Analysis of time-domain integration methods for the simulation of thermal convection in an annulus

Venkatesh Gopinath

UNIVERSITÉ SORBONNE PARIS CITÉ

Thèse de doctorat de l'Université Sorbonne Paris Cité
Préparée à l'Université Paris Diderot
École Doctorale STEP'UP – ED 560
Institut de Physique du Globe de Paris – Équipe de Géomagnétisme

Analysis of time-domain integration methods for the simulation of thermal convection in an annulus

par

Venkatesh Gopinath

présentée et soutenue publiquement le
15 Juillet 2019

Thèse de doctorat de Sciences de la Terre et de l'environment

dirigée par **Alexandre Fournier**
et par **Thomas Gastine**

devant un jury composé de

Nathanaël Schaeffer Chercheur CNRS (ISTerre)	Rapporteur
Philip Livermore Associate Professor (Leeds University)	Rapporteur
Laurène Jouve Maître de conférences (Université Toulouse III)	Membre
Jean-Pierre Vilotte Professeur (IPGP)	Président du jury
Thomas Gastine CR-CNRS (IPGP)	Co-encadrant
Alexandre Fournier Professeur (IPGP)	Directeur de thèse

UNIVERSITÉ SORBONNE PARIS CITÉ

USPC
Université Sorbonne
Paris Cité

Analysis of time-domain integration methods for the simulation of thermal convection in an annulus

Venkatesh Gopinath

Résumé

Les simulations numériques de la convection thermique dans le noyau externe de la Terre sont un outil essentiel dans la compréhension de la dynamique à l'origine du champ magnétique terrestre. Des stratégies efficaces de résolution du système d'équations gouvernant cette dernière sont particulièrement intéressantes pour la communauté de recherche sur la Terre profonde qui tente actuellement de faire opérer les simulations numérique de géodynamo à des paramètres terrestres. Il existe de nombreuses pistes d'améliorations des modèles numériques. Dans cette thèse, nous avons axer notre recherche sur les techniques d'intégration en temps afin d'étendre la gamme des paramètres. Nous résolvons un problème de convection thermique dans un anneau en 2 dimensions en appliquant une méthode pseudospectrale pour la discrétisation spatiale. Concernant la discrétisation temporelle, le système d'équations comprend à la fois la composante raide (diffusive) et la composante non-raide (advective). Une pratique courante est de traiter la partie diffusive de façon implicite et la partie advective de façon explicite afin de limiter la restriction du pas de temps qui se produit lorsque nous utilisons une méthode purement explicite. Ceux-ci sont connus sous le nom d'intégrateurs en temps IMEX. Nous nous concentrons sur ces méthodes IMEX et analysons leur performance lorsqu'elles sont appliquées à ce problème. Nous considérons deux familles de méthodes IMEX, les méthodes *multistep* et les méthodes *multistage* IMEX Runge-Kutta (IMEX-RK). Nous réalisons une analyse systématique des paramètres d'entrée, à savoir le nombre de Rayleigh (Ra) et le nombre de Prandtl (Pr), qui contrôlent respectivement le forçage thermique et le rapport entre viscosité cinématique et diffusivité thermique. Notre intérêt se porte sur les régimes d'écoulement fortement non-linéaires et nous observons que, lorsque le nombre de Reynolds augmente, peu de méthodes IMEX-RK fonctionnent mieux que les méthodes *multistep*. Plus précisément, nous comparons les performances des méthodes IMEX-RK avec la méthode de second ordre Crank-Nicholson et Adams-Bashforth (CNAB2), largement utilisées par la communauté de recherche en géodynamo. Nous trouvons que certaines des méthodes d'ordre supérieur fonctionnent mieux que CNAB2 pour de grands nombres de Reynolds. Ce résultat ouvre la possibilité d'utiliser de telles méthodes pour les calculs de dynamo 3D. Cependant, dans la plupart des autres cas, les méthodes *multistep* d'un ordre donné sont plus performantes ques les méthodes IMEX-RK du même ordre.

Mots-clés: géodynamo, simulations numériques, convection thermique, intégration temporelle, méthodes IMEX

Abstract

Numerical simulations of outer core thermal convection of the Earth have been an essential tool in understanding the dynamics of magnetic field generation which surrounds the Earth. Efficient numerical strategies to solve this system of governing equations are of interest in the community of deep Earth research because, current numerical geodynamo models are on the quest to operate at the actual parameters of the Earth. There are many avenues for the improvement of the numerical model. In this thesis, we focus on the time domain integration techniques for solving such problems so that we may push the parameter boundaries further. We solve for a thermal convection problem in a 2D annulus. We use a pseudospectral method for spatial discretization. With respect to the time discretization part, the governing equations contain both numerically stiff (diffusive) and non-stiff (advective) components. A common practice is to treat the diffusive part implicitly and the advective part explicitly so as to alleviate the timestep restriction which happens when we use a purely explicit method. These are known as the IMEX time integrators. We focus on these IMEX methods and analyze their performance when applied to this problem. We consider two families of IMEX methods, the multistep methods and the multistage IMEX Runge-Kutta methods (IMEX-RK). We do a systematic survey of input parameters namely the Rayleigh number (Ra) and the Prandtl number (Pr), which control the thermal forcing and the ratio of momentum to thermal diffusivities respectively. Our focus is on the strongly nonlinear flow regimes and we observe that, as the Reynolds number increases, few of the IMEX-RK methods perform better than multistep methods. Specifically, we compare the performances of the IMEX-RK methods with the second order Crank-Nicholson and Adams-Bashforth (CNAB2) method, which is widely used in the geodynamo community. We find some of the higher order methods to perform better than CNAB2 for large Reynolds numbers. This result opens up the possibility of utilizing such higher order methods for the full 3D dynamo calculations. However, in most other cases, multistep methods of a given order outperform IMEX-RK methods of the same order.

Keywords: geodynamo, numerical simulations, thermal convection, time integration, IMEX methods

Acknowledgement

Over the last three years, it has been a great experience to meet fantastic people here in Paris and especially at IPGP. Their positivity has always influenced me in getting the work done during the course of this PhD. I thank god for giving me this opportunity.

First of all, I will be forever indebted to my thesis advisors Alexandre Fournier and Thomas Gastine. Without their guidance and helpful nature, this thesis would not have materialized. They were extremely patient with me throughout my PhD duration. During my initial days, when they asked me to write the code in a modular fashion, I remember after I finished writing the code from scratch in Fortran in a 'single program' and I told Alex "I completed writing the code, so I will make the 'cosmetic' changes i.e. all the modules over the weekend". He was silent for some seconds and had a poker face and I was wondering "did I say something wrong ?". I realized the meaning of the silence and my naivety after six months when I finished writing the first version of all the modules and subroutines. It was one of the most important parts of the work as it made the code modular and easy to correct. Both Alex and Thomas had a great quality control on the project. While I was showing my code and the results for corrections, passing every Gastine's constraint (a stringent quality control parameter) was something to cheer on that particular day. It has been a thoroughly rewarding process through out the duration of the PhD. The mentorship of Alex and Thomas was fantastic throughout the PhD and they always made time for me in their busy schedule and made me feel at ease while talking to them. Both Alex and Thomas are role models for scientists and advisors. I hope to strive in the future to live up to their standards of doing high quality work.

I would thank the jury members of my thesis defence, Jean-Pierre Vilotte, Nathanaël Schaeffer, Phil Livermore and Laurène Jouve for reading my thesis and for having asked important questions during my defence which enabled a fruitful discussion and closure on some of the important results.

I would also like to thank my "Comité de thèse" members composed of Jean-Pierre Vilotte, Nathanaël Schaeffer and Ludovic Metevier. We met at the end of each academic year during the PhD. They were very helpful with their comments. Their questions and answers with respect to the technicality and status of the thesis was quite important to resolve the problems encountered in the thesis. Their clarity of direction was one of the reasons this thesis was completed. For that, I am grateful to them.

I am grateful to Binod Sreenivasan of Indian Institute of Science (IISc), Bangalore for giving me an opportunity to work with him. Without the work done under his supervision, I would not have gotten this PhD position.

I thank all my labmates in geomagnetism team, Marie, Martin, Stanislava, Tatiana, Delphine, Guillaume, Tobias, Jenny, Marina (from volcanic systems), Rafael, Sabrina and Boris for their kindness and accommodating nature. During the early days, Tatiana, Marina and Boris were quite helpful in telling me important information about Paris. Along the course of PhD all of them had made my stay a pleasant one in IPGP. I specially thank Marie and Martin for their great company and patient French lessons sometimes during lunch. There was always laughter and camaraderie among us and they never made me feel like an outsider. I thank fellow metal-heads Stanislava and Pierre (from Siesmology) for discussions on music and Guillaume for finely detailed discussions on Starwars. I will

never forget the day when the Runge-Kutta time schemes worked using the Johnston's fix for the vorticity at the boundaries. Trivial as the work may be, it was an infamous evening on that day at the Les Arene bar. I spent the next two years being made fun of by the great beer tenders in the lab when ever the name "Les Arene" came up in a conversation.

I also thank all my Indian friends in IPGP, Pranav, Venkata, Shipra, Gaurav and Satish for their company and cheerfulness. Over the three years of my stay at Cite Universitaire, I am grateful to Yashas, Gaurav, Shreyas, Devi, Akhila, Inaya, Amit, Madhukar, Abhijeet and Omkar from Maison de l'Inde. They made my stay a memorable one at Maison de l'Inde. Discovering new heavy metal music with Yashas was quite the experience. I thank him for his good friendship, wit and endless supply of coronas on Friday evenings. I also thank Devi for her friendship, company and for imparting good culinary skills to me. During the writing process of the thesis, talking to Abhijeet and Madhukar helped take some pressure off because of our discussions on politics and science on weekend evenings which was a much needed distraction. I also thank fellow Chennaiite, Kannabiran for his company and good friendship, especially in the first year where we set off to Gare du Nord frequently to have Tamil cuisine. I am grateful to him for his time and discussions on basic fluid mechanics.

Outside of Paris, I am grateful to my friends and previous labmates, Ghanesh, Swarandeep and Subhajit from my IISc years before coming to Paris. Our constant technical discussions during the IISc years inspired me to apply for this PhD position. Their friendship and insights during those years are invaluable to me. I also thank my friends from high school and college, Santhosh, Maruthi, Sriram, Raghavendran and Srinivasan for their support at different stages of my life even though we were split among different countries. Although we interacted very less during the course of this PhD, they understood and were silently wishing well for me.

I am also thankful to my mentors in during master studies, Antonio Gil and Paul Ledger for imparting their technical knowledge of numerical methods and scientific computing. It will always stick with me.

I am forever grateful to Professor Shevare of Indian Institute of Technology, Bombay (IITB) for his great guidance and mentorship during my years working in his company, Zeus Numerix (ZN). I also thank Sandeep, Praveen, Basant, Irshad and Abhishek from ZN for their friendship, guidance and priceless insights to life. Working with them armed me with some of the skills that were required during this PhD.

Through out the PhD, there had been ups and downs. There has been moments of panic as well as great moments of jumping and fist pumping alone even for trivial successes. During these times, music was a constant support to me. So, I dedicate this thesis to fallen musicians, Layne Stayley, Paul Gray, Chris Cornell and Chester Bennington.

Most importantly, I would not have been able to do this PhD without the love and support of my mother, father, my sister and my girlfriend. They are and always will be my core source of strength and stability in my life.

Thank you, everyone.

Contents

Chapter 1

Introduction

1.1 State of the art 3D dynamo models

The geomagnetic field of the Earth is thought to be generated by the motion of an electrically conducting fluid inside the outer core. It is shown as a schematic in plot (a) of Fig. 1.1. In order to maintain such magnetic fields over vast time scales, it is theorized that a mechanism operating in the Earth's core continuously generates the geomagnetic field. This mechanism is called the geodynamo. The first theory of a dynamo model was proposed in the year 1946 by Walter Elsasser (Elsasser, 1946), where, he mentions that the convection in the outer core would sustain the magnetic field via electromagnetic induction which compensates the Ohmic dissipation. It is believed that the dynamo process consists of a rotating, convecting, and electrically conductive fluid motion inside the Earth's outer core. Such convective motions can either be driven by thermal convection or by compositional convection (Jones, 2015, Chapter 8.05). Thermal convection happens due to temperature differences between the inner core boundary and the core mantle boundary. Compositional convection on the other hand happens due to rising light elements such as sulfur from the inner core as it crystallizes. In order to understand the underlying physics of such processes, a mathematical model is required which describes the governing dynamics of the geodynamo. Such models are called dynamo models. We concern ourselves with dynamo models based on such convective phenomena.

The use of such numerical simulations has been an essential tool to unravel the fundamental nature of the geodynamo and it remains to be so for the foreseeable future since such phenomena cannot be directly observed. A 3D dynamo model often involves simulating the convective flow of an electrically conducting fluid in a spherical shell domain which mimics the outer core of the Earth. The dynamo model is successful if it produces a self-sustaining magnetic field for the given input parameters. The first numerical study on a 3D solar dynamo was done by Glatzmaier and Gilman (1981). A decade after that, the first self-consistent 3D dynamo model of the Earth's geodynamo was done by Glatzmaier and Roberts (1995). It was called the Glatzmaier-Roberts model. The magnetic field lines from their simulation are shown in plot (b) of Fig. 1.1. It is a dominant dipolar

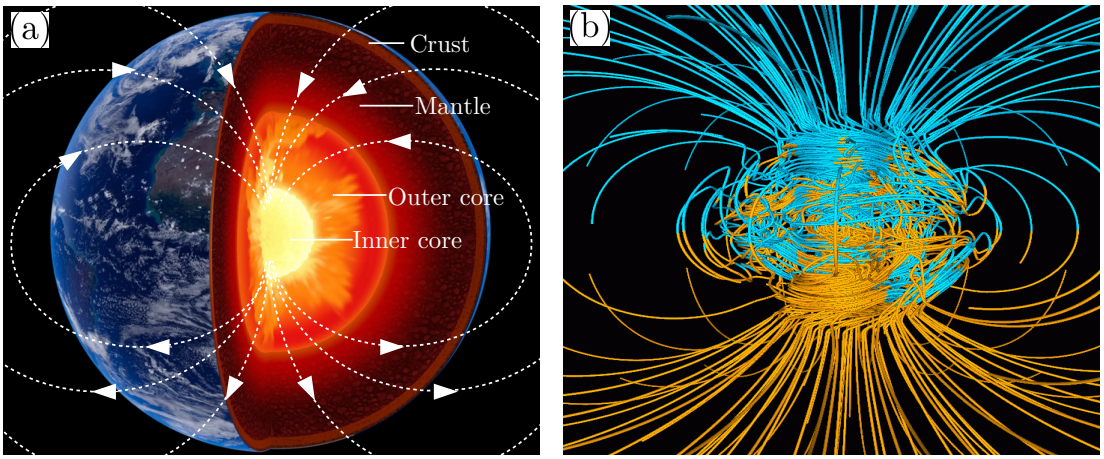

Figure 1.1: (a) An artistic description of the cut section of the Earth with the respective labels of the four major regions namely the crust, mantle, outer core and inner core. The white dashed lines indicate magnetic field lines and the white arrows indicate the direction of the field. (b) Magnetic field lines from the first self-consistent 3D numerical geodynamo model by Glatzmaier and Roberts (1995). Blue lines show inward directed field and yellow lines show outward directed field.

field. Since this ground breaking simulation, there has been a significant development in High Performance Computing (HPC) systems and numerical algorithms. A dynamo model consists of a set of Partial Differential Equations (PDEs) which are to be solved by the use of numerical methods and computer simulations. The basic geometry of a dynamo operating in a rotating spherical shell is shown in Fig. 1.2. This set of PDEs consists of the magnetic induction equation, the Navier-Stokes equation and the transport equation for heat or composition (Christensen and Wicht, 2015; Schaeffer et al., 2017). At this juncture, it is logical to introduce the governing equations for a dynamo model because it will help us to base our further discussion on the numerical aspects of the dynamo models. We introduce the variables in the governing equations. $\mathbf{u}$ is the fluid velocity field, T is the temperature and $\mathbf{B}$ is the magnetic field. Most dynamo models solve for the dimensionless equations. Therefore, one would need to choose a way to scale the equations. Following, (Christensen and Wicht, 2015, p. 247), we use the set up and definitions of the benchmark dynamo for illustration. The fundamental length scale is the thickness $D = r_o - r_i$ of the spherical shell, time t is scaled by the viscous diffusion time D^2/ν, where ν is the kinematic viscosity, temperature is scaled by ΔT and magnetic induction by $(\rho\mu\lambda\Omega)^{1/2}$, where ρ is the density, μ is the magnetic permeability, λ is the magnetic diffusivity and Ω is the rotation rate. This leads to the following set of dimensionless equations

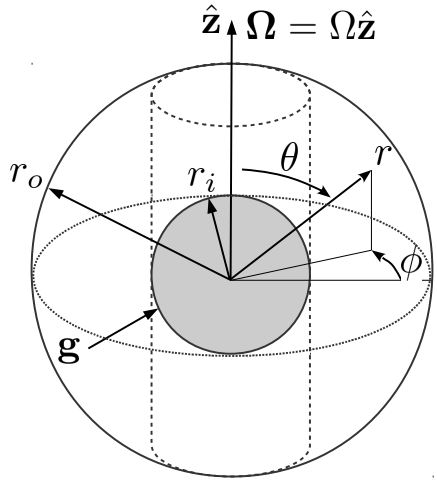

Figure 1.2: Basic geometry for a dynamo operating inside a rotating spherical shell with rotation rate Ω. Shaded region shows the inner core and the dotted cylinder is the inner core tangent cylinder.

$$E\left(\frac{\partial u}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) + 2\hat{\mathbf{z}}\times\mathbf{u} + \nabla\Pi = E\nabla^2\mathbf{u} + \frac{RaE}{Pr}\frac{\mathbf{r}}{r_o}T + \frac{1}{Pm}(\nabla\times\mathbf{B})\times\mathbf{B}, \tag{1.1}$$

$$\frac{\partial\mathbf{B}}{\partial t} - \nabla\times(\mathbf{u}\times\mathbf{B}) = \frac{1}{Pm}\nabla^2\mathbf{B}, \tag{1.2}$$

$$\frac{\partial T}{\partial t} + \mathbf{u}\cdot\nabla T = \frac{1}{Pr}\nabla^2 T + Q, \tag{1.3}$$

$$\nabla\cdot\mathbf{u} = 0, \tag{1.4}$$

$$\nabla\cdot\mathbf{B} = 0. \tag{1.5}$$

Equation (1.1) is the Navier-Stokes equation (detailed derivation in Chapter 2) with Lorentz force term ($\frac{1}{Pm}(\nabla\times\mathbf{B})\times\mathbf{B}$) and the Coriolis term $2\hat{\mathbf{z}}$, equation (1.2) is the induction equation, equation (1.3) is the temperature equation, equation (1.4) is the continuity equation and (1.5) is Gauss's law for magnetism. The unit vector $\hat{\mathbf{z}}$ indicates the direction of the rotation axis. Π is the non-hydrostatic pressure and Q is the volumetric heating. The four dimensionless control parameters are the Ekman number, $E = \nu/\Omega D^2$, the Rayleigh number, $Ra = g_o\alpha\Delta T D^3/\nu\kappa$, the Prandtl number, $Pr = \nu/\kappa$ and the magnetic Prandtl number, $Pm = \nu/\lambda$. g_o is the value of gravity present at the outer boundary, κ is the thermal diffusivity and α is the coefficient of thermal expansion. Also of importance is the magnetic interaction parameter or the Stuart number and the Rossby number. The Stuart number is given as $N = B^2 D\sigma/\rho U$, where, B is the magnetic field, σ is the electrical conductivity, U is the characteristic velocity scale. It measures the ratio of electromagnetic to inertial forces. It can also be given as $N = Ha^2/Re$, where $Ha = BD\sqrt{\frac{\sigma}{\mu}}$ is the Hartmann number, where μ is the dynamic viscosity and $Re = UD/\nu$ is the Reynolds number. The Rossby number is given as $Ro = u_{\text{rms}}D/\nu$, where, u_{rms} is the root mean square velocity. It measures the ratio of inertial to Coriolis forces. Another important dimensionless number to measure is the magnetic Reynolds number Rm which is the ratio of advection of the magnetic field to magnetic diffusion. This system of PDEs

is strongly nonlinear if the Stuart number is small and Rossby number is large which corresponds to large Reynolds number. For the Earth, $Re = 10^8$ which makes the system strongly nonlinear and turbulent. As for the boundary conditions, the models generally use a no-slip condition on the inner and outer boundaries for the velocities, insulating boundary condition at the outer core boundary and conducting boundary condition at the inner core boundary. Also, in the majority of the cases, a fixed temperature contrast is imposed between the inner and outer boundary, practically setting Q to zero. As mentioned in (Christensen and Wicht, 2015, p. 250), this condition is used for simplicity and there is no physical basis and results have been found to not so largely depend on the thermal boundary conditions. Thus, we seek to solve the governing equation with the associated boundary conditions using numerical methods and HPC. We will now discuss the present day modelling capabilities of various computer codes existing in the dynamo community as shown in Table 1.1.

The 3D dynamo models of the present generation (Matsui et al., 2016; Davies et al., 2011) involve numerical simulations which are highly parallelized along the spatial coordinates of the fluid domain. Such parallelism often involves the use of advanced HPC architectures. A full performance study of the modern 3D dynamo codes was done in Matsui et al. (2016). They compare and analyse the parallelization capabilities and the performances of most of the present day dynamo codes. For this study they fixed the dimensionless parameters to be $E = 10^{-3}$, $Ra = 10^5$, $Pr = 1$ and $Pm = 5$. According to (Christensen and Wicht, 2015, p.262) these parameters correspond to the magnetic Reynolds number to be $Rm = 39$ which is one of the lowest values for which a self-sustained dynamo has been found. Therefore, it is a nonlinear problem albeit with weakly driven convection. However, since the focus is on the performance comparison of various codes, they solve for weakly driven dynamo problem. With respect to parallelization capabilities, they perform a weak scaling and a strong scaling test. In a weak scaling test, the problem size increases at the same rate as the number of processors, maintaining constant amount of work per processor. It is based on the Gustafson's law (Gustafson, 1988) which states that speedup $= s_t + p_t \times N$, where s_t is the proportion of execution time spent on the serial part, p_t is the proportion of execution time spent on the parallel part and N is the number of processor cores. According to this speedup, the parallel part scales linearly with the amount of processors and there is no upper limit for the speedup. On the other hand, in a strong scaling test, the problem size remains the same as the number of processors increases. It is based on the Amdahl's law (Amdahl, 1967) which states that, speedup $= 1/(s_t + p_t/N)$. Therefore, the upper limit of the scaled speedup is determined by the serial part of the simulation. Therefore, strong scaling is usually more useful and more difficult to achieve. Matsui et al. (2016) gives a measure to quantify the deviations from ideal strong scaling as

$$\mathcal{E} = \frac{t_{\text{ref}}}{t} \frac{N_{\text{ref}}}{N_{\text{Core}}}, \tag{1.6}$$

where, $\mathcal{E}$ is the efficiency, t and N denote time and number of processor cores and 'ref'

Table 1.1: List of dynamo codes in the literature.

Code	Spatial method	Temporal method	Reference
MagIC	SH with CCM	CNAB2	Christensen (2001)
Rayleigh	SH with CCM	CNAB2	Featherstone and Hindman (2016)
SBS	SH with CCM	CNAB2	Simitev and Busse (2005)
SPmodel	SH with CCM	CNAB2	Sasaki et al. (2012)
USSC code	SH with CCM	CNAB2	Glatzmaier and Roberts (1995)
SpF-MoSST	SH with radial compact FDM	CN-Predictor-Corrector	Kuang and Bloxham (1999)
TITECH code	SH with radial compact FDM	CNAB3	Takahashi (2012)
Calypso	SH with radial FDM	CNAB2	Matsui et al. (2014)
ETH code	SH with radial FDM	Predictor-Corrector	Willis et al. (2007)
H2000	SH with radial FDM	CNAB2	Hollerbach (2000)
LSD code	SH with radial FDM	Predictor-Corrector	Davies et al. (2011)
PARODY-JA	SH with radial FDM	CNAB2	Landeau and Aubert (2011)
XSHELLS	SH with radial FDM	CNAB2	Schaeffer (2013)
GeoFEM	FEM	CNAB2	Matsui and Okuda (2004)
SFEMaNS	Fourier spectral method with FEM	SBDF2	Guermond et al. (2007)

Headings from left to right are the name of the code, spatial discretization method, time discretization method and the corresponding reference. The abbreviations SH indicates Spherical Harmonic Expansion, CCM indicates Chebyshev Collocation Method, FDM indicates Finite Difference Method, FEM indicates Finite Element Method, CNAB2 indicates Crank-Nicholson and Adams-Bashforth second order method, and SBDF2 indicates Semi-implicit Backward Difference Formula, a second order method.

denotes the reference. Fig. 1.3 shows the efficiency versus the number of processor cores for all the codes shown in Matsui et al. (2016) as well as in Table 1.1 except for the code H2000. We observe from the figure that efficiency for strong scaling is difficult to maintain as the number of processor cores increases beyond 10^3 and only codes which have 2-D or 3-D MPI parallelization (Message Passing Interface Forum, 1994) capabilities maintain high efficiency $\mathcal{E} > 0.6$ up to N_{Core} 10^4. To quantify this, they define a parallelization limit which is the number of processor cores N_{ref} where $\mathcal{E} = 0.6$. For this limit, the scaling capabilities of the codes that were used for the strong scaling test are shown in Fig. 1.4. The legends in the figures for e.g. 1-D, 2-D, 3-D indicate the number of directions which are parallelized with Message Passing Interface (MPI). Also, 1-D+OpenMP indicate the number of directions which are parallelized with MPI and OpenMP. From this figure we can observe that the codes (Rayleigh, LSD, and Calypso) have the capability to run efficiently up to $16,384$ processor cores and are estimated to scale efficiently up to 10^7 processor cores. Regarding the raw performance of the codes at a given core count, Matsui et al. (2016) perform a comparison of all the codes. The result for the strong scaling test is shown in Fig. 1.5. From the figure, it can be seen that, among the SH-FDM category, the XSHELLS code is the fastest and among the SH-CHY category, MagIC and Rayleigh are very close. At this point, it is interesting to note that, even with such good scaling, raw performance and the available advanced HPC resources, the present day simulations are not yet close to operating at the actual parameters and regimes of the Earth. These parameters for the Earth are shown in Table 1.2 which is adapted from Roberts and King (2013). An 'Earth-like' regime is a regime where we operate the simulations with parameters as close as possible to that of the Earth. We can see from Table 1.2 that, we

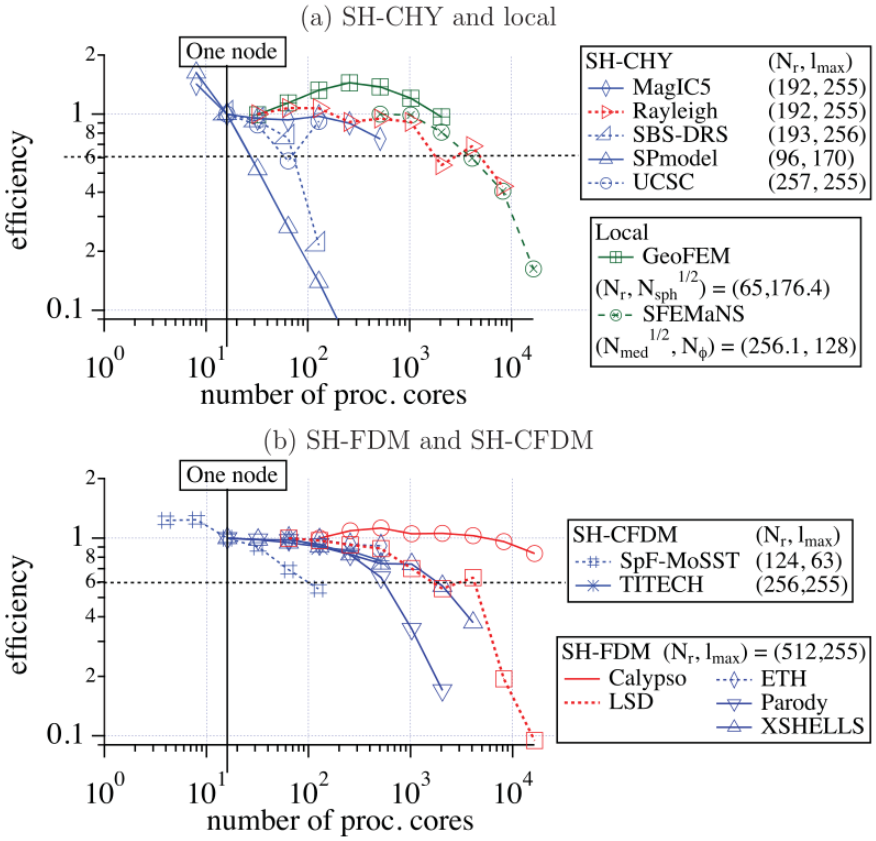

Figure 1.3: Parallel efficiency $\mathcal{E}$ as a function of the number of processor cores in the strong scaling. The efficiency is referred by the result using 16 processor cores. Results for codes with spatial discretization using spherical harmonic expansion in the angular directions and Chebyshev polynomials in the radial directions (SH-CHY) and local methods are shown in plot (a) and those with spherical harmonic-FDM (Finite Difference Method) and spherical harmonic-CFDM (Compact Finite Difference Method) are shown in plot (b). Codes with 1-D parallelization are plotted in blue, codes with 2-D parallelization are plotted in red and those with 3-D parallelization are plotted in green. Solid line is used for codes with hybrid parallelization (MPI and OpenMP) and a dotted line is used for the other codes. Also, $\mathcal{E} = 1$ means ideal scaling. From Matsui et al. (2016).

are not exactly performing our numerical simulations close to that of the parameters of the Earth. In fact, the most advanced simulation of the geodynamo to date was recently done

Table 1.2: Table for dimensionless control parameters for the Earth and the numerical simulations. Adapted from Roberts and King (2013).

Symbol	Dimensionless number	Definition	Earth	Numerical simulation
E	Ekman	$\nu/\Omega D^2$	10^{-15}	$10^{-3} - 10^{-7}$
Ra	Rayleigh	$g_o \alpha \Delta T D^3/\nu\kappa$	10^{24}	$10^5 - 10^{11}$
Pr	Prandtl	ν/κ	0.1	$0.1 - 30$
Pm	Magnetic Prandtl	ν/λ	10^{-6}	$0.01 - 20$
Re	Reynolds	UD/ν	10^8	$1 - 3 \times 10^3$
Rm	Magnetic Reynolds	UD/λ	10^3	$10^2 - 10^3$

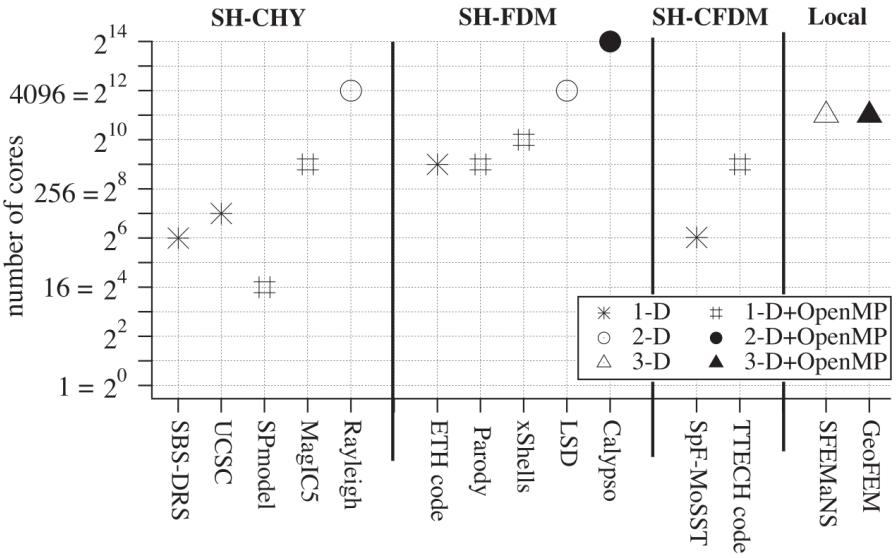

Figure 1.4: Maximum number of processor cores for the strong scaling to keep parallel efficiency $\mathcal{E} > 0.60$. The different symbols indicate the number of directions of MPI and OpenMP parallelization. From Matsui et al. (2016).

by Schaeffer et al. (2017). They performed their simulations at $E = 10^{-7}$ which is eight orders higher than the value of Earth's Ekman number. Therefore, there is a constant need to push towards these Earth-like regimes and since we are far from operating at these parameters, these regimes can also be coined as 'extreme parameter regimes'. With the existing parallelization in space, one of the possible avenues for exploring such extreme parameter regimes is to make use of a more optimally efficient time discretization method and it is particularly attractive to make use of a parallel-in-time discretization.

Although not part of the work done in this thesis, we will now discuss the widely used parallel-in-time method known as the parareal method created by the seminal work of Lions et al. (2001). We discuss it because we will eventually show the relevance of the findings from this thesis in relation to parareal methods. The parareal method is based on a predictor-corrector approach. The schematic for a parareal parallel-in-time method is shown in Fig. 1.6. Following Samaddar et al. (2010) and Samuel (2011), let us define a single Ordinary Differential Equation (ODE) of the type

$$\frac{du}{dt} = A(u,t),\ u(0) = u_0, \tag{1.7}$$

where, A is an arbitrary function. We thus require to find the solution u at time $T > 0$. After discretizing in time we can advance this equation in time as

$$u(t + \Delta t) = \mathcal{A}_{\Delta t} \cdot u(t), \tag{1.8}$$

where, $\mathcal{A}$ is our advancing operator. The parareal algorithm dictates two such advancing operators namely the fine operator $\mathcal{F}$ and the coarse operator $\mathcal{C}$. The fine operator is in general more accurate and has smaller timestep sizes than the coarse operator, making

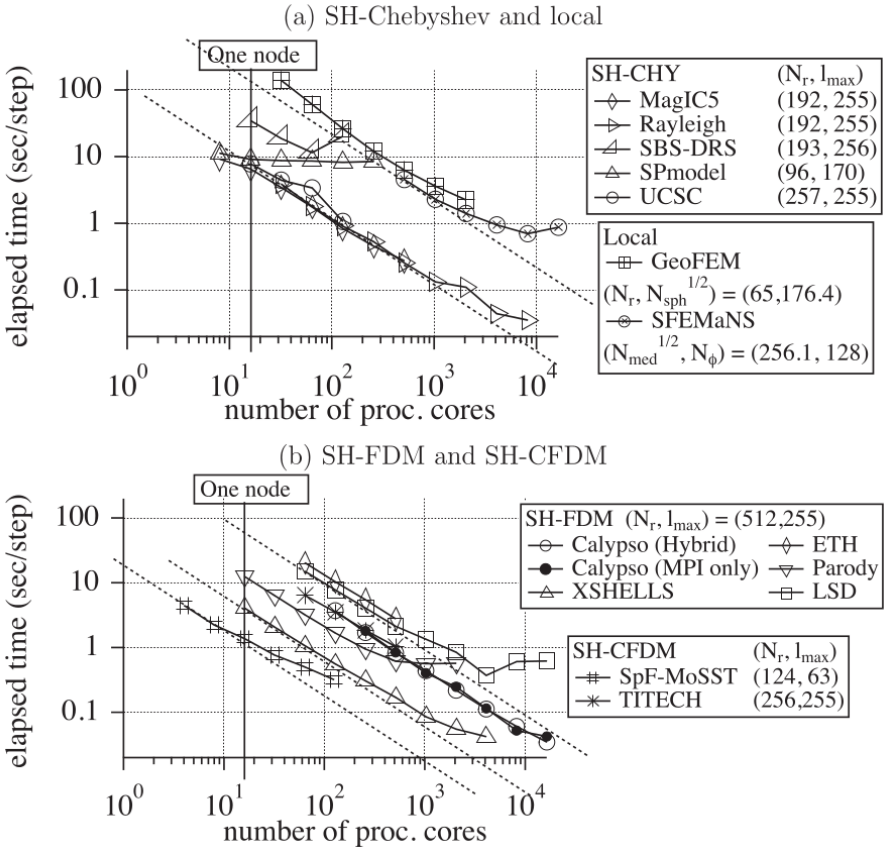

Figure 1.5: Elapsed time for each time step in the strong scaling as a function of the number of processor cores. The ideal scaling ($t_{elapsed} \propto N_{core}^{-1}$) for SFEMaNS, Rayleigh, LSD code, XSHELLS and SpF-MoSST is plotted by dotted lines. Results of Calypso using hybrid (MPI and OpenMP) parallelization and MPI parallelization are plotted by open and solid circles. From Matsui et al. (2016).

it more expensive to compute. $\mathcal{C}$ is always run serially and $\mathcal{F}$ is always run in parallel by splitting it in to several chunks of time. Let N represent the number of processors and the total simulation time T be divided into N smaller chunks of size $\Delta t = T/N$. Let $t_n = n\Delta t$ for $n = 0, 1, 2, \cdots, N$ and the index $k = 0, 1, 2, \cdots$ represent the parareal iteration number. Then, u_n^k represents the solution at time t_n at the k^{th} parareal iteration. The initial value u_0 is now u_0^0. The parareal steps are shown as follows

- At $k = 0$: Use the $\mathcal{C}$ operator to calculate u_n^0 for the start of every time chunk t_n. This step is indicated by the orange line in Fig. 1.6.
- At $k > 0$:

  Step 1: Each processor propagates the solution in time using the initial value u_n^{k-1}. This is where the time chunks are run in parallel. This step is indicated by the blue line in Fig. 1.6.

  Step 2: Update initial value at each time chunk as

$$u_{n+1}^{k+1} = \mathcal{C}_{\Delta t}(u_n^{k+1}) + \mathcal{F}_{\Delta t}(u_n^k) - \mathcal{C}_{\Delta t}(u_n^k). \tag{1.9}$$

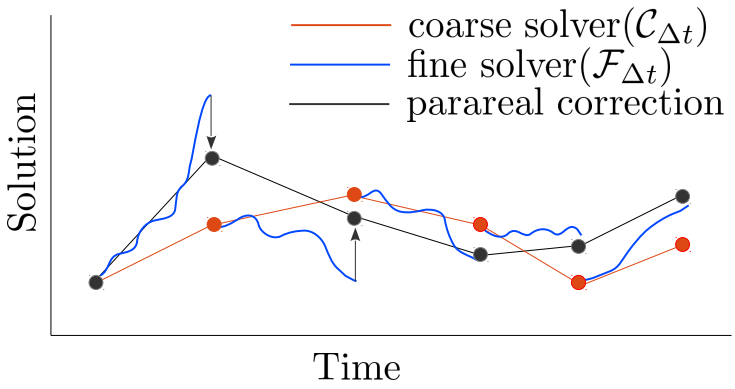

Figure 1.6: Schematic of the parareal parallel-in-time method. Initial approximation to the solution of an initial-value problem on the coarse time grid is shown in orange. The blue line indicates the parallel-in-time fine solver which is started from the coarse initial value. The black line indicates the updated solution after a parareal iteration. Adapted from Haut and Wingate (2014).

This step is done serially because the first term on the right hand side has to be done serially. This step is indicated by the black line in Fig. 1.6.

Step 3: Check for convergence in all the time chunks and exit the parareal iteration if converged.

Next, the choice of $\mathcal{C}$ and $\mathcal{F}$ is decided based on the requirement that the iterations $k <$ number of time chunks N in order to obtain speedup. From Samuel (2011), a parareal speedup is defined as

$$S = \frac{\tau_{\text{serial}}}{\tau_{\text{parareal}}}, \tag{1.10}$$

where,

$$\tau_{\text{parareal}} \approx N\tau_C + k(\tau_f + N\tau_c), \tag{1.11}$$

where, τ_c and τ_f are the computational time associated to the use of coarse and fine operators respectively. Also,

$$\tau_{\text{serial}} \approx N\tau_f. \tag{1.12}$$

Therefore, the parareal speedup is defined as

$$S = \frac{N}{N\beta + k(1 + N\beta)}, \tag{1.13}$$

where, $\beta = \tau_c/\tau_f$.

The Navier-Stokes equations are an integral part of the dynamo models as shown in equations (1.1). The use of parallel in time methods such as parareal (Lions et al., 2001) was applied to Navier-Stokes equations has been explored in Fisher et al. (2003), Samaddar et al. (2010), Hupp et al. (2016) and many others. One of the few application of parallel-in-time approach to solid Earth geosciences was done by Samuel (2011), where,

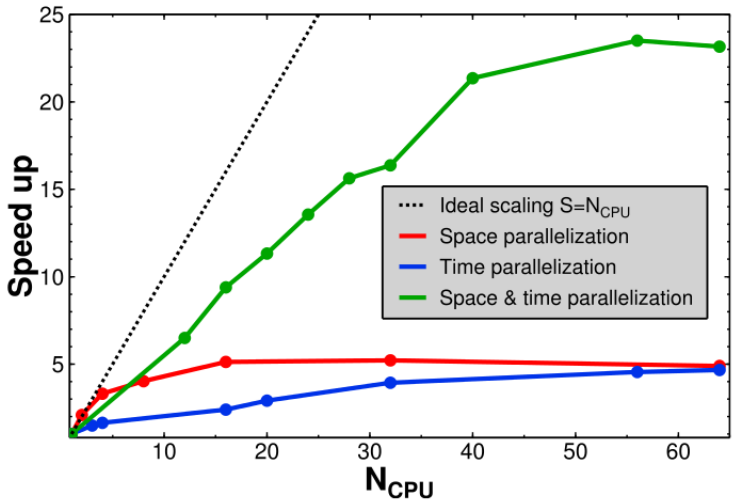

Figure 1.7: Result of the convection simulation test. Parallel speedup as a function of the number of processor cores. From Samuel (2011).

they perform numerical simulations of mantle convection. They solve for the continuity equation, momentum equation and the energy equation. However, they assume infinite Prandtl number and this makes the explicit time dependence appearing only in the energy equation. In such a case, temporal dependence for the velocity field is only due to the coupling of the Navier-Stokes equations and the energy equations via the buoyancy term and the viscosity which in their case, is a function of temperature. They report a good speedup of their solution run times for their space-time parallelization as shown in Fig. 1.7. They perform the simulation in a box of grid size 512×256. From the Fig. 1.7, we can observe that the speedup achieved with space-time parallelization is five times better than the simulation performed with only the space or time parallelization for $\mathrm{N_{CPU}} > 30$. Another recent study was done by Clarke et al. (2019) where they perform a parallel-in-time analysis applied to the kinematic dynamo problem. A kinematic dynamo problem consists of solely solving for the induction equation with a prescribed velocity as shown in equation (1.2). They solve with both constant and time-varying velocity fields which are called Roberts flow and Galloway-Proctor flow respectively. They define their parallel speedup as

$$s \leq \min\left\{\frac{N_p}{k_{con}}, \frac{R_f}{R_c}\right\}, \tag{1.14}$$

where, N_p is the number of processors, k_{con} is the number of parareal iterations to reach convergence, R_f is the runtime of the fine method and R_c is the run time of the coarse method. They hence defined the parallel efficiency to be $\epsilon = s/N_p$. They found that parallel efficiency did not suffer as Rm increases and in the case of $Rm = 3000$ which is a highly advective case, they find the parallel efficiency to be holding. Their speedup and parallel efficiency are shown in Fig. 1.8 for $Rm = 3000$. From this figure, we observe that the scaling saturation has not even reached saturation for the parareal approach and that the parallel efficiency is maintained closer to the theoretical maximum of $1/k_{con}$.

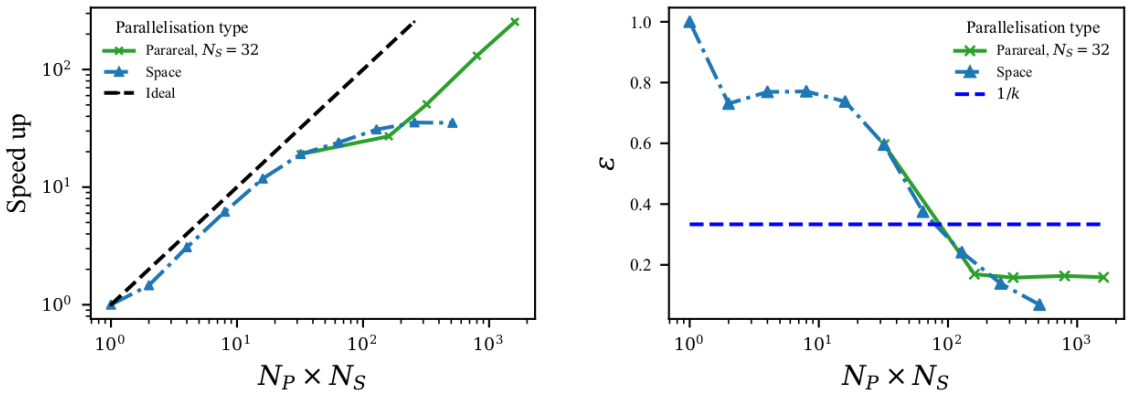

Figure 1.8: Speedup and parallel efficiency of the parareal method compared to parallelization in space for $Rm = 3000$. Total number of processors is calculated as number of processors in space (N_S) multiplied by the number of processors for Parareal (N_P). From Clarke et al. (2019).

Such parallel-in-time approaches are warranted to be used in a 3D dynamo model because all the existing models make use of parallelization in space and they run sequentially in time. In particular, when we want to explore possible extreme parameter regimes with the available computing resources, we would require a higher spatial resolution and a small timestep size. The reason for the small timestep size is due the stability constraints which will be briefly touched upon in the next section. With respect to spatial parallelization, even if the dynamo model scales well with the available number processor cores, it still has to run sequentially in time with small timestep sizes and thus it becomes an obstacle to overall efficiency of the simulation. Thus, a parallel-in-time approach to the dynamo models is a possible way to explore extreme parameter regimes closer to the Earth as it increases the efficiency of the simulations by having a combined space-time parallelization. The outcomes from this thesis could be used in part to construct efficient methods for parallel-in-time approaches applied to the dynamo models and it will be discussed in the final chapter.

1.2 Survey of time integration methods

The study of efficient time integration methods for numerical simulation is the main theme of this thesis. Along with this theme, we are also interested in strongly nonlinear flow regimes as we want to push towards solutions having input parameters closer to the Earth's values. We will talk about the possible avenues towards such regimes in the subsequent section.

Mathematically, the dynamo models consist of advection-diffusion type equations. For spatial discretization, a majority of the current dynamo codes use a spherical harmonic expansion along the angular directions and a Chebyshev expansion or a finite difference method along the radius. Such spectral discretization in space was first applied to a

dynamo problem by Bullard and Gellman (1954). For discretizing the equations in time, it can either be a fully implicit method or a fully explicit method or a semi-implicit method. For a more detailed explanation of such methods, the reader is referred to chapter 3.

Fully implicit methods offer increased stability and allow for larger timesteps, but they do have a caveat that the nonlinear terms couple all the modes in spectral space (Christensen and Wicht, 2015, Chapter 8.10). Therefore, for a strongly nonlinear problem, either a fully explicit or a semi-implicit or Implicit-Explicit (IMEX) method (Ascher et al., 1995, 1997) is generally used. The Navier-Stokes equation, which is an important part of the dynamo model comprises an advective or nonlinear component and a diffusive or linear component. To briefly illustrate the advantages of an IMEX method over a fully explicit strategy, let us consider a scalar advection-diffusion equation in one dimension x of the form

$$\frac{\partial u}{\partial t} = u\frac{\partial u}{\partial x} + \nu\frac{\partial^2 u}{\partial x^2}, \tag{1.15}$$

where, $u(x)$ is the required solution, ν is the kinematic viscosity, $u\frac{\partial u}{\partial x}$ is the nonlinear term and $\nu\frac{\partial^2 u}{\partial x^2}$ is the linear term. The stability conditions for the above equation depends on whether we treat the time discretization explicitly or semi-implicitly can be given as

$$\Delta t_{\text{explicit}} = \alpha \ \min\left[\min\left(\frac{\Delta x}{|u|_{\max}}\right), \min\left(\frac{\Delta x^2}{\nu}\right)\right], \tag{1.16}$$

$$\Delta t_{\text{IMEX}} = \beta \ \min\left(\frac{\Delta x}{|u|_{\max}}\right), \tag{1.17}$$

where, α and β are known as the Courant-Friedrichs-Lewy (CFL) numbers (Courant et al., 1928) and Δx is an element of the spatial grid span. From the above equations, it is clear that if we use a fully explicit method in time, the timestep size is restricted to the square of the smallest grid size. Since we are interested in strongly nonlinear regimes with large Reynolds numbers, we would be using large grid sizes. Even though there are studies which use explicit time schemes in strongly nonlinear regimes such as in Johnston and Doering (2009), the large grid sizes which are required would imply that Δx will be smaller and thus the IMEX timestep size will be larger than the explicit one. Hence, with respect to integrating these equations in time, an ideal way to treat them is to use the IMEX class of time integration methods, where, the diffusive part is handled implicitly in time and the nonlinear advective part is handled explicitly in time. More on why we use IMEX strategies will be discussed in chapter 3. Most of the current 3D dynamo models use this IMEX method to deal with the time integration. The present generation of dynamo codes given in Matsui et al. (2016) are shown in the Table 1.1 along with their spatial and time discretization methods and the corresponding reference for each code.

We observe from Table 1.1 that, a majority of the current dynamo codes uses a second order method for time integration, especially since the seminal work by Glatzmaier (1984),the CNAB2 method, which is an IMEX method, seems to be widely used.

1.2.1 Work done in the geophysical and astrophysical community

Here, we will briefly discuss the work done in the dynamo community regarding the exploration of time integration methods and its relevance with respect to the work done in this thesis.

- There are dynamo models in a rotating spherical shell geometry like in Livermore (2007), where they compare IMEX multistep methods with the exponential time integrators. Exponential time integration involves the exact integration of the linear components which helps reduce the stiffness. They report that, if the timestep is constrained by the stability of the nonlinear terms, the IMEX multistep method (CNAB2), is of same accuracy as the second order exponential methods and thus they prefer CNAB2 as the best choice as it is simpler to implement. They speculate that the higher order exponential methods will be more accurate than the higher order IMEX methods.
  - Although we do not study the exponential methods, we study many higher order methods with small and large nonlinear timescales in this thesis.
- A recent work of Garcia et al. (2014) compares the higher order exponential time integrators versus the higher order IMEX multistep methods when applied to a thermal convection problem in a rotating spherical shell. They perform their numerical simulations for different cases, each having a particular critical azimuthal wave number m_c and grid resolution, N_r and L, where, N_r is the number of radial grid values and L is the spherical harmonic truncation. They confirm the superior accuracy of the exponential methods for a strongly supercritical nonlinear problem with $Ra = 53.58Ra_c$ and $Pr = 0.1$, where $Ra_c = 1.86 \times 10^5$ is the critical Rayleigh number. For a range of Rayleigh numbers including the highest Rayleigh number they perform, they find the exponential methods being more than one order accurate than the IMEX multistep methods. However, they find that higher order exponential methods are more expensive than the IMEX multistep methods as indicated by plot b) of Fig. 1.9. In Fig. 1.9, the black line plots are from the IMEX multistep family and other colors are from the different exponential methods. They also perform runs with decreasing Ekman numbers where they once again find the exponential methods more accurate than the IMEX multistep methods. They also find that, for their lowest Ekman number of $E = 10^{-6}$, the higher order exponential methods start to show better performance than the higher order IMEX multistep methods. They speculate that the exponential methods could be a viable alternative to the IMEX multistep methods at Ekman numbers closer to the Earth's value.
  - We do not analyse the exponential methods because we pursue strongly nonlinear regimes as we perform simulations for strongly supercritical Rayleigh numbers and Garcia et al. (2014) does mention that the exponential methods are more expensive than the IMEX methods as shown in Fig. 1.9. We analyse a range of multistep and multistage methods. The exponential methods can be considered as for future extension of this thesis.

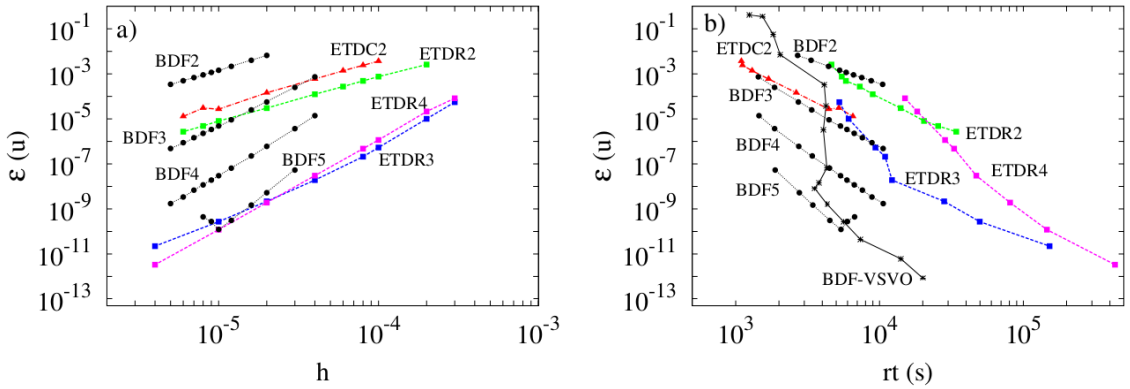

Figure 1.9: Plot a) shows the the relative error $\varepsilon(u)$ plotted versus the constant timestep size h for the exponential and the IMEX multistep methods. Plot b) shows the relative error versus the run time in seconds (rt) for the same methods shown in plot a). The input parameters are $E = 10^{-4}$ and $Ra = 1.42Ra_c$. The resolution is $N_r = 32$, $L = 54$ and $m_c = 6$. From Garcia et al. (2014).

- A recent work by Marti et al. (2016) uses an IMEX Runge-Kutta (IMEX-RK) (Ascher et al., 1997) method where a Diagonally implicit Runge-Kutta (DIRK) method is used for treating the diffusive component and an Explicit Runge-Kutta (ERK) method is used for treating the advective component. They use second, third and fourth order IMEX-RK methods from Cavaglieri and Bewley (2015). They perform stability and accuracy of the numerical time integration by evolving the solution for 0.1 diffusion times. Compared to the explicit implementation, for the purely hydrodynamic case ($E = 10^{-5},\ Ra = 1.5 \times 10^7 \sim 3Ra_c,\ Pr = 1$), they report that the second order method allows for a 16 times larger timestep size and fourth order method allows for a 4 times larger timestep size. For the dynamo case ($E = 10^{-5},\ Ra = 1.5 \times 10^7 \sim 3Ra_c,\ Pr = 1,\ Pm = 2$), they report that the second order method allows for a 12 times larger timestep size and fourth order method allows for a 3 times larger timestep size.
  - We study the third and fourth order schemes which Marti et al. (2016) tried and we also report their properties for a nonlinear regime.
- Grooms and Majda (2013) did quasigeostrophic turbulence simulations with hyper-diffusivity in a periodic two dimensional box geometry. The nonlinear timescales are small suggesting that the problem is not stiff. They use the KC564 time scheme (Kennedy and Carpenter, 2003) with adaptive stepsize control.
  - We also use this scheme in this thesis but we study a more realistic problem with non-periodic boundary walls and we also study both linear and nonlinear regimes.
- Grooms and Julien (2011) did a study where they analyse 24 third and fourth order IMEX and exponential methods applied to many numerical experiments including quasigeostrophic equations and stratified 2D Boussinesq equations. They solve the 2D Boussinesq equations in a periodic domain using a Fourier spectral discretization

in space with grid size of 256×256. They report a linear time scale of $\tau_L = 10^{-3}$ and a nonlinear time scale of $\tau_{NL} = 0.14$. The disparity in the values of these timescales makes the problem stiff and this measure of stiffness will be discussed in chapter 3 and chapter 4. Their solution is in a nearly linear regime for which they report that the exponential methods were more accurate than IMEX methods by a factor of 10^5. Also, they report that the IMEX Runge-Kutta methods were more accurate than the IMEX multistep methods. They also found the BR553 time scheme (Boscarino and Russo, 2009) to be super-convergent, meaning it performs better than its theoretical order of 3. Another important result from their study was the order reduction phenomena which they observe for certain IMEX Runge-Kutta time schemes.

  - As mentioned before, we study a more realistic problem with non-periodic boundary walls and we also study both strongly linear and nonlinear flow regimes. For our study, we also analyse both IMEX multistep and IMEX Runge-Kutta methods some of which that are used in Grooms and Julien (2011). We also observe the order reduction phenomena for certain IMEX-RK methods which we will discuss in chapter 4.

- Although CNAB2 became a widely used method for time integration in the dynamo community, around the time of the Glatzmaier and Roberts dynamo model, they also performed an anelastic dynamo simulation in Glatzmaier and Roberts (1996) using a third order semi-implicit Runge-Kutta time integration method which was introduced by Spalart et al. (1991). They mention that the method is computationally expensive even though its stability was better than a second order method.

  - We analyse many third order IMEX Runge-Kutta schemes and we analyse their stability properties as compared to the CNAB2 scheme.

- Stellmach and Hansen (2008) performed three dimensional Cartesian dynamo simulations exploring parallel computing capabilities. They use the SBDF2 scheme for time stepping. Also, Verhoeven et al. (2015) performed three dimensional Cartesian compressible convection for various Rayleigh numbers. They used the SBDF3 scheme for time stepping.

  - We also study the second, third and fourth order Backward Difference Formula (BDF) methods. Stellmach and Hansen (2008) also mention that in conjunction with local methods (FDM or FVM for e.g.), the fully implicit time steppers will be cheaper than IMEX methods for low Ekman numbers and strong magnetic fields (Harder and Hansen, 2005). This is because the Alvén wave velocity becomes large and start to limit timestep sizes if the nonlinear terms are treated explicitly. However, since most of the dynamo codes are using spectral methods in space, a fully implicit approach in time will be detrimental to the performance because the nonlinear terms couple all the modes in spectral space as this causes difficulty in proper implementation. Since we

will be considering spectral method for spatial discretization of our problem, we stick to the IMEX approaches in time.

- A fairly recent work by Gastine (2019) performs a spherical quasi-geostrophic convection with different time integration methods. They compare few IMEX Runge-Kutta methods with the CNAB2 method which we indicated to be widely used in the dynamo community. They analyse the parameters $E = 10^{-7}$, $Pr = 1$, and $Ra = 2 \times 10^{11}$, which indicate a strongly nonlinear problem. Performance wise, they indicate that the methods CNAB2 and ARS222 (IMEX-RK) are the best. They speculate whether there are other IMEX-RK methods that will perform better than the CNAB2 method for turbulent convection.
  - We do a similar performance study of the time integration methods in this thesis based on the stability analysis. We study with a larger pool of time schemes for a wide range of parameters for e.g. for $Ra = 10^4$ to $Ra = 10^9$ which cover both the linear and nonlinear flow regimes of thermal convection. We will analyse the low to high order IMEX-RK methods with respect to CNAB2 for performance comparison.

Now that we have looked into these different studies, we get a sense that although there are inroads made by many in exploring new time integration schemes in the dynamo community, a variety of attractive time integration methods in the literature still remain unused. Thus, from the work done in the dynamo community regarding the use of time integration methods, we have an inference and thus the main theme of the thesis:

- An assessment of different time integration strategies would help determine the methods that can improve the efficiency of present day dynamo models and in turn motivate us to explore new parameter regimes.

1.3 Avenues towards Earth-like regimes

Glatzmaier (2002) proposed the grand challenge to achieve the Earth-like geodynamo simulation. They proposed that the grand challenge is to run a 3D geodynamo simulation with the rotation rate, dimensions, density and heatflux of the Earth's core and with all diffusivities set to Earth's magnetic diffusivity 2 m^2/s. The Earth's values of the corresponding control parameters (E, Ra, Pr, Pm) are given in Table 1.2. To perform a simulation with Earth-like parameter regimes, we need to improve the numerical methods used to model the geodynamo equations. The 3D dynamo models in a spherical shell geometry generally use a pseudo-spectral method for spatial resolution. Usually, as shown in Table 1.1, a spherical harmonic expansion is used along the zonal and meridional directions and a Chebyshev method or a finite difference method is used along the radial direction. The use of such spectral methods generally results in exponential convergence (Orzag, 1980). Such convergence properties are reported in Fournier et al. (2005) where they use a Fourier-spectral element method for thermal convection in a spherical shell

Table 1.3: Time discretization errors for the rotating Gaussian cone problem for different 2D grid resolutions.

Resolution	Method	
	Crowley-4	RK3
050×050	0.522×10^{-1}	0.598×10^{-1}
100×100	0.490×10^{-2}	0.530×10^{-2}
200×200	0.306×10^{-3}	0.344×10^{-3}
400×400	0.173×10^{-4}	0.219×10^{-4}
800×800	0.173×10^{-5}	0.144×10^{-5}

Crowley-4 is a fourth order method and RK3 is a third order method. A fourth order finite difference spatial discretization has been used. Adapted from Wicker and Skamarock (2002).

geometry. However, the present day dynamo models calculate the nonlinear terms in the physical grid space and they are transformed back into spectral space using spherical harmonic transforms. For simulations with high resolution in space, these transforms become a hindrance to efficiency of the simulation (Glatzmaier, 2002; Christensen and Wicht, 2015). One way to deal with them is to improve the Legendre transforms or to use local methods such as finite difference or finite element methods instead of spectral methods. The local methods are attractive even if they do not have exponential convergence because they exhibit better spatial parallelization capabilities than the spectral methods. However, they will not be as accurate as spectral methods for the same spatial resolutions and the local methods require larger resolution to maintain comparable accuracy with spectral methods. Also, for the implementation of insulating boundary conditions, spectral methods offer an advantage as the derivatives are calculated with high accuracy and a larger number of grid points are required by the local methods to obtain comparable accuracy (Christensen and Wicht, 2015, p. 260). While pushing for the extreme parameter regimes, whether good spatial parallelization properties of the local methods will make them more efficient than the spectral methods is still speculated.

Apart from parallel-in-time methods, another avenue to improve the numerical model is to look at the time discretization part of the dynamo model which we focus in this thesis. As discussed in (Christensen and Wicht, 2015, Chapter 8.08) and in the previous subsection, most of the dynamo models have a second order IMEX multistep method or a predictor-corrector method.

Although some of the recent works in the literature Marti et al. (2016) started to use new time integration methods, the exploration of various higher order methods for dynamo models is still premature. The majority of the present day dynamo models use a higher order spatial discretization and mostly a second order time discretization. Such disparity of orders between discretization in space and time can affect the overall performance of the simulation. Such drop in efficiency has been shown before in Wicker and Skamarock

(2002) and Park and Lee (2009). Wicker and Skamarock (2002) perform tests based on higher order Runge-Kutta methods and Crowley advection schemes on linear and non-linear compressible Navier-Stokes equations. They clearly show that as the spatial resolution increases, higher order time integration methods are more accurate than the lower order ones as indicated in Table 1.3 which is adapted from Wicker and Skamarock (2002). Park and Lee (2009) on the other hand compare higher order explicit Runge-Kutta methods with Leap Frog (LF) methods for linear equations. They show that higher order time integration methods may not be needed when used alongside lower order spatial discretization. However, they also show that the the accuracy of the higher order time integration method is higher than a lower order one when used alongside a higher order spatial discretization method, as found by Wicker and Skamarock (2002). They also speculate on how well the higher order time integration methods perform on practical high resolution simulations. Thus a motivating question is borne out from these studies which will be addressed in this thesis:

- Whether the higher order time integration methods will be more efficient for the existing 3D dynamo models which are highly resolved in space ?

To answer this question, we need to conduct a study of variety of higher order time integration strategies. We concern ourselves with the IMEX methods and there are a wide variety of them described in the literature. These methods can be currently classed in to three categories namely, the IMEX multistep, IMEX-RK and the IMEX General Linear Methods (IMEX-GLM) which were introduced by Butcher and Jackiewicz (1993). In the recent work of Zhang et al. (2014), they applied the IMEX-GLM methods to 2D Burger's equations and 3D compressible Euler equations. They found the IMEX-GLM methods do not suffer any order reduction (drop in convergence rate) and found them to be much more efficient compared to the IMEX-RK methods. However, further analysis of practical simulations is needed to ascertain their advantages over IMEX-RK methods. This thesis does not study the newly developed IMEX-GLM methods. In our work, we analyze the second, third and fourth order IMEX multistep and IMEX-RK methods in the literature which include the recently developed methods by Cavaglieri and Bewley (2015), Boscarino and Russo (2009), Boscarino et al. (2013) and Boscarino et al. (2017). We analyse these two classes of time integration methods applied to a problem of thermal convection in a 2D annulus geometry. We chose this model for our analysis because it is faster to test new time integration schemes and at the same time the physics is still relevant when extended to a 3D spherical shell geometry.

1.4 Outline of the Thesis

The thesis is divided into five chapters. This introduction is followed by chapter 2 where we describe the physical problem which includes the mathematical description of the governing equations and the diagnostic measures which are used to check the solution correctness. Chapter 3 describes the numerical strategies that are used. First, the spatial discretization strategy is explained followed by the time discretization strategy. The governing equations are shown next in their semi-discrete form and then the numerical implementation strategies for these equations are shown in a matrix form. After that, the different properties of the time integration methods used are summarized. Subsequently, the validation of the code which was developed is shown. Chapter 4 deals with the results of the study. The solution physics is discussed first. After that, the accuracy and cost of all the time integration methods are shown. Next, the order reduction phenomena is discussed followed by a detailed comparison of the stability properties for all the methods. Finally, a summary of the results is shown in chapter 5 followed by a discussion on parallel-in-time methods.

Chapter 2

Thermal convection model

This chapter focusses on the description of the physical model of thermal convection in an annulus and the associated governing equations.

2.1 Physical setup

We consider Rayleigh-Bénard convection (Chandrasekhar (1981)) of a Boussinesq fluid contained in a cylindrical annulus geometry with outer radius r_o and inner radius r_i. It is shown as a schematic in Fig. 2.1. As mentioned before in the introduction, we reiterate that, physics of convection in an annulus is relevant enough to be considered as a test model for the application of efficient numerical strategies. The independent variables for this Eulerian representation (Batchelor (2000)) are time, t, and the spatial coordinates, r, ϕ and z. The acceleration due to gravity is assumed to be constant and it is given as, $\mathbf{g} = -g_o\hat{\mathbf{r}}$, directed radially inward. The inner and outer boundary walls are maintained at constant temperatures T_i and T_o respectively. The inner boundary temperature is higher than the outer boundary temperature by an amount $\Delta T = T_i - T_o$. We consider fluid flow only in two directions namely (r, ϕ). Therefore, the z-component of the fluid velocity vanishes.

2.2 Governing equations

In this section we will derive the governing equations that are used to model thermal convection in an annulus from basic conservation laws (Hirsch, 2007, Chapter 1). The conservation law for a quantity U can be stated as, "The variation of the total amount of quantity U in a given domain Ω is equal to the net amount of that quantity entering and leaving the domain plus the amount of that quantity generated by internal sources". It

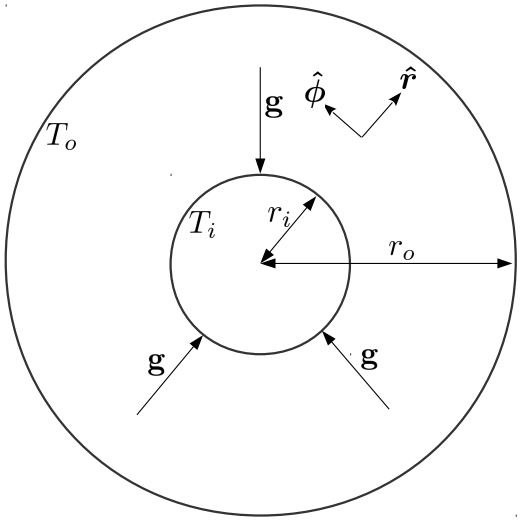

Figure 2.1: Schematic of the problem set-up.

is shown as a schematic diagram in Fig. 2.2. Mathematically, this law can be written for a scalar quantity U as

$$\frac{d}{dt}\int_{\Omega} U \mathrm{d}\Omega = -\oint_{S} \mathbf{F}\cdot\mathbf{dS} + \int_{\Omega} Q_V \mathrm{d}\Omega + \oint_{S} \mathbf{Q_S}\cdot\mathbf{dS}, \tag{2.1}$$

where, $\mathbf{F}$ is the flux contribution, Q_V is the volume source and $\mathbf{Q_S}$ is the surface source. Now, Gauss's theorem states that, surface integral of the flux is equal to volume integral of the divergence of the flux. It is mathematically given as

$$\oint_{S} \mathbf{F}\cdot\mathbf{dS} = \int_{\Omega} \boldsymbol{\nabla}\cdot\mathbf{F}\mathrm{d}\Omega. \tag{2.2}$$

Using Gauss's theorem, equation (2.1) can be rewritten as

$$\frac{\partial}{\partial t}\int_{\Omega} U \mathrm{d}\Omega = -\int_{\Omega} \boldsymbol{\nabla}\cdot\mathbf{F}\mathrm{d}\Omega + \int_{\Omega} Q_V \mathrm{d}\Omega + \int_{\Omega} \boldsymbol{\nabla}\cdot\mathbf{Q_S}\mathrm{d}\Omega. \tag{2.3}$$

Equation (2.3) is valid for an arbitrary volume Ω. So, it must be valid at any point in the domain. Hence, we can write the differential form of equation (2.3) as

$$\frac{\partial U}{\partial t} + \boldsymbol{\nabla}\cdot\mathbf{F} = Q_V + \boldsymbol{\nabla}\cdot\mathbf{Q_S}. \tag{2.4}$$

Next, we define the vector conservation law for a vector $\mathbf{U}$ as

$$\frac{\partial}{\partial t}\int_{\Omega} \mathbf{U}\mathrm{d}\Omega = -\int_{\Omega} \boldsymbol{\nabla}\cdot\overline{\overline{F}}\mathrm{d}\Omega + \int_{\Omega} \mathbf{Q_V}\mathrm{d}\Omega + \int_{\Omega} \boldsymbol{\nabla}\cdot\overline{\overline{Q}}_S\mathrm{d}\Omega, \tag{2.5}$$

where, the fluxes and the surfaces sources are given in tensor notation and a tensor is denoted by the double bars. Subsequently, we can write the differential form of equation

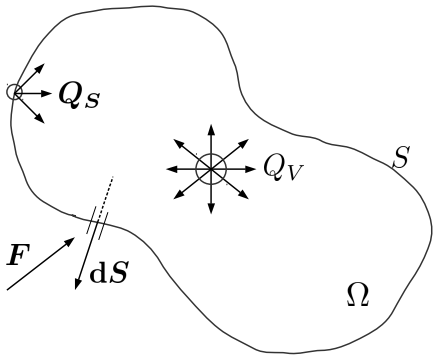

Figure 2.2: Schematic for the general form of a conservation law for a scalar quantity.

(2.5) as

$$\frac{\partial \mathbf{U}}{\partial t} + \boldsymbol{\nabla} \cdot \overline{\overline{F}} = \mathbf{Q_V} + \boldsymbol{\nabla} \cdot \overline{\overline{Q}}_S. \tag{2.6}$$

We will use the differential form of the conservation laws defined in equations (2.4) and (2.6) to derive the governing equations for our problem. The governing equations for any fluid flow system consists of conservation of mass, momentum and energy quantities. The equations for these three quantities are discussed next.

2.2.1 Continuity equation

We first derive the mass conservation equation also known as the continuity equation. As mass is a scalar, we must use the scalar conservation law given in equation (2.4). We use specific mass which is nothing but the density ρ, so we have $U \equiv \rho$. There are no external mass sources and no diffusion of mass. Thus, there is only the convective flux which is given as $\mathbf{F_C} = \rho\mathbf{u}$. Here, $\mathbf{u}$ is the fluid flow velocity and its three cylindrical components are denoted as (u_r, u_ϕ, u_z). Therefore, the mass conservation equation given in differential form will be

$$\frac{\partial \rho}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \mathbf{u}) = 0. \tag{2.7}$$

It can be rewritten as

$$\frac{d\rho}{dt} + \rho \boldsymbol{\nabla} \cdot \mathbf{u} = 0. \tag{2.8}$$

where, the material derivative is defined as

$$\frac{d}{dt} \equiv \frac{\partial}{\partial t} + \mathbf{u} \cdot \boldsymbol{\nabla}. \tag{2.9}$$

2.2.2 Momentum equation

Next, we derive the momentum conservation equation. Momentum is the product of mass and velocity which becomes a product of density and velocity if we consider specific mass. So, referring to the vector conservation law in equation (2.6) we have, $\mathbf{U} \equiv \rho\mathbf{u}$. The convective flux can be given as

$$\overline{\overline{F}}_C = \rho\mathbf{u} \otimes \mathbf{u}, \tag{2.10}$$

where, $\otimes$ denotes tensor product of two vectors. Next, we define the forces which contribute to the source terms. The external force per unit mass is $\mathbf{f}_e$ and the internal force per unit mass is $\mathbf{f}_i$. The external force in our case is the gravity. Hence, $\mathbf{f}_e = \mathbf{g}$ but we will use $\mathbf{f}_e$ for generality for now. The internal force is defined as

$$\mathbf{f}_i = (\overline{\overline{\sigma}} \cdot \hat{\mathbf{n}})_i, \tag{2.11}$$

where, $\overline{\overline{\sigma}}$ is the internal stress tensor and $\hat{\mathbf{n}}$ is the unit normal vector to the surface. For a Newtonian fluid flow, the internal stress tensor consists of the normal stress and the shear stress components. It is given as

$$\overline{\overline{\sigma}} = -p\overline{\overline{I}} + \overline{\overline{\tau}}, \tag{2.12}$$

where, p is the isotropic pressure, $\overline{\overline{I}}$ is the identity tensor and $\overline{\overline{\tau}}$ is the viscous shear stress tensor which is given as

$$\tau_{ij} = \mu\left[\left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j}\right)\right] - \frac{2}{3}(\boldsymbol{\nabla} \cdot \mathbf{u})\delta_{ij}, \tag{2.13}$$

where, μ is the dynamic viscosity and δ_{ij} is the Kronecker delta. Therfore, $\mathbf{Q}_V = \rho\mathbf{f}_e$ and $\mathbf{Q}_S = \overline{\overline{\sigma}} \cdot \hat{\mathbf{n}}$. Now, using the general form of the vector conservation law given in equation (2.6), the momentum conservation equation is given to be

$$\frac{\partial \rho\mathbf{u}}{\partial t} + \boldsymbol{\nabla} \cdot (\rho\mathbf{u} \otimes \mathbf{u} + p\overline{\overline{I}} - \overline{\overline{\tau}}) = \rho\mathbf{f}_e. \tag{2.14}$$

Subtracting the continuity equation multiplied by $\mathbf{u}$ from above equation, we get

$$\rho\frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u} \cdot \boldsymbol{\nabla})\mathbf{u} = -\boldsymbol{\nabla}p + \boldsymbol{\nabla} \cdot \overline{\overline{\tau}} + \rho\mathbf{f}_e. \tag{2.15}$$

And for constant viscosity coefficients we get

$$\rho\frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u} \cdot \boldsymbol{\nabla})\mathbf{u} = -\boldsymbol{\nabla}p + \mu\left[\nabla^2\mathbf{u} + \frac{1}{3}\boldsymbol{\nabla}(\boldsymbol{\nabla} \cdot \mathbf{u})\right] + \rho\mathbf{f}_e. \tag{2.16}$$

This equation is called the Navier-Stokes equations which governs the motion of wide variety of fluid physics.

2.2.3 Energy equation

The next equation we derive is the energy equation where $U \equiv (e + \mathbf{u}^2/2)$, where, e is the internal energy and $\mathbf{u}^2/2$ is the kinetic energy per unit mass. Their sum is denoted by the total energy E. The energy equation is a result of the first law of thermodynamics (Tritton (1988)). It states that, the rate of change of energy inside a fluid element is equal to the net flux of heat into the element plus the rate of work done by surface and body forces. From equation (2.4), we have $U \equiv E$. We define the fluxes first. The convective flux now becomes

$$\mathbf{F}_C = \rho\mathbf{u}\left(e + \frac{\mathbf{u}^2}{2}\right). \tag{2.17}$$

The diffusive flux is given as

$$\mathbf{F}_D = -k\boldsymbol{\nabla}T, \tag{2.18}$$

which is coming from the Fourier's law of heat conduction, where, k is the thermal conductivity. The volume source of energy is given as

$$Q_V = \rho\mathbf{f}_e \cdot \mathbf{u} + Q, \tag{2.19}$$

where, $\mathbf{f}_e$ is the volume force and Q is a heat source other than conduction. The surface source of energy is due to the internal stresses acting on the surface. It is given as

$$\mathbf{Q_S} = \overline{\overline{\sigma}} \cdot \mathbf{u} = -p\mathbf{u} + \overline{\overline{\tau}} \cdot \mathbf{u}. \tag{2.20}$$

Referring to equation (2.4), the differential form of the energy equation can now be written as

$$\frac{\partial \rho E}{\partial t} + \boldsymbol{\nabla} \cdot (\rho\mathbf{u}E - k\boldsymbol{\nabla}T - \overline{\overline{\tau}} \cdot \mathbf{u} + p\mathbf{u}) = \rho\mathbf{f}_e \cdot \mathbf{v} + Q. \tag{2.21}$$

Now, multiplying $\mathbf{u}$ to the momentum equation in (2.16) and subtracting it from equation (2.21) gives the equation

$$\rho\frac{de}{dt} + p\nabla \cdot \mathbf{u} = k\boldsymbol{\nabla}^2 T + Q. \tag{2.22}$$

This is the equation for internal energy. Thus, we concern ourselves with equations (2.7), (2.16) and (2.22) which are the three main governing equations.

2.2.4 Boussinesq approximation

Now, with respect to our problem of thermal convection in an annulus, we make a physical approximation called the Boussinesq approximation for the fluid. This approximation

implies that, density variations are negligible with respect to the background density ρ_o everywhere in the momentum equation except in the term where gravity is present. It assumes that the vertical extent of the modeled domain is small relative to the hydrostatic scale heights of density. The density scale height is given as $L_e = -(\mathrm{d}ln\rho/\mathrm{d}r)^{-1}$ and if our domain length is L, it means that $L \ll L_e$. The density has a linear relationship with the temperature which is shown as

$$\rho = \rho_o(1 - \alpha\Delta T), \tag{2.23}$$

where, ρ_o is the background density and α is the coefficient of thermal expansion. This approximation means that we filter out the sound waves (Glatzmaier, 2014, Chapter 1, p. 11). Since, sound waves travel faster than the fluid and represent small perturbations in density, the numerical timestep size required to capture such perturbations has to be naturally small for the solution to be resolved in time. This makes the numerical computation expensive. To alleviate this limitation, we use the Boussinesq approximation. Hence, continuity equation (2.7) with the Boussinesq approximation will become

$$\boldsymbol{\nabla} \cdot \mathbf{u} = 0. \tag{2.24}$$

Likewise, the momentum equation in (2.16) with Boussinesq approximation will become

$$\frac{\partial \mathbf{u}}{\partial t} = -\mathbf{u} \cdot \boldsymbol{\nabla}\mathbf{u} - \frac{1}{\rho_o}\boldsymbol{\nabla}p + \alpha g_o T\hat{\mathbf{r}} + \nu\nabla^2\mathbf{u}, \tag{2.25}$$

where, ν is the kinematic viscosity, which is given as $\nu = \mu/\rho_o$. We consider the fluid to be a perfect gas. Also, refering to the momentum equation given in equation (2.16), we observe that the external force $\mathbf{f}_e = \mathbf{g}$. Hence, $\rho_o de/dt$ is $\rho_o c_v dT/dt$, where, c_v is the specific heat capacity at constant volume and T is the temperature. Also, $k = \rho_o \kappa c_p$ and $c_p = c_v$, where c_p is the specific heat capacity at constant pressure and κ is the thermal diffusivity. Substituting these conditions into equation (2.22) will result in the energy equation to be

$$\frac{\partial T}{\partial t} = -\mathbf{u} \cdot \boldsymbol{\nabla}T + \kappa\nabla^2 T. \tag{2.26}$$

The heating term Q in equation (2.22) is absent in case a fixed temperature contrast is adopted (Glatzmaier, 2014, Chapter 1, p. 13). Regarding the boundary conditions for the equations, we either have the no-slip or stress-free boundary conditions for the velocities and we have a constant boundary condition for the temperature. They will be detailed after we show the vorticity-streamfunction formulation.

2.2.5 Vorticity-streamfunction formulation

Next, we consider the vorticity-streamfunction formulation of the governing equations. We use this formulation for two reasons. First reason is, we do not need to solve for the

divergence of velocity which is numerically cumbersome. Second reason is, we reduce the number of equations so that it is computationally cheaper. We know that vorticity can be expressed as $\boldsymbol{\omega} = \boldsymbol{\nabla} \times \mathbf{u}$ and we solve for a two dimensional system, where, $u_z = 0$ and $\partial/\partial z = 0$. Therefore, vorticity is represented only by its z-component as

$$\boldsymbol{\omega} = \frac{1}{r}\left[\frac{\partial(r u_\phi)}{\partial r} - \frac{\partial u_r}{\partial \phi}\right]\hat{\mathbf{z}} = \omega\hat{\mathbf{z}}. \tag{2.27}$$

As for the velocity, it is written in terms of a streamfunction, Ψ as $\mathbf{u} = \boldsymbol{\nabla} \times (\Psi\hat{\mathbf{z}})$. Since we have an annular geometry, the model allows for a non-vanishing mean flow $\overline{u_\phi}$, where the bar indicates the average over ϕ-direction (Plaut and Busse (2002)). This mean flow is given as

$$\overline{u_\phi} = \frac{1}{2\pi}\int_0^{2\pi} u_\phi d\phi. \tag{2.28}$$

Now, by separating this mean flow from Ψ, the expression for $\mathbf{u}$ can now be written as

$$\mathbf{u} = \overline{u_\phi}\hat{\boldsymbol{\phi}} + \boldsymbol{\nabla} \times (\psi\hat{\mathbf{z}}), \tag{2.29}$$

where, ψ is the non-axisymmetric streamfunction without the meanflow. The corresponding individual velocity components are

$$u_r = \frac{1}{r}\frac{\partial\psi}{\partial\phi}, \tag{2.30a}$$

$$u_\phi = \overline{u_\phi} - \frac{\partial\psi}{\partial r}. \tag{2.30b}$$

This representation of velocity automatically satisfies the divergence free constraint shown in equation (2.24) and when substituted into the vorticity definition, we get

$$\omega = \frac{1}{r}\frac{\partial(r\overline{u_\phi})}{\partial r} - \nabla^2\psi, \tag{2.31}$$

which is the constraint equation to solve for streamfunction along with the impermeable boundary conditions that are discussed in one of the forthcoming sections. The following set of equations are the governing equations with the vorticity-streamfunction formulation

$$\frac{\partial T}{\partial t} = -\mathbf{u}\cdot\boldsymbol{\nabla}T + \kappa\nabla^2 T, \tag{2.32a}$$

$$\frac{\partial \omega}{\partial t} = -\mathbf{u}\cdot\boldsymbol{\nabla}\omega + \nu\nabla^2\omega - \alpha g_o\left(\frac{1}{r}\right)\frac{\partial T}{\partial\phi}, \tag{2.32b}$$

$$\omega = \frac{1}{r}\frac{\partial(r\overline{u_\phi})}{\partial r} - \nabla^2\psi, \tag{2.32c}$$

$$\frac{\partial\overline{u_\phi}}{\partial t} = -\overline{(\mathbf{u}\cdot\boldsymbol{\nabla}\mathbf{u})}_\phi + \tilde{\Delta}\overline{u_\phi}, \tag{2.32d}$$

where, $\tilde{\Delta} = \partial_r(\partial_r + 1/r)$. Equation (2.32b) is the vorticity equation and it is obtained by taking the curl of the momentum conservation equation (2.25). Equation (2.32d) is the equation for mean flow $\overline{u_\phi}$. It is given as a supplement to the vorticity equation to ensure periodicity of pressure (Plaut and Busse (2002)). We then nondimensionalize these equations. Length, time, and temperature are scaled by the gap width ($D = r_o - r_i$), the viscous diffusion time D^2/ν, where, ν is the kinematic viscosity, and ΔT, respectively. Using these scalings, the full set of dimensionless governing equations on which we focus henceforth are

$$\frac{\partial T}{\partial t} = -\mathbf{u}\cdot\boldsymbol{\nabla}T + \frac{1}{Pr}\nabla^2 T, \tag{2.33a}$$

$$\frac{\partial \omega}{\partial t} = -\mathbf{u}\cdot\boldsymbol{\nabla}\omega + \nabla^2\omega - \frac{Ra}{Pr}\left(\frac{1}{r}\right)\frac{\partial T}{\partial \phi}, \tag{2.33b}$$

$$\omega = \frac{1}{r}\frac{\partial(r\overline{u_\phi})}{\partial r} - \nabla^2\psi, \tag{2.33c}$$

$$\frac{\partial \overline{u_\phi}}{\partial t} = -\overline{(\mathbf{u}\cdot\boldsymbol{\nabla}\mathbf{u})}_\phi + \tilde{\Delta}\overline{u_\phi}, \tag{2.33d}$$

The two nondimensional numbers shown in the equations are the Rayleigh number and the Prandtl number and they are discussed in the following section. Regarding the initial conditions, we provide a Gaussian perturbation on top of the conductive state. We introduce the conductive profile of the temperature T_c in an annulus geometry. It is given as a solution of

$$\frac{\mathrm{d}}{\mathrm{d}r}\left(r\frac{\mathrm{d}T_c}{\mathrm{d}r}\right) = 0, \tag{2.34}$$

with boundary conditions $T_c = 1$ at $r = r_i$ and $T_c = 0$ at $r = r_o$. The resulting solution is given as

$$T_c = \frac{\log r - \log r_o}{\log r_i - \log r_o}. \tag{2.35}$$

2.3 Dimensionless control parameters

The main dimensionless control parameters for our numerical simulations are the Rayleigh number and the Prandtl number. They are defined as

$$Ra = \frac{g_o \alpha \Delta T D^3}{\nu\kappa}, \tag{2.36}$$

$$Pr = \frac{\nu}{\kappa}, \tag{2.37}$$

The Rayleigh number measures the strength of convection. The Prandtl number is the ratio of momentum diffusivity to thermal diffusivity. It indicates about the flow structure compared to the thermal structure (Glatzmaier, 2014, Chapter 1). The radius ratio is an input to the problem and it is given as $\eta = r_i/r_o$.

2.4 Boundary conditions

Regarding the boundary conditions for the annulus, it is naturally periodic in the azimuthal direction and it can have either a Stress-Free (SF) or a No-Slip (NS) boundary condition along the curved inner and outer boundary walls along with the impermeable condition. The impermeable boundary condition states

$$\psi = 0 \text{ at } r = r_i,\ r_o. \tag{2.38}$$

The no-slip boundary condition is defined as

$$\frac{\partial \psi}{\partial r} = 0 \text{ at } r = r_i,\ r_o, \tag{2.39}$$

which results in vorticity at the boundaries to be $\omega = -\frac{\partial^2 \psi}{\partial r^2}$. The stress-free boundary condition is defined as

$$\frac{\partial^2 \psi}{\partial r^2} - \frac{1}{r}\frac{\partial \psi}{\partial r} = 0 \text{ at } r = r_i,\ r_o, \tag{2.40}$$

which results in vorticity at the boundaries to be $\omega = -\frac{2}{r}\frac{\partial \psi}{\partial r}$. Thus, we have four boundary conditions on ψ. Likewise, for equation (2.33d), the no-slip boundary condition is

$$\overline{u_\phi} = 0 \text{ at } r = r_i,\ r_o, \tag{2.41}$$

and stress-free boundary condition is

$$\frac{\partial \overline{u_\phi}}{\partial r} - \frac{1}{r}\overline{u_\phi} = 0 \text{ at } r = r_i,\ r_o. \tag{2.42}$$

As for the temperaure, the boundary conditions are

$$T = 1 \text{ at } r = r_i, \tag{2.43a}$$

$$T = 0 \text{ at } r = r_o. \tag{2.43b}$$

Chapter 3

Numerical implementation

This chapter focusses on the description of the numerical methods which were used to model thermal convection in an annulus. The spatial discretization methods are first described followed by their application to the governing equations. After that, the time discretization methods are discussed. Subsequently, the practical implementation details of the said numerical methods for the governing equations are described. After that, the various time integration methods that were implemented in the code are summarized followed by the description of the diagnostic parameters. Finally, the validation of the code is shown.

3.1 Spectral methods

Spectral methods are a class of numerical methods used to solve differential equations. These methods are extensively applied to many problems in fluid dynamics. In this thesis we use spectral methods to solve the governing differential equations that we saw in the previous chapter.

These methods belong to a class of weighted residual methods (Peyret, 2002; Boyd, 2001). It is a method where we seek an approximate solution in the form of a series expansion. For clarity, one dimensional functions are used while defining these methods. A function $u(x)$, x being the independent variable, can be approximated as

$$u_N(x) \approx \sum_{m=0}^{N} \hat{u}_m \Phi_m(x), \ a \leq x \leq b, \tag{3.1}$$

where, $N+1$ is the number of modes, $u_N(x)$ is the numerical approximation of $u(x)$, $\hat{u}_m$ is the spectral coefficient and $\Phi_m(x)$ is the basis function, also called as the trial function. The basis function for a Fourier expansion is a trigonometric function e^{imx} and for a Chebyshev expansion, they are Chebyshev polynomials $\mathrm{T}_k(x)$. a and b are the end points

of the physical space in x. These methods usually define a residual which has to be zero in an approximate sense. A residual R_N is given to be

$$R_N(x) = u - u_N. \tag{3.2}$$

Consider a differential equation

$$Lu - f = 0, \tag{3.3}$$

where, L is the operator of the differential equation and f is a function of x. The residual is defined as

$$R_N(x) = Lu_N - f \tag{3.4}$$

The weighted residual method constists of setting R_N to zero such that

$$\int_a^b R_N \Psi_j w dx = 0, \ j = 0 \text{ to } N, \tag{3.5}$$

where, Ψ_i is the test function and w is the weight. It is to be noted that in collocation type methods, the test function is defined as

$$\Psi_j = \delta(x - x_j), \ j = 0 \text{ to } N, \tag{3.6}$$

where, δ is the Kronecker delta function. Also, the weight becomes $w = 1$ and the points x_i are called the collocation points. Therefore, from equation (3.4), we observe that $R_N(x_i) = 0$. Hence, the residual is exactly zero at the collocation points. Thus, we make the residual to be zero at a set of points which is equal in number to the undetermined coefficients or the number of modes $(N + 1)$. Since we use the collocation points, the method we use is known as the "pseudospectral" method (Boyd, 2001, Chapter 1, p. 2). However, the name "pseudospectral" also comes from using such spectral methods to a problem where nonlinear products of two dependent variables for e.g. a quadratic nonlinear term is transformed in to physical (grid) space where the variables are simply multiplied and then the product is transformed back in to the spectral (mode) space (Glatzmaier, 2014). It is notable to find the equivalence of collocation and residual methods for certain choices of the test function as in Boyd (2001).

3.1.1 Choice of pseudospectral methods

One reason we use the pseudospectral methods for spatial discretization of our problem is because, for a given number of degrees of freedom, these methods provide solutions which are always more accurate than the local methods such as the finite element methods or the finite difference methods when the solution sought is smooth enough, meaning there

are no sudden steep gradients of variable values at any location in the solution domain as it might occur for e.g. in the presence of a shockwave. We know that the collocation type spectral methods result in matrix problems with a full matrix. So, when the number of grid points N increases, so does the memory requirement for the resulting matrix problem. Therefore, one can root for the local methods to be of good choice in such situations as they result in a sparse matrix problem and hence less memory requirement. However, for the problems we are concerned with, where we require very high spatial accuracy of order $O(10^{-9})$ or less, the amount of grid points needed in the local methods to obtain such high accuracy makes them more memory intensive than the pseudospectral methods. Moreover, given the number of grid points N, the pseudospectral error (Boyd, 2001, Chapter 1, p. 8) is given to be

$$\text{Pseudospectral error} \approx O[(1/N)^N]. \tag{3.7}$$

This means that the error is decreasing faster than any finite power of N because the power in this function always increases. Such a property is known as the exponential convergence (Boyd, 2001; Orzag, 1980). Therefore, considering both memory requirement and accuracy as compared to the local methods, the pseudospectral method is the method of choice for spatial discretization for our particular problem. Moreover, there are recent advances in spectral methods where they obtain a sparse matrix problem by using a Chebyshev integration method, see for e.g. in Marti et al. (2016), Gastine (2019). However, their implementation is out of scope in this work but it will be considered for future implementation. Next, we briefly describe the Fourier and Chebyshev methods used in this work.

3.1.2 Fourier method

In the Fourier method, we have the basis functions to be a trigonometric function. In equation (3.1), it is given as

$$\Phi_m(x) = e^{imx}, \tag{3.8}$$

where, $i^2 = -1$. In our problem, we use the Fourier method only along the azimuthal direction since it is periodic. The assumption of periodicity in a direction along which we apply the Fourier method is mandatory because if we apply these methods along a non-periodic direction, we will introduce the Gibbs oscillations (Gibbs (1899)) because such oscillations result in the presence of sharp gradients in the solution for example near a velocity boundary layer. Such oscillations will contaminate the solution. We will now obtain expressions for the Fourier coefficients. Let the function $u(x)$ be approximated by a truncated trigonometric series as

$$u_K(x) = \sum_{m=-K}^{K} \hat{u}_m e^{imx}. \tag{3.9}$$

Therefore, we have to find $N = 2K + 1$ coefficients. Since $u(x)$ is assumed to be real, we have

$$\hat{u}_{-m} = \hat{u}_m^*, \tag{3.10}$$

where, the star indicates complex conjugate. Next, we show the orthogonality property of an exponential complex function as

$$\int_0^{2\pi} e^{imx} e^{-ilx} dx = \begin{cases} 2\pi \text{ if } m = l, \\ 0 \text{ if } m \neq l, \end{cases} \tag{3.11}$$

where, e^{-ilx} is the test function. According to the weighted residual method, we have the scalar product

$$(R_K, e^{-ilx})_w = 0, \ l = -K, \cdots, K, \tag{3.12}$$

where, $w = 1$. Expanding the above equation gives

$$\int_0^{2\pi} R_K e^{-ilx} \mathrm{d}x = 0, \tag{3.13}$$

which can be further expanded as

$$\int_0^{2\pi} \left[u e^{-ilx} - \sum_{m=-K}^{K} \hat{u}_m e^{i(m-l)x} \right] dx = 0. \tag{3.14}$$

Using equation (3.11) in (3.14) results in the expression for the Fourier coefficients

$$\hat{u}_m = \frac{1}{2\pi} \int_0^{2\pi} u e^{-imx} dx, \ m = -K, \ldots, K. \tag{3.15}$$

In order to express the equation (3.15) as a discrete sum, we can apply the collocation method where the residual is zero at each of the collocation points. The collocation points are defined to be

$$x_j = \frac{2\pi j}{N}, \ j = 0, \ldots, N, \tag{3.16}$$

where, $x_0 = 0$ and $x_N = 2\pi$. Since the residual is zero at the collocation points, it is given as

$$R_K(x_j) = u(x_j) - u_K(x_j) = 0, \ j = 1, \ldots, N, \tag{3.17}$$

and therefore

$$u(x_j) = \sum_{m=-K}^{K} \hat{u}_m e^{imx_j}, \ j = 1, \ldots, N, \tag{3.18}$$

where $N = 2K + 1$. The discrete orthogonality relation is now given to be

$$\sum_{i=1}^{N} e^{i(m-l)\frac{2\pi j}{N}} = \begin{cases} N \text{ if } m - l = kN,\ k = 0, \pm 1, \pm 2, \dots \\ 0 \text{ otherwise.} \end{cases} \tag{3.19}$$

Multiplying equation (3.18) on both sides by e^{-ilx} and summing from $j = 1, \cdots, N$ and by using the discrete orthogonality relation, we get the discrete form of equation (3.15) to be

$$\hat{u}_m = \frac{1}{N} \sum_{j=1}^{N} u(x_j) e^{-imx_j},\ m = -K, \cdots, K. \tag{3.20}$$

This equation is the forward Fourier transform. For clarity, we rewrite the equation (3.9) again as

$$u(x) = \sum_{m=-K}^{K} \hat{u}_m e^{imx}. \tag{3.21}$$

This equation is called the inverse Fourier transform or generally the Discrete Fourier Transform (DFT) (Peyret (2002)). For the implementation of these forward and inverse Fourier transforms in the code, we use the Fast Fourier Transform FFT subroutines given by the FFTW libraries (Frigo (1999)). Next, we will describe the Chebyshev method that was used in this work.

3.1.3 Chebyshev method

The Chebyshev methods are generally utilized for spatial discretization along the non-periodic directions (Peyret, 2002; Glatzmaier, 2014). We use the same independent variable x as we used in the previous section for simplicity and will consider one dimensional functions to illustrate the forward and inverse Chebyshev transforms. Following (Glatzmaier, 2014, Chapter 9), a Chebyshev polynomial of the first kind $\mathrm{T}_k(x)$ is a polynomial of degree k defined for $x \in [-1, 1]$. It is given as

$$\mathrm{T}_k(x) = \cos(k \cos^{-1} x). \tag{3.22}$$

From (Zwillinger, 2018, Chapter 6, p. 453), it can also be expressed as,

$$\mathrm{T}_k(x) = \frac{k}{2} \sum_{m=0}^{\lfloor k/2 \rfloor} (-1)^m \frac{(k-m-1)!}{m!(k-2m)!} (2x)^{k-2m}. \tag{3.23}$$

The degree of the Chebyshev polynomial is given as the number of zeroes of the polynomial in between $-1 \leq x \leq 1$. It is shown in Fig. 3.1. When we increases the polynomial degree, the number of zero crossings cluster towards the boundaries which makes Chebyshev

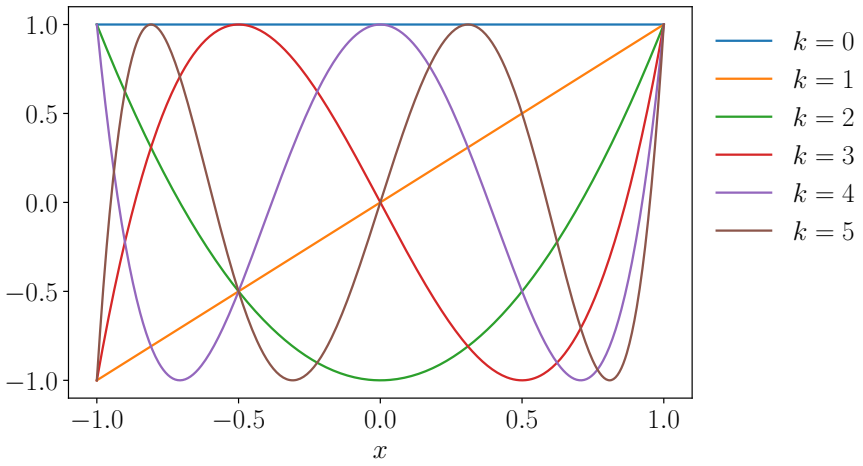

Figure 3.1: Plot of the Chebyshev polynomials versus x for degrees $k = 0 \rightarrow 5$.

methods an attractive method for numerical simulation of fluid flows with thin boundary layers. It is given from the definition of a Chebyshev polynomial of the first kind, $\mathrm{T}_k(1) = 1$ and $\mathrm{T}_k(-1) = (-1)^k$. Now, given

$$\mathrm{T}_0(x) = 1, \tag{3.24}$$

$$\mathrm{T}_1(x) = x, \tag{3.25}$$

higher degree polynomials are obtained by the recursive relation given as

$$\mathrm{T}_{k+1}(x) = 2x\mathrm{T}_k(x) - \mathrm{T}_{k-1}(x). \tag{3.26}$$

For obtaining the recursion relations for the first and second derivatives of the Chebyshev polynomials, we have to use the derivatives of equations (3.24) and (3.25). The recursive relation for the first derivative is

$$\frac{\mathrm{dT}_{k+1}}{\mathrm{d}x} = 2\mathrm{T}_k + 2x\frac{\mathrm{dT}_k}{dx} - \frac{\mathrm{dT}_{k-1}}{\mathrm{d}x}, \tag{3.27}$$

and the recursive relation for the second derivative is

$$\frac{\mathrm{d}^2\mathrm{T}_{k+1}}{\mathrm{d}x^2} = 4\frac{\mathrm{dT}_k}{\mathrm{d}x} + 2x\frac{\mathrm{d}^2\mathrm{T}_k}{\mathrm{d}x^2} - \frac{\mathrm{d}^2\mathrm{T}_{k-1}}{\mathrm{d}x^2}. \tag{3.28}$$

These recursive relations will be used to construct the derivative matrices that we will introduce later in this chapter. Next, we define the collocation grid points we used for our problem. We use the Chebyshev-Gauss-Lobatto points (Glatzmaier, 1984). Supposing we need to define the collocation grid points between the inner boundary x_1 and outer boundary x_{N_x}, it is generally given as

$$x_l = x_1 + \frac{x_{N_x} - x_1}{2}\left[1 + \cos\left(\frac{(N_x - l)\pi}{N_x - 1}\right)\right], \tag{3.29}$$

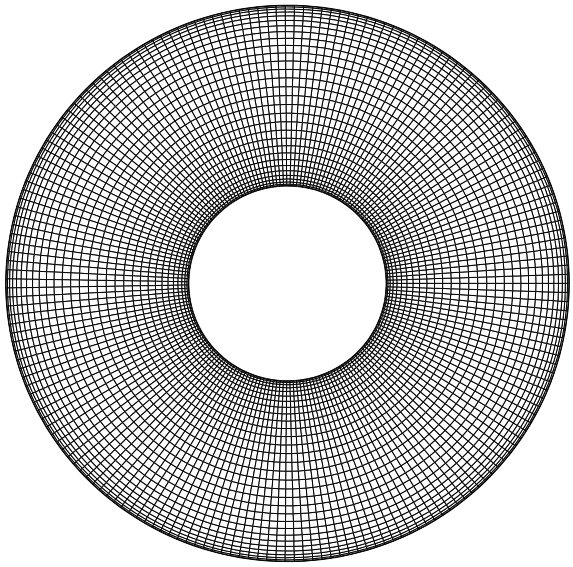

Figure 3.2: The physical grid for $N_r = 32$ and $N_m = 192$.

where, $1 \leq l \leq N_x$. Following (Glatzmaier, 2014, Chapter 9), the function $u(x)$ expanded using Chebyshev polynomials is given as

$$u(x_l) = \left(\frac{2}{N_l - 1}\right)^{1/2} \sum_{k=0}^{N_l-1}{}'' u_k \mathrm{T}_k(x_l), \tag{3.30}$$

where, the double prime indicates that the first and last terms in the summation need to be multiplied by 1/2. The above equation is the Chebyshev inverse transform. The Chebyshev forward transform which gives us the Chebyshev coefficients is given as

$$u_k = \left(\frac{2}{N_l - 1}\right)^{1/2} \sum_{l=1}^{N_l}{}'' u(x_l)\mathrm{T}_k(x_l). \tag{3.31}$$

For obtaining these Chebyshev forward and inverse transforms in the code, we use the Discrete Cosine Transform (DCT) subroutines from the FFTW library. Next, we move on to the details of the Fourier-Chebyshev pseudospectral method applied to the governing equations for our problem. A typical grid generated after the spatial discretization is shown in Fig. 3.2. Along the azimuthal direction, the grid is equispaced and along the radial direction, it is clustered at the boundaries because of the use of Chebyshev-Gauss-Lobatto points. The no-slip boundary condition dictates the velocities (u_r, u_ϕ) to be zero at the inner and outer boundaries of the annulus. For large Reynolds numbers (Re) flows with no-slip boundary conditions, such clustering of the grid points at the boundary is an advantage as it helps to resolve the steep gradients of velocities in the boundary layer. Also, for low Prandtl number (Pr) simulations, it helps us to resolve the thermal

boundary layers.

3.2 Application of the pseudospectral method

We now move on to discuss the application of the method to our problem, see for e.g. (Glatzmaier, 2014, chapter 9, pp. 108-114).

3.2.1 Fourier expansion

We initially present the Fourier expansion of the variables in equations (2.33) as follows

$$T(r,\phi,t) \approx 2\sum_{m=0}^{N_m}{}' \mathrm{Re}(T_m(r,t)e^{im\phi}), \tag{3.32a}$$

$$\omega(r,\phi,t) \approx 2\sum_{m=0}^{N_m}{}' \mathrm{Re}(\omega_m(r,t)e^{im\phi}), \tag{3.32b}$$

$$\psi(r,\phi,t) \approx 2\sum_{m=1}^{N_m} \mathrm{Re}(\psi_m(r,t)e^{im\phi}), \tag{3.32c}$$

where, $N_m + 1$ is the number of Fourier modes, $\mathrm{Re}(f)$ is the real part of a complex function f and the single prime on the summation symbol means the first term in the series is multiplied by $1/2$. Recalling equations (2.30a) and (2.30b), this results in the following expansion for the individual velocities

$$u_r = \frac{2}{r}\sum_{m=1}^{N_m} \mathrm{Re}(im\psi_m e^{im\phi}), \tag{3.33a}$$

$$u_\phi = \overline{u_\phi} - 2\sum_{m=1}^{N_m} \mathrm{Re}\left(\frac{\partial \psi_m}{\partial r}e^{im\phi}\right). \tag{3.33b}$$

Substituting these expressions into the set of equations in (2.33) results in the following set of nondimensional spectral equations for the annulus

$$\frac{\partial T_m(r,t)}{\partial t} = -[\mathbf{u}\cdot\boldsymbol{\nabla}T]_m(r,t) + \frac{1}{Pr}\nabla^2 T_m(r,t), \text{ for } m \geq 0, \tag{3.34a}$$

$$\frac{\partial \omega_m(r,t)}{\partial t} = -[\mathbf{u}\cdot\boldsymbol{\nabla}\omega]_m(r,t) + \nabla^2\omega_m(r,t) - \frac{Ra}{Pr}\frac{im}{r}T_m(r,t), \text{ for } m > 0, \tag{3.34b}$$

$$\omega_m(r,t) = -\nabla^2\psi_m(r,t), \text{ for } m > 0, \tag{3.34c}$$

$$\frac{\partial \overline{u_\phi}(r,t)}{\partial t} = -\overline{(\mathbf{u}\cdot\boldsymbol{\nabla}\mathbf{u})}_\phi(r,t) + \tilde{\Delta}\overline{u_\phi}(r,t), \text{ for } m = 0, \tag{3.34d}$$

$$\overline{\omega}(r,t) = \frac{1}{r}\frac{\partial(r\overline{u_\phi}(r,t))}{\partial r}, \text{ for } m = 0, \tag{3.34e}$$

where, the Laplacian becomes, $\nabla^2 = \partial_{rr} + (1/r)\partial_r - (m^2/r^2)$ and the m^{th} mode of the term inside the square brackets is indicated by $[\cdots]_m$. $\overline{\omega}(r,t)$ from equation (3.34e) is obtained after solving for (3.34d) at $m = 0$. Recalling that we satisfy the divergence constraint in equation (2.24), we rewrite the nonlinear terms in conservative form[1], i.e., as divergences of temperature and vorticity fluxes respectively (Glatzmaier, 2014, Chapter 9, p. 105). Therefore, these terms in equations (3.34a) and (3.34b) are written as, $\mathbf{u}\cdot\boldsymbol{\nabla}T = \boldsymbol{\nabla}\cdot(T\mathbf{u})$ and $\mathbf{u}\cdot\boldsymbol{\nabla}\omega = \boldsymbol{\nabla}\cdot(\omega\mathbf{u})$ respectively. We write in such forms because it is easier to evaluate them in the physical space and then obtain their transforms back into Fourier space.

3.2.2 Chebyshev expansion

Now, coming to the Chebyshev expansion along the radius, the variables $(T_m(r,t), \omega_m(r,t), \psi_m(r,t), \overline{u_\phi}(r,t))$ can be expanded further using the Chebyshev polynomials as

$$T_m(r_j,t) = \left(\frac{2}{N_r-1}\right)^{1/2} \sum_{k=0}^{N_r-1}{}'' T_{mk}(t)\mathrm{T}_k(r_j), \tag{3.35a}$$

$$\omega_m(r_j,t) = \left(\frac{2}{N_r-1}\right)^{1/2} \sum_{k=0}^{N_r-1}{}'' \omega_{mk}(t)\mathrm{T}_k(r_j), \tag{3.35b}$$

$$\psi_m(r_j,t) = \left(\frac{2}{N_r-1}\right)^{1/2} \sum_{k=0}^{N_r-1}{}'' \psi_{mk}(t)\mathrm{T}_k(r_j), \tag{3.35c}$$

$$\overline{u_\phi}(r_j,t) = \left(\frac{2}{N_r-1}\right)^{1/2} \sum_{k=0}^{N_r-1}{}'' u_{\phi_{0k}}(t)\mathrm{T}_k(r_j), \tag{3.35d}$$

where, N_r is the number of radial grid points with $1 \leq j \leq N_r$, k is the degree of Chebyshev polynomial T_k and the double prime on the summation symbol means the first and last terms in the series are multiplied by 1/2. r_j consists of the Chebyshev-Gauss-Lobatto

[1] a conservative form is a native form that comes from the conservation laws which were discussed in the previous chapter.

points defined as

$$r_j = r_i + \frac{r_o - r_i}{2}\left[1 + \cos\left(\frac{(N_r - j)\pi}{N_r - 1}\right)\right]. \tag{3.36}$$

For each Fourier mode m, let the vectors $(\mathtt{T}_m, \zeta_m, \Psi_m, \mathtt{u}_{\phi 0})$ denote the radial values of $(T_{mk}, \omega_{mk}, \psi_{mk}, u_{\phi 0k})$ respectively for all k in the range $(0 \leq k \leq N_r - 1)$. They are in Fourier-Chebyshev space. Now, substituting these expressions into the equations in (3.34), we get the following set of governing equations in semi-discrete form, where, for each Fourier mode m running from 0 to N_m, we have to solve for the variables $\mathtt{T}_m, \zeta_m, \Psi_m, \mathtt{u}_{\phi 0}$ such that

$$\mathbf{T}\frac{\mathrm{d}\mathtt{T}_m}{\mathrm{d}t} = -\mathbf{N}_1\mathtt{N}_{T_1} - \mathbf{N}_{2m}\mathtt{N}_{T_2} + \mathbf{L}_m\mathtt{T}_m, \text{ for } m \geq 0, \tag{3.37a}$$

$$\mathbf{T}\frac{\mathrm{d}\zeta_m}{\mathrm{d}t} = -\mathbf{N}_1\mathtt{N}_{\omega_1} - \mathbf{N}_{2m}\mathtt{N}_{\omega_2} + \mathbf{L}_m\zeta_m - \frac{Ra}{Pr}\mathbf{N}_{2m}\mathtt{T}_m, \text{ for } m > 0, \tag{3.37b}$$

$$\mathbf{T}\zeta_m = -\mathbf{L}_m\Psi_m, \text{ for } m > 0, \tag{3.37c}$$

$$\mathbf{T}\frac{\mathrm{d}\mathtt{u}_{\phi 0}}{\mathrm{d}t} = -\mathbf{T}\mathtt{C}_0 + \mathbf{L}_0\mathtt{u}_{\phi 0}, \text{ for } m = 0, \tag{3.37d}$$

where, $\mathtt{C}_0 = [(\mathbf{u}\cdot\boldsymbol{\nabla}\mathbf{u})_\phi]_{(m=0)k}$. ζ_0 is updated immediately after solving for $\mathtt{u}_{\phi 0}$. Also, the vectors $\mathtt{N}_{T_1}$ and $\mathtt{N}_{T_2}$ are radial values of $(u_r T)_{mk}$ and $(u_\phi T)_{mk}$. Similarly, $\mathtt{N}_{\omega_1}$ and $\mathtt{N}_{\omega_2}$ are radial profiles of $(u_r\omega)_{mk}$ and $(u_\phi\omega)_{mk}$ respectively. The products in these nonlinear terms $(u_r T, u_\phi T)$ and $(u_r\omega, u_\phi\omega)$ respectively are computed pseudo-spectrally using the 2/3 rule (Boyd, 2001, Chapter 9). The concept of de-aliasing is discussed in appendix B. The operator matrices $\mathbf{T}$, $\mathbf{N}_1$, $\mathbf{N}_{2m}$, $\mathbf{L}_m$ and $\mathbf{L}_0$ are expressed by

$$\mathbf{T} = \left(\frac{2}{N_r - 1}\right)^{1/2} {\sum_{k=0}^{N_r-1}}'' \mathrm{T}_k, \tag{3.38a}$$

$$\mathbf{N}_1 = \left(\frac{2}{N_r - 1}\right)^{1/2} {\sum_{k=0}^{N_r-1}}'' \left[\frac{1}{r}\mathrm{T}_k + \frac{d\mathrm{T}_k}{dr}\right], \tag{3.38b}$$

$$\mathbf{N}_{2m} = \left(\frac{2}{N_r - 1}\right)^{1/2} {\sum_{k=0}^{N_r-1}}'' \left[\frac{im}{r}\mathrm{T}_k\right], \tag{3.38c}$$

$$\mathbf{L}_m = \left(\frac{2}{N_r - 1}\right)^{1/2} {\sum_{k=0}^{N_r-1}}'' \left[\frac{d^2\mathrm{T}_k}{dr^2} + \frac{1}{r}\frac{d\mathrm{T}_k}{dr} - \frac{m^2}{r^2}\mathrm{T}_k\right], \tag{3.38d}$$

$$\mathbf{L}_0 = \left(\frac{2}{N_r - 1}\right)^{1/2} {\sum_{k=0}^{N_r-1}}'' \left[\frac{d^2\mathrm{T}_k}{dr^2} + \frac{1}{r}\frac{d\mathrm{T}_k}{dr}\right]. \tag{3.38e}$$

The semi-discrete equations in (3.37) can now be cast into a system of Discrete Algebraic Equations (DAEs) as

$$\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}t} = \mathbf{f}(\mathbf{y}, \Psi_m), \tag{3.39a}$$

$$\mathbf{0} = \mathbf{h}(\mathbf{y}, \Psi_m), \tag{3.39b}$$

where $\mathbf{y}$ is a vector of the unknowns $(\mathtt{T}_m, \zeta_m, \mathtt{u}_{\phi_0})$, and both $\mathbf{f}$ and $\mathbf{h}$ are vectors which are functions of the unknowns according to the equations shown in (3.37). A DAE system of equations consists of both the differential and the algebraic equations. The equations shown above comes in the class of index-2 DAEs (Ascher and Petzold, 1998, Chapter 9, p. 234). The number of times the DAE is to be differentiated in time so as to obtain an ODE after elimination gives the index of the DAE. The equation (3.39a) is the differential equation and the equation (3.39b) is the algebraic equation or the constraint and the variable Ψ_m is called the algebraic variable.

3.3 Temporal discretization

In order to obtain a solution from a problem which is time dependent, we would be required to discretize in time. The theme of this thesis is based on time domain discretization for which there are variety of numerical techniques available. Before diving into the application of the time domain discretization to the problem at hand, we will discuss some basic concepts of time domain discretization and its classes of methods. We describe methods based on finite differences in time. Although we use spectrally accurate spatial discretization, a spectrally accurate time domain discretization for e.g. a Chebyshev approximation in time would be very expensive to compute. Therefore, for simulations of fluid flows, especially in the dynamo community, we use finite differences in time. As mentioned before in the introduction, most of the present day dynamo codes use a second order time integration method alongside a highly resolved spatial discretization. In order to preserve the high accuracy of the spectral method, the truncation error associated with the finite difference approximation in time must be sufficiently small (Peyret (2002)). Therefore, despite the low order of the scheme, the stability will demand a small timestep size as we shall see in the coming sections. As we noted in the introduction, whether a higher order method will be a better choice for our practical problem is one of the questions we will answer in the next chapter that discusses the results.

3.3.1 Stiffness: Brief definition

The concept of stiffness arises when we seek numerical solution of Ordinary Differential Equations (ODEs). The term "stiff" is used by many authors in the literature in different contexts with respect to their problem. It was first coined by Curtiss and Hirschfelder

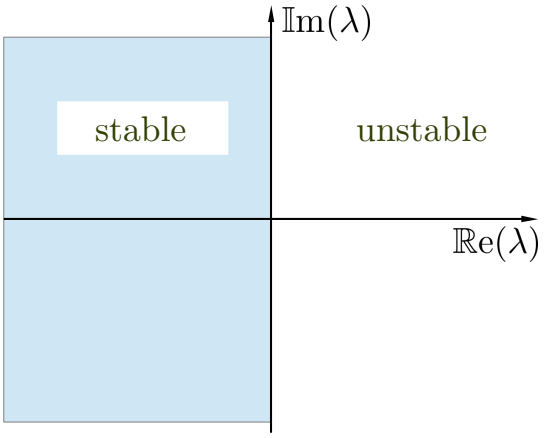

Figure 3.3: Domain of analytical stability.

(1952). Various aspects of stiffness in the application of numerical methods for ODEs are covered by Spijker (1996) and according to Lambert (1991), the main characteristics of stiffness can be stated as follows

1. A linear system is stiff if all its eigenvalues have large negative real part. It will be explained in the upcoming subsections.
2. Stiffness occurs when the stability requirements force the timestep size to small values than that is required to obtain accurate solution.
3. Stiffness occurs when some components of the solution decay faster than the other.

Point 1 assumes that the eigenvalues of the linear operator have large negative real part. Therefore, large negative real values indicate that we require small timestep sizes to maintain stability. We will now illustrate Point 1 using a simple example for basic understanding. Consider a linear system of the form

$$\frac{dy}{dt} = Ay, \; y(t_0) = y_0, \tag{3.40}$$

where, A is the operator matrix also known as the Jacobian. The solution of such a system can be analytically given as

$$y(t) = \exp(A \cdot t) \cdot y_0. \tag{3.41}$$

This solution is called analytically stable if it remains bounded as time goes to infinity. In other words, it is analytically stable if and only if all the eigenvalues of A have negative real parts, which means

$$\mathrm{Re}(\lambda) < 0, \tag{3.42}$$

where, λ is the eigenvalue. In a complex λ-plane, the stable and unstable regions are shown in Fig. 3.3.

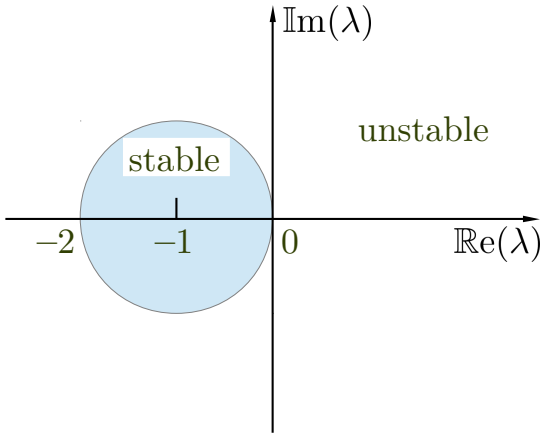

Figure 3.4: Domain of numerical stability for forward Euler method.

We will be introducing different classes of time integration methods later in this chapter. However, to illustrate numerical stability, we use a forward Euler method to equation (3.40) which results in

$$y_{n+1} = R(hA)y_n, \tag{3.43}$$

where, $h = \Delta t$, is the timestep size and R is the stability function which is defined in this case as $R(z) = 1 + z$. Suppose A is diagonalizable with eigenvalues $v_1, \cdots, v_n$, we can express y_0 using the eigenvector basis as

$$y_0 = \sum_{i=1}^{n} \alpha_i v_i. \tag{3.44}$$

From equation (3.43), we know that,

$$y_1 = R(hA)y_0, \tag{3.45}$$
$$y_2 = R(hA)y_1 = (R(hA))^2 y_0, \tag{3.46}$$
$$\vdots$$
$$y_n = (R(hA))^n y_0. \tag{3.47}$$

When using equation (3.44) in the above expression, we get

$$y_n = \sum_{i=1}^{n} (R(h\lambda_i))^n \alpha_i v_i, \tag{3.48}$$

Following (Hairer and Wanner, 2010, Chapter 4, p. 16), y_n remains bounded for $n \to \infty$, if for all eigenvalues, the complex number $z = h\lambda_i$ lies in the set

$$S = \{z \in \mathbb{C}; |R(z)| \leq 1\} = \{z \in \mathbb{C}; |z - (-1)| \leq 1\}$$

which is a circle of radius 1 with centre at -1. It is illustrated in Fig. 3.4. Therefore, if we use the forward Euler method for the example problem shown in equation 3.40, we get a stable solution if the eigenvalues lie inside the unit circle shown in Fig. 3.4. Also, suppose $\lambda = -100$, then h has a constraint $0 \leq h \leq 2/100$. So, the larger the negative real eigenvalue, the smaller the timestep size. Also of importance to the time schemes are the properties A-stablity and L-stability. The time scheme is A-stable if the stability region contains the set $\{z \in \mathbb{C} | Re(z) < 0\}$ and it is L-stable if it is A-stable and if $|R(z)| \to 0$ as $z \to \infty$.

Point 2 means that when we use an explicit method, we will be forced to choose small timestep sizes to maintain stable solution. Let us consider a scalar advection-diffusion equation in one dimension x of the form

$$\frac{\partial u}{\partial t} = u \cdot \frac{\partial u}{\partial x} + \nu \frac{\partial^2 u}{\partial x^2}, \tag{3.49}$$

where, $u(x)$ is the required solution, ν is the kinematic viscosity, $u\frac{\partial u}{\partial x}$ is the nonlinear term and $\nu\frac{\partial^2}{\partial x^2}$ is the linear term. The stability conditions for the above equation if we treat the time discretization explicitly can be given as

$$\Delta t_{\text{explicit}} = \alpha \ \min\left[\min\left(\frac{\Delta x}{|u|_{\max}}\right), \min\left(\frac{\Delta x^2}{\nu}\right)\right], \tag{3.50}$$

where, α is known as the Courant-Friedrichs-Lewy (CFL) number (Courant et al. (1928)) and Δx is the grid spacing. From the above equations, it is clear that if we use a fully explicit method in time, the timestep size is restricted by the square of the smallest grid size or the strongest u or the largest magnitude of u. Usually it is the grid size which affects the Δt, however, it remains to be seen if that changes at very high Re.

Point 3 means that if the timescale for diffusion, τ_L, is smaller compared to the nonlinear timescale, τ_{NL}, in other words, if there is a disparity in timescales, stiffness will ensue.

Large eigenvalues (point 1) means we need to use smaller timesteps which also results when we use a less stable method (point 2) and small τ_L (point 3) values can result from stringent stability conditions as well. Therefore, in a way all three points are connected to each other. We will continue to identify stiffness in a general sense with these three points in the future. However, we will discuss how we perceive stiffness for our problem when we discuss the results.

3.3.2 Approaches for integration in time

Now, following Peyret (2002), the way to treat the equations in time depends on the type of equations we handle which signify the fluid flow type. As mentioned in the introduction, we deal with the governing equations which contain both diffusive and advective components. Generally the different ways to integrate in time are categorized into three different types as follows

- *Fully explicit method* These methods are generally used for compressible flows or incompressible flows with variable viscosity cases where nonlinearity exists in both the advective and diffusive components.
- *Fully implicit method* These methods are generally applied for very "stiff" problems and especially in conjunction with local methods for spatial discretization. The term "stiff" generally means that, when any explicit scheme applied to such problems creates extremely small timestep sizes.
- *Implicit-Explicit (IMEX) method* These methods are generally used for problems which we concern ourselves with, where, an implicit scheme is applied on the linear component of the equation and an explicit scheme is applied on the nonlinear component of the equation. Thus, for the problem at hand we will use the IMEX approach for time domain discretization.

Before we proceed with the IMEX methods, we will now introduce the two classes of time stepping strategies which we follow in this thesis. They are the multistep and the multistage methods. We will discuss them as follows.

3.3.3 Multistep methods: Basic construction

3.3.3.1 Methods based on integration

In this section, we will give an introduction to the multistep methods. The multistep methods use several time levels simultaneously and the order of accuracy of the method is usually defined by the number of time levels. Let us consider an Initial Value Problem (IVP) of the form

$$\frac{dy}{dt} = f(t, y),\ y(t_0) = y_0. \tag{3.51}$$

Initially, we show the first time integration method provided by Leonard Euler (Euler, 1758). Suppose we choose Δt as the timestep size and $t_n = t_0 + n\Delta t$, the Euler method says

$$y_{n+1} = y_n + \Delta t f(t_n, y_n), \tag{3.52}$$

where, the value of y_n is an approximate solution to the ordinary differential equation in equation (3.51). This method is a one-step method where it uses one previous time level and it is an explicit method of order one. The first multistep method was introduced in the article Bashforth (1883). His problem was solved using a multistep method introduced by J.C. Adams. In one-step methods, one seeks the numerical solution only from the differential equation and the initial value. Adams devised a multistep method consisting of two parts (Hairer et al., 2008). First, a starting procedure which provides $y_1, \cdots, y_{k-1}$ which are approximations to the exact solutions at the points $t_0+\Delta t, \cdots, t_0+(k-1)\Delta t$ and

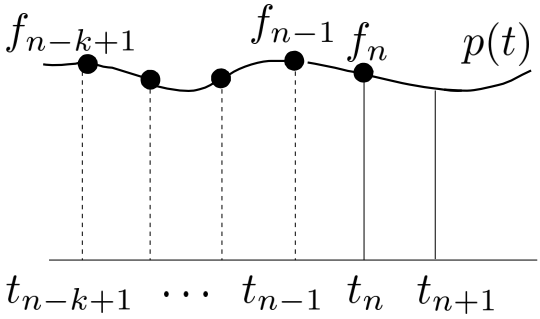

Figure 3.5: Polynomial interpolation of function values at previous timelevels for Explicit Adams methods.

then the multistep scheme to obtain an approximation to the exact solution $y(t_0 + k\Delta t)$. Based on the k successive steps, it is applied recursively to obtain $y(t_0 + (k-1)\Delta t)$ and so on. One way to obtain the starting values is to use any one-step method or start with a low order Adams method and use very small timestep sizes. For illustrative purposes, we will now derive the Adams method. This was the first multistep method and it is known as the Adams-Bashforth method. Consider the notation $t_i = t_0 + i\Delta t$ and suppose we know the numerical approximations $y_n, y_{n-1}, \cdots, y_{n-k+1}$ to the exact solution $y(x_n), \cdots, y(x_{n-k+1})$ of the Initial Value Problem (IVP) shown in equation (3.51). The integrated form of the IVP can be written as

$$y(t_{n+1}) = y(t_n) + \int_{t_n}^{t_{n+1}} f[t, y(t)]\mathrm{d}t. \tag{3.53}$$

We have assumed that we know the previous timestep values of $y(t)$ $(y_{n-k+1}, \cdots, y_n)$ and thus the function values

$$f_i = f(t_i, y_i) \text{ for } i = n-k+1, \cdots, n. \tag{3.54}$$

Therefore, the function $f(t, y(t))$ can be approximated as a polynomial $p(t)$ through the points $\{(t_i, y_i) | i = n-k+1, \cdots, n\}$ as shown in Fig. 3.5. The polynomial is expressed in terms of Newton's backward differences

$$\nabla^0 f_n = f_n, \;\; \nabla^{j+1} f_n = \nabla^j f_n - \nabla^j f_{n-1}. \tag{3.55}$$

For clarity, we show for $j = 1$,

$$\nabla^1 f_n = \nabla^0 f_n - \nabla^0 f_{n-1} = f_n - f_{n-1} \tag{3.56}$$

and for $j = 2$,

$$\begin{aligned} \nabla^2 f_n &= \nabla^1 f_n - \nabla^1 f_{n-1}, \\ &= f_n - f_{n-1} - (f_{n-1} - f_{n-2}), \\ &= f_n - 2f_{n-1} + f_{n-2}. \end{aligned} \tag{3.57}$$

Now, the polynomial is given as

$$p(t) = p(t_n + h\Delta t) = \sum_{j=0}^{k-1} (-1)^j \binom{-h}{j} \nabla^j. \tag{3.58}$$

where,

$$h = \frac{t - t_n}{\Delta t}, \tag{3.59}$$

and

$$(-1)^j \binom{-h}{j} = \frac{h(h+1)\cdots(h+j-1)}{j!}. \tag{3.60}$$

This is the Newton's interpolation formula (Newton, 1711). Therefore, we can write equation (3.53) as

$$y_{n+1} = y_n + \int_{t_n}^{t_{n+1}} p(t)\mathrm{d}t. \tag{3.61}$$

Inserting equation (3.58) in equation (3.61) gives

$$y_{n+1} = y_n + \int_{t_n}^{t_{n+1}} \sum_{j=0}^{k-1} (-1)^j \binom{-h}{j} \nabla^j f_n \, \mathrm{d}t. \tag{3.62}$$

We know that $r = (t - t_n)/\Delta t$, therefore, we do a change of variables from t to r in the integral which means, at $t = t_n$, $r = 0$ and at $t = t_{n+1}$, $r = 1$. Also $\mathrm{d}t = \Delta t \mathrm{d}r$. Therefore, we get

$$y_{n+1} = y_n + \Delta t \sum_{j=1}^{k-1} \left[\int_0^1 (-1)^j \binom{-h}{j} \mathrm{d}h \right] \nabla^j f_n, \tag{3.63}$$

which can be simplified as

$$y_{n+1} = y_n + \Delta t \sum_{j=0}^{k-1} \gamma_j \nabla^j f_n, \tag{3.64}$$

where the coefficients γ_j satisfy

$$\gamma_j = (-1)^j \int_0^1 \binom{-h}{j} \mathrm{d}h. \tag{3.65}$$

The coefficients γ_j are usually obtained by a recursive relation (Hairer et al., 2008, Chapter 3, p. 359). It is given as

$$\gamma_m + \frac{1}{2}\gamma_{m-1} + \frac{1}{3}\gamma_{m-2} + \cdots + \frac{1}{m+1}\gamma_0 = 1. \tag{3.66}$$

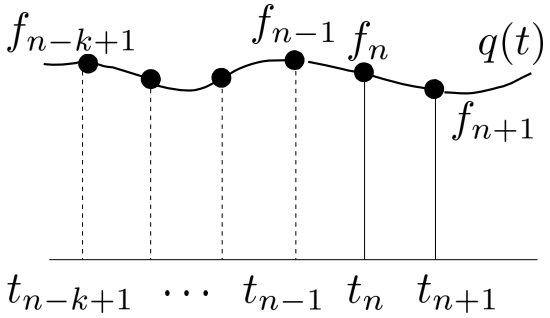

Figure 3.6: Poynomial interpolation of function values at current and previous timelevels for Adams-Moulton methods.

For illustrative purposes, let us derive the time scheme for $k = 2$. For continuity, we once again write the general time scheme as

$$y_{n+1} = y_n + \Delta t \sum_{j=0}^{k-1} \gamma_j \nabla^j f_n. \tag{3.67}$$

Therefore, for $k = 2$, we get

$$y_{n+1} = y_n + \Delta t \sum_{j=0}^{1} \gamma_j \nabla^j f_n, \tag{3.68}$$

which can be further simplified as

$$y_{n+1} = y_n + \Delta t[\gamma_0 \nabla^0 f_n + \gamma_1 \nabla^1 f_n]. \tag{3.69}$$

We will first get the values of the coefficients γ_0 and γ_1 as

$$\gamma_0 = \int_0^1 (-1)^0 \binom{-h}{0} \mathrm{d}h, \tag{3.70}$$

which becomes

$$\gamma_0 = \int_0^1 \mathrm{d}h = 1. \tag{3.71}$$

Likewise,

$$\gamma_1 = \int_0^1 (-1)^1 \binom{-h}{1} \mathrm{d}h, \tag{3.72}$$

which becomes

$$\gamma_1 = \int_0^1 \frac{h}{1!} \mathrm{d}h = \frac{1}{2}. \tag{3.73}$$

Therefore, equation (3.68) can be written as

$$y_{n+1} = y_n + \Delta t[f_n + \frac{1}{2}(f_n - f_{n-1})]. \tag{3.74}$$

Thus, one obtains

$$y_{n+1} = y_n + \Delta t\left(\frac{3}{2}f_n - \frac{1}{2}f_{n-1}\right), \tag{3.75}$$

which is the second order Adams-Bashforth method. We use it for the treatment of the advective component of the governing equations. Adams also constructed an implicit scheme that involves the evaluation of the function f_{n+1} at the point (t_{n+1}, f_{n+1}). This method is also known as the Adams-Moulton method. The polynomial expressing the functions of previous timesteps and the present one as is the case now is shown in Fig. 3.6. It is given as

$$q(t) = q(t_n + \Delta t) = \sum_{j=0}^{k}(-1)\begin{pmatrix}-h+1\\ j\end{pmatrix}\nabla^j f_{n+1}, \tag{3.76}$$

Therefore,

$$y_{n+1} = y_n + \int_{t_n}^{t_{n+1}} q(t)\mathrm{d}t. \tag{3.77}$$

The implicit scheme can now be written as

$$y_{n+1} = y_n + \Delta t\sum_{j=0}^{k}\alpha_j\nabla^j f_{n+1}, \tag{3.78}$$

where, the coefficient α_j satisfies

$$\alpha_j = (-1)^j\int_0^1\begin{pmatrix}-h+1\\ j\end{pmatrix}\mathrm{d}r. \tag{3.79}$$

For order $k = 2$, the method becomes

$$y_{n+1} = y_n + \frac{\Delta t}{2}(f_{n+1} + f_n). \tag{3.80}$$

This method is also known as the Crank-Nicholson method which results from the Adams-Moulton method for $k = 2$. It can also be obtained as a combination of a forward Euler method and a backward Euler method. As discussed in chapter 1, this is a commonly used time scheme since Glatzmaier (1984) and it is used to treat the diffusive component of the governing equations.

3.3.3.2 Methods based on differentiation

Now that we have seen the workings of a multistep method based on integration, we will now describe a multistep method based on differentiation. These methods are called the Backward Difference Formula (BDF) methods. They were first introduced by Curtiss and Hirschfelder (1952). We will be using the BDF methods to treat the diffusive component of the governing equations. Assume the approximations as $y_{n-k+1}, \cdots, y_n$ to the exact solution of equation (3.51) are known. As with the multistep methods shown before, we choose a polynomial $l(t)$ which interpolates the values $\{(t_i, y_i)|i = n-k+1, \cdots, n+1\}$. This polynomial is expressed in terms of backward differences as

$$l(t) = l(t_n + \Delta t) = \sum_{j=0}^{k} (-1)^j \binom{-h+1}{j} \nabla^j y_{n+1}. \tag{3.81}$$

y_{n+1} will be determined when the polynomial $l(t)$ satisfies the differential equation at one grid point at least, i.e.

$$l'(t_{n+1-i}) = f(t_{n+1-i}, y_{n+1-i}). \tag{3.82}$$

If we put $i = 1$ we can obtain explicit time schemes. For example, $i = 1$ will give the explicit Euler method. In this case, we put $i = 0$ and we can draw a polynomial like the one shown in Fig. 3.7. The implicit backward difference formula is now given as

$$\sum_{j=0}^{k} \delta_j \nabla^j y_{n+1} = \Delta t f_{n+1}, \tag{3.83}$$

where the coefficients are

$$\delta_j = (-1)^j \frac{d}{dh} \binom{-h+1}{j} \Bigg|_{r=1}. \tag{3.84}$$

By using binomial expansion of

$$(-1)^j \binom{-h+1}{j} = \frac{1}{j!}(h-1)h(h+1), \cdots, (h+j-2), \tag{3.85}$$

the coefficients now become

$$\delta_0 = 0, \ \delta_j = \frac{1}{j} \text{ for } j \geq 1. \tag{3.86}$$

Therefore, the BDF method now reads

$$\sum_{j=1}^{k} \frac{1}{j} \nabla^j y_{n+1} = \Delta t f_{n+1}. \tag{3.87}$$

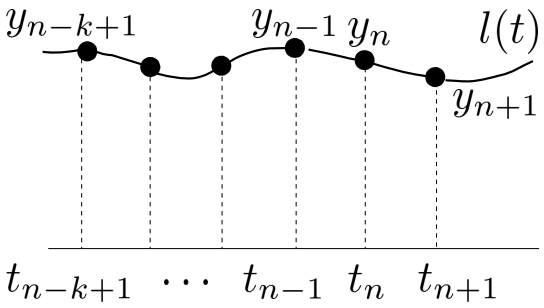

Figure 3.7: Poynomial interpolation of solution values at current and previous timelevels for Backward Difference Formula (BDF) methods.

For illustrative purposes we will list the second, third and fourth order BDF scheme as follows:
For $k = 2$ (second order),

$$\frac{3}{2}y_{n+1} - 2y_n + \frac{1}{2}y_{n-1} = \Delta t f_{n+1}. \tag{3.88}$$

For $k = 3$ (third order),

$$\frac{11}{6}y_{n+1} - 3y_n + \frac{3}{2}y_{n-1} - \frac{1}{3}y_{n-2} = \Delta t f_{n+1}. \tag{3.89}$$

For $k = 4$ (fourth order),

$$\frac{25}{12}y_{n+1} - 4y_n + 3y_{n-1} - \frac{4}{3}y_{n-2} + \frac{1}{4}y_{n-3} = \Delta t f_{n+1}. \tag{3.90}$$

We have thus seen the construction of the multistep methods with a simple ODE equation given in (3.51). Also, these time schemes are valid for fixed timestep sizes only. For variable timestep sizes, we will have to change the weights. However, for the problem at hand, we have already mentioned that we will be using an IMEX formalism which we will describe soon. Next, we discuss about the multistage methods.

3.3.4 Multistage methods: Basic construction

The multistage methods are basically one-step methods which involve calculation of intermediate values at each time level. Let us once again consider an Initial Value Problem (IVP) of the form

$$\frac{dy}{dt} = f(t, y),\ y(t_0) = y_0. \tag{3.91}$$

Suppose we choose Δt as the timestep size and $t_n = t_0 + n\Delta t$, as shown in the previous section, the explicit one-step Euler method (Euler, 1758) reads

$$y_{n+1} = y_n + \Delta t f(t_n, y_n). \tag{3.92}$$

The value of y_n is an approximate solution to the ODE shown in equation (3.91). Runge (1895) used the midpoint rule to this ODE which results in

$$y(t_0 + \Delta t_0) \approx y_0 + \Delta t f\left[\left(t_0 + \frac{\Delta t}{2} + y\left(t_0 + \frac{\Delta t}{2}\right)\right]\right. . \tag{3.93}$$

He then used the Euler method with timestep size $\Delta t/2$ and wrote the above equation as

$$\begin{aligned} k_1 &= f(t_0, y_0), \\ k_2 &= f\left(t_0 + \frac{\Delta t}{2}, y_0 + \frac{\Delta t}{2} k_1\right), \\ y_1 &= y_0 + \Delta t k_2. \end{aligned} \tag{3.94}$$

Thus, Runge (1895) introduced a method that is second order accurate.

3.3.4.1 General formulation of explicit Runge-Kutta methods

Runge (1895) and Heun (1900) made inroads to the development of the first multistage methods. After that, Kutta (1901) gave a general framework to obtain explicit multistage methods of any order, methods now known as the Runge-Kutta methods. It can be described as follows. Let s be the number of stages of the Runge-Kutta method and $a_{21}, a_{31}, a_{32}, \cdots, a_{s1}, a_{s2}, \cdots, a_{s,s-1}, b_1, \cdots, b_s, c_2, \cdots, c_s$ be real coefficients. The general s-stage Explicit Runge-Kutta (ERK) method can be given as

$$\begin{aligned} k_1 &= f(t_0, y_0) \\ k_2 &= f(t_0 + c_2\Delta t, y_0 + \Delta t a_{21} k_1) \\ k_3 &= f(t_0 + c_3\Delta t, y_0 + \Delta t(a_{31} k_1 + a_{32} k_2) \\ &\cdots \\ k_s &= f(t_0 + c_s\Delta t, y_0 + \Delta t(a_{s1} k_1 + \cdots + a_{s,s-1} k_{s-1})) \\ y_1 &= y_0 + \Delta t(b_1 k_1 + \cdots + b_s k_s). \end{aligned} \tag{3.95}$$

The other condition which has to be satisfied is

$$c_i = \sum_{j=1}^{i-1} a_{ij}, \text{ for } i = 1, 2, \cdots, s. \tag{3.96}$$

These conditions express that all points where f is evaluated are first order approximations to the solution. Following (Hairer et al., 2008, Chapter 2, p. 134), a Runge-Kutta method has an order p if for sufficiently smooth problems

$$\|y(t_0 + \Delta t) - y_1\| \leq K \Delta t^{p+1}, \tag{3.97}$$

where, K is a constant. This equation means that the Taylor's series for the exact solution $y(t_0 + \Delta t)$ and the numerical solution coincide up to and including the term h^p. From

Butcher (1964b), it became customary to write the constants in the method in a particular format known as the Butcher's tableau. For the s-stage ERK, the Butcher's tableau is given as

$$\begin{array}{c|ccccc} 0 & 0 & 0 & 0 & \cdots & 0 \\ c_2 & a_{21} & 0 & 0 & \cdots & 0 \\ c_3 & a_{31} & a_{32} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ c_s & a_{s1} & a_{s2} & \cdots & a_{s,s-1} & 0 \\ \hline & b_1 & b_2 & \cdots & b_{s-1} & b_s \end{array} \tag{3.98}$$

Therefore, the second order Runge's method introduced before can now be given in Butcher's table as

$$\begin{array}{c|cc} 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ \hline & 0 & 1 \end{array} \tag{3.99}$$

The Butcher's tableau will be extensively used to show all the different multistage time integration methods used in this thesis. Other important requirements of the Runge-Kutta method are the order conditions which are to be satisfied to maintain the said order of the method (Butcher, 1964b; Hairer et al., 2008). For illustrative purposes we show the order conditions for the ERK up to order three as follows
For first order accuracy

$$\sum_{i=1}^{s} b_i = 1. \tag{3.100}$$

For second order accuracy

$$\sum_{i=1}^{s} b_i c_i = \frac{1}{2}. \tag{3.101}$$

For third order accuracy

$$\sum_{i=1}^{s} b_i c_i^2 = \frac{1}{3}, \tag{3.102a}$$

$$\sum_{i=1}^{s} \sum_{j=1}^{s} b_i a_{ij} c_j = \frac{1}{6}. \tag{3.102b}$$

Such order conditions vary depending on the type of Runge-Kutta method being used. Also, stage order is an important ingredient of a multistage Runge-Kutta method. From Bendsten (1997), a stage order is defined as follows. An s-stage Runge-Kutta method is

said to have a stage order q if q is the largest number such that the condition $C(q)$, i.e.

$$\sum_{j=1}^{s} a_{ij} c_j^{k-1} = \frac{c_i^k}{k} \text{ for } k = 1, \cdots, q \text{ and all } i, \tag{3.103}$$

holds. It is considered that, when a Runge-Kutta method is applied to a stiff problem, the order of accuracy is reduced to the stage order of the method (Ascher and Petzold, 1998, Chapter 4, p. 107). Therefore, a low stage order method may result in overall order reduction when applied to stiff equations. Also, when these Runge-Kutta methods are applied to our index-2 DAE system of governing equations, severe order reduction will ensue unless we satisfy the necessary additional order conditions pertaining to the type of the method (Ascher and Petzold, 1998, Chapter 10, p. 271). We will study such phenomena for our problem when we discuss the results.

Now that we have discussed the basic workings of multistep and multistage time integration methods, we will next talk about the IMEX methods and their application to our problem.

3.3.5 IMEX methods: Application to our problem

The mixed set of DAEs resulting from the Fourier-Chebyshev discretization described in the previous section has to be integrated in time. We solve the system of DAEs simultaneously in time using time integration methods which are developed for ODEs. We focus on the well known Implicit-Explicit (IMEX) time integrators (Ascher et al., 1995, 1997) and analyze their performance when applied to this problem. There are different families of IMEX methods, and here we are interested in both IMEX multistep and IMEX multistage methods (Ascher et al., 1997; Kennedy and Carpenter, 2003). At each timestep, the temperature equation is solved first as it is not directly coupled with the vorticity or the streamfunction equations. After we solve for temperature, we solve for the mean azimuthal velocity and the two coupled equations for vorticity and streamfunction. We will describe the multistep and multistage methods in the upcoming subsections.

3.3.6 IMEX multistep methods

For our problem, the DAEs shown in the equations (3.39) can be rewritten as

$$\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}t} = \mathbf{f}^a(\mathbf{y}) + \mathbf{f}^d(\mathbf{y}), \tag{3.104a}$$

$$\mathbf{0} = \mathbf{h}(\mathbf{y}, \Psi_m), \tag{3.104b}$$

The terms $\mathbf{f}^a$ and $\mathbf{f}^d$ denote the advective and the diffusive components of the equation. In our problem however, we deal with the algebraic constraint by solving the DAE simultaneously. Such approaches for discretizing DAEs in time were introduced by Gear (1971).

Since we advance equation (3.104a) in time, an IMEX method consists of applying an implicit discretization for $\mathbf{f}^d$ and an explicit one for $\mathbf{f}^a$ (Ascher et al. (1995)). Now, consider s to be the number of multisteps of an IMEX method with $s \geq 1$. Let Δt be the timestep size and $\mathbf{y}_n$ denote the approximate solution at time $t_n = n\Delta t$. Then, for fixed Δt, according to Ascher et al. (1995), a general linear multistep IMEX method can be written as

$$\frac{1}{\Delta t}\mathbf{y}_{n+1} + \frac{1}{\Delta t}\sum_{j=0}^{s-1} a_j \mathbf{y}_{n-j} - c_{-1}\mathbf{f}^d(\mathbf{y}_{n+1}) = \sum_{j=0}^{s-1} b_j \mathbf{f}^a(\mathbf{y}_{n-j}) + \sum_{j=0}^{s-1} c_j \mathbf{f}^d(\mathbf{y}_{n-j}), \tag{3.105}$$

where $c_{-1} \neq 0$. An s-step IMEX method cannot have order of accuracy greater than s (Ascher et al. (1995)). The IMEX methods we use have the same order as the number of steps. For the case with variable Δt, following Wang and Ruuth (2008), we denote $\Delta t_{n+j} = t_{n+j+1} - t_{n+j}$ and assume previous s values of $\mathbf{y}^{n+j},\ j = 0, 1, \cdots, s-1$, are known. Correspondingly, a general linear multistep method for variable Δt can be written as

$$\frac{1}{\Delta t_{n+s-1}}\sum_{j=0}^{s} \alpha_{j,n}\mathbf{y}_{n+j} = \sum_{j=0}^{s-1} \beta_{j,n}\mathbf{f}^a(\mathbf{y}_{n+j}) + \sum_{j=0}^{s} \gamma_{j,n}\mathbf{f}^d(\mathbf{y}_{n+j}), \tag{3.106}$$

where $\alpha_{s,n} \neq 0$, $\gamma_{s,n} \neq 0$ and $s \geq 2$. The variable coefficients $\alpha_{j,n}$, $\beta_{j,n}$ and $\gamma_{j,n}$ are functions of the step-size ratios $\Delta t_i / \Delta t_{i-1}$ for $i = n+1, \cdots, n+s-1,\ s \geq 2$ and they satisfy the order conditions shown in Wang and Ruuth (2008).

3.3.7 IMEX Runge-Kutta methods (IMEX-RK)

The IMEX-RK method is constructed as a multistage method (Ascher et al. (1997), Gardner et al. (2018)). At each stage we solve the system of DAEs simultaneously. More details of how we solve them will be discussed in the next section. Considering equation (3.104a) which is advanced in time, the diffusive part is treated implicitly using s-stage Diagonally Implicit Runge-Kutta (DIRK) method and advective part is treated explicitly using Explicit Runge-Kutta (ERK) method. To advance from time t_n to time t_{n+1}, IMEX-RK solution requires s sub-stages and one final stage of assembly depending on the type of the method. The coefficients are given in the form of Butcher's tables (Ascher et al. (1997)). The Butcher's tableau for the implicit part is given as

$$\begin{array}{c|ccccc} 0 & 0 & 0 & 0 & \cdots & 0 \\ c_2 & a_{21} & a_{22} & 0 & \cdots & 0 \\ c_3 & a_{31} & a_{32} & a_{33} & 0 & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ c_s & a_{s1} & a_{s2} & \cdots & a_{s,s-1} & a_{ss} \\ \hline & b_1 & b_2 & \cdots & b_{s-1} & b_s \end{array} \tag{3.107}$$

The Butcher's tableau for the explicit part is given as

$$\begin{array}{c|ccccc} 0 & 0 & 0 & 0 & \cdots & 0 \\ c_2 & \hat{a}_{21} & 0 & 0 & \cdots & 0 \\ c_3 & \hat{a}_{31} & \hat{a}_{32} & 0 & 0 & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ c_s & \hat{a}_{s1} & \hat{a}_{s2} & \cdots & \hat{a}_{s,s-1} & 0 \\ \hline & \hat{b}_1 & \hat{b}_2 & \cdots & \hat{b}_{s-1} & \hat{b}_s \end{array} \tag{3.108}$$

with the condition

$$c_i = \sum_{j=1}^{s} a_{ij} = \sum_{j=1}^{s} \hat{a}_{ij},\ i = 1, 2, \cdots s. \tag{3.109}$$

Now, the general solution at each stage ($i = 1, 2, \cdots, s$) can be written as

$$\mathbf{Y}_i = \mathbf{y}_n + \Delta t \sum_{j=1}^{i} a_{ij} \mathbf{f}^d(t_n + c_j \Delta t, \mathbf{Y}_j) + \Delta t \sum_{j=1}^{i-1} \hat{a}_{ij} \mathbf{f}^a(t_n + c_j \Delta t, \mathbf{Y}_j). \tag{3.110}$$

So, at each stage we need to solve for $\mathbf{Y}_i$ and find $\mathbf{f}^d(\mathbf{Y}_i), \mathbf{f}^a(\mathbf{Y}_i)$. We can find $\mathbf{Y}_i$ by rearranging equation (3.110) as

$$\begin{aligned} (I - \Delta t a_{ii} \mathbf{f}^d)\mathbf{Y}_i = \quad & \mathbf{y}_n + \Delta t \sum_{j=1}^{i-1} a_{ij} \mathbf{f}^d(t_n + c_j \Delta t, \mathbf{Y}_j) \\ + \quad & \Delta t \sum_{j=1}^{i-1} \hat{a}_{ij} \mathbf{f}^a(t_n + c_j \Delta t, \mathbf{Y}_j). \end{aligned} \tag{3.111}$$

The final assembly stage for advancing to the next time step is written as

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \Delta t \sum_{j=1}^{s} b_j \mathbf{f}^d(t_n + c_j \Delta t, \mathbf{Y}_j) + \Delta t \sum_{j=1}^{s} \hat{b}_j \mathbf{f}^a(t_n + c_j \Delta t, \mathbf{Y}_j). \tag{3.112}$$

However, there are methods where the assembly stage is avoided. These methods are known as Stiffly Accurate (SA) IMEX-RK methods (Ascher et al., 1997; Boscarino et al., 2013). A DIRK method is stiffly accurate when

$$b_j = a_{s,j},\ i = 1, 2, \cdots s, \tag{3.113}$$

and an ERK method is stiffly accurate when

$$\hat{b}_j = \hat{a}_{s,j},\ i = 1, 2, \cdots s. \tag{3.114}$$

3.3.8 IMEX methods: Implementation details

With respect to time discretization of the equations in (3.37), we will always be using an IMEX strategy which was discussed in the previous two subsections. The problem in the equations shown in (3.37) will always be reduced to solving for each timestep, for each Fourier mode m, a matrix system $\mathbf{Ax} = \mathbf{b}$ along the non-periodic radial direction, where $\mathbf{A}$ is the operator matrix, $\mathbf{x}$ is the required solution in Fourier-Chebyshev space and $\mathbf{b}$ is the right hand side (rhs) of the equations which contains the nonlinear terms that are treated explicitly and the rhs is represented in the Fourier-real space. The construction of the right hand side $\mathbf{b}$ will vary with the time integration method used. Given that we have the variables at time $t_n = n\Delta t$, in order to obtain the value of variables solved at time t_{n+1}, generally, an IMEX time discretization method applied to temperature equation (3.37a) can be expanded and written as

$$\left(\frac{2}{N_r-1}\right)^{1/2} {\sum_{k=0}^{N_r-1}}'' \left\{ \mathrm{T}_k(r_j) - \frac{\lambda\Delta t}{Pr}\left[\frac{d^2\mathrm{T}_k(r_j)}{dr^2} + \frac{1}{r}\frac{d\mathrm{T}_k(r_j)}{dr} - \frac{m^2}{r^2}\mathrm{T}_k(r_j)\right]\right\} T_{mk}^{n+1} = \mathrm{b}_m^n(r_j), \tag{3.115}$$

where, the value of constant coefficient λ depends on the time integration method used. In matrix notation, for $m \geq 0$, equation (3.115) becomes,

$$\mathbf{A}_m \mathbf{x}_{m_T}^{n+1} = \mathbf{b}_{m_T}^n, \tag{3.116}$$

where,

$$\mathbf{x}_{m_T}^{n+1} = [T_{m0}^{n+1}, T_{m1}^{n+1}, \cdots, T_{m(N_r-1)}^{n+1}]^\intercal, \tag{3.117}$$

$$\mathbf{b}_{m_T}^n = [\mathrm{b}_{m1}^n(T), \mathrm{b}_{m2}^n(T), \cdots, \mathrm{b}_{mN_r}^n(T)]^\intercal. \tag{3.118}$$

$\mathbf{A}_m$ is a $N_r \times N_r$ matrix. We have fixed temperature at the boundaries which is a Dirichlet type boundary condition and it has to be applied in the first and the last rows of $\mathbf{A}_m$. This can be the schematically represented (Glatzmaier, 2014, Chapter 9, p. 113) for $N_r = 5$ as

$$C\begin{bmatrix} \mathrm{T}_{10}/2 & \mathrm{T}_{11} & \mathrm{T}_{12} & \mathrm{T}_{13} & \mathrm{T}_{14}/2 \\ \mathrm{A}_{20m}/2 & \mathrm{A}_{21m} & \mathrm{A}_{22m} & \mathrm{A}_{23m} & \mathrm{A}_{24m}/2 \\ \mathrm{A}_{30m}/2 & \mathrm{A}_{31m} & \mathrm{A}_{32m} & \mathrm{A}_{33m} & \mathrm{A}_{34m}/2 \\ \mathrm{A}_{40m}/2 & \mathrm{A}_{41m} & \mathrm{A}_{42m} & \mathrm{A}_{43m} & \mathrm{A}_{44m}/2 \\ \mathrm{T}_{50}/2 & \mathrm{T}_{51} & \mathrm{T}_{52} & \mathrm{T}_{53} & \mathrm{T}_{54}/2 \end{bmatrix} \begin{bmatrix} T_{m0}^{n+1} \\ T_{m1}^{n+1} \\ T_{m2}^{n+1} \\ T_{m3}^{n+1} \\ T_{m4}^{n+1} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathrm{b}_{m2}^n(T) \\ \mathrm{b}_{m3}^n(T) \\ \mathrm{b}_{m4}^n(T) \\ 0 \end{bmatrix}, \tag{3.119}$$

where, $C = [2/(N_r-1)]^{1/2}$. The first and last rows of $\mathbf{A}_m$ are replaced by the Chebyshev functions of degree k at r-level j, $\mathrm{T}_{jk} \equiv \mathrm{T}_k(r_j)$. The updated temperature is in the Fourier-Chebyshev space and the right-hand side elements for r-level j and Fourier mode m are $\mathrm{b}_{mj}^n(T) \equiv \mathrm{b}_m^n[T(r_j)]$. Regarding the equation for mean azimuthal velocity, proceeding as

before, after discretizing (3.37d) in time, the matrix system we get is

$$\mathbf{A}_0 \mathbf{x}_{0_{\overline{u_\phi}}}^{n+1} = \mathbf{b}_{0_{\overline{u_\phi}}}^{n}, \tag{3.120}$$

where,

$$\mathbf{x}_{0_{\overline{u_\phi}}}^{n+1} = [{u_\phi}_{00}^{n+1}, {u_\phi}_{01}^{n+1}, \cdots, {u_\phi}_{0(N_r-1)}^{n+1}]^\intercal, \tag{3.121}$$

$$\mathbf{b}_{0_{\overline{u_\phi}}}^{n} = [\mathrm{b}_{01}^{n}(\overline{u_\phi}), \mathrm{b}_{02}^{n}(\overline{u_\phi}), \cdots, \mathrm{b}_{0N_r}^{n}(\overline{u_\phi})]^\intercal. \tag{3.122}$$

The method to apply boundary condition for this equation is equivalent to that shown in equation (3.119).

Now, turning to the vorticity and streamfunction equations (3.37b) and (3.37c) respectively, we note here that we solve this DAE simultaneously. After discretizing equation (3.37b) in time, the resulting coupled block matrix system for $m > 0$ is

$$\begin{bmatrix} \mathbf{A}_{m_{11}} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{m_{22}} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{m_\psi}^{n+1} \\ \mathbf{x}_{m_\omega}^{n+1} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{b}_{m_\omega}^{n} \end{bmatrix}, \tag{3.123}$$

where, matrix $\mathbf{A}_{m_{11}}$ is the operator matrix $\mathbf{L}_m$ given in equation (3.38d), matrix $\mathbf{A}_{12}$ is the operator matrix $\mathbf{T}$ given in equation (3.38a), matrix $\mathbf{A}_{21}$ is a matrix filled with zeros and matrix $\mathbf{A}_{m_{22}}$ is the same matrix $\mathbf{A}_m$ shown in equations (3.116). Likewise, the vectors in equation (3.123) are given as

$$\mathbf{x}_{m_\psi}^{n+1} = [\psi_{m0}^{n+1}, \psi_{m1}^{n+1}, \cdots, \psi_{m(N_r-1)}^{n+1}]^\intercal, \tag{3.124}$$

$$\mathbf{x}_{m_\omega}^{n+1} = [\omega_{m0}^{n+1}, \omega_{m1}^{n+1}, \cdots, \omega_{m(N_r-1)}^{n+1}]^\intercal, \tag{3.125}$$

$$\mathbf{b}_{m_\omega}^{n} = [\mathrm{b}_{m1}^{n}(\omega), \mathrm{b}_{m2}^{n}(\omega), \cdots, \mathrm{b}_{mN_r}^{n}(\omega)]^\intercal. \tag{3.126}$$

The impermeability condition $\psi = 0$ is incorporated along the top and bottom rows of $\mathbf{A}_{m_{11}}$ and $\mathbf{A}_{12}$. The no-slip or stress-free conditions involving radial derivatives are incorporated along the top and bottom rows of $\mathbf{A}_{21}$ and $\mathbf{A}_{m_{22}}$. All four boundary conditions are represented by a series of Chebyshev coefficients along the same lines as done in equation (3.119). The matrix equations are solved using standard matrix solvers available in the *LAPACK* routines `dgetrf` and `dgetrs` (Anderson et al., 1999). The `dgetrf` routine is used for computing the LU factorization and the `dgetrs` routine is used for solving the system using the factored matrix obtained by using the `dgetrf` routine.

3.3.8.1 Treatment of IMEX-RK methods with an assembly stage

We explained how we solve the matrix problem in the previous subsection. However, not all methods are stiffly accurate and we have used the IMEX-RK methods which require

an assembly stage. When we perform the assembly step as shown in equation (3.112), it is trivial to apply the boundary condition for a variable if it is of constant value independent of other variables. However, we have a situation where we solve the matrix problem for the vorticity equation coupled with the streamfunction equation along with four boundary conditions on ψ. So, we don't impose any boundary conditions on the vorticity. Now, when we perform the assembly step we need to take care of the boundary value of vorticity to make sure we get an accurate and stable solution. When we get to the assembly step we first assemble ω using equation (3.112). Using this value of ω, we solve for the stream function equation given in equation (2.33c) along with its Dirichlet boundary conditions given in equation (2.38). For $m > 0$, it takes the matrix form

$$\mathbf{L}_m \mathbf{x}_{m_\psi}^{n+1} = -\mathbf{b}_{m_\omega}^{n+1}, \tag{3.127}$$

where, $\mathbf{L}_m$ is the operator matrix shown in equation (3.38d) and the vectors are

$$\mathbf{x}_{m_\psi}^{n+1} = [\psi_{m0}^{n+1}, \psi_{m1}^{n+1}, \cdots, \psi_{m(N_r-1)}^{n+1}]^\intercal, \tag{3.128}$$

$$\mathbf{b}_{m_\omega}^{n+1} = [\omega_{m1}^{n+1}, \omega_{m2}^{n+1}, \cdots, \omega_{mN_r}^{n+1}]^\intercal. \tag{3.129}$$

We use the impermeable boundary condition for ψ and we apply it on the top and bottom rows of $\mathbf{L}_m$. Once we solve this equation, we recover the velocities u_r and u_ϕ using equations (3.33a) and (3.33b). In order to update the value of ω at the boundaries, for each $m > 0$, we use the interior nodal values of ψ along the radius. According to the no-slip boundary condition, ω at the boundaries is defined to be

$$\omega = -\frac{\partial^2 \psi}{\partial r^2} \text{ at } r = r_i,\ r_o. \tag{3.130}$$

We define a local polynomial interpolant of ψ to approximate ω. Let $L(r)$ be the unique interpolating polynomial of degree at most $J+1$ determined by $J+2$ conditions

$$L(r_j) = \psi_{m,j} \text{ for } 0 \le j \le J,\ L'(r_0) = K, \tag{3.131}$$

where, K is a prescribed constant and r_0 can be either r_i or r_o. Now, $L(r)$ can be written as

$$L(r) = \sum_{j=0}^{J} \psi_{m,j} l_j(r) + \left[K - \sum_{j=0}^{J} \psi_{m,j} l_j'(r_0) \right] (r - r_0) l_0(r), \tag{3.132}$$

where, l_j are the Lagrange interpolation basis functions (Abramowitz and Stegun, 1965) and l_j' is the first derivative of the Lagrange polynomial. For the boundary condition on ω shown in equation (3.130), the local vorticity boundary condition in terms of Lagrange

interpolants at $r = r_i$ and $r = r_o$ reads as

$$\omega_{m,0} = -L''(r_0) = -\left(\sum_{j=0}^{J} \psi_{m,j}\left[l_j''(r_0) - 2l_j'(r_0)l_0'(r_0)\right] + 2Kl_0'(r_0)\right), \tag{3.133}$$

where l_j'' is the second derivative of the Lagrange polynomial. The first and second derivatives of the Lagrange polynomials are expanded and written as

$$l_j'(r) = \sum_{i=0, i\neq j}^{l}\left[\frac{1}{r_j - r_i}\prod_{p=0, p\neq(i,j)}^{l}\frac{r - r_p}{r_j - r_p}\right], \tag{3.134}$$

$$l_j''(r) = \sum_{i=0, i\neq j}^{l}\frac{1}{r_j - r_i}\left[\sum_{p=0, p\neq(i,j)}^{l}\left(\frac{1}{r_j - r_p}\prod_{q=0, q\neq(i,j,p)}^{l}\frac{r - r_q}{r_j - r_q}\right)\right]. \tag{3.135}$$

The no-slip boundary condition on ψ will result in $K = 0$. Therefore, we implement equation (3.133) in the code to make sure we satisfy the vorticity boundary condition in the case of an assembly step while using the IMEX-RK methods. In the case of a stress-free boundary condition, one would set $K = u_\phi$ in equation (3.133) to obtain the vorticity at the boundaries.

Also, there is an additional complication with the use of stress-free boundary condition while using methods which have an assembly stage. It arises in the assembly step for the mean flow velocity, $\overline{u_\phi}$. For stress-free boundaries, mean flow velocity is not zero at the boundaries and while we perform an assembly step, the solution is advanced in time only at the interior collocation nodes. Therefore, the boundary values must be updated by another way. After we perform the assembly step, we follow Canuto (1986) to update the boundary values of $\overline{u_\phi}$. It is explained as follows. The stress-free boundary condition for $\overline{u_\phi}$ is given as

$$\frac{\partial \overline{u_\phi}}{\partial r} - \frac{1}{r}\overline{u_\phi} = 0 \text{ at } r = r_i,\ r_o. \tag{3.136}$$

Following (Canuto, 1986, p. 816), we write the boundary conditions as

$$\beta\frac{\partial \overline{u_\phi}}{\partial r} + \alpha\overline{u_\phi} = 0 \text{ at } r = r_o, \tag{3.137}$$

$$\delta\frac{\partial \overline{u_\phi}}{\partial r} + \gamma\overline{u_\phi} = 0 \text{ at } r = r_i. \tag{3.138}$$

Therefore, from equation (3.136), we find the coefficients to be

$$\alpha = -\frac{1}{r_o},\ \beta = 1,\ \gamma = -\frac{1}{r_i},\ \delta = 1. \tag{3.139}$$

Now, following (Canuto, 1986, p. 818), we find the boundary values of $\overline{u_\phi}$ at the new time level by solving the system of equations shown below

$$(\alpha + \beta d_{00})u_o^N + \beta d_{0N} u_N^N = -\beta \sum_{j=1}^{N-1} d_{0j} u_j^N, \tag{3.140}$$

$$(\gamma + \delta d_{NN})u_N^N + \delta d_{N0} u_o^N = -\delta \sum_{j=1}^{N-1} d_{Nj} u_j^N, \tag{3.141}$$

where, $u = \overline{u_\phi}$, $N = N_r$ and $\{d_{ij}\}$ is the matrix representing the first derivative at the collocation points in physical space. Following (Peyret, 2002, Chapter 3, pp. 49-50), to understand how $\{d_{ij}\}$ is constructed, let us define a solution $u_N(x)$ where, $x_i = \cos(\pi i/N)$, $i = 0, \cdots, N$, as follows

$$u_N(x) = \sum_{j=0}^{N} l_j(x) u(x_j), \tag{3.142}$$

where, $u_N(x_j) = u(x_j)$ and $l_j(x)$ is the Lagrange polynomial. Now, we write the general expression for the pth derivative of $u_N^{(p)}$ by differentiating equation (3.142) p times as

$$u_N^{(p)}(x) = \sum_{j=0}^{N} l_j^{(p)} u_N(x_j), \ i = 0, \cdots, N, \tag{3.143}$$

where, $d_{i,j}^{(p)} = l_j^{(p)}(x_i)$. The expression of the first derivative is thus given as

$$\begin{aligned} d_{i,j}^{(1)} &= \frac{\overline{c}_i}{\overline{c}_j} \frac{(-1)^{i+j}}{x_i - x_j}, \ 0 \le i, j \le N, \ i \ne j, \\ d_{i,i}^{(1)} &= -\frac{x_i}{2(1 - x_i^2)}, \ 1 \le i \le N-1, \\ d_{0,0}^{(1)} &= -d_{N,N}^{(1)} = \frac{2N^2 + 1}{6}, \end{aligned} \tag{3.144}$$

where, $x_i = \cos(\pi i/N)$, $\overline{c}_o = \overline{c_N} = 2$, $\overline{c}_j = 1$ for $1 \le j \le N-1$.

3.3.9 Summary of time integration methods

We investigate the accuracy and performance of 24 IMEX methods in Table 3.1 from the literature. Among the IMEX multistep methods, we investigate four different methods namely the CNAB2 (Crank-Nicholson, Adams-Bashforth) which has order 2 and the three Backward Difference Formula (BDF) methods namely SBDF2, SBDF3 and SBDF4 with orders $2, 3,$ and 4 respectively (Ascher et al., 1995). The stencils for these multistep methods are shown in the appendix. Regarding the IMEX-RK methods, we investigate a variety of methods from the literature. We adopt the following naming convention for the

methods listed in Table 3.1. The letters in each of the method names denote the initials of the authors in the reference and the numbers from left to right denote the number of implicit solves, number of explicit solves and the overall order of the method respectively (Grooms and Julien, 2011). The properties of the methods are shown along each column of Table 3.1 from left to right. The last column shows the references. Each IMEX-RK method listed in Table 3.1 is given in Butcher's tableau format in the appendix.

Table 3.1: Properties of the IMEX-RK methods used.

Method	$n\mathbf{f}^d$	$n\mathbf{f}^a$	Order			Stability		S.A.	S.A.	$b_j = \hat{b}_j$	$c_j = \hat{c}_j$	Reference
			E	I	A	A	L	DIRK	ERK			
ARS222	2	3	2	2	2	✓	✓	✓	✓	X	✓	Ascher et al. (1997)
ARS232	2	3	2	2	2	✓	✓	✓	X	✓	✓	Ascher et al. (1997)
BPR442	4	4	2	2	2	✓	✓	✓	✓	X	✓	Boscarino et al. (2017)
PC332	3	3	2	2	2	✓	X	✓	✓	X	✓	-
D332	3	3	2	2	2	✓	✓	✓	✓	✓	✓	Duan (2016)
ARS443	4	4	3	3	3	✓	✓	✓	✓	X	✓	Ascher et al. (1997)
ARS343	3	4	3	3	3	✓	✓	✓	X	✓	✓	Ascher et al. (1997)
ARS233	2	3	3	4	3	✓	X	X	X	✓	✓	Ascher et al. (1997)
KC343	3	4	3	3	3	✓	✓	✓	X	✓	✓	Kennedy and Carpenter (2003)
CFN343	3	4	3	3	3	✓	✓	✓	X	✓	✓	Calvo et al. (2001)
BR553	5	5	3	3	3	✓	✓	X	X	✓	✓	Boscarino and Russo (2009)
BPR433	4	3	3	3	3	✓	✓	✓	✓	X	✓	Boscarino et al. (2013)
BR343	3	4	3	3	3	✓	✓	X	X	✓	✓	Boscarino and Russo (2007)
D543	5	4	3	3	3	✓	✓	✓	✓	✓	✓	Duan (2016)
CB343	3	4	3	3	3	✓	✓	✓	X	✓	✓	Cavaglieri and Bewley (2015)
LZ443	4	4	3	3	3	✓	✓	✓	✓	✓	✓	Liu and Zou (2006)
KC564	5	6	4	4	4	✓	✓	✓	X	✓	✓	Kennedy and Carpenter (2003)
CB564	5	6	4	4	4	✓	X	✓	X	✓	✓	Cavaglieri and Bewley (2015)
LZ664	6	6	4	4	4	✓	✓	✓	✓	✓	✓	Liu and Zou (2006)
CFN564	5	6	4	4	4	✓	✓	✓	X	✓	✓	Calvo et al. (2001)

$n\mathbf{f}^d$ indicates the number of implicit solves per step that involve the diffusive component $\mathbf{f}^d$, $n\mathbf{f}^a$ indicates the number of explicit stages that involve the advective component $\mathbf{f}^a$. Order denotes the theoretical order of accuracy where E is the order of the ERK method, I is the order of the DIRK method and A is the overall order of the IMEX-RK method. Stability denotes the type of stability property, A and L for each of the IMEX-RK methods. S.A. indicates whether the ERK or DIRK part of the method is stiffly accurate, $b_j = \hat{b}_j$ indicates if the DIRK and ERK methods have the same solution weights to compute the final assembly of $\mathbf{y}_{n+1}$ and $c_j = \hat{c}_j$ indicates if the DIRK and ERK methods are evaluated at the same stage time.

3.4 Diagnostics

There are certain diagnostic parameters to compute from the solution to obtain useful information and to check solution correctness. Next, we denote the spatial average over whole area of the annulus by $\langle \cdots \rangle$. For a field $f(r, \phi)$, it is given as

$$\langle f \rangle = \frac{1}{A} \iint_A f(r, \phi) r \mathrm{d}r \mathrm{d}\phi, \tag{3.145}$$

where, A is the total area of the annulus. Also, overbars $\overline{\cdots}$ indicate time average. We compute the Kinetic Energy (KE) at specified time intervals of the simulation. At a given instant in time, it is given by

$$\mathrm{KE} = \frac{1}{2} \langle u_r^2 + u_\phi^2 \rangle. \tag{3.146}$$

The Nusselt number (Nu) is the ratio of total heat flux to the heat carried by conduction. We calculate it at the inner and outer boundaries of the flow domain at specified time intervals of the simulation run. It is denoted as Nu_{r_i} and Nu_{r_o} respectively. The Nusselt number at the outer boundary is given as

$$Nu_{r_o} = \frac{(\frac{\mathrm{d}T}{\mathrm{d}r})_{r_o}}{(\frac{\mathrm{d}T_c}{\mathrm{d}r})_{r_o}} = \frac{\eta \log \eta (\frac{\mathrm{d}T}{\mathrm{d}r})_{r_o}}{1 - \eta}, \tag{3.147}$$

and the Nusselt number at the inner boundary is given as

$$Nu_{r_i} = \frac{(\frac{\mathrm{d}T}{\mathrm{d}r})_{r_i}}{(\frac{\mathrm{d}T_c}{\mathrm{d}r})_{r_i}} = \frac{\log \eta (\frac{\mathrm{d}T}{\mathrm{d}r})_{r_i}}{1 - \eta}. \tag{3.148}$$

The balance of both these Nusselt numbers indicates that thermal relaxation has been reached and it is a good indicator for convergence of the solution. Now, we define the Reynolds number (Re) which measures the ratio of inertial to viscous forces. With our choice of scales, it is given as the Root Mean Square (RMS) value of the velocities as

$$Re = \overline{\sqrt{\langle u_r^2 + u_\phi^2 \rangle}}. \tag{3.149}$$

The next diagnostic quantity we compute from the solution at specified time intervals is the power balance. From the solution, we check if the viscous dissipation balances the buoyancy power. This balance indicates that we satisfy the kinetic energy budget (Petschel et al., 2015). The expression for viscous dissipation is given as

$$\epsilon_U = \langle \mathbf{u} \cdot \nabla^2 \mathbf{u}) \rangle. \tag{3.150}$$

Using the vector identity $\boldsymbol{\nabla} \times \boldsymbol{\nabla} \times \mathbf{u} = \boldsymbol{\nabla}(\boldsymbol{\nabla} \cdot \mathbf{u}) - \nabla^2 \mathbf{u}$, the divergence constraint given in equation (2.24) and the vorticity definition, the viscous dissipation term now becomes

$$\epsilon_U = -\langle [\mathbf{u} \cdot (\boldsymbol{\nabla} \times \boldsymbol{\omega})] \rangle. \tag{3.151}$$

It can further be expanded and written as

$$\epsilon_U = -\left[\int_0^{2\pi}\int_{r_i}^{r_o} u_r \frac{\partial \omega}{\partial \phi} \mathrm{d}r\mathrm{d}\phi - \int_0^{2\pi}\int_{r_i}^{r_o} u_\phi \frac{\partial \omega}{\partial r} r\mathrm{d}r\mathrm{d}\phi\right]. \tag{3.152}$$

After performing integration by parts on the above equation, we end up with the following expression

$$\epsilon_U = -\left[\langle \omega^2 \rangle - 2\int_0^{2\pi} \left(u^2_{\phi_{r=r_o}} - u^2_{\phi_{r=r_i}}\right)\mathrm{d}\phi\right]. \tag{3.153}$$

Now, with regards to the boundary conditions, for stress-free boundaries, we can use equation (3.153) to calculate the viscous dissipation and for no-slip boundaries we use only the 1st term of equation (3.153). The buoyancy power is given by

$$P = \frac{Ra}{Pr}\langle u_r T \rangle. \tag{3.154}$$

We also define the kinetic energy spectra $E(m)$. It is represented as a function of the Fourier modes m as

$$E(m) = \frac{1}{2}\int_{r_i}^{r_o} [u_r^2(m,r) + u_\phi^2(m,r)] r\mathrm{d}r, \tag{3.155}$$

where, the velocities u_r and u_ϕ are represented in the Fourier-real space (m, r). The kinetic energy spectra is an useful measure as it can provide the Fourier mode at which the energy is maximum, which in turn can signify the number of convection rolls in the flow domain. Also, the plot of kinetic energy versus the number of Fourier modes can be an indication of the spectral convergence of the method. We also show the Chebyshev spectra for the variables T and ω to check if the solution is spatially resolved along the r-direction. More on Fourier-Chebyshev discretization will be discussed in the next chapter. Next, we define the error norms which are used to compute errors from the solution. We define a L^2 error norm and a L^2 relative error norm and denote them as error and relative error respectively. For temperaure T and the velocities u_r and u_ϕ, we define a combined relative error as

$$\text{relative}\|\epsilon\|_{L^2} = \left[\frac{\langle (T - T_R)^2 \rangle}{\langle T_R^2 \rangle} + \frac{\langle (u_r - u_{r_R}^2 \rangle}{\langle (u_{r\,R})^2 \rangle} + \frac{\langle (u_\phi - u_{\phi_R})^2 \rangle}{\langle u_{\phi_R}^2 \rangle}\right]^{1/2} \tag{3.156}$$

where subscript R indicates the reference value. As for the vorticity ω, we define a L^2 measure of the error to be

$$\|\epsilon\|_{L^2} = \sqrt{\frac{1}{A}\langle (\omega - \omega_R)^2 \rangle}. \tag{3.157}$$

Also, we will be using the same L^2 measure of the error for the radial velocity u_r.

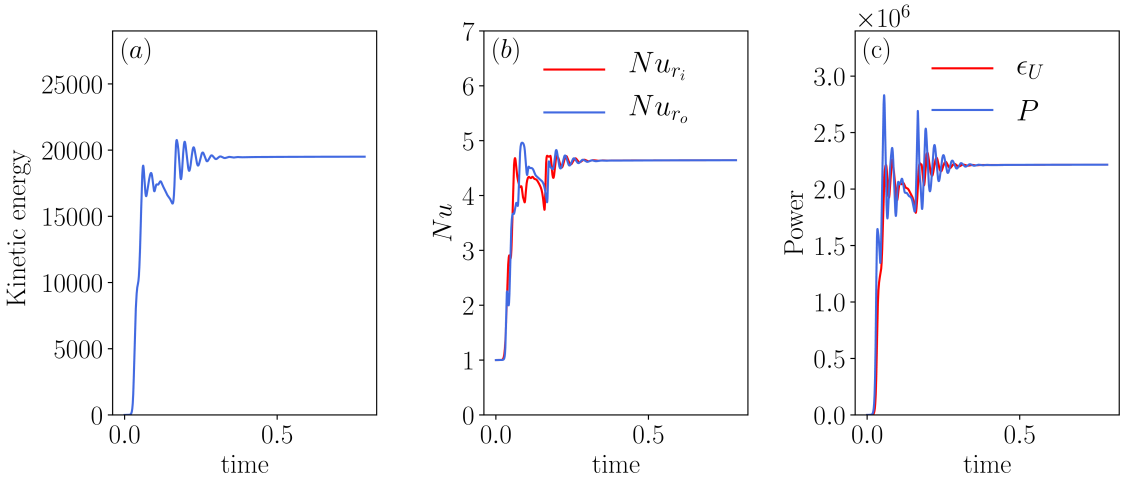

Figure 3.8: (a) Plot of kinetic energy as a function of time. (b) Plot of Nusselt numbers as a function of time. (c) Plot of the power as a function of time. The run was performed with $\eta = 0.3$, $Ra = 10^5$, $Pr = 1.0$, $N_r = 64$ and $N_m = 64$ with no-slip boundary conditions and with the SBDF4 time integration method.

3.5 Benchmark

The code for solving the problem using the aforementioned pseudospectral methods and time-stepping strategies was written from scratch in the Fortran programming language. The code contains several modules and subroutines where each module has specific dependencies. We perform a simulation with the parameters $Ra = 10^5$, $Pr = 1.0$, with no-slip boundary condition and with the SBDF4 time integration method. We then check for the convergence of the solution. Fig. 3.8 shows the diagnostic plots from the solution. From Fig. 3.8, plot (a) shows the saturation of the kinetic energy. Plot (b) indicates that the Nusselt number at the inner radius matches the Nusselt number at the outer radius implying that the thermal equilibrium has been reached. Plot (c) indicates that the power budget is satisfied as the viscous dissipation and the buoyancy power matches well and they attain a saturated value. From these plots we ascertain that solution is converged.

Furthermore, to benchmark the code with peer reviewed results, it is compared with a reference solution obtained by Alonso et al. (2000) who also studied the problem of thermal convection in an annulus. They performed their numerical simulations using spectral methods with a fixed radius ratio of $\eta = 0.3$ and Prandtl number $Pr = 0.025$. The benchmark simulation was performed for exactly the same set of parameters as in Alonso et al. (2000). Fig. 3.9 shows that the plot of Nu versus Ra coincide for the solution from the code and the reference. This match between the reference and the code is also shown in the Table 3.2. Furthermore, for the range of Rayleigh numbers shown in Fig. 3.9, we observe an oscillation of the solution about the periodic azimuthal direction. We performed a simulation with the parameters $Ra = 6500$, $Pr = 0.025$, $N_r = 32$, $N_m = 192$ from the reference Alonso et al. (2000) along with no-slip boundary condition and with

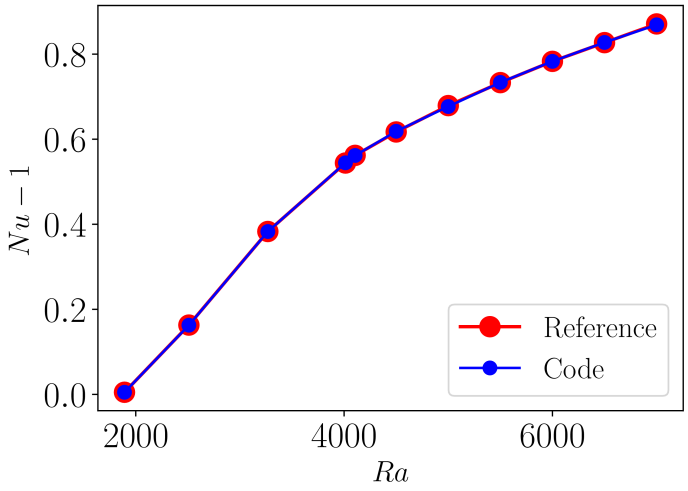

Figure 3.9: Plot of Nusselt number versus the Rayleigh number for the solutions from the code and the reference. The runs are performed with $\eta = 0.3$, $Pr = 0.025$, $N_r = 32$ and $N_m = 192$ with no-slip boundary conditions. We use the ARS443 time integration method for this analysis.

Table 3.2: Table for comparison of Nusselt numbers.

Ra	Reference $Nu-1$	Code $Nu-1$
1892	0.005	0.005
2510	0.163	0.162
3268	0.383	0.383
4013	0.544	0.544
4106	0.562	0.561
4500	0.617	0.618
5000	0.679	0.678
5500	0.733	0.733
6000	0.783	0.783
6500	0.827	0.827
7000	0.871	0.869

the ARS443 time integration method. We chose ARS443 randomly out of all the IMEX-RK methods to test if such methods are accurate enough. We obtained the frequency of oscillation to be $f = 5.15$ which exactly matched with the published value. Thus we ascertain that the code was benchmarked and ready to be used for the study of various time integration methods. The code is christened as **L**ight weight **F**ramework of time schemes for **A**nnular **Con**vection (**FALCon**). It will be submitted in an open source online repository in due time.

Chapter 4

Results

This chapter focuses on the description and detailed analysis of the results which are obtained after performing numerical simulations using the chosen set of time integration methods for the problem of thermal convection in an annulus.

4.1 Summary of cases

We perform multiple simulations with a wide range of input parameter values of the Rayleigh number Ra and the Prandtl number Pr. The 11 simulation cases that were performed are listed in Table 4.1 along with their respective input and output parameters. All the cases performed have no-slip boundary conditions. We analyzed the accuracy and performance of all the IMEX and IMEX-RK methods mentioned in the previous section for each case in Table 4.1. This enables us to study how the time integration methods perform in both strongly linear and strongly nonlinear flow regimes. A majority of seven cases are done with Prandtl number $Pr = 1$. Low Prandtl number simulations are of interest in the dynamo community since liquid metals usually have low Pr. Also, it is of interest in the solar dynamo community where $Pr \sim 10^{-3}$ (Chillà and Schumacher, 2012). Therefore, we perform three cases with $Pr = 0.025$. We perform one high Pr simulation with $Pr = 40$. We study the accuracy and the cost for all the 24 time schemes listed in Table 3.1 of the previous chapter. Therefore, we analyse in total 264 separate cases. For each of these cases, we study the accuracy and the cost for a wide range of timestep sizes Δt. On an average, we have a set of ten Δts for each case, and therefore, we perform $2,640$ simulation runs in total. Apart from these runs, we also perform a detailed stability analysis for a set of chosen time schemes which are applied to all the cases shown in Table 4.1. That is a separate study which will be shown in due course of this chapter.

Table 4.1: Summary table of numerical simulations performed.

Case	Ra	Pr	N_m	N_r	BC	Re	Nu	ϵ_U	P
0	2×10^3	1.00	36	36	NS	2.77	1.15	1.89×10^3	1.89×10^3
1	1×10^4	1.00	48	48	NS	18.33	2.51	9.30×10^4	9.28×10^4
2	1×10^7	40.0	256	256	NS	27.34	12.68	4.40×10^5	4.40×10^5
3	1×10^5	1.00	64	64	NS	75.28	4.54	2.21×10^6	2.21×10^6
4	1×10^6	1.00	128	96	NS	273.61	8.07	4.20×10^7	4.19×10^7
5	1×10^4	0.025	192	64	NS	547.63	2.11	1.13×10^8	1.11×10^8
6	1×10^7	1.00	160	128	NS	937.34	13.86	7.23×10^8	7.24×10^8
7	1×10^5	0.025	320	128	NS	2205.39	4.27	3.15×10^9	3.15×10^9
8	1×10^8	1.00	256	256	NS	3586.14	22.95	1.33×10^{10}	1.33×10^{10}
9	1×10^6	0.025	384	160	NS	6733.84	6.16	5.14×10^{10}	5.27×10^{10}
10	1×10^9	1.00	384	384	NS	13159.46	44.60	2.59×10^{11}	2.60×10^{11}

All simulation runs are performed with radius ratio of $\eta = 0.35$, with constant acceleration due to gravity and no-slip boundary conditions. All the 24 time integration methods are analysed for all the cases shown here.

Although our focus is to analyse the time integration methods, we need to make sure that the physics of the solution is technically sound. Therefore we will first discuss the solution physics in the following section.

4.1.1 Solution physics

We will now show the solution physics for three different Rayleigh numbers, $Ra = 10^7$, $Ra = 10^8$ and $Ra = 10^9$ with $Pr = 1$. We choose this particular set of input parameters because these three Rayleigh numbers are the largest ones among all the simulation cases shown in Table 4.1. Also, they are one order of magnitude apart which means that the physical solution for each of the three cases are different. We first show the plots of the time averaged solution spectra in the Fourier and Chebyshev space for the three cases in Fig. 4.1. Plot (a) of Fig. 4.1 shows the kinetic energy spectra as a function of the azimuthal wavenumber. The kinetic energy peaks at $m = 3$ for the three Rayleigh numbers. As the number of Fourier modes increases, we see that the kinetic energy spectra shows almost five decades of decay in amplitude. This indicates that the number of grid points in the azimuthal direction suffices. Coming to plot (b) and plot (c) of Fig. 4.1, it shows the normalized Chebyshev spectra of temperature and vorticity respectively as a function of the number of Chebyshev modes for one given azimuthal wavenumber. We see at least six orders of fall for the coefficients as the number of Chebyshev modes increases. This indicates that we have enough grid points along the radial direction. Therefore, we are assured that the solution is finely resolved. This enables us to focus on errors in the time discretization part.

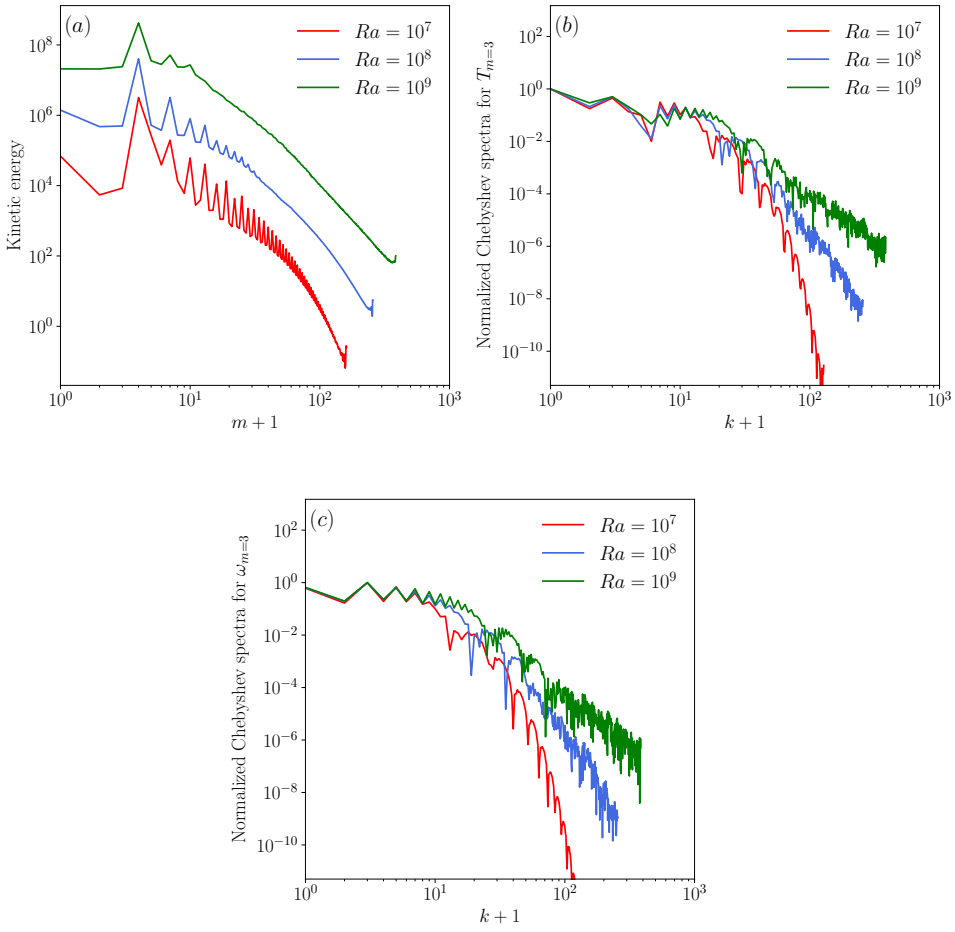

Figure 4.1: (a) Kinetic energy spectra with maximum energy for the three highest Rayleigh numbers. (b) Normalized Chebyshev spectra of T scaled with respect to their maximum value at $m = 3$ for three highest Rayleigh numbers. (c) Normalized Chebyshev spectra of ω scaled with respect to their maximum value at $m = 3$ for three highest Rayleigh numbers. All the plots shown are for $Pr = 1$ and are averaged in time.

We also checked the balance of Nusselt number Nu at the inner and outer boundaries for all the cases and we make sure they balance each other when we attain saturation of kinetic energy. This indicates that the thermal equilibrium has been reached. Also, we check the balance of viscous dissipation and buoyancy power. They are of the same order of magnitude for all the cases indicating that the power budget has been satisfied. For example, for case 10, $\epsilon_U = 2.59 \times 10^{11}$ and $P = 2.60 \times 10^{11}$ indicating that they are of same order of magnitude. Now, to visualize the solution, we show the contour plots of the temperature (T) and the vorticity (ω) overplotted with the streamlines, for the three Rayleigh numbers $Ra = 10^7$, $Ra = 10^8$ and $Ra = 10^9$ in Fig. 4.2. From the figure, we can observe that, as the Rayleigh number increases, the flow becomes more and more chaotic referring to the time dependency of the system, indicating that the nonlinear terms in

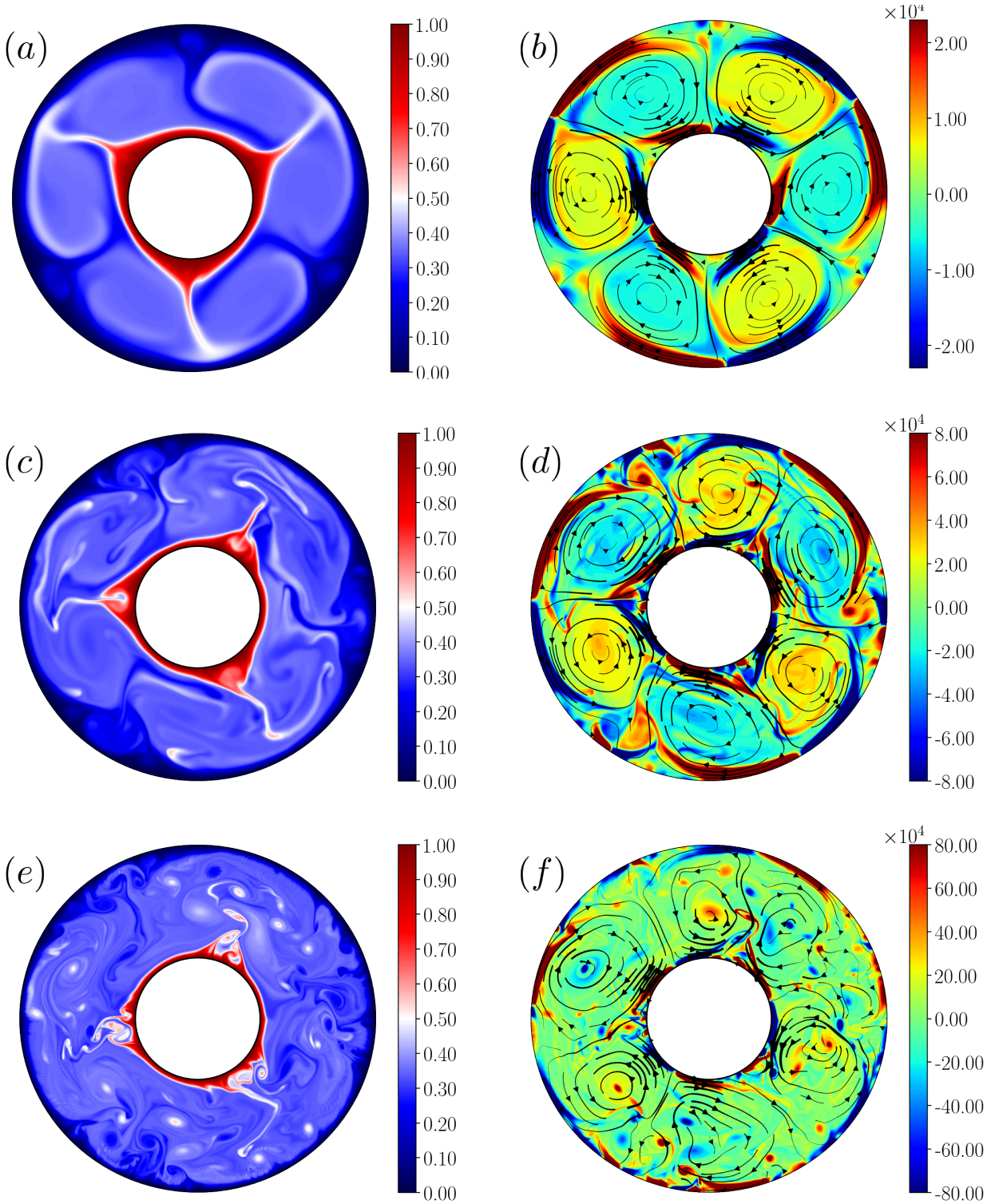

Figure 4.2: Left panel of figures indicate contour plots of total temperature T. Right panel of figures indicate contour plots of vorticity ω and the streamlines in the flow. The plots in the top row are for $Ra = 10^7, Pr = 1$, the plots in the middle row are for $Ra = 10^8, Pr = 1$ and the plots in the bottom row are for $Ra = 10^9, Pr = 1$.

the governing equations become stronger and the Reynolds number Re increases. We also note from Fig. 4.2 that the three temperature plumes shown in the left panel of figures indicate three upwellings and three downwellings of convection cells which are shown in

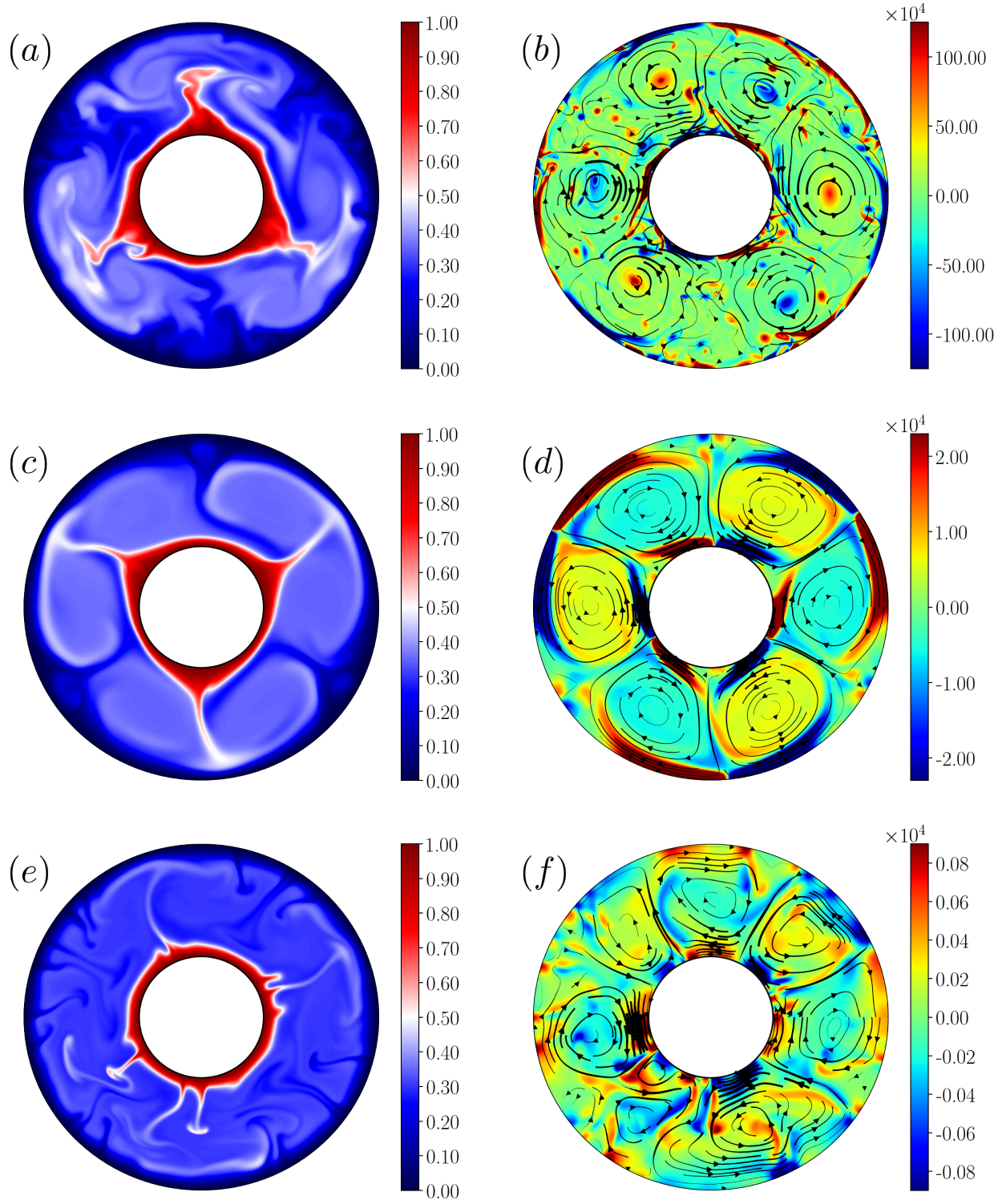

Figure 4.3: Left panel of figures indicate contour plots of total temperature T. Right panel of figures indicate contour plots of vorticity ω and the streamlines in the flow. The plots in the top row are for $Ra = 10^7, Pr = 0.025$, the plots in the middle row are for $Ra = 10^7, Pr = 1$ and the plots in the bottom row are for $Ra = 10^7, Pr = 40$.

the vorticity plots along the right panel of figures. The three plumes also correspond to the maximum kinetic energy observed at $m = 3$ in plot (a) of Fig. 4.1. Also, the thermal boundary layers and plume width decreases when Ra increases. For illustrative purposes

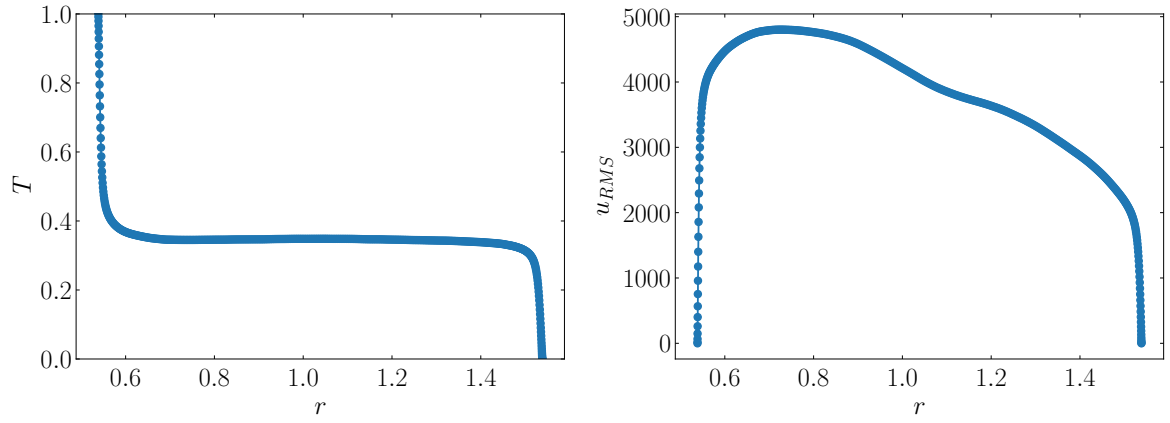

Figure 4.4: (a) Time and azimuthally-averaged temperature along the radius r. (b) Time average of the RMS value of velocity along the radius r.

we also show the contour plots for three different Prandtl numbers for $Ra = 10^7$. We observe from Fig. 4.3 that, for $Pr = 0.025$ and $Pr = 1$, the number of temperature plumes is three which corresponds to maximum kinetic energy at $m = 3$. Comparing the solution for these two Prandtl numbers, due to the presence of strong mean flow and large inertia in the case of $Pr = 0.025$, the Reynolds number is already of order 10^4 whereas for the case with $Pr = 1$, the Re is of order close to 10^3. The bottom row of Fig. 4.3 shows the contour plots for $Pr = 40$. For such large Pr, the momentum diffusivity is higher than the thermal diffusivity and the resulting Reynolds number is $Re = 27.34$ which is very low in this case as compared to a case with $Pr = 1$ where $Re = 937.34$.

Next, in Fig. 4.4, we show the time and azimuthally-averaged temperature and RMS value of velocity as a function of radius for the highest Reynolds number (case 10: $Ra = 10^9$, $Pr = 1$, $Re = 13159.46$). The thermal boundary layer scales inversely to the Nusselt number and the viscous boundary layer scales inversely to the square of the Reynolds number (Scheel and Schumacher, 2014) which makes

$$\delta_T \approx \frac{1}{Nu}, \text{ and } \delta_u \approx \frac{1}{\sqrt{Re}}, \tag{4.1}$$

where, δ_T and δ_u are the thermal and viscous boundary layer thicknesses. In our case we have more than 10 points in these boundary layers. We make sure all the simulation cases are well resolved along the boundaries.

4.1.2 Nusselt and Reynolds number scalings

There are two input parameters to the problem, the Rayleigh number Ra and the Prandtl number Pr. The two important output parameters we measure are the Nusselt number Nu and the Reynolds number Re which were defined in chapter 2. It is of interest to obtain the quantitative behaviour of Nu, Re as a function of Ra and Pr e.g. (Johnston

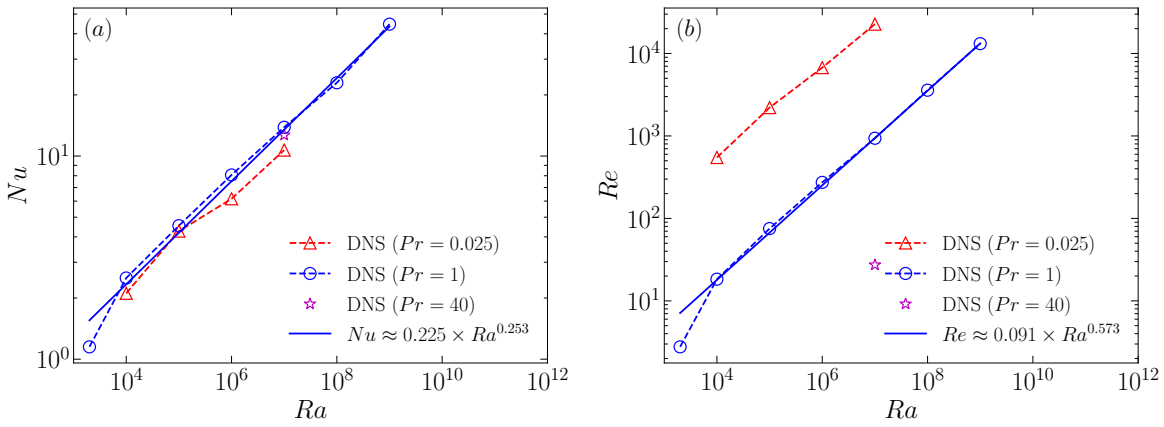

Figure 4.5: (a) Nusselt number versus Rayleigh number. (b) Reynolds number versus Raleigh number.

and Doering, 2009; Gastine et al., 2015). Since we have performed simulations with a wide range of Rayleigh numbers, it is only natural to study the scalings of Nusselt and Reynolds numbers with respect to the Rayleigh number. We restrict our attention in this case to $Pr = 1$ to obtain a scaling law since for this value of Pr, we have maximum number of cases using which we can obtain a reasonable scalings of Nu and Re. Plot (a) of Fig. 4.5 shows the plot of Nu versus Ra for three different values of Pr. We obtain a fit for the Nusselt number as $Nu \approx 0.225 \times Ra^{0.253}$ for $Pr = 1$. Plot (b) of Fig. 4.5 shows the plot of Re versus Ra for three different values of Pr. For $Pr = 1$, we obtain a fit for the Reynolds number as $Re \approx 0.091 \times Ra^{0.573}$. In Fig. 4.6, we also plot the compensated Nusselt and Reynolds numbers as a function of the Rayleigh number. We see from plot (a) of Fig. 4.6 that, as Ra increases, Nu tends to scale as $Nu \sim Ra^{1/3}$. Also, the solution from the code closely matches with the reference (Zhu et al., 2018). Plot (b) of Fig. 4.6 shows the compensated Re as a function of Ra. From the literature (Grossmann and Lohse (2000)), we expect Re to scale as $Re \sim Ra^{1/2}$. Gastine et al. (2015) verified this scaling law in their results where they showed that the compensated Re initially rises for increasing Ra and dips at a large value of Ra. We expect the plot to go down for higher Ra than shown. We also observe that Nu does not change considerably for different values of Pr. However, Re varies by at least one order of magnitude for different values of Pr we tried, keeping the same value of Ra. This is because, the low value of $Pr = 0.025$ makes way for a strong inertia and mean flow in the system compared with a system with $Pr = 1$ for the same Ra value and hence the flow is more nonlinear and hence the increase in Re. On the other hand, $Pr = 40$ makes the system more linear for the same value of Ra which results in smaller value of Re.

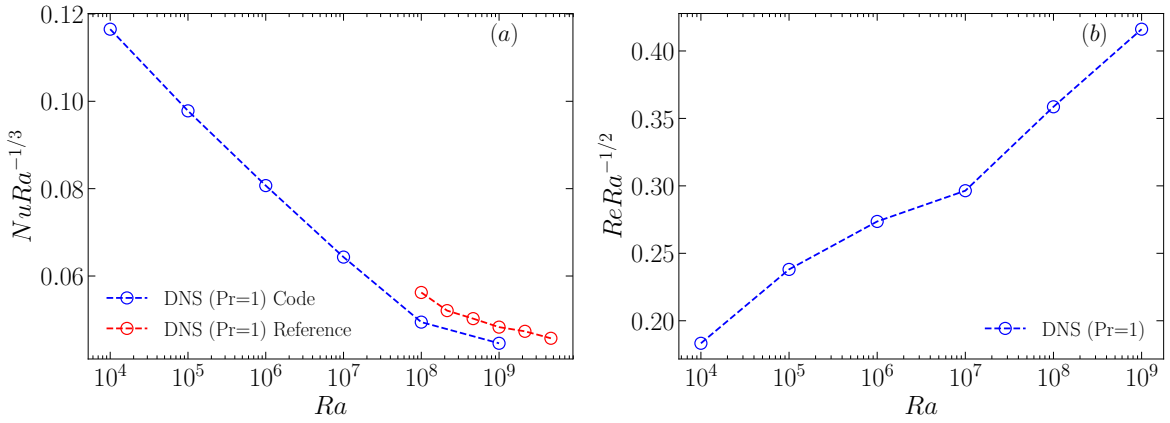

Figure 4.6: (a) Compensated Nusselt number versus Rayleigh number shown for the solutions from the code and the reference. (b) Compensated Reynolds number versus Rayleigh number.

4.2 Time integration: Accuracy and cost

We now proceed to analyse all the time schemes given in Table 3.1 for their accuracy and cost. Accuracy is measured by the L^2 error and relative error as defined in chapter 3 and cost is the amount of physical time taken in seconds. We use constant timestep sizes for this analysis. We make sure we use enough grid resolution so that no major spatial errors creep in to the temporal errors which we are interested here. An illustration of how we calculate the accuracy and the cost is shown in Fig. 4.7. Regarding the accuracy plots, for scheme$_A$, for each Δt, we plot a point which corresponds to the calculated error. After plotting for all the timestep sizes from the largest to the smallest ones, we will obtain the convergence rate or slope of the time scheme. The maximum timestep size shown for each method is the timestep size above which the solution blows up. For our analysis, we use the first three points of the accuracy curve for each scheme to calculate the slope, which is shown in red color in the schematic. As for the cost calculation, the cost$_{A_1}$ shown in the schematic for e.g. can be determined as

$$\text{cost}_{A_1} = \frac{C_1 \times \text{time duration of simulation}}{\Delta t_{A_1}}, \tag{4.2}$$

where, C_1 is the physical time taken for one iteration using the corresponding time scheme. Therefore, for the given time scheme, for each Δt we calculate the associated error and the cost and plot them. To illustrate the selection of the best performing time scheme, we show the blue continuous line which shows the required accuracy or the level of the error we require. We see that it first intersects along the cost curve of scheme$_A$ and correspondingly we obtain the lowest cost possible among the two schemes shown. Therefore, scheme$_A$ shows better performance than scheme$_B$.

We will select two important cases out of the dataset for illustrative purposes and compare all the time schemes with each other. We use the Reynolds number Re to select the two

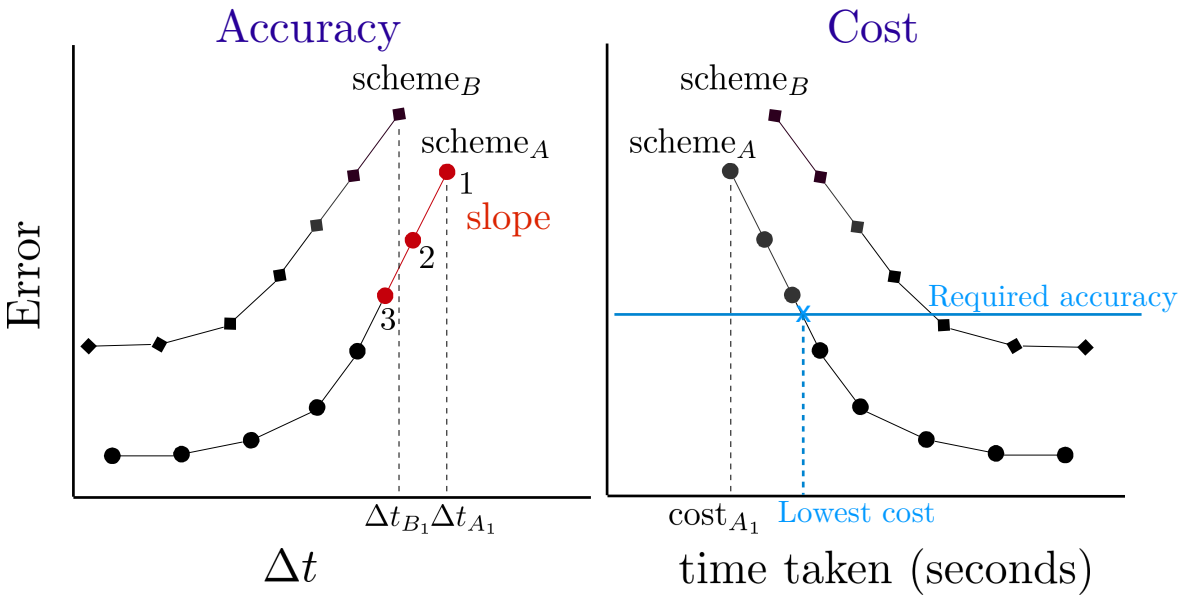

Figure 4.7: Schematic illustration of the calculation of accuracy and cost for each time scheme.

cases because Re is a good measure of whether a flow regime is linear or nonlinear. As we mentioned in the introduction, our goal is to find good time integration methods which can help us reach extreme parameter regimes. Therefore, our focus is on the highest Re simulation which is a strongly nonlinear one. We will therefore select a strongly nonlinear case and a more laminar case and discern the differences thereafter. We will show the accuracy and cost of the variables (T, u_r, u_ϕ) using the combined measure of the relative error as defined in equation (3.156) of chapter 3. After that, we will show the accuracy comparisons of the two variables ω and u_r using the relative error measure. These two variables are the differential and algebraic variables of our mixed DAE system of equations. We will now proceed to describe the results for the strongly nonlinear case first.

4.2.1 Case 10 : $Ra = 10^9, Pr = 1$

The output variables from the simulation are always in the Fourier-real (m, r) space. The reference solution with which we measure the solution error is computed using the SBDF4 method with the smallest Δt considered, $\Delta t = 5 \times 10^{-10}$. For the variables T, u_r and u_ϕ, we use the combined relative error measure defined in equation (3.156) of chapter 3. For each of the time integration methods used, we show in Fig. 4.8, the convergence rates in the left figure panels and their corresponding cost in the right figure panels. The top left panel of Fig. 4.8 indicates the convergence rate for all the second order methods and we observe that all the IMEX-RK methods are more accurate than the IMEX methods CNAB2 and SBDF2. Especially, ARS232 is more accurate than all the others and it is almost three orders of magnitude more accurate than the IMEX methods. Also important to note here is the super-convergent nature of ARS232 for a range of

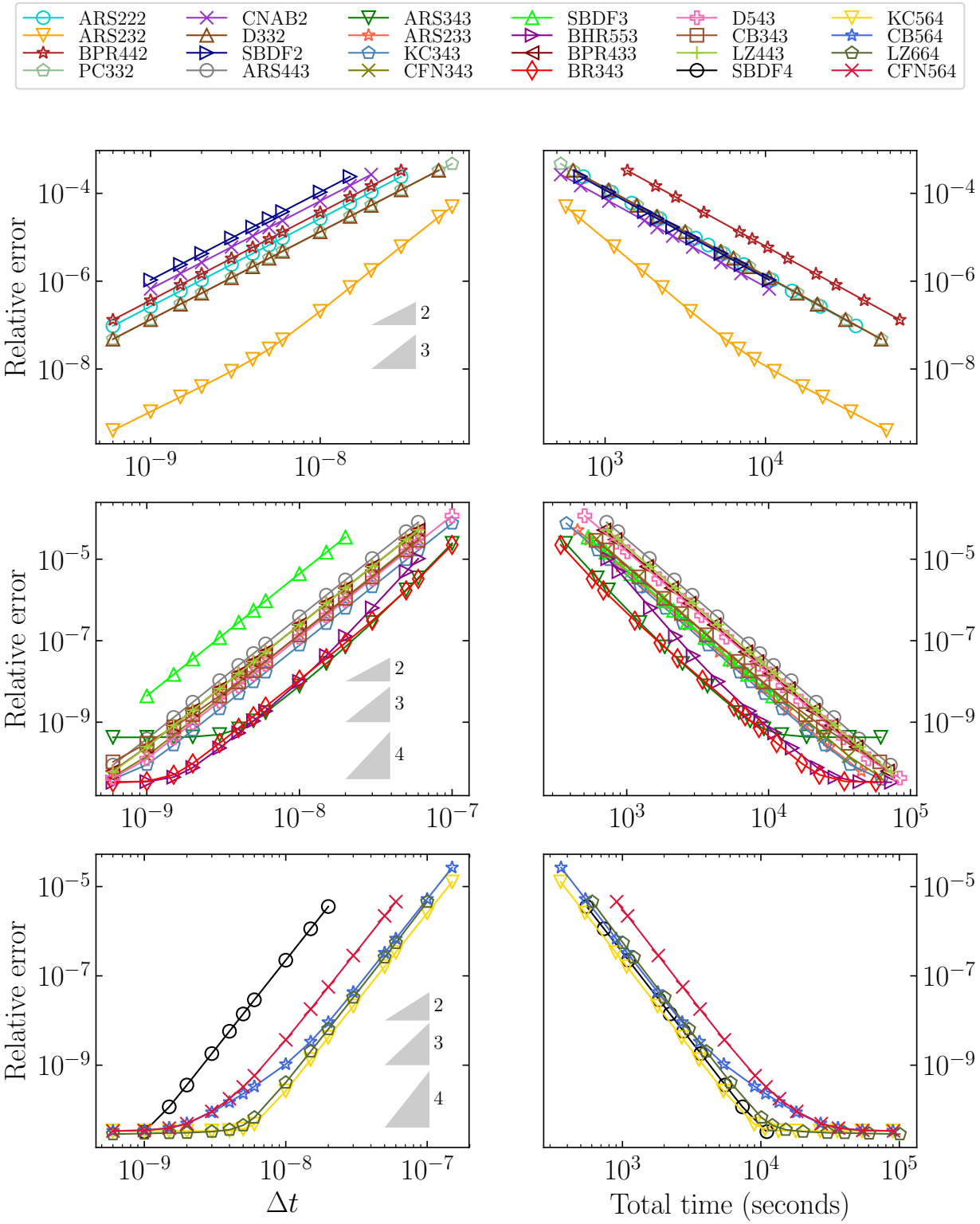

Figure 4.8: The top left, middle left and lower left hand panels show the combined relative error of T, u_r and u_ϕ as a function of timestep size Δt for the second, third and fourth order methods respectively; the top right, middle right and lower right hand panels show the same error as a function of phyisical time taken in seconds for second, third and fourth order methods respectively. All plots shown in this figure are based on case 10 ($Ra = 10^9$, $Pr = 1$).

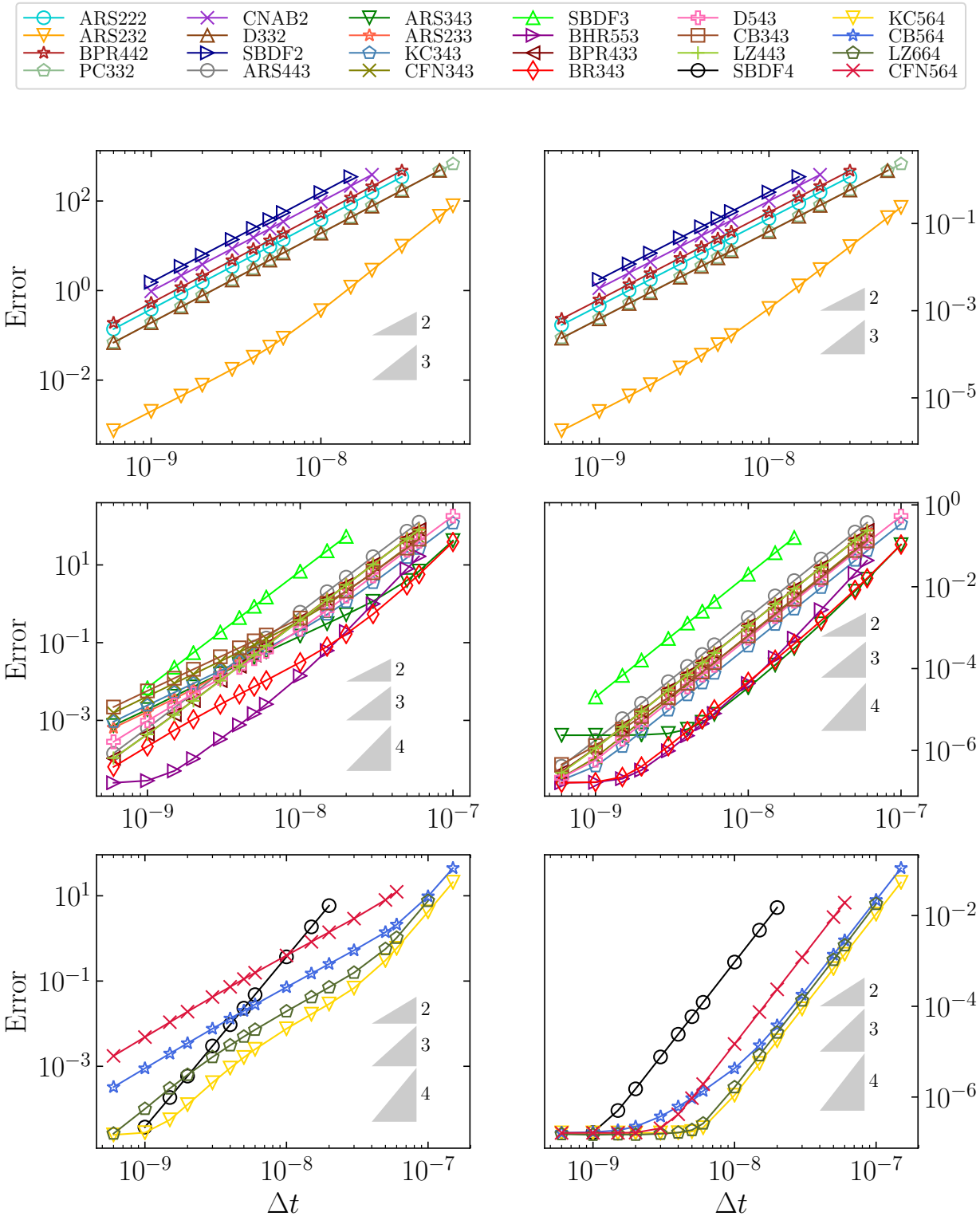

Figure 4.9: The top left, middle left and lower left hand panels show the error in ω as a function of timestep size Δt for the second, third and fourth order methods, respectively; the top right, middle right and lower right hand panels show the same error in u_r as a function of timestep size Δt for the second, third and fourth order methods, respectively. All plots shown in this figure are based on case 10 ($Ra = 10^9$, $Pr = 1$).

larger timestep sizes, meaning it performs better than its theoretical order. Regarding the cost, supposing we fix the required error threshold to be 10^{-7}, ARS232 outperforms all the other second order methods. This was also reported by Kennedy and Carpenter (2003) where they found the ARS232 method to be substantially more efficient than the SBDF2 method. Among the third order methods shown in the middle panel of Fig. 4.8, all the IMEX-RK methods are more accurate than the IMEX method SBDF3. The most accurate methods among the IMEX-RK methods are the ARS343, BR343 and BR553 methods with BR553 being super-convergent for a range of larger timestep sizes. This is in agreement with Grooms and Julien (2011) who also reported super-convergent nature of the BR553 method. Regarding the cost, ARS343 and BR343 perform best among the third order methods. Turning to the fourth order methods shown in the bottom panel of Fig. 4.8, the IMEX-RK methods are more accurate again compared to their IMEX counterpart SBDF4 and with respect to the cost, the method KC564 performs best and it is in fact as cheap as the third order methods ARS343 and BR343. We also observe that all methods shown in the left figure panels of Fig. 4.8 retain their theoretical convergence rates. Next, we show the accuracy plots for ω and u_r in Fig. 4.9. We use the error measure defined in equation (3.157) of chapter 3 for ω and u_r. We choose to show these plots for ω and u_r separately because, they are part of the DAE system where ω is the differential variable and ψ the algebraic variable. We show u_r as the algebraic variable in this plot since it is directly proportional to ψ after discretization along the azimuthal direction using Fourier expansion. For comparison, among the second and third order methods shown in the top and middle panels of Fig. 4.9, we observe that all the methods maintain their theoretical convergence rate and none of them suffer from order reduction. In fact, BR553 shows super-convergence for both the differential and the algebraic variables. Turning to the bottom panel of Fig. 4.9, among the fourth order methods however, we see that CFN564 suffers order reduction for the differential variable ω and the rest of the fourth order methods suffer order reduction for ω only at smaller timestep sizes. None of the fourth order methods suffer order reduction for the variable u_r. Grooms and Julien (2011) also performed analysis of the time schemes which we used, such as SBDF3, SBDF4, ARS343, CFN343, KC343, BR553 and KC564 for 2D Boussinesq convection in a periodic box and they found the IMEX-RK time schemes to be better performing than the IMEX multistep time schemes SBDF3 and SBDF4. We are also in agreement for this case. We know that this case has the highest Reynolds number. So, naturally one wants to know what happens at lower Re. Therefore, we next show the case which has a strongly linear flow regime.

4.2.2 Case $4 : Ra = 10^6, Pr = 1$

We show the accuracy and cost plot for the variables T, u_r, u_ϕ as a combined measure of the relative error in Fig. 4.10. The reference solution with which we measure the solution error is computed using the SBDF4 method with $\Delta t = 5 \times 10^{-8}$. The top left panel of Fig. 4.10 indicates the convergence rate for all the second order methods and we observe

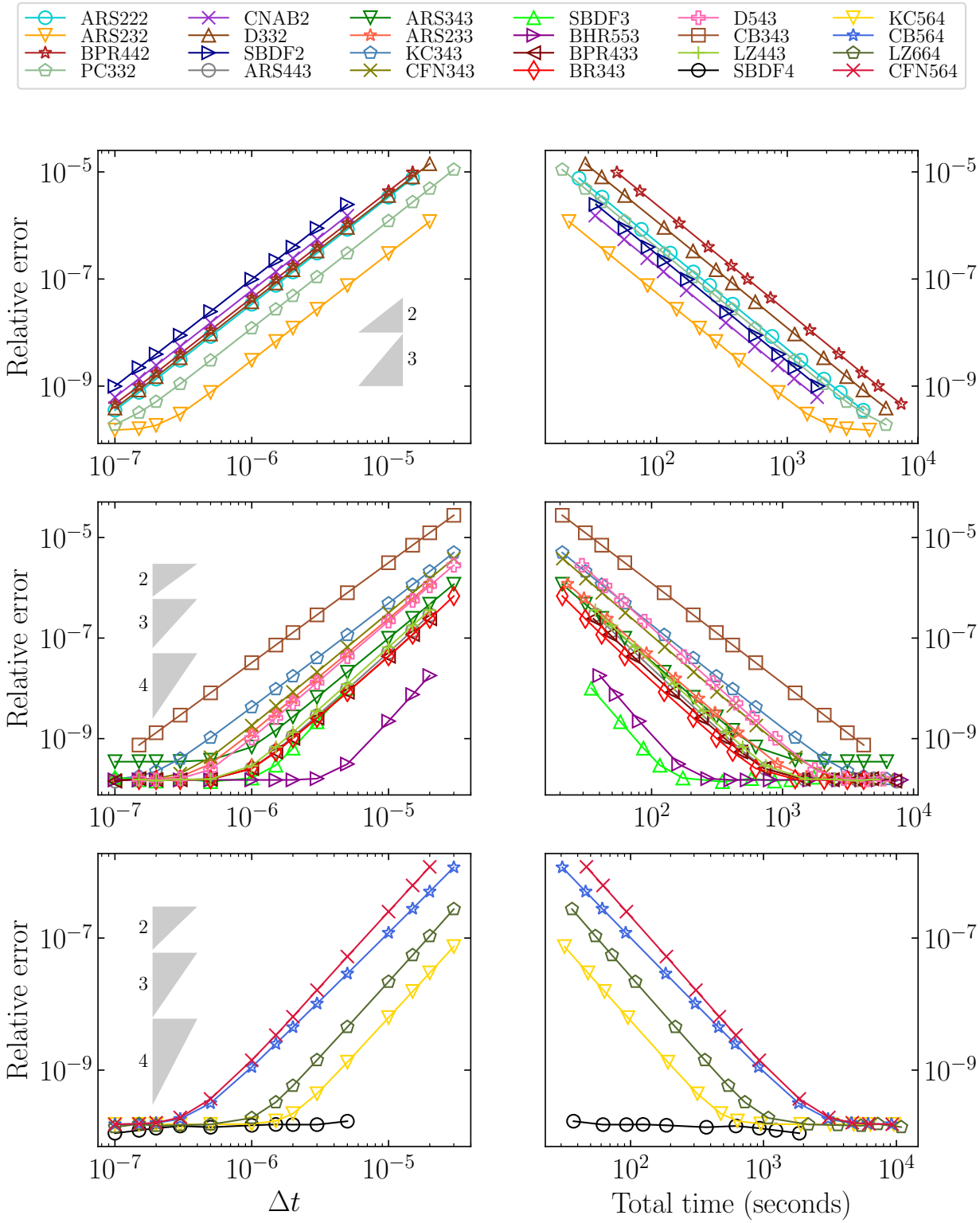

Figure 4.10: The top left, middle left and lower left hand panels show the combined relative error of T, u_r and u_ϕ as a function of timestep size Δt for the second, third and fourth order methods respectively; the top right, middle right and lower right hand panels show the same error as a function of physical time taken in seconds for second, third and fourth order methods respectively. All plots shown in this figure are based on case 4 ($Ra = 10^6$, $Pr = 1$).

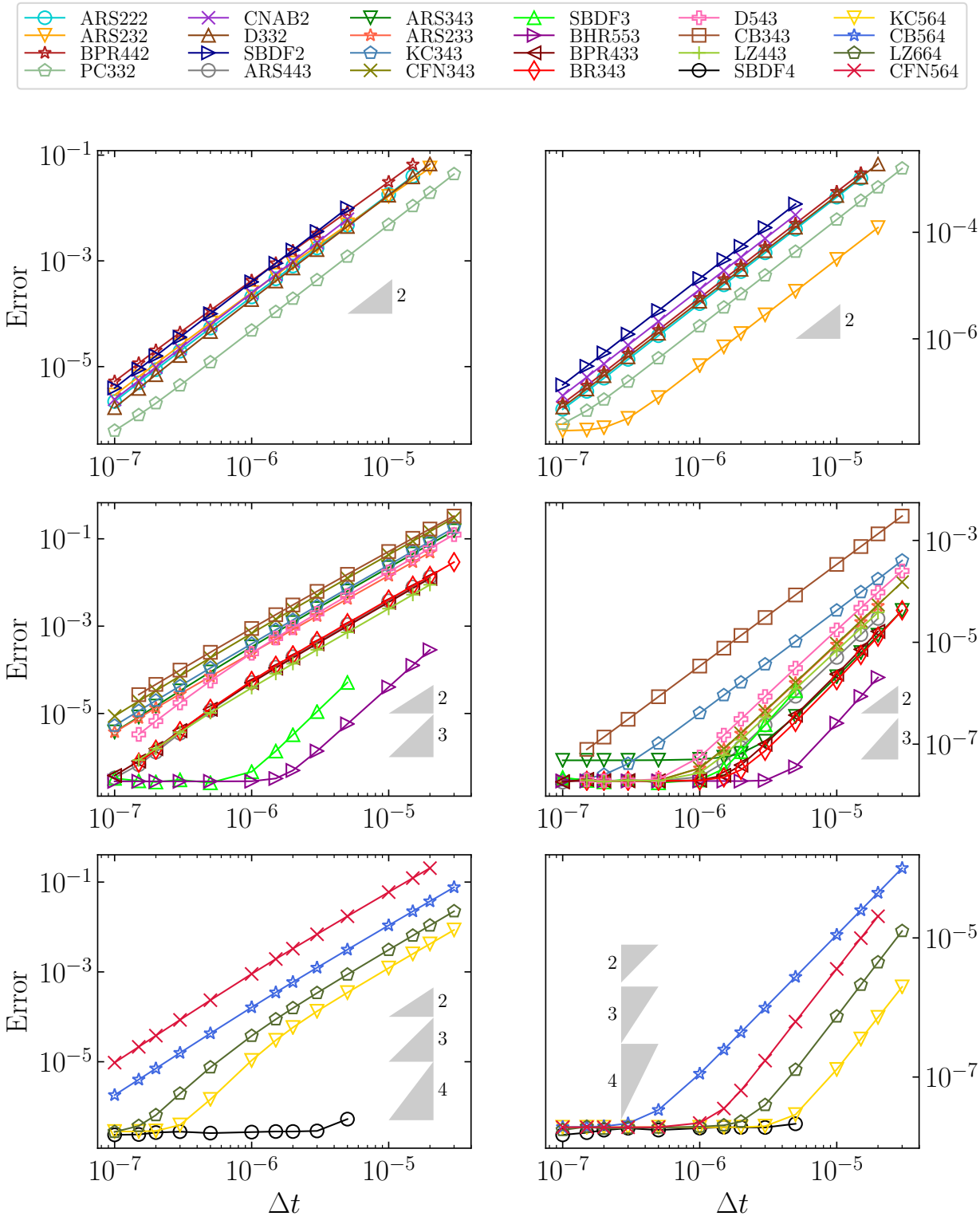

Figure 4.11: The top left, middle left and lower left hand panels show the error in ω as a function of timestep size Δt for the second, third and fourth order methods respectively; the top right, middle right and lower right hand panels show the same error in u_r as a function of timestep size Δt for the second, third and fourth order methods respectively. All plots shown in this figure are based on case 4 ($Ra = 10^6$, $Pr = 1$).

that all the IMEX-RK methods are more accurate than the IMEX methods CNAB2 and SBDF2. With respect to the cost, we see that the method ARS232 outperforms all other methods for any value of the required accuracy. This performance is similar to that of the strongly nonlinear case 10 which was shown in the previous subsection. Among the third order methods, we now see that all the IMEX-RK methods except for BR553 are less accurate than SBDF3 whereas in the strongly nonlinear case shown before, all IMEX-RK cases were more accurate than the SBDF3 method. Regarding the cost, for high values of error, say 10^{-7}, BR343 and ARS343 perform best. However, if the required error is low, say 10^{-9}, the IMEX multistep method SBDF3 outperforms all the other methods. Among the fourth order methods, we see that all methods except SBDF4 show drop in convergence rates meaning that the slopes are lower than their theoretical convergence rates. Therefore, we witness a phenomenon of order reduction here. We could not measure the slope for SBDF4 as the error for the highest Δt was very low. However, it remains the most accurate method among the fourth order methods. From the plot for the cost, we observe that SBDF4 remains the best method as the error for the highest Δt was very low. Next, we compare the accuracies of the differential and algebraic variables for this case. We show the plots of convergence rates in Fig. 4.11 for ω and u_r for case 4. Among the second order methods shown in the top panels of Fig. 4.11, none of them suffer from order reduction. Among the third order methods shown in the middle panel of Fig. 4.11, we observe except for SBDF3 and BR553 all the other third order methods suffer from order reduction for ω. BR553 is a fairly new method derived in Boscarino and Russo (2009). It satisfies additional order conditions for index-2 DAE which ensure that it never suffers from order reduction. For the variable u_r however, the methods CB343 and KC343 suffer order reduction but none of the other methods do. As we shall see in the next section, this case is a stiff problem. For such stiff cases, Kennedy and Carpenter (2003) does a cost comparison of the IMEX-RK methods KC343, KC564 with the IMEX multistep methods SBDF3 and SBDF4 for a Van der Pol's equation. They found that the IMEX multistep methods SBDF3 and SBDF4 perform better than KC343 and KC564 methods. For this case, we are also in agreement with them. Also, among the fourth order methods shown in the bottom panel of Fig. 4.11, all of them suffer order reduction for ω. Regarding u_r, all the methods suffer milder order reduction as compared to ω. Thus we see this phenomena of order reduction prevalent among more number of IMEX-RK methods in low Re situations. More analysis on this is shown in the next subsection.

4.2.3 Order reduction

The phenomenon of order reduction is well known for IMEX-RK methods when applied to index-2 DAEs (Boscarino, 2007; Kennedy and Carpenter, 2003; Ascher and Petzold, 1998). It is said that when the problem is stiff an order reduction phenomenon is observed (Boscarino, 2007). So, before we proceed to show the observed order reduction of the methods, we need to understand what stiffness means for our problem. We estimate

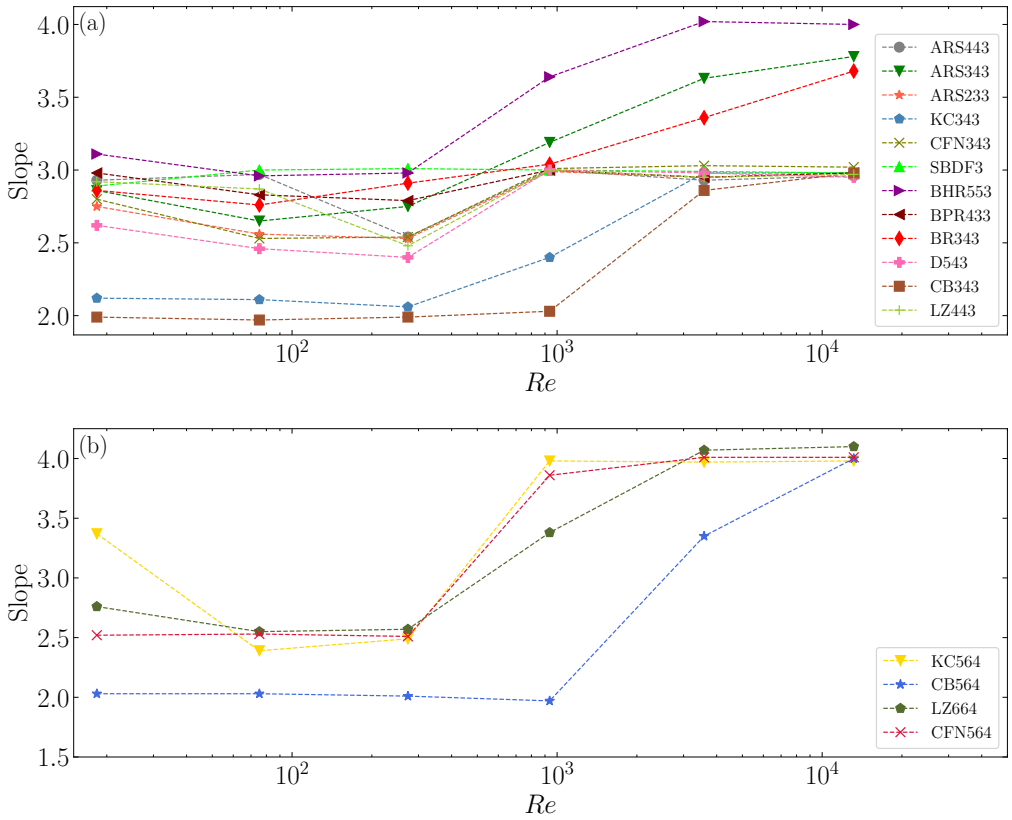

Figure 4.12: (a) Slopes of the third order methods as a function of the Reynolds number Re. (b) Slopes of the fourth order methods as a function of the Reynolds number Re. Both plots indicate slopes for radial velocity u_r and $Pr = 1$.

stiffness as

$$\sigma = \frac{1}{\epsilon}, \tag{4.3}$$

where,

$$\epsilon = \frac{\tau_{\mathrm{L}}}{\tau_{\mathrm{NL}}} \sim Re_{\Delta x}. \tag{4.4}$$

τ_{L} is the linear (diffusive) time scale and τ_{NL} is the nonlinear (advective) time scale and $Re_{\Delta x}$ is the grid Reynolds number. Thus, when $\sigma \gg 1$, a fully explicit treatment in time becomes expensive and we opt for an implicit treatment of the diffusive components of the equation, which we do in the case of the IMEX methods that we explore. The stiffness can also be measured macroscopically using the global Reynolds number which is the ratio of diffusive to advective timescales which emperically means

$$\sigma \sim \frac{1}{Re} \tag{4.5}$$

In our simulation runs, we observe order reduction for the differential variable ω and the algebraic variable ψ. As for the algebraic variable, we look at u_r since it is just ψ multiplied

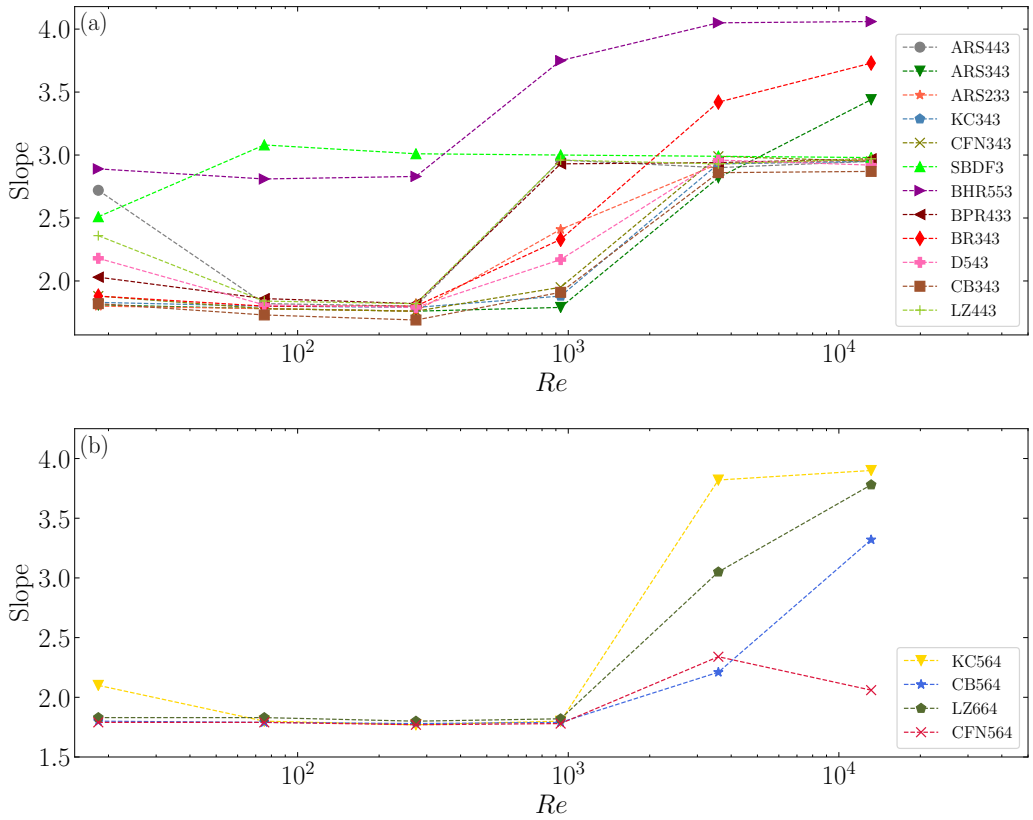

Figure 4.13: (a) Slopes of the third order methods as function of the Reynolds number Re. (b) Slopes of the fourth order methods as function of the Reynolds number Re. Both plots indicate slopes for vorticity ω and $Pr = 1$.

by the Fourier mode by its definition. So, we plot the convergence rates or slopes for ω and u_r for increasing values of Re using the same L^2 error measure introduced before in equation (3.157) in chapter 2. The slopes are measured from the accuracy plots using the first three points starting with the largest Δt. Fig. 4.12 shows the slope versus Re for the radial velocity u_r at $Pr = 1$. From plot (a) of Fig. 4.12, we observe that except for KC343 and CB343, none of the other methods suffer significant order reduction, meaning they always have slopes > 2.5 and they suffer this milder order reduction only till $Re = 1000$ after which they reach their theoretical convergence rates. The IMEX method SBDF3 never suffers from order reductions. We also notice that the IMEX-RK method BR553 never suffers order reduction as it satisfies appropriate order conditions to maintain its theoretical order in stiff regimes (Boscarino and Russo, 2009). In fact, for $Re > 10^3$, BR553 along with BR343 and ARS343 shows super-convergence (Grooms and Julien, 2011). From plot (b) of Fig. 4.12, we observe that, all the fourth order methods suffer order reduction untill $Re = 300$ except CB564, after which the slopes increase as Re increases. For CB564, the slope starts to show an increase at $Re = 1000$. All the fourth order schemes attain their theoretical convergence rates for the highest value of Re.

As for the vorticity, plot (a) and plot (b) of Fig. 4.13 show the slopes for the third order and fourth order methods respectively. Among the third order methods, we notice

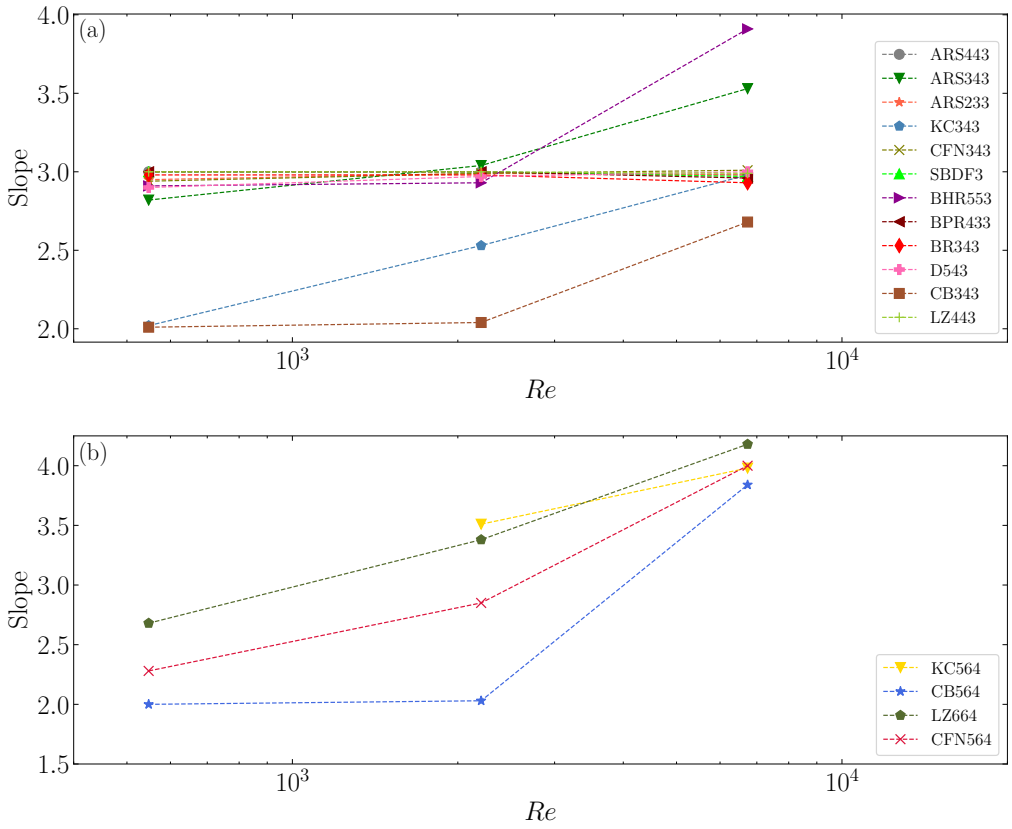

Figure 4.14: (a) Slopes of the third order methods as function of the Reynolds number Re. (b) Slopes of the fourth order methods as function of the Reynolds number Re. Both plots indicate slopes for radial velocity u_r and $Pr = 0.025$.

the IMEX-RK methods showing order reduction until $Re \sim 300$ above which the slopes start to increase. The IMEX method SBDF3 and the IMEX-RK method BR553 never suffers order reduction and for $Re > 10^3$ and BR553, along with BR343 and ARS343 shows super-convergence. Among the fourth order methods, KC564, CB564 and LZ664 show order reduction till $Re \sim 1000$ and thereafter the slopes increase as Re increases. However, CFN564 always shows order reduction for any Re. For the $Pr = 0.025$ cases, we plot the slope versus Re for u_r in Fig. 4.14. Among the third order schemes, we observe that none of the time schemes except KC343 and CB343 suffer from order reduction and all of them attain the theoretical convergence rates at the highest value of Re. Among the fourth order methods, all of them suffer order reduction except for the highest value of Re. As for the vorticity, we plot the slope versus Re in Fig. 4.15. Among the third order methods, we observe from plot (a) of Fig. 4.15 that SBDF3 and BR553 never show any order reduction. All the other methods show order reduction until $Re = 2000$ after which they show increase in the slopes and they attain theoretical convergence rates for the highest value of Re. Among the fourth order methods, we see that all methods suffer order reduction till $Re = 2000$ as in the case of the third order methods and thereafter their slopes increase till they reach their theoretical rates at the highest Re.

We have seen how the slopes are affected by the global Reynolds number which is a

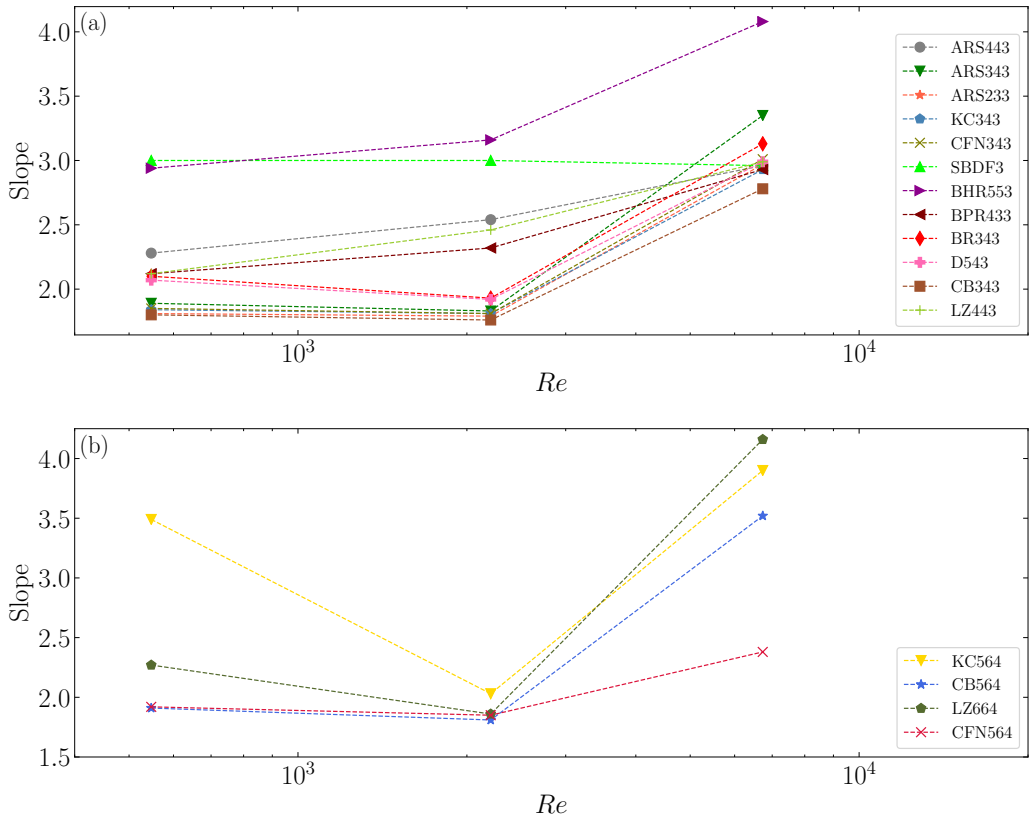

Figure 4.15: (a) Slopes of the third order methods as a function of the Reynolds number Re. (b) Slopes of the fourth order methods as a function of the Reynolds number Re. Both plots indicate slopes for vorticity ω and $Pr = 0.025$.

Table 4.2: Table for slope (ω) for different grid resolutions. All runs are done for case 7 and the time scheme D543. Star indicates the reference run.

N_r	N_m	Slope
048	160	2.80
064	160	2.48
080	160	2.25
096	160	2.17
112	160	2.13
128*	160*	2.12*
216	160	1.86
128	080	1.95
128	300	1.91

macroscopic measure of stiffness. In our case we do not have the luxury of a direct hands on measure of stiffness but now we attempt to obtain a tangible measure. For doing so, we choose a particular case (case : $7, Ra = 10^7, Pr = 1$) and a particular method (D543). We chose this case and method because it suffers order reduction and we want to know

how sensitive it is w.r.t to changes in the resolution of the grid. We now define

$$(\tau_L)_r = \min\left(\frac{\Delta r^2}{\nu+\kappa}\right), \tag{4.6}$$

$$(\tau_L)_\phi = \min\left(\frac{r^2\Delta\phi^2}{\nu+\kappa}\right), \tag{4.7}$$

$$(\tau_{NL})_r = \min\left[\frac{\Delta r}{\max(u_r)}\right], \tag{4.8}$$

$$(\tau_{NL})_\phi = \min\left[\frac{r\Delta\phi}{\max(u_\phi)}\right], \tag{4.9}$$

where, the subscripts r and ϕ denote the directions. These are the linear and nonlinear time scales along both r and ϕ. We also define the ratios

$$\epsilon_r = \frac{(\tau_L)_r}{(\tau_{NL})_r}, \tag{4.10}$$

$$\epsilon_\phi = \frac{(\tau_L)_\phi}{(\tau_{NL})_\phi}. \tag{4.11}$$

ϵ_r is chosen from a radial profile as the value at the intersection of the edge of thermal boundary layer δ_T and in our case, $\epsilon = \epsilon_r = \min(\epsilon_r, \epsilon_\phi)$. Therefore, when $\epsilon_r \ll 1$ means that the problem is stiff. This meaning of ϵ is used in Kennedy and Carpenter (2003) where their direct control of stiffness was the ratio $1/\epsilon$ which was an input to the problem. In our case we measure it based on the time scale definitions. Our input parameters Ra and Pr control the stiffness of the problem. The resulting stiffness definitions will be

$$\sigma_r = \frac{1}{\epsilon_r}, \tag{4.12}$$

$$\sigma_\phi = \frac{1}{\epsilon_\phi}. \tag{4.13}$$

For this analysis, we perform simulation runs with different grid resolutions. We show the different runs performed along with the slope for ω in Table 4.2. In this table, we vary N_r and N_m and we found that slope is sensitive to N_r and not for N_m. Thus the main observation here is that even if the global Reynolds number is constant, by changing grid resolution, we change the grid Reynolds number $Re_{\Delta r} \sim \epsilon_r$. To have more clarity, this observation is shown as a plot in Fig. 4.16. All the black markers are slope values at different radial grid resolutions, the $Nr = 128$ being the reference run.

Kennedy and Carpenter (2003) did a study where they plotted the slopes of the differential and the algebraic variables versus their stiffness measure for particular time schemes KC343 and KC564 as shown in Fig. 4.17. However, they had a direct control of the

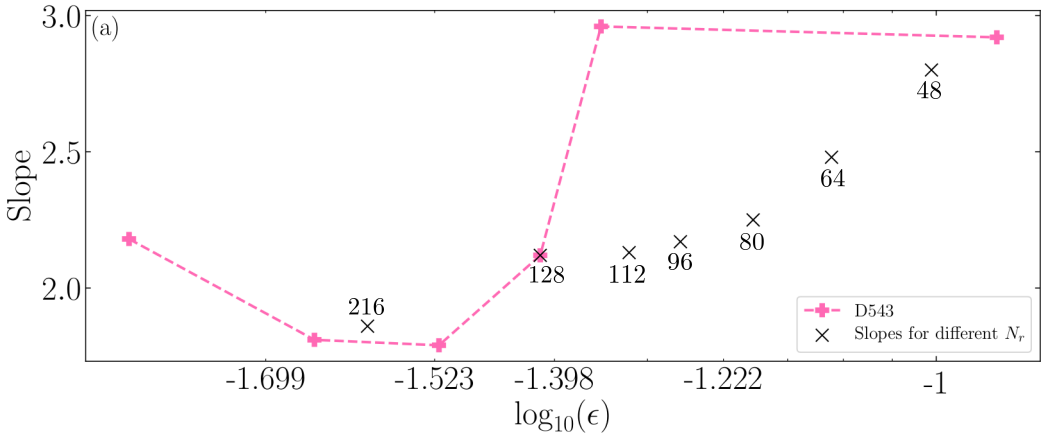

Figure 4.16: Slope versus ϵ for D543 method for the variable u_r. The black marker indicates different grid resolutions (N_r) along the radial direction.

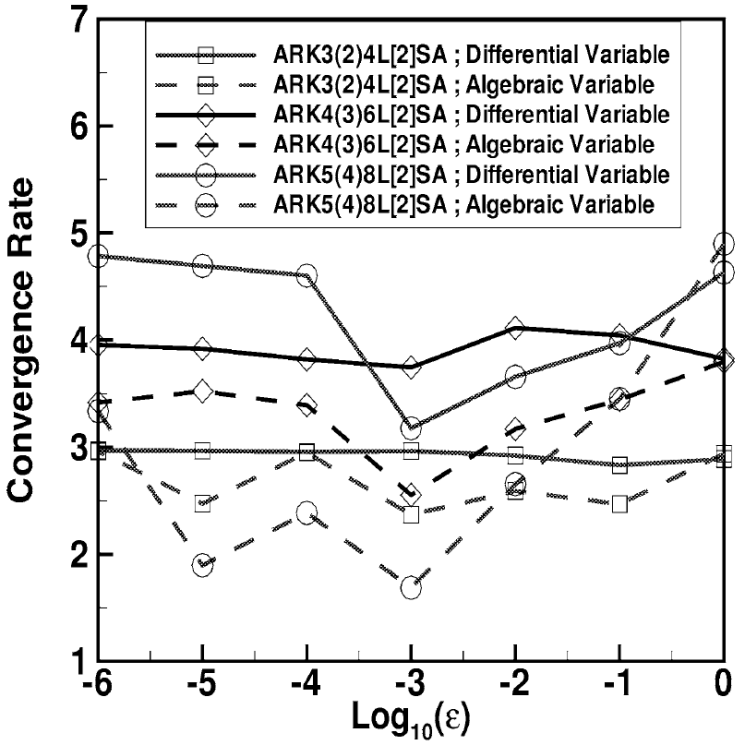

Figure 4.17: Slope versus ϵ from Kennedy and Carpenter (2003).

stiffness it was a parameter in their problem, the Van der Pol equation. In order to compare our result with Kennedy and Carpenter (2003), we plot slope versus our direct measure of ϵ in Fig. 4.18. We observe the same trend as we saw in the plots with slope versus global Reynolds numbers, the trend being that, we observe order reduction for lower values of the grid Reynolds number but as it increases, the slopes retain their theoretical values just as it is shown by Kennedy and Carpenter (2003).

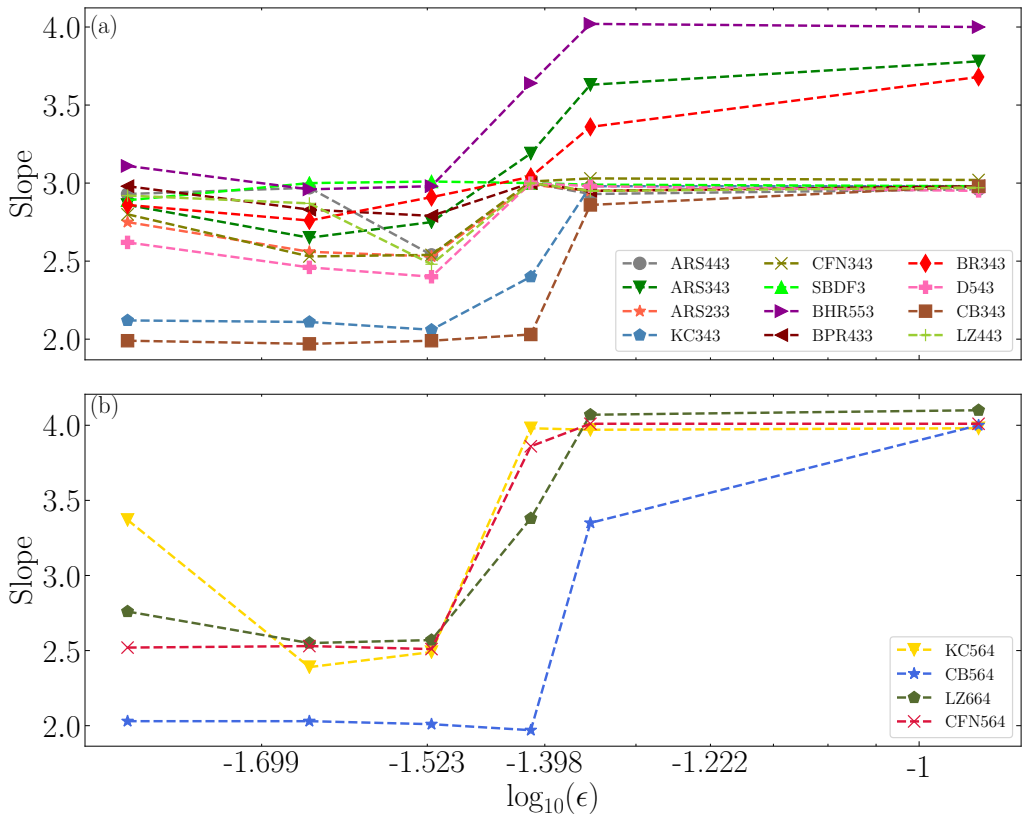

Figure 4.18: (a) Slopes of the third order methods as a function of ϵ or the grid Reynolds number $Re_{\Delta r}$. (b) Slopes of the fourth order methods as a function of ϵ. Both plots indicate slopes for radial velocity u_r and $Pr = 1$.

4.3 Stability

In this section we discuss the stability of the time integration methods shown so far. Since we are interested in the physics happening at the nonlinear time scale τ_{NL}, one must use a timestep size Δt which must be sufficient to resolve the nonlinear time scales in order to maintain accuracy. In order to find the maximum Δt possible so as to maintain stable and accurate solution, usually a constraint called as the Courant-Friedreichs-Lewy (CFL) is used (Courant et al., 1928). For our problem the Δt is given to be

$$\Delta t = \alpha \min\left[\min\left(\frac{\mathrm{d}r}{|u_r|_{\max}}\right), \min\left(\frac{r\mathrm{d}\phi}{|u_\phi|_{\max}}\right)\right], \tag{4.14}$$

where α is the CFL number. The maximum Δt to obtain a stable solution is now controlled by the CFL constraint $\alpha_{\max}$ which is the maximum CFL number. Thus, the value of $\alpha_{\max}$ is to be found for all the methods to ascertain their performance. Usually $\alpha_{\max}$ is chosen such that, for any value above this threshold would make the solution to blow up. For our problem however, we define a more stringent measure to determine $\alpha_{\max}$. We plot the viscous dissipation ϵ_U with respect to time and see if there are spurious spikes in the values of ϵ_U at any given time. Such spikes indicate numerical instability and at those

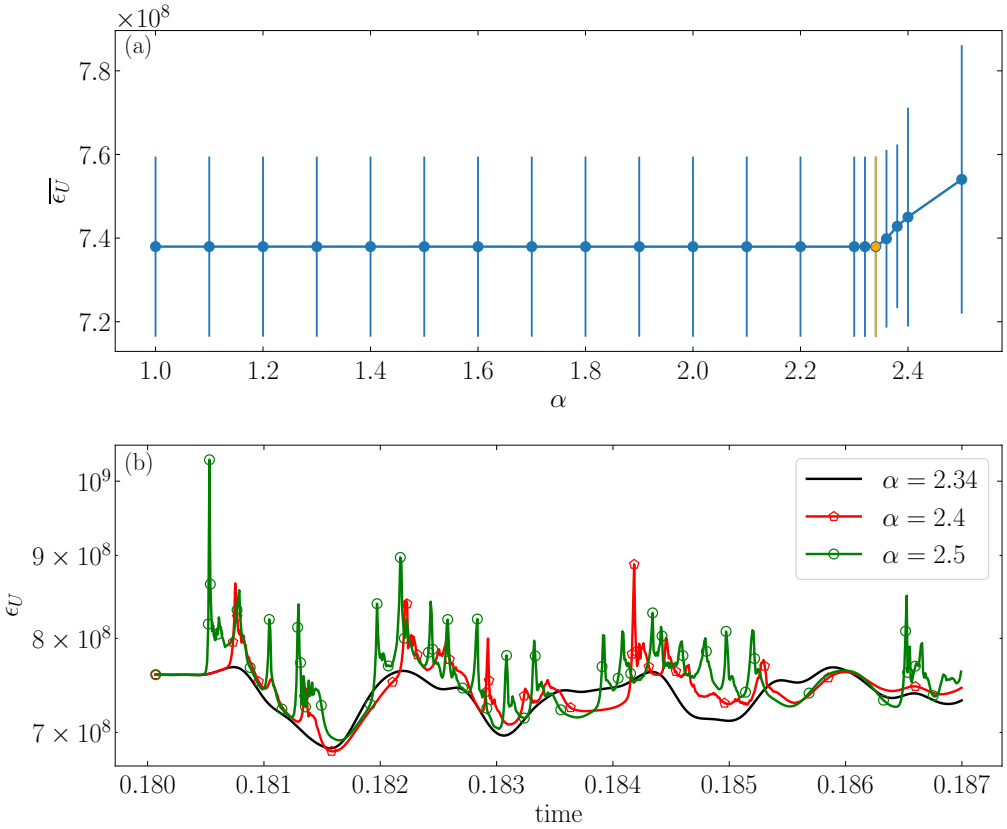

Figure 4.19: (a) Mean viscous dissipation $\overline{\epsilon_U}$ as a function of α. Vertical bar indicates standard deviation of the mean. The orange vertical bar indicates the $\alpha_{\max} = 2.34$. (b) Viscous dissipation ϵ_U as a function of time for different values of α. Plots shown in this figure are based on case 7 and the time integration method used is KC564.

instances in time, the power budget is not satisfied. So, even if the solution does not blow up in the presence of such spikes, it is not accurate and it will eventually blow up at longer times. Thus, we devise a method to know if there are such spurious values of ϵ_U in the the solution. For case 7, with the use of KC564 method, we show an illustration of the approach in Fig. 4.19. Plot (a) of Fig. 4.19 shows the mean viscous dissipation as a function of different values of α. From this figure we observe that the maximum value of α above which the solution blows up is 2.5. However, there is a range ($2.3 < \alpha < 2.5$) where larger values of $\overline{\epsilon_U}$ is observed. We measure $\overline{\epsilon_U}$ for an increment of 0.02 from $\alpha = 2.3$ to $\alpha = 2.4$ to narrow down to the best $\alpha_{\max}$. The viscous dissipation ϵ_U at these values of α is shown in plot (b) of Fig. 4.19. We notice that the maximum value of α for which we do not see any spikes in ϵ_U is at $\alpha = 2.34$. Thus, we obtain our $\alpha_{\max}$ as 2.34 using this procedure. Thereafter, we select the best performing set of methods for case 10 using the cost plots which were shown before in Fig. 4.8 and we obtain $\alpha_{\max}$ for all the cases in Table 4.1 except cases 0, 2 and 8 which were highly stable. Table 4.3 shows the values of $\alpha_{\max}$ for these cases as a function of increasing values of Reynolds numbers. We use this stringent measure to obtain $\alpha_{\max}$ for all the cases shown in Table 4.3.

Using these values of $\alpha_{\max}$ and the knowledge of cost for each of the methods, one can

Table 4.3: $\alpha_{\max}$ values for the selected methods.

Method	Reynolds number (Re)							
	75.28	273.61	547.63	937.34	2205.39	3586.14	6733.84	13159.46
CNAB2	0.38	0.46	0.42	0.38	0.34	0.24	0.24	0.16
ARS222	0.90	0.98	0.90	0.68	0.64	0.42	0.38	0.26
PC332	1.92	1.68	1.52	1.28	1.14	0.78	0.80	0.58
ARS232	1.14	1.50	0.84	1.24	0.82	0.76	0.80	0.58
SBDF2	0.42	0.42	0.38	0.30	0.28	0.18	0.18	0.13
ARS343	1.58	1.98	1.80	1.60	1.48	1.04	1.10	0.80
ARS233	0.96	1.24	0.66	1.08	0.64	0.72	0.76	0.52
KC343	1.42	1.78	1.64	1.46	1.34	0.98	1.04	0.72
BR343	1.22	1.58	0.92	1.78	0.86	1.12	1.04	0.82
BR553	1.14	1.48	1.32	1.22	1.14	0.78	0.84	0.62
BPR433	1.20	1.42	1.28	1.16	1.04	0.74	0.78	0.54
SBDF3	0.28	0.38	0.34	0.32	0.30	0.22	0.24	0.17
KC564	2.34	2.92	2.64	2.34	2.12	1.48	1.54	1.12
LZ664	1.38	1.84	1.76	1.52	1.54	1.10	1.28	0.92
CB564	1.66	2.26	2.04	1.84	1.78	1.20	1.38	1.10
SBDF4	0.20	0.42	0.26	0.24	0.24	0.18	0.22	0.15

From left to right, the cases shown in this table are placed according to increasing values of the Reynolds number Re.

construct an efficiency plot which indicates the relative performance of all the methods with respect to a chosen reference method. This can be measured as

$$\mathcal{P} = \frac{\varsigma}{\varsigma_{\text{ref}}}, \tag{4.15}$$

where, $\mathcal{P}$ is a performance measure and ref indicates the time integration method with respect to which we measure the performance and ς is defined as

$$\varsigma = \frac{\alpha_{\max}}{C_1}, \tag{4.16}$$

where, C_1 is the cost or the time taken for one iteration. This measure ignores the accuracy.

4.3.1 Performance analysis with respect to CNAB2

As mentioned in the introduction, most dynamo codes in the community use a second order method known as CNAB2 for their simulations. Therefore, it is natural to do a comparative analysis of performances of all the time integration methods with respect

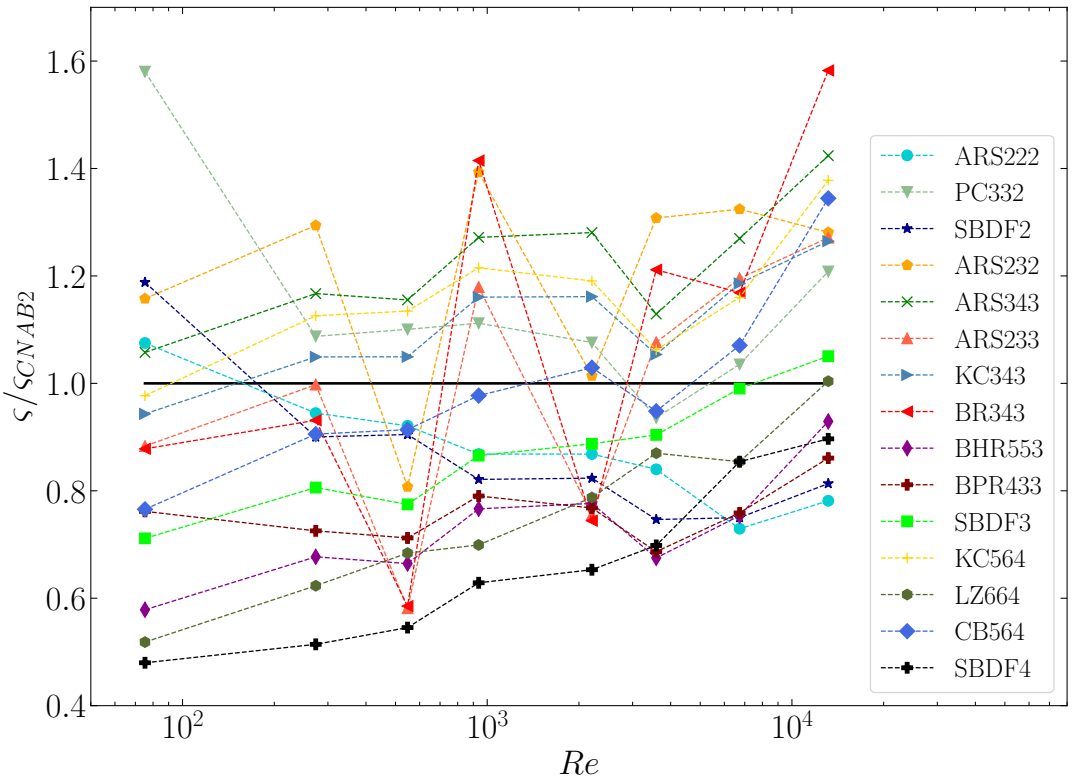

Figure 4.20: Performance $\mathcal{P}$ of each method as a function of the Reynolds number Re. The horizontal black line indicates the reference method CNAB2.

to CNAB2. We use the performance measure as mentioned in equation (4.15). We now show in Fig. 4.20 the performance of all the methods with respect to CNAB2 for increasing values of the Reynolds number Re. Regarding the second order methods, ARS232 performs best for large Re but for the majority of the cases, PC332 does a better job. It is around 1.2 times more efficient as compared to CNAB2 at the highest Reynolds number Re. The two second order methods, namely the ARS222 and SBDF2 show decreasing performance for the entire range of Re except for the highest value and they perform less efficiently compared to CNAB2. Among the third order methods, BR343 and ARS343 are more than 1.5 times efficient than CNAB2 at the highest Re. They are followed by KC343 and ARS233 which are also around 1.5 times more efficient than CNAB2. The IMEX method SBDF3 remains less efficient than CNAB2 for the entire range of Re except at the highest value where its slightly more efficient than CNAB2. Among the fourth order methods, KC564 and CB564 perform best and they are 1.5 times more efficient than CNAB2 at the highest Re. The methods ARS343, KC343 and KC564 remain more efficient than CNAB2 for almost the entire range of Re. Most of the schemes start to show increasing performance at high values of Re and even if some of them are less efficient than CNAB2 for lower ranges of Re, we predict that they will be more efficient than CNAB2 at higher ranges of Re. In fact, the steepest growing performance is shown by LZ664 which is as good as CNAB2 at the highest Re and if projected for higher Re, it will become more efficient than CNAB2. Also, the methods ARS232, ARS233 and BR343

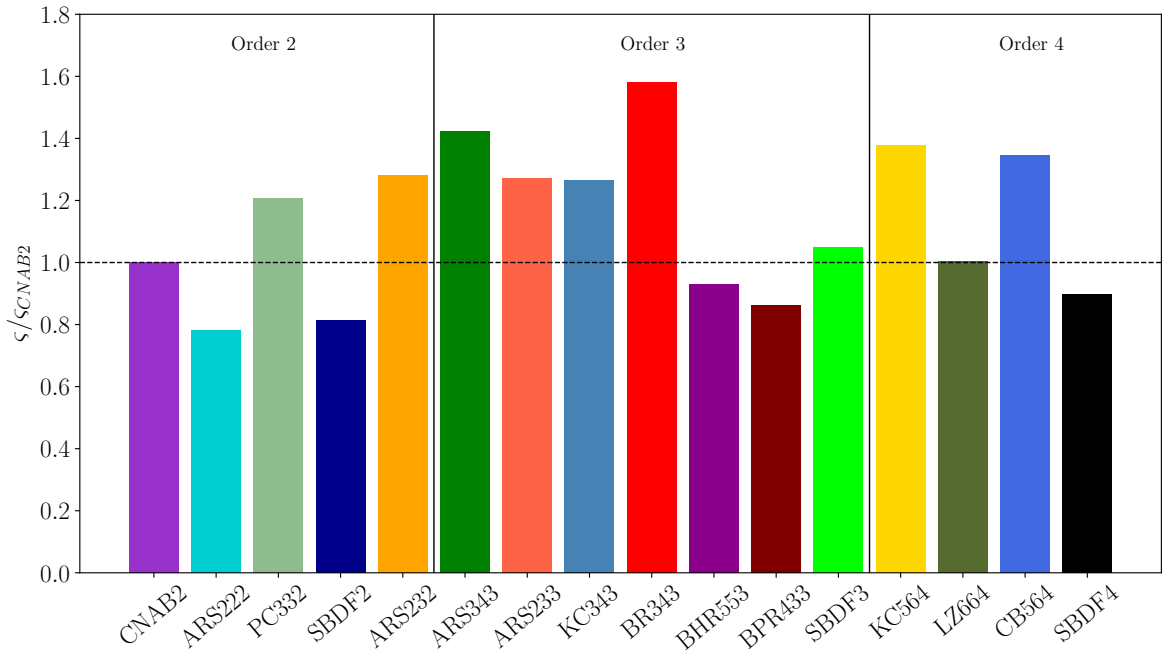

Figure 4.21: Performance of each method with respect to CNAB2 for the highest Reynolds number, $Re = 13159.46$.

are sensitive to Pr as their performance follows a zig-zag pattern. However, for large Re cases, they show a steady trend. To have a visually clear picture, for the highest Re, a histogram consisting of the performances of all the schemes with respect to CNAB2 is shown in Fig. 4.21. We observe that the top three methods which perform better than CNAB2 are BR343, ARS343 and KC564. Also of worthy note is that the IMEX-RK methods with assembly steps perform better than the stiffly accurate ones. It is surprising to note that two fourth order methods namely, KC564 and CB564 perform better than a second order method CNAB2. This is because they have better stability, i.e., they have higher values of $\alpha_{\max}$ than CNAB2 even though the time taken for one iteration is approximately 6 times shorter in the case of the CNAB2 method. The implication of this result in the dynamo community is significant because, we can now use a fourth order method for a high Re flow regime which can perform better than a second order method such as CNAB2 which was also shown by Marti et al. (2016) where, they use the CB564 method.

Now, we define another performance measure for the solutions which includes an additional accuracy measure. It is expressed by

$$\mathcal{P}_\varepsilon = \frac{\xi}{\xi_{\text{ref}}}, \tag{4.17}$$

where, $\mathcal{P}_\varepsilon$ is a performance measure, and ξ is defined as

$$\xi = \frac{\alpha_{\max}}{C_1}\frac{\varepsilon_{\text{ref}}}{\varepsilon}, \tag{4.18}$$

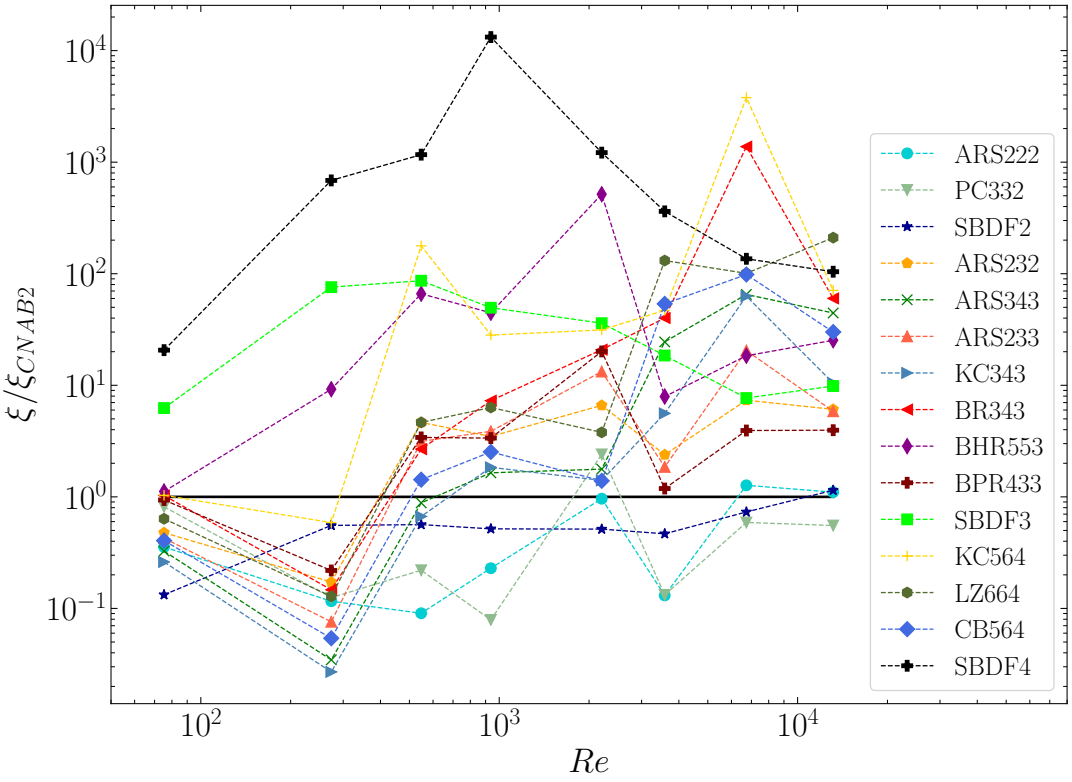

Figure 4.22: Performance $\mathcal{P}_\varepsilon$ of each method as a function of the Reynolds number Re. The horizontal black line indicates the reference method CNAB2.

where, ε is the L^2 error of ω measured at the average timestep size of the simulation run that was performed with α_{max}. We measure this error by linear interpolation using the previously determined value of the convergence rate or slope of the simulation run. We use the accuracy measure inside the performance measure because some times the methods lose their accuracy because of order reduction and it mostly happens to be with the variable ω. The choice of accuracy of ω as a component of the performance measure makes it a stringent one. This performance measure as a function of Re is shown in Fig. 4.22. From the figure, we observe that, among the second order methods, as Re increases, only the ARS232 method performs better than CNAB2. The methods ARS222 and SBDF2 perform worse than CNAB2 for the entire range of Re except for the highest value. The performance of PC332 is lower than CNAB2 at any value of Re. Among the third order methods, for almost all values of Re, SBDF3 outperforms all the other third order methods. Around $Re \sim 500$, the performance of the third order IMEX-RK methods starts to increase and they start to be better than CNAB2. At the highest Re, BR343 performs better than all the other third order methods. It is followed by ARS343 and BR553. Among the fourth order methods, SBDF4 performs better than all the methods for the entire range of Re except for the two largest values of Re. At the largest value of Re, LZ664 outperforms all the other methods. It is followed by SBDF4, KC564 and CB564. It is interesting to observe that the IMEX methods SBDF3 and SBDF4 starts to show drop in performance starting from $Re = 10^3$ as the Re increases and all the IMEX-

RK methods show increasing performance at this same range of Re. So, if we project the performance trends of these curves for higher Re than that is shown, the IMEX-RK methods should perform better than their IMEX counterparts.

Chapter 5

Discussion

This chapter discusses the findings of the results obtained and possible implications for further studies.

5.1 Summary of work done

This thesis focusses on the implementation, validation and analysis of variety of time integration schemes in the literature. As a starting point, a thermal convection model in an annulus was adopted as the problem domain. As for the numerical methods, a pseudospectral spatial discretization strategy was followed and for the time discretization, the IMEX multistep and multistage methods were used. The code for the work was then developed from scratch and it was validated and benchmarked. It was then parallelized in space using OpenMP pragmas. The code was written in the Fortran programming language and it contains many modules and subroutines. It was written in such a way that, one can easily implement additional time integration methods into the existing framework. The code and its associated documentation will be submitted to an open online repository in due time. Once all the time schemes were in place, the simulation cases were performed. The data was saved in to an ascii file and it was used to plot the accuracy and the cost diagrams. A stability analysis was performed for select sub-group of the time schemes to obtain information on their performance. Their data is also saved on to a second ascii file. These data will also be submitted to an open online repository.

5.2 Goals revisited

Before we summarize the findings of the study, we reiterate the main goals of the thesis which were

- To assess different time integration strategies with the hope of finding efficient ones based on multiple simulation cases with varying input parameters (Ra and Pr).

- To find whether higher order time integration methods perform better than the lower order ones.
- A more specific question to answer was whether there were better performing methods than CNAB2 which is widely used in the dynamo community and if so, how much better are they quantitatively.

5.3 Summary of findings

The following points are the important results from the study

- **Accuracy and cost**: We perform multiple simulation runs for this study for various input parameters. We base our reference to be the Reynolds number (Re). From this analysis, for low Re cases (Re $\mathcal{O}(100)$), among the second order methods, the IMEX-RK methods ARS232 and PC332 perform best. Among the third order methods, the multistep method SBDF3 performs best and is closely followed by the IMEX-RK method BHR553. Among the fourth order methods, SBDF4 remains the best performing method.

  For high Re cases ($Re > \mathcal{O}(1000)$), among the second order methods, ARS232 performs best. Among the third order methods, ARS343 and BR343 perform best and among the fourth order methods, KC564 outperforms all the others and it is closely followed by CB564.

- **Stiffness and order reduction**: We macroscopically measure stiffness as $\sigma = 1/Re$ and we find that as Re increases, the order reduction for the differential variable (ω) and the algebraic variable (ψ) using the IMEX-RK time schemes diminishes. We also define a tangible measure of the stiffness based on the linear and nonlinear time scales along the radial direction and the azimuthal direction to be $\sigma_r = (\tau_{NL})_r/(\tau_L)_r$ and $\sigma_p = (\tau_{NL})_p/(\tau_L)_p$ and we find the true measure of stiffness as $\sigma_r = \max(\sigma_r, \sigma_p)$. As the number of grid points along the radial direction increases, this stiffness also increases.

- **Stability and performance**: For this study, we use variable timestep sizes based on the CFL criterion. One of the goals of this thesis is to find whether there are other time integration methods which are better than CNAB2. For finding them, we define two performance measures. One that includes the accuracy measure denoted as $\mathcal{P}_\varepsilon$ and the other that does not which is denoted as $\mathcal{P}$. We find that, for the highest Re that we are interested in, based on the performance measure $\mathcal{P}$, the top three time schemes better than CNAB2 were BR343, ARS343 and KC564. Based on the performance measure with accuracy $\mathcal{P}_\varepsilon$, the top methods better than CNAB2 were SBDF4 and KC564. Also, for this measure, it is interesting to note that SBDF4 and SBDF3 methods show a decreasing performance trends for increasing Re.

– Therefore we find that higher order time integration methods such as KC564, ARS343, BR343 perform better than the second order accurate CNAB2 method. This means that we could perform simulations which are faster and more accurate. We foresee that this finding will improve the future 3D dynamo simulations as did the work done by Marti et al. (2016). Therefore, the findings from this study is another step towards the grand challenge put forward by Glatzmaier (2002). Also, moving from 2D convection to 3D dynamo models will incorporate additional stiffness from the Coriolis force and Lorentz force that may change our findings.

5.4 Implication for parallel-in-time methods

We discussed about the parallel-in-time methods in chapter 1 as it is one of the avenues to explore Earth-like parameter regimes. In order to understand how the findings from this thesis can be applied to the parallel-in-time methods, the reader is encouraged to review the parareal method which was discussed in chapter 1. Based on that discussion, one has to carefully choose $\mathcal{C}$ and $\mathcal{F}$ so as to achieve parareal speedups. We now correlate our findings from this thesis in choosing the coarse and fine operators as follows

- The choice of the IMEX multistage methods, IMEX-RK is a preferable choice to be used for $\mathcal{C}$ and $\mathcal{F}$ because, in such methods, we have the self-restart capability which means that they do not rely on solutions at previous time levels. The parareal method involves several time chunks and such self-restart capability is preferred whereas with the use of the IMEX multistep methods, we require solution information at several previous time levels making it a complicated method while calculating the parallel solution of the fine solver. From this thesis, one can now choose a variety of IMEX-RK methods for a parareal version of the same problem.
- Based on the performance analysis which we did in the previous chapter, for the highest Reynolds number, we may choose $\mathcal{C} \sim$ ARS232 or PC332 since we might reduce the grid size for the coarse solver and in doing so, the cost will be lower as compared to the higher order methods resulting in improved performance for the lower order IMEX-RK methods and a trial and error analysis has to be performed to ascertain the best method for the coarse solver. As for $\mathcal{F}$, we could use the KC564 or the CB564 method. From our study, we reiterate that we found the fourth order IMEX-RK methods to be one of the best performing methods among all for the highest Re. Therefore, they could be a valuable choice for the fine solver as compared to the regularly used explicit Runge-Kutta method RK4 in the parareal community for example in Samaddar et al. (2010) because the IMEX-RK methods do not face severe timestep restriction associated with the RK4 method. Thus, we could in principle have enhanced parareal speedup in using such IMEX-RK methods for the fine solver. In fact, one could in theory even use these fourth order IMEX-

RK methods for the coarse solver with decreased grid resolution as their stability is very high resulting in increased performance.

Appendix A

Time integration methods

We now provide the stencils for all the IMEX multi-step methods and the Butcher's tables for all the IMEX Runge-Kutta methods. All the following time integration methods were first shown in Table 3.1.

A.1 IMEX multi-step methods

Our governing equations are in a general form

$$\frac{\partial y}{\partial t} = F^D[t, y(t)] + F^A[t, y(t)] + S, \tag{A.1}$$

where, y is the solution we seek, F^D and F^A are the implicit and explicit components of the equation and S is a source term which in our case, it is the buoyancy term if we consider the vorticity equation. The following notations will be used:

$$\begin{aligned}
\Delta t &= t_{n+1} - t_n, \\
\Delta t_n &= t_n - t_{n-1}, \\
\Delta t_{n-1} &= t_{n-1} - t_{n-2}, \\
\Delta t_{n-2} &= t_{n-2} - t_{n-3}, \\
r_1 &= \frac{\Delta t_{n-1}}{\Delta t_n}, \\
r_2 &= \frac{\Delta t_{n-2}}{\Delta t_{n-1}}, \\
r_3 &= \frac{\Delta t_{n-3}}{\Delta t_{n-2}}.
\end{aligned} \tag{A.2}$$

We now proceed to show the stencils for the variable timestep IMEX multistep methods we used in the thesis.

A.1.1 CNAB2

$$y_{n+1}-\frac{\Delta t}{2}F^D_{n+1} = y_n+\frac{\Delta t}{2}F^D_n+\Delta t\left[\left((1+\frac{\Delta t_n}{2\Delta t_{n-1}}\right)F^A_n+\left(\frac{-\Delta t_n}{2\Delta t_{n-1}}\right)F^A_{n-1}\right]+\frac{\Delta t}{2}S_n. \quad \text{(A.3)}$$

A.1.2 SBDF2

$$a_0 y_{n+1} - \Delta t F^D_{n+1} = -a_1 y_n - a_2 y_{n-1} + \Delta t(b_0 F^A_n + b_1 F^A_{n-1}) + \Delta t S_n, \quad \text{(A.4)}$$

where,

$$\begin{aligned} a_0 &= (2+r_1)/(1+r_1),\\ a_1 &= 1+1/r_1,\\ a_2 &= -(1/r_1)/(1+r_1),\\ b_0 &= 1+1/r_1,\\ b_1 &= -1/r_1. \end{aligned} \quad \text{(A.5)}$$

A.1.3 SBDF3

$$a_0 y_{n+1}-\Delta t F^D_{n+1} = -a_1 y_n - a_2 y_{n-1} - a_3 y_{n-2} + \Delta t(b_0 F^A_n + b_1 F^A_{n-1} + b_2 F^A_{n-2}) + \Delta t S_n, \quad \text{(A.6)}$$

where,

$$\begin{aligned} a_0 &= 1+1/(1+r_1)+1/(1+r_1+r_2),\\ a_1 &= (1+r_1)(1+r_1+r_2)/[r_1(r_1+r_2)],\\ a_2 &= -(1+r_1+r_2)/[r_1 r_2(1+r_1)],\\ a_3 &= (1+r_1)/[r_2(r_1+r_2)(1+r_1+r_2)],\\ b_0 &= (1+r_1)(1+r_1+r_2)/[r_1(r_1+r_2)],\\ b_1 &= -(1+r_1+r_2)/(r_1 r_2),\\ b_2 &= (1+r_1)/[r_2(r_1+r_2)]. \end{aligned} \quad \text{(A.7)}$$

A.1.4 SBDF4

$$\begin{aligned} a_0 y_{n+1} - \Delta t F^D_{n+1} = \quad & - a_1 y_n - a_2 y_{n-1} - a_3 y_{n-2}\\ & - a_4 y_{n-3} + \Delta t(b_0 F^A_n + b_1 F^A_{n-1} + b_2 F^A_{n-2} - b_3 F^A_{n-3}) + \Delta t S_n, \end{aligned} \quad \text{(A.8)}$$

where,

$$
\begin{aligned}
c_1 &= 1 + r_3(1 + r_2), \\
c_2 &= 1 + r_2(1 + r_1), \\
c_3 &= 1 + r_3 c_2, \\
a_0 &= 1 + r_1/(1 + r_1) + r_1 r_2/c_2 + r_1 r_2 r_3/c_3, \\
a_1 &= -1 - r_1[1 + r_2(1 + r_1)(1 + r_3 + c2/c1)/(1 + r_2)], \\
a_2 &= r_1[r_1/(1 + r_1) + r_1 r_2(c_3 + r_3)/(1 + r_3)], \\
a_3 &= -r_2^3 r_1^2(1 + r_1)/(1 + r_2)c_3/c_2, \\
a_4 &= (1 + r_1)/(1 + r_3)c_2 c_3/c_1(r_1^2 r_2^3 r_3^4), \\
b_0 &= r_2(1 + r_1)/(1 + r_2)[(1 + r_1)(c_3 + r_3) + (1 + r_3)/r_2]/c_1, \\
b_1 &= -c_2 c_3 r_1/(1 + r_3), \\
b_2 &= r_2^2 r_1 c_3(1 + r_1)/(1 + r_2), \\
b_3 &= -r_3^3 r_2^2 r_1(c_2/c_1)(1 + r_1)/(1 + r_3).
\end{aligned}
\tag{A.9}
$$

A.2 IMEX Runge-Kutta methods

The Butcher's table on the left/top indicates the DIRK part and the Butcher's table on the right/bottom indicates the ERK part.

A.2.1 ARS222

$$
\begin{array}{c|ccc}
0 & 0 & 0 & 0 \\
\gamma & 0 & \gamma & 0 \\
1 & 0 & 1-\gamma & \gamma \\
\hline
 & 0 & 1-\gamma & \gamma
\end{array}
\qquad
\begin{array}{c|ccc}
0 & 0 & 0 & 0 \\
\gamma & \gamma & 0 & 0 \\
1 & \delta & 1-\delta & 0 \\
\hline
 & \delta & 1-\delta & 0
\end{array}
$$

$\gamma = (2 - \sqrt{2})/2$, $\delta = 1 - 2/\gamma$

A.2.2 ARS232

$$
\begin{array}{c|ccc}
0 & 0 & 0 & 0 \\
\gamma & 0 & \gamma & 0 \\
1 & 0 & 1-\gamma & \gamma \\
\hline
 & 0 & 1-\gamma & \gamma
\end{array}
\qquad
\begin{array}{c|ccc}
0 & 0 & 0 & 0 \\
\gamma & \gamma & 0 & 0 \\
1 & \delta & 1-\delta & 0 \\
\hline
 & 0 & 1-\gamma & \gamma
\end{array}
$$

$\gamma = (2 - \sqrt{2})/2$, $\delta = -2\sqrt{2}/3$

A.2.3 BPR442

0	0	0	0	0	0
$\frac{1}{4}$	$\frac{1}{4}$	0	0	0	0
$\frac{1}{4}$	$\frac{13}{4}$	-3	0	0	0
$\frac{3}{4}$	$\frac{1}{4}$	0	$\frac{1}{2}$	0	0
1	0	$\frac{1}{3}$	$\frac{1}{6}$	$\frac{1}{2}$	0
	0	$\frac{1}{3}$	$\frac{1}{6}$	$\frac{1}{2}$	0

0	0	0	0	0	0
$\frac{1}{4}$	0	$\frac{1}{4}$	0	0	0
$\frac{1}{4}$	0	0	$\frac{1}{4}$	0	0
$\frac{3}{4}$	0	$\frac{1}{24}$	$\frac{11}{24}$	$\frac{1}{4}$	0
1	0	$\frac{11}{24}$	$\frac{1}{6}$	$\frac{1}{8}$	$\frac{1}{4}$
	0	$\frac{11}{24}$	$\frac{1}{6}$	$\frac{1}{8}$	$\frac{1}{8}$

A.2.4 PC332

0	0	0	0	0
1	0.5	0.5	0	0
1	0.5	0	0.5	0
1	0.5	0	0	0.5
	0.5	0	0	0.5

0	0	0	0	0
1	1	0	0	0
1	0.5	0.5	0	0
1	0.5	0	0.5	0
	0.5	0	0.5	0

A.2.5 D332

0	0	0	0	0
$\frac{1}{2}$	0	$\frac{1}{2}$	0	0
1	0	$\frac{1}{2}$	$\frac{1}{2}$	0
1	0	1	$-\frac{1}{2}$	$\frac{1}{2}$
	0	1	$-\frac{1}{2}$	$\frac{1}{2}$

0	0	0	0	0
$\frac{1}{2}$	$\frac{1}{2}$	0	0	0
1	$\frac{1}{2}$	$\frac{1}{2}$	0	0
1	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	0
	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	0

A.2.6 ARS443

0	0	0	0	0	0
$\frac{1}{2}$	0	$\frac{1}{2}$	0	0	0
$\frac{2}{3}$	0	$\frac{1}{6}$	$\frac{1}{2}$	0	0
$\frac{1}{2}$	0	$-\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0
1	0	$\frac{3}{2}$	$-\frac{3}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
	0	$\frac{3}{2}$	$-\frac{3}{2}$	$\frac{1}{2}$	$\frac{1}{2}$

0	0	0	0	0	0
$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	0
$\frac{2}{3}$	$\frac{11}{18}$	$\frac{1}{18}$	0	0	0
$\frac{1}{2}$	$\frac{5}{6}$	$-\frac{5}{6}$	$\frac{1}{2}$	0	0
1	$\frac{1}{4}$	$\frac{7}{4}$	$\frac{3}{4}$	$-\frac{7}{4}$	0
	$\frac{1}{4}$	$\frac{7}{4}$	$\frac{3}{4}$	$-\frac{7}{4}$	0

A.2.7 ARS343

0	0	0	0	0
0.4358665215	0	0.4358665215	0	0
0.7179332608	0	0.2820667392	0.4358665215	0
1	0	1.208496649	-0.644363171	0.4358665215
	0	1.208496649	-0.644363171	0.4358665215

0	0	0	0	0
0.4358665215	0.4358665215	0	0	0
0.7179332608	0.321278886	0.3966543747	0	0
1	-0.105858296	0.5529291479	0.5529291479	0
	-0.105858296	0.5529291479	0.5529291479	0

A.2.8 ARS233

0	0	0	0
γ	0	γ	0
$1-\gamma$	0	$1-2\gamma$	γ
	0	$\frac{1}{2}$	$\frac{1}{2}$

0	0	0	0
γ	γ	0	0
$1-\gamma$	$\gamma-1$	$2(1-\gamma)$	0
	0	$\frac{1}{2}$	$\frac{1}{2}$

$\gamma = (3+\sqrt{3})/6$

A.2.9 KC343

0	0	0	0	0
$\frac{1767732205903}{2027836641118}$	$\frac{1767732205903}{4055673282236}$	$\frac{1767732205903}{4055673282236}$	0	0
$\frac{3}{5}$	$\frac{2746238789719}{10658868560708}$	$\frac{-640167445237}{6845629431997}$	$\frac{1767732205903}{4055673282236}$	0
1	$\frac{1471266399579}{7840856788654}$	$\frac{-4482444167858}{7529755066697}$	$\frac{11266239266428}{11593286722821}$	$\frac{1767732205903}{4055673282236}$
	$\frac{1471266399579}{7840856788654}$	$\frac{-4482444167858}{7529755066697}$	$\frac{11266239266428}{11593286722821}$	$\frac{1767732205903}{4055673282236}$

0	0	0	0	0
$\frac{1767732205903}{2027836641118}$	$\frac{1767732205903}{2027836641118}$	0	0	0
$\frac{3}{5}$	$\frac{5535828885825}{10492691773637}$	$\frac{788022342437}{10882634858940}$	0	0
1	$\frac{6485989280629}{16251701735622}$	$\frac{-4246266847089}{9704473918619}$	$\frac{10755448449292}{10357097424841}$	0
	$\frac{1471266399579}{7840856788654}$	$\frac{-4482444167858}{7529755066697}$	$\frac{11266239266428}{11593286722821}$	$\frac{1767732205903}{4055673282236}$

A.2.10 CFN343

0	0	0	0	0
γ	0	γ	0	0
0.717933260754229	0	$\frac{(1-\gamma)}{2}$	γ	0
1	0	1.208496649176010	-0.644363170684469	γ
	0	1.208496649176010	-0.644363170684469	γ

0	0	0	0	0
γ	γ	0	0	0
0.717933260754229	$\frac{(1+\gamma)}{2}+0.35$	-0.35	γ	0
1	0	1.989175724679846	-0.989175724679846	0
	0	1.208496649176010	-0.644363170684469	γ

$\gamma = 0.435866521508459$

A.2.11 BR553

0	0	0	0	0	0
c_2	γ	γ	0	0	0
c_3	a_{31}	a_{32}	γ	0	0
c_4	a_{41}	a_{41}	a_{43}	γ	0
1	b_1	0	b_3	b_4	γ
	b_1	0	b_3	b_4	γ

0	0	0	0	0	0
c_2	2γ	0	0	0	0
c_3	$\hat{a}_{31}$	$\hat{a}_{32}$	0	0	0
c_4	$\hat{a}_{41}$	0	$\hat{a}_{43}$	0	0
1	$\hat{a}_{51}$	$\hat{a}_{52}$	$\hat{a}_{53}$	$\hat{a}_{54}$	0
	b_1	0	b_3	b_4	γ

$\gamma = 0.435866521508482$

$\hat{a}_{31} = \gamma$

$\hat{a}_{32} = \hat{a}_{31}$

$\hat{a}_{41} = \frac{-475883375220285986033264}{59411272693343784570 4163}$

$\hat{a}_{42} = 0$

$\hat{a}_{43} = \frac{186623344982202682770 8736}{5941127269343784570 4163}$

$\hat{a}_{51} = \frac{628288458180731695856358816860913917376103082 47}{176112910684412105319781630311686343715753056000}$

$\hat{a}_{52} = \frac{-302987763081184622639300143137943089}{153535994420329331863918012936815 6500}$

$\hat{a}_{53} = \frac{262315887293043739337088563996093207}{297427554730376353252081786906492000}$

$\hat{a}_{54} = \frac{-98761823189417658143812471 7087}{238773376602029693195269018560 00}$

$a_{31} = \hat{a}_{31}$

$a_{32} = \frac{-31733082319927313}{455705377221960889379854647102}$

$a_{41} = \frac{-30123785410849220273619967617949193605163013778096 10}{45123394056585269977907753045030512597955897345819349}$

$a_{42} = \frac{-628655892978071532942 68}{10255967344161067230558732701909504 7}$

$a_{43} = \frac{41876979692085529960314626700141490094521427700 0}{21245436038525770855595459809987481860321716713 9}$

$b_1 = \frac{48769850233674067860351 1}{11811596369281859202602 08}$

$b_2 = 0$

$b_3 = -\hat{a}_{52}$

$b_4 = \frac{-10523592833510061607293821886 3}{22825544520646617565757271980 00}$

$c_2 = 2\gamma$

$c_3 = \frac{902905985686}{1035759735069}$

$c_4 = \frac{2684624}{1147171}$

A.2.12 BPR433

0	0	0	0	0	0
$\frac{1}{4}$	0	$\frac{1}{4}$	0	0	0
$\frac{1}{4}$	0	0	$\frac{1}{4}$	0	0
$\frac{3}{4}$	0	$\frac{1}{24}$	$\frac{11}{24}$	$\frac{1}{4}$	0
1	0	$\frac{11}{24}$	$\frac{1}{6}$	$\frac{1}{8}$	$\frac{1}{4}$
	0	$\frac{11}{24}$	$\frac{1}{6}$	$\frac{1}{8}$	$\frac{1}{4}$

0	0	0	0	0	0
$\frac{1}{4}$	$\frac{1}{4}$	0	0	0	0
$\frac{1}{4}$	$\frac{13}{4}$	-3	0	0	0
$\frac{3}{4}$	$\frac{1}{4}$	0	$\frac{1}{2}$	0	0
1	0	$\frac{1}{3}$	$\frac{1}{6}$	$\frac{1}{2}$	0
	0	$\frac{1}{3}$	$\frac{1}{6}$	$\frac{1}{2}$	0

A.2.13 BR343

0	0	0	0	0
0.435866521508458	0	0.4358665215	0	0
0.717933260754228	0	0.28206673924577	0.435866521508458	0
1	0	1.20849664917601	-0.64436317068446	0.435866521508458
	0	1.20849664917601	-0.64436317068446	0.435866521508458

0	0	0	0	0	
0.435866521508458	0.435866521508458	0	0	0	
0.7179332607542289	0.535396540307354	0.182536720446875	0	0	
1	0.63041255815287	-0.83193390106308	1.20152134291021	0	
	0	1.20849664917601	-0.64436317068446	0.435866521508458	

A.2.14 D543

0	0	0	0	0	0	0
$\frac{1}{2}$	0	$\frac{1}{2}$	0	0	0	0
1	0	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0
$\frac{1}{2}$	0	$\frac{1}{2}$	$-\frac{1}{2}$	$\frac{1}{2}$	0	0
0	0	$-1-\alpha$	$\frac{1}{2}$	α	$\frac{1}{2}$	0
1	0	$\frac{2}{3}$	$-\frac{1}{3}$	0	$\frac{1}{6}$	$\frac{1}{2}$
	0	$\frac{2}{3}$	$-\frac{1}{3}$	0	$\frac{1}{6}$	$\frac{1}{2}$

0	0	0	0	0	0	0
$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	0	0
1	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	0
$\frac{1}{2}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	0	0	0
0	0	0	0	0	0	0
1	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{2}$	0	0
	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{2}$	0	0

$\alpha = -3$

A.2.15 CB343

0	0	0	0	0
c_2	a_{21}	a_{22}	0	0
c_3	a_{31}	a_{32}	a_{33}	0
1	b_1	b_2	b_3	b_4
	b_1	b_2	b_3	b_4

0	0	0	0	0
c_2	$\hat{a}_{21}$	0	0	0
c_3	$\hat{a}_{31}$	$\hat{a}_{32}$	0	0
1	b_1	$\hat{a}_{42}$	$\hat{a}_{43}$	0
	b_1	b_2	b_3	b_4

$c_2 = 49/50$

$c_3 = 1/25$

$a_{21} = c2/2.0$

$a_{22} = c2/2.0$

$a_{31} = -785157464198/1093480182337$

$a_{32} = -30736234873/978681420651$

$a_{33} = 983779726483/1246172347126$

$b_1 = -2179897048956/603118880443$

$b_2 = 99189146040/891495457793$

$b_3 = 6064140186914/1415701440113$

$b_4 = 146791865627/668377518349$

$\hat{a}_{21} = c2$

$\hat{a}_{31} = 13244205847/647648310246$

$\hat{a}_{32} = 13419997131/686433909488$

$\hat{a}_{42} = 231677526244/1085522130027$

$\hat{a}_{43} = 3007879347537/683461566472$

A.2.16 LZ443

0	0	0	0	0	0
$\frac{1}{4}$	$\frac{3}{20}$	$\frac{1}{10}$	0	0	0
$\frac{1}{2}$	$\frac{9}{10}$	$-\frac{13}{10}$	$\frac{9}{10}$	0	0
$\frac{3}{4}$	$\frac{17}{10}$	$-\frac{11}{4}$	$\frac{3}{2}$	$\frac{3}{10}$	0
1	1	$-\frac{10}{3}$	$\frac{17}{3}$	$-\frac{10}{3}$	1
	1	$-\frac{10}{3}$	$\frac{17}{3}$	$-\frac{10}{3}$	1

0	0	0	0	0	0
$\frac{1}{4}$	$\frac{1}{4}$	0	0	0	0
$\frac{1}{2}$	0	$\frac{1}{2}$	0	0	0
$\frac{3}{4}$	$-\frac{1}{2}$	$\frac{5}{4}$	0	0	0
1	0	$\frac{2}{3}$	$-\frac{1}{3}$	$\frac{2}{3}$	0
	0	$\frac{2}{3}$	$-\frac{1}{3}$	$\frac{2}{3}$	0

A.2.17 KC564

0	0	0	0	0	0	0
c_2	a_{21}	a_{22}	0	0	0	0
c_3	a_{31}	a_{32}	a_{33}	0	0	0
c_4	a_{41}	a_{42}	a_{43}	a_{44}	0	0
c_5	a_{51}	a_{52}	a_{53}	a_{54}	a_{55}	0
1	b_1	b_2	b_3	b_4	b_5	b_6
	b_1	b_2	b_3	b_4	b_5	b_6

0	0	0	0	0	0	0
c_2	$\hat{a}_{21}$	0	0	0	0	0
c_3	$\hat{a}_{31}$	$\hat{a}_{32}$	0	0	0	0
c_4	$\hat{a}_{41}$	$\hat{a}_{42}$	$\hat{a}_{43}$	0	0	0
c_5	$\hat{a}_{51}$	$\hat{a}_{52}$	$\hat{a}_{53}$	$\hat{a}_{54}$	0	0
1	$\hat{a}_{61}$	$\hat{a}_{62}$	$\hat{a}_{63}$	$\hat{a}_{64}$	$\hat{a}_{65}$	0
	b_1	b_2	b_3	b_4	b_5	b_6

$c_2 = 1/2$
$c_3 = 83/250$
$c_4 = 31/50$
$c_5 = 17/20$
$a_{21} = 1/4$
$a_{22} = 1/4$
$a_{31} = 8611/62500$
$a_{32} = -1743/31250$
$a_{33} = 1/4$
$a_{41} = 5012029/34652500$
$a_{42} = -654441/2922500$
$a_{43} = 174375/388108$
$a_{44} = 1/4$
$a_{51} = 15267082809/155376265600$
$a_{52} = -71443401/120774400$
$a_{53} = 730878875/902184768$
$a_{54} = 2285395/8070912$
$a_{55} = 1/4$
$b_1 = 82889/524892$
$b_2 = 0$
$b_3 = 15625/83664$
$b_4 = 69875/102672$
$b_5 = -2260/8211$
$b_6 = 1/4$
$\hat{a}_{21} = 1/2$
$\hat{a}_{31} = 13861/62500$

$\hat{a}_{32} = 6889/62500$
$\hat{a}_{41} = -116923316275/2393684061468$
$\hat{a}_{42} = -2731218467317/15368042101831$
$\hat{a}_{43} = 9408046702089/11113171139209$
$\hat{a}_{51} = -451086348788/2902428689909$
$\hat{a}_{52} = -2682348792572/7519795681897$
$\hat{a}_{53} = 12662868775082/11960479115383$
$\hat{a}_{54} = 3355817975965/11060851509271$
$\hat{a}_{61} = 647845179188/3216320057751$
$\hat{a}_{62} = 73281519250/8382639484533$
$\hat{a}_{63} = 552539513391/3454668386233$
$\hat{a}_{64} = 3354512671639/8306763924573$
$\hat{a}_{65} = 4040/17871$

A.2.18 CB564

0	0	0	0	0	0	0
c_2	a_{21}	a_{22}	0	0	0	0
c_3	a_{31}	a_{32}	a_{33}	0	0	0
c_4	a_{41}	a_{42}	a_{43}	a_{44}	0	0
c_5	a_{51}	a_{52}	a_{53}	a_{54}	a_{55}	0
1	b_1	b_2	b_3	b_4	b_5	b_6
	b_1	b_2	b_3	b_4	b_5	b_6

0	0	0	0	0	0	0
c_2	$\hat{a}_{21}$	0	0	0	0	0
c_3	$\hat{a}_{31}$	$\hat{a}_{32}$	0	0	0	0
c_4	$\hat{a}_{41}$	$\hat{a}_{42}$	$\hat{a}_{43}$	0	0	0
c_5	$\hat{a}_{51}$	$\hat{a}_{52}$	$\hat{a}_{53}$	$\hat{a}_{54}$	0	0
1	$\hat{a}_{61}$	$\hat{a}_{62}$	$\hat{a}_{63}$	$\hat{a}_{64}$	$\hat{a}_{65}$	0
	b_1	b_2	b_3	b_4	b_5	b_6

$c_2 = 0.25$
$c_3 = 0.75$
$c_4 = 3/8$
$c_5 = 0.5$
$a_{21} = c2/2$
$a_{22} = c2/2$
$a_{31} = 216145252607/961230882893$
$a_{32} = 257479850128/1143310606989$
$a_{33} = 30481561667/101628412017$
$a_{42} = -381180097479/1276440792700$
$a_{43} = -54660926949/461115766612$
$a_{44} = 344309628413/552073727558$

$a_{53} = -100836174740/861952129159$

$a_{54} = -250423827953/1283875864443$

$a_{55} = 0.5$

$\hat{a}_{21} = c_2$

$\hat{a}_{31} = 153985248130/1004999853329$

$\hat{a}_{32} = 902825336800/1512825644809$

$\hat{a}_{42} = 99316866929/820744730663$

$\hat{a}_{43} = 82888780751/969573940619$

$\hat{a}_{53} = 57501241309/765040883867$

$\hat{a}_{54} = 76345938311/676824576433$

$\hat{a}_{64} = -4099309936455/6310162971841$

$\hat{a}_{65} = 1395992540491/933264948679$

$b_1 = 232049084587/1377130630063$

$b_2 = 322009889509/2243393849156$

$b_3 = -195109672787/1233165545817$

$b_4 = -340582416761/705418832319$

$b_5 = 463396075661/409972144477$

$b_6 = 323177943294/1626646580633$

A.2.19 LZ664

0	0	0	0	0	0	0	0
$\frac{1}{3}$	$-\frac{1}{6}$	$\frac{1}{2}$	0	0	0	0	0
$\frac{1}{3}$	$\frac{1}{6}$	$-\frac{1}{3}$	$\frac{1}{2}$	0	0	0	0
$\frac{1}{2}$	$\frac{3}{8}$	$-\frac{3}{8}$	0	$\frac{1}{2}$	0	0	0
$\frac{1}{2}$	$\frac{1}{8}$	0	$\frac{3}{8}$	$-\frac{1}{2}$	$\frac{1}{2}$	0	0
1	$-\frac{1}{2}$	0	3	-2	0	$\frac{1}{2}$	0
1	$\frac{1}{6}$	0	0	0	$\frac{2}{3}$	$\frac{1}{2}$	$\frac{2}{3}$
	$\frac{1}{6}$	0	0	0	$\frac{2}{3}$	$\frac{1}{2}$	$\frac{2}{3}$

0	0	0	0	0	0	0	0
$\frac{1}{3}$	$\frac{1}{3}$	0	0	0	0	0	0
$\frac{1}{3}$	$\frac{1}{6}$	$\frac{1}{6}$	0	0	0	0	0
$\frac{1}{2}$	$\frac{1}{8}$	0	$\frac{3}{8}$	0	0	0	0
$\frac{1}{2}$	$\frac{1}{8}$	0	$\frac{3}{8}$	0	0	0	0
1	$\frac{1}{2}$	0	$-\frac{3}{2}$	1	1	0	0
1	$\frac{1}{6}$	0	0	0	$\frac{2}{3}$	$\frac{1}{6}$	0
	$\frac{1}{6}$	0	0	0	$\frac{2}{3}$	$\frac{1}{6}$	0

A.2.20 CFN564

0	0	0	0	0	0	0
$\frac{1}{4}$	0	$\frac{1}{4}$	0	0	0	0
$\frac{3}{4}$	0	$\frac{1}{2}$	$\frac{1}{4}$	0	0	0
$\frac{11}{20}$	0	$\frac{17}{50}$	$-\frac{1}{25}$	$\frac{1}{4}$	0	0
$\frac{1}{2}$	0	$\frac{371}{1360}$	$-\frac{137}{2720}$	$\frac{15}{544}$	$\frac{1}{4}$	0
1	0	$\frac{25}{24}$	$-\frac{49}{48}$	$\frac{125}{16}$	$-\frac{85}{12}$	$\frac{1}{4}$
	0	$\frac{25}{24}$	$-\frac{49}{48}$	$\frac{125}{16}$	$-\frac{85}{12}$	$\frac{1}{4}$

0	0	0	0	0	0	0
$\frac{1}{4}$	$\frac{1}{4}$	0	0	0	0	0
$\frac{3}{4}$	$-\frac{1}{4}$	1	0	0	0	0
$\frac{11}{20}$	$-\frac{13}{100}$	$\frac{43}{75}$	$\frac{8}{75}$	0	0	0
$\frac{1}{2}$	$-\frac{6}{85}$	$\frac{42}{85}$	$\frac{179}{1360}$	$-\frac{15}{272}$	0	0
1	0	$\frac{79}{24}$	$-\frac{5}{8}$	$\frac{25}{2}$	$-\frac{85}{6}$	0
	0	$\frac{79}{24}$	$-\frac{5}{8}$	$\frac{25}{2}$	$-\frac{85}{6}$	0

Appendix B

Dealiasing explained

The concept of dealiasing was introduced by Orzag (1980). In numerical modelling of complex physical processes, we come across non-linear terms. In the present context, we have quadratically nonlinear terms which represent the advective part of the equation. For example, let us take terms like,

$$u_1(x) \times u_2(x), \tag{B.1}$$

where, u_1 and u_2 are solution variables and x is a physical dimension. Now, when we use Fourier spectral collocation methods such as those described in chapter 3, such products in Fourier space will look like,

$$\hat{u}_1(k) \times \hat{u}_2(k), \tag{B.2}$$

where, $\hat{u}$ represent the solution variables in the spectral space or Fourier space and k is a Fourier mode. Such calculations are called convolutions and they are expensive to calculate in the Fourier space. So, we transform these terms in to the physical space and there we multiply them. After that, we bring the product back to Fourier space for further calculations. However, in doing so, we have contributions from both the modes of $\hat{u}_1$ and $\hat{u}_2$, for example,

$$e^{ik_1x}e^{ik_2x} = e^{i(k_1+k_2)x}. \tag{B.3}$$

The solution is represented by the modes $k \leq N_m$. However, when $k_1 + k_2 > N_m$, the solution is supposed to be aliased. This is explained with an example. Let us take $u_1 = \sin(k_1x)$ and $u_2 = \sin(k_2x)$. So, their product will look like,

$$u_1 \times u_2 = \frac{1}{2}\Big\{\cos[(k_1 - k_2)x) - \cos((k_1 + k_2)x]\Big\}. \tag{B.4}$$

We would like the mode $k_1 + k_2$ to be lying below $N_p/2$ in the mode space shown in the number line in figure B.1. Now, let us consider the case when it is not.

If we look at figure B.2, we can see that $k_1 + k_2$ is aliased to $k_1 + k_2 - N_p$ which is inside the range of wavenumbers which represent the solution. The word alias implies that same solution is represented by modes of two different waves. This is unphysical and at each time step of the numerical simulation, when calculating the nonlinear part of the equation, the larger modes will contribute their energy like this. This will lead the solution to blow up. To see how $k_1 + k_2$ aliased to $k_1 + k_2 - N_p$, let us look at the complex to real Fourier transform which is given as

$$u(x_j) = \sum_{k=-K}^{K} \hat{u}_k e^{ikx_j}, \; j = 1, \cdots, N, \tag{B.5}$$

where, $N = 2K + 1$ and K is the number of Fourier modes. The exponential from this transformation is e^{ikx_j}. If we put $k = k_1 + k_2 = N_p/2 + d$ inside the exponential, and make some small changes in the terms like $N_p/2 + d = N_p - N_p/2 + d$, then,

$$e^{i(N_p - N_p/2 + d)2\pi j/N} = e^{i2\pi j} e^{(-N_p/2+d)x_j}, \tag{B.6}$$

where $e^{i2\pi j} = 1$ for all j, which means $N_p/2 + d$ is aliased to $-N_p/2 + d$ which is nothing but $k_1 + k_2 - N_p$. Conversely, $-(k_1 + k_2)$ will be aliased to $N_p - (k_1 + k_2)$. Suppose we fix a mode k such that we do not want this aliasing inside $(-k, k)$, then,

$$N_p - (k_1 + k_2) > k, \tag{B.7}$$

and when if we take $k_1 = k_2 = k$, then $N_p - 2k > k$, in other words $k < N_p/3$ or

$$k < (2/3)N_p/2, \tag{B.8}$$

which is famously known as the 2/3 rule (Boyd, 2001, Chapter 9) for dealiasing. So, when we restrict ourselves with the modes $(-k, k)$ with this rule, and put rest of the modes to zero, we have dealiased the solution. It is vivid if we continue to explain the dealiasing with numbers. Let us say $N_m = 10$, $Np = 2N_m$. According to the 2/3 rule, $k < 7$. If we use $k_1 = 2$ and $k_2 = 3$, $k_1 + k_2 < 10$, they do not alias. However, if we use $k_1 = 5$ and $k_2 = 6$, then we have $k_1 + k_2 = 11$. Now, the wavenumber line looks like the one shown in figure B.3. It must be noted that $k_1 + k_2 = 11$ is aliased to -10 which is outside the range of k which is $(-7, 7)$. Now, consider $k_1 = 8$ and $k_2 = 9$ which will be aliased to $k_1 + k_2 - 20 = -3$. This lies inside $-7, 7$ as shown in figure B.4. For intuitive purposes, consider the problem of thermal convection where the nonlinear terms such as $(\mathbf{u} \cdot \boldsymbol{\nabla\omega})$

has contributions from large modes in between $(k, N_p/2)$. Such modes will unwantedly appear inside $(-k, k)$ at each time step and starts accumulating additional energy over time. This will eventually lead to blow up of energy. So if we keep $k_1, k_2 < (2/3)N_p/2$ and put all other $k_1, k_2 = 0$ in between $(k, N_p/2)$, then the solution is dealiased.

An alternative way to ensure dealiasing is to use the (3/2) rule. Here, the number of physical points must be taken as $N_p = 3N_m$ which means that, $k < 2/3(N_p/2)$. Therefore, the preferred number of modes must follow $k < N_m$ which in turn means we are not required to truncate any modes for dealiasing. However, we deal with $3/2N_m$ modes while performing the forward or inverse Fourier transforms and only N_m modes are used for solving the problem. Thus, the remaining $N_m/2$ must be put to zero and padded with the non-zero N_m modes and only after doing this we must take the complex to real Fourier transform. It is one's choice to use the 2/3 or the 3/2 rule. However, we can note that 3/2 rule does not involve any truncation of the modes, so for a given N_m we may see the complete power spectral density or the kinetic energy spectra for all the modes N_m. Following the above discussion, if we consider the same example as above, when we have $k_1 = 2$ and $k_2 = 3$, the power spectral density for the product after taking FFT can be shown as in figure B.5.

It is noted that $k_1 + k_2 < 7$, so there is no problem of aliasing. However, when we have $k_1 = 8$ and $k_2 = 9$, then $k_1 + k_2 > 7$. So when we don't apply the (3/2) rule, it has to alias which is shown in figure B.6. The aliased mode is $k_1 + k_2 = 17$ which is aliased to $k_1 + k_2 - 17 = -3$. Since we consider transform of real variables i.e. we have modes running from 0-$N_p/2$, it will be aliased to 4 as it has to include the $k = 0$ as well. However, when we apply the dealising rule, we do not see this aliasing happening as shown in figure B.7. Therefore, we can conclude that dealiasing plays a very important role while using Fourier spectral methods for solving a nonlinear problem.

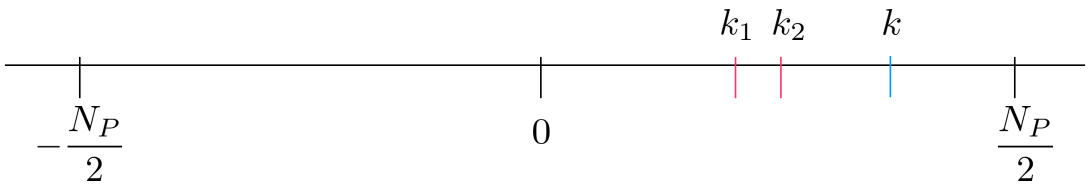

Figure B.1

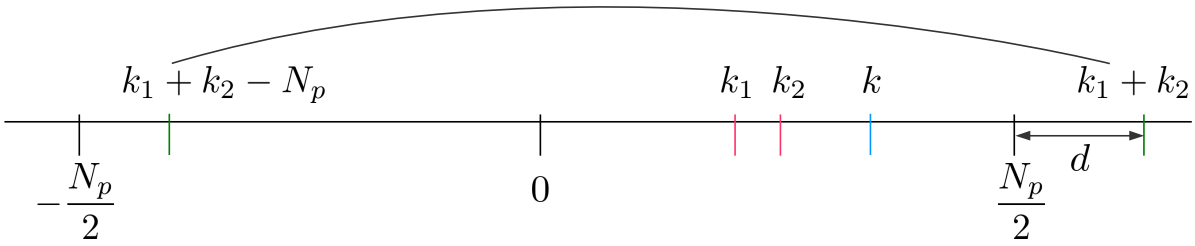

Figure B.2

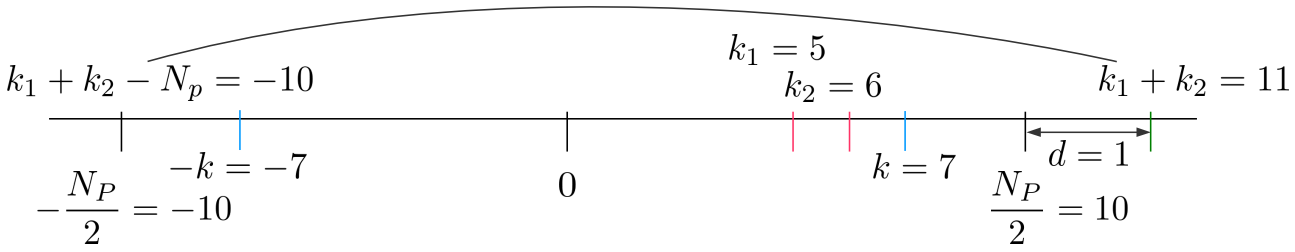

Figure B.3

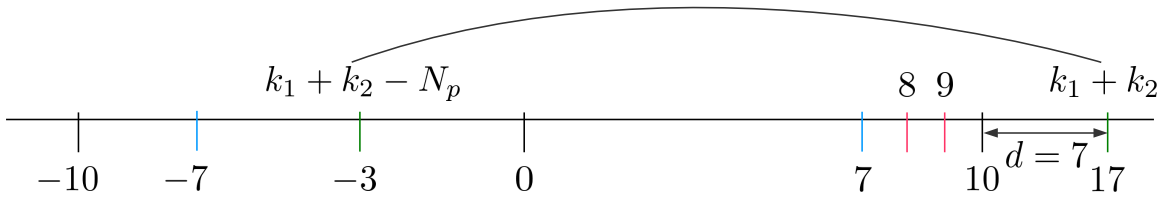

Figure B.4

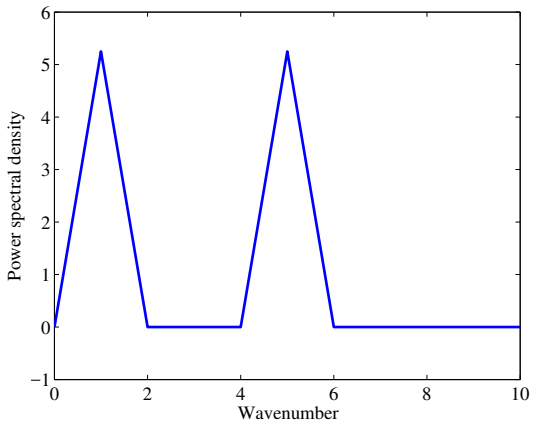

Figure B.5: Power spectral density when $k_1 = 2$ and $k_2 = 3$

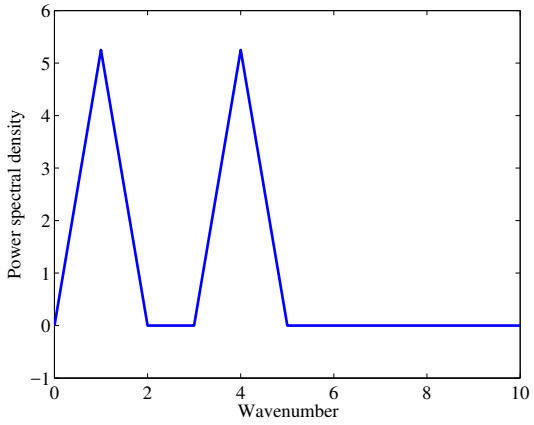

Figure B.6: Power spectral density without 3/2 rule when $k_1 = 8$ and $k_2 = 9$.

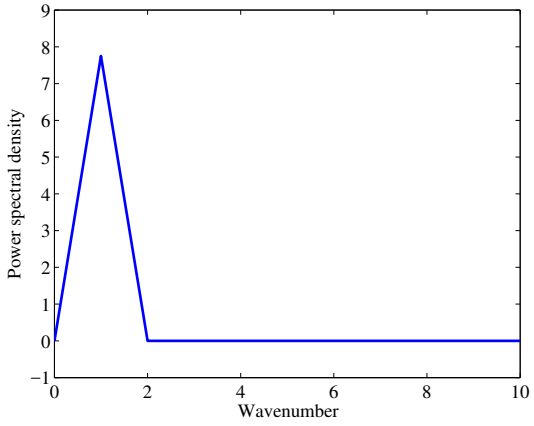

Figure B.7: Power spectral density with 3/2 rule when $k_1 = 8$ and $k_2 = 9$.

Appendix C

Simulation data

All the simulation cases shown performed for all the time integration methods were stored in ascii files. The data for the accuracy and cost analysis are saved in data1.txt. The different columns of the ascii file are described in Table C.1 below.

Table C.1

Column	Description
1	Case number
2	Rayleigh number (Ra)
3	Prandtl number (Pr)
4	Reynolds number (Re)
5	Nusselt number (Nu)
6	Viscous dissipation
7	Buoyancy power
8	Maximum order (N_m)
9	Number of Chebyshev points along radius (N_r)
10	Simulation start time
11	Simulation end time
12	Diffusive time scale along radius $(\tau_L)_r$
13	Advective time scale along radius $(\tau_{NL})_r$
14	Diffusive time scale along azimuth $(\tau_L)_r$
15	Advective time scale along azimuth $(\tau_{NL})_r$
16	Boundary condition
17	Name of the time scheme
18	Physical time taken or cost for one iteration
19	Number of iterations
20	Timestep size (Δt)
21	Infinite norm of error in temperature ($\|\epsilon\|_\infty(T)$)
22	Infinite norm of error in vorticity ($\|\epsilon\|_\infty(\omega)$)
23	Infinite norm of error in radial velocity ($\|\epsilon\|_\infty(u_r)$)
24	Infinite norm of error in azimuthal velocity ($\|\epsilon\|_\infty(u_\phi)$)
25	L^2 norm of error in temperature ($\|\epsilon\|_{L^2}(T)$)
26	L^2 norm of error in vorticity ($\|\epsilon\|_{L^2}(\omega)$)
27	L^2 norm of error in radial velocity ($\|\epsilon\|_{L^2}(u_r)$)
28	L^2 norm of error in azimuthal velocity ($\|\epsilon\|_{L^2}(u_\phi)$)
29	Slope of Temperature w.r.t infinite norm
31	Slope of Vorticity w.r.t infinite norm
32	Slope of Radial Velocity w.r.t infinite norm
33	Slope of Azimuthal velocity w.r.t infinite norm
34	Slope of Temperature w.r.t L^2 norm
35	Slope of Vorticity w.r.t L^2 norm
36	Slope of Radial Velocity w.r.t L^2 norm
37	Slope of Azimuthal velocity w.r.t L^2 norm

Likewise, the data from the stability analysis which is shown in Table 4.4 of chapter 4 is saved in data2.txt.

Bibliography

M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover, First edition, 1965.

A. Alonso, Sánchez J., and M. Net. Transition to Temporal Chaos in an $O(2)$-Symmetric Convective System for Low Prandtl Numbers. *Prog. Theor. Phys. Supp.*, 139, 2000.

G. M. Amdahl. Validity of the Single Processor Approach to Achieving Large Scale Computing Capabilities. *AFIPS Conference Proceedings*, 30:483–485, 1967.

E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK User's Guide*. SIAM, First edition, 1999.

U.M. Ascher and L.R. Petzold. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. SIAM, First edition, 1998.

U.M. Ascher, S.J. Ruuth, and B.T.R. Wetton. Implicit-explicit methods for time-dependent partial differential equations. *SIAM J. Numer. Anal.*, 32:797–823, 1995.

U.M. Ascher, S. J. Ruuth, and R. J. Spiteri. Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations. *Appl. Numer. Math.*, 25:151–167, 1997.

A. Bashforth. *An attempt to test the theories of capillary action by comparing the theoretical and measured forms of drops of fluid. With an explanation of the method of integration employed in constructing the tables which give the theoretical form of such drops, by J.C.Adams*. Cambridge Univ. Press, 1883.

G.K. Batchelor. *An Introduction to Fluid Dynamics*. Cambridge University Press, Second edition, 2000.

C. Bendsten. On implicit Runge-Kutta methods with high stage order. *BIT Numer. Math.*, 37:1:221–226, 1997.

S. Boscarino. Error analysis of IMEX Runge-Kutta methods derived from differential-algebraic systems. *SIAM J. Numer. Anal.*, 45:1600–1621, 2007.

S. Boscarino and G. Russo. On the uniform accuracy of IMEX Runge-Kutta schemes and applications to hyperbolic systems with relaxation. *Communications to SIMAI Congress*, 2, 2007.

S. Boscarino and G. Russo. On a class of uniformly accurate IMEX Runge-Kutta schemes and applications to hyperbolic systems with relaxation. *SIAM J. Sci. Comput.*, 31: 1926–1945, 2009.

S. Boscarino, L. Pareschi, and G. Russo. Implicit-Explicit Runge-Kutta schemes for hyperbolic systems and kinetic equations in the diffusion limit. *SIAM J. Sci. Comput.*, 35:A22–A51, 2013.

S. Boscarino, L. Pareschi, and G. Russo. A unified IMEX Runge-Kutta approach for hyperbolic systems with multiscale relaxation. *SIAM J. Numer. Anal.*, 55:2085–2109, 2017.

J.P. Boyd. *Chebyshev and Fourier Spectral Methods*. Dover, Second edition, 2001.

E. C. Bullard and H. Gellman. Homogeneous dynamos and terrestrial magnetism. *Phil. Trans. Royal Soc. A*, 30:237–257, 1954.

J. C. Butcher and Z. Jackiewicz. Diagonally implicit general linear methods for ordinary differential equations. *BIT Numerical Mathematics*, 33:452–472, 1993.

J.C. Butcher. On runge-kutta processes of high order. *J. Austral. Math. Soc.*, 4:176–194, 1964b.

M.P. Calvo, Frutos J., de, and J. Novo. Linearly implicit Runge-Kutta methods for advection-reaction-diffusion equations. *Appl. Numer. Math.*, 37:535–549, 2001.

C. Canuto. Boundary conditions in Chebyshev and Legendre methods. *SIAM J. Numer. Anal.*, 23:815–831, 1986.

Daniele Cavaglieri and Thomas Bewley. Low-storage implicit/explicit Runge-Kutta schemes for the simulation of stiff high-dimensional ODE systems. *J. Comput. Appl. Math.*, 286:172–193, 2015.

Subrahmanyan Chandrasekhar. *Hydrodynamic and Hydromagnetic Stability*. Dover, Third edition, 1981.

F. Chillà and J Schumacher. New perspectives in turbulent Rayleigh-bénard convection. 35:58, 2012.

U. R. Christensen. A numerical dynamo benchmark. *Phys. Earth. Planet. Inter.*, 128: 25–34, 2001.

U. R. Christensen and J. Wicht. *Treatise on Geophysics 2nd Edition, Volume 8*. Elsevier, First edition, 2015.

A.T. Clarke, C.J. Davies, D. Ruprecht, and S.M. Tobias. Parallel-in-time integration of Kinematic Dynamos. *arXiv:1902.00387*, 2019.

R. Courant, K. Friedrichs, and H. Lewy. Partial differential equations of mathematical physics. *Mathematische Annalen*, 100:32–74, 1928.

C.F. Curtiss and J.O. Hirschfelder. Integration of stiff equations. *Proc. Nat. Acad. Sci.*, 38:235–243, 1952.

C. J. Davies, D. Gubbins, and P. K. Jimack. Scalability of pseudospectral methods for geodynamo simulations. *Concurrency and Computation: Practice and Experience*, 23: 38–56, 2011.

S.-C Duan. All-stages-implicit and strong-stability-preserving implicit-explicit Runge-Kutta time discretization schemes for hyperbolic systems with stiff relaxation terms. *arXiv:1606.02053*, 2016.

W. M. Elsasser. Induction Effects in Terrestrial Magnetism Part II. The Secular Variation. *Phys. Rev.*, 70:202, 1946.

L. Euler. Du mouvement de rotation des corps solides autour d'un axe variable. *Hist. de l'Acad. Royale de Berlin, Tom.14, Ann e MDCCLVIII, pp.154-193. Opera Omnia Ser*, 8:200–235, 1758.

N. Featherstone and B. W. Hindman. The spectral amplitude of stellar convection and its scaling in the high-Rayleigh-number regime. *Astrophys. J.*, 818:32, 2016.

P.F. Fisher, F. Hecht, and Y. Maday. A parareal in time semi-implicit approximation of the Navier-Stokes equations. *Proceedings of the 15th International Domain Decomposition Conference, Lect. Notes Comput. Sci. Eng. 40, R. Kornhuber, R. H. W. Hoppe, J. Périaux, O. Pironneau, O. B. Widlund, and J. Xu, eds., Springer, Berlin*, pages 433–440, 2003.

A. Fournier, H.-P. Bunge, R. Hollerbach, and J.-P. Vilotte. A fourier-spectral element algorithm for thermal convection in rotating axisymmetric containers. *J. Comput. Phys.*, 204:462–489, 2005.

M. Frigo. A Fast Fourier Transform Compiler. *Proceedings of the 1999 ACM SIGPLAN Conference on Programming Language Design and Implementation*, 1999.

F. Garcia, L. Bonaventura, M. Net, and J. Sánchez. Exponential versus IMEX high-order time integrators for thermal convection in rotating spherical shells. *J. Comput. Phys.*, 264:41–54, 2014.

D.J. Gardner, J.E. Guerra, F.P. Hamon, D.R. Reynolds, P.A. Ullrich, and C.S. Woodward. Implicit-Explicit (IMEX) Runge-Kutta methods for non-hydrostatic atmospheric models. *Geosci. Model Dev.*, 11:1497–1515, 2018.

T. Gastine. pizza: an open-source pseudo-spectral code for spherical quasi-geostrophic convection. *Geophys. J. Int.*, 200:1–20, 2019.

T. Gastine, J. Wicht, and J. Aurnou. Turbulent Rayleigh-Bénard convection in spherical shells. *J. Fluid Mech.*, 778:721–764, 2015.

C.W. Gear. The simultaneous numerical solution of differential-algebraic equations. *IEEE Trans. Circuit Theory*, CT-18:89–95, 1971.

J. W. Gibbs. Fourier's Series. *Nature*, 59:606, 1899.

G. A. Glatzmaier. Numerical Simulations of Stellar Convective Dynamos. I. The Model and Method. *J. Comput. Phys.*, 55:461–484, 1984.

G. A. Glatzmaier. Geodynamo Simulations-How Realistic Are They ? *Philos. Trans. Royal Soc. A*, 247:213–278, 2002.

G. A. Glatzmaier. *Introduction to Modeling Convection in Planets and Stars.* Princeton University Press, First edition, 2014.

G. A. Glatzmaier and P. A. Gilman. *Solar Phenomena in Stars and Stellar Systems, ed. Bonnet, 145.* Dordrecht: Reidel, 1981.

G. A. Glatzmaier and P.H. Roberts. A three-dimensional self-consistent computer simulation of a geomagnetic field reversal. *Nature*, 377:203–209, 1995.

G.A. Glatzmaier and P.H. Roberts. An anelastic evolutionary geodynamo simulation driven by compositional and thermal convection. *Physica D*, 97:81–94, 1996.

I. Grooms and K. Julien. Linearly implicit methods for nonlinear PDEs with linear dispersion and dissipation. *J. Comput. Phys.*, 230:3630–3650, 2011.

I. Grooms and A. Majda. Efficient stochastic superparameterization for geophysical turbulence. *Proc. Nat. Acad. Sci.*, 110(12):4464–4469, 2013.

S. Grossmann and D. Lohse. Scaling in thermal convection: a unifying theory. *J. Fluid Mech.*, 407:27–56, 2000.

J.-L. Guermond, R. Laguerre, J. Léorat, and C. Nore. An interior penalty Galerkin method for the MHD equations in hetergenous domains. *J. Comput. Phys.*, 221:349–369, 2007.

J. L. Gustafson. Reevaluating Amdahl's law. *Commun. ACM*, 31:532–533, 1988.

E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II, Stiff and Differential-Algebraic Problems.* Springer, First edition, 2010.

E. Hairer, S.P. Nø rset, and G. Wanner. *Solving Ordinary Differential Equations I, Non-stiff Problems.* Springer, Third edition, 2008.

H. Harder and U. Hansen. A finite volume solution method thermal convection and dynamo problems in spherical shells. *Geophys. J. Int.*, 162:522–532, 2005.

T. Haut and B. Wingate. Asymptotic Parallel-in-Time Method for Highly Oscillatory PDEs. *SIAM J. Sci. Comput.*, 36(2):A693–A713, 2014.

K. Heun. Neue methode zur approximativen integration der differentialgleichungen einer unabhöngigen veränderlichen. *Zeitschr. für Math. u. Phys.*, 45:23–38, 1900.

C. Hirsch. *Numerical Computation of Internal and External Flows: The Fundamentals of Computational Fluid Dynamics.* Butterworth-Heinemann, Second edition, 2007.

R. Hollerbach. A spectral solution of the magneto-convection equations in spherical geometry. *Int. J. Numer. Methods Fluids*, 32:773–797, 2000.

D. Hupp, P. Arbenz, and D. Obrist. A parallel Navier-Stokes solver using spectral discretization in time. *Int. J. Comput. Fluid Dyn.*, 30:489–494, 2016.

H. Johnston and C. Doering. Comparison of turbulent thermal convection between conditions of constant temperature and constant flux. *Phys. Rev. Lett.*, 102:064501, 2009.

C. A. Jones. *Treatise on Geophysics 2nd Edition, Volume 8.* Elsevier, First edition, 2015.

C.A. Kennedy and M.H. Carpenter. Additive Runge-Kutta schemes for convection-diffusion-reaction equations. *Appl. Numer. Math.*, 44:139–181, 2003.

W. Kuang and J. Bloxham. Numerical modeling of magnetohydrodynamic convection in a rapidly rotating spherical shell: weak and strong field dynamo action. *J. Comput. Phys.*, 153:51–81, 1999.

W. Kutta. Beitrag zur näherungsweisen integration totaler differentialgleichungen. *Zeitschr. für Math. u. Phys.*, 46:435–453, 1901.

J.D. Lambert. *Numerical Methods for Ordinary Differential Systems*. Wiley, First edition, 1991.

M. Landeau and J. Aubert. Equatorially asymmetric convection inducing a hemispherical magnetic field in rotating spheres and implications for the past martian dynamo. *Phys. Earth. Plan. Int.*, 185:61–73, 2011.

J.-L. Lions, Y. Maday, and G. Turinici. A "parareal" in time discretization of PDE's. *C. R. Acad. Sci. Paris Sér. I Math.*, 332:661–668, 2001.

Hongyu Liu and Jun Zou. Some new additive Runge-Kutta methods and their applications. *J. Comput. Appl. Math.*, 190:74–98, 2006.

P. W. Livermore. An implementation of the exponential time differencing scheme to the magnetohydrodynamic equations in a spherical shell. *J. Comput. Phys.*, 220:824–838, 2007.

P. Marti, M. A. Calkins, and K. Julien. A computationally efficient spectral method for modeling core dynamics. *Geochem. Geophys. Geosyst.*, 17, 2016.

H. Matsui and H. Okuda. Development of a simulation code for MHD dynamo processes using the GeoFEM platform. *Int. J. Comput. Fluid Dyn.*, 18:323–332, 2004.

H. Matsui, E. King, and B. A. Buffet. Multi-scale convection in a geodynamo simulation with uniform heat flux along the outer boundary. *Geochem. Geophys. Geosyst.*, 15: 3212–3225, 2014.

H. Matsui, E. Heien, J. Aubert, J. M. Arnou, and M. Avery. Performance benchmarks for a next generation numerical dynamo model. *Geochem. Geophys. Geosyst.*, 17:1586–1607, 2016.

Message Passing Interface Forum. MPI: A Message-Passing Interface standard. *Int. J. High Perform. Comput. Appl.*, 8, 1994.

I. Newton. *Methodus differentialis (Analysis per quantitatum, series, fluxiones, ac differentias: cum enumeratione linearum tertii ordinis)*. London, 1711.

S. A. Orzag. Spectral methods for problems in complex geometries. *J. Comput. Phys.*, 37:70–92, 1980.

S.-H. Park and T.-Y. Lee. Higher-Order Time-Integration Schemes with Explicit Time-Splitting Methods. *Mon. Wea. Rev.*, 137:4047–4060, 2009.

K. Petschel, S. Stellmach, M. Wilczek, J. Lülff, and U. Hansen. Kinetic energy transport in Rayleigh-Bénard convection. *J. Fluid Mech.*, 773:395–417, 2015.

Roger Peyret. *Spectral Methods for Incompressible Viscous Flow*. Springer, First edition, 2002.

E. Plaut and F. H. Busse. Low-Prandtl-number convection in a rotating cylindrical annulus. *J. Fluid. Mech.*, 464:345–363, 2002.

P.H. Roberts and E. King. On the genesis of the earth's magnetism. *Rep. Prog. Phys.*, 76, 2013.

C. Runge. Ueber die numerische auflöung von differentialgleichungen. *Math. Ann.*, 46: 167–178, 1895.

D. Samaddar, D. E. Newman, and R. Sánchez. Parallelization in time of numerical simulations of fully-developed plasma turbulence using the parareal algorithm. *J. Comput. Phys.*, 229:6558–6573, 2010.

H. Samuel. Time domain parallelization for computational geodynamics. *Geochem. Geophys. Geosyst.*, 13:Q01003, 2011.

Y. Sasaki, S. Takehiro, Y.-Y. Hayashi, and SPMODEL Development Group. *Project of MHD Dynamo in Rotating Spheres and Spherical Shells*. Kyoto, Japan. [Available at http://www.gfd-dennou.org/library/dynamo/.], 2012.

N. Schaeffer. Efficient spherical harmonic transforms aimed at pseudospectral numerical simulations. *Geochem. Geophys. Geosyst.*, 14:751–758, 2013.

N. Schaeffer, D. Jault, H.-C. Nataf, and A. Fournier. Turbulent geodynamo simulations: a leap towards earth's core. *Geophys. J. Int.*, 211:1–29, 2017.

J. D. Scheel and J. Schumacher. Local boundary layer scales in turbulent Rayleigh-Bénard convection. *J. Fluid Mech.*, 758:344–373, 2014.

R. D. Simitev and F. H Busse. Prandtl number dependence of convection driven dynamos in rotating spherical fluid shells. *J. Fluid Mech.*, 532:365–388, 2005.

P.R. Spalart, R.D. Moser, and M.M. Rogers. Spectral methods for the Navier-Stokes equations with one infinite and two periodic directions. *J. Comp. Phys.*, 96:297–324, 1991.

M.N. Spijker. Stiffness in numerical initial-value problems. *J. Comput. Appl. Math*, 72: 393–406, 1996.

S. Stellmach and U. Hansen. An efficient spectral method for the simulation of dynamos in Cartesian geometry and its implementation on massively parallel computers. *Geochem. Geophys. Geosys.*, 9:Q05003, 2008.

F. Takahashi. Implementation of a high-order combined compact difference scheme in problems of thermally driven convection and dynamo in rotating spherical shells. *Geophys. Astrophys. Fluid Dyn.*, 106:231–249, 2012.

D.J. Tritton. *Physical Fluid Dynamics*. Oxford University Press, Second edition, 1988.

J. Verhoeven, T. Wiesehöfer, and S. Stellmach. Anelastic versus fully compressible turbulent rayleigh-bénard convection. *Astrophys. J.*, 805:62, 2015.

D. Wang and S. Ruuth. Variable step-size implicit-explicit linear multistep methods for time-dependent partial differential equations. *J. Comput. Math*, 26:838–855, 2008.

J. L. Wicker and W. C. Skamarock. Time-splitting Methods for Elastic Models Using Forward Time Schemes. *Mon. Wea. Rev.*, 130:2088–2097, 2002.

A. P. Willis, B. Sreenivasan, and D. Gubbins. Thermal core–mantle interaction: Exploring regimes for 'locked' dynamo action. *Phys. Earth Planet. Inter.*, 165:83–92, 2007.

H. Zhang, A. Sandu, and S. Blaise. Partitioned and implicit explicit general linear methods for ordinary differential equations. *Journal of Scientific Computing*, pages 1–26, 2014.

X. Zhu, V. Mathai, R. J. A. M. Stevens, R. Verzicco, and D. Lohse. Transition to the Ultimate Regime in Two-Dimensional Rayleigh-Bénard Convection. *Phys. Rev. Lett.*, 120:144502, 2018.

D. Zwillinger. *CRC Standard Mathematical Tables and Formulas*. Springer, Thirty third edition, 2018.