TOWARDS ASYNCHRONOUS SIMULATIONS OF TURBULENT FLOWS: ACCURACY, PERFORMANCE, AND OPTIMIZATION

A Dissertation

by

KOMAL KUMARI

Submitted to the Graduate and Professional School of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Diego A. Donzis
Committee Members,	Raktim Bhattacharya
	Sharath Girimaji
	Lawrence Rauchwerger
Head of Department,	Srinivas R. Vadali

December 2021

Major Subject: Aerospace Engineering

Copyright 2021 Komal Kumari

ABSTRACT

Our understanding of turbulence has heavily relied on high-fidelity Direct Numerical Simulations (DNS) that resolve all dynamically relevant scales. But because of the inherent complexities of turbulent flows, these simulations are computationally very expensive and practically impossible at realistic conditions. Advancements in high performance computing provided much needed boost to the computational resources through increasing levels of parallelism and made DNS realizable, even though only in a limited parameter range. As the number of processing elements (PEs) in parallel machines increases, the penalties incurred in current algorithms due to necessary communications and synchronizations between PEs to update data become significant. These overheads are expected to pose a serious challenge to scalability on the next-generation exascale machines. An effective way to mitigate this bottleneck is through relaxation of strict communication and synchronization constraints and proceed with computations asynchronously $i.e.$ without waiting for updated information from the other PEs. In this work, we investigate the viability of such asynchronous computing using high-order Asynchrony-Tolerant (AT) schemes for accurate and scalable simulations of reacting and non-reacting turbulence at extreme scales. For this, we first assess the important numerical properties of AT schemes, including conservation, stability, and spectral accuracy. Through rigorous mathematical analysis, we expose the breakdown of the standard von Neumann analysis for stability of multi-level schemes, even for widely used synchronous schemes. We overcome these limitations through what we call the generalized von Neumann analysis that is then used to assess stability of the AT schemes. Following which, we propose and implement two computational algorithms to introduce asynchrony in a three-dimensional compressible flow solver. We use these to perform first of a kind asynchronous simulation of compressible turbulence and analyze the effect of asynchrony on important physical characteristics of turbulence. Specifically we show that both large-scale and scale-scale features including highly intermittent instantaneous events, are accurately resolved by these algorithms. We also show excellent strong and weak scaling of asynchronous algorithms up to a processor count of $P = 262144$

because of significant reduction in communication overheads. As a precursor to the development of asynchronous combustion codes for simulations of more challenging problems with additional physical and numerical complexities, we investigate the effect of asynchrony on several canonical reacting flows. Furthermore, for problems with shocks and discontinuities, such as detonations, we derive and verify AT-WENO (weighted essentially non-oscillatory) schemes. With the ultimate goal to derive new optimal AT schemes we also develop a unified framework for the derivation of finite difference schemes. We show explicit trade-offs between order of accuracy, spectral accuracy and stability under this unifying framework, which can be exploited to devise very accurate numerical schemes for asynchronous computations on extreme scales with minimal overheads.

DEDICATION

To Mummy, Papa and Bhai

ACKNOWLEDGMENTS

It is an absolute honor to thank all the wonderful people in my life who helped me achieve my dream of earning a Ph.D degree and made every bit of this journey worth treasuring.

First, I would like to express my sincere gratitude to my Ph.D advisor, Dr. Diego A. Donzis, for this opportunity and for his invaluable guidance throughout this work. I have always been in awe of his passion, commitment and rigour towards his research, teaching and every task that he undertakes. Over these past five years, besides skills essential to succeed as a researcher, through him I have also learned important life lessons that I will forever try to imbibe. He gave me ample opportunities to explore ideas, share research with the experts in the community and develop collaborations with peers. He always made himself available to answer any questions and address concerns of all sorts even when his schedule was busy. I am truely grateful to Prof Donzis for his constant encouragement, support, kindness and patience all these years. He has been an incredible mentor and a wonderful role model.

I am thankful to Dr. Raktim Bhattacharya with whom I had the opportunity to work in close collaboration. I thoroughly enjoyed our meetings that often led to very interesting tangential discussions. I would also like to thank Dr. Lawrence Rauchwerger for taking the time to meet with me even when he was visiting A&M for very short times. I am grateful to Dr. Sharath Girmaji for suggestions during my preliminary examination, discussions during APS and for his very insightful courses on turbulence. I want to thank Dr. Jacqueline Chen for the opportunity to work with her as part of NSF-INTERN program. Her insights and perspectives made my internship an excellent learning experience. I am also immensely thankful to Dr. Aditya Konduri for all his help and guidance during our research collaboration and even after that. I would also like to thank our excellent department staff–Gail, Colleen and Kelsie– for all the administrative help and ensuring everything went by smoothly.

A very big thanks to my awesome lab mates–Dr. John, Dr. Sualeh and Akanksha. The long thought-provoking discussions and the flux of ideas during our group meetings and casual hangouts

helped me improve my work and presentation skills tremendously. A very special thanks to Vedang for his constant encouragemnt and for bearing with me through both good times and not-so-great times. I also want to thank all my wonderful friends– Sneha, Priya, Pawan, Riya, Godara, Ankita, Divya, Rishita, Dammati– for making this journey fun, exciting and memorable.

Finally I want to thank my family for always being so supportive of all my decisions. Thank you Papa, Mummy and Bhai for showering me with unconditional love, and for innumerble prayers throughout these years. Without their continuous support and confidence in me, this PhD would have been an impossible task. I dedicate my Ph.D to them.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a dissertation committee consisting of Professors Diego Donzis (advisor), Raktim Bhattacharya of the Department of Aerospace Engineering, Sharath Girimaji of the Department of Ocean Engineering, and Lawrence Rauchwerger of the Department of Computer Engineering.

All other work conducted for the dissertation was completed by the student independently under the supervision of Dr Diego A. Donzis. The work on reacting flows was done as part of NSF-INTERN program with Dr Jacqueline Chen at Sandia National Laboratories.

Funding Sources

Graduate study and research were supported by NSF grants OCI-1054966, CCF-1439145, and CBET-1605914 and used resources provided by the Extreme Science and Engineering Discovery Environment (XSEDE) supported by NSF.

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

1.1 Asynchronous simulations of turbulent flows: An Overview

Turbulence is one of the most common state of fluid motion, prevalent in both nature and engineering flows. It finds application in a diverse range of fields including high speed air and ground transportation, engine combustion, atmospheric mixing, astrophysics, medicine and innumerable other processes of academic and industrial relevance. Under the continuum limits, turbulent flows can be mathematically modelled by the Navier-Stokes (NS) equations. These NS equations are conservation laws that are highly non-local and non-linear which makes them notoriously hard to handle analytically or theoretically. Turbulence is also chaotic in nature and thus highly sensitive to initial and boundary conditions. Another inherent feature of turbulent flows is a wide range of unsteady spatial and temporal scales stemming from the non-linear interactions. This range of scales grows with the Reynolds Number (R_λ) (ratio between the inertial and viscous forces) which is typically very high for real applications. Because of all the complexities engendered by turbulence, despite sustained efforts spanning several decades from mathematicians, physicists and engineers alike, it still remains a major scientific and computational challenge.

For over four decades now, fundamental understanding of turbulence has relied extensively on numerical simulations. These simulations give invaluable insights into the complex physical processes that are beyond the scope of theoretical studies [1, 2] and are critical for development of turbulence models used in low-cost industrial simulations. Typically, for numerical simulations to be accurate, the computational domain should be very finely discretized in order to resolve the so-called Kolmogorov scale [3], the smallest dynamically relevant scale in a turbulent flow. At the same time, the computational domain should be large enough to accommodate the largest scales of motion in the flow. Furthermore, the simulation time should be sufficiently long to capture the slow evolution of the largest scale while the time-step size should be small enough to capture the fast Kolmogorov time scale characteristic of the smallest scales. Simulations that follow these

stringent constraints and consequently, accurately resolve the physics of all relevant scales are known as Direct Numerical Simulations (DNS) [4, 5]. Using classical scaling relations based on Kolmogorov ideas [3] for grid spacing and a CFL condition for time-step size, the computational work grows steeply as Re_λ^6, though more recent work suggests Re_λ^8 if all intermittent events are to be resolved [6]. Due to this steep power-law dependence, DNS is computationally prohibitively expensive. However, the advancements in high performance computing platforms enabling deployment of hundreds of thousands of processors concurrently made DNS realizable, though only up to a modest Reynolds Number. In order to assess the evolution of turbulent simulations over the years, the growth in peak computing power expressed in terms of operations per second along with the highest Reynolds number in incompressible homogeneous isotropic turbulence simulations is shown in Fig. 1.1. We see that both computational power and Reynolds number exhibit an exponential growth in time. A similar observation was also made in DNS of compressible turbulence [7]. This sustained growth in the problem size of DNS or highest achievable Reynolds numbers in the past decades has been made possible by increasing levels of parallelism. Besides DNS in a triple-periodic domain, the massive computing resources have also been used for simulations of more complex problems, including turbulence combution with detailed reaction mechanisms and complex geometries.

From these trends, one can anticipate that on the next-generation exascale machines that are envisioned to have billions of processing elements (PEs) and perform operations three orders of magnitude faster than the current state-of-the-art supercomputers, DNS at much higher Reynolds number would be feasible. However, virtually all numerical methods used in current state-of-the-art solvers require PEs to communicate and synchronize at all times in order to meet accuracy requirements. The resulting overheads become exceedingly expensive at extreme scales and are expected to be a major bottleneck to scalability on the future exascale machines [8, 9]. This will ultimately limit the anticipated Reynolds number on these machines.

Because of their extreme levels of parallelism, exascale machines will be subject to node failures, intense data movements, load imbalances, and system noise. These coupled with limited scal-

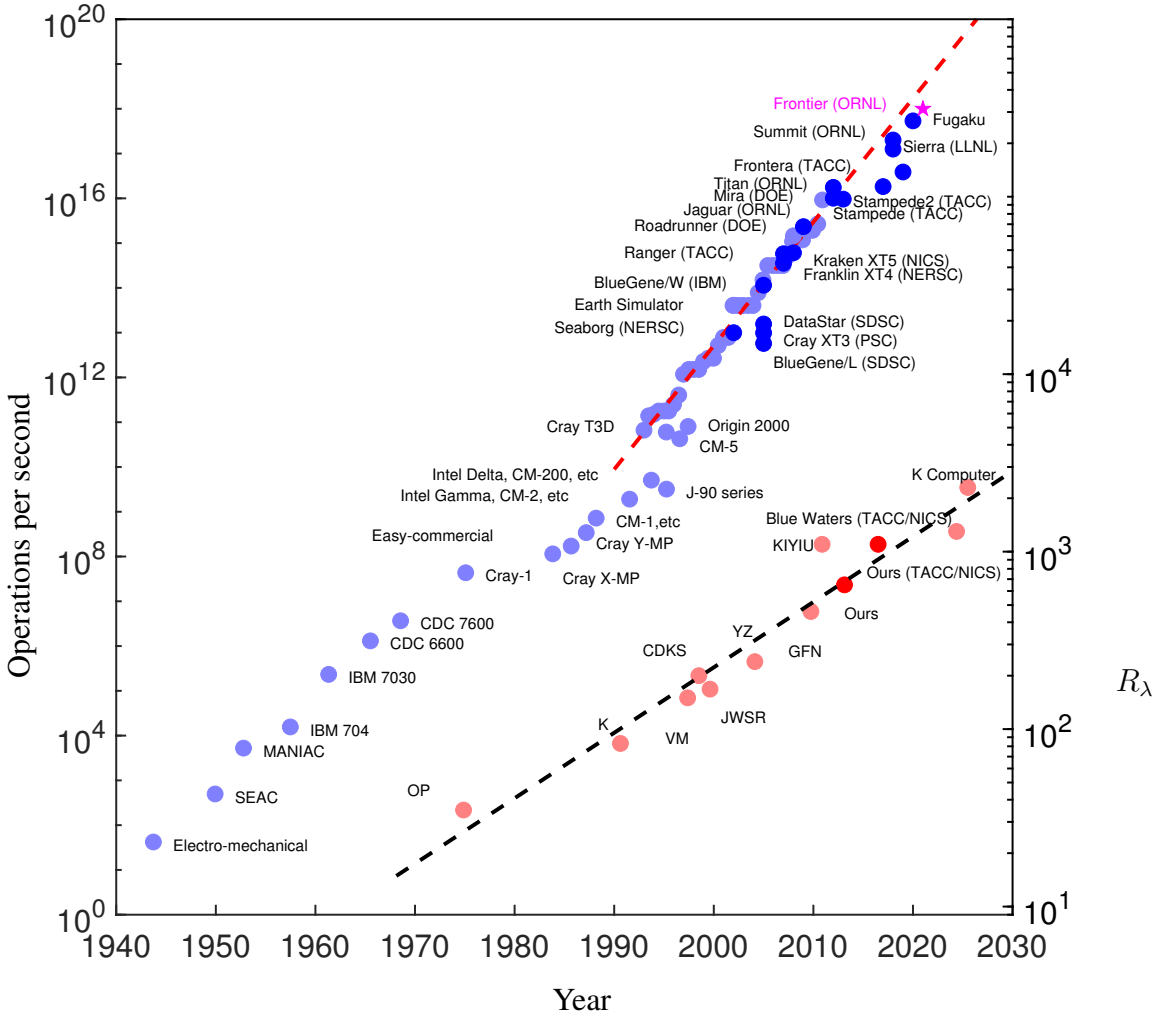

Figure 1.1: Evolution of computational power (left axes) and Reynolds number (right axes) over past decades. The magenta start indicates the next-generation exascale machine to be deployed in 2021.

ability of current approaches, clearly suggest that paradigm shifts at both hardware and software levels are required for efficient utilization of the massive computing power of exascale machines [8, 10, 11]. This includes development and validation of novel and resilient numerical schemes and computational algorithms that can scale to an increasing number of PEs, whilst incurring minimal overheads and accurately resolving the multi-scale physics engendered by the NS equations. This is the main thrust of the present work.

In the past years, some work has focused on relaxing the synchronization requirements among the processors and perform so-called asynchronous numerical simulations, realized either by modifying the governing equations or by modifying the numerical methods. However, these efforts were severely limited to lower orders of accuracy and restricted to certain class of PDEs [12, 13, 14, 15]. A new and more generalized approach, extensible to arbitrarily high orders of accuracy, has been recently developed [9, 16] to derive the so-called Asynchrony-Tolerant (AT) finite-difference schemes. But even in these studies numerical accuracy is investigated only for simplified model problems in low dimensions. The ability of these schemes to accurately simulate realistic three-dimensional turbulent flows or more challenging reacting flows has not been done before. Without careful assessment of the numerical and parallel performance of these schemes it is unclear whether they can indeed provide a path towards exascale simulations in future massively parallel systems.

1.2 Asynchrony-tolerant (AT) finite difference schemes

For the numerical simulation of any time-dependent partial differential equation (PDE), like the NS equations, the first step is to discretize the physical domain into N grid-points. Fig. 1.2(a) illustrates one such discretized domain in one-dimension with a uniform grid spacing Δx. The spatial derivatives in the PDE are then approximated, at each grid point i, using an appropriate numerical method. These spatial derivatives, computed at a given time level n, are used for advancing the system in time in small increments of Δt.

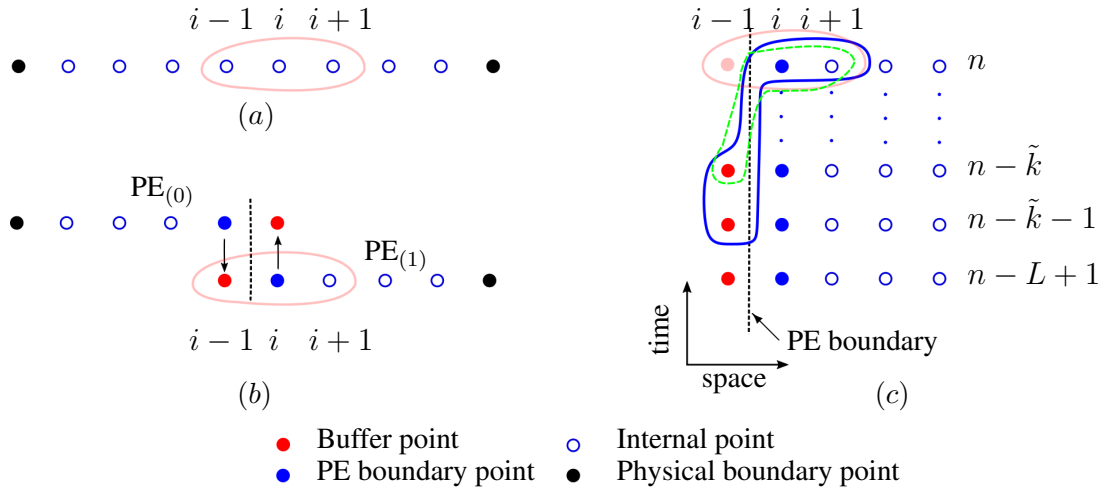

Figure 1.2: 1D domain discretization: (a) into one PE (serial) (b) into two PEs in parallel. (c) shows the left boundary of $\text{PE}_{(1)}$ with L time levels. Different lines are: standard synchronous scheme in faded-red (Eq. (1.2)), standard scheme used asynchronously in dashed-green (Eq. (1.3)), second-order AT scheme in solid-blue (Eq. (1.9)) at point i and PE boundary in dashed-black. For all the schemes $M = 1$.

1.2.1 Concept

Explicit finite difference schemes have been extensively used for approximation of derivatives in PDEs including in massive simulations of turbulent reacting flows [17, 18]. In finite difference schemes, the derivative at the i^{th} grid point can be approximated as the weighted average of the values at its neighboring points. For example, for a spatially and temporally varying function $u(x,t)$, at a grid point i and time level n, the d-th spatial derivative can be written as,

$$\left.\frac{\partial^{(d)}u}{\partial x^{(d)}}\right|_i^n \approx \frac{1}{(\Delta x)^d}\sum_{m=-M}^{M} a_m u_{i+m}^n + \mathcal{O}(\Delta x^{p+1}), \tag{1.1}$$

where M is the stencil size in each direction and $u(x_i,t_n) = u_i^n$ and $p+1$ is the order of accuracy. Here the weights or coefficients a_m are computed by imposing order of accuracy constraints on the Taylor expansion of u_{i+m} in space. The resulting schemes are unique and have a minimum truncation error for a given stencil size M. A standard second-order finite difference scheme is

obtained for $M = 1$, which for the second derivative reads as,

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_i^n = \frac{u_{i-1}^n - 2u_i^n + u_{i+1}^n}{\Delta x^2}, \tag{1.2}$$

and is outlined in faded red in Fig. 1.2. In Eq. (1.2), the computation of spatial derivatives at time level n, requires data at the same time level. Thus, these computations can be trivially done using a serial code, where only one PE is used and therefore all data is available locally. This is shown explicitly in Fig. 1.2(a).

However, with the advent and rapid developments in parallel computing paradigm, the general trend is to have hundreds of thousands of PEs working concurrently rather than one huge PE doing computations single-handedly. When such is the case, the discretized physical domain is decomposed into P sub-domains, where P is the number of PEs. Such a domain decomposition in 1D using two PEs is shown in Fig. 1.2(b). Spatial derivatives can be computed using data available within the PE at the so-called internal points. However, at the PE boundaries, values at the grid points that are a part of the neighboring PE are required for accurately computing the derivative. This data is typically communicated into the buffer arrays (red circles in Fig. 1.2) by the neighboring PEs at each time level. In order to have the most updated values at these buffer points, PEs are forced to stall and synchronize, leading to overheads that are detrimental to parallel performance and limit the scalability to extreme scales.

If instead, one allows for computation to proceed without waiting for the most updated values, that is, using delayed values at the buffer points, the derivative can be computed as

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_i^n = \frac{u_{i-1}^{n-\tilde{k}} - 2u_i^n + u_{i+1}^n}{\Delta x^2}. \tag{1.3}$$

Here $n - \tilde{k}$ is the latest available time level written in terms of the delay $\tilde{k}$. This is essentially asynchronous computation of derivatives and is schematically shown as a dashed green curve in Fig. 1.2(c). While conceptually this leads to reduction in synchronization and communication overheads, the delay $(\tilde{k})$ severely degrades the accuracy of the standard scheme to zeroth order [9].

This motivated the derivation of a new family of the so-called Asynchrony-Tolerant (AT) schemes that are resilient to asynchrony.

1.2.2 Derivation

AT schemes can be seen as a generalization of standard finite differences, where the computation of spatial derivatives use function values of neighboring points in both space and time. We can write a general d-th spatial derivative of a spatially and temporally varying function $u(x,t)$ at a grid point i and time level n as,

$$\left.\frac{\partial^{(d)}u}{\partial x^{(d)}}\right|_i^n \approx \frac{1}{(\Delta x)^d}\sum_{l=0}^{L}\sum_{m=-M}^{M} a_m^l u_{i+m}^{n-l} + \mathcal{O}(\Delta x^p), \tag{1.4}$$

where M is the stencil size in each direction, L is the number of time levels used and a_m^l's are the coefficients. Following the general rules for derivation of finite difference, these coefficients can be computed by imposing order of accuracy constraints on the Taylor series expansion of u_{i+m}^{n-l} about point u_i^n in both space and time which is given by

$$u_{i+m}^{n-l} = \sum_{q=0}^{\infty}\sum_{s=0}^{\infty} u^{(q,s)}\big|_i^n \frac{(m\Delta x)^q(-l\Delta x)^s}{q!s!}, \tag{1.5}$$

where $u^{(q,s)}$ is the q-th and s-th partial derivative in space and time, respectively, of u. Substituting Eq. (1.5) in the right hand side of Eq. (1.4), we get

$$\sum_{l=0}^{L}\sum_{m=-M}^{M} a_m^l u_{i+m}^{n-l} = \sum_{l=0}^{L}\sum_{m=-M}^{M} a_m^l \sum_{q=0}^{\infty}\sum_{s=0}^{\infty} u^{(q,s)}\big|_i^n \frac{(m\Delta x)^q(-l\Delta x)^s}{q!s!}. \tag{1.6}$$

Eq. (1.6) represents a numerical approximation of the second derivative of order $p+1$, if the coefficient of term $u^{(2,0)}$ in this linear combination is equal to one and all terms in yielding a truncation error with the power of Δx less than $p+1$ are zero. However, due to presence of coupled terms involving both Δt and Δx, in order to correctly identify and eliminate low-order terms, we have to use a CFL type relation of the form $\Delta t \sim \Delta x^r$. We can then write the order of

accuracy constraint as

$$\sum_{l=0}^{L}\sum_{m=-M}^{M} a_m^l \frac{(m\Delta x)^q(-l\Delta x)^s}{q!s!} = \begin{cases} 1, & (q,s) = (d,0) \\ 0, & q + rs \leq d + p;\ (q,s) \neq (d,0) \end{cases} \tag{1.7}$$

where for the d^{th} derivative of order p. In the absence of delays, *i.e.* when $L = 0$, Eq. (1.7) reduces to

$$\sum_{m=-M}^{M} a_m \frac{(m\Delta x)^q}{q!} = \begin{cases} 1, & q = d \\ 0, & q \leq d + p;\ q \neq d \end{cases} \tag{1.8}$$

which is equivalent to the constraints imposed on standard finite difference schemes.

While Eq. (1.7) gives the exact constraints that need to be satisfied to get the AT scheme of desired order. The coefficients of these AT schemes, depends upon a number of choices, including stencil size M, time levels L, relation between the time-step and gird-size ($\Delta t \sim \Delta x^r$), nature (symmetric $|c^0_{-m}| = |c^0_m|$ or asymmetric $|a^0_{-m}| = |a^0_m|$). These choices have direct implications on the accuracy, memory requirements and performance of the schemes. While schemes can directly be computed by constructing a linear system of equation from Eq. (1.7), because of several linearly dependent terms a solution is not always guaranteed. Additional constraints can be imposed to get schemes that reduce to standard difference schemes in the absence of delays. The details of both of these methodologies has been described detail in [16]. As an example, a second-order AT scheme for the second derivative at the left boundary Eq. (1.9) with stencil $M = 1$ in space, and a delay of $\tilde{k}$ can be written as,

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_i^n = \frac{-\tilde{k}u_{i-1}^{n-\tilde{k}-1} + (\tilde{k}+1)u_{i-1}^{n-\tilde{k}} - 2u_i^n + u_{i+1}^n}{\Delta x^2}, \tag{1.9}$$

where a diffusive CFL relation of the form $\Delta t \sim \Delta x^2$ is used to relate spatial and temporal resolutions. This scheme is shown schematically in Fig. 1.2 with a solid blue curve. These schemes use multiple consecutive time levels on the delayed side, depending upon the order of accuracy. Also, the coefficients are a function of delays $\tilde{k}$ that reduce to constant coefficients for SFD when

$\tilde{k} = 0$. The delays seen by a processor depend upon machine characteristics such as network latency and bandwidth, clock rate and routing algorithm, and are essentially random. Therefore, the coefficients of the AT schemes that depend upon these delays, have to be computed dynamically at runtime. The specific details of how delays are introduced in a three-dimensional Navier-Stokes solver are described in Chapter 5

1.3 Objectives of the Present Work

The objectives of the present work can be now listed:

1. Assess the important numerical properties of novel asynchrony-tolerant (AT) finite difference schemes including, conservation and global order of accuracy with multi-stage temporal schemes.

2. Assess the stability of the AT schemes using a new generalized von Neuman analysis that overcomes severe limitations identified in the standard analysis.

3. Develop of a unified framework for derivation of optimized finite difference schemes, both synchronous and asynchrony-tolerant.

4. Perform first-of-its-kind asynchronous simulations of compressible turbulence

   (a) Develop a flow solver that facilitates asynchronous communications using two proposed algorithms.

   (b) Verify the efficacy of AT schemes in resolving important large and small scale characteristics of turbulence.

   (c) Perform strong and weak scaling study of the asychronous solver.

5. Investigate the effect of data asynchrony on canonical reacting flows with one-step and detailed reaction mechanism and develop AT-WENO schemes for solving problems with shocks and discontinuties.

2. NUMERICAL PROPERTIES OF ASYNCHRONY-TOLERANT SCHEMES*

2.1 Introduction

For numerical simulation of any physical process governed by partial differential equations (PDEs), the choice of numerical schemes used for the approximation of both spatial and temporal derivatives is very critical. The numerical fidelity of the simulation explicitly depends upon how closely the discrete approximation reproduces the continuous system. This in turn depends upon the properties of the underlying numerical discretization scheme, for example, its order of accuracy, spectral accuracy and stability. In general, the finite difference schemes used for compuation of spatial derivatives are derived solely based on the order of accuracy requirements. However, the order only dictates how steeply the truncation error decreases as the spatial grid is refined. The asynchrony-tolerant (AT) schemes are also derived by eliminating sufficient lower order terms, as described in the previous Chapter, to obtain a scheme of required order of accuracy. Numerical experiments involving both linear and non-linear equations for the verification of the order of the AT schemes were presented in [16]. The current chapter focuses on the investigation of other important numerical properties of AT schemes before using these schemes to perform simulations of complex physical systems.

A property of interest for accurate simulations of PDEs that represent conservation laws, is the conservation property of the numerical discretization of these laws. While standard finite difference schemes are known to satisy the conservation property due to the telescoping effect of the fixed coefficients, the AT schemes present some challenges since their coefficients depend upon the delays encountered at the PE boundary. Due to this dependence of the coefficients on the delays, the scheme at each boundary of each PE may be different. As a result, the telescoping property of the coefficients of the standard schemes is affected. This could lead to spurious numerical source and thus, alter the true physics of the system being solved. It is therefore very important to clearly

*Parts of this chapter are reprinted from Journal of Computational Physics, Volume 419, Komal Kumari and Diego A. Donzis, “Direct numerical simulations of turbulent flows using high-order asynchrony-tolerant schemes: Accuracy and performance", Pages 109626, 15 October 2020, with permission from Elsevier.

understand how accurately AT schems represent conservation laws and this is done in the current chapter. Through analysis of a simple domain we compute the exact residual that is obtained when AT schemes are used for discretization of conservation laws.

Since the PDEs involve derivatives in both space and time, the order of accuracy of the fully-discrete system depends upon the order of both spatial and temporal discretization scheme. Both multi-level Adams-Bashforth (AB) and multi-stage Runge-Kutta (RK) temporal schemes are used for high-order tempral discretization along with finite difference schemes for spatial derivatives. While the global order of accuracy (space + time) wherein AT schemes are used in conjuction with AB schemes has been verified in [16], their coupling with RK schemes has not been investigated. Furthemore, due to the computation of multiple stages in high-order RK schemes, these require efficient algorithms in order to facilitate asynchronous computations with AT schemes. In this chapter the global order of accuracy of AT schemes with RK schemes is evaluated analytically and a methodology is presented to use them efficiently in an actual simulation. The trade-off between communications and computations is also discussed.

The stability of AT schemes is yet another property that is essential to ensure that perturbations, for example, due to round-off or truncation error do not grow unboundedly in time. While the effect of data asynchrony on the stability of standard finite difference schemes is discussed in [9], stability of AT schemes has not assessed before. Due to the random nature of coefficients of the AT schemes that eventually leads to a different scheme being used at different processor boundaries at every time-step, the stability analysis of these schemes is feasible only under simplifying assumptions. In this chapter we assess the stability of AT schemes using standard methodologies in physical and spectral space. We define stability limit of AT schemes in terms of an asynchronous CFL that equal to the standard synchronous stability limit. This, as we show here, yields an effective asynchronous time-step that can be used to explain the reduction in stability limit with increasing delay level at the processor boundaries. The subsequent sections discuss each of these numerical properties in detail [19].

2.2 Conservation property of AT schemes

The Navier-Stokes equations represent conservation laws (mass, momentum and energy). For accurate numerical simulations, it is important that the numerical discretization of these laws also satisfy the global conservation. This ensures that no spurious sources are introduced because of numerical artifacts that can result in violation of the underlying conservation. To investigate the effect of numerical discretization on conservation property, it is common to consider a simpe one-dimensional form of a conservation law,

$$\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} = 0, \tag{2.1}$$

where $f(x,t)$ is the flux of some conserved quantity $u(x,t)$, for example, mass or momentum. The total variation of $u(x,t)$ over a domain $[0,1]$ depends only upon the flux f through the boundaries. This can be expressed more precisely by integrating Eq. (2.1) over the domain,

$$\frac{d}{dt}\int_0^1 u(x,t)dx = \int_0^1 \left(\frac{\partial f}{\partial x}\right) dx = f(1,t) - f(0,t), \tag{2.2}$$

showing explicit dependence of variation in $u(x,t)$ only on the flux at the boundaries. For periodic boundary condition *i.e* $f(0,t) = f(1,t)$, this flux is equal to zero. When the derivatives are approximated numerically, it is desirable that the discrete form of the above conservation law is also satisfied to a given accuracy. Consider a generalized spatial discretization given by Eq. (1.4), for N grid points and time level n, to yield

$$\int_0^1 \left.\frac{\partial f}{\partial x}\right|^n dx = \sum_{i=1}^{N}\left(\frac{1}{\Delta x}\sum_{l=0}^{L}\sum_{m=-M}^{M} c_{ml} f_{i+m}^{n-l}\right). \tag{2.3}$$

For $M = 1$, corresponding to an AT scheme with leading truncation error term of order $\mathcal{O}(\Delta x^a)$ where $a = 2$ when $\Delta t \sim \Delta x^2$ [16], and a domain decomposed into 2 PEs such that PE$^{(1)}$ holds gridpoints $i \in [1, N/2]$ and PE$^{(2)}$ holds gridpoints $i \in [N/2+1, N]$ and satisfies periodic boundary

conditions, we can write Eq. (2.3) as

$$\begin{aligned}\int_0^1 \left|\frac{\partial f}{\partial x}\right|^n dx =& \frac{f_2^n - (\tilde{k}_l^{(1)}+1) f_N^{n-\tilde{k}_l^{(1)}} + \tilde{k}_l^{(1)} f_N^{n-\tilde{k}_l^{(1)}-1}}{2\Delta x} + \sum_{i=2}^{N/2-1}\left(\frac{f_{i+1}^n - f_{i-1}^n}{2\Delta x}\right) \\ &+ \frac{(\tilde{k}_r^{(1)}+1) f_{N/2+1}^{n-\tilde{k}_r^{(1)}} - \tilde{k}_r^{(1)} f_{N/2+1}^{n-\tilde{k}_r^{(1)}-1} - f_{N/2-1}^n}{2\Delta x} \\ &+ \frac{f_{N/2+2}^n - (\tilde{k}_l^{(2)}+1) f_{N/2}^{n-\tilde{k}_l^{(2)}} + \tilde{k}_l^{(2)} f_{N/2}^{n-\tilde{k}_l^{(2)}-1}}{2\Delta x} \\ &+ \sum_{i=N/2+2}^{N-1}\left(\frac{f_{i+1}^n - f_{i-1}^n}{2\Delta x}\right) + \frac{(\tilde{k}_r^{(2)}+1) f_1^{n-\tilde{k}_r^{(2)}} - \tilde{k}_r^{(2)} f_1^{n-\tilde{k}_r^{(2)}-1} - f_{N-1}^n}{2\Delta x},\end{aligned} \tag{2.4}$$

where $\tilde{k}_l^{(1)}$ and $\tilde{k}_r^{(1)}$ are the delays on left and right boundary for PE$^{(1)}$ and $\tilde{k}_l^{(2)}$ and $\tilde{k}_r^{(2)}$ are the delays on left and right boundary for PE$^{(2)}$ and periodic boundary conditions are used. Because of the telescoping effect, the above expression can be simplified to

$$\begin{aligned}\int_0^1 \left|\frac{\partial f}{\partial x}\right|^n dx =& \frac{-(\tilde{k}_l^{(1)}+1) f_N^{n-\tilde{k}_l^{(1)}} + \tilde{k}_l^{(1)} f_N^{n-\tilde{k}_l^{(1)}-1}}{2\Delta x} + \left(\frac{f_{N/2}^n - f_1^n}{2\Delta x}\right) \\ &+ \frac{(\tilde{k}_r^{(1)}+1) f_{N/2+1}^{n-\tilde{k}_r^{(1)}} - \tilde{k}_r^{(1)} f_{N/2+1}^{n-\tilde{k}_r^{(1)}-1}}{2\Delta x} + \frac{-(\tilde{k}_l^{(2)}+1) f_{N/2}^{n-\tilde{k}_l^{(2)}} + \tilde{k}_l^{(2)} f_{N/2}^{n-\tilde{k}_l^{(2)}-1}}{2\Delta x} \\ &+ \left(\frac{f_N^n - f_{N/2+1}^n}{2\Delta x}\right) + \frac{(\tilde{k}_r^{(2)}+1) f_1^{n-\tilde{k}_r^{(2)}} - \tilde{k}_r^{(2)} f_1^{n-\tilde{k}_r^{(2)}-1}}{2\Delta x}.\end{aligned} \tag{2.5}$$

For the standard sychronous case, $\tilde{k}_l^{(1)} = \tilde{k}_r^{(1)} = \tilde{k}_l^{(2)} = \tilde{k}_r^{(2)} = 0$, that is, when delays are absent, all terms on the right-hand side of Eq. (2.5) cancel each other and the conservative property is trivially satisfied. In the presence of delays, on the other hand, this is not immediately obvious from Eq. (2.5). Further simplification of this equation can be done using a Taylor series expansion in time which leads to similar cancellation of all low-order terms yielding a residual of the order of Δx^3. More generally, for larger M, that is, for AT schemes of order $a = 2M$ and $\Delta t \sim \Delta x^2$, the residual is found to be

$$\int_0^1 \left|\frac{\partial f}{\partial x}\right|^n dx = \mathcal{O}(\Delta x^{a+1}). \tag{2.6}$$

Thus, we conclude that the AT schemes retain the conservative property up to an order higher than the order of the scheme. For the fourth-order schemes used for the simulation presented in Chapter 6 and Chapter 7, conservation is satisfied up to $\mathcal{O}(\Delta x^5)$. While this is a small residual, its effects on the quantities of interest will become clear from the numerical simulations presented subsequent Chapters.

2.3 Global order of accuracy

For the evolution of PDEs in time, spatial schemes need to be coupled with a temporal scheme of appropriate order, yielding a fully-discrete sytem. The global order of accuracy of the fully-discrete system so obtained depends on both the spatial and temporal discretizations. In order to compute this global order, a relation of the form $\Delta t \sim \Delta x^r$ is used to express the leading order truncation error term of the time discretization scheme also in terms of grid size (Δx). For example, for a fourth-order spatial scheme ($\mathcal{O}(\Delta x^4)$), if a second-order temporal discretization scheme ($\mathcal{O}(\Delta t^2)$) is used then the global spatial order is two if $\Delta t \sim \Delta x$ and four if $\Delta t \sim \Delta x^2$. One of the most widely used schemes for high-order temporal discretization is the multi-stage Runge-Kutta (RK) method. An S-stage explicit RK scheme for an equation of the form $du/dt = f(u,t)$ over a time step Δt is given by

$$u^{n+1} = u^n + \Delta t \sum_{s=0}^{S} b_s k_s, \tag{2.7}$$

where the stages k_s are computed using

$$\begin{aligned} k_1 &= f(u^n, t^n) \\ k_s &= f(u^n + \Delta t \sum_{i=1}^{s-1} (a_{si} k_i), t^n + c_s \Delta t). \end{aligned} \tag{2.8}$$

Here a_{si}, b_s, c_s are the coefficients of the RK scheme [20]. For an RK scheme, in advancing from time level n to $n+1$, intermediate stages k_s are computed. Each of these stages depends upon the previous stages as can be seen from Eq. (2.8). Thus, for simulations performed in parallel,

RK schemes require PEs to communicate and synchronize after every stage in order to update the corresponding values at buffer points. However, if simulations are performed with asynchronous communications using AT schemes, then a fractional delay will be encountered at the intermediate stages and different AT schemes will be required for each stage. An alternate method is proposed to effectively use multi-stage RK schemes with AT that circumvents the need to communicate at every stage and avoids fractional delays.

In the presence of delays, each stage k_s is computed using AT schemes at PE boundary points. This AT scheme uses delayed data at buffer points from multiple consecutive time levels. For the correct computation of k_s at the boundaries, all previous stages $k_{s-1}, k_{s-2}, \ldots, k_1$ are computed locally at both PE boundary and buffer points of each PE. Furthermore, at the delayed time levels required by the AT scheme, these stages are also computed at the internal points close to the PE boundary. In order to perform the additional computations at buffer points, a larger message of size (= *No. of stages in RK* $\times$ *spatial stencil*) has to be communicated across processors at every time step. However, the PEs no longer communicate at every stage of the RK. Thus, there exists a trade-off between communication which is expensive and computation which is cheap.

To illustrate that the RK schemes preserve the order of accuracy when used with high-order AT schemes, the one-dimensional diffusion equation is considered,

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2} \tag{2.9}$$

where α is the diffusivity and $u(x,t)$ is the velocity field. Denoting the right-hand-side of Eq. (2.9) by $f(u^n, t^n) = \partial^2 u^n / \partial x^2$ (without α), the temporal discretization at the j-th spatial point using a two-stage second-order RK scheme (RK2) is given by

$$u_j^{n+1} = u_j^n + \alpha \Delta t \left(\frac{k_{j,1}}{2} + \frac{k_{j,2}}{2} \right) \tag{2.10}$$

where $k_{j,1} = f(u_j^n, t^n)$, $k_{j,2} = f\left((u_j^n)^*, t^n + \Delta t\right)$ and $(u_j^n)^* = u_j^n + \Delta t(f(u_j^n, t^n)) = u_j^n + \Delta t \alpha k_1$. When a standard fourth-order central difference scheme is used for numerical approximation of

$f(u_j^n, t)$, the following $k_{j,1}$ and $k_{j,2}$ are obtained,

$$\begin{aligned} k_{j,1} =& \frac{-u_{j-2}^n + 16u_{j-1}^n - 30u_j^n + 16u_{j+1}^n - u_{j+2}^n}{12\Delta x^2} \\ k_{j,2} =& -\frac{\alpha\Delta t(k_{j-2,1}) + u_{j-2}^n}{12\Delta x^2} + \frac{4\left(\alpha\Delta t(k_{j-1,1}) + u_{j-1}^n\right)}{3\Delta x^2} - \frac{5\left(\alpha\Delta t\left(k_{j,1}\right) + u_j^n\right)}{2\Delta x^2} \\ &+ \frac{4\left(\alpha\Delta t(k_{j+1,1}) + u_{j+1}^n\right)}{3\Delta x^2} - \frac{\alpha\Delta t\left(k_{j+2,1}\right) + u_{j+2}^n}{12\Delta x^2}. \end{aligned} \tag{2.11}$$

The Taylor series expansion of Eq. (2.10) on substituting Eq. (2.11) then yields a truncation error of the form

$$\text{TE}_j^n|_{\text{sync}} = \frac{1}{6}\left(-f^{(0,3)}(x,t)\right)\Delta t^2 - \frac{1}{90}\alpha f^{(6,0)}(x,t)\Delta x^4 + \mathcal{O}(\Delta x^6, \Delta t^3, \Delta x^4\Delta t) \tag{2.12}$$

or

$$\text{TE}_j^n|_{\text{sync}} = (-\frac{r_\alpha^2 u^{(0,3)}(x,t)}{6\alpha^2} - \frac{1}{90}\alpha u^{(6,0)}(x,t))\Delta x^4 + \mathcal{O}(\Delta x^6), \tag{2.13}$$

where $r_\alpha = \alpha\Delta t/\Delta x^2$ is the diffusive CFL and the subscript "sync" denotes synchronous. Note that from Eq. (2.13) the global spatial order of accuracy is four.

When a fourth-order AT scheme is used at the boundary points for computation of spatial derivatives, we have

$$\begin{aligned} k_{j,11} =& \left(\tilde{k}^2 + \tilde{k}\right)\frac{\left(-u_{j-2}^{n-\tilde{k}-2} + 16u_{j-1}^{n-\tilde{k}-2} - 30u_j^n + 16u_{j+1}^n - u_{j+2}^n\right)}{24\Delta x^2} \\ &- \left(\tilde{k}^2 + 2\tilde{k}\right)\frac{\left(-u_{j-2}^{n-\tilde{k}-1} + 16u_{j-1}^{n-\tilde{k}-1} - 30u_j^n + 16u_{j+1}^n - u_{j+2}^n\right)}{12\Delta x^2} \\ &+ \left(\tilde{k}^2 + 3\tilde{k} + 2\right)\frac{\left(-u_{j-2}^{n-\tilde{k}} + 16u_{j-1}^{n-\tilde{k}-1} - 30u_j^n + 16u_{j+1}^n - u_{j+2}^n\right)}{24\Delta x^2} \end{aligned} \tag{2.14}$$

which is used to compute $k_{j,2} = f(u_j^n + \Delta t\alpha k_1, t^n + \Delta t)$ with $f(.,.)$ being evaluated using the fourth-order AT-scheme as well. Once again, the Taylor series expansion is used to obtain the

truncation error,

$$\text{TE}_j^n|_{\text{AT},\tilde{k}} = -\frac{5}{24}\alpha\tilde{k}\left(\tilde{k}^2+3\tilde{k}+2\right)u^{(0,3)}(x,t)\frac{\Delta t^3}{\Delta x^2} - \frac{1}{6}u^{(0,3)}(x,t)\Delta t^2 - \frac{1}{90}\alpha u^{(6,0)}(x,t)\Delta x^4 + \mathcal{O}(\tilde{k}^3\frac{\Delta t^3}{\Delta x}). \tag{2.15}$$

The above expression simplifies to give a global fourth-order truncation error term when r_α is substituted,

$$\text{TE}_j^n|_{\text{AT},\tilde{k}} = -\left(\frac{75\tilde{k}\left(\tilde{k}^2+3\tilde{k}+2\right)r_\alpha^3 u^{(0,3)}(x,t) + 60r_\alpha^2 u^{(0,3)}(x,t) + 4\alpha^3 u^{(6,0)}(x,t)}{360\alpha^2}\right)\Delta x^4 + \mathcal{O}(\Delta x^6). \tag{2.16}$$

When the delay is zero *i.e.* $\tilde{k}=0$, the truncation error in Eq. (2.16) reduces to $\text{TE}_j^n|_{\text{sync}}$ (Eq. (2.13)) which is the synchronous truncation error at the internal points. Therefore, RK schemes can be used with the AT-schemes without affecting the global order of accuracy. During a simulation the stencil data at older time levels for the buffer as well as internal points is available, and hence standard synchronous schemes can be used to compute the sub-stages $k_{j,s}$ at these points for efficient implementation of RK with AT. Numerical simulations of linear equations were performed using this approach and the global order of accuracy was preserved.

2.3.1 Runge-Kutta vs. Adams-Bashforth schemes

High order explicit temporal methods including multistage Runge-Kutta (RK) schemes and multistep Adams-Bashforth schemes, are very common choices of temporal discretizations for numerical simulations of complex physical processes governed by PDEs. While RK schemes are known for their good stability characteristics [20], the computation of a stage of RK requires communication across all neighbors after every stage. This communication frequency can be reduced by increasing the size of the message such that all the stages can be computed within the processor, as described in the previous section. However, in a 3D domain this requires processors to communicate across all 26 (faces + edges + corners) neighbors. The right-side of the PDE also needs to be evaluated s-times, per PE, per time-step, for an $s\text{-}stage$ RK scheme. Consequently, RK schemes are computation and communication intensive. On the other hand, multi-step Adams-Bashforth

(AB) schemes offer more flexibility in terms of implementation and also require less communications. A general AB scheme with T steps for an equation of the form $\partial u/\partial t = f$, can be written as,

$$u_i^{n+1} = u_i^n + \Delta t \sum_{m=0}^{T-1} \beta_m f_i^{n-m}, \tag{2.17}$$

where the coeffcients β_m depend upon the desired order of accuracy [21]. Not only can AB be efficiently implemented with only six communications per PE per time-step, it only requires computation of f^n every time-step since $f^{n-m}, m > 0$ is used from previous steps. Furthermore, the computation of f^{n-m} using AT schemes does not alter the order of accuracy of AB schemes [16]. Thus, for our three-dimensional compressible flow solver we use second-order AB schemes for the temporal evolution in both synchronous and asynchronous simulations (Chapter 6). For one-dimensional simulations of reacting flow presented in Chapter 7 we use RK schemes with the methodology described in the previous section.

2.4 Stability analysis

Asynchrony at the processor boundaries and the associated random nature of delays and the coefficients of AT schemes, can introduce random numerical errors. These error can in turn trigger instabilities, especially if the delay ($\tilde{k}$), bounded by L, is very large [16]. Since the delays are random with some machine-dependent probability distribution, the instantaneous delay varies with both time step and processor boundary. Consequently, the parameter space or the number of states in this discrete random system grows exponentially [22] and makes sability analysis in a general sense intractable. Thus, some simplifying assumptions need to be made in order to assess the stability of AT schemes. For example, we assume that same delay (hence same AT scheme) is used across all processors at all times. We also assume that each processor has exactly one grid-point and delays can be considered on either both sides or only one side. Under these assumptions we can compute the maximum CFL for a given delay L, such that the AT scheme is stable. This analysis is perfomed in both physical and spectral space.

2.4.1 Matrix stability analysis

We consider a simple 1D diffusion equation,

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2} \tag{2.18}$$

where α is the diffusivity constant and $u(x,t)$ is the velocity or temperature field. This equation is discretized using a second-order AT scheme in space and forward Euler in time. Following [16], we can discretize Eq. (2.18) at the i^{th} grid point with delay $\tilde{k}_l$ at the left boundary and $\tilde{k}_r$ at the right boundary as,

$$u_i^{n+1} = u_i^n + \frac{\alpha \Delta t}{\Delta x^2}\left((\tilde{k}_l + 1)u_{i-1}^{n-\tilde{k}_l} - \tilde{k}_l u_{i-1}^{n-\tilde{k}_l-1} - 2u_i^n + (\tilde{k}_r + 1)u_{i+1}^{n-\tilde{k}_r} - \tilde{k}_r u_{i+1}^{n-\tilde{k}_r-1}\right). \tag{2.19}$$

For the above discretization we have considered an extreme case scenario where $P = N$ and $N_T = 1$ $i.e.$ every PE has only one grid point. It can be shown that Eq. (2.19) preserves the order of accuracy despite delays on both boundaries. Next we define $U^n := [u_0^n, u_1^n, ..., u_N^n]$ and $V^n := [U^n,\ ...\ ,\ U^{n-\tilde{k}-1}]^T$, where $\tilde{k} = max(\tilde{k}_l, \tilde{k}_r)$. Using these definitions, we can write the matrix form of the evolution equation as,

$$V^{n+1} = \mathbf{A}(\tilde{k}_l, \tilde{k}_r)V^n \tag{2.20}$$

where the coefficient matrix is

$$\mathbf{A}(\tilde{k}_l, \tilde{k}_r) = \begin{bmatrix} \mathbf{A}_0 & \mathbf{A}_1 & \dots & \mathbf{A}_{\tilde{k}} & \mathbf{A}_{\tilde{k}+1} \\ \mathbf{I} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{I} & \mathbf{0} \end{bmatrix}. \tag{2.21}$$

While this equation is very general, we specialize this system to same delay on both sides ($\tilde{k} = \tilde{k}_l = \tilde{k}_r$) for all processors, which can be thought as a worst case scenario. Defining $r_d = \alpha \Delta t / \Delta x^2$

as the diffusive CFL, we can then write,

$$\mathbf{A}_0(\tilde{k}) = \begin{bmatrix} 1-2r_d & \mathcal{L}_1^0 r_d & 0 & \dots & 0 & \mathcal{L}_1^0 r_d \\ \mathcal{L}_1^0 r_d & 1-2r_d & \mathcal{L}_1^0 r_d & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathcal{L}_1^0 r_d & 0 & 0 & \dots & \mathcal{L}_1^0 r_d & 1-2r_d \end{bmatrix}, \quad \mathbf{A}_{\tilde{k}}(\tilde{k}) = \begin{bmatrix} 0 & \mathcal{J} & 0 & \dots & 0 & \mathcal{J} \\ \mathcal{J} & 0 & \mathcal{J} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathcal{J} & 0 & 0 & \dots & \mathcal{J} & 0 \end{bmatrix}$$

with $\mathcal{J}$ defined as,

$$\mathcal{J} = \mathcal{L}_1^m r_d(\tilde{k}+1) - \mathcal{L}_2^m r_d \tilde{k} \tag{2.22}$$

which is used to set the coefficient as $r_d(\tilde{k}+1)$ for $\mathbf{U}^{n-\tilde{k}}$ and $-r_d\tilde{k}$ for $\mathbf{U}^{n-\tilde{k}-1}$. For this we use $\mathcal{L}^m$, which is the Lagrange polynomial of order L,

$$\mathcal{L}_1^m(\tilde{k}) = \prod_{l\neq m}^{L} \frac{\tilde{k}-l}{m-l} \qquad \mathcal{L}_2^m(\tilde{k}) = \prod_{l\neq m}^{L} \frac{\tilde{k}+1-l}{m-l}. \tag{2.23}$$

By definition, $\mathcal{L}_1^m(\tilde{k})$ takes value 1 if $m = \tilde{k}$ and zero otherwise. Similarly $\mathcal{L}_2^m(\tilde{k})$ is 1 if $m = \tilde{k}+1$ and 0 for other values of m. The number of Lagrange polynomials is equal to the number of time levels in the AT scheme, which for the second-order scheme used here is equal to two. In the absence of delays we have, $\mathbf{A} = \mathbf{A}_0$, which is equivalent to the standard second-order finite difference system.

For stability, the spectral radius of $\mathbf{A}(\tilde{k})$ should be bounded by unity to ensure that the numerical perturbations do not grow unboundedly in time[20]. Because of the complexity of the system, the spectrum has to be computed numerically. Again as a worst case scenario [16], we assume a Dirac delta distribution of delays, such that, $\tilde{k} = L$ at all points. For a given L, we compute the maximum or critical r_d for which all the eigenvalues of the evolution matrix are less than unity. This is the largest value for which the numerical scheme is stable, and is denoted by $r_{d,m}(L)$. The results of this analysis are shown in Fig. 2.1. In the synchronous limit ($L = 0$), we obtain the well known stability limit for a second order central difference scheme in space with forward Euler in time, $r_{d,m}(0) = 0.5$ [20]. As we increase L, this stability limit decreases as can be seen from the

solid red circles in Fig. 2.1(a). Similar analysis was also done for the advection-diffusion equation which has both first and second derivatives and thus both convective ($r_c = c\Delta t/\Delta x$) and diffusive CFLs are used to determine stability. Here again we fix the delay $\tilde{k} = L$ and compute the stability limit in the r_c-r_d plane. The procedure was repeated for different values of L. The result is plotted in Fig. 2.1(b). For $L = 0$, we get the well known stability bound, $r_{c,m}(0) \leq 2(r_{d,m}(0))^2 \leq 1$ [20]. As L is increased we see that both $r_{c,m}(L)$ and $r_{d,m}(L)$ decrease resulting in a smaller stability region. Thus, for stability a time-step smaller than that for $L = 0$ is required whenever $L > 0$. We do point out that these stability bounds are based on worst case scenario assumptions and are thus strict. In more realistic scenarios (2.4.2), the effect of asynchrony on stability is relatively weaker.

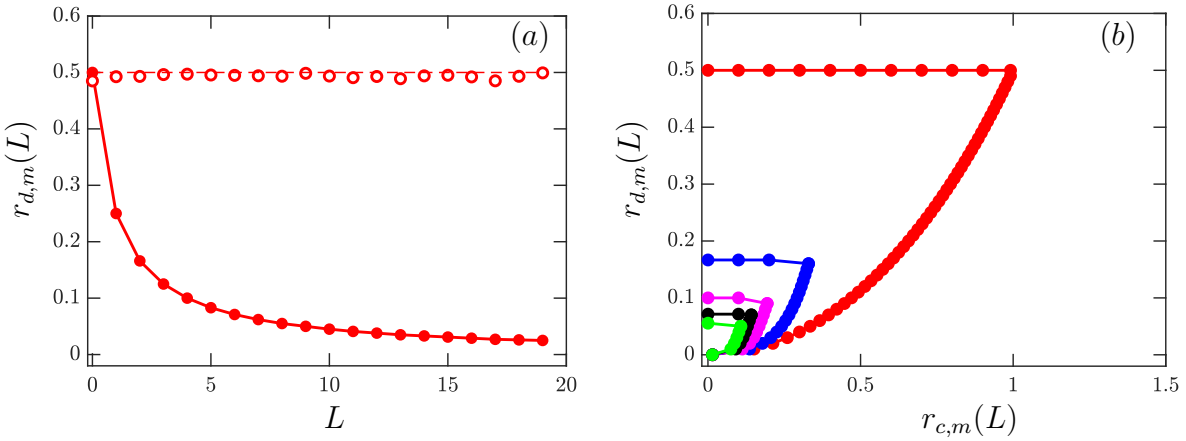

Figure 2.1: (a) Variation of stability limit r_d (solid) and $\widetilde{r_d} = (L+1)r_d$ (hollow) with L for diffusion equation. (b) Stability limit in r_c-r_d plane for advection-diffusion equation for $L = 0$ (red), $L = 2$ (blue), $L = 4$ (magenta), $L = 6$ (black) and $L = 8$ (green).

In order to characterize the reduction in stability limits, it is of interest to obtain the stability limit in an asynchronous simulations from the known stability limit of a synchronous implementation. This can be written as

$$r_{d,m}(L) = r_{d,m}(0)/f(L), \tag{2.24}$$

where the yet unknown function $f(L)$ characterizes the effect of delays. Clearly, $f(0) = 1$. Some

guidance on a plausible functional form for $f(L)$ can be obtained by a careful examination of Eq. (2.19) where we observe that, in the presence of delays, r_d at the boundary points always appears in conjunction with functions of delays that are also the coefficients of the AT scheme. In the present case, from Eq. (2.22) we have $r_d(\tilde{k}+1)$ and $-r_d\tilde{k}$ in the evolution matrix $\mathbf{A}(\tilde{k})$. Since both terms are linear in the delay, it is natural to expect that, for $\tilde{k} = L$, stability, and thus $f(L)$, would be a linear function of L. In fact, a best fit approximation for $r_{d,m}(L) = r_{d,m}(0)/f(L)$ does yield a linear relation $f(L) \approx L+1$. Both $r_{d,m}(0)/(L+1)$ (solid line) and $r_{d,m}(L)$ (solid circles) are plotted in Fig. 2.1(a) and are in excellent agreement with each other. Furthermore, we can re-arrange $r_{d,m}(L) = r_{d,m}(0)/f(L)$ to read as $r^a_{d,m} = r_{d,m}(L) \times f(L) = r_{d,m}(0)$. This implies that with a correct approximation for $f(L)$, we can express stability in terms of an *effective asynchronous* CFL ($r^a_{d,m}$), which is independent of delay L and essentially equal to the synchronous stability limit ($r_{d,m}(0)$). The numerical data do support this argument as can be seen from Fig. 2.1(a) where $r^a_{d,m}$ (hollow circles) are constant for all L and close to $r_{d,m}(0) = 0.5$ (dashed line).

We also computed the stability limit for the schemes used for the turbulence simulations in this work, namely, fourth-order AT schemes coupled with AB2 in time. This is shown in Fig. 2.2(a) for the diffusion equation. In this case, r_d appears multiplied by the coefficients in this fourth-order AT scheme (Appendix B) in the discrete equation which are seen to be quadratic in L. Then, based on the argument above, we expect $f(L)$ also to be quadratic in L. From Fig. 2.2(a) we can see that that $r_{d,m}(L)$ (solid circles) decreases with L and is in good agreement with $r_{d,m}(0)/f(0.74L^2 + 0.47L + 1)$ (solid line). Moreover, $r^a_{d,m}$ (hollow circles) is close to $r_{d,m}(0) \approx 0.18$ (dashed line) for all L. This again supports the proposed rescaling in Eq. (2.24).

One can understand this effect more intuitively as follows. When there is a delay at the PE boundaries, data from multiple delayed time levels is used at these points for computation of derivatives. As a result the effective time-step, as seen by the numerical scheme, increases. This effective time-step is essentially equivalent to $\Delta t_L = \Delta t \times f(L)$ and is apparent when $r^a_{d,m}$ is written as

$$r^a_{d,m} = r_{d,m}(L) \times f(L) = \frac{\alpha(\Delta t \times f(L)}{\Delta x^2} = \frac{\alpha(\Delta t_L)}{\Delta x^2}, \tag{2.25}$$

For fixed grid spacing Δx, this increase in time-step is compensated by a decrease in $r_{d,m}(L)$ to ensure stability. On the other hand, $r^a_{d,m}$ which is already expressed in terms of Δt_L, remains approximately constant with L and is equal to $r_{d,m}(0)$.

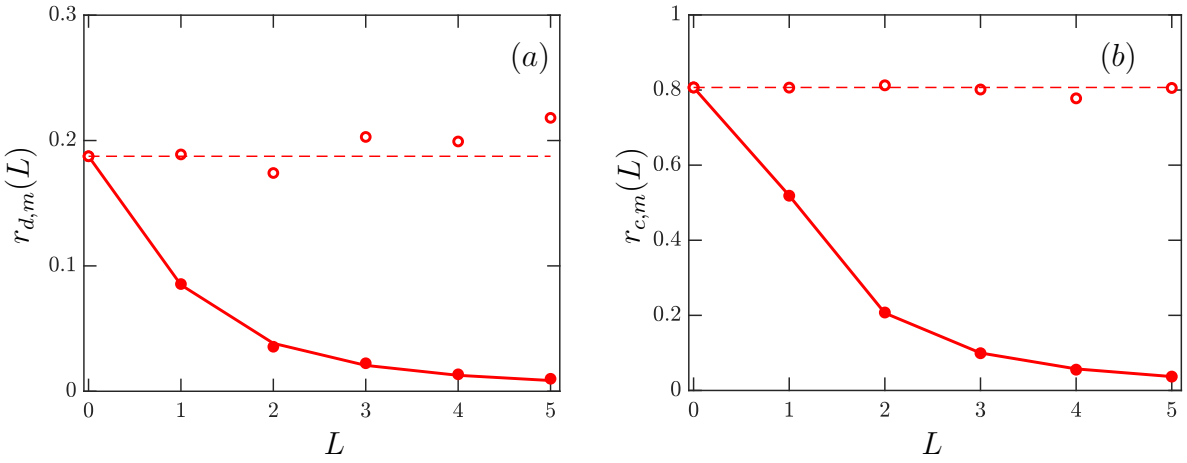

Figure 2.2: Variation of stability limit (a) r_d (solid) and $\widetilde{r_d} = (0.74L^2 + 0.47L + 1)r_d$ (hollow) with L for diffusion equation and (b) r_c (solid) and $\widetilde{r_c} = (0.90L^2 - 0.35L + 1)r_c$ (hollow) for NS equation, using fourth-order AT scheme in space and AB2 in time.

For a complex system of equations, such as the Navier-Stokes equations, an analytical stability analysis is difficult. However, stability limits can be computed numerically either by gradually increasing the CFL until the system becomes unstable or by using the bisection method. We obtained the stability limit for decaying turbulence at $Re_\lambda \approx 35$, by imposing a fixed delay L at all the six faces at every time step. Since both diffusive and convective terms are present in the NS equations, the time-step is determined by the smallest physical time scale, which for the simulations presented is always the latter. Thus, the stability limit is obtained in terms of a convective CFL (r_c) and is shown in Fig. 2.2(b) with $r_{c,m}(L)$ (solid circles) decreasing with L. As before, this effect is accurately captured by $r_{c,m}(L) = r_{c,m}(0)/f(L)$ (solid line), where $f(L) \approx 0.90L^2 - 0.35L + 1$. Here again, $r^a_{c,m} = r_{c,m}(L) \times f(L)$ (hollow circles) is seen to be a constant consistent with the synchronous limit $r_{c,m}(0) \approx 0.8$ for all L.

2.4.2 Standard Von-neumann analysis of AT schemes

Stability analysis in the frequency domain, more commonly known as the *Von Neumann analysis*, has been widely used for linear problems with constant coefficients [20]. While the standard von Neumann analysis is discussed in great detail in Chapter 3, in the current section we directly use this approach to assess stability of AT schemes. Since this method also requires all the points in the domain to use the same numerical scheme, we assume that each processor has only one grid point ($P = N$) and the AT scheme use same delay ($\tilde{k}$) on both sides for all PEs. To proceed further, consider a diffusion equation discretized using a second-order AT scheme in space and forward Euler in time,

$$u_i^{n+1} = u_i^n + \frac{\alpha \Delta t}{\Delta x^2} \left((\tilde{k}+1) u_{i-1}^{n-\tilde{k}} - \tilde{k} u_{i-1}^{n-\tilde{k}-1} - 2u_i^n + (\tilde{k}+1) u_{i+1}^{n-\tilde{k}} - \tilde{k} u_{i+1}^{n-\tilde{k}-1} \right). \tag{2.26}$$

where $\tilde{k}$ is the delay at both left and right boundary. Using a Fourier decomposition, $u_i^n = v^n e^{I i \phi}$ where $I = \sqrt{-1}$ and $\phi = \kappa \Delta x$, we can simplify Eq. (2.26) as

$$v^{n+1} = r_d \left(v^{n-L}(L+1) - v^{n-L-1} L \right) e^{-I\phi} + (1 - 2r_d) v^n + r_d \left(v^{n-L}(L+1) - v^{n-L-1} L \right) e^{I\phi}. \tag{2.27}$$

where $r_d = \alpha \Delta t / \Delta x^2$ is the diffusive CFL and delay is equal to the maximum allowed delay *ie* $\tilde{k} = L$. On taking the Z-transform of Eq. (2.27) with

$$v^n = v(z); \quad v^{n-L} = z^{-L} v(z) \tag{2.28}$$

we get a polynomial of order $L + 2$ in z which reads as,

$$z^{L+2} - z^{L+1}(1 - 2r_d) - r_d (e^{-I\phi} - e^{I\phi})(L+1) z - r_d (e^{-I\phi} - e^{I\phi}) L = 0 \tag{2.29}$$

For stability we require the amplitude all the harmonics to not grow in time. This is possible only if all the roots of Eq. (2.29) are within a unit disc in the complex Z-plane and the root $z = 1$ has

multiplicity one [20]. The roots of the above equation can be computed numerically for different L and these can be used to obtain the maximum r_d such that all roots lie within a unit disc. This is the largest limit for which the numerical scheme is stable and as before, we denote it as $r_{d,m}(L)$. Instead of computing the roots, the sufficient conditions that guarantee that all the roots lie within the disc of radius one can also be computed using the *Schur-Cohn* criteria [23]. As an example, we use *Schur-Cohn* criteria to get these conditions for $L = 1$. Since for $L = 1$, Eq. (2.29) reduces to a cubic polynomial, we get a total of three conditions,

$$4r_d^2\cos^2(\phi) - 1 < 0, \tag{2.30}$$

$$16r_d^4\cos^4(\phi) - 4r_d^2(4r_d(\text{rd}+1)+3)\cos^2(\phi) + 1 > 0, \tag{2.31}$$

$$\begin{aligned} &8r_d\sin^2\left(\frac{\phi}{2}\right)\left(-2r_d^2 + 2r_d(2r_d\cos(\phi) - r_d\cos(2\phi) + \cos(\phi)) + 1\right)^2 \\ &\left(3r_d^3\cos(3\phi) + 3\left(3r_d^2 - 1\right)r_d\cos(\phi) + (r_d - 1)\left(2r_d^2\cos(2\phi) + 2r_d^2 - 1\right)\right) > 0. \end{aligned} \tag{2.32}$$

These equations have only two parameters which can be varied to determine $r_{d,m}(L)$ that satisfies all the three conditions, for all values of $\phi \in [0, \pi]$. This gives us $r_{d,m}(1) \approx 0.25$ which is consistent with the stability limit obtained from the matrix stability analysis presented in section 3. For larger values of r_d, at least one of the above conditions is violated and thus instabilities can be triggered. We can see that this value is less than the traditional stability limit for a synchronous scheme $(r_{d,m}(0) = 0.5)$. It is worthwhile to note that similar constraints can also be written for $L = 0$ or the standard synchronous scheme which give us the expected limit of $r_{d,m}(0) = 0.5$. This idea can be extended to get the constraints for the stability for higher L as well. [23] discusses techniques to extend the *Schur-cohn* criteria to higher order polynomials efficiently.

The above analysis assumes a worst case scenario with two-sided delays at all points. A more practical scenario would be to have delays on either the left or the right PE boundary. Since the schemes are symmetric, the stability limit for an AT scheme with delays on left boundary, is also applicable to AT scheme with delays at the right boundary. Repeating the above procedure, but now considering delay $\tilde{k} = L$ on the left boundary and $\tilde{k} = 0$ on the right, gives us following three

conditions for all the roots of the z equation to lie with a unit disc,

$$r_d^2 - 1 < 0, \tag{2.33}$$

$$1 - r_d^2\left(3 + 4r_d(1 + r_d)\right) + r_d\left(2r_d^2 + 4r_d^3\right)cos(\phi) > 0, \tag{2.34}$$

$$\begin{aligned} &8r_d(-1 + r_d^2)\sin^2\left(\frac{\phi}{2}\right)\big(-1 + r_d + 4r_d^2 + r_d^3 - 4r_d^4 + \\ &r_d(-1 + 2r_d)(1 + 2r_d)^2 cos(\phi) + r_d^2\left(1 - 4r_d(1 + r_d)\right)cos(2\phi) + r_d^3 cos(3\phi)\big) > 0. \end{aligned} \tag{2.35}$$

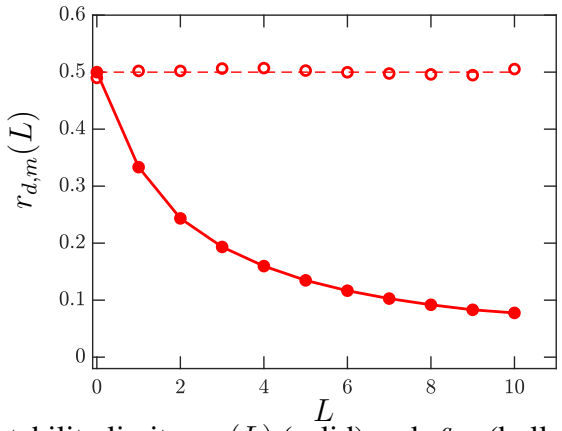

Figure 2.3: Variation of stability limit $r_{d,m}(L)$ (solid) and $r^a_{d,m}$ (hollow) with L for diffusion equation with delays on only one side.

These new conditions are satisfied for $r_{d,m}(1) \approx 0.33$ for all $\phi \in [0, \pi]$, which is lower than the previously computed bound of $r_{d,m}(1) \approx 0.25$. In Fig. 2.3 we show the $r_{d,m}(L)$ (solid circles) obtained for varying L for one-sided delays. Also plotted is $r_{d,m}(0)/f(L)$ (solid line) defined in Eq. (2.24), where $f(L) \approx 0.56L + 0.9$, and $r^a_{d,m} = r_{d,m}(0) \times f(L)$ in hollow circles with a dashed line corresponding to the synchronous limit $r_{d,m}(0) = 0.5$. We see that while $r_{d,m}(L)$ decreases as delay increases, the *effective asynchronous* CFL ($r^a_{d,m}$) is close to 0.5 for all L and supports the argument presented in the previous section.

2.5 A note on time-step

The CFL, relating time-step size to the grid spacing, can be used to determine the leading order error term of a fully discretized PDE in order to ensure that global order of accuracy is preserved (see section 2.3). Furthermore, the CFL is also used to determine the spatial and temporal resolution to ensure stability of the numerical simulation. The stability analysis presented here shows that the stability limit for the AT schemes decreases with L as $r_{d,m}(0)/f(L)$. This dependency can also be expressed using the *effective asynchronous* CFL ($r^a_{c,m}$ or $r^a_{d,m}$) which satisfies the same limit as the synchronous case ($r_{c,m}(0)$ or $r_{d,m}(0)$) and uses an effective time-step ($\Delta t_L = f(L) \times \Delta t$). Here $f(L)$, which is of the same order in L as the coefficients in the corresponding AT scheme, gives a quantitative measure of the effect of delays on the stability limit. For example, for a large value of $f(L)$, in order to keep $r^a_{c,m} = f(L) \times r_{c,m}(L)$ constant, $r_{c,m}(L)$ needs to be small. This implies that a small Δt is required for the asynchronous simulation to be stable, which in turn can increase the computational cost. However, we note that while simulations of turbulent flows at $r_c = 1$ are prevalent in literature, recent studies have shown that for adequate temporal resolution, a much smaller r_c should be used [24]. Thus, the CFL (or Δt) dictated by those resolution requirements, could be much smaller than the reduced stability limit discussed above.

In general, a fixed CFL simulations leads to additional overheads. For example, for a convective CFL (r_c), the time-step Δt is computed as,

$$\Delta t = \frac{r_c \Delta x}{u_{max}} \tag{2.36}$$

where u_{max} is the global maximum velocity. Since this maximum is computed across all PEs, it requires a collective blocking communication call at every time step and leads to more synchronization overheads. In order to avoid this, instead of a CFL condition, one can use a fixed Δt [18, 25]. This is the approach we adopt here. For consistency, synchronous simulations are also done at the same fixed Δt.

2.6 Conclusions

In this chapter we looked at some of the numerical properties of asynchrony-tolerant (AT) schemes. Specifically, we showed that these schemes can be used for simulations of conservation laws and they incur only a small residual despite delays at processor boundaries. This residual is an order higher than the order of AT schemes and thus decreases with increasing resolution. We also verified that the multi-stage Runge-Kutta schemes when used for temporal discretization with AT schemes, preserve the global order of accuracy. While we presented an algorithm for efficient implementation of RK with AT schemes, in a three-dimensionsional domain the temporal integration with RK has a high computation, communication and synchronization intensity. This necessitates the need of alternate temporal integration methods including Adam-Bashforth for asynchronous simulations in higher dimensions.

We performed stability analysis of AT schemes in both physical and spectral space using standard methodologies under the symplifying assumptions that only one grid-point is present in every processing element and the same AT scheme (or same delay) is used throughout the domain. While very strict stability bounds are obtained when delays are considered at both sides, these bounds for delays on only one side are more relaxed. The stability limit for AT schemes is also expressed in terms of the known stability limit for the corressponding synchronous scheme and a function that characterizes the effect of delays. This function can also be used to write the effective time-step that is seen at the processor boundaries due to delayed data.

For the stability analysis of AT schemes using standard approaches, we implicitly assume that delay is time-invariant. However, in an actual simulation this delay is essentially a random variable that depends upon machine characteristics and varies with time step. While considering a random delay at every time step renders the stability problem intractable, we attempt to relax the assumption of fixed delay at all times in Chapter 3. In particular, we look at a specific implementation of AT schemes where the delay and consequently the scheme changes periodically. We are also interested in the spectral accuracy of AT schemes and this is done in the next Chapter using the proposed generalized von Neumann analysis.

3. GENERALIZED VON-NEUMANN ANALYSIS*

3.1 Introduction

A large number of complex physical processes in nature and engineering are governed by non-linear ordinary differential equations (ODEs) and partial differential equations (PDEs). Due to the general lack of analytical solutions for non-linear equations, the fundamental understanding of such processes relies heavily on numerical simulations. These simulations require approximation of derivatives by appropriate numerical methods. Consequently, truncation and round-off errors are introduced that can affect the accuracy of the numerical solutions. High-order numerical methods are preferred as they incur smaller truncation errors for a finely discretized domain. High-order schemes also have the ability to resolve a wide range of scales or wavenumbers, thereby making them suitable for simulations of multi-scale phenomena. However, a numerical scheme must be stable in order to be usable in practice. The basic idea of stability is to ensure that the numerical scheme does not allow the error or perturbations (such as round-off) to grow unbounded in time. According to the definition of stability by Lax and Richtmyer [26], all components of the initial solution should be uniformly bounded. Mathematically, we can express stability by writing out the fully discretized PDE in matrix form

$$U^{(n)} = \mathbf{C}U^{(n-1)}, \tag{3.1}$$

where $\mathbf{C}$ is the evolution matrix and $U^{(n)}$ is a vector comprising the values of the solution $u_i^{(n)}$ at all mesh points i at some time level n. Similar matrix-evolution equation was also constructed in the previous chapter (see Eq. (2.20)) to assess the stability of asynchrony-tolerant (AT) schemes.

*Parts of this chapter are reprinted from Journal of Computational Physics, Volume 424, Komal Kumari and Diego A. Donzis, "A generalized von Neumann analysis for multi-level schemes: Stability and spectral accuracy", Pages 109868, 1 January 2021, with permission from Elsevier

If the initial solution $U^{(0)}$ is known and bounded, then one can write

$$U^{(n)} = (\mathbf{C})^n U^{(0)} \tag{3.2}$$

where $(\mathbf{C})^n$ is the operator raised to the power n. Then $U^{(n)}$ is bounded when the operator $(\mathbf{C})^n$ remains uniformly bounded for all n, Δt and Δx, or mathematically,

$$\|(\mathbf{C})^n\| < K \text{ for } \begin{cases} 0 < \Delta t < \tau \\ 0 \leq n\Delta t \leq T \end{cases} \tag{3.3}$$

for fixed values of τ and T, where K is independent of n, τ and T and $\|.\|$ is some matrix norm [20].

When periodic boundary conditions are used, stability can also be studied using the von Neumann analysis which provides a wealth of additional information on the performance of numerical schemes. The von Neumann analysis is one of the most widely used and well-known techniques for the analysis of numerical methods and can be easily found in multitude of introductory and advanced CFD books [20, 27, 28, 29, 30] and the references therein. For this, we express the solution as a Fourier series and compute the one-step amplification factor, $G(\eta) = \hat{u}^{(n)}/\hat{u}^{(n-1)}$, where $\eta = k\Delta x$ is the normalized wavenumber and $\hat{u}^{(n)}$ is the amplitude of a Fourier mode at wavenumber η and time level n. Stability in the von Neumann sense then requires the amplitude of the amplification factor to be bounded by unity [20], *i.e*

$$|G(\eta)| \leq 1, \quad \forall \eta \in [0, \pi]. \tag{3.4}$$

However, the standard von Neumann analysis relies on an implicit, yet fundamental, assumption that the amplification factor $G(\eta)$ is independent of time level n. A simple test to assess the validity of this assumption is to compute the amplification at different n for a model equation whose von Neumann amplification factor is well known. An example is shown in Fig. 3.1 where

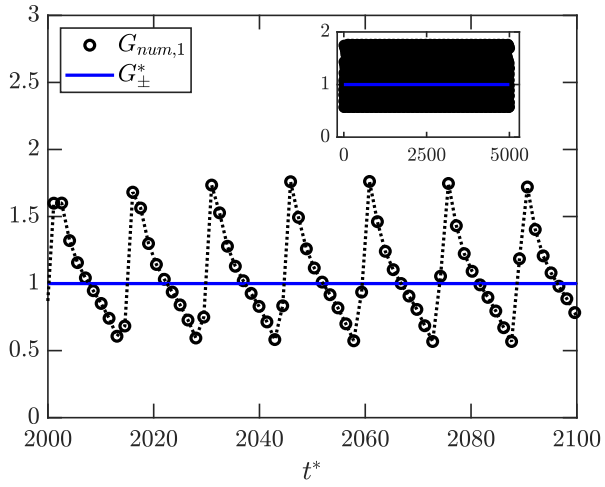

Figure 3.1: Time evolution of the amplification factor for the advection equation using the leapfrog (LF) scheme for wavenumber $k = 32$ or $\eta \approx 1.57$. Comparison is shown between the numerical amplification (black circles) and the analytical amplification obtained from the standard von Neumann analysis (blue horizontal line). Inset is same evolution for all times.

the amplification factor ($G^{(n)}_{num,1} = \hat{u}^{(n)}/\hat{u}^{(n-1)}$) has been obtained from a numerical simulation of the linear advection equation using a three-level leapfrog scheme for an arbitrary wavenumber $k = 32$ or $\eta \approx 1.57$ at every time step (or n). While the exact discretization and Fourier analysis of the equation are given in later sections, here it suffices to note that the numerical amplification (black circles) varies with time (or n) and exhibits a local pattern with no clear tendency to converge to a fixed value. This behavior persists at both long at short times, as is evident from the inset in Fig. 3.1. Also shown in the figure is the magnitude of the leapfrog scheme obtained using the standard von Neumann analysis, $G^*_{\pm} = -jr_c \sin\eta \pm \sqrt{1 - r_c^2 \sin^2\eta}$ (blue line), where $r_c = \beta_1 \Delta t/dx$ is the convective CFL and β_1 is the propagation speed. Two important conclusions emerge from this simple example. First, the instantaneous values of numerical amplification (black circles) are not equal to the amplification factor obtained from the von Neumann analysis (blue line). Second, the amplification factor takes values larger than unity even though the scheme is stable at the CFL used in this simulation. Thus, it seems necessary to reevaluate the standard von Neumann analysis. This will also have implications on the stability definition in Eq. (3.4) which would need to be modified in order to account for variation of the amplification factor with n.

In this chapter we identify and address these concerns [31]. In particular, we show the limitations of the standard analysis using both theoretical and numerical results for two multi-level schemes. As we discuss in great detail below, the disagreement between the amplification observed in numerical simulations and that computed using the standard von Neumann analysis, which is clearly apparent in Fig. 3.1, is because the standard analysis fails to account for the variation of amplification with time. To overcome this limitation, we propose a modification to the standard von Neumann analysis, in which we retain the time dependent behavior of amplification factor in what we call here the generalized von Neumann analysis. This generalized analysis allows us to compute the correct amplification factor for multi-level numerical methods. This amplification factor is then used to compute the spectral accuracy of fully-discrete systems which also exposes the degrading effect of temporal schemes on the spectral accuracy of the spatial scheme. We then propose a generalized definition of stability based on the correct amplification factor. Finally, we use this generalized von Neumann analysis to assess the stability and spectral accuracy of the more challenging asynchrony-tolerant (AT) schemes whose coefficients change periodically in time.

3.2 Preliminaries: modified wavenumber analysis

The $d\text{-}th$ spatial derivative of a function $u(x,t)$ at a point i or x_i and time level n using a finite difference scheme can be written as

$$\left.\frac{\partial^d u(x,t)}{\partial x^d}\right|_i^{(n)} \approx \frac{1}{(\Delta x)^d}\sum_{m=-M}^{M} a_{d,m} u_{i+m}^{(n)} + \mathcal{O}(\Delta x^{p+1}). \tag{3.5}$$

where M points are used on both sides of i, $p+1$ is the order of accuracy and $a_{d,m}$ are the coefficients of the finite difference scheme. For a given scheme, it is then common to assess the ability of the scheme to capture adequately different wavenumbers. This can be done by performing the so-called von Neumann analysis which is based on considering a single Fourier mode for the solution,

$$u(x,t) = \hat{u}(t)e^{jkx}, \tag{3.6}$$

where $\hat{u}$ is the Fourier coefficient at wavenumber k, and $j = \sqrt{-1}$. The exact $d\text{-}th$ derivative is then given by

$$\frac{\partial^d u}{\partial x^d} = \hat{u}(t)(jk)^d e^{jkx} = (jk)^d u(x,t). \tag{3.7}$$

The discrete approximation Eq. (3.5) can now be written, using the following notation

$$\begin{aligned} u_i^{(n)} &:= u(x_i, t_n) = \hat{u}^{(n)} e^{jkx_i}, \\ u_{i+m}^{(n)} &:= u(x_i + m\Delta x, t_n) = \hat{u}^{(n)} e^{jkx_i} e^{jkm\Delta x} = u_i^{(n)} e^{jkm\Delta x} \end{aligned} \tag{3.8}$$

as

$$\left.\frac{\partial^d u(x,t)}{\partial x^d}\right|_i^{(n)} = \left(\frac{1}{\Delta x^d}\sum_m a_{d,m} e^{jkm\Delta x}\right) u_i^{(n)}. \tag{3.9}$$

Comparing Eq. (3.7) and Eq. (3.9) and defining $\eta := k\Delta x$ as the normalized wavenumber, we can write

$$\left.\frac{\partial^d u(x,t)}{\partial x^d}\right|_i^{(n)} = \left(\frac{1}{\Delta x^d}\sum_m a_{d,m} e^{jm\eta}\right) u_i^{(n)} = \frac{1}{\Delta x^d}(j\overline{\eta})^d u_i^{(n)} \tag{3.10}$$

where $\overline{\eta}$ is the so-called modified wavenumber for the spatial operator. Spectral resolution is then assessed by studying how accurately $\overline{\eta}$ approximates η. For example, for a second-derivative computed using a second-order central difference scheme, we have $\overline{\eta}^2 = 2(1 - \cos\eta)$ which is shown as a dashed green line in Fig. 3.2. While this is close to the exact wavenumber η^2 (magenta line) for low wavenumbers, it departs significantly at high wavenumbers. Therefore, the scheme will not capture accurately high wavenumber fluctuations and is thus considered inadequate for multi-scale phenomena such as turbulence.

We note, however, that the modified wavenumber analysis is based exclusively on the ability of a particular spatial discretization to compute gradients accurately, and involves no consideration of the temporal discretization needed to perform an actual numerical simulation. It is somewhat surprising that the effect of temporal discretization on the modified wavenumber for the fully discretized PDE, has received little attention, even though it follows naturally from the results of the

von Neumann analysis. We show through numerical experiments that the temporal schemes in fact degrade the spectral accuracy of spatial schemes, especially at high CFL values. This is presented in the next section.

For completion, we introduce here the amplification factor as the ratio between Fourier coefficients at different time levels. In particular we define

$$\widetilde{G}_m^{(n)} \coloneqq \frac{\hat{u}^{(n)}}{\hat{u}^{(n-m)}} \tag{3.11}$$

as the amplification factor across m steps. For clarity in the exposition, we will use a tilde over the amplification factor for a particular numerical scheme as in Eq. (3.11) while the exact amplification factor resulting from the exact solution of a particular PDE will be denoted by $G_m^{(n)}$. We finally mention that it also possible to obtain the modified wavenumber in terms of the amplification factor, as we show below, for specific PDEs.

3.3 Von-Neumann analysis

In order to introduce the standard von Neumann approach, we present the analysis for standard two-level and three-level schemes for linear equations. In each case we start by computing the amplification factor and the modified wavenumber from it, followed by discussion with an emphasis on the limits of the standard approach.

3.3.1 Two-level scheme

Consider a 1D diffusion equation,

$$\frac{\partial u}{\partial t} = \beta_2 \frac{\partial^2 u}{\partial x^2}, \tag{3.12}$$

where β_2 is the diffusion coefficient. If Eq. (3.12) is discretized using a forward Euler scheme in time and second-order central difference in space, the fully-discrete equation can be written as,

$$u_i^{(n)} = u_i^{(n-1)} + r_d \left(u_{i-1}^{(n-1)} - 2u_i^{(n-1)} + u_{i+1}^{(n-1)} \right), \tag{3.13}$$

where $r_d = \beta_2 \Delta t / \Delta x^2$ is the diffusive CFL. The stability of Eq. (3.13) can then be computed from the amplification factor defined in Eq. (3.11) yielding

$$\widetilde{G}_1^{(n)} = 1 + 2r_d \left(\cos \eta - 1\right), \tag{3.14}$$

Using the stability definition given by Eq. (3.4) we can easily compute the well known stability bound $r_d \leq 0.5$.

It is worth noting that, besides stability, Eq. (3.14) can also be used to compute spectral accuracy of the fully discrete system. For Eq. (3.12), the exact amplification can be obtained by substituting Eq. (3.8) in Eq. (3.12),

$$G_1^{(n)} = e^{-r_d \eta^2}. \tag{3.15}$$

This expression can be solved for the wavenumber as $\eta^2 = \frac{\log(|G_1^{(n)}|)}{-r_d}$. Similarly, we can also obtain the numerical or modified wavenumber for the fully discrete system from Eq. (3.14) as

$$\widetilde{\eta}^2(r_d) = \frac{\log(|\widetilde{G_1}^{(n)}|)}{-r_d}. \tag{3.16}$$

This modified wavenumber $\widetilde{\eta}$ gives spectral accuracy of the fully discrete system and can be used to study the effect of the temporal scheme on the spectral accuracy of the spatial scheme (given by $\overline{\eta}$). Unlike the modified wavenumber for the spatial scheme $\overline{\eta}$, the modified wavenumber for the fully discrete system varies with the diffusive CFL r_d. In Fig. 3.2, we show the exact wavenumber (η^2) for the second derivative in magenta, the modified wavenumber ($\bar{\eta}^2$) for the spatial scheme used in Eq. (3.12) in green and the modified wavenumber ($\widetilde{\eta}^2(r_d)$) obtained from Eq. (3.16) for two different CFL values in black. We see that at low CFL ($r_d = 0.1$), the dotted-black line is slightly closer to the magenta line (η^2) than the dashed-blue line ($\bar{\eta}$), implying that the temporal operator seems to improve the spectral accuracy of the spatial operator at this CFL value. On the other hand, at higher CFL ($r_d = 0.45$), the modified wavenumber for the fully discrete system shown in dashed-dotted black line shows large variations from both the exact wavenumber and

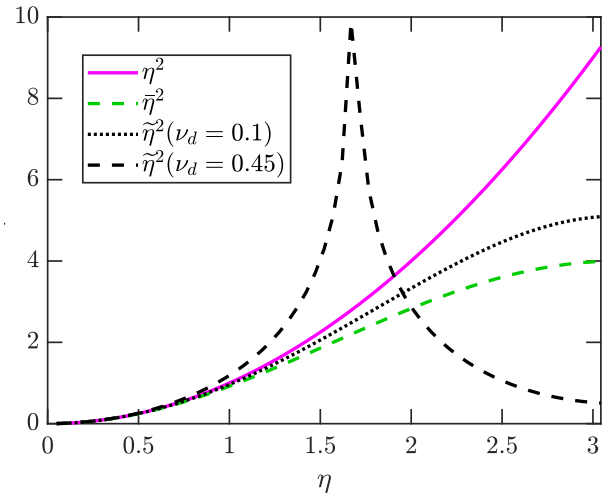

Figure 3.2: Spectral accuracy of the second-order central difference scheme for second-derivative in space and fully-discrete diffusion equation. Different lines correspond to exact modified wavenumber for second derivative (magenta), modified wavenumber for spatial scheme (green), modified wavenumber for fully discrete system (black) at diffusive CFL $r_d = 0.10$ (dotted) and $r_d = 0.45$ (dashed).

the modified wavenumber for the spatial scheme. Thus, depending on the CFL, the temporal scheme can drastically change the spectral accuracy of the spatial scheme. This understanding is crucial because in simulations high-order spatial schemes are preferred for their good spectral accuracy. Without careful consideration of the effect of temporal scheme, especially at large CFLs, the spectral accuracy of the resulting fully discrete system could be severely deteriorated.

3.3.2 Three-level schemes

As examples for standard three-level schemes, we will consider the leapfrog (LF) scheme for the advection equation

$$\frac{\partial u}{\partial t} + \beta_1 \frac{\partial u}{\partial x} = 0 \tag{3.17}$$

where $\beta_1 > 0$ is the propagation speed and the Du-Fort and Frankel (DFF) scheme [32] for the diffusion equation Eq. (3.12).

3.3.2.1 Leapfrog scheme

We can write the discretized form of Eq. (3.17) using the LF scheme as,

$$u_i^{(n)} = u_i^{(n-2)} - r_c \left(u_{i+1}^{(n-1)} - u_{i-1}^{(n-1)} \right), \tag{3.18}$$

where $r_c = \beta_1 \Delta t / \Delta x$ is the convective CFL. Now, for the von Neumann analysis we first substitute Eq. (3.8) in Eq. (3.18) to yield

$$\hat{u}^{(n)} = \hat{u}^{(n-2)} \underbrace{-j \left(2 r_c \sin \eta \right)}_{:= f_1(r_c, \eta)} \hat{u}^{(n-1)}. \tag{3.19}$$

This can be rearranged to compute the amplification across one time step as

$$\frac{\hat{u}^{(n)}}{\hat{u}^{(n-1)}} = \frac{\hat{u}^{(n-2)}}{\hat{u}^{(n-1)}} + f_1(r_c, \eta), \tag{3.20}$$

or across two time steps as

$$\frac{\hat{u}^{(n)}}{\hat{u}^{(n-2)}} = 1 + f_1(r_c, \eta) \frac{\hat{u}^{(n-1)}}{\hat{u}^{(n-2)}}. \tag{3.21}$$

Since the standard von Neumann analysis assumes $\widetilde{G}_1 = \hat{u}^{(n)} / \hat{u}^{(n-1)} = \hat{u}^{(n-1)} / \hat{u}^{(n-2)}$, which gives $\widetilde{G}_2 = \hat{u}^{(n)} / \hat{u}^{(n-2)} = \widetilde{G}_1^2$, both Eq. (3.20) and Eq. (3.21) reduce to the following quadratic equation

$$\widetilde{G}_1^2 - f_1(r_c, \eta) \widetilde{G}_1 - 1 = 0, \tag{3.22}$$

independent of the time level n. The two roots of Eq. (3.22),

$$G_{\pm}^{*} = -j r_c \sin \eta \pm \sqrt{1 - r_c^2 \sin^2 \eta}. \tag{3.23}$$

give two amplification factors for the leapfrog scheme, even though the exact amplification factor has a single value, $G = e^{-j r_c \eta}$, for all η. Of these two roots, one is called the physical or principal root while the other is known as the spurious root generated by the scheme [20]. For stability, the

amplitude of both $G_{\pm}^{*}$ should be bounded by unity. Imposing this constraint in Eq. (3.23) results in the stability limit $r_c \leq 1$. While this analysis is well established in the literature, in order to ascertain that it is indeed the amplification factor for the LF scheme, we performed simulations and compared the numerical amplification factor with $G_{\pm}^{*}$.

For the simulations, we iterated Eq. (3.18) using a broadband initial condition with a maximum wavenumber of k_{max}. Since LF is a three-level scheme, in order to start the computation, two time levels, $u^{(1)}$ and $u^{(0)}$ need to be initialized. Here $u^{(1)}$ can be computed exactly (if the exact solution is known) or numerically using two-level schemes such as Runge-Kutta from the initial condition $u^{(0)}$. For simplicity here, we used former approach using the exact solution, that is, $\widetilde{G}_1^{(1)} = G$. The different amplification factors for $r_c = 0.25$ and $r_c = 0.95$ are shown in Fig. 3.3 at a normalized time $t^* = k_{max}\beta_1 t \approx 77$. We see that the one-step (black circles) and the square-root of two-step (red triangles) numerical amplification factors are different from each other and neither of these agree with the standard von Neumann amplification (blue line). Also, these numerical amplifications vary with η and attain values much larger than unity for certain wavenumbers even though $G_{\pm}^{*}$ is unity for all η. The time t^* here is chosen arbitrarily and similar results are observed at other times as well.

We can also look at the spectral accuracy of the LF scheme by computing the modified wavenumber. Following the exact amplification factor, which gives the wavenumber as $\eta = -arg[G]/r_c$, we can obtain the modified wavenumber $\eta_{\pm}^{*} = -arg[G_{\pm}^{*}]/r_c$ from the standard amplification factor. This is compared with the one-step (black circles) and two-step (red triangles) modified wavenumber computed from the numerical simulations in Fig. 3.4. Similar to the amplification factor, we see that the wavenumber obtained from the simulations (circles and triangles) overshoots or undershoots $\eta_{\pm}^{*}$ (blue lines). It is interesting to note that while at $r_c = 0.25$, the numerical wavenumber has a tendency to oscillate only about η_{+}^{*} (solid blue), the behavior changes with r_c. Specifically, for $r_c = 0.95$, the numerical wavenumber is close to η_{+}^{*} for low η and transitions to η_{-}^{*} (dashed blue) for larger η. This suggests that the classification of the roots as physical and spurious as established in the literature, may need to be revisited.

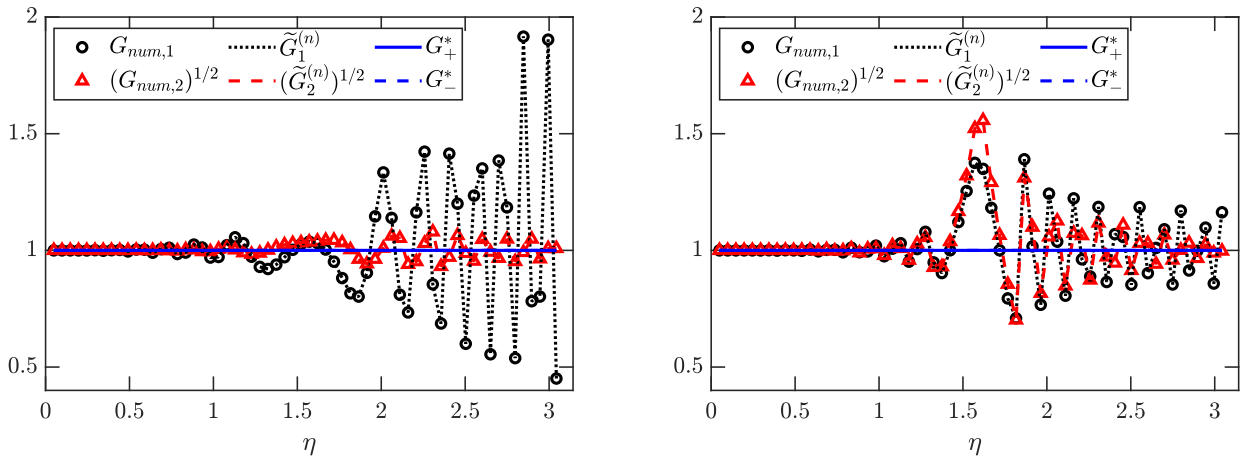

Figure 3.3: Amplification factor for the advection equation using the leapfrog (LF) schemse. Comparison of numerical and analytical amplification factor for $r_c = 0.25$ (left) and $r_c = 0.95$ (right) at $t^* = n\Delta t \beta_1 k_{max} \approx 77$. Here G_{num} is the numerical amplification factor, $\widetilde{G}^{(n)}$ is the amplification factor computed using the generalized von Neumann analysis, and $G_\pm^*$ are the standard amplification factors. Subscript $1, 2$ specifies number of time steps across which the corresponding quantity is computed.

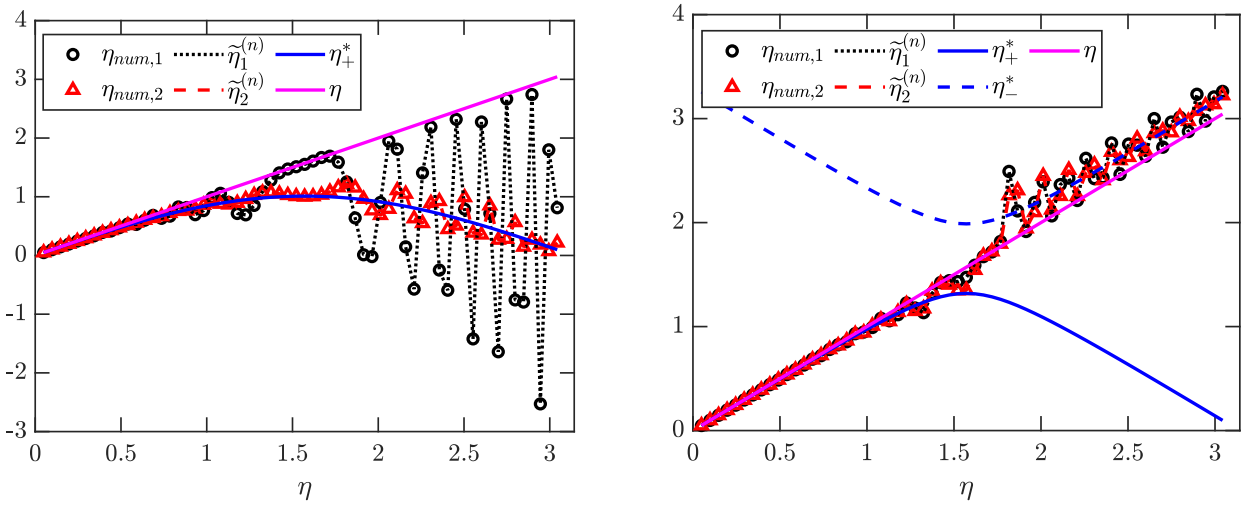

Figure 3.4: Spectral accuracy for the advection equation using the leapfrog (LF) scheme. Comparison of numerical and analytical modified wavenumber for $r_c = 0.25$ (left) and $r_c = 0.95$ (right) at $t^* = n\Delta t \beta_1 k_{max} \approx 77$. η_-^* is not shown in (a) as it takes large values. Here η is the exact wavenumber, η_{num} is the numerical modified wavenumber, $\widetilde{\eta}^{(n)}$ is the modified wavenumber computed using the generalized von Neumann analysis, and $\eta_\pm^*$ are the modified number computed from standard von Neumann analysis. Subscript 1 and 2 specifies the number of time steps across which the corresponding quantity is computed.

3.3.2.2 *Du-Fort and Frankel scheme*

The discrete form of the diffusion equation (Eq. (3.12)) using the DFF scheme is given by

$$u_i^{(n)} = \underbrace{\frac{1-2r_d}{1+2r_d}}_{:=f_2(r_d)} u_i^{(n-2)} + \frac{2r_d}{1+2r_d}\left(u_{i+1}^{(n-1)} + u_{i-1}^{(n-1)}\right), \tag{3.24}$$

which reduces to

$$\hat{u}^{(n)} = f_2(r_d)\hat{u}^{(n-2)} + \underbrace{\frac{4r_d\cos\eta}{1+2r_d}}_{:=f_3(r_d,\eta)}\hat{u}^{(n-1)}, \tag{3.25}$$

in wavenumber space, with r_d being the diffusive CFL. Similar to the LF scheme, we can obtain the one-step ($\widetilde{G}_1$) or two-step amplification factor ($\widetilde{G}_2 = \widetilde{G}_1^2$) from Eq. (3.25), both of which lead to a quadratic equation independent of time level n and given by,

$$\widetilde{G}_1^2 - f_3(r_d,\eta)\widetilde{G}_1 - f_2(r_d) = 0, \tag{3.26}$$

The two roots of Eq. (3.26),

$$G_\pm^* = \frac{2r_d\cos(\eta) \pm \sqrt{1-4r_d^2\sin^2(\eta)}}{1+2r_d}, \tag{3.27}$$

then give us the two standard amplification factors for the DFF scheme. Since the amplitude of $G_\pm^*$ is bounded by unity for all r_d, the scheme is said to be unconditionally stable. Simply said, the maximum amplification at any time level is at most unity for all wavenumbers. However, if we iterate Eq. (3.24) for a broadband initial condition and compare the numerical amplification so obtained with that given by $G_\pm^*$, as shown in Fig. 3.5, we note important discrepancies. First, the one-step (black circles) and the square-root of two-step (red triangles) numerical amplification factors are not equal to just one of the two roots $G_\pm^*$ for all η. Instead, the numerical amplification remains closer to G_+^* (solid blue line) for lower η and to G_-^* (dashed blue line) at higher η, with a different transition region for different r_d. Second, for $r_d = 0.95$, the numerical amplification for

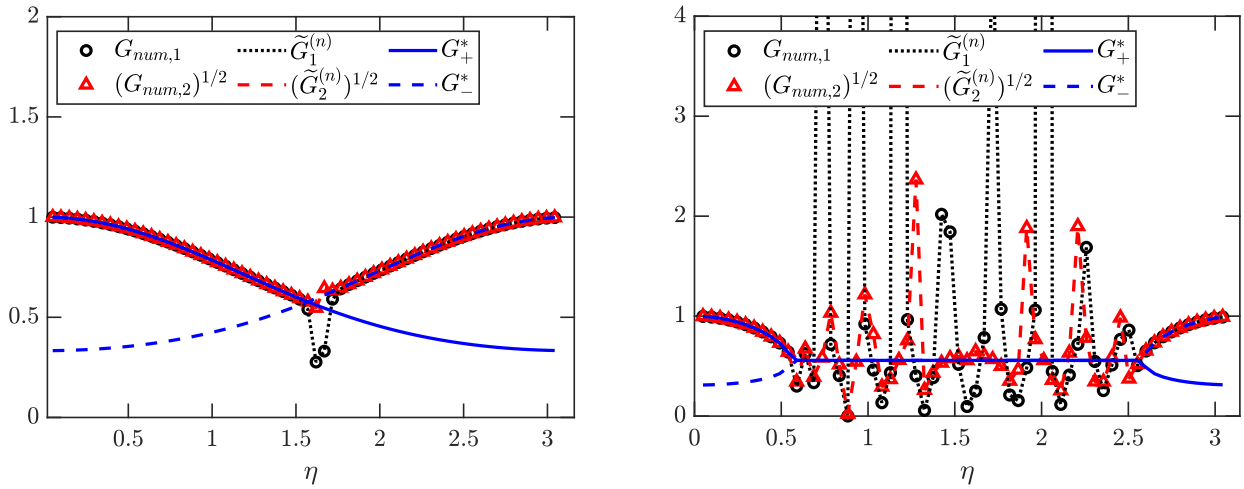

Figure 3.5: Amplification factor for the diffusion equation using the Du-fort and Frankel (DFF) scheme. Comparison of numerical and analytical amplification factor for $r_d = 0.25$ (left) and $r_d = 0.95$ (right) at $t^* = n\Delta t\beta_2 k_{max}^2 \approx 88$. Here G_{num} is the numerical amplification factor, $\widetilde{G}^{(n)}$ is the amplification factors computed using generalized von Neumann analysis and $G_\pm^*$ is the standard amplification factor. Subscript $1, 2$ specifies number of time steps across which the corresponding quantity is computed.

intermediate wavenumbers exhibits large fluctuations with magnitude much larger than unity and clearly not equal to the standard amplification factor (blue lines). Similar disagreements are also seen between the numerical modified wavenumber and modified wavenumber computed from $G_\pm^*$ (not shown here).

The clear inability of the standard von Neumann analysis to accurately compute the amplification factor (and modified wavenumber) for both LF and DFF schemes is a direct consequence of the invalid assumption that the amplification factor is independent of time level. We overcome this limitation by proposing a generalized von Neumann analysis, which we describe next.

3.4 Generalized von Neumann analysis

The generalized von Neumann analysis is also based on a Fourier decomposition of the solution but unlike the standard analysis, the amplification factor retains the dependency on time level. As a result, the amplification factor at any time instant is computed using a recursive relation, instead of the characteristic polynomial found in the standard analysis. Thus, the essence of the generalized

Von-Neumann approach is to simply relax the assumption that the amplification factor in multi-level schemes is same at all times. Additionally, unlike the standard von Neumann method that gives multiple amplification factors corresponding to each root of the characteristic polynomial, the generalized method computes a unique amplification factor at every time level, which as we show below agrees well with the numerical simulations. Moreover, for certain multi-level schemes we can express the amplification factor obtained from the generalized von Neumann analysis as a simple continued fraction. Doing so allows us to obtain the exact conditions under which the standard analysis, in fact, can be used to compute the amplification factor. Finally, we re-define the stability condition in order to account for the variation of amplification factor with time.

We now introduce and illustrate the concept of generalized von Neumann analysis using two well-known multi-level schemes. We note that for two level schemes, the generalized von Neumann analysis reduces to the standard von Neumann analysis.

3.4.1 Leapfrog scheme

The generalized von Neumann analysis for the LF scheme follows directly from Eq. (3.20) and Eq. (3.21). Instead of assuming that the amplification is independent of time level n, we write the general one-step and two-step amplifications as functions of n as

$$\widetilde{G}_1^{(n)} := \frac{\hat{u}^{(n)}}{\hat{u}^{(n-1)}} = f_1(r_c, \eta) + \frac{1}{\widetilde{G}_1^{(n-1)}} \tag{3.28}$$

$$\widetilde{G}_2^{(n)} := \frac{\hat{u}^{(n)}}{\hat{u}^{(n-2)}} = 1 + f_1(r_c, \eta)\widetilde{G}_1^{(n-1)}. \tag{3.29}$$

While Eq. (3.28) or Eq. (3.29) give the corresponding amplification at any time level n, we are also interested in computing conditions under which the generalized amplification reduces to the amplification obtained from the standard von Neumann analysis. For this, we note that Eq. (3.28)

is a recursive function that can also be expressed as a continued fraction,

$$\widetilde{G}_1^{(n)} = f_1(r_c,\eta) + \cfrac{1}{f_1(r_c,\eta) + \cfrac{1}{f_1(r_c,\eta) + \cfrac{1}{\ddots\ f_1(r_c,\eta) + \frac{1}{\widetilde{G}_1^{(1)}}}}} \tag{3.30}$$

when $\widetilde{G}_1^{(n-1)}$ is written in terms of $\widetilde{G}_1^{(n-2)}$ which itself is written in terms of $\widetilde{G}_1^{(n-3)}$ and so on. We can write Eq. (3.30) compactly as

$$\widetilde{G}_1^{(n)} = b_1 + \frac{1|}{|b_2} + \frac{1|}{|b_3} + \cdots + \frac{1|}{|b_n} = \frac{A_n}{B_n}, \text{ where } \begin{cases} b_i = f_1(r_c,\eta), i \neq n \\ b_n = \widetilde{G}_1^{(1)} \end{cases} \tag{3.31}$$

Here A_n and B_n are the so-called n^{th} numerator and denominator, respectively, and $\widetilde{G}_1^{(n)}$ is called the n^{th} approximant of the continued fraction [33]. This continued fraction is said to converge if the sequence of its approximants $\{A_n/B_n\}$ converges, that is, $\lim_{n\to\infty} A_n/B_n$ exists and is finite. Clearly, if the continued fraction converges, then it must converge to one of the roots of the quadratic equation Eq. (3.22) that follows directly from Eq. (3.28) or Eq. (3.30). However, there is no reason *a priori* to believe that the continued fraction converges. If it does not converge, then Eq. (3.22) loses its meaning in the traditional sense, as there is no n-independent $\widetilde{G}_1$ which reduces Eq. (3.28) to Eq. (3.22). Yet, as we show here, Eq. (3.22) can provide critical information about whether such a $\widetilde{G}_1$ exists or not. The condition for existence of $\lim_{n\to\infty} A_n/B_n$ (*i.e.* convergence) and its value is given by the following lemma [33].

Lemma 1. *Let G_+^* and G_-^* be the roots of the quadratic equation Eq. (3.22), then*

1. *If $|G_+^*| \neq |G_-^*|$ or $G_+^* = G_-^*$ the continued fraction converges to the root with larger absolute value*

2. *If* $|G_+^*| = |G_-^*|$ *and* $G_+^* \neq G_-^*$*, the continued fraction does not converge.*

Proof. See Appendix 3.A. □

Now, we can readily identify different cases when looking at the roots of Eq. (3.22) given by Eq. (3.23). For a convective CFL r_c that satisfies $r_c < 1$, we have $|G_+^*| = |G_-^*|$ for all η. Thus, according to Lemma $1(ii)$, approximants of the continued fraction do not converge to either of these roots. In other words, the amplification factor at any time level n, which is equal to the n^{th} approximant obtained from Eq. (3.31), is different from $G_\pm^*$, the amplification factors obtained with the standard von Neumann analysis. If, on the other hand, $r_c \geq 1$, then $|G_+^*| \neq |G_-^*|$ for $\eta \in [0, \alpha) \cup (\pi - \alpha, \pi]$, $|G_+^*| = |G_-^*|$ for $\eta \in (\alpha, \pi - \alpha)$ and $G_+^* = G_-^*$ for $\eta = \{\alpha, \pi - \alpha\}$, where $\alpha = \sin^{-1}(1/r_c)$. In this case, as a consequence of Lemma 1, the amplification factor is equal to the root in Eq. (3.23) with larger absolute value for $\eta \in [0, \alpha) \cup (\pi - \alpha, \pi]$ and equal to the approximant of the continued fraction for $\eta = (\alpha, \pi - \alpha)$. When $\eta = \{\alpha, \pi - \alpha\}$, since both roots are equal, the continued fraction converges to $G_+^* = G_-^*$. The conclusion from this analysis, thus, is that the standard von Neumann analysis is applicable only for a certain range of η depending upon r_c. Eq. (3.31), o the other hand, always gives the correct value of amplification, for all r_c, η and n.

We refer back to Fig. 3.3, where we see that the one-step numerical amplification (black circles) agrees exactly with $\widetilde{G}_1^{(n)}$ (dotted-black line) computed using Eq. (3.28), as expected. Similar agreement is seen between the square-root of the two-step numerical amplification (red triangles) and $(\widetilde{G}_2^{(n)})^{1/2}$ (dashed-red line) computed from Eq. (3.29). The amplification factors given by Eq. (3.28) and Eq. (3.29) can also be used to assess the actual spectral accuracy of the fully discrete system by computing the time-dependent modified wavenumber

$$\widetilde{\eta}_m^{(n)} = -arg[\widetilde{G}_m^{(n)}]/m\, r_c, \tag{3.32}$$

where m is the number of steps across which the amplification is computed. This is shown in Fig. 3.4. Similar to the amplification factor, we see that both the one-step (black circles) and

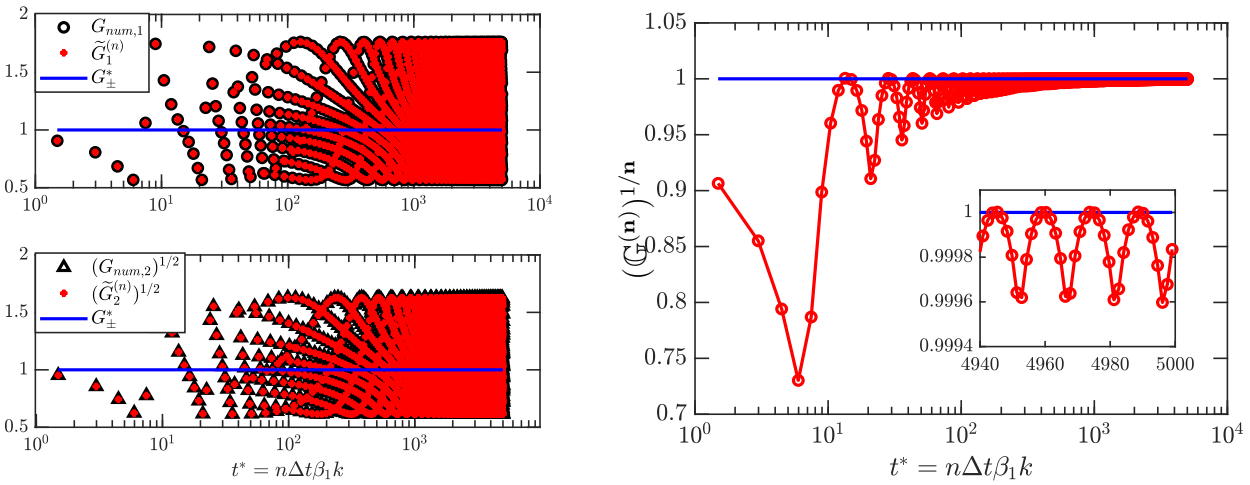

Figure 3.6: Time evolution of one-step (top-left) and two-step (bottom-left) amplification factor and stability parameter $(\mathbb{G}^{(\mathbf{n})})^{1/\mathbf{n}}$ (right) for advection equation using leapfrog (LF) scheme for wavenumber $k = 32$ or $\eta \approx 1.57$. Inset in plot on the right is a zoom-in on short time interval. Here G_{num} is the numerical amplification factor, $\widetilde{G}^{(n)}$ is the amplification factor computed using generalized von Neumann analysis and $G_\pm^*$ is the standard amplification factor. Subscript $1, 2$ specifies number of time steps across which the corresponding quantity is computed.

two-step (red triangles) numerical modified wavenumber obtained from simulations agree exactly with $\widetilde{\eta}_1^{(n)}$ (black line) and $\widetilde{\eta}_2^{(n)}$ (red line), respectively. Since the modified wavenumber is directly computed from the amplification factor, it also exhibits fluctuations depending upon r_c and η if the latter does not converge. Moreover, when the modified wavenumber varies with time, the accuracy of the corresponding numerical gradients and thus, the accuracy of simulations will also vary with time.

It is now clear, from both theoretical and numerical results, that the amplification factor depends, in general, upon time level n. To illustrate this further, one can also look at the time evolution of the amplification factor for an arbitrary wavenumber ($k = 32$ or $\eta \approx 1.57$), as shown in in Fig. 3.6(a) for $r_c = 0.95$ (or Fig. 3.1). We immediately see that the instantaneous numerical amplification factor (black circles and triangles) varies with time (t^* or n) and is clearly different from the amplification factor $G_\pm^*$ (blue) obtained from the standard analysis, which gives a constant value at all times. However, this oscillatory instantaneous behavior of numerical amplification is accurately predicted at all times by $\widetilde{G}_1^{(n)}$ or $\widetilde{G}_2^{(n)}$ (red) obtained from the generalized von Neumann

analysis. Furthermore, we observe that the instantaneous amplitude, albeit bounded, is frequently greater than unity. Since the scheme is stable in a broader sense, one needs to reevaluate the classical stability definition in Eq. (3.4). It seems natural that a stability notion in a broader sense could be based on the product of amplification factors to be bounded instead of the values at individual steps. Clearly, the latter is a sufficient condition for the former. We, thus, propose the following stability criterion

$$\mathbb{G}^{(\mathbf{n})} := \left(\prod_{p=1}^{n} |\widetilde{G}^{(p)}| \right) < K \tag{3.33}$$

where K is finite and independent of n. This is closer to the stability definition given by Eq. (3.3) and requires all modes to be uniformly bounded. This allows the amplitude of different modes to grow instantaneously, though not in a long-term unbounded manner. We do note that while a close expression for Eq. (3.33) may be possible in some cases, in others it may be difficult. Thus, Eq. (3.33) may need to be studied for each scheme. We also note that for a scheme with $\widetilde{G}^{(n)} = \widetilde{G}^{(n-1)} = \cdots = G$, Eq. (3.33) will reduce to the stability defined in the standard von Neumann sense. In other words, the standard von Neumann analysis is a special case of the generalized analysis presented here.

In Fig. 3.6(b) we show the time evolution of $(\mathbb{G}^{(\mathbf{n})})^{\mathbf{1/n}}$ for $r_c = 0.95$ and $k = 32$ or $\eta \approx 1.57$. After large fluctuations in the initial steps, wherein the initial condition effects are important, we see that the amplitude oscillates but remains bounded and exhibits a tendency to converge towards unity. In general, for $r_c \leq 1$, Eq. (3.33) is satisfied for all wavenumbers but for $r_c > 1$, K in Eq. (3.33) increases with n for at least some wavenumbers. Since there exists a finite K, independent of n for all η only for $r_c \leq 1$, the LF scheme is stable for this range of CFL. It is important to note here that even though the generalized von Neumann analysis gives the same stability bound as the standard von Neumann analysis, the latter uses an amplification factor which is never realizable in actual simulations.

3.4.2 Du Fort and Frankel scheme

The general one-step and two-step amplification factors at any time level n follow from Eq. (3.25) and are given by,

$$\widetilde{G}_1^{(n)} = f_3(r_d, \eta) + \frac{f_2(r_d)}{\widetilde{G}_1^{(n-1)}} \tag{3.34}$$

$$\widetilde{G}_2^{(n)} = f_3(r_d, \eta)\widetilde{G}_1^{(n-1)} + f_2(r_d), \tag{3.35}$$

respectively. Once again, in order to determine whether or not the amplification computed from the generalized von Neumann analysis is equivalent to that known from the standard analysis, we restate the recursive relation in Eq. (3.34) as the following continued fraction

$$\widetilde{G}_1^{(n)} = b_1 + \frac{a_2|}{|b_2} + \frac{a_3|}{|b_3} + \cdots + \frac{a_n|}{|b_n} = \frac{A_n}{B_n}, \text{ where } \begin{cases} b_i = f_3(r_d, \eta), i \neq n \\ b_n = \widetilde{G}_1^{(1)} \\ a_i = f_2(r_d). \end{cases} \tag{3.36}$$

This fraction converges (if the limit exists) to one of the roots Eq. (3.26) that follows from Eq. (3.34) or Eq. (3.36) if $\widetilde{G}_1$ is independent of n. The condition for the existence of a limit of Eq. (3.36) and its value is determined by the following Lemma.

Lemma 2. *Let G_+^* and G_-^* be the roots of the quadratic equation Eq. (3.26) with real coefficients and discriminant $D = f_3(r_d, \eta)^2 + 4f_2(r_d)$, then*

1. *If $D > 0$, the continued fraction converges to the root with larger absolute value.*
2. *If $D = 0$, the continued fraction converges to the single root with multiplicity two.*
3. *If $D < 0$, the continued fraction does not converge.*

Proof. See Appendix 3.A. □

For a diffusive CFL $r_d < 0.5$, we have $D > 0$ for all η and thus the continued fraction converges, by Lemma 2, to the root $G^*_\pm$ of Eq. (3.26), listed in Eq. (3.27), with larger absolute value. This can also be seen from simulation results shown in Fig. 3.5(a). We observe that both the one-step (black circles) and the square-root of the two-step (red triangles) numerical amplification factors agree exactly with the amplification computed using Eq. (3.34) (dotted black line) and Eq. (3.35) (dashed red line), respectively, and exhibit a tendency to converge to the branch of $G^*_\pm$ (blue line) with large magnitude. We do however see some discrepancies at intermediate values of η. This difference is observed because the dissipative nature of the equation leads to a solution which decays very rapidly to round-off levels, and the simulation time is thus necessarily short. For short times (small n), the effect of initial condition or $\widetilde{G}^1_1$ is still prominent and thus, we do not see the numerical amplification converging exactly to standard amplification $G^*_\pm$ for all wavenumbers. However, we can also obtain the long-term behavior of the amplification factor by iterating equation Eq. (3.34) or Eq. (3.36) to obtain $\widetilde{G}^{(n)}_1$ and Eq. (3.35) to obtain $\widetilde{G}^{(n)}_2$, for a sufficiently large n. This is shown in Fig. 3.7(a) where we observe that both $\widetilde{G}^{(n)}_1$ (dotted black) and $\widetilde{G}^{(n)}_2$Eq. (3.35) (dashed red) converge to the branch of $G^*_\pm$ (blue) with higher magnitude. This clearly implies that while the amplification factor obtained from the generalized von Neumann analysis agrees well with the numerical amplification at any arbitrary time, the standard amplification factor only captures the long term behavior for $r_d < 0.5$.

Now for $r_d \geq 0.5$, we have $D > 0$ for $\eta \in [0, \alpha) \cup (\pi - \alpha, \pi]$, $D < 0$ for $\eta \in (\alpha, \pi - \alpha)$ and $D = 0$ for $\eta = \{\alpha, \pi - \alpha\}$, where $\alpha = \sin^{-1}(1/2r_d)$. Thus, as a consequence of Lemma 2, the amplification factor converges to the root $G^*_\pm$ with larger magnitude for $\eta \in [0, \alpha) \cup (\pi - \alpha, \pi]$, converges to $G^*_+ = G^*_-$ for $\eta = \{\alpha, \pi - \alpha\}$, and oscillates without converging to either of $G^*_\pm$ for $\eta \in (\alpha, \pi - \alpha)$. This is evident from both Fig. 3.5(b) and Fig. 3.7(b), where both amplification factors obtained from Eq. (3.34) or Eq. (3.36) (black) and Eq. (3.35) (red) converge to the larger of the two roots $G^*_\pm$ at low and high wavenumbers that lie in the interval $\eta \in [0, \alpha) \cup (\pi - \alpha, \pi]$. For the intermediate wavenumbers $\eta \in (\alpha, \pi - \alpha)$, the amplification exhibits an oscillatory behavior. Moreover, for these wavenumbers, where Eq. (3.36) does not converge to either of

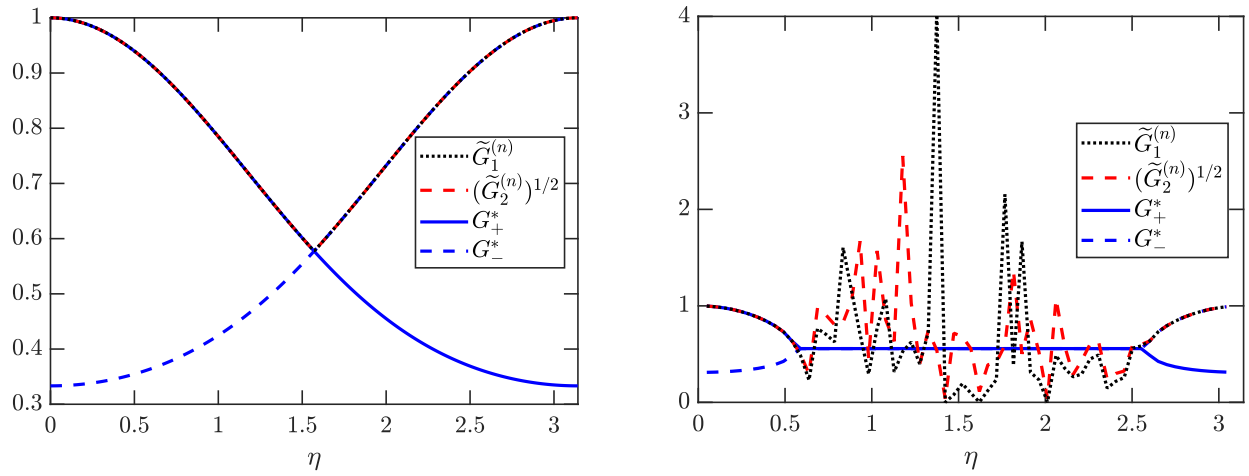

Figure 3.7: Amplification factor for the diffusion equation using the Du-fort and Frankel (DFF) scheme. Comparison of amplification factor obtained from the generalized and standard von Neumann analyses for $r_d = 0.25$ (left) and $r_d = 0.95$ (right) at long time. Here $\widetilde{G}^{(n)}$ is the amplification factor computed using generalized von Neumann analysis and $G_\pm^*$ is the standard amplification factor. Subscripts 1 and 2 denote number of time steps across which the corresponding quantity is computed.

$G_\pm^*$, the erratic instantaneous amplification is not necessarily bounded by unity, unlike the constant standard amplification, $G_\pm^*$. Thus, as in the case of LF scheme, the standard von Neumann analysis of DFF scheme gives correct measure of amplification only for a range of wavenumber and CFL. The generalized von Neumann analysis proposed here, on the other hand, reproduces the numerical data for any n, η, and r_d. Here also the stability of the DFF scheme can be determined from Eq. (3.33). For instance, the stability parameter defined in Eq. (3.33) for an arbitrary wavenumber $k = 10$ or $\eta \approx 0.49$ for $r_d = \{0.95, 2\}$, remains bounded for both r_d values at all times, as shown in Fig. 3.8. It is interesting to note here that for $r_d = 0.95$, this specific η belongs in the interval $[0, \alpha)$ and thus, $\widetilde{G}_1^{(n)}$ and consequently $(\mathbb{G}^{(\mathbf{n})})^{1/\mathbf{n}}$ (dotted-black), converges to G_+ (blue), with small differences only at small n. However, for $r_d = 2$, this specific η belongs to a different interval, that is $(\alpha, \pi - \alpha)$, and therefore, $(\mathbb{G}^{(\mathbf{n})})^{1/\mathbf{n}}$ approaches $|G_\pm|$ with persistent oscillations that are bounded.

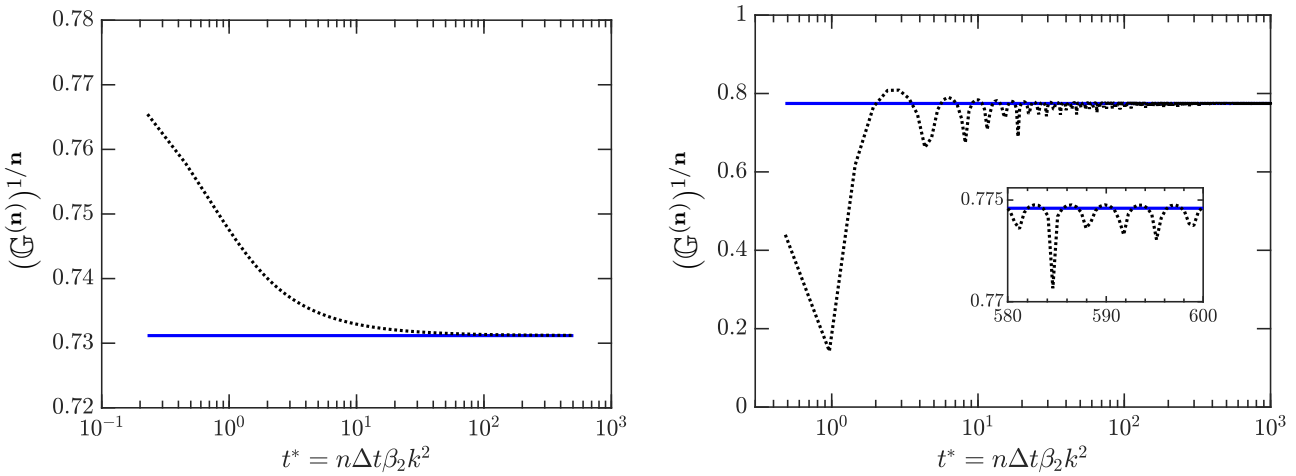

Figure 3.8: Time evolution of stability parameter $(\mathbb{G}^{(\mathbf{n})})^{1/\mathbf{n}}$ for the diffusion equation using the Du-fort and Frankel (DFF) scheme for $k = 10$ or $\eta \approx 0.49$ for $r_d = 0.95$ (left) and $r_d = 2$ (right). Blue line corresponds to $|G_+|$ (left) and to $|G_\pm|$ (right). Inset (right) is a zoom-in on short time interval.

3.5 Beyond the standard von Neumann analysis: asynchrony-tolerant schemes

In the previous section we discussed the breakdown of standard von Neumann analysis for simple synchronous schemes. We overcame these limitations with our proposed generalized von Neumann analysis. We use this generalized analysis to assess the stability and spectral accuracy of multi-level asynchronous schemes for which the standard von Neumann analysis is not applicable, even in principle. This is the case of so-called asynchrony tolerant (AT) schemes [16]. These AT schemes, as mentioned in Chapter 1, have been derived to improve the parallel performance of large-scale simulations where communications between processors becomes exceedingly expensive [9]. To do so, the algorithm does not enforce synchronization across processors to complete communications; instead processors continue computations uninterrupted with data available locally. This leads to a random delay at each processor boundary at every time step and thus the number of states grow exponentially rendering the stability analysis intractable. Thus, we limit our analysis to a simple, and yet, a practical implementaion of AT schemes that is more general than the stability analyisis perfomed in Chapter 2.

In particular, here we focus here on the Communication Avoiding Algorithm (CAA) for AT

schemes introduced in Chapter 5, where processors communicate only periodically. As a result, the specific numerical scheme used for derivatives also varies periodically making the standard von Neumann analysis inapplicable. Since the generalized von Neumann approach presented in previous sections does not require the amplification factor to be same at every step, it can be easily used when the scheme changes in time. Here again we assume that the same AT schemes are used at all grid points, a practice that has been used before as a worst case scenario for stability and accuracy [9, 16, 19]. For simplicity in the exposition, we also assume that the delays are present only on one-side as one would expect in an actual simulation.

The important parameter in the CAA is the rate of communication T, which bounds the maximum allowed delay L, as $L = T - 1$. The instantaneous delay then changes periodically from 0 (no delay) to L. For simplicity, we consider $L = 1$ ($T = 2$) which corresponds to AT schemes with delay of one and the standard synchronous schemes used in alternate steps. For this illustration we will consider the advection-diffusion equation,

$$\frac{\partial u}{\partial t} + \beta_1 \frac{\partial u}{\partial x} = \beta_2 \frac{\partial^2 u}{\partial x^2}, \tag{3.37}$$

where β_1 is the propagation speed and β_2 is the diffusion coefficient, discretized using second-order AT schemes [16] for both spatial derivatives,

$$\left.\frac{\partial u}{\partial x}\right|_i^{(n)} = \frac{-u_{i-1}^{(n)} + (\ell+1)u_{i+1}^{(n-\ell)} - \ell u_{i+1}^{(n-\ell-1)}}{2\Delta x}, \tag{3.38}$$

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_i^{(n)} = \frac{u_{i-1}^{(n)} - 2u_i^{(n)} + (\ell+1)u_{i+1}^{(n-\ell)} - \ell u_{i+1}^{(n-\ell-1)}}{\Delta x^2}\,. \tag{3.39}$$

Here ℓ is the delay seen at the right processor boundary which, with $L = 1$, is given by the sequence $\ell = \{0, 1, 0, 1, 0, 1, \dots\}$ in time. Both Eq. (3.38) and Eq. (3.39) reduce to a standard finite difference scheme when $\ell = 0$. We use a standard forward Euler method in time, leading to the following fully discrete system

$$
\begin{aligned}
u_i^{(n+1)} =& u_i^{(n)} - \frac{r_c}{2}\left(-u_{i-1}^{(n)} + (\ell+1)u_{i+1}^{(n-\ell)} - \ell u_{i+1}^{(n-\ell-1)}\right) \\
&+ r_d\left(u_{i-1}^{(n)} - 2u_i^{(n)} + (\ell+1)u_{i+1}^{(n-\ell)} - \ell u_{i+1}^{(n-\ell-1)}\right),
\end{aligned}
\tag{3.40}
$$

with r_c and r_d being the convective and diffusive CFL, respectively. which are used to compute $\widetilde{G}_1^{(1)}$, the first CAA step is synchronous with $\ell = 0$ and therefore, we get

$$
\widetilde{G}_1^{(2)} := \frac{\hat{u}^{(2)}}{\hat{u}^{(1)}} = 1 - 4r_d \sin^2(\eta/2) - jr_c \sin\eta = \widetilde{G}_{1,s}, \tag{3.41}
$$

where $\widetilde{G}_{1,s}$ is the well-known synchronous amplification factor [20]. In the next CAA step, delay is non-zero and thus the fully discrete system is given by Eq. (3.40) with $\ell = 1$. Substitution of Eq. (3.6) in Eq. (3.40) then yields,

$$
\hat{u}^{(3)} = \underbrace{\left(1 - 2r_d + (r_d + r_c/2)e^{-j\eta}\right)}_{:=f_4(r_c,r_d,\eta)}\hat{u}^{(2)} + \underbrace{(r_d - r_c/2)(\ell+1)e^{j\eta}}_{:=f_5(r_c,r_d,\ell,\eta)}\hat{u}^{(1)} + \underbrace{(r_c/2 - r_d)\ell e^{j\eta}}_{:=f_6(r_c,r_d,\ell,\eta)}\hat{u}^{(0)}, \tag{3.42}
$$

or simply

$$
\hat{u}^{(3)} = \underbrace{\left(f_4(r_c,r_d,\eta) + \frac{1}{\widetilde{G}_{1,s}}\left(f_5(r_c,r_d,\ell,\eta) + \frac{f_6(r_c,r_d,\ell,\eta)}{\widetilde{G}_1^{(1)}}\right)\right)}_{:=G_1^{(3)}}\hat{u}^{(2)}, \tag{3.43}
$$

if $\hat{u}^{(1)}$ is rewritten in terms of $\hat{u}^{(2)}$ using Eq. (3.41). In the next step we have $\ell = 0$ which gives a synchronous amplification factor followed by an asynchronous amplification factor in the following step when $\ell = 1$ and so on. Because of this periodicity in delay and consequently the schemes, the general one-step amplification for any time level $n \geq 2$ has T distinct functional forms given by,

$$
\widetilde{G}_1^{(n)} = \begin{cases} f_4(r_c,r_d,\eta) + \frac{1}{\widetilde{G}_{1,s}}\left(f_5(r_c,r_d,\ell,\eta) + \frac{f_6(r_c,r_d,\ell,\eta)}{\widetilde{G}_1^{(n-2)}}\right), & \text{if } n \text{ is odd } \textit{i.e. } \ell = 1 \\ \widetilde{G}_{1,s}\ , & \text{if } n \text{ is even } \textit{i.e. } \ell = 0 \end{cases} \tag{3.44}
$$

It is clear from Eq. (3.44) that the one-step amplification, and the corresponding modified wavenumbers, will alternate between synchronous and asynchronous values. This prohibits the use of the standard von Neumann stability constraint given by Eq. (3.4) to compute stability regions.

We now express the asynchronous amplification for all odd $n \geq 3$ in Eq. (3.44) as a continued fraction of the form,

$$\widetilde{G}_1^{(n)} = b_1 + \frac{a_2|}{|b_2} + \frac{a_3|}{|b_3} + \cdots + \frac{a_n|}{|b_n} = \frac{A_n}{B_n}, \text{ where } \begin{cases} b_i = f_4(r_c, r_d, \eta) + \frac{f_5(r_c, r_d, 1, \eta)}{\widetilde{G}_{1,s}}, i \neq n \\ b_n = \widetilde{G}_1^{(1)} \\ a_i = \frac{f_5(r_c, r_d, 1, \eta)}{\widetilde{G}_{1,s}}. \end{cases} \tag{3.45}$$

The conditions of convergence and the limit of this continued fraction follows from Lemma 1 with $G_{\pm}^*$ being the roots of the quadratic equation

$$\widetilde{G}^2 - \left(f_4(r_c, r_d, \eta) + \frac{f_5(r_c, r_d, 1, \eta)}{\widetilde{G}_{1,s}} \right) \widetilde{G} - \frac{f_6(r_c, r_d, 1, \eta)}{\widetilde{G}_{1,s}} = 0. \tag{3.46}$$

On computing the roots of Eq. (3.46) we found that $|G_-^*| \neq |G_+^*|$ for $r_c, r_d \leq 2$ and $\eta \in [0, \pi]$. Thus, the continued fraction in Eq. (3.36) converges to the root of Eq. (3.46) with larger absolute value within this range of parameters. Since, both the amplification factors in Eq. (3.44) converge to two unique functions, their product converges to a single function. This product is the two-step amplification given by

$$\widetilde{G}_T^{(n)} = \widetilde{G}_2^{(n)} = f_4(r_c, r_d, \eta)\widetilde{G}_{1,s} + f_5(r_c, r_d, 1, \eta) + \frac{f_6(r_c, r_d, 1, \eta)}{\widetilde{G}_1^{(n-3+\ell)}}, \tag{3.47}$$

or simply the product of $\widetilde{G}_{1,s}$ and the n^{th} approximant of Eq. (3.45). While the two-step amplification, $\widetilde{G}_2^{(n)}$, is expected to show variations at small n due to initial conditions, it eventually attains the same value irrespective of n and can therefore be used to determine the spectral accuracy of CAA.

The stability region in the r_c-r_d plane for second-order CAA can be obtained using the new

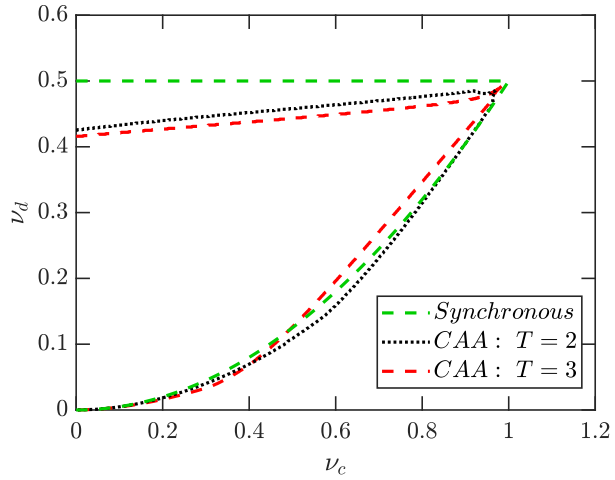

Figure 3.9: Stability region of the fully discrete advection-diffusion equation using asynchrony-tolerant (AT) schemes with a communication-avoiding algorithm (CAA) with periods $T = 2$ and 3 in the r_c-r_d plane. Also shown in a dashed-green line is the stability region for the synchronous scheme.

definition of stability in Eq. (3.33). Due to the difficulties in obtaining an analytical closed result, this is accomplished numerically by iterating Eq. (3.40), computing $\mathbb{G}^{(\mathbf{n})}$ and check whether the stability constraint Eq. (3.33) is satisfied or not. This is shown in Fig. 3.9, along with the stability limit for the standard synchronous scheme in dashed green. We see that the stability region for both $T = 2$ (black) and $T = 3$ (red) are close to that for the synchronous case, showing that the effect of periodic delays on the stability of AT schemes is minimal. Within this stability region, the instantaneous amplification, corresponding to the steps with $\ell > 0$, can take values larger than unity, similar to the instantaneous amplification being greater than unity for stable synchronous multi-level schemes discussed in the previous section. We also note that if instead, standard von Neumann analysis is used to compute the stability limits, then it would require bounding individual amplification factors obtained for each delay value up to L. This analysis for fixed delay at all times was recently done in [19] where it was shown that the stability region shrinks drastically as L is increased. However, by allowing for temporal variation of amplification and not requiring the amplification to be bounded by unity at each time level, we obtain a bigger stability region in r_c-r_d plane.

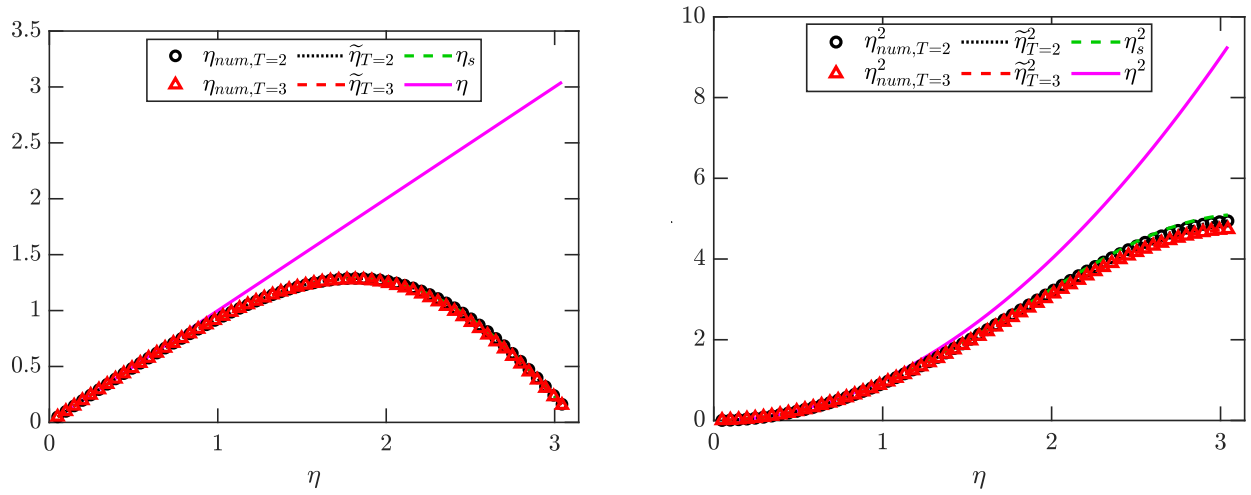

Figure 3.10: Spectral accuracy for the fully discrete advection-diffusion equation using asynchrony-tolerant (AT) schemes with a communication-avoiding algorithm (CAA) with periods $T = 2$ and 3, and $r_d = r_c = 0.1$. Comparison of analytical and numerical modified wavenumbers for first derivative (left) and second derivative (right). Here $\widetilde{\eta}_T$ and $\widetilde{\eta}^2_T$ are given by Eq. (3.32) and Eq. (3.48), respectively and the subscript s denotes the modified wavenumber for the synchronous scheme.

As in the synchronous case, the spectral accuracy of CAA can be assessed using the amplification factor given in Eq. (3.47) by computing the modified wavenumber $\widetilde{\eta}_T$ for the first derivative from Eq. (3.32) and the modified wavenumber $\widetilde{\eta}^2_T$ for the second derivative from

$$\widetilde{\eta}^2_T = \frac{\log(|\widetilde{G}_T|)}{-Tr_d}, \tag{3.48}$$

where the superscript (n) for the time level is omitted for simplicity. We compute $\widetilde{\eta}_T$ and $\widetilde{\eta}^2_T$ for $(r_c, r_d) = (0.1, 0.1)$ and $(0.2, 0.5)$. In Fig. 3.10, where both CFLs are small, we see that the modified wavenumbers for first and second derivatives in CAA are close to the corresponding modified wavenumbers for the synchronous scheme (dashed green). However, for larger CFL as shown in Fig. 3.11, the modified wavenumber exhibits a tendency to move further away from both synchronous (dashed green) and the exact solution (magenta) as L is increased from one to two, especially for large η. We also note that the spectral accuracy for the second derivative is deteriorated at larger CFL for synchronous as well as AT schemes, with the modified wavenumber

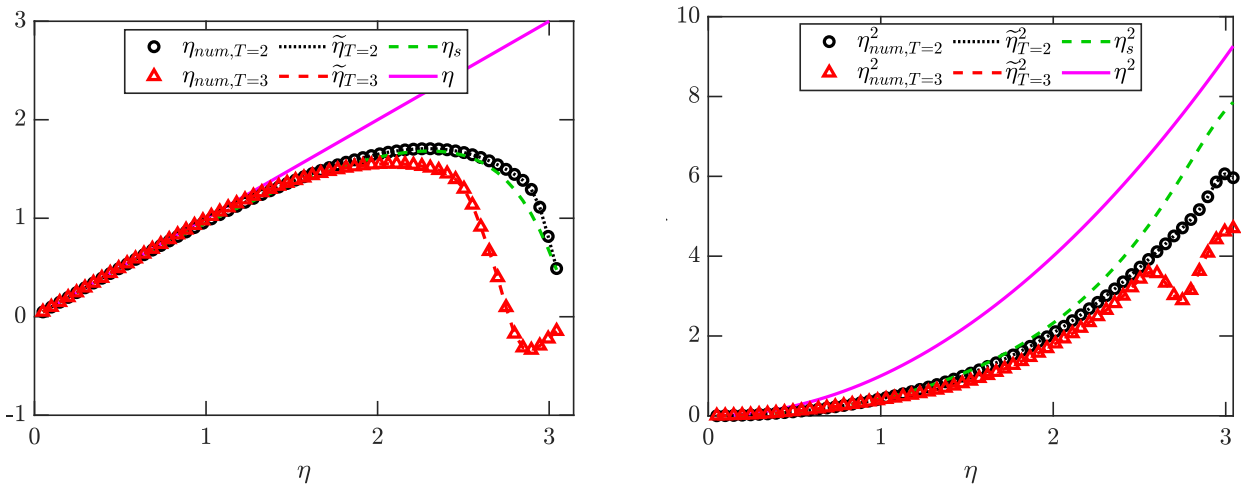

Figure 3.11: Spectral accuracy for the fully discrete advection-diffusion equation using asynchrony-tolerant (AT) schemes with a communication-avoiding algorithm (CAA) with periods $T = 2$ and 3, and $r_d = 0.2$, $r_c = 0.5$. Comparison of analytical and numerical modified wavenumbers for first derivative (left) and second derivative (right). Here $\widetilde{\eta}_T$ and $\widetilde{\eta}^2_T$ are given by Eq. (3.32) and Eq. (3.48), respectively and the subscript s denotes the modified wavenumber for the synchronous scheme.

close to the exact wavenumber only for a very small range of η.

We finally note that Eq. (3.47) can be generalized to a T-step amplification factor for a general CAA with delay $\ell = \{0, 1, \ldots, L\}$ and period $T = L + 1$ as,

$$\widetilde{G}^{(n)}_T = f_4(r_c, r_d, \eta) \prod_{l=1}^{L} \widetilde{G}^{(n-l)}_1 + f_5(r_c, r_d, L, \eta) + \frac{f_6(r_c, r_d, L, \eta)}{\widetilde{G}^{(n-T)}_1}. \tag{3.49}$$

whose analysis provides insight into the stability and spectral accuracy of this schemes.

In conclusion, through specific examples, we have shown how the proposed generalized von Neumann analysis can be used to study diverse numerical schemes including challenging cases such as asynchrony-tolerant schemes, which are not tractable with the standard von Neumann analysis.

3.6 Conclusions

Von Neumann analysis has been extensively used for the stability analysis of numerical schemes in wavenumber space. This involves the computation of the amplification factor which is a mea-

sure of growth or decay of different wavenumbers at every time step. Stability is then assured by bounding the magnitude of this amplification factor by unity. However, an underlying assumption in this analysis is that the amplification factor is independent of time. This leads to multiple amplification factors for fully discrete systems with more than two time levels. For example, for three-level schemes such as leapfrog (LF) and Du-fort and Frankel (DFF) schemes, the standard analysis results in a quadratic equation which can be solved to obtain two amplification factors ($G^*_\pm$). However, simple numerical simulations show that for both DFF and LF the observed amplification factor (i) is not equal to either of these roots $G^*_\pm$, (ii) varies in time, and (iii) can take values larger than unity even though the scheme is stable.

We, thus, introduce a generalized von Neumann analysis wherein we relax the assumption that the amplification factor does not vary in time. The resulting time-dependent amplification factor ($\widetilde{G}_1^{(n)}$) is a recursive function that can be expressed as a periodic continued fraction. The amplification $\widetilde{G}_1^{(n)}$ at any time level n is then simply equal to the n^{th} approximant of this continued fraction. In general, $\widetilde{G}_1^{(n)}$ depends upon the CFL, time level n and wavenumber η. By analyzing the convergence properties of the corresponding continued fraction, we show that $\widetilde{G}_1^{(n)}$ may either converge to a fixed value or oscillate with finite amplitudes in the long-time limit. In fact, when it does converge, $\widetilde{G}_1^{(n)}$ becomes independent of n and is equal to the standard amplification factor ($G^*_\pm$) with larger magnitude. While the amplification obtained from the standard von Neumann analysis is applicable only under certain conditions and for a range of wavenumbers—both of which that we determine exactly—the proposed generalized analysis gives the correct measure of amplification for all conditions and wavenumbers.

Since, in general $\widetilde{G}_1^{(n)}$ varies with n and may be greater than unity even if the scheme is stable, we define stability in terms of the product of amplification at each time level and require this product to be finite at all times (Eq. (3.33)). This allows for the amplitude to grow instantaneously as observed in actual simulations. While the modified definition of stability gives us the same stability bound for LF scheme as obtained by the standard analysis, this definition uses the actual value of amplification at all n.

We extended the generalized von Neumann analysis to the asynchronous Communication Avoiding Algorithm (CAA) wherein, asynchrony-tolerant (AT) schemes with periodically changing coefficients are used. Since the scheme changes with time, the one-step amplification factor is always time-dependent and thus, the standard von Neumann analysis cannot be used. The stability limit for CAA using the time-dependent amplification factor and Eq. (3.33) was numerically obtained. Results show that the stability region for CAA remains close to that obtained for corresponding synchronous scheme. Thus, the AT schemes when used with periodic delays exhibit superior stability in comparison to the stricter stability bounds obtained for fixed delays in recent work by [19].

Besides stability, we also assessed the spectral accuracy of the fully discrete systems, by computing the modified wavenumber from the amplification factor. While the modified wavenumber for the spatial differential operator is independent of CFL, for fully discrete system, the modified wavenumber varies with CFL because of spatio-temporal coupling. Somewhat surprisingly, this result, which is true also for the standard von Neumann analysis, seems to have not been studied in the literature before. Our results allowed to extend this even further to situations where when the amplification factor varies with time, which thus leads to a modified wavenumber which also varies with time. The main conclusion is that even for same the spatial scheme with a fixed CFL, the accuracy of the gradients and consequently, the accuracy of the simulation, can change in time.

In summary, we presented theoretical and numerical results that expose the limitations of the standard von Neumann analysis. We developed a generalized von Neumann analysis which overcomes these limitations and thus provides an effective alternate approach for the stability and spectral analysis of multi-level schemes, including asynchrony-tolerant schemes.

Appendix 3.A: Convergence of a periodic continued fraction

Let u and v be the two roots the quadratic equation

$$x^2 - bx - a = 0, \quad (a \neq 0, b \neq 0), \tag{3.50}$$

such that $|u| \geq |v|$ and the discriminant is $D = b^2 + 4a$. We can rearrange Eq. (3.50) to give $x = b + a/x$, which can be expressed in the form of the following periodic continued fraction,

$$x = b + \cfrac{a}{b + \cfrac{a}{b + \cfrac{a}{b + \ddots}}}. \tag{3.51}$$

Now, since $a = -uv$ and $b = u + v$, Eq. (3.51) can be re-written in terms of the roots u and v as

$$x = u + v - \cfrac{uv}{u + v - \cfrac{uv}{u + v - \cfrac{uv}{u + v - \ddots}}} = u + v - \cfrac{v}{1 + \rho - \cfrac{\rho}{1 + \rho - \cfrac{\rho}{1 + \rho - \ddots}}}, \tag{3.52}$$

where $\rho = v/u$. Following Theorem 2.1 in [33], we can then write the n^{th} approximant of Eq. (3.52) as

$$u + v - \cfrac{v}{\frac{v}{u} + \cfrac{1}{\sum_{p=0}^{n-1}(v/u)^p}}. \tag{3.53}$$

Eq. (3.53) gives important information about convergence of continued fraction Eq. (3.51).

Case 1: At least one of the coefficients a or b is imaginary

(i) If $|u| \neq |v|$, such that $|v/u| < 1$, the complex geometric series, $\sum_{p=0}^{\infty}(v/u)^p$, converges to $\frac{1}{1-v/u}$. Thus, Eq. (3.53) converges or equivalently Eq. (3.51) converges to u which is the root of Eq. (3.50) with larger magnitude. If $v = u$, the sum $\sum_{p=0}^{\infty}(v/u)^p$ is infinite and thus, the continued fraction Eq. (3.53) or Eq. (3.51) converges to $u = v$.

(ii) If $|u| = |v|$, $\lim_{n\to\infty}\left(\sum_{p=0}^{n}(v/u)^p\right)$ does not exist. In general, for $|v/u| = 1, u \neq v$, the sequence

$v/u, (v/u)^2, (v/u)^3, \ldots$ has atleast two different limit-points. Consequently, the continued fraction oscillates.

Case 2: Both a and b are real

(i) If $D > 0$, both u and v are real. Since $|u| > |v|$, the ratio $|v/u| < 1$ and thus the sum of geometric progression, $\sum_{p=0}^{\infty}(v/u)^p$, that appears in the denominator in Eq. (3.53) is equal to $\frac{1}{1-v/u}$. Thus Eq. (3.53) or equivalently Eq. (3.51) converges to u which is the root of Eq. (3.50) with larger magnitude.

(ii) If $D = 0$, the roots u and v are real and equal. In this case, the sum $\sum_{p=0}^{\infty}(v/u)^p$ is infinite and thus, the continued fraction Eq. (3.53) or Eq. (3.51) converges to $u = v$.

(iii) If $D < 0$, then the roots u, v are complex conjugates that satisfy $|u| = |v|$. The continued fraction oscillates (Refer Case 1(ii)).

4. ASYNCHRONY-TOLERANT WEIGHTED ESSENTIALLY NON-OSCILLATORY SCHEMES

4.1 Introduction

In the presence of sharp gradients or discontinuities in density, temperature and composition due to detonations, shocks or high pressure flames, the central finite difference schemes are prone to large oscillations and instabilities. An alternate well established numerical technique to accurately simulate flows with piecewise smooth solutions between discontinuities is the so called weighted essentially non-oscillatory schemes (WENO) [34, 35]. These schemes aim at achieving high-order accuracy at the smooth regions of the flow and resolve the discontinuities with minimal oscillations by automatically selecting the locally smoothest stencil. WENO schemes are considerably more expensive than regular central difference schemes due to the need to evaluate the same flux functions from multiple candidate stencils [36]. However, the numerical method is suitable for implementation in finite difference codes using stencil operations very similar to the central difference schemes. Also, the usual advantages of DNS using finite difference schemes including parallelism and scalability apply to the WENO schemes as well. Apart from DNS [37, 38], the advantages of WENO schemes have also been demonstrated in implicit large eddy simulations (iLES) [39, 40, 41].

In this chapter after a brief overview of the strandard WENO schemes, we look at the degrading effect of data asynchrony on these schemes. Following which we derive asynchrony-tolerant weighted essentially non-oscillatory (AT-WENO) schemes using the general procedure described in [16]. We also obtain the smoothness indicators with necessary accuracy requirements with delayed data at the PE boundaries. Further analysis of these AT-WENO schemes shows that by selectively eliminating certain low-order terms in the truncation error, the convexity in the so-called ideal weights is preserved. Moreover, the schemes so derived reduce to the standard WENO schemes in the absence of delays in both small and large stencils. We verify the accuracy of these

AT-WENO schemes through numerical simulations of both linear and non-linear equations.

4.2 Standard WENO schemes

We begin with a quick overview of the standard WENO schemes. Consider the one-dimensional (x-direction) version of the governing equations:

$$\frac{\partial Q}{\partial t} + \frac{\partial C}{\partial x} + \frac{\partial D}{\partial x} = S, \tag{4.1}$$

where Q is the solution vector, C is the vector comprising the convective flux terms, D is the viscous and molecular diffusion flux vector, and S is the vector of source terms. The exact form of terms in Q, C and D is defined in Chapter 7In the WENO framework, the terms contained in the D vector do not require any special treatment and are evaluated using central difference schemes (standard or AT). However, the convective flux terms in C have to be computed appropriately to ensure both stability and accuracy in regions near and at discontinuities. Specifically, the derivative of the convective flux $\partial C/\partial x$ at a point j is approximated using the flux at the edges, *i.e.*, $\widehat{C}_{j\pm\frac{1}{2}}$ of a cell $\mathcal{I}_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$. The flux at the edges is computed using the WENO approximation procedure that can be carried out using interpolation or reconstruction, and accordingly, requires point values or cell averages of the fluxes. Note that the hat notation $(\widehat{\ })$ is used to denote the variables at cell edges. The derivative is then computed using the relation

$$\left.\frac{\partial C}{\partial x}\right|_j = \frac{\widehat{C}_{j+\frac{1}{2}} - \widehat{C}_{j-\frac{1}{2}}}{\Delta x} \tag{4.2}$$

for a uniform grid and the order depends upon the order of the numerical flux approximation $\widehat{C}_{j\pm\frac{1}{2}}$. When using finite-differences, the WENO approximation directly yields the fluxes at the edges in terms of the fluxes at the grid points. For stability, appropriate up-winding of fluxes is also required and is achieved through splitting the flux, for example using the local Lax-Friedrichs flux splitting

methodology,

$$\begin{aligned} C &= C^+ + C^-, \quad \text{where} \\ C^\pm &= \frac{1}{2}(C \pm \lambda_{max} Q), \end{aligned} \tag{4.3}$$

at every grid-point. The quantity λ_{max} in Eq. (4.3) is the maximum local wave propagation speed which can be computed as

$$\lambda_{max}|_j = \max\{(|u|)_{j-1}, (|u \pm c|)_{j-1}, (|u|)_j, (|u \pm c|)_j\}, \tag{4.4}$$

where u and c are the local flow velocity and the speed of sound, respectively. The derivative is then simply evaluated using

$$\left.\frac{\partial C}{\partial x}\right|_j = \frac{\widehat{C}^+_{j+\frac{1}{2}} - \widehat{C}^+_{j-\frac{1}{2}}}{\Delta x} + \frac{\widehat{C}^-_{j+\frac{1}{2}} - \widehat{C}^-_{j-\frac{1}{2}}}{\Delta x} \tag{4.5}$$

with the appropriate upwind stencil for both approximating positive and negative fluxes [38]. For computation of derivative in Eq. (4.5) using the WENO procedure, the first step is to approximate the fluxes at the edges ($j \pm 1/2$) through an interpolation or a reconstruction procedure. For simplicity, the approximation of these fluxes using an interpolation technique is discussed here in detail. However, the approximation through reconstruction that treats grid-point values as cell averages can also be performed following the steps detailed here. In order to do so, the primitive function defined in [34] is required.

For illustration, consider a specific WENO scheme that uses three sub-stencils each of which comprises three grid-points denoted by $S^{(i)}$, $i = \{0, 1, 2\}$, in Fig. 4.1 to approximate the value of a quantity $\hat{u}$ at cell edges, say $j + 1/2$, and time-level n. Taken together the larger stencil obtained by combining these $S^{(i)}$ contains five points. Since the approximation order depends upon the number of points, a fifth-order accurate approximation can be obtained in the full stencil for a smooth function. This is the classical three-fifth order WENO scheme, where the three smaller candidate stencils $S^{(0)} = \{j-2, j-1, j\}$, $S^{(1)} = \{j-1, j, j+1\}$, and $S^{(2)} = \{j, j+1, j+2\}$

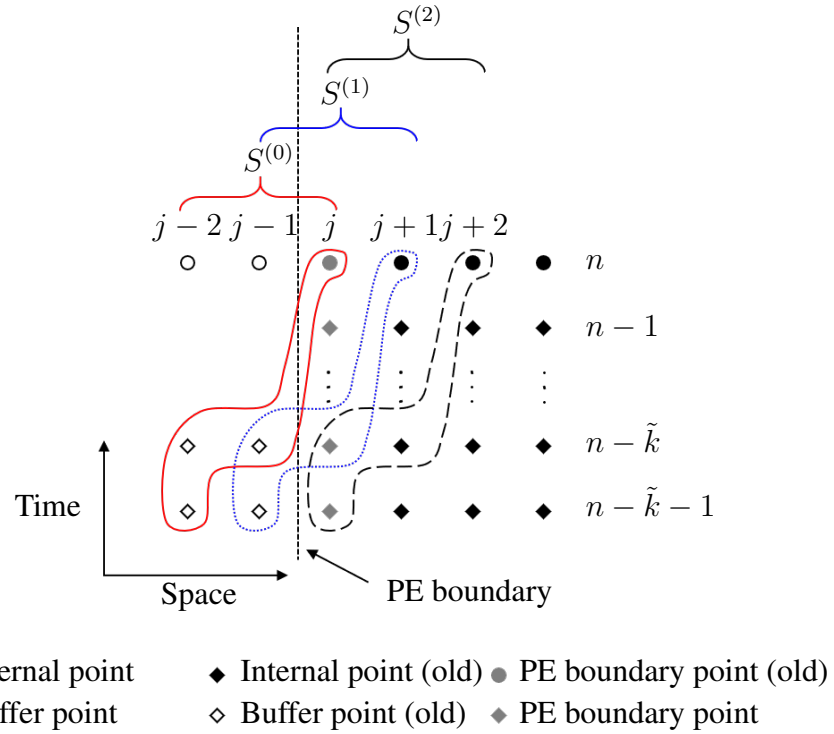

Figure 4.1: Four-point stencils for asynchronous WENO approximation at the left processor boundary. $S^{(0)}, S^{(1)}$ and $S^{(2)}$ are the three-point synchronous WENO stencils for approximation at point $j+1/2$ and the corresponding curves are the AT candidate stencil with delays on the left.

give a degree two polynomial interpolation of $\hat{u}^{(n)}_{j+\frac{1}{2}}$ at time level n. The synchronous third-order interpolant in each of the three stencils can be computed using Taylor series expansion or Lagrange interpolation [34, 35],

$$\begin{aligned}
\hat{u}^{n,(0)}_{j+\frac{1}{2}} &= \frac{3}{8}u^n_{j-2} - \frac{5}{4}u^n_{j-1} + \frac{15}{8}u^n_j + \mathcal{O}(\Delta x^3) \\
\hat{u}^{n,(1)}_{j+\frac{1}{2}} &= -\frac{1}{8}u^n_{j-1} + \frac{3}{4}u^n_j + \frac{3}{8}u^n_{j+1} + \mathcal{O}(\Delta x^3) \\
\hat{u}^{n,(2)}_{j+\frac{1}{2}} &= \frac{3}{8}u^n_j + \frac{3}{4}u^n_{j+1} - \frac{1}{8}u^n_{j+2} + \mathcal{O}(\Delta x^3).
\end{aligned} \tag{4.6}$$

The final approximation is taken as a convex combination of the above three third-order approximations, yielding a higher order interpolant in the larger stencil. Ideally, if $u(x,t)$ is smooth in the large stencil $S = \bigcup_{i=0}^{2} S^{(i)} = \{j-2, j-1, j, j+1, j+2\}$, a fifth-order approximation can

be achieved using the following expression:

$$\hat{u}^n_{j+\frac{1}{2}} = \sum_{i=0}^{2} \omega_i \hat{u}^{n,(i)}_{j+\frac{1}{2}}, \tag{4.7}$$

where ω_i's are the non-linear weights

$$\omega_i = \frac{\alpha_i}{\sum_{i=0}^{2} \alpha_i}, \quad \alpha_i = \frac{\gamma_i}{(\epsilon + \beta^{(i)})^2}. \tag{4.8}$$

The non-linear weights in Eq. (4.8) satisfy $w_i \geq 0$ and $\sum_{i=0}^{2} \omega_i = 1$ and are related to theso-called ideal or linear weights γ_i through the smoothness indicator $\beta^{(i)}$ [42, 35, 43]. Simple Taylor series expansion yields the ideal weights $\gamma = \{\frac{1}{16}, \frac{5}{8}, \frac{5}{16}\}$, such that

$$\hat{u}^n_{j+\frac{1}{2}} = \frac{3}{128}u^n_{j-2} - \frac{5}{32}u^n_{j-1} + \frac{45}{64}u^n_j + \frac{15}{32}u^n_{j+1} - \frac{5}{128}u^n_{j+2} + \mathcal{O}(\Delta x^5). \tag{4.9}$$

We now focus on the effect of data asynchrony on each of the individual approximations as well as the approximation in the full stencil.

4.2.1 Effect of asynchrony on standard WENO

Consider a point j at the left processor boundary with a delay $\tilde{k}$ encountered at each of the buffer points. The order of accuracy of the interpolants in the first two stencils is affected by this delay,

$$\begin{aligned} \hat{u}^{n,(0)}_{j+\frac{1}{2}} &= \frac{3}{8}u^{n-\tilde{k}}_{j-2} - \frac{5}{4}u^{n-\tilde{k}}_{j-1} + \frac{15}{8}u^n_j + \mathcal{O}(\Delta t, \Delta x \Delta t, \Delta x^3) \\ \hat{u}^{n,(1)}_{j+\frac{1}{2}} &= -\frac{1}{8}u^{n-\tilde{k}}_{j-1} + \frac{3}{4}u^n_j + \frac{3}{8}u^n_{j+1} + \mathcal{O}(\Delta t, \Delta x \Delta t, \Delta x^3). \end{aligned} \tag{4.10}$$

More precisely, there are terms in the truncation error that depend upon Δt. On relating $\Delta t \sim \Delta x^r$, the interpolants with delays reduce to first order when $r = 1$ (convective CFL) and to second order when $r = 2$ (diffusive CFL). The convex combination of the three interpolants with the ideal weights, γ_i, also has a leading order truncation term that scales as $\mathcal{O}(\Delta t)$, thereby degrading the

order even in the full stencil, *i.e*

$$\hat{u}^n_{j+\frac{1}{2}} = \frac{3}{128}u^{n-\tilde{k}}_{j-2} - \frac{5}{32}u^{n-\tilde{k}}_{j-1} + \frac{45}{64}u^n_j + \frac{15}{32}u^n_{j-1} - \frac{5}{128}u^n_{j+2} + \mathcal{O}(\Delta t, \Delta x\Delta t, \Delta t^2, \Delta x^2\Delta t). \quad (4.11)$$

Clearly, the standard interpolants cannot be used in the presence of delays. A similar degradation in order was observed when standard finite difference schemes are used with asynchrony [9] for computation of spatial derivatives. This necessitates the need to derive WENO schemes that are resilient to asynchony at a mathematical level. This is done next.

4.3 Asynchrony-tolerant WENO schemes

Following the general methodology to overcome this loss of accuracy in [16], the asynchrony-tolerant WENO schemes (AT-WENO) can be derived. For this, the AT-interpolant for each of the substencils in the presence of delays is computed. When multiple time-levels are used, the interpolation is two-dimensional i.e. in space and time, and thus, requires larger numbers of points to achieve a given order of accuracy. Since Taylor series expansion is used to obtain the coefficients, a relation of the form $\Delta t \sim \Delta x^r$ is also needed in order to identify the lower order terms that should be eliminated so as to achieve the desired order of accuracy. For example, for a third-order accurate approximation, when $\Delta t \sim \Delta x$ six equations corresponding to terms of the form $(1, \Delta t, \Delta x, \Delta t^2, \Delta t\Delta x, \Delta x^2)$ are needed, and therefore, each candidate stencil is comprised of six points. Similarly, if $r = 2$ only four equations $(1, \Delta t, \Delta x, \Delta x^2)$ are required, and therefore, a four-point stencil is sufficient to achieve a third-order approximation. Since the stencil now depends upon the delays observed at the processor boundaries, the stencil itself is random because of the random nature of the delays. One can represent this stochastic stencil as $\tilde{S}$. For the first candidate stencil, $\tilde{S}^{(0)} = \{u^{-\tilde{k}+n-1}_{j-2}, u^{n-\tilde{k}}_{j-2}, u^{n-\tilde{k}}_{j-1}, u^n_j\}$ with delay $\tilde{k}$ and $r = 2$, the AT interpolant is

$$\tilde{u}^{n,(0)}_{j+\frac{1}{2}} = \frac{7}{8}\tilde{k}u^{-\tilde{k}+n-1}_{j-2} + \frac{1}{8}\left(3 - 7\tilde{k}\right)u^{n-\tilde{k}}_{j-2} - \frac{5}{4}u^{n-\tilde{k}}_{j-1} + \frac{15}{8}u^n_j + \text{TE}^{(0)}. \quad (4.12)$$

The leading order truncation error in Eq. (4.12) term depends on the delay $\tilde{k}$ and is equal to

$$\text{TE}^{(0)} = \frac{-5}{4}\tilde{k}\dot{u}'(x,t)\Delta t\Delta x + \frac{5}{16}u'''(x,t)\Delta x^3 \sim \mathcal{O}(\Delta x^3), \tag{4.13}$$

where $\dot{u} = \partial u/\partial t$ and $u' = \partial u/\partial x$ and $\Delta t \sim \Delta x^2$ gives the overall order to be $\mathcal{O}(\Delta x^3)$. When $\tilde{k} = 0$, Eq. (4.12) reduces to the standard third-order synchronous interpolant in Eq. (4.6). For the second stencil $\tilde{S}^{(1)} = \{u_{j-1}^{-\tilde{k}+n-1}, u_{j-1}^{-\tilde{k}}, u_j^n, u_{j+1}^n\}$, with delays considered only at the buffer point (u_{j-1}), the third-order interpolant is

$$\tilde{u}_{j+\frac{1}{2}}^{n,(1)} = \frac{1}{8}\tilde{k}u_{j-1}^{-\tilde{k}+n-1} + \frac{1}{8}\left(-\tilde{k}-1\right)u_{j-1}^{n-\tilde{k}} + \frac{3}{4}u_j^n + \frac{3}{8}u_{j+1}^n + \text{TE}^{(1)}, \tag{4.14}$$

where the leading order truncation error term

$$\text{TE}^{(1)} = -\frac{1}{16}u'''(x,t)\Delta x^3 \sim \mathcal{O}(\Delta x^3) \tag{4.15}$$

is independent of delay $\tilde{k}$, unlike Eq. (4.13). Lastly, for stencil $\tilde{S}^{(2)} = \{u_j^n, u_{j+1}^n, u_{j+2}^n\} = S^{(2)}$, since all the points are within the PE, we can readily use the standard interpolant *i.e.* $\tilde{u}_{j+\frac{1}{2}}^{n,(2)} = \hat{u}_{j+\frac{1}{2}}^{n,(2)}$. The full stencil is comprised of seven points $\tilde{S} = \bigcup_{i=0}^{2} \tilde{S}^{(i)} = \{u_{j-2}^{-\tilde{k}+n-1}, u_{j-2}^{n-\tilde{k}}, u_{j-1}^{-\tilde{k}+n-1}, u_{j-1}^{n-\tilde{k}}, u_j^n, u_{j+1}^n, u_{j+2}^n\}$. With this choice of stencil, at most, a fourth-order accurate approximation can be obtained for the full stencil since the degrees of freedom are sufficient only to eliminate seven low-order terms. Extension to higher orders would require extending the stencil in both space and time. This is explained in great detail in [16] in the context of AT finite difference schemes.

On evaluation of the ideal weights for this larger stochastic stencil $\tilde{S}$ *i.e.*

$$\tilde{u}_{j+\frac{1}{2}}^{n} = \sum_{i=0}^{2} \gamma_i \tilde{u}_{j+\frac{1}{2}}^{n,(i)}, \tag{4.16}$$

yields

$$\gamma = \left\{0, \frac{1}{2}, \frac{1}{2}\right\} \tag{4.17}$$

which, in turn, gives a fourth-order truncation error term in Eq. (4.16). These ideal weights eliminate the first stencil $\tilde{S}^{(0)}$ and reduce the full stencil to a four-point synchronous stencil in the absence of delays. If the second stencil is modified $\tilde{S}^{(1)} = \{u_{j-1}^{-\tilde{k}+n-1}, u_{j-1}^{n-\tilde{k}}, u_j^{n-\tilde{k}}, u_{j+1}^n\}$, the ideal weights are no longer positive, $\gamma = \left\{-\frac{3}{2}, -\frac{5}{2}, 5\right\}$. Further changing the third stencil to $\tilde{S}^{(2)} = \{u_j^{-\tilde{k}+n-1}, u_j^{n-\tilde{k}}, u_{j+1}^{n-\tilde{k}}, u_{j+2}^n\}$ such that all three stencils now have similar stencil structure and thus use four points as shown in Fig. 4.1, the ideal weights obtained are

$$\gamma = \left\{\frac{3}{8}, \frac{5}{4}, -\frac{5}{8}\right\}. \tag{4.18}$$

This exercise shows that irrespective of the choice of asynchronous stencil with four points, at least one of the ideal or linear weights is non-positive for integer values of $\tilde{k}$. This non-convex nature of ideal weights leads to instabilities and oscillations [44]. The procedure to deal with the non-positive weights has been described in [44] and it involves splitting all the weights into positive and negative parts,

$$\begin{aligned} \tilde{\gamma}_i^+ &= \frac{1}{2}\left(\gamma_i + \theta|\gamma_i|\right) \\ \tilde{\gamma}_i^- &= \tilde{\gamma}_i^+ - \gamma_i \end{aligned} \tag{4.19}$$

for $i = \{0, 1, 2\}$. The split ideal weights are then scaled by the parameters

$$\sigma^\pm = \sum_{i=1}^{3} \tilde{\gamma}_i^\pm; \quad \gamma_i^\pm = \tilde{\gamma}_i^+ / \sigma^\pm, \tag{4.20}$$

which are used to write the split polynomial interpolations

$$\left(\hat{u}_{j+\frac{1}{2}}^n\right)^\pm = \sum_{i=0}^{2} \gamma_i^\pm \hat{u}_{j+\frac{1}{2}}^{n,(i)}, \tag{4.21}$$

which is equivalent to replacing the ideal positive weight γ_i in Eq. (4.8) by the corresponding scaled split weight $\gamma_i^\pm$ [44]. However, this procedure involve a series of additional computations to compute the derivative of the flux accurately.

For this particular choice of stencil with delayed data at the left boundaries, the reconstruction

polynomial can also be derived using the primitive function [34]. For example, for a stencil of the form $\tilde{S}^{(i)} = \{u_{j-1+i}^{-\tilde{k}+n-1}, u_{j+i}^{n-\tilde{k}}, u_{j+1+i}^{n-\tilde{k}}, u_{j+1+i}^{n}\}$ where $i = \{0, 1, 2\}$, one obtains the following AT approximations,

$$\begin{aligned}\tilde{u}_{j+\frac{1}{2}}^{n,(i)} =& \frac{1}{6}\left(3i^2 - 12i + 5\right)\tilde{k}u_{i+j-2}^{-\tilde{k}+n-1} + \frac{1}{6}\left(-3i^2\tilde{k} + 12i\tilde{k} - 5\tilde{k} + 3i^2 - 6i + 2\right)u_{i+j-2}^{n-\tilde{k}} \\ &+ \frac{1}{6}\left(-6i^2 + 18i - 7\right)u_{i+j-1}^{n-\tilde{k}} + \frac{1}{6}\left(3i^2 - 12i + 11\right)u_{i+j}^{n}.\end{aligned} \tag{4.22}$$

The reconstruction polynomials in Eq. (4.22) reduce to the traditional candidate reconstruction polynomials in [43, 34] when $\tilde{k} = 0$ corressponding to the classical three-fifth order WENO scheme with ideal weights $\{1/10, 6/10, 3/10\}$.

As is evident from Eq. (4.8), the non-linear weights ω_i that are integral to the WENO formulation depend upon the smoothness indicators β_i defined in [34] as

$$\beta^{(i)} = \sum_{\ell=1}^{k} \Delta x^{2\ell-1} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \left(\frac{d^\ell}{dx^\ell} p^{(i)}(x)\right)^2 dx \tag{4.23}$$

where $p^{(i)}(x)$ is the polynomial of degree k (=2, in the example considered here) for stencil $S^{(i)}$. For a one-dimensional polynomial interpolation Eq. (4.23) can be easily evaluated to give a smoothness indicator for each stencil. However, for a simpler extension to an asynchronous stencil, Simpson's 3/8 rule is used to compute $\beta^{(i)}$,

$$\begin{aligned}\beta^{(i)} = \frac{\Delta x}{6}\Bigg(& \sum_{\ell=1}^{k} \Delta x^{2\ell-1}\left(\left.\frac{d^\ell}{dx^\ell} p^{(j)}(x)\right|_{x=x_{j-\frac{1}{2}}}\right)^2 + 4\sum_{\ell=1}^{k} \Delta x^{2\ell-1}\left(\left.\frac{d^\ell}{dx^\ell} p^{(j)}(x)\right|_{x=x_{j-\frac{1}{6}}}\right)^2 \\ &+ 4\sum_{\ell=1}^{k} \Delta x^{2\ell-1}\left(\left.\frac{d^\ell}{dx^\ell} p^{(j)}(x)\right|_{x=x_{j+\frac{1}{6}}}\right)^2 + \sum_{\ell=1}^{k} \Delta x^{2\ell-1}\left(\left.\frac{d^\ell}{dx^\ell} p^{(j)}(x)\right|_{x=x_{j+\frac{1}{2}}}\right)^2\Bigg),\end{aligned} \tag{4.24}$$

such that the derivatives of the interpolant are computed numerically at each of $\{x_{j-\frac{1}{2}}, x_{j-\frac{1}{6}}, x_{j+\frac{1}{6}}, x_{j+\frac{1}{2}}\}$ using all the points in the stencil $S^{(i)}$. This yields the smoothness indicator expressed in the

form

$$\begin{aligned}
\beta^{(0)} &= \frac{1}{3}\left(4\left(u^n\right)_{j-2}^2 + \left(11u_j^n - 19u_{j-1}^n\right)u_{j-2}^n + 25\left(u^n\right)_{j-1}^2 + 10\left(u^n\right)_j^2 - 31u_{j-1}^n u_j^n\right) \\
\beta^{(1)} &= \frac{1}{3}\left(4\left(u^n\right)_{j-1}^2 + \left(5u_{j+1}^n - 13u_j^n\right)u_{j-1}^n + 13\left(u^n\right)_j^2 + 4\left(u^n\right)_{j+1}^2 - 13u_j^n u_{j+1}^n\right) \\
\beta^{(2)} &= \frac{1}{3}\left(10\left(u^n\right)_j^2 + \left(11u_{j+2}^n - 31u_{j+1}^n\right)u_j^n + 25\left(u^n\right)_{j+1}^2 + 4\left(u^n\right)_{j+2}^2 - 19u_{j+1}^n u_{j+2}^n\right)
\end{aligned} \tag{4.25}$$

which is consistent with $\beta^{(j)}$ in the literature. Extending this to the asynchronous stencil, the smoothness indicator of the following form is obtained when a four-point stencil is used for all three candidate stencils,

$$\begin{aligned}
\tilde{\beta}^{(0)} = \frac{1}{3}\Bigg(&\left(10\tilde{k}^2 - 11\tilde{k} + 4\right)\left(u_{j-2}^{n-\tilde{k}}\right)^2 + \Big(- 20\tilde{k}^2 u_{j-2}^{n-\tilde{k}-1} + \tilde{k}\left(11u_{j-2}^{n-\tilde{k}-1} + 31u_{j-1}^{n-\tilde{k}} - 20u_j^n\right) \\
&- 19u_{j-1}^{n-\tilde{k}} + 11u_j^n\Big)u_{j-2}^{n-\tilde{k}} + 10\tilde{k}^2\left(u_{j-2}^{n-\tilde{k}-1}\right)^2 + 25\left(u_{j-1}^{n-\tilde{k}}\right)^2 - 31u_j^n u_{j-1}^{n-\tilde{k}} \\
&+ \tilde{k}u_{j-2}^{n-\tilde{k}-1}\left(20u_j^n - 31u_{j-1}^{n-\tilde{k}}\right) + 10\left(u_j^n\right)^2\Bigg)
\end{aligned} \tag{4.26}$$

$$\begin{aligned}
\tilde{\beta}^{(1)} = \frac{1}{3}\Bigg(&\left(4\tilde{k}^2 - 5\tilde{k} + 4\right)\left(u_{j-1}^{n-\tilde{k}}\right)^2 + \Big(- 8\tilde{k}^2 u_{j-1}^{n-\tilde{k}-1} + \tilde{k}\left(5u_{j-1}^{n-\tilde{k}-1} + 13u_j^{n-\tilde{k}} - 8u_{j+1}^n\right) \\
&- 13u_j^{n-\tilde{k}} + 5u_{j+1}^n\Big)u_{j-1}^{n-\tilde{k}} + 4\tilde{k}^2\left(u_{j-1}^{n-\tilde{k}-1}\right)^2 + 13\left(u_j^{n-\tilde{k}}\right)^2 - 13u_{j+1}^n u_j^{n-\tilde{k}} \\
&+ \tilde{k}u_{j-1}^{n-\tilde{k}-1}\left(8u_{j+1}^n - 13u_j^{n-\tilde{k}}\right) + 4\left(u_{j+1}^n\right)^2\Bigg)
\end{aligned} \tag{4.27}$$

$$\begin{aligned}
\tilde{\beta}^{(2)} = \frac{1}{3}\Bigg(&\left(4\tilde{k}^2 - 11\tilde{k} + 10\right)\left(u_j^{n-\tilde{k}}\right)^2 + \Big(- 8\tilde{k}^2 u_j^{n-\tilde{k}-1} + \tilde{k}\left(11u_j^{n-\tilde{k}-1} + 19u_{j+1}^{n-\tilde{k}} - 8u_{j+2}^n\right) \\
&- 31u_{j+1}^{n-\tilde{k}} + 11u_{j+2}^n\Big)u_j^{n-\tilde{k}} + 4\tilde{k}^2\left(u_j^{n-\tilde{k}-1}\right)^2 + 25\left(u_{j+1}^{n-\tilde{k}}\right)^2 - 19u_{j+2}^n u_{j+1}^{n-\tilde{k}} \\
&+ \tilde{k}u_j^{n-\tilde{k}-1}\left(8u_{j+2}^n - 19u_{j+1}^{n-\tilde{k}}\right) + 4\left(u_{j+2}^n\right)^2\Bigg).
\end{aligned} \tag{4.28}$$

Each of the $\tilde{\beta}^{(i)}$ listed in Eq. (4.26), Eq. (4.27), Eq. (4.28) reduce to the corresponding $\beta^{(j)}$ expressions in Eq. (4.25) when $\tilde{k} = 0$. With the approximations and smoothness indicators in the candidate stencil known along with the ideal weights, the derivative of the flux can then be com-

puted using the standard WENO procedure.

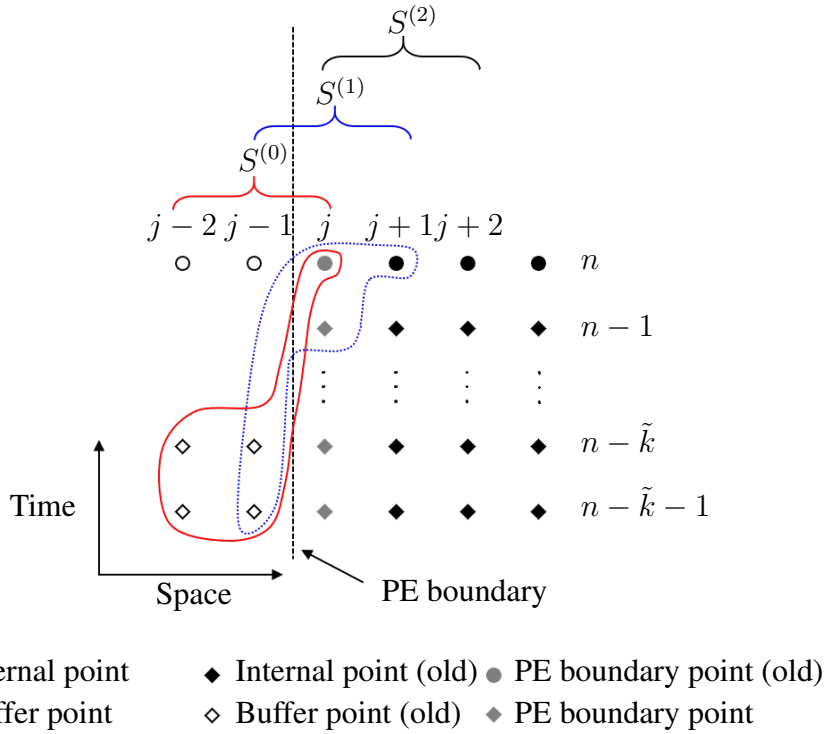

Figure 4.2: Updated stencil for WENO at the left processor boundary.

4.4 Improving AT-WENO schemes

For the choice of stencil discussed above, if the delay is zero then instead of achieving fifth-order approximation in the larger stencil, only fourth-order accuracy is obtained. This is explained next. The ideal weights $\{\gamma_0, \gamma_1, \gamma_2\}$ in Eq. (4.16) satisfy $\sum_{i=0}^{2} \gamma_i = 1$ and thus, two additional constraints need to be imposed to find a unique solution when seeking a combination of smaller stencils. Upon upgrading from a third-order approximation in the smaller stencil $\tilde{S}^{(j)}$ to a fourth-order approximation in the full stencil $\tilde{S}$, only two lower order truncation error terms of the form $\{\Delta t \Delta x, \Delta x^3\}$ can be eliminated. Thus, a unique solution for γ_i is obtained. However, such an approximation will never recover fifth-order accuracy in the absence of delays since the Δx^4 truncation error term is non-zero. Consequently, when there is no delay, the large stencil $\tilde{S}$ will

have at most four points and the approximation is only fourth-order accurate. For example, the ideal weights Eq. (4.18) yield non-zero coefficients only for points $\{j-2, j-1, j, j+2\}$ in the full stencil when $\tilde{k}=0$. Moreover, this stencil is spatially discontinuous.

The reduction in the number of points in the larger stencil when the delay is zero and a corresponding lower order accurate approximation can be overcome by selectively eliminating additional truncation error terms. For an asynchronous fifth-order approximation, terms of the form $\{\Delta t^2, \Delta t\Delta x^2, \Delta x^4\}$ in addition to $\{\Delta t\Delta x, \Delta x^3\}$ also need to be eliminated. Since this yields a system with six constraints and only three unknowns, a solution can not be uniquely determined. However, it is still possible to achieve a fifth-order approximation in the absence of delays. Addition of one more point in the smaller asynchronous stencil and using this degree of freedom to eliminate the $\Delta t\Delta x$ truncation term does not affect the order of the resulting interpolant. However, when the ideal weights in the large stencil are computed, one of the constraints can now be imposed on the Δx^4 term. This exercise then yields convex ideal weights that are exactly equal to the ones obtained when all three small stencils are synchronous. While the asynchronous approximation in the full stencil is still of order four, the synchronous approximation recovers fifth-order accuracy when the delay goes to zero. Moreover, a convex combination of the candidate stencil with none of the weights being negative is achieved. Thus, the additional computation involved in the treatment on negative weights is no-longer required. This new asynchronous stencil at the left boundary is given as

$$\begin{aligned}
\tilde{S}_L^{(0)} &= \{u_{j-2}^{n-\tilde{k}-1}, u_{j-2}^{n-\tilde{k}}, u_{j-1}^{n-\tilde{k}-1}, u_{j-1}^{n-\tilde{k}}, u_j^n\},\\
\tilde{S}_L^{(1)} &= \{u_{j-1}^{n-\tilde{k}-1}, u_{j-1}^{n-\tilde{k}}, u_j^{n-\tilde{k}}, u_j^n, u_{j+1}^n\}\\
\tilde{S}_L^{(2)} &= S^{(2)} = \{u_j^n, u_{j-1}^n, u_{j+2}^n\}.
\end{aligned} \tag{4.29}$$

and is shown in Fig. 4.2. For this stencil the third-order interpolation of the form

$$\begin{aligned}
\tilde{u}_{j+\frac{1}{2}}^{n,(0)} &= -\frac{3}{8}\tilde{k}u_{j-2}^{-\tilde{k}+n-1} + \frac{5}{4}\tilde{k}u_{j-1}^{-\tilde{k}+n-1} + \frac{1}{8}\left(3\tilde{k}+3\right)u_{j-2}^{n-\tilde{k}} + \frac{1}{4}\left(-5\tilde{k}-5\right)u_{j-1}^{n-\tilde{k}} + \frac{15}{8}u_j^n\\
\tilde{u}_{j+\frac{1}{2}}^{n,(1)} &= \frac{1}{8}\tilde{k}u_{j-1}^{-\tilde{k}+n-1} + \frac{1}{8}\left(-\tilde{k}-1\right)u_{j-1}^{n-\tilde{k}} + \frac{3}{4}u_j^n + \frac{3}{8}u_{j+1}^n\\
\tilde{u}_{j+\frac{1}{2}}^{n,(2)} &= \frac{3}{8}8u_j^n + \frac{3}{4}u_{j+1}^n - \frac{1}{8}u_{j+2}^n
\end{aligned} \tag{4.30}$$

is obtained when a constraint is imposed on the truncation error term $\Delta t \Delta x$ for the first two stencils, $\tilde{S}^{(0)}$ and $\tilde{S}^{(1)}$. The ideal weights are then computed to be

$$\gamma = \left\{ \frac{1}{16}, \frac{5}{8}, \frac{5}{16} \right\}. \tag{4.31}$$

While the interpolation procedure for approximating the fluxes at the edges is explained in detail, approximation using reconstruction, which is equivalent to the WENO approximation for the first derivative, is the main relevant WENO procedure when designing finite volume or finite difference schemes to solve hyperbolic conservation laws [45]. Similar constraints in terms of the order-of-accuracy and elimination of specific terms can also be used to derive a reconstruction function with an asynchronous stencil using a primitive variable. For the choice of stencil listed in Eq. (4.29), by treating point values as cell-averages the third-order reconstruction approximation at the left boundary assumes the following functional form

$$\begin{aligned}
\tilde{u}^{n,(0)}_{j+\frac{1}{2}} &= -\frac{1}{3}\tilde{k}u^{-\tilde{k}+n-1}_{j-2} + \frac{7}{6}\tilde{k}u^{-\tilde{k}+n-1}_{j-1} + \frac{1}{6}\left(2\tilde{k}+2\right)u^{n-\tilde{k}}_{j-2} + \frac{1}{6}\left(-7\tilde{k}-7\right)u^{n-\tilde{k}}_{j-1} + \frac{11}{6}u^{n}_{j} \\
\tilde{u}^{n,(1)}_{j+\frac{1}{2}} &= \frac{1}{6}\tilde{k}u^{-\tilde{k}+n-1}_{j-1} + \frac{1}{6}\left(-\tilde{k}-1\right)u^{n-\tilde{k}}_{j-1} + \frac{5}{6}u^{n}_{j} + \frac{1}{3}u^{n}_{j+1} \\
\tilde{u}^{n,(2)}_{j+\frac{1}{2}} &= \frac{1}{3}u^{n}_{j} + \frac{5}{6}u^{n}_{j+1} - \frac{1}{6}u^{n}_{j+2}
\end{aligned} \tag{4.32}$$

such that the coefficients reduce to their corresponding synchronous values when $\tilde{k} = 0$. The ideal or linear weights in this case can then be computed to be

$$\gamma = \left\{ \frac{1}{10}, \frac{3}{5}, \frac{3}{10} \right\} \tag{4.33}$$

which provides the fifth-order reconstruction polynomial in the larger stencil when the delay is zero, and is fourth-order, otherwise. Finally, the smoothness indicator for stencil Eq. (4.29) is

computed using Simpson's 3/8 rule,

$$\tilde{\beta}^{(0)} = \frac{1}{3}\Bigg(4\left(u_{j-2}^{n-\tilde{k}}\right)^2 + 25\left(u_{j-1}^{n-\tilde{k}}\right)^2 + \tilde{k}^2\Big(4\left(u_{j-2}^{n-\tilde{k}-1}\right)^2 + 4\left(u_{j-2}^{n-\tilde{k}}\right)^2 + 19u_{j-2}^{n-\tilde{k}}\left(u_{j-1}^{n-\tilde{k}-1} - u_{j-1}^{n-\tilde{k}}\right) + 25\left(u_{j-1}^{n-\tilde{k}-1} - u_{j-1}^{n-\tilde{k}}\right)^2 + u_{j-2}^{n-\tilde{k}-1}\left(19\left(u_{j-1}^{n-\tilde{k}} - u_{j-1}^{n-\tilde{k}-1}\right) - 8u_{j-2}^{n-\tilde{k}}\right)\Big) - 31u_j^n u_{j-1}^{n-\tilde{k}} + 10\left(u_j^n\right)^2 + u_{j-2}^{n-\tilde{k}}\left(11u_j^n - 19u_{j-1}^{n-\tilde{k}}\right) + \tilde{k}\Big(\left(19u_{j-1}^{n-\tilde{k}-1} - 38u_{j-1}^{n-\tilde{k}} + 11u_j^n\right)u_{j-2}^{n-\tilde{k}} + 8\left(u_{j-2}^{n-\tilde{k}}\right)^2 + u_{j-2}^{n-\tilde{k}-1}\left(-8u_{j-2}^{n-\tilde{k}} + 19u_{j-1}^{n-\tilde{k}} - 11u_j^n\right) - \left(u_{j-1}^{n-\tilde{k}-1} - u_{j-1}^{n-\tilde{k}}\right)\left(50u_{j-1}^{n-\tilde{k}} - 31u_j^n\right)\Big)\Bigg) \tag{4.34}$$

$$\tilde{\beta}^{(1)} = \frac{1}{3}\Bigg(4\tilde{k}^2\left(u_{j-1}^{n-\tilde{k}-1} - u_{j-1}^{n-\tilde{k}}\right)^2 + 4\left(u_{j-1}^{n-\tilde{k}}\right)^2 + 13\left(u_j^n\right)^2 + 4\left(u_{j+1}^n\right)^2 + \left(5u_{j+1}^n - 13u_j^n\right)u_{j-1}^{n-\tilde{k}} - 13u_j^n u_{j+1}^n - \tilde{k}\left(u_{j-1}^{n-\tilde{k}-1} - u_{j-1}^{n-\tilde{k}}\right)\left(8u_{j-1}^{n-\tilde{k}} - 13u_j^n + 5u_{j+1}^n\right)\Bigg) \tag{4.35}$$

$$\tilde{\beta}^{(2)} = \frac{1}{3}\Bigg(10\left(u_j^n\right)^2 - 31u_{j+1}^n u_j^n + 25\left(u_{j+1}^n\right)^2 + 4\left(u_{j+2}^n\right)^2 + \left(11u_j^n - 19u_{j+1}^n\right)u_{j+2}^n\Bigg) = \beta^{(2)}. \tag{4.36}$$

In terms of the leading order truncation error, these smoothness indicators have the same properties as their synchronous counterparts, and therefore, preserve the order characteristics of the non-linear weights in Eq. (4.7). Thus, the AT-WENO approximation in the simulations presented here is carried out using the reconstruction polynomials in Eq. (4.32), ideal weights in Eq. (4.33) and the smoothness indicators listed in Eq. (4.36). The corressponding approximations and the smoothness indicators in the asynchronous stencil at the right PE boundary are listed in Appendix 4.A. While only a specific example of AT-WENO is presented here, similar steps can be followed to derive high-order schemes as well by extending the stencil in both space and time. Furthermore, additional low-order truncation error terms can also be selectively eliminated in both candidate and full stencil such that the AT-WENO scheme in the full stencil reduces to the corresponding synchronous scheme in the absence of delays.

4.5 Order of accuracy test

Before proceeding with the validation simulations in Chapter 7, the order of accuracy of the synchronous and the new AT-WENO schemes is tested on a linear advection equation,

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0 \tag{4.37}$$

in a periodic domain $x \in [-1, 1]$ for initial condition $u(x, 0) = \sin^4(\pi x + 0.25)$ and a convective velocity of $c = 5$ for different spatial resolutions. Since the AT-WENO scheme is derived using the relation $\Delta t \sim \Delta x^2$, we retain this power-law for computing the time step in the simulations to assess the order of accuracy. The temporal scheme used is RK-3 implemented using the procedure listed in section 2.3. Delays with uniform probability are introduced at every grid-point. The L_1 and L_∞ norm of the error at $t = 1.0$ is tabulated in Table 4.1 for the synchronous case, standard WENO used asynchronously (AS-WENO) and the asynchrony-tolerant WENO (AT-WENO). As expected, the order degrades to two when the standard WENO sees delays at the boundaries and the error is orders of magnitude larger than the synchronous error. On the contrary, the AT-WENO scheme exhibit errors comparable to the synchronous case and the order of accuracy is close to four.

The analysis was repeated for an inviscid Burgers equation

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(\frac{u^2}{2}\right) = 0 \tag{4.38}$$

in a periodic domain with an initial condition, $u(x, 0) = 0.5 + \sin(\pi x)$. The order of accuracy both before and after the formation of the shock is computed. At $t = 0.15$, the solution is smooth throughout the domain and the order deterioration only occurs when the standard WENO is used asynchronously (AS-WENO). Both synchronous and AT-WENO have order higher than the theoretical order when the grid is refined. For L_1 and L_∞ norm of error after shock formation, only the smooth region at a distance of 0.05 on both sides of the shock, *ie.* $|x - \mathrm{x_{shock}}| > 0.05$ is considered

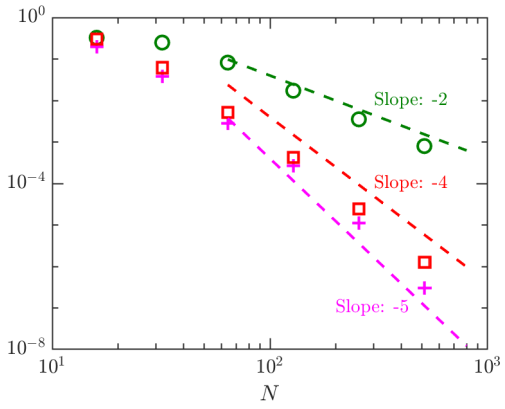

Figure 4.3: L_2 norm of error for linear advection equation with initial condition $u(x,0) = \sin^4(\pi x + 0.25)$ at $t = 1.0$ for synchronous (magenta pluses), AS-WENO (green circles) and AT-WENO (red squares).

[42]. Once again, errors in both the L_1 and L_∞ norms and the order of accuracy for AT-WENO are consistent with that for standard WENO. Thus, AT-WENO exhibits expected numerical accuracy for both linear and non-linear equations.

4.6 Conclusions

In this chapter we presented a brief overview of the standard weighted essentially non-oscillatory (WENO) schemes that are widely used for simulations of problems with jumps and discontinuties. The WENO schemes comprise several ingredients, including selection of small stencils in the vicinity of the point of interest, approximation of flux using grid-points in each of these stencils, computation of the ideal or linear weights, the smoothness indicators in each candidate stencils and the non-linear weights in terms of these linear weights and smoothness indicators. The final approximation of the flux at the cell edge is then a combination of the flux approximation in each stencil weighted by the non-linear weights. This procedure is used for computation of derivatives of the convective terms in the presence of sharp jumps, such as seen in high pressure flames and shocks or detonations. If delayed data is used at the processor boundary, then the numerical accuracy degrades incurring large errors. This is illustrated here through truncation error analysis as well as through numerical simulations.

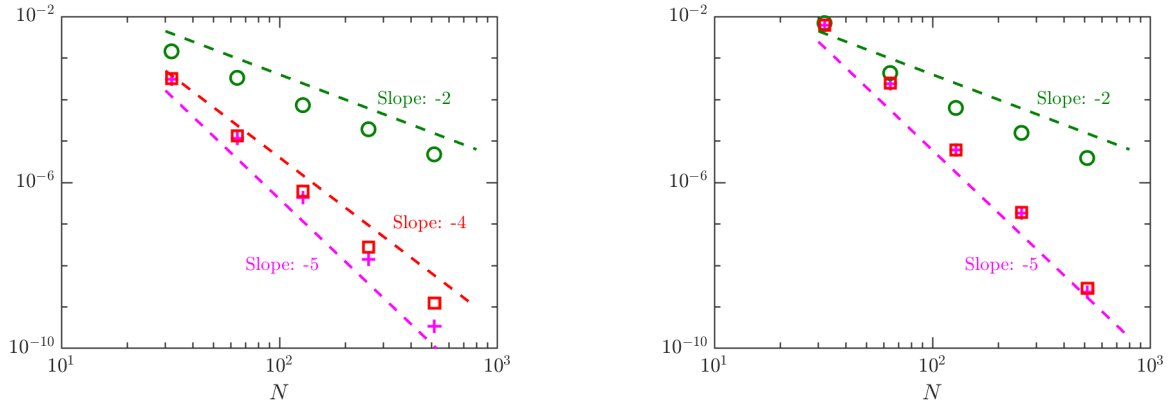

Figure 4.4: L_2 norm of error for invisicd Burgers' equation (a) before and (b) after shock with initial condition $u(x,0) = \sin^4(\pi x + 0.25)$ at $t = 1.0$ for synchronous (magenta pluses), AS-WENO (green circles) and AT-WENO (red squares).

Case	N	L_1 error	L_1 order	L_∞ error	L_∞ order
Synchronous (Order 5)	16	1.95E-01	-	3.93E-01	-
	32	3.87E-02	2.3	9.09E-02	2.1
	64	2.79E-03	3.8	5.57E-03	4.0
	128	2.65E-04	3.4	1.07E-03	2.4
	256	1.09E-05	4.6	7.00E-05	3.9
	512	3.01E-07	5.8	2.18E-06	5.0
AS-WENO $p = [0.5, 0.5]$ (Order 5)	16	3.25E-01	-	6.10E-01	-
	32	1.42E-01	1.2	2.93E-01	1.0
	64	2.84E-02	2.3	6.38E-02	2.2
	128	5.70E-03	2.3	1.32E-02	2.3
	256	1.22E-03	2.2	2.90E-03	2.2
	512	2.82E-04	2.1	6.79E-04	2.1
AT-WENO $p = [0.5, 0.5]$ (Order 4)	16	2.00E-01	-	3.86E-01	-
	32	4.15E-02	2.3	1.00E-01	1.9
	64	2.98E-03	3.8	6.70E-03	3.9
	128	2.90E-04	3.4	9.54E-04	2.9
	256	1.42E-05	4.4	7.04E-05	3.8
	512	5.79E-07	4.6	2.40E-06	4.9
AT-WENO $p = [0.3, 0.3, 0.3]$ (Order 4)	16	3.01E-01	-	5.09E-01	-
	32	6.18E-02	2.3	1.37E-01	1.9
	64	5.20E-03	3.8	1.13E-02	3.6
	128	4.22E-04	3.6	1.12E-03	3.3
	256	2.44E-05	4.1	7.73E-05	3.9
	512	1.26E-06	4.3	2.86E-06	4.8

Table 4.1: Order of accuracy for linear advection equation with initial condition $u(x,0) = \sin^4(\pi x + 0.25)$ at $t = 1.0$ for synchronous, AS-WENO and AT-WENO.

$t = 0.15$: Before shock

Case	N	L_1 error	L_1 order	L_∞ error	L_∞ order
Synchronous (Order 5)	32	3.03E-04	-	2.36E-03	-
	64	1.14E-05	4.7	1.08E-04	4.4
	128	4.38E-07	4.7	3.71E-06	4.9
	256	1.41E-08	5.0	2.12E-07	4.1
	512	3.37E-10	5.4	4.53E-09	5.5
AS-WENO $p = [0.5, 0.5]$ (Order 5)	32	7.99E-04	-	4.77E-03	-
	64	1.44E-04	2.5	7.21E-04	2.7
	128	3.39E-05	2.1	1.49E-04	2.3
	256	8.61E-06	2.0	3.76E-05	2.0
	512	2.05E-06	2.1	8.93E-06	2.1
AT-WENO $p = [0.5, 0.5]$ (Order 4)	32	3.11E-04	-	2.48E-03	-
	64	1.21E-05	4.7	1.15E-04	4.4
	128	4.85E-07	4.6	4.14E-06	4.8
	256	1.82E-08	4.7	2.07E-07	4.3
	512	6.61E-10	4.8	4.82E-09	5.4
AT-WENO $p = [0.3, 0.3, 0.3]$ (Order 4)	32	3.23E-04	-	2.67E-03	-
	64	1.32E-05	4.6	1.23E-04	4.4
	128	6.14E-07	4.4	4.88E-06	4.7
	256	2.74E-08	4.5	1.99E-07	4.6
	512	1.25E-09	4.5	7.42E-09	4.7

Table 4.2: Order of accuracy for inviscid Burgers' equation with initial condition $u(x, 0) = 0.5 + \sin(\pi x)$ before the shock for synchronous, AS-WENO and AT-WENO.

$t = 0.55$: After shock

Case	N	L_1 error	L_1 order	L_∞ error	L_∞ order
Synchronous (Order 5)	32	6.26E-03	-	1.20E-01	-
	64	2.43E-04	4.7	1.14E-02	3.4
	128	6.09E-06	5.3	3.00E-04	5.2
	256	1.83E-07	5.1	2.18E-05	3.8
	512	2.25E-09	6.3	3.79E-07	5.8
AS-WENO $p = [0.5, 0.5]$ (Order 5)	32	6.49E-03	-	1.20E-01	-
	64	3.21E-04	4.3	1.10E-02	3.4
	128	3.17E-05	3.3	3.48E-04	5.0
	256	6.96E-06	2.2	1.75E-05	4.3
	512	1.70E-06	2.0	4.14E-06	2.1
AT-WENO $p = [0.5, 0.5]$ (Order 4)	32	6.26E-03	-	1.20E-01	-
	64	2.44E-04	4.7	1.14E-02	3.4
	128	6.10E-06	5.3	2.99E-04	5.2
	256	1.85E-07	5.0	2.18E-05	3.8
	512	2.43E-09	6.2	3.78E-07	5.8
AT-WENO $p = [0.3, 0.3, 0.3]$ (Order 4)	32	6.26E-03	-	1.20E-01	-
	64	2.44E-04	4.7	1.14E-02	3.4
	128	6.15E-06	5.3	2.99E-04	5.3
	256	1.89E-07	5.0	2.18E-05	3.8
	512	2.80E-09	6.1	3.76E-07	5.9

Table 4.3: Order of accuracy for inviscid Burgers' equation with initial condition $u(x, 0) = 0.5 + \sin(\pi x)$ after the shock for synchronous, AS-WENO and AT-WENO.

To overcome the loss of accuracy in the presence of delays, we derive, for the first time, the asynchrony-tolerant weighted essentially non-oscillatory (AT-WENO) schemes using the methodology presented in [16]. This also includes computation of smoothness indicators with desired accuracy requirements with delayed. However, by direct extension of the [16] to derive AT-WENO schemes, we observed the standard WENO schemes are not recovered in the absence of delays. Furthermore, the linear weights so obtained are non-convex and can therefore lead to numerical instabilities and oscillations [44]. Through close analysis of the truncation error terms, we note that by introducing an additional grid point in the asynchronous stencil, we can selectively eliminate a low-order term. This in turn allows us to recover the standard WENO schemes when delay is zero and the resulting ideal weights are also convex. The superior accuracy of AT-WENO schemes in comparison to the standard WENO schemes in the presence of delays is exhibited through the order-of-accuracy tests for linear and non-linear equations. In Chapter 7 we use these AT-WENO schemes for propagation of a detonation wave.

Appendix 4.A: Reconstruction approximation at the right processor boundary

For points located at the right processor boundary the modified smaller asynchronous stencil is given by,

$$\begin{aligned}
\tilde{S}_R^{(0)} &= \{u_{j-2}^n, u_{j-1}^n, u_j^n\} \\
\tilde{S}_R^{(1)} &= \{u_{j-1}^n, u_j^n, u_j^{n-\tilde{k}}, u_{j+1}^{n-\tilde{k}} u_{j+1}^{n-\tilde{k}-1}\} \\
\tilde{S}_R^{(2)} &= \{u_j^n, u_{j+1}^{n-\tilde{k}}, u_{j+1}^{n-\tilde{k}-1}, u_{j+2}^{n-\tilde{k}} u_{j+2}^{n-\tilde{k}-1}\}
\end{aligned} \tag{4.39}$$

Here the delay appears at the rightmost gird points. We derive the following reconstruction approximation at the right boundary in each of these stencils,

$$\begin{aligned}
\tilde{u}_{j+\frac{1}{2}}^{n,(0)} &= \frac{u_{j-2}^n}{3} - \frac{7u_{j-1}^n}{6} + \frac{11u_j^n}{6} \\
\tilde{u}_{j+\frac{1}{2}}^{n,(1)} &= -\frac{1}{3}\tilde{k}u_{j+1}^{-\tilde{k}+n-1} + \frac{1}{6}\left(2\tilde{k}+2\right)u_{j+1}^{n-\tilde{k}} - \frac{1}{6}u_{j-1}^n + \frac{5u_j^n}{6} \\
\tilde{u}_{j+\frac{1}{2}}^{n,(2)} &= -\frac{5}{6}\tilde{k}u_{j+1}^{-\tilde{k}+n-1} + \frac{1}{6}\tilde{k}u_{j+2}^{-\tilde{k}+n-1} + \frac{1}{6}\left(5\tilde{k}+5\right)u_{j+1}^{n-\tilde{k}} + \frac{1}{6}\left(-\tilde{k}-1\right)u_{j+2}^{n-\tilde{k}} + \frac{u_j^n}{3}
\end{aligned} \tag{4.40}$$

Similarly, the approximation at point $u_{j-1/2}$ on the right boundary at each of the candidate stencils is listed below

$$\begin{aligned}
\tilde{u}^{n,(0)}_{j-\frac{1}{2}} &= -\frac{1}{6}u^n_{j-2} + \frac{5u^n_{j-1}}{6} + \frac{u^n_j}{3} \\
\tilde{u}^{n,(1)}_{j-\frac{1}{2}} &= \frac{1}{6}\tilde{k}u^{-\tilde{k}+n-1}_{j+1} + \frac{1}{6}\left(-\tilde{k}-1\right)u^{n-\tilde{k}}_{j+1} + \frac{u^n_{j-1}}{3} + \frac{5u^n_j}{6} \\
\tilde{u}^{n,(2)}_{j-\frac{1}{2}} &= \frac{7}{6}\tilde{k}u^{-\tilde{k}+n-1}_{j+1} - \frac{1}{3}\tilde{k}u^{-\tilde{k}+n-1}_{j+2} + \frac{1}{6}\left(-7\tilde{k}-7\right)u^{n-\tilde{k}}_{j+1} + \frac{1}{6}\left(2\tilde{k}+2\right)u^{n-\tilde{k}}_{j+2} + \frac{11u^n_j}{6}
\end{aligned} \tag{4.41}$$

Following the procedure described in the previous sections, the smoothness indicator at the right-boundary can be computed to be

$$\tilde{\beta}^{(0)} = \frac{1}{3}\left(4\left(u^n\right)^2_{j-2} + \left(11u^n_j - 19u^n_{j-1}\right)u^n_{j-2} + 25\left(u^n\right)^2_{j-1} + 10\left(u^n\right)^2_j - 31u^n_{j-1}u^n_j\right) \tag{4.42}$$

$$\begin{aligned}
\tilde{\beta}^{(1)} = \frac{1}{3}\Bigg(& u^n_{j-1}\left(-5\tilde{k}u^{n-\tilde{k}-1}_{j+1} + 5\left(\tilde{k}+1\right)u^{n-\tilde{k}}_{j+1} - 13u^n_j\right) + 4\left(\tilde{k}u^{n-\tilde{k}-1}_{j+1} - \left(\tilde{k}+1\right)u^{n-\tilde{k}}_{j+1}\right)^2 \\
& + 13u^n_j\left(\tilde{k}u^{n-\tilde{k}-1}_{j+1} - \left(\tilde{k}+1\right)u^{n-\tilde{k}}_{j+1}\right) + 4\left(u^n_{j-1}\right)^2 + 13\left(u^n_j\right)^2\Bigg)
\end{aligned} \tag{4.43}$$

$$\begin{aligned}
\tilde{\beta}^{(2)} = \frac{1}{3}\Bigg(& \tilde{k}^2\Bigg(25\left(u^{n-\tilde{k}-1}_{j+1}\right)^2 + \left(19\left(u^{n-\tilde{k}}_{j+2} - u^{n-\tilde{k}-1}_{j+2}\right) - 50u^{n-\tilde{k}}_{j+1}\right)u^{n-\tilde{k}-1}_{j+1} + 25\left(u^{n-\tilde{k}}_{j+1}\right)^2 \\
& + 4\left(u^{n-\tilde{k}-1}_{j+2} - u^{n-\tilde{k}}_{j+2}\right)^2 + 19u^{n-\tilde{k}}_{j+1}\left(u^{n-\tilde{k}-1}_{j+2} - u^{n-\tilde{k}}_{j+2}\right)\Bigg) + \tilde{k}\Big(50\left(u^{n-\tilde{k}}_{j+1}\right)^2 \\
& + 19\left(u^{n-\tilde{k}-1}_{j+2} - 2u^{n-\tilde{k}}_{j+2}\right)u^{n-\tilde{k}}_{j+1} + 8u^{n-\tilde{k}}_{j+2}\left(u^{n-\tilde{k}}_{j+2} - u^{n-\tilde{k}-1}_{j+2}\right) \\
& + u^{n-\tilde{k}-1}_{j+1}\left(19u^{n-\tilde{k}}_{j+2} - 50u^{n-\tilde{k}}_{j+1}\right)\Big) + 25\left(u^{n-\tilde{k}}_{j+1}\right)^2 + 4\left(u^{n-\tilde{k}}_{j+2}\right)^2 - 19u^{n-\tilde{k}}_{j+1}u^{n-\tilde{k}}_{j+2} \\
& + u^n_j\Big(-31u^{n-\tilde{k}}_{j+1} + 11u^{n-\tilde{k}}_{j+2} \\
& + \tilde{k}\left(31u^{n-\tilde{k}-1}_{j+1} - 31u^{n-\tilde{k}}_{j+1} - 11u^{n-\tilde{k}-1}_{j+2} + 11u^{n-\tilde{k}}_{j+2}\right)\Big) + 10\left(u^n_j\right)^2\Bigg)
\end{aligned} \tag{4.44}$$

Appendix 4.B: Reconstruction approximation at the left processor boundary

The approximation of $u_{j-1/2}$ at the left-boundary for the asynchronous stencils shown in Fig. 4.2 is

$$\begin{aligned}
\tilde{u}^{n,(0)}_{j-\frac{1}{2}} &= \frac{1}{6}\tilde{k}u^{-\tilde{k}+n-1}_{j-2} - \frac{5}{6}\tilde{k}u^{-\tilde{k}+n-1}_{j-1} + \frac{1}{6}\left(-\tilde{k}-1\right)u^{n-\tilde{k}}_{j-2} + \frac{1}{6}\left(5\tilde{k}+5\right)u^{n-\tilde{k}}_{j-1} + \frac{u^n_j}{3} \\
\tilde{u}^{n,(1)}_{j-\frac{1}{2}} &= -\frac{1}{3}\tilde{k}u^{-\tilde{k}+n-1}_{j-1} + \frac{1}{6}\left(2\tilde{k}+2\right)u^{n-\tilde{k}}_{j-1} + \frac{5u^n_j}{6} - \frac{u^n_{j+1}}{6} \\
\tilde{u}^{n,(2)}_{j-\frac{1}{2}} &= \frac{11}{6}u^n_j - \frac{7}{6}u^n_{j+1} + \frac{1}{3}u^n_{j+2}.
\end{aligned} \tag{4.45}$$

5. COMPUTATIONAL ALGORITHMS FOR ASYNCHRONOUS SIMULATIONS*

5.1 Introduction

The preceeding chapters focussed on devising asynchronous numerical methods and investigating their underlying properties. We now look into the application of the asynchrony-tolerant (AT) schemes for performing asynchronous simulations of compressible turbulence. An important first step in achieving this goal, is the development of a 3D Navier-Stokes (NS) solver that facilitates asynchronous communications between processing elements (PEs). One also needs to consider how asynchrony is introduced in the solver since it has implications in terms of both numerical and computational performance. We propose two approaches for introducing asynchrony: one that avoids synchronizations and the other that avoids communications . While the former leads to reduction in processor idling time and results in machine-dependent random delays at processor boundaries, the latter leads to periodic delays and reduction in the net volume of communications. Since power consumption, especially for data movement is expected to be a major concern for the next generation exascale machines, the reduced frequency of communications in communication avoiding algorithm, make it a viable energy efficient alternative to standard approaches. Moreover, as shown in Chapter 3 (section 3.5), the communication avoiding algorithms with periodic delays, have stability and spectral accuracy characteristics very similar to the standard synchronous algorithms.

5.2 Domain decomposition

The compressible flow solver is parallelized using a 3D domain decomposition and each PE is responsible for computations in a piece of this 3D domain. Communications between PEs are localized to the nearest neighbors only. In Fig. 5.1 we see a simple 1D domain decomposition where every PE has N_T grid points and two neighbors. The number of internal points (N_I) $i.e.$ the

*Parts of this chapter are reprinted from Journal of Computational Physics, Volume 419, Komal Kumari and Diego A. Donzis, "Direct numerical simulations of turbulent flows using high-order asynchrony-tolerant schemes: Accuracy and performance", Pages 109626, 15 October 2020, with permission from Elsevier.

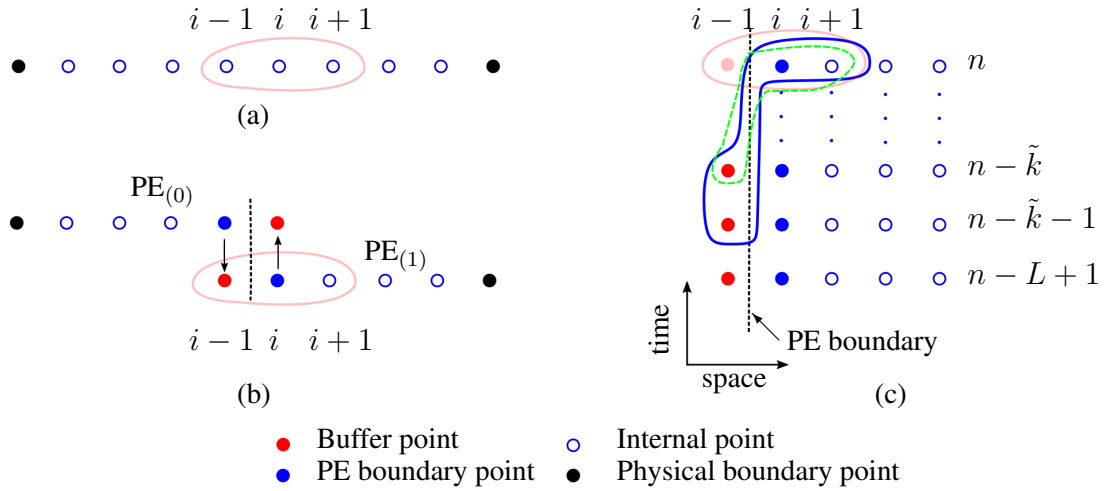

Figure 5.1: 1D domain discretization: (a) into one PE (serial) (b) into two PEs in parallel. (c) shows the left boundary of PE$_{(1)}$ with L time levels. Different lines are: standard synchronous scheme in faded-red (Eq. (1.2)), standard scheme used asynchronously in dashed-green (Eq. (1.3)), second-order AT scheme in solid-blue (Eq. (1.9)) at point i and PE boundary in dashed-black. For all the schemes $M = 1$.

points that use standard (synchronous) finite differences, with M points in each direction is equal to $N_T - 2M$. The total number of boundary points (N_B) and buffer points (N_{Bf}) in this case are equal to $2M$. Clearly $N_B \cup N_I = N_T$ and $N_B \cap N_I = \varnothing$.

Extending this idea to a 3D computational topology, such that each PE has a total of twenty six neighboring PEs, we can compute the total number of internal and boundary points. Consider a general 3D domain with N_x,N_y, and N_z grid points and P_x, P_y, and P_z processors in the x, y, and z directions, respectively. Then the total number of grid points (N_T) per PE is

$$N_T = \frac{N_x N_y N_z}{P_x P_y P_z}. \tag{5.1}$$

Using spatial schemes which require M points on each side for all three directions, it is easy to show that the number internal points (N_I) is

$$N_I = \left(\frac{N_x}{P_x} - 2M\right)\left(\frac{N_y}{P_y} - 2M\right)\left(\frac{N_z}{P_z} - 2M\right). \tag{5.2}$$

Since communications are done across all six faces of a PE, Eq. (5.1) and Eq. (5.2) gives us the exact number of boundary points (N_B) or the points that use AT schemes for the computation of spatial derivatives,

$$N_B = \frac{N_x N_y N_z}{P_x P_y P_z} - \left(\frac{N_x}{P_x} - 2M\right)\left(\frac{N_y}{P_y} - 2M\right)\left(\frac{N_z}{P_z} - 2M\right). \tag{5.3}$$

We can then compute the percentage of points that use AT schemes,

$$N_B(\%) = 100\left(\frac{N_B}{N_T}\right), \tag{5.4}$$

which can be used as a metric of the extent in space in which asynchrony affects the computations of derivatives directly.

5.3 Parameters and algorithms

We solve the NS equation for five variables $(\rho, \rho u_1, \rho u_2, \rho u_3, \rho e)$ at every time step. Since data at older time levels is used for AT schemes, each PE stores $5 \times (N_I + N_B + N_{bf}) \times (L + \mathtt{t}_\ell)$ data points, where L is the maximum allowed delay that can also be used as a control parameter for error and stability as we show below and $\mathtt{t}_\ell$ is the number of consecutive time-levels required for the computation of derivatives by AT schemes. We use two-sided non-blocking MPI calls (*MPI_Isend, MPI_Irecv*) for asynchronous communications between the PEs across the six faces of the 3D computational domain. In each direction, these communications are limited to immediate neighbors only. The status of these non-blocking communications is checked using *MPI_test* and it is utilized to compute delay at each PE boundary. To control the manner in which asynchrony appears we utilize two control parameters $\mathtt{c}_r$ and L. The *communication rate* $\mathtt{c}_r$ specifies the frequency of communication in each direction, that is to say, PEs initiate communication calls every $\mathtt{c}_r$ consecutive time steps. The second parameter is the maximum allowed delay, L. PEs impose explicit synchronization by invoking *MPI_Wait* whenever instantaneous delays at PE boundaries cross this threshold L. This synchronization is imposed only in the direction in which the delay is larger than

L and is thus local in nature. These two parameters determine the nature of delays. For example, if $\mathtt{c}_r > 1, L > 1$ delays are periodic and if $\mathtt{c}_r = 1, L > 1$ then delays at PE boundaries are random. In both the cases the delays are however bounded by L. A synchronous simulations is realized when $\mathtt{c}_r = 1$ and $L = 1$. Irrespective of $\mathtt{c}_r$ and L, global communications and synchronizations involving all PEs are done only for I/O.

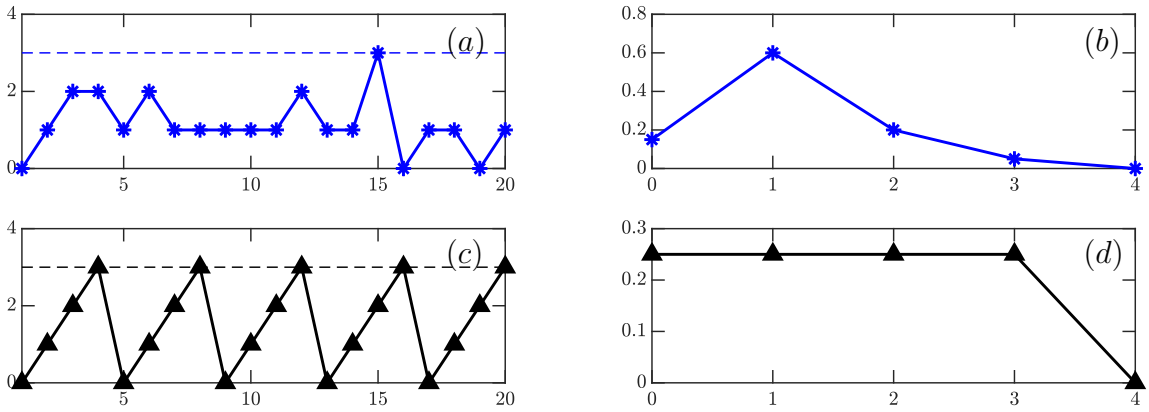

Figure 5.2: Simulated time series of delays for (a) SAA with $\mathtt{c}_r = 1, L = 3$ and (c) CAA with $\mathtt{c}_r = 4, L = 3$. PDF of these simulated delays for (b) SAA and (d) CAA.

5.3.1 Synchronization Avoiding Algorithm (SAA): random delays

For $\mathtt{c}_r = 1$ and $L > 1$, we have what we call a synchronization avoiding algorithm: a local synchronization is applied $if\ and\ only\ if$ the delay ($\tilde{k}$) at a PE boundary is greater than the maximum allowed delay L. We use circular send (U_{send}) and receive (U_{recv}) buffers in each direction for communicating to and from the neighboring PEs, respectively. At each time step, PEs sent data at only one time level across the boundaries. A generalized SAA is listed in Algorithm 1, where computations can proceed without waiting for updated values whenever $\tilde{k} \leq L$. When synchronization is not imposed, communications can complete in the background, facilitating overlap between communications and computations. Despite the reduction of synchronization overheads

and PE idling, this method does require communication at every step. The delay observed at PE boundaries is a function of machine characteristics, such as, network performance, processor and memory speeds etc., and is therefore a random variable, with a different value at each of the six PE boundaries. Since the delays and, consequently, coefficients of AT schemes may be different for each PE boundary, some additional numerical errors can be introduced due to random nature of these delays. Numerical simulations show that this effect is negligible for values of L that satisfy stability (section 5.4).

A typical time series of random delays for SAA is shown in Fig. 5.2(a) along with its PDF in Fig. 5.2(b). In this example, the delay is bounded by $L = 3$ as shown by the dashed blue line in Fig. 5.2(a). The statistical moments of the distribution of delays have an effect on the accuracy of the solution [16]. Since statistical characteristics of the delays can be controlled by forced synchronization, L becomes a parameter for error control.

Algorithm 1: Synchronization Avoiding Algorithm (SAA). Here U^n is the variable array at time level n, U_{send} is the send buffer, U_{recv} is the receive buffer, U^n_{buffer} is the data at buffer points, $U^n_{boundary}$ is the data at boundary points and f evaluates the discretized NS equation using AT schemes in space and AB schemes in time for number of time steps equal to st.

Synchronous Loop: Initialize $L + \mathtt{t}_\ell$ levels of $U, U_{send} \leftarrow U_{boundary}, U_{buffer} \leftarrow U_{recv}$

Asynchronous Loop:

for $n = L + \mathtt{t}_\ell + 1, \cdots, st$ **do**

- $U^{n+1} = f(U^n, U^{n-1}, ..., U^{n-T+1})$
- $U_{send} \leftarrow U^{n+1}_{boundary}$
- Send data across 6 faces: MPI_Isend
- **for** $face = 1 : 6$ **do**
  - Check communication status:MPI_Test
  - Compute delay $(\tilde{k})$
  - **if** delay $(\tilde{k}) \leq L$ **then**
    - $U_{buffer} \leftarrow U_{recv}$
  - **else**
    - Force synchronization: MPI_Wait
    - Update delay $(\tilde{k})$
    - $U_{buffer} \leftarrow U_{recv}$
  - **end**
  - Compute coefficients of the AT schemes (Appendix B)
- **end**

end

5.3.2 Communication Avoiding Algorithm (CAA): periodic delays

As alternative to communicating at every time step, we propose the so-called communication avoiding algorithm, in which the PEs communicate periodically every $\mathtt{c}_r > 1$ steps. As a result, the delay changes periodically from 0 (no delay) to a maximum allowed delay L which satisfies

$L = \mathtt{c}_r - 1$. Because of this periodicity, the delay across all the PE boundaries is the same in every direction. Since PEs communicate every $\mathtt{c}_r$ time steps, the send and receive buffers now have data at $min(\mathtt{t}_\ell, \mathtt{c}_r)$ time levels. This multiple time level data is required for computation of derivatives using AT schemes at the communication avoiding time steps. We have listed a generalized implementation of CAA in Algorithm 2, where the delay is incremented by one when PEs do not communicate. A typical time series of delays bounded by $L = 3$ (dashed black line) is shown in Fig. 5.2(c) for CAA with $\mathtt{c}_r = 4$. The delay in this case is deterministic and the PDF shown in Fig. 5.2(d) has a uniform distribution. Both delay and its PDF are independent of the machine characteristics and depend only upon the control parameters, contrary to SAA where the delay is random and its PDF is machine specific. CAA reduce the total latency time by a factor of $\mathtt{c}_r$ in comparison to synchronous avoiding or standard synchronous algorithms and are therefore particularly effective in latency-dominated machines. Furthermore, because of the reduction in frequency of communications, the energy consumption for these algorithms is also expected to be reduced. We also assessed the stability of the CAA in section 3.5 and showed that these algorithms permit a time-step that is comparable to its standard synchronous counterpart. One drawback of the communication avoiding algorithms is the larger size of send and receive buffers that could adversely affect performance for bandwidth-dominated machines.

Algorithm 2: Communication avoiding algorithm. Here U^n is the variable array at time level n, U_{send} is the send buffer, U_{recv} is the receive buffer, U^n_{buffer} is the data at buffer points, $U^n_{boundary}$ is the data at boundary points and f evaluates the discretized NS equation using AT schemes in space and AB schemes in time for number of time steps equal to st.

Synchronous Loop: Initialize $L + \mathtt{t}_\ell$ levels of $U, U_{send} \leftarrow U_{boundary}, U_{buffer} \leftarrow U_{recv}$

Compute $\ell = min(\mathtt{t}_\ell, \mathtt{c}_r)$

Asynchronous Loop:

for $\underline{n = L + 1, \cdots, st}$ **do**

  $U^{n+1} = f(U^n, U^{n-1}, ..., U^{n-T+1})$

  for $\underline{face = 1 : 6}$ **do**

    if $\underline{(mod(n, \mathtt{c}_r) == 0)}$ **then**

      $U_{send} \leftarrow U^{n+1}_{boundary}, ..., U^{n-\ell}_{boundary}$

      MPI_Isend and MPI_Irecv

      $U_{buffer} \leftarrow U_{recv}$

      $\tilde{k} \leftarrow 0$

    else

      Update delay: $\tilde{k} \leftarrow \tilde{k} + 1$

    end

    Compute coefficients of the AT schemes (Appendix B)

  end

end

5.4 Maximum delay L

The maximum allowed delay L is an important control parameter as it determines the error and stability of the AT schemes as well as the computational performance of the solver. As shown in [16], the error due to asynchrony in AT schemes is a function of statistical moments of delays which depend upon the architecture of the machine, communication links and patterns, latency, bandwidth and clock speed. Since the asynchronous error grows with L [16], very large values

of L can affect the accuracy of simulations. Furthermore, the memory requirement of all stored variables, the size of send and receive buffers and the rate of synchronizations and communications are also directly affected by the choice of parameter L. It is therefore critical that L be chosen judiciously in simulations and this choice can be based on two main factors, including stability that is discussed in Chapter 2 and performance that is described next.

The maximum allowed delay L has implications in terms of the computational implementation of the solver. Increasing L increases the number of times levels that need to be stored which increases memory requirement. At the same time, if L is too small then synchronization will be forced more often than required and asynchrony will not be leveraged efficiently. In practice, one can run a short simulation with a very large L and obtain the PDF of the delays ($\tilde{k}$). From this data, one can calculate an appropriate L by requiring $P(\tilde{k} > L) \lesssim c$, that is to say, one would expect forced synchronization $c\%$ of the time. Thus, c exposes tradeoff between performance and accuracy through the degree of asynchrony. For example, at $c = 0$ the simulation is completely asynchronous, $i.e.$, synchronization is never imposed, which is detrimental to accuracy if L is large.

For illustration purposes, in Fig. 5.3 we show PDF of delays $(\tilde{k})$ on three large systems at Texas Advanced Computing Center (TACC), namely, Stampede2, Frontera and Lonestar5 for $L = 10$ and different processor counts. From the black lines, we can clearly see that the probability of delays decreases with increasing delay and $P(\tilde{k} > 3) \lesssim 0.05$ on Stampede2. This implies that for a simulation with $L = 3$, synchronizations will be forcefully imposed less than 5% of the time. The trend is consistent even if we double the number of processors from $P = 8192$ (solid black) to $P = 16384$ (dashed black). For Frontera (red lines in Fig. 5.3) we see that the probability of $\tilde{k} = 1$ is higher than the probability of $\tilde{k} = 0$ for all the three cases. This points to a slow network that is expected to adversely affect the scaling for standard synchronous simulations. We see similar behavior for Lonestar5 (blue), with probability of $\tilde{k} = 1$ being the maximum. For both Frontera and Lonestar5, $P(\tilde{k} > 3) \lesssim 0.05$, for all the processor counts shown in Fig. 5.3. Thus, $L = 3$ is a reasonable choice for these three machines. Note that for CAA this is equivalent to a reduction

in the volume of communications by a factor of four. This reduction will be particularly critical when the PE count is high as envisioned in the exascale machines.

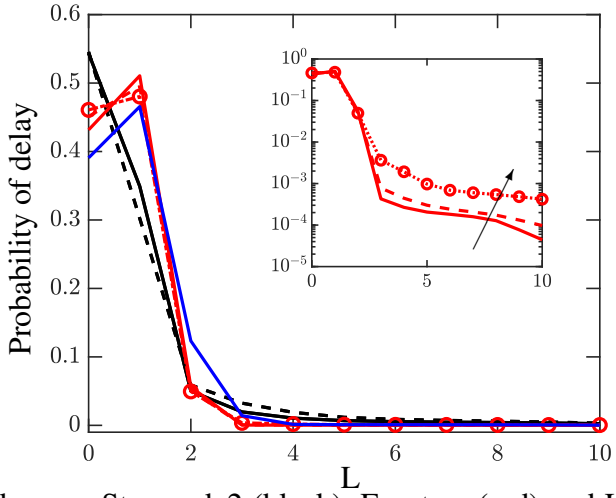

Figure 5.3: PDF of delays on Stampede2 (black), Frontera (red) and Lonestar5 (blue) with maximum allowed delay of $L = 10$. Different lines are $P = 4096$ (solid red, solid blue), $P = 8192$ (solid black, dashed red) and $P = 16384$ (dashed black) and $P = 262144$ (dotted-circles). Inset is PDF of delays on Frontera in linear-log scale with arrow denoting increasing P.

It is also worth noting the non-vanishing probability of delays as large as $L = 10$ in Fig. 5.3 (inset) for Frontera, indicating that at least some fraction of communications were synchronized. While this is not of much consequence at low processor counts, for an increasingly large number of PEs, even a small probability of large delays can account for severe overheads. For example, for a seemingly low probability of $P(\tilde{k} = 10) \sim \mathcal{O}(10^{-4})$, at a processor count of $\mathcal{O}(10^5)$, at least $\mathcal{O}(10^1)$ processors see a delay of $L = 10$ at the boundaries and are forced to synchronize at every time step. Considering that the probability of large delays increases with increasing number of PEs as seen in Fig. 5.3 (inset), a much larger fraction of processors would see large delays in the next generation exascale machines where the number of PEs is expected to be of $\mathcal{O}(10^6)$-$\mathcal{O}(10^9)$ with increased architectural inhomogeneity. Thus, even with a large value for the maximum allowed delay, a significant number of PEs would be subject to forced synchronization at extreme scales. However, these synchronizations would still be extremely small in comparison to the standard

synchronous algorithms that require all PEs to synchronize at all times.

In general, the maximum allowable delay (L) is chosen such that the PEs incur in minimal overheads because of forced synchronization and communications, without additional computational cost to ensure stability.

5.5 Conclusions

We developed a first-of-its-kind threee-dimensional asynchronous Navier-Stokes solver that uses AT schemes for computation of spatial derivatives at the processor boundaries. We introduced two ways to allow for asynchrony, namely, communication avoiding and synchronization avoiding algorithms (CAA and SAA, respectively). The former leads to deterministic delays with a uniform probability distribution, whereas the latter leads to random delays with a machine specific delay distribution. While in this work CAA and SAA are presented as two separate algorithms, a combination of the two can also be used, such that, both communication and synchronization rates are greater than one. This in principle can potentially lead to further reduction in overheads associated with the communication and synchronization. We use CAA and SAA for asynchronous simulations of decaying and solenoidally forced turbulence that are presented in the next Chapter where we also look at the computational perfomance of both of these algorithms.

6. ASYNCHRONOUS DNS OF COMPRESSIBLE TURBULENCE*

6.1 Introduction

Several numerical methods have been used for Direct Numerical Simulations (DNS) [4] of the Navier-Stokes equations to study turbulence, depending upon the complexity of the domain and the nature of problem of interest. Spectral methods [46], known for accurate computation of derivatives, have been used extensively in incompressible simulations. However, these present challenges when extended to non-periodic boundary conditions. An alternative to these methods, that is more amenable to the choice of boundary conditions, is the compact difference schemes that have spectral like resolution [47]. These are widely used for simulations of multi-scale phenomena like turbulence [48, 49, 50, 51, 52]. However, computation of derivatives using compact schemes involves a system of linear equations. This imposes constraints on the computational domain since each processor must have entire range of data in the direction of computation of derivative. Such codes require multiple collective communication calls, which in turn can make communication time quite significant [53, 54, 50] for both compact and spectral implementations.

Explicit finite difference schemes have also been extensively used for approximation of derivatives in partial differential equations (PDEs) including in massive simulations of turbulent reacting flows [17, 18]. For explicit schemes, the derivative at a grid point in the domain is approximated as a linear combination of the values at its neighboring points only. Because of this local dependence, different processors can work concurrently on different parts of the domain. However, at the processor boundaries, processors need to communicate to obtain data from the neighboring processors in order to compute the derivatives. Although these are local communications as opposed to the collective communications for compact or spectral schemes, processors still incur in overheads due to the need to communicate and synchronize at every time step to meet accuracy requirements. While simulations have been successfully done using hundreds of thousands

*Parts of this chapter are reprinted from Journal of Computational Physics, Volume 419, Komal Kumari and Diego A. Donzis, "Direct numerical simulations of turbulent flows using high-order asynchrony-tolerant schemes: Accuracy and performance", Pages 109626, 15 October 2020, with permission from Elsevier.

of processors [50, 55, 56], the synchronizations and communication overheads, irrespective of the choice of numerical methods, pose a serious challenge to scalability at extreme scales [8]. In order to overcome this bottleneck, some work has focused on relaxing the synchronization requirements among the processors and perform so-called asynchronous numerical simulations. Early work in the literature dealt with asynchronous simulations but severely limited to lower orders of accuracy and restricted to certain class of PDEs [12, 13, 14, 15]. A new and more generalized approach, extensible to arbitrarily high orders of accuracy, has been recently developed [9, 16] to derive the so-called Asynchrony-Tolerant (AT) finite-difference schemes.

However, these studies investigated numerical accuracy and stability for simplified model problems in low dimensions. The ability of these schemes to accurately simulate realistic three-dimensional turbulent flows have not been done before. Without careful assessment of the numerical and parallel performance of these schemes it is unclear whether they can indeed provide a path towards exascale simulations in future massively parallel systems. This is the main thrust of this chapter.

Specifically, in this chapter, we use AT schemes together with asynchronous algorithms described in Chapter 5 to perform, a first of a kind, asynchronous simulation of three-dimensional compressible turbulence. Our focus is on the effect of asynchrony on important turbulent characteristics such as evolution of the turbulent kinetic energy, the spectra and PDFs of velocity gradients, enstrophy and dissipation.

6.2 Governing equations and numerical schemes

The NS equations, which represent conservation of mass, momentum and energy can be written as,

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i) = 0, \tag{6.1}$$

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_i u_j) = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}(\sigma_{ij}) + \rho f_i, \tag{6.2}$$

$$\frac{\partial}{\partial t}(\rho e) + \frac{\partial}{\partial x_i}(\rho e u_i) = -p\frac{\partial u_i}{\partial x_i} + \frac{\partial}{\partial x_i}\left(k\frac{\partial T}{\partial x_i}\right) + \sigma_{ij}S_{ij}, \tag{6.3}$$

with ρ being the density, u_i the i^{th} component of velocity, e the internal energy per unit mass which depends upon temperature (T) according to the perfect gas law, k the coefficient of thermal conductivity, p the pressure, and f_i the external forcing. The viscous stress and the strain rate tensors are given, respectively, by,

$$\sigma_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\delta_{ij}\frac{\partial u_k}{\partial x_k} \right), \tag{6.4}$$

$$S_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right), \tag{6.5}$$

where the dynamic viscosity, μ, follows Sutherland viscosity law.

In this work we use fourth-order AT schemes at processor boundaries for spatial derivatives in each direction which require communication across six faces of each PE in a 3D domain. At the internal points we use standard fourth-order finite differences for spatial derivatives. Computation of mixed derivatives is challenging as they require communication across more neigboring PEs or communication of additional quantities such as gradients. Both of these are detrimental to parallel performance. As an alternative, we limit our communications per PE to six by computing mixed derivatives at the boundary points in three steps. For example, for $(\partial \left(\partial u/\partial y\right)/\partial x)$, we first compute $\partial u/\partial y$ and $\partial u/\partial x$ using AT schemes at the boundaries. Next we compute $(\partial \left(\partial u/\partial y\right)/\partial x)$ and $(\partial \left(\partial u/\partial x\right)/\partial y)$ using standard one sided finite difference schemes in x and y directiion, respectively. Since $(\partial \left(\partial u/\partial y\right)/\partial x)$ = $(\partial \left(\partial u/\partial x\right)/\partial y)$, we take the average of $\partial \left(\partial u/\partial y\right)/\partial x$ and $(\partial \left(\partial u/\partial x\right)/\partial y$ to minimize errors and use this value as the final approximation of the corresponding mixed derivatives.

6.3 Numerical results

We have implemented the synchronous and asynchronous numerical methods and algorithms described in the previous Chapters to perform DNS of decaying and forced isotropic turbulence at different Reynolds numbers to assess the effect (or lack thereof) of asynchrony. The resolution used for both synchronous and asynchronous implementations is $\eta/\Delta x \approx 0.5$ or $\kappa_{max}\eta \approx 1.5$, where

$\eta = (\nu^3/\langle\epsilon\rangle)^{1/4}$ is the Kolmogorov length scale, ν is the kinematic viscosity and $\kappa_{max} = \sqrt{2}N/3$ is the highest resolvable wave number for commonly used pseudospectral simulations in a cubic domain of length 2π on each side and N^3 points [46, 24]. This resolution has been shown to lead to well-resolved simulations for the conditions and quantities of interest presented here [57, 58]. As discussed in section 2.5, the time-step size Δt is fixed at a value that yields an initial CFL of $\mathcal{O}(0.1)$ consistent with the recommendation in [24]. To facilitate comparisons both synchronous and asynchronous simulations use the same time step. We use periodic boundary condition in all directions. The initial velocity field is a stationary state obtained by forcing the large scales of motion as done in [59, 57] and is same for both synchronous and asynchronous simulations. The important simulation parameters including resolution, percentage of points directly affected by asynchrony ($N_B\%$), Re_λ, and simulation time in terms of eddy turnover time $T_e = \mathcal{L}/u_{rms}$, where $\mathcal{L}$ is the integral length scale and u_{rms} is the root mean square of velocity fluctuations, are tabulated in Table 6.1. The level of compressibility is commonly defined in terms of the turbulent Mach number $M_t = \langle u_i u_i\rangle^{1/2}/c$, where c is the mean speed of sound, u_i is the velocity fluctuation, $\langle\cdot\rangle$ is the average computed across the entire domain and summation convention is used. For the simulations in this paper $M_t{\approx}0.3$ which represents a case where dilatational effects start becoming important [57].

For the rest of this section, we will refer the synchronous simulations using standard finite differences as SFD. The asynchronous simulations using AT schemes with random delays will be referred to as SAA and that with periodic delays will be referred to as CAA. We also have tenth-order compact schemes (C10) with third order RK scheme in time for one of the cases for comparison purposes to highlight that our finite difference simulations are comparable to the most well resolved simulations of compressible turbulence in literature [60, 50, 59, 58].

Decaying					
N^3	$N_B(\%)$	$Re_\lambda(0)$	$\eta(0)/\Delta x$	$\kappa_{max}\eta(0)$	$t/T_e(0)$
256^3	57.8	100	0.5	1.4	24
512^3	50.8	145	0.5	1.5	24

Forced					
N^3	$N_B(\%)$	Re_λ	$\eta/\Delta x$	$\kappa_{max}\eta$	t/T_e
64^3	57.8	35	0.5	1.6	10
256^3	57.8	100	0.5	1.8	19

Table 6.1: DNS parameters: number of grid points N^3, percentage of boundary points $N_B\%$, Taylor Reynolds number Re_λ, resolution $\eta/\Delta x$ and $\kappa_{max}\eta$ and normalized simulation time t/T_e. Normalization is done using the initial values $(Re_\lambda(0), \eta(0), T_e(0))$ for the decay cases and using average computed over stationary state for the forced case.

6.3.1 Decaying turbulence

6.3.1.1 Low order statistics in physical space

It is important for any numerical scheme to accurately capture the large scale behavior of the system. An important and widely studied [61, 62] large scale quantity in fluid turbulence is the mean turbulent kinetic energy per unit mass defined as,

$$K = \frac{1}{2}\langle \rho u_i u_i \rangle. \tag{6.6}$$

In the absence of energy input to the system, K decays in time as shown in Fig. 6.1(a, c), where K is normalized by its initial value K_0 and time is normalized by initial eddy turnover time, $T_e(0) = \mathcal{L}/u_{rms}$. After an initial transient, the decay obeys a power-law in time observed as a straight line on a log-log scale in Fig. 6.1(a, c). The decay exponent is seen to be consistent with that found in the literature for similar conditions [62, 63]. The excellent agreement between SFD, CAA and SAA in Fig. 6.1(a, c) at all times shows that asynchronous implementations have accuracy comparable to SFD.

The rate at which kinetic energy is dissipated is given by $\langle \epsilon \rangle = 2\langle \sigma_{ij} S_{ij} \rangle$. Because most of

the contribution to dissipation comes from small scales (or high wavenumbers) it is therefore sensitive to how accurately high wavenumbers are resolved by the numerical methods. The decay of $\langle\epsilon\rangle$ (normalized by its initial value) is shown in Fig. 6.1(b, d) for SFD, CAA and SAA with no observable differences. Thus, we find that the asynchronous implementations are able to capture the evolution of low-order large and small scale quantities with accuracy comparable to the standard finite differences. Also shown in Fig. 6.1(a, b) is the evolution obtained for C10 (magenta line), which is identical to the evolution obtained for both asynchronous and synchronous finite difference.

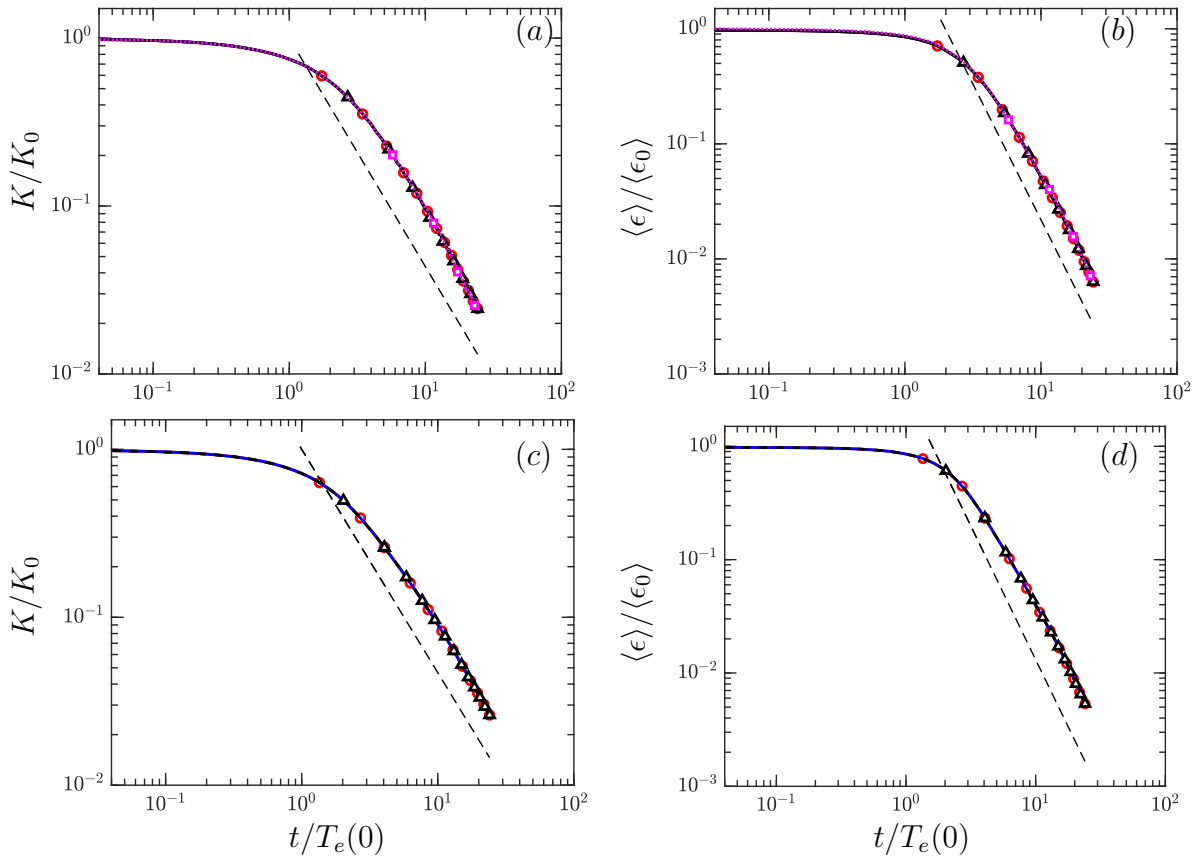

Figure 6.1: Evolution of space averaged turbulent kinetic energy normalized by the initial turbulent kinetic energy K_0 (left) and evolution of space averaged dissipation rate normalized by the initial dissipation rate ϵ_0 (right) for $Re_\lambda(0) \approx 100$ (a, b) and $Re_\lambda(0) \approx 145$ (c, d). Different lines are: SFD (red-circle), CAA (black-triangle) and SAA (blue) both with $L = 3$. The black-dashed line corresponds to $K/K_0 \propto (t/T_e(0))^{-1.4}$ in (a) and $\langle\epsilon\rangle/\langle\epsilon_0\rangle \propto (t/T_e(0))^{-2.4}$ in (b). Magenta line in (a, b) is C10.

6.3.1.2 *Low order statistics in spectral space*

Fluid turbulence comprises a wide range of interacting scales [64]. The energy distribution across these scales is characterized by the energy spectrum, which according to Kolmogorov self-similarity hypothesis (K41) [3] is given by,

$$E(\kappa) = C\langle\epsilon\rangle^{2/3}\kappa^{-5/3}f(\kappa\eta), \tag{6.7}$$

where C is the Kolmogorov constant, κ is the wavenumber and $\eta = (\nu^3/\langle\epsilon\rangle)^{1/4}$ is the Kolmogorov length scale [3] and $f(\kappa\eta)$ is a universal function. This has been compared against simulations and experiments extensively and shown to be a good representation of the spectrum across different flows and Reynolds numbers for incompressible [65] and compressible flows [59, 66, 61] at low M_t. In the so-called inertial range ($1/\mathcal{L} \ll k \ll 1/\eta$), $f(k\eta) = 1$ and the classical $5/3$ scaling for the energy spectrum [67, 65, 5] can be seen as a flat region in the compensated energy spectrum,

$$\frac{E(\kappa)}{\langle\epsilon\rangle^{2/3}\kappa^{-5/3}} = C, \tag{6.8}$$

which becomes wider with an increase in Reynolds number. The height of this flat region gives the Kolmogorov constant which has been estimated to be $C = 1.6$ from simulations and experiments in incompressible turbulence [68, 69, 70]. This value has been shown to be consistent for compressible simulations [59]. At high wavenumbers, $f(k\eta)$ is a decaying exponential [71, 72, 73, 74] which may retain a weak Reynolds number effect at very high wavenumbers [75].

In Fig. 6.2(a, c) we show the compensated energy spectrum at $t/T_e(0) \approx 1$ and 4 for $Re_\lambda(0) \approx 100$ and 145 for SFD, CAA and SAA implementations. A plateau in this normalization corresponding to the inertial range can be seen at short times over a narrow range of scales. Because of the decrease in Re_λ with time due to the decay, the inertial range becomes less prominent at later times. We also see that the high wavenumbers are universal as expected from Eq. (6.7). Both SAA and CAA retain the universality at small scales and accurately capture the evolution of inertial and

large scales. We see a virtually perfect agreement even at the smallest scales (inset inFig. 6.2(a, c)) for CAA as well as SAA with SFD. Moreover, the energy spectrum is also identical to the one obtained with C10 (magenta line in Fig. 6.2(a)) from some of the most well-resolved simulation of compressible turbulence [59, 57].

Similar to the energy spectrum, K41 also predicts a scaling in the inertial range for pressure fluctuations [1, 76] which reads,

$$E_p(\kappa) = C_p \langle \epsilon \rangle^{4/3} \kappa^{-7/3}. \tag{6.9}$$

The inertial range can be identified as the plateau in the compensated pressure spectrum plot, if Re_λ is high enough. Since $M_t \approx 0.3$ for our simulation is fairly low, the pressure spectrum should be similar to the incompressible spectrum [59, 76]. This is indeed observed in Fig. 6.2(b, d) for $Re_\lambda(0) \approx 100$ and 145 at $t/T_e(0) \approx 1$ and 4 for the universal part of the spectrum. A horizontal dashed line at $C_p = 8$ is also included for reference obtained from incompressible flows [76]. These spectra are consistent with those in the literature at similar conditions [59] with a collapse at the high wave-numbers similar to the energy spectrum. The data for CAA and SAA agree closely with that for SFD at both times for both Re_λ. However, for $k\eta \geq 1.5$, SAA spectrum has a small pileup at the high wavenumbers. This difference in the spectrum for CAA and SAA can be attributed to the difference in the nature of delays which for the former is deterministic and random for the latter. The randomness associated with SAA can lead to numerical errors that are absent in CAA and can cause a small pileup of energy at the high wavenumbers as seen in the pressure spectrum in Fig. 6.2(b) for SAA. The differences in Fig. 6.2(b) are magnified because of the prefactor $k^{7/3}$ but they are concentrated only in a few wavenumbers and represent an extremely small contribution to *e.g.*, pressure variance. We have also performed simulations at higher $M_t \approx 0.6$ and found that this small pileup disappears. Thus, this seems to be a low-M_t effect which can be explained by noting that as M_t increases, there is stronger interaction between the so-called solenoidal and dilatational velocity components [77] which can help mix these already

small perturbations at PE boundaries for SAA. To illustrate this, we can decompose the pressure field into solenoidal (p_s) and dilatational ($p_d = p - p_s$) components as done in [78, 59, 57]. Here the fluctuations in solenoidal pressure are written in terms of local enstrophy ($\Omega = \omega_i\omega_i$) and dissipation ($\epsilon$) as ($\nabla^2(p_s/\rho) = (\Omega - \epsilon/\nu)/2$). These components of pressure spectrum at two different M_t values are shown in Fig. 6.3. We note that the small pile-up at high wavenumbers for SAA is observed in the lower M_t case only for the dilatational component. However, the solenoidal components exhibit no such pile-up for both CAA and SAA irrespective of the turbulent Mach number.

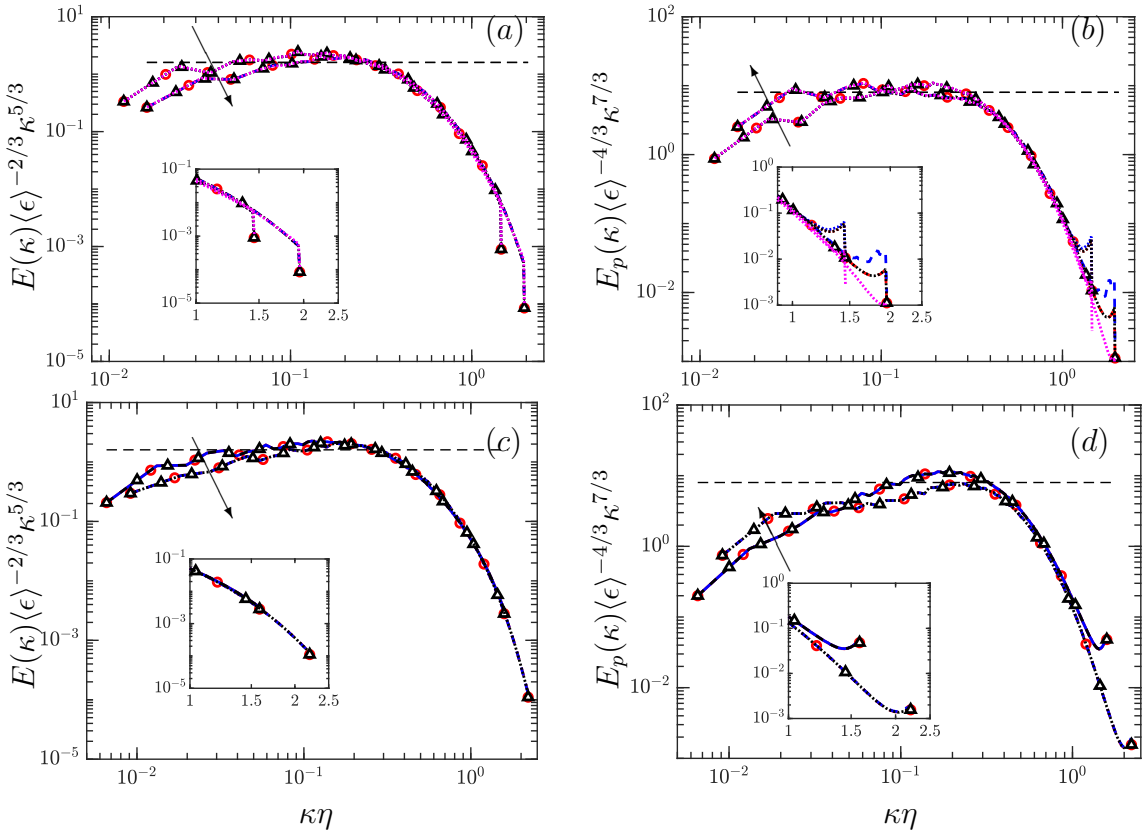

Figure 6.2: Compensated energy spectrum (left) and compensated pressure spectrum (right) for $Re_\lambda(0) \approx 100$ (a, b) and $Re_\lambda(0) \approx 145$ (c, d) at $t/T_e(0) \approx 1$ and 4. Different lines are: SFD (red-circle), CAA (black-triangle) and SAA (blue) with $L = 3$. The arrow denotes increasing time. Magenta line in (a, b) is C10.

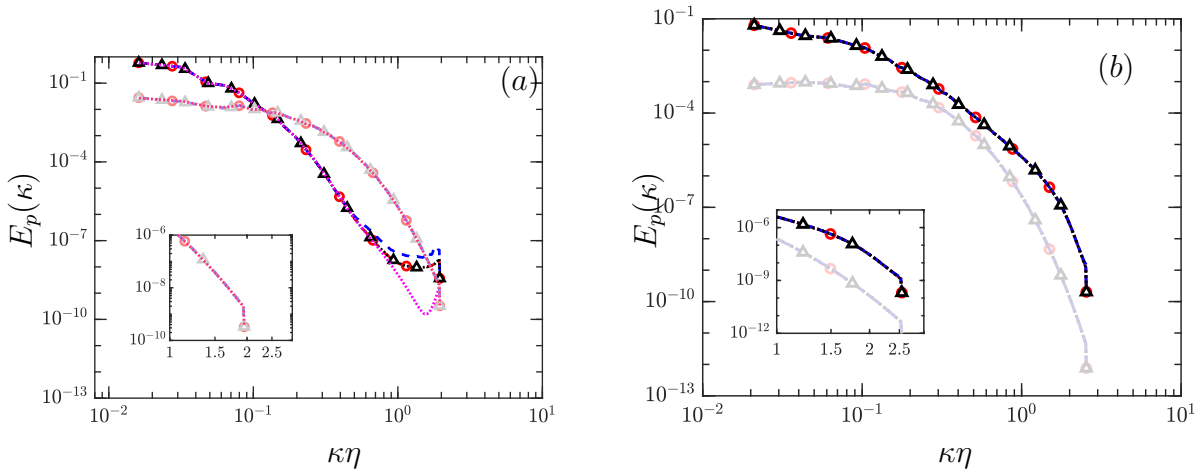

Figure 6.3: Solenoidal (faded lines) and dilatational (dark lines) components of pressure spectrum for $Re_\lambda(0) \approx 100$ and (a) $M_t(0) \approx 0.3$ and (b) $M_t(0) \approx 0.6$ at $t/T_e(0) \approx 4$. Inset: (a) solenoidal spectrum at large wavenumbers and (b) both solenoidal (faded) and dialatational spectrum at large wavenumbers. Colors of the lines are same as in Fig. 6.2

A general conclusion one can draw from both energy and pressure spectrum plots, is that the dynamics of the flow at the scales of interest is accurately captured despite asynchrony even though there are some very small deviations at the high wavenumbers in the pressure spectrum for SAA. Furthermore, we see from Fig. 6.2(b) that the pressure spectrum for SFD itself is not identical to the spectrum obtained for C10 at higher wavenumbers. Thus, it is not unexpected that asynchronous schemes present a different behavior at high wavenumbers. The errors in SAA, though already very small, can be mitigated if higher order schemes or higher resolution is used. As an example, in Fig. 6.4(b), the compensated pressure spectrum is shown for $Re_\lambda(0) \approx 100$ at $t/T_e(0) \approx 4$ using fourth and sixth order AT scheme (included in the appendix) for SAA. While the SAA with fourth-order AT scheme (solid blue) peels off at $\kappa\eta \approx 1.5$, SAA with sixth-order AT scheme (faded-blue square), follows the SFD spectrum till the highest $\kappa\eta$.

6.3.1.3 Statistics of velocity gradients

An important feature of 3D turbulence is the generation of vortical motions, often quantified with the so-called enstrophy ($\Omega = \langle \omega_i \omega_i \rangle$, where $\boldsymbol{\omega} = \nabla \times \boldsymbol{u}$ is the vorticity vector). A normalized metric for the production of enstrophy, which is also representative of the non-linear transfer of

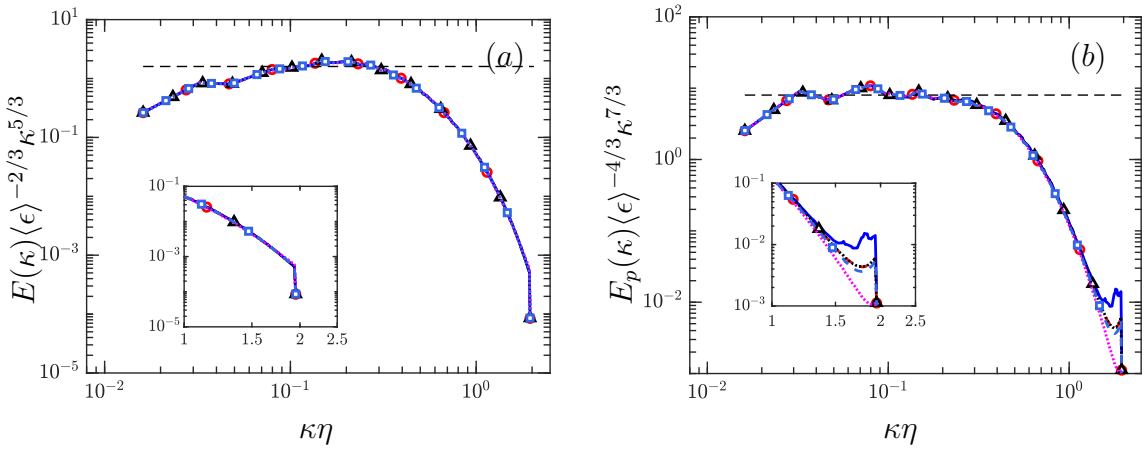

Figure 6.4: (a) Compensated energy spectrum and (b) compensated pressure spectrum for $Re_\lambda(0) \approx 100$ at $t/T_e(0) \approx 4$. Faded dashed-blue line with squares is the sixth-order asynchronous scheme with random delays and solid blue line is fourth-order AT scheme with random delays. Rest of the lines are same as in Fig. 6.2. Insets zoom in on high wavenumbers.

energy from large scales to small scales, is the skewness of the longitudinal velocity gradient, $S = \langle(\partial u_1/\partial x_1)^3\rangle/(\langle(\partial u_1/\partial x_1)^2\rangle)^{3/2}$ [1, 65, 79]. The negative of the skewness $(-S)$ is constant at about ~ 0.5 as long as the Reynolds number is not too low. This has been extensively documented in experiments and numerical simulations [80, 65, 81, 5]. In Fig. 6.5(a, c) we show the time evolution of $-S$ for initial Re_λ of 100 and 145, respectively. We see that $-S$ is close to 0.5 and this is consistent for SFD, CAA and SAA, with some small differences at later times. Despite odd-order moments being more sensitive to resolution [82] and susceptible to numerical errors, we see that the asynchronous algorithms capture skewness very well and close to the skewness computed using C10.

Another intrinsic characteristic of turbulent flows is the phenomena of intermittency which is a tendency to have localized events of fluctuations that are orders of magnitude larger than the mean [72, 65, 79, 83, 84, 85]. These events add to the complexity of the turbulent flows, specifically at the smallest scales. One way to quantify this phenomena is through the moments of velocity gradients as most of their contribution stems from the small scales and it is thus an excellent quantity to check small scale resolution. These moments transition from Gaussian to anomalous as Reynolds number increases [86, 87, 88]. In Fig. 6.5(b, d) we show the normalized fourth-order

moment or flatness ($F = \langle(\partial u_1/\partial x_1)^4\rangle/(\langle(\partial u_1/\partial x_1)^2\rangle)^2$) of the longitudinal velocity gradient. The flatness is close to 6 [80, 65] at initial times and tends to decrease because of decrease in Reynolds number for decaying turbulence. We see an excellent agreement between synchronous and both the asynchronous simulations with no observable differences from C10. Even though the computation of the gradient $\partial u_1/\partial x_1$ is directly affected by asynchrony, the higher order moments of the same exhibit trends similar to SFD and C10.

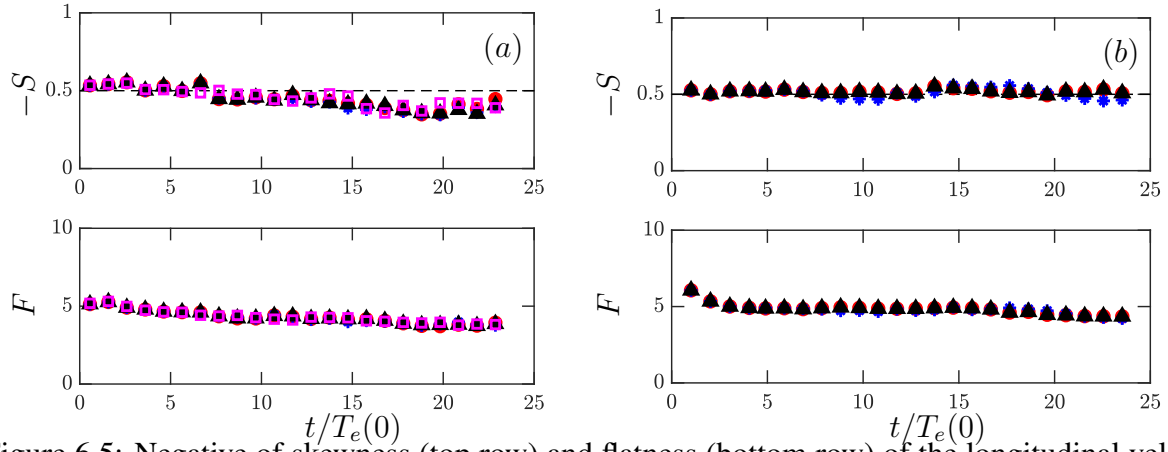

Figure 6.5: Negative of skewness (top row) and flatness (bottom row) of the longitudinal velocity gradient vs. normalized time for (a) $Re_\lambda(0) \approx 100$ and (b) $Re_\lambda(0) \approx 145$. Different symbols are: SFD (red-circle), CAA (black-triangle) and SAA (blue asterik) with $L = 3$. The dashed black line indicates skewness of 0.5 and magenta squares in $(a, c)$are C10.

6.3.1.4 Instantaneous enstrophy field

The average quantities discussed in the sections show good agreement between the asynchronous and synchronous simulations. A stricter test of accuracy would comprise the instantaneous flow fields which can potentially show some differences because of different truncation errors for different schemes in the computation of derivatives at the boundaries. As argued above, enstrophy is known to be sensitive to small scale resolution and is highly intermittent [82, 89] and thus provides a stringent test of the numerical performance of schemes. In Fig. 6.6 we show the contours of the enstrophy normalized by its mean ($\Omega/\langle\Omega\rangle$) in the yz plane at $x = \pi$. Qualitatively,

all the large and small structures look identical for SFD, CAA and SAA. In particular, a concern with asynchronous schemes is the behavior close to the processor boundaries. If we closely look along these PE boundaries (faded lines in Fig. 6.6) there are no perceptible differences between enstrophy contours for SAA, CAA and SFD. Moreover, even complex structures spanning across multiple PE boundaries, for example, inside black circle in Fig. 6.6, is consistent for all the three cases. Besides some very small but not apparent localized differences in the intensity of enstrophy for SAA, the asynchronous algorithms accurately resolve the highly intermittent instantaneous enstrophy field. The instantaneous dissipation field (not shown here) exhibits similar behavior and is captured accurately.

6.3.2 Forced Turbulence

In the preceding section we focused on the DNS of decay of stationary state initial velocity field and observed a close agreement between the synchronous and asynchronous numerical simulations. In this section we discuss the effect of asynchrony on forced turbulence. Here, energy is injected at the large scales, or wavenumbers (κ) in a spherical shell of radius κ_f, where $\kappa \leq \kappa_f$, $(\kappa_f = 3)$, through the term f in the momentum equation (Eq. (6.2)). The details of the stochastic forcing implemented can be found in [90] and has been extensively used in [91, 59, 57, 92] for compressible turbulence. Through the non-linear interactions this injected energy cascades down to the inertial and small scales, where it is dissipated into internal energy by the action of viscosity. One can derive the evolution equation of the mean turbulent kinetic energy (K) by multiplying Eq. (6.2) by u_i and taking the mean, which reads as

$$\frac{dK}{dt} = \langle p'\theta' \rangle - \langle \epsilon \rangle + \langle f_i u_i \rangle \tag{6.10}$$

where $\theta = \partial u_i / \partial x_i$ is the dilatation, $\langle p'\theta' \rangle$ is the mean pressure-dilatation correlation and the mean dissipation $\langle \epsilon \rangle$. The external forcing f acts against the dissipative effect of viscosity to sustain turbulent fluctuations. We can also write the equation of the mean internal energy ($\langle e \rangle$)

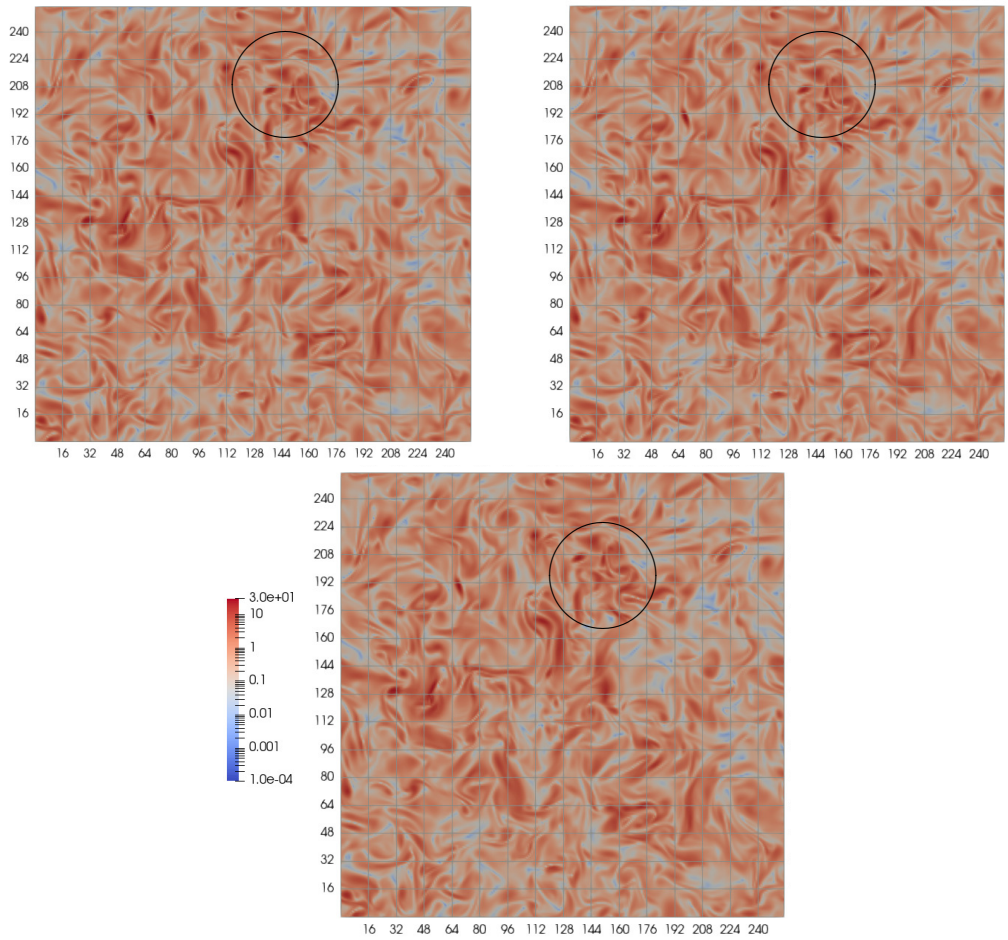

Figure 6.6: Normalized instantaneous enstrophy $(\Omega/\langle\Omega\rangle)$ field at $t/T_e(0) \approx 4$ for (a) SFD (b) CAA ($L = 3$) and (c) SAA ($L = 3$) in the yz plane at $x = \pi$ for $Re_\lambda(0) \approx 100$. The faded lines represent processor boundaries

from Eq. (6.3) as

$$\frac{d\langle e\rangle}{dt} = -\langle p'\theta'\rangle - \langle\epsilon\rangle. \tag{6.11}$$

The pressure-dilatation and viscous dissipation are responsible for the exchange between kinetic and internal energy. While the former is a bi-directional exchange depending upon the value of turbulent Mach number, M_t [59, 57], the latter converts kinetic energy into internal energy irreversibly. Since no external sink is added to the energy equation, the internal energy of the system always increases. The time evolution of K and $\langle\epsilon\rangle$, normalized by their initial values, is plotted in Fig. 6.7. We can see that K increases initially, because of the input of energy due to forcing

at large scales. Once the cascade develops and transfers energy to the smallest dissipative scales, the mean kinetic energy starts to decrease. At the same time, dissipation also increases initially, after an initial lag, until it reaches an equilibrium. At this point the rate of energy input is equal to rate of dissipation and a quasi-stationary state is reached [66]. In Fig. 6.7, this state is achieved at $t/T_e \approx 5$ for $Re_\lambda \approx 35$ (a), and $t/T_e \approx 6$ for $Re_\lambda \approx 100$ (b), where T_e is the average eddy turnover time. The average eddy turnover time is computed form the average taken at ten checkpoints from $t/T_e \geq 5$ for $Re_\lambda \approx 35$ and at fifteen checkpoints from $t/T_e \geq 6$ for $Re_\lambda \approx 100$. The net increase in the total energy is, at this point, equal to the increase in the internal energy. As in the case of decaying turbulence, we see a good agreement between the synchronous and asynchronous simulations in Fig. 6.7 for both high and low Re_λ.

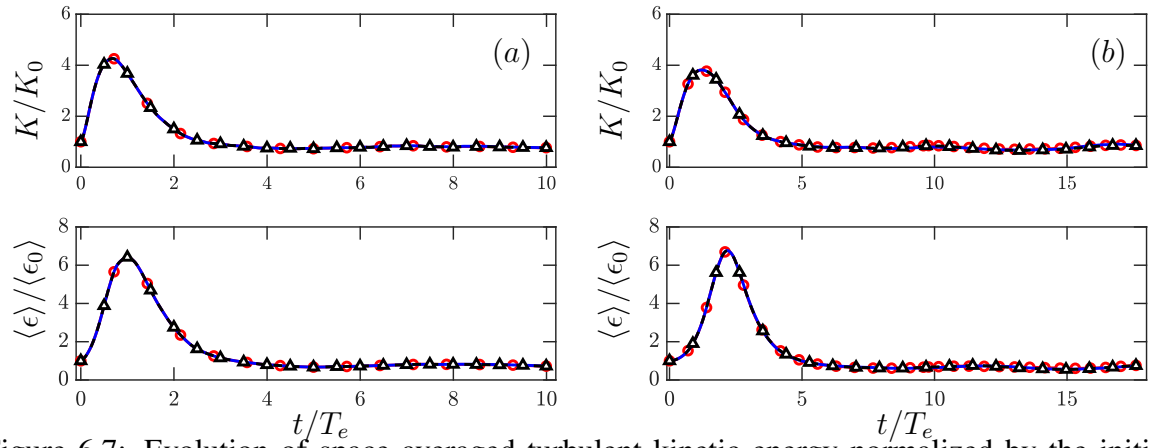

Figure 6.7: Evolution of space averaged turbulent kinetic energy normalized by the initial turbulent kinetic energy K_0 (top row) and space averaged dissipation rate normalized by the initial dissipation rate ϵ_0 (bottom row) for(a) $Re_\lambda \approx 35$ and (b) $Re_\lambda \approx 100$. Different lines are: SFD (red-circle), CAA (black-triangle) and SAA (blue) with maximum allowed delay level of $L = 3$. Time is normalized by the average eddy turnover time (T_e).

We are also interested in the energy and pressure spectrum, which are plotted in Fig. 6.8. These spectra are the average taken for ten and fifteen checkpoints, respectively for $Re_\lambda \approx 35$ and 100, after the quasi-stationary state is reached. These energy spectra are shown in Fig. 6.8(a) where we see that both CAA and SAA simulations are accurately resolved, with good collapse at all

wavenumbers. For the pressure spectrum in Fig. 6.8(b), the CAA and SAA agree equally well with the SFD, unlike the decaying case where small errors were seen at the large wavenumbers for SAA. These spectra are also consistent with [59] at similar conditions. For a quantitative metric of difference between the asynchronous and synchronous spectra, we look at the relative error defined as

$$e(\kappa^*) = \max_{\kappa} \frac{|E_{\text{CAA or SAA}}(\kappa) - E_{\text{SFD}}(\kappa)|}{E_{\text{SFD}}(\kappa)}, \tag{6.12}$$

where κ^* is the wavenumber at which maximum error occurs. This error is tabulated in Table 6.2 for both energy and pressure spectrum and is within 95% confidence interval for the standard synchronous case. We note that the error is maximum at scales larger than the dissipative scales ($\kappa^*\eta < 1$).

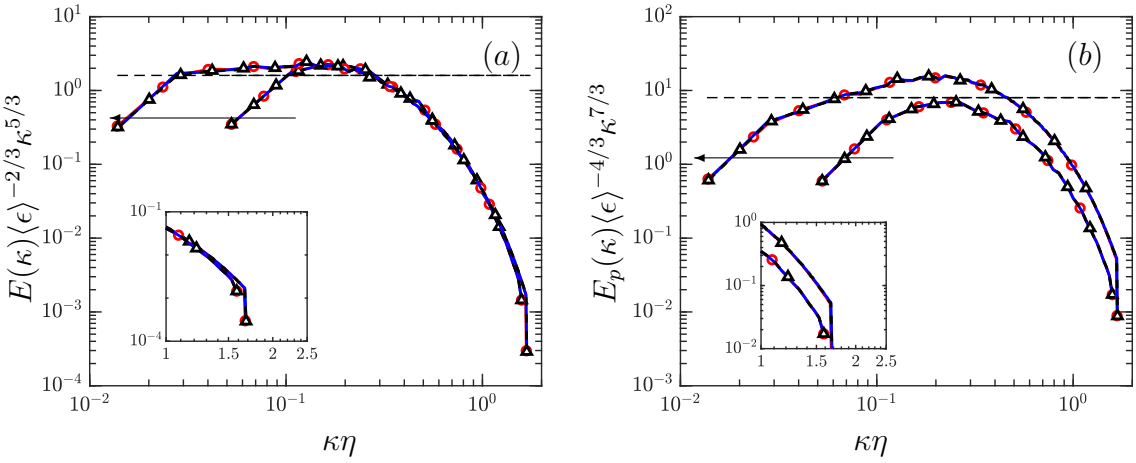

Figure 6.8: (a) Compensated energy spectrum and (b) compensated pressure spectrum for $Re_\lambda \approx 35$ and $Re_\lambda \approx 100$. Different lines are: SFD (red-circle), CAA (black-triangle) with $L = 3$ and SAA (blue) with $L = 2$. Arrow indicated increasing Re_λ. Dashed line in is Kolmogorov constant $C = 1.6$ in (a) and $C_p = 8$ in (b).

The higher order moments of the longitudinal velocity gradients are also plotted in Fig. 6.9. We see that $-S$ fluctuates around 0.5 [88] and the values are fairly consistent for SAA, CAA and SFD. Even better agreement is seen for F in Fig. 6.9(c, d), with value close to 6 for $Re_\lambda \approx 100$ and smaller for $Re_\lambda \approx 35$ [88]. The average of value skewness and flatness in the quasi-stationaly

Energy Spectrum				
	$R_\lambda \approx 35$		$R_\lambda \approx 100$	
	$e(\kappa^*)$	$\kappa^*\eta$	$e(\kappa^*)$	$\kappa^*\eta$
SAA	1.87%	0.32	1.69%	0.08
CAA	0.17%	0.42	0.97%	0.08

Pressure Spectrum				
	$R_\lambda \approx 35$		$R_\lambda \approx 100$	
	$e(\kappa^*)$	$\kappa^*\eta$	$e(\kappa^*)$	$\kappa^*\eta$
SAA	3.06%	0.42	2.37%	0.08
CAA	0.32%	0.42	3.14%	0.08

Table 6.2: Maximum relative error in enegry and pressure spectra and the wavenumber at which it occurs for both the Reynolds numbers.

state is tabulated in Table 6.3. Also included in this table is the relative error for both CAA and SAA computed with respect to SFD. In general the error is small for CAA as compared to SAA and for both the cases this relative errors are within 95% confidence interval of the synchronous simlation.

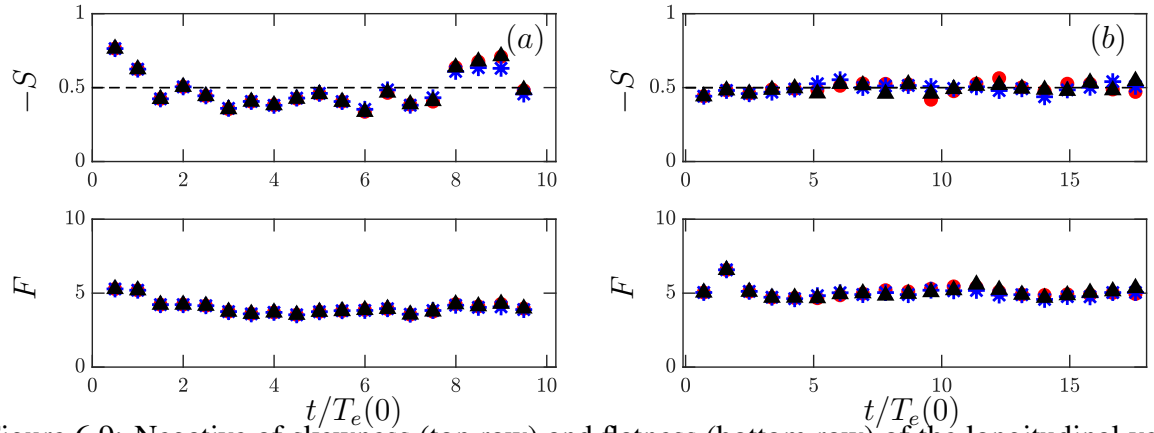

Figure 6.9: Negative of skewness (top row) and flatness (bottom row) of the longitudinal velocity gradient vs. normalized time for (a) $Re_\lambda \approx 35$ and (b) $Re_\lambda \approx 100$. Different symbols are: SFD (red-circle), CAA (black-triangle) with $L = 2$ and SAA (blue asterik) with $L = 2$. The dashed black line indicates skewness of 0.5.

$Skewness\ (S)$				
	$R_\lambda \approx 35$		$R_\lambda \approx 100$	
	S	$Relative\ error$	S	$Relative\ Error$
SFD	-0.489	-	-0.503	-
SAA	-0.480	-1.93%	-0.504	0.18%
CAA	-0.491	0.36%	-0.499	072%

$Flatness\ (F)$				
	$R_\lambda \approx 35$		$R_\lambda \approx 100$	
	F	$Relative\ error$	F	$Relative\ Error$
SFD	3.887	-	5.041	-
SAA	3.859	0.73%	4.946	1.89%
CAA	3.893	0.14%	5.031	0.21%

Table 6.3: Skewness and flatness averaged over quasi-stationary states and the relative error computed with respect to the SFD value for both R_λ.

Finally we look at the PDF of enstrophy density (Ω) and dissipation rate (ϵ) [82, 78, 89]. Both dissipation and enstrophy are crucial in the understanding of the small-scale motions [65] and are highly intermittent. Because of extreme events in ϵ and ω, the corresponding PDFs of the normalized quantities, $\epsilon/\langle\epsilon\rangle$ and $\langle\omega\rangle$, are characterized by wide tails. The PDF of $\epsilon/\langle\epsilon\rangle$ and $\Omega/\langle\Omega\rangle$, averaged over checkpoints as in case of averaged spectrum, are plotted in Fig. 6.10(a, b). We can clearly see the tails of both the PDFs become wider as Reynolds number is increased from 38 to 100. This suggests that the propensity of events that are an order of magnitude more intense than the mean, increases with the Reynolds number [82, 89]. Furthermore, we also observe that the tails for the PDF of $\Omega/\langle\Omega\rangle$ in Fig. 6.10(b) are wider than the tails for PDF of $\epsilon/\langle\epsilon\rangle$ in Fig. 6.10(a). This implies that enstrophy is more intermittent than dissipation and this has been consistently established in several past studies [80, 93, 65, 94, 82]. These features of the PDF are captured well by both the asynchronous algorithms with very small differences at the far tails. Thus, the AT schemes accurately resolve even the finest scales of turbulence including very highly intermittent events in dissipation and enstrophy.

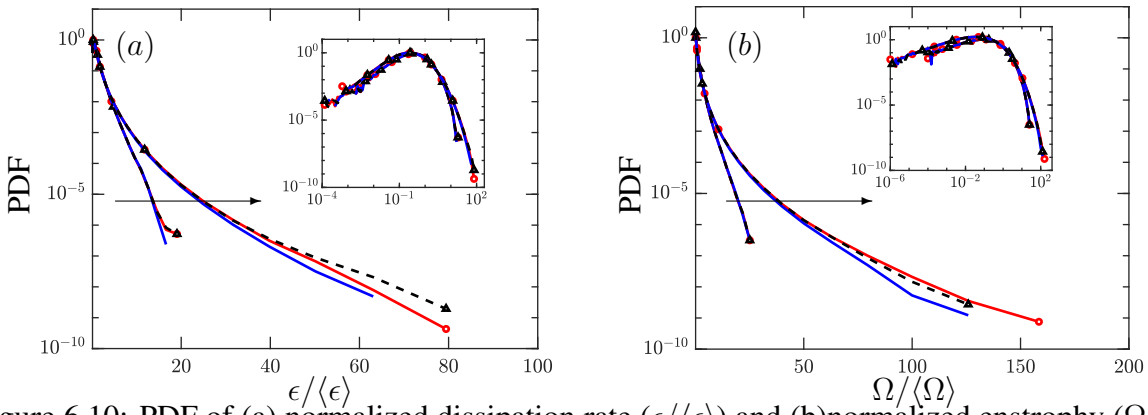

Figure 6.10: PDF of (a) normalized dissipation rate ($\epsilon/\langle\epsilon\rangle$) and (b)normalized enstrophy ($\Omega/\langle\Omega\rangle$) in log-linear scale. The insets are the same PDFs in log-log scale. Different lines are: SFD (solid red with circle), CAA (dashed black with triangle) and SAA (solid blue) with $L = 3$. The arrow indicates increasing Re_λ

6.3.3 Computational performance

The preceding sections demonstrated the ability of asynchronous algorithms in resolving important physical characteristics of turbulent flows including instantaneous field and high order statistics. Now we show that the asynchronous simulations are computationally more efficient than their synchronous counterpart. To study this we look at so-called strong and weak scaling of the solver. In the former the problem size remains fixed, while in the latter the computational work is kept constant. Ideally, for a fixed problem size, the computation time should decrease linearly on increasing the processor count. However, with increasing number of processors, the necessary communications and synchronizations increase the communication time until it eventually dominates the total execution time. This is essentially the communication bottleneck and is expected to be a major challenge to scalability [8, 50, 9, 16]. In Fig. 6.11(a), we have plotted the total execution time for synchronous and asynchronous implementations for our compressible flow solver. These times are an average of five runs of 6000 steps each and a maximum allowed delay of $L = 4$ for both SAA and CAA. For reference we have also plotted ideal scaling as a dashed black line. In Fig. 6.11(a) clear departures from ideal scaling are seen at $P = 512$ for SFD. This, as is evident from Fig. 6.11(b), happens because the percentage of communication time (dashed

red) grows with processor count (P) until it becomes comparable to the computation time. On the other hand, both CAA and SAA (black and blue lines) are close to the ideal scaling in Fig. 6.11(a) for a much larger processor count of $P = 8192$. The improved scaling is attributed to the fact that only a small percentage ($\sim 20\%$) of the overall time is spent on communications. This percentage (Fig. 6.11(b)) remains fairly constant on increasing the number of processors for the asynchronous implementations, whereas grows to larger than 50% for the synchronous case.

Next we look at the weak scaling, where ideally because of fixed computational work, the time per step should remain constant on increasing the processor count. The time per step for a computational load of $N^3/P = 2048$ is plotted in Fig. 6.12. For the synchronous case, this time per step scaling grows by a factor of 60% because of increase in communication and synchronization ovearheads at large core count ($P = 262,144$). This can only be expected to get worse at much higher levels of parallelism expected in exascale machines. On the other hand, the asynchronous algorithms show improved scaling, with a much smaller 21% increase in time per step for SAA and only 14% increase for CAA on increasing the number of processors from $P = 128$ to $P = 262,144$. This also implies that reduction in the overall volume of communication (CAA) at extreme scales provides more improvement in scaling than reducing forced synchronizations (SAA).

Both weak and strong scaling analysis lead us to the same conclusion that the asynchronous algorithms remove synchronization and communication overheads, leading to an effective overlap between communications and computations and, consequently, an improvement in scaling.

6.4 Conclusions

Numerical simulations of PDEs, governing complex natural and engineering phenomena, using standard numerical methods on parallel supercomputers, require PEs to communicate and synchronize frequently to ensure accuracy. This synchronization and communication cost and the resulting PE idling grows with increasing levels of parallelism and presents a major challenge to scalability to exascale computing. In order to mitigate this bottleneck, these constraints were relaxed at a mathematical level to derive the so-called Asynchrony-Tolerant (AT) of arbitrary order of accu-

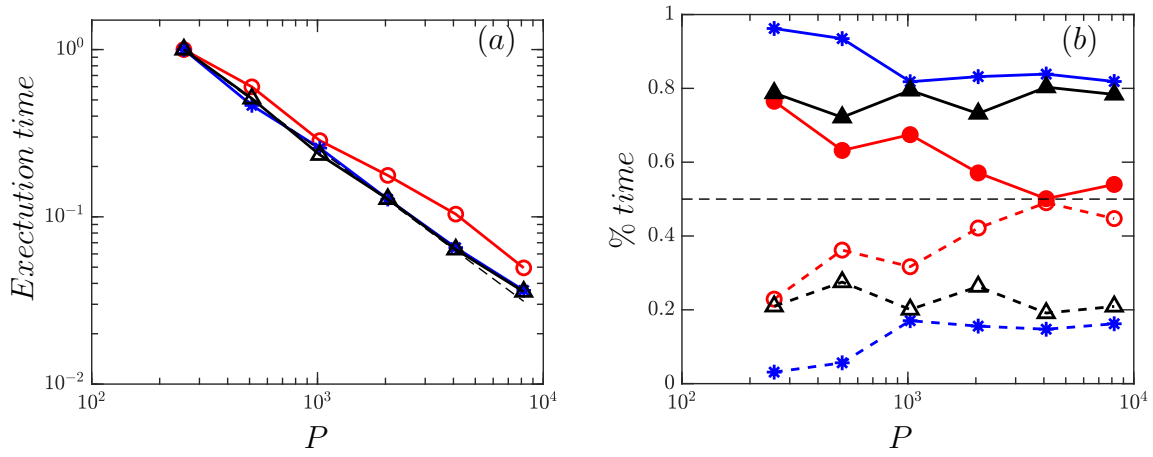

Figure 6.11: Strong scaling for $N = 128$. (a): Total execution time normalized by the execution time for $P = 256$. (b): Computation time and communication time as a percentage of the total execution time. Different lines are: SFD (red), CAA (black) and SAA (blue), dotted black in (a) is ideal scaling and in (b) is 50% of total time. Dashed lines with hollow symbols in (b) is communication time and solid lines with solid symbols is computation time.

racy in [16]. By allowing for asynchrony, these AT schemes can be used to allow computations to proceed in a PE without having to wait for updated values at the boundaries, thus removing synchronizations.

In this chapter we presented, first of a kind, asynchronous simulations of compressible turbulence using high-order Asynchrony-Tolerant (AT) schemes to study the effect of asynchrony on the physics of turbulence at different scales and on the computational performance of the solver. The numerical properties of these schemes, including stability analysis was presented in Chapter 2 and Chapter 3. The details of the asynchronous solver and two ways to allow for asynchrony, namely, communication avoiding and synchronization avoiding algorithms (CAA and SAA, respectively) that were discussed in Chapter 5.

The aforementioned asynchronous algorithms are used for the simulation of decaying and solenoidally forced turbulence. Important low and high order statistics obtained for the asynchronous algorithms are compared with that for the standard synchronous finite differences (SFD) at the same resolution and order and also with high-order compact difference schemes (C10). We found excellent agreement between SFD and CAA for the time evolution of turbulent kinetic en-

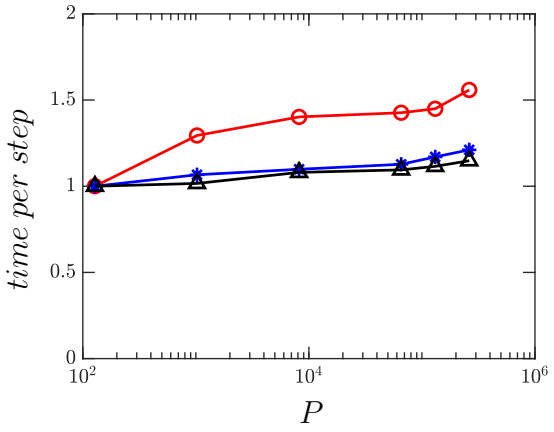

Figure 6.12: Weak scaling: time per step for $N^3/P = 2048$ normalized by time per step for $P = 128$. Different lines are: SFD (red), CAA (black) and SAA (blue).

ergy and dissipation for both decaying and forced turbulence, including the transients for the latter. The distribution of energy at different scales as shown by the velocity and the pressure spectrum is resolved by CAA with same level of accuracy as SFD and C10, even at the largest wavenumbers. Higher-order moments of longitudinal velocity gradient, including skewness and flatness, also showed excellent agreement between SFD, C10 and CAA. No observable differences are seen in the complex distribution of the contours of instantaneous enstrophy field. The PDF of highly intermittent quantities such as dissipation and enstrophy, that are also very sensitive to the accuracy of numerical schemes and small scale resolution, are also captured well by CAA, with some statistical differences at extreme tails.

For SAA as well, the evolution of turbulent kinetic energy and dissipation for decaying and forced turbulence (including transients) is in excellent agreement with SFD and C10. While no differences were seen for the energy spectrum, the pressure spectrum showed some small differences at high wavenumbers. However, these differences do not affect the dynamics of the scales of interest and, as we show, are easily mitigated if higher order AT schemes are used. Similar to CAA, the instantaneous enstrophy field, the flatness and skewness of longitudinal velocity gradient and the PDF of dissipation and enstrophy is shown to be in excellent agreement with the synchronous simulations.

Taken together, the results obtained for both CAA and SAA and their comparison with synchronous simulations (SFD and C10), clearly show that the physics of turbulence even at the finest scales is resolved accurately by the asynchronous algorithms, even though more than 50% of total gridpoints are affected directly by asynchrony.

We also presented the effect of asynchrony on computational performance. In particular, both strong and weak scaling results showed a near ideal scaling for the asynchronous algorithms and significant departures from the same for synchronous case. This improved scaling can be traced back to a significant reduction in communications (CAA) and synchronizations (SAA), resulting in an overall lower fraction of communication compared to computation for both CAA and SAA. We also observed that at very high processor count $(P = 262144)$, the reduction in overall volume of communications (CAA) is more effective in improving the scaling than relaxing explicit synchronization (SAA). This improvement in scaling is expected to be more consequential as we increase the problem size and processor count to levels anticipated on exascale machines.

In conclusion, asynchronous simulations can accurately resolve the physics at all scales and provide better parallel performance as problem size increases. Thus, asynchronous computing presents an effective alternative to standard computing approaches for simulation of turbulence and other complex phenomena at unprecedented levels of physical realism on the next generation exascale machines.

7. ASYNCHRONOUS SIMULATIONS OF REACTING FLOWS

7.1 Introduction

Over the past few decades, the advancements in high performance computation resources have provided a significant boost to high-fidelity simulations of turbulent reacting flows [95, 96, 97]. The so-called Direct Numerical Simulations (DNS) [4] with detailed chemistry provide invaluable insights into the fundamental details of fine-grained interactions between chemistry and turbulence [98, 99, 100]. In the DNS approach for reacting flows the time-dependent governing Navier-Stokes, energy and species continuity equations are solved with high accuracy, and all dynamically relevant ranges of unsteady spatial and temporal scales are numerically resolved. This stringent resolution criteria imposes prohibitive computational cost and consequently limits the attainable parametric range of relevant DNS. Even for moderate Reynolds number or number of transported species, for example, DNS requires massive supercomputers with hundreds of thousands of processing elements (PEs) working concurrently. In some of the most well resolved DNS of such complex flows [101, 18, 102, 100, 103, 25, 104, 105, 106, 107, 38, 108] high-order finite difference schemes have been used extensively to approximate the spatial derivatives. The parallel efficiency of these schemes is high since they use local stencils extending only to the nearest grid-points to approximate the derivatives. However, in a data parallel decomposition of the domain, where multiple PEs are working on different parts of the computational domain in parallel, the PEs also need to communicate across the processor boundaries of the nearest computational domain neighbors in each direction. The standard schemes inherently necessitate synchronizations at PE boundaries and consequently incur severe penalties due to processor idling, especially when a large number of PEs are used. This is the communication and synchronization bottleneck that is expected to pose a major challenge in efficiently scaling to next-generation exascale machines [8]. As described in previous chapters, an efficient way to mitigate this bottleneck is to relax the strict communication and synchronization requirements and perform simulations asynchronously using asynchrony-tolerant

(AT) schemes [16].

The AT schemes were used to perform accurate asynchronous simulations of Burgers' turbulence [109] and compressible turbulence [19] and exhibited superior scaling to their synchronous counterpart. To further advance the applicability of such asynchronous simulations with AT schemes, multi-physics simulations of turbulent combustion, that demand massive computations are a natural next choice. As a first step to evaluate the numerical performance of AT schemes for computationally expensive and highly nonlinear turbulent combustion simulations, several canonical reacting flow problems in one-dimension have been performed here. Precisely, the effect of data asynchrony is studied on autoignition, premixed flame propagation and non-premixed autoignition. Both one-step and detailed chemical mechanisms with stiff reactions are used to test the efficacy of the AT schemes. Moreover, the accuracy of AT-WENO schemes derived in Chapter 4 is also demonstrated.

7.2 Governing equations

In the following sections the effect of asynchrony is evaluated for a set of canonical reacting flow configurations. The AT schemes described the earlier Chapters are implemented in a compressible reacting flow solver in one-dimension with periodic and open boundaries. The governing conservation equations and constitutive laws are presented in this section. The one-dimensional form of the conservation equations for mass, momentum, total energy and species continuity are

$$\begin{aligned}
\frac{\partial \rho}{\partial t} &= -\frac{\partial \rho u}{\partial x} \\
\frac{\partial (\rho u)}{\partial t} &= -\frac{\partial (\rho u u)}{\partial x} + \frac{\partial \tau}{\partial x} - \frac{\partial p}{\partial x} \\
\frac{\partial (\rho e_0)}{\partial t} &= -\frac{\partial [u(\rho e_0 + p)]}{\partial x} + \frac{\partial (\tau u)}{\partial x} - \frac{\partial q}{\partial x} \\
\frac{\partial (\rho Y_i)}{\partial t} &= -\frac{\partial (\rho u Y_i)}{\partial x} - \frac{\partial (\rho Y_i V_i)}{\partial x} + W_i \dot{\omega}_i,
\end{aligned} \tag{7.1}$$

where Y_i is the mass fraction, W_i is the molecular weight, V_i is the species mass diffusion, $\dot{\omega}_i$ is the molar production rate of species i and e_0 is the specific total energy

$$e_0 = \frac{u^2}{2} - \frac{p}{\rho} + h. \tag{7.2}$$

$h = \sum_{i=1}^{N_s} Y_i h_i = \sum_{i=1}^{N_s} Y_i \left(h_i^0 + \int_{T_0}^{T} c_{p,i} dT \right)$ is the total enthalpy expressed in terms of h_i^0 which is the enthalpy of formation of species i at temperature T_0 and the isobaric heat capacity $c_p = \sum_{i=1}^{N_s} Y_i c_{p,i}$. For an ideal gas mixture, $p = \rho R_u T/W$ and $c_p - c_v = R_u/W$ where $W = \left(\sum_{i=1}^{N_s} Y_i/W_i \right)^{-1}$ and R_u is the universal gas constant, are used to compute the pressure and specific heats. The viscous stress τ is

$$\tau = \frac{4}{3}\mu \frac{\partial u}{\partial x}, \tag{7.3}$$

and the heat flux and species diffusion velocities are given by

$$\begin{aligned} q &= -\lambda \frac{\partial T}{\partial x} + \sum_{i=1}^{N_s} h_i J_i \\ V_i &= -\frac{D_i^{\text{mix}}}{X_i} \frac{\partial X_i}{\partial x} \end{aligned} \tag{7.4}$$

where $J_i = \rho Y_i V_i$ is the species diffusive flux, D_i^{mix} is the mixture-averaged diffusion coefficient, and $X_i = Y_i W/W_i$ is the mole fraction. Barodiffusion and the Soret and Dufour effects are not considered. CHEMKIN [110] and TRANSPORT [111] software libraries were linked with the solver and used for evaluating reaction rates, thermodynamic and mixture-averaged transport properties. Eq. (7.1) can be written in a compact form as in Eq. (4.1) where

$$Q = \begin{pmatrix} \rho \\ \rho u \\ \rho e_0 \\ \rho Y_i \end{pmatrix}, \quad C = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ u(\rho e_0 + p) \\ \rho u Y_i \end{pmatrix}, \quad D = \begin{pmatrix} 0 \\ -\tau \\ -\tau u + q \\ \rho Y_i V_i \end{pmatrix}, \text{ and } \quad S = \begin{pmatrix} 0 \\ 0 \\ 0 \\ W_i \dot{\omega}_i \end{pmatrix}. \tag{7.5}$$

The derivatives of product terms in Eq. (7.1) are expanded using the chain rule. For example, $\partial(\rho u Y_i)/\partial x := (\rho u Y_i)_x = \rho u (Y_i)_x + \rho u_x Y_i + \rho_x u Y_i$. This essentially allows computation of derivatives of such terms using AT schemes without having to retain every product term at multiple time-levels. Apart from the expansion using the chain rule, the one-dimensional asynchronous solver used in the present study is largely based on S3D [18] that is widely used to perform DNS of turbulent combustion.

7.3 Numerical Results

In this section, five different flow configurations are selected to assess the effect of asynchrony on canonical combustion problems. For the first case, two types of asynchronous simulations are considered,

1. Standard schemes used asynchronously (AS-SFD or AS, AS-WENO as applicable)

2. AT schemes for asynchronous computation (AT, AT-WENO as applicable).

For the remainder of the cases, only the the asynchronous simulation performed using AT schemes is compared with the synchronous simulation. The domain is decomposed into P processors and delays are introduced using a random number generator at each processor boundary similar to [16]. The maximum allowed delay levels are three with the probability of non-zero delay gradually increasing from Set-1 to Set-3. The different probability sets considered in numerical simulations are tabulated in Table 7.1 where Set-4 represents a synchronous simulation. The probability for Set-2 and Set-3 is similar to the probability of delays observed on TACC supercomputers [19]. Summary of all the numerical experiments performed and their relevance is listed in Table 7.2.

7.3.1 Non-reacting case: acoustic wave propagation

For the non-reacting case, the propagation of an acoustic wave in air is considered. By doing so, the effect of source of the acoustic wave, for example an ignition kernel is decoupled from its propagation, and the only focus is on whether the asynchrony-tolerant framework can accurately capture waves traversing with the speed of sound. Furthermore, if an error in gradients due to

	Probability $[p_0\ p_1\ p_2]$	Legend
Set-1	[0.8 0.1 0.1]	___ ○
Set-2	[0.6 0.3 0.1]	- . *
Set-3	[0.4 0.5 0.1]	- - □
Set-4	[1.0 0.0 0.0]	. . . +

Table 7.1: Probability of simulated delays for different sets used in the numerical experiments presented in Section 5. This legend (color and/or symbol) is used in all the figures from Fig. 7.1 to Fig. 7.10.

Case number	Case name	Relevant processes to be resolved
5.1	Acoustic wave propagation (non-reacting)	Pressure perturbation travelling at speed of sound
5.2	Auto-ignition of H_2 (periodic domain)	Spontaneous ignition dominated by reaction term
5.3	Auto-ignition of C_2H_4 (temperature fluctuations at inflow)	Unsteadiness, oscillatory ignition front
5.4	Premixed flame propagation	Reactive-diffusive balance in reaction zone
5.5	Non-premixed ignition	Diffusion controlled reaction front
5.6	Propagation of a detonation wave	Jump due to shock front followed by a reaction zone

Table 7.2: Summary of numerical simulations and their relevance.

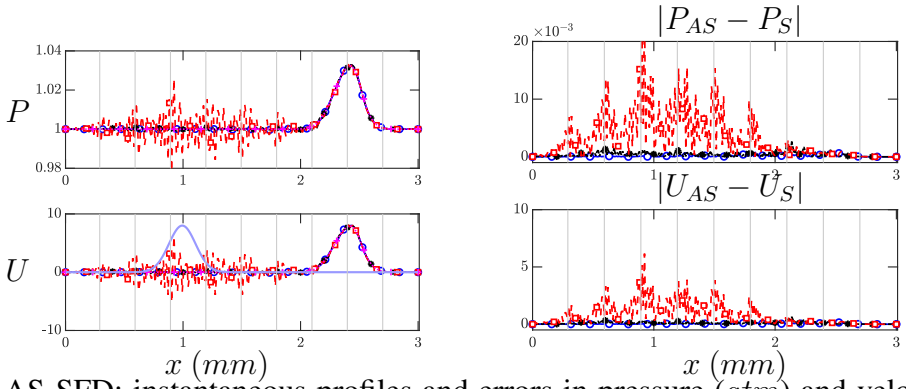

Figure 7.1: AS-SFD: instantaneous profiles and errors in pressure (atm) and velocity (ms^{-1}) at $t = 4 \times 10^{-6}s$. The different lines: blue (Set-1), black (Set-2), red (Set-3) and magenta (Set-4) are defined in Table 7.1. The faded blue line indicates the initial condition and faded-black lines represent processor boundaries. AS: Asynchronous with standard finite-difference schemes, S: synchronous.

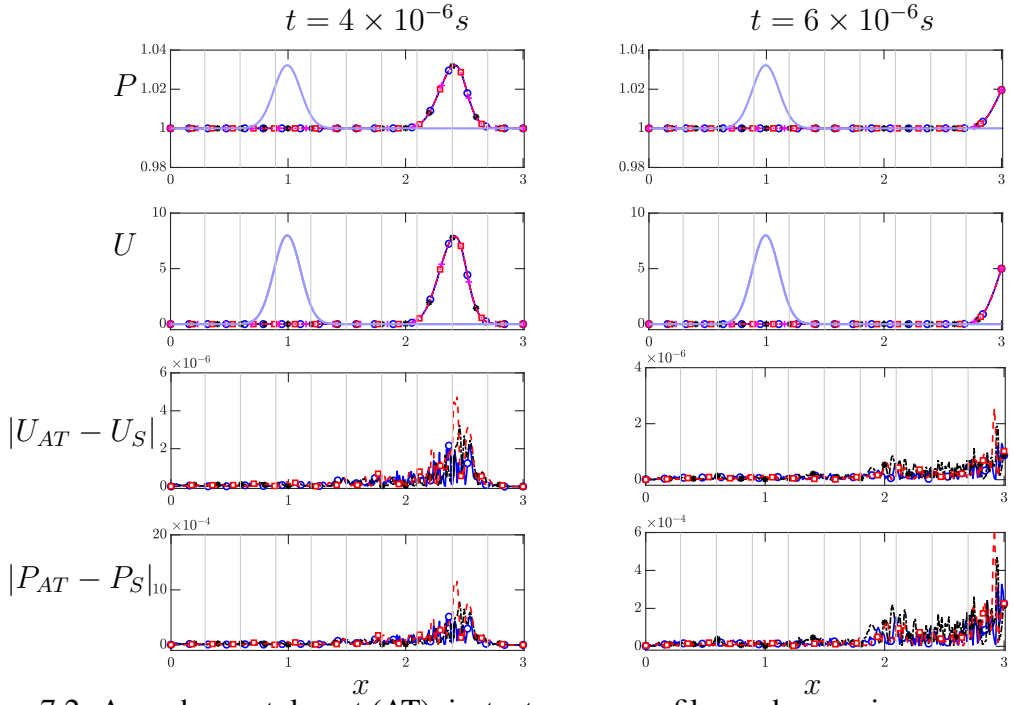

Figure 7.2: Asynchrony-tolerant (AT): instantaneous profiles and errors in pressure (atm) and velocity (ms^{-1}) at $t = 4 \times 10^{-6}s$ (left) and $t = 6 \times 10^{-6}s$ (right). The different lines: blue (Set-1), black (Set-2), red (Set-3) and magenta (Set-4) are defined in Table 7.1. The faded blue line indicates the initial condition and faded-black lines represent processor boundaries. AT: asynchronous with asynchrony-tolerant, S: synchronous.

delayed data at the boundaries manifest themselves in the form of dilatational modes then one should observe these effects in the propagation of acoustic waves through larger errors. In the current problem the acoustic wave is considered to be a perturbation in the pressure field that is described with the following initial condition [112],

$$\begin{aligned} u(x,0) &= u_0 + \mathcal{A}\exp\left[-\mathcal{B}\left(\frac{x-x_0}{L}\right)\right] \\ P(x,0) &= P_0 + \rho_0 c_0\left(u-u_0\right) \\ \rho(x,0) &= \rho_0 + \frac{\rho_0\left(u-u_0\right)}{c_0} \\ T &= P/\rho R \end{aligned} \tag{7.6}$$

where u_0, ρ_0, P_0 prescribe the uniform mean values, and ideal gas law is used to compute the temperature field. Here $\mathcal{A}$ and $\mathcal{B}$ determine the magnitude and stiffness of the acoustic fluctuation and x_0 is the location of the fluctuation peak. This initial field is shown as a faded blue line in Fig. 7.1 and Fig. 7.2. Non-reflecting inflow/outflow boundary conditions [113, 114] are used and the initial fluctuation is allowed to traverse to at least one processing element boundary where it encounters delays. While at the internal points standard fourth-order central difference schemes are used, at the physical boundary points the derivatives are computed using second-order finite difference schemes. For computation of derivatives at the processor boundaries, fourth-order AT schemes are used (see Appendix A).

When the standard central-difference schemes are used asynchronously (AS-SFD), there are visible fluctuations in the pressure field even at early times. This is evident from Fig. 7.1 where both instantaneous pressure and velocity fields at time $t = 4 \times 10^{-6}s$ show large numerical errors and clear deviation from the corresponding synchronous field. For larger delay values corresponding to Set-3 in Table 7.1, the numerical perturbations become significant at much shorter times and render the simulation unstable eventually. These errors are amplified even after the acoustic wave leaves the domain. However, when AT schemes are used for propagation of acoustic wave with delays at processor boundaries, the solution remains in close agreement with its synchronous counterpart

		$t = 4 \times 10^{-6} s$		$t = 6 \times 10^{-6} s$	
AS-SFD					
	Case	$U\ (ms^{-1})$	$P\ (atm)$	$U\ (ms^{-1})$	$P\ (atm)$
L_1 error	1	3.33e-02	1.36e-04	3.30e-02	1.34e-04
	2	9.87e-02	4.00e-04	3.78e-01	1.52e-03
	3	8.01e-01	3.26e-03	2.28e+01	9.31e-02
L_∞ error	1	2.27e-01	9.43e-04	2.72e-01	1.08e-03
	2	6.53e-01	2.69e-03	2.03e+00	8.08e-03
	3	6.16e+00	2.57e-02	1.45e+02	7.56e-01
Asynchrony-Tolerant (AT)					
	Case	$U\ (ms^{-1})$	$P\ (atm)$	$U\ (ms^{-1})$	$P\ (atm)$
L_1 error	1	5.88e-05	2.43e-07	3.38e-05	1.40e-07
	2	6.70e-05	2.75e-07	4.82e-05	1.95e-07
	3	8.83e-05	3.64e-07	3.81e-05	1.57e-07
L_∞ error	1	5.25e-04	2.18e-06	3.27e-04	1.29e-06
	2	7.84e-04	3.23e-06	4.70e-04	1.92e-06
	2	1.16e-03	4.73e-06	6.09e-04	2.56e-06

Table 7.3: L_1 and L_∞ norm of error in velocity and pressure in acoustic wave propagation simulations.

and the errors are several orders of magnitude smaller (see Fig. 7.2). Similar behavior is also observed at later times when the wave almost leaves the right boundary. Moreover, the errors in both pressure and velocity are not localized to near processing element boundary points where AT schemes are used to compute derivatives.

The L_1 and L_∞ norms of the errors computed with respect to the synchronous simulation are defined as

$$\begin{aligned} Z_{L_1} &= \langle Z_{\text{synchronous}} - Z_{\text{AS or AT}} \rangle \\ Z_{L_\infty} &= \max \left(Z_{\text{synchronous}} - Z_{\text{AS or AT}} \right), \end{aligned} \tag{7.7}$$

where Z is any quantity, for example temperature, pressure or mass fractions, and $\langle . \rangle$ is the spatial average. These errors are listed in Table 7.3 for both AS-SFD and AT simulations at two different times. The AT schemes exhibit significantly less numerical error at both times, and does not grow monotonically with time like it does for the AS-SFD.

7.3.2 Auto-ignition of H_2 using one-step chemistry

A canonical problem of practical relevance to compression ignition in internal combustion engines is the auto-ignition in a premixed fuel-air mixture. Here auto-ignition of a lean H_2/air mixture at an equivalence ratio of 0.4895 at $400K$ is considered. A Gaussian temperature spike with a peak value of $1040K$, which is above the ignition limit, is introduced at the center of a one-dimensional closed domain and the mixture eventually auto-ignites after an induction period. Initially radical chain-branching reactions occur accumulating reactive intermediate species at nearly isothermal conditions. This eventually leads to thermal explosion and the temperature increases rapidly to nearly 2000K. Two reaction fronts then emanate from the spontaneous ignition, one propagating to the left, while the other to the right. The gas expansion from the heat release during auto-ignition results in an induced velocity from an initially quiescent flow field.

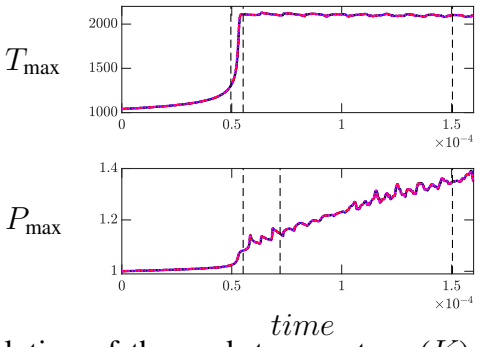

Figure 7.3: Temporal evolution of the peak temperature (K) and pressure (atm) during auto-ignition of a premixture of hydrogen/air in a periodic domain. The different lines (without symbols): blue (Set-1), black (Set-2), red (Set-3) and magenta (Set-4) are defined in Table 7.1. The dashed line indicates the time at which the instantaneous scalar and velocity profiles and errors are listed in Table 7.4 and Table 7.5.

The time elapsed from the initial hot-spot to a time at which the temporal gradient of temperature or heat release rate is maximum is the so-called ignition delay time (τ_{ign}). The time evolution of maximum temperature and pressure is shown in Fig. 7.3. It is clear that while the temperature increases slowly initially, there is an exponential increase in the temperature at around

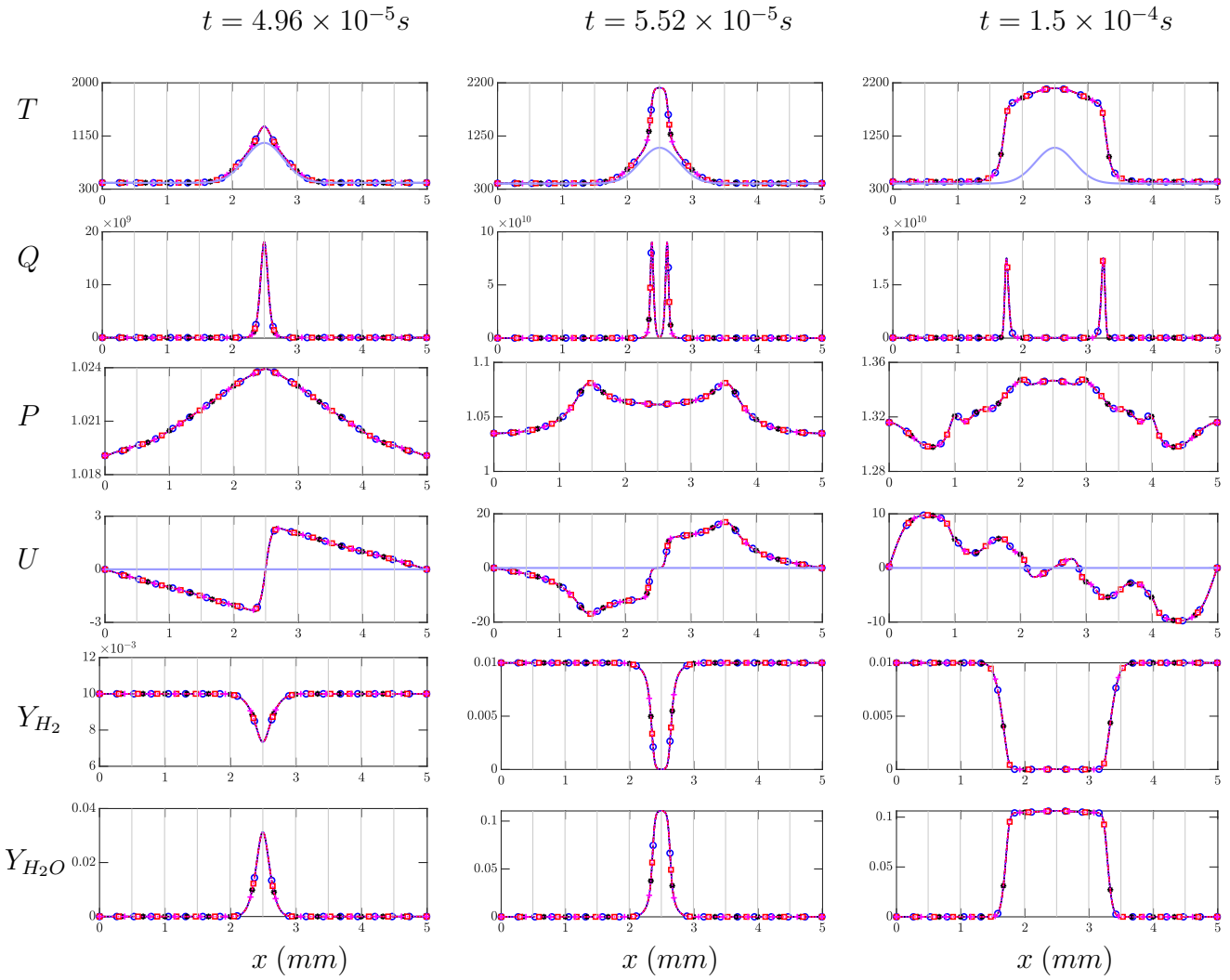

Figure 7.4: Asynchrony-tolerant (AT): auto-ignition of a premixed H_2/air mixture: instantaneous profiles of temperature (K), heat-release rate ($Jm^{-3}s^{-3}$), pressure (atm), velocity (ms^{-1}), and mass fractions of Y_{H_2} and Y_{H_2O}. The different lines: blue (Set-1), black (Set-2), red (Set-3) and magenta (Set-4) are defined in Table 7.1. The faded blue line indicates the initial condition and faded-black lines represent processor boundaries.

$t = 5 \times 10^{-5}s$. The temporal evolution of the peak temperature and pressure is captured accurately by the asynchrony-tolerant schemes. Following ignition, the two fronts propagate towards the left and right boundary and are allowed to traverse across at least one processing element boundary where delays are explicitly encountered. For simplicity we consider a periodic domain that allows us to compute spectral characteristics of both thermodynamic and hydrodynamic quantities. Note that the periodic boundary conditions considered here provide a closed volume with compression

heating from the ignition and leads to gradual rise in pressure with time. This is evident from Fig. 7.3 where the pressure increases from $1\ atm$ to $1.4\ atm$ after the ignition.

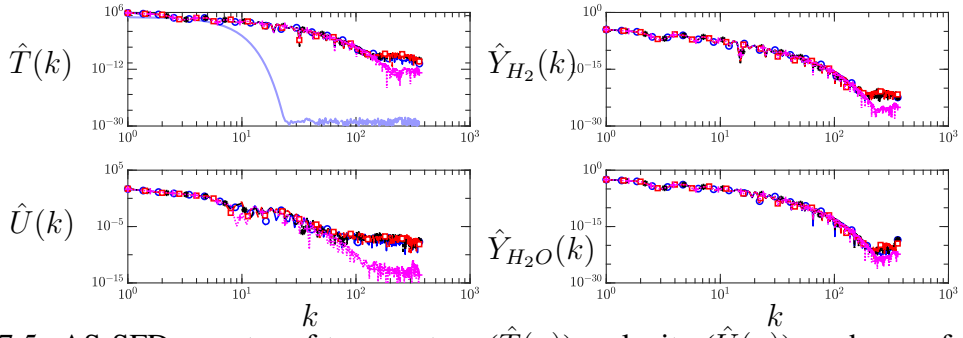

Figure 7.5: AS-SFD: spectra of temperature ($\hat{T}(\kappa)$), velocity ($\hat{U}(\kappa)$), and mass fractions $\hat{Y}_{H_2}(\kappa)$ and $\hat{Y}_{H_2O}(\kappa)$ for auto-ignition of a premixed H_2/air mixture. The different lines: blue (Set-1), black (Set-2), red (Set-3) and magenta (Set-4) are defined in Table 7.1.

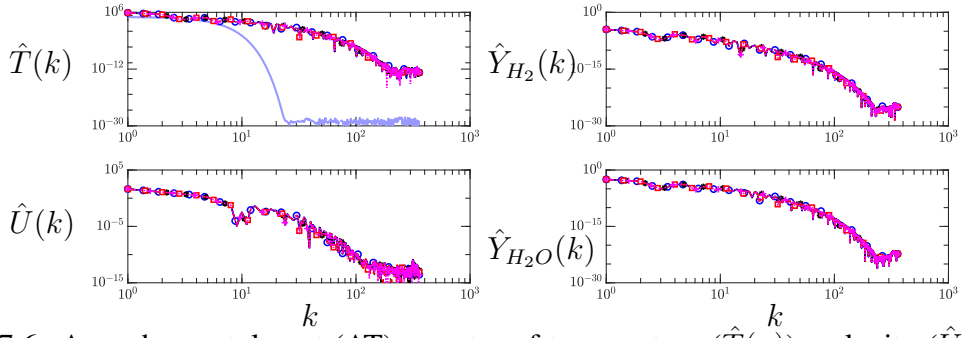

Figure 7.6: Asynchrony-tolerant (AT): spectra of temperature ($\hat{T}(\kappa)$), velocity ($\hat{U}(\kappa)$), and mass fractions $\hat{Y}_{H_2}(\kappa)$ and $\hat{Y}_{H_2O}(\kappa)$ for auto-ignition of premixed H_2. The different lines: blue (Set-1), black (Set-2), red (Set-3) and magenta (Set-4) are defined in Table 7.1.

To investigate the effect of delays at processor boundaries on important quantities of interest both before and after spontaneous ignition, the spatial evolution of temperature, velocity, pressure, heat release, and mass fractions of hydrogen and water are shown in Fig. 7.4. The time instants considered are indicated with dashed-black lines in Fig. 7.3. Qualitatively, no difference exists

Time	Case	$U\ (ms^{-1})$	$P\ (atm)$	$T\ (K)$	H_2	H_2O
AS-SFD						
1.5e-4	1	2.43e-01	8.1737e-04	3.2713e-01	1.2700e-06	2.5152e-05
	2	3.99e-01	1.3420e-03	5.3992e-01	2.1059e-06	4.1778e-05
	3	5.66e-01	1.9061e-03	7.6244e-01	2.9054e-06	5.8324e-05
Asynchrony-tolerant (AT)						
4.96e-5	1	1.07e-07	3.16e-10	8.50e-08	1.89e-13	2.29e-12
	2	1.23e-07	3.68e-10	8.19e-08	2.96e-13	2.31e-12
	3	1.33e-07	4.21e-10	1.11e-07	4.22e-13	3.66e-12
7.2e-5	1	6.80e-07	2.18e-09	7.14e-07	2.49e-12	2.53e-11
	2	6.46e-07	2.20e-09	9.74e-07	3.77e-12	4.89e-11
	3	1.07e-06	3.52e-09	1.09e-06	2.79e-12	3.67e-11
1.5e-4	1	1.17e-04	2.70e-07	7.67e-04	4.56e-09	4.87e-08
	2	5.67e-05	1.49e-07	2.52e-04	1.54e-09	1.61e-08
	3	1.10e-04	2.70e-07	7.17e-04	4.33e-09	4.62e-08

Table 7.4: L_1 norm of error in temperature, velocity, and mass fraction of H_2 and H_2O for auto-ignition of premixed Hydrogen.

between the instantaneous values of these quantities for synchronous simulation and asynchronous simulations with AT schemes. The L_1 and L_∞ norm of error computed with respect to the synchronous simulation is listed in Table 7.4 and Table 7.5, respectively, for three time instants. Also listed are the errors when the standard schemes are used asynchronously. Note that in this case the errors are several orders of magnitude larger than AT schemes, irrespective of the quantity. Moreover, for the former case, the maximum error in temperature is approximately 9K which can trigger reactions and result in nonphysical ignition in otherwise quiescent flow due to the strong temperature dependence of reaction rates. Such numerical errors are not observed for AT schemes even when the gradients for example in temperature or species mass fractions exist at the processing element boundaries.

The spectra of temperature, velocity and mass fraction of reactants and products, are shown in Fig. 7.5 and Fig. 7.6. The AT schemes exhibit an excellent agreement with the synchronous simulations at all wavenumbers (κ). Furthermore, non-physical accumulation of energy at high wavenumbers due to numerical errors does not exist which is in fact observed in Fig. 7.5 when

Time	Case	$U\ (ms^{-1})$	$P\ (atm)$	$T\ (K)$	H_2	H_2O
AS-SFD						
1.5e-4	1	8.73e-01	2.72e-03	3.50e+00	1.80e-05	3.19e-04
	2	1.51e+00	4.46e-03	5.78e+00	2.96e-05	5.25e-04
	3	2.14e+00	6.23e-03	8.79e+00	4.41e-05	7.88e-04
Asynchrony-tolerant (AT)						
2e-5	1	1.30e-06	2.44e-09	1.48e-06	4.61e-12	6.60e-11
	2	1.64e-06	2.30e-09	2.07e-06	1.07e-11	9.12e-11
	3	1.83e-06	2.80e-09	1.18e-06	1.19e-11	9.79e-11
5.52e-5	1	7.55e-06	2.64e-08	9.52e-06	7.27e-11	8.87e-10
	2	5.70e-06	1.95e-08	1.88e-05	1.36e-10	1.90e-09
	3	9.45e-06	3.26e-08	1.19e-05	9.30e-11	1.10e-09
1.5e-4	1	1.08e-03	2.44e-06	1.14e-02	5.51e-08	9.58e-07
	2	4.08e-04	7.97e-07	3.60e-03	1.77e-08	3.05e-07
	3	8.06e-04	1.75e-06	1.07e-02	5.12e-08	8.93e-07

Table 7.5: L_∞ norm of error in temperature, velocity, and mass fraction of H_2 and H_2O for auto-ignition of premixed Hydrogen.

the standard schemes are used with asynchrony. Hence, AT schemes demonstrate an excellent resolving efficiency in both physical and spectral space despite delayed data being used for the computation of derivatives at processing element boundaries.

7.3.3 Auto-ignition: temperature fluctuations at the inflow boundary

It is common to investigate the effect of turbulent fluctuations on fuel-air mixing, flame or spontaneous ignition front propagation, hotspots, etc. To simulate this effect in one-dimension and to ensure that the AT schemes can propagate the fluctuations accurately across processing element boundaries, a temperature perturbation is forced at the left inflow boundary. The initial condition, shown by the faded blue line in Fig. 7.7, is the steady state solution obtained from the auto-ignition of premixed C_2H_4/air mixture at a pressure of $2\ atm$. A 22 species 18-step reduced mechanism describing the oxidation kinetics of ethylene/air [115] is used. The steady solution is perturbed with sinusoidal temperature fluctuations of magnitude $180K$ and frequency $20Khz$ at the left boundary using an oscillatory inflow boundary condition [116, 117]. As the perturbations approach the igniting front, the temperature in its vicinity rises as is evident from the increase in

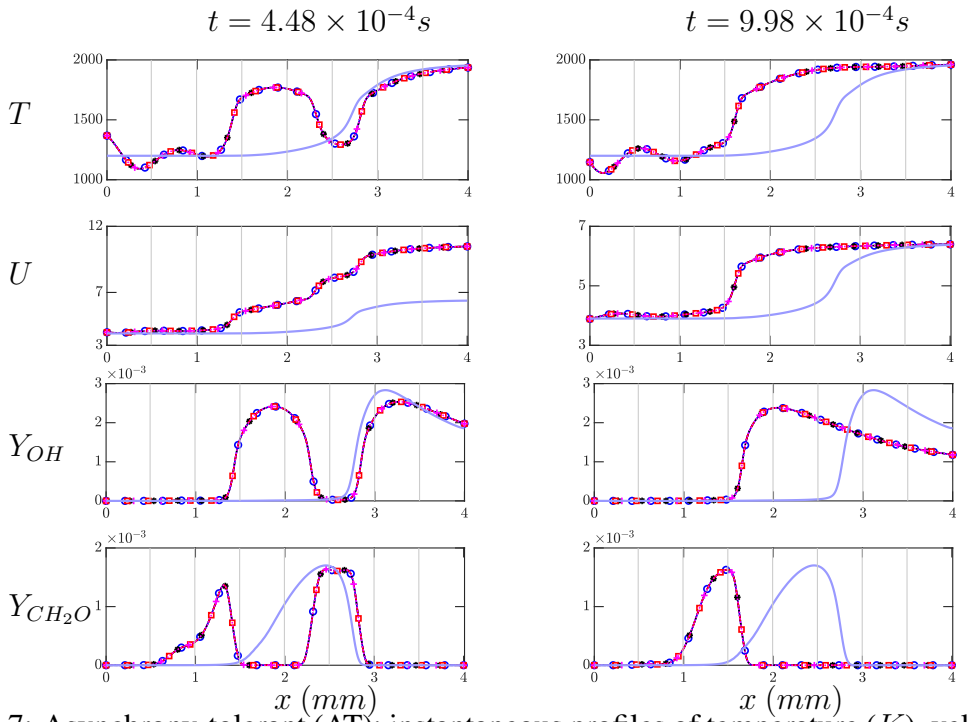

Figure 7.7: Asynchrony-tolerant (AT): instantaneous profiles of temperature (K), velocity (ms^{-1}), and mass fractions of OH and CH_2O radicals for autoignition of premixed C_2H_4/air mixtures with temperature fluctuations at the inflow. The different lines: blue (Set-1), black (Set-2), red (Set-3) and magenta (Set-4) are defined in Table 7.1. The faded blue line indicates the initial condition and faded-black lines represent processor boundaries.

the mass fraction of CH_2O radical in Fig. 7.7. This induces a secondary ignition kernel to the left of the initial front as can be seen in Fig. 7.7 (left column). The two kernels eventually interact and as time progresses, a steady ignition front develops to the left of the initial kernel as shown in the plots in the right column of Fig. 7.7. This front oscillates about a mean location, with the peak temperature and pressure also oscillating in response to the incoming sinusoidal fluctuations. The AT schemes accurately capture the transient and steady state evolution of both the temperature and intermediate species even in the presence of delays at the processor boundaries. An excellent qualitative agreement between the instantaneous profiles for AT and synchronous simulations in Fig. 7.7 is observed. Both L_1 and L_∞ norms of the errors tabulated in Table 7.6 and Table 7.7 are also reasonably small at both times.

Time	Case	$U\ (ms^{-1})$	$P\ (atm)$	$T\ (K)$	OH	CH_2O	H_2O
Asynchrony-tolerant (AT)							
4.48e-4	1	3.20e-08	1.10e-10	1.32e-07	6.67e-13	3.53e-13	4.48e-12
	2	3.89e-08	1.27e-10	1.70e-07	8.75e-13	4.82e-13	5.91e-12
	3	4.97e-08	1.66e-10	1.75e-07	9.06e-13	4.90e-13	6.08e-12
9.98e-4	1	3.09e-09	1.09e-11	3.32e-08	1.50e-13	9.12e-14	1.16e-12
	2	3.99e-09	1.38e-11	1.93e-08	1.20e-13	4.42e-14	6.33e-13
	3	4.77e-09	1.69e-11	2.82e-08	1.69e-13	7.69e-14	8.95e-13

Table 7.6: Asynchrony-tolerant (AT): L_1 norm of error in temperature, velocity, and mass fraction of OH and CH_2O for auto-ignition of C_2H_4/air flame with temperature fluctuations at the inflow.

Time	Case	$U\ (ms^{-1})$	$P\ (atm)$	$T\ (K)$	OH	CH_2O	H_2O
Asynchrony-tolerant (AT)							
4.48e-4	1	2.10e-07	8.22e-10	1.08e-06	5.76e-12	4.36e-12	5.12e-11
	2	2.20e-07	9.66e-10	1.53e-06	9.46e-12	6.58e-12	7.44e-11
	3	3.46e-07	1.30e-09	1.33e-06	8.23e-12	5.76e-12	6.54e-11
1.9e-3	1	2.05e-08	7.70e-11	5.33e-07	2.81e-12	2.15e-12	2.58e-11
	2	2.78e-08	1.01e-10	3.13e-07	1.68e-12	1.26e-12	1.50e-11
	3	3.57e-08	1.28e-10	4.41e-07	2.42e-12	1.83e-12	2.14e-11

Table 7.7: Asynchrony-tolerant (AT): L_∞ norm of error in temperature, velocity, and mass fraction of OH and CH_2O for auto-ignition of C_2H_4/air flame with temperature fluctuations at the inflow.

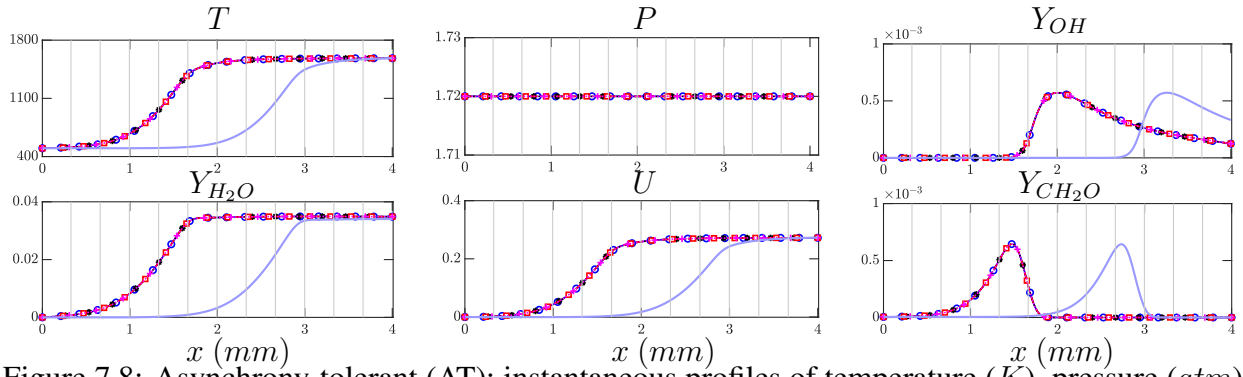

Figure 7.8: Asynchrony-tolerant (AT): instantaneous profiles of temperature (K), pressure (atm), velocity (ms^{-1}), and mass fractions of product, H_2O, and radicals, OH and CH_2O, at $t = 1 \times 10^{-2}s$ for premixed C_2H_4 flame propagation. The different lines: blue (Set-1), black (Set-2), red (Set-3) and magenta (Set-4) are defined in Table 7.1. The faded blue line indicates the initial condition and faded-black lines represent processor boundaries.

7.3.4 Premixed flame propagation

The previous cases focused on investigation of the effect of asynchrony on spontaneous ignition dominated by unsteadiness and advection-reaction balance. The current example is considered to study the effect of delayed data on laminar premixed flame propagation. The flame propagates at a subsonic velocity and is characterized by a balance between the reactive and diffusive terms in the steady species conservation equations. A mixture of C_2H_4/air at $\phi = 0.42$, $T = 500\ K$ and $P = 1.72\ atm$ is considered. The one-dimensional domain is $4\ mm$ and is discretized with 576 grid-points that are distributed across 12 processors. The 22 species, 18-step reduced mechanism from [115] for ethylene/air used in the earlier case is also used here. The initial condition is generated using an auto-ignition case, and non-reflecting inflow and subsonic outflow are imposed at the left and right boundaries, respectively. The flame is initially located close to the right edge of the domain and with time propagates to the left while consuming the reactants mixture. The initial flame front, comprising pre-heat and reaction zones, spans across three PEs as shown by the faded-blue line Fig. 7.8. This flame traverses across multiple processing element boundaries and the errors are computed to assess the effect of the asynchronous data encountered at the boundaries on flow and flame quantities.

Time	Case	$U\ (ms^{-1})$	$P\ (atm)$	$T\ (K)$	OH	CH_2O	H_2O
L_1 error							
1e-2	1	6.22e-11	2.32e-13	2.72e-09	2.51e-15	3.21e-15	8.60e-14
	2	9.96e-11	3.81e-13	7.12e-10	8.73e-16	8.04e-16	2.04e-14
	3	9.42e-11	3.61e-13	1.27e-09	1.16e-15	1.35e-15	3.42e-14
L_∞ error							
1e-2	1	5.72e-10	2.07e-12	1.42e-08	3.02e-14	4.66e-13	2.38e-14
	2	1.06e-09	3.07e-12	4.18e-09	8.81e-15	1.23e-13	6.31e-15
	3	6.52e-10	3.36e-12	6.61e-09	1.42e-14	2.09e-13	1.19e-14

Table 7.8: Asynchrony-tolerant (AT): L_1 and L_∞ norm of error in velocity, pressure, temperature, and mass fractions of OH, CH_2O and H_2O for premixed C_2H_4 flame propagation.

The flame structure remains the same with time, therefore, only the instantaneous profiles and errors at the time instant when the flame approaches near the left boundary is considered for accuracy analysis. There is an excellent agreement between the spatial profiles for the synchronous and the AT simulations, as shown in Fig. 7.8 for temperature, velocity as well as major and minor species. The drop in pressure in the reaction zone is negligibly small and thus the pressure values depicted in Fig. 7.8 are nearly constant. As is evident from Table 7.8, both L_1 and L_∞ norms of the error between synchronous and AT simulations are negligibly small and as low as $\mathcal{O}(10^{-14})$ for some of the species. Furthermore, the flame speed computed from the time evolution of the location of the peak heat release rate is equal to $12.59\ cm/s$ for both synchronous and asynchronous simulations. The thermal flame thickness is equal to $\delta_T = \frac{T_2 - T_1}{\max(\partial T/\partial x)} = 7.81 \times 10^{-4} m$, irrespective of the delays.

7.3.5 Non-premixed ignition

An ignition of non-premixed H_2 using Burke's mechanism [118] with 9 species is considered next. The setup is similar to the one used in [112] and [119] with fuel diluted with nitrogen on the left and air heated to $1500K$ on the right, as shown in Fig. 7.9. An inflow boundary condition from [116] is used on the left boundary with zero velocity and an outflow boundary condition is used on the right boundary with appropriate viscous conditions on each side. The one-dimensional domain is $4\ mm$ in length, discretized into 576 grid-points that are distributed across 12 PEs.

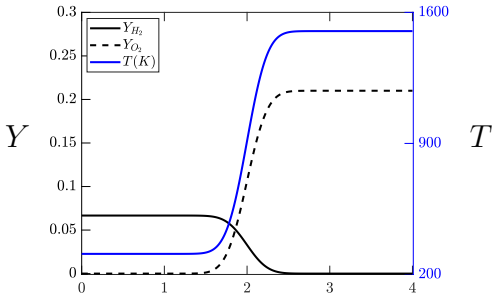

Figure 7.9: Initial condition for non-premixed ignition with diluted fuel on the left and vitiated air on the right.

Time	Case	$U\ (ms^{-1})$	$P\ (atm)$	$T\ (K)$	H_2	O	H	H_2O
Asynchrony-tolerant (AT)								
7.92e-5	1	1.94e-08	4.43e-11	1.52e-07	6.04e-12	3.63e-12	4.85e-13	2.54e-11
	2	2.59e-08	5.94e-11	1.97e-07	6.72e-12	5.40e-12	6.60e-13	3.74e-11
	3	3.14e-08	6.44e-11	3.13e-07	1.12e-11	7.77e-12	9.38e-13	5.42e-11
3.19e-4	1	5.94e-10	1.14e-12	2.57e-08	6.04e-13	2.69e-13	1.95e-14	4.43e-12
	2	5.25e-10	1.06e-12	3.45e-08	6.98e-13	2.55e-13	2.22e-14	5.91e-12
	3	6.18e-10	1.30e-12	5.01e-08	1.61e-12	2.56e-13	4.91e-14	4.80e-12

Table 7.9: Asynchrony-tolerant (AT): L_1 norm of error in temperature, velocity, and mass fractions of H_2, O, H and H_2O for non-premixed $H2/air$ ignition.

As time progresses and diffusion process tends to homogenize the gradients, there is ignition close to the stoichiometric mixture fraction, resulting in a rise in peak temperature and concentration of intermediate species. The instantaneous profiles of key quantities are shown in Fig. 7.10 at two times. The first series of plots on the left column in Fig. 7.10 are at an earlier time when ignition is localized. With time both fuel and intermediate species diffuse across processing element boundaries and the flame expands, which is shown in the plots on the right column in Fig. 7.10. The evolution of both major and minor species as well as temperature and velocity is accurately resolved by the AT schemes. The average and maximum errors, tabulated in Table 7.9 and Table 7.10, respectively, are both negligibly small. Similar low errors were also observed in other species that are not shown here.

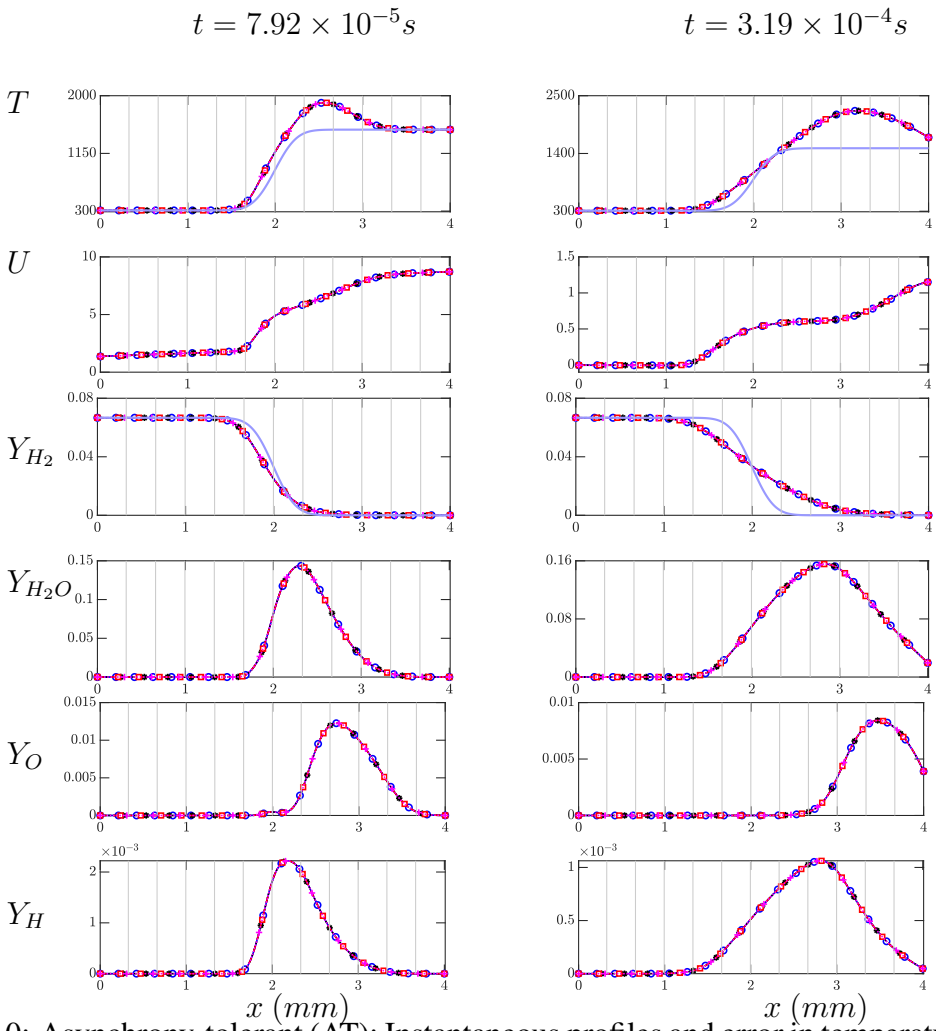

Figure 7.10: Asynchrony-tolerant (AT): Instantaneous profiles and error in temperature (K), velocity (ms^{-1}) and mass fractions of H_2, H_2O, O and H for non-premixed ignition of H_2. The different lines: blue (Set-1), black (Set-2), red (Set-3) and magenta (Set-4) are defined in Table 7.1. The faded blue line indicates the initial condition and faded-black lines represent processor boundaries.

7.3.6 Propagation of a detonation wave

In order to test the numerical accuracy of AT-WENO schemes for reacting flows, a detonation of stoichiometric hydrogen/oxygen mixture diluted with argon is considered. The mixture with molar ratio $H_2 : O_2 : Ar = 2 : 1 : 7$ is at an initial temperature of 305K and pressure of 6670 Pa. A detailed hydrogen/air mechanism [120] comprising of 9 species and 34 reversible reactions is used. A similar setup has also been used as a test case in previous studies to investigate the efficacy of numerical schemes [121, 122, 123]. Supersonic outflow condition is used at the left boundary and subsonic inlet condition at the right boundary. The simulation is initialized with a shock that

Time	Case	$U\ (ms^{-1})$	$P\ (atm)$	$T\ (K)$	H_2	O	H	H_2O
Asynchrony-tolerant (AT)								
7.92e-5	1	1.09e-07	1.91e-10	6.10e-07	3.89e-11	1.31e-10	2.63e-11	4.05e-12
	2	1.11e-07	2.25e-10	8.52e-07	4.83e-11	1.85e-10	3.69e-11	5.24e-12
	3	1.21e-07	2.14e-10	1.25e-06	7.16e-11	2.73e-10	5.27e-11	7.10e-12
7.92e-5	1	5.08e-09	9.80e-12	9.89e-08	2.38e-12	1.65e-11	1.50e-12	7.53e-14
	2	4.01e-09	8.79e-12	8.94e-08	2.49e-12	1.64e-11	1.29e-12	6.47e-14
	3	4.93e-09	7.91e-12	1.52e-07	5.15e-12	1.41e-11	1.17e-12	1.38e-13

Table 7.10: Asynchrony-tolerant (AT): L_∞ norm of error in temperature, velocity, and mass fractions of H_2, O, H and H_2O for non-premixed $H2/air$ ignition.

ignites the mixture at the left end of the computational domain. The domain length is $30\ cm$ which is discretized into 6000 grid-points that are distributed across 100 PEs.

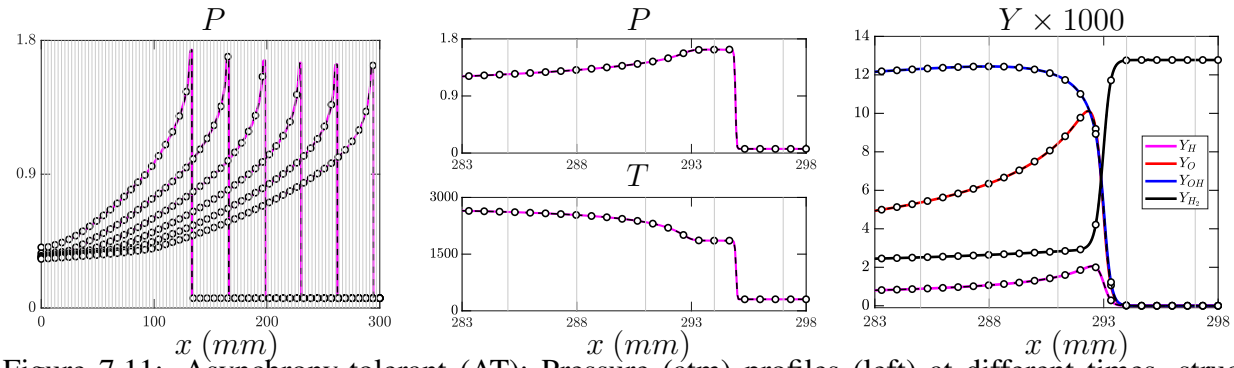

Figure 7.11: Asynchrony-tolerant (AT): Pressure (atm) profiles (left) at different times, structure of detonation wave (middle) for temperature (K) and pressure (atm) and mass fractions of H, O, OH, H_2 (right). The solid lines are standard WENO and black-dashed line with black symbols (Set-3 in Table 7.1) corresponds to AT-WENO. Faded-back lines represent processor boundaries.

The simulation results are shown in Fig. 7.11 with pressure at different time instants in the leftmost plot. There is a leading shock front that compresses the fuel/air mixture followed by the induction and the reaction zones. The pressure obtained with synchronous WENO scheme is shown in solid magenta. It is in excellent agreement with the pressure computed with AT-WENO

scheme shown in black circles and dashed lines. A zoom-in of the detonation structure for temperature and pressure is shown in the centre plots of Fig. 7.11. The mixture reacts after the induction zone leading to an increase in temperature and a corresponding decrease in pressure. The mass fractions for different radicals including fuel H_2 are plotted on the rightmost plot in Fig. 7.11. The results of this simulation show that the detonation structure is captured accurately with the AT-WENO scheme. Furthermore, the detonation velocity obtained from standard WENO and AT-WENO simulations is $1619.7\ ms^{-1}$ and $1620.1\ ms^{-1}$, respectively. These values are in good agreement with detonation velocity reported in previous studies [121, 122, 123]. The peak von-Neumann pressure is $165.35\ kPa$ and $165.34\ kPa$ for WENO and AT-WENO which are also close to the previous works. Overall the AT-WENO schemes derived in this paper exhibit excellent numerical accuracy despite the use of delayed data at processor boundaries. Since multidimensional simulations of detonation phenomena in combustion devices face highly intensive computational resource requirements [124, 125], AT-WENO schemes hold the potential of offsetting a certain portion of communication overhead associated with the standard WENO schemes at large node counts, thereby leading to improved scalability.

7.4 Discussions

DNS of turbulent combustion at higher Reynolds numbers with detailed chemical mechanisms and at conditions relevant for practical devices will require efficient utilization of massive computing resources anticipated on the next generation Exascale machines. These simulations at scales and conditions that might be currently infeasible will provide fine-grained details into the interactions between turbulence and chemistry and help us understand the physical processes that result in pollutant formation, flame stabilization, blow-off, etc. This in turn will not only aide in the design of fuel-flexible, low-emission combustion engines but also in the development of physics-based models for low-cost engineering simulations. However, a successful transition of DNS codes to the Exascale machines is possible only through redesign and development of new computational algorithms in order to overcome some of the obvious bottlenecks and challenges.

The anticipated computing power of a quintillion (10^{18}) double precision floating point oper-

ations per second on the Exascale machine will be realized through extreme levels of parallelism with millions of processing elements (PEs), including CPUs, GPUs and FPGAs. For example, Frontier at the Oak Ridge Leadership Computing Facility (OCLF) will have 1:4 CPU to GPU ratio with unified memory. Despite the use of state-of-the-art low-latency, high-bandwidth communication links, the sheer number of communications will significantly hinder scaling to large node counts. To this end, efficient strategies to mask communications and data movement will improve scalability. The use of AT schemes at the PE boundaries for computation of derivatives accurately, despite delayed data, will facilitate further overlap between communications and computations. However, a challenge in extending the AT schemes to three-dimensions is that the derivatives of convective and diffusive terms need to be expanded in terms of derivatives of primitive variables using chain rule. For example, $\partial(\rho ue)/\partial x := (\rho ue)_x = \rho u e_x + \rho e u_x + e u \rho_x$. This expansion increases the total number of derivatives that need to be computed, especially when a large number of species are involved. Alternatively, one can compute the convective and diffusive terms (for example, $\rho ue,\ \rho Y_i u,\ J_i,\ \tau$) from the primitive variables at different time levels as required by the AT schemes. These computations can be performed on the fly in order to avoid storing such product terms at multiple time-levels. Essentially there is a trade-off between memory (data movement) and computations. Re-design of data structures for better memory coherency will also be extremely helpful since AT schemes use data from multiple time levels and can thus incur high cache miss rate. Besides these challenges, a critical first step in using AT schemes for scalable simulations of turbulent combustion on massive supercomputers is to ascertain that the numerical artifacts due to delayed data at PE boundary are significantly small and do not affect the underlying physics. The comprehensive set of numerical experiments presented in the current work specifically highlight the excellent numerical accuracy of AT schemes for reacting flows.

A key component of simulations of reacting flows even for simple geometries include the prescribing of boundary conditions. The NSCBC [113, 114] with improvements for reacting flows [116, 117, 119, 126] has been successfully used for these simulations. In the normal direction, the terms in the NSCBC boundary conditions are not affected by AT schemes. For example, at

the outflow, one-sided schemes are used, which require only the local information within the PE. At the inflow, relaxation terms are used which again do not require PEs to communicate. Since transverse and viscous terms are also included for reacting flows [116, 117, 119], the AT schemes can be utilized in the transverse direction for computation of all the spatial derivatives involved. Extension to corners can be done following [126] with AT schemes used wherever necessary.

Another aspect of using asynchronous computations on Exascale machines is to leverage the high flop-rate of GPUs. In hybrid computing architectures, GPUs are expected to handle most of the computations, while CPUs facilitate communications between PEs. With AT schemes, GPUs do not have to wait on the CPUs for the most updated data from the neighboring PEs. This would enhance utilization of GPUs without affecting numerical accuracy or introducing idling penalties. A similar approach has been utilized in [127] where asynchronous copies between CPUs and GPUs are used to overlap computations and data movement, but delayed data with asynchronous computations have not been used. At a compiler level, new asynchronous run-time systems that are capable of dynamic task parallelism are being developed to improve the computation-communication overlap [128, 129, 130]. The AT schemes, which relax synchronization at a mathematical level, can be coupled with such programming models to create highly scalable PDE solvers. Asynchrony has also been utilized in [131] for scalable resilience to soft faults. However, in [131] all derivatives are still computed with the most updated data but the computations are re-arranged to ensure maximum overlap between communications and computations. The asynchronization approach utilized in [131] coupled with mathematical level asynchrony with AT schemes can be an effective in pushing the scaling wall. Furthermore, since AT schemes do not need the updated data, these schemes can also be used to recover node failures without halting the simulation altogether.

7.5 Conclusions

A series of numerical simulations that utilize delayed data at the processor boundaries are presented for canonical reacting flows with one-step and detailed reaction mechanisms were presented in this chapter. It is shown that large numerical errors are incurred in physical as well as spectral space when standard schemes are used asynchronously even for the simplest problems. To

overcome this loss of accuracy, the novel asynchrony-tolerant (AT) schemes are used for accurate computation of spatial derivatives with delayed data at the boundaries. These asynchronous simulations using AT schemes are compared with standard synchronous simulations and are shown to exhibit excellent qualitative and quantitative agreement. Both spatial and temporal evolution of hydrodynamic and thermo-chemical quantities are accurately captured by the AT schemes.

For simulations of high-speed flows with shocks and jump discontinuities, the standard shock-resolving WENO schemes have degraded numerical accuracy. Thus, following the procedure listed in [16] AT-WENO schemes are derived in Chapter 4. The order of accuracy of AT-WENO schemes was verified in Chapter 4 for both linear and non-linear equations. In the current chapter we used these schemes for simulations of detonation wave with delays and showed that the AT-WENO schemes are in excellent agreement with their synchronous counterpart.

A common conclusion one can arrive at through these simulations is that the AT schemes have an excellent accuracy despite relaxed synchronizations at the processor boundaries. Thus, AT schemes provide a potential path for highly scalable asynchronous simulations of turbulent combustion.

8. UNIFIED FRAMEWORK FOR DERIVING OPTIMAL FINITE DIFFERENCES*

8.1 Introduction

Ordinary differential equations (ODEs) as well as partial differential equations (PDEs) are pervasive in science and engineering as they model accurately a large number of natural and man-made systems. Unfortunately, many of these equations are exceedingly complex at realistic conditions and analytical solutions are virtually impossible. Thus, significant advances in understanding these phenomena have relied on the use of computer simulations for which an appropriate numerical method needs to be used to assure certain degree of accuracy in the solution.

Perhaps the most widely used method to discretize these governing equations in order to solve them on a computer is the so-called finite differences. The derivation of explicit finite difference schemes is in general very well known and has been studied extensively [20]. The general idea is to linearly combine the values of the function to be differentiated at neighboring points. The weights in this linear combination are determined so as to minimize error in some sense. The specific choice of this objective function, we show here, has a critical effect on the resulting schemes.

More formally, the standard procedure starts with an approximation of the derivative of a function f at a point x_i of the form

$$f_i' = \left.\frac{\partial f}{\partial x}\right|_{x_i} \approx \frac{1}{\Delta x}\sum_{m=-M}^{M} a_m f_{i+m} + \mathcal{O}(\Delta x^{p+1}). \tag{8.1}$$

The last term indicates that the truncation error of the approximation is of order $p+1$. In a traditional derivation one first selects the stencil size, that is the number of neighboring points to use in the approximation which is $(2M+1)$ in Eq. (8.1), and then finds the coefficients a_m such that $p+1$, the order of the truncation error, is largest. This is done by eliminating all terms of order lower than $p+1$ in a Taylor expansion of the right-hand-side of Eq. (8.1). This approach has also

*Parts of this chapter are reprinted from Journal of Computational Physics, Volume 399, Komal Kumari, Raktim Bhattacharya and Diego A. Donzis, "A unified approach for deriving optimal finite differences", Pages 108957, 15 December 2019, with permission from Elsevier.

been utilized to obtain the asynchrony-tolerant schemes in Chapter 1.

However, in many problems, especially those which involve multi-scale phenomena, one is concerned not only with a global measure of the error but also with the error incurred at different scales or, in Fourier space, at different wavenumbers. The range of wavenumbers and, thus, the resolution requirements depend on non-dimensional parameters such as Reynolds number for turbulent flows, or the Prandtl and Schmidt numbers for mixing problems, as they define length and time scales of interest in the problem. It is therefore of consequence to use schemes that are optimal for the time and length scale posed by the physics of the problems. This concern is typically addressed by analyzing the spectral characteristics of scheme derived above by using, for example, von Neumann analysis [20] or a modified wavenumber approach [47]. The result of this analysis is used to determine if the scheme derived from an order-of-accuracy consideration is indeed appropriate to resolve all relevant scales in the problem at hand. Note that this is in general *a posteriori* evaluation of the scheme.

There has been efforts in the literature to devise schemes with general properties in terms of spectral accuracy. Early work on acoustic computations [132] showed that some desired spectral behavior can be obtained by solving an optimization problem where the objective function to be minimized is some measure of these errors. A number of different applications based on the same general approach have been presented in the literature [132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142]. In all these studies specific requirements were put forth typically based on the physics of interest which often resulted in subjective criteria to account for those specific requirements. Some were limited in scope, for exmaple, by limiting the results to approximating only the first derivative [132, 133, 137, 135, 139]. These optimizations were also limited in the sense that the nature of the unknown coefficients (e.g. whether they are symmetric or antisymmetric) is specified a priori which resulted in the objective function comprising either only the real part or only the imaginary part of the spectral error, depending upon the order of the derivative being computed. However, as we show below, that symmetry and antisymmetry of the coefficients for even and odd derivatives respectively, can be obtained as a *consequence* of the minimization problem without any exter-

nally imposed conditions. Some others [141] formulated an optimization based on the maximum norm to minimize spectral errors but the algorithm adheres to certain ad-hoc rules (such as the nature of the coefficients or the relative magnitude of the coefficients in a stencil) and presents challenges in finding the global optimum. Other specialized optimizations have been conducted to obtain schemes with lower errors that, for example, can resolve very strong gradients in fluid flow calculations [138, 134, 136], though some degree of subjectivity and trial-and-error was used in the formulations. Both [134, 136] utilize a two level optimization to achieve better resolving efficiency and [136] also incorporated a weight function to emphasize on the relevant scales. Another approach is presented in [140] where the unknown coefficient is selected according to the level of dissipation required without carrying out any formal optimization. A general conclusion one can arrive at from all these studies, is that more sensible choices than commonly made can lead to numerical schemes that can outperform standard finite differences [47, 132]. Here we support this idea and show that the precise meaning of this metric has a clear impact on the scheme obtained. Another observation is that formulations are typically tailored with specific applications and constraints in mind due to the different requirements dictated by the physics of interest. It is thus not surprising that there seems to be no general rigorous mathematical framework under which these particular cases can be derived. Here, in a first step, we provide such a framework along with some rigorous results on the nature of the error that result from optimization formulations. As we will show, the framework can incorporate the different requirements needed for different cases and conditions.

A third critical aspect when considering numerical schemes is their stability. Obviously, to be usable, a scheme must be numerically stable when utilized to solve an ODE or PDE. Again, this has traditionally been *a posteriori* undertaking: after selecting a scheme of a given order, with a desired spectral accuracy, one would check if the scheme is stable or not and under which conditions. This has been the case for standard or optimized schemes [132, 143]. This is also the case for the two-step optimization of [144] in which an optimal spatial scheme is obtained first, followed by an optimization to get a stable time marching scheme. This also highlights the

importance of considering the relation between the space and time discretization to increase the computational efficiency of the spatial operators. In [142], the spatial and the temporal schemes are decoupled and optimized separately in order to achieve maximum resolving efficiency for both the operators. The optimization of the temporal scheme is also subject to the stability constraint. Schemes that remain stable for a broader range in the appropriate parameter space (time step size, grid spacing, non-dimensional groups such as CFL number, etc.) are typically preferred as simulations with larger time steps are computationally less expensive. While the time step and grid spacing are conventionally subject to the stability and resolution requirements, [142] shows that optimal values for these that minimize computational cost for some error level can also be obtained.

An overriding question is, thus, whether it is possible to find optimal schemes of given order, that are stable and that minimize the spectral error in a suitably defined manner typically informed by the physical characteristics of the problem being solved. This is the main motivation of the work presented in this chapter. The mathematical framework in which this can be achieved reduces to an optimization problem with equality and inequality constraints which can be solved, under certain conditions, analytically. The importance of this work is that it allows us to express physically meangingful desired properties and constraints into a unified mathematical framework which results in highly-accurate schemes for a particular problem of interest. Another important aspect of the proposed framework is that it also exposes explicitly tradeoffs that can be profitably used in specific circumstances. For example, we show that it is possible to construct explicit schemes that can remain stable for very large time steps (even an order of magnitude larger than equivalent standard schemes) when constraints on accuracy at some scales can be relaxed. We can also show that unlike traditional finite differences, one can design "spectrally flat" schemes which present a more homogeneous accuracy distribution across wavenumbers. Furthermore, because of optimality in spectral properties, we also show that the resulting schemes present better performance in terms of important physical properties like its dispersion relation, and group and phase velocity [145, 139, 146].

8.2 The framework for deriving finite differences

As described above, in this work we construct a framework to derive finite differences in which three important design characteristics, namely order of accuracy, spectral accuracy, and stability, are combined into a rigorous mathematical framework. We now present each in turn.

8.2.1 Order of accuracy

A generalization of the approximation in Eq. (8.1) to the d-th spatial derivative is given by

$$f_i^{(d)} = \frac{1}{(\Delta x)^d} \sum_{m=-M}^{M} a_m f_{i+m}, \tag{8.2}$$

where, as before, we have M points on either side of the i-th grid point where the derivative is sought. The stencil size is then $S \equiv 2M + 1$. A Taylor series for a term on the right-hand-side of Eq. (8.2) can be written as

$$f_{i+m} = f_i + (m\Delta x) f_i' + (m\Delta x)^2 f_i''/2! + \ldots.$$

Upon constructing the entire sum in Eq. (8.2), a $(p+1)$-th order approximation of the d-th derivative requires that the term with the d-th derivative be equal to $d!$ and that the rest of the terms up to order p be zero. After some algebra these constraints can be written as

$$\sum_{m=-M}^{M} m^q a_m = \begin{cases} 0 & q \neq d, \\ d! & q = d, \end{cases} \tag{8.3}$$

for $q \leq d + p$, or more compactly as

$$\mathbf{a}_d^T \mathbf{X}_d = \mathbf{y}_d, \tag{8.4}$$

where

$$\mathbf{a}_d^T := \begin{bmatrix} a_{-M} & a_{-M+1} & \cdots a_{M-1} & a_M \end{bmatrix}, \tag{8.5}$$

$$\mathbf{m}^T := \begin{bmatrix} -M & -M+1 & \cdots & M-1 & M \end{bmatrix}, \tag{8.6}$$

$$\mathbf{X}_d := \begin{bmatrix} \mathbf{1}_{S\times 1} & \mathbf{m} & \cdots & \mathbf{m}^{d-1} & \mathbf{m}^d & \cdots & \mathbf{m}^{d+p} \end{bmatrix}, \tag{8.7}$$

$$\mathbf{y}_d := \begin{bmatrix} \mathbf{0}_{1\times d} & d! & \mathbf{0}_{1\times p} \end{bmatrix}. \tag{8.8}$$

with $\mathbf{1}_{S\times 1}$ is an $S \times 1$ vector with ones as its elements and $\mathbf{0}_{1\times d}$ a $1 \times d$ vector with zeros as its elements. The vectors $\mathbf{m}^n$ in Eq. (8.7) are defined as vectors composed of each element of $\mathbf{m}$ raised to the power n.

Equation Eq. (8.4) is the linear system that, for a given stencil size M, results in a finite difference scheme of order $p + 1$. This approach, though presented in different forms across the literature, forms the basis for standard derivation of finite differences when $S = d+p$. In this case, the number of unknown coefficients in $\mathbf{a}_d$ equals the number of terms that need to be eliminated to maintain a certain order of accuracy. If on the other hand, $S > d+p$ then the solution to Eq. (8.4) is not unique. The remaining degrees of freedom can then be used to, e.g., assure spectral accuracy. This is presented next.

8.2.2 Spectral accuracy

In order to understand the behavior of discrete differentiation operators at different scales or frequencies, it is common to evaluate them utilizing a Fourier representation. This was also presented briefly in Chapter 3 (see section 3.2). Here again, for simplicity we consider a single mode in a spatial discrete Fourier series:

$$f(x) = \hat{f} e^{jkx} \tag{8.9}$$

where $\hat{f}$ is the Fourier coefficient of the mode at wavenumber k, and $j = \sqrt{-1}$. Clearly, the exact d-th derivative is given by

$$f^{(d)}(x) = (jk)^d f(x). \tag{8.10}$$

Now consider a discrete approximation of the derivative of the form Eq. (8.2). Since,

$$f_i := f(x_i) = \hat{f} e^{jkx_i}, \tag{8.11}$$

$$f_{i+m} := f(x_i + m\Delta x) = \hat{f} e^{jkx_i} e^{jkm\Delta x} = f_i e^{jkm\Delta x} \tag{8.12}$$

equation Eq. (8.2) becomes

$$f_i^{(d)} = \left(\frac{1}{(\Delta x)^d} \sum_m a_m e^{jkm\Delta x} \right) f_i \tag{8.13}$$

in terms of this single Fourier mode.

Comparison between the numerical approximation Eq. (8.13) and exact differentiation Eq. (8.10) suggests a natural way to define the error at a given waveumber as

$$e(\eta) = \sum_m a_m e^{jm\eta} - (j\eta)^d, \tag{8.14}$$

where $\eta := k\Delta x$ is a convenient normalized wavenumber in the interval $[0, \pi]$. More compactly, this can be expressed in matrix form as

$$e(\eta) = \left(\mathbf{C}^T(\eta) + j\mathbf{S}^T(\eta) \right) \mathbf{a}_d - (j\eta)^d, \tag{8.15}$$

where

$$\mathbf{C}(\eta) := \begin{bmatrix} \cos(-M\eta) \\ \vdots \\ \cos(-\eta) \\ 1 \\ \cos\eta \\ \vdots \\ \cos(M\eta) \end{bmatrix}, \text{ and } \mathbf{S}(\eta) := \begin{bmatrix} \sin(-M\eta) \\ \vdots \\ \sin(-\eta) \\ 0 \\ \sin(\eta) \\ \vdots \\ \sin(M\eta) \end{bmatrix}. \tag{8.16}$$

Equation Eq. (8.14) can also be written in terms of the so-called modified wavenumber $(j\widetilde{\eta})^d = \sum_m a_m e^{jm\eta} = \left(\mathbf{C}^T(\eta) + j\mathbf{S}^T(\eta)\right)\mathbf{a}_d$ as

$$e(\eta) = (j\widetilde{\eta})^d - (j\eta)^d = j^d(\widetilde{\eta}^d - \eta^d). \tag{8.17}$$

Clearly, the difference between $\widetilde{\eta}^d$ and η^d provides a measure of the spectral error at wavenumber η. The ratio of the modified wavenumber to the actual wavenumber is then

$$G_d := \frac{\widetilde{\eta}^d}{\eta^d} = \frac{e(\eta)}{(j\eta)^d} + 1, \tag{8.18}$$

which is a complementary measure of error across wavenumbers and will be used later on when comparing different schemes.

A global figure of merit to assess how accurately the scheme captures spectral content can be defined as the weighted $\mathcal{L}_2$ norm of the error $e(\eta)$:

$$\|e(\eta)\|^2_{\mathcal{L}_2} := \int_0^\pi \gamma(\eta) e^*(\eta) e(\eta) d\eta =: \langle e^*(\eta) e(\eta) \rangle \,. \tag{8.19}$$

Here $e^*(\eta)$ is the complex conjugate of $e(\eta)$, and $\gamma(\eta)$ is a weighting function introduced to provide control over which wavenumbers are to be more accurately resolved. The selection of $\gamma(\eta)$ would

depend, in general, on the physical characteristics of the system of interest. For example, for PDEs with multi-scale broadband solutions a natural choice would be a constant $\gamma(\eta)$ over the range of η of interest and zero elsewhere. For a system with two well defined wavenumber bands, on the other hand, one can define $\gamma(\eta)$ presenting relatively large values around those bands but negligible values everywhere else. Examples on the impact of this choice will be provided in section 8.3.

We can now frame the problem of the derivation of finite difference schemes combining spectral resolution and order of accuracy. Formally, our goal is to determine $\mathbf{a}_d$ such that $\|e(\eta)\|^2_{\mathcal{L}_2}$ is minimized, subject to a given order of accuracy defined by Eq. (8.4), i.e.

$$\min_{\mathbf{a}_d \in \mathbb{R}^{2M+1}} \|e(\eta)\|^2_{\mathcal{L}_2}, \text{ subject to Eq. (8.4).} \tag{8.20}$$

Equation Eq. (8.20) provides, then, the unifying formalism to find the coefficients in Eq. (8.2) that both provides a given order of accuracy and minimizes error in spectral space [132]. This formulation, written in different ways, has been used extensively as pointed out in the introduction, but here is presented in a very general form. Note that if the number of unknowns ($S = 2M + 1$) is equal to the number of terms to be removed from the truncation error to achieve a given order, then Eq. (8.4) has a unique solution and no optimization is possible. This is the case of standard finite difference schemes. If, on the other hand, the stencil size makes the number of unknowns greater that those needed to achieve a given order, the system will utilize those degrees of freedom to minimize $\|e(\eta)\|^2_{\mathcal{L}_2}$.

We note here that in standard derivations of finite differences based only on order-of-accuracy considerations, the spectral behavior is found a posteriori. Spectral resolution is then coupled to (dependent on) order of accuracy. While typically increasing the formal order of accuracy of the scheme, leads to a better spectral resolution, standard techniques provide no mechanism to constrain the spectral behavior of the resulting schemes. In the approach presented above, on the other hand, specifications on spectral accuracy are independent of order of accuracy. Thus, we see that by *coupling the two mathematical systems* into a unified formulation, we effectively

decouple the requirements for order of accuracy and spectral resolution. The result is that when the stencil size is increased, the extra degrees of freedom can be used to either increase the order of accuracy further or improve specific spectral behavior. In section 8.3 we present specific examples of optimized low-order schemes that are shown to have better spectral resolution than standard high-order finite differences.

To develop the theory further, it is convenient to distinguish between odd-order and even-order derivatives which leads to different behavior in Fourier space. In both cases an analytical solution can be found and is presented next.

8.2.2.1 Even derivatives

For $d = 2q$, $q = \{1, 2, \cdots\}$, $e(\eta)$ becomes

$$e(\eta) = \left(\mathbf{C}^T(\eta)\mathbf{a}_d - (-1)^q\eta^d\right) + j\mathbf{S}(\eta)^T\mathbf{a}_d.$$

The $\mathcal{L}_2$ norm of the error is therefore,

$$\begin{aligned}\|e(\eta)\|_{\mathcal{L}_2}^2 &:= \int_0^\pi \gamma(\eta)\left[\left(\mathbf{C}^T(\eta)\mathbf{a}_d - (-1)^q\eta^d\right)^2 + \left(\mathbf{S}(\eta)^T\mathbf{a}_d\right)^2\right]d\eta, \\ &= \mathbf{a}_d^T\underbrace{\left(\left\langle\gamma(\eta)\mathbf{C}(\eta)\mathbf{C}^T(\eta)\right\rangle + \left\langle\gamma(\eta)\mathbf{S}(\eta)\mathbf{S}^T(\eta)\right\rangle\right)}_{\mathbf{Q}_d}\mathbf{a}_d \\ &- 2\mathbf{a}_d^T\underbrace{\left\langle\gamma(\eta)(-1)^q\eta^d\mathbf{C}(\eta)\right\rangle}_{\mathbf{r}_d} + \left\langle\gamma(\eta)\eta^{2d}\right\rangle,\end{aligned} \tag{8.21}$$

and the optimization problem Eq. (8.20) can be written as

$$\min_{\mathbf{a}_d\in\mathbb{R}^{2M+1}}(\mathbf{a}_d^T\mathbf{Q}_d\mathbf{a}_d - 2\mathbf{a}_d^T\mathbf{r}_d), \text{ subject to Eq. (8.4)}, \tag{8.22}$$

where the constant term $\left\langle\eta^{2d}\right\rangle$ is ignored in the minimization. The system Eq. (8.22) is a quadratic programming problem, with linear equality constraints. We can solve this problem analytically, which is determined from the Karush-Kuhn-Tucker (KKT) condition [147]. The optimal solution for the coefficients $\mathbf{a}_d$, which will be denoted with an asterisk ($\mathbf{a}_d^*$), satisfies the following KKT

condition

$$\underbrace{\begin{bmatrix} \mathbf{Q}_d & \mathbf{X}_d \\ \mathbf{X}_d^T & \mathbf{0}_{(d+p+1)\times(d+p+1)} \end{bmatrix}}_{:=\mathbf{K}} \begin{pmatrix} \mathbf{a}_d^* \\ \boldsymbol{\lambda}_d^* \end{pmatrix} = \begin{pmatrix} \mathbf{r}_d \\ \mathbf{y}_d^T \end{pmatrix}, \tag{8.23}$$

where $\boldsymbol{\lambda}_d \in \mathbb{R}^{d+p+1}$ is the Lagrange multiplier associated with the constraint in Eq. (8.4). Since $\mathbf{X}_d^T \in \mathbb{R}^{(d+p+1)\times S}$ has full row rank, then the KKT matrix $\mathbf{K}$ is nonsingular and Eq. (8.23) has unique solution $(\mathbf{a}_d^*, \boldsymbol{\lambda}_d^*)$ given by

$$\begin{pmatrix} \mathbf{a}_d^* \\ \boldsymbol{\lambda}_d^* \end{pmatrix} = \begin{bmatrix} \mathbf{Q}_d & \mathbf{X}_d \\ \mathbf{X}_d^T & \mathbf{0}_{(d+p+1)\times(d+p+1)} \end{bmatrix}^{-1} \begin{pmatrix} \mathbf{r}_d \\ \mathbf{y}_d^T \end{pmatrix}. \tag{8.24}$$

Since everything on the right-hand-side is known, Eq. (8.24) provides the coefficients $\mathbf{a}_d^*$ for the finite difference approximation Eq. (8.2) of order $p+1$ which minimize the spectral error Eq. (8.19).

As an example consider a second-order scheme for a second derivative. In this case, a 3-point stencil ($S = 2M+1 = 3$) yields a unique solution to Eq. (8.4), namely the common approximation $\mathbf{a}_2^* = \begin{bmatrix} 1 & -2 & 1 \end{bmatrix}^T$ or $f_i' \approx (f_{i-1} - 2f_i + f_{i+1})/\Delta x^2$. If one now retains a second-order approximation but increases the stencil size S, the additional degrees of freedom are utilized to reduce, through the minimization process, the spectral error in the approximation. For illustration purposes assume we would like to resolve as accurately as possible all wavenumbers in the range $\eta \in [0, 2.5]$. In this case, one can naturally choose $\gamma(\eta) = 1$ for $\eta \in [0, 2.5]$ and $\gamma(\eta) = 0$ otherwise. Upon solving Eq. (8.24), we obtain the coefficients of the resulting schemes which are shown in Fig. 8.1 and Table 8.1.

The spectral accuracy of these schemes is shown in Fig. 8.2 where we compare $\widetilde{\eta}^2$ (magenta) to η^2 (dashed black) as in Eq. (8.17): the difference between these two curves correspond the error representing the derivative at a given frequency. The standard second order scheme, $M = 1$, is shown with the red line for comparison. In the top panels of the figure, we clearly see that as we

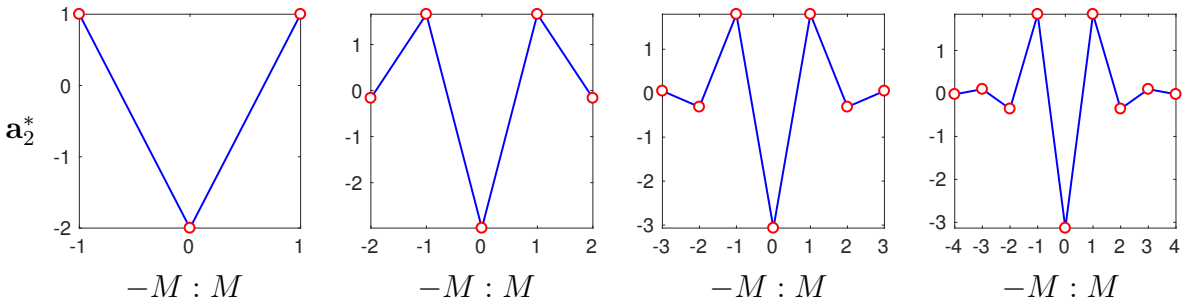

Figure 8.1: Optimal coefficients for various stencil sizes $S = 2M + 1$, for a second derivative with second order accuracy. The coefficients were obtained using $\gamma(\eta) = 1$, for $\eta \in [0, 2.5]$ and $\gamma(\eta) = 0$ otherwise.

	$M = 1$	$M = 2$	$M = 3$	$M = 4$
a_0^*	-2	-2.986945912146335	-3.067324780469417	-3.132525936497260
a_1^*	1	1.657963941430890	1.795865984254199	1.843958787844204
a_2^*		-0.164490985357722	-0.312793272384242	-0.357929955982910
a_3^*			0.050589678364752	0.099426449444277
a_4^*				-0.019192313056941

Table 8.1: Numerical coefficients for the schemes in Fig. 8.1. Note that since the resulting schemes are symmetric around the central grid point, only coefficients on one side are shown.

increase M a better representation of the exact derivative is achieved in the range $\eta \in [0, 2.5]$ as expected. There is also a stark difference between the standard and the optimized schemes which becomes more prominent as η or M increases.

Further information about the nature of the error can be obtained by analyzing the real and imaginary components of $e(\eta)$. This is shown in the middle and bottom panels of Fig. 8.2. While there are significant errors in the real component $\Re[e(\eta)]$ for $\eta \in [0, 2.5]$ when $M = 1$, they decrease substantially when M is increased, as expected. The errors outside of the support of $\gamma(\eta)$ are marginally affected since those are not modes for which the minimization process seeks optimal solutions. The imaginary part $\Im[e(\eta)]$ on the other hand presents zero error. This fact can indeed be proven and is presented here as,

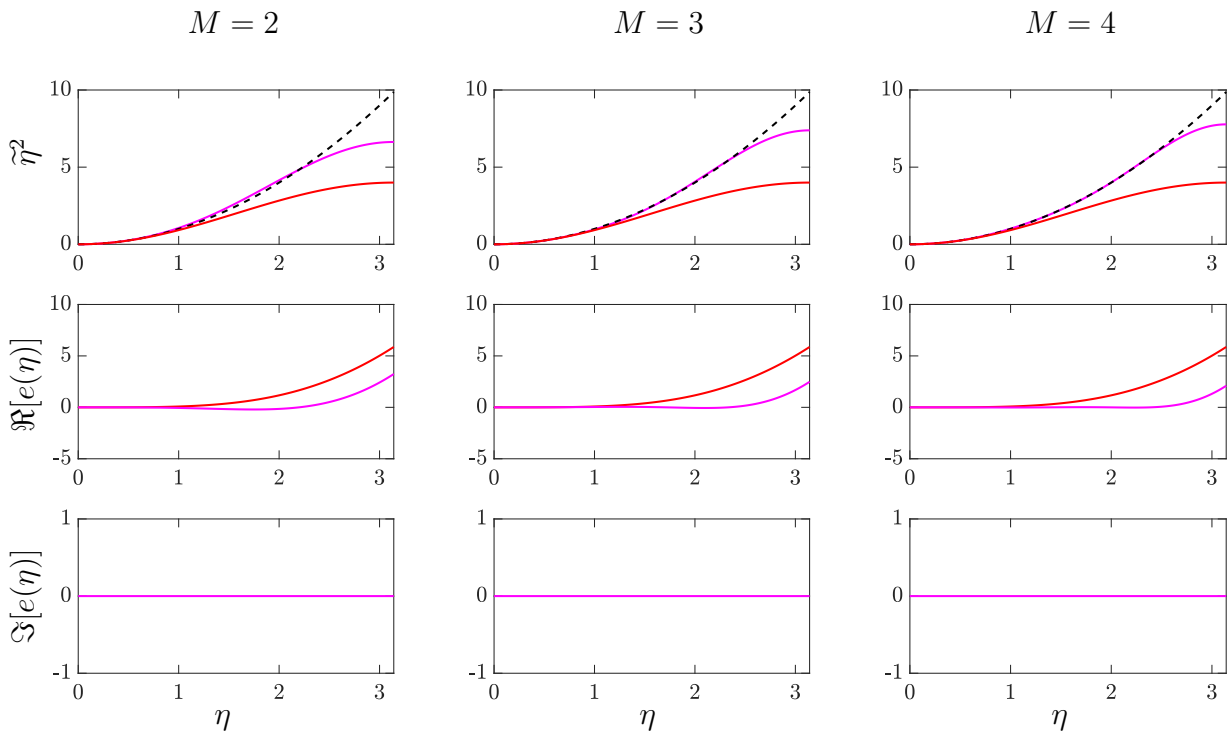

Figure 8.2: Spectral accuracy for the schemes presented in Fig. 8.1 and Table 8.1. Top, middle and bottom rows show the modified wavenumber, the real part, and the imaginary part of the spectral error, respectively. Left, middle and right columns correspond to $M = 2$, 3, and 4, respectively. Dashed black line: exact differentiation ($\widetilde{\eta}^2 = \eta^2$). Magenta solid line: $\widetilde{\eta}$ for optimized schemes. In all plots, red solid line corresponds to standard second-order scheme ($M = 1$) for comparison.

Lemma 3. *Finite difference approximations Eq. (8.2) obtained from Eq. (8.20) with d even, yield a spectral error Eq. (8.14) which satisfies $\Im[e(\eta)] = 0$.*

Proof. See Appendix 8.A. □

This result has implications in terms of the type of error expected (dispersive versus dissipative) when the scheme is used in a fully discretized PDE. In particular, from Eq. (8.18) we have $G_2 = \widetilde{\eta}^2/\eta^2 = (-e(\eta)/\eta^2 + 1)$ which implies that the modified wavenumber will not have a phase error since $\Im[e(\eta)] = 0$. Only dissipative errors are thus expected.

We note here that the error has also been defined using an additional arbitrary coefficient [138, 134] to emphasize the relative importance of the dissipative and dispersive errors. That is, one modifies Eq. (8.21) to $\|e(\eta)\|^2_{\mathcal{L}_2} = \int_0^\pi \gamma(\eta)[\sigma(\mathbf{C}^T(\eta)\mathbf{a}_d - (-1)^q\eta^d)^2 + (1-\sigma)(\mathbf{S}(\eta)^T\mathbf{a}_d)^2]d\eta$

where $\sigma \in [0, 1]$. However, from the proof of lemma 3, the minimum value of the error is obtained when one of the integrals is identically zero. Therefore the parameter σ will only appear as a constant and will have no effect on the optimal coefficients. The parameter σ, thus, is of relevance only when the grid is biased in one direction. In this scenario, neither of the integrals in Eq. (8.21) will be 0, resulting in error with both real and imaginary part. The optimal solution will then be a minimum of the summation of dispersive and dissipative errors.

It is also of interest to understand how errors change with S (or M). This is so because increasing N (number of grid points) or S (stencil size) leads to more computations and, thus, more computationally intensive simulations. For standard schemes, when one increases the number of points in the stencil (S), one also increases the order of accuracy. In particular, the error for a given S (or M) is proportional to Δx^{2M} where the proportionality constant also decreases with M. While there is no easy way to represent these constants in closed form, from their structure one may then still expects an approximately exponential decrease in the error as the stencil size increases. This is indeed what we see in Fig. 8.3 where we show the error $\|e(\eta)\|_{\mathcal{L}_2}^2$ as a function of M (black symbols). For optimized schemes of fixed order, on the other hand, it is not obvious a priori the rate of convergence with the number of grid points used in the stencil. However, we can readily evaluate the error numerically to assess this convergence rate. This is shown in Fig. 8.3 where we include the result of such numerical calculations with the second-order optimized schemes (blue symbols) in Fig. 8.1 as M is increased. It can be seen that the error also decreases approximately exponentially with M. While for $M = 1$ the two curves coincide as expected since both approaches lead to the same scheme, optimized schemes (blue) present a better convergence rate as the stencil size increases.

8.2.2.2 Odd derivatives

For $d = 2q + 1$, $q = \{0, 1, \cdots\}$, $e(\eta)$ becomes

$$e(\eta) = \mathbf{C}^T(\eta)\mathbf{a}_d + j\left(\mathbf{S}^T(\eta)\mathbf{a}_d - (-1)^q\eta^d\right).$$

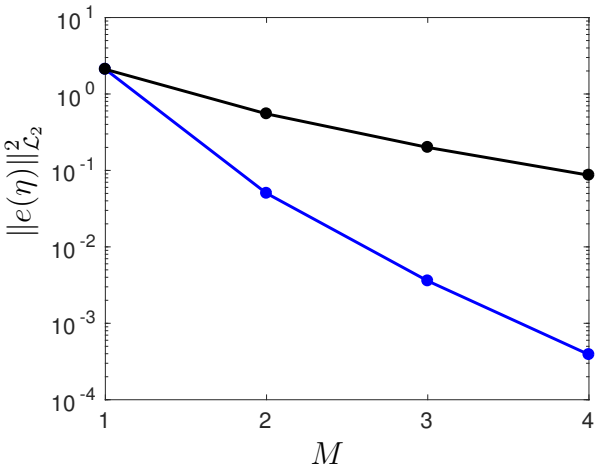

Figure 8.3: Optimal spectral error for various stencil size M, approximating second derivative. The blue line corresponds to the second order optimized schemes presented in Fig. 8.1 and Table 8.1. The black line corresponds to the standard schemes with stencil size M and $2M$ order of accuracy.

The $\mathcal{L}_2$-norm of the error for this case is

$$\begin{aligned}\|e(\eta)\|^2_{\mathcal{L}_2} &:= \int_0^\pi \gamma(\eta)\left[\left(\mathbf{C}^T(\eta)\mathbf{a}_d\right)^2 + \left(\mathbf{S}^T(\eta)\mathbf{a}_d - (-1)^q\eta^d\right)^2\right] d\eta, \\ &= \mathbf{a}_d^T \underbrace{\left(\left\langle\gamma(\eta)\mathbf{C}(\eta)\mathbf{C}^T(\eta)\right\rangle + \left\langle\gamma(\eta)\mathbf{S}(\eta)\mathbf{S}^T(\eta)\right\rangle\right)}_{\mathbf{Q}_d}\mathbf{a}_d \\ &- 2\mathbf{a}_d^T \underbrace{\left\langle\gamma(\eta)(-1)^q\eta^d\mathbf{S}(\eta)\right\rangle}_{\mathbf{r}_d} + \left\langle\gamma(\eta)\eta^{2d}\right\rangle.\end{aligned} \tag{8.25}$$

The optimization problem is the same as Eq. (8.22) with $\mathbf{r}_d$ given by Eq. (8.25), and the optimal solution is given by Eq. (8.24).

As an example, we consider here again a second-order approximation but of the first derivative for increasing values of M. The resulting schemes are shown in Fig. 8.4 and their spectral behavior is shown in Fig. 8.5. Similar conclusions to the second derivative example shown above are observed. As M increases the wavenumbers where $\gamma(\eta)$ is non-zero are increasingly well resolved which is seen as $\widetilde{\eta}$ becoming closer to η in the top panels of Fig. 8.5. We can also see that the real component of the spectral error (middle panels) is zero. This is due to the following lemma.

Lemma 4. *Finite difference approximations Eq. (8.2) obtained from Eq. (8.20) with d odd, yield a spectral error Eq. (8.14) which satisfies $\Re[e(\eta)] = 0$.*

Proof. See Appendix 8.B. □

Since $G_1 = e(\eta)/(j\eta) + 1$, and $e(\eta)$ has only an imaginary component, then G_1 is real. As in the example before, errors are then expected to be dispersive in nature though the exact nature of the error would depend on the PDE in which such a scheme is used. As before, we found the global error to decrease exponentially with M as seen in Fig. 8.6 and faster than non-optimized schemes.

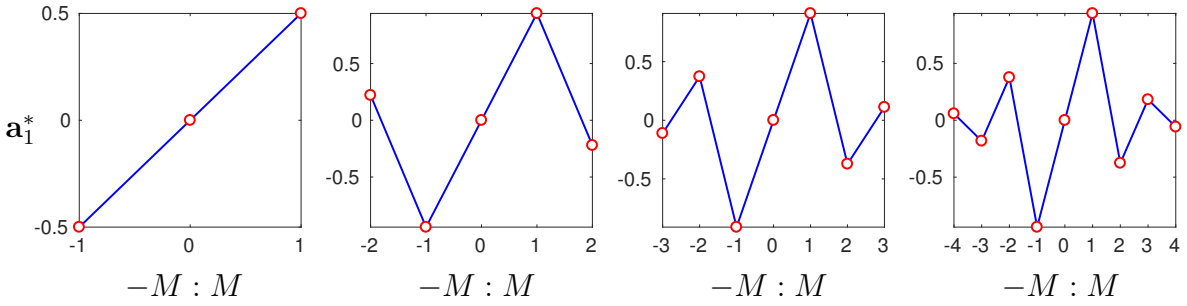

Figure 8.4: Optimal coefficients for various stencil size M, approximating first derivative with second order accuracy. The coefficients were obtained using $\gamma(\eta) = 1$, for $\eta \in [0, 2.5]$; and $\gamma(\eta) = 0$ otherwise.

	$M=1$	$M=2$	$M=3$	$M=4$
a_1^*	0.50	0.941502204636976	0.911624839168511	0.939273151104227
a_2^*		-0.220751102318488	-0.372951233396604	-0.376375957228243
a_3^*			0.111425875874899	0.182092697439389
a_4^*				-0.058199832241477

Table 8.2: Numerical coefficients for the schemes in Fig. 8.4. Note that since the resulting schemes are anti-symmetric ($a_{-M}^* = -a_M^*$) around the central grid point, only coefficients on one side are shown.

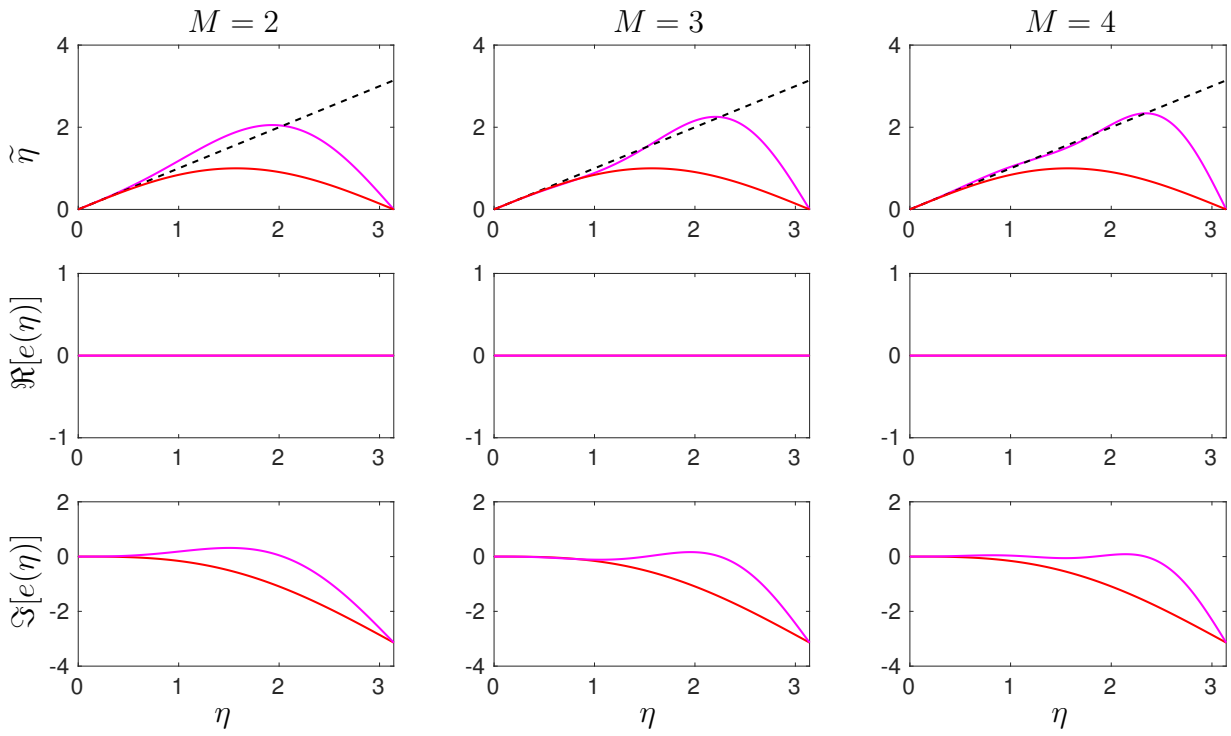

Figure 8.5: Spectral accuracy for the schemes presented in Fig. 8.4 and Table 8.2. Top, middle and bottom rows show the modified wavenumber, the real part, and the imaginary part of the spectral error, respectively. Left, middle and right columns correspond to $M = 2$, 3, and 4, respectively. Dashed black line: exact differentiation ($\widetilde{\eta} = \eta$). Magenta solid line: $\widetilde{\eta}$ for optimized schemes. In all plots, red solid line corresponds to standard second-order scheme ($M = 1$) for comparison.

8.2.2.3 Effect of $\gamma(\eta)$

An important feature of the framework presented here for optimized schemes is that depending upon the physics of the problem, one can choose what scales need to be properly resolved. By setting $\gamma(\eta)$ in Eq. (8.19) as unity, equal weight is given to all wavenumbers while solving the minimization problem. The formulation, however, is more general and allow us to change $\gamma(\eta)$ to, for example, resolve a subset of wavenumbers more accurately than others. To illustrate this, consider an eight-order standard scheme ($M = 4$) for the second derivative whose modified wavenumber can be readily computed analytically. The relative error at each wavenumber, which with Eq. (8.18) can be written as $|\widetilde{\eta}^2 - \eta^2|/\eta^2 = G_2 - 1$, is shown in Fig. 8.7 as a black line. We

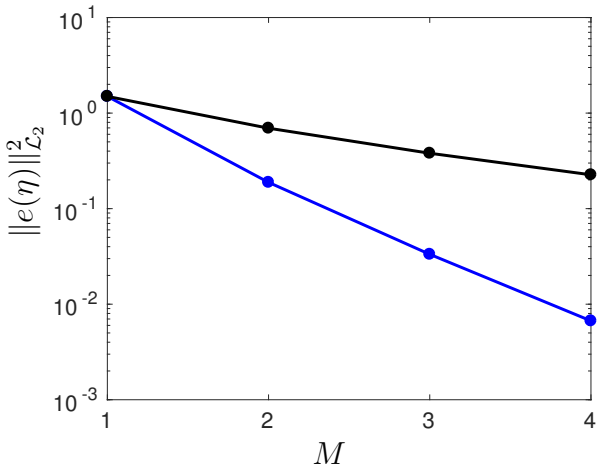

Figure 8.6: Optimal spectral error for various stencil size M, approximating first derivative. The blue line corresponds to the second order optimized schemes presented in Fig. 8.4 and Table 8.2. The black line corresponds to the standard schemes with stencil size M and $2M$ order of accuracy.

can see that standard schemes have very low errors at low η but become progressively worse at high η. In fact, there are about 15 orders of magnitude difference between the error incurred at low and high wavenumbers. This situation may present some challenges when these schemes are used to resolve multiscale problems where all wavenumbers contribute to the dynamics. In contrast, the optimized scheme with $\gamma(\eta) = 1$ in $\eta \in [0, 2.5]$ shown in magenta in the figure, results in a much flatter error in spectral space. The oscillatory nature of the error is due to the following. The optimized schemes minimize a global measure of the error based on the $\mathcal{L}_2$ norm of difference between $\widetilde{\eta}^d$ and η^d. Pointwise however, the resulting scheme produces a $\widetilde{\eta}$ that can be above or below η; only the appropriate integral is minimized. An easy-to-see example of this behavior is seen in Fig. 8.5 for $M = 3$ (top-middle panel). The crossing points between $\widetilde{\eta}$ and η leads then to zero error which correspond to the down peaks in Fig. 8.7 for the corresponding scheme.

The other curves in Fig. 8.7 illustrate the effect of changing $\gamma(\eta)$ on the spectral error. For this illustration we use $\gamma(\eta) = e^{-\eta/\psi}$ where ψ is a constant that controls how quickly the weighting function drops to zero near the origin. A similar form has been used in some applications [138, 136] though for some ψ. As $\gamma(\eta)$ becomes steeper (small ψ), the error for lower and higher

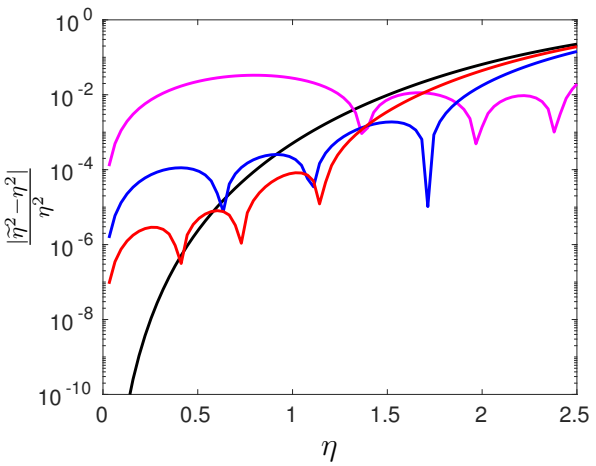

Figure 8.7: Spectral error for various $\gamma(\eta)$ for second derivative. Standard eighth order scheme (black), $\gamma(\eta) = 1$ (magenta), $\gamma(\eta) = exp(-\eta/0.1)$ (blue) and $\gamma(\eta) = exp(-\eta/0.06)$ (red) for $\eta \in [0, 2.5]$ and 0 otherwise.

wavenumbers decreases and increases, respectively. The optimized schemes then get closer to the strongly non-uniform distribution of error in spectral space of standard schemes.

Clearly one can use, for example, a banded $\gamma(\eta)$, such that it is non-zero only for some intermmediate wavenumbers. Or one may need to resolve to distinct bands in wavenumber space which can easily be accommodated by an appropriate choice of $\gamma(\eta)$. The choice of $\gamma(\eta)$ provides a high flexibility to select how the available information (the value of the function at S grid points) to resolve only the scales relevant to the problem being solved.

8.2.3 Stability

As discussed in the introduction, it is common to first choose a scheme of given order and then verify that spectral accuracy is acceptable. That scheme is then used to discretize a PDE. However, before this numerical arrangement can be utilized, one needs to determine whether the fully discretized PDE is stable. Different methods to assess stability of a given discretized equation have been discussed extensively [20]. Our objective here is not simply to determine whether the schemes developed above are stable or not. Rather is to incorporate the stability requirement into a unified formulation. This framework would thus provide, for a given M, a stable scheme of given order with the best spectral resolution possible for wavenumbers of interest.

For this, consider the general linear partial differential equation

$$\frac{\partial f}{\partial t} = \sum_{d=1}^{D} \beta_d \frac{\partial^d f}{\partial x^d}. \tag{8.26}$$

discretized over the entire domain with $N := 2M_{\text{max}} + 1$ grid points. As before, spatial derivatives are approximated using a stencil of size $S = 2M + 1$, where M can take values from 1 to M_{max}. The d^{th} derivative at the i^{th} grid point is parameterized by $\mathbf{a}_{i,d} \in \mathbb{R}^S$. Define $\mathbf{A}_d \in \mathbb{R}^{N \times S}$ to be the vertical stacking of $\mathbf{a}_{i,d}^T$, i.e.

$$\mathbf{A}_d := \begin{bmatrix} \mathbf{a}_{1,d}^T \\ \vdots \\ \mathbf{a}_{N,d}^T \end{bmatrix}. \tag{8.27}$$

The order accuracy constraint Eq. (8.4) for every $\mathbf{a}_{i,d}$, can then be compactly written as

$$\mathbf{A}_d \mathbf{X}_d = \mathbf{Y}_d, \tag{8.28}$$

for $d = 1, \cdots, D$; and $\mathbf{Y}_d := \mathbf{1}_{N \times 1} \otimes \mathbf{y}_d$.

The cost function to be minimized is the sum of the spectral error Eq. (8.19) at all locations i:

$$\begin{aligned} \sum_{i=1}^{N} \|e_i(\eta)\|_{\mathcal{L}_2}^2 &:= \sum_{i=1}^{N} \sum_{d=1}^{D} \left(\mathbf{a}_{i,d}^T \mathbf{Q}_d \mathbf{a}_{i,d} - 2\mathbf{a}_{i,d}^T \mathbf{r}_d \right) + N \left\langle \eta^{2d} \right\rangle, \\ &= \sum_{d=1}^{D} \mathbf{v}_d^T (\mathbf{I}_N \otimes \mathbf{Q}_d) \mathbf{v}_d - 2\mathbf{v}_d^T (\mathbf{1}_{N \times 1} \otimes \mathbf{r}_d), \end{aligned} \tag{8.29}$$

where $\mathbf{v}_d := \mathbf{vec}(\mathbf{A}_d^T)$, and $\mathbf{vec}(\cdots)$ vectorizes a matrix by vertically stacking the columns. The minimization of Eq. (8.29) subjected to Eq. (8.28) leads to optimal schemes with a given order of accuracy. To include stability in the formulation, we first define the vectors

$$\mathbf{F} := \begin{pmatrix} f_1 \\ \vdots \\ f_N \end{pmatrix}, \text{ and } \mathbf{F}^{(d)} := \begin{pmatrix} f_1^{(d)} \\ \vdots \\ f_N^{(d)} \end{pmatrix}. \tag{8.30}$$

Using these definitions, the finite-difference approximation for Eq. (8.26) for all the grid points can be written compactly as

$$\mathbf{F}^{(d)} = \frac{1}{(\Delta x)^d} \mathbf{A}_d^{\mathbf{\Phi}} \mathbf{F}, \tag{8.31}$$

where the matrix $\mathbf{A}_d^{\mathbf{\Phi}}$ contains the unknown coefficients $\mathbf{a}_{i,d}$ arranged appropriately for the correct computation of the d^{th} derivative at i^{th} location. The matrix $\mathbf{A}_d^{\mathbf{\Phi}}$ can be constructed using shift operators the details of which are included in Appendix 8.C. The important element here is that the spatial derivative is a linear operator acting on the value of the function f at all grid points.

8.2.3.1 Stability of Semi-discrete Scheme

With the spatial discretization from Eq. (8.31), Eq. (8.26) can be written as

$$\dot{\mathbf{F}} = \left(\sum_{d=1}^{D} \frac{1}{(\Delta x)^d} \beta_d \mathbf{A}_d^{\mathbf{\Phi}} \right) \mathbf{F}, \tag{8.32}$$

whose analytical solution can be readily obtained as

$$\mathbf{F}(t) := \exp\left(\sum_{d=1}^{D} \frac{1}{(\Delta x)^d} \beta_d t \mathbf{A}_d^{\mathbf{\Phi}} \right) \mathbf{F}_0, \tag{8.33}$$

where $\mathbf{F}_0$ is the initial condition. Clearly the stability of Eq. (8.33) is governed by the coefficients $\mathbf{A}_d^{\mathbf{\Phi}}$.

In order to understand the conditions under which the scheme defined by $\mathbf{A}_d^{\mathbf{\Phi}}$ is stable consider a single Fourier mode as in Eq. (8.11). The approximate d^{th} derivative can be conveniently written

in terms of the modified wavenumber as $(j\widetilde{\eta})^d \hat{f}$ and the original PDE becomes

$$\frac{d\hat{f}}{dt} = \sum_{d=1}^{D} \beta_d (j\widetilde{\eta})^d \hat{f}. \tag{8.34}$$

The solution, with $\hat{f}_0$ being the value of $\hat{f}$ at $t = 0$, is given by

$$\frac{\hat{f}}{\hat{f}_0} = \exp\left[\sum_{d=1}^{D} \beta_d (j\widetilde{\eta})^d t\right]. \tag{8.35}$$

If the solution to the original PDE is non-increasing in time, then the discretization is considered stable if no Fourier mode grows in time. This is clearly satisfied if

$$\Re\left\{\sum_{d=1}^{D} \beta_d (j\widetilde{\eta})^d\right\} \leq 0 \tag{8.36}$$

From lemma 3 and 4, we can conclude that

$$(j\widetilde{\eta})^d = (\mathbf{C}^T(\eta) + j\mathbf{S}^T(\eta))\mathbf{a}_d = \begin{cases} j\mathbf{S}^T(\eta)\mathbf{a}_d, \text{ for odd derivative,} \\ \mathbf{C}(\eta)^T\mathbf{a}_d, \text{ for even derivative,} \end{cases}$$

because $\mathbf{C}^T(\eta)\mathbf{a}_d = 0$ for odd derivatives and $\mathbf{S}^T(\eta)\mathbf{a}_d = 0$ for even derivatives. Then,

$$\begin{aligned} \sum_{d=1}^{D} \beta_d (j\widetilde{\eta})^d &= j\beta_1 \mathbf{S}^T(\eta)\mathbf{a}_1 + \beta_2 \mathbf{C}^T(\eta)\mathbf{a}_2 + j\beta_3 \mathbf{S}^T(\eta)\mathbf{a}_3 + \beta_4 \mathbf{C}^T(\eta)\mathbf{a}_4 + \cdots, \\ &= \left(\beta_2 \mathbf{C}^T(\eta)\mathbf{a}_2 + \beta_4 \mathbf{C}^T(\eta)\mathbf{a}_4 + \cdots\right) + j\left(\beta_1 \mathbf{S}^T(\eta)\mathbf{a}_1 + \beta_3 \mathbf{S}^T(\eta)\mathbf{a}_3 + \cdots\right). \end{aligned} \tag{8.37}$$

Therefore, Eq. (8.36) implies

$$\beta_2 \mathbf{C}^T(\eta)\mathbf{a}_2 + \beta_4 \mathbf{C}^T(\eta)\mathbf{a}_4 + \cdots \leq 0 \tag{8.38}$$

which provides a general constraint to assure stability of the semidiscrete system.

Clearly stability for the semi-discrete system depends only on the even derivatives. In that case

d can be written as $2q$ for $q = 1, 2, \cdots$, and $(j\widetilde{\eta})^d = (-1)^q \widetilde{\eta}^{2q}$. Since the optimization of spectral error guarantees $(j\widetilde{\eta})^d = \mathbf{C}^T(\eta)\mathbf{a}_{2q}$, we have $\mathbf{C}^T(\eta)\mathbf{a}_{2q} = (-1)^q \widetilde{\eta}^{2q}$. Consequently, the sign of $\mathbf{C}^T(\eta)\mathbf{a}_{2q}$ alternates with q, i.e.

$$\begin{aligned}
&\text{for } q = 1,\ \mathbf{C}^T(\eta)\mathbf{a}_2 = -\widetilde{\eta}^2 <= 0,\\
&\text{for } q = 2,\ \mathbf{C}^T(\eta)\mathbf{a}_4 = \widetilde{\eta}^4 >= 0,\\
&\text{for } q = 3,\ \mathbf{C}^T(\eta)\mathbf{a}_6 = -\widetilde{\eta}^6 <= 0,
\end{aligned}$$

and so on. Therefore, if the coefficients of the original PDE can be written as $\beta_{2q} := (-1)^{q+1}\gamma_{2q}^2$ for some γ_{2q}, then Eq. (8.38) is implicitly satisfied as it reduces to a sum of negative numbers. In general, if β_{2q} does not have sign as required by $(-1)^{q+1}\gamma_{2q}^2$, then, while that particular derivative is unstable, the system is still stable if

$$-\beta_2\widetilde{\eta}^2 + \beta_4\widetilde{\eta}^4 - \beta_6\widetilde{\eta}^6 + \cdots \leq 0. \tag{8.39}$$

8.2.3.2 Stability of Fully-Discrete Scheme

While semi-discrete analyses provide some information about the temporal behavior of the discretized spatial derivatives in the PDEs, one is ultimately interested in the stability of the fully discretized system. For concreteness assume that the time discretization of Eq. (8.26) is done using a *forward difference*. With Eq. (8.31) for the spatial discretization we write Eq. (8.26) as:

$$\mathbf{F}^{k+1} = \left(\mathbf{I}_N + \sum_d \frac{\Delta t}{(\Delta x)^d}\beta_d \mathbf{A}_d^{\mathbf{\Phi}}\right)\mathbf{F}^k. \tag{8.40}$$

This is a linear discrete-time system in $\mathbf{F}$, where the system matrix is linearly dependent on the stencil coefficients $\mathbf{A}_d$, discretization parameters Δt and Δx, and the coefficients β_d. In what follows we assume Δx and β_d are given which is a typical situation in simulations of physical systems. The objective then is to determine Δt and $\mathbf{A}_d$ so that stability is achieved subjected to

specifications in order of accuracy and spectral resolution.

In general, stability is guaranteed if the spectral radius of the evolution matrix is bounded by unity [20]:

$$\lambda_{\max}\left(\mathbf{I}_N + \sum_d \frac{\Delta t}{(\Delta x)^d}\beta_d \mathbf{A}_d^{\mathbf{\Phi}}\right) \leq 1. \tag{8.41}$$

Because the spectral radius is bounded by matrix norms, we guarantee stability by bounding the 2-norm, i.e.

$$\left\|\mathbf{I}_N + \sum_d \frac{\Delta t}{(\Delta x)^d}\beta_d \mathbf{A}_d^{\mathbf{\Phi}}\right\|_2 \leq 1. \tag{8.42}$$

Using Schur complements, this inequality is equivalent to

$$\begin{bmatrix} \mathbf{I}_N & \left(\mathbf{I}_N + \sum_d \frac{\Delta t}{(\Delta x)^d}\beta_d \mathbf{A}_d^{\mathbf{\Phi}}\right)^T \\ \left(\mathbf{I} + \sum_d \frac{\Delta t}{(\Delta x)^d}\beta_d \mathbf{A}_d^{\mathbf{\Phi}}\right) & \mathbf{I}_N \end{bmatrix} \geq 0. \tag{8.43}$$

This (inequality) constraint coupled with the order of accuracy (equality) constraint Eq. (8.28) completes the set of constraints for the optimization problem aimed at minimizing the spectral error given by Eq. (8.29).

Unfortunately, the matrix inequality in Eq. (8.43) has products of Δt and $\mathbf{A}_d$, which makes the problem non convex. Thus, more general techniques than convex optimization have to be employed to solve the problem. Here we present two approaches. In the first approach, we determine $\mathbf{A}_d$ analytically that minimizes the spectral error and then maximize Δt for which stability is achieved. In the second approach, we assume a value for Δt and determine $\mathbf{A}_d$ which minimizes spectral error and guarantees stability. Both these approaches are discussed in detail below.

8.2.3.3 Given spectrally optimal $\mathbf{A}_d$, maximize Δt and guarantee stability simultaneously

In this approach, we solve for the optimal $\mathbf{a}_{i,d}$ analytically that, for a given order, minimizes the spectral error and then maximize Δt for which stability is guaranteed. Optimization of the spectral

error with given order of accuracy is given by previous formulation as

$$\min_{\{\mathbf{A}_d\}_{d=1}^{D}} \quad \sum_{d=1}^{D} \mathbf{v}_d^T(\mathbf{I}_N \otimes \mathbf{Q}_d)\mathbf{v}_d - 2\mathbf{v}_d^T(\mathbf{1}_{N\times 1} \otimes \mathbf{r}_d),$$

subject to

$$\mathbf{A}_d\mathbf{X}_d = \mathbf{Y}_d, \text{ for } d = 1, \cdots, D,$$

where $\mathbf{v}_d := \mathbf{vec}(\mathbf{A}_d^T)$.

We observe that the cost and constraint functions are separable with respect to d, and thus can be independently optimized using,

$$\min_{\mathbf{A}_d} \quad \sum_{d=1}^{D} \mathbf{v}_d^T(\mathbf{I}_N \otimes \mathbf{Q}_d)\mathbf{v}_d - 2\mathbf{v}_d^T(\mathbf{1}_{N\times 1} \otimes \mathbf{r}_d), \text{ subject to } \mathbf{A}_d\mathbf{X}_d = \mathbf{Y}_d, \tag{8.44}$$

for $d = 1, \cdots, D$. The linear constraint $\mathbf{A}_d\mathbf{X}_d = \mathbf{Y}_d$ can be written as

$$\left(\mathbf{I}_N \otimes \mathbf{X}_d^T\right)\mathbf{v}_d = \mathbf{Y}_d^T.$$

Therefore, the optimization problem in Eq. (8.44) can be written as

$$\min_{\mathbf{v}_d} \quad \mathbf{v}_d^T\left(\mathbf{I}_N \otimes \mathbf{Q}_d\right)\mathbf{v}_d - 2\mathbf{v}_d^T\left(\mathbf{1}_{N\times 1} \otimes \mathbf{r}_d\right), \tag{8.45}$$

$$\text{subject to } \left(\mathbf{I}_N \otimes \mathbf{X}_d^T\right)\mathbf{v}_d = \mathbf{Y}_d^T, \tag{8.46}$$

which has the analytical solution

$$\begin{pmatrix} \mathbf{v}_d \\ \mathbf{\Lambda}_d \end{pmatrix}^* = \begin{bmatrix} \mathbf{I}_N \otimes \mathbf{Q}_d & \mathbf{I}_N \otimes \mathbf{X}_d \\ \mathbf{I}_N \otimes \mathbf{X}_d^T & \mathbf{I}_N \otimes \mathbf{0} \end{bmatrix}^{-1} \begin{pmatrix} \mathbf{1}_{N\times 1} \otimes \mathbf{r}_d \\ \mathbf{Y}_d^T \end{pmatrix}, \tag{8.47}$$

where $\mathbf{\Lambda}_d$ is the vector of Lagrange multipliers associated with the constraints. Finally, we can

recover $\mathbf{A}_d^*$, the coefficients of the scheme that minimizes the spectral error and is stable, from $\mathbf{v}_d^*$. While intuitively it is clear that this solution should be the same at all grid points in the domain, here we present, for completion, a proof as:

Lemma 5. *Optimal explicit symmetrical finite-difference approximation is invariant of the grid point location in the domain.*

Proof. See Appendix 8.D. □

From Lemma 5, we write the complete system as a stacking of local solutions:

$$\mathbf{A}_d^* = \mathbf{1}_{N\times 1} \otimes \mathbf{a}_d^{*T}. \tag{8.48}$$

Therefore, the maximum Δt for which stability is guaranteed is obtained by solving the optimization problem Eq. (8.43), that is

$$\max_{\Delta t} \text{ subject to } \begin{bmatrix} \mathbf{I}_N & \left(\mathbf{I} + \sum_d \frac{\Delta t}{(\Delta x)^d}\beta_d \mathbf{A}_d^{\mathbf{\Phi}}\right)^T \\ \left(\mathbf{I} + \sum_d \frac{\Delta t}{(\Delta x)^d}\beta_d \mathbf{A}_d^{\mathbf{\Phi}}\right) & \mathbf{I}_N \end{bmatrix} \geq 0, \tag{8.49}$$

where $\mathbf{A}_d^{\mathbf{\Phi}}$ is determined using Eq. (8.76) and $\mathbf{A}_d^*$ from Eq. (8.48). The optimization in Eq. (8.49) is a convex optimization problem in Δt and can be efficiently solved numerically using software such as `cvx`[148]. This formulation can then be used to obtain stability limits of the fully-discrete system.

As an example, consider an advection-diffusion equation, that is Eq. (8.26) with $D = 2$. For given values of β_1, $\beta_2 > 0$, and Δx, one can solve the optimization problem to obtain the largest Δt for which a previously obtained optimal $\mathbf{A}_d^*$ remain stable. Note that the stability of the scheme will depend only on the non-dimensional parameters $\Delta t\beta_d/\Delta x^d$. For $D = 2$, these are commonly called convective and diffusive CFL, that is, $r_c := \Delta t\beta_1/\Delta x$ and $r_d := \Delta t\beta_2/\Delta x^2$, respectively. Stability regions can then be obtained by e.g. sweeping values of Δx and plot results in terms of r_c and r_d.

This is what we show in Fig. 8.8 for the advection-diffusion equation for different values of M. For $M = 1$ the numerically evaluated stability region is the same as the analytical form found in textbooks for the standard second-order 3-point stencil for first and second derivatives, namely $r_c^2 \leq 2r_d \leq 1$ [20]. As M is increased the stability region becomes smaller. This is also generally consistent with standard schemes for which as the order (and stencil size) increases the stability region shrinks. In both cases, this is related to the decreasing dissipation at high wavenumbers as spectral resolution improves which could trigger instabilities.

In summary, the formulation here allows us to obtain, for optimal schemes of given order and maximum spectral resolution, the largest step size which guarantees stability or, more generally, regions of stability.

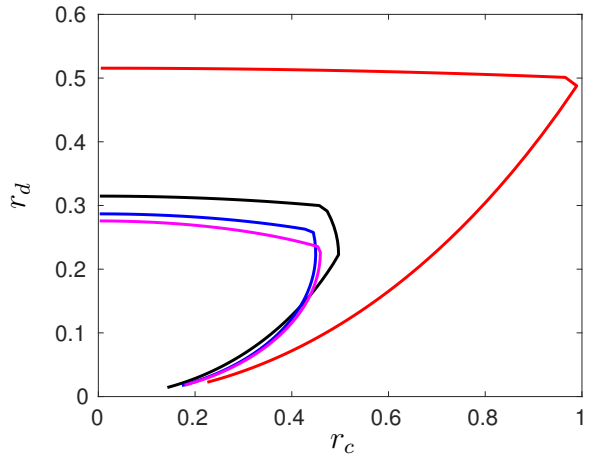

Figure 8.8: Stability region for the advection-diffusion equation with optimal coefficients $\mathbf{a}_1^*$ and $\mathbf{a}_2^*$ for different stencil sizes: $M = 1$ (red), $M = 2$ (black), $M = 3$ (blue), $M = 4$ (magenta).

8.2.3.4 Given Δt, optimize spectral error and guarantee stability simultanously

In this approach, for a given Δt we solve for $\mathbf{A}_d$ that simultaneously minimizes spectral error, achieves given order accuracy, and guarantees stability. This approach is of practical relevance as the time-step Δt would depend on the fastest physical process in the problem. For example, when

chemical reactions are present in a flow, the Damkohler number (the ratio of flow time scales to chemistry time scales) can be very high which means the time scales of reactions are considerably smaller than flow time scales which forces Δt to be small for an accurate solution.

The formulation here is done by combining the results in previous sections to write the following optimization problem,

$$\left.\begin{aligned} &\min_{\{\mathbf{A}_d\}_{d=1}^D} \quad \sum_{d=1}^{D}\sum_{i=1}^{N} \left(\mathbf{a}_{i,d}^T \mathbf{Q}_d \mathbf{a}_{i,d} - 2\mathbf{a}_{i,d}^T \mathbf{r}_d\right), \\ &\text{subject to} \\ &\mathbf{A}_d \mathbf{X}_d = \mathbf{Y}_d, \text{ for } d = 1, \cdots, D; \\ &\begin{bmatrix} \mathbf{I}_N & \left(\mathbf{I} + \sum_d \frac{\Delta t}{(\Delta x)^d}\beta_d \mathbf{A}_d^{\mathbf{\Phi}}\right)^T \\ \left(\mathbf{I} + \sum_d \frac{\Delta t}{(\Delta x)^d}\beta_d \mathbf{A}_d^{\mathbf{\Phi}}\right) & \mathbf{I}_N \end{bmatrix} \geq 0. \end{aligned}\right\} \tag{8.50}$$

Equation Eq. (8.50) is a convex optimization problem [149] in variables $\{\mathbf{A}_d\}_{d=1}^D$, with $\mathbf{A}_d \in \mathbb{R}^{N\times S}$, and can be efficiently solved numerically using `cvx` [148]. The time step Δt can be maximized by iteratively solving Eq. (8.50) with increasing Δt until the problem becomes unfeasible. Since the spectra of a matrix is continuous in terms of the elements, the maximum Δt can be determined using a bisection algorithm [21].

In Fig. 8.9 we show the result of such a computation again for an advection-diffusion equation discretized with second-order approximations. In particular, we show contours of the sum of the spectral error $\|e(\eta)\|_{\mathcal{L}_2}^2$ in the first and second derivatives for different values of M normalized by the value for $M = 1$. Colored areas represent regions where schemes are stable. For $M = 1$, the stable region is well known as discussed above and presents values of 1.0 due to the normalization chosen. Interestingly, as M increases the stability area increases instead of decreasing as in Fig. 8.8. This illustrates an interesting aspect of the formulation. By providing additional degrees of freedom (bigger stencil size) but fixing the formal order of accuracy, Eq. (8.50) minimizes spectral error with the condition that the scheme be stable when some fixed Δt is used. As can be seen, for $M = 4$, schemes can remain stable for very large CFL numbers (an order of magnitude larger

than for standard schemes) though with larger spectral error. In other words, the framework allows one to trade off error with stability. If one is interested in stationary states, for example, one can use the optimal scheme with $M = 4$ with very large time steps in a fully explicit arrangement to solve transients. After the desired steady state is attained one can reduce Δt with corresponding optimized schemes to reduce errors. In fact, by comparing part (a) and (d) in the figure, we can see that for regions where $M = 1$ is stable, $M = 4$ provides errors which could be two orders of magnitude smaller. Conversely, for similar accuracy, schemes with $M = 4$ can be used with much larger Δt than a scheme with $M = 1$. Note that this approach could prove beneficial at very large levels of parallelism where implicit schemes in time, while providing good stability characteristics for large time steps, become challenging due to the necessity to invert large matrices.

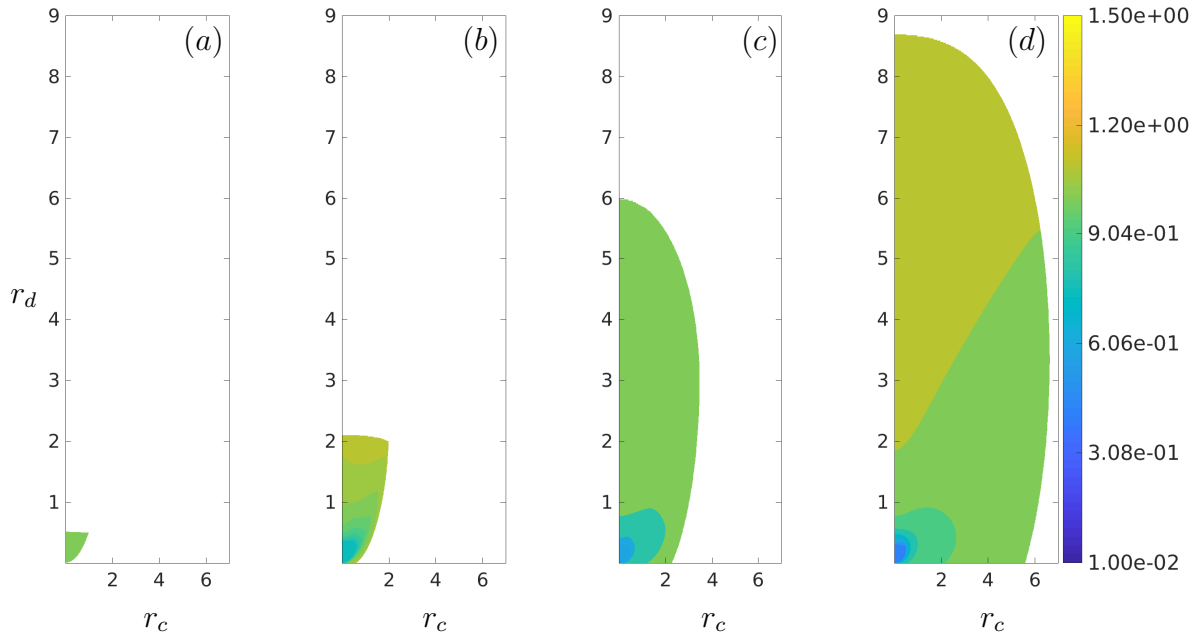

Figure 8.9: Tradeoff between stability and spectral accuracy with respect to (r_d, r_c), for $\beta_1 = 10$ and $\beta_2 = 1$. Contours represent sum of spectral errors $\|e(\eta)\|^2_{\mathcal{L}_2}$ for first and second derivatives using optimal coefficients obtained as a solution of Eq. (8.50) with fixed Δt for the advection-diffusion equation. Errors are normalized by the spectral error obtained for $M = 1$. Different stencil sizes shown in (a) $M = 1$, (b) $M = 2$, (c) $M = 3$, and (d) $M = 4$.

It is interesting to observe how optimal coefficients change to maintain stability for very large time steps. Since biased (upwind) schemes tend to be more dissipative and stable, one would expect that gains in stability are mediated by losses in accuracy due to biasing. This is, in fact, what is observed. In order to quantify deviations from symmetry ($a_i = a_{-i}$, where a_is are the coefficients in Eq. (8.2) that are obtained from $\mathbf{a}_d$) for even derivatives or anti-symmetry ($a_i = -a_{-i}$) for odd derivatives we define the metric:

$$A = \sum_{i=1}^{M} \frac{|a_{-i} + (-1)^{d+1} a_i|}{|a_{-i}| + |a_i|}, \tag{8.51}$$

for the d-th derivative. Clearly $A = 0$ when the coefficients are symmetric (for even derivatives) or anti-symmetric (for odd derivatives). Larger values of A are associated with increasingly biased approximations. The denominator in Eq. (8.51) is included such that the contribution from all coefficients are of the same order. This is needed because coefficients tend to decrease in magnitude with distance from the point where derivatives are computed.

In Fig. 8.10(a) we plot A for the first (a) and second (b) derivatives, respectively for $M = 2$. As before, colored areas correspond to stable conditions. The black line in Fig. 8.10(a) corresponds to the stability region given in Fig. 8.8 for $M = 2$. When we are within this boundary, the value of A for the first derivative is very small which suggests that all extra degrees of freedom are used to minimize the spectral error for which symmetric stencils are generally better. However, as we move outside of this area, these extra degrees of freedom are needed to satisfy the stability constraint. By losing symmetry or anti-symmetry error increases (as seen in Fig. 8.9) but stability improves. In fact, asymmetry for the first derivative implies that the error has both real and imaginary parts as seen in Fig. 8.11. The biased nature of coefficients leads not only to dispersion errors (as assured by lemma 4 with symmetric coefficients) but also to dissipation errors as $\Re[e(\eta)] \neq 0$. Both dissiption and dispersion error increase as we increase the value of r_c and r_d.

A different behavior is observed for the second derivative whose contours of A are shown in Fig. 8.10(b). Here we see that A is very small for any r_c and r_d within the limits of stability.

The implication of this is that stability is essentially governed by the first derivative, which may be intuitive. It is indeed common to stabilize fluid flow simulation codes by special treatment of the convective terms while leaving diffusive terms approximated by standard central differences. What our results show, beyond heuristic stabilization considerations is that, indeed, biasing convective terms and keeping central schemes for diffusion is the spectrally optimal way of achieving stability. They also show that, unlike [145], so-called anti-diffusion does not lead to instability due to the compensating effect of the other term in the equation. For completion, in Fig. 8.10(c) we show the change in optimal coefficients for the first derivative for the conditions represented by different symbols in part (a). We clearly see how the scheme changes from a completely anti-symmetric configuration (red circles) to a more biased set of coefficients. We emphasize this stable configurations are optimal in spectral space.

We close this section with two remarks about the stability limits computed here. First, in terms of computational cost we note that as a consequence of Lemma 5 for periodic domains, the optimization problem in Eq. (8.50) has a total of $n = Sd$ degrees of freedom, which is independent of the total number of grid points N. Since the computational cost in terms of time and memory for state-of-the-art optimization algorithms is $\mathcal{O}(n)$ [150], obtaining the numerical scheme represents a small fraction of the total computational cost. Furthermore, since the time step size is typically controlled either by r_c or r_d, the computational cost of solving the equation scales as N^2 or N^3, respectively, making the optimization cost increasingly smaller as the problem size increases. Second, we note that the stability limits shown in Fig. 8.9 were obtained numerically with machine precision. Thus, caution needs to be exercised when selecting conditions in the r_c-r_d plane very close to the stability boundaries.

8.2.3.5 Higher-order temporal schemes

In the previous section we illustrated how stability can be incorporated into the framework using a first-order forward temporal discretization. However, in practice higher order schemes are typically used. Thus, we generalize Eq. (8.40) to include higher order multi-step temporal schemes, such as Adams-Bashforth. For this we define a vector $\underline{\mathbf{F}}$ as a stack of the solution at L

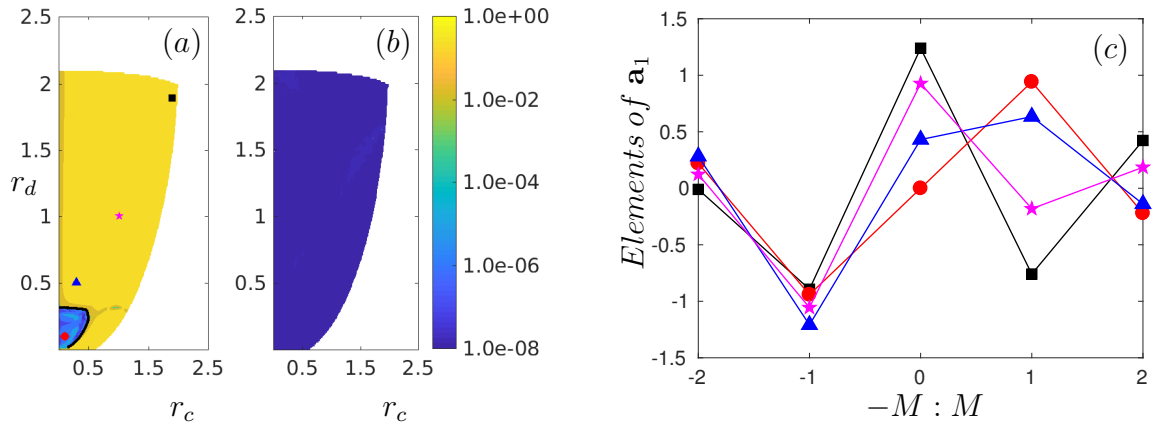

Figure 8.10: Contours of measure A for $M = 2$ for (a) first derivative (b) second derivative. (c) Optimal coefficients for different values of r_c and r_d as marked, with the same symbols, in (a).

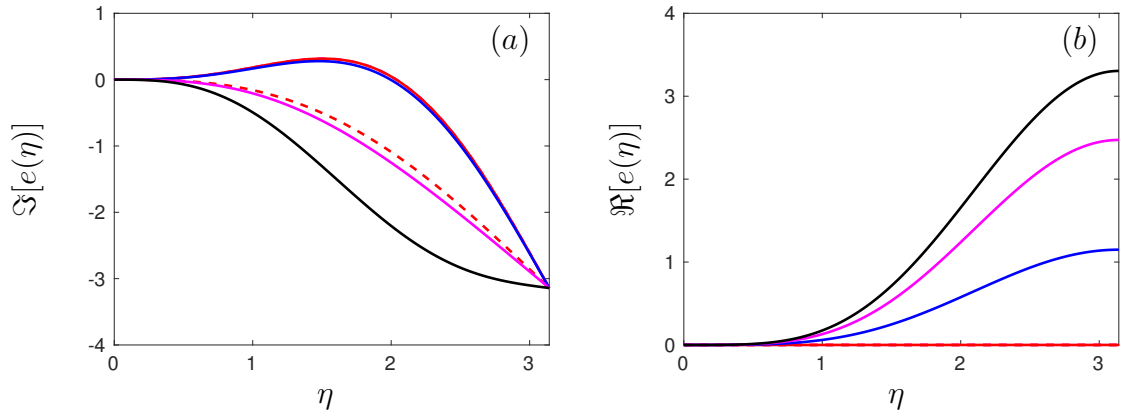

Figure 8.11: Dispersion error (a) and dissipation error (b) for the first derivative with $M = 2$ at different values of r_c and r_d as marked with the same color in Fig. 8.10(a). The dashed red line corresponds to standard second order scheme with $M = 1$ and the dashed magenta line (hidden) corresponds to optimized scheme obtained for $M = 2$ without stability constraint.

consecutive time levels,

$$\underline{\mathbf{F}}^{k+1} = \begin{bmatrix} \mathbf{F}^{k+1} & \mathbf{F}^{k} & \dots & \mathbf{F}^{k-L+1} \end{bmatrix}^T. \tag{8.52}$$

The evolution equation can then be written as

$$\underline{\mathbf{F}}^{k+1} = \left(\mathbf{A}_t + \sum_d \frac{\Delta t}{(\Delta x)^d} \beta_d \mathbf{B}_t \right) \underline{\mathbf{F}}^{k}, \tag{8.53}$$

where

$$\mathbf{A}_t := \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix} \quad \text{and} \quad \mathbf{B}_t := \begin{bmatrix} \mathbf{b}^T \otimes \mathbf{A}_d^{\mathbf{\Phi}} \\ \mathbf{0}_{NL\times N(L+1)} \end{bmatrix}, \tag{8.54}$$

$$\mathbf{A}_1 := \mathbf{e}_{1\times(L+1)} \otimes \mathbf{I}_{N\times N} \quad \text{and} \quad \mathbf{A}_2 := \begin{bmatrix} \mathbf{I}_{NL\times NL} & \mathbf{0}_{NL\times N} \end{bmatrix}, \tag{8.55}$$

with $\mathbf{e} = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix}$. The vector $\mathbf{b}^T$ is the vector of the coefficients of the temporal scheme that uses $L+1$ time levels. By examining Eq. (8.53) it is clear that multistep methods have the same structure as the example in section 8.2.3.4 where, for a given Δt, the unknowns are contained in $\mathbf{B}_t$. Thus, the stability condition can then be written as

$$\lambda_{max}\left(\mathbf{A}_t + \sum_d \frac{\Delta t}{(\Delta x)^d}\beta_d \mathbf{B}_t\right) \leq 1 \tag{8.56}$$

Following the same procedure as before, this condition, and thus stability, is guaranteed by bounding the 2-norm, i.e.

$$\left\|\mathbf{A}_t + \sum_d \frac{\Delta t}{(\Delta x)^d}\beta_d \mathbf{B}_t\right\|_2 \leq 1. \tag{8.57}$$

which is the generalization for the inequality constraint Eq. (8.42).

Thus, the unified generalized formulation for arbitrary multi-step temporal discretizations consists of an optimization problem that minimizes the spectral error given by Eq. (8.29) subjected to the order of accuracy (equality) constraint Eq. (8.28), and the stability (inequality) constraint Eq. (8.57).

The two approaches discussed in sections 8.2.3.3 and 8.2.3.4, can then be applied to the generalized multi-step schemes as well. That is, we can compute maximum Δt for given $\mathbf{A}_d^{\mathbf{\Phi}}$ or we can compute $\mathbf{A}_d^{\mathbf{\Phi}}$ by fixing Δt. We do note, however, that bounding the spectral radius by the $2\text{-}norm$ can be unnecessarily restrictive for some multistep schemes especially for long temporal stencils which leads to long vectors resulting from the stacking of increasingly large number of time levels.

8.3 Numerical results

In order to test the theoretical results from previous sections we conducted several test on model PDEs of increasing complexity. The focus would be in a comparison between the schemes developed here and standard schemes. In particular, we will compare our optimized schemes against

1. standard scheme of the same order

2. standard scheme with the same stencil size

While (a) allows us to assess how the additional degrees of freedom are used to increase spectral accuracy and/or maintain stability, (b) provides comparison between two schemes with the same computational cost in computing spatial derivatives since both schemes use the same stencil size. As pointed out later on, however, optimal schemes of lower order may indeed provide a computational advantage when they are coupled with a temporal scheme of matching order to solve a PDE.

We begin our analysis with the diffusion equation and the linear advection-diffusion equation for which exact analytical solutions are known and the error in numerical solutions can be evaluated accurately. We then turn to the non-linear advection-diffusion (Burgers) equation which is a widely used proxy to study important features of fluid flow motion governed by the Navier-Stokes equations. We will conclude the numerical section by a brief analysis of the wave equation as well as a discussion of the effect on dispersion relations.

For short, we will refer to the standard and optimized finite difference schemes as SFD and OFD, respectively in what follows.

8.3.1 Diffusion equation

Consider the equation:

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}, \tag{8.58}$$

where α is the diffusivity. Since this equation is linear, different Fourier modes do not interact and wavenumbers present in the solution are only due to them being present in the initial conditions. The dissipative action of the second derivative causes the decay of amplitude of all modes with time. This decay becomes more prominent as the wavenumber increases. Equation Eq. (8.58) is solved in a periodic domain of length $L = 2\pi$. The initial condition is a superimposition of sinusoidal waves,

$$u(x,0) = \sum_k A(k)\sin(kx + \phi_k), \tag{8.59}$$

where k denotes the wavenumber, ϕ_k is a random phase angle corresponding to each wavenumber, and $A(k)$ is the amplitude of each mode taken here to be represented as a power law of the form $A(k) = A(1)k^\sigma$. The value of the exponent was chosen to be $\sigma = -1/6$ which, by being small, corresponds to a shallow spectrum representative of a solution with a wide range of energetic modes. The reason for this is to critically assess the ability of schemes to represent accurately a wide range of scales.

The analytical solution of Eq. (8.58) is known:

$$u_a(x,t) = \sum e^{-\alpha k^2 t} A(k)\sin(kx + \phi_k) \tag{8.60}$$

For the semi-discrete analysis, we discretize Eq. (8.58) using an optimized second order scheme (OFD2) with $M = 4$ in space whose coefficients can be found in Tables 8.1 and 8.2. The numerical results so obtained are compared with standard second order (SFD2) and standard eighth order (SFD8) in space. For the fully discrete system, we match the order of accuracy of time and space discretization. Time and space step sizes (Δt, Δx) are related through a diffusive CFL condition ($r_d = \alpha\Delta t/\Delta x^2$). Thus, we use a forward first order discretization in time for OFD2 and SFD2, and a fourth-order five-stage Runge-Kutta scheme for SFD8.

To assess the error across scales, we compute the relative difference between the energy at individual Fourier modes (at a given wavenumber) of the numerical and analytical solutions, that is $|\hat{u}(\eta)^2 - \hat{u}_a(\eta)^2|/|\hat{u}_a(\eta)|^2$ where as before $\eta = k\Delta x$. In Fig. 8.12 we show this error for the

semi-discrete system (solid lines) at a normalized time of $t\alpha k_0^2 \approx 0.002$ where $k_0 = 1$ is the lowest wavenumber in the simulation. This small normalized time was selected to ensure that the solution has evolved enough to present measurable errors while, at the same time, energy in high wavenumbers is not completely dissipated. In terms of the highest wavenumber in the simulation he normalized time is $t\alpha k_{max}^2 \approx 20$. The error for SFD2 is larger than OFD2 and grows drastically with increasing wavenumber, illustrating the inability of low-order standard schemes to capture rapid spatial fluctuations. However, this is clearly not the case of optimized second-order schemes. For SFD8, the error is very small at low wavenumbers but also increases significantly at high wavenumbers. In fact, we see that the error spans more than fifteen orders of magnitude showing a dramatic disparity in resolution capabilities for multiscale problems.

The optimized scheme OFD2, on the other hand, shows a more uniform error distribution across the entire wavenumber space shown. In fact, OFD2 presents much smaller errors than even the eighth order SFD8 at high wavenumbers. By fixing the formal order the scheme to two, the additional degrees of freedom provided are used to increase the resolution capabilities of increasingly large regions of wavenumber space. The fact that OFD2 is more *spectrally flat* than the other schemes stems for the choice we have made for the function $\gamma(\eta)$ in Eq. (8.19) as unity in the region of interest $\eta \in [0, 2.5]$ and zero otherwise. This assigns equal weights to all wavenumbers in that interval and, thus, leads to spectrally flat schemes. Such performance is highly desired when one wants to study multiscale physical processes like turbulence. As we have shown in section 8.2.2.3, however, the formulation allows for non-uniform weights which can be used to obtain better resolution in one or more arbitrary regions of the wavenumber space.

In the figure we also include the fully-discrete system integrated with a very small time step ($r_d = 0.0005$) as a dashed line. This is seen to be very similar to the semi-dicrete case. However, if we increase the time-step, that is, increase r_d, then the error increases and the time discretization errors may dominate the solution. Note that this is true both for optimized and standard schemes.

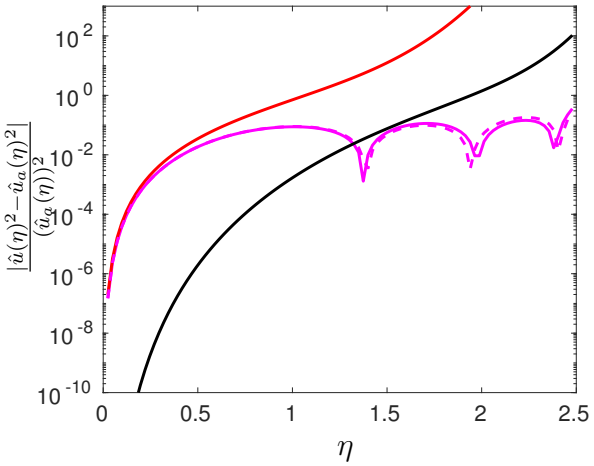

Figure 8.12: Normalized error in spectral energy for diffusion equation. Solid lines correspond to semi-discrete integration with different numerical schemes: SFD2 (red), SFD8 (black), OFD2 (solid magenta). Dashed magenta line is OFD2 for the fully-discrete system using forward first order in time with $r_d = 0.0005$.

8.3.2 Linear advection-diffusion equation

We now consider, next in complexity, the linear advection-diffusion equation which has both the first and second derivatives in space which, as shown in previous sections, typically present different types of errors:

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = \alpha\frac{\partial^2 u}{\partial x^2}, \tag{8.61}$$

In this case, Fourier modes are convected at the velocity c and dissipated at a rate determined by the diffusivity α. Here we use the same initial condition Eq. (8.59) as in the previous section. The analytical solution for Eq. (8.61) is given by

$$u_a(x,t) = \sum e^{-\alpha k^2 t} A(k) \sin(k(x-ct) + \phi_k) \tag{8.62}$$

Because of the presence of both first and second derivatives we expect the numerical solution to be affected by both dispersion and dissipation errors. Dispersion error typically due to the first derivative, distorts phase relations between different waves and tend to create distorted shapes. Dissipation errors, as explained before, affect the amplitude of different waves. In Fig. 8.13 we show

again the relative error of spectral energy which represents dissipation error across the wavenumber space. The trend is similar to what we observed for the diffusion equation.

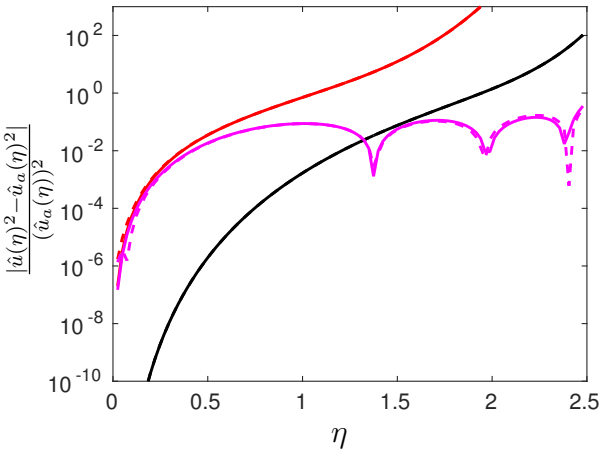

Figure 8.13: Normalized error in spectral energy for advection-diffusion equation. Different lines correspond to different numerical schemes: standard second order in space (red), standard eighth order in space (black), optimized (M-4) second order in space (solid magenta) and optimized (M-4) second order in space and forward first order in time (dashed magenta)

It is also possible to quantify dispersion errors by computing the effective propagation speed c^* of Fourier modes at wavenumber k which can readily shown to be given by $c^*(k) = \arg[\hat{u}_t(k)/\hat{u}_0(k)]/kt$, where $\hat{u}_0(k)$ and $\hat{u}_t(k)$ are the Fourier coefficients at the beginning of the simulations and at a time t, respectively. The ratio of the numerical speed c^* to the actual propagation speed c is a measure of phase (dispersive) errors. This ratio is unity if there is no phase error which implies that the numerical solution travels with the same speed as the actual solution. In fig. 8.14(a) we show results for a normalized time of $ck_0t = 0.1$, or in terms of the highest wavnumber in the simulation $ck_{max}t \approx 10$. We see that, for SDF2, the numerical wave speed becomes much smaller compared to the actual speed c as the wavenumber increases, resulting in large phase errors. For OFD2 the numerical speed remains much closer to the actual speed, though with some oscillations. OFD2 is also seen to be better than even the eighth order scheme SFD8 for which, as the wavenumber increases, the numerical speed decreases monotonically leading to very large phase error. The

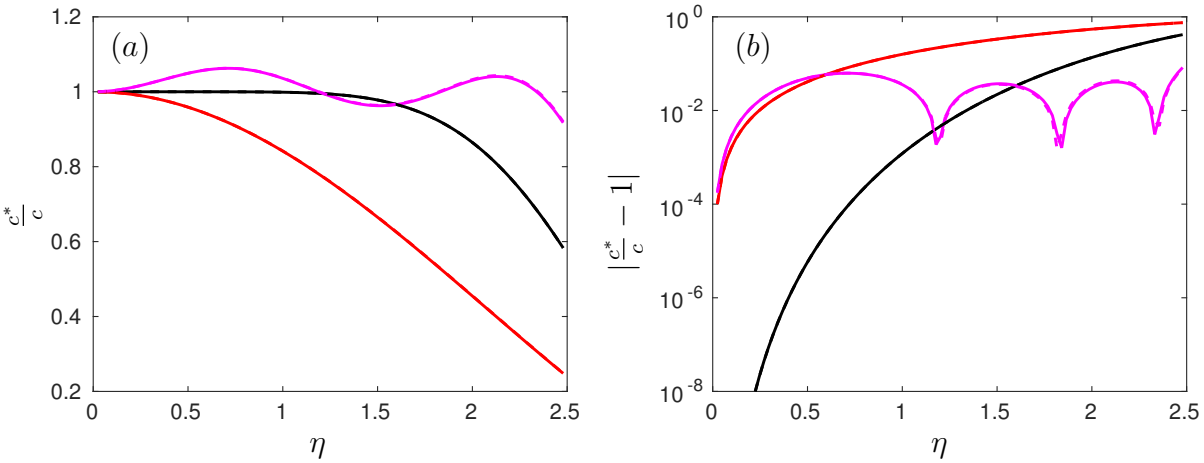

Figure 8.14: (a) Ratio of numerical to actual speed. (b) Normalized phase error for advection-diffusion equation. Different lines correspond to different numerical schemes: SFD2 (red), SFD8 (black), semi-discrete with OFD2 in space (solid magenta) and OFD2 in space and forward first order in time (dashed magenta).

normalization of fig. 8.14(b) highlights this trend in the phase error. In particular, it shows that, because of the particular choice of $\gamma(\eta)$, the numerical scheme presents a more spectrally flat response.

A complementary assessment of the numerical performance of these schemes can be obtained by comparing the time evolution of different Fourier modes. In the fully discrete system, order of accuracy in time is chosen to be consistent with the spatial discretization. With a fixed diffusive CFL, this implies that the second and eight order schemes in space need a first and fourth order temporal discretization, respectively. In the examples below we use, thus, forward Euler (first order) and RK4 (fourth order) for time integration.

For illustration purposes we consider waves at the extreme ends ($k = 10, 100$) of the wavenumber interval over which the schemes have been optimized, and an intermediate wave-number ($k = 50$). These correspond to $\eta = \{0.245, 1.227, 2.454\}$. In fig. 8.15 we show the evolution of energy for these three wavenumbers. The decay in energy occurs at a faster rate as the wavenumber increases. At lower η, the energy decay is well captured by all three schemes. As we increase η, SFD2 over-predicts the energy content and fails to capture the actual dissipation. The disparity

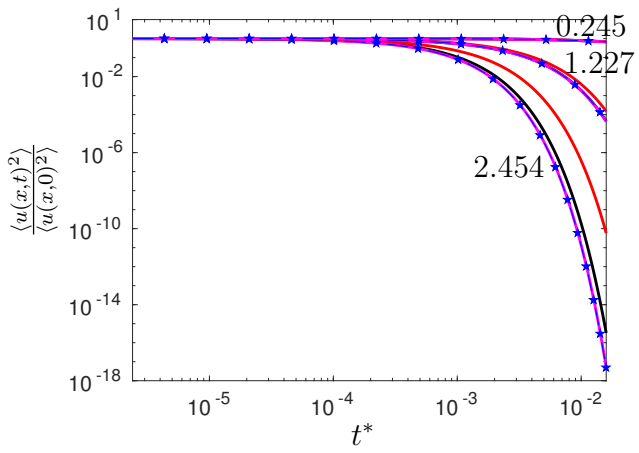

Figure 8.15: Evolution of space-averaged energy normalized by the initial space-averaged energy with normalized time $t^* = tc/L$ for three different wave-numbers ($\eta = 0.245, 1.227, 2.454$). Different line styles correspond to different numerical schemes: SFD8 with RK4 in time (black), OFD2 with forward Euler in time (solid magenta), SFD2 with forward Euler in time (red). The blue dashed-star line is the analytical solution.

between SFD2 and OFD2 increases with increasing wavenumber with the latter remaining close to the analytical solution. For very high wavenumbers $\eta = 2.454$, the second-order optimized scheme is in fact visibly closer to the analytical solution than a standard eighth-order scheme. Clearly by using the additional information provided by neighboring points to increase spectral accuracy instead of formal order of accuracy leads to better resolved physics, especially at high wavenumbers where dissipation is strongest.

We have also compared schemes derived with our framework with those in [137, 132]. For example, we have computed fourth-order optimized schemes for $M = 4$ with the same conditions as those in those references and found essentially the same coefficients. Thus, our general framework can reproduce other particular results in the literature.

8.3.3 Non-linear advection-diffusion equation

Of fundamental and practical interest is the non-linear advection-diffusion equation as it resembles the one-dimensional version of the Navier-Stokes equation that governs the motion of fluid flows,

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \alpha\frac{\partial^2 u}{\partial x^2}. \tag{8.63}$$

Here $u(x,t)$ is the velocity and α is viscosity. The non-linear term causes interaction between Fourier modes which redistributes energy among the different scales (wavenumbers) in the solution and produces new scales of motion. Because of the absence of a constant input of energy, the amplitude of different modes decay with time due to the dissipative action which becomes more effective at smaller scales.

Eq. (8.63) is solved in a periodic domain of length $L = 2\pi$, subject to initial conditions given by Eq. (8.59). The value of the exponent in Eq. (8.59) was chosen to be $\sigma = -5/6$. This exponent, which corresponds to an energy spectrum decaying as $k^{-5/3}$ consistent with fully developed turbulence, ensures that the spectrum is shallow enough to ensure high energy content at high wavenumbers while remaining stable.

Although Eq. (8.63) is non-linear, we can apply the Cole-Hopf transformation and find an analytical solution to the problem [151, 152]. Define a transformation variable ϕ, such that

$$u = -2\alpha \frac{1}{\phi} \frac{\partial \phi}{\partial x}, \tag{8.64}$$

Then Eq. (8.63) reduces to a simple diffusion equation in ϕ, which can be readily solved analytically. The result in terms of the primitive variable $u(x,t)$ is

$$u(x,t) = \frac{\int_{-\infty}^{\infty} \frac{(x-y)\mathcal{I}(y,x)}{t} dy}{\int_{-\infty}^{\infty} \mathcal{I}(y,x) dy}. \tag{8.65}$$

where $\mathcal{I}(y,x) = e^{-\frac{(x-y)^2}{4\alpha t}} \phi(y,0)$. and the initial condition for ϕ, is computed from the initial condition for u as

$$\phi(x,0) = e^{\left(-\int_0^x \frac{u(y,0)}{2\alpha} dy\right)}. \tag{8.66}$$

While Eq. (8.65) is the exact solution, the integrals involved are computed numerically with standard integral techniques which were tested for grid convergence. This is compared with the numerical solution of Eq. (8.63) using, as before, SFD2, SFD8, and OFD2. The fully discrete system is formed with compatible temporal scheme as described in previous section. We performed grid

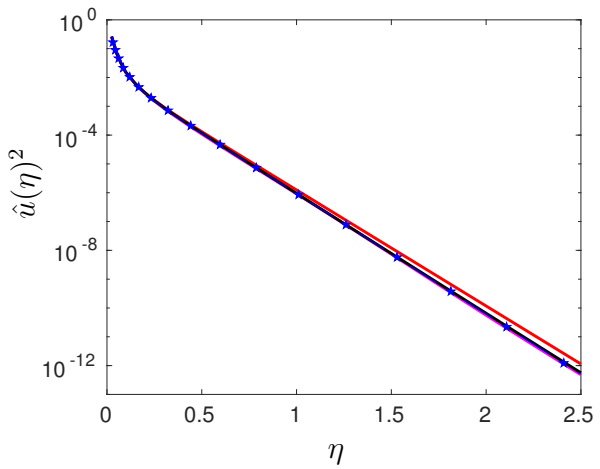

Figure 8.16: Energy spectrum for the non-linear advection-diffusion equation at $t^* = t/t_0 \approx 0.464$. Different lines correspond to different numerical schemes: SFD8 with RK4 in time (black), OFD2 with forward first-order in time (magenta), and SFD2 with forward first-order in time (red). The blue dashed-star line is the analytical solution.

convergence studies and found that e.g. the space-averaged kinetic energy evolution in time becomes independent of resolution at $N = 256$. This is the resolution used for the comparisons that follow.

In Fig. 8.16 we show the energy spectrum obtained for these three schemes along with the analytical solution Eq. (8.65), at $t/t_0 \approx 0.464$, where $t_0 = K_0/\epsilon_0$ is a characteristic time scale defined by the initial energy ($K_0 \equiv \langle u_0^2 \rangle/2$, where angular brackets denote space averages and a subscript 0 denotes initial conditions) and the energy dissipation rate ($\epsilon_0 \equiv \alpha \langle (\partial u_0/\partial x)^2 \rangle$). We observe that OFD2 and SFD8 agree closely with each other and with the analytical solution throughout the range of η. Results for SFD2, however, exhibit clear departures especially at high wavenumbers. The ability of OFD2 to capture high wavenumbers accurately is expected since, as shown above, this scheme presents a more spectrally flat response. In fact, this optimized scheme presents better resolution than SFD8 at very high waveumbers though the converse is true at low waveumbers.

Because of the better small-scale resolution of OFD2, one would expect it to capture quantities that depend sensitively on the small scales more accurately than SFD8 as well. An ex-

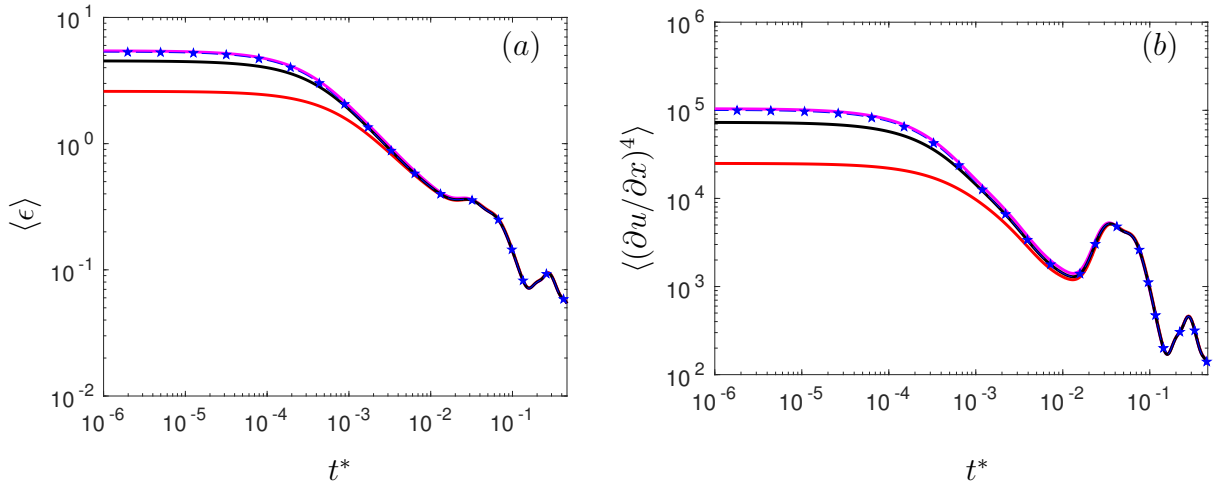

Figure 8.17: (a) Evolution of space-averaged dissipation (b) Evolution of space-averaged fourth order moment of velocity gradient with normalized time ($t^* = t/t_0$) . Different lines correspond to different numerical schemes: SFD8 with RK4 in time (black), OFD2 with forward first-order in time (magenta), and SFD2 with forward first-order in time (red). The blue dashed-star line is the analytical solution.

ample of such a quantity is the dissipation rate introduced above which is proportional to the second order moment of the velocity gradient and thus is dominated by high wavenumber activity. This is indeed observed in Fig. 8.17(a) where we show the evolution of $\langle\epsilon\rangle$ as the flow decays along with the analytical value (dashed-star line) which can be computed easily by taking derivative of Eq. (8.65), $\langle\epsilon\rangle = \alpha\langle(du/dx)^2\rangle = \alpha\langle\big(\int_{-\infty}^{\infty}\mathcal{I}(y,x)dy\int_{-\infty}^{\infty}\frac{\big(2\alpha t-(x-y)^2\big)\mathcal{I}(y,x)}{2\alpha t^2}dy - \int_{-\infty}^{\infty}\frac{(x-y)\mathcal{I}(y,x)}{t}dy\int_{-\infty}^{\infty}\frac{(y-x)\mathcal{I}(y,x)}{2\alpha t}dy\big)\left(\int_{-\infty}^{\infty}\mathcal{I}(y,x)dy\right)^{-2}\rangle$.We can clearly see that initially OFD2 is very close to the analytical value followed by SFD8 and SFD2. As time evolves, diffusive effects damp high wavenumbers faster then small scales and the main contribution to dissipation moves to lower wavenumbers where the three schemes present similar resolution capabilities. The same conclusion holds for the more challenging higher order moments. This is seen in Fig. 8.17(b). where we show the fourth-order moment of velocity gradients. Again OFD2 is more accurate than second and eight order standard schemes.

8.3.4 Wave Equation

Another equation that has been used to assess the performance of numerical schemes is the second order linear PDE for the discription of waves [144], commonly used in acoustics and given by,

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2} \tag{8.67}$$

where c is the propagation speed. The general solution to this equation is a standing wave formed by the superposition of two travelling waves moving with velocity c to the right and -c to the left respectively. Following [144], this equation can be expressed as the following system of equations,

$$\begin{pmatrix} \partial u/\partial t \\ \partial v/\partial t \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ c^2 \partial^2/\partial x^2 & 0 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix}. \tag{8.68}$$

We solved this system for a Gaussian initial condition, $u_0(x) = 0.2e^{-64x^2}$, that has a narrow width in the physical domain and therefore the spectrum spans a wide range of wavenumbers in Fourier space. We use the optimized second-order OFD2 for the second derivative in space computed with a weight function, $\gamma(w) = e^{-w^2/256}$ for $w = [0, 2]$ and $\gamma(w) = 0$ elsewhere in Eq. (8.21). This was done to emphasize wavenumbers relevant to the problem. Because of the use of a CFL condition given by $r_c = c\Delta t/\Delta x = 0.1$, we also employ second-order temporal discretization (RK2). The solution was advanced until a physical time of $t = 2$ for periodic boundary conditions and the results have been plotted in Fig. 8.18. We can see that the SFD2 scheme has already developed significant oscillations that trigger instabilities, whereas the OFD2 is comparable to both SFD8 and the exact solution. The space averaged 2-norm of the error for OFD2 is an order of magnitude smaller than the error for SFD2. The 2-norm of the error for OFD2 scheme is comparable to the error obtained for SFD8. Besides this, the OFD2 scheme is three times computationally less expensive as compared to the SFD8.

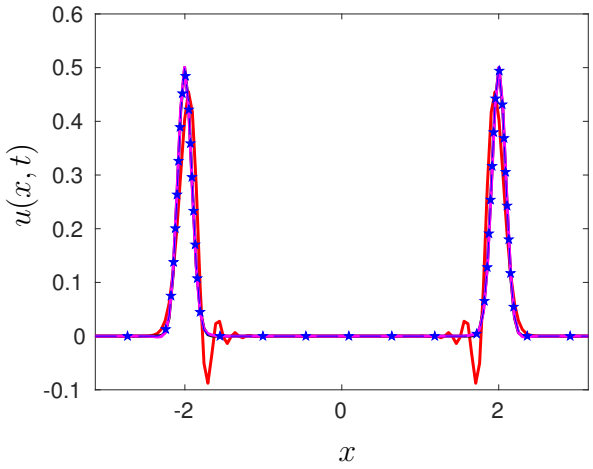

Figure 8.18: Solution of the wave-equation for $t = 2$ with a Gaussian initial condition. Different lines correspond to different numerical schemes: SFD8 with RK4 in time (black), OFD2 with RK2 in time (magenta), and SFD2 with RK2 in time (red). The blue dashed-star line is the analytical solution.

8.3.5 Space-time errors

In addition to spectral accuracy, other performance metrics have been utilized in the literature to assess the appropriateness of a numerical scheme to reproduce physics of interest. For strongly convective problems, such as in acoustics, the dispersion relation provides important information about propagation speeds and characteristics. Thus, the interest in so-called dispersion-relation preserving schemes [132].

Consider again the convection-diffusion equation:

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = \alpha\frac{\partial^2 u}{\partial x^2}, \quad c, \alpha > 0\,, \tag{8.69}$$

This linear equation propagates the initial condition to the right at the speed c and because of the diffusive term the amplitude decreases with time at a rate determined by the diffusivity coefficient α. Using $u = \hat{u}e^{-j\omega t}e^{jkx}$ it is easy to obtain the exact dispersion relation for this equation, $\omega = ck - j\alpha k^2$, where k is the wavenumber. Two physically meaningful quantities related to the dispersion relation are the phase and group velocities (c_p and c_g respectively) which are given by the real

part of $c_p = \omega/k$ and $c_g = d\omega/dk$. Note that both are equal to c for the analytical solution. The equivalent numerical dispersion relation is, on the other hand, given by $\omega^* = c^*k - j\alpha^*k$, where c^* and α^* are the numerical velocity and diffusivity respectively. One is thus interested in how the phase and group velocities from the computed solution (c_p^* and c_g^*) compares to their exact values.

A convenient way to obtain dispersion relations and derived quantities is through the so-called amplification factor G^* defined as the ratio of Fourier modes of the numerical solution at two consecutive time steps. In general, one can write $G^* = |G^*|e^{-j\beta}$, where $|G^*|$ is the magnitude of the amplification factor and the phase β is related to the numerical phase speed (c_p^*). Note that the exact amplification factor is given by $G = e^{-i\omega\Delta t}$. It is then readily shown that the numerical group velocity can be written as [139],

$$\frac{c_g^*}{c_g} = \frac{1}{r_c\Delta x}\frac{d\beta}{dk}. \tag{8.70}$$

where, as noted above, $c_g = c$. In a dispersion-relation preserving scheme, this ratio as well as c_p^*/c_p should be close to unity.

As an illustration, consider Eq. (8.69) discretized with a forward difference in time and the spatial schemes in Table 8.1 and Table 8.2:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{-c}{\Delta x}\sum_{m=-M}^{M} a_{m,1}u_{i+m} + \frac{\alpha}{\Delta x^2}\sum_{m=-M}^{M} a_{m,2}u_{i+m}. \tag{8.71}$$

Since c_g^*/c_g and c_p^*/c_p depend on both $k\Delta x$ and $\omega\Delta t$, one can assess performance by measuring the area in the $k\Delta x - \omega\Delta t$ plane in which these ratios area within some percentage of unity, say 5% [145]. The larger this area, the larger the range of wavenumbers that preserve the dispersion relation. And since the phase errors are more prominent in convection dominated problems, we will consider relatively small values of r_d. From table 8.3, we can see that when the optimized schemes derived above are coupled with forward difference in time, the region in the $k\Delta x - \omega\Delta t$ for which c_g^*/c_g is close to unity, increases considerably for different values of r_d. For $M = 4$, this area is five times larger than that for the standard second order scheme. We can also see that the ratio c_p^*/c_p

is close to unity for a much larger area for optimized schemes. This is in fact not unexepected as the objective function in the minimization problem is designed to bring the modified wavenumber close to the actual wavenumber, resulting in more accurate derivatives leading, in general, to lower errors in phase and group velocities. Similar results were also observed for the diffusivity ratio α^*/α.

	$r_d = 0$		$r_d = 0.02$	
M	c_g^*/c_g	c_p^*/c_p	c_g^*/c_g	c_p^*/c_p
1(S)	0.58	1.73	0.62	1.84
2(O)	2.29	6.87	2.27	6.89
3(O)	1.82	5.13	1.87	5.34
4(O)	2.91	10.09	3.04	10.59

Table 8.3: Percentage area in the $k\Delta x - \omega\Delta t$ plane for which c_g^*/c_g and c_p^*/c_p is within 5% of unity for $r_d = 0$ and $r_d = 0.02$. In parenthesis, O stands for optimized scheme, and S for standard scheme. The schemes used here are those in Table 8.1 and Table 8.2.

We close this section by noting that although optimized schemes seem to naturally preserve dispersion relations better, constraints on these dispersion relations can be directly included in the general framework proposed here. For example, one can construct objective functions as in [132] where one can emphasize either propagation characteristics or damping characteristics. In the present framework, this can be done in such a way to ensure, at the same time, stability of the resulting scheme. Some of the effects of such an implementation were discussed in section 8.2.2.1.

8.4 Conclusions

Standard finite difference schemes are commonly derived to maximize its formal order of accuracy for a given stencil. The spectral accuracy and stability of the schemes so obtained are typically checked *a posteriori*. While efforts have been devoted to merge order-of-accuracy constraints and spectral accuracy, no general mathematical framework has been put forth which, perhaps more

importantly, introduced stability as an additional constraint. Here we develop such a framework to derive finite differences that accounts for order of accuracy, spectral resolution and stability. The most general formulation is given by Eq. (8.50).

While order of accuracy is defined as the power of the lowest order term in the truncation error, spectral accuracy is defined through an objective function that minimizes the spectral error in some specific way. This definition of this error is rather general and includes a weighting function $\gamma(\eta)$ which can be used to emphasize different scales relevant to the physical problem being solved. The fusion of order-of-accuracy constraints and spectral accuracy leads to a minimization problem which is convex and thus leads to a global minimum. The optimal coefficients, which are obtained analytically from the minimization problem, were shown to be symmetric for the even derivatives and anti-symmetric for the odd derivatives. This was shown to correspond to vanishing imaginary and real part of the error $e(\eta)$. This is the case, regardless of the functional form of the weighting function $\gamma(\eta)$. In other words, we have shown that the minimization of spectral errors leads to symmetric or anti-symmetric stencils for even and odd derivatives respectively, regardless of which range of wavenumber is optimized.

We have also incorporated stability into the unified framework. The semi-discrete system is shown to depend upon the sign of the parameters β_d for even d in the PDE. For the fully-discrete system, stability is assured by requiring the spectral radius of the evolution matrix to be smaller than unity which can in turn be bounded by its 2-norm. The latter can be written as a linear matrix inequality. This provides an additional constraint imposed on the minimization problem to ensure that the resulting scheme is stable. The final unified mathematical framework consists on the minimization of an objective function representative of spectral error constrained by given order of accuracy and stability guarantees. Due to the non-linearity inherent in this constraint, this problem is no longer convex and therefore cannot be solved directly using standard optimization tools. Two approaches were proposed to tackle this non-linear optimization.

In the first approach, given an optimal scheme (subjected to constraints in order and spectral accuracy) one finds the largest Δt for which the scheme remains stable. This approach is similar

to standard practices in which given a scheme, one aims at finding the largest time step that assures stability. Here, however, the largest time step results from a convex optimization which gives global extrema and can be solved efficiently. Through an example, we showed that the stability region decreased with increasing stencil size M consistent with a reduction of (artificial) dissipation at higher wavenumbers. This approach illustrates how accuracy and stability are separate requirements in the formulation: the additional degrees of freedom available for longer stencils are used to maximize spectral resolution regardless of stability. The result is a reduced stability region in the r_d-r_c space. In the second approach, the three elements are combined: optimal coefficients are obtained with both order of accuracy and stability constraints for a given time-step Δt. This is a common situation when the time step is set by physical considerations (e.g. shortest time scale in the problem). In this approach, the additional information provided by neighboring grid points is used to extend stability which make the use of much larger time-step feasible with spectral error comparable to the standard schemes. We showed that explicit schemes both in time and space, can remain stable for very large time steps. This can provide significant advantages for massively parallel simulations for which implicit schemes become increasingly challenging at large processor counts.

Several numerical results were presented to illustrate the numerical performance against standard finite differences of different orders. In particular, we compared optimized schemes against standard schemes of the same order and the same stencil size. While the latter presents the same computational cost in terms of spatial derivatives, the fully discrete system may be less expensive if the order of the temporal discretization is to be compatible with spatial order. Introducing the effect of temporal discretization in the unified framework presented here is part of our own ongoing research. Another application of the framework presented here is the inclusion of more general boundary conditions. Although we have limited our results to periodic domains, the framework can straightforwardly be extended to solve problems with non-periodic boundary conditions. The overall minimization problem and the constraints remain the same but the structure of some matrices has to be changed to restrict the stencil used close to boundaries. In particular, the first few and

last elements of $\mathbf{A}_d$ in Eq. (8.27) will be biased so that only grid points to the right and left of the boundary are used respectively [153]. The size of each stencil as well as the order or accuracy can be set individually through the corresponding entries in $\mathbf{X}_d$ in Eq. (8.28). As commonly done in simulations of complex flows, a progressive reduction of order of accuracy close to the boundary can thus be easily accomodated in this formulation. The stability constraint is identical to that presented for periodic domains. Thus, here too Eq. (8.50) will yield spectrally optimal schemes for a given Δt and Eq. (8.47) will yield the maximum Δt for which a spectrally optimal scheme remain stable.

In summary, we integrated order of accuracy, spectral resolution and stability in the derivation of finite differences in a unified framework. We have shown specific properties of the resulting schemes in terms of the kind of error expected. The coupling of these three critical elements in a unified formulation allows one to decouple requirements in terms of e.g. order of accuracy and spectral accuracy. This coupling also manifests itself in the trading of accuracy with stability. We showed, for example, how spectrally optimal finite differences bias odd order derivatives to maintain stability at the expense of accuracy. Other tradeoffs can be analyzed similarly within the framework presented here.

Appendix 8.A: Imaginary component of the optimal spectral error $e(\eta)$ is zero for even derivatives

Proof. The spectral error for even derivative is

$$e(\eta) = \left(\mathbf{C}^T(\eta)\mathbf{a}_d - (-1)^q\eta^d\right) + j\mathbf{S}(\eta)^T\mathbf{a}_d.$$

Therefore,

$$\|e(\eta)\|_{\mathcal{L}_2}^2 = \int_0^\pi \gamma(\eta)\left[\left(\mathbf{C}^T(\eta)\mathbf{a}_d - (-1)^q\eta^d\right)^2 + (\mathbf{S}(\eta)^T\mathbf{a}_d)^2\right]d\eta,$$

or

$$\min_{\mathbf{a}_d} \|e(\eta)\|^2_{\mathcal{L}_2} = \min_{\mathbf{a}_d} \int_0^\pi \gamma(\eta) \left[\left(\mathbf{C}^T(\eta)\mathbf{a}_d - (-1)^q \eta^d \right)^2 \right] d\eta + \min_{\mathbf{a}_d} \int_0^\pi \gamma(\eta) \left(\mathbf{S}(\eta)^T \mathbf{a}_d \right)^2 d\eta.$$

Since the cost function is sum of squares, it is minimized if and only if individual terms are minimized.

Without loss of generality, we can write $\mathbf{a}_d := \mathbf{a}_d^s + \mathbf{a}_d^{as}$, where $\mathbf{a}_d^s$ is symmetrical about the central element, and $\mathbf{a}_d^{as}$ is anti-symmetric about central element. Therefore,

$$\mathbf{S}^T(\eta)\mathbf{a}_d = \mathbf{S}^T(\eta)(\mathbf{a}_d^s + \mathbf{a}_d^{as}) = \mathbf{S}^T(\eta)\mathbf{a}_d^{as},$$
$$\mathbf{C}^T(\eta)\mathbf{a}_d = \mathbf{C}^T(\eta)(\mathbf{a}_d^s + \mathbf{a}_d^{as}) = \mathbf{C}^T(\eta)\mathbf{a}_d^s,$$

since $\mathbf{S}^T(\eta)\mathbf{a}_d^s = 0$ for symmetric coefficients and $\mathbf{C}^T(\eta)\mathbf{a}_d^{as} = 0$ for anti-symmetric coefficients.

Consequently,

$$\begin{aligned} &\min_{\mathbf{a}_d} \int_0^\pi \gamma(\eta) \left[\left(\mathbf{C}^T(\eta)\mathbf{a}_d - (-1)^q \eta^d \right)^2 \right] d\eta + \min_{\mathbf{a}_d} \int_0^\pi \gamma(\eta) \left(\mathbf{S}(\eta)^T \mathbf{a}_d \right)^2 d\eta, \\ =&\min_{\mathbf{a}_d^s} \int_0^\pi \gamma(\eta) \left[\left(\mathbf{C}^T(\eta)\mathbf{a}_d^s - (-1)^q \eta^d \right)^2 \right] d\eta + \min_{\mathbf{a}_d^{as}} \int_0^\pi \gamma(\eta) \left(\mathbf{S}(\eta)^T \mathbf{a}_d^{as} \right)^2 d\eta. \end{aligned}$$

Therefore, the two optimizations are independent of each other. For a positive real valued function $\gamma(\eta)$, the second term is zero if and only if $\mathbf{a}_d^{as} = 0$. Consequently, for $\mathcal{L}_2$ optimal spectral errors, the imaginary part of the spectral error is zero for even derivatives, and the optimal $\mathbf{a}_d$ is symmetrical about central element.

We next analyze the feasibility of the order constraint with $\mathbf{a}_d^{as} = 0$. Let $\mathbf{T}_s := (\mathbf{I} + \mathbf{J})/2$, and $\mathbf{T}_{as} := (\mathbf{I} - \mathbf{J})/2$, where $\mathbf{J}$ represents anti-diagonal matrix. Wth these transformation matrices, we can write $\mathbf{a}_d^s := \mathbf{T}_s\mathbf{a}_d$ and $\mathbf{a}_d^{as} := \mathbf{T}_{as}\mathbf{a}_d$. Therefore,

$$\mathbf{a}_d^T\mathbf{X}_d = (\mathbf{a}_d^s + \mathbf{a}_d^{as})^T\mathbf{X}_d = (\mathbf{a}_d^s)^T\mathbf{X}_d = \mathbf{a}_d^T\mathbf{T}_s^T\mathbf{X},$$

and the order accuracy constraint can be written as

$$\mathbf{a}_d^T \mathbf{T}_s^T \mathbf{X} = \mathbf{y_d}.$$

We observe that the structure of $\mathbf{X}_d$ is such that the odd columns are symmetric and the even columns are anti-symmetric, about the central element. Therefore, for the even columns, $\mathbf{T}_s^T(\mathbf{X}_d)_i = 0$. Noting that the even columns of $\mathbf{y_d}$ are zero, we can conclude that the constraints corresponding to the even columns are trivially satisfied for symmetric coefficients. For the odd columns of $\mathbf{X}_d$, we observe that $\mathbf{T}_s^T(\mathbf{X}_d)_i = (\mathbf{X}_d)_i$. That is, the constraints corresponding to the odd columns are unaffected. Consequently, if $\mathbf{a}_d^T \mathbf{X}_d = \mathbf{y}_d$ is feasible, then $\mathbf{a}_d^T \mathbf{T}_s^T \mathbf{X} = \mathbf{y_d}$ is also feasible. □

Appendix 8.B: Real component of the optimal spectral error $e(\eta)$ is zero for odd derivatives

Proof. The proof follows similarly to above. The spectral error in this case is

$$e(\eta) := \mathbf{C}^T(\eta)\mathbf{a}_d + j\left(\mathbf{S}^T(\eta)\mathbf{a}_d - (-1)^q \eta^d\right).$$

Using the same decomposition for $\mathbf{a}_d$ as above, we get

$$\min_{\mathbf{a}_d} \|e(\eta)\|_{\mathcal{L}_2}^2 = \min_{\mathbf{a}_d} \int_0^\pi \gamma(\eta)\left(\mathbf{C}^T(\eta)\mathbf{a}_d\right)^2 d\eta + \min_{\mathbf{a}_d} \int_0^\pi \gamma(\eta)\left[\mathbf{S}^T(\eta)\mathbf{a}_d - (-1)^q\eta^d\right]^2 d\eta,$$
$$\min_{\mathbf{a}_d^s} \int_0^\pi \gamma(\eta)\left(\mathbf{C}^T(\eta)\mathbf{a}_d^s\right)^2 d\eta + \min_{\mathbf{a}_d^{as}} \int_0^\pi \gamma(\eta)\left[\mathbf{S}^T(\eta)\mathbf{a}_d^{as} - (-1)^q\eta^d\right]^2 d\eta.$$

Using similar arguments as above, the optimal solution will guarantee $\mathbf{a}_d^s = 0$ and consequently, the real part of the spectral error is zero for odd derivatives. With $\mathbf{a}_d^s = 0$, the optimal $\mathbf{a}_d$ will be anti-symmetric about its central element.

The proof for feasibility of the accuracy order constraint, with $\mathbf{a}_d^s = 0$, is similar to the feasibility proof for the even derivative. □

Appendix 8.C: Construction of coefficient matrix $\mathbf{A}_d^{\mathbf{\Phi}}$

Proof. Define a shift operator $\mathbf{\Phi}_k$, which is an $N \times N$ matrix, with elements

$$\Phi_{k_{ij}} := \delta((i-j-k) \mod N), \tag{8.72}$$

where $\delta(\cdot)$ is the Kronecker delta function defined as

$$\delta(i) = \begin{cases} 0 \text{ if } i \neq 0, \\ 1 \text{ if } i = 0. \end{cases}$$

For a column vectors, the operator $\mathbf{\Phi}_k$ cyclically shifts the elements down, k times. For example, for

$$\mathbf{v} := \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix}, \mathbf{\Phi}_1 \mathbf{v} = \begin{pmatrix} 4 \\ 1 \\ 2 \\ 3 \end{pmatrix},$$

where

$$\mathbf{\Phi}_1 := \delta((i-j-1) \mod 4) = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

For a row vector, the operator cyclically shifts the elements left, k times. That is,

$$\mathbf{v}^T := \begin{pmatrix} 1 & 2 & 3 & 4 \end{pmatrix}, \mathbf{v}^T \mathbf{\Phi}_1 = \begin{pmatrix} 2 \\ 3 \\ 4 \\ 1 \end{pmatrix},$$

From the definition of vector $\mathbf{F}$ and $\mathbf{F}^{(d)}$

$$\mathbf{F} := \begin{pmatrix} f_1 \\ \vdots \\ f_N \end{pmatrix}, \text{ and } \mathbf{F}^{(d)} := \begin{pmatrix} f_1^{(d)} \\ \vdots \\ f_N^{(d)} \end{pmatrix}. \tag{8.73}$$

we can write the finite difference approximation at the i-th grid point as

$$f_i^{(d)} = \frac{1}{(\Delta x)^d} \mathbf{a}_{i,d}^T \mathbf{T} \mathbf{\Phi}_{M_{\max}+i-1} \mathbf{F},$$

for $i = \{1, \cdots, N\}$, and $\mathbf{T} \in \mathbb{R}^{S \times N}$ is a transformation matrix defined by

$$\mathbf{T} := \begin{bmatrix} \mathbf{0}_{S \times (M_{\max} - M)} & \mathbf{I}_S & \mathbf{0}_{S \times (M_{\max} - M)} \end{bmatrix}. \tag{8.74}$$

The matrix $\mathbf{T}\mathbf{\Phi}_{M_{\max}+i-1}$ is a linear operator, or simply a mask, that picks the correct elements from $\mathbf{F}$ in determining the derivative at the i^{th} location.

Now, let $\mathbf{A}_d^{\mathbf{\Phi}}$ be the vertical stacking of $\mathbf{a}_{i,d}^T \mathbf{T} \mathbf{\Phi}_{M_{\max}+i}$, for $i = \{1, \cdots, N\}$, i.e.

$$\mathbf{A}_d^{\mathbf{\Phi}} := \begin{bmatrix} \mathbf{a}_{1,d}^T \mathbf{T} \mathbf{\Phi}_{M_{\max}} \\ \vdots \\ \mathbf{a}_{N,d}^T \mathbf{T} \mathbf{\Phi}_{M_{\max}+N-1} \end{bmatrix} := \sum_i^N \boldsymbol{\delta}_i \boldsymbol{\delta}_i^T \mathbf{A}_d \mathbf{T} \mathbf{\Phi}_{M_{\max}+i-1}, \tag{8.75}$$

where $\boldsymbol{\delta}_i \in \mathbb{R}^n$ is a vector whose k^{th} element is defined by $\delta(i-k)$, i.e. the i^{th} element of $\boldsymbol{\delta}_i \in \mathbb{R}^n$ is equal to one and the rest are zero. The vector $\boldsymbol{\delta}_i$ in Eq. (8.76) is defined for $n = N$.

The definition of $\mathbf{A}_d^{\mathbf{\Phi}}$ can be compactly written as

$$\mathbf{A}_d^{\mathbf{\Phi}} = \mathbf{M}_1 \left(\mathbf{I}_N \otimes \mathbf{A}_d \right) \mathbf{M}_{\mathbf{2}}, \tag{8.76}$$

where

$$\mathbf{M}_1 := \begin{bmatrix} \boldsymbol{\delta}_1 \boldsymbol{\delta}_1^T & \cdots & \boldsymbol{\delta}_N \boldsymbol{\delta}_N^T \end{bmatrix}, \qquad \mathbf{M}_2 := \begin{bmatrix} \mathbf{T}\boldsymbol{\Phi}_{M_{\max}} \\ \vdots \\ \mathbf{T}\boldsymbol{\Phi}_{M_{\max}+N-1} \end{bmatrix}. \tag{8.77}$$

Equation Eq. (8.76) shows that $\mathbf{A}_d^{\boldsymbol{\Phi}}$ is linear in $\mathbf{A}_d$.

Thus, the finite-difference approximation for the d^{th} derivative for all the grid points is

$$\mathbf{F}^{(d)} = \frac{1}{(\Delta x)^d} \mathbf{A}_d^{\boldsymbol{\Phi}} \mathbf{F}. \tag{8.78}$$

□

9. CONCLUSIONS AND FUTURE WORK

9.1 Conclusions

A vast number of natural and engineering phenomena are governed by complex differential equations that resist analytical treatment. One such ubiquitous phenomenon is fluid turbulence, governed by the Navier-Stokes (NS) equations. These equations have been the subject of experimental validation, and theoretical and numerical exploration for over a century. Over the last four decades direct numerical simulations have become an indispensable tool for improved insights into turbulence. These computations are challenging due to the characteristic wide range of unsteady spatio-temporal scales of turbulence. This places stringent and expensive requirements on high fidelity simulations and rendering them unachievable at realistic conditions. Parallel computing which employs a large number of processors for computation is a major advance in mitigating this challenge. But, virtually all current approaches require frequent communications and synchronization between processors which becomes a bottleneck in performance and scalability to high processor counts. On the next-generation exascale computers, envisioned to increase the computational power by three orders of magnitude over current state-of-the-art supercomputers, communication will likely be an insurmountable obstacle with current numerical approaches. It is now a common consensus that in order to scale to these exascale machine, paradigm shifts and co-design at both hardware and software levels are required. To this end, we focus on numerical methods and computational algorithms that reduce communications and synchronizations to perform the so-called asynchronous simulations of complex phenomena including turbulence.

The current effort is based on previous work [154] which was mainly confined to the derivation of asynchrony-tolerant (AT) schemes. Building up on that, we first assess the important numerical properties of these schemes, including coversation, stability, and spectral accuracy. We note that since the coefficients of AT schemes depend on the delay seen at the processor boundary, the schemes used in an actual simulation are time varying. Because the scheme changes, the standard

von Neumann analysis for stability can not be used for the AT schemes, even in principle. Moreover, we rigorously prove that the well-established standard von Neumann analysis, in fact fails even for some of the widely used synchronous schemes. We overcome these limitations through our proposed generalized von Neuamann analysis that is also used to assess the spectral accuracy of fully-discrete systems. We show that, despite common belief, the modified wavenumber of spatial schemes can vary with time-step even in the absence of asynchrony. This has implications on the numerical accuracy of multi-scale processes.

Following extensive analysis of the properties of AT schemes, we developed an asynchronous three-dimensional compressible Navier-Stokes solver that facilitiates relaxations of communications and synchronizations. The solver uses two proposed algorithms to effectively introduce delays at processor boundaries. These are the synchronization avoiding algorithm (SAA) and the the communication avoiding algorithms (CAA) that lead to machine-dependent random delays and deterministic periodic delays, respectively. We use this solver to verify the efficacy of AT schemes in resolving the important small and large scale characteristics of compressible homogeneous isotropic turbulence. We also benchmarked the performance of our asynchronous solver on TACC supercomputers and showed that the asynchronous algorithms have better scaling than their synchronous counterpart, primarily due to reduction in communication time. Several steps were also taken to identify and eliminate major bottlenecks in the solver through profiling tools, such as vTune Amplifier, TAU and PAPI. We extended the AT framework to investigate the effect of data asynchrony on reacting flows with one-step and detailed reactions mechanisms. Here again we observed the high numerical accuracy of AT scheme, both qualitatively and quantitatively. We also derived, for the first time, asynchrony-tolerant weighted essentially non-oscillatory schemes (AT-WENO) for problems with shock and discontinuities. The excellent accuracy of these schemes is demonstrated through the propagation of detonation wave.

Finally, with the ultimate goal to derive new optimal AT schemes we developed a unified framework for the derivation of finite difference schemes [155]. We show explicit trade-offs between order of accuracy, spectral accuracy and stability under this unifying framework. We also concretely

proved some properties relating to the nature of the coeffiicients (symmetric/anti-symmteric) of standard finite difference schemes, which albeit well-known have never been rigorously proved before.

9.2 Future Research Directions

We finish this dissertation by pondering on some plausible future research directions that can stem out from our present work. This includes extension of our framework for deriving optimal synchronous spatial schemes to a more a general framework that can be used to obtain optimal asynchrony-tolerant schemes. Besides numerical methods, the other major part of future work includes optimization of the performance of the code. This included restructuring the computations intensive modules of the code, new and more efficient data strutures and wiser use of asynchrony. Each of these directions are described next.

9.2.1 Numerical methods

- In our communication avoiding algorithms, a larger chunk of data needs to be sent and received at every communicating step since the AT schemes use multiple consecutive time-levels. While the CAA reduces the volume of communications by reducing the frequency of communications, further reduction in the communication time in this algorithm can be achieved through new AT schemes which use only one time-level information from the neighboring processor and multiple delayed levels at internal points instead. Once example of such a second order scheme for second derivative is,

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_i^n \approx \frac{\tilde{k}u_{i-1}^{n-\tilde{k}-1} - 2\tilde{k}u_i^{n-\tilde{k}-1} + (\tilde{k}+1)u_{i+1}^{n-\tilde{k}} - 2u_i^n + u_{i+1}^n}{\Delta x^2(\tilde{k}+1)}, \tag{9.1}$$

  where instead of using multiple time levels at the buffer points, multiple time levels are used at the internal points. The size of message in CAA for these new AT schemes is the same as that for the algorithms which communicate at every time step. The effect of these schemes on the performance and accuracy needs to be explored.

- The stability constraint used in our optimization work ensure that solutions are non-growing and are applied to the entire PDE which may involve terms of different characteristics (convection, diffusion, etc). While this type of constraint is standard practice, it seems possible to extend the framework to other approaches that can capture other details of the error dynamics [156], such as group velocity. Furthermore, our current focus was on spatial discretization with a given temporal scheme which could be of arbitrary order. Because of the potential additional effects when coupling space-time operators as is also shown in Chapter 3, it is a natural next step to optimize spatial and temporal schemes simultaneously. Unfortunately, this coupled optimization problem yields non-linear objective functions [144] which makes the mathematical problem much more challenging. While we have tried approaches, such as linearization of the constraints and objective function, a more unified theory for deriving optimal AT schemes requires more work.

- Although not shown here, from some of our recent work we identified another compuationally expensive problem of interest that pretains to propagation of electromagnetic wave through compressible turbulence. The simplified governing equation, which is the paraxial approximation of the Maxwell's equations, has complex coefficients, could only be stablized when implicit schemes are used for propagation. Since implicit schemes require solving a system of linear equations, the scalability of the solver is severely limited. The extension of the optimization framework for deriving stable explicit (synchronous and asynchrony-tolerant) schemes for such a problem will be extremely useful for simulations of the paraxial equations at conditions and scales of interest.

9.2.2 High Performance Computing

- In this work CAA and SAA were presented as two separate algorithms, however, a combination of the two can also be used. This will potentially lead to further reduction in overheads associated with the communication and synchronization.

- There are generalizations that can be introduced where the maximum delay level (L) is dif-

ferent across different regions in the domain, depending upon the level of accuracy required. This does require critical analysis of load balancing to ensure that the processors synchronizing more often (smaller L), have less computational work, so that the synchronization time in these processors does not affect the total execution time.

- From the performance analysis, we observed that while the asynchronous algorithms showed an improved scaling compared to the standard synchronous algorithm, the cache miss rate for the former was found to be higher. Though this miss rate reduces as the processor count is increased, optimization in implementation will help further push the limits of scaling and reduce the overall computation time.

- Another aspect of using asynchronous computations on Exascale machines is to leverage the high flop-rate of GPUs. In hybrid computing architectures, GPUs are expected to handle most of the computations, while CPUs facilitate communications between PEs. With AT schemes, GPUs do not have to wait on the CPUs for the most updated data from the neighboring PEs. This would enhance utilization of GPUs without affecting numerical accuracy or introducing idling penalties. A similar approach has been utilized in [127] where asynchronous copies between CPUs and GPUs are used to overlap computations and data movement, but delayed data with asynchronous computations have not been used. At a compiler level, new asynchronous run-time systems that are capable of dynamic task parallelism are being developed to improve the computation-communication overlap [128, 129, 130, 157, 158]. The AT schemes, which relax synchronization at a mathematical level, can be coupled with such programming models to create highly scalable PDE solvers.

- Asynchrony has also been utilized in [131] for scalable resilience to soft faults. However, in [131] all derivatives are still computed with the most updated data but the computations are re-arranged to ensure maximum overlap between communications and computations. The asynchronization approach utilized in [131] coupled with mathematical level asynchrony with AT schemes can be an effective in pushing the scaling wall. Furthermore, since AT

schemes do not need the updated data, these schemes can also be used to recover node failures without halting the simulation altogether.

Taken together, with optimal-asynchrony-tolerant numerical methods used in a performance optimized flow solver, we can perform scalable, fast and high-fidelity simulations of complex physical processes that are governed by Partial Differential Equations (PDEs). This includes extending the compressible DNS capability to perform simulations at unprecedented range of flow parameters for new insights into fundamental physics of turbulence. The resulting optimal-asynchronous framework can also be leveraged for simulations of turbulent combustion, shock-turbulence interactions and turbulent boundary layer, where different levels of asynchrony and numerical methods can be used in different regions in the domain.

REFERENCES

[1] A. S. Monin and A. M. Yaglom, Statistical Fluid Mechanics, Vol. II. MIT Press, 1975.

[2] U. Frisch, Turbulence. The legacy of A.N. Kolmogorov. 1995.

[3] A. N. Kolmogorov, "Local structure of turbulence in an incompressible fluid for very large reynolds numbers," Dokl. Akad. Nauk. SSSR, vol. 30, pp. 299–303, 1941.

[4] P. Moin and K. Mahesh, "Direct Numerical Simulation: A Tool in Turbulence Research," Annu. Rev. Fluid Mech., vol. 30, no. 1, pp. 539–578, 1998.

[5] T. Ishihara, T. Gotoh, and Y. Kaneda, "Study of High–Reynolds Number Isotropic Turbulence by Direct Numerical Simulation," Annu. Rev. Fluid Mech., vol. 41, no. 1, pp. 165–180, 2009.

[6] V. Yakhot and K. R. Sreenivasan, "Anomalous scaling of structure functions and dynamic constraints on turbulence simulations," J. Stat. Phys., vol. 121, no. 5, pp. 823–841, 2005.

[7] A. Gruber, "Direct Numerical Simulations: Present and Future Role in Fundamental and Applied Combustion Research," in AIAA Scitech 2019 Forum, (San Diego, California), AIAA, 2019.

[8] J. Dongarra, P. Beckman, T. Moore, P. Aerts, G. Aloisio, J.-C. Andre, D. Barkai, J.-Y. Berthou, T. Boku, B. Braunschweig, F. Cappello, B. Chapman, Xuebin Chi, A. Choudhary, S. Dosanjh, T. Dunning, S. Fiore, A. Geist, B. Gropp, R. Harrison, M. Hereld, M. Heroux, A. Hoisie, K. Hotta, Zhong Jin, Y. Ishikawa, F. Johnson, S. Kale, R. Kenway, D. Keyes, B. Kramer, J. Labarta, A. Lichnewsky, T. Lippert, B. Lucas, B. Maccabe, S. Matsuoka, P. Messina, P. Michielse, B. Mohr, M. S. Mueller, W. E. Nagel, H. Nakashima, M. E. Papka, D. Reed, M. Sato, E. Seidel, J. Shalf, D. Skinner, M. Snir, T. Sterling, R. Stevens, F. Streitz, B. Sugar, S. Sumimoto, W. Tang, J. Taylor, R. Thakur, A. Trefethen, M. Valero, A. van

der Steen, J. Vetter, P. Williams, R. Wisniewski, and K. Yelick, "The International Exascale Software Project roadmap," Int. J. High Perform. C., vol. 25, no. 1, pp. 3–60, 2011.

[9] D. A. Donzis and K. Aditya, "Asynchronous finite-difference schemes for partial differential equations," J. Comput. Phys., vol. 274, no. 0, pp. 370–392, 2014.

[10] J. Slotnick, A. Khodadoust, J. Alonso, D. Darmofal, W. Gropp, E. Lurie, and D. Mavriplis, "CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences," 2014.

[11] P. Messina, "The U.S. D.O.E. Exascale Computing Project – Goals and Challenges," 2017.

[12] D. Amitai, A. Averbuch, S. Itzikowitz, and M. Israeli, "Parallel adaptive and time-stabilizing schemes for constant-coefficient parabolic PDEs," Comput. Math. Appl., vol. 24, pp. 33–53, 1992.

[13] D. Amitai, A. Averbuch, S. Itzikowitz, and E. Turkel, "Asynchronous and corrected-asynchronous finite difference solutions of pdes on mimd multiprocessors," Numer. Algorithms, vol. 6, no. 2, pp. 275–296, 1994.

[14] D. Mudigere, S. D. Sherlekar, and S. Ansumali, "Delayed Difference Scheme for Large Scale Scientific Simulations," Phys. Rev. Lett., vol. 113, no. 21, p. 218701, 2014.

[15] A. Mittal and S. Girimaji, "Proxy-equation paradigm: A strategy for massively parallel asynchronous computations," Phys. Rev. E, vol. 96, p. 033304, 2017.

[16] K. Aditya and D. A. Donzis, "High-order asynchrony-tolerant finite difference schemes for partial differential equations," J. Comput. Phys., vol. 350, pp. 550–572, 2017.

[17] J. B. Chen and H. G. Im, "Stretch effects on the burning velocity of turbulent premixed hydrogen/air flames," P. Combust. Inst., vol. 28, no. 1, pp. 211–218, 2000.

[18] J. H. Chen, A. Choudhary, B. de Supinski, M. DeVries, E. R. Hawkes, S. Klasky, W. K. Liao, K. L. Ma, J. Mellor-Crummey, N. Podhorszki, R. Sankaran, S. Shende, and C. S. Yoo,

"Terascale direct numerical simulations of turbulent combustion using S3D," Comput. Sci. Disc., vol. 2, no. 1, p. 015001, 2009.

[19] K. Kumari and D. A. Donzis, "Direct numerical simulations of turbulent flows using high-order asynchrony-tolerant schemes: Accuracy and performance," J. Comput. Phys., vol. 419, p. 109626, 2020.

[20] C. Hirsch, Numerical computation of internal and external flows, vol. 1. Wiley, New York, 1994.

[21] J. Stoer and R. Bulirsch, Introduction to Numerical Analysis, vol. 12. Springer Science & Business Media, 2013.

[22] K. Lee, Analysis of Large-Scale Asynchronous Switched Dynamical Systems. PhD thesis, Texas A&M University, 2015.

[23] J. Miller, "On the location of zeros of certain classes of polynomials with applications to numerical analysis," J. I. Math. Appl., vol. 8, 1971.

[24] P. K. Yeung, K. R. Sreenivasan, and S. B. Pope, "Effects of finite spatial and temporal resolution in direct numerical simulations of incompressible isotropic turbulence," Phys. Rev. Fluids, vol. 3, p. 064603, 2018.

[25] A. Gruber, E. S. Richardson, K. Aditya, and J. H. Chen, "Direct numerical simulations of premixed and stratified flame propagation in turbulent channel flow," Phys. Rev. Fluids, vol. 3, no. 11, p. 110507, 2018.

[26] P. D. Lax and R. D. Richtmyer, "Survey of the stability of linear finite difference equations," Comm. Pure Appl. Math., vol. 9, no. 2, pp. 267–293, 1956.

[27] J. G. Charney, R. FjÖrtoft, and J. V. Neumann, "Numerical integration of the barotropic vorticity equation," Tellus, vol. 2, no. 4, pp. 237–254, 1950.

[28] J. V. Neumann and R. D. Richtmyer, "A method for the numerical calculation of hydrodynamic shocks," J. of Appl. Phys., vol. 21, no. 3, pp. 232–237, 1950.

[29] C. Fletcher, Computational Techniques for Fluid Dynamics 1: Fundamental and General Techniques. Springer Series in Computational Physics, Springer Berlin Heidelberg, 2013.

[30] D. Anderson, J. Tannehill, and R. Pletcher, Computational Fluid Mechanics and Heat Transfer. Series in Computational and Physical Processes in Mechanics and Thermal Sciences, CRC Press, 2016.

[31] K. Kumari and D. A. Donzis, "A generalized von Neumann analysis for multi-level schemes: Stability and spectral accuracy," J. Comput. Phys., vol. 424, p. 109868, 2021.

[32] E. C. D. Fort and S. P. Frankel, "Stability conditions in the numerical treatment of parabolic differential equations," Mathematical Tables and Other Aids to Computation, vol. 7, no. 43, pp. 135–152, 1953.

[33] H. S. Hall, Analytic Theory of Continued Fractions. Chelsea, New York, 1948.

[34] C.-W. Shu, "Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws," in Advanced Numerical Approximation of Nonlinear Hyperbolic Equations: Lectures given at the 2nd Session of the Centro Internazionale Matematico Estivo (C.I.M.E.) Held in Cetraro, Italy, June 23–28, 1997 (A. Quarteroni, ed.), pp. 325–432, Berlin, Heidelberg: Springer Berlin Heidelberg, 1998.

[35] C.-W. Shu, "High Order Weighted Essentially Nonoscillatory Schemes for Convection Dominated Problems," SIAM Rev., vol. 51, no. 1, pp. 82–126, 2009.

[36] S. Pirozzoli, "Numerical methods for high-speed flows," Annu. Rev. Fluid Mech., vol. 43, pp. 163–194, 2011.

[37] I. Bermejo-Moreno, J. Bodart, J. Larsson, B. M. Barney, J. W. Nichols, and S. Jones, "Solving the compressible navier-stokes equations on up to 1.97 million cores and 4.1 trillion grid points," in SC '13: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, pp. 1–10, 2013.

[38] S. Desai, Y. J. Kim, W. Song, M. B. Luong, F. E. H. Pérez, R. Sankaran, and H. G. Im, "Direct numerical simulations of turbulent reacting flows with shock waves and stiff chemistry using many-core/gpu acceleration," Comput. Fluids, vol. 215, p. 104787, 2021.

[39] A. Mosedale and D. Drikakis, "Assessment of Very High Order of Accuracy in Implicit LES models," J. Fluids Eng., vol. 129, no. 12, pp. 1497–1503, 2007.

[40] K. Ritos, I. W. Kokkinakis, and D. Drikakis, "Physical insight into the accuracy of finely-resolved iLES in turbulent boundary layers," Comput. Fluids, vol. 169, pp. 309–316, 2018.

[41] K. Ritos, I. W. Kokkinakis, and D. Drikakis, "Performance of high-order implicit large eddy simulations," Comput. Fluids, vol. 173, pp. 307–312, 2018.

[42] C.-W. Shu, "Essentially non-oscillatory and weighted essentially non-oscillatory schemes," Acta Numer., p. 63, 2020.

[43] G.-S. Jiang and C.-W. Shu, "Efficient implementation of weighted eno schemes," J. Comput. Phys., vol. 126, no. 1, pp. 202–228, 1996.

[44] J. Shi, C. Hu, and C.-W. Shu, "A Technique of Treating Negative Weights in WENO Schemes," J. Comput. Phys., vol. 175, no. 1, pp. 108–127, 2002.

[45] Y.-y. Liu, C.-w. Shu, and M.-p. Zhang, "On the positivity of linear weights in weno approximations," Acta Math. Appl. Sin.-E, vol. 25, no. 3, pp. 503–538, 2009.

[46] C. Canuto, ed., Spectral Methods in Fluid Dynamics. Springer Series in Computational Physics, Berlin ; New York: Springer-Verlag, corr. 3rd print ed., 1988.

[47] S. K. Lele, "Compact finite difference schemes with spectral-like resolution," J. Comput. Phys., vol. 103, no. 1, pp. 16–42, 1992.

[48] K. Mahesh, S. Lee, S. K. Lele, and P. Moin, "The interaction of an isotropic field of acoustic waves with a shock wave," J. Fluid Mech., vol. 300, pp. 383–407, 1995.

[49] M. R. Petersen and D. Livescu, "Forcing for statistically stationary compressible isotropic turbulence," Phys. Fluids, vol. 22, no. 11, p. 116101, 2010.

[50] S. Jagannathan and D. A. Donzis, "Massively parallel direct numerical simulations of forced compressible turbulence: A hybrid mpi/openmp approach," in Proceedings of the 1st Conference of the Extreme Science and Engineering Discovery Environment: Bridging from the eXtreme to the Campus and Beyond, XSEDE '12, (New York, NY, USA), pp. 23:1–23:8, ACM, 2012.

[51] C. H. Chen and D. A. Donzis, "Shock–turbulence interactions at high turbulence intensities," J. Fluid Mech., vol. 870, pp. 813–847, 2019.

[52] S. Khurshid and D. A. Donzis, "Decaying compressible turbulence with thermal non-equilibrium," Phys. Fluids, vol. 31, no. 1, p. 015103, 2019.

[53] A. W. Cook, W. H. Cabot, P. L. Williams, B. J. Miller, B. R. d. Supinski, R. K. Yates, and M. L. Welcome, "Tera-scalable algorithms for variable-density elliptic hydrodynamics with spectral accuracy," in Proceedings of the 2005 ACM/IEEE conference on Supercomputing, no. 1 in SC'05, (Washington, DC, USA), p. 60, IEEE Computer Society, 2005.

[54] D. A. Donzis, P. K. Yeung, and D. Pekurovsky, "Turbulence simulations on $O(10^4)$ processors," TeraGrid 2008 Conference., 2008.

[55] M. Lee, N. Malaya, and R. D. Moser, "Petascale direct numerical simulation of turbulent channel flow on up to 786K cores," in SC '13: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, pp. 1–11, 2013.

[56] Y. Minamoto and J. H. Chen, "DNS of a turbulent lifted DME jet flame," Combust. Flame, vol. 169, pp. 38–50, 2016.

[57] S. Jagannathan and D. A. Donzis, "Reynolds and mach number scaling in solenoidally-forced compressible turbulence using high-resolution direct numerical simulations," J. Fluid Mech., vol. 789, pp. 669–707, 2016.

[58] J. Wang, T. Gotoh, and T. Watanabe, "Spectra and statistics in compressible isotropic turbulence," Phys. Rev. Fluids, vol. 2, no. 1, p. 013403, 2017.

[59] D. A. Donzis and S. Jagannathan, "Fluctuations of thermodynamic variables in stationary compressible turbulence," J. Fluid Mech., vol. 733, p. 221–244, 2013.

[60] J. Wang, L.-P. Wang, Z. Xiao, Y. Shi, and S. Chen, "A hybrid numerical simulation of isotropic compressible turbulence," J. Comput. Phys., vol. 229, no. 13, pp. 5257–5279, 2010.

[61] S. Kida and S. A. Orszag, "Energy and spectral dynamics in decaying compressible turbulence," J. Sci. Comput., vol. 7, no. 1, pp. 1–34, 1992.

[62] R. Samtaney, D. I. Pullin, and B. Kosović, "Direct numerical simulation of decaying compressible turbulence and shocklet statistics," Phys. Fluids, vol. 13, no. 5, pp. 1415–1430, 2001.

[63] S. R. Yoffe and W. D. McComb, "Onset criteria for freely decaying isotropic turbulence," Phys. Rev. Fluids, vol. 3, p. 104605, 2018.

[64] S. B. Pope, Turbulent Flows. Cambridge University Press, 2000.

[65] K. R. Sreenivasan and R. A. Antonia, "The phenomenology of small-scale turbulence," Annu. Rev. Fluid Mech., vol. 29, no. 1, pp. 435–472, 1997.

[66] S. Kida and S. A. Orszag, "Energy and spectral dynamics in forced compressible turbulence," J. Sci. Comput., vol. 5, no. 2, pp. 85–125, 1990.

[67] S. K. Lele, "Compressibility effects on turbulence," Annu. Rev. Fluid Mech., vol. 26, no. 1, pp. 211–254, 1994.

[68] K. R. Sreenivasan, "On the universality of the kolmogorov constant," Phys. Fluids, vol. 7, no. 11, pp. 2778–2784, 1995.

[69] P. K. Yeung and Y. Zhou, "Universality of the kolmogorov constant in numerical simulations of turbulence," Phys. Rev. E, vol. 56, pp. 1746–1752, 1997.

[70] D. A. Donzis and K. R. Sreenivasan, "The bottleneck effect and the kolmogorov constant in isotropic turbulence," J. Fluid Mech., vol. 657, p. 171–188, 2010.

[71] R. H. Kraichnan, "The structure of isotropic turbulence at very high Reynolds numbers," J. Fluid Mech., vol. 5, no. 04, p. 497, 1959.

[72] R. H. Kraichnan, "Intermittency in the Very Small Scales of Turbulence," Phys. Fluids, vol. 10, no. 9, p. 2080, 1967.

[73] L. Sirovich, L. Smith, and V. Yakhot, "Energy spectrum of homogeneous and isotropic turbulence in far dissipation range," Phys. Rev. Lett., vol. 72, no. 3, pp. 344–347, 1994.

[74] C. Foias, O. Manley, and L. Sirovich, "Empirical and Stokes eigenfunctions and the far-dissipative turbulent spectrum," Phys. Fluids A: Fluid, vol. 2, no. 3, pp. 464–467, 1990.

[75] S. Khurshid, D. A. Donzis, and K. R. Sreenivasan, "Energy spectrum in the dissipation range," Phys. Rev. Fluids, vol. 3, no. 8, p. 082601, 2018.

[76] T. Gotoh and D. Fukayama, "Pressure spectrum in homogeneous turbulence," Phys. Rev. Lett., vol. 86, pp. 3775–3778, 2001.

[77] D. A. Donzis and J. P. John, "Universality and scaling in homogeneous compressible turbulence," Phys. Rev. Fluids, vol. 5, p. 084609, 2020.

[78] P. K. Yeung, D. A. Donzis, and K. R. Sreenivasan, "Dissipation, enstrophy and pressure statistics in turbulence simulations at high Reynolds numbers," J. Fluid Mech., vol. 700, pp. 5–15, 2012.

[79] P. Davidson, Turbulence: An Introduction for Scientists and Engineers, vol. 2. Oxford University Press, 2015.

[80] R. M. Kerr, "Higher-order derivative correlations and the alignment of small-scale structures in isotropic numerical turbulence," J. Fluid Mech., vol. 153, p. 31–58, 1985.

[81] T. Gotoh, D. Fukayama, and T. Nakano, "Velocity field statistics in homogeneous steady turbulence obtained using a high-resolution direct numerical simulation," Phys. Fluids, vol. 14, no. 3, pp. 1065–1081, 2002.

[82] D. A. Donzis, P. K. Yeung, and K. R. Sreenivasan, "Dissipation and enstrophy in isotropic turbulence: Resolution effects and scaling in direct numerical simulations," Phys. Fluids, vol. 20, no. 4, p. 045108, 2008.

[83] D. A. Donzis and S. Jagannathan, "On the Relation between Small-scale Intermittency and Shocks in Turbulent Flows," Procedia IUTAM, vol. 9, pp. 3–15, 2013.

[84] D. Porter, A. Pouquet, and P. Woodward, "Intermittency in Compressible Flows," in Advances in Turbulence VII (R. Moreau and U. Frisch, eds.), vol. 46, pp. 255–258, Dordrecht: Springer Netherlands, 1998.

[85] L. Pan, P. Padoan, and A. G. Kritsuk, "Dissipative Structures in Supersonic Turbulence," Phys. Rev. Lett., vol. 102, no. 3, p. 034501, 2009.

[86] V. Yakhot and D. A. Donzis, "Anomalous exponents in strong turbulence," Physica D, vol. 384-385, pp. 12–17, 2018.

[87] V. Yakhot and D. Donzis, "Emergence of Multiscaling in a Random-Force Stirred Fluid," Phys. Rev. Lett., vol. 119, no. 4, p. 044501, 2017.

[88] J. Schumacher, J. D. Scheel, D. Krasnov, D. A. Donzis, V. Yakhot, and K. R. Sreenivasan, "Small-scale universality in fluid turbulence," P. Natl. Acad. Sci. USA, vol. 111, no. 30, pp. 10961–10965, 2014.

[89] P. K. Yeung, X. M. Zhai, and K. R. Sreenivasan, "Extreme events in computational turbulence," P. Natl. Acad. Sci. USA, vol. 112, no. 41, pp. 12633–12638, 2015.

[90] V. Eswaran and S. Pope, "An examination of forcing in direct numerical simulations of turbulence," Comput. Fluids, vol. 16, no. 3, pp. 257 – 278, 1988.

[91] M. R. Petersen and D. Livescu, "Forcing for statistically stationary compressible isotropic turbulence," Phys. Fluids, vol. 22, no. 11, p. 116101, 2010.

[92] D. A. Donzis and A. F. Maqui, "Statistically steady states of forced isotropic turbulence in thermal equilibrium and non-equilibrium," J. Fluid Mech., vol. 797, pp. 181–200, 2016.

[93] E. D. Siggia, "Numerical study of small-scale intermittency in three-dimensional turbulence," J. Fluid Mech., vol. 107, no. -1, p. 375, 1981.

[94] T. Zhou and R. A. Antonia, "Reynolds number dependence of the small-scale structure of grid turbulence," J. Fluid Mech., vol. 406, p. 81–107, 2000.

[95] C. K. Westbrook, Y. Mizobuchi, T. J. Poinsot, P. J. Smith, and J. Warnatz, "Computational combustion," P. Combust. Inst., vol. 30, no. 1, pp. 125–157, 2005.

[96] H. Pitsch, "Large-Eddy Simulation of Turbulent Combustion," Annu. Rev. Fluid Mech., vol. 38, no. 1, pp. 453–482, 2006.

[97] Y. Mizobuchi, S. Tachibana, J. Shinio, S. Ogawa, and T. Takeno, "A numerical analysis of the structure of a turbulent hydrogen jet lifted flame," P. Combust. Inst., vol. 29, no. 2, pp. 2009–2015, 2002.

[98] E. R. Hawkes, R. Sankaran, J. C. Sutherland, and J. H. Chen, "Scalar mixing in direct numerical simulations of temporally evolving plane jet flames with skeletal CO/H2 kinetics," P. Combust. Inst., vol. 31, no. 1, pp. 1633–1640, 2007.

[99] J. B. Bell, M. S. Day, J. F. Grcar, M. J. Lijewski, J. F. Driscoll, and S. A. Filatyev, "Numerical simulation of a laboratory-scale turbulent slot flame," P. Combust. Inst., vol. 31, no. 1, pp. 1299–1307, 2007.

[100] K. Aditya, A. Gruber, C. Xu, T. Lu, A. Krisman, M. R. Bothien, and J. H. Chen, "Direct numerical simulation of flame stabilization assisted by autoignition in a reheat gas turbine combustor," P. Combust. Inst., vol. 37, no. 2, pp. 2635–2642, 2019.

[101] M. Visbal, D. Gaitonde, and D. Rizzetta, "High-order schemes for DNS/LES and CAA on curvilinear dynamic meshes," DNS/LES Progress and Challenges, Greyden Press, 2001.

[102] A. Attili, F. Bisetti, M. E. Mueller, and H. Pitsch, "Formation, growth, and transport of soot in a three-dimensional turbulent non-premixed jet flame," Combust. Flame, vol. 161, no. 7, pp. 1849–1865, 2014.

[103] B. Savard, E. R. Hawkes, K. Aditya, H. Wang, and J. H. Chen, "Regimes of premixed turbulent spontaneous ignition and deflagration under gas-turbine reheat combustion conditions," Combust. Flame, vol. 208, pp. 402–419, 2019.

[104] J. Zhang, M. B. Luong, F. E. H. Pérez, D. Han, H. G. Im, and Z. Huang, "Exergy loss characteristics of DME/air and ethanol/air mixtures with temperature and concentration fluctuations under HCCI/SCCI conditions: A DNS study," Combust. Flame, vol. 226, pp. 334–346, 2021.

[105] L. Berger, R. Hesse, K. Kleinheinz, M. J. Hegetschweiler, A. Attili, J. Beeckmann, G. T. Linteris, and H. Pitsch, "A DNS study of the impact of gravity on spherically expanding laminar premixed flames," Combust. Flame, vol. 216, pp. 412–425, 2020.

[106] Y. J. Kim, B. J. Lee, and H. G. Im, "Effects of Differential Diffusion on the Stabilization of Unsteady Lean Premixed Flames Behind a Bluff-Body," Flow Turbul. Combust., pp. 1125–1141, 2021.

[107] G. Nivarti and S. Cant, "Direct numerical simulation of the bending effect in turbulent premixed flames," P. Combust. Inst., vol. 36, no. 2, pp. 1903–1910, 2017.

[108] G. Beardsell and G. Blanquart, "Fully compressible simulations of the impact of acoustic waves on the dynamics of laminar premixed flames for engine-relevant conditions," P. Combust. Inst., vol. 38, no. 2, pp. 1923–1931, 2021.

[109] K. Aditya, T. Gysi, G. Kwasniewski, T. Hoefler, D. A. Donzis, and J. H. Chen, "A scalable weakly-synchronous algorithm for solving partial differential equations," 2019.

[110] R. J. Kee, F. M. Rupley, E. Meeks, and J. A. Miller, "Chemkin-iii: A fortran chemical kinetics package for the analysis of gas-phase chemical and plasma kinetics," tech. rep., Sandia National Labs., Livermore, CA (United States), 1996.

[111] R. J. Kee, G. Dixon-Lewis, J. Warnatz, M. E. Coltrin, and J. A. Miller, "A fortran computer code package for the evaluation of gas-phase multicomponent transport properties," Sandia National Laboratories Report SAND86-8246, vol. 13, pp. 80401–1887, 1986.

[112] M. Baum, T. Poinsot, and D. Thevenin, "Accurate boundary conditions for multicomponent reactive flows," J. Comput. Phys., vol. 116, no. 2, pp. 247 – 261, 1995.

[113] K. W. Thompson, "Time dependent boundary conditions for hyperbolic systems," J. Comput. Phys., vol. 68, no. 1, pp. 1–24, 1987.

[114] T. J. Poinsot and S. K. Lele, "Boundary conditions for direct simulations of compressible viscous flows," J. Comput. Phys., vol. 101, no. 1, pp. 104–129, 1992.

[115] Z. Luo, C. S. Yoo, E. S. Richardson, J. H. Chen, C. K. Law, and T. Lu, "Chemical explosive mode analysis for a turbulent lifted ethylene jet flame in highly-heated coflow," Combust. Flame, vol. 159, no. 1, pp. 265–274, 2012.

[116] C. S. Yoo, Y. Wang, A. Trouvé, and H. G. Im, "Characteristic boundary conditions for direct simulations of turbulent counterflow flames," Combust. Theor. Model., vol. 9, no. 4, pp. 617–646, 2005.

[117] C. S. Yoo and H. G. Im, "Characteristic boundary conditions for simulations of compressible reacting flows with multi-dimensional, viscous and reaction effects," Combust. Theor. Model., vol. 11, no. 2, pp. 259–286, 2007.

[118] M. P. Burke, M. Chaos, Y. Ju, F. L. Dryer, and S. J. Klippenstein, "Comprehensive h2/o2 kinetic model for high-pressure combustion," Int. J. Chem. Kinet., vol. 44, no. 7, pp. 444–474, 2012.

[119] J. C. Sutherland and C. A. Kennedy, "Improved boundary conditions for viscous, reacting, compressible flows," J. Comput. Phys., vol. 191, no. 2, pp. 502–524, 2003.

[120] C. K. Westbrook, "Chemical kinetics of hydrocarbon oxidation in gaseous detonations," Combust. Flame, vol. 46, pp. 191 – 210, 1982.

[121] R. Deiterding, "Parallel adaptive simulation of multi-dimensional detonation structures," 2003.

[122] Y. Lv and M. Ihme, "Discontinuous Galerkin method for multicomponent chemically reacting flows and combustion," J. Comput. Phys., vol. 270, pp. 105–137, 2014.

[123] R. W. Houim and K. K. Kuo, "A low-dissipation and time-accurate method for compressible multi-component flow with variable specific heat ratios," J. Comput. Phys., vol. 230, no. 23, pp. 8527–8553, 2011.

[124] M. B. Luong, S. Desai, F. E. H. Pérez, R. Sankaran, B. Johansson, and H. G. Im, "A statistical analysis of developing knock intensity in a mixture with temperature inhomogeneities," P. Combust. Inst., 2020.

[125] M. B. Luong, S. Desai, F. E. Hernández Pérez, R. Sankaran, B. Johansson, and H. G. Im, "Effects of turbulence and temperature fluctuations on knock development in an ethanol/air mixture," Flow Turbul. Combust., pp. 1–21, 2020.

[126] G. Lodato and H. Pitsch, "Characteristic outflows with optimal transverse terms: The edges and corners coupling algorithm," Center for Turbulence Research, Annual Research Briefs, p. 12, 2011.

[127] P. K. Yeung and K. Ravikumar, "Advancing understanding of turbulence through extreme-scale computation: Intermittency and simulations at large problem sizes," Phys. Rev. Fluids, vol. 5, p. 110517, 2020.

[128] E. Slaughter, W. Lee, S. Treichler, M. Bauer, and A. Aiken, "Regent: A high-productivity programming language for hpc with logical regions," in Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC '15, (New York, NY, USA), Association for Computing Machinery, 2015.

[129] B. Acun, A. Gupta, N. Jain, A. Langer, H. Menon, E. Mikida, X. Ni, M. Robson, Y. Sun, E. Totoni, L. Wesolowski, and L. Kale, "Parallel programming with migratable objects: Charm++ in practice," in SC '14: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 647–658, 2014.

[130] A. Kulkarni and A. Lumsdaine, "A comparative study of asynchronous many-tasking runtimes: Cilk, charm++, parallex and am++," arXiv preprint arXiv:1904.00518, 2019.

[131] H. Kolla, J. R. Mayo, K. Teranishi, and R. C. Armstrong, "Improving scalability of silent-error resilience for message-passing solvers via local recovery and asynchrony," 2020

IEEE/ACM 10th Workshop on Fault Tolerance for HPC at eXtreme Scale (FTXS), pp. 1–10, 2020.

[132] C. K. Tam and J. C. Webb, "Dispersion-relation-preserving finite difference schemes for computational acoustics," J. Comput. Phys., vol. 107, no. 2, pp. 262 – 281, 1993.

[133] J. W. Kim and D. J. Lee, "Optimized compact finite difference schemes with maximum resolution," AIAA Journal, vol. 34, no. 5, pp. 887–893, 1996.

[134] Z. Wang and R. Chen, "Optimized weighted essentially nonoscillatory schemes for linear waves with discontinuity," J. Comput. Phys., vol. 174, no. 1, pp. 381 – 404, 2001.

[135] M. Zhuang and R. F. Chen, "Applications of high-order optimized upwind schemes for computational aeroacoustics," AIAA Journal, vol. 40, no. 3, pp. 443–449, 2002.

[136] D. Ponziani, S. Pirozzoli, and F. Grasso, "Development of optimized weighted-ENO schemes for multiscale compressible flows," Int. J. Numer. Meth. Fluids, vol. 42, no. 9, pp. 953–977, 2003.

[137] C. Bogey and C. Bailly, "A family of low dispersive and low dissipative explicit schemes for flow and noise computations," J. Comput. Phys., vol. 194, no. 1, pp. 194 – 214, 2004.

[138] M. P. Martin, E. M. Taylor, M. Wu, and V. G. Weirs, "A bandwidth-optimized WENO scheme for the effective direct numerical simulation of compressible turbulence," J. Comput. Phys., vol. 220, no. 1, pp. 270–289, 2006.

[139] T. K. Sengupta, S. K. Sircar, and A. Dipankar, "High accuracy schemes for DNS and acoustics," J. Sci. Comput., vol. 26, no. 2, pp. 151–193, 2006.

[140] J. Fang, Z. Li, and L. Lu, "An optimized low-dissipation monotonicity-preserving scheme for numerical simulations of high-speed turbulent flows," J. Sci. Comput., vol. 56, no. 1, pp. 67–95, 2013.

[141] J.-H. Zhang and Z.-X. Yao, "Optimized explicit finite-difference schemes for spatial derivatives using maximum norm," J. Comput. Phys., vol. 250, pp. 511 – 526, 2013.

[142] S. Pirozzoli, "Performance analysis and optimization of finite-difference schemes for wave propagation problems," J. Comput. Phys., vol. 222, no. 2, pp. 809–831, 2007.

[143] G. Ashcroft and X. Zhang, "Optimized prefactored compact schemes," J. Comput. Phys., vol. 190, no. 2, pp. 459 – 477, 2003.

[144] Z. Haras and S. Ta'asan, "Finite difference schemes for long-time integration," J. Comput. Phys., vol. 114, no. 2, pp. 265 – 279, 1994.

[145] T. Sengupta, G. Ganeriwal, and S. De, "Analysis of central and upwind compact schemes," J. Comput. Phys., vol. 192, no. 2, pp. 677 – 694, 2003.

[146] T. K. Sengupta and A. Bhole, "Error dynamics of diffusion equation: Effects of numerical diffusion and dispersive diffusion," J. Comput. Phys., vol. 266, pp. 240 – 251, 2014.

[147] D. P. Bertsekas, Nonlinear programming. Athena scientific Belmont, 1999.

[148] M. Grant, S. Boyd, and Y. Ye, "CVX: Matlab software for disciplined convex programming," 2008.

[149] A. Ben-Tal and A. Nemirovski, Lectures on modern convex optimization: analysis, algorithms, and engineering applications. SIAM, 2001.

[150] R. Y. Zhang and J. Lavaei, "Efficient algorithm for large-and-sparse LMI feasibility problems," IEEE CDC, pp. 6868–6875, 2018.

[151] H. Eberhard, "The partial differential equation $u_t + uu_x = \mu u_{xx}$," Commun. Pure Appl. Math., vol. 3, no. 3, pp. 201–230, 1950.

[152] M. I. Cole, "Algorithmic Skeletons: Structured Management of Parallel Computation," p. 137.

[153] V. M. Deshpande, R. Bhattacharya, and D. A. Donzis, "A unified framework to generate optimized compact finite difference schemes," J. Comput. Phys., vol. 432, p. 110157, 2021.

[154] K. Aditya, Highly Scalable Asynchronous Computing Method for Partial Differential Equations: A Path Towards Exascale. PhD thesis, Texas A&M University, 2017.

[155] K. Kumari, R. Bhattacharya, and D. A. Donzis, "A unified approach for deriving optimal finite differences," J. Comput. Phys., vol. 399, p. 108957, 2019.

[156] Y. Shokin and K. Roesner, The Method of Differential Approximation. Springer Berlin Heidelberg, 2011.

[157] M. Zandifar, M. Abdul Jabbar, A. Majidi, D. Keyes, N. M. Amato, and L. Rauchwerger, "Composing Algorithmic Skeletons to Express High-Performance Scientific Applications," in Proceedings of the 29th ACM on International Conference on Supercomputing - ICS '15, (Newport Beach, California, USA), pp. 415–424, ACM Press, 2015.

[158] M. Zandifar, N. Thomas, N. M. Amato, and L. Rauchwerger, "The stapl Skeleton Framework," in Languages and Compilers for Parallel Computing (J. Brodman and P. Tu, eds.), vol. 8967, pp. 176–190, Cham: Springer International Publishing, 2015.

APPENDIX A

ASYNCHRONY-TOLERANT SCHEMES

(Derivative, Order)	Boundary	Scheme
(2,4)	Left	$\frac{1}{2}(\tilde{k}^2+3\tilde{k}+2)\left(-u^n_{i+2}+16u^n_{i+1}-30u^n_i+16u^{n-\tilde{k}}_{i-1}-u^{n-\tilde{k}}_{i-2}\right)/12\Delta x^2$ $-(\tilde{k}^2+2\tilde{k})\left(-u^n_{i+2}+16u^n_{i+1}-30u^n_i+16u^{n-\tilde{k}-1}_{i-1}-u^{n-\tilde{k}-1}_{i-2}\right)/12\Delta x^2$ $+\frac{1}{2}(\tilde{k}^2+\tilde{k})\left(-u^n_{i+2}+16u^n_{i+1}-30u^n_i+16u^{n-\tilde{k}-2}_{i-1}-u^{n-\tilde{k}-2}_{i-2}\right)/12\Delta x^2$
(2,4)	Right	$\frac{1}{2}(\tilde{k}^2+3\tilde{k}+2)\left(-u^{n-\tilde{k}}_{i+2}+16u^{n-\tilde{k}}_{i+1}-30u^n_i+16u^n_{i-1}-u^n_{i-2}\right)/12\Delta x^2$ $-(\tilde{k}^2+2\tilde{k})\left(-u^{n-\tilde{k}-1}_{i+2}+16u^{n-\tilde{k}-1}_{i+1}-30u^n_i+16u^n_{i-1}-u^n_{i-2}\right)/12\Delta x^2$ $+\frac{1}{2}(\tilde{k}^2+\tilde{k})\left(-u^{n-\tilde{k}-2}_{i+2}+16u^{n-\tilde{k}-2}_{i+1}-30u^n_i+16u^n_{i-1}-u^n_{i-2}\right)/12\Delta x^2$
(1,4)	Left	$\frac{1}{2}(\tilde{k}^2+3\tilde{k}+2)\left(-u^n_{i+2}+8u^n_{i+1}-8u^{n-\tilde{k}}_{i-1}+u^{n-\tilde{k}}_{i-2}\right)/12\Delta x$ $-(\tilde{k}^2+2\tilde{k})\left(-u^n_{i+2}+8u^n_{i+1}-8u^{n-\tilde{k}-1}_{i-1}+u^{n-\tilde{k}-1}_{i-2}\right)/12\Delta x$ $+\frac{1}{2}(\tilde{k}^2+\tilde{k})\left(-u^n_{i+2}+8u^n_{i+1}-8u^{n-\tilde{k}-2}_{i-1}+u^{n-\tilde{k}-2}_{i-2}\right)/12\Delta x$
(1,4)	Right	$\frac{1}{2}(\tilde{k}^2+3\tilde{k}+2)\left(-u^{n-\tilde{k}}_{i+2}+8u^{n-\tilde{k}}_{i+1}-8u^n_{i-1}+u^n_{i-2}\right)/12\Delta x$ $-(\tilde{k}^2+2\tilde{k})\left(-u^{n-\tilde{k}-1}_{i+2}+8u^{n-\tilde{k}-1}_{i+1}-8u^n_{i-1}+u^n_{i-2}\right)/12\Delta x$ $+\frac{1}{2}(\tilde{k}^2+\tilde{k})\left(-u^{n-\tilde{k}-2}_{i+2}+8u^{n-\tilde{k}-2}_{i+1}-8u^n_{i-1}+u^n_{i-2}\right)/12\Delta x$

Table A.1: Fourth-order asynchrony-tolerant (AT) schemes for left and right boundary used in numerical simulations for first and second derivative.

(Derivative, Order)	Boundary	Scheme
(2,6)	Left	$\frac{1}{6}(\tilde{k}^3+6\tilde{k}^2+11\tilde{k}+6)\frac{\left(2u^n_{i+3}-27u^n_{i+2}+270u^n_{i+1}-490u^n_i+270u^{n-\tilde{k}}_{i-1}-27u^{n-\tilde{k}}_{i-2}+2u^{n-\tilde{k}}_{i-3}\right)}{180\Delta x^2}$ $-\frac{1}{2}(\tilde{k}^3+5\tilde{k}^2+6\tilde{k})\frac{\left(2u^n_{i+3}-27u^n_{i+2}+270u^n_{i+1}-490u^n_i+270u^{n-\tilde{k}-1}_{i-1}-27u^{n-\tilde{k}-1}_{i-2}+2u^{n-\tilde{k}-1}_{i+3}\right)}{180\Delta x^2}$ $+\frac{1}{2}(\tilde{k}^3+4\tilde{k}^2+3\tilde{k})\frac{\left(2u^n_{i+3}-27u^n_{i+2}+270u^n_{i+1}-490u^n_i+270u^{n-\tilde{k}-2}_{i-1}-27u^{n-\tilde{k}-2}_{i-2}+2u^{n-\tilde{k}-2}_{i+3}\right)}{180\Delta x^2}$ $-\frac{1}{6}(\tilde{k}^3+3\tilde{k}^2+2\tilde{k})\frac{\left(2u^n_{i+3}-27u^n_{i+2}+270u^n_{i+1}-490u^n_i+270u^{n-\tilde{k}-3}_{i-1}-27u^{n-\tilde{k}-3}_{i-2}+2u^{n-\tilde{k}-3}_{i+3}\right)}{180\Delta x^2}$
(2,6)	Right	$\frac{1}{6}(\tilde{k}^3+6\tilde{k}^2+11\tilde{k}+6)\frac{\left(2u^{n-\tilde{k}}_{i+3}-27u^{n-\tilde{k}}_{i+2}+270u^{n-\tilde{k}}_{i+1}-490u^n_i+270u^n_{i-1}-27u^n_{i-2}+2u^n_{i-3}\right)}{180\Delta x^2}$ $-\frac{1}{2}(\tilde{k}^3+5\tilde{k}^2+6\tilde{k})\frac{\left(2u^{n-\tilde{k}-1}_{i+3}-27u^{n-\tilde{k}-1}_{i+2}+270u^{n-\tilde{k}-1}_{i+1}-490u^n_i+270u^n_{i-1}-27u^n_{i-2}+2u^n_{i-3}\right)}{180\Delta x^2}$ $+\frac{1}{2}(\tilde{k}^3+4\tilde{k}^2+3\tilde{k})\frac{\left(2u^{n-\tilde{k}-2}_{i+3}-27u^{n-\tilde{k}-2}_{i+2}+270u^{n-\tilde{k}-2}_{i+1}-490u^n_i+270u^n_{i-1}-27u^n_{i-2}+2u^n_{i-3}\right)}{180\Delta x^2}$ $-\frac{1}{6}(\tilde{k}^3+3\tilde{k}^2+2\tilde{k})\frac{\left(2u^{n-\tilde{k}-3}_{i+3}-27u^{n-\tilde{k}-3}_{i+2}+270u^{n-\tilde{k}-3}_{i+1}-490u^n_i+270u^n_{i-1}-27u^n_{i-2}+2u^n_{i-3}\right)}{180\Delta x^2}$
(1,6)	Left	$\frac{1}{6}(\tilde{k}^3+6\tilde{k}^2+11\tilde{k}+6)\left(u^n_{i+3}-9u^n_{i+2}+45u^n_{i+1}-45u^{n-\tilde{k}}_{i-1}+9u^{n-\tilde{k}}_{i-2}-u^{n-\tilde{k}}_{i-3}\right)/60\Delta x$ $-\frac{1}{2}(\tilde{k}^3+5\tilde{k}^2+6\tilde{k})\left(u^n_{i+3}-9u^n_{i+2}+45u^n_{i+1}-45u^{n-\tilde{k}-1}_{i-1}+9u^{n-\tilde{k}-1}_{i-2}-u^{n-\tilde{k}-1}_{i+3}\right)/60\Delta x$ $+\frac{1}{2}(\tilde{k}^3+4\tilde{k}^2+3\tilde{k})\left(u^n_{i+3}-9u^n_{i+2}+45u^n_{i+1}-45u^{n-\tilde{k}-2}_{i-1}+9u^{n-\tilde{k}-2}_{i-2}-u^{n-\tilde{k}-2}_{i+3}\right)/60\Delta x$ $-\frac{1}{6}(\tilde{k}^3+3\tilde{k}^2+2\tilde{k})\left(u^n_{i+3}-9u^n_{i+2}+45u^n_{i+1}-45u^{n-\tilde{k}-3}_{i-1}+9u^{n-\tilde{k}-3}_{i-2}-u^{n-\tilde{k}-3}_{i+3}\right)/60\Delta x$
(1,6)	Right	$\frac{1}{6}(\tilde{k}^3+6\tilde{k}^2+11\tilde{k}+6)\left(u^{n-\tilde{k}}_{i+3}-9u^{n-\tilde{k}}_{i+2}+45u^{n-\tilde{k}}_{i+1}-45u^n_{i-1}+9u^n_{i-2}-u^n_{i-3}\right)/60\Delta x$ $-\frac{1}{2}(\tilde{k}^3+5\tilde{k}^2+6\tilde{k})\left(u^{n-\tilde{k}-1}_{i+3}-9u^{n-\tilde{k}-1}_{i+2}+45u^{n-\tilde{k}-1}_{i+1}-45u^n_{i-1}+9u^n_{i-2}-u^n_{i-3}\right)/600\Delta x$ $+\frac{1}{2}(\tilde{k}^3+4\tilde{k}^2+3\tilde{k})\left(u^{n-\tilde{k}-2}_{i+3}-9u^{n-\tilde{k}-2}_{i+2}+45u^{n-\tilde{k}-2}_{i+1}-45u^n_{i-1}+9u^n_{i-2}-u^n_{i-3}\right)/60\Delta x$ $-\frac{1}{6}(\tilde{k}^3+3\tilde{k}^2+2\tilde{k})\left(u^{n-\tilde{k}-3}_{i+3}-9u^{n-\tilde{k}-3}_{i+2}+45u^{n-\tilde{k}-3}_{i+1}-45u^n_{i-1}+9u^n_{i-2}-u^n_{i-3}\right)/60\Delta x$

Table A.2: Sixth-order asynchrony-tolerant (AT) schemes for left and right boundary used in numerical simulations for first and second derivative.