat{H}_h = H_h^{(in)}, \text{ on } \partial\Omega. \tag{4.19}$$

4.3.1 The stability analysis

In this subsection, we will show that the solutions of our proposed semi-discrete DG method satisfy the same energy identity as in the continuous level (4.7), which leads to the stability of the method.

Theorem 4.3.1. For the semi-discrete DG method (4.10)-(4.13) with alternating fluxes (4.14)-(4.19), we define the energy:

$$
\begin{aligned}
ENG_h(t) := \Big[&\|\partial_t \boldsymbol{D}_h\|^2 + \|M_C^{\frac{1}{2}}\boldsymbol{D}_h\|^2 + \epsilon_0\lambda_2 \Big(\|M_A^{-\frac{1}{2}}\partial_{t^2}\boldsymbol{E}_h\|^2 \\
&+ 2\omega_p^2\|M_A^{-\frac{1}{2}}\partial_t\boldsymbol{E}_h\|^2 + \omega_p^4\|M_A^{-\frac{1}{2}}\boldsymbol{E}_h\|^2 \Big) + \mu_0\mu\left(\omega_p^2\|\partial_t H_h\|^2 + \|\partial_{t^2}H_h\|^2\right)\Big](t),
\end{aligned}
\tag{4.20}
$$

then, the energy satisfies the following energy identity: For any $t \geq 0$:

$$
\begin{aligned}
ENG_h(t) - ENG_h(0) = 2\int_0^t \Big[&\epsilon_0\lambda_2 \left(M_A^{-1}\partial_{t^2}\boldsymbol{E}_h + \omega_p^2 M_A^{-1}\boldsymbol{E}_h, \partial_t\boldsymbol{D}_h \right) \\
&+ (M_C\partial_t\boldsymbol{D}_h, \partial_{t^2}\boldsymbol{E}_h) + \omega_p^2\left(M_C\boldsymbol{D}_h, \partial_t\boldsymbol{E}_h\right)\Big](s)ds.
\end{aligned}
\tag{4.21}
$$

Furthermore, it leads to the stability:

$$
ENG_h(t) \leq \exp(C^* t)\cdot ENG_h(0), \qquad \forall t \in [0, T],
\tag{4.22}
$$

with the constant C^* depending only on the physical parameters $\varepsilon_0, \mu_0, d, H_1, H_2$ and ω_p.

Proof. To make our proof easy to follow, we divide it into several major parts.

(I) Choosing $\boldsymbol{u} = \partial_t\boldsymbol{D}_h$ in (4.12), we obtain

$$
\frac{1}{2}\frac{d}{dt}\left[\|\partial_t\boldsymbol{D}_h\|^2 + \|M_C^{\frac{1}{2}}\boldsymbol{D}_h\|^2\right] = \epsilon_0\lambda_2\left(M_A^{-1}\partial_{t^2}\boldsymbol{E}_h + \omega_p^2 M_A^{-1}\boldsymbol{E}_h, \partial_t\boldsymbol{D}_h\right).
\tag{4.23}
$$

Differentiating (4.12) with respect to t and choosing $\boldsymbol{u} = \partial_{t^2}\boldsymbol{E}_h$, we have

$$
\frac{\epsilon_0\lambda_2}{2}\frac{d}{dt}\left[\|M_A^{-\frac{1}{2}}\partial_{t^2}\boldsymbol{E}_h\|^2 + \omega_p^2\|M_A^{-\frac{1}{2}}\partial_t\boldsymbol{E}_h\|^2\right] = \left(\partial_{t^3}\boldsymbol{D}_h + M_C\partial_t\boldsymbol{D}_h, \partial_{t^2}\boldsymbol{E}_h\right).
\tag{4.24}
$$

Adding (4.23) and (4.24) together, we obtain

$$\begin{aligned}\frac{1}{2}\frac{d}{dt}&\left[||\partial_t \boldsymbol{D}_h||^2 + ||M_C^{\frac{1}{2}}\boldsymbol{D}_h||^2 + \epsilon_0\lambda_2\left(||M_A^{-\frac{1}{2}}\partial_{t^2}\boldsymbol{E}_h||^2 + \omega_p^2||M_A^{-\frac{1}{2}}\partial_t\boldsymbol{E}_h||^2\right)\right] \\ &= \epsilon_0\lambda_2\left(M_A^{-1}\partial_{t^2}\boldsymbol{E}_h + \omega_p^2 M_A^{-1}\boldsymbol{E}_h, \partial_t\boldsymbol{D}_h\right) + \left(\partial_{t^3}\boldsymbol{D}_h + M_C\partial_t\boldsymbol{D}_h, \partial_{t^2}\boldsymbol{E}_h\right).\end{aligned} \tag{4.25}$$

(II) To control the term $\boldsymbol{E}_h$ on the right hand side (RHS) of (4.25), we choose $\boldsymbol{u} = \partial_t\boldsymbol{E}_h$ in (4.12) to obtain

$$\frac{\epsilon_0\lambda_2}{2}\frac{d}{dt}\left[||M_A^{-\frac{1}{2}}\partial_t\boldsymbol{E}_h||^2 + \omega_p^2||M_A^{-\frac{1}{2}}\boldsymbol{E}_h||^2\right] = \left(\partial_{t^2}\boldsymbol{D}_h + M_C\boldsymbol{D}_h, \partial_t\boldsymbol{E}_h\right). \tag{4.26}$$

Multiplying (4.26) by ω_p^2, then adding the result to (4.25), we have

$$\begin{aligned}\frac{1}{2}\frac{d}{dt}&\Big[||\partial_t\boldsymbol{D}_h||^2 + ||M_C^{\frac{1}{2}}\boldsymbol{D}_h||^2 + \epsilon_0\lambda_2\Big(||M_A^{-\frac{1}{2}}\partial_{t^2}\boldsymbol{E}_h||^2 + 2\omega_p^2||M_A^{-\frac{1}{2}}\partial_t\boldsymbol{E}_h||^2 \\ &+ \omega_p^4||M_A^{-\frac{1}{2}}\boldsymbol{E}_h||^2\Big)\Big] = \epsilon_0\lambda_2\left(M_A^{-1}\partial_{t^2}\boldsymbol{E}_h + \omega_p^2 M_A^{-1}\boldsymbol{E}_h, \partial_t\boldsymbol{D}_h\right) \\ &+ \left(\partial_{t^3}\boldsymbol{D}_h + M_C\partial_t\boldsymbol{D}_h, \partial_{t^2}\boldsymbol{E}_h\right) + \omega_p^2\left(\partial_{t^2}\boldsymbol{D}_h + M_C\boldsymbol{D}_h, \partial_t\boldsymbol{E}_h\right).\end{aligned} \tag{4.27}$$

(III) Now we need to control the terms $\partial_{t^3}\boldsymbol{D}_h$ and $\partial_{t^2}\boldsymbol{D}_h$ on the RHS of (4.27).

Differentiating both (4.10) and (4.11) with respect to t, choosing $\phi_x = \partial_t E_{xh}$ and $\phi_y = \partial_t E_{yh}$ in (4.10) and (4.11), respectively, then adding the results together, we have

$$\begin{aligned}&\int_e \partial_{t^2}\boldsymbol{D}_h \cdot \partial_t\boldsymbol{E}_h + \int_e \partial_t H_h(\partial_y\partial_t E_{xh} - \partial_x\partial_t E_{yh}) \\ &- \int_{\partial e} \partial_t\hat{H}_h\partial_t E_{xh}^{(\mathrm{in})} n_y^{(in)} + \int_{\partial e} \partial_t\hat{H}_h\partial_t E_{yh}^{(\mathrm{in})} n_x^{(in)} = 0.\end{aligned} \tag{4.28}$$

Differentiating (4.13) with respect to t, choosing $\psi = \partial_t H_h$, then using integration by parts, we have

$$\begin{aligned}\mu_0\mu \int_e \partial_{t^2} H_h \partial_t H_h + \int_e \partial_t H_h (\partial_x \partial_t E_{yh} - \partial_y \partial_t E_{xh}) - \int_{\partial e} \partial_t E_{yh}^{(\mathrm{in})} \partial_t H_h^{(in)} n_x^{(in)} \\ + \int_{\partial e} \partial_t E_{xh}^{(\mathrm{in})} \partial_t H_h^{(in)} n_y^{(in)} + \int_{\partial e} (\partial_t \hat{E}_{yh} n_x^{(\mathrm{in})} - \partial_t \hat{E}_{xh} n_y^{(in)}) \partial_t H_h^{(in)} = 0.\end{aligned} \tag{4.29}$$

Adding (4.28) and (4.29) together over all elements, we have

$$\sum_{e \in \mathcal{T}_h} \int_e \partial_{t^2} \boldsymbol{D}_h \cdot \partial_t \boldsymbol{E}_h + \frac{\mu_0 \mu}{2} \frac{d}{dt} \|\partial_t H_h\|^2 + F_x - F_y = 0, \tag{4.30}$$

where we define

$$F_x = \sum_{e \in \mathcal{T}_h} \int_{\partial e} \left(-\partial_t \hat{H}_h \partial_t E_{xh}^{(in)} n_y^{(in)} + \partial_t H_h^{(in)} \partial_t E_{xh}^{(in)} n_y^{(in)} - \partial_t H_h^{(in)} \partial_t \hat{E}_{xh} n_y^{(in)} \right), \tag{4.31}$$

$$F_y = \sum_{e \in \mathcal{T}_h} \int_{\partial e} \left(\partial_t \hat{H}_h \partial_t E_{yh}^{(in)} n_x^{(in)} - \partial_t H_h^{(in)} \partial_t E_{yh}^{(in)} n_x^{(in)} + \partial_t H_h^{(in)} \partial_t \hat{E}_{yh} n_x^{(in)} \right). \tag{4.32}$$

By regrouping terms by sides of the elements and using the definitions of the numerical fluxes $\hat{H}_h$ and $\hat{E}_{xh}$, we have:

$$\begin{aligned} F_x = & \sum_{s \in \mathcal{S}_I} n_y^R \int_s \Big(-\partial_t H_h^L \partial_t E_{xh}^R + \partial_t H_h^L \partial_t E_{xh}^L + \partial_t H_h^R \partial_t E_{xh}^R \\ & \qquad - \partial_t H_h^L \partial_t E_{xh}^L - \partial_t H_h^R \partial_t E_{xh}^R + \partial_t H_h^L \partial_t E_{xh}^R \Big) + \\ & \sum_{s \in \mathcal{S}_{Top}} n_y^R \int_s \left(-\partial_t \hat{H}_h \partial_t E_{xh}^{(in)} + \partial_t H_h^{(in)} \partial_t E_{xh}^{(in)} - \partial_t H_h^{(in)} \partial_t \hat{E}_{xh} \right) + \\ & \sum_{s \in \mathcal{S}_{Bottom}} n^R \int_s \left(-\partial_t \hat{H}_h \partial_t E_{xh}^{(in)} + \partial_t H_h^{(in)} \partial_t E_{xh}^{(in)} - \partial_t H_h^{(in)} \partial_t \hat{E}_{xh} \right) = 0, \end{aligned} \tag{4.33}$$

where $\mathcal{S}_I$ denotes the set of all non-boundary sides, $\mathcal{S}_{Top}$ represents the set of sides on $y = d$, and $\mathcal{S}_{Bottom}$ on $y = c$.

Similarly, we can prove that $F_y = 0$.

Then using the results of $F_x = F_y = 0$ in (4.30), we obtain

$$\sum_{e\in\mathcal{T}_h}\int_e \partial_{t^2}\boldsymbol{D}_h \cdot \partial_t \boldsymbol{E}_h = -\frac{\mu_0\mu}{2}\frac{d}{dt}||\partial_t H_h||^2. \tag{4.34}$$

Following the same argument, we can prove that

$$\sum_{e\in\mathcal{T}_h}\int_e \partial_{t^3}\boldsymbol{D}_h \cdot \partial_{t^2} \boldsymbol{E}_h = -\frac{\mu_0\mu}{2}\frac{d}{dt}||\partial_{t^2} H_h||^2. \tag{4.35}$$

Substituting (4.34) and (4.35) into (4.27), we obtain

$$\begin{aligned}
&\frac{1}{2}\frac{d}{dt}\Big[||\partial_t\boldsymbol{D}_h||^2 + ||M_C^{\frac{1}{2}}\boldsymbol{D}_h||^2 + \epsilon_0\lambda_2\Big(||M_A^{-\frac{1}{2}}\partial_{t^2}\boldsymbol{E}_h||^2 + 2\omega_p^2||M_A^{-\frac{1}{2}}\partial_t\boldsymbol{E}_h||^2 + \\
&\qquad\quad \omega_p^4||M_A^{-\frac{1}{2}}\boldsymbol{E}_h||^2\Big) + \mu_0\mu\left(\omega_p^2||\partial_t H_h||^2 + ||\partial_{t^2}H_h||^2\right)\Big] \\
&= \epsilon_0\lambda_2\left(M_A^{-1}\partial_{t^2}\boldsymbol{E}_h + \omega_p^2 M_A^{-1}\boldsymbol{E}_h, \partial_t\boldsymbol{D}_h\right) + \\
&\qquad\quad (M_C\partial_t\boldsymbol{D}_h, \partial_{t^2}\boldsymbol{E}_h) + \omega_p^2\left(M_C\boldsymbol{D}_h, \partial_t\boldsymbol{E}_h\right).
\end{aligned} \tag{4.36}$$

Integrating (4.36) with respect to t from 0 to t, we obtain the energy identity (4.21). Then we apply the Cauchy-Schwarz inequality to all terms in the RHS of (4.36), and use the Gronwall inequality to complete the proof.

■

4.3.2 The error analysis

In this subsection, we will show the sub-optimal error estimate of the semi-discrete DG method on unstructured meshes, and the optimal error estimate of the DG method with a modified alternating flux on tensor-product rectangular meshes with tensor-product DG spaces.

4.3.2.1 The error analysis on unstructured meshes

The errors between the exact solutions (E_x, E_y, D_x, D_y, H) of (4.1)-(4.4) and the corresponding numerical solutions $(E_{xh}, E_{yh}, D_{xh}, D_{yh}, H_h)$ of the semi-discrete scheme (4.10)-(4.13) are denoted as

$$\begin{aligned}&\mathcal{E}_{E_x} = E_x - E_{xh},\ \mathcal{E}_{E_y} = E_y - E_{yh},\ \mathcal{E}_{D_x} = D_x - D_{xh},\\&\mathcal{E}_{D_y} = D_y - D_{yh},\ \mathcal{E}_H = H - H_h,\end{aligned}$$

and we define $\mathcal{E}_{\boldsymbol{D}} = (\mathcal{E}_{D_x},\ \mathcal{E}_{D_y})$, and $\mathcal{E}_{\boldsymbol{E}} = (\mathcal{E}_{E_x},\ \mathcal{E}_{E_y})$.

Subtracting (4.10)-(4.13) from the weak formulation of the PDEs (4.1)-(4.4), we obtain the following error equations:

$$\int_e \partial_t \mathcal{E}_{D_x} \phi_x + \int_e \mathcal{E}_H \partial_y \phi_x - \int_{\partial e} \hat{\mathcal{E}}_H \phi_x^{(\mathrm{in})} n_y^{(in)} = 0, \tag{4.37}$$

$$\int_e \partial_t \mathcal{E}_{D_y} \phi_y - \int_e \mathcal{E}_H \partial_x \phi_y + \int_{\partial e} \hat{\mathcal{E}}_H \phi_y^{(\mathrm{in})} n_x^{(in)} = 0, \tag{4.38}$$

$$\varepsilon_0 \lambda_2 \int_e \Big(M_A^{-1} \partial_{t^2} \mathcal{E}_{\boldsymbol{E}} + \omega_p^2 M_A^{-1} \mathcal{E}_{\boldsymbol{E}} \Big) \cdot \boldsymbol{u} = \int_e \Big(\partial_{t^2} \mathcal{E}_{\boldsymbol{D}} + M_C \mathcal{E}_{\boldsymbol{D}} \Big) \cdot \boldsymbol{u}, \tag{4.39}$$

$$\mu_0 \mu \int_e \partial_t \mathcal{E}_H \psi - \int_e \mathcal{E}_{E_y} \partial_x \psi + \int_e \mathcal{E}_{E_x} \partial_y \psi + \int_{\partial e} (\hat{\mathcal{E}}_{E_y} n_x^{(in)} - \hat{\mathcal{E}}_{E_x} n_y^{(in)}) \psi^{(\mathrm{in})} = 0. \tag{4.40}$$

Then, we have the following theorem:

Theorem 4.3.2. Suppose that the analytical solutions (E_x, E_y, D_x, D_y, H) of (4.1)-(4.4) are smooth enough, and $(E_{xh}, E_{yh}, D_{xh}, D_{yh}, H_h)$ are the corresponding numerical solutions of (4.10)-(4.13). With the alternating flux (4.14)-(4.16) and the PEC boundary condition (4.17)-(4.19), we have the following error estimate:

$$\begin{aligned}
&\Bigg[||\partial_t \boldsymbol{D} - \partial_t \boldsymbol{D}_h||^2 + ||M_C^{\frac{1}{2}}(\boldsymbol{D} - \boldsymbol{D}_h)||^2 + \epsilon_0\lambda_2\Big(||M_A^{-\frac{1}{2}}(\partial_{t^2}\boldsymbol{E} - \partial_{t^2}\boldsymbol{E}_h)||^2 \\
&\quad + 2\omega_p^2||M_A^{-\frac{1}{2}}(\partial_t \boldsymbol{E} - \partial_t \boldsymbol{E}_h)||^2 + \omega_p^4||M_A^{-\frac{1}{2}}(\boldsymbol{E} - \boldsymbol{E}_h)||^2\Big) \\
&\quad + \mu_0\mu\left(\omega_p^2||\partial_t H - \partial_t H_h||^2 + ||\partial_{t^2}H - \partial_{t^2}H_h||^2\right)\Bigg](t) \\
&\leq Ch^{2k} + \Bigg[||\partial_t \boldsymbol{D} - \partial_t \boldsymbol{D}_h||^2 + ||M_C^{\frac{1}{2}}(\boldsymbol{D} - \boldsymbol{D}_h)||^2 + \\
&\quad \epsilon_0\lambda_2\Big(||M_A^{-\frac{1}{2}}(\partial_{t^2}\boldsymbol{E} - \partial_{t^2}\boldsymbol{E}_h)||^2 + 2\omega_p^2||M_A^{-\frac{1}{2}}(\partial_t \boldsymbol{E} - \partial_t \boldsymbol{E}_h)||^2 + \\
&\quad \omega_p^4||M_A^{-\frac{1}{2}}(\boldsymbol{E} - \boldsymbol{E}_h)||^2\Big) + \mu_0\mu\left(\omega_p^2||\partial_t H - \partial_t H_h||^2 + ||\partial_{t^2}H - \partial_{t^2}H_h||^2\right)\Bigg](0).
\end{aligned} \tag{4.41}$$

Here $k \geq 1$ is the order of the basis function V_h^k, and C is a positive constant independent of the mesh size h.

Proof. We first decompose each of the error function $(\mathcal{E}_{E_x}, \mathcal{E}_{E_y}, \mathcal{E}_{D_x}, \mathcal{E}_{D_y}, \mathcal{E}_H)$ into two parts respectively:

$$\mathcal{E}_{E_x} = E_x - E_{xh} = (\Pi E_x - E_{xh}) - (\Pi E_x - E_x) := \xi_{E_x} - \eta_{E_x},$$

$$\mathcal{E}_{E_y} = E_y - E_{yh} = (\Pi E_y - E_{yh}) - (\Pi E_y - E_y) := \xi_{E_y} - \eta_{E_y},$$

$$\mathcal{E}_{D_x} = D_x - D_{xh} = (\Pi D_x - D_{xh}) - (\Pi D_x - D_x) := \xi_{D_x} - \eta_{D_x},$$

$$\mathcal{E}_{D_y} = D_y - D_{yh} = (\Pi D_y - D_{yh}) - (\Pi D_y - D_y) := \ \xi_{D_y} - \eta_{D_y},$$

$$\mathcal{E}_H \ = \ H - H_h \ = \ (\Pi H - H_h) - (\Pi H - H) \ := \ \xi_H - \eta_H,$$

where Π presents the standard L_2 projection onto V_h^k.

Similar as the stability proof, we take $\boldsymbol{u} = \partial_t \xi_{\boldsymbol{D}}$ and $\boldsymbol{u} = \partial_t \xi_{\boldsymbol{E}}$ respectively in (4.39), and we differentiate (4.39) with respect to t and let $\boldsymbol{u} = \partial_{t^2}\xi_{\boldsymbol{D}}$. Then we sum over all elements in the domain. By putting all terms containing η to the RHS, and the rest terms to the left hand side (LHS), we get:

$$\begin{aligned}\frac{1}{2}\frac{d}{dt}\Big[||\partial_t\xi_{\boldsymbol{D}}||^2 + ||M_C^{\frac{1}{2}}\xi_{\boldsymbol{D}}||^2\Big] &- \epsilon_0\lambda_2\left(M_A^{-1}\partial_{t^2}\xi_{\boldsymbol{E}} + \omega_p^2 M_A^{-1}\xi_{\boldsymbol{E}}, \partial_t\xi_{\boldsymbol{D}}\right)\\ &= \left(\partial_{t^2}\eta_{\boldsymbol{D}}, \partial_t\xi_{\boldsymbol{D}}\right) + \left(M_C\eta_{\boldsymbol{D}}, \partial_t\xi_{\boldsymbol{D}}\right)\\ &\quad - \varepsilon_0\lambda_2\left(M_A^{-1}\partial_{t^2}\eta_{\boldsymbol{E}} + M_A^{-1}\eta_{\boldsymbol{E}}, \partial_t\xi_{\boldsymbol{D}}\right),\end{aligned} \tag{4.42}$$

$$\begin{aligned}\frac{\epsilon_0\lambda_2}{2}\frac{d}{dt}\Big[||M_A^{-\frac{1}{2}}\partial_t\xi_{\boldsymbol{E}}||^2 + \omega_p^2||M_A^{-\frac{1}{2}}\xi_{\boldsymbol{E}}||^2\Big] &- \left(\partial_{t^2}\xi_{\boldsymbol{D}} + M_C\xi_{\boldsymbol{D}}, \partial_t\xi_{\boldsymbol{E}}\right)\\ &= \varepsilon_0\lambda_2\left(M_A^{-1}\partial_{t^2}\eta_{\boldsymbol{E}}, \partial_t\xi_{\boldsymbol{E}}\right) + \varepsilon_0\lambda_2\omega_p^2\left(M_A^{-1}\eta_{\boldsymbol{E}}, \partial_t\xi_{\boldsymbol{E}}\right)\\ &\quad - \left(\partial_{t^2}\eta_{\boldsymbol{D}} + M_C\eta_{\boldsymbol{D}}, \partial_t\xi_{\boldsymbol{E}}\right),\end{aligned} \tag{4.43}$$

$$\begin{aligned}\frac{\epsilon_0\lambda_2}{2}\frac{d}{dt}\Big[||M_A^{-\frac{1}{2}}\partial_{t^2}\xi_{\boldsymbol{E}}||^2 + \omega_p^2||M_A^{-\frac{1}{2}}\partial_t\xi_{\boldsymbol{E}}||^2\Big] &- \left(\partial_{t^3}\xi_{\boldsymbol{D}} + M_C\partial_t\xi_{\boldsymbol{D}}, \partial_{t^2}\xi_{\boldsymbol{E}}\right)\\ &= \varepsilon_0\lambda_2\left(M_A^{-1}\partial_{t^3}\eta_{\boldsymbol{E}}, \partial_{t^2}\xi_{\boldsymbol{E}}\right) + \varepsilon_0\lambda_2\omega_p^2\left(M_A^{-1}\partial_t\eta_{\boldsymbol{E}}, \partial_{t^2}\xi_{\boldsymbol{E}}\right)\\ &\quad - \left(\partial_{t^3}\eta_{\boldsymbol{D}} + M_C\partial_t\eta_{\boldsymbol{D}}, \partial_{t^2}\xi_{\boldsymbol{E}}\right),\end{aligned} \tag{4.44}$$

where $\xi_{\boldsymbol{D}} = (\xi_{D_x}, \xi_{D_y})$, and $\xi_{\boldsymbol{E}} = (\xi_{E_x}, \xi_{E_y})$. $\eta_{\boldsymbol{D}}$ and $\eta_{\boldsymbol{E}}$ are defined similarly.

Next, we differentiate (4.37), (4.38), and (4.40) with respect to t, and choose $\phi_x = \partial_t\xi_{E_x}$, $\phi_y = \partial_t\xi_{E_y}$ and $\psi = \partial_t\xi_H$ respectively. Then we sum up these three equations, and sum over all elements in the domain to obtain:

$$\begin{aligned}&\left(\partial_{t^2}\mathcal{E}_{\boldsymbol{D}}, \partial_t\xi_{\boldsymbol{E}}\right) + \mu_0\mu(\partial_{t^2}\mathcal{E}_H, \partial_t\xi_H) + \left(\partial_t\mathcal{E}_H, \partial_y\partial_t\xi_{E_x} - \partial_x\partial_t\xi_{E_y}\right)\\ &- (\partial_t\mathcal{E}_{E_y}, \partial_x\partial_t\xi_H) + \left(\partial_t\mathcal{E}_{E_x}, \partial_y\partial_t\xi_H\right) + \sum_{e\in\mathcal{T}_h}\Bigg(-\int_{\partial e}\partial_t\hat{\mathcal{E}}_H\partial_t\xi_{E_x}^{(\mathrm{in})}n_y^{(in)}\\ &+ \int_{\partial e}\partial_t\hat{\mathcal{E}}_H\partial_t\xi_{E_y}^{(\mathrm{in})}n_x^{(in)} + \int_{\partial e}(\partial_t\hat{\mathcal{E}}_{E_y}n_x^{(\mathrm{in})} - \partial_t\hat{\mathcal{E}}_{E_x}n_y^{(in)})\partial_t\xi_H^{(in)}\Bigg) = 0.\end{aligned}\tag{4.45}$$

By applying the error decomposition, we have:

$$\begin{aligned}&\left(\partial_{t^2}\xi_{\boldsymbol{D}}, \partial_t\xi_{\boldsymbol{E}}\right) + \mu_0\mu(\partial_{t^2}\xi_H, \partial_t\xi_H) + \left(\partial_t\xi_H, \partial_y\partial_t\xi_{E_x} - \partial_x\partial_t\xi_{E_y}\right)\\ &\quad - (\partial_t\xi_{E_y}, \partial_x\partial_t\xi_H) + \left(\partial_t\xi_{E_x}, \partial_y\partial_t\xi_H\right) + \sum_{e\in\mathcal{T}_h}\Bigg(-\int_{\partial e}\partial_t\hat{\xi}_H\partial_t\xi_{E_x}^{(\mathrm{in})}n_y^{(in)}\\ &\quad + \int_{\partial e}\partial_t\hat{\xi}_H\partial_t\xi_{E_y}^{(\mathrm{in})}n_x^{(in)} + \int_{\partial e}(\partial_t\hat{\xi}_{E_y}n_x^{(\mathrm{in})} - \partial_t\hat{\xi}_{E_x}n_y^{(in)})\partial_t\xi_H^{(in)}\Bigg)\\ &= \left(\partial_{t^2}\eta_{\boldsymbol{D}}, \partial_t\xi_{\boldsymbol{E}}\right) + \mu_0\mu(\partial_{t^2}\eta_H, \partial_t\xi_H) + \left(\partial_t\eta_H, \partial_y\partial_t\xi_{E_x} - \partial_x\partial_t\xi_{E_y}\right)\\ &\quad - (\partial_t\eta_{E_y}, \partial_x\partial_t\xi_H) + \left(\partial_t\eta_{E_x}, \partial_y\partial_t\xi_H\right) + \sum_{e\in\mathcal{T}_h}\Bigg(-\int_{\partial e}\partial_t\hat{\eta}_H\partial_t\xi_{E_x}^{(\mathrm{in})}n_y^{(in)}\\ &\quad + \int_{\partial e}\partial_t\hat{\eta}_H\partial_t\xi_{E_y}^{(\mathrm{in})}n_x^{(in)} + \int_{\partial e}(\partial_t\hat{\eta}_{E_y}n_x^{(\mathrm{in})} - \partial_t\hat{\eta}_{E_x}n_y^{(in)})\partial_t\xi_H^{(in)}\Bigg).\end{aligned}\tag{4.46}$$

Using the same argument as the stability analysis on the LHS, we obtain:

$$\begin{aligned}
&\left(\partial_{t^2}\xi_{\boldsymbol{D}}, \partial_t\xi_{\boldsymbol{E}}\right) + \frac{\mu_0\mu}{2}\partial_t\|\partial_t\xi_H\|^2 \\
&= \left(\partial_{t^2}\eta_{\boldsymbol{D}}, \partial_t\xi_{\boldsymbol{E}}\right) + \mu_0\mu(\partial_{t^2}\eta_H, \partial_t\xi_H) + \left(\partial_t\eta_H, \partial_y\partial_t\xi_{E_x} - \partial_x\partial_t\xi_{E_y}\right) \\
&\quad - (\partial_t\eta_{E_y}, \partial_x\partial_t\xi_H) + \left(\partial_t\eta_{E_x}, \partial_y\partial_t\xi_H\right) + \sum_{e\in\mathcal{T}_h}\Bigg(-\int_{\partial e}\partial_t\hat{\eta}_H\partial_t\xi_{E_x}^{(\text{in})}n_y^{(in)} \\
&\quad + \int_{\partial e}\partial_t\hat{\eta}_H\partial_t\xi_{E_y}^{(\text{in})}n_x^{(in)} + \int_{\partial e}(\partial_t\hat{\eta}_{E_y}n_x^{(\text{in})} - \partial_t\hat{\eta}_{E_x}n_y^{(in)})\partial_t\xi_H^{(in)}\Bigg).
\end{aligned} \tag{4.47}$$

Similarly, by differentiating (4.37), (4.38), and (4.40) with respect to t^2, and choosing $\phi_x = \partial_{t^2}\xi_{E_x}$, $\phi_y = \partial_{t^2}\xi_{E_y}$ and $\psi = \partial_{t^2}\xi_H$ respectively, we have:

$$\begin{aligned}
&\left(\partial_{t^3}\xi_{\boldsymbol{D}}, \partial_{t^2}\xi_{\boldsymbol{E}}\right) + \frac{\mu_0\mu}{2}\partial_t\|\partial_{t^2}\xi_H\|^2 \\
&= \left(\partial_{t^3}\eta_{\boldsymbol{D}}, \partial_{t^2}\xi_{\boldsymbol{E}}\right) + \mu_0\mu(\partial_{t^3}\eta_H, \partial_{t^2}\xi_H) + \left(\partial_{t^2}\eta_H, \partial_y\partial_{t^2}\xi_{E_x} - \partial_x\partial_{t^2}\xi_{E_y}\right) \\
&\quad - (\partial_{t^2}\eta_{E_y}, \partial_x\partial_{t^2}\xi_H) + \left(\partial_{t^2}\eta_{E_x}, \partial_y\partial_{t^2}\xi_H\right) + \sum_{e\in\mathcal{T}_h}\Bigg(-\int_{\partial e}\partial_{t^2}\hat{\eta}_H\partial_{t^2}\xi_{E_x}^{(\text{in})}n_y^{(in)} \\
&\quad + \int_{\partial e}\partial_{t^2}\hat{\eta}_H\partial_{t^2}\xi_{E_y}^{(\text{in})}n_x^{(in)} + \int_{\partial e}(\partial_{t^2}\hat{\eta}_{E_y}n_x^{(\text{in})} - \partial_{t^2}\hat{\eta}_{E_x}n_y^{(in)})\partial_{t^2}\xi_H^{(in)}\Bigg).
\end{aligned} \tag{4.48}$$

We multiply (4.43) and (4.47) by ω_p^2 and sum them with (4.42), (4.44) and (4.48) to attain the formula for the LHS:

$$\begin{aligned}
LHS = \frac{1}{2}\frac{d}{dt}\Bigg[&\|\partial_t\xi_{\boldsymbol{D}}\|^2 + \|M_C^{\frac{1}{2}}\xi_{\boldsymbol{D}}\|^2 + \\
&\epsilon_0\lambda_2\left(\|M_A^{-\frac{1}{2}}\partial_{t^2}\xi_{\boldsymbol{E}}\|^2 + 2\omega_p^2\|M_A^{-\frac{1}{2}}\partial_t\xi_{\boldsymbol{E}}\|^2 + \omega_p^4\|M_A^{-\frac{1}{2}}\xi_{\boldsymbol{E}}\|^2\right) \\
&+ \mu_0\mu\left(\omega_p^2\|\partial_t\xi_H\|^2 + \|\partial_{t^2}\xi_H\|^2\right)\Bigg] - \epsilon_0\lambda_2\left(M_A^{-1}\partial_{t^2}\xi_{\boldsymbol{E}} + \omega_p^2 M_A^{-1}\xi_{\boldsymbol{E}}, \partial_t\xi_{\boldsymbol{D}}\right) \\
&- \left(M_C\partial_t\xi_{\boldsymbol{D}}, \partial_{t^2}\xi_{\boldsymbol{E}}\right) - \omega_p^2\left(M_C\xi_{\boldsymbol{D}}, \partial_t\xi_{\boldsymbol{E}}\right).
\end{aligned} \tag{4.49}$$

Next, we consider the RHS. Using the fact that ξ_{E_x}, ξ_{E_y}, ξ_{D_x}, ξ_{D_y} and ξ_H are in space $P_k(e)$, the property of the projections $(\Pi u)_t = \Pi u_t$, and the definition of the L_2 projection:

$$\int_e (\Pi u - u) v dx = 0 \quad \forall v \in \mathcal{P}_k(e),$$

we conclude that all inner products of η and ξ terms equal to zero. Therefore, we obtain:

$$\begin{aligned} RHS = \sum_{e \in \mathcal{T}_h} \Bigg(\omega_p^2 \Big(& - \int_{\partial e} \partial_t \hat{\eta}_H \partial_t \xi_{E_x}^{(\text{in})} n_y^{(in)} + \int_{\partial e} \partial_t \hat{\eta}_H \partial_t \xi_{E_y}^{(\text{in})} n_x^{(in)} \\ & + \int_{\partial e} (\partial_t \hat{\eta}_{E_y} n_x^{(\text{in})} - \partial_t \hat{\eta}_{E_x} n_y^{(in)}) \partial_t \xi_H^{(in)} \Big) - \int_{\partial e} \partial_{t^2} \hat{\eta}_H \partial_{t^2} \xi_{E_x}^{(\text{in})} n_y^{(in)} \\ & + \int_{\partial e} \partial_{t^2} \hat{\eta}_H \partial_{t^2} \xi_{E_y}^{(\text{in})} n_x^{(in)} + \int_{\partial e} (\partial_{t^2} \hat{\eta}_{E_y} n_x^{(\text{in})} - \partial_{t^2} \hat{\eta}_{E_x} n_y^{(in)}) \partial_{t^2} \xi_H^{(in)} \Bigg). \end{aligned} \tag{4.50}$$

Consider the first term on the RHS, by applying the Cauchy-Schwarz inequality firstly, and using the approximating property of polynomial preserving operators (Theorem 3.4.1 in [12]) on the η_H and the standard inverse inequality [12] on the ξ_{E_x}, we have

$$\begin{aligned} \sum_{e \in \mathcal{T}_h} \int_{\partial e} \partial_t \hat{\eta}_H \partial_t \xi_{E_x}^{(\text{in})} n_y^{(in)} & \leq \sum_{e \in \mathcal{T}} \frac{1}{\delta h} \int_{\partial e} \left| \partial_t \hat{\eta}_H \right|^2 + \delta h \int_{\partial e} \left| \partial_t \xi_{E_x}^{(\text{in})} \right|^2 \\ & \leq C \sum_{e \in \mathcal{T}_h} \left(\frac{1}{\delta} \|\partial_t \eta_H\|_{L^\infty(e)}^2 + \delta h^2 \|\partial_t \xi_{E_x}\|_{L^\infty(e)}^2 \right) \\ & \leq C h^{2k} \|\partial_t H\|_{H^{k+1}(\Omega)}^2 + \mathrm{C} \|\partial_t \xi_{E_x}\|_{L^2(\Omega)}^2, \end{aligned} \tag{4.51}$$

with any $\delta > 0$. Note that the constant C may have different values in each term, but is independent of the mesh size h.

Using the same arguments on the remaining terms, we obtain the following inequality:

$$
\begin{aligned}
&\frac{1}{2}\frac{d}{dt}\Bigg[\|\partial_t\xi_{\boldsymbol{D}}\|^2 + \|M_C^{\frac{1}{2}}\xi_{\boldsymbol{D}}\|^2 + \epsilon_0\lambda_2\Big(\|M_A^{-\frac{1}{2}}\partial_{t^2}\xi_{\boldsymbol{E}}\|^2 + 2\omega_p^2\|M_A^{-\frac{1}{2}}\partial_t\xi_{\boldsymbol{E}}\|^2 \\
&\qquad + \omega_p^4\|M_A^{-\frac{1}{2}}\xi_{\boldsymbol{E}}\|^2\Big) + \mu_0\mu\left(\omega_p^2\|\partial_t\xi_H\|^2 + \|\partial_{t^2}\xi_H\|^2\right)\Bigg] \\
&\leq Ch^{2k}\Big(\|\partial_t\boldsymbol{E}\|_{H^{k+1}}^2 + \|\partial_{t^2}\boldsymbol{E}\|_{H^{k+1}}^2 + \|\partial_t H\|_{H^{k+1}}^2 + \|\partial_{t^2}H\|_{H^{k+1}}^2\Big) \\
&\qquad + C\Big(\|\partial_t\xi_{\boldsymbol{E}}\|^2 + \|\partial_{t^2}\xi_{\boldsymbol{E}}\|^2 + \|\partial_t\xi_H\|^2 + \|\partial_{t^2}\xi_H\|^2\Big) \\
&\qquad + \epsilon_0\lambda_2\left(M_A^{-1}\partial_{t^2}\xi_{\boldsymbol{E}} + \omega_p^2 M_A^{-1}\xi_{\boldsymbol{E}}, \partial_t\xi_{\boldsymbol{D}}\right) + \left(M_C\partial_t\xi_{\boldsymbol{D}}, \partial_{t^2}\xi_{\boldsymbol{E}}\right) \\
&\qquad + \omega_p^2\left(M_C\xi_{\boldsymbol{D}}, \partial_t\xi_{\boldsymbol{E}}\right).
\end{aligned}
\tag{4.52}
$$

Finally, applying the Cauchy-Schwarz inequality, and then using the Gronwall inequality, the error estimates of the L_2 projections, and the triangle inequality, we can conclude the proof. ■

4.3.2.2 The error analysis on rectangular meshes

In general, using the flux and boundary condition (4.14)-(4.19), the stability and error analysis on tensor-product rectangular meshes are the same as those on triangular meshes. However, inspired by [53], if we modify the fluxes at the PEC boundary by adding suitable jump terms, and by using tensor-product DG spaces, the optimal error accuracy can be proved mathematically, and can be observed in the numerical tests.

We firstly define the rectangular mesh. For simplicity, we consider a rectangular domain $\Omega = [a_x, b_x] \times [a_y, b_y]$, which is discretized by the cells $I_{ij} = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}] := I_i \times J_j$ for $1 \leq i \leq N_x$ and $1 \leq j \leq N_y$. The mesh sizes are defined as $h_i^x = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}$ and $h_j^y = y_{j+\frac{1}{2}} - y_{j-\frac{1}{2}}$, with $h^x = \max_{1\le i\le N_x} h_i^x$, $h^y = \max_{1\le j\le N_y} h_j^y$, and $h = \max(h^x, h^y)$. The finite element space V_h^k is chosen as

$$V_h^k = \{v : v|_{I_{ij}} \in Q^k(I_{ij})\},$$

where $Q^k(I_{ij})$ is the space of the tensor products of one dimensional polynomials with degree at most k over the cell I_{ij}. For simplicity, let $u_h(x^+_{i+\frac{1}{2}}, y)$ (or $u_h^+(x_{i+\frac{1}{2}}, y)$ or $(u_h)^+_{i+\frac{1}{2},y}$) and $u_h(x^-_{i+\frac{1}{2}}, y)$ (or $u_h^-(x_{i+\frac{1}{2}}, y)$ or $(u_h)^-_{i+\frac{1}{2},y}$) denote the limit value of u_h at $x_{i+\frac{1}{2}}$ from the right cell $I_{i+1,j}$, and from the left cell $I_{i,j}$ respectively. $u_h(x, y^+_{j+\frac{1}{2}})$ (or $u_h^+(x, y_{j+\frac{1}{2}})$ or $(u_h)^+_{x,j+\frac{1}{2}}$), and $u_h(x, y^-_{j+\frac{1}{2}})$ (or $u_h^-(x, y_{j+\frac{1}{2}})$ or $(u_h)^-_{x,j+\frac{1}{2}}$) are defined similarly. By setting the fixed direction $\boldsymbol{\beta} = (1, 1)$, the alternating fluxes become:

$$\hat{E}_{xh}(x, y_{j+\frac{1}{2}}) = E^+_{xh}(x, y_{j+\frac{1}{2}}), \tag{4.53}$$

$$\hat{E}_{yh}(x_{i+\frac{1}{2}}, y) = E^+_{yh}(x_{i+\frac{1}{2}}, y), \tag{4.54}$$

$$\hat{H}_h(x, y_{j+\frac{1}{2}}) = H^-_h(x, y_{j+\frac{1}{2}}), \tag{4.55}$$

$$\hat{H}_h(x_{i+\frac{1}{2}}, y) = H^-_h(x_{i+\frac{1}{2}}, y). \tag{4.56}$$

To achieve the optimal convergence, instead of letting the fluxes $\hat{H}_h(x, y_{\frac{1}{2}}) = H^+(x, y_{\frac{1}{2}})$ and $\hat{H}_h(x_{\frac{1}{2}}, y) = H^+(x_{\frac{1}{2}}, y)$ as in (4.19), we apply the PEC boundary condition as stated below:

$$\hat{E}_{xh}(x, y_{\frac{1}{2}}) = 0, \tag{4.57}$$

$$\hat{E}_{yh}(x_{\frac{1}{2}}, y) = 0, \tag{4.58}$$

$$\hat{H}_h(x, y_{\frac{1}{2}}) = H_h^+(x, y_{\frac{1}{2}}) + c_0 \left[\!\!\left[E_{xh}(x, y_{\frac{1}{2}}) \right]\!\!\right], \tag{4.59}$$

$$\hat{H}_h(x_{\frac{1}{2}}, y) = H_h^+(x_{\frac{1}{2}}, y) - c_0 \left[\!\!\left[E_{yh}(x_{\frac{1}{2}}, y) \right]\!\!\right]. \tag{4.60}$$

The constant c_0 is independent of the mesh size h, and in the following numerical tests, c_0 is chosen as $\frac{1}{2}$. The jump cross the cell boundaries is denoted as $[\![u]\!] = u^+ - u^-$. Here $[\![E_{xh}(x, y_{\frac{1}{2}})]\!] = E_{xh}^+(x, y_{\frac{1}{2}}) - 0$, and $[\![E_{yh}(x_{\frac{1}{2}}, y)]\!] = E_{yh}^+(x_{\frac{1}{2}}, y) - 0$.

Using the fluxes and boundary conditions (4.53)-(4.60), and following the same argument as in Section 3.1, we can verify the stability of the method. For the error analysis, we have the following theorem.

Theorem 4.3.3. Suppose that the analytical solutions (E_x, E_y, D_x, D_y, H) of (4.1)-(4.4) are smooth enough, and $(E_{xh}, E_{yh}, D_{xh}, D_{yh}, H_h)$ are the corresponding numerical solutions of (4.10)-(4.13) on the rectangular mesh. With the alternating flux (4.53)-(4.56) and the PEC boundary condition (4.57)-(4.60), we have the following error estimate:

$$\begin{aligned}
&\Bigg[\|\partial_t \boldsymbol{D} - \partial_t \boldsymbol{D}_h\|^2 + \|M_C^{\frac{1}{2}}(\boldsymbol{D} - \boldsymbol{D}_h)\|^2 + \epsilon_0\lambda_2\Big(\|M_A^{-\frac{1}{2}}(\partial_{t^2}\boldsymbol{E} - \partial_{t^2}\boldsymbol{E}_h)\|^2 \\
&\qquad + 2\omega_p^2\|M_A^{-\frac{1}{2}}(\partial_t\boldsymbol{E} - \partial_t\boldsymbol{E}_h)\|^2 + \omega_p^4\|M_A^{-\frac{1}{2}}(\boldsymbol{E} - \boldsymbol{E}_h)\|^2\Big) \\
&\qquad + \mu_0\mu\left(\omega_p^2\|\partial_t H - \partial_t H_h\|^2 + \|\partial_{t^2} H - \partial_{t^2} H_h\|^2\right)\Bigg](t) \\
&\leq Ch^{2k+2} + \Bigg[\|\partial_t \boldsymbol{D} - \partial_t \boldsymbol{D}_h\|^2 + \|M_C^{\frac{1}{2}}(\boldsymbol{D} - \boldsymbol{D}_h)\|^2 + \\
&\qquad \epsilon_0\lambda_2\Big(\|M_A^{-\frac{1}{2}}(\partial_{t^2}\boldsymbol{E} - \partial_{t^2}\boldsymbol{E}_h)\|^2 + 2\omega_p^2\|M_A^{-\frac{1}{2}}(\partial_t\boldsymbol{E} - \partial_t\boldsymbol{E}_h)\|^2 + \\
&\omega_p^4\|M_A^{-\frac{1}{2}}(\boldsymbol{E} - \boldsymbol{E}_h)\|^2\Big) + \mu_0\mu\left(\omega_p^2\|\partial_t H - \partial_t H_h\|^2 + \|\partial_{t^2} H - \partial_{t^2} H_h\|^2\right)\Bigg](0).
\end{aligned} \tag{4.61}$$

Here $k \geq 1$ is the order of the basis function V_h^k, and C is a positive constant independent of the mesh size h.

Proof. To prove the theorem, we firstly need to define some new projections [53]. The $1D$ projections in the x direction

$$P_x^{\pm} : H^1(I_i) \rightarrow \mathcal{P}_k(I_i)$$

are defined as the functions in the k-th degree polynomial space that satisfy

$$\int_{I_i} (P_x^+ u - u) v dx = 0 \quad \forall v \in \mathcal{P}_{k-1}(I_i), \qquad \text{and} \quad P_x^+ u(x_{i-\frac{1}{2}}^+) = u(x_{i-\frac{1}{2}}^+), \tag{4.62}$$

$$\int_{I_i} (P_x^- u - u) v dx = 0 \quad \forall v \in \mathcal{P}_{k-1}(I_i), \qquad \text{and} \quad P_x^- u(x_{i+\frac{1}{2}}^-) = u(x_{i+\frac{1}{2}}^-). \tag{4.63}$$

The $1D$ projections in the y direction $P_y^{\pm}$ are defined in the same way. Besides, the standard L_2 projections in the x and y directions are denoted as

$$P_x : H^1(I_i) \rightarrow \mathcal{P}_k(I_i), \qquad P_y : H^1(J_j) \rightarrow \mathcal{P}_k(J_j).$$

Next, we use the tensor products of the $1D$ projections to define the $2D$ projections in cell I_{ij}. In particular, we define the projection

$$\Pi_1 = P_x \otimes P_y^+ : H^2(I_{ij}) \rightarrow Q_k(I_{ij}),$$

which satisfies that: For any $u \in H^2(I_{ij})$ and any test function $\phi \in Q_k(I_{ij})$:

$$\int_{I_{ij}} \Pi_1 u(x,y) \frac{\partial \phi(x,y)}{\partial y} dxdy = \int_{I_{ij}} u(x,y) \frac{\partial \phi(x,y)}{\partial y} dxdy, \tag{4.64}$$

$$\int_{I_i} \Pi_1 u\left(x, y^+_{j-\frac{1}{2}}\right) \phi\left(x, y^+_{j-\frac{1}{2}}\right) dx = \int_{I_i} u\left(x, y^+_{j-\frac{1}{2}}\right) \phi\left(x, y^+_{j-\frac{1}{2}}\right) dx. \tag{4.65}$$

The projection

$$\Pi_2 = P_x^+ \otimes P_y : H^2(I_{ij}) \rightarrow Q_k(I_{ij}),$$

which satisfies that: For any $u \in H^2(I_{ij})$ and any $\phi \in Q_k(I_{ij})$:

$$\int_{I_{ij}} \Pi_2 u(x,y) \frac{\partial \phi(x,y)}{\partial x} dxdy = \int_{I_{ij}} u(x,y) \frac{\partial \phi(x,y)}{\partial x} dxdy, \tag{4.66}$$

$$\int_{J_j} \Pi_2 u\left(x^+_{i-\frac{1}{2}}, y\right) \phi\left(x^+_{i-\frac{1}{2}}, y\right) dy = \int_{J_j} u\left(x^+_{i-\frac{1}{2}}, y\right) \phi\left(x^+_{i-\frac{1}{2}}, y\right) dy. \tag{4.67}$$

The projection

$$\Pi_3 = P_x^- \otimes P_y^- : H^2(I_{ij}) \to Q_k(I_{ij}),$$

which satisfies that: For any $u \in H^2(I_{ij})$ and any $\phi \in Q_{k-1}(I_{ij})$:

$$\int_{I_{ij}} \Pi_3 u(x,y)\phi(x,y)dxdy = \int_{I_{ij}} u(x,y)\phi(x,y)dxdy, \tag{4.68}$$

$$\int_{I_i} \Pi_3 u\left(x, y^-_{j+\frac{1}{2}}\right)\phi\left(x, y^-_{j+\frac{1}{2}}\right)dx = \int_{I_i} u\left(x, y^-_{j+\frac{1}{2}}\right)\phi\left(x, y^-_{j+\frac{1}{2}}\right)dx, \tag{4.69}$$

$$\int_{J_j} \Pi_3 u\left(x^-_{i+\frac{1}{2}}, y\right)\phi\left(x^-_{i+\frac{1}{2}}, y\right)dy = \int_{J_j} u\left(x^-_{i+\frac{1}{2}}, y\right)\phi\left(x^-_{i+\frac{1}{2}}, y\right)dy, \tag{4.70}$$

$$\Pi_3 u\left(x^-_{i+\frac{1}{2}}, y^-_{j+\frac{1}{2}}\right) = u\left(x^-_{i+\frac{1}{2}}, y^-_{j+\frac{1}{2}}\right). \tag{4.71}$$

Finally, the usual 2D L_2 projection is denoted as

$$\Pi_4 = P_x \otimes P_y : H^2(I_{ij}) \to Q_k(I_{ij}).$$

The good properties of the projections including the uniqueness and the optimal error estimate can be found in Lemmas 3.1-3.3 in [53].

The errors between the exact solutions and the numerical solutions can be decomposed by using the above projections:

$$\mathcal{E}_{E_x} = E_x - E_{xh} = (\Pi_1 E_x - E_{xh}) - (\Pi_1 E_x - E_x) := \ \xi_{E_x} - \eta_{E_x},$$

$$\mathcal{E}_{E_y} = E_y - E_{yh} = (\Pi_2 E_y - E_{yh}) - (\Pi_2 E_y - E_y) := \ \xi_{E_y} - \eta_{E_y},$$

$$\mathcal{E}_{D_x} = D_x - D_{xh} = (\Pi_4 D_x - D_{xh}) - (\Pi_4 D_x - D_x) := \ \xi_{D_x} - \eta_{D_x},$$

$$\mathcal{E}_{D_y} = D_y - D_{yh} = (\Pi_4 D_y - D_{yh}) - (\Pi_4 D_y - D_y) := \ \xi_{D_y} - \eta_{D_y},$$

$$\mathcal{E}_H \ = \ H - H_h \ = \ (\Pi_3 H - H_h) - (\Pi_3 H - H) \ := \ \xi_H - \eta_H,$$

Then, following the exact steps in Sect. 4.3.2.1, and using definitions of the projections (4.62)-(4.71) and the property $(\Pi u)_t = \Pi u_t$, we obtain the equation of the errors:

$$\begin{aligned}
&\frac{1}{2}\frac{d}{dt}\bigg[\|\partial_t \xi_{\boldsymbol{D}}\|^2 + \|M_C^{\frac{1}{2}} \xi_{\boldsymbol{D}}\|^2 + \epsilon_0 \lambda_2 \Big(\|M_A^{-\frac{1}{2}} \partial_{t^2} \xi_{\boldsymbol{E}}\|^2 + 2\omega_p^2 \|M_A^{-\frac{1}{2}} \partial_t \xi_{\boldsymbol{E}}\|^2 \\
&\qquad + \omega_p^4 \|M_A^{-\frac{1}{2}} \xi_{\boldsymbol{E}}\|^2\Big) + \mu_0 \mu \left(\omega_p^2 \|\partial_t \xi_H\|^2 + \|\partial_{t^2} \xi_H\|^2\right)\bigg] \\
&\leq GD + \omega_p^2 \left(\sum_{j=1}^{N_y} \mathrm{TEX}_j(\partial_t \eta_H, \partial_t \xi_{E_x}) + \sum_{i=1}^{N_x} \mathrm{TEY}_i(\partial_t \eta_H, \partial_t \xi_{E_y})\right) \\
&\qquad + \sum_{j=1}^{N_y} \mathrm{TEX}_j(\partial_{t^2} \eta_H, \partial_{t^2} \xi_{E_x}) + \sum_{i=1}^{Nx} \mathrm{TEY}_i(\partial_{t^2} \eta_H, \partial_{t^2} \xi_{E_y}) \\
&\qquad - \omega_p^2 \left(\sum_{i=1}^{N_x} c_0 \int_{I_i} (\partial_t \xi_{E_x}^+(x, y_{\frac{1}{2}}))^2 + \sum_{j=1}^{N_y} c_0 \int_{I_i} (\partial_t \xi_{E_y}^+(x_{\frac{1}{2}}, y))^2\right) \\
&\qquad - \left(\sum_{i=1}^{N_x} c_0 \int_{I_i} (\partial_{t^2} \xi_{E_x}^+(x, y_{\frac{1}{2}}))^2 + \sum_{j=1}^{N_y} c_0 \int_{I_i} (\partial_{t^2} \xi_{E_y}^+(x_{\frac{1}{2}}, y))^2\right).
\end{aligned} \tag{4.72}$$

The GD, which contains all good terms, is defined as:

$$\begin{aligned}
GD = &-\varepsilon_0 \lambda_2 \left(M_A^{-1} \partial_{t^2} \eta_{\boldsymbol{E}}, \partial_t \xi_{\boldsymbol{D}}\right) + \varepsilon_0 \lambda_2 \omega_p^2 \left(M_A^{-1} \eta_{\boldsymbol{E}}, \partial_t \xi_{\boldsymbol{D}}\right) \\
&+ \varepsilon_0 \lambda_2 \omega_p^2 \left(M_A^{-1} \partial_{t^2} \eta_{\boldsymbol{E}}, \partial_t \xi_{\boldsymbol{E}}\right) + \varepsilon_0 \lambda_2 \omega_p^4 \left(M_A^{-1} \eta_{\boldsymbol{E}}, \partial_t \xi_{\boldsymbol{E}}\right) \\
&+ \varepsilon_0 \lambda_2 \left(M_A^{-1} \partial_{t^3} \eta_{\boldsymbol{E}}, \partial_{t^2} \xi_{\boldsymbol{E}}\right) + \varepsilon_0 \lambda_2 \omega_p^2 \left(M_A^{-1} \partial_t \eta_{\boldsymbol{E}}, \partial_{t^2} \xi_{\boldsymbol{E}}\right) \\
&+ \mu_0 \mu \omega_p^2 (\partial_{t^2} \eta_H, \partial_t \xi_H) + \mu_0 \mu (\partial_{t^3} \eta_H, \partial_{t^2} \xi_H) \\
&+ \epsilon_0 \lambda_2 \left(M_A^{-1} \partial_{t^2} \xi_{\boldsymbol{E}} + \omega_p^2 M_A^{-1} \xi_{\boldsymbol{E}}, \partial_t \xi_{\boldsymbol{D}}\right) \\
&+ \left(M_C \partial_t \xi_{\boldsymbol{D}}, \partial_{t^2} \xi_{\boldsymbol{E}}\right) + \omega_p^2 \left(M_C \xi_{\boldsymbol{D}}, \partial_t \xi_{\boldsymbol{E}}\right),
\end{aligned} \tag{4.73}$$

and the terms $\text{TEX}_j(\eta_H, \xi_{E_x})$ and $\text{TEX}_i(\eta_H, \xi_{E_y})$ are defined as

$$\text{TEX}_j = \sum_{i=1}^{N_x}\Big(-\int_{I_i}\big(\hat{\eta}_H\xi_{E_x}^{-}(x, y_{j+\frac{1}{2}}) - \hat{\eta}_H\xi_{E_x}^{+}(x, y_{j-\frac{1}{2}})\big) + \int_{I_{ij}} \eta_H \partial_y \xi_{E_x}\Big), \tag{4.74}$$

$$\text{TEY}_i = \sum_{j=1}^{N_y}\Big(\int_{I_i}\big(\hat{\eta}_H\xi_{E_y}^{-}(x_{i+\frac{1}{2}}, y) - \hat{\eta}_H\xi_{E_y}^{+}(x_{i-\frac{1}{2}}, y)\big) - \int_{I_{ij}} \eta_H \partial_x \xi_{E_y}\Big). \tag{4.75}$$

By using the following inequalities in [53, Lemmas 3.3-3.4]:

$$\begin{aligned}
&\sum_{j=2}^{N_y} \text{TEX}_j(\eta_H, \xi_{E_x}) \leq Ch^{2k+2} + \|\xi_{E_x}\|^2,\\
&\sum_{i=2}^{N_x} \text{TEY}_i(\eta_H, \xi_{E_y}) \leq Ch^{2k+2} + \|\xi_{E_y}\|^2,\\
&\text{TEX}_1(\eta_H, \xi_{E_x}) - \sum_{i=1}^{N_x} c_0 \int_{I_i} (\xi_{E_x}^{+}(x, y_{\frac{1}{2}}))^2 \leq Ch^{2k+2} + \|\xi_{E_x}\|^2,\\
&\text{TEY}_1(\eta_H, \xi_{E_y}) - \sum_{j=1}^{N_y} c_0 \int_{I_j} (\xi_{E_y}^{+}(x_{\frac{1}{2}}, y))^2 \leq Ch^{2k+2} + \|\xi_{E_y}\|^2,
\end{aligned} \tag{4.76}$$

we have

$$RHS \leq GD + \omega_p^2(Ch^{2k+2} + C\|\partial_t \xi_{\boldsymbol{E}}\|^2) + (Ch^{2k+2} + C\|\partial_{t^2} \xi_{\boldsymbol{E}}\|^2). \tag{4.77}$$

Applying the Cauchy-Schwarz inequality on the good terms, and using the optimal error estimate of the projections (see Lemma 3.3 in [74]), and finally applying the Gronwall inequality and the triangle inequality, we conclude the proof. ■

Chapter Five

The fully-discrete LDG methods for the carpet cloak model

5.1 The fully-discrete DG method

In this section, we propose the fully discrete leap-frog DG method to solve the carpet cloak model on unstructured meshes. The stability analysis will be given and the numerical results of the carpet cloak simulations will be provided. Before we define the fully-discrete scheme, we introduce the following central difference operators in time: For any time sequence function u^n,

$$\delta_\tau u^{n+\frac{1}{2}} := \frac{u^{n+1} - u^n}{\tau}, \ \ \delta_\tau^2 u^n := \frac{\delta_\tau u^{n+\frac{1}{2}} - \delta_\tau u^{n-\frac{1}{2}}}{\tau} = \frac{u^{n+1} - 2u^n + u^{n-1}}{\tau^2}.$$

The averaging operators are defined as:

$$\overline{u}^n = \frac{u^{n+1} + u^{n-1}}{2}, \ \ \breve{u}^n = \frac{u^{n+\frac{1}{2}} + u^{n-\frac{1}{2}}}{2}.$$

Moreover, we need the following discrete Gronwall inequality to prove the discrete stability:

Lemma 5.1.1. [68, Lemma 4.1.2] Assume that the sequence u_n satisfies

$$u_0 \leq g_0, \quad and \quad u_0 \leq g_0 + r\tau \sum_{s=0}^{n-1} u_s, \ \forall n \geq 1,$$

where g_0, r and τ are some positive constants. Then we have

$$u_n \leq g_0 \cdot (1 + r\tau)^n \leq g_0 \cdot \exp(rn\tau), \ \forall n \geq 1.$$

Now we consider the following leap-frog LDG scheme: For any $n \geq 0$, find

$D_{xh}^{n+1}, D_{yh}^{n+1}, H_h^{n+\frac{1}{2}}, E_{xh}^{n+1}, E_{yh}^{n+1} \in V_h^k$ such that

$$\int_e \delta_\tau D_{xh}^{n+\frac{1}{2}}\phi_x + \int_e H_h^{n+\frac{1}{2}}\partial_y\phi_x - \int_{\partial e}\hat{H}_h^{n+\frac{1}{2}}\phi_x^{(in)}n_y^{(in)} = 0, \tag{5.1}$$

$$\int_e \delta_\tau D_{yh}^{n+\frac{1}{2}}\phi_x - \int_e H_h^{n+\frac{1}{2}}\partial_x\phi_y + \int_{\partial e}\hat{H}_h^{n+\frac{1}{2}}\phi_y^{(in)}n_x^{(in)} = 0, \tag{5.2}$$

$$\varepsilon_0\lambda_2\int_e\left(M_A^{-1}\delta_\tau^2\boldsymbol{E}_h^n + \omega_p^2 M_A^{-1}\overline{\boldsymbol{E}}_h^n\right)\cdot\boldsymbol{u} = \int_e\left(\delta_\tau^2\boldsymbol{D}_h^n + M_C\overline{\boldsymbol{D}}_h^n\right)\cdot\boldsymbol{u}, \tag{5.3}$$

$$\mu_0\mu\int_e \delta_\tau H_h^n\psi - \int_e E_{yh}^n\partial_x\psi + \int_e E_{xh}^n\partial_y\psi + \int_{\partial e}\left(\hat{E}_{yh}^n n_x^{(in)} - \hat{E}_{xh}^n n_y^{(in)}\right)\psi = 0, \tag{5.4}$$

for all test functions $\phi_x, \phi_y, \boldsymbol{u}, \psi \in V_h^k$, with the following fluxes consistent with (4.14)-(4.19):

$$\hat{E}_{xh}^n = E_{xh}^{n,R} \tag{5.5}$$

$$\hat{E}_{yh}^n = E_{yh}^{n,R} \tag{5.6}$$

$$\hat{H}_h^{n+\frac{1}{2}} = H_h^{n+\frac{1}{2},L} \tag{5.7}$$

$$\hat{E}_{xh}^n = 0, \text{ on } y = c, d \tag{5.8}$$

$$\hat{E}_{yh}^n = 0, \text{ on } x = a, b \tag{5.9}$$

$$\hat{H}_h^{n+\frac{1}{2}} = H_h^{n+\frac{1}{2},(in)}, \text{ on } \partial\Omega. \tag{5.10}$$

With the above preparation, we can now prove the following energy identity, which is really the discrete form of the energy identity (4.21).

Theorem 5.1.2. For the solution $(\boldsymbol{D}_h^{n+1}, H_h^{n+\frac{1}{2}}, \boldsymbol{E}_h^{n+1})$ of the leap-frog LDG scheme

(5.1)-(5.4), we define the discrete energy at time level m:

$$
\begin{aligned}
ENG_{lf}(m) &= \|\delta_\tau \boldsymbol{D}_h^{m+\frac{1}{2}}\|^2 + \frac{1}{2}\Big(\|M_C^{\frac{1}{2}}\boldsymbol{D}_h^{m+1}\|^2 + \|M_C^{\frac{1}{2}}\boldsymbol{D}_h^{m}\|^2\Big) \\
&+ \varepsilon_0\lambda_2\|M_A^{-\frac{1}{2}}\delta_\tau^2\boldsymbol{E}_h^{m+1}\|^2 + \frac{\varepsilon_0\lambda_2\omega_p^2}{2}\Big(3\|M_A^{-\frac{1}{2}}\delta_\tau\boldsymbol{E}_h^{m+\frac{1}{2}}\|^2 + \|M_A^{-\frac{1}{2}}\delta_\tau\boldsymbol{E}_h^{m-\frac{1}{2}}\|^2\Big) \\
&+ \frac{\varepsilon_0\lambda_2\omega_p^4}{2}\Big(\|M_A^{-\frac{1}{2}}\boldsymbol{E}_h^{m+1}\|^2 + \|M_A^{-\frac{1}{2}}\boldsymbol{E}_h^{m}\|^2\Big) \\
&+ \mu_0\mu\left[\omega_p^2(\delta_\tau H_h^{m+1}, \delta_\tau H_h^{m}) + (\delta_\tau^2 H_h^{m+\frac{1}{2}}, \delta_\tau^2 H_h^{m-\frac{1}{2}})\right].
\end{aligned}
\tag{5.11}
$$

Suppose the time step satisfies the constraint:

$$
\tau \leq \min\left\{ \frac{1}{\sqrt{\varepsilon_0\lambda_2}\|M_A^{-\frac{1}{2}}\| + \frac{\|M_A^{-\frac{1}{2}}M_B\|}{2\sqrt{\varepsilon_0\lambda_2}}}, \quad \frac{\sqrt{\varepsilon_0\lambda_2}}{\omega_p\|M_C^{\frac{1}{2}}M_A^{\frac{1}{2}}\|} \right\}, \tag{5.12}
$$

then we have

$$
ENG_{lf}(m) \leq C \cdot ENG_{lf}(0) \cdot \exp(cm\tau), \qquad \forall m \geq 1, \tag{5.13}
$$

where C and c are positive constants independent of mesh size h and time step τ.

Proof. To make the proof easy to follow, we divide it into several major parts.

(I) Choosing $\boldsymbol{u} = \frac{\tau}{2}(\delta_\tau\boldsymbol{D}_h^{n+\frac{1}{2}} + \delta_\tau\boldsymbol{D}_h^{n-\frac{1}{2}}) = \tau\delta_\tau\breve{\boldsymbol{D}}_h^n$ in (5.3), and using the identity

$$
\begin{aligned}
&\Big(M_C\overline{\boldsymbol{D}}_h^n, \frac{\tau}{2}(\delta_\tau\boldsymbol{D}_h^{n+\frac{1}{2}} + \delta_\tau\boldsymbol{D}_h^{n-\frac{1}{2}})\Big) \\
&= \Big(M_C\frac{\boldsymbol{D}_h^{n+1} + \boldsymbol{D}_h^{n-1}}{2}, \frac{(\boldsymbol{D}_h^{n+1} - \boldsymbol{D}_h^{n}) + (\boldsymbol{D}_h^{n} - \boldsymbol{D}_h^{n-1})}{2}\Big) \\
&= \frac{1}{4}(\|M_C^{\frac{1}{2}}\boldsymbol{D}_h^{n+1}\|^2 - \|M_C^{\frac{1}{2}}\boldsymbol{D}_h^{n-1}\|^2),
\end{aligned}
\tag{5.14}
$$

we have

$$\begin{aligned}
&\frac{1}{2}(\|\delta_\tau \boldsymbol{D}_h^{n+\frac{1}{2}}\|^2 - \|\delta_\tau \boldsymbol{D}_h^{n-\frac{1}{2}}\|^2) + \frac{1}{4}(\|M_C^{\frac{1}{2}}\boldsymbol{D}_h^{n+1}\|^2 - \|M_C^{\frac{1}{2}}\boldsymbol{D}_h^{n-1}\|^2)\\
&= \tau\varepsilon_0\lambda_2\left[(M_A^{-1}\delta_\tau^2\boldsymbol{E}_h^n, \delta_\tau\breve{\boldsymbol{D}}_h^n) + \omega_p^2(M_A^{-1}\overline{\boldsymbol{E}}_h^n, \delta_\tau\breve{\boldsymbol{D}}_h^n)\right].
\end{aligned}\tag{5.15}$$

(II) Choosing $\boldsymbol{u} = \frac{\tau}{2}(\delta_\tau\boldsymbol{E}_h^{n+\frac{1}{2}} + \delta_\tau\boldsymbol{E}_h^{n-\frac{1}{2}}) = \tau\delta_\tau\breve{\boldsymbol{E}}_h^n$ in (5.3), and using the identity

$$\begin{aligned}
&\left(M_A^{-1}\overline{\boldsymbol{E}}_h^n, \frac{\tau}{2}(\delta_\tau\boldsymbol{E}_h^{n+\frac{1}{2}} + \delta_\tau\boldsymbol{E}_h^{n-\frac{1}{2}})\right)\\
&= \left(M_A^{-1}\frac{\boldsymbol{E}_h^{n+1} + \boldsymbol{E}_h^{n-1}}{2}, \frac{(\boldsymbol{E}_h^{n+1} - \boldsymbol{E}_h^n) + (\boldsymbol{E}_h^n - \boldsymbol{E}_h^{n-1})}{2}\right)\\
&= \frac{1}{4}(\|M_A^{-\frac{1}{2}}\boldsymbol{E}_h^{n+1}\|^2 - \|M_A^{-\frac{1}{2}}\boldsymbol{E}_h^{n-1}\|^2),
\end{aligned}\tag{5.16}$$

we obtain

$$\begin{aligned}
&\frac{\varepsilon_0\lambda_2}{2}(\|M_A^{-\frac{1}{2}}\delta_\tau\boldsymbol{E}_h^{n+\frac{1}{2}}\|^2 - \|M_A^{-\frac{1}{2}}\delta_\tau\boldsymbol{E}_h^{n-\frac{1}{2}}\|^2)\\
&\qquad + \frac{\varepsilon_0\lambda_2\omega_p^2}{4}(\|M_A^{-\frac{1}{2}}\boldsymbol{E}_h^{n+1}\|^2 - \|M_A^{-\frac{1}{2}}\boldsymbol{E}_h^{n-1}\|^2)\\
&= \tau\left(\delta_\tau^2\boldsymbol{D}_h^n + M_C\overline{\boldsymbol{D}}_h^n, \delta_\tau\breve{\boldsymbol{E}}_h^n\right).
\end{aligned}\tag{5.17}$$

Using (5.3) to subtract themselves with n replaced by $n-1$, then choosing $\boldsymbol{u} = \frac{1}{2}(\delta_\tau^2\boldsymbol{E}_h^n + \delta_\tau^2\boldsymbol{E}_h^{n-1}) = \delta_\tau^2\breve{\boldsymbol{E}}_h^{n-\frac{1}{2}}$, and using the identity

$$\begin{aligned}
&\left(M_A^{-1}(\overline{\boldsymbol{E}}_h^n - \overline{\boldsymbol{E}}_h^{n-1}), \frac{1}{2}(\delta_\tau^2\boldsymbol{E}_h^n + \delta_\tau^2\boldsymbol{E}_h^{n-1})\right)\\
&= \left(M_A^{-1}\frac{(\boldsymbol{E}_h^{n+1} + \boldsymbol{E}_h^{n-1}) - (\boldsymbol{E}_h^n + \boldsymbol{E}_h^{n-2})}{2},\right.\\
&\qquad\qquad \left.\frac{\delta_\tau\boldsymbol{E}_h^{n+\frac{1}{2}} - \delta_\tau\boldsymbol{E}_h^{n-\frac{1}{2}}}{2\tau} + \frac{\delta_\tau\boldsymbol{E}_h^{n+\frac{1}{2}} - \delta_\tau\boldsymbol{E}_h^{n-\frac{3}{2}}}{2\tau}\right)\\
&= \frac{1}{4}\left(M_A^{-1}(\delta_\tau\boldsymbol{E}_h^{n+\frac{1}{2}} + \delta_\tau\boldsymbol{E}_h^{n-\frac{3}{2}}), \delta_\tau\boldsymbol{E}_h^{n+\frac{1}{2}} - \delta_\tau\boldsymbol{E}_h^{n-\frac{3}{2}}\right)\\
&= \frac{1}{4}(\|M_A^{-\frac{1}{2}}\delta_\tau\boldsymbol{E}_h^{n+\frac{1}{2}}\|^2 - \|M_A^{-\frac{1}{2}}\delta_\tau\boldsymbol{E}_h^{n-\frac{3}{2}}\|^2),
\end{aligned}\tag{5.18}$$

we obtain

$$\begin{aligned}
&\frac{\varepsilon_0\lambda_2}{2}(\|M_A^{-\frac{1}{2}}\delta_\tau^2\boldsymbol{E}_h^n\|^2 - \|M_A^{-\frac{1}{2}}\delta_\tau^2\boldsymbol{E}_h^{n-1}\|^2) \\
&\quad + \frac{\varepsilon_0\lambda_2\omega_p^2}{4}(\|M_A^{-\frac{1}{2}}\delta_\tau\boldsymbol{E}_h^{n+\frac{1}{2}}\|^2 - \|M_A^{-\frac{1}{2}}\delta_\tau\boldsymbol{E}_h^{n-\frac{3}{2}}\|^2) \\
&= \tau\left(\delta_\tau^3\boldsymbol{D}_h^{n-\frac{1}{2}} + M_C\delta_\tau\overline{\boldsymbol{D}}_h^{n-\frac{1}{2}}, \delta_\tau^2\breve{\boldsymbol{E}}_h^{n-\frac{1}{2}}\right).
\end{aligned} \tag{5.19}$$

(III) Multiplying (5.17) by ω_p^2, and adding the result together with (5.15) and (5.19), we have

$$\begin{aligned}
&\frac{1}{2}(\|\delta_\tau\boldsymbol{D}_h^{n+\frac{1}{2}}\|^2 - \|\delta_\tau\boldsymbol{D}_h^{n-\frac{1}{2}}\|^2) + \frac{1}{4}(\|M_C^{\frac{1}{2}}\boldsymbol{D}_h^{n+1}\|^2 - \|M_C^{\frac{1}{2}}\boldsymbol{D}_h^{n-1}\|^2) \\
&\quad + \frac{\varepsilon_0\lambda_2}{2}(\|M_A^{-\frac{1}{2}}\delta_\tau^2\boldsymbol{E}_h^{n+1}\|^2 - \|M_A^{-\frac{1}{2}}\delta_\tau^2\boldsymbol{E}_h^{n}\|^2) \\
&\quad + \frac{\varepsilon_0\lambda_2\omega_p^2}{4}(\|M_A^{-\frac{1}{2}}\delta_\tau\boldsymbol{E}_h^{n+\frac{3}{2}}\|^2 - \|M_A^{-\frac{1}{2}}\delta_\tau\boldsymbol{E}_h^{n-\frac{1}{2}}\|^2) \\
&\quad + \frac{\varepsilon_0\lambda_2\omega_p^2}{2}(\|M_A^{-\frac{1}{2}}\delta_\tau\boldsymbol{E}_h^{n+\frac{1}{2}}\|^2 - \|M_A^{-\frac{1}{2}}\delta_\tau\boldsymbol{E}_h^{n-\frac{1}{2}}\|^2) \\
&\quad + \frac{\varepsilon_0\lambda_2\omega_p^4}{4}(\|M_A^{-\frac{1}{2}}\boldsymbol{E}_h^{n+1}\|^2 - \|M_A^{-\frac{1}{2}}\boldsymbol{E}_h^{n-1}\|^2) \\
&= \tau\varepsilon_0\lambda_2(M_A^{-1}\delta_\tau^2\boldsymbol{E}_h^n + \omega_p^2 M_A^{-1}\overline{\boldsymbol{E}}_h^n, \delta_\tau\breve{\boldsymbol{D}}_h^n) \\
&\quad + \tau\left(\delta_\tau^3\boldsymbol{D}_h^{n+\frac{1}{2}} + M_C\delta_\tau\overline{\boldsymbol{D}}_h^{n+\frac{1}{2}}, \delta_\tau^2\breve{\boldsymbol{E}}_h^{n+\frac{1}{2}}\right) + \tau\omega_p^2\left(\delta_\tau^2\boldsymbol{D}_h^n + M_C\overline{\boldsymbol{D}}_h^n, \delta_\tau\breve{\boldsymbol{E}}_h^n\right).
\end{aligned} \tag{5.20}$$

After dividing both sides of (5.20) by τ, we can see that (5.20) is really a discrete form of (4.27)!

(IV) Similar to the semi-discrete case, now we need to bound the terms $\delta_\tau^3\boldsymbol{D}_h^{n+\frac{1}{2}}$ and $\delta_\tau^2\boldsymbol{D}_h^n$ on the RHS of (5.20).

Using (5.1) and (5.2) to subtract themselves with n replaced by $n-1$, respectively, then letting $\phi_x = \frac{1}{\tau}\delta_\tau \breve{\boldsymbol{E}}^n_{xh}$ and $\phi_y = \frac{1}{\tau}\delta_\tau \breve{\boldsymbol{E}}^n_{yh}$ and adding the results together, we have

$$\begin{aligned} &\int_e \delta_\tau^2 \boldsymbol{D}_h^n \cdot \delta_\tau \breve{\boldsymbol{E}}_h^n + \int_e \delta_\tau H_h^n \cdot \partial_y \delta_\tau \breve{E}^n_{xh} - \int_e \delta_\tau H_h^n \cdot \partial_x \delta_\tau \breve{E}^n_{yh} \\ &\quad - \int_{\partial e} \delta_\tau \hat{H}_h^n \cdot \delta_\tau \breve{E}_{xh}^{n(in)} n_y^{(in)} + \int_{\partial e} \delta_\tau \hat{H}_h^n \cdot \delta_\tau \breve{E}_{yh}^{n(in)} n_x^{(in)} = 0. \end{aligned} \tag{5.21}$$

Using (5.4) with n replaced by $n+1$ to subtract itself with n replaced by $n-1$, then choosing $\psi = \frac{1}{2\tau}\delta_\tau H_h^n$, and using the identity

$$\begin{aligned} E_{yh}^{n+1} - E_{yh}^{n-1} &= \tau\left(\frac{E_{yh}^{n+1} - E_{yh}^n}{\tau} + \frac{E_{yh}^n - E_{yh}^{n-1}}{\tau}\right) \\ &= \tau(\delta_\tau E_{yh}^{n+\frac{1}{2}} + \delta_\tau E_{yh}^{n-\frac{1}{2}}) \\ &= 2\tau \delta_\tau \breve{E}^n_{yh}, \end{aligned} \tag{5.22}$$

we have

$$\begin{aligned} &\frac{\mu_0 \mu}{2\tau} \int_e (\delta_\tau H_h^{n+1} - \delta_\tau H_h^{n-1}) \delta_\tau H_h^n - \int_e \delta_\tau \breve{E}^n_{yh} \cdot \partial_x \delta_\tau H_h^n + \int_e \delta_\tau \breve{E}^n_{xh} \cdot \partial_y \delta_\tau H_h^n \\ &\quad + \int_{\partial e} (\delta_\tau \hat{\breve{E}}^n_{yh} n_x^{(in)} - \delta_\tau \hat{\breve{E}}^n_{xh} n_y^{(in)}) \delta_\tau H_h^{n(in)} = 0. \end{aligned} \tag{5.23}$$

Adding (5.21) and (5.23) together, then using integration by parts, and summing up the result over all elements, we obtain

$$\begin{aligned} &(\delta_\tau^2 \boldsymbol{D}_h^n, \delta_\tau \breve{\boldsymbol{E}}_h^n) + \frac{\mu_0 \mu}{2\tau}(\delta_\tau H_h^{n+1} - \delta_\tau H_h^{n-1}, \delta_\tau H_h^n) \\ &\quad + \sum_{e \in \mathcal{T}_h} \int_{\partial e} (-\delta_\tau \hat{H}_h^n \cdot \delta_\tau \breve{E}_{xh}^{n(in)} n_y^{(in)} + \delta_\tau \hat{H}_h^n \cdot \delta_\tau \breve{E}_{yh}^{n(in)} n_x^{(in)}) \\ &\quad + \sum_{e \in \mathcal{T}_h} \int_{\partial e} (-\delta_\tau H_h^{n(in)} \cdot \delta_\tau \breve{E}_{yh}^{n(in)} n_x^{(in)} + \delta_\tau H_h^{n(in)} \cdot \delta_\tau \breve{E}_{xh}^{n(in)} n_y^{(in)}) \\ &\quad + \sum_{e \in \mathcal{T}_h} \int_{\partial e} (\delta_\tau \hat{\breve{E}}^n_{yh} n_x^{(in)} - \delta_\tau \hat{\breve{E}}^n_{xh} n_y^{(in)}) \delta_\tau H_h^{n(in)} = 0. \end{aligned} \tag{5.24}$$

We assign all boundary integral terms of (5.24) into G_x and G_y classes:

$$\begin{aligned} G_x &= \sum_{e\in\mathcal{T}_h} \int_{\partial e} (-\delta_\tau \hat{H}_h^n \cdot \delta_\tau \breve{E}_{xh}^n n_y^{(in)} + \delta_\tau H_h^n \cdot \delta_\tau \breve{E}_{xh}^n n_y^{(in)} - \delta_\tau H_h^n \cdot \delta_\tau \breve{\hat{E}}_{xh}^n n_y^{(in)}), \\ G_y &= \sum_{e\in\mathcal{T}_h} \int_{\partial e} (\delta_\tau \hat{H}_h^n \cdot \delta_\tau \breve{E}_{yh}^n n_x^{(in)} - \delta_\tau H_h^n \cdot \delta_\tau \breve{E}_{yh}^n n_x^{(in)} + \delta_\tau H_h^n \cdot \delta_\tau \breve{\hat{E}}_{yh}^n n_x^{(in)}). \end{aligned} \tag{5.25}$$

By regrouping terms by sides of the elements and using the definitions of the numerical fluxes $\hat{H}_h^n$ and $\hat{E}_{xh}^n$, we have:

$$\begin{aligned} G_x = &\sum_{s\in\mathcal{S}_I} n_y^R \int_s \Big(-\delta_\tau H_h^{n,L} \cdot \delta_\tau \breve{E}_{xh}^{n,R} + \delta_\tau H_h^{n,L} \cdot \delta_\tau \breve{E}_{xh}^{n,L} + \delta_\tau H_h^{n,R} \cdot \delta_\tau \breve{E}_{xh}^{n,R} \\ &- \delta_\tau H_h^{n,L} \cdot \delta_\tau \breve{E}_{xh}^{n,L} - \delta_\tau H_h^{n,R} \cdot \delta_\tau \breve{E}_{xh}^{n,R} + \delta_\tau H_h^{n,L} \cdot \delta_\tau \breve{E}_{xh}^{n,R} \Big) + \\ &\sum_{s\in\mathcal{S}_{Top}} n_y^R \int_s \Big(-\delta_\tau H_h^{n,(in)} \delta_\tau \breve{E}_{xh}^{n,(in)} + \delta_\tau H_h^{n,(in)} \delta_\tau \breve{E}_{xh}^{n,(in)} - \delta_\tau H_h^{n,(in)} \delta_\tau \breve{\hat{E}}_{xh}^n \Big) + \\ &\sum_{s\in\mathcal{S}_{Bottom}} n_y^R \int_s \Big(-\delta_\tau H_h^{n,(in)} \delta_\tau \breve{E}_{xh}^{n,(in)} + \delta_\tau H_h^{n,(in)} \delta_\tau \breve{E}_{xh}^{n,(in)} - \delta_\tau H_h^{n,(in)} \delta_\tau \breve{\hat{E}}_{xh}^n \Big) \\ &= 0, \end{aligned} \tag{5.26}$$

where $\mathcal{S}_I$ denotes the set of all non-boundary sides, $\mathcal{S}_{Top}$ represents the set of sides on $y = d$, and $\mathcal{S}_{Bottom}$ on $y = c$.

Similarly, we can prove that $G_y = 0$. Substituting $G_x = G_y = 0$ into (5.24), we have

$$(\delta_\tau^2 \boldsymbol{D}_h^n, \delta_\tau \breve{\boldsymbol{E}}_h^n) = -\frac{\mu_0 \mu}{2\tau} (\delta_\tau H_h^{n+1} - \delta_\tau H_h^{n-1}, \delta_\tau H_h^n), \tag{5.27}$$

which is the discrete form of (4.34).

(V) Following the same technique as (IV), we can obtain

$$(\delta_\tau^3 \boldsymbol{D}_h^{n-\frac{1}{2}}, \delta_\tau^2 \breve{\boldsymbol{E}}_h^{n-\frac{1}{2}}) = -\frac{\mu_0\mu}{2\tau}(\delta_\tau^2 H_h^{n+\frac{1}{2}} - \delta_\tau^2 H_h^{n-\frac{3}{2}}, \delta_\tau^2 H_h^{n-\frac{1}{2}}), \tag{5.28}$$

which is the discrete form of (4.35).

Substituting (5.27) and (5.28) into (5.20), we have

$$\begin{aligned}
&\frac{1}{2}(\|\delta_\tau \boldsymbol{D}_h^{n+\frac{1}{2}}\|^2 - \|\delta_\tau \boldsymbol{D}_h^{n-\frac{1}{2}}\|^2) + \frac{1}{4}(\|M_C^{\frac{1}{2}} \boldsymbol{D}_h^{n+1}\|^2 - \|M_C^{\frac{1}{2}} \boldsymbol{D}_h^{n-1}\|^2) \\
&\quad + \frac{\varepsilon_0\lambda_2}{2}(\|M_A^{-\frac{1}{2}} \delta_\tau^2 \boldsymbol{E}_h^n\|^2 - \|M_A^{-\frac{1}{2}} \delta_\tau^2 \boldsymbol{E}_h^{n-1}\|^2) \\
&\quad + \frac{\varepsilon_0\lambda_2\omega_p^2}{4}(\|M_A^{-\frac{1}{2}} \delta_\tau \boldsymbol{E}_h^{n+\frac{1}{2}}\|^2 - \|M_A^{-\frac{1}{2}} \delta_\tau \boldsymbol{E}_h^{n-\frac{3}{2}}\|^2) \\
&\quad + \frac{\varepsilon_0\lambda_2\omega_p^2}{2}(\|M_A^{-\frac{1}{2}} \delta_\tau \boldsymbol{E}_h^{n+\frac{1}{2}}\|^2 - \|M_A^{-\frac{1}{2}} \delta_\tau \boldsymbol{E}_h^{n-\frac{1}{2}}\|^2) \\
&\quad + \frac{\varepsilon_0\lambda_2\omega_p^4}{4}(\|M_A^{-\frac{1}{2}} \boldsymbol{E}_h^{n+1}\|^2 - \|M_A^{-\frac{1}{2}} \boldsymbol{E}_h^{n-1}\|^2) \\
&\quad + \frac{\mu_0\mu}{2}\left[\omega_p^2(\delta_\tau H_h^{n+1} - \delta_\tau H_h^{n-1}, \delta_\tau H_h^n) + (\delta_\tau^2 H_h^{n+\frac{1}{2}} - \delta_\tau^2 H_h^{n-\frac{3}{2}}, \delta_\tau^2 H_h^{n-\frac{1}{2}})\right] \\
&= \tau\varepsilon_0\lambda_2(M_A^{-1}\delta_\tau^2 \boldsymbol{E}_h^n + \omega_p^2 M_A^{-1} \overline{\boldsymbol{E}}_h^n, \delta_\tau \breve{\boldsymbol{D}}_h^n) + \tau\left(M_C \delta_\tau \overline{\boldsymbol{D}}_h^{n-\frac{1}{2}}, \delta_\tau^2 \breve{\boldsymbol{E}}_h^{n-\frac{1}{2}}\right) \\
&\quad + \tau\omega_p^2\left(M_C \overline{\boldsymbol{D}}_h^n, \delta_\tau \breve{\boldsymbol{E}}_h^n\right),
\end{aligned} \tag{5.29}$$

which is the discrete form of (4.36).

Multiplying (5.29) by 2, then summing up the result from $n = 1$ to $n = m$, and using the identity

$$\sum_{n=1}^{m}(a_{n+1} - a_{n-1}, a_n) = (a_{m+1}, a_m) - (a_1, a_0), \tag{5.30}$$

we obtain

$$
\begin{aligned}
ENG_{LF}(m) - ENG_{LF}(0) = \sum_{n=1}^{m} 2\Big[&\tau\varepsilon_0\lambda_2(M_A^{-1}\delta_\tau^2\boldsymbol{E}_h^n + \omega_p^2 M_A^{-1}\overline{\boldsymbol{E}}_h^n, \delta_\tau\breve{\boldsymbol{D}}_h^n) \\
&+ \tau(M_C\delta_\tau\overline{\boldsymbol{D}}_h^{n-\frac{1}{2}}, \delta_\tau^2\breve{\boldsymbol{E}}_h^{n-\frac{1}{2}}) + \tau\omega_p^2(M_C\overline{\boldsymbol{D}}_h^n, \delta_\tau\breve{\boldsymbol{E}}_h^n)\Big].
\end{aligned}
\tag{5.31}
$$

Then, we just need to bound the RHS terms of (5.31) and use Lemma 5.1.1 to finish the proof. By using the following two inequalities and estimating the RHS terms one by one:

$$
2ab \le a^2 + b^2, \qquad \left(\frac{a+b}{2}\right)^2 \le \frac{1}{2}(a+b)^2,
$$

we obtain the following four estimates:

$$
\begin{aligned}
&\sum_{n=1}^{m} 2\tau\varepsilon_0\lambda_2(M_A^{-1}\delta_\tau^2\boldsymbol{E}_h^n, \delta_\tau\breve{\boldsymbol{D}}_h^n) \\
&\le \sum_{n=1}^{m} 2\tau\varepsilon_0\lambda_2 ||M_A^{-\frac{1}{2}}|| ||M_A^{-\frac{1}{2}}\delta_\tau^2\boldsymbol{E}_h^n|| ||\frac{1}{2}\delta_\tau(\boldsymbol{D}_h^{n+\frac{1}{2}} + \boldsymbol{D}_h^{n-\frac{1}{2}})|| \\
&\le \tau\sqrt{\varepsilon_0\lambda_2}||M_A^{-\frac{1}{2}}|| \sum_{n=1}^{m}\Big(\varepsilon_0\lambda_2||M_A^{-\frac{1}{2}}\delta_\tau^2\boldsymbol{E}_h^n||^2 \\
&\qquad + \frac{1}{2}(||\delta_\tau\boldsymbol{D}_h^{n+\frac{1}{2}}||^2 + ||\delta_\tau\boldsymbol{D}_h^{n-\frac{1}{2}}||^2)\Big),
\end{aligned}
\tag{5.32}
$$

$$\begin{aligned}
&\sum_{n=1}^{m} 2\tau\varepsilon_0\lambda_2\omega_p^2(M_A^{-1}\overline{\boldsymbol{E}}_h^n, \delta_\tau\breve{\boldsymbol{D}}_h^n) \\
&\leq \sum_{n=1}^{m} 2\tau\varepsilon_0\lambda_2\omega_p^2\|M_A^{-\frac{1}{2}}\|\|M_A^{-\frac{1}{2}}\frac{1}{2}(\boldsymbol{E}_h^{n+1} + \boldsymbol{E}_h^{n-1})\|\|\frac{1}{2}\delta_\tau(\boldsymbol{D}_h^{n+\frac{1}{2}} + \boldsymbol{D}_h^{n-\frac{1}{2}})\| \\
&\leq \tau\sqrt{\varepsilon_0\lambda_2}\|M_A^{-\frac{1}{2}}\|\sum_{n=1}^{m} 2\sqrt{\varepsilon_0\lambda_2}\omega_p^2\|\frac{1}{2}(M_A^{-\frac{1}{2}}\boldsymbol{E}_h^{n+1} + M_A^{-\frac{1}{2}}\boldsymbol{E}_h^{n-1})\| \\
&\qquad \cdot \|\frac{1}{2}\delta_\tau(\boldsymbol{D}_h^{n+\frac{1}{2}} + \boldsymbol{D}_h^{n-\frac{1}{2}})\| \\
&\leq \tau\sqrt{\varepsilon_0\lambda_2}\|M_A^{-\frac{1}{2}}\|\sum_{n=1}^{m}\Big(\frac{\varepsilon_0\lambda_2\omega_p^4}{2}(\|M_A^{-\frac{1}{2}}\boldsymbol{E}_h^{n+1}\|^2 + \|M_A^{-\frac{1}{2}}\boldsymbol{E}_h^{n-1}\|^2) \\
&\qquad + \frac{1}{2}(\|\delta_\tau\boldsymbol{D}_h^{n+\frac{1}{2}}\|^2 + \|\delta_\tau\boldsymbol{D}_h^{n-\frac{1}{2}}\|^2)\Big),
\end{aligned} \tag{5.33}$$

$$\begin{aligned}
&\sum_{n=1}^{m} 2\tau(M_C\delta_\tau\overline{\boldsymbol{D}}_h^{n-\frac{1}{2}}, \delta_\tau^2\breve{\boldsymbol{E}}_h^{n-\frac{1}{2}}) \\
&= \sum_{n=1}^{m} 2\tau\Big(M_A^{-1}M_B\cdot\frac{1}{2}\delta_\tau(\boldsymbol{D}_h^{n+\frac{1}{2}} + \boldsymbol{D}_h^{n-\frac{3}{2}}), \frac{1}{2}\delta_\tau^2(\boldsymbol{E}_h^n + \boldsymbol{E}_h^{n-1})\Big) \\
&\leq \sum_{n=1}^{m}\frac{\tau\|M_A^{-\frac{1}{2}}M_B\|}{\sqrt{\varepsilon_0\lambda_2}} \\
&\qquad \cdot 2\|\frac{1}{2}\delta_\tau(\boldsymbol{D}_h^{n+\frac{1}{2}} + \boldsymbol{D}_h^{n-\frac{3}{2}})\|\cdot\sqrt{\varepsilon_0\lambda_2}\|M_A^{-\frac{1}{2}}\frac{1}{2}\delta_\tau^2(\boldsymbol{E}_h^n + \boldsymbol{E}_h^{n-1})\| \\
&\leq \frac{\tau\|M_A^{-\frac{1}{2}}M_B\|}{\sqrt{\varepsilon_0\lambda_2}}\sum_{n=1}^{m}\Big(\frac{1}{2}(\|\delta_\tau\boldsymbol{D}_h^{n+\frac{1}{2}}\|^2 + \|\delta_\tau\boldsymbol{D}_h^{n-\frac{3}{2}}\|^2) \\
&\qquad + \frac{\varepsilon_0\lambda_2}{2}(\|M_A^{-\frac{1}{2}}\delta_\tau^2\boldsymbol{E}_h^n\|^2 + \|M_A^{-\frac{1}{2}}\delta_\tau^2\boldsymbol{E}_h^{n-1}\|^2)\Big),
\end{aligned} \tag{5.34}$$

and

$$\begin{aligned}
&\sum_{n=1}^{m} 2\tau\omega_p^2(M_C\overline{\boldsymbol{D}}_h^n, \delta_\tau\breve{\boldsymbol{E}}_h^n) \\
&= \sum_{n=1}^{m} 2\tau\omega_p^2\Big(M_C^{\frac{1}{2}}\frac{1}{2}(\boldsymbol{D}_h^{n+1} + \boldsymbol{D}_h^{n-1}), M_C^{\frac{1}{2}}M_A^{\frac{1}{2}}M_A^{-\frac{1}{2}}\frac{1}{2}\delta_\tau(\boldsymbol{E}_h^{n+\frac{1}{2}} + \boldsymbol{E}_h^{n-\frac{1}{2}})\Big) \\
&\leq \frac{\tau\omega_p\|M_C^{\frac{1}{2}}M_A^{\frac{1}{2}}\|}{\sqrt{\varepsilon_0\lambda_2}} \sum_{n=1}^{m} 2\|M_C^{\frac{1}{2}}\frac{1}{2}(\boldsymbol{D}_h^{n+1} + \boldsymbol{D}_h^{n-1})\| \\
&\qquad \cdot \sqrt{\varepsilon_0\lambda_2}\omega_p\|M_A^{-\frac{1}{2}}\frac{1}{2}\delta_\tau(\boldsymbol{E}_h^{n+\frac{1}{2}} + \boldsymbol{E}_h^{n-\frac{1}{2}})\| \\
&\leq \frac{\tau\omega_p\|M_C^{\frac{1}{2}}M_A^{\frac{1}{2}}\|}{\sqrt{\varepsilon_0\lambda_2}} \sum_{n=1}^{m}\Big(\frac{1}{2}(\|M_C^{\frac{1}{2}}\boldsymbol{D}_h^{n+1}\|^2 + \|M_C^{\frac{1}{2}}\boldsymbol{D}_h^{n-1})\|^2) \\
&\qquad + \frac{\varepsilon_0\lambda_2\omega_p^2}{2}(\|M_A^{-\frac{1}{2}}\delta_\tau\boldsymbol{E}_h^{n+\frac{1}{2}}\|^2 + \|M_A^{-\frac{1}{2}}\delta_\tau\boldsymbol{E}_h^{n-\frac{1}{2}}\|^2)\Big)
\end{aligned} \tag{5.35}$$

Substituting (5.32)-(5.35) into (5.31), and choosing the time step τ to make the coefficients of $\|M_C^{\frac{1}{2}}\boldsymbol{D}_h^{n+1}\|$, $\|\delta_\tau\boldsymbol{D}_h^{n+\frac{1}{2}}\|$, $\|M_A^{-\frac{1}{2}}\boldsymbol{E}_h^{n+1}\|$, $\|M_A^{-\frac{1}{2}}\delta_\tau\boldsymbol{E}_h^{n+\frac{1}{2}}\|$ on the LHS smaller than those on the RHS:

$$\begin{aligned}
&\frac{\tau\omega_p\|M_C^{\frac{1}{2}}M_A^{\frac{1}{2}}\|}{2\sqrt{\varepsilon_0\lambda_2}} \leq \frac{1}{2}, \quad \tau\sqrt{\varepsilon_0\lambda_2}\|M_A^{-\frac{1}{2}} + \frac{\tau\|M_A^{-\frac{1}{2}}M_B\|}{2\sqrt{\varepsilon_0\lambda_2}} \leq 1, \\
&\tau\sqrt{\varepsilon_0\lambda_2}\|M_A^{-\frac{1}{2}}\| \leq 1, \quad \frac{\tau\omega_p\|M_C^{\frac{1}{2}}M_A^{\frac{1}{2}}\|}{\sqrt{\varepsilon_0\lambda_2}} \leq 3,
\end{aligned} \tag{5.36}$$

which is equivalent to (5.12). Finally by applying the discrete Gronwall inequality given in Lemma 5.1.1, we finish the proof. ■

5.2 Numerical results

We now present two accuracy tests of the leap-frog DG methods (5.1)-(5.4) on the rectangular mesh and unstructured mesh, to verify the proved convergence results. Additionally, some numerical simulations of the cloaking phenomenon will be shown.

5.2.1 The error table on triangular meshes

We use the model in [54, Sect. 5] to test the convergence rate of our model:

$$\partial_t D_x = \frac{\partial H}{\partial y}, \tag{5.37}$$

$$\partial_t D_y = -\frac{\partial H}{\partial x}, \tag{5.38}$$

$$\varepsilon_0 \lambda_2 \left(M_A^{-1} \partial_{t^2} \boldsymbol{E} + \omega_p^2 M_A^{-1} \boldsymbol{E} \right) = \partial_{t^2} \boldsymbol{D} + M_C \boldsymbol{D} + \boldsymbol{f}(t_n), \tag{5.39}$$

$$\mu_0 \mu \partial_t H = -\nabla \times \boldsymbol{E}, \tag{5.40}$$

where the source term $\boldsymbol{f}$ is

$$\boldsymbol{f}(x, y, t) = \varepsilon_0 \lambda_2 \left(M_A^{-1} \partial_{t^2} \boldsymbol{E} + \omega_p^2 M_A^{-1} \boldsymbol{E} \right) - \partial_{t^2} \boldsymbol{D} - M_C \boldsymbol{D}. \tag{5.41}$$

The model has exact solutions

$$E_x(x, y, t) = \cos(\omega x)\sin(\omega y)e^{-\omega_f t}, \tag{5.42}$$

$$E_y(x, y, t) = -\sin(\omega x)\cos(\omega y)e^{-\omega_f t}, \tag{5.43}$$

$$D_x(x, y, t) = \frac{-2\omega}{\mu_0\mu\omega_f^2}\cos(\omega x)\sin(\omega y)e^{-\omega_f t}, \tag{5.44}$$

$$D_y(x, y, t) = \frac{-2\omega}{\mu_0\mu\omega_f^2}(-\sin(\omega x)\cos(\omega y))e^{-\omega_f t}, \tag{5.45}$$

$$H(x, y, t) = \frac{-2\omega}{\mu_0\mu\omega_f}\cos(\omega x)\cos(\omega y)e^{-\omega_f t}. \tag{5.46}$$

We use the unit square as our physical domain, which is partitioned by the triangular mesh. Fig. 5.1 shows a sample coarse mesh.

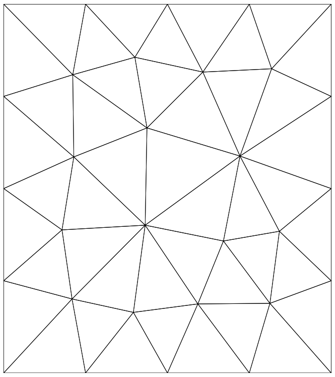


Figure 5.1: Sample mesh for the unit region.

We couple the leap-frog time discretization with the second and third order DG methods, and apply the alternating fluxes and the PEC boundary conditions (5.5)-(5.10) to solve the model. The physical parameters in the test are chosen as:

$$H_1 = 0.05,\ H_2 = 0.2,\ d = 0.2,\ \epsilon_0 = \mu_0 = \pi,\ \mu = 4\pi,\ T = 0.1.$$

Table 5.1: L_2 errors and orders obtained the DG method for $\boldsymbol{E}$,$\boldsymbol{D}$, and H on the unstructured mesh.

			k=1			
Level of refinement	$\boldsymbol{E}$ error	order	$\boldsymbol{D}$ error	order	H error	order
1	1.15 E-01		5.31 E-01		9.91 E-02	
2	4.17 E-02	1.29	2.09 E-01	1.34	3.06 E-02	1.69
3	1.38 E-02	1.77	5.98 E-02	1.80	1.33 E-02	1.19
4	4.67 E-03	1.56	1.88 E-02	1.66	2.49 E-03	2.42
5	2.24 E-03	1.05	8.69 E-03	1.11	6.63 E-04	1.91

			k=2			
Level of refinement	$\boldsymbol{E}$ error	order	$\boldsymbol{D}$ error	order	H error	order
1	2.66 E-02		1.21 E-01		1.99 E-02	
2	4.60 E-03	2.53	2.07 E-02	2.54	3.49 E-03	2.51
3	7.31 E-04	2.65	3.11 E-03	2.73	4.71 E-04	2.88
4	1.27 E-04	2.51	4.90 E-04	2.66	6.68 E-05	2.82
5	2.82 E-05	2.17	1.50 E-04	2.21	9.10 E-06	2.87

The time step is chosen as $\tau = 0.01h$ for the second order DG method, and $\tau = 0.01h^{\frac{3}{2}}$ for the third order DG method, where h is the mesh size. The L_2 errors and the corresponding convergence rates of $||\boldsymbol{E}_h^{n+1} - \boldsymbol{E}(t_{n+1})||$, $||\boldsymbol{D}_h^{n+1} - \boldsymbol{D}(t_{n+1})||$, and $||H_h^{n+\frac{1}{2}} - H(t_{n+\frac{1}{2}})||$ are shown in Table 5.1. We observe the sub-optimal convergence rates of $O(h^k)$ in the L^2 norm, which is consistent with Theorem 4.3.2.

5.2.2 The error table on rectangular meshes

Next, we partition the unit square domain with the rectangular mesh, and apply the alternating fluxes with additional jump terms on the PEC boundary conditions (4.53)-(4.60) to simulate the model (5.37)-(5.40). To achieve the optimal order of convergence, we set the initial conditions as:

$$E_{xh}(0) = \Pi_1 E_x(0), \;\; E_{yh}(0) = \Pi_2 E_y(0), \;\; D_{xh}(0) = \Pi_4 D_x(0),$$
$$D_{yh}(0) = \Pi_4 D_y(0), \;\; H_h(0) = \Pi_3 H(0).$$

Table 5.2: L_2 errors and orders obtained from the leap-frog DG method for ***E***,***D***, and H on the rectangular mesh.

k=1						
# cells	***E*** error	order	***D*** error	order	H error	order
10×10	1.38 E-02		8.98 E-02		1.38 E-02	
20×20	3.40 E-03	1.95	2.26 E-02	1.96	3.40 E-03	1.95
40×40	8.67 E-04	1.97	5.73 E-03	1.98	8.67 E-04	1.97
80×80	2.18 E-04	1.99	1.43 E-03	1.99	2.18 E-04	1.99
160×160	5.47 E-05	2.00	3.60 E-04	1.99	5.47 E-05	2.00
k=2						
# cells	***E*** error	order	***D*** error	order	H error	order
10×10	7.93 E-03		2.18 E-02		6.57 E-03	
20×20	9.85 E-04	3.00	2.75 E-03	2.98	8.41 E-04	2.96
40×40	1.22 E-04	3.01	3.47 E-04	2.98	1.10 E-04	2.97
80×80	1.52 E-05	3.00	4.33 E-05	2.99	1.37 E-05	3.01
160×160	1.90 E-06	3.00	5.42 E-06	2.99	1.71 E-06	3.00

Table. 5.2 shows the L^2 errors and the convergence rates in this case. As proved in Theorem 4.3.3, the optimal order of accuracy is obtained.

5.2.3 The wave propagation cross the cloaking region

To see the invisibility cloaking phenomenon, we test our leap-frog DG scheme on Example 2 in [54], where the physical domain is $[-0.6, 0.6]$m $\times$ $[0, 0.6]$m, and the physical parameters for the simulation are

$$H_1 = 0.1m, \;\; H_2 = 0.4m, \;\; d = 0.4m, \;\; \tau = 1e - 13s.$$

The domain is partitioned by the unstructured triangular mesh with mesh size $h = 0.01$m, and it is surrounded by a perfectly matched layer (PML) of thickness

$15h$ to absorb outgoing waves. Here we use the classical 2D Berenger PML, whose governing equations are [50]:

$$\epsilon_0 \partial_t \boldsymbol{E} + \begin{pmatrix} \sigma_y & 0 \\ 0 & \sigma_x \end{pmatrix} \boldsymbol{E} = \nabla \times H_z, \tag{5.47}$$

$$\mu_0 \partial_t H_{zx} + \sigma_{mx} H_{zx} = -\frac{\partial E_y}{\partial x}, \tag{5.48}$$

$$\mu_0 \partial_t H_{zy} + \sigma_{my} H_{zy} = \frac{\partial E_x}{\partial y}, \tag{5.49}$$

where $H_z = H_{zx} + H_{zy}$ represents the magnetic field, and the parameters σ_i and $\sigma_{m,i}$, $i = x, y$ denote the electric and the magnetic conductivities in the x- and y-directions respectively.

In the domain, an incident Gaussian wave

$$H(x, y, t) = \sin(2\pi f) \exp(-\frac{|\boldsymbol{x} - \boldsymbol{x}_c|^2}{L^2})$$

is imposed along a line segment with endpoints $(-d, d/2)$ and $(-d/2, d)$. We set the frequency $f = 2\text{GHz}$, $L = 0.25\sqrt{2}d$, and $\boldsymbol{x}_c = (-3d/4, 3d/4)$, where $\boldsymbol{x} = (x, y)$ is an arbitrary point on the segment. The Fig. 5.2 shows that the computational domain is wrapped by the green PML region. The red quadrilateral region represents the cloaking region, where the carpet cloak model (4.1-4.4) is solved. The rest blue region is vacuum, where the standard Maxwell equation is solved. The numerical magnetic field H at different time steps are shown in Fig. 5.3, and it can be observed that the wave looks like the one reflecting from the flat ground, and the the hidden region is invisible to the observers at the far end.

For the comparison, the simulation of the magnetic field H without the cloaking material is presented in Fig. 5.4, and the cloak phenomenon disappears in this case.

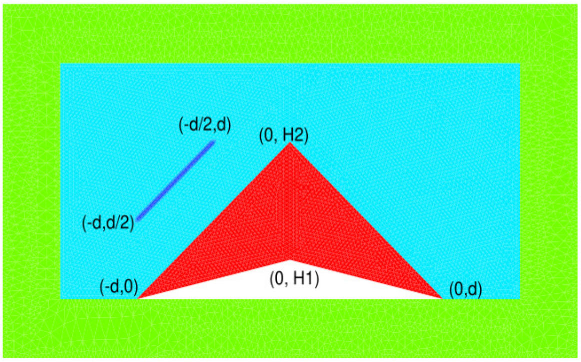


Figure 5.2: The computational domain for Examples 5.3 and 5.4.

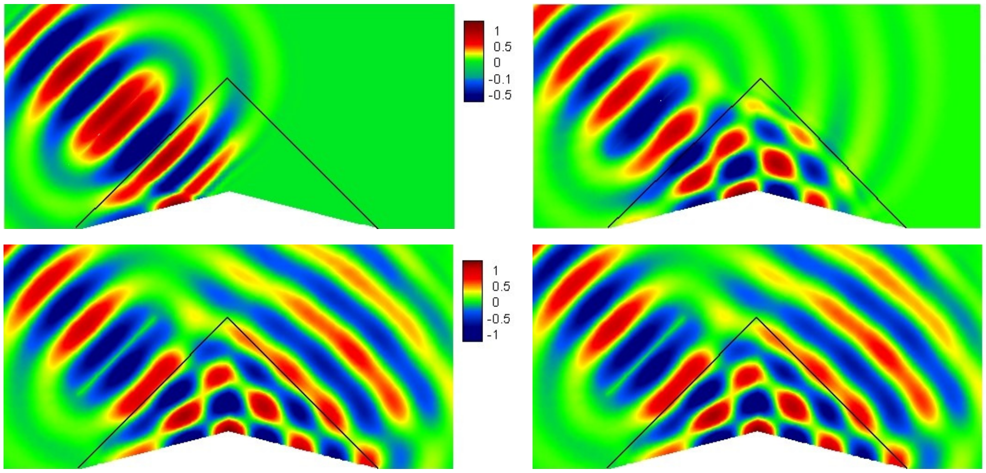


Figure 5.3: Example 5.3 (with metamaterial). The magnetic field H obtained at 12000, 24000, 40000, and 50000 time steps (oriented counterclockwise).

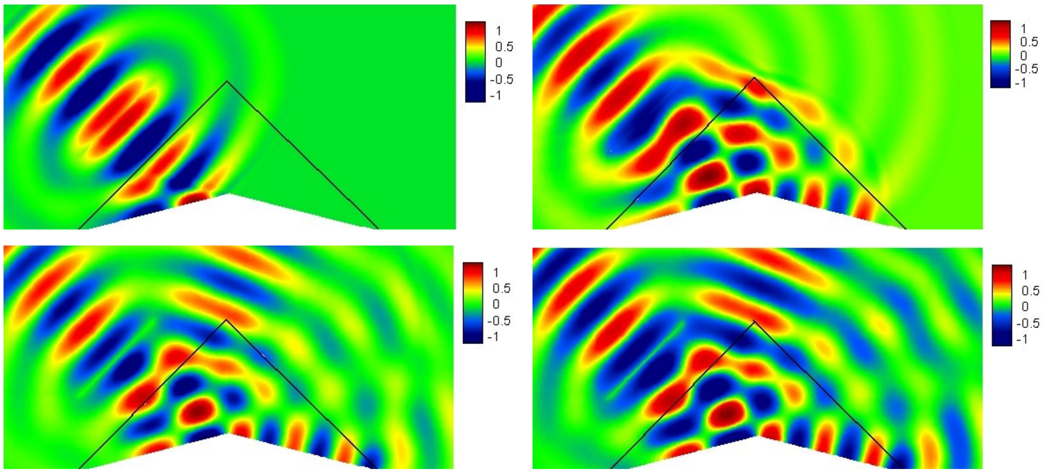


Figure 5.4: Example 5.3 (without metamaterial). The magnetic field H obtained at 12000, 24000, 40000, and 50000 time steps (oriented counterclockwise).

5.2.4 The wave propagation with a vertical incident wave source

We repeat Example 5.2, and substitute the incident Gaussian wave to a vertical source wave $H(x, y, t) = 0.1\sin(2\pi f)$ with the frequency $f = 2$GHz on edge $x = -0.6$m. The numerical solutions of H at each time step are shown in Fig. 5.5. This result shows that the plane wave pattern is perfectly recovered after passing through the cloaking region, and we conclude that the cloaking phenomenon is also achieved in this case.

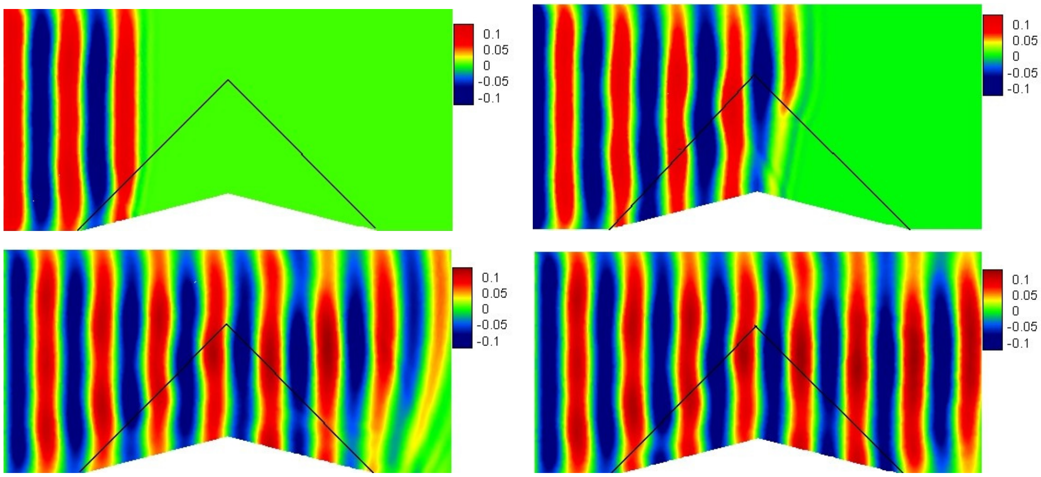


Figure 5.5: Example 5.4. The magnetic field H obtained at 12000, 24000, 40000, and 50000 time steps (oriented counterclockwise).

Chapter Six

Conclusion

This dissertation presents two topics: the first one is the development of the limiter applied on the DG methods, and the second one is about the application of the LDG methods on a physical model.

In the first part, we design a MLP based TVB limiter for solving hyperbolic conservation laws in one and two dimensional scalar and system cases using DG schemes. In comparison with the classical minmod-based TVB limiter with an empirically chosen TVB constant M, the new MLP based TVB limiter has following advantages. First of all, the MLP limiter is able to control spurious oscillations near discontinuities without excessive smearing, while maintaining the original high order accuracy in smooth regions, including near smooth extrema. Furthermore, the MLP procedure automates the choice of the TVB constant M, thus eliminates the need to choose M in an ad hoc fashion. This is especially important for hyperbolic systems, for which no rigorous mathematical guidance exists for the choice of M. Secondly, the model training can be performed offline, leaving the online computation efficient involving only a few low-cost matrix multiplications.Thus it is simple to modify the standard DG code to apply the new limiter, and the extra coding only involves a few lines. Last but not the least, the MLP based TVB limiter works well for the DG scheme of various orders of accuracy, and give the same or even better performance than the classical TVB limiter with manually chosen TVB constant M through trial and error, for an extensive list of numerical test problems in 1D and 2D.

In the second part, we develop the leap-frog DG scheme for solving the time-domain carpet cloak model. We prove the stability and the sub-optimal order of convergence for the semi-discrete scheme on triangular meshes, and the optimal order of convergence on rectangular meshes with tensor-product DG spaces. Then, the conditional stability for the fully-discrete scheme with the time step constraint $\tau = O(h)$ is proved. Numerically, the sub-optimal convergence rate on unstructured meshes and the optimal convergence rate on rectangular meshes with tensor-product DG spaces are verified in the error accuracy tests. Moreover, simulations of wave propagation in the carpet cloak region are presented.

Bibliography

[1] H. Ammari, H. Kang, H. Lee, M. Lim and S. Yu, Enhancement of near cloaking for the full Maxwell equations, SIAM Journal on Applied Mathematics, 73, 2013, 2055-2076.

[2] A. Anees and L. Angermann, Time domain finite element method for Maxwell's equations, IEEE Access, 7, 2019, 63852-63867.

[3] R. Biswas, K. Devine and J. Flaherty, Parallel, adaptive finite element methods for conservation laws, Applied Numerical Mathematics, 14, 1994, 255-283.

[4] V.A. Bokil, Y. Cheng, Y. Jiang and F. Li, Energy stable discontinuous Galerkin methods for Maxwell's equations in nonlinear optical media, Journal of Computational Physics, 350, 2017, 420-452.

[5] S.C. Brenner, J. Gedicke and L.-Y. Sung, An adaptive P1 finite element method for two-dimensional transverse magnetic time harmonic Maxwell's equations with general material properties and general boundary conditions, Journal of Scientific Computing, 68, 2016, 848-863.

[6] A. Buffa, P. Houston and I. Perugia, Discontinuous Galerkin computation of the Maxwell eigenvalues on simplicial meshes, Journal of Computational and Applied Mathematics, 204, 2007, 317-333.

[7] M. Cassier, P. Joly and M. Kachanovska, Mathematical models for dispersive electromagnetic waves: An overview, Computers & Mathematics with Applications, 74, 2017, 2792-2830.

[8] T. Chen and C.-W. Shu, Entropy stable high order discontinuous Galerkin methods with suitable quadrature rules for hyperbolic conservation laws, Journal of Computational Physics, 345, 2017, 427-461.

[9] T. Chen and C.-W. Shu, Review of entropy stable discontinuous Galerkin methods for systems of conservation laws on unstructured simplex meshes, CSIAM Transactions on Applied Mathematics (CSAM), 1, 2020, 1-52.

[10] E.T. Chung, P.Ciarlet Jr., T.F. Yu, Convergence and superconvergence of staggered discontinuous Galerkin methods for the three-dimensional Maxwell equations on Cartesian grids,Journal of Computational Physics, 235, 2013, 14–31.

[11] E.T. Chung and P. Ciarlet Jr., A staggered discontinuous Galerkin method for wave propagation in media with dielectrics and meta-materials, Journal of Computational and Applied Mathematics, 239, 2013, 189-207.

[12] P.G. Ciarlet, Finite Element Method for Elliptic Problems, SIAM, Philadelphia, 2002.

[13] B.Cockburn, F.Li, C.-W.Shu, Locally divergence-free discontinuous Galerkin methods for the Maxwell equations, Journal of Computational Physics, 194, 2004, 588–610.

[14] B. Cockburn, S. Hou and C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case, Mathematics of Computation, 54, 1990, 545-581.

[15] B. Cockburn, S.-Y. Lin and C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one-dimensional systems, Journal of Computational Physics, 84, 1989, 90-113.

[16] B. Cockburn and C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework, Mathematics of Computation, 52, 1989, 411-435.

[17] B. Cockburn and C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation law V: multidimensional systems, Journal of Computational Physics, 141, 1998, 199-224.

[18] B. Cockburn and C.-W. Shu, The local discontinuous Galerkin method for time-dependent convection-diffusion systems, SIAM Journal on Numerical Analysis, 35, 1998, 2240-2463.

[19] B. Cockburn and C.-W. Shu, Runge-Kutta discontinuous Galerkin methods for convection-dominated problems, Journal of Scientific Computing, 16, 2001, 173-261.

[20] G. Cybenko, Continuous valued neural networks with two hidden layers are sufficient, Technical Report, Department of Computer Science, Tufts University, Medford, MA, 1988.

[21] L. Demkowicz, J. Kurtz, D. Pardo, M. Paszynski, W. Rachowicz and A. Zdunek, Computing with hp-Adaptive Finite Elements. Vol.2: Frontiers: Three Dimensional Elliptic and Maxwell Problems with Applications, Chapman & Hall/CRC, 2008.

[22] N. Discacciati, J.S. Hesthaven,and D. Ray, Controlling oscillations in high-order Discontinuous Galerkin schemes using artificial viscosity tuned by neural networks,Journal of Computational Physics, 409, 2020.

[23] L. Fezoui, S. Lanteri, S. Lohrengel, S. Piperno, Convergence and stability of a discontinuous Galerkin time-domain method for the 3D heterogeneous Maxwell equations on unstructured meshes, Mathematical Modelling and Numerical Analysis, 39, 2005, 1149–1176.

[24] G. Fu and C.-W. Shu, A new troubled-cell indicator for discontinuous Galerkin methods for hyperbolic conservation laws, Journal of Computational Physics, 347, 2017, 305-327.

[25] Z. Gao, X. Wen and W.S. Don, Enhanced robustness of the hybrid compact-WENO finite difference scheme for hyperbolic conservation laws with multi-resolution analysis and Tukey's boxplot method, Journal of Computational Physics, 73, 2017, 736-752.

[26] S. Golak, A MLP solver for first and second order partial differential equations, J.M. de Sá, L.A. Alexandre, W. Duch, and D. Mandic, (eds), Artificial Neural Networks-ICANN 2007, Springer, Berlin, Heidelberg, 2007, 789-797.

[27] A. Greenleaf, Y. Kurylev, M. Lassas and G. Uhlmann, Cloaking devices, electromagnetics wormholes and transformation optics, SIAM Review, 51, 2009, 3-33.

[28] F. Guevara Vasquez, G.W. Milton and D. Onofrei, Broadband exterior cloaking, Optics Express, 17, 2009, 14800-14805.

[29] N.J. Guliyev and V.E. Ismailov, A single hidden layer feedforward network with only one neuron in the hidden layer can approximate any univariate function, Neural Computation, 28, 2016, 1289-1304.

[30] Y. Hao and R. Mittra, FDTD Modeling of Metamaterials: Theory and Applications, Artech House Publishers, 2008.

[31] A. Harten, High resolution schemes for hyperbolic conservation laws, Journal of Computational Physics, 49, 1983, 357-393.

[32] J.S. Hesthaven and T. Warburton, Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications, Springer, New York, 2008.

[33] R. Hiptmair, Finite elements in computational electromagnetism. Acta Numerica, 11, 2002, 237-339.

[34] J. Hong, L. Ji and L. Kong, Energy-dissipation splitting finite-difference time-domain method for Maxwell equations with perfectly matched layers, Journal of Computational Physics, 269, 2014, 201-214.

[35] S. Hou and X.-D. Liu. Solutions of multi-dimensional hyperbolic systems of conservation laws by square entropy condition satisfying discontinuous Galerkin method, Journal of Scientific Computing, 31, 2007, 127-151.

[36] G.-S. Jiang and C.-W. Shu, On cell entropy inequality for discontinuous Galerkin methods, Mathematics of Computation, 62, 1994, 531-538.

[37] D.P. Kingma and J. Ba, Adam: a method for stochastic optimization, arXiv:1412.6980, 2014.

[38] D. Kriesel, A brief introduction to neural networks, http://www.dkriesel.com, 2007.

[39] R.V. Kohn, D. Onofrei, M.S. Vogelius and M.I. Weinstein, Cloaking via change of variables for the Helmholtz equation, SIAM Journal on Numerical Analysis, 63, 2010, 973-1016.

[40] K. Kontzialis, K. Panourgias and J. Ekaterinaris, A limiting approach for DG discretizations on mixed type meshes, Computer Methods in Applied Mechanics and Engineering, 285, 2015, 587-620.

[41] L. Krivodonova, J. Xin, J.-F. Remacle, N. Chevaugeon and J. E. Flaherty, Shock detection and limiting with discontinuous Galerkin methods for hyperbolic conservation laws, Applied Numerical Mathematics, 48, 2004, 323-338.

[42] I.E. Lagaris, A. Likas and D.I. Fotiadis, Artificial neural networks for solving ordinary and partial differential equations, IEEE Transactions on Neural Networks, 9, 1998, 987-1000.

[43] S. Lanteri, C. Scheid, Convergence of a discontinuous Galerkin scheme for the mixed time domain Maxwells equations in dispersive media, IMA Journal of Numerical Analysis, 33, 2013, 432–459.

[44] J.J. Lee, A mixed method for time-transient acoustic wave propagation in metamaterials, Journal of Scientific Computing, 84, 2020, 1-23.

[45] H. Lee and N. Lee, Wet-dry moving boundary treatment for Runge-Kutta discontinuous Galerkin shallow water equation model, KSCE Journal of Civil Engineering, 20, 2016, 978-989.

[46] U. Leonhardt, Optical conformal mapping, Science, 312, 2006, 1777-1780.

[47] J. Li, Development of discontinuous Galerkin methods for Maxwell's equations in metamaterials and perfectly matched layers, Journal of Computational and Applied Mathematics, 236, 2011, 950–961.

[48] J. Li and J.S. Hesthaven, Analysis and application of the nodal discontinuous Galerkin method for wave propagation in metamaterials, Journal of Computational Physics, 258, 2014, 915-930.

[49] J. Li and Y. Huang, Time-Domain Finite Element Methods for Maxwell's Equations in Metamaterials, Springer, Berlin Heidelberg, 2013.

[50] J. Li, Y. Huang and W. Yang, Well-posedness study and finite element simulation of time- domain cylindrical and elliptical cloaks, Mathematics of Computation, 84, 2015, 543-562.

[51] J. Li, Y. Huang, W. Yang and A. Wood, Mathematical analysis and time-domain finite element simulation of carpet cloak, SIAM Journal of Applied Mathematics, 74(4), 2014, 1136-1151.

[52] J. Li, C. Meng and Y. Huang, Improved analysis and simulation of a time-domain carpet cloak model, Computational Methods in Applied Mathematics, 19(2), 2019, 359-378.

[53] J. Li, C. Shi and C.-W. Shu, Optimal non-dissipative discontinuous Galerkin methods for Maxwell's equations in Drude metamaterials, Computers & Mathematics with Applications, 73, 2017, 1768-1780.

[54] J. Li, C.-W. Shu and W. Yang, Development and analysis of two new finite element schemes for a time-domain carpet cloak model, Advances in Computational Mathematics, submitted.

[55] J. Li, J.W. Waters and E.A. Machorro, An implicit leap-frog discontinuous Galerkin method for the time-domain Maxwell's equations in metamaterials, Computer Methods in Applied Mechanics and Engineering, 223-224, 2012, 43–54.

[56] W. Li, D. Liang and Y. Lin, Symmetric energy-conserved S-FDTD scheme for two-dimensional Maxwell's equations in negative index metamaterials, Journal of Scientific Computing, 69, 2016, 696-735.

[57] T. Lu, P. Zhang, W. Cai, Discontinuous Galerkin methods for dispersive and lossy Maxwells equations and PML boundary conditions, Journal of Computational Physics, 200, 2004, 549–580.

[58] A.L. Maas, A.Y. Hannun and A.Y. Ng, Rectifier nonlinearities improve neural network acoustic models, In Proc. International Conference on Machine Learning, 30, 2013.

[59] P. Monk, Finite Element Methods for Maxwell's Equations, Oxford University Press, 2003.

[60] S. Nicaise and J. Venel, A posteriori error estimates for a finite element approximation of transmission problems with sign changing coefficients, Journal of Computational and Applied Mathematics, 235, 2011, 4272-4282.

[61] A. B. Novikoff, On convergence proofs on perceptrons, Symposium on the Mathematical Theory of Automata, 12, 1962, 615-622.

[62] S. Osher, Convergence of generalized MUSCL schemes, SIAM Journal on Numerical Analysis, 22, 1985, 947-961.

[63] S. Osher and S. Chakravarthy, High resolution schemes and the entropy condition, SIAM Journal on Numerical Analysis, 21, 1984, 955-984.

[64] K.T. Panourgias and J.A. Ekaterinaris, A discontinuous Galerkin approach for high-resolution simulations of three-dimensional flows, Computer Methods in Applied Mechanics and Engineering, 299, 2016, 245-282.

[65] J.B. Pendry, D. Schurig and D.R. Smith, Controlling electromagnetic fields, Science, 312, 2006, 1780-1782.

[66] J. Qiu and C.-W. Shu, Runge-Kutta discontinuous Galerkin method using WENO limiters, SIAM Journal on Scientific Computing, 26, 2005, 907-929.

[67] J. Qiu and C.-W. Shu, A comparison of troubled-cell indicators for runge–kutta discontinuous galerkin methods using weighted essentially nonoscillatory limiters, SIAM Journal on Scientific Computing, 27(3), 995-19, 2005.

[68] A. Quarteroni, and V. Alberto, Numerical approximation of partial differential equations, Springer Series in Computational Mathematics, 23, 2008.

[69] D. Ray and J.S. Hesthaven, An artificial neural network as a troubled-cell indicator, Journal of Computational Physics, 367, 2018, 166-191.

[70] D. Ray and J.S. Hesthaven, Detecting troubled-cells on two-dimensional unstructured grids using a neural network, Journal of Computational Physics, 397, 2019, 108-845.

[71] W. Reed and T. Hill, Triangular mesh methods for neutron transport equation, Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory, Los Alamos, NM, 1973.

[72] F. Rosenblatt, The perceptron: A probabilistic model for information storage and organization in the brain, Psychological Review, 65, 1958, 386-408.

[73] K. Rudd and S. Ferrari, A constrained integration (cint) approach to solving partial differential equations using artificial neural networks, Neurocomputing, 155, 2015, 277-285.

[74] C. Shi, J. Li and C.-W. Shu, Discontinuous Galerkin methods for Maxwell's equations in Drude metamaterials on unstructured meshes, Journal of Applied Mathematics, 342, 2018, 147-163.

[75] C.-W. Shu, TVB uniformly high-order schemes for conservation laws, Mathematics of Computation, 49, 1987, 105-121.

[76] C.-W. Shu and S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, Journal of Computational Physics, 77, 1988, 439-471.

[77] C.-W. Shu and S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, II, Journal of Computational Physics, 83, 1989, 32-78.

[78] Z. Sun, S. Wang, L.-B. Chang, Y. Xing and D. Xiu, Convolution neural network shock detector for numerical Solution of Conservation Laws, Communications in Computational Physics, 28, 2020, 2075-2108.

[79] A. Suresh and H. Huynth, Accurate monotonicity-preserving schemes with Runge-Kutta time stepping, Computational Fluid Dynamics Conference, 13, 1997, 83-99.

[80] M.J. Vuik and J.K. Ryan, Automated parameters for troubled-cell indicators using outlier detection, SIAM Journal on Scientific Computing, 38, 2016, A84-A104.

[81] J. Wang, Z. Xie and C. Chen, Implicit DG method for time domain maxwell's equations involving metamaterials, Advances in Applied Mathematics and Mechanics, 7, 2015, 796-817.

[82] X. Wen, W.S. Don, Z. Gao and J.S. Hesthaven, An edge detector based on artificial neural network with application to hybrid compact-WENO finite difference scheme, Journal of Scientific Computing, 83, 2020.

[83] P. Woodward and P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, Journal of Computational Physics, 54, 1984, 115-173.

[84] Y. Xing and X. Zhang, Positivity-preserving well-balanced discontinuous Galerkin methods for the shallow water equations on unstructured triangular meshes, Journal of Computational Physics, 57, 2013, 19-41.

[85] Y. Xu and C.-W. Shu, Local discontinuous Galerkin methods for high-order time-dependent partial differential equations, Communications in Computational Physics, 7, 2010, 1-46.

[86] Z. Yang, L.-L. Wang, Z. Rong, B. Wang and B. Zhang, Seamless integration of global Dirichlet-to-Neumann boundary condition and spectral elements for transformation electromagnetics, Computer Methods in Applied Mechanics and Engineering, 301, 2016, 137-163.

[87] S. Zhang, C.G. Xia and N. Fang, Broadband acoustic cloak for ultrasound waves, Physical Review Letters, 106, 2011, 024301.

[88] J. Zhao and H. Tang, Runge-Kutta central discontinuous Galerkin methods for the special relativistic hydrodynamics, Communications in Computational Physics, 22, 2017, 643-682.

[89] H. Zhu, Y. Cheng and J. Qiu, A comparison of the performance of limiters for Runge-Kutta discontinuous Galerkin methods. Advances in Applied Mathematics and Mechanics, 5, 2013, 365-390.