# HIGH-ORDER NUMERICAL METHODS FOR TIME-DEPENDENT PROBLEMS WITH APPLICATIONS

---

A Dissertation

Presented to

the Faculty of the Department of Mathematics

University of Houston

---

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

---

By

Pei Yang

August 2015

# HIGH-ORDER NUMERICAL METHODS FOR TIME-DEPENDENT PROBLEMS WITH APPLICATIONS

Pei Yang

APPROVED:

Dr. Jingmei Qiu, Chairman
Department of Mathematics, University of Houston

Dr. Tsorng-Whay Pan,
Department of Mathematics, University of Houston

Dr. Jiwen He,
Department of Mathematics, University of Houston

Dr. Guoning Chen,
Department of Computer Science,
University of Houston

Dean, College of Natural Sciences and Mathematics

# HIGH-ORDER NUMERICAL METHODS FOR TIME-DEPENDENT PROBLEMS WITH APPLICATIONS

---

An Abstract of a Dissertation

Presented to

the Faculty of the Department of Mathematics

University of Houston

---

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

---

By

Pei Yang

August 2015

# Abstract

In this dissertation, several high-order numerical methods for solving time dependent problems are studied.

In the first part, a maximum principle preserving (MPP) finite-volume (FV) weighted essentially non-oscillatory (WENO) Runge Kutta (RK) scheme is proposed for convection-dominated problems. Such problems possess the maximum principle at the theoretical level, hence it is hoped that the numerical solution preserves the maximum principle. However, normal high-order FV WENO RK scheme doesn't satisfy such property. We propose a modified high-order FV WENO scheme by adding locally-parametrized flux limiters to maintain the maximum principle. In this work, for the first time under the finite-volume framework, such flux limiters are proved to maintain the high-order accuracy of the original WENO scheme for linear advection problems without any additional time-step restriction. And for general nonlinear convection-dominated problems, the flux limiters are proved to introduce up to $\mathcal{O}(\Delta x^3 + \Delta t^3)$ modification to the high-order temporal integrated flux in the original WENO scheme without extra time-step constraint. The MPP property of the proposed scheme is validated by several numerical tests.

In the second part, an integral deferred correction (InDC) method with adaptive non-polynomial basis is presented to solve stiff time dependent problems whose solutions contain initial or internal layers. Several non-polynomial bases with exponential functions are proposed, in the hope that the stiff layers in the solution can be better resolved by the exponentials than by polynomials. The stability and accuracy

properties of the non-polynomial InDC schemes are comparable to those of the polynomial InDC schemes. Finally, numerical test shows that the newly proposed InDC scheme outperforms the traditional polynomial-based scheme when it is applied to solve initial value problems with layers, in the sense that the former scheme takes fewer time steps than the latter one given the same error tolerance.

# Contents

# CHAPTER 1

# Introduction

## 1.1 Background

Many problems in computational fluid dynamics and other areas can be formulated in the form of time-dependent partial differential equations (PDEs). An example is the convection-diffusion problem, which mathematically takes the form

$$u_t + f(u)_x = a(u)_{xx}. \tag{1.1.1}$$

A tremendous amount of work has been done in designing schemes for numerically solving time-dependent PDEs, as it is hard to get analytical solutions in most cases. Available schemes include the finite element method, finite-difference method, finite volume method, and spectral method, etc..

In this dissertation, we mainly focus on the finite volume (FV) scheme for solving the convection-diffusion problem. Consider the one-dimensional problem (1.1.1) for $x \in [a, b]$. Suppose that we have the uniform spatial grids

$$a = x_{\frac{1}{2}} < x_{\frac{3}{2}} < \cdots < x_{N-\frac{1}{2}} < x_{N+\frac{1}{2}} = b,\ \Delta x = \frac{b-a}{N}, \tag{1.1.2}$$

with a computational cell and the corresponding cell center defined as

$$I_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}],\ x_j = \frac{1}{2}(x_{j-\frac{1}{2}} + x_{j+\frac{1}{2}}),\ j = 1, 2, \cdots, N. \tag{1.1.3}$$

Let $\bar{u}_j$ denote the approximation to the cell average of $u$ over cell $I_j$, i.e., $\bar{u}_j \approx \frac{1}{\Delta x}\int_{I_j} u(x)dx$. The FV scheme is designed by integrating equation (1.1.1) over each computational cell $I_j$ and then dividing it by $\Delta x$, which gives

$$\frac{d\bar{u}_j}{dt} = -\frac{1}{\Delta x}(\hat{H}^C_{j+\frac{1}{2}} - \hat{H}^C_{j-\frac{1}{2}}) + \frac{1}{\Delta x}(\hat{H}^D_{j+\frac{1}{2}} - \hat{H}^D_{j-\frac{1}{2}}), \tag{1.1.4}$$

where $\hat{H}^C_{j+\frac{1}{2}}$ and $\hat{H}^D_{j+\frac{1}{2}}$ are the numerical fluxes for convection and diffusion terms respectively. $\hat{H}^C_{j+\frac{1}{2}}$ and $\hat{H}^D_{j+\frac{1}{2}}$ can be understood as approximations to $f(u(x,t))|_{x=x_{j+\frac{1}{2}}}$ and $(a(u(x,t))_x)|_{x=x_{j+\frac{1}{2}}}$ respectively, which can be approximated via reconstructions from neighbouring cell averages of $I_j$, e.g., $\{\bar{u}_{j-r}, \cdots, \bar{u}_{j+s}\}$ (which will thereafter

be called the reconstruction stencil) for a $(r+s+1)$-th order accurate reconstruction. Among all the different reconstruction methods, the weighted essentially non-oscillatory (WENO) (see [24]) reconstruction procedure is well known due to its robustness in adaptively selecting appropriate reconstruction stencil for stability, high-order accuracy, as well as a non-oscillatory resolution of shocks. One should be aware that using information from appropriate directions plays an important role in designing numerical fluxes for convection or convection-dominated problems. The reader may refer to [23] for more details about reconstruction procedures, as well as for an exhaustive study on how different numerical fluxes have been developed in the last few decades. The procedures of reconstructing numerical fluxes used in this dissertation will be presented in details in Chapter 2, which can also be found in [22].

With the right hand-side of (1.1.4) properly addressed as just discussed, we discretize the time derivative on the left hand side of (1.1.4) by ODE solvers. Usually an ODE system is obtained after spatial discretization is performed on a time-dependent PDE problem. In general, these ODE problems can be written as

$$\mathbf{u}' = \mathbf{f}(t, \mathbf{u}), \tag{1.1.5}$$

where $\mathbf{f}$ is a vector function of $\mathbf{u} = (\bar{u}_1, \cdots, \bar{u}_M)$.

The total variation diminishing (TVD) Runge-Kutta (RK) ([21]) scheme is widely-used for convection dominated problems and is given by

$$\mathbf{u}^{(1)} = \mathbf{u}^n + \Delta t \mathbf{f}(t_n, \mathbf{u}^n),$$

$$\mathbf{u}^{(2)} = \mathbf{u}^n + \Delta t(\frac{1}{4}\mathbf{f}(t_n, \mathbf{u}^n) + \frac{1}{4}\mathbf{f}(t_n + \Delta t, \mathbf{u}^{(1)})), \tag{1.1.6}$$
$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t(\frac{1}{6}\mathbf{f}(t_n, \mathbf{u}^n) + \frac{1}{6}\mathbf{f}(t_n + \Delta t, \mathbf{u}^{(1)}) + \frac{2}{3}\mathbf{f}(t_n + \frac{1}{2}\Delta t, \mathbf{u}^{(2)})).$$

We apply the TVD RK scheme (1.1.6) on the semi-discretized scheme (1.1.4) and get a fully-discretized scheme

$$\bar{u}_j^{n+1} = \bar{u}_j^n - \lambda(\hat{H}^{rk}_{j+\frac{1}{2}} - \hat{H}^{rk}_{j-\frac{1}{2}}), \quad j = 1, 2, \cdots, N \tag{1.1.7}$$

with $\lambda = \frac{\Delta t}{\Delta x}$ and

$$\hat{H}^{rk}_{j+\frac{1}{2}} = \frac{1}{6}(\hat{H}^{C,n}_{j+\frac{1}{2}} - \hat{H}^{D,n}_{j+\frac{1}{2}}) + \frac{1}{6}(\hat{H}^{C,(1)}_{j+\frac{1}{2}} - \hat{H}^{D,(1)}_{j+\frac{1}{2}}) + \frac{2}{3}(\hat{H}^{C,(2)}_{j+\frac{1}{2}} - \hat{H}^{D,(2)}_{j+\frac{1}{2}}). \tag{1.1.8}$$

Here $\hat{H}^{C,(s)}_{j+\frac{1}{2}}$, $\hat{H}^{D,(s)}_{j+\frac{1}{2}}$ $(s = 1, 2)$ are the numerical fluxes at the intermediate stages in the TVD RK scheme (1.1.6).

Besides the convection-diffusion problem, there are many other physical phenomena of great importance for applications described by a multi-scale system of differential equations of the form

$$u' = f(u) + \frac{1}{\epsilon}g(u), \tag{1.1.9}$$

with $\epsilon > 0$ being the stiffness parameter. Systems of such form arise in various application problems such as chemical reaction, mechanics, hyperbolic systems with relaxation where a method of lines approach is used, etc.. In general, in order to treat and handle problems of this form, it is important to develop suitable numerical

methods that work in an accurate, stable and efficient way. So far, many numerical solvers for (1.1.9) have been developed, including forward Euler (FE), backward Euler (BE), Runge Kutta (RK), implicit-explicit (IMEX), integral deferred correction (InDC) ([27]), spectral deferred correction (SDC) ([29]) and so on.

Both FE and BE methods are low-order (first-order) schemes. It is easy to implement FE scheme due to its explicit property, however FE scheme suffers from the small time-step constraint because its stability region is small, and this disadvantage renders solving stiff ODE problems almost impossible. For BE scheme, the stability region is open in the complex plane, which makes it a good choice for solving stiff ODEs. However BE scheme is implicit, so if the right hand side in (1.1.9) is nonlinear with respect to $u$ then one needs to (numerically) solve nonlinear equations to get the numerical solution at each time-step, which might be a very costly task.

InDC method has been gaining more and more popularity ([35], [36], [27], [28]) in recent years. The derivation for InDC scheme is quite straightforward, compared with the complicated algebraic manipulations for deriving high-order Runge Kutta schemes, and it can effectively improve order of accuracy just by relatively simple iterations. In this dissertation, we make use of the InDC framework to construct a new family of time integrators. We review the InDC scheme as follows. For the initial value problem

$$u' = f(t,u), \quad t \in [0,1], \quad u(0) = u_0, \tag{1.1.10}$$

let the discretization for the time domain $[0, T]$ be

$$0 = t_1 < t_2 < \cdots < t_n < \cdots < t_N = T, \tag{1.1.11}$$

and let the numerical approximation to $\{u(t_n)\}_{n=1}^{n=N}$ be $\{u_n\}_{n=1}^{n=N}$. Then each interval $I_n = [t_{n-1}, t_n]$ is uniformly discretized into subintervals as

$$t_{n-1} = t_{n,0} < t_{n,1} < \cdots < t_{n,m} < \cdots < t_{n,M} = t_n. \tag{1.1.12}$$

We let $H_n = t_n - t_{n-1}$ and $h_n = \frac{H_n}{M}$. For a given interval $I_n$, the numerical solutions on the grid points $\{t_{n,m}\}_{m=0}^{m=M}$ may be obtained with a low-order scheme like FE, BE, or a low-order IMEX scheme. Let $\eta^{[0]} = (\eta_0^{[0]}, \cdots, \eta_M^{[0]})$ denote the obtained numerical solutions. This step is call the *prediction step*.

The *correction loop* in the InDC scheme starts with considering the *error function*

$$e^{(k-1)}(t) = u(t) - \eta^{(k-1)}(t) \tag{1.1.13}$$

where $\eta^{k-1}(t)$ is the polynomial function that interpolates $\eta^{[k-1]}$, and $k$ denotes the $k$-th correction step. The error function satisfies the *error equation* (see, e.g., [28])

$$(e^{(k-1)}(t) + \int_0^t \epsilon^{(k-1)}(\tau)d\tau)'(t) = f(t, \eta^{(k-1)}(t) + e^{(k-1)}(t)) - f(t, \eta^{(k-1)}(t)), \tag{1.1.14}$$

where $\epsilon^{(k-1)}(t) = (\eta^{(k-1)})'(t) - f(t, \eta^{(k-1)}(t))$ is the *residual function*. The error equation may be numerically solved by a low-order scheme, with initial value 0, over the grid points $\{t_{n,m}\}_{m=0}^{m=M}$. Let the numerical solutions to $\{e^{(k-1)}(t_{n,m})\}_{m=0}^{M}$ be

$\{\delta_m\}_{m=0}^{M}$. For example, applying FE scheme to solve (1.1.14) gives

$$\delta_{m+1}^{[k]} = \delta_m^{[k]} + h(f(\tau_m, \eta_m^{[k-1]} + \delta_m^{[k]}) - f(\tau_m, \eta_m^{[k-1]})) - \int_{\tau_m}^{\tau_{m+1}} \epsilon^{(k-1)}(t)\, dt,\ m = 0, \ldots, M-1, \tag{1.1.15}$$

with

$$\int_{t_{n,m}}^{t_{n,m+1}} \epsilon^{(k-1)}(t)\, dt = \eta_{m+1}^{[k-1]} - \eta_m^{[k-1]} - \int_{t_{n,m}}^{t_{n,m+1}} f(t, \eta^{(k-1)}(t))dt$$

where $\int_{t_{n,m}}^{t_{n,m+1}} f(t, \eta^{(k-1)}(t))dt$ can be approximated by replacing the integrand by the polynomial function that interpolates $\{f(t_{n,m}, \eta_m^{(k-1)})\}_{m=0}^{M}$.

After $\{\delta_m\}_{m=0}^{M}$ are obtained, $\eta^{[k-1]}$ is updated to be $\eta^{[k]} = \eta^{[k-1]} + \delta^{[k]}$ with $\delta^{[k]} = (\delta_0, \delta_1, \cdots, \delta_M)$, and this is called a *correction loop*, after which one may continue to apply the same procedure discussed above for several more times to improve the accuracy of $(\eta_0^{[k]}, \cdots, \eta_M^{[k]})$ as approximations to $(u(t_{n,0}), \cdots, u(t_{n,M}))$.

Finally after $J$ correction loops, $\eta_M^{[J]}$ is taken to approximate $u(t_n)$, i.e., $u_n = \eta_M^{[J]}$. With $\eta_M^{[J]}$ as an initial value, one proceeds to apply the InDC scheme for the next interval $I_{n+1}$. More details about the InDC scheme can be found in [27].

Another ODE solver quite similar to InDC is spectral deferred correction (SDC) method, which is proposed in [29]. It is called SDC method because it uses spectral points (e.g., Gauss points) rather than uniformly distributed points to divide each time interval $I_n$ into subintervals. Apart from that, SDC scheme and InDC scheme are exactly the same.

## 1.2 Main Topics of the Dissertation

This dissertation mainly addresses two topics related to time-dependent problems:

- In Chapter 2, a high-order maximum principle preserving (MPP) finite volume method for convection dominated problems is proposed and tested on several problems in fluid dynamics. Recall that at theoretical level, the convection-diffusion problem

$$u_t + f(u)_x = a(u)_{xx}, \quad x \in [a,b],\ t \in [0,T], \quad u(0) = u_0(x), \tag{1.2.16}$$

with $a'(u) > 0$ satisfies the maximum principle:

$$\text{if } u_M = \max_x u_0(x), u_m = \min_x u_0(x),\ \text{then } u(x,t) \in [u_m, u_M]. \tag{1.2.17}$$

So it is desirable that numerical solutions should preserve the maximum principle in the discrete form:

$$\text{if } u_M = \max_x u_0(x), u_m = \min_x u_0(x),\ \text{then } \bar{u}_j^n \in [u_m, u_M] \text{ for any } n,\ j, \tag{1.2.18}$$

where $n$ is the index for temporal discretization, i.e., $n$ corresponds to time $t_n$ and $j$ is for spatial discretization, i.e., $j$ corresponds to interval $I_n$. Unfortunately, the high-order FV RK scheme doesn't have the MPP property. In the last few years, a lot of work ([12], [14], [15], [16], etc.) has been done to modify the existing high-order numerical schemes to maintain maximum principle. For

example, in [16] an MPP finite-difference scheme was presented for hyperbolic conservation law problems. In this dissertation, we incorporate a similar strategy to modify the FV RK scheme so that it satisfies the maximum principle. The basic idea is to replace the numerical flux (1.1.8) with a new flux

$$\tilde{H}^{rk}_{j+\frac{1}{2}} = \theta_{j+\frac{1}{2}} \hat{H}^{rk}_{j+\frac{1}{2}} + (1 - \theta_{j+\frac{1}{2}}) \hat{h}_{j+\frac{1}{2}}, \tag{1.2.19}$$

where $\hat{h}_{j+\frac{1}{2}}$ is a low-order flux (e.g., Lax-Friedrichs flux) satisfying maximum principle and $\theta_{j+\frac{1}{2}}$ is the local weight parameter, so that the modified scheme

$$\bar{u}^{n+1}_j = \bar{u}^n_j - \lambda(\tilde{H}^{rk}_{j+\frac{1}{2}} - \tilde{H}^{rk}_{j-\frac{1}{2}}) \tag{1.2.20}$$

possesses MPP property. The strategy for determining the local parameter $\theta_{j+\frac{1}{2}}$ and other related issues will be addressed in Chapter 2.

- In Chapter 3, an integral deferred correction method with adaptive non-polynomial basis for stiff ODE problems is proposed and tested. Recall that in the InDC scheme discussed in the previous section, there is the term

$$\int_{t_{n,m}}^{t_{n,m+1}} \epsilon^{(k-1)}(t)\, dt = \eta^{[k-1]}_{m+1} - \eta^{[k-1]}_m - \int_{t_{n,m}}^{t_{n,m+1}} f(t, \eta^{(k-1)}(t))dt \tag{1.2.21}$$

in which the integral $\int_{t_{n,m}}^{t_{n,m+1}} f(t, \eta^{(k-1)}(t))dt$ needs to be numerically computed by replacing the integrand by the polynomial function that interpolates $\{f(t_{n,m}, \eta^{(k-1)}_m)\}_{m=0}^M$. Traditionally, polynomial basis (e.g., Lagrangian

polynomials) is used to do the interpolation. In this dissertation, we investigated the possibility of using several non-polynomial basis to interpolate $\{f(t_{n,m}, \eta_m^{(k-1)})\}_{m=0}^{M}$. Specifically, the following bases

1. $\left\{y : y|_{I_n} \in span\{e^{\lambda\tau}, \tau, \tau^2, \cdots, \tau^M\}, \forall n\right\}$,
2. $\left\{y : y|_{I_n} \in span\{e^{\lambda\tau}(1, \tau, \tau^2, \cdots, \tau^M)\}, \forall n\right\}$,
3. $\left\{y : y|_{I_n} \in span\{1, \tau, \tau^2, \cdots, \tau^M, e^{\lambda\tau}\}, \forall n\right\}$

with $\tau = \frac{t-(t_{n-1}+t_n)/2}{H_n/2}$ are proposed as alternatives for polynomial basis. The incorporation of the function $e^{\lambda\tau}$ is motivated by the fact that many stiff ODE problems have structures in the form of $e^{ct}$ with $|c|$ very large. The scheme is designed so that $e^{\lambda\tau}$ can contribute to effectively capture such structures. Furthermore, it is proposed that the parameter $\lambda$ should be adaptively chosen according to the local structure of the solution. The idea of adopting local parameter $\lambda_n$ for non-polynomial basis is earlier presented in designing non-polynomial based discontinuous Garlerkin scheme in [39], in which a simple strategy for computing the local parameter $\lambda_n$ was discussed and the good performance of the strategy was verified by numerical tests. We therefore adopt the same strategy in our work, which will be presented in Chapter 3 of this dissertation.

In Chapter 4, the conclusions from this dissertation are presented.

# CHAPTER 2

# High-Order MPP Finite Volume Method

## 2.1 Introduction

Recently, there is a growing interest in designing high-order maximum principle preserving (MPP) schemes for solving scalar convection-dominated problems [17, 16, 15, 9, 10, 12] and positivity preserving schemes for compressible Euler and Navier-Stokes equations [8, 13, 11, 18]. The motivation of this family of work arises from the observation that many existing high-order conservative methods break down when they are applied to simulate fluid dynamics in extreme cases such as near-vacuum state. To illustrate the purpose of the family of the MPP methods, we shall consider

the solution to the following problem

$$u_t + f(u)_x = a(u)_{xx}, \qquad u(x,0) = u_0(x), \tag{2.1.1}$$

with $a'(u) > 0$. The solution to (2.1.1) satisfies the maximum principle, i.e.,

$$\text{if } u_M = \max_x u_0(x), u_m = \min_x u_0(x), \text{ then } u(x,t) \in [u_m, u_M]. \tag{2.1.2}$$

Within the high-order finite volume (FV) Runge-Kutta (RK) weighted essentially non-oscillatory (WENO) framework, we would like to maintain a discrete form of (2.1.2):

$$\text{if } u_M = \max_x u_0(x), u_m = \min_x u_0(x), \text{ then } \bar{u}_j^n \in [u_m, u_M] \text{ for any } n,\ j, \tag{2.1.3}$$

where $\bar{u}_j^n$ approximates the cell average of the exact solution with high-order accuracy on a given $j$th spatial interval at time $t^n$.

Efforts for designing MPP high-order schemes to solve (2.1.1) can be found in recent work by Zhang et al. [16, 19], as a continuous research effort to design high-order FV and discontinuous Galerkin (DG) MPP schemes based on a polynomial rescaling limiter on the reconstructed (for FV) or representing (for DG) polynomials [17]. This approach requires the updated *cell average* to be written as a convex combination of some local quantities within the range $[u_m, u_M]$. For convection-diffusion problems which do not have a finite speed of propagation, it is difficult to generalize such approach to design MPP schemes that are higher than third-order accurate.

In [9], an alternative approach via a parametrized flux limiter, developed earlier by Xu et al. [15, 12], is proposed for the finite-difference (FD) RK WENO method in solving convection diffusion equations. The flux limiter is applied to convection and diffusion fluxes together to achieve (2.1.3) for the approximated point values in the finite-difference framework. In this chapter, we try to apply the MPP flux limiters to high-order FV RK WENO methods to maintain (2.1.3) with efficiency. Furthermore, we provide some theoretical analysis on the preservation of high-order accuracy for the proposed flux limiter in FV framework. Finally, we remark that our current focus is on convection-dominated diffusion problems for which explicit temporal integration proves to be efficient. For the regime of medium to large diffusion, where implicit temporal integration is needed for simulation efficiency, we refer to earlier work in [5, 3, 2, 4] and references therein for the construction of the MPP schemes with finite element framework. The generalization of the current flux limiter is not yet available and is subject to future investigation.

The MPP methods in [17, 15, 12] are designed based on the observation that first-order monotone schemes in general satisfy MPP property (2.1.3) with proper Courant-Friedrichs-Lewy (CFL) numbers, while regular high-order conservative schemes often fail to maintain (2.1.3). The MPP flux limiting approach is to seek a linear combination of the first-order monotone flux with the high-order flux, in the hope of that such combination can achieve both MPP property and high-order accuracy under certain conditions, e.g., some mild time-step constraint. This line of approach is proven to be successful in [12, 9] for the FD RK WENO schemes and

it is later generalized to the high-order semi-Lagrangian WENO method for solving the Vlasov-Poisson system [14]. A positivity preserving flux limiting approach is developed in [13] to ensure positivity of the computed density and pressure for compressible Euler simulations. Technically, the generalization of such MPP flux limiters from FD WENO [9] to FV WENO method is rather straightforward. Taking into the consideration that FV method offers a more natural framework for mass conservation and flexibility in handling irregular computational domain, we propose to apply the MPP flux limiters to the high-order FV RK WENO method to solve (2.1.1). The proposed flux limiting procedure is rather easy to implement even with the complexity of the flux forms in multi-dimensional FV computation. Moreover, a general theoretical proof on preserving both MPP and high-order accuracy without additional time-step constraint can be done for FV methods when solving a linear advection equation; such result does not hold for high-order FD schemes [12].

In our work, for the first time, we establish a general proof that, there is no further time-step restriction, besides the CFL condition under the linear stability requirement, to preserve high-order accuracy when the high-order flux is limited toward an upwind first-order flux for solving linear advection problem, when the parametrized flux limiters are applied to FV RK WENO method. In other words, both the MPP property and high-order accuracy of the original scheme can be maintained without additional time-step constraint. For a general nonlinear convection problem, we prove that the flux limiter preserves up to third-order accuracy and the discrete maximum principle with no further CFL restriction. This proof relies on tedious

Taylor expansions, and it is difficult to generalize it to results with higher order accuracy (fourth order or higher). On the other hand, such analysis can be extended to a convection-dominated diffusion problem as done in [9]. Furthermore, numerical results indicate that mild CFL restriction is needed for the MPP flux limiting finite volume scheme without sacrificing accuracy.

This chapter is organized as follows. In Section 2, we provide the numerical algorithm of the high-order FV RK WENO scheme with MPP flux limiters for the one-dimensional problem and in Section 3 the scheme is generalized to two-dimensional case. In Section 4, theoretical analysis is given for a linear advection problem and general nonlinear problems. Numerical experiments are demonstrated in Section 5.

## 2.2 MPP FV Method for One-Dimensional Problems

In this section, we propose a high-order FV scheme for the one-dimensional convection-diffusion equation. In the proposed scheme, the high-order WENO reconstruction of flux is used for the convection term, while a high-order compact reconstruction of flux is proposed for the diffusion term.

For simplicity, we first consider a one-dimensional (1D) case. The following uniform spatial discretization is used for a 1D bounded domain $[a, b]$,

$$a = x_{\frac{1}{2}} < x_{\frac{3}{2}} < \cdots < x_{N-\frac{1}{2}} < x_{N+\frac{1}{2}} = b,\ \Delta x = \frac{b-a}{N}. \tag{2.2.1}$$

with the computational cell and cell center defined as

$$I_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}],\ x_j = \frac{1}{2}(x_{j-\frac{1}{2}} + x_{j+\frac{1}{2}}),\ j = 1, 2, \cdots, N. \tag{2.2.2}$$

Let $\bar{u}_j$ denote approximation to the cell average of $u$ over cell $I_j$. The FV scheme is designed by integrating equation (2.1.1) over each computational cell $I_j$ and then dividing it by $\Delta x$,

$$\frac{d\bar{u}_j}{dt} = -\frac{1}{\Delta x}(\hat{H}^C_{j+\frac{1}{2}} - \hat{H}^C_{j-\frac{1}{2}}) + \frac{1}{\Delta x}(\hat{H}^D_{j+\frac{1}{2}} - \hat{H}^D_{j-\frac{1}{2}}), \tag{2.2.3}$$

where $\hat{H}^C_{j+\frac{1}{2}}$ and $\hat{H}^D_{j+\frac{1}{2}}$ are the numerical fluxes for convection and diffusion terms respectively.

For the convection term, one can adopt any monotone flux. For example, in our simulations, we use the Lax-Friedrichs flux

$$\hat{H}^C_{j+\frac{1}{2}}(u^-_{j+\frac{1}{2}}, u^+_{j+\frac{1}{2}}) = \frac{1}{2}\big(f(u^-_{j+\frac{1}{2}}) + \alpha u^-_{j+\frac{1}{2}}\big) + \frac{1}{2}\big(f(u^+_{j+\frac{1}{2}}) - \alpha u^+_{j+\frac{1}{2}}\big),\ \alpha = \max_{u_m \le u \le u_M} |f'(u)|. \tag{2.2.4}$$

Here $u^-_{j+\frac{1}{2}} \doteq P(x_{j+\frac{1}{2}})$, where $P(x)$ is obtained by reconstructing a $(2k+1)^{th}$ order polynomial whose averages agree with those in a left-biased stencil $\{\bar{u}_{j-k}, \cdots, \bar{u}_{j+k}\}$,

$$\frac{1}{\Delta x}\int_{I_l} P(x)dx = \bar{u}_l,\ l = j-k, \cdots, j+k.$$

The reconstruction procedure for $u^+_{j+\frac{1}{2}}$ can be done similarly from a right-biased stencil. To suppress oscillation around discontinuities and maintain high-order accuracy

around smooth regions of the solution, the WENO mechanism can be incorporated in the reconstruction. Details of such procedure can be found in [1].

For the diffusion term, we propose the following *compact* reconstruction strategy for approximating fluxes at cell boundaries $a(u)_x|_{x_{j+\frac{1}{2}}}$. Without loss of generality, we consider a fourth order reconstruction, while similar strategies can be extended to schemes with arbitrary high-order. Below we let $u_j$ denote approximation to the point values of $u$ at $x_j$.

1. Reconstruct $\{u_l\}_{l=j-1}^{j+2}$ from the cell averages $\{\bar{u}_l\}_{l=j-1}^{j+2}$ by constructing a cubic polynomial $P(x)$, such that

$$\frac{1}{\Delta x}\int_{I_l} P(x)dx = \bar{u}_l,\ l = j-1, \cdots, j+2.$$

   Then $u_l = P(x_l)$, $l = j-1, \cdots j+2$. We use $\mathcal{R}_1$ to denote such reconstruction procedure,

$$(u_{j-1}, u_j, u_{j+1}, u_{j+2}) = \mathcal{R}_1(\bar{u}_{j-1}, \bar{u}_j, \bar{u}_{j+1}, \bar{u}_{j+2}).$$

   As a reference, the reconstruction formulas for $\mathcal{R}_1$ are provided below,

$$u_{j-1} = \frac{11}{12}\bar{u}_{j-1} + \frac{5}{24}\bar{u}_j - \frac{1}{6}\bar{u}_{j+1} + \frac{1}{24}\bar{u}_{j+2}, \quad u_j = -\frac{1}{24}\bar{u}_{j-1} + \frac{13}{12}\bar{u}_j - \frac{1}{24}\bar{u}_{j+1},$$
$$u_{j+1} = -\frac{1}{24}\bar{u}_j + \frac{13}{12}\bar{u}_{j+1} - \frac{1}{24}\bar{u}_{j+2}, \quad u_{j+2} = \frac{1}{24}\bar{u}_{j-1} - \frac{1}{6}\bar{u}_j + \frac{5}{24}\bar{u}_{j+1} + \frac{11}{12}\bar{u}_{j+2}.$$

2. Construct an interplant $Q(x)$ such that $Q(x_l) = a(u_l),\ l = j-1, \cdots, j+2$.

Then let $\hat{H}^D_{j+\frac{1}{2}} = Q'(x)|_{x_{j+\frac{1}{2}}}$. Such procedure is denoted as

$$\hat{H}^D_{j+\frac{1}{2}} = \mathcal{R}_2(a(u_{j-1}), a(u_j), a(u_{j+1}), a(u_{j+2})).$$

As a reference, we provide the formula for $\mathcal{R}_2$ below

$$\hat{H}^D_{j+\frac{1}{2}} = \frac{1}{24}a(u_{j-1}) - \frac{9}{8}a(u_j) + \frac{9}{8}a(u_{j+1}) - \frac{1}{24}a(u_{j+2}).$$

**Remark 2.2.1.** The reconstruction processes for $\mathcal{R}_1$ and $\mathcal{R}_2$ operators are designed such that $\hat{H}^D_{j+\frac{1}{2}}$ is reconstructed from a compact stencil with a given order of accuracy. Because of such design, for the linear diffusion term $a(u) = u$, $\mathcal{R}_1$ and $\mathcal{R}_2$ can be combined and the strategy above turns out to be a classical fourth order central difference from a five-cell stencil with

$$\hat{H}^D_{j+\frac{1}{2}} = \frac{1}{\Delta x}(\frac{1}{2}\bar{u}_{j-1} - \frac{15}{12}\bar{u}_j + \frac{15}{12}\bar{u}_{j+1} - \frac{1}{12}\bar{u}_{j+2}).$$

If each of $u_l$ ($l = j-1, \cdots j+2$) in Step 1 is reconstructed from symmetrical stencils (having the same number of cells from left and from right), the reconstruction of $\hat{H}^D_{j+\frac{1}{2}}$ will depend on a much wider stencil $\{u_{j-3}, \cdots u_{j+4}\}$. Such non-compact way of reconstructing numerical fluxes for diffusion terms will introduce some numerical instabilities when approximating nonlinear diffusion terms in our numerical tests, whereas the proposed compact strategy does not encounter such difficulty.

We use the following third-order total variation diminishing (TVD) RK method

[6] for the time discretization of (2.2.3), which reads

$$\begin{aligned} u^{(1)} &= \bar{u}^n + \Delta t L(\bar{u}^n), \\ u^{(2)} &= \bar{u}^n + \Delta t(\frac{1}{4}L(\bar{u}^n) + \frac{1}{4}L(u^{(1)})), \\ \bar{u}^{n+1} &= \bar{u}^n + \Delta t(\frac{1}{6}L(\bar{u}^n) + \frac{1}{6}L(u^{(1)}) + \frac{2}{3}L(u^{(2)})), \end{aligned} \tag{2.2.5}$$

where $L(\bar{u}^n)$ denotes the right hand side of equation (2.2.3). Here $\bar{u}^n$ and $u^{(s)}$, $s = 1, 2$ denote the numerical solution of $u$ at time $t^n$ and corresponding RK stages. The fully discretized scheme (2.2.5) can be rewritten as

$$\bar{u}_j^{n+1} = \bar{u}_j^n - \lambda(\hat{H}^{rk}_{j+\frac{1}{2}} - \hat{H}^{rk}_{j-\frac{1}{2}}) \tag{2.2.6}$$

with $\lambda = \frac{\Delta t}{\Delta x}$ and

$$\hat{H}^{rk}_{j+\frac{1}{2}} = \frac{1}{6}(\hat{H}^{C,n}_{j+\frac{1}{2}} - \hat{H}^{D,n}_{j+\frac{1}{2}}) + \frac{1}{6}(\hat{H}^{C,(1)}_{j+\frac{1}{2}} - \hat{H}^{D,(1)}_{j+\frac{1}{2}}) + \frac{2}{3}(\hat{H}^{C,(2)}_{j+\frac{1}{2}} - \hat{H}^{D,(2)}_{j+\frac{1}{2}}).$$

Here $\hat{H}^{C,(s)}_{j+\frac{1}{2}}$, $\hat{H}^{D,(s)}_{j+\frac{1}{2}}$ $(s = 1, 2)$ are the numerical fluxes at the intermediate stages in the RK scheme (2.2.5).

It has been known that the numerical solutions from schemes with a first-order monotone flux for the convection term together with a first-order flux for the diffusion term satisfy the maximum principle, if the time-step is small enough [19]. However, if the numerical fluxes are of high-order such as the one from the reconstruction process proposed above, the MPP property for the numerical solutions does not necessarily hold under the same time-step constraint. Next we apply the parametrized flux

limiters proposed in [12] to the scheme (2.2.6) to preserve the discrete maximum principle (2.1.3).

We modify the numerical flux $\hat{H}^{rk}_{j+\frac{1}{2}}$ in equation (2.2.6) with

$$\tilde{H}^{rk}_{j+\frac{1}{2}} = \theta_{j+\frac{1}{2}}\hat{H}^{rk}_{j+\frac{1}{2}} + (1-\theta_{j+\frac{1}{2}})\hat{h}_{j+\frac{1}{2}}, \tag{2.2.7}$$

by carefully seeking local parameters $\theta_{j+\frac{1}{2}}$, such that the numerical solutions enjoy the MPP property yet $\theta_{j+\frac{1}{2}}$ is as close to 1 as possible. In other words, $\tilde{H}^{rk}_{j+\frac{1}{2}}$ is as close to the original high-order flux $\hat{H}^{rk}_{j+\frac{1}{2}}$ as possible. Here $\hat{h}_{j+\frac{1}{2}}$ denotes the first-order flux for convection and diffusion terms, using which in the scheme (2.2.3) with a forward Euler time discretization guarantees the maximum principle of numerical solutions. For example, we can take

$$\hat{h}_{j+\frac{1}{2}} = \hat{h}^{C}_{j+\frac{1}{2}} - \hat{h}^{D}_{j+\frac{1}{2}} = \frac{1}{2}\big(f(\bar{u}_j) + \alpha\bar{u}_j\big) + \frac{1}{2}\big(f(\bar{u}_{j+1}) - \alpha\bar{u}_{j+1}\big) - \frac{a(\bar{u}_{j+1}) - a(\bar{u}_j)}{\Delta x}$$

with $\alpha = \max_{u_m \le u \le u_M}|f'(u)|$. The goal of the procedures outlined below is to adjust $\theta_{j+\frac{1}{2}}$, so that with the modified flux $\tilde{H}^{rk}_{j+\frac{1}{2}}$, the numerical solutions satisfy the maximum principle,

$$u_m \le \bar{u}^n_j - \lambda(\tilde{H}^{rk}_{j+\frac{1}{2}} - \tilde{H}^{rk}_{j-\frac{1}{2}}) \le u_M, \quad \forall j. \tag{2.2.8}$$

Detailed procedures in decoupling the above inequalities have been intensively discussed in our previous work, e.g., [12]. Below we only briefly describe the computational algorithm for the proposed limiter.

Let $F_{j+\frac{1}{2}} \doteq \hat{H}^{rk}_{j+\frac{1}{2}} - \hat{h}_{j+\frac{1}{2}}$ and

$$\Gamma^M_j \doteq u_M - (\bar{u}^n_j - \lambda(\hat{h}_{j+\frac{1}{2}} - \hat{h}_{j-\frac{1}{2}})),\ \Gamma^m_j \doteq u_m - (\bar{u}^n_j - \lambda(\hat{h}_{j+\frac{1}{2}} - \hat{h}_{j-\frac{1}{2}})).$$

The MPP property is satisfied with the modified flux (2.2.7) when the following inequalities are hold,

$$\lambda\theta_{j-\frac{1}{2}}F_{j-\frac{1}{2}} - \lambda\theta_{j+\frac{1}{2}}F_{j+\frac{1}{2}} - \Gamma^M_j \le 0, \tag{2.2.9}$$

$$\lambda\theta_{j-\frac{1}{2}}F_{j-\frac{1}{2}} - \lambda\theta_{j+\frac{1}{2}}F_{j+\frac{1}{2}} - \Gamma^m_j \ge 0. \tag{2.2.10}$$

We first consider the inequality (2.2.9). We seek a local pair of numbers $(\Lambda^M_{-\frac{1}{2},I_j}, \Lambda^M_{+\frac{1}{2},I_j})$ such that (1) $\Lambda^M_{\pm\frac{1}{2},I_j} \in [0,1]$ and is as close to 1 as possible, (2) for any $\theta_{j-\frac{1}{2}} \in [0, \Lambda^M_{-\frac{1}{2},I_j}]$, $\theta_{j+\frac{1}{2}} \in [0, \Lambda^M_{+\frac{1}{2},I_j}]$, the inequality (2.2.9) holds. The inequality (2.2.9) can be decoupled based on the following four different cases:

(a) If $F_{j-\frac{1}{2}} \le 0$ and $F_{j+\frac{1}{2}} \ge 0$, then $(\Lambda^M_{-\frac{1}{2},I_j}, \Lambda^M_{+\frac{1}{2},I_j}) = (1,1)$.

(b) If $F_{j-\frac{1}{2}} \le 0$ and $F_{j+\frac{1}{2}} < 0$, then $(\Lambda^M_{-\frac{1}{2},I_j}, \Lambda^M_{+\frac{1}{2},I_j}) = (1, \min(1, \frac{\Gamma^M_j}{-\lambda F_{j+\frac{1}{2}}}))$.

(c) If $F_{j-\frac{1}{2}} > 0$ and $F_{j+\frac{1}{2}} \ge 0$, then $(\Lambda^M_{-\frac{1}{2},I_j}, \Lambda^M_{+\frac{1}{2},I_j}) = (\min(1, \frac{\Gamma^M_j}{\lambda F_{j-\frac{1}{2}}}), 1)$.

(d) If $F_{j-\frac{1}{2}} > 0$ and $F_{j+\frac{1}{2}} < 0$, then

$$(\Lambda^M_{-\frac{1}{2},I_j}, \Lambda^M_{+\frac{1}{2},I_j}) = (\min(1, \frac{\Gamma^M_j}{\lambda F_{j-\frac{1}{2}} - \lambda F_{j+\frac{1}{2}}}), \min(1, \frac{\Gamma^M_j}{\lambda F_{j-\frac{1}{2}} - \lambda F_{j+\frac{1}{2}}})).$$

Similarly, we can find a local pair of numbers $(\Lambda^m_{-\frac{1}{2},I_j}, \Lambda^m_{+\frac{1}{2},I_j})$ such that for any

$$\theta_{j-\frac{1}{2}} \in [0, \Lambda^m_{-\frac{1}{2},I_j}],\ \theta_{j+\frac{1}{2}} \in [0, \Lambda^m_{+\frac{1}{2},I_j}]$$

(2.2.10) holds. There are also four different cases:

(a) If $F_{j-\frac{1}{2}} \geq 0$ and $F_{j+\frac{1}{2}} < 0$, then $(\Lambda^m_{-\frac{1}{2},I_j}, \Lambda^m_{+\frac{1}{2},I_j}) = (1, 1)$.

(b) If $F_{j-\frac{1}{2}} \geq 0$ and $F_{j+\frac{1}{2}} > 0$, then $(\Lambda^m_{-\frac{1}{2},I_j}, \Lambda^m_{+\frac{1}{2},I_j}) = (1, \min(1, \frac{\Gamma^m_j}{-\lambda F_{j+\frac{1}{2}}}))$.

(c) If $F_{j-\frac{1}{2}} < 0$ and $F_{j+\frac{1}{2}} < 0$, then $(\Lambda^m_{-\frac{1}{2},I_j}, \Lambda^m_{+\frac{1}{2},I_j}) = (\min(1, \frac{\Gamma^m_j}{\lambda F_{j-\frac{1}{2}}}), 1)$.

(d) If $F_{j-\frac{1}{2}} < 0$ and $F_{j+\frac{1}{2}} \geq 0$, then

$$(\Lambda^m_{-\frac{1}{2},I_j}, \Lambda^m_{+\frac{1}{2},I_j}) = (\min(1, \frac{\Gamma^m_j}{\lambda F_{j-\frac{1}{2}} - \lambda F_{j+\frac{1}{2}}}), \min(1, \frac{\Gamma^m_j}{\lambda F_{j-\frac{1}{2}} - \lambda F_{j+\frac{1}{2}}})).$$

Finally, the local limiter parameter $\theta_{j+\frac{1}{2}}$ at the cell boundary $x_{j+\frac{1}{2}}$ is defined as

$$\theta_{j+\frac{1}{2}} = min(\Lambda^M_{+\frac{1}{2},I_j}, \Lambda^m_{+\frac{1}{2},I_j}, \Lambda^M_{-\frac{1}{2},I_{j+1}}, \Lambda^m_{-\frac{1}{2},I_{j+1}}), \tag{2.2.11}$$

so that the numerical solutions $\bar{u}^{n+1}_j$, $\forall j, n$ satisfy the maximum principle.

**Remark 2.2.2.** The proposed generalization of the parametrized flux limiter to convection-diffusion problems is rather straightforward. In comparison, it is much more difficult to generalize the polynomial rescaling approach in [17] to schemes with higher than third-order accuracy for convection diffusion problems. The approach there relies on rewriting the updated cell average as a convex combination of some

local quantities within the range $[u_m, u_M]$; this is more difficult to achieve with the diffusion terms [16, 19]. Moreover, the proposed flux limiter introduces very mild time-step constraint to preserve both MPP and high-order accuracy of the original FV RK scheme, see Section 2.4 for more discussions.

## 2.3 MPP FV Method for Two-Dimensional Problems

The extension of the FV RK scheme from 1D case to two-dimensional (2D) convection-diffusion problems is straightforward. For example, we consider a 2D problem on a rectangular domain $[a, b] \times [c, d]$,

$$u_t + f(u)_x + g(u)_y = a(u)_{xx} + b(u)_{yy}. \tag{2.3.12}$$

Without loss of generality, we consider a set of uniform mesh

$$a = x_{\frac{1}{2}} < x_{\frac{3}{2}} < \cdots < x_{N-\frac{1}{2}} < x_{N_x+\frac{1}{2}} = b,\ \Delta x = \frac{b-a}{N_x},$$

$$c = y_{\frac{1}{2}} < y_{\frac{3}{2}} < \cdots < y_{N-\frac{1}{2}} < y_{N_y+\frac{1}{2}} = d,\ \Delta y = \frac{d-c}{N_y},$$

with $I_{i,j} = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}]$. A semi-discrete FV discretization of (2.3.12) gives

$$\frac{d}{dt}\bar{u}_{i,j} + \frac{1}{\Delta x}(\hat{f}_{i+\frac{1}{2},j} - \hat{f}_{i-\frac{1}{2},j}) + \frac{1}{\Delta y}(\hat{g}_{i,j+\frac{1}{2}} - \hat{g}_{i,j-\frac{1}{2}})$$

$$= \frac{1}{\Delta x}(\widehat{(a_x)}_{i+\frac{1}{2},j} - \widehat{(a_x)}_{i-\frac{1}{2},j}) + \frac{1}{\Delta y}(\widehat{(b_y)}_{i,j+\frac{1}{2}} - \widehat{(b_y)}_{i,j-\frac{1}{2}}), \tag{2.3.13}$$

where $\bar{u}_{i,j} = \frac{1}{\Delta x \Delta y}\int\int_{I_{i,j}} u dx dy$ and $\hat{f}_{i+\frac{1}{2},j} = \frac{1}{\Delta y}\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} f(x_{i+\frac{1}{2}}, y)dy$ is the average of the flux over the right boundary of cell $I_{i,j}$. $\hat{g}_{i,j+\frac{1}{2}}$, $\widehat{(a_x)}_{i+\frac{1}{2},j}$, $\widehat{(b_y)}_{i,j+\frac{1}{2}}$ can be defined similarly. The flux $\hat{f}_{i+\frac{1}{2},j}$ is evaluated by applying the Gaussian quadrature rule for integration,

$$\hat{f}_{i+\frac{1}{2},j} = \frac{1}{2}\sum_{i_g}\omega_{i_g} f(u_{i+\frac{1}{2},i_g}). \tag{2.3.14}$$

Here $\sum\limits_{i_g}$ represents the summation over the Gaussian quadratures with $\omega_{i_g}$ being quadrature weights and $u_{i+\frac{1}{2},i_g}$ is the approximated value to $u(x_{i+\frac{1}{2}}, y_{i_g})$ with $y_{i_g}$ being the Gaussian quadrature points over $[y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}]$. $u_{i+\frac{1}{2},i_g}$ can be reconstructed from $\{\bar{u}_{i,j}\}$ in the following two steps. Firstly, we reconstruct $\frac{1}{\Delta x}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u(x, y_{i_g})dx$ from $\{\bar{u}_{i,j}\}$. To do this, we construct a polynomial $Q(y)$ such that

$$\frac{1}{\Delta y}\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} Q(y)dy = \frac{1}{\Delta x \Delta y}\int_{I_{i,j}} u(x,y)dxdy = \bar{u}_{i,j}, \tag{2.3.15}$$

with $j$ belongs to a reconstruction stencil in the $y$-direction as in the one-dimensional case. Then $Q(y_{i_g})$ is a high-order approximation to $\frac{1}{\Delta x}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u(x, y_{i_g})dx$. We let $\mathcal{R}_y$ to denote such reconstruction process in $y$-direction. Secondly, we construct a polynomial $P(x)$ such that

$$\frac{1}{\Delta x}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} P(x)dx = \frac{1}{\Delta x}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u(x, y_{i_g})dx, \tag{2.3.16}$$

with $i$ belongs to a reconstruction stencil in the $x$-direction as in the one-dimensional case. Then $u_{i+\frac{1}{2},i_g} = P(x_{i+\frac{1}{2}})$. Such 1D reconstruction process is denoted as $\mathcal{R}_x$. The 2D reconstructing procedure can be summarized as the following flowchart

$$\{\bar{u}_{i,j}\} \xrightarrow{\mathcal{R}_y} \{\frac{1}{\Delta x}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u(x, y_{i_g})dx\} \xrightarrow{\mathcal{R}_x} \{u_{i+\frac{1}{2},i_g}\}. \tag{2.3.17}$$

Detailed information on the 2D reconstruction procedure is also available in [1].

The basic idea for deriving the MPP flux limiters is the same as for one-dimensional problem, i.e., necessary conditions for the numerical solutions to satisfy maximum principle will be derived, based on similar inequalities as (2.2.9) and (2.2.10).

After being discretized temporally with TVD Runge-Kutta method in the way similar to the case for the one-dimensional problem, the scheme (2.3.13) becomes

$$u_{i,j}^{n+1} = u_{i,j}^{n} - \lambda_x(\hat{H}_{i+\frac{1}{2},j}^{rk} - \hat{H}_{i-\frac{1}{2},j}^{rk}) - \lambda_y(\hat{G}_{i,j+\frac{1}{2}}^{rk} - \hat{G}_{i,j-\frac{1}{2}}^{rk}), \tag{2.3.18}$$

where $\lambda_x = \frac{\triangle t}{\triangle x}$ and $\lambda_y = \frac{\triangle t}{\triangle y}$, and

$$\hat{H}_{i+\frac{1}{2},j}^{rk} = \frac{1}{6}(\hat{f}_{i+\frac{1}{2},j}^{n} - \widehat{(a_x)}_{i+\frac{1}{2},j}^{n}) + \frac{1}{6}(\hat{f}_{i+\frac{1}{2},j}^{1} - \widehat{(a_x)}_{i+\frac{1}{2},j}^{1}) + \frac{2}{3}(\hat{f}_{i+\frac{1}{2},j}^{2} - \widehat{(a_x)}_{i+\frac{1}{2},j}^{2}), \tag{2.3.19}$$

$$\hat{G}_{i,j+\frac{1}{2}}^{rk} = \frac{1}{6}(\hat{g}_{i+\frac{1}{2},j}^{n} - \widehat{(b_x)}_{i+\frac{1}{2},j}^{n}) + \frac{1}{6}(\hat{g}_{i+\frac{1}{2},j}^{1} - \widehat{(b_x)}_{i+\frac{1}{2},j}^{1}) + \frac{2}{3}(\hat{g}_{i+\frac{1}{2},j}^{2} - \widehat{(b_x)}_{i+\frac{1}{2},j}^{2}). \tag{2.3.20}$$

$\hat{H}_{i+\frac{1}{2},j}^{rk}$ and $\hat{G}_{i,j+\frac{1}{2}}^{rk}$ can be understood as the average integral of the numerical fluxes in the temporal direction.

Similarly as for the one-dimensional case, we modify the fluxes as follows,

$$\tilde{H}^{rk}_{i+\frac{1}{2},j} = \theta_{i+\frac{1}{2},j}\hat{H}^{rk}_{i+\frac{1}{2},j} + (1-\theta_{i+\frac{1}{2},j})\hat{h}_{i+\frac{1}{2},j}, \tag{2.3.21}$$

$$\tilde{G}^{rk}_{i,j+\frac{1}{2}} = \theta_{i,j+\frac{1}{2}}\hat{G}^{rk}_{i,j+\frac{1}{2}} + (1-\theta_{i,j+\frac{1}{2}})\hat{g}_{i,j+\frac{1}{2}}, \tag{2.3.22}$$

where $\hat{h}_{i+\frac{1}{2},j}$ and $\hat{g}_{i,j+\frac{1}{2}}$ are low-order monotone flux that satisfy maximum principle, so that

$$u_m \leq u^n_{i,j} - \lambda_x(\tilde{H}^{rk}_{i+\frac{1}{2},j} - \tilde{H}^{rk}_{i-\frac{1}{2},j}) - \lambda_y(\tilde{G}^{rk}_{i,j+\frac{1}{2}} - \tilde{G}^{rk}_{i,j-\frac{1}{2}}) \leq u_M, \tag{2.3.23}$$

with $u_m = \underset{x,y}{min}\ u_0(x,y)$ and $u_M = \underset{x,y}{max}\ u_0(x,y)$.

Introducing the notations

$$\begin{aligned}
F_{i-\frac{1}{2},j} &= \lambda_x(\hat{H}^{rk}_{i-\frac{1}{2},j} - \hat{h}_{i-\frac{1}{2},j}),\\
F_{i+\frac{1}{2},j} &= -\lambda_x(\hat{H}^{rk}_{i+\frac{1}{2},j} - \hat{h}_{i+\frac{1}{2},j}),\\
F_{i,j-\frac{1}{2}} &= \lambda_y(\hat{G}^{rk}_{i,j-\frac{1}{2}} - \hat{g}_{i,j-\frac{1}{2}}),\\
F_{i,j+\frac{1}{2}} &= -\lambda_y(\hat{G}^{rk}_{i,j+\frac{1}{2}} - \hat{g}_{i,j+\frac{1}{2}}),
\end{aligned}$$

and plugging the modified fluxes (2.3.21) and (2.3.22) into (2.3.23), we have

$$\theta_{i+\frac{1}{2},j}F_{i+\frac{1}{2},j} + \theta_{i-\frac{1}{2},j}F_{i-\frac{1}{2},j} + \theta_{i,j+\frac{1}{2}}F_{i,j+\frac{1}{2}} + \theta_{i,j-\frac{1}{2}}F_{i,j-\frac{1}{2}} \leq \Gamma^M_{i,j}, \tag{2.3.24}$$

$$\theta_{i+\frac{1}{2},j}F_{i+\frac{1}{2},j} + \theta_{i-\frac{1}{2},j}F_{i-\frac{1}{2},j} + \theta_{i,j+\frac{1}{2}}F_{i,j+\frac{1}{2}} + \theta_{i,j-\frac{1}{2}}F_{i,j-\frac{1}{2}} \geq \Gamma^m_{i,j}, \tag{2.3.25}$$

where

$$\Gamma^M_{i,j} = u_M - (u_{i,j} - \lambda_x(\hat{h}_{i+\frac{1}{2},j} - \hat{h}_{i-\frac{1}{2},j}) - \lambda_y(\hat{g}_{i,j+\frac{1}{2}} - \hat{g}_{i,j-\frac{1}{2}})) \geq 0, \tag{2.3.26}$$

$$\Gamma^m_{i,j} = u_m - (u_{i,j} - \lambda_x(\hat{h}_{i+\frac{1}{2},j} - \hat{h}_{i-\frac{1}{2},j}) - \lambda_y(\hat{g}_{i,j+\frac{1}{2}} - \hat{g}_{i,j-\frac{1}{2}})) \leq 0. \tag{2.3.27}$$

Similarly as in the one-dimensional case, we need to find numbers $\Lambda_{L,i,j}, \Lambda_{R,i,j}, \Lambda_{D,i,j}, \Lambda_{U,i,j}$ such that if

$$(\theta_{i-\frac{1}{2},j}, \theta_{i+\frac{1}{2},j}, \theta_{i,j-\frac{1}{2}}, \theta_{i,j+\frac{1}{2}}) \in [0, \Lambda_{L,i,j}] \times [0, \Lambda_{R,i,j}] \times [0, \Lambda_{D,i,j}] \times [0, \Lambda_{U,i,j}], \tag{2.3.28}$$

then (2.3.24) and (2.3.25) hold. Both the cases for maximum-value and minimum value should be considered, so the numbers $\Lambda_{L,i,j}, \Lambda_{R,i,j}, \Lambda_{D,i,j}, \Lambda_{U,i,j}$ are

$$\begin{cases} \Lambda_{L,i,j} = min(\Lambda^M_{L,i,j}, \Lambda^m_{L,i,j}), \\ \Lambda_{R,i,j} = min(\Lambda^M_{R,i,j}, \Lambda^m_{R,i,j}), \\ \Lambda_{D,i,j} = min(\Lambda^M_{D,i,j}, \Lambda^m_{D,i,j}), \\ \Lambda_{U,i,j} = min(\Lambda^M_{U,i,j}, \Lambda^m_{U,i,j}). \end{cases} \tag{2.3.29}$$

Finally we define the local limiter parameters as

$$\begin{cases} \theta_{i+\frac{1}{2},j} = min(\Lambda_{R,i,j}, \Lambda_{L,i+1,j}), \\ \theta_{i,j+\frac{1}{2}} = min(\Lambda_{U,i,j}, \Lambda_{D,i,j+1}). \end{cases} \tag{2.3.30}$$

With these limiters, the numerical solution at each time-step will satisfy the maximum principle.

## 2.4 Theoretical Properties

In this section, we provide accuracy analysis for the MPP flux limiter applied to the high-order FV RK scheme solving pure convection problems. Specifically, we will prove that the proposed parametrized flux limiter as in equation (2.2.7) introduces a high-order modification in space and time to the temporal integrated flux of the original scheme, assuming that the solution is smooth enough. A general proof on preservation of *arbitrary* high-order accuracy will be provided for linear problems. Then by performing Taylor expansions around extrema, we prove that the modification from the proposed flux limiter is of at least third-order, for FV RK schemes that are third-order or higher in solving general nonlinear problems.

The entropy solution $u(x,t)$ to a scalar convection problem

$$u_t + f(u)_x = 0, \quad u(x,0) = u_0(x). \tag{2.4.1}$$

satisfies

$$\frac{d}{dt}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} u(x,t)dx = f(u(x_{j+\frac{1}{2}},t)) - f(u(x_{j-\frac{1}{2}},t)). \tag{2.4.2}$$

Integrating (2.4.2) over the time period $[t^n, t^{n+1}]$, we have

$$\bar{u}_j(t^{n+1}) = \bar{u}_j(t^n) - \lambda(\check{f}_{j+\frac{1}{2}} - \check{f}_{j-\frac{1}{2}}), \tag{2.4.3}$$

where $\lambda = \Delta t/\Delta x$ and

$$\bar{u}_j(t) = \frac{1}{\Delta x}\int_{x_{j-1/2}}^{x_{j+1/2}} u(x,t)dx, \quad \check{f}_{j-1/2} = \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} f(u(x_{j-1/2},t))dt. \tag{2.4.4}$$

The entropy solution satisfies the maximum principle in the form of

$$u_m \le \bar{u}_j(t^n) - \lambda(\check{f}_{j+\frac{1}{2}} - \check{f}_{j-\frac{1}{2}}) \le u_M. \tag{2.4.5}$$

For schemes with $(2k+1)^{th}$ order finite volume spatial discretization (2.2.6) and $p^{th}$ order RK time discretization, we assume

$$|\check{f}_{j+\frac{1}{2}} - \hat{H}^{rk}_{j+\frac{1}{2}}| = \mathcal{O}(\Delta x^{2k+1} + \Delta t^p), \quad \forall j. \tag{2.4.6}$$

Our analysis is in the sense of local truncation analysis assuming the difference between $\bar{u}_j(t^n)$ and $\bar{u}^n_j$ is of high-order ($\mathcal{O}(\Delta x^{2k+1} + \Delta t^p)$). Under a corresponding $(2k+1)^{th}$ order reconstruction, the difference between the point values $u(x_j, t^n)$ and $u^n_j$ is also of high-order. In the following, we use them interchangeably when such high-order difference allows.

For the MPP flux limiter, we only consider the maximum-value part as in equation (2.2.9). The proof of equation (2.2.10) for the minimum value would be similar. We

would like to prove that the difference between $\hat{H}^{rk}_{j+\frac{1}{2}}$ and $\tilde{H}^{rk}_{j+\frac{1}{2}}$ in (2.2.7) is of high-order in both space and time, that is

$$|\hat{H}^{rk}_{j+\frac{1}{2}} - \tilde{H}^{rk}_{j+\frac{1}{2}}| = \mathcal{O}(\Delta x^{2k+1} + \Delta t^p), \quad \forall j. \tag{2.4.7}$$

There are four cases of the maximum-value part (2.2.9) outlined in the previous section. The estimate (2.4.7) can be easily checked for case (a) and (d) under the assumption (2.4.6) and the fact (2.4.5), see arguments in [12]. Below we will only discuss case (b), as the argument for case (c) would be similar.

First we give the following lemma:

**Lemma 2.4.1.** Consider applying the MPP flux limiter (2.2.7) for the maximum-value part (2.2.9) with case (b), to prove (2.4.7), it suffices to have

$$|u_M - (\bar{u}_j - \lambda(\check{f}_{j+\frac{1}{2}} - \hat{h}_{j-\frac{1}{2}}))| = \mathcal{O}(\Delta x^{2k+1} + \Delta t^p), \tag{2.4.8}$$

if $u_M - (\bar{u}_j - \lambda(\hat{H}^{rk}_{j+\frac{1}{2}} - \hat{h}_{j-\frac{1}{2}})) < 0$.

*Proof.* For case (b), we are considering the case when $\Lambda_{+\frac{1}{2},I_j} = \frac{\Gamma^M_j}{-\lambda F_{j+\frac{1}{2}}} < 1$. It is equivalent to $u_M - (\bar{u}_j - \lambda(\hat{H}^{rk}_{j+\frac{1}{2}} - \hat{h}_{j-\frac{1}{2}})) < 0$, and

$$\tilde{H}^{rk}_{j+\frac{1}{2}} - \hat{H}^{rk}_{j+\frac{1}{2}} = \frac{\Gamma^M_j + \lambda F_{j+\frac{1}{2}}}{-\lambda} = \frac{u_M - (\bar{u}_j - \lambda(\hat{H}^{rk}_{j+\frac{1}{2}} - \hat{h}_{j-\frac{1}{2}}))}{-\lambda},$$

which indicates that it suffices to have (2.4.8) to obtain (2.4.7) with the assumption (2.4.6). □

**Theorem 2.4.2.** *Assuming $f'(u) > 0$ and $\lambda \max_u |f'(u)| \leq 1$, we have*

$$\bar{u}_j(t^n) - \lambda(\check{f}_{j+\frac{1}{2}} - f(\bar{u}_{j-1}(t^n))) \leq u_M \tag{2.4.9}$$

*if $u(x,t)$ is the entropy solution to (2.4.1) subject to initial data $u_0(x)$.*

*Proof.* Consider the problem (2.4.1) with a different initial condition at time level $t^n$,

$$\tilde{u}(x,t^n) = \begin{cases} u(x,t^n) & x \geq x_{j-\frac{1}{2}}, \\ \bar{u}_{j-1}(t^n) & x < x_{j-\frac{1}{2}}, \end{cases} \tag{2.4.10}$$

here $u(x,t^n)$ is the exact solution of (2.4.1) at time level $t^n$. Assuming $\tilde{u}(x,t)$ is its entropy solution corresponding to the initial data $\tilde{u}(x,t^n)$, instantly we have

$$\bar{\tilde{u}}_j(t^n) = \bar{u}_j(t^n). \tag{2.4.11}$$

Since $f'(u) > 0$, we have

$$f(\tilde{u}(x_{j-\frac{1}{2}},t)) = f(\bar{u}_{j-1}(t^n)), \tag{2.4.12}$$

for $t \in [t^n, t^{n+1}]$. Since $\lambda \max_u |f'(u)| \leq 1$, the characteristic starting from $x_{j-\frac{1}{2}}$ would not hit the side $x_{j+\frac{1}{2}}$, therefore

$$\tilde{u}(x_{j+\frac{1}{2}},t) = u(x_{j+\frac{1}{2}},t) \tag{2.4.13}$$

for $t \in [t^n, t^{n+1}]$. Also since $\tilde{u}$ satisfies the maximum principle $\tilde{u} \leq u_M$, we have

$$\bar{\tilde{u}}_j^{n+1} = \bar{\tilde{u}}_j^n - \lambda(\check{\tilde{f}}_{j+\frac{1}{2}} - \check{\tilde{f}}_{j-\frac{1}{2}}) \leq u_M,$$

where

$$\check{\tilde{f}}_{j-1/2} = \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} f(\tilde{u}(x_{j-1/2}, t))dt. \tag{2.4.14}$$

Substituting (2.4.11), (2.4.12) and (2.4.13) into the above inequality, it follows that

$$\bar{u}_j(t^n) - \lambda(\check{f}_{j+\frac{1}{2}} - f(\bar{u}_{j-1}(t^n)) \leq u_M.$$

□

For the case $f'(u) < 0$, we have the following

**Theorem 2.4.3.** *Assuming $f'(u) < 0$ and $\lambda \max_u |f'(u)| \leq 1$, we have*

$$\bar{u}_j(t^n) - \lambda(\check{f}_{j+\frac{1}{2}} - f(\bar{u}_j(t^n))) \leq u_M, \tag{2.4.15}$$

*if $u(x,t)$ is the entropy solution to problem (2.4.1) subject to initial data $u_0(x)$.*

*Proof.* The proof is similar. The only difference is that in this case, we shall consider an auxiliary problem (2.4.1) with initial data

$$\tilde{\tilde{u}}(x, t^n) = \begin{cases} u(x, t^n) & x \geq x_{j+\frac{1}{2}}, \\ \bar{u}_j(t^n) & x < x_{j+\frac{1}{2}}. \end{cases} \tag{2.4.16}$$

□

Theorem 2.4.2 and 2.4.3 implies the first **main result**.

**Theorem 2.4.4.** *For the cases stated in Theorem 2.4.2 and 2.4.3:* $f'(u) > 0$ *or* $f'(u) < 0$*, with* $\lambda \max_u |f'(u)| \leq 1$*, the estimate*

$$|\hat{H}^{rk}_{j+\frac{1}{2}} - \tilde{H}^{rk}_{j+\frac{1}{2}}| = \mathcal{O}(\Delta x^{2k+1} + \Delta t^p), \quad \forall j$$

*holds if equation*

$$|\check{f}_{j+\frac{1}{2}} - \hat{H}^{rk}_{j+\frac{1}{2}}| = \mathcal{O}(\Delta x^{2k+1} + \Delta t^p), \quad \forall j$$

*holds, when* $\hat{h}_{j-\frac{1}{2}}$ *is the first-order Godunov flux for the modification in (2.2.7).*

*Proof.* The theorem can be proved by combining earlier arguments in this section, observing that $\hat{h}_{j-\frac{1}{2}} = f(\bar{u}^n_{j-1})$ if $f'(u) > 0$, otherwise $\hat{h}_{j-\frac{1}{2}} = f(\bar{u}^n_j)$. □

The conclusion from Theorem 2.4.4 is that the MPP flux limiters for high-order FV RK scheme does not introduce extra CFL constraint to preserve the high-order accuracy of the original scheme. In the linear advection case, Theorem 2.4.4 simply indicates that

**Remark 2.4.5.** The MPP flux limiters preserve high-order accuracy under the CFL requirement $\lambda \max_u |f'(u)| \leq 1$ for linear advection problems when high-order numerical fluxes are limited to the first-order upwind flux. Without much difficulty, we can generalize the results in Theorem 2.4.2, 2.4.3 to two-dimensional linear advection problems.

It is difficult to generalize the above approach to general convection-dominated diffusion problems. However, we believe this is one important step toward a complete proof. Below, by performing Taylor expansions around extrema, we provide a proof of (2.4.7) with third-order spatial and temporal accuracy ($k = 1, p = 3$) for a general nonlinear problem. We consider a first-order monotone flux $\hat{h}_{j-\frac{1}{2}} = \hat{h}(\bar{u}_{j-1}, \bar{u}_j)$ in the proposed parametrized flux limiting procedure (2.2.7). And we define

$$L_{1,j} = \frac{\hat{h}(\bar{u}_{j-1}, \bar{u}_j) - f(\bar{u}_{j-1})}{\bar{u}_j - \bar{u}_{j-1}}, \quad L_{2,j} = -\frac{f(\bar{u}_j) - \hat{h}(\bar{u}_{j-1}, \bar{u}_j)}{\bar{u}_j - \bar{u}_{j-1}}, \tag{2.4.17}$$

where $L_{1,j}$ and $L_{2,j}$ are two coefficients related to the monotonicity condition [7]. Let $L = \max_j |L_{1,j} + L_{2,j}|$, we have

**Theorem 2.4.6.** *Consider a third-order (or higher) finite volume RK discretization for a pure convection problem* (2.4.1)*, with a first-order monotone flux* $\hat{h}_{j-\frac{1}{2}} = \hat{h}(\bar{u}_{j-1}, \bar{u}_j)$ *in* (2.2.7)*. The estimate* (2.4.7) *holds with* $k = 1, p = 3$ *under the CFL condition* $1 - \lambda L \geq 0$*.*

*Proof.* Using the earlier argument, we will only prove (2.4.8), assuming $u_M - (\bar{u}_j - \lambda(\hat{H}^{rk}_{j+\frac{1}{2}} - \hat{h}_{j-\frac{1}{2}})) < 0$. We mimic the proof for the finite-difference scheme in [12]. First we use the 3-point Gauss Lobatto quadrature to approximate $\check{f}_{j+\frac{1}{2}}$,

$$\check{f}_{j+\frac{1}{2}} = \frac{1}{6} f(u(x_{j+\frac{1}{2}}, t^n + \Delta t)) + \frac{2}{3} f((x_{j+\frac{1}{2}}, t^n + \frac{\Delta t}{2})) + \frac{1}{6} f((x_{j+\frac{1}{2}}, t^n)) + \mathcal{O}(\Delta t^3). \tag{2.4.18}$$

Following the characteristics, we get

$$\check{f}_{j+\frac{1}{2}} = \frac{1}{6}f(u(x_{j+\frac{1}{2}} - \lambda_1\Delta x, t^n)) + \frac{2}{3}f(u(x_{j+\frac{1}{2}} - \lambda_2\Delta x, t^n)) + \frac{1}{6}f(u(x_{j+\frac{1}{2}}, t^n)) + \mathcal{O}(\Delta t^3), \tag{2.4.19}$$

where $\lambda_1$ and $\lambda_2$ can be determined from

$$\lambda_1 = \lambda f'(u(x_{j+\frac{1}{2}} - \lambda_1\Delta x, t^n)), \quad \lambda_2 = \frac{\lambda}{2}f'(u(x_{j+\frac{1}{2}} - \lambda_2\Delta x, t^n)). \tag{2.4.20}$$

For the finite volume method, $u(x^*, t^n)$ in (2.4.19) can be approximated by a second order polynomial reconstruction from $\bar{u}_{j-1}$, $\bar{u}_j$ and $\bar{u}_{j+1}$. Denoting $u_1 = u(x_{j+\frac{1}{2}} - \lambda_1\Delta x, t^n)$, $u_2 = u(x_{j+\frac{1}{2}} - \lambda_2\Delta x, t^n)$ and $u_3 = u(x_{j+\frac{1}{2}}, t^n)$, we have

$$u_1 = \frac{1}{6}\left((5 + 6\lambda_1 - 6\lambda_1^2)\bar{u}_j + (-1 + 3\lambda_1^2)\bar{u}_{j-1} + (2 - 6\lambda_1 + 3\lambda_1^2)\bar{u}_{j+1}\right) + O(\Delta x^3), \tag{2.4.21a}$$

$$u_2 = \frac{1}{6}\left((5 + 6\lambda_2 - 6\lambda_2^2)\bar{u}_j + (-1 + 3\lambda_2^2)\bar{u}_{j-1} + (2 - 6\lambda_2 + 3\lambda_1^2)\bar{u}_{j+1}\right) + O(\Delta x^3), \tag{2.4.21b}$$

$$u_3 = \frac{1}{6}\left(5\bar{u}_j - \bar{u}_{j-1} + 2\bar{u}_{j+1}\right) + O(\Delta x^3). \tag{2.4.21c}$$

We prove (2.4.8) case by case. We first consider the case $x_M \in I_j$, with $u_M = u(x_M)$, $u'_M = 0$ and $u''_M \leq 0$. We perform Taylor expansions of $\{\bar{u}_{j-1}, \bar{u}_j, \bar{u}_{j+1}\}$ around $x_M$ with up to third-order, denoting $z = (x_j - x_M)/\Delta x$, (2.4.21) can be

rewritten as

$$u_1 = u_M + u'_M\Delta x(\frac{1}{2} - \lambda_1 + z) + u''_M\frac{\Delta x^2}{2}(\frac{1}{4} - \lambda_1 + \lambda_1^2 + z - 2\lambda_1 z + z^2) + O(\Delta x^3), \tag{2.4.22a}$$

$$u_2 = u_M + u'_M\Delta x(\frac{1}{2} - \lambda_2 + z) + u''_M\frac{\Delta x^2}{2}(\frac{1}{4} - \lambda_2 + \lambda_2^2 + z - 2\lambda_2 z + z^2) + O(\Delta x^3), \tag{2.4.22b}$$

$$u_3 = u_M + u'_M\Delta x(\frac{1}{2} + z) + u''_M\frac{\Delta x^2}{2}(\frac{1}{4} + z + z^2) + O(\Delta x^3). \tag{2.4.22c}$$

Now denoting $\lambda_1 = \lambda_0 + \eta_1\Delta x + \mathcal{O}(\Delta x^2)$ and $\lambda_2 = \frac{\lambda_0}{2} + \eta_2\Delta x + \mathcal{O}(\Delta x^2)$, where $\lambda_0 = \lambda f'(u_M)$, based on the approximation (2.4.22) and Taylor expansions of $\{f'(u_1), f'(u_2)\}$ around $f'(u_M)$ up to second order, $\eta_1$ and $\eta_2$ can be determined by substituting $\lambda_1$ and $\lambda_2$ into (2.4.20) and we have

$$\begin{aligned}
\lambda_1 &= \lambda_0 + f''(u_M)u'_M\lambda(z + \frac{1}{2} - \lambda_0)\Delta x + \mathcal{O}(\Delta x^2),\\
\lambda_2 &= \frac{\lambda_0}{2} + f''(u_M)u'_M\frac{\lambda}{2}(z + \frac{1}{2} - \frac{\lambda_0}{2})\Delta x + \mathcal{O}(\Delta x^2).
\end{aligned}$$

For the first-order monotone flux $\hat{h}_{j-\frac{1}{2}} = \hat{h}(\bar{u}_{j-1}, \bar{u}_j)$, it can be written as

$$\hat{h}_{j-\frac{1}{2}} = f(\bar{u}_{j-1}) + L_{1,j}(\bar{u}_j - \bar{u}_{j-1}), \quad L_{1,j} = \frac{\hat{h}(\bar{u}_{j-1}, \bar{u}_j) - f(\bar{u}_{j-1})}{\bar{u}_j - \bar{u}_{j-1}}, \tag{2.4.23}$$

where $f(\bar{u}_{j-1}) = \hat{h}(\bar{u}_{j-1}, \bar{u}_{j-1})$ due to consistence. $L_{1,j}$ is negative and bounded due to the monotonicity and Lipschitz continuous conditions. On the other hand, $\hat{h}_{j-\frac{1}{2}}$

can also be written as

$$\hat{h}_{j-\frac{1}{2}} = f(\bar{u}_j) + L_{2,j}(\bar{u}_j - \bar{u}_{j-1}), \quad L_{2,j} = -\frac{f(\bar{u}_j) - \hat{h}(\bar{u}_{j-1}, \bar{u}_j)}{\bar{u}_j - \bar{u}_{j-1}}, \tag{2.4.24}$$

where $f(\bar{u}_j) = \hat{h}(\bar{u}_j, \bar{u}_j)$, and $L_{2,j}$ is negative and bounded.

With above notations, by performing Taylor expansions of $\{\bar{u}_{j-1}, \bar{u}_j\}$ around $u_M$ and Taylor expansions of $\{f(\bar{u}_{j-1}), f(\bar{u}_j)\}$ around $f(u_M)$ with up to third-order and with the fact that $u'_M = 0$, we now discuss the following two cases:

- If $f'(u_M) \geq 0$, we have $\lambda_0 = \lambda f'(u_M) \in [0, 1]$ since $\lambda \max_u |f'(u)| \leq 1$. We take $\hat{h}_{j-\frac{1}{2}}$ as in (2.4.23), we have

$$\bar{u}_j - \lambda \left( \check{f}_{j+\frac{1}{2}} - \hat{h}_{j-\frac{1}{2}} \right) = u_M + \frac{u''_M}{12} \Delta x^2 g(z, \lambda_0) + \mathcal{O}(\Delta x^3 + \Delta t^3), \tag{2.4.25}$$

  where

$$g(z, \lambda_0) = g_1(z, \lambda_0) - 6\lambda L_{1,j}(1 - 2z), \tag{2.4.26}$$

  with

$$g_1(z, \lambda_0) = \frac{1}{2} + (5\lambda_0 + 3\lambda_0^2 - 2\lambda_0^3) + 6(-3\lambda_0 + \lambda_0^2)z + 6z^2. \tag{2.4.27}$$

  $\lambda L_{1,j}(1 - 2z) \leq 0$ for $z \in [-\frac{1}{2}, \frac{1}{2}]$ and $L_{1,j} \leq 0$. The minimum value of function $g_1$ with respect to $z$ is

$$(g_1)_{min} = g_1(z, \lambda_0)\Big|_{z=-\frac{1}{2}\lambda_0(\lambda_0-3)} = \frac{1}{2} + \frac{\lambda_0}{2}(\lambda_0 - 2)(\lambda_0 - 1)(5 - 3\lambda_0) \geq 0, \tag{2.4.28}$$

so that $g(z,\lambda_0) \geq 0$. Since $u''_M \leq 0$, from (2.4.25) we obtain (2.4.8).

- If $f'(u_M) < 0$, we have $\lambda_0 \in [-1,0]$. We take $\hat{h}_{j-\frac{1}{2}}$ in (2.4.24), similarly we have (2.4.25) and

$$g(z,\lambda_0) = g_2(z,\lambda_0) - 6\lambda L_{2,j}(1-2z), \tag{2.4.29}$$

with

$$g_2(z,\lambda_0) = \frac{1}{2} + (-\lambda_0 + 3\lambda_0^2 - 2\lambda_0^3) + 6(-\lambda_0 + \lambda_0^2)z + 6z^2. \tag{2.4.30}$$

$\lambda L_{2,j}(1-2z) \leq 0$ for $z \in [-\frac{1}{2}, \frac{1}{2}]$ and $L_{2,j} \leq 0$. The minimum value of $g_2$ with respect to $z$ is

$$(g_2)_{min} = g_2(z,\lambda_0)\Big|_{z=-\frac{1}{2}\lambda_0(\lambda_0-1)} = \frac{1}{2} + \frac{\lambda_0}{2}(\lambda_0+1)(\lambda_0-1)(2-3\lambda_0) \geq 0, \tag{2.4.31}$$

that is $g(z,\lambda_0) \geq 0$. Since $u''_M \leq 0$, from (2.4.25) we also obtain (2.4.8).

Now if $x_M \notin I_j$, however there is a local maximum point $x_M^{loc}$ inside the cell of $I_j$, the above analysis still holds. We then consider that $u(x)$ reaches its local maximum $u_M^{loc}$ over $I_j$ at $x_M^{loc} = x_{j-\frac{1}{2}}$, we have $u'_{j-\frac{1}{2}} < 0$. We take $\hat{h}_{j-\frac{1}{2}}$ as an average of (2.4.23) and (2.4.24). Following the same Taylor-expansion procedure as above, with $z = (x_j - x_M^{loc})/\Delta x = (x_j - x_{j-\frac{1}{2}})/\Delta x = 1/2$, we have

$$\bar{u}_j - \lambda\left(\check{f}_{j+\frac{1}{2}} - \hat{h}_{j-\frac{1}{2}}\right) = u_{j-\frac{1}{2}} + u'_{j-\frac{1}{2}}\Delta x s_1 + (u'_{j-\frac{1}{2}})^2\Delta x^2 s_2 + u''_{j-\frac{1}{2}}\frac{\Delta x^2}{2}s_3 + \mathcal{O}(\Delta x^3 + \Delta t^3), \tag{2.4.32}$$

where

$$s_1 = \frac{1}{2}(-2\lambda_0 + \lambda_0^2) + \frac{1}{2}(1 + \lambda(L_{1,j} + L_{2,j})),$$
$$s_2 = -f''(u_{j-\frac{1}{2}})\frac{\lambda}{8}(3 - 4\lambda_0 + 4\lambda_0^2), \qquad s_3 = \frac{1}{3}(1 - 2\lambda_0 + 3\lambda_0^2 - \lambda_0^3).$$

(2.4.32) can be rewritten as

$$\begin{aligned}\bar{u}_j - \lambda\left(\check{f}_{j+\frac{1}{2}} - \hat{h}_{j-\frac{1}{2}}\right) =& u(x_{j-\frac{1}{2}} - \sqrt{s_3}\Delta x) + u'_{j-\frac{1}{2}}\Delta x\big(\frac{1}{2}(-2\lambda_0 + \lambda_0^2) + \sqrt{s_3} \\ &+\frac{1}{2}(1 + \lambda(L_{1,j} + L_{2,j}))\big) + (u'_{j-\frac{1}{2}})^2\Delta x^2 s_2 + \mathcal{O}(\Delta x^3 + \Delta t^3).\end{aligned} \tag{2.4.33}$$

It is easy to check that $s_3 > 0$ and $\frac{1}{2}(-2\lambda_0 + \lambda_0^2) + \sqrt{s_3} > 0$ for $\lambda_0 = \lambda f'(u_M) \in [-1, 1]$. From the CFL condition $1 + \lambda(L_{1,j} + L_{2,j}) \geq 1 - \lambda L \geq 0$, we obtain $u'_{j-\frac{1}{2}}\Delta x\big(\frac{1}{2}(-2\lambda_0 + \lambda_0^2) + \sqrt{s_3} + \frac{1}{2}(1 + \lambda(L_{1,j} + L_{2,j}))\big) \leq 0$ since $u'_{j-\frac{1}{2}} < 0$.

Now to prove (2.4.8), it is sufficient to show $u(x_{j-\frac{1}{2}} - \sqrt{s_3}\Delta x) + \Delta x^2(u'_{j-\frac{1}{2}})^2 s_2 \leq u_M$ or $u'_{j-\frac{1}{2}} = \mathcal{O}(\Delta x)$. If $[x_{j-\frac{1}{2}} - \sqrt{s_3}\Delta x - \Delta x, x_{j-\frac{1}{2}} - \sqrt{s_3}\Delta x]$ is not a monotone region, there is a point $x^{\#,1}$ in this region, such that $u'(x^{\#,1}) = 0$. Similarly, if $[x_{j-\frac{1}{2}} - \sqrt{s_3}\Delta x - \Delta x, x_{j-\frac{1}{2}} - \sqrt{s_3}\Delta x]$ is a monotone increasing region, since $u'_{j-\frac{1}{2}} < 0$, there is one point $x^{\#,2}$ in $[x_{j-\frac{1}{2}} - \sqrt{s_3}\Delta x, x_{j-\frac{1}{2}}]$, such that $u'(x^{\#,2}) = 0$. For these two cases, $u'_{j-\frac{1}{2}} = \mathcal{O}(\Delta x)$. We then focus on the case when $[x_{j-\frac{1}{2}} - \sqrt{s_3}\Delta x - \Delta x, x_{j-\frac{1}{2}} - \sqrt{s_3}\Delta x]$ is a monotone decreasing region. We assume

$$u(x_{j-\frac{1}{2}} - \sqrt{s_3}\Delta x) + c\Delta x^2 > u_M$$

where $c = |(u'_{j-\frac{1}{2}})^2 s_2|$. Since

$$u(x_{j-\frac{1}{2}} - \sqrt{s_3}\Delta x) = u(x_{j-\frac{1}{2}} - \sqrt{s_3}\Delta x - \Delta x) + u'(x^{\#,3})\Delta x,$$

where $u'(x^{\#,3}) < 0$, we have

$$u'(x^{\#,3})\Delta x + c\Delta x^2 > 0,$$

which implies $|u'(x^{\#,3})| \leq c\Delta x$, therefore, $u'_{j-\frac{1}{2}} = \mathcal{O}(\Delta x)$.

$x_M^{loc} = x_{j+\frac{1}{2}}$ with $u'_{j+\frac{1}{2}} \geq 0$ can be proved similarly. Combining the above discussion, (2.4.8) is proved. □

Therefore, for the general nonlinear convection problem, the MPP flux limiters preserve the third-order accuracy of the original FV RK scheme without extra CFL constraint.

**Remark 2.4.7.** The above proof relies on characteristic tracing. It is difficult to directly generalize such approach to the convection-diffusion problem. On the other hand, similar strategy as that used in [9] by using a Lax-Wendroff strategy, i.e., transforming temporal derivatives into spatial derivatives by repeating using PDEs and its differentiation versions, can be directly applied here. A similar conclusion can be obtained that the MPP flux limiters preserve the third-order accuracy of the original FV RK scheme for the convection dominated diffusion equation without extra CFL constraint. To save some space, we will not repeat the algebraically tedious details here.

**Remark 2.4.8.** It is technically difficult to generalize the proof in Theorem 2.4.6 to higher than third-order, especially with the use of general monotone fluxes, for example, global Lax-Friedrich flux

$$\hat{h}_{j-\frac{1}{2}} = \hat{h}(\bar{u}_{j-1}, \bar{u}_j) = \frac{1}{2}\big(f(\bar{u}_j) + f(\bar{u}_{j-1}) - \alpha(\bar{u}_j - \bar{u}_{j-1})\big), \quad \alpha = \max_u |f'(u)|. \tag{2.4.34}$$

On the other hand, the use of the global Lax-Friedrich flux with an extra large $\alpha$ is not unusual; yet it is quite involved to theoretically or numerically investigate such issue in a nonlinear system. Instead, we use a monotone but over-diffusive flux with

$$\hat{h}_{j+\frac{1}{2}} = \frac{1}{2}\big((1+\alpha)\bar{u}_j + (1-\alpha)\bar{u}_{j+1}\big), \quad \alpha > \max_u |f'(u)| = 1, \tag{2.4.35}$$

for a linear advection equation $u_t + u_x = 0$ with a set of carefully chosen initial conditions. Such scenario is set up to mimic the use of global Lax-Friedrich flux with an extra large $\alpha$ for general nonlinear systems. In Table 2.4.1-2.4.3 below, we present the accuracy test for using the parametrized flux limiter with an over-diffusive first-order monotone flux (2.4.35) with $\alpha = 1.2$ on a linear 5th, 7th and 9th order FV RK schemes, which denoted to be "FVRK5", "FVRK7", "FVRK9" respectively. A mild CFL constraint around 0.7 with time-step $\Delta t = CFL\Delta x/\alpha$ is observed to be sufficient to maintain the high-order accuracy of the underlying scheme with the MPP flux limiter.

| $CFL$ | | mesh | $L^1$ error | order | $L^\infty$ error | order | Umin | Umax |
|---|---|---|---|---|---|---|---|---|
| 0.9 | Non-MPP | 20 | 1.29E-02 | – | 2.00E-02 | – | -0.013805229 | 0.960012218 |
| | | 40 | 5.62E-04 | 4.52 | 9.27E-04 | 4.43 | -0.000670411 | 0.988524452 |
| | | 80 | 1.87E-05 | 4.91 | 3.13E-05 | 4.89 | -0.000025527 | 0.998060523 |
| | | 160 | 5.96E-07 | 4.97 | 9.94E-07 | 4.98 | -0.000000471 | 0.999076363 |
| | | 320 | 1.87E-08 | 4.99 | 3.12E-08 | 4.99 | -0.000000025 | 0.999931894 |
| | | 640 | 5.85E-10 | 5.00 | 9.76E-10 | 5.00 | -0.000000001 | 0.999980112 |
| | | 1280 | 1.83E-11 | 5.00 | 3.05E-11 | 5.00 | 0.000000000 | 0.999992161 |
| | MPP | 20 | 9.97E-03 | – | 1.82E-02 | – | 0.000000000 | 0.960132209 |
| | | 40 | 5.52E-04 | 4.18 | 1.31E-03 | 3.80 | 0.000000000 | 0.988525623 |
| | | 80 | 1.89E-05 | 4.87 | 4.62E-05 | 4.83 | 0.000000000 | 0.998060523 |
| | | 160 | 6.04E-07 | 4.96 | 2.01E-06 | 4.52 | 0.000000325 | 0.999076363 |
| | | 320 | 1.91E-08 | 4.98 | 7.25E-08 | 4.79 | 0.000000010 | 0.999931894 |
| | | 640 | 6.04E-10 | 4.99 | 2.95E-09 | 4.62 | 0.000000001 | 0.999980112 |
| | | 1280 | 1.90E-11 | 4.99 | 1.33E-10 | 4.47 | 0.000000000 | 0.999992161 |
| 0.7 | Non-MPP | 20 | 1.30E-02 | – | 2.01E-02 | – | -0.014015296 | 0.959761206 |
| | | 40 | 5.66E-04 | 4.52 | 9.35E-04 | 4.43 | -0.000680048 | 0.988513480 |
| | | 80 | 1.89E-05 | 4.90 | 3.17E-05 | 4.88 | -0.000025848 | 0.998060157 |
| | | 160 | 6.03E-07 | 4.97 | 1.01E-06 | 4.98 | -0.000000482 | 0.999076351 |
| | | 320 | 1.89E-08 | 4.99 | 3.16E-08 | 4.99 | -0.000000026 | 0.999931893 |
| | | 640 | 5.92E-10 | 5.00 | 9.87E-10 | 5.00 | -0.000000001 | 0.999980112 |
| | | 1280 | 1.85E-11 | 5.00 | 3.09E-11 | 5.00 | 0.000000000 | 0.999992161 |
| | MPP | 20 | 9.95E-03 | – | 1.81E-02 | – | 0.000000000 | 0.959688278 |
| | | 40 | 5.55E-04 | 4.16 | 1.40E-03 | 3.70 | 0.000000000 | 0.988514505 |
| | | 80 | 1.91E-05 | 4.86 | 4.90E-05 | 4.84 | 0.000000000 | 0.998060157 |
| | | 160 | 6.09E-07 | 4.97 | 1.86E-06 | 4.72 | 0.000000000 | 0.999076351 |
| | | 320 | 1.91E-08 | 5.00 | 6.03E-08 | 4.94 | 0.000000002 | 0.999931893 |
| | | 640 | 5.95E-10 | 5.00 | 1.91E-09 | 4.98 | 0.000000000 | 0.999980112 |
| | | 1280 | 1.85E-11 | 5.00 | 5.61E-11 | 5.09 | 0.000000000 | 0.999992161 |

Table 2.4.1: $L^1$ and $L^\infty$ errors and orders for $u_t + u_x = 0$ with initial condition $u(x,0) = \sin^4(x)$. $T = 1$. The over-diffusive global Lax-Friedrichs flux (2.4.35) is used with $\alpha = 1.2$. FVRK5.

## 2.5 Numerical Tests

In this section, we present numerical tests of the proposed MPP high-order FV RK WENO method for convection diffusion problems. Schemes with and without MPP limiters are compared. In these tests, the time-step size for the RK method is chosen

| $CFL$ | | mesh | $L^1$ error | order | $L^\infty$ error | order | Umin | Umax |
|---|---|---|---|---|---|---|---|---|
| 0.9 | Non-MPP | 20 | 4.13E-03 | – | 6.38E-03 | – | -0.004489835 | 0.972363581 |
| | | 40 | 4.69E-05 | 6.46 | 7.37E-05 | 6.44 | -0.000005603 | 0.989301523 |
| | | 80 | 3.99E-07 | 6.88 | 6.38E-07 | 6.85 | 0.000001412 | 0.998091183 |
| | | 160 | 3.20E-09 | 6.96 | 5.10E-09 | 6.97 | 0.000000392 | 0.999077344 |
| | | 320 | 2.51E-11 | 6.99 | 4.01E-11 | 6.99 | 0.000000002 | 0.999931925 |
| | | 640 | 1.97E-13 | 7.00 | 3.14E-13 | 6.99 | 0.000000000 | 0.999980113 |
| | MPP | 20 | 3.60E-03 | – | 6.39E-03 | – | 0.000517069 | 0.972406897 |
| | | 40 | 4.78E-05 | 6.23 | 1.04E-04 | 5.94 | 0.000064524 | 0.989302277 |
| | | 80 | 6.29E-07 | 6.25 | 2.95E-06 | 5.15 | 0.000003451 | 0.998091182 |
| | | 160 | 1.42E-08 | 5.47 | 2.09E-07 | 3.82 | 0.000000602 | 0.999077344 |
| | | 320 | 4.87E-10 | 4.87 | 1.44E-08 | 3.86 | 0.000000012 | 0.999931925 |
| | | 640 | 1.78E-11 | 4.78 | 1.01E-09 | 3.83 | 0.000000001 | 0.999980113 |
| 0.7 | Non-MPP | 20 | 4.12E-03 | – | 6.38E-03 | – | -0.004485289 | 0.972368315 |
| | | 40 | 4.69E-05 | 6.46 | 7.37E-05 | 6.44 | -0.000005556 | 0.989301572 |
| | | 80 | 3.98E-07 | 6.88 | 6.38E-07 | 6.85 | 0.000001412 | 0.998091183 |
| | | 160 | 3.19E-09 | 6.96 | 5.10E-09 | 6.97 | 0.000000392 | 0.999077344 |
| | | 320 | 2.51E-11 | 6.99 | 4.00E-11 | 6.99 | 0.000000002 | 0.999931925 |
| | | 640 | 1.96E-13 | 7.00 | 3.14E-13 | 7.00 | 0.000000000 | 0.999980113 |
| | MPP | 20 | 3.62E-03 | – | 6.59E-03 | – | 0.000515735 | 0.972263646 |
| | | 40 | 4.65E-05 | 6.28 | 8.94E-05 | 6.20 | 0.000054894 | 0.989301394 |
| | | 80 | 3.98E-07 | 6.87 | 6.38E-07 | 7.13 | 0.000001412 | 0.998091183 |
| | | 160 | 3.19E-09 | 6.96 | 5.10E-09 | 6.97 | 0.000000392 | 0.999077344 |
| | | 320 | 2.51E-11 | 6.99 | 4.00E-11 | 6.99 | 0.000000002 | 0.999931925 |
| | | 640 | 1.96E-13 | 7.00 | 3.14E-13 | 7.00 | 0.000000000 | 0.999980113 |

Table 2.4.2: $L^1$ and $L^\infty$ errors and orders for $u_t + u_x = 0$ with initial condition $u(x,0) = \sin^4(x)$. $T = 1$. The over-diffusive global Lax-Friedrichs flux (2.4.35) is used with $\alpha = 1.2$. FVRK7.

such that

$$\Delta t = \min\Big(\frac{CFLC}{\max|f'(u)|}\Delta x, \frac{CFLD}{\max|a'(u)|}\Delta x^2\Big), \tag{2.5.1}$$

for one-dimensional problems and

$$\Delta t = \min\Big(\frac{CFLC}{\max|f'(u)|/\Delta x + \max|g'(u)|/\Delta y}, \frac{CFLD}{\max|a'(u)|/\Delta x^2 + \max|b'(u)|/\Delta y^2}\Big), \tag{2.5.2}$$

for two-dimensional problems. Here CFLC (CFLD resp.) represents the CFL number for the convection (diffusion resp.) term. In our tests, we will take $CFLC = 0.6$

| $CFL$ | | mesh | $L^1$ error | order | $L^\infty$ error | order | Umin | Umax |
|---|---|---|---|---|---|---|---|---|
| 0.9 | Non-MPP | 20 | 1.29E-03 | – | 2.00E-03 | – | -0.001216056 | 0.975890071 |
| | | 40 | 3.99E-06 | 8.34 | 6.19E-06 | 8.34 | 0.000053321 | 0.989362841 |
| | | 80 | 8.67E-09 | 8.85 | 1.37E-08 | 8.82 | 0.000002016 | 0.998091807 |
| | | 160 | 1.75E-11 | 8.95 | 2.76E-11 | 8.96 | 0.000000397 | 0.999077349 |
| | | 320 | 3.44E-14 | 8.99 | 5.51E-14 | 8.97 | 0.000000002 | 0.999931925 |
| | MPP | 20 | 1.20E-03 | – | 2.37E-03 | – | 0.000393260 | 0.975868904 |
| | | 40 | 8.91E-06 | 7.08 | 3.54E-05 | 6.06 | 0.000092174 | 0.989363425 |
| | | 80 | 2.90E-07 | 4.94 | 2.72E-06 | 3.70 | 0.000003586 | 0.998091812 |
| | | 160 | 1.15E-08 | 4.65 | 2.02E-07 | 3.75 | 0.000000600 | 0.999077349 |
| | | 320 | 4.32E-10 | 4.74 | 1.30E-08 | 3.96 | 0.000000013 | 0.999931925 |
| 0.7 | Non-MPP | 20 | 1.29E-03 | – | 2.00E-03 | – | -0.001216106 | 0.975890020 |
| | | 40 | 3.99E-06 | 8.34 | 6.19E-06 | 8.34 | 0.000053321 | 0.989362841 |
| | | 80 | 8.67E-09 | 8.85 | 1.37E-08 | 8.82 | 0.000002016 | 0.998091807 |
| | | 160 | 1.75E-11 | 8.95 | 2.76E-11 | 8.96 | 0.000000397 | 0.999077349 |
| | | 320 | 3.44E-14 | 8.99 | 5.60E-14 | 8.94 | 0.000000002 | 0.999931925 |
| | MPP | 20 | 1.20E-03 | – | 2.47E-03 | – | 0.000419926 | 0.975868183 |
| | | 40 | 3.99E-06 | 8.23 | 6.19E-06 | 8.64 | 0.000053321 | 0.989362841 |
| | | 80 | 8.67E-09 | 8.85 | 1.37E-08 | 8.82 | 0.000002016 | 0.998091807 |
| | | 160 | 1.75E-11 | 8.95 | 2.76E-11 | 8.96 | 0.000000397 | 0.999077349 |
| | | 320 | 3.44E-14 | 8.99 | 5.59E-14 | 8.95 | 0.000000002 | 0.999931925 |

Table 2.4.3: $L^1$ and $L^\infty$ errors and orders for $u_t + u_x = 0$ with initial condition $u(x,0) = \sin^4(x)$. $T = 1$. The over-diffusive global Lax-Friedrichs flux (2.4.35) is used with $\alpha = 1.2$. FVRK9.

for convection-dominated problems and $CFLD = 0.8$ for pure diffusion problems. Herein we let "MPP" and "NonMPP" denote the scheme with and without the MPP limiter, and $U_{\max}$ ($U_{\min}$ resp.) denote the maximum (minimum resp.) value among the numerical cell averages $\bar{u}_j$. To better illustrate the effectiveness of the MPP limiters, we use linear weights instead of WENO weights in the reconstruction procedure for the convection term.

### 2.5.1 Basic Tests

**Example 2.5.1.** (1D Linear Problem)

$$u_t + u_x = \epsilon u_{xx},\ x \in [0, 2\pi],\ \epsilon = 0.00001. \tag{2.5.3}$$

We test the proposed scheme on the problem (2.5.3) with initial condition $u(x, 0) = \sin^4(x)$ and periodic boundary condition. The exact solution is

$$u(x,t) = \frac{3}{8} - \frac{1}{2}\exp(-4\epsilon t)\cos(2(x-t)) + \frac{1}{8}\exp(-16\epsilon t)\cos(4(x-t)). \tag{2.5.4}$$

The $L_1$ and $L_\infty$ errors and orders of convergence for the scheme with and without MPP limiters are shown in Table 2.5.1. It is observed that the MPP limiter avoids overshooting and undershooting of the numerical solution while preserve high-order accuracy.

| | mesh | $L_1$ error | order | $L_\infty$ error | order | Umax | Umin |
|---|---|---|---|---|---|---|---|
| Non-MPP | 50 | 1.68E-04 | — | 2.76E-04 | — | 0.996998594480 | -0.000182938402 |
| | 100 | 5.47E-06 | 4.94 | 9.11E-06 | 4.92 | 0.997933416789 | -0.000005718342 |
| | 200 | 1.72E-07 | 4.99 | 2.87E-07 | 4.99 | 0.999579130130 | -0.000000153518 |
| | 400 | 5.38E-09 | 5.00 | 9.00E-09 | 5.00 | 0.999905929907 | -0.000000002134 |
| | 800 | 1.68E-10 | 5.00 | 2.81E-10 | 5.00 | 0.999945898951 | 0.000000001890 |
| MPP | 50 | 1.71E-04 | — | 2.87E-04 | — | 0.996998296191 | 0.000000000000 |
| | 100 | 5.46E-06 | 4.93 | 1.34E-05 | 4.42 | 0.997933416819 | 0.000000016274 |
| | 200 | 1.72E-07 | 5.00 | 4.91E-07 | 4.77 | 0.999579130130 | 0.000000013987 |
| | 400 | 5.38E-09 | 5.03 | 1.25E-08 | 5.29 | 0.999905929907 | 0.000000001048 |
| | 800 | 1.68E-10 | 5.01 | 2.81E-10 | 5.48 | 0.999945898951 | 0.000000001890 |

Table 2.5.1: Accuracy tests for 1D linear equation (2.5.3) with exact solution (2.5.4) at time $T = 1.0$.

We then test problem (2.5.3) with the initial condition having rich solution structures

$$u_0(x) = \begin{cases} \frac{1}{6}(G(x,\beta,z-\delta) + G(x,\beta,z+\delta) + 4G(x,\beta,z)), & -0.8 \le x \le -0.6; \\ 1, & -0.4 \le x \le -0.2; \\ 1 - |10(x-0.1)|, & 0 \le x \le 0.2; \\ \frac{1}{6}(F(x,\gamma,a-\delta) + F(x,\gamma,a+\delta) + 4F(x,\gamma,a)), & 0.4 \le x \le 0.6; \\ 0, & \text{otherwise.} \end{cases} \tag{2.5.5}$$

where $G(x,\beta,z) = e^{-\beta(x-z)^2}$ and $F(x,\gamma,a) = \sqrt{\max(1-\gamma^2(x-a)^2, 0)}$. The constants involved are $a = 0.5, z = -0.7, \delta = 0.005, \gamma = 10$ and $\beta = \log 2/(36\delta^2)$ and the boundary condition is periodic. The maximum and minimum cell averages are listed in Table 2.5.2. In Figure 2.5.1, the effectiveness of the MPP limiters in controlling the numerical solution within theoretical bounds can be clearly observed.

| | NonMPP | | MPP | |
|---|---|---|---|---|
| mesh | Umax | Umin | Umax | Umin |
| 50 | 1.106238399422 | -0.114766938420 | 1.000000000000 | 0.000000000000 |
| 100 | 1.056114534445 | -0.067351423479 | 1.000000000000 | 0.000000000000 |
| 200 | 1.054864483784 | -0.054928012204 | 1.000000000000 | 0.000000000000 |
| 400 | 1.048250067722 | -0.048250171364 | 1.000000000000 | 0.000000000000 |
| 800 | 1.031246517796 | -0.031246517794 | 1.000000000000 | 0.000000000000 |

Table 2.5.2: The maximum and minimum values of the numerical cell averages for problem (2.5.3) with initial conditions (2.5.5) at time $T = 1.0$.

**Example 2.5.2.** (1D Nonlinear Equation) We test the FV RK scheme with and

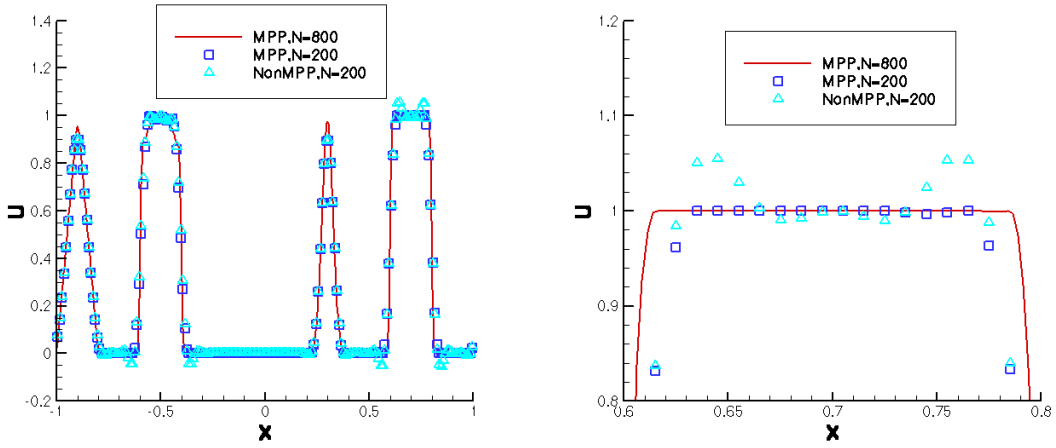


Figure 2.5.1: Left: Comparison of the FV RK scheme with and without MPP limiters for 1d linear problem (2.5.3) with initial condition (2.5.5) at $T = 1.0$. Right: Zoom-in around the overshooting.

without MPP limiters on Burgers' equation

$$u_t + (\frac{u^2}{2})_x = \epsilon u_{xx}, \ x \in [-1, 1], \ \epsilon = 0.0001, \tag{2.5.6}$$

with initial condition

$$u(x, 0) = \begin{cases} 2, & |x| < 0.5; \\ 0, & \text{otherwise}, \end{cases}$$

and periodic boundary conditions. The results in Table 2.5.3 shows that the numerical solution goes beyond the theoretical bounds if no limiters are applied and stays within the theoretical range if MPP limiters are applied.

| | NonMPP | | MPP | |
|---|---|---|---|---|
| mesh | Umax | Umin | Umax | Umin |
| 50 | 2.349929038912 | -0.063536142936 | 1.818784698878 | 0.000000000000 |
| 100 | 2.438970633433 | -0.135799476071 | 1.879377697365 | 0.000000000000 |
| 200 | 2.217068598684 | -0.095548979222 | 1.913720603302 | 0.000000000000 |
| 400 | 2.216719764740 | -0.095114086983 | 1.938439146468 | 0.000000000000 |
| 800 | 2.210614277385 | -0.092745597929 | 1.959770865698 | 0.000000000000 |

Table 2.5.3: The maximum and minimum values of the numerical cell averages for Burgers' equation (2.5.6) at time $T = 0.05$.

**Example 2.5.3.** (2D Linear Problem)

$$u_t + u_x + u_y = \epsilon(u_{xx} + u_{yy}), \quad (x, y) \in [0, 2\pi]^2, \quad \epsilon = 0.001. \tag{2.5.7}$$

We first consider the problem with initial condition $u(x, y, 0) = \sin^4(x + y)$ and periodic boundary condition. The exact solution to the problem is

$$u(x, y, t) = \frac{3}{8} - \frac{1}{2}\exp(-8\epsilon t)\cos(2(x + y - 2t)) + \frac{1}{8}\exp(-32\epsilon t)\cos(4(x + y - 2t)). \tag{2.5.8}$$

The $L_1$ and $L_\infty$ errors and orders of convergence for the FV RK scheme with and without MPP limiters are shown in Table 2.5.4. High-order accuracy is preserved when the MPP limiters are applied to control the numerical solution within the theoretical bounds.

We then consider problem (2.5.7) with initial condition

$$u(x, 0) = \begin{cases} 1, & (x, y) \in [\frac{\pi}{2}, \frac{3\pi}{2}] \times [\frac{\pi}{2}, \frac{3\pi}{2}]; \\ 0, & \text{otherwise on } [0, 2\pi] \times [0, 2\pi], \end{cases} \tag{2.5.9}$$

| | mesh | $L_1$ error | order | $L_\infty$ error | order | Umax | Umin |
|---|---|---|---|---|---|---|---|
| NonMPP | $16 \times 16$ | 4.86E-03 | — | 9.30E-03 | — | 0.919696089900 | 0.000159282060 |
| | $32 \times 32$ | 2.85E-04 | 4.29 | 4.49E-04 | 4.37 | 0.986054820018 | -0.000283832731 |
| | $64 \times 64$ | 9.82E-06 | 4.84 | 1.62E-05 | 4.79 | 0.995960434630 | -0.000004482350 |
| | $128 \times 128$ | 3.12E-07 | 4.96 | 5.22E-07 | 4.95 | 0.998407179488 | 0.000001288422 |
| | $256 \times 256$ | 9.73E-09 | 5.00 | 1.63E-08 | 5.01 | 0.998990497491 | 0.000000740680 |
| MPP | $16 \times 16$ | 4.86E-03 | — | 9.30E-03 | — | 0.919696089900 | 0.000159282060 |
| | $32 \times 32$ | 2.87E-04 | 4.27 | 4.49E-04 | 4.37 | 0.986054818813 | 0.000000000000 |
| | $64 \times 64$ | 9.82E-06 | 4.85 | 1.64E-05 | 4.77 | 0.995960434630 | 0.000000000000 |
| | $128 \times 128$ | 3.12E-07 | 4.97 | 5.22E-07 | 4.97 | 0.998407179488 | 0.000001288422 |
| | $256 \times 256$ | 9.73E-09 | 5.00 | 1.63E-08 | 5.01 | 0.998990497491 | 0.000000740680 |

Table 2.5.4: Accuracy tests for 2D linear equation (2.5.7) with exact solution (2.5.8) at time $T = 1.0$.

and periodic boundary condition. The results are shown in Table 2.5.5, which indicates the effectiveness of the MPP limiter.

| | NonMPP | | MPP | |
|---|---|---|---|---|
| mesh | Umax | Umin | Umax | Umin |
| $16 \times 16$ | 1.196476571354 | -0.102486638966 | 1.000000000000 | 0.000000000000 |
| $32 \times 32$ | 1.317444117818 | -0.169214623680 | 1.000000000000 | 0.000000000000 |
| $64 \times 64$ | 1.341696522446 | -0.182902057169 | 1.000000000000 | 0.000000000000 |
| $128 \times 128$ | 1.225931525834 | -0.116989442889 | 1.000000000000 | 0.000000000000 |
| $256 \times 256$ | 1.108731559448 | -0.055808238605 | 1.000000000000 | 0.000000000000 |

Table 2.5.5: Maximum and minimum cell averages in the 2D linear problem (2.5.7) with initial condition (2.5.9) at time $T = 0.1$.

**Example 2.5.4.** (1D Buckley-Leverett Equation) Consider the problem

$$u_t + f(u)_x = \epsilon(\nu(u)u_x)_x, \quad \epsilon = 0.01, \tag{2.5.10}$$

where

$$\nu(u) = \begin{cases} 4u(1-u), & 0 \le u \le 1; \\ 0, & \text{otherwise}, \end{cases} \quad \text{and} \quad f(u) = \frac{u^2}{u^2 + (1-u)^2}.$$

The initial condition is

$$u(x,0) = \begin{cases} 1-3x, & 0 \le x < \frac{1}{3}; \\ 0, & \frac{1}{3} \le x \le 1, \end{cases}$$

and the boundary conditions are $u(0,t) = 1$ and $u(1,t) = 0$. The numerical results are shown in Table 2.5.6. The numerical solution goes below 0 if MPP limiters are not applied, and stays within the theoretical bounds $[0,1]$ when MPP limiters are applied. Figure 2.5.2 illustrates the effectiveness of MPP limiters near the undershooting of the numerical solution.

| | NonMPP | | MPP | |
|---|---|---|---|---|
| mesh | Umax | Umin | Umax | Umin |
| 50 | 1.000000000000000 | -0.002643266424381 | 1.000000000000000 | 0.000000000000000 |
| 100 | 1.000000000000000 | -0.001813338703220 | 1.000000000000000 | 0.000000000000000 |
| 200 | 1.000000000000000 | -0.000942402907667 | 1.000000000000000 | 0.000000000000000 |
| 400 | 1.000000000000000 | -0.000491323673758 | 1.000000000000000 | 0.000000000000000 |
| 800 | 1.000000000000000 | -0.000247268741213 | 1.000000000000000 | 0.000000000000000 |

Table 2.5.6: The maximum and minimum values for 1D Buckley-Leverett problem (2.5.10) at time $T = 0.2$.

**Example 2.5.5.** (2D Buckley-Leverett Equation) Consider

$$u_t + f(u)_x + g(u)_y = \epsilon(u_{xx} + u_{yy}), \quad (x,y) \in [-1.5, 1.5]^2, \quad \epsilon = 0.01 \qquad (2.5.11)$$

where

$$f(u) = \frac{u^2}{u^2 + (1-u)^2}, \quad g(u) = f(u)(1 - 5(1-u)^2),$$

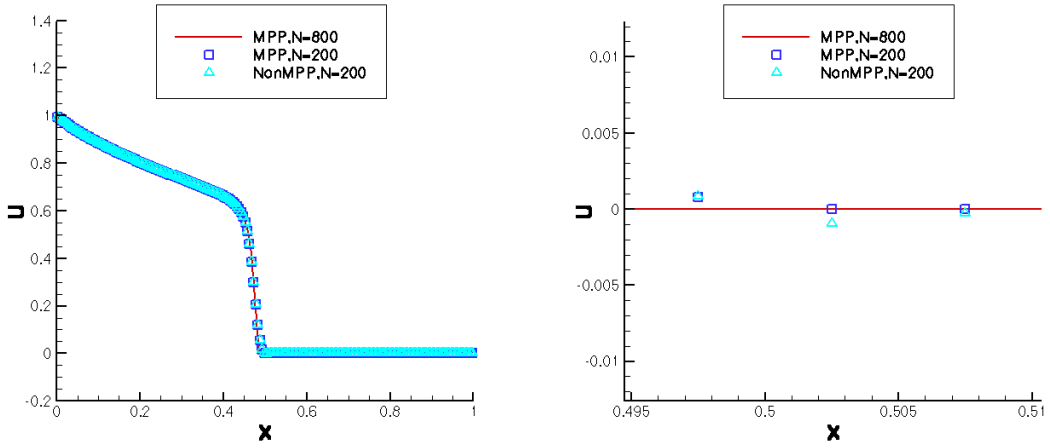


Figure 2.5.2: Left: Solutions for 1D Buckley-Leverett equation (2.5.10) at $T = 0.2$. Right: Zoom-in around the undershooting.

with initial condition

$$u(x, y, 0) = \begin{cases} 1, & x^2 + y^2 < 0.5; \\ 0, & \text{otherwise on } [-1.5, 1.5]^2, \end{cases}$$

and periodic boundary conditions. The numerical results in Table 2.5.7 show that the MPP limiters effectively control the numerical solution within the theoretical range $[0, 1]$.

| | NonMPP | | MPP | |
|---|---|---|---|---|
| mesh | Umax | Umin | Umax | Umin |
| $16 \times 16$ | 1.190542402917 | -0.142603740886 | 1.000000000000 | 0.000000000000 |
| $32 \times 32$ | 1.183357844800 | -0.174592560044 | 1.000000000000 | 0.000000000000 |
| $64 \times 64$ | 1.148424330885 | -0.167227853261 | 1.000000000000 | 0.000000000000 |
| $128 \times 128$ | 1.084563025034 | -0.083883559766 | 1.000000000000 | 0.000000000000 |
| $256 \times 256$ | 0.998736899089 | -0.018463025969 | 0.998566263416 | 0.000000000000 |

Table 2.5.7: Maximum and minimum cell averages for 2D Buckley-Leverett problem (2.5.11) at time $T = 0.5$.

**Example 2.5.6.** (1D Porous Medium Equation) Consider

$$u_t = (u^m)_{xx},\ m > 1, \quad x \in [-2\pi, 2\pi] \tag{2.5.12}$$

whose solution is the Barenblatt solution in the following form

$$B_m(x,t) = t^{-k}\Big[(1 - \frac{k(m-1)}{2m}\frac{|x|^2}{t^{2k}})_+\Big]^{\frac{1}{m+1}}, \tag{2.5.13}$$

with $k = \frac{1}{m+1}$ and $u_+ = \max(u, 0)$. The boundary conditions are assumed to be zero at both ends. Starting from time $T_0 = 1$, we compute the numerical solution of the problem up to time $T = 2$ by the FV RK scheme and the results are shown in Table 2.5.8. Obviously, there are undershoots when regular FV RK scheme are applied. And the MPP limiters can effectively eliminate the overshoots in the numerical solution. Also the plot in Figure 2.5.3 shows the effectiveness of the MPP limiters.

| $N = 100$ | NonMPP | | MPP | |
|---|---|---|---|---|
| m | Umax | Umin | Umax | Umin |
| 2 | 0.793283780606 | -0.000338472445 | 0.793283375962 | 0.000000000000 |
| 3 | 0.840666629482 | -0.001792679096 | 0.840663542409 | 0.000000000000 |
| 5 | 0.890829374423 | -0.005693908465 | 0.890821177490 | 0.000000000000 |
| 8 | 0.925837535365 | -0.003841778007 | 0.925826127818 | 0.000000000000 |

Table 2.5.8: Maximum and minimum cell average values for 1D porous medium problem (2.5.12) with $m = 2, 3, 5, 8$ at time $T = 2$.

**Example 2.5.7.** (2D Porous Medium Equation) Consider

$$u_t = (u^m)_{xx} + (u^m)_{yy}, \quad m = 2, \quad (x, y) \in [-1, 1]^2 \tag{2.5.14}$$

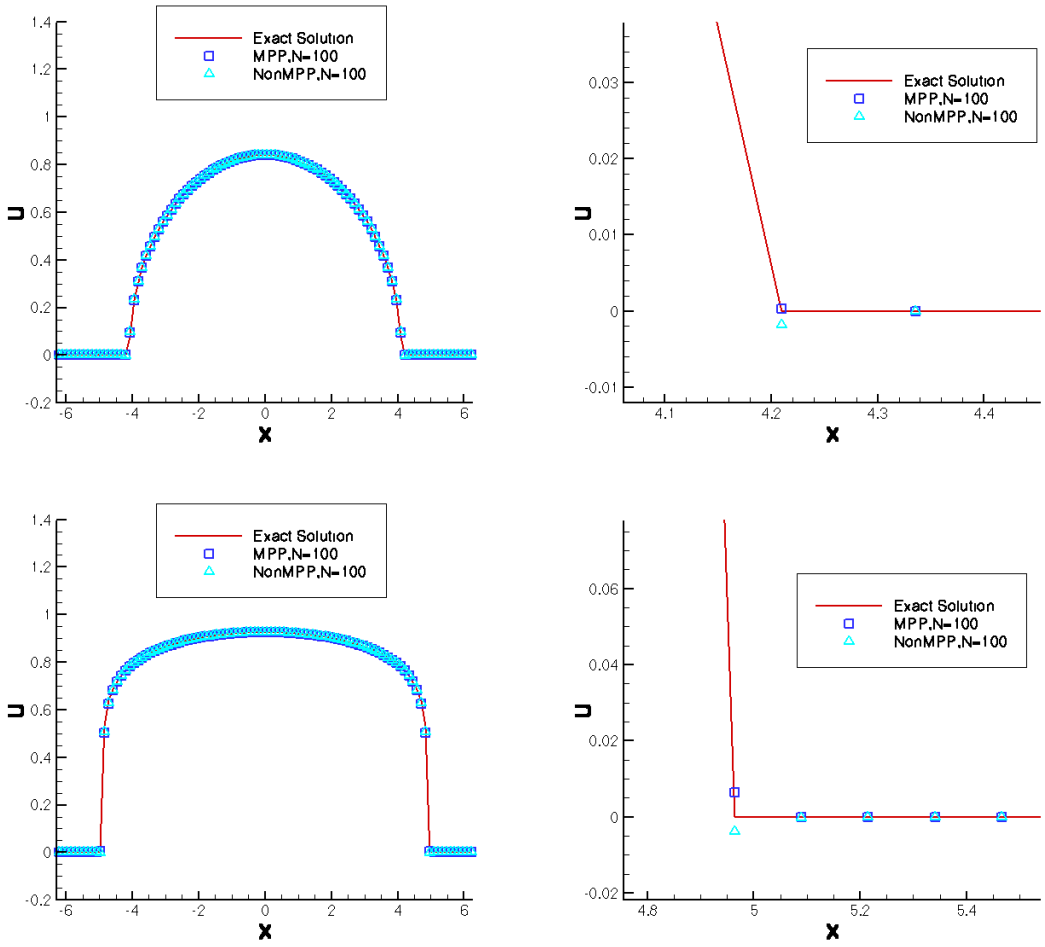


Figure 2.5.3: Left: Plot for 1D porous medium problem (2.5.12) with N=100 at $T = 2$. Top is for m=3 and bottom is for m=8. Right: Zoom-in around the undershooting.

with initial condition

$$u(x,y,0) = \begin{cases} 1, & (x,y) \in [-\frac{1}{2},\frac{1}{2}]^2; \\ 0, & \text{otherwise on } [-\frac{1}{2},\frac{1}{2}]^2, \end{cases}$$

and periodic boundary conditions. We produce the numerical results at time $T = 0.005$, as shown in Table 2.5.9. The results show that the MPP limiters perform effectively at avoiding overshooting and undershooting of the numerical solution.

| | NonMPP | | MPP | |
|---|---|---|---|---|
| mesh | Umax | Umin | Umax | Umin |
| $16 \times 16$ | 1.000485743751 | -0.000349298087 | 0.999827816078 | 0.000000000000 |
| $32 \times 32$ | 0.999625786453 | -0.001200636807 | 0.999573139639 | 0.000000000000 |
| $64 \times 64$ | 0.999537081790 | -0.000855830629 | 0.999533087178 | 0.000000000000 |
| $128 \times 128$ | 0.999527411822 | -0.000474775257 | 0.999526635569 | 0.000000000000 |
| $256 \times 256$ | 0.999525567240 | -0.000261471521 | 0.999525309113 | 0.000000000000 |

Table 2.5.9: Maximum and minimum cell average values for 2D porous medium problem (2.5.14) at time $T = 0.005$.

### 2.5.2 Incompressible-Flow Problems

In this subsection, we test the proposed scheme on incompressible-flow problems in the form

$$\omega_t + (u\omega)_x + (v\omega)_y = \frac{1}{Re}(\omega_{xx} + \omega_{yy}), \tag{2.5.15}$$

where $\langle u, v\rangle$ is the divergence-free velocity field and Re is the Reynold number. The theoretical solution satisfies the maximum principle due to the divergence-free property of the velocity field. For the numerical solution to satisfy the maximum

principle, discretized divergence-free condition needs to be considered, hence special treatment needs to be taken when low-order flux for the convection term is designed. For details, see [12], according to which we design the low-order monotone flux for the following incompressible problems.

**Example 2.5.8.** (Rotation with Viscosity)

$$u_t + (-yu)_x + (xu)_y = \frac{1}{Re}(u_{xx} + u_{yy}), \quad (x, y) \in [-\pi, \pi]^2. \tag{2.5.16}$$

The initial condition is shown in Figure 2.5.4 and the boundary condition is assumed to be periodic. The numerical solution at time $T = 0.1$ is shown in Table 2.5.10, which indicates that there are overshooting and undershooting in the numerical solution by regular FV RK scheme and they can be avoided by applying the MPP limiter. The solutions with and without MPP limiter are also compared in Figure 2.5.5. From Table 2.5.10 and Figure 2.5.5, the effectiveness of the MPP limiter can be better illustrated when Renold number is larger. This is because the overshooting and undershooting are more apparent when Reynold number is larger, which corresponds to less diffusion.

**Example 2.5.9.** (Swirling Deformation with Viscosity)

$$u_t + (-\cos^2(\frac{x}{2})\sin(y)g(t)u)_x + (\sin(x)\cos^2(\frac{y}{2})t(t))u)_y = \frac{1}{Re}(u_{xx} + u_{yy}), \tag{2.5.17}$$

where $(x, y) \in [-\pi, \pi]^2$ and $g(t) = \cos(\pi t/T)\pi$. The initial condition is the same as in Example 4.8 and the boundary conditions are also periodic. Similarly, we also compare the results for different Reynold numbers Re=100 and Re=10000. As

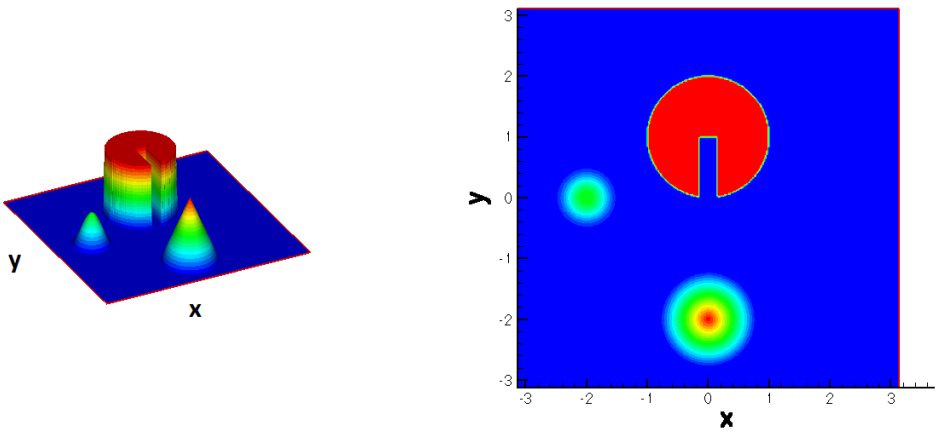


Figure 2.5.4: Initial condition for Example 3.8 and Example 3.9.

shown in Table 2.5.11, the MPP limiter plays the role of eliminating overshooting and undershooting in the numerical solution, especially for problems with larger Reynold number. This can also be observed in Figure 2.5.6.

**Example 2.5.10.** (Vortex Patch) Consider the problem

$$\omega_t + (u\omega)_x + (v\omega)_y = \frac{1}{Re}(\omega_{xx} + \omega_{yy}), \tag{2.5.18}$$

$$\Delta\psi = \omega, \ \langle u, v\rangle = \langle -\psi_y, \psi_x\rangle, \tag{2.5.19}$$

with the following initial condition

$$\omega(x, y, 0) = \begin{cases} -1, & \frac{\pi}{2} \le x \le \frac{3\pi}{2}, \ \frac{\pi}{4} \le \frac{3\pi}{4}, \\ 1, & \frac{\pi}{2} \le x \le \frac{3\pi}{2}, \ \frac{5\pi}{4} \le \frac{7\pi}{4}, \\ 0, & \text{otherwise}, \end{cases} \tag{2.5.20}$$

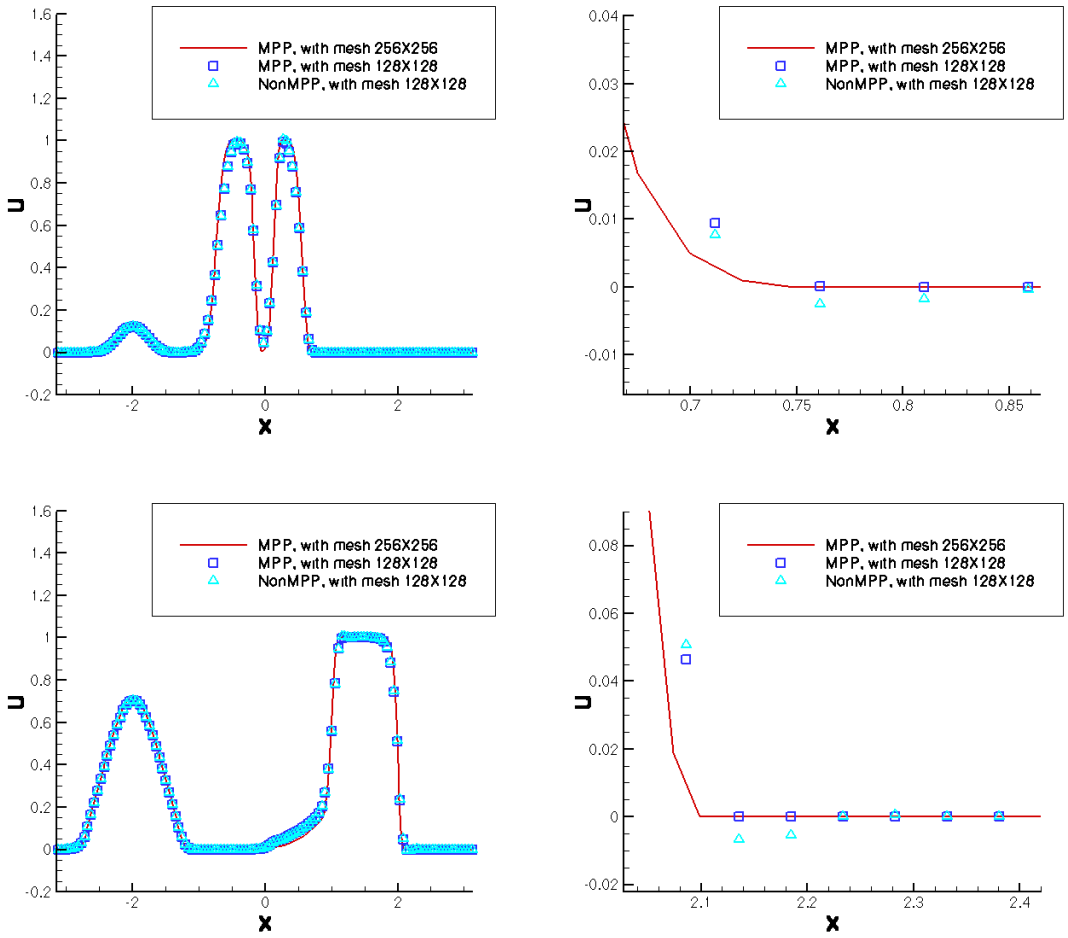


Figure 2.5.5: Left: Cutting plots for rotation problem (2.5.16) for Re=10000 at $T = 0.1$. Right: Zoom-in around the undershooting. Top: cutting along $y = 5\Delta y$ for $N_y = 128$; Bottom: cutting along $x = 0$.

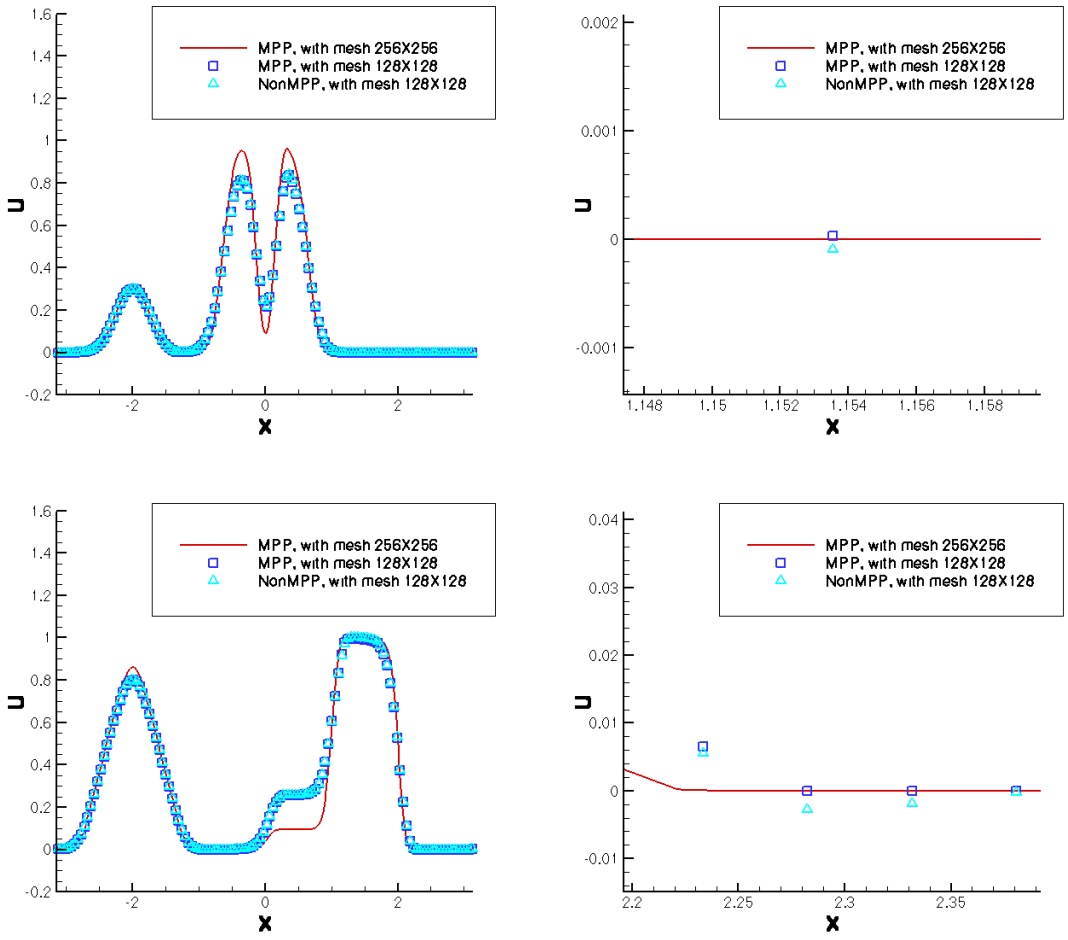


Figure 2.5.6: Left: Cutting plots for swirling deformation problem (2.5.17) for Re=10000 at T=0.1. Right: Zoom-in around the undershooting. Top: cutting along $y = 5\Delta y$ for $N_y = 128$; Bottom: cutting along $x = 0$.

| Re=100 | NonMPP | | MPP | |
|---|---|---|---|---|
| mesh | Umax | Umin | Umax | Umin |
| $16 \times 16$ | 0.947915608973 | -0.041388485669 | 0.947719795318 | 0.000000000000 |
| $32 \times 32$ | 0.999789765557 | -0.048836983632 | 0.996173203589 | 0.000000000000 |
| $64 \times 64$ | 1.008171330748 | -0.039241271474 | 0.999999999928 | 0.000000000000 |
| $128 \times 128$ | 1.002125190412 | -0.027962451582 | 0.999999999920 | 0.000000000000 |
| $256 \times 256$ | 1.000099518450 | -0.012262487330 | 0.999999999983 | 0.000000000000 |
| Re=10000 | NonMPP | | MPP | |
| mesh | Umax | Umin | Umax | Umin |
| $16 \times 16$ | 0.949247968412 | -0.042285048496 | 0.949049295419 | 0.000000000000 |
| $32 \times 32$ | 1.002247494119 | -0.053653247391 | 0.996943318800 | 0.000000000000 |
| $64 \times 64$ | 1.012845607701 | -0.049914946698 | 0.999999462216 | 0.000000000000 |
| $128 \times 128$ | 1.009050027036 | -0.050526262050 | 0.999999999977 | 0.000000000000 |
| $256 \times 256$ | 1.007608558521 | -0.058482843302 | 0.999999999995 | 0.000000000000 |

Table 2.5.10: The maximum and minimum cell averages for rotation problem (2.5.16) with two different Reynold numbers at $T = 0.1$.

and periodic-boundary condition. The maximum and minimum cell averages of the numerical solution with two Reynold numbers Re=100 and Re=10000, obtained by regular FV RK scheme and the scheme with the MPP limiter are compared in Table 2.5.12, from which we can observe the effectiveness of the MPP limiter in controlling overshooting and undershooting in the numerical solution. The contour plot of the solution is presented in Figure 2.5.7, which shows that the solution obtained by FV RK scheme with the MPP limiter is comparable to that obtained by regular FV RK scheme.

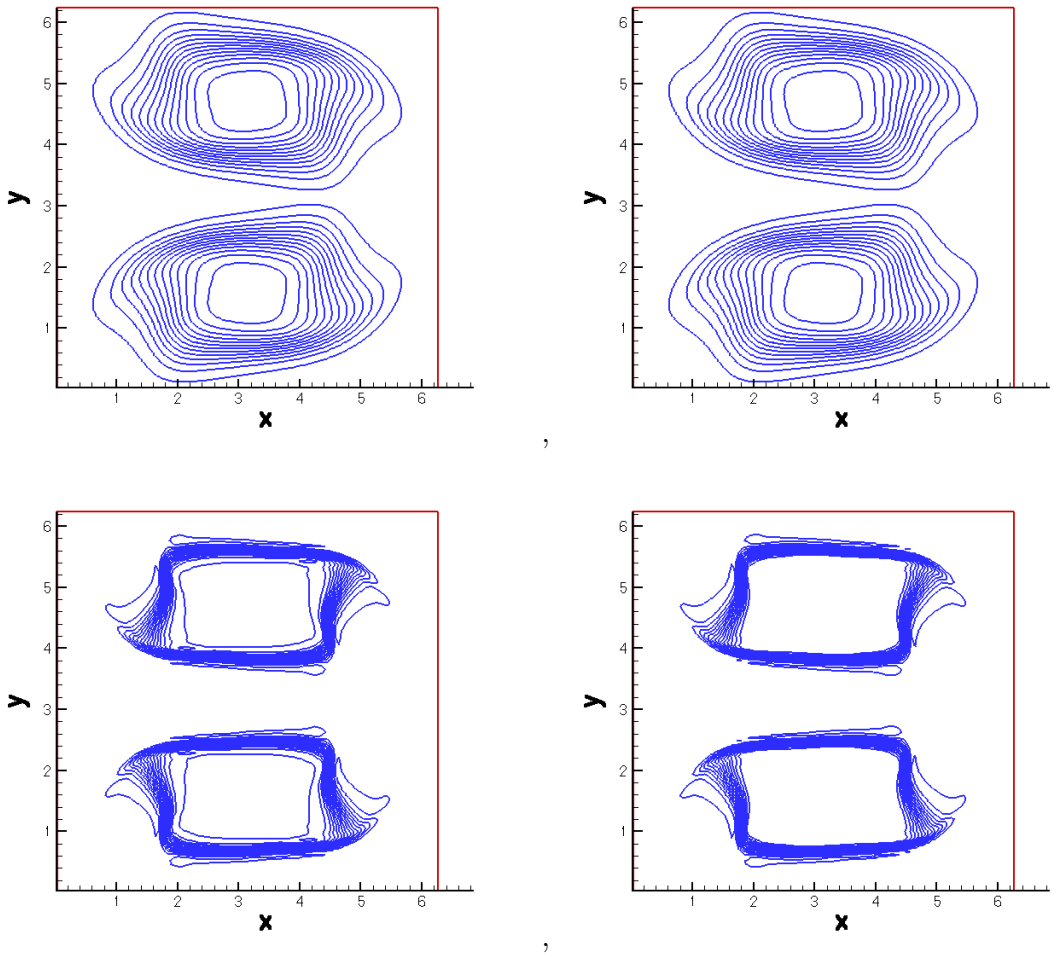


Figure 2.5.7: Contours of the numerical solution for vortex patch problem (2.5.18) with Re=100 (top) and Re=10000 (bottom) at time $T = 5$. The contours on the left are for the NonMPP scheme and those on the right are for the MPP scheme. For both Re=100 and Re=10000, 30 equally spaced contour lines within the range $[-1.1, 1.1]$ are plotted.

| Re=100 | NonMPP | | MPP | |
|---|---|---|---|---|
| mesh | Umax | Umin | Umax | Umin |
| $16 \times 16$ | 0.873440241699 | -0.010737472197 | 0.842184825192 | 0.000000000000 |
| $32 \times 32$ | 0.971822334038 | -0.011947680561 | 0.942384582101 | 0.000000000000 |
| $64 \times 64$ | 0.997563271155 | -0.005935366467 | 0.986960253479 | 0.000000000000 |
| $128 \times 128$ | 1.000886437426 | -0.001258903421 | 0.998925498573 | 0.000000000000 |
| $256 \times 256$ | 1.000040508119 | -0.000036182185 | 0.999992956155 | 0.000000000000 |
| Re=10000 | NonMPP | | MPP | |
| mesh | Umax | Umin | Umax | Umin |
| $16 \times 16$ | 0.874953790056 | -0.011212471543 | 0.846813512747 | 0.000000000000 |
| $32 \times 32$ | 0.973964125865 | -0.014299538733 | 0.942368749644 | 0.000000000000 |
| $64 \times 64$ | 1.000873875979 | -0.006640227946 | 0.988604733672 | 0.000000000000 |
| $128 \times 128$ | 1.002350640870 | -0.002755842119 | 0.999375840770 | 0.000000000000 |
| $256 \times 256$ | 1.000734372263 | -0.000563730690 | 0.999998986667 | 0.000000000000 |

Table 2.5.11: The maximum and minimum cell averages for swirling deformation problem (2.5.17) with two different Reynold numbers at T=0.1.

| Re=100 | NonMPP | | MPP | |
|---|---|---|---|---|
| mesh | Umax | Umin | Umax | Umin |
| $16 \times 16$ | 1.035853749815 | -1.035699868274 | 1.000000000000 | -1.000000000000 |
| $32 \times 32$ | 1.054573231517 | -1.054663726026 | 1.000000000000 | -1.000000000000 |
| $64 \times 64$ | 1.044017351861 | -1.044000125346 | 1.000000000000 | -1.000000000000 |
| $128 \times 128$ | 1.010637311054 | -1.010641150928 | 1.000000000000 | -1.000000000000 |
| $256 \times 256$ | 1.000000232315 | -1.000000231632 | 1.000000000000 | -1.000000000000 |
| Re=10000 | NonMPP | | MPP | |
| mesh | Umax | Umin | Umax | Umin |
| $16 \times 16$ | 1.036117022938 | -1.035951331163 | 1.000000000000 | -1.000000000000 |
| $32 \times 32$ | 1.060652217270 | -1.060764279809 | 1.000000000000 | -1.000000000000 |
| $64 \times 64$ | 1.086490500643 | -1.086296444198 | 1.000000000000 | -1.000000000000 |
| $128 \times 128$ | 1.127323843780 | -1.127407543973 | 1.000000000000 | -1.000000000000 |
| $256 \times 256$ | 1.129384376147 | -1.129395445889 | 1.000000000000 | -1.000000000000 |

Table 2.5.12: The maximum and minimum cell averages for vortex patch problem (2.5.18) at time T=0.1 with Re=100 and Re=10000.

# CHAPTER 3

# Integral Deferred Correction Method with Adaptive Non-Polynomial Basis

## 3.1 Introduction

In this chapter, we consider a new class of integral deferred correction (InDC) methods using adaptive non-polynomial basis for stiff time dependent problems to better capture sharp solution structures such as initial or internal layers. We will first briefly review the literature on classical time integrators for stiff and multi-scale ODE problems, the InDC framework and its development. After that, we will introduce

features about the proposed InDC methods with adaptive basis.

Developing efficient and effective time integrators for ODE systems has been a classical subject discussed in many books [30, 31]. Various types of time integrators (e.g., explicit multi-step method, multi-stage method, Taylor-series method or the so-called "Cauchy-Kowalewski procedure") with different features and advantages, have been shown to be very efficient in solving non-stiff ODEs. When the problem becomes stiff, implicit methods, integration factor (IF) method, and exponential time differencing method have been shown to be effective in resolving the stiffness by allowing large numerical time steps (independent of the stiffness parameter). However, some of these methods are well known for their restrictions. For example, the fully-implicit method for nonlinear ODE systems may require a nonlinear solver for large systems that could be computationally expensive. The integration factor and exponential time differencing methods are known to be effective for problems with only linear stiff terms. When the stiff terms become nonlinear (for example for the stiff Van der Pol system), the partitioned or additive implicit-explicit (IMEX) Runge-Kutta (RK) methods could be more computationally effective. Error estimates on the IMEX RK methods for a singular perturbation problem (SPP) are available in recent work [26], explaining the appearance of the order reduction phenomena. Another open challenge for stiff problems is to develop effective and efficient numerical integrators to resolve sharp layers such as initial layers and internal layers. To the best of the authors' knowledge, most of the numerical integrators nowadays assume that these layers are well resolved with very small mesh size, until the stage of smooth solution structure is reached. There is a recent attempt in resolving the initial layers

analytically as proposed in [40]. However, the proposed strategy there is restricted to the case where the stiff term is linear.

InDC method [27, 28], along with the deferred correction (DC) [25, 38] and spectral deferred correction (SDC) [29, 37, 36, 34, 35, 32] methods, is an automatic procedure of building up very high-order numerical integrators based on lower order ones for ODEs. The InDC procedure consists of one prediction step and several iteration steps for correction. The high-order accuracy is accomplished by using a lower order numerical method to solve a series of error equations in each correction step. In each correction iteration, an integration matrix, based on numerical quadratures derived from polynomial approximation, is built to approximate the residuals. Compared with the classical DC method, the recently developed SDC and InDC methods are based on Picard integral equation and a deferred correction procedure is applied to an integral formulation of the error equation in DC methods. It has been shown that SDC and InDC outperform DC in many problems with better stability and accuracy properties [29, 28]. The main difference between SDC and InDC is the distribution of quadrature nodes: the SDC method uses Gaussian/Lobatto/Radau points for better stability and accuracy properties, while InDC method uses uniform quadrature points to guarantee high-order accuracy increasing when high-order RK methods are applied in correction steps [28]. In [32], the authors pointed out that the SDC/InDC correction iterations converge to a collocation discretization of ODE problems. When the problem becomes stiff, the SDC and InDC methods require small time steps, which increases the cost of the computation. In [41], a scheme that combines exponential time differencing (ETD) method and SDC/InDC method was

proposed for effectively resolving the stiffness of stiff problems without refining the time-step size. However the scheme only works for linear stiff problems.

Traditional ODE integrators use approximation space with polynomial functions to construct high-order methods, e.g., the collocation methods at various collocation quadrature points [30, 31], RK methods, etc. In this work, we consider to augment the polynomial basis with some non-polynomial elementary functions such as an exponential function. The exponential function has an adaptive parameter which can be constructed via local solution structure. The new augmented basis leads to a new class of InDC methods that are able to better capture sharp solution structures such as initial layers and internal layers. Specifically, in the prediction step of the InDC framework, the exponential function $e^{\lambda t}$ with the stiffness parameter $\lambda$ is adaptively selected to better capture the dynamic solution structure. The new adaptive basis is then used to derive numerical quadratures for approximating residuals in correction iterations. When the InDC solutions converge with correction iterations, it would converge to a new class of collocation solution with the adaptive non-polynomial basis. For details, see discussions on the traditional polynomial basis in [32]. Note that the adaptive non-polynomial basis has been successfully developed in [39] in the discontinuous Galerkin framework for solving hyperbolic, parabolic problems with specific solution structures. We also comment that there are some recent development on using exponentials for solution approximations for stiff ODEs in the SDC framework [33]. Their idea of adaptively selecting proper exponential basis functions is via skeletonization, and the construction of basis functions is different from the proposed augmented polynomial basis in our work.

InDC scheme with adaptive non-polynomial basis has advantages over InDC scheme with polynomial basis, due to the fact that adaptive non-polynomial basis can approximate a function better than traditional polynomial basis. To be specific, smooth functions can be approximated by the non-polynomial basis as well as by polynomial basis because in this case the local adaptive parameter $\lambda$ is very close to 0, while functions that have sharp structures such as initial and internal layers can be approximated better by the adaptive non-polynomial basis than by polynomial basis because the former has the adaptive parameter $\lambda$ to automatically capture the sharp structure. At the same time, when the adaptive non-polynomial basis is incorporated into the InDC framework, the stability and accuracy regions turn out to be almost the same as those for polynomial InDC scheme, as is shown in Section 3.4 of this chapter. And it can be observed from numerical tests that the adaptive non-polynomial InDC scheme is comparable to polynomial InDC scheme if the ODE problem is not stiff or the initial condition is well-prepared (in this case the solution doesn't have sharp structures), while for ODE problems that have sharp solution structures, the adaptive non-polynomial InDC outperforms polynomial InDC, in the sense that the former with coarser meshes obtains the same error level as the latter with refined meshes does.

This chapter is organized as follows. In Section 3.2 we test and compare the performances of different bases in approximating functions that are smooth and functions that have layers. In Section 3.3, the InDC scheme with non-polynomial basis is presented and the stability and accuracy are studied in Section 3.4. Moreover, the adaptive step-size control technique is presented in Section 3.5. Then in Section

3.6 the adaptive non-polynomial InDC scheme is tested on several ODE examples.

## 3.2 Function Approximation with Adaptive Non-Polynomial Basis

In this section, we first introduce examples of approximation spaces as in [39]. We investigate and compare the performance of different approximation spaces in approximating functions of different shapes. Since our focus is the time integrator, we consider approximation in one-dimensional domain $[0, T]$ discretized as $\bigcup_n [t_{n-1}, t_n]$. On each interval $I_n = [t_{n-1}, t_n]$, we let the rescaled variable $\tau = 2\frac{t-t_{n-\frac{1}{2}}}{h_n} \in [-1, 1]$ with $t_{n-\frac{1}{2}}$ and $h_n$ being the mid-point and the length of the interval. Various approximation spaces we consider include the following.

1. The regular piecewise polynomial space:

$$B_1^M = \left\{ y : y|_{I_n} \in span\{1, \tau, \tau^2, \cdots, \tau^M\}, \forall n \right\}.$$

2. The exponential space I:

$$B_2^M(\lambda) = \left\{ y : y|_{I_n} \in span\{e^{\lambda\tau}, \tau, \tau^2, \cdots, \tau^M\}, \forall n \right\}.$$

3. The exponential space II:

$$B_3^M(\lambda) = \left\{ y : y|_{I_n} \in span\{e^{\lambda\tau}(1, \tau, \tau^2, \cdots, \tau^M)\}, \forall n \right\}.$$

4. The exponential space III:

$$B_4^M(\lambda) = \left\{ y : y|_{I_n} \in span\{1, \tau, \tau^2, \cdots, \tau^M, e^{\lambda\tau}\}, \forall n \right\}.$$

There are several remarks we would like to make in terms of finding the best function approximating a given function $f(t)$ using the approximation spaces specified above.

**Remark 3.2.1.** There is a parameter $\lambda$ in the exponential spaces. Below, we will consider the case when $\lambda$ is fixed (non-adaptive) and the case when $\lambda$ is adaptively selected according to the local function structures as in [39]. For the adaptive basis, we find a local parameter $\lambda_n$ for each cell $I_n$ such that the following quantity

$$\int_{I_n} [log|f(t)| - log(ce^{\lambda_n\tau})]^2 dt, \ c > 0, \tag{3.2.1}$$

is minimized. Taking the derivative of (3.2.1) with respect to $\lambda_n$ and setting the derivative to be zero, one gets

$$-\int_{I_n} 2[log|f(t)| - log|c| - \lambda_n\tau]\tau dt = 0 \tag{3.2.2}$$

which gives

$$\lambda_n = \frac{6}{h_n^2}\int_{I_n} (t - t_{n-\frac{1}{2}}) log|f(t)| dt. \tag{3.2.3}$$

The effectiveness of the above approach in identifying the adaptive parameter $\lambda_n$ has been extensively tested in [39]. Note that if the function $f(t)$ changes sign over interval $I_n$, then we propose to replace $log|f(t)|$ with $log(f(t) - \min_{t\in I_n} f(t))$ in eq. (3.2.1). When the function being approximated undergoes mild changes, then

the adaptive $\lambda$ would be chosen to close to 0, then the exponential basis $B_2^M(\lambda)$ and $B_3^M(\lambda)$ is very close to the regular polynomial basis.

**Remark 3.2.2.** The function in the approximation space is chosen as the $L^2$ projection of the original function $f$ into a approximation space with basis $B = \{\phi_1, \cdots, \phi_M\}$, denoted as $\mathcal{P}_h f$. In order to perform the $L^2$ projection, one needs to perform integration for the mass matrix $\mathcal{M}_{M\times M} = (m_{ij})$ with $m_{ij} = \int_{I_n} \phi_i \phi_j dt$ as well as a column vector $\vec{b}_{M\times 1}$ with $b_j = \int_{I_n} f(t)\phi_j dt$. Usually, the integrations are performed via numerical quadrature rules. However, when the parameter $|\lambda_n|$ is large, corresponding to fast decay or growth of the solution over a small interval, then a more careful integration (either exact integration or numerical integration with a smaller resolution scale) needs to be performed. The $L^2$ projection of the function $f$ can be expressed as

$$\mathcal{P}_h f = (\phi_1, \cdots, \phi_M)(\mathcal{M}^{-1}\vec{b}) \approx f(t), \quad \text{on} \quad I_n.$$

**Remark 3.2.3.** For the exponential space III $B_4$, the mass matrix $M$ will be ill-conditioned if $|\lambda|$ is too small, as the two bases $e^{\lambda\tau}$ and 1 become very close to each other (linearly dependent). A threshold value $\lambda_{threshold}$ is set to prevent the ill-conditioning of the mass matrix $M$. Specifically, when $|\lambda| < \lambda_{threshold}$, we set $\lambda = sign(\lambda)\lambda_{threshold}$. In our numerical tests, we let $\lambda_{threshold} = \frac{1}{2}$.

**Remark 3.2.4.** In [39], it is proved that if each of the basis functions $\phi_i$, $i = 1, \cdots k$ can be well-approximated by polynomials, then the $L^2$ projection of the function onto the approximation space $\mathcal{P}_h(f)$ is a high-order approximation to the original function $f$, see Proposition 3.2 in [39].

In tables 3.2.1—3.2.4, we present the function approximation via the $L^2$ projection onto the approximation space $B_1$, $B_2$, $B_3$ and $B_4$ and compare their performances. We consider the exponential space using a fixed parameter $\lambda = Ch_n$ where $C = -1$ and with adaptive selection of parameter according to Remark 3.2.1. The functions being approximated include a smooth one $f(t) = \sin(t)$, as well as functions with sharp layers $f(t) = e^{-50t}$, $f(t) = te^{-50t}$ and $f(t) = 1/(1+exp(-100t))+cos(\pi t)$. In general, an adaptive exponential basis performs better than the same space with a fixed parameter $\lambda$. Expected $k+1$-th order accuracy is observed for $B_1$, $B_2$ and $B_3$ approximation space, and $k+2$-th order accuracy is observed for the $B_4$ basis when the function being approximated is smooth and well resolved, e.g. $f(t) = \sin(t)$. When the functions being approximated have sharp layers, we compare the order of magnitudes of errors for different approximation spaces. Again, the adaptive non-polynomial space performs better than the non-adaptive one. It is also observed that, in general the adaptive exponential bases perform better (smaller error in magnitude) than the regular polynomial basis, especially when the mesh resolution is coarse.

| $B_1^k$ | | | | | | |
|---|---|---|---|---|---|---|
| | k=1 | | k=2 | | k=3 | |
| N | $L_1$ error | order | $L_1$ error | order | $L_1$ error | order |
| 20 | 2.01e-003 | — | 6.32e-005 | — | 1.17e-006 | — |
| 40 | 5.02e-004 | 2.00 | 7.89e-006 | 3.00 | 7.28e-008 | 4.00 |
| 80 | 1.25e-004 | 2.00 | 9.85e-007 | 3.00 | 4.55e-009 | 4.00 |
| 160 | 3.14e-005 | 2.00 | 1.23e-007 | 3.00 | 2.84e-010 | 4.00 |
| $B_2^k$ | | | | | | |
| | k=1 | | k=2 | | k=3 | |
| N | $L_1$ error | order | $L_1$ error | order | $L_1$ error | order |
| 20 | 9.46e-003 | — | 5.14e-004 | — | 1.73e-005 | — |
| 40 | 2.44e-003 | 1.95 | 6.40e-005 | 3.01 | 1.08e-006 | 4.00 |
| 80 | 6.19e-004 | 1.98 | 7.96e-006 | 3.01 | 6.79e-008 | 4.00 |
| 160 | 1.56e-004 | 1.99 | 9.94e-007 | 3.00 | 4.25e-009 | 4.00 |
| Adaptive $B_2^k$ | | | | | | |
| | k=1 | | k=2 | | k=3 | |
| N | $L_1$ error | order | $L_1$ error | order | $L_1$ error | order |
| 20 | 2.12e-003 | — | 6.43e-005 | — | 1.16e-006 | — |
| 40 | 5.28e-004 | 2.00 | 8.01e-006 | 3.00 | 7.24e-008 | 4.00 |
| 80 | 1.32e-004 | 2.00 | 1.00e-006 | 3.00 | 4.52e-009 | 4.00 |
| 160 | 3.30e-005 | 2.00 | 1.25e-007 | 3.00 | 2.83e-010 | 4.00 |
| $B_3^k$ | | | | | | |
| | k=1 | | k=2 | | k=3 | |
| N | $L_1$ error | order | $L_1$ error | order | $L_1$ error | order |
| 20 | 7.56e-003 | — | 7.94e-004 | — | 2.56e-005 | — |
| 40 | 2.21e-003 | 1.77 | 9.48e-005 | 3.07 | 1.68e-006 | 3.93 |
| 80 | 5.91e-004 | 1.90 | 1.15e-005 | 3.05 | 1.09e-007 | 3.94 |
| 160 | 1.52e-004 | 1.96 | 1.40e-006 | 3.03 | 6.97e-009 | 3.97 |
| Adaptive $B_3^k$ | | | | | | |
| | k=1 | | k=2 | | k=3 | |
| N | $L_1$ error | order | $L_1$ error | order | $L_1$ error | order |
| 20 | 3.21e-003 | — | 5.93e-006 | — | 2.11e-006 | — |
| 40 | 8.05e-004 | 2.00 | 6.93e-007 | 3.10 | 1.32e-007 | 4.00 |
| 80 | 2.01e-004 | 2.00 | 8.39e-008 | 3.05 | 8.26e-009 | 4.00 |
| 160 | 5.03e-005 | 2.00 | 1.04e-008 | 3.02 | 5.16e-010 | 4.00 |
| $B_4^k$ | | | | | | |
| | k=1 | | k=2 | | k=3 | |
| N | $L_1$ error | order | $L_1$ error | order | $L_1$ error | order |
| 20 | 2.27e-004 | — | 3.66e-006 | — | 8.93e-008 | — |
| 40 | 5.48e-005 | 2.05 | 4.41e-007 | 3.05 | 5.38e-009 | 4.05 |
| 80 | 1.36e-005 | 2.01 | 5.46e-008 | 3.01 | 3.33e-010 | 4.01 |
| 160 | 3.38e-006 | 2.00 | 6.80e-009 | 3.00 | 2.07e-011 | 4.01 |
| Adaptive $B_4^k$ | | | | | | |
| | k=1 | | k=2 | | k=3 | |
| N | $L_1$ error | order | $L_1$ error | order | $L_1$ error | order |
| 20 | 1.56e-004 | — | 4.86e-006 | — | 6.76e-008 | — |
| 40 | 4.31e-005 | 1.85 | 5.34e-007 | 3.19 | 4.67e-009 | 3.85 |
| 80 | 1.15e-005 | 1.90 | 6.21e-008 | 3.10 | 3.12e-010 | 3.90 |
| 160 | 2.99e-006 | 1.95 | 7.48e-009 | 3.05 | 2.02e-011 | 3.95 |

Table 3.2.1: Comparison of different bases for approximating $sin(t),\ t \in [0, 2\pi]$.

| $B_1^k$ | | | | | | |
|---|---|---|---|---|---|---|
| | k=1 | | k=2 | | k=3 | |
| N | $L_1$ error | order | $L_1$ error | order | $L_1$ error | order |
| 20 | 4.29e-003 | — | 8.16e-004 | — | 1.32e-004 | — |
| 40 | 1.08e-003 | 1.99 | 1.17e-004 | 2.80 | 9.07e-006 | 3.87 |
| 80 | 2.62e-004 | 2.04 | 1.53e-005 | 2.94 | 5.76e-007 | 3.98 |
| 160 | 6.42e-005 | 2.03 | 1.93e-006 | 2.98 | 3.60e-008 | 4.00 |
| $B_2^k$ | | | | | | |
| | k=1 | | k=2 | | k=3 | |
| N | $L_1$ error | order | $L_1$ error | order | $L_1$ error | order |
| 20 | 4.29e-003 | — | 8.16e-004 | — | 1.32e-004 | — |
| 40 | 1.08e-003 | 1.99 | 1.17e-004 | 2.80 | 9.07e-006 | 3.87 |
| 80 | 2.62e-004 | 2.04 | 1.53e-005 | 2.94 | 5.76e-007 | 3.98 |
| 160 | 6.41e-005 | 2.03 | 1.93e-006 | 2.98 | 3.60e-008 | 4.00 |
| Adaptive $B_2^k$ | | | | | | |
| | k=1 | | k=2 | | k=3 | |
| N | $L_1$ error | order | $L_1$ error | order | $L_1$ error | order |
| 20 | 4.22e-003 | — | 8.15e-004 | — | 1.32e-004 | — |
| 40 | 1.04e-003 | 2.02 | 1.16e-004 | 2.81 | 9.06e-006 | 3.87 |
| 80 | 2.52e-004 | 2.05 | 1.51e-005 | 2.94 | 5.75e-007 | 3.98 |
| 160 | 6.14e-005 | 2.04 | 1.91e-006 | 2.98 | 3.59e-008 | 4.00 |
| $B_3^k$ | | | | | | |
| | k=1 | | k=2 | | k=3 | |
| N | $L_1$ error | order | $L_1$ error | order | $L_1$ error | order |
| 20 | 4.53e-003 | — | 9.45e-004 | — | 1.52e-004 | — |
| 40 | 1.15e-003 | 1.98 | 1.34e-004 | 2.82 | 1.05e-005 | 3.85 |
| 80 | 2.82e-004 | 2.03 | 1.73e-005 | 2.95 | 6.70e-007 | 3.97 |
| 160 | 6.91e-005 | 2.03 | 2.18e-006 | 2.99 | 4.20e-008 | 4.00 |
| Adaptive $B_3^k$ | | | | | | |
| | k=1 | | k=2 | | k=3 | |
| N | $L_1$ error | order | $L_1$ error | order | $L_1$ error | order |
| 20 | 3.50e-003 | — | 4.90e-004 | — | 8.15e-005 | — |
| 40 | 7.91e-004 | 2.15 | 6.47e-005 | 2.92 | 4.68e-006 | 4.12 |
| 80 | 1.83e-004 | 2.11 | 8.36e-006 | 2.95 | 2.80e-007 | 4.06 |
| 160 | 4.36e-005 | 2.06 | 1.06e-006 | 2.97 | 1.71e-008 | 4.03 |
| $B_4^k$ | | | | | | |
| | k=1 | | k=2 | | k=3 | |
| N | $L_1$ error | order | $L_1$ error | order | $L_1$ error | order |
| 20 | 1.24e-003 | — | 1.75e-004 | — | 3.21e-005 | — |
| 40 | 2.28e-004 | 2.45 | 1.54e-005 | 3.51 | 1.46e-006 | 4.46 |
| 80 | 4.29e-005 | 2.41 | 1.41e-006 | 3.45 | 6.77e-008 | 4.43 |
| 160 | 8.76e-006 | 2.29 | 1.42e-007 | 3.31 | 3.43e-009 | 4.30 |
| Adaptive $B_4^k$ | | | | | | |
| | k=1 | | k=2 | | k=3 | |
| N | $L_1$ error | order | $L_1$ error | order | $L_1$ error | order |
| 20 | 4.36e-004 | — | 8.29e-005 | — | 1.41e-005 | — |
| 40 | 2.10e-005 | 4.38 | 1.90e-006 | 5.45 | 1.67e-007 | 6.41 |
| 80 | 8.23e-006 | 1.35 | 3.63e-007 | 2.39 | 1.61e-008 | 3.37 |
| 160 | 3.83e-006 | 1.10 | 8.31e-008 | 2.12 | 1.85e-009 | 3.12 |

Table 3.2.2: Comparison of different bases for $exp(-50t),\ t \in [0, 1]$.

| $B_1^k$ | | | | | | |
|---|---|---|---|---|---|---|
| | k=1 | | k=2 | | k=3 | |
| N | $L_1$ error | order | $L_1$ error | order | $L_1$ error | order |
| 20 | 1.01e-004 | — | 2.94e-005 | — | 7.35e-006 | — |
| 40 | 2.75e-005 | 1.88 | 4.68e-006 | 2.65 | 5.41e-007 | 3.76 |
| 80 | 6.80e-006 | 2.02 | 6.32e-007 | 2.89 | 3.49e-008 | 3.95 |
| 160 | 1.65e-006 | 2.04 | 8.08e-008 | 2.97 | 2.19e-009 | 4.00 |
| $B_2^k$ | | | | | | |
| | k=1 | | k=2 | | k=3 | |
| N | $L_1$ error | order | $L_1$ error | order | $L_1$ error | order |
| 20 | 1.01e-004 | — | 2.94e-005 | — | 7.35e-006 | — |
| 40 | 2.75e-005 | 1.88 | 4.68e-006 | 2.65 | 5.41e-007 | 3.76 |
| 80 | 6.80e-006 | 2.02 | 6.32e-007 | 2.89 | 3.49e-008 | 3.95 |
| 160 | 1.65e-006 | 2.04 | 8.08e-008 | 2.97 | 2.19e-009 | 4.00 |
| Adaptive $B_2^k$ | | | | | | |
| | k=1 | | k=2 | | k=3 | |
| N | $L_1$ error | order | $L_1$ error | order | $L_1$ error | order |
| 20 | 1.01e-004 | — | 2.94e-005 | — | 7.35e-006 | — |
| 40 | 2.75e-005 | 1.88 | 4.68e-006 | 2.65 | 5.41e-007 | 3.76 |
| 80 | 6.80e-006 | 2.02 | 6.32e-007 | 2.89 | 3.49e-008 | 3.95 |
| 160 | 1.65e-006 | 2.04 | 8.08e-008 | 2.97 | 2.19e-009 | 4.00 |
| $B_3^k$ | | | | | | |
| | k=1 | | k=2 | | k=3 | |
| N | $L_1$ error | order | $L_1$ error | order | $L_1$ error | order |
| 20 | 1.02e-004 | — | 3.23e-005 | — | 7.97e-006 | — |
| 40 | 2.79e-005 | 1.86 | 5.08e-006 | 2.67 | 5.96e-007 | 3.74 |
| 80 | 7.00e-006 | 1.99 | 6.79e-007 | 2.90 | 3.87e-008 | 3.95 |
| 160 | 1.70e-006 | 2.04 | 8.67e-008 | 2.97 | 2.43e-009 | 3.99 |
| Adaptive $B_3^k$ | | | | | | |
| | k=1 | | k=2 | | k=3 | |
| N | $L_1$ error | order | $L_1$ error | order | $L_1$ error | order |
| 20 | 1.01e-004 | — | 2.94e-005 | — | 7.35e-006 | — |
| 40 | 2.75e-005 | 1.88 | 4.70e-006 | 2.65 | 5.44e-007 | 3.76 |
| 80 | 6.81e-006 | 2.01 | 6.36e-007 | 2.88 | 3.52e-008 | 3.95 |
| 160 | 1.66e-006 | 2.04 | 8.14e-008 | 2.97 | 2.21e-009 | 3.99 |
| $B_4^k$ | | | | | | |
| | k=1 | | k=2 | | k=3 | |
| N | $L_1$ error | order | $L_1$ error | order | $L_1$ error | order |
| 20 | 3.91e-005 | — | 8.78e-006 | — | 2.19e-006 | — |
| 40 | 7.44e-006 | 2.39 | 7.89e-007 | 3.48 | 1.02e-007 | 4.43 |
| 80 | 1.32e-006 | 2.49 | 6.90e-008 | 3.52 | 4.59e-009 | 4.47 |
| 160 | 2.53e-007 | 2.39 | 6.60e-009 | 3.39 | 2.24e-010 | 4.36 |
| Adaptive $B_4^k$ | | | | | | |
| | k=1 | | k=2 | | k=3 | |
| N | $L_1$ error | order | $L_1$ error | order | $L_1$ error | order |
| 20 | 3.71e-005 | — | 8.82e-006 | — | 2.17e-006 | — |
| 40 | 6.94e-006 | 2.42 | 7.22e-007 | 3.61 | 8.91e-008 | 4.61 |
| 80 | 1.23e-006 | 2.50 | 5.99e-008 | 3.59 | 3.80e-009 | 4.55 |
| 160 | 2.16e-007 | 2.51 | 5.58e-009 | 3.42 | 1.86e-010 | 4.36 |

Table 3.2.3: Comparison of different bases for $texp(-50t),\ t \in [0,1]$.

| $B_1^k$ | | | | | | |
|---|---|---|---|---|---|---|
| | k=1 | | k=2 | | k=3 | |
| N | $L_1$ error | order | $L_1$ error | order | $L_1$ error | order |
| 20 | 7.40e-003 | — | 3.59e-003 | — | 1.02e-003 | — |
| 40 | 2.81e-003 | 1.40 | 5.25e-004 | 2.77 | 1.04e-004 | 3.29 |
| 80 | 6.66e-004 | 2.08 | 5.09e-005 | 3.37 | 2.00e-005 | 2.38 |
| 160 | 1.55e-004 | 2.10 | 1.27e-005 | 2.01 | 1.02e-006 | 4.29 |
| $B_2^k$ | | | | | | |
| | k=1 | | k=2 | | k=3 | |
| N | $L_1$ error | order | $L_1$ error | order | $L_1$ error | order |
| 20 | 8.16e-003 | — | 3.58e-003 | — | 1.02e-003 | — |
| 40 | 3.01e-003 | 1.44 | 5.23e-004 | 2.77 | 1.04e-004 | 3.29 |
| 80 | 7.14e-004 | 2.08 | 5.07e-005 | 3.37 | 2.00e-005 | 2.38 |
| 160 | 1.67e-004 | 2.09 | 1.26e-005 | 2.01 | 1.02e-006 | 4.29 |
| Adaptive $B_2^k$ | | | | | | |
| | k=1 | | k=2 | | k=3 | |
| N | $L_1$ error | order | $L_1$ error | order | $L_1$ error | order |
| 20 | 7.47e-003 | — | 3.59e-003 | — | 1.02e-003 | — |
| 40 | 2.83e-003 | 1.40 | 5.25e-004 | 2.77 | 1.04e-004 | 3.29 |
| 80 | 6.71e-004 | 2.08 | 5.11e-005 | 3.36 | 2.00e-005 | 2.38 |
| 160 | 1.56e-004 | 2.10 | 1.27e-005 | 2.01 | 1.02e-006 | 4.29 |
| $B_3^k$ | | | | | | |
| | k=1 | | k=2 | | k=3 | |
| N | $L_1$ error | order | $L_1$ error | order | $L_1$ error | order |
| 20 | 8.54e-003 | — | 3.80e-003 | — | 1.05e-003 | — |
| 40 | 3.09e-003 | 1.47 | 5.44e-004 | 2.80 | 1.06e-004 | 3.30 |
| 80 | 7.44e-004 | 2.05 | 5.30e-005 | 3.36 | 1.99e-005 | 2.42 |
| 160 | 1.76e-004 | 2.08 | 1.28e-005 | 2.05 | 1.02e-006 | 4.29 |
| Adaptive $B_3^k$ | | | | | | |
| | k=1 | | k=2 | | k=3 | |
| N | $L_1$ error | order | $L_1$ error | order | $L_1$ error | order |
| 20 | 8.08e-003 | — | 3.60e-003 | — | 1.02e-003 | — |
| 40 | 2.91e-003 | 1.47 | 5.29e-004 | 2.77 | 1.08e-004 | 3.24 |
| 80 | 6.93e-004 | 2.07 | 5.65e-005 | 3.23 | 1.95e-005 | 2.47 |
| 160 | 1.63e-004 | 2.09 | 1.25e-005 | 2.17 | 9.99e-007 | 4.29 |
| $B_4^k$ | | | | | | |
| | k=1 | | k=2 | | k=3 | |
| N | $L_1$ error | order | $L_1$ error | order | $L_1$ error | order |
| 20 | 3.99e-003 | — | 1.04e-003 | — | 4.07e-004 | — |
| 40 | 6.12e-004 | 2.71 | 1.16e-004 | 3.17 | 9.66e-005 | 2.08 |
| 80 | 9.37e-005 | 2.71 | 1.94e-005 | 2.57 | 2.81e-006 | 5.10 |
| 160 | 1.94e-005 | 2.27 | 1.03e-006 | 4.24 | 1.17e-007 | 4.58 |
| Adaptive $B_4^k$ | | | | | | |
| | k=1 | | k=2 | | k=3 | |
| N | $L_1$ error | order | $L_1$ error | order | $L_1$ error | order |
| 20 | 3.94e-003 | — | 1.04e-003 | — | 4.07e-004 | — |
| 40 | 5.94e-004 | 2.73 | 1.15e-004 | 3.18 | 9.63e-005 | 2.08 |
| 80 | 8.97e-005 | 2.73 | 1.93e-005 | 2.57 | 2.78e-006 | 5.11 |
| 160 | 1.86e-005 | 2.27 | 1.00e-006 | 4.26 | 1.15e-007 | 4.59 |

Table 3.2.4: Comparison of different bases for $1/(1 + exp(-100t)) + cos(\pi t),\ t \in [-1, 1]$.

## 3.3 InDC Method with Adaptive Non-Polynomial Basis

### 3.3.1 Review of the Traditional SDC/InDC Methods

We review the SDC/InDC methods for a standard initial-value problem

$$y' = f(t, y), \quad t \in [0, T], \quad \text{with the initial data} \quad y(0) = y_0. \tag{3.3.1}$$

The time domain $[0, T]$ is discretized into intervals

$$0 = t_1 < t_2 < \cdots < t_n < \cdots < t_N = T,$$

and each interval $I_n = [t_{n-1}, t_n]$ is further discretized into $M$ sub-intervals

$$t_{n-1} = t_{n,0} = t_{n,1} < \cdots < t_{n,m} < \cdots < t_{n,M} = t_n. \tag{3.3.2}$$

We let $\tau_m := t_{n,m}$, $\forall m = 0, \cdots M$ in (3.3.2), and refer to them as quadrature nodes. When these quadrature points are Gaussian points, the method is called the SDC method [29]; when the quadrature points are equally spaced, then the method is called the InDC method [28]. For each time interval $I_n$, let $H \doteq t_n - t_{n-1}$ and $h = H/M$, then the InDC method on $I_n$ is described below.

- (prediction step) Use a low-order numerical integrator to obtain a numerical solution, $\vec{\eta}^{[0]} = (\eta_0^{[0]}, \ldots, \eta_M^{[0]})$, which is a low-order approximation to the exact

solution at quadrature points $\tau_m$. For example, applying a first-order forward Euler method to (3.3.1) gives $\eta_{m+1}^{[0]} = \eta_m^{[0]} + hf(\tau_m, \eta_m^{[0]})$, $m = 0, \ldots, M-1$.

- (correction loop) Use the error function to improve the accuracy of the scheme at each iteration. For $k = 1, \ldots, K$ ($K$ is number of correction steps),

  1. Denote the *error function* from the previous step as

  $$e^{(k-1)}(t) = y(t) - \eta^{(k-1)}(t), \tag{3.3.3}$$

  where $y(t)$ is the exact solution and $\eta^{(k-1)}(t)$ is an $M^{th}$ degree polynomial interpolating $\vec{\eta}^{[k-1]}$. Note that the error function, $e^{(k-1)}(t)$, is not a polynomial in general.

  2. Let the *residual function* be $\epsilon^{(k-1)}(t) = (\eta^{(k-1)})'(t) - f(t, \eta^{(k-1)}(t))$.

  3. Compute the *numerical error vector*, $\vec{\delta}^{[k]} = (\delta_0^{[k]}, \ldots, \delta_m^{[k]}, \ldots, \delta_M^{[k]})$, using a low-order numerical method to discretize the integral form of the *error equation*,

  $$\left(e^{(k-1)} + \int_0^t \epsilon^{(k-1)}(\tau)\, d\tau\right)'(t) = f(t, \eta^{(k-1)}(t) + e^{(k-1)}(t)) - f(t, \eta^{(k-1)}(t)). \tag{3.3.4}$$

  Let $\delta_m^{[k]}$ be the corresponding numerical approximation to the exact error function at $\tau_m$. For example, applying a first-order forward Euler method to (3.3.4) gives,

  $$\delta_{m+1}^{[k]} = \delta_m^{[k]} + h(f(\tau_m, \eta_m^{[k-1]} + \delta_m^{[k]}) - f(\tau_m, \eta_m^{[k-1]})) - \int_{\tau_m}^{\tau_{m+1}} \epsilon^{(k-1)}(t)\, dt,$$

$$m = 0, \ldots, M-1, \tag{3.3.5}$$

where the integral term $\int_{\tau_m}^{\tau_{m+1}}$ in the above equation is approximated by

$$\int_{\tau_m}^{\tau_{m+1}} \epsilon^{(k-1)}(t)\, dt = \eta_{m+1}^{[k-1]} - \eta_m^{[k-1]} - \sum_j S_{m,j} f(t_j, \eta_j^{[k-1]})$$

with the Lagrangian polynomial integration coefficients

$$S_{m,j} = \frac{1}{h} \int_{\tau_m}^{\tau_{m+1}} \alpha_j(s) ds, \quad \text{for} \quad m = 0, \cdots, M-1, \quad j = 1, \cdots M,$$

where $\alpha_j(s) = \prod_i \frac{s-\tau_i}{\tau_j - \tau_i}$ is the Lagrangian polynomial basis function based on the node $\tau_j$, $j = 1, \cdots M$. Let

$$S^m(\vec{f}) = \sum_{j=1}^{M} S_{m,j} f(t_j, y_j), \tag{3.3.6}$$

then

$$hS^m(\vec{f}) - \int_{\tau_m}^{\tau_{m+1}} f(s, y(s)) ds = \mathcal{O}(h^{M+1}),$$

for any smooth function $f$. In other words, the quadrature formula given by $hS^m(\vec{f})$ approximates the exact integration with $(M+1)^{th}$ order of accuracy *locally*.

4. Update the numerical solution $\vec{\eta}^{[k]} = \vec{\eta}^{[k-1]} + \vec{\delta}^{[k]}$.

Notationally, superscripts with a round bracket, e.g., $(k)$, denote a function, while superscripts with a square bracket, e.g., $[k]$, denote a vector at the $k^{th}$ correction step.

English letters are reserved for functions or vectors in the exact solution space, e.g., an exact solution $y(t)$ and an exact error function $e(t)$, while Greek letters denote functions or vectors in the numerical solution space, e.g., a numerical solution $\eta(t)$, and a numerical error function $\delta(t)$.

**Remark 3.3.1.** It was pointed out in [32] that if the SDC/InDC correction iteration converges, it converges to the collocation solution with given quadrature nodes $(\tau_1, \cdots, \tau_M)$. Specifically, let $\vec{\eta} = \lim_{k\to\infty} \vec{\eta}^{[k-1]}$, then $\vec{\eta} = \eta_0 \mathbf{1} + S\vec{\eta}$, where $\mathbf{1} = (1, \cdots, 1)'$ is of size $M \times 1$ and $S$ is the integration matrix of size $M \times M$.

### 3.3.2 InDC Methods with Adaptive Non-Polynomial Basis

When the ODE problem (3.3.1) becomes stiff or multi-scale, many numerical challenges need to be addressed. For example, one may apply implicit or exponential time differencing methods to ensure stability and accuracy of time integrators with large time-step size. For layer structures, very-fine resolution is usually needed to resolve the layer. Observing the fast growth and decay of the solution structure around layers, we propose to use an adaptive exponential basis as discussed in Section 3.2, rather than the standard polynomial basis in constructing numerical integrators in the InDC framework. Specifically, we propose the following modification to the standard InDC algorithm. We take the exponential space $B_2^M(\lambda)$ with adaptive choice of $\lambda$ as an example to illustrate the idea, while the algorithm can be readily generalized to other adaptive non-polynomial spaces.

- (prediction step) The prediction step remains the same as the standard InDC

algorithm. When the problem becomes stiff or multi-scale, an implicit, implicit-explicit or exponential time differencing method may be used.

- (correction loop) In the modified scheme, we first need to find an appropriate parameter $\lambda$ in exponential basis $e^{\lambda\tau}$ by fitting the solution from prediction or the previous correction iteration. The fitting procedure would be the same as that outlined in Remark 3.2.1, where the function values are available as approximations at quadrature points. The numerical solution and integration of residual are then approximated by using the exponential basis space, rather than the standard polynomials. Specifically, the modified procedure for the correction iteration is outlined below. For $k = 1, \ldots, K$ ($K$ is number of correction steps),

  1. Find an adaptive parameter $\lambda$ in a time-step evolution via

$$\lambda = \frac{6}{H^2} \int_{t_{n-1}}^{t_n} (t - t_{n-\frac{1}{2}}) log|\eta^{(k-1)}(t)|dt, \tag{3.3.7}$$

     where the integration is numerically approximated by a regular quadrature rule based on uniform nodes. Note that since *log* scale is taken, the regular quadrature rule based on polynomial approximation is appropriate.

  2. Denote the *error function* from the previous step as

$$e^{(k-1)}(t) = y(t) - \eta^{(k-1)}(t), \tag{3.3.8}$$

     where $y(t)$ is the exact solution and $\eta^{(k-1)}(t) \in B_2^M(\lambda)$ interpolates $\vec{\eta}^{[k-1]}$.

3. Let the *residual function* be $\epsilon^{(k-1)}(t) = (\eta^{(k-1)})'(t) - f(t, \eta^{(k-1)}(t))$.

4. Compute the *numerical error vector*, $\vec{\delta}^{[k]} = (\delta_0^{[k]}, \ldots, \delta_m^{[k]}, \ldots, \delta_M^{[k]})$, using a low-order numerical method to discretize the integral form of the *error equation*,

$$\left(e^{(k-1)} + \int_0^t \epsilon^{(k-1)}(\tau)\, d\tau\right)'(t) = f(t, \eta^{(k-1)}(t) + e^{(k-1)}(t)) - f(t, \eta^{(k-1)}(t)). \tag{3.3.9}$$

Let $\delta_m^{[k]}$ be the corresponding numerical approximation to the exact error function at $\tau_m$. For example, applying a first-order forward Euler method to (3.3.9) gives,

$$\delta_{m+1}^{[k]} = \delta_m^{[k]} + h(f(\tau_m, \eta_m^{[k-1]} + \delta_m^{[k]}) - f(\tau_m, \eta_m^{[k-1]})) - \int_{\tau_m}^{\tau_{m+1}} \epsilon^{(k-1)}(t)\, dt, \quad m = 0, \ldots, M-1, \tag{3.3.10}$$

where integral term $\int_{\tau_m}^{\tau_{m+1}}$ for the residual in the above equation is approximated by

$$\int_{\tau_m}^{\tau_{m+1}} \epsilon^{(k-1)}(t)\, dt = \eta_{m+1}^{[k-1]} - \eta_m^{[k-1]} - \sum_j S_{m,j}^{\lambda} f(t_j, \eta_j^{[k-1]})$$

where the integration coefficients $S^{\lambda}_{m,j}$ are obtained from integrating functions in $B^M_2(\lambda)$ interpolating $\vec{\eta}^{(k-1)}$. Specifically,

$$S^{\lambda}_{m,j} = \frac{1}{h}\int_{\tau_m}^{\tau_{m+1}} \alpha^{\lambda}_j(s)ds, \quad \text{for} \quad m = 0, \cdots, M-1, \quad j = 1, \cdots M, \tag{3.3.11}$$

where $\alpha^{\lambda}_j(s) \in B^M_2(\lambda)$ is the Lagrangian basis function based on the node $\tau_j$ satisfying $\alpha^{\lambda}_j(\tau_i) = \delta_{ij}$. Similar to the polynomial basis, the quadrature formula based on exponential basis approximates the exact integration with $(M+1)^{th}$ order of accuracy *locally*.

5. Update the numerical solution $\vec{\eta}^{[k]} = \vec{\eta}^{[k-1]} + \vec{\delta}^{[k]}$.

**Remark 3.3.2.** We remark that, for smooth solutions with well resolved time-step resolution, the numerical solution changes mildly over the time interval. The adaptive $\lambda$ could be close to 0. In this case, the exponential basis $B^M_2(\lambda)$ behaves in a very similar way to the regular polynomial basis. On the other hand, when the solution undergoes rapid change over a time-step, the adaptive $\lambda$ could be away from zero to better capture sharp transition of the solution.

**Remark 3.3.3.** If the InDC correction iterations converge, using similar argument as those in [32], the InDC solution converges to the exponential basis collocation solution with given quadrature nodes $(\tau_1, \cdots, \tau_M)$. Specifically, let $\vec{\eta} = \lim_{k\to\infty} \vec{\eta}^{[k-1]}$, then $\vec{\eta} = \eta_0 \mathbf{1} + S^{\lambda}\vec{\eta}$, where $\mathbf{1} = (1, \cdots, 1)'$ is of size $M \times 1$ and $S^{\lambda}$ is the integration matrix of size $M \times M$ constructed based on an exponential basis parameterized by $\lambda$.

## 3.4 Stability and Accuracy Properties

In this section, we investigate the stability and accuracy properties of the InDC methods using adaptive exponential bases via standard linear analysis. Below, we first introduce the concepts of stability and accuracy regions ([28]), which are widely used as a measurement to compare performance of different numerical integrators.

**Definition 3.4.1.** The amplification factor for a numerical method, $Am(\lambda)$, can be interpreted as the numerical solution to

$$y'(t) = \lambda y(t), \quad y(0) = 1, \tag{3.4.1}$$

after one time-step of size 1 for $\lambda \in \mathbb{C}$, i.e., $Am(\lambda) = y(1)$.

**Definition 3.4.2.** The stability region, $S$, for a numerical method, is the subset of the complex plane $\mathbb{C}$, consisting of all $\lambda$ such that $Am(\lambda) \leq 1$,

$$S = \{\lambda : Am(\lambda) \leq 1\}.$$

**Definition 3.4.3.** Let $e(\lambda)$ be the error at $T = 1$, obtained using a numerical method to solve IVP (3.4.1), $\lambda \in \mathbb{C}$, with a fixed number of function evaluations (i.e., $dt$ is chosen so that the total number of function evaluations for the method can be controlled). Then, the accuracy plot for that numerical method is defined to be a contour plot of the error, $e(\lambda)$.

We study the stability and accuracy properties of the InDC schemes with different lower-order schemes in the prediction and correction steps. For example, if forward

Euler scheme is used at the prediction and correction steps, then the scheme is denoted as $FEInDC$. So if backward Euler or implicit-explicit (IMEX) schemes are used, we have $BEInDC$ or $IMEXInDC$ schemes respectively. Suppose the dimension of the approximation space is $M$, and the number of correction steps is $J$, then the corresponding InDC scheme can be denoted as $FEInDC^J_M$, $BEInDC^J_M$, and $IMEXInDC^J_M$ respectively. For example, if the quadrature nodes for $I_n$ are $t_{n-1}+(0,\ 1/4,\ 2/4,\ 3/4,\ 1)h$ with $h=(t_n-t_{n-1})/4$, then $M=5$ and $J$ takes values of $0,\ 1,\ 2,\ 3,\ 4$ and if the quadrature nodes for $I_n$ are $t_{n-1}+(1/4,\ 2/4,\ 3/4,\ 1)h$ with $h=(t_n-t_{n-1})/4$, then $M=4$ and $J$ takes values of $0,\ 1,\ 2,\ 3$.

The stability regions for various InDC schemes with different approximation bases are presented in figures (3.4.1)—(3.4.6), from which the following observations can be made.

1. For $FEInDC$ schemes, the stability regions are the domains enclosed by the closed curves in figures (3.4.1) and (3.4.2). For $FEInDC$ with a given basis, the stability region shrinks as $J$ (the number of correction steps) increases. And in general, the stability regions for $FEInDC$ with adaptive B2, adaptive B2 and adaptive B3 are comparable to that for $FEInDC$ with polynomial basis B1.

2. For $BEInCD$ schemes, the stability regions are the open domain outside the closed curves in figures (3.4.1) and (3.4.2). Similarly, the stability region shrinks as $J$ increases and overall the stability regions corresponding to adaptive B2,

adaptive B2 and adaptive B3 are similar to that for B1. However, the stability regions for $M = 4$ and $M = 5$ are quite different, since the plots indicate that the stability region for $M = 5$ is smaller than that for $M = 4$ for a given $J$. From this point of view, $BEInDC$ schemes with sub-intervals $(1/4,\ 2/4,\ 3/4,\ 1)h$ that exclude the left-most quadrature point are better than those with sub-intervals $(0, 1/4,\ 2/4,\ 3/4,\ 1)h$ including the left-most quadrature point.

3. For $IMEXInDC$ schemes, the stability regions are the left parts of two pieces of the complex plane divided by the open curves in figure (3.4.5) and (3.4.6). The same as the previous observations, the stability region shrinks as $J$ increases and all the stability regions for B1, adaptive B2, adaptive B3, adaptive B4 are similar.

The accuracy regions with time-step $\triangle t = 0.2$ for various $InDC$ schemes are presented in figures (3.4.7)—(3.4.12). It can be observed that for given $M$ and $J$, the accuracy regions for a scheme ($FEInDC$, $BEInDC$ and $IMEXInDC$) with different approximation bases are almost the same. Moreover, for a scheme with given $M$ and a given basis, the accuracy region for a given tolerance $\epsilon$ increases as $J$ increases, which is consistent with that the order of accuracy of the scheme increases as $J$ increases.

So overall, the stability and accuracy regions for the schemes $FEInDC$, $BEInDC$ and $IMEXInDC$ with adaptive non-polynomial bases are comparable to those for the corresponding scheme with traditional polynomial basis B1, hence replacing B1

by an adaptive non-polynomial basis doesn't hurt the stability and accuracy properties of the traditional $InDC$ schemes.

## 3.5 Adaptive Step Size Control

As is well known, for problems with solution that have both smooth structures and sharp structures (e.g., the solution to the Van de Pol system), it is more efficient, or even exclusively the only choice in some cases, to implement an ODE solver with adaptive step sizes. To adaptively select the step size $h_n$, we need a proper error estimate that serves as a criterion to determine whether or not $h_n$ has to be refined, and if so, how it should be refined.

For the InDC scheme, recall that after $k$ iterations we obtain

$$\eta_m^{[k]}, \quad m = 0, 1, 2, \cdots, M, \tag{3.5.1}$$

the numerical solutions for $y(t)$ at the quadrature nodes $t_n = \tau_0 < \tau_1 < \cdots < \tau_M = t_{n+1}$ on the interval $[t_n, t_{n+1}]$. We propose the following error estimate

$$e_M = |y_{n+1}^* - y_{n+1}^{**}| \tag{3.5.2}$$

where

$$y_{n+1}^* = y_n + \int_{t_n}^{t_{n+1}} P_M^{B_i^M}(f(y,t))dt \tag{3.5.3}$$

and

$$y_{n+1}^{**} = y_n + \int_{t_n}^{t_{n+1}} P_{M-1}^{B_i^M}(f(y,t))dt. \tag{3.5.4}$$

Here $P_M^{B_i^M}(f(y,t))$ is the function in the space $B_i^M$ that interpolates $\{f(\eta_m^{[k]}, \tau_m)\}_{m=0}^{m=M}$ and $P_{M-1}^{B_i^M}(f(y,t))$ is the function in the space $B_i^M$ that interpolates $\{f(\eta_m^{[k]}, \tau_m)\}_{m=1}^{m=M}$. Note that the only difference between $P_M^{B_i^M}(f(y,t))$ and $P_{M-1}^{B_i^M}(f(y,t))$ is that the former includes the left-most point $\tau_0$ while the latter doesn't. When $B_i^M = B_1^M$ (the case of the polynomial InDC scheme), $P_M^{B_1^M}(f(y,t))$ and $P_M^{B_1^{M-1}}(f(y,t))$ are just interpolating polynomials.

With the error estimate $e_M$ and a given error tolerance $e_{tol}$, we adopt the step size prediction formula

$$h_n^{new} = h_n \cdot min(\ 6,\ max(\ 0.2,\ 0.9(\frac{e_{tol}}{e_M})^{\frac{1}{p}}\ )\ ), \tag{3.5.5}$$

with $p = 1$, as is suggested in [31] (Equation (7.28) on page 112), in chapter IV.8 of which a thorough study for step size selection can be found.

**Remark 3.5.1.** For our newly designed adaptive non-polynomial InDC scheme, it is expected that for a given $e_{tol}$, the step size $h_n^{new}$ should be larger than that for traditional polynomial InDC scheme, especially in the region where the initial or internal layers reside, thanks to the robustness of the adaptive basis element $e^{\lambda_n t}$ in capturing the sharp structures of the layers.

## 3.6 Numerical Tests

In this section, we test the InDC schemes with different adaptive non-polynomial bases on several ODE problems. The solution to each of these problems has initial or internal layers. For example, the solution to the scaler ODE problem

$$y' = -2\pi sin(2\pi t) - \frac{1}{\varepsilon}(y - cos(2\pi t)), \quad y(0) = 1 + \alpha \tag{3.6.1}$$

is

$$y(t) = \alpha e^{-t/\varepsilon} + cos(2\pi t). \tag{3.6.2}$$

$\alpha e^{-t/\varepsilon}$ is the part that depends on the stiffness parameter $\varepsilon$ and $cos(2\pi t)$ is the smooth part that is independent of $\varepsilon$. When $\varepsilon$ is very small, e.g. $\varepsilon = 1.0 \times 10^{-6}$, $\alpha e^{-t/\varepsilon}$ has a sharp shape, which is called initial layer because it exists mostly at the initial part of the whole solution.

One may notice that solution to problem (3.6.1) is independent of the stiffness parameter $\varepsilon$ when the initial data $y(0) = 1$, *i.e.*, $\alpha = 0$. In this case, the solution only has the smooth part, hence an implicit InDC scheme with regular polynomial basis can accurately solve the problem without the need to take extremely small time-step size. For general initial data, the adaptive step size selection technique is needed since for the initial layer, small time steps should be used to get accurate enough solution, while for the region after the initial layer, normal step size should be used due to the consideration of computational efficiency. This motivates us to combine the non-polynomial InDC scheme with polynomial InDC scheme: in the stiff region, we use

the non-polynomial basis while in the smooth region we use the regular polynomial basis. To determine whether or not an time interval $I_n$ is in smooth region, we compute the local stiffness parameter $\lambda_n$ based on the predicted solution obtained by the prediction step in the InDC scheme, and if $|\lambda_n/(0.5H_n)|$ is larger than a critical value $\lambda_c$, adaptive non-polynomial bases will be used to solve the problem over $I_n$, otherwise polynomial basis will be used. In the following tests, we use $InDC_M^J(B_1)$ to denote the regular polynomial InDC scheme, and $InDC_M^J(B_1, B_n),\ n = 2, 3, 4$ to denote the mixed InDC scheme that adopts $B_1$ in smooth regions and adopts $B_n$ (the non-polynomial basis) in stiff regions. To test if the proposed non-polynomial InDC schemes outperform traditional polynomial InDC schemes, we compare the number of time steps that are needed to obtain a given error tolerance when the schemes are applied to solve the ODE problems.

**Example 3.6.1.** (Cosine problem with an initial layer)

$$y' = -2\pi sin(2\pi t) - \frac{1}{\varepsilon}(y - cos(2\pi t)), \quad y(0) = 1 + \alpha. \tag{3.6.3}$$

We consider the case with $\alpha = 1.0$ and $\epsilon = 1.0 \times 10^{-6}$, in which the solution has a very-sharp initial layer. We solve the problem with schemes $BEInDC_4^1$, $BEInDC_4^2$, $BEInDC_5^1$, and $BEInDC_5^2$ with different bases. In our test $\lambda_c$ is set to be $10^4$. And in the adaptive step size selection procedure the error tolerance $e_{tol}$ is set to be $1.0 \times 10^{-5}$.

The results are shown in figures 3.6.1—3.6.4, in which NOS is the abbreviation for *number of steps*. Take the scheme $BEInDC_4^2$ as an example, we can observe the

following properties.

1. For final time $T = \epsilon,\ 10\epsilon,\ 10^5\epsilon$, $BEInDC_4^2$ with mixed bases costs fewer time steps than $BEInDC_4^2$ with $B_1$ (polynomial basis), given the same error tolerance measured in the $L_1$ norm. For example, at time $T = \epsilon$, $BEInDC_4^2(B_1)$ costs 9 steps while $BEInDC_4^2(B_1, B_4)$ costs 7 steps.

2. If the final time $T$ is in the stiff region, i.e., $T = \epsilon$, then the non-polynomial basis takes all the time steps. For example, for $BEInDC_4^2(B_1, B_4)$, $B_4$ is used in all the 7 steps, and $B_1$ is not used.

3. If the final time $T = 10\epsilon$, then both the non-polynomial basis and polynomial basis are used for some time steps. For example, for $BEInDC_4^2(B_1, B_4)$, $B_4$ costs 12 steps and $B_1$ costs 4 steps.

4. If the final time $T = 10^5\epsilon$, then the NOS for $B_4$ is still the same as that for time $T = 10\epsilon$. This can be explained by that after time $T = 10\epsilon$, the solution is in the smooth region, hence the problem is solely solved by $B_1$ and $B_4$ is not used anymore.

5. Overall, the performance of $BEInDC_4^2(B_1, B_4)$ is the best.

Similar properties for $InDC_4^1$, $InDC_5^1$ and $InDC_5^2$ can be observed. And we found that schemes with $M = 5$ take more time steps than schemes with $M = 4$ for $T = 100000\epsilon$. So overall, $BEInDC_4^J(B_1, B_4)$ is a better choice for improving computational efficiency when one solves stiff problems with sharp initial layers.

**Example 3.6.2.** (Van der Pol system)

Consider the Van der Pol system

$$\begin{cases} y_1' = y_2, \\ \varepsilon y_2' = (1 - y_1^2)y_2 - y_1, \end{cases} \tag{3.6.4}$$

with $\epsilon = 0.01$. The initial condition is

$$\begin{cases} y_1(0) = 2, \\ y_2(0) = -0.66. \end{cases} \tag{3.6.5}$$

We solve the problem with scheme $InDC_4^3$ with different bases. And in the adaptive step size selection procedure $e_{tol}$ is set to be $1.0 \times 10^{-2}$.

The results are shown in figures 3.6.5 and 3.6.6. Both the solutions to $y_1$ and $y_2$ have internal layers. It can be observed that $InDC_4^3(B_1, B_4)$ takes 12 fewer steps than $InDC_4^3(B_1)$ if the final time is $T = 1$ for which the solution has one internal layer, and takes 29 fewer steps if the final time is $T = 2$ for which the solution has two internal layers. So $InDC_4^3(B_1, B_4)$ improves the computational efficiency by saving time steps, and the more layers the solution has, the more time steps it saves. However, $InDC_4^3(B_1, B_2)$ and $InDC_4^3(B_1, B_3)$ do not have such advantage.

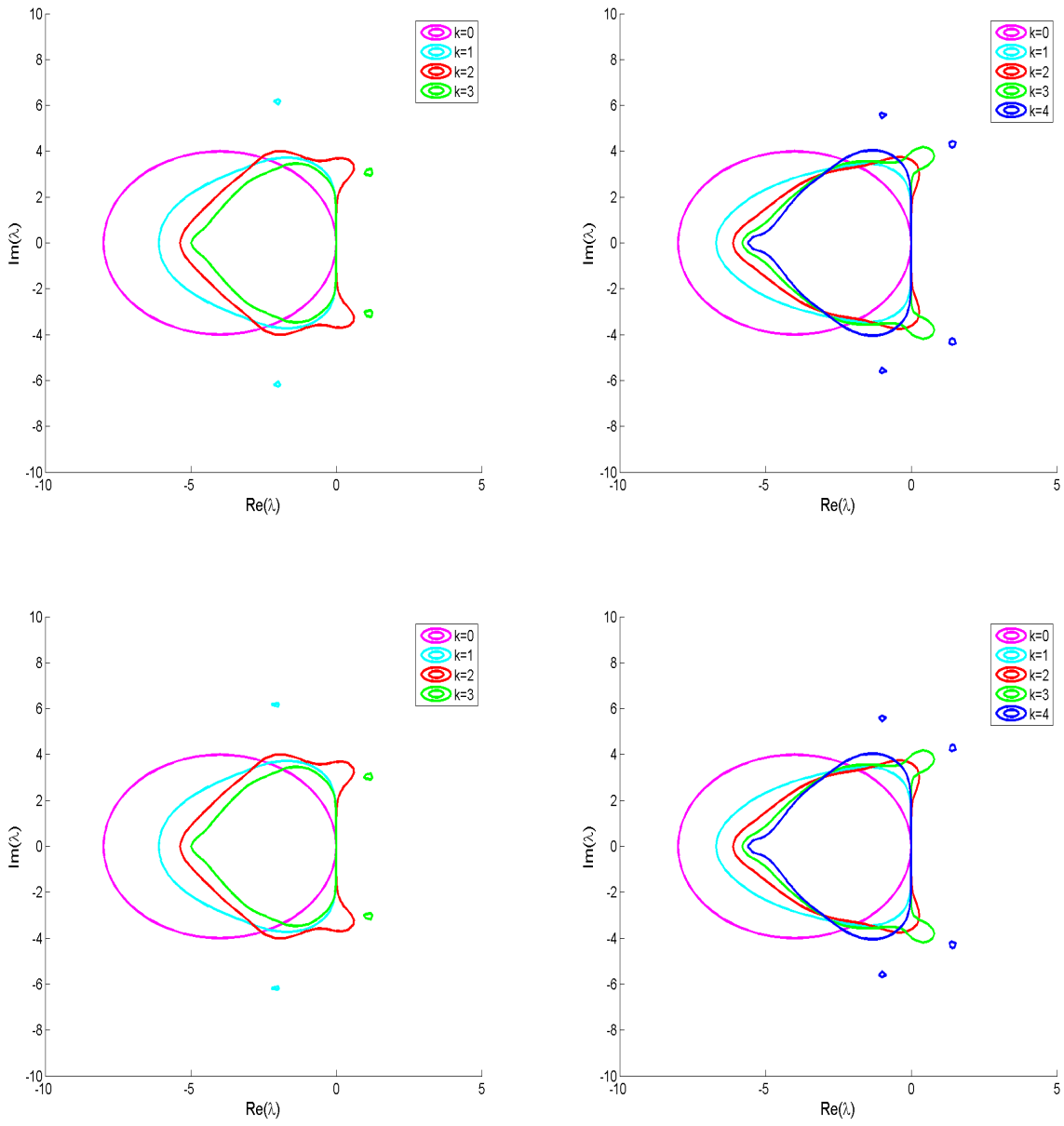


Figure 3.4.1: Stability regions for forward Euler (FE) InDC schemes. Top-left: $FEInDC_4$ with B1; Top-right: $FEInDC_5$ with B1; Bottom-left: $FEInDC_4$ with adaptive B2; Bottom-right: $FEInDC_5$ with adaptive B2. $k$ is the number of correction steps.

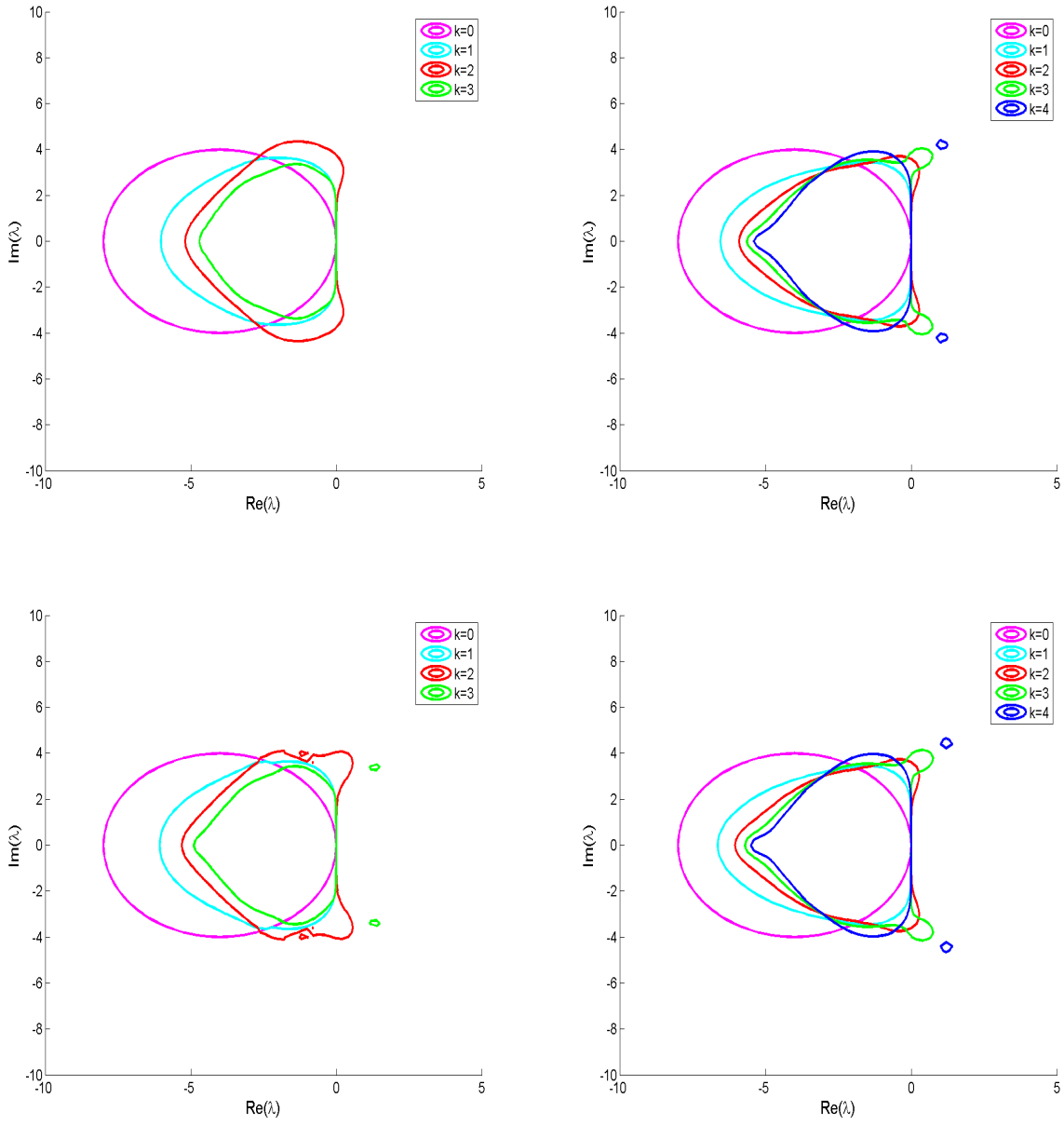


Figure 3.4.2: Stability regions for forward Euler (FE) InDC schemes. Top-left: $FEInDC_4$ with adaptive B3; Top-right: $FEInDC_5$ with adaptive B3; Bottom-left: $FEInDC_4$ with adaptive adaptive B4; Bottom-right: $FEInDC_5$ with adaptive adaptive B4. $k$ is the number of correction steps.

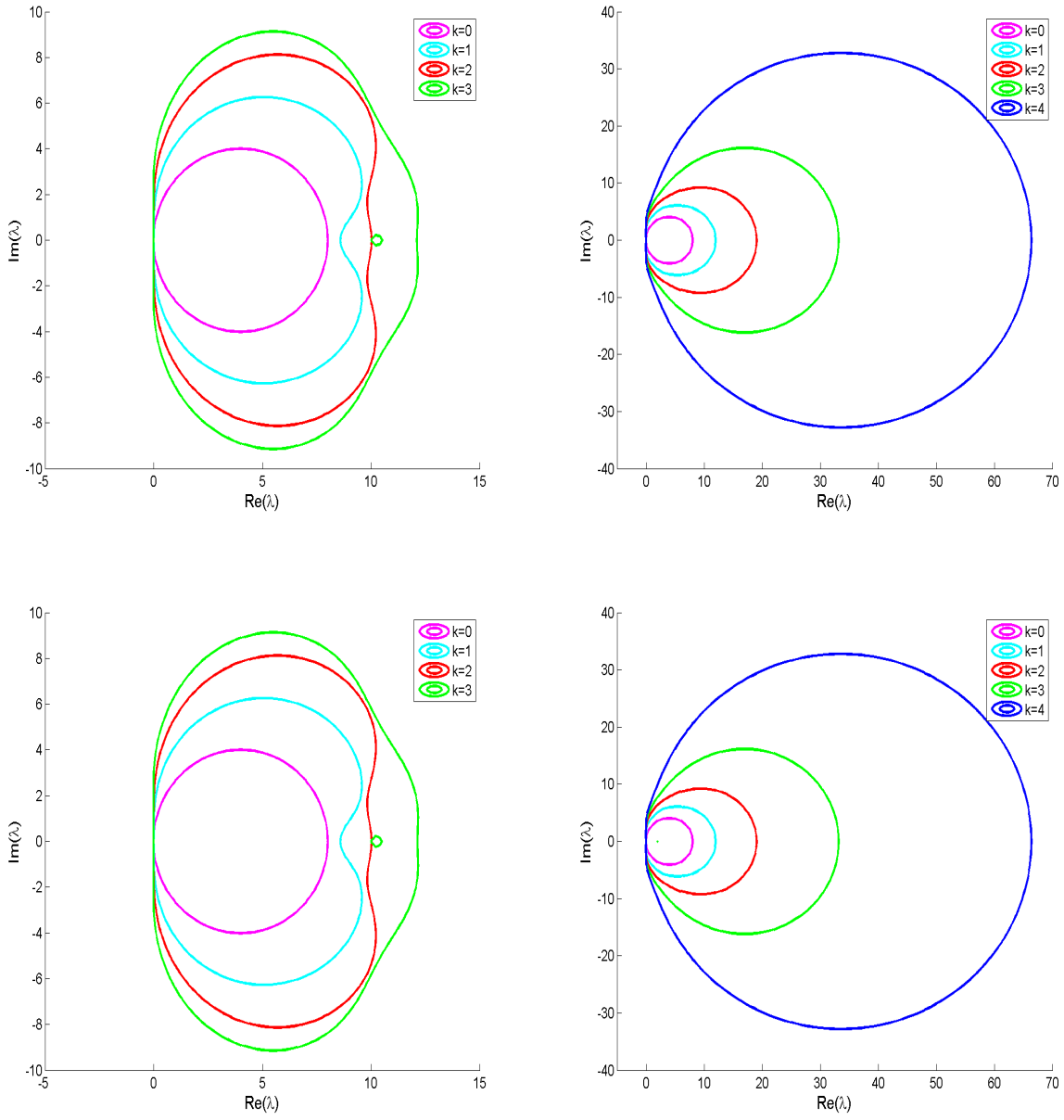


Figure 3.4.3: Stability regions for backward Euler (BE) InDC schemes. Top-left: $BEInDC_4$ with B1; Top-right: $BEInDC_5$ with B1; Bottom-left: $BEInDC_4$ with adaptive adaptive B2; Bottom-right: $BEInDC_5$ with adaptive adaptive B2. $k$ is the number of correction steps.

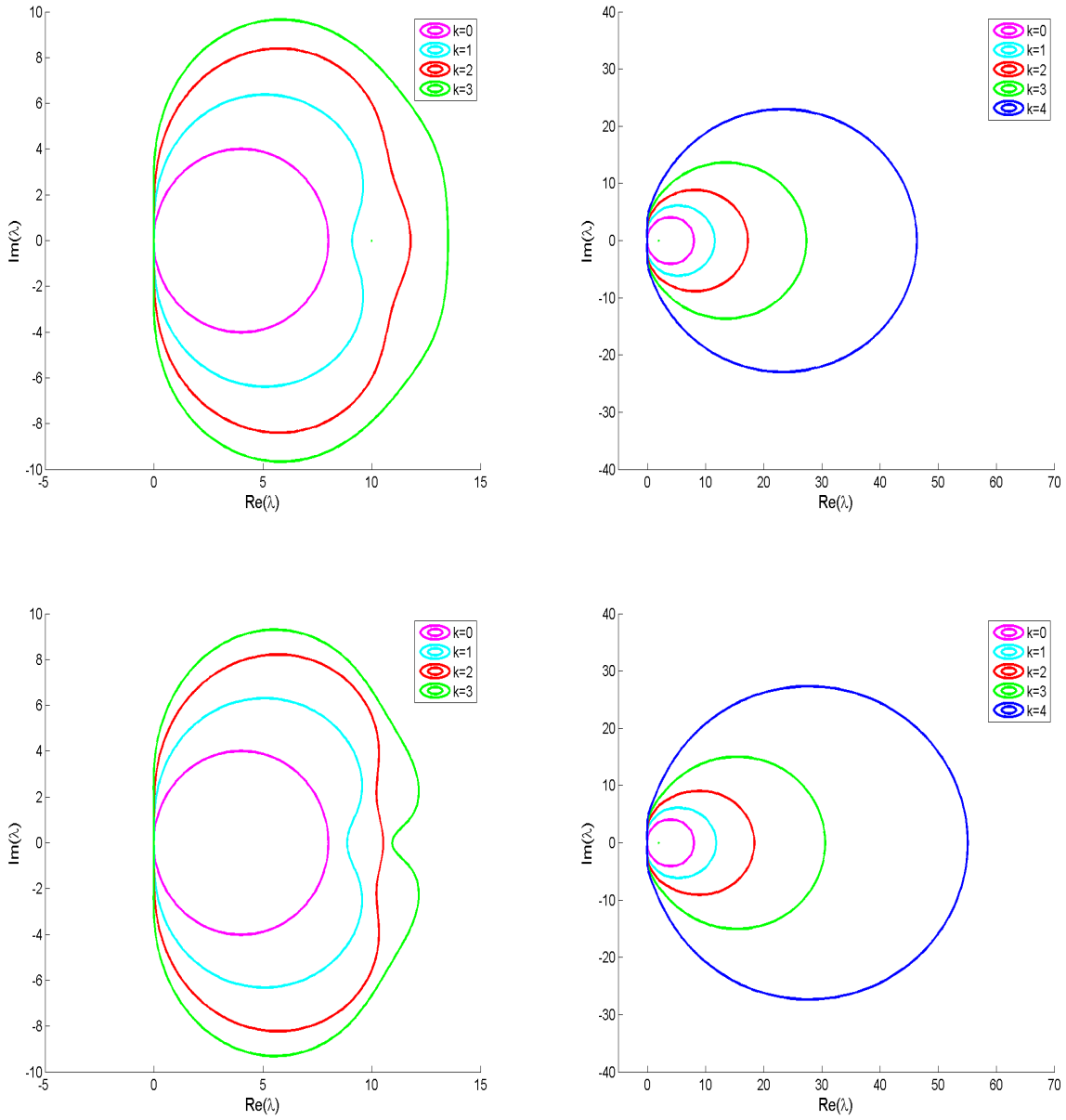


Figure 3.4.4: Stability regions for backward Euler (BE) InDC schemes. Top-left: $BEInDC_4$ with adaptive B3; Top-right: $BEInDC_5$ with adaptive B3; Bottom-left: $BEInDC_4$ with adaptive adaptive B4; Bottom-right: $BEInDC_5$ with adaptive adaptive B4. $k$ is the number of correction steps.

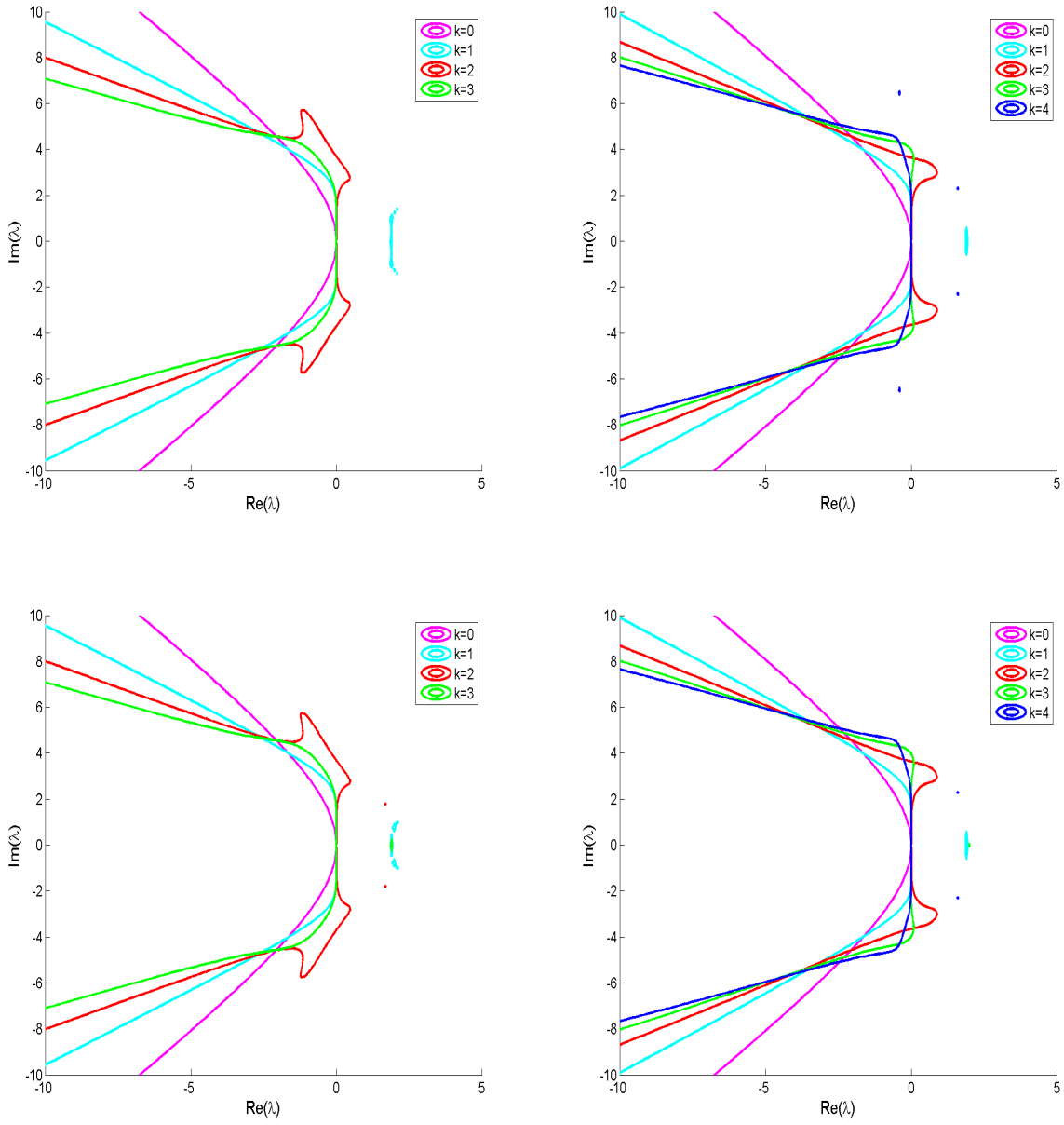


Figure 3.4.5: Stability regions for implicit-explicit (IMEX) InDC schemes. Top-left: $IMEXInDC_4$ with B1; Top-right: $IMEXInDC_5$ with B1; Bottom-left: $IMEXInDC_4$ with adaptive adaptive B2; Bottom-right: $IMEXInDC_5$ with adaptive adaptive B2. $k$ is the number of correction steps.

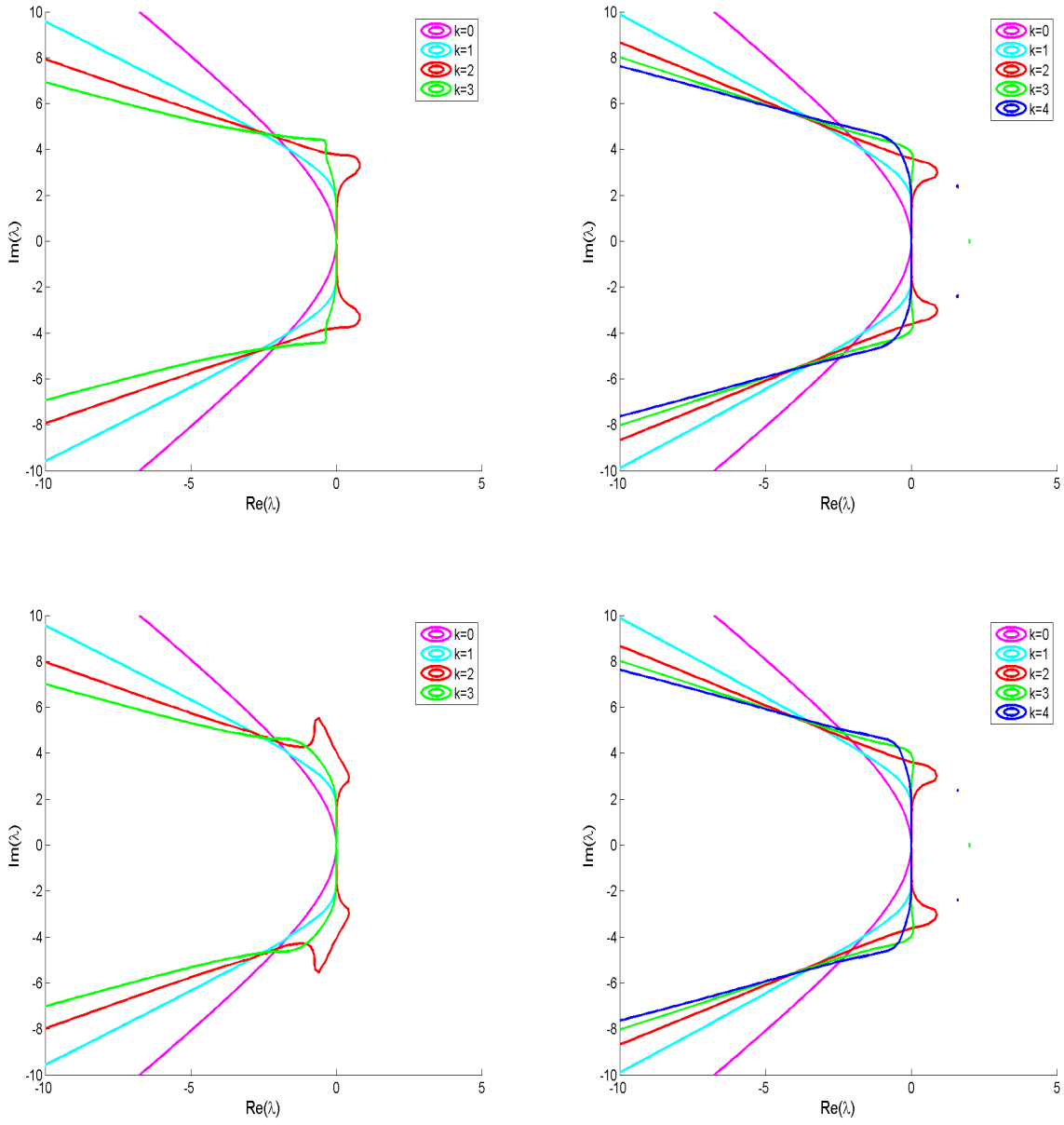


Figure 3.4.6: Stability regions for implicit-explicit (IMEX) InDC schemes. Top-left: $IMEXInDC_4$ with adaptive B3; Top-right: $IMEXInDC_5$ with adaptive B3; Bottom-left: $IMEXInDC_4$ with adaptive adaptive B4; Bottom-right: $IMEXInDC_5$ with adaptive adaptive B4. $k$ is the number of correction steps.

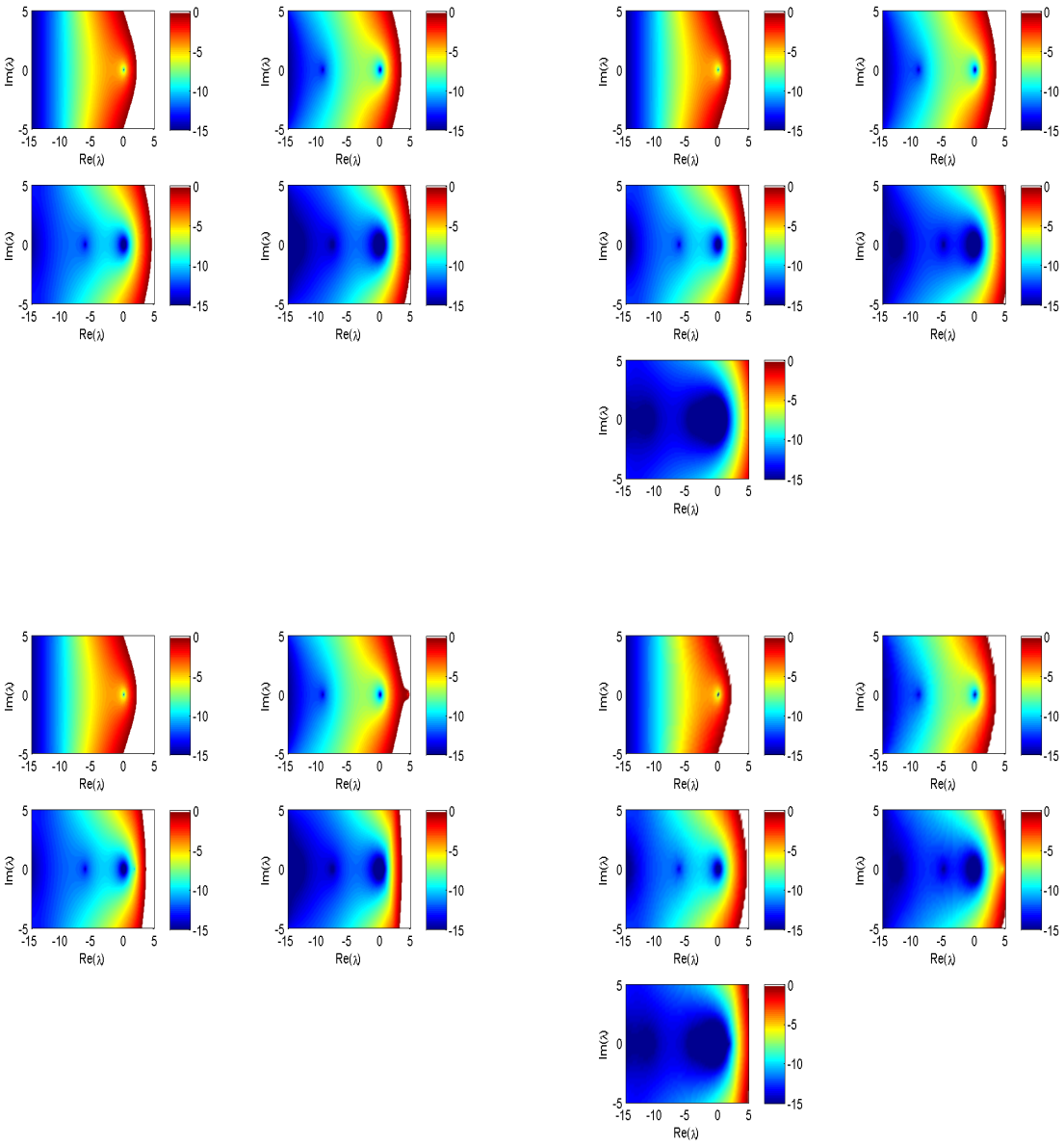


Figure 3.4.7: (Accuracy regions) Top-left: $FEInDC_4$ with B1; Top-right: $FEInDC_5$ with B1; Bottom-left: $FEInDC_4$ with adaptive B2; Bottom-right: $FEInDC_5$ with adaptive B2. For each of the figures for $InDC_4$, the top-left, top-right, bottom-left, bottom-right sub-figures are for $InDC_4^0$, $InDC_4^1$, $InDC_4^2$, $InDC_4^3$ respectively and for each of the figures for $InDC_5$, the top-left, top-right, middle-left, middle-right, bottom-left sub-figures are for $InDC_5^0$, $InDC_5^1$, $InDC_5^2$, $InDC_5^3$, $InDC_5^4$ respectively.

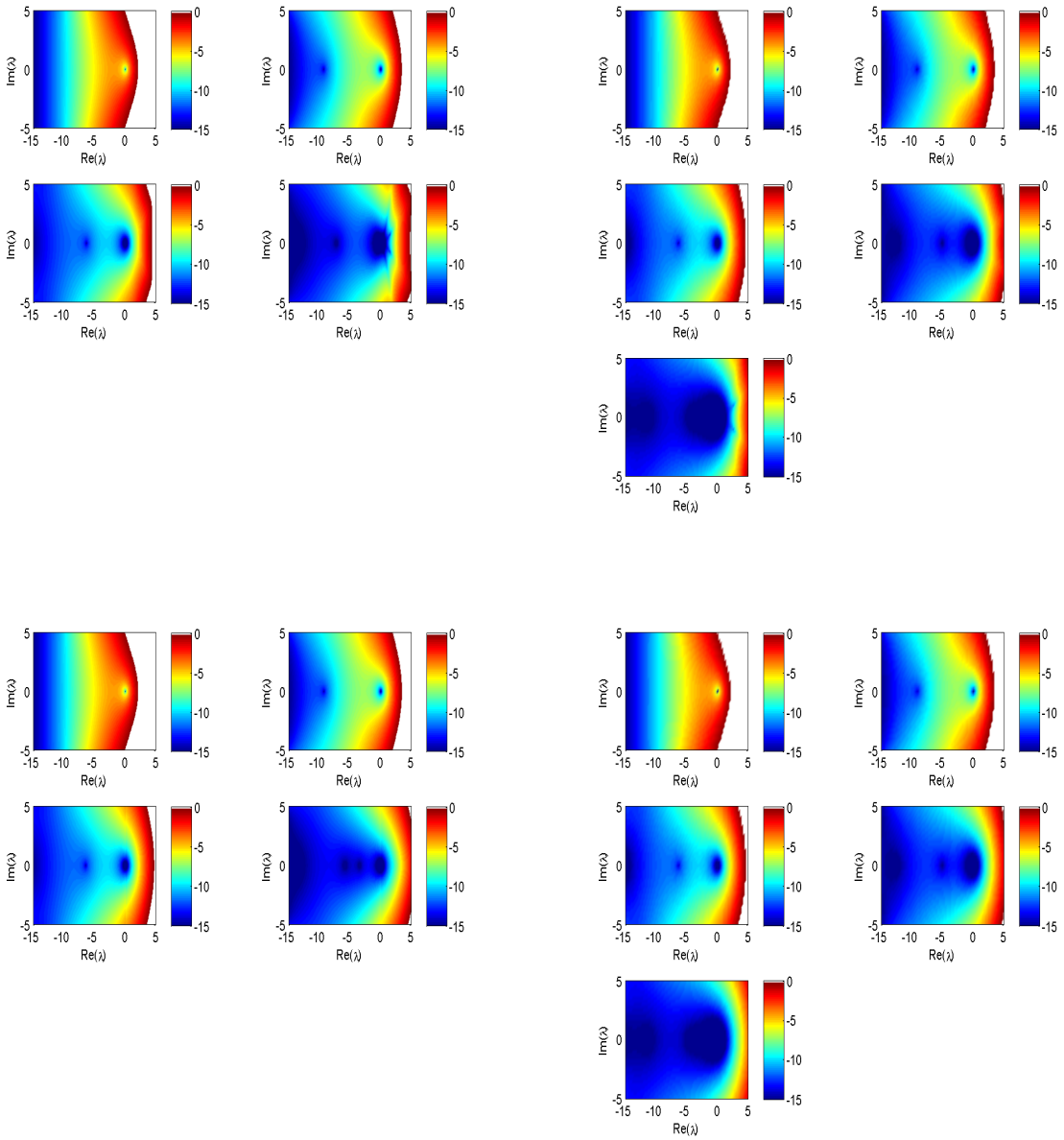


Figure 3.4.8: (Accuracy regions) Top-left: $FEInDC_4$ with adaptive B3; Top-right: $FEInDC_5$ with adaptive B3; Bottom-left: $FEInDC_4$ with adaptive B4; Bottom-right: $FEInDC_5$ with adaptive B4.

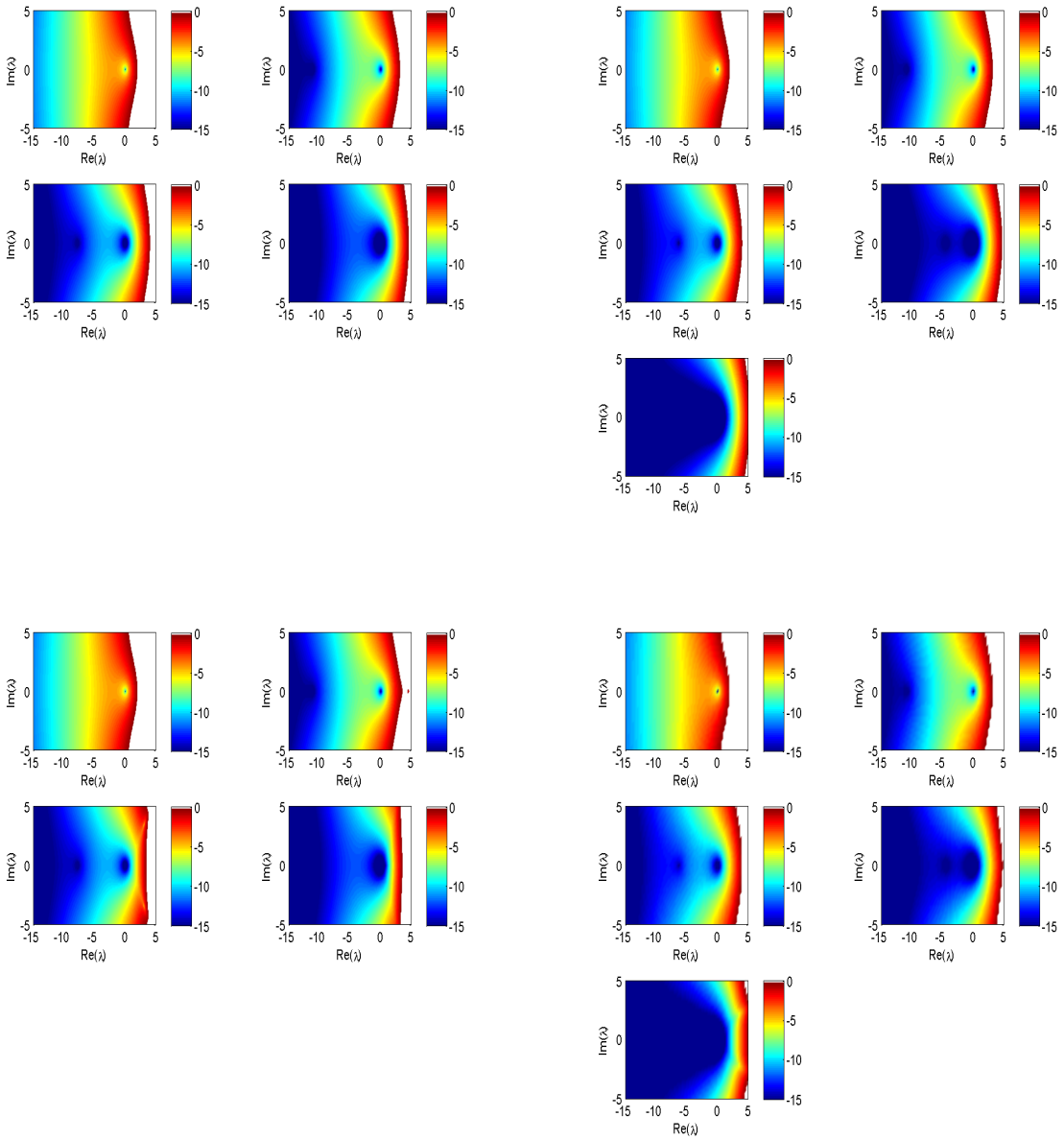


Figure 3.4.9: (Accuracy regions) Top-left: $BEInDC_4$ with B1; Top-right: $BEInDC_5$ with B1; Bottom-left: $BEInDC_4$ with adaptive B2; Bottom-right: $BEInDC_5$ with adaptive B2.

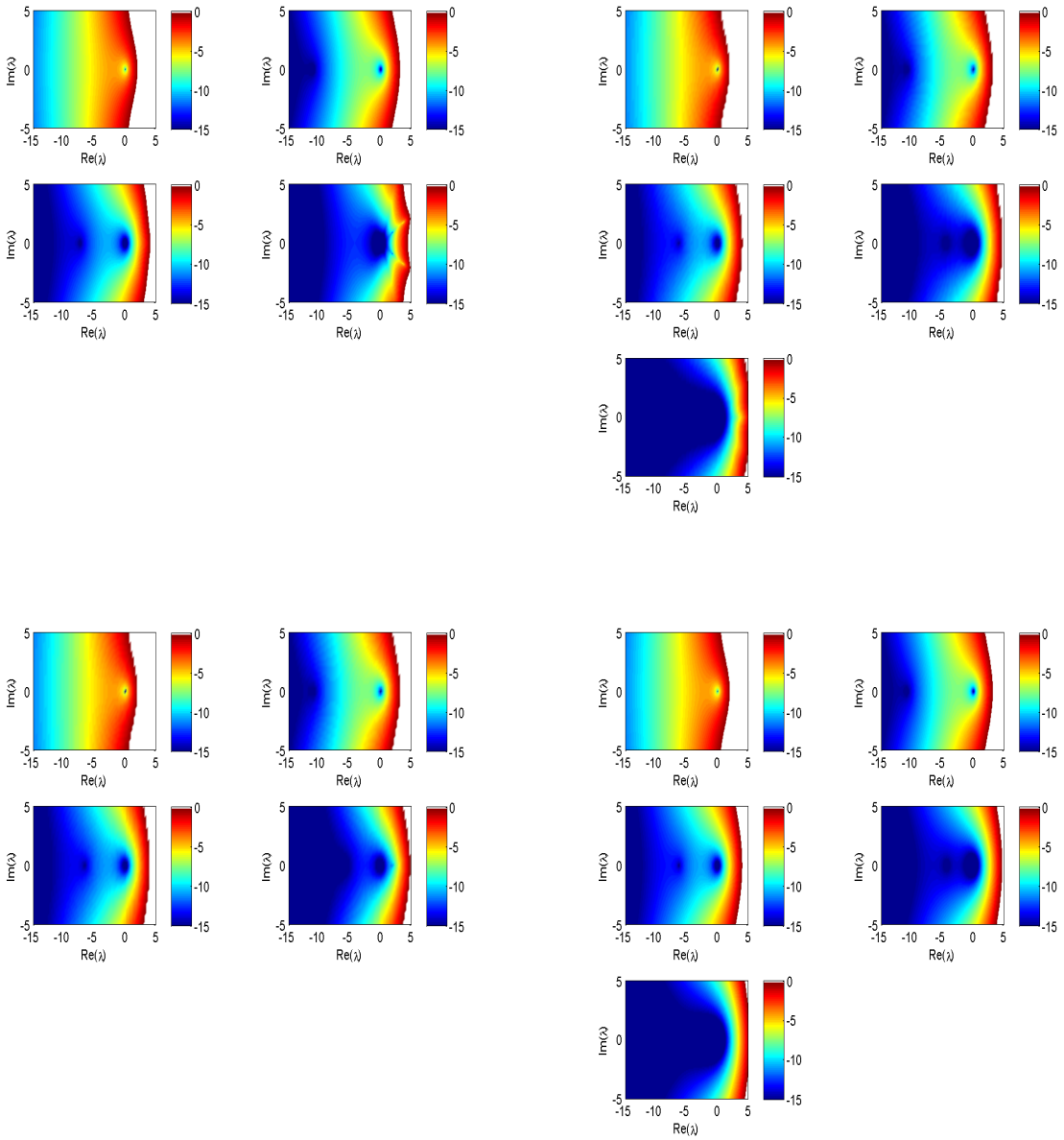


Figure 3.4.10: (Accuracy regions) Top-left: $BEInDC_4$ with adaptive B3; Top-right: $BEInDC_5$ with adaptive B3; Bottom-left: $BEInDC_4$ with adaptive B4; Bottom-right: $BEInDC_5$ with adaptive B4.

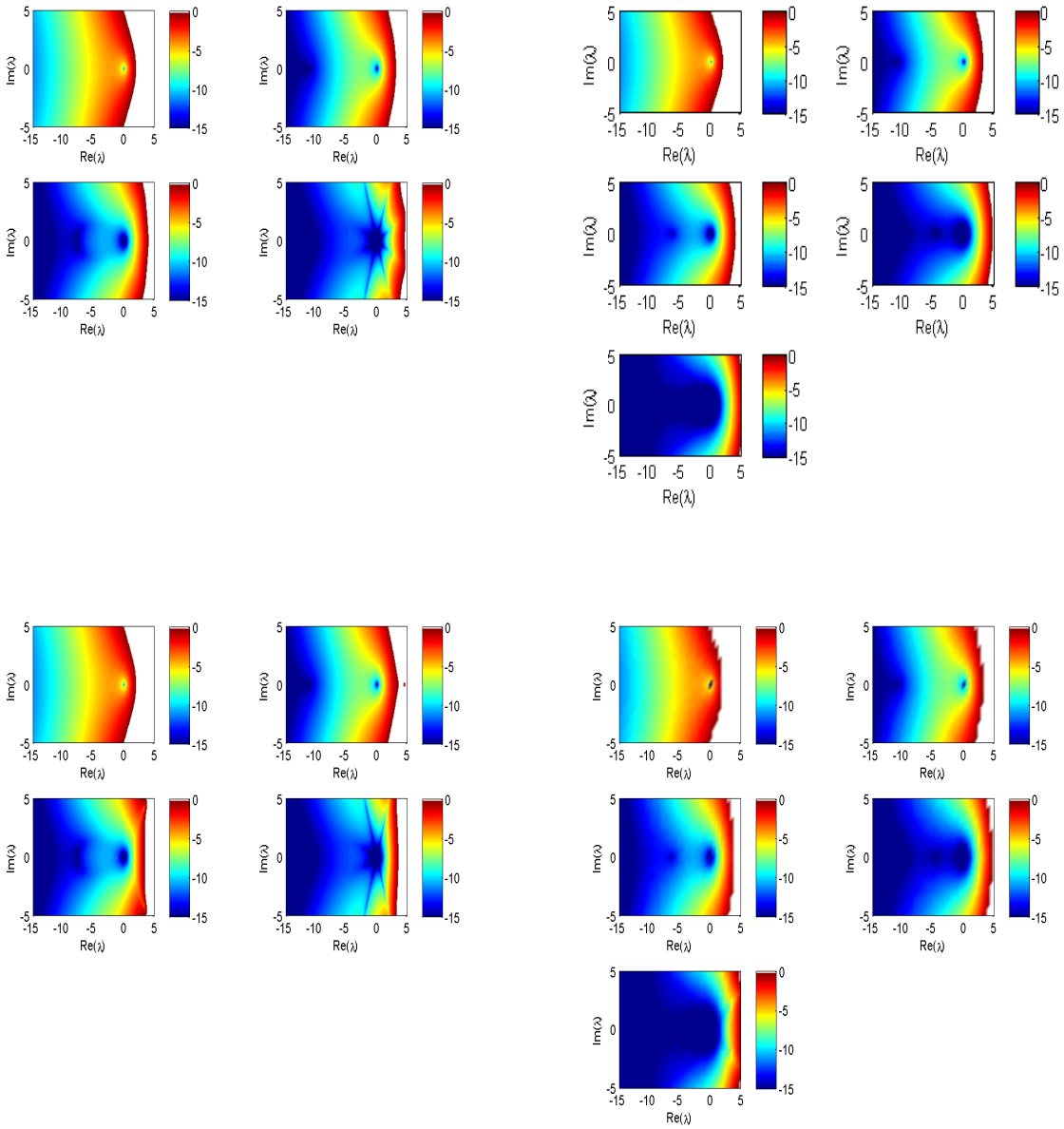


Figure 3.4.11: (Accuracy regions) Top-left: $IMEXInDC_4$ with B1; Top-right: $IMEXInDC_5$ with B1; Bottom-left: $IMEXInDC_4$ with adaptive B2; Bottom-right: $IMEXInDC_5$ with adaptive B2.

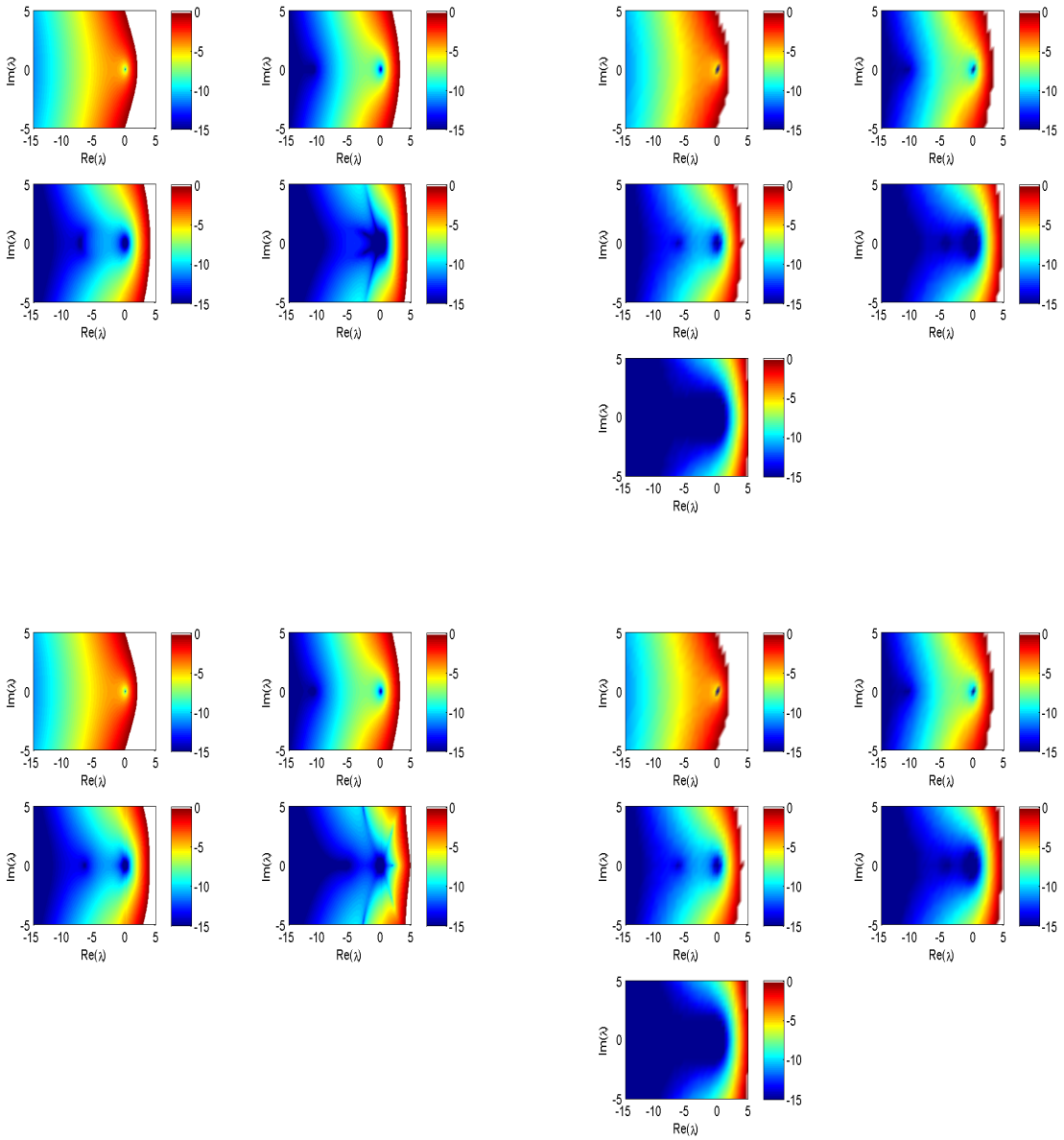


Figure 3.4.12: (Accuracy regions) Top-left: $IMEXInDC_4$ with adaptive B3; Top-right: $IMEXInDC_5$ with adaptive B3; Bottom-left: $IMEXInDC_4$ with adaptive B4; Bottom-right: $IMEXInDC_5$ with adaptive B4.

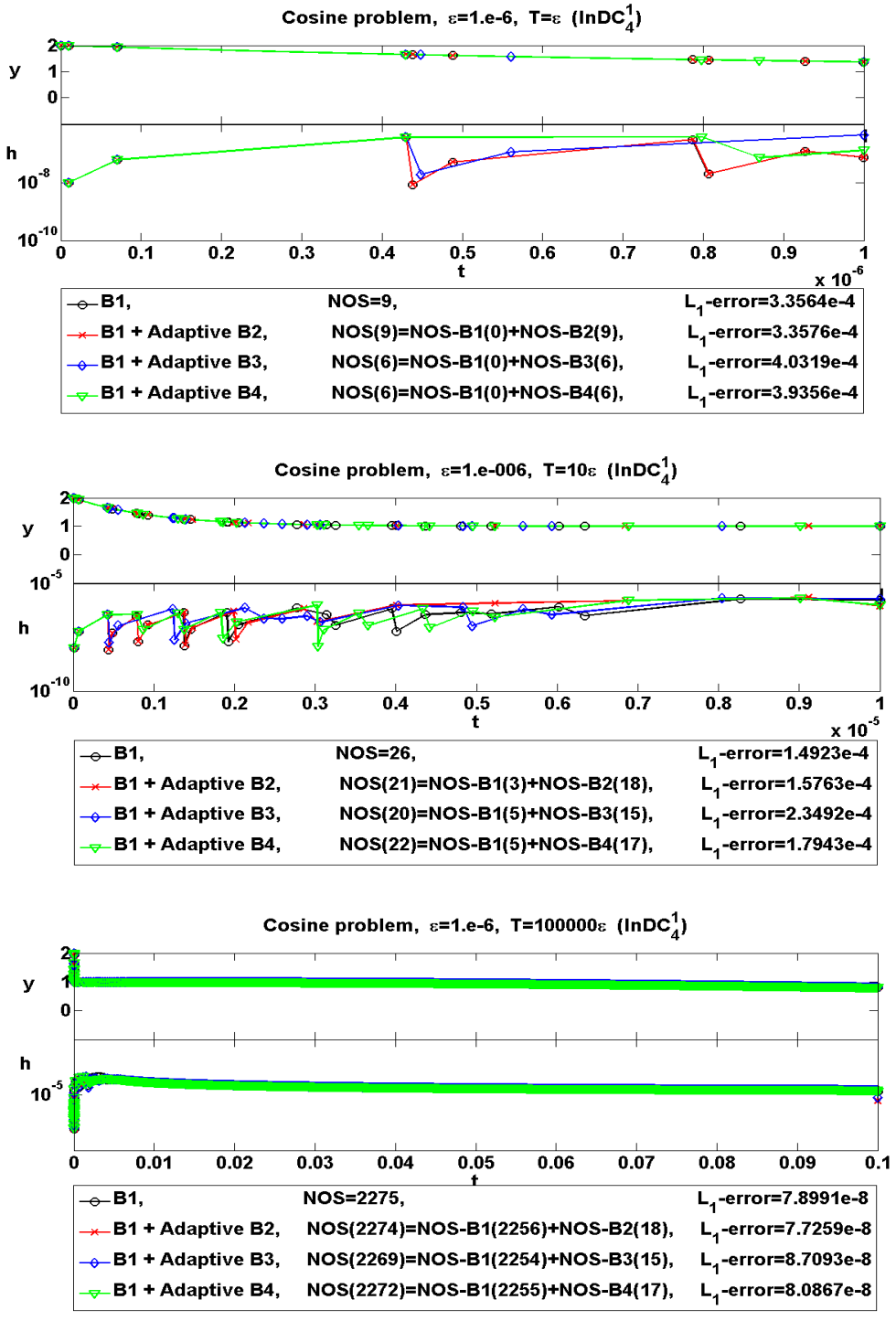


Figure 3.6.1: (Example 3.6.1) NOS (*number of steps*) for $InDC_4^1$ schemes with different bases. Top: $T = \epsilon$; Middle: $T = 10\epsilon$; Bottom: $T = 10^5\epsilon$.

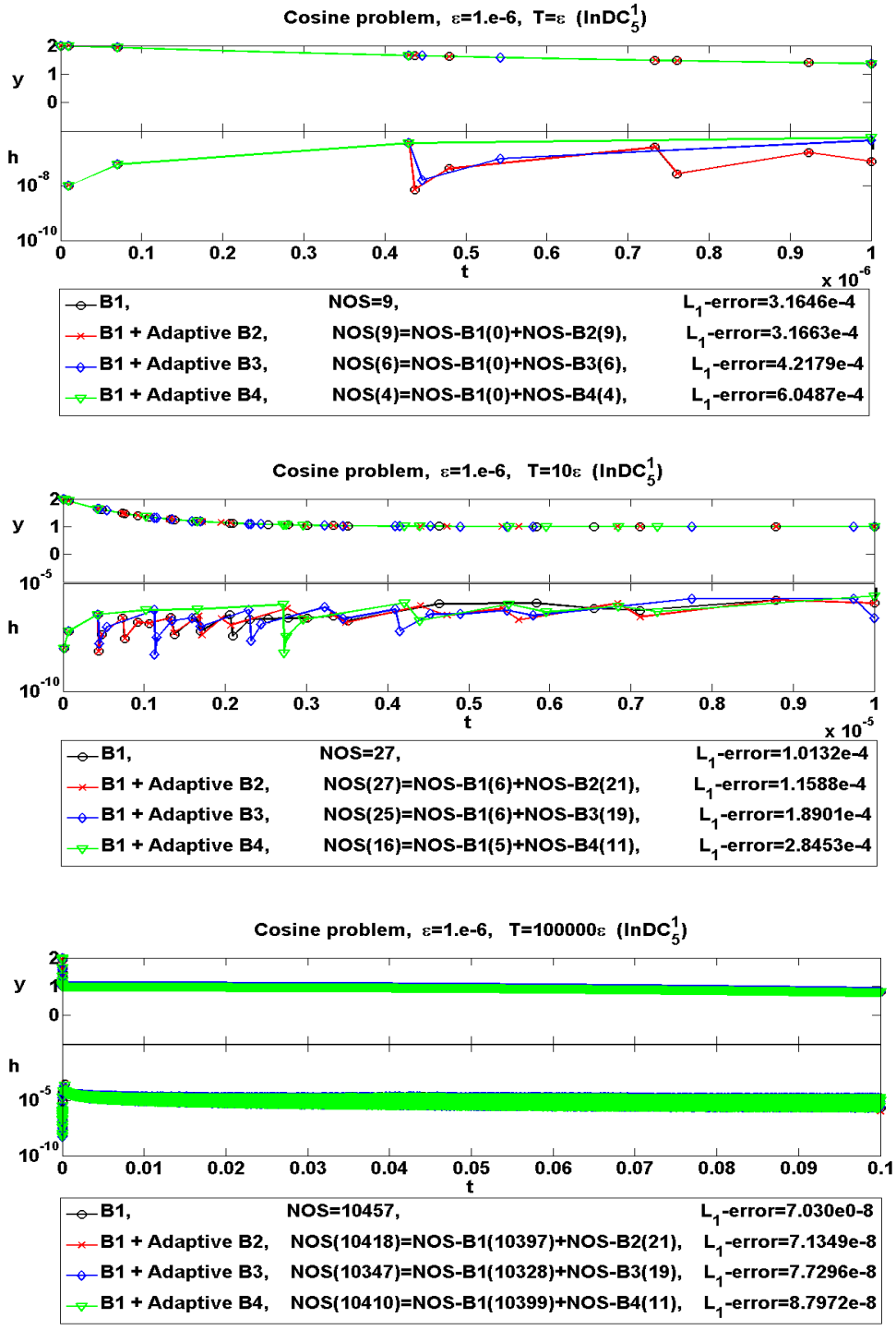


Figure 3.6.2: (Example 3.6.1) NOS for $InDC_5^1$ schemes with different bases. Top: $T = \epsilon$; Middle: $T = 10\epsilon$; Bottom: $T = 10^5\epsilon$.

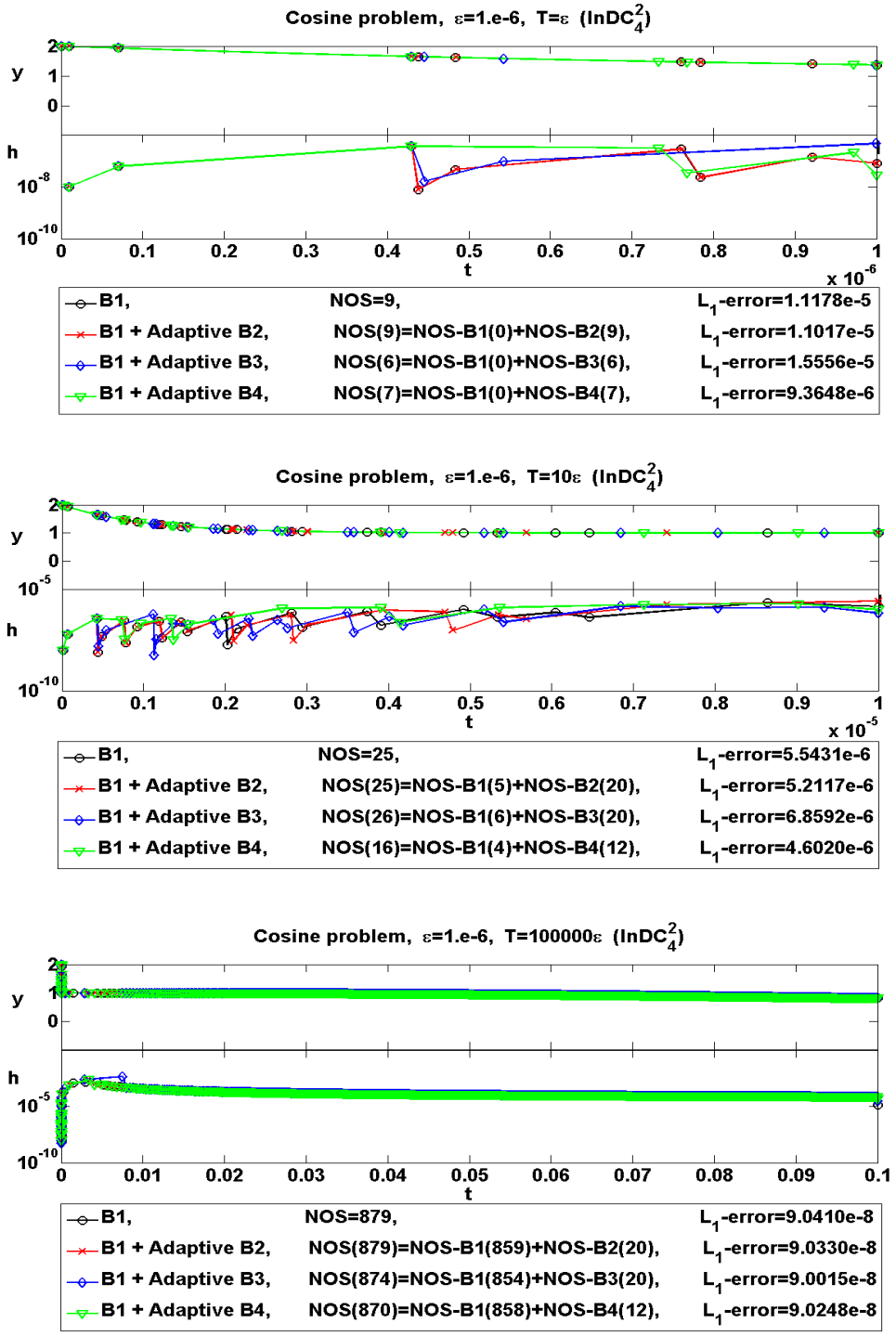


Figure 3.6.3: (Example 3.6.1) NOS for $InDC_4^2$ schemes with different bases. Top: $T = \epsilon$; Middle: $T = 10\epsilon$; Bottom: $T = 10^5\epsilon$.

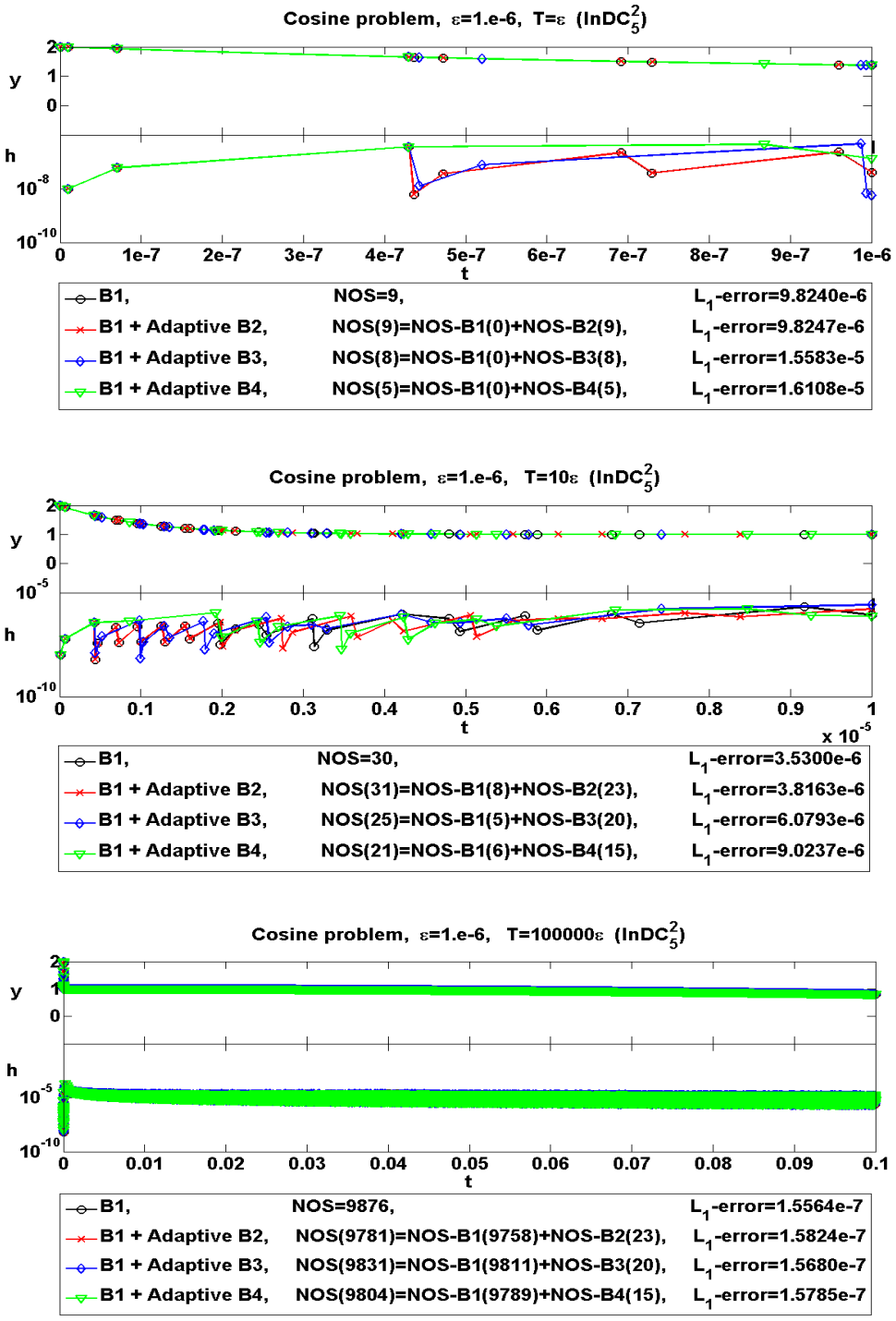


Figure 3.6.4: (Example 3.6.1) NOS for $InDC_5^2$ schemes with different bases. Top: $T = \epsilon$; Middle: $T = 10\epsilon$; Bottom: $T = 10^5\epsilon$.

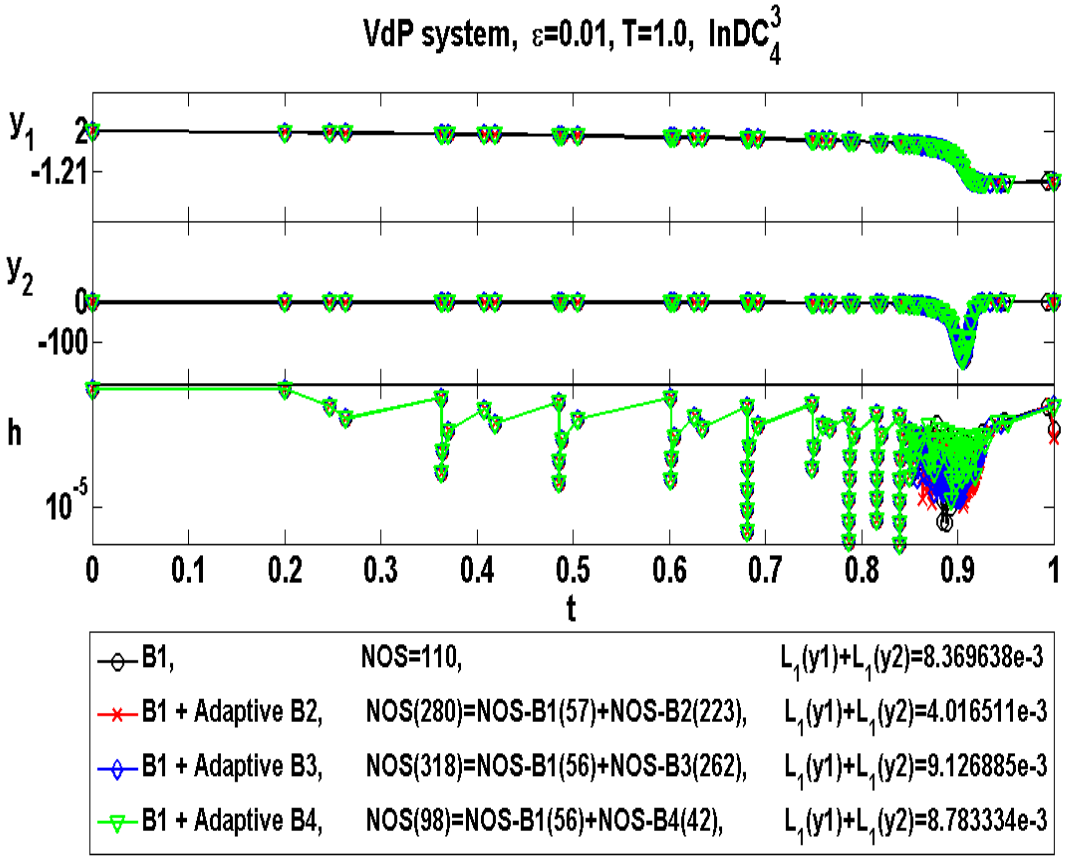


Figure 3.6.5: (Example 3.6.2) NOS for $InDC_4^3$ schemes with different bases. (T=1)

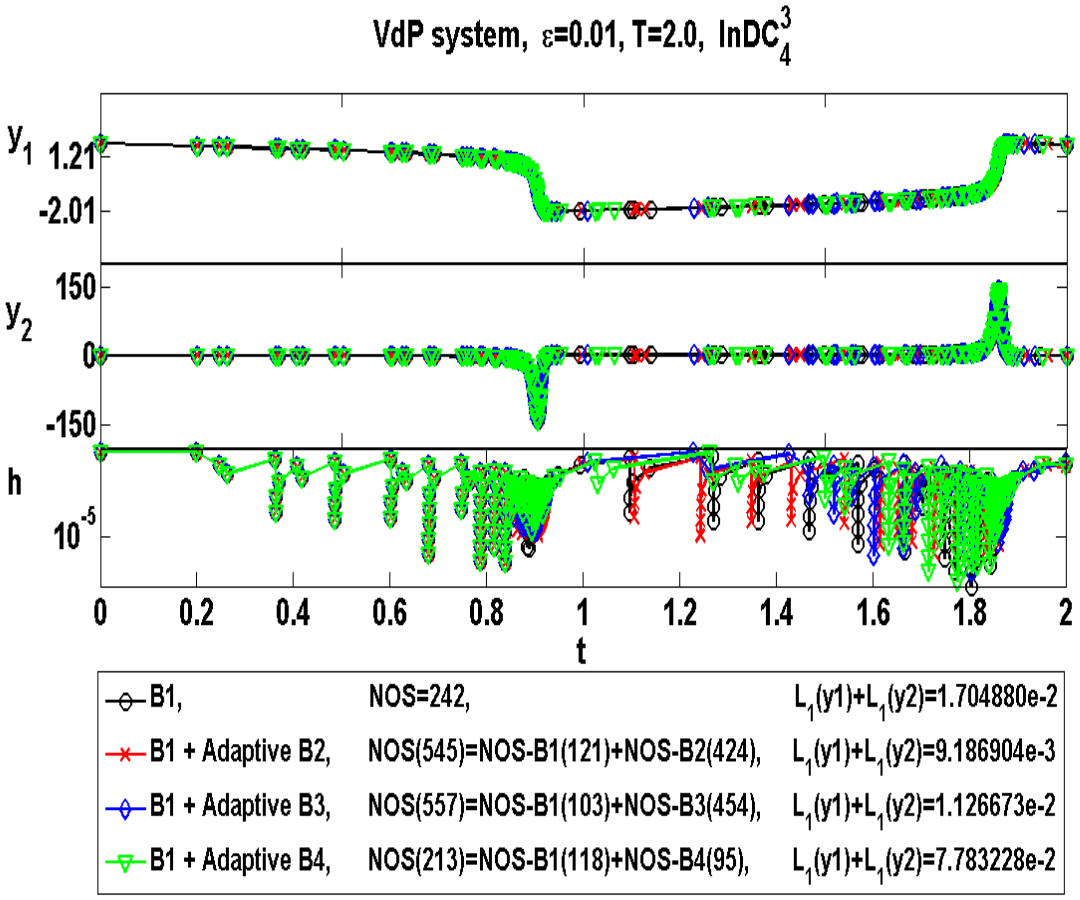


Figure 3.6.6: (Example 3.6.2) NOS for $InDC_4^3$ schemes with different bases. (T=2)

# CHAPTER 4

# Conclusions

The conclusions from the work in this dissertation are summarized as follows.

In the first part, the locally-parametrized flux limiters were successfully incorporated to modify the high-order numerical flux in the original finite volume (FV) weighted essentially non-oscillatory (WENO) Runge Kutta (RK) scheme toward a lower-order monotone flux that is maximum principle preserving (MPP), so that the new scheme satisfies the maximum principle while at the same time maintains the high-order accuracy. For linear advection problems, if the convection term $f(u)$ satisfies $f'(u) > 0$ for all $u$ or $f'(u) < 0$ for all $u$, it was proved that the high-order accuracy is maintained if the lower order flux is of Godnov type. And for a general

setting, by Taylor-expansion around extrema it was proved that the FV RK WENO scheme preserves up to third-order accuracy without extra Courant-Friedrichs-Lewy (CFL) restriction. Extensive numerical tests were performed to verify the MPP property of numerical solutions as well as preservation of high-order accuracy.

In the second part, the traditional integral deferred correction (InDC) scheme based on polynomial basis was modified by replacing the polynomial basis with several non-polynomial bases that contain exponential functions. Specifically, three new bases $B2$, $B3$ and $B4$ were proposed as replacements for the polynomial basis in the original scheme. Numerical investigations in the dissertation showed that all these three new bases approximate functions with sharp layers slightly better than polynomial, and among them $B4$ performs the best. We investigate the stability and accuracy properties of the non-polynomial InDC scheme coupled with backward Euler (BE), forward Euler (FE), and implicit-explicit (IMEX) schemes. Specifically, the stability and accuracy regions of the InDC method with adaptive non-polynomial basis are comparable to those of the traditional polynomial-based InDC scheme. The newly proposed InDC scheme is applied to various stiff problems and it is observed that in the presence of layers, the scheme with adaptive non-polynomial basis B4 uses less computational time steps than the scheme with regular polynomial basis, given the same error tolerance. This fact indicates better computational efficiency of the new scheme.

# Bibliography


[1] B. Cockburn, C. Johnson, C.-W. Shu, and E. Tadmor, *Advanced numerical approximation of nonlinear hyperbolic equations*, Springer New York, 1998.

[2] I. Faragó and R. Horváth, *Discrete maximum principle and adequate discretizations of linear parabolic problems*, SIAM Journal on Scientific Computing, 28 (2006), pp. 2313–2336.

[3] I. Faragó, R. Horváth, and S. Korotov, *Discrete maximum principle for linear parabolic problems solved on hybrid meshes*, Applied Numerical Mathematics, 53 (2005), pp. 249–264.

[4] I. Faragó and J. Karátson, *Discrete maximum principle for nonlinear parabolic PDE systems*, IMA Journal of Numerical Analysis, (2012).

[5] H. Fujii, *Some remarks on finite element analysis of time-dependent field problems*, Theory and Practice in Finite Element Structural Analysis, (1973), pp. 91–106.

[6] S. Gottlieb, D. Ketcheson, and C.-W. Shu, *High order strong stability preserving time discretizations*, Journal of Scientific Computing, 38 (2009), pp. 251–289.

[7] A. Harten, *High resolution schemes for hyperbolic conservation laws*, Journal of Computational Physics, 49 (1983), pp. 357–393.

[8] X. Y. Hu, N. A. Adams, and C.-W. Shu, *Positivity-preserving method for high-order conservative schemes solving compressible Euler equations*, Journal of Computational Physics, 242 (2013), pp. 169–180.

[9] Y. Jiang and Z. Xu, *Parametrized maximum principle preserving limiter for finite difference WENO schemes solving convection-dominated diffusion equations*, SIAM Journal on Scientific Computing, 35 (2013), pp. A2524–A2553.

[10] C. Liang and Z. Xu, *Parametrized maximum principle preserving flux limiters for high order schemes solving multi-dimensional scalar hyperbolic conservation laws*, Journal of Scientific Computing, 58 (2014), pp. 41–60.

[11] B. Perthame and C.-W. Shu, *On positivity preserving finite volume schemes for Euler equations*, Numerische Mathematik, 73 (1996), pp. 119–130.

[12] T. Xiong, J.-M. Qiu, and Z. Xu, *A parametrized maximum principle preserving flux limiter for finite difference RK-WENO schemes with applications in incompressible flows* , Journal of Computational Physics, 252 (2013), pp. 310–331.

[13] T. Xiong, J.-M. Qiu, and Z. Xu, *Parametrized positivity preserving flux limiters for the high order finite difference WENO scheme solving compressible Euler equations*.

[14] T. Xiong, J.-M. Qiu, and Z. Xu, *Semi-Lagrangian finite difference WENO scheme with parametrized MPP flux limiters for the Vlasov equation*.

[15] Z. Xu, *Parametrized maximum principle preserving flux limiters for high order scheme solving hyperbolic conservation laws: one-dimensional scalar problem*, Mathematics of Computation.

[16] X. Zhang, Y. Liu, and C.-W. Shu, *Maximum-principle-satisfying high order finite volume weighted essentially nonoscillatory schemes for convection-diffusion equations*, SIAM Journal on Scientific Computing, 34 (2012), pp. A627–A658.

[17] X. Zhang and C.-W. Shu, *On maximum-principle-satisfying high order schemes for scalar conservation laws*, Journal of Computational Physics, 229 (2010), pp. 3091–3120.

[18] Xiangxiong Zhang and Chi-Wang Shu, *On positivity-preserving high order discontinuous Galerkin schemes for compressible Euler equations on rectangular meshes*, Journal of Computational Physics, 229 (2010), pp. 8918–8934.

[19] Y. Zhang, X. Zhang, and C.-W. Shu, *Maximum-principle-satisfying second order discontinuous Galerkin schemes for convection-diffusion equations on triangular meshes*, Journal of Computational Physics, 234 (2012), pp. 295–316.

[20] Jan S. Hesthaven, Sigal Gottlieb, and David Gottlieb, *Spectral Methods for Time-Dependent Problems*, Cambridge University Press, 2007.

[21] Sigal Gottlieb and Chi-Wang Shu, *Total Variation Diminishing Runge Kutta Schemes*, Mathematics of Computation, Volume 67, Number 221, January 1998, Pages 73-85.

[22] Pei Yang, Tao Xiong, Jingmei Qiu and Zhengfu Xu, *High order maximum principle preserving finite volume method for convection dominated problems.*

[23] Chi-Wang Shu, *Essentially Non-Oscillatory and Weighted Essentially Non-Oscillatory Schemes for Hyperbolic Conservation Laws*, ICASE Report No. 97-65, (1997).

[24] Xu-Dong Liu, Stanley Osher, and Tony Chan, *Weighted Essentially Non-oscillatory Schemes*, Journal of Computational Physics, Volume 115, Issue 1, November 1994, Pages 200-212.

[25] K. Böhmer and H. Stetter, *Defect correction methods. Theory and applications*, (1984).

[26] S. Boscarino, *Error analysis of IMEX Runge-Kutta methods derived from differential-algebraic systems*, SIAM Journal on Numerical Analysis, 45 (2008), pp. 1600–1621.

[27] A. Christlieb, B. Ong, and J. Qiu, *Comments on high order integrators embedded within integral deferred correction methods*, Comm. Appl. Math. Comput. Sci, 4 (2009), pp. 27–56.

[28] Andrew Christlieb, Benjamin Ong, and Jing-Mei Qiu, *Integral deferred correction methods constructed with high order Runge-Kutta integrators*, Mathematics of Computation, 79 (2009), p. 761.

[29] A. Dutt, L. Greengard, and V. Rokhlin, *Spectral deferred correction methods for ordinary differential equations*, BIT Numerical Mathematics, 40 (2000), pp. 241–266.

[30] E. Hairer, S. Nørsett, and G. Wanner, *Solving ordinary differential equations: Nonstiff problems*, vol. 1, Springer Verlag, 1993.

[31] E. Hairer and G. Wanner, *Solving ordinary differential equations II: stiff and differential algebraic problems*, vol. 2, Springer Verlag, 1993.

[32] J. Huang, J. Jia, and M. Minion, *Accelerating the convergence of spectral deferred correction methods*, Journal of Computational Physics, 214 (2006), pp. 633–656.

[33] D. Kushnir and V. Rokhlin, *A highly accurate solver for stiff ordinary differential equations*, SIAM Journal on Scientific Computing, 34 (2012), pp. A1296–A1315.

[34] A. Layton, *On the choice of correctors for semi-implicit picard deferred correction methods*, Applied Numerical Mathematics, 58 (2008), pp. 845–858.

[35] A. Layton and M. Minion, *Implications of the choice of quadrature nodes for picard integral deferred corrections methods for ordinary differential equations*, BIT Numerical Mathematics, 45 (2005), pp. 341–373.

[36] Anita T. Layton and Michael L. Minion, *Implications of the choice of predictors for semi-implicit picard integral deferred corrections methods*, Comm. Appl. Math. Comput. Sci, 1 (2007), pp. 1–34.

[37] M. Minion, *Semi-implicit spectral deferred correction methods for ordinary differential equations*, Commun. Math. Sci, 1 (2003), pp. 471–500.

[38] R. D. Skeel, *A theoretical framework for proving accuracy results for deferred corrections*, SIAM J. Numer. Anal., 19 (1982), pp. 171–196.

[39] L. Yuan and C.-W. Shu, *Discontinuous galerkin method based on non-polynomial approximation spaces*, Journal of Computational Physics, 218 (2006), pp. 295–323.

[40] Chang-Yeol Jung and Thien Binh Nguyen, *New time differencing methods for spectral methods*, Journal of Scientific Computing, 2015.

[41] Tommaso Buvoli, *A class of exponential integrators based on spectral deferred correction*, 2015.